Predicting Spatial Self-Organization with Statistical Moments

Linge Bai*, Robert Gilmore† and David E. Breen*
*College of Computing and Informatics †Department of Physics
Drexel University, Philadelphia, PA 19104
Email: {lb353,david}@cs.drexel.edu, robert.gilmore@drexel.edu

***Abstract*—We have developed a self-organizing shape formation system based on locally interacting agents whose behaviors are inspired by living cells. Given a predefined macroscopic shape, genetic programming is used to find a finite field function that defines the agents' interactions. By following the gradient of the cumulative field the agents form into a desired shape. It has been seen that the self-organization process may form two or more stable final configurations. In order to control the outcome of the shape formation process, it is first necessary to accurately predict the outcome of the dynamic simulation. This paper describes an approach to predicting the final configurations produced by our spatial self-organization system at an early stage in the process. The approach calculates statistical moments of the coordinates of the agents, and employs Support Vector Machines to predict the final shape of the agent swarm based on the moments and their time derivatives.**

I. Introduction

Motivated by the ability of living cells to form into specific shapes and structures, we have developed chemotaxis-inspired software agents for self-organizing shape formation [1], [2]. The actions of the agents, which we call Morphogenetic Primitives (MPs), are based on the behaviors exhibited by living cells. Cells emit chemicals into the environment. Neighboring cells detect the overall chemical concentration at their surfaces and respond to the chemical stimulus by moving along the chemical field's gradients [3]. Similarly, in our system the agents emit a virtual chemical, with its concentration defined by an explicit mathematical expression. All agents start with an initial random configuration and stochastically follow the gradient of the cumulative concentration field. These chemotaxis-based local interactions can direct the agents to self-organize into user-specified shapes (Figure 1), a capability that could be used to control a robotic swarm.

In some cases, we have observed that the agents do not spatially self-organize into a unique shape, but instead form two or more stable final configurations. It would be useful to control the outcome of these divergent spatial self-organization processes. This would allow us to guarantee that all of our MP simulations would produce a single, desired shape. The first step toward providing the guarantee involves predicting, at an early stage of the evolution, which final shape will be formed. This will then make it possible to apply changes that direct the agent swarm towards the correct spatial configuration. Towards this end, we have discovered features, based on statistical moments and their derivatives of the agents' positions, that can be computed early in the swarm's evolution and may be used to predict the final shape of the self-organizing system. Analyzing these features with a Support Vector Machine [4], a supervised machine learning technique, classifies the agent system, i.e. provides a prediction for the system's final configuration. Given a prediction that a swarm is aggregating into an "undesired" configuration, it may be possible to perturb the positions of the agents to redirect their shape evolution. We aim to design biased perturbations that can change the path of the evolution towards a "desired" shape.

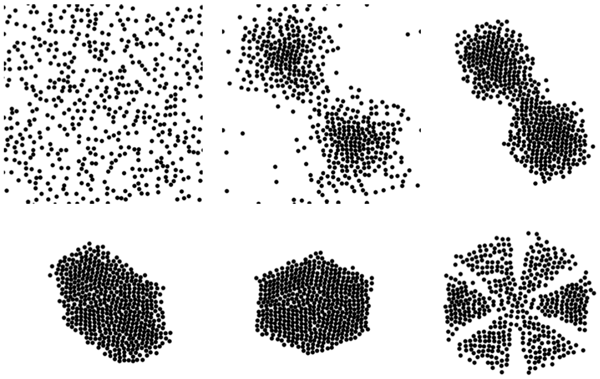
Fig. 1. MPs self-organizing into a pinwheel shape.

II. Related Work

Multi-agent approaches have been widely utilized to model self-organizing systems [5], [6], [7]. Our current work focuses on how to analyze and predict the outcome of such systems. To date, a variety of methods have been used to analyze these systems. These methods usually require identifying global variables [8] and/or deriving quantitative properties of the whole system [9]. Differential equations also are used to capture state transitions of a system [10], [11]. These methods require identifying global variables and the relationships between them, which can be obscure and quite complex.

Various analytical methods have been used to predict the outcomes of self-organizing systems. Time series analysis has been applied to predict self-organizing maps [12]. The analysis aims to predict long-term trends, rather than predicting specific final states. Hamann [13] presents a data-driven, iterative process, based on a swarm urn model, Markov chains and abductive reasoning, for determining, testing, and refining hypotheses about how self-organizing decision-making system operate. The process is able to identify influential feedback loops and important subsections of the configuration space in these types of systems.

Machine learning algorithms have also been applied to self-organizing systems. An unsupervised learning algorithm is utilized to construct a self-organizing map for information retrieval [14]. Support Vector Machines have been used together with self-organizing maps for shape-based image retrieval [15].

Statistical moments are measures that quantify the shape of distributions. In statistics, the method of moments is used to estimate population parameters [16]. In our work, we gain insights from the moments of the coordinates of moving agents in a $2D$ arena and predict the final outcome of the system based on the moment information.

III. Self-Organizing Shape Formation

Our previous work [2], [17] is inspired by developmental biology [18] and morphogenesis [19] and defines software agents with behaviors similar to living cells. This work builds upon a chemotaxis-based cell aggregation simulation system [20]. Morphogenesis is the process that develops the shape or structure of an organism through cell shape change, movement, attachment, growth and death. The motions induced by chemotaxis (one of the mechanisms of morphogenesis) may produce patterns or sorting of cells [21]. These natural phenomena provide the proof-of-concept that chemotaxis-based cell aggregation offers a promising approach to guiding spatial self-organization processes.

A. Morphogenetic Primitives

In our system, virtual cells (agents) are initially placed inside a $2D$ environment with a random uniform distribution, and interact with each other via a chemotaxis paradigm. This interaction produces movements that lead the agents to aggregate into a single user-specified shape. We call these self-organizing agents Morphogenetic Primitives (MPs).

Each MP is represented by a small disc and emits a 'chemical' into the environment within a fixed distance. Every MP emits the identical local chemical field. An MP detects the cumulative chemical field at eight receptors on its surface, and calculates the field gradient from this input. MPs move in the direction of the field gradient with a speed proportional to the magnitude of the gradient. By employing these relatively simple chemotaxis-inspired behaviors MPs are able to self-organize into specific macroscopic shapes.

B. Local Interactions

While MPs' fundamental interactions are based on a chemotaxis-inspired paradigm, we do not limit their behaviors/properties to be physically realistic or completely consistent with biology. Instead, developmental biology provides a motivating starting point for MPs. As a way to customize chemotaxis-inspired agents for shape formation, we alter the chemical concentration fields around individual cells. Instead of the chemical concentration dropping off as a function of distance, e.g. $1/r$, in our system we define the concentration field with an explicit function of cell-cell distance d, one cell's angular location θ in another cell's local coordinate system and simulation time t.

Currently, there is no prescriptive way to specify a particular local field function that will direct MPs to form a specific macroscopic shape, we therefore employ genetic programming [22] to produce the mathematical expression that explicitly specifies the field function. In order to meet the substantial computational requirement imposed by our evolutionary computing approach, we have implemented a master-slave form of the distributed genetic programming process [1]. The fitness measure associated with each individual field function is based on the shape that emerges from the chemical-field-driven aggregation simulation, and determines which functions will be passed along to later generations. The genetic process stops once an individual (i.e., a mathematical expression) in the population produces the desired shape via a chemotaxis simulation, or after a certain number of generations have been produced and evaluated.

With this algorithm, we have successfully evolved local MP chemical field functions for a number of simple shapes. These results support the proposition that biological phenomena offer paradigms for designing cellular primitives for self-organizing shape formation. Additionally, evolutionary computing techniques, specifically genetic programming, have been crucial for discovering the detailed local interactions that lead to the emergence of macroscopic shapes and structures.

However, given the MPs' initial random configurations and the stochastic nature of the self-organization process, the outcomes of the simulations with a specific field function are not always the same. We have found that the shape formation simulations can generate bifurcating results. For some field functions, if we run numerous simulations each starting with a different random distribution of MPs, two sets of final configurations will be formed. In most cases an equal number of each configuration is formed, but in a few cases the ratio of the numbers is not one. Since it would be useful to control the outcomes of the self-organization process, we are developing methods for predicting the final configuration of a bifurcating simulation at an early stage of the process.

IV. Moment Analysis

In order to predict the final outcome of a self-organizing shape formation simulation, we first extract features that capture the spatial distribution of the MPs. Moments provide a quantitative way to describe a distribution. Since MPs are defined as small discs, we use the center of each disc to represent each MP's location. We therefore can simplify the collection of MP locations as a set of $2D$ points, and apply moment analysis to this set over the duration of the MP simulation.

A. Moment Calculation

We calculate the mean (first moment), variance (second central moment), skewness (third central moment) and kurtosis (fourth central moment) from the x and y coordinates of the MP centers. We analyze the locations X_i of all points (MPs) as a whole, rather than tracking the location and movement of each individual point. The population size of the agents is denoted as n, $(n = 500)$, and the formulas of the four moments M_1 to M_4 are given in Equations 1 to 4,

$$M_1 = \frac{1}{n}\sum_{i=1}^{n} X_i, \tag{1}$$

$$M_2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - M_1)^2, \tag{2}$$

$$M_3 = [\frac{1}{n}\sum_{i=1}^{n}(X_i - M_1)^3]/(M_2)^{3/2}, \tag{3}$$

$$M_4 = [\frac{1}{n}\sum_{i=1}^{n}(X_i - M_1)^4]/(M_2)^2. \tag{4}$$

These statistical moments provide quantitative information about the shape of histograms/distributions. When computed for the x and y coordinates of the MPs, these moments capture the symmetry and asymmetry of the spatial distribution of the whole population. Since the x and y coordinates of the points change over time, so do the four moments of the distribution of the x and y values. The change of the moments as a function of simulation time also provides insight into the dynamic nature of a particular MP simulation.

We define the time derivative of the moments as the slope of a linear interpolating function of consecutive moment values. A series of values are obtained by sampling the simulation time with a fixed interval τ. At each simulation time t, the four moments $M_i(t)$ $(i = 1\ to\ 4)$ of the overall distribution are calculated. For each moment M_i, we consider an interval of 4τ which containing 5 samples, that is, $M_i(t-2\tau)$, $M_i(t-\tau)$, $M_i(t)$, $M_i(t+\tau)$ and $M_i(t+2\tau)$. A straight line is fit to the five points with a least squares method. For each moment M_i at time t, we find the k_i and b_i that minimize Equation 5,

$$min \sum_{j=-2}^{2} |k_i \cdot (t + j\tau) + b_i - M_i(t + j\tau)|^2. \tag{5}$$

The slope k_i of the fitted line is then used as an approximation of the time derivative of $M_i(t)(i = 1\ to\ 4)$.

By calculating the moments and their time derivatives for both the x and y coordinates of the point set, at a given time t, we obtain a 16-dimensional vector to represent the distribution,

$$\begin{aligned}[M_{x_1}(t), M_{y_1}(t), M_{x_2}(t), M_{y_2}(t),\\ M_{x_3}(t), M_{y_3}(t), M_{x_4}(t), M_{y_4}(t),\\ k_{x_1}(t), k_{y_1}(t), k_{x_2}(t), k_{y_2}(t),\\ k_{x_3}(t), k_{y_3}(t), k_{x_4}(t), k_{y_4}(t)]\end{aligned} \tag{6}$$

Our simulations consist of $(n = 500)$ $2D$ points and therefore have a configuration space of dimension $2n$. By distilling an MP spatial distribution down to this feature vector, we reduce the dimensionality of the simulation from 1000 to 16.

Given the sensitivity of non-linear dynamical systems to initial conditions [23], it makes it extremely difficult, if not impossible, to predict the outcome of our complex, self-organizing system from its initial, random spatial configuration. We therefore attempt to predict the final spatial configuration at an early stage of the aggregation, usually before it is visually evident what shape will emerge from the process. We have found that applying our analysis at a time in the simulation that is a small percentage of the total time needed for the final aggregated shape to form (e.g. 5% to 10%) produces acceptable results (81% to 91% accuracy).

B. Using Support Vector Machines

We consider prediction of the bifurcating outcomes as a classification problem and utilize support vector machines (SVMs) [4] to solve it. An SVM applies kernel methods to labeled training data and transforms the data into a high dimensional space. It computes the optimal separating hyperplane, which has the largest margin between the two classes. Doing so produces a trained SVM model with kernel and constraint parameters. Classification is performed by mapping new data into the same high dimensional space and determining on which side of the hyperplane the new data lie.

Since our data are not linearly separable, we apply a Radial Basis Function (RBF) kernel in our SVM. We use LIBSVM [24] to perform SVM training and testing. To train the SVM, we employ a leave-one-out cross-validation method. We use grid search to find the best cost parameter C for the SVM and γ parameter for the RBF kernel. Once the optimal C and γ values were identified, accuracy statistics were gathered from numerous simulations, each utilizing different random initial conditions.

To obtain training data for a specific target shape, we run a certain number (200) of MP simulations starting with different, random spatial configurations. Each simulation consists of a fixed number of time steps $(10,000$ to $35,000)$, depending on how long it takes for the target shape to form. We save the spatial distribution of the agents every 100 steps of the simulation, that is, $\tau = 100$. Each simulation will produce an aggregated structure at the end of the simulation, and we manually label all final structures with one of two categories.

V. Results

Our system is able to produce a large number of interesting and stable shapes. However, it occasionally (∼8% of interaction functions) will produce two or more different shapes from the same agent interaction function. In these cases we divide the results into two “bifurcated” groups, one group with the dominant, i.e. the most prevalent, configuration, and the second ”other” group with the remaining less numerous configurations. Given this grouping scheme, we have applied the proposed prediction method to a number of bifurcating shape evolutions. The target shapes are a quarter-moon, an

Quarter-moon				
Total Time	Prediction Time	Positive Instances	Negative Instances	Kernel
35,000	3,500	100	100	RBF
Cost (C)	gamma (γ)	Accuracy	Sensitivity	Specificity
512	0.03125	91.5%	92.0%	91.0%

Ellipse				
Total Time	Prediction Time	Positive Instances	Negative Instances	Kernel
10,000	500	100	100	RBF
Cost (C)	gamma (γ)	Accuracy	Sensitivity	Specificity
2048	4.883e-04	89.0%	89.0%	89.0%

Four Discs				
Total Time	Prediction Time	Positive Instances	Negative Instances	Kernel
15,000	1,500	100	100	RBF
Cost (C)	gamma (γ)	Accuracy	Sensitivity	Specificity
512	1.221e-04	83.0%	84.0%	82.0%

Parallel Line Segments				
Total Time	Prediction Time	Positive Instances	Negative Instances	Kernel
10,000	1,000	508	100	RBF
Cost (C)	gamma (γ)	Accuracy	Sensitivity	Specificity
4	0.125	(80.9 ± 2.5)%	(80.4 ± 3.5)%	(81.4 ± 2.6)%

TABLE I
DATASET INFORMATION AND PREDICTION ACCURACIES FOR BIFURCATING SHAPES AT 10% (5% FOR ELLIPSE) OF SIMULATION TIME.

ellipse, a set of four discs, and a set of two parallel line segments. Once the SVM parameters with the best overall accuracy are determined through grid search and leave-one-out cross-validation, we calculate overall accuracy, sensitivity and specificity for each example. These results are summarized, along with the dataset descriptions and SVM parameters, in Table I. The quarter-moon, ellipse and four-discs datasets each contains 200 simulations, with 100 belonging to each class. The parallel line segments dataset is unbalanced, with 508 instances belonging to one class and 100 belonging to the other. We have seen that the SVM approach yields higher accuracies when the bifurcation is balanced.

The quarter-moon dataset contains 100 left-facing structures and 100 right-facing ones. The quarter-moon interaction function did produce a few shapes of a third type (a diagonal shape) ($\sim$9% of simulation runs), but these were not included in this study. Instead we only used the balanced left- and right-facing outcomes. A typical shape evolution for this dataset is shown in Figure 2. The simulation reaches a stable state by $35,000$ simulation steps. We extract feature vectors in the form of Vector 6. At step $3,500$ (10% of the total simulation time) we are able to predict the left-facing or right-facing outcome with an overall accuracy of 91.5%.

The ellipse dataset contains 100 single ellipses and 100 non-single ellipses. The non-single ellipse group included both two ellipses and single malformed structures. Figure 3 demonstrates a typical ellipse shape evolution. The simulation reaches a stable state by $10,000$ steps. At 500 steps (5% of simulation time) we are able to predict whether the simulation will produce a single ellipse or not with an overall accuracy of 89%.

The four-discs dataset contains 100 four discs structures and 100 structures not containing four discs, with a typical shape evolution shown in Figure 4. The second group contains a variety of disc numbers, including 3, 5, 6 and 7, with the most prevalent number being 5. The simulation reaches a stable state by $15,000$ steps. At $1,500$ steps (10% of simulation time) we are able to predict whether the simulation will form four discs or not with an overall accuracy of 83%.

The parallel line dataset contains 508 instances of two vertical parallel line segments and 100 instances of either ”Y-shape” structures or one vertical line, as seen in Figure 5. The simulation reaches a stable state by $10,000$ steps. We calculate the prediction of the final configuration at step $1,000$. Since this is an unbalanced dataset, we under-sample the majority set. We randomly choose 100 out of the 508 two-line-segments instances and merge them with the minority set. Leave-one-out cross validation is then performed on the newly constructed balanced dataset. We achieve an average overall accuracy of 80.9 ± 2.5% over 100 randomly constructed balanced datasets when predicting the formation of two vertical lines.

Since the overall line-segment results are produced via averaging the results from numerous randomly generated partitions, we include a standard deviation with the average. The

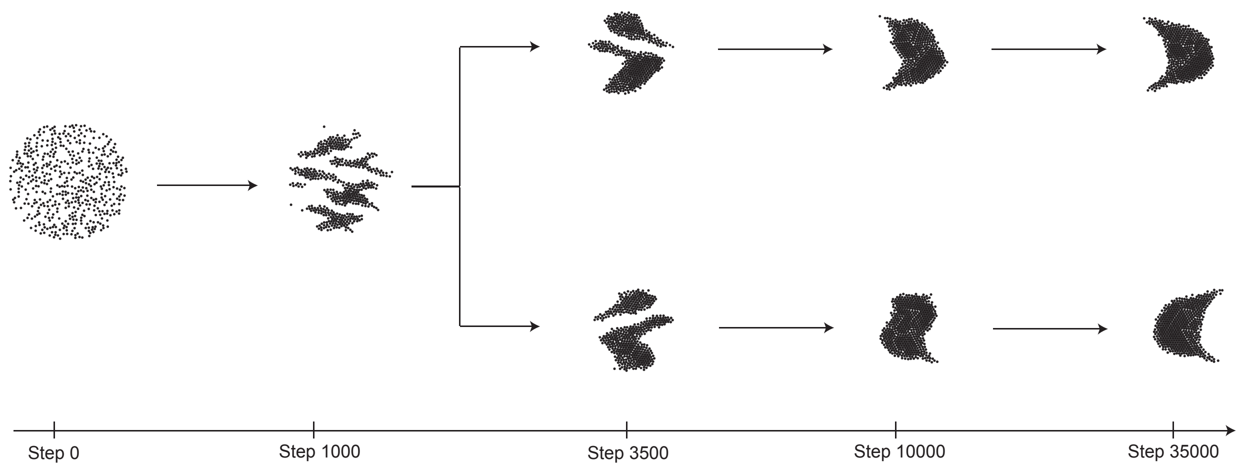

Fig. 2. Shape evolution of the quarter-moon.

other examples have balanced datasets, and therefore leave-one-out cross-validation produces a unique partitioning and no deviations.

VI. Conclusion and Future Work

Based on the statistical moments of agents' positions and the time derivatives of these moments, we are able to predict the final outcome of a spatial self-organization process at 5% or 10% percent of the simulation time with an overall accuracy of 81% to 91%. Now that we can predict the outcome of this process at an early stage, we are developing algorithms that will control or influence the agents' interactions so that they consistently form into one stable configuration.

Acknowledgment: The authors would like to thank Santiago Ontañón and Christian Kuehn for many helpful discussions. This work was funded by National Science Foundation grants CCF-0636323 and IIS-0845415.

References

[1] L. Bai, M. Eyiyurekli, and D. Breen, "Automated shape composition based on cell biology and distributed genetic programming," in *Proc. Genetic and Evolutionary Computation Conference*, 2008, pp. 1179–1186.

[2] L. Bai and D. E. Breen, "Chemotaxis-inspired cellular primitives for self-organizing shape formation," in *Morphogenetic Engineering*, R. Doursat, H. Sayama, and O. Michel, Eds. Berlin: Springer, 2012, pp. 209–237.

[3] E. Eisenbach *et al.*, *Chemotaxis*. London: Imperial College Press, 2004.

[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[5] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Computer Graphics (Proc. ACM SIGGRAPH)*, vol. 21, no. 4, 1987, pp. 25–34.

[6] G. Serugendo, A. Karageorgos, O. Rana, and F. Zambonelli, Eds., *Engineering Self-Organising Systems*. Berlin: Springer, 2004.

[7] G. Weiss, Ed., *Multiagent Systems, 2nd Edition*. Cambridge, MA: MIT Press, 2013.

[8] I. G. Kevrekidis, C. W. Gear, and G. Hummer, "Equation-free: The computer-aided analysis of complex multiscale systems," *AIChE Journal*, vol. 50, no. 7, pp. 1346–1355, 2004.

[9] J. Sudeikat and W. Renz, "A systemic approach to the validation of self–organizing dynamics within MAS," in *Agent-Oriented Software Engineering IX*. Springer, 2009, pp. 31–45.

[10] G. Ferrari-Trecate, A. Buffa, and M. Gati, "Analysis of coordination in multi-agent systems through partial difference equations," *IEEE Trans. on Automatic Control*, vol. 51, no. 6, pp. 1058–1063, 2006.

[11] L. Cardelli, "On process rate semantics," *Theoretical Computer Science*, vol. 391, no. 3, pp. 190–215, 2008.

[12] G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort, and M. Verleysenothers, "Time series forecasting: Obtaining long term trends with self-organizing maps," *Pattern Recognition Letters*, vol. 26, no. 12, pp. 1795–1808, 2005.

[13] H. Hamann, "A reductionist approach to hypothesis-catching for the analysis of self-organizing decision-making systems," in *Proc. IEEE Conference on Self-Adaptive and Self-Organizing Systems*, 2013, pp. 227–236.

[14] X. Lin, D. Soergel, and G. Marchionini, "A self-organizing semantic map for information retrieval," in *Proc. ACM Conference on Research and Development in Information Retrieval*, 1991, pp. 262–269.

[15] W.-T. Wong, F. Y. Shih, and J. Liu, "Shape-based image retrieval using support vector machines, fourier descriptors and self-organizing maps," *Information Sciences*, vol. 177, no. 8, pp. 1878–1891, 2007.

[16] J. M. Wooldridge, "Applications of generalized method of moments estimation," *The Journal of Economic Perspectives*, vol. 15, no. 4, pp. 87–100, 2001.

[17] L. Bai, M. Eyiyurekli, P. Lelkes, and D. Breen, "Self-organized sorting of heterotypic agents via a chemotaxis paradigm," *Science of Computer Programming*, vol. 78, no. 5, pp. 594–611, 2013.

[18] S. Gilbert, *Developmental Biology*, 8th ed. Sunderland, MA: Sinauer Associates, Inc., 2006.

[19] J. Davies, *Mechanisms of Morphogenesis: The Creation of Biological Form*. Amsterdam: Elsevier, 2005.

[20] M. Eyiyurekli, P. Manley, P. Lelkes, and D. Breen, "A computational model of chemotaxis-based cell aggregation," *BioSystems*, vol. 93, no. 3, pp. 226–239, September 2008.

[21] T. Sekimura, S. Noji, N. Ueno, and P. Maini, Eds., *Morphogenesis and Pattern Formation in Biological Systems*. Tokyo: Springer, 2003.

[22] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[23] S. Wiggins, Ed., *Introduction to Applied Nonlinear Dynamical Systems and Chaos, 2nd Edition*. New York: Springer, 2003.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

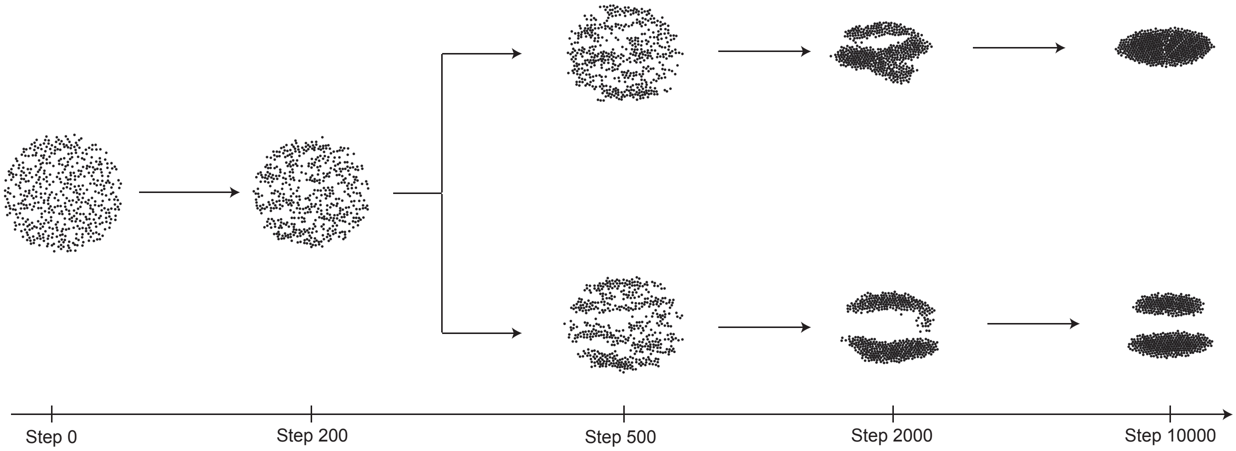

Fig. 3. Shape evolution of the ellipse.

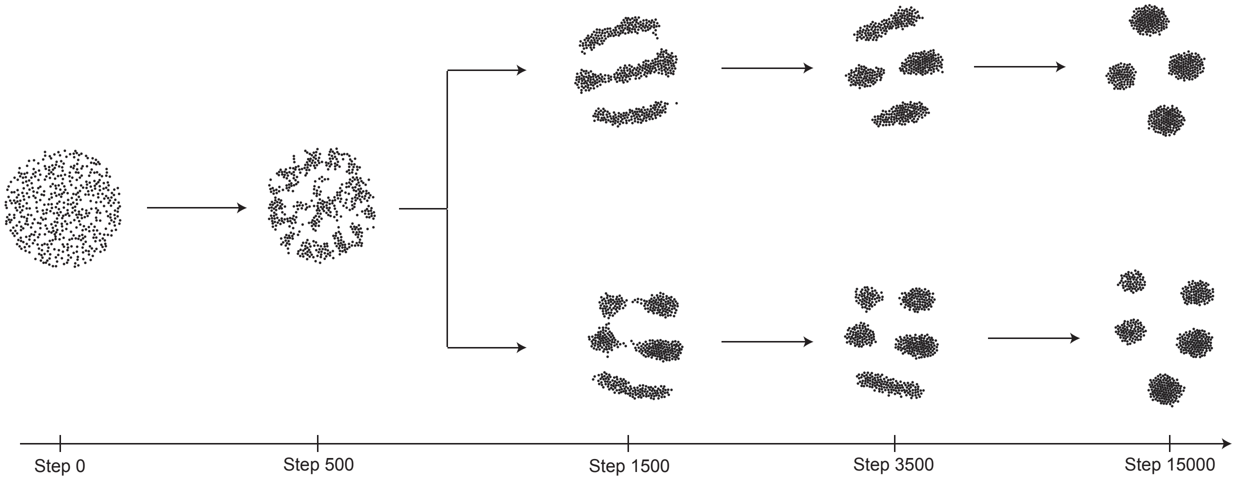

Fig. 4. Shape evolution of the discs structure.

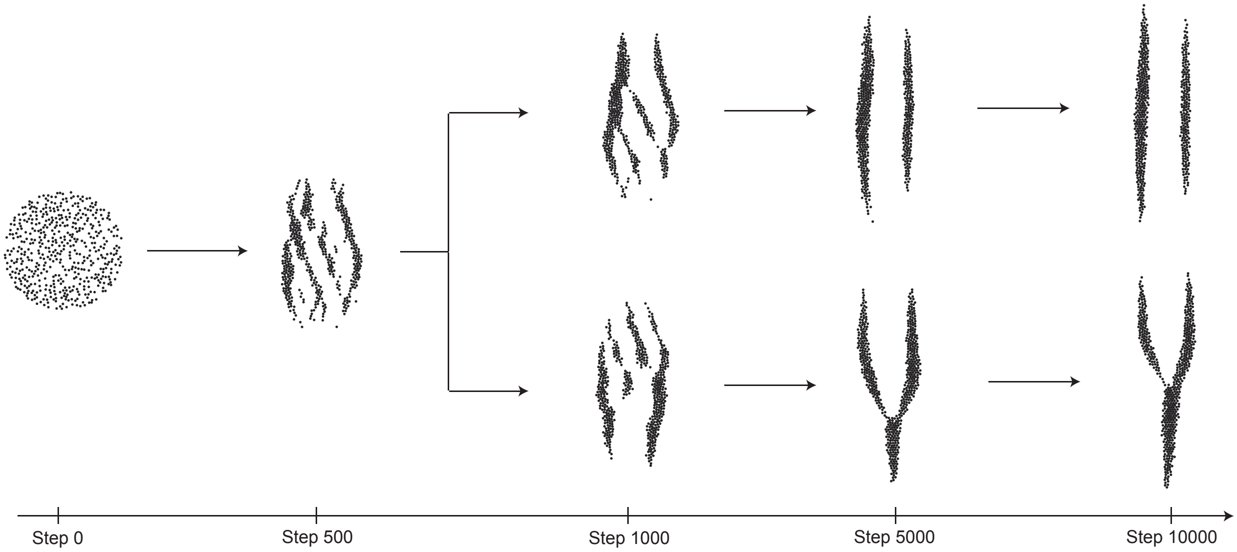

Fig. 5. Shape evolution of the line segments structure.