Discovering Grid-Cell Models Through Evolutionary Computation

Lin Wang
Shandong Provincial Key Laboratory of Network Based Intelligent Computing
University of Jinan
Jinan, 250022, China
Email: ise_wanglin@ujn.edu.cn

Bo Yang*
Shandong Provincial Key Laboratory of Network Based Intelligent Computing
University of Jinan
Jinan, 250022, China
Email: yangbo@ujn.edu.cn

Jeff Orchard
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
Email: jorchard@uwaterloo.ca

***Abstract*—One of the main tasks in neuroscience research is to interpret the activity of neurons. Given some neuroscientific data, such as spike trains, one tries to decipher how the activity of the neurons relate to the outside world and/or the behaviour of the animal. The discovery of place cells and grid cells are great examples – discoveries that garnered a Nobel Prize in 2014. However, the spatial patterns exhibited by such cells are only the beginning of our understanding of spatial representation in the brain. In this paper, we apply an evolutionary algorithm to discover spatial patterns exhibited in cells from the entorhinal cortex to see (1) if we can automatically deduce an accurate model for the hexagonal-grid pattern, and (2) if we can discover a more general model that also incorporates grid-cell-like variants that have been observed, but not understood.**

***Index Terms*—Grid Cell; Genetic Expression Programming; Particle Swarm Optimization; Neuroscience**

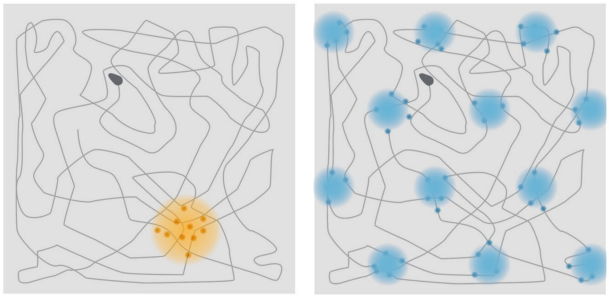
I. Introduction

The 2014 Nobel Prize for Physiology or Medicine went jointly to John O'Keefe, May-Britt Moser, and Edvard Moser for their discoveries of neurons in the brain that seem to encode an animal's location in space. John O'Keefe discovered cells in the rat hippocampus that start firing action potentials (spikes) only when the animal is in a particular region of a room [1]. There are many such "place cells", and each has a different receptive field (region where it becomes active). This startling discovery showed that some neural representations of space and location can be very simple. Figure 1(a) illustrates the activity of a place cell.

Later, O'Keefe uncovered a phase relationship between place-cell spikes and the theta cycle, an internal clock that operates at between 4 and 12 Hz in the rat [2].

Subsequently, other cells were found that exhibit spatially regular receptive fields. The lab of May-Britt and Edvard Moser discovered cells in the entorhinal cortex (which projects to the hippocampus) that fire bursts of spikes at an array of locations that seem to be arrange in a hexagonal grid [4] (Fig. 1(b)). However, many other cells in the entorhinal cortex exhibit firing patterns that are hexagon-like, but vary in a number of ways.

In this paper, we bring the power of evolutionary computing to this fascinating problem, to search for a parsimonious model for the spatial pattern of activity exhibited by entorhinal grid cells.



(a) Place Cell (b) Grid Cell

Fig. 1. Activation maps of (a) place cell, and (b) grid cell (adapted from [3]).

The rest of the paper is organized as follows. Section II reviews grid-cell models and evolutionary computation methods for reverse engineering. Then, the details of our methodology for discovering grid-cell models are described in Section III. Section IV provides details about our experiments, and discussion followed by conclusions in Section V.

II. Related Works

Even before the 2014 Nobel Prizes were awarded, researchers had been coming up with models to explain how grid-cell patterns and place-cell patterns emerged from neurons. At the time of this writing, there seem to be two main camps: attractor network models, and oscillator interference models. In attractor network (AN) models, neurons are imagined in a 2-D sheet, each connected to neurons in its neighbourhood. The efficacy of these connections follow the "local excitation / lateral inhibition" pattern. The activity of neurons in such a mesh can self-organize and exhibit many stable patterns, including a single local region of activity (like a place field), or (with different parameters) a hexagonal array of bumps [5].

The oscillator interference (OI) models are based on neural oscillators, populations of neurons that generate oscillating output. However, the frequency of oscillation varies with the velocity vector of the animal. As the animal moves around,

* Corresponding Author

various oscillators get in and out of phase with each other [6], resulting in the animal's position being coded into oscillator phase [7]. The OI explanation for grid cells is that it is the interference pattern between 3 oscillators tuned to movement in divergent directions.

In recent years, with the increase in scientific data and computing power, evolutionary computation methods have shown promise at reverse engineering models from data. Schmidt et al [8] extracted natural physical laws from observed data without help from any prior human knowledge using genetic programming. Floares [9] adopts genetic programming to discover structure and parameters automatically for drug gene regulatory networks. Based on time-series measurements, Qian [10] distilled nonlinear differential equation based gene regulatory networks using evolutionary computation. Chakraborty [11] adopts evolutionary programming to infer dynamic and static models for solid oxide fuel cells. In our previous research, we use gene expression programming to extract the hydration kinetic equations [12], [13] and image-feature based models [14] from cement hydration data.

III. Methodology

In this section, the method of distilling the model for grid cell firing pattern is described in detail. Grid cells fire whenever the animal moves to any of a set of sites distributed in hexagonal grid. The grid cell attains its maximal firing rate at the centre of each grid field. We express the firing rate model of a grid cell as

$$\lambda = f_{C_1,C_2,\cdots,C_n}(x,y), \tag{1}$$

where f represents the form of the firing rate model, and $C_1, C_2, \cdots, C_n$ represent coefficients in the model f, and (x,y) represents the animal's position in the environment. The coefficients $C_1, C_2, \cdots, C_n$ adjust the model to fit different grid cells, each exhibiting a different grid-like firing pattern. In the traditional, manually-derived models, they have different physical meaning. For example, in the interference models , the coefficients represent the weights of spatial wave vectors patterns.

The adopted method consists of two stages: the discovery of a unified form of model, and the optimization of coefficients in the discovered model to fit different systems. In the first stage, we use a hybrid evolution method consisting of Gene Expression Programming (GEP) [15] and Particle Swarm Optimization (PSO) [16]. The general form is optimized by gene expression programming. To evaluate the fitness of individuals in GEP, the particle swarm optimization is performed – but with fewer generations – to fit the potential model to each observed activation map. The final output of the first stage is a model with several automatically-discovered coefficients. At the second stage, particle swarm optimization is performed with a larger population size, more populations, and more generations, to get the best fit to the activation maps.

Fig. 2 illustrates the basic procedure of distilling a grid-cell model from the data. Firstly, the spiking train and position information are recorded while an animal is running in an environment. Then, the firing fields (activation map) is characterized using a spatial smoothing algorithm [17]. The first stage, which adopts the hybrid evolution consisting of GEP and PSO, is used to evolve the unified form f of grid-cell firing. Considering the high time-complexity of discovering a model, a CUDA-based parallel acceleration method is used on a GPU platform. After obtaining the unified model, the second stage optimizes the coefficients in the model for each activation map.

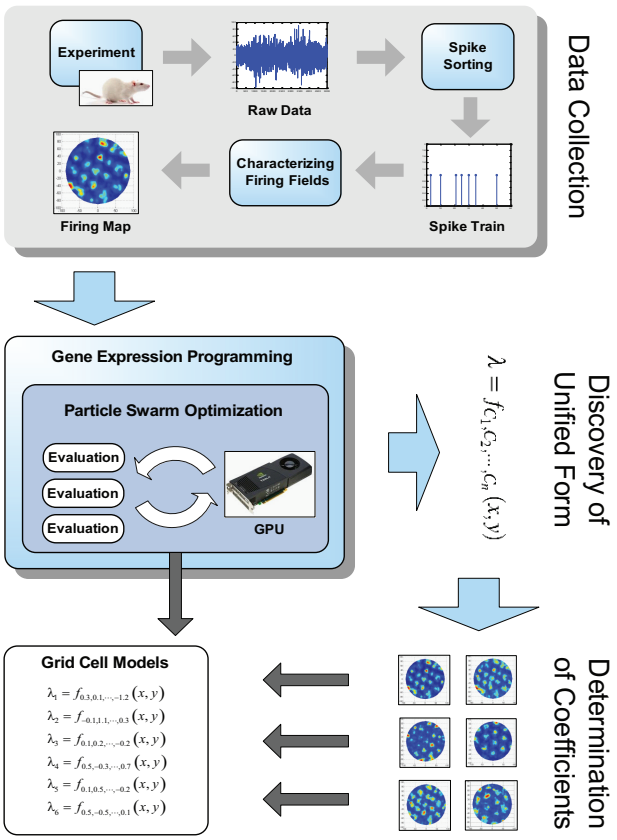


Fig. 2. The main framework of distilling grid cell firing model from observed data. It consists of data collection, discovery of unified form, and the determination of coefficients.

A. Data Collection

For our study, we used data from Hafting et al [17]. Data was collected using tetrodes from the medial entorhinal cortex of rats as they wandered around a circular arena. The rat's position was tracked simultaneously. The raw tetrode data was spike-sorted [18] to infer individual spikes from nearby neurons.

The spike trains and position traces were used to get an activation map using a spatial smoothing algorithm proposed by Hafting et al. [17]. The average firing rate at a location x in the environment is calculated by

$$\lambda(x) = \frac{\sum_{i=1}^{n} g\left(\frac{s_i - x}{h}\right)}{\int_0^T g\left(\frac{y(t)-x}{h}\right) dt} \tag{2}$$

where g is a Gaussian kernel, h is the smoothing factor and is set to 3 according to [17], n represents the total number of spikes in the experiment, s_i represents the position of ith spike, $y(t)$ represents the position of the animal at time t, and T represents the duration of the recording. A bin size of 5cm $\times$ 5cm is used to divide the map into a lattice.

B. Discovery of Unified Form

In this subsection, we describe our approach for finding a unified form of a grid-cell model. We used gene expression programming (GEP) to evolve the form of the model based on the rat data.

However, there is a key question; how to evaluate whether a possible model fits the observed data. This question is further related to how to find the best combination of coefficients for a given model to fit the data. We used particle swarm optimization (PSO) because of its lower requirement for population size, and better convergence speed. It is embedded into the evaluation step of the GEP and performed for all possible solutions. In order to reduce the running time, a GPU platform is used to speed up evaluations in parallel. In the following subsections, some key issues within this stage are described in detail.

1) Basic Methods: Gene Expression Programming [15] is a variant of genetic programming [19], which optimizes a program and function using operations on a tree structure. GEP combines the advantages of genetic algorithms (linear encoding) and genetic programming (tree structure optimization) [20]. It evolves the complex program by operating on simple linear sequences efficiently. Its chromosomes consists of several genes, which are k-expression based linear sequences and can be connected by a link function. The tree structure is obtained by decoding the genes, traversing from top to bottom and from left to right. GEP chromosomes select symbols from a function set F, consisting of operators, and a terminal set T, consisting of constants, variables, and coefficients. Each gene in a chromosome can be further divided into the head part and tail part. This division is used to maintain the completeness of the tree structure, selecting components from both sets for the head part and selecting components from the terminal set for the tail part. Fig. 3 shows an example GEP chromosome, and the corresponding expression tree and algebraic form for a grid cell model.

Particle Swarm Optimization [16] is a global numerical optimization method inspired by the foraging movement of animals. Each solution is encoded as a particle that flies over the solution space. In this research, the particle stands for a vector of coefficients for an evaluated model. Each particle changes its velocity and position by following its personal historical best solution and the global historical best solution. In each generation, each particle's velocity and position are updated using

$$v^i = v^i + \varphi_1 r(pbest^i - x^i) + \varphi_2 r(gbest^i - x^i),$$

where v represents the velocity of a particle, and x represents its position (a possible set of coefficients in an evaluated grid

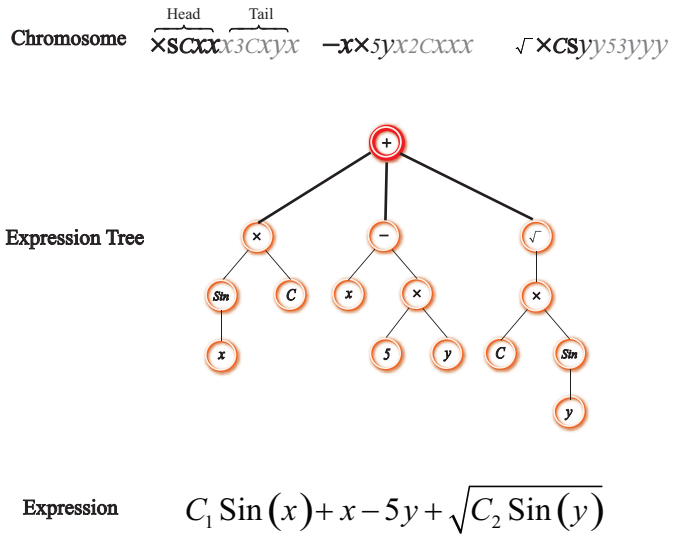


Fig. 3. GEP chromosome, expression tree, and equation. The chromosome comprises three genes which are linked by the plus operator. The head length is five and the total length is eleven.

cell model). *pbest* represents the historical best solution found by this particle, and *gbest* represents the historical best solution found by the whole population. The superscript i indexes the ith element in a vector. r is a random number and selected from the interval [0,1]. φ_1 and φ_2 stand for the acceleration constants. Using the new velocity v, the new position of a particle is updated using

$$x^i = x^i + v^i. \tag{3}$$

When the termination criterion is met, the historical best solution found by the whole population gives the optimal values for the coefficients in a given grid-cell model.

2) Components: To narrow down the searching scope and speed up the evolution process, we decompose the interference model into basic components and add them into function sets and terminal sets. In addition to these, the exponential and logarithm functions are added.

$$F = \left\{+, -, \times, \div, \sin, \cos, (\ )x + (\ )y, -(\ ), \frac{1}{(\ )}, e^{(\ )}, \log|\ |, \sqrt{|\ |}\right\}$$

$$T = \{\pi,\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ x,\ y,\ C\}$$

The sets F and T represent the function set and terminal set, respectively. The terminal set consists of tree parts: constants, variables, and coefficients. Coefficients are used to fit the model to different systems, i.e., different animals and environments. They are optimized by PSO for each system.

Notice that not all component types have the same number of members. For example, there are 8 constants to choose from, 12 functions to choose from, two variables, and only one coefficient. To avoid over-selection of the more numerous components, we first randomly select one of the four types (function, constant, variable, or coefficient) with equal probability, then randomly select a member from that type (using equal probabilities).

3) Cost Function: One important issue in this evolutionary method is how to evaluate the goodness of a possible model. In this stage, a cost function is defined to evaluate candidate models. The mean absolute error (MAE) is added to our cost function to measure the model's deviation from the measured activation maps. Furthermore, in order to restrict bloating and control the overfitting of a model (common problems in genetic programming), the minimum description length principle (MDL) is also added into the model [21]. Based on the principle of Occam's razor, it provides a penalty for model complexity. It helps to find a simple, yet accurate, model. In this research, the MDL is defined as the length of the coding regions. Thus, the cost function is defined as

$$Cost = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left|x_{ij}-\hat{x}_{ij}\right|}{MN} + \alpha\frac{\sum_{i=1}^{L} l\left(I^{i}\right)}{LK}, \tag{4}$$

where N represents the number of samples, M represents the number of points in each sample, and x_{ij} and $\hat{x}_{ij}$ represent the observed and estimated (respectively) firing rate of the jth point in ith sample. L represents the number of genes. α is a weight parameter that adjusts the relative weight between MAE and MDL.

4) Parallelization: The nested relationship between GEP and PSO, both of which are high time-complexity algorithms, increases the run time sharply. Therefore, to get the results in a reasonable time, a GPU platform using the CUDA [22] architecture was adopted to perform the processing in parallel. A master/slave model is used where the CPU is responsible for the main procedure of GEP and PSO except evaluation, and the GPU is responsible for the evaluation of all solutions in parallel at the thread level. The GPU threads do not run until all chromosomes and their corresponding coefficients are ready at the CPU for evaluation in a batch. Each GPU thread evaluates a combination of chromosomes and corresponding coefficients. Then, the CPU waits until all evaluation results are returned from the GPU side.

C. Determination of Coefficients

The purpose of the previous stage is to generate a unified form, a general but parsimonious model of the data. However, the remaining question is how to apply it to different sets of data? In other words, how do we determine the value of the coefficients? The PSO is adopted here to look for the optimal coefficient values for each sample. In the previous stage, the coefficients are optimized with only a few iterations to get a preliminary evaluation of a given model. Compared with that, the PSO's population number, population size, and the maximum generations are increased greatly at this stage to find the best coefficient values. As far as the cost function is concerned, this stage omits the MDL part because the form of the model has already been confirmed. The cost function is now changed to

$$Cost = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left|x_{ij}-\hat{x}_{ij}\right|}{MN}. \tag{5}$$

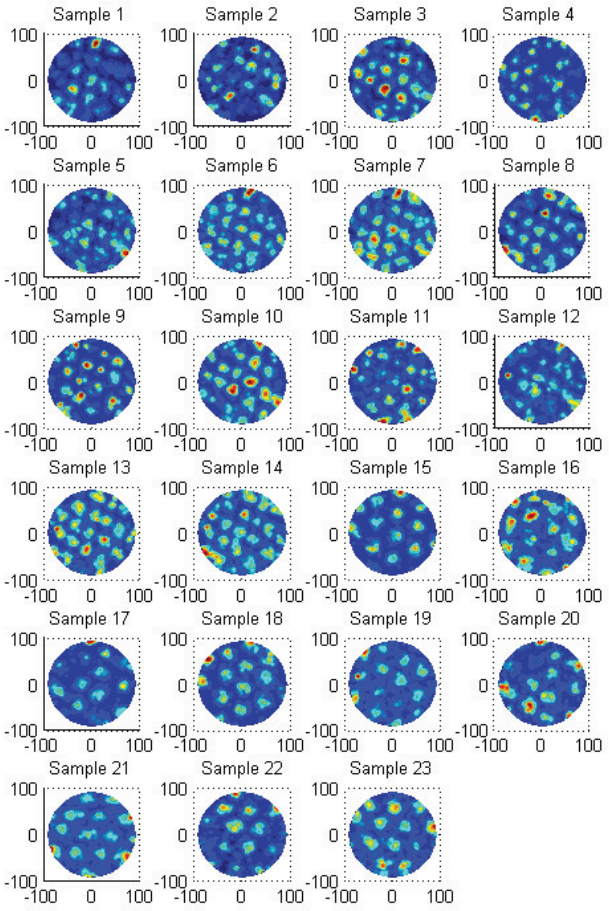


Fig. 4. The characterized rate maps of grid cells which are collected by Hafting et al [17]. The more red a site is, the higher firing rate the grid cell has at that site.

IV. Experiments

A. Observed Data

The observed data used in this paper was published by Hafting et al [17]. It was available from the Mosers' lab website: http://www.ntnu.edu/kavli/research/grid-cell-data. This data is comprised of spike-train and position data collected from rats. The information was recorded as the rat was running in a circular environment with diameter 180 cm. The recorded grid cells are located in layer II of the dorsocaudal pole of the medial entorhinal cortex. The data attained contained spike trains for ten neurons. Two trials were performed for the first rat, and one trail was performed for the second rat. Therefore, there are 23 spike trains in total. The first twenty trains come from the first rat and the other tree come from the second rat. We characterized the firing pattern (activation map) with the use of Eq. 2. Each map was smoothed and binned into matrices of 5cm $\times$ 5cm pixels. Fig. 4 illustrates the processed activation

TABLE I
EXPERIMENTAL SETTINGS AT THE FIRST STAGE

	Parameters	Settings
GEP	Number of population	8
	Population size	192
	Number of generations	1000
	Number of generations in epoch	10
	Number of genes	5
	Migration probability	10%
	One-point recombination probability	40%
	Two-points recombination probability	40%
	Gene transposition probability	0.5%
	Gene recombination probability	5%
	Mutation probability	2%
	Insertion sequence probability	10%
	Root insertion sequence probability	10%
	Head length	20
	Initialization maximum cost	2
	α	0.04
	Tournament Size	7
PSO	Population size	16
	Number of generations	100
	φ_0	1
	φ_1	1.8
	φ_2	1.8
	Number of trails	2
GPU	Number of threads in each block	768
	Number of blocks in each device	144
	Number of devices	4

maps of all samples. Nine of the samples showing a typical hexagonal pattern were selected as representative samples and used to search for a grid-cell model; they are samples 6, 8, 11, 15, 16, 17, 21, 22, and 23. The other samples were used to test the generalizability of the distilled model.

B. Experiment Settings and Comparison

A migration strategy [23], which follows the theory of equilibrium [23], is adopted to increase the performance of evolution. This strategy divides a large population into small subpopulations and transfers information between them every several generations, i.e., after each epoch. Furthermore, we performed selection for the GEP using a diversity preservation tournament based on a tree-similarity measure and global probability weights. This strategy was designed to maintain the diversity of the population and avoid premature convergence to a suboptimal solution.

In order to find the best parameters, we first ran with a small population and lower maximum generation; this is helpful to tune parameters before we increase the population size and number of generations. Then, the population size and generations are increased to get better results since the performance of evolutionary computation increases with the size of the population and number of generations. After trial and error, the settings in Table I were used in our first stage.

A desktop supercomputer with four NVIDIA Tesla K10 GPU accelerators was used in the experiment. Each GPU has 1536 CUDA cores, each with a clock rate of 745 MHz. The total amount of global memory is 3.5 GB. The total amount of shared memory per block and total number of registers

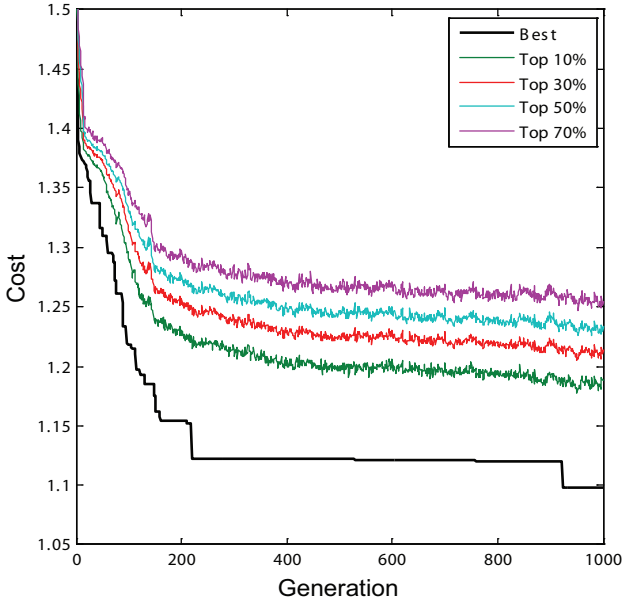


Fig. 5. The evolution of historical best solution as well as the evolution of average cost of top 10%, 30%, 50%, 70% chromosomes over time.

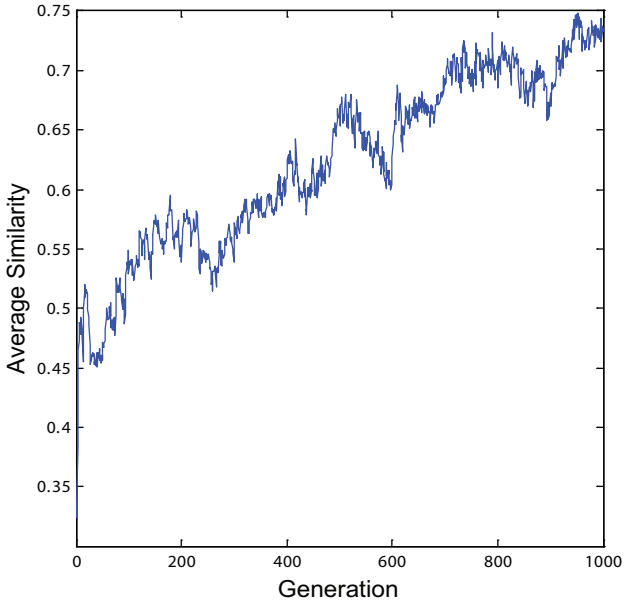


Fig. 6. The development of average similarity of all chromosomes over time.

available per block are 49152 bytes and 65536, respectively. The peak single-precision floating point performance of each accelerator is 4.58 TFlops.

C. Results and Analysis

The formal running for the first stage took 184495 seconds (51.25 hours). The time-evolution of cost is shown in Fig. 5. It illustrates the evolution of the historical best solution as well as the evolution of the average cost of the top 10%, 30%, 50%, and 70% chromosomes in all populations. It can be observed that the evolution almost converged after 400 generations.

Furthermore, Fig. 6 demonstrates the development of average similarity of all chromosomes over time. It starts from a lower value and ends at the highest value. This result indicates that uniformity increases gradually with time.

After the first stage, the best model form, with cost equal to 1.098061, is

$$\begin{aligned} f(x,y) = C + 2 + \sin\left(\frac{5-C-\sqrt{|C|}x-Cy}{6}\right) \\ + C\cos\left(\frac{C-\sin(C)x-\cos(C)y}{6}\right) \\ + C\cos\left(\frac{Cx+\cos(C)y-C}{6}\right) \\ + C\sin\left(\frac{C+C-\cos\left(\sqrt{|C|}\right)x-\sin(\ln|C|)y}{6}\right). \end{aligned} \tag{6}$$

We further simplified this original model by trigonometric identities, and combined like terms to get

$$\begin{aligned} f(x,y) = \cos(C_1x + C_2y + C_3) \\ + C_4\cos(C_5x + C_6y + C_7) \\ + C_8\cos(C_9x + C_{10}y + C_{11}) \\ + C_{12}\cos(C_{13}x + C_{14}y + C_{15}) \\ + C_{16} \end{aligned} \tag{7}$$

where C_1 to C_{16} represent the coefficients. This is the grid-cell firing model our method discovered from the neural data. The unit of x and y is centimetres. It should be noted that, despite the first cosine term, the model is consistent with oscillatory-interference models [6]. It confirms the effectiveness of evolutionary methods in discovering models for grid-cell firing.

At the second stage, to verify the generalization ability of the distilled model, the other 14 samples not used in the first stage were used as testing samples. Particle swarm optimization was used for determining coefficients on these samples. In order to search the coefficients thoroughly to get the best results, the PSO parameters were changed significantly. The population size was increased to 8192, the number of generations increased to 5000, and the number of trials increased to 4. Fig. 7 illustrates the activation maps of test samples, as well as their corresponding coefficients and cost. The average cost of all testing data is 1.2222385. It can be observed that the distilled model captures the typical hexagonal pattern of grid-cell firing on samples 1, 2, 3, 4, 9, 10, 13, 18, 19, and 20.

Fig. 7 also shows some of the other subtle patterns, besides hexagonal patterns. These results further verify the effectiveness of the proposed evolutionary computation based method on discovering grid-cell-like models. The simulated activation maps on samples 5, 7, 12, and 14 do not exhibit a typical hexagonal pattern. Among them, according to [17], samples 5 and 12 belong to cell t7c1 of the first rat, and samples 7 and 14 belong to cell t7c3 of the first rat. Even though the model was discovered by feeding the algorithm typical grid-cell activation maps, the results exhibits other structure, implying that the simple, canonical grid cell may be only part of a larger picture of navigation in the brain. As an example, Julija Krupic et al [24] showed some stripe-like firing patterns and non-spatially periodic cells.

V. Conclusion

Evolutionary computation was able to discover a model of grid-cell activation maps. The model is consistent with the oscillator interference models proposed by other researchers [6]. Our evolutionary method involved gene expression programming, combined with particle swarm optimization. The simulations were accelerated on a parallel computing architecture, taking advantage of four GPU boards. Not only did our method discover a model for common grid cells, but it also identified variants consistent with other observed grid-cell-like activation maps [24].

This study is only the beginning of this project. In the future, we plan to investigate in more detail the fit of our model to grid-cell variants. Also, we noticed that the coefficients in Eq. 7 come from various algebraic expressions involving coefficients, constants, and functions in Eq. 6. Some of those functions have limits on their range. For example, a cosine function can only yield values between -1 and 1. We would like to examine how the constraints implicit in those algebraic expressions might impact the models.

Acknowledgment

The authors would like to thank Centre for Neural Computation at Kavli Institute for Systems Neuroscience for providing raw grid cell data. This work was partially supported by National Natural Science Foundation of China under Grant No. 61573166, No. 61572230, No. 61373054, No. 61472164, No. 81301298, No.61302128, No. 61472163, Natural Sciences and Engineering Research Council of Canada (NSERC), Shandong Provincial Natural Science Foundation, China, under Grant ZR2015JL025. Science and technology project of Shandong Province under Grant No. 2015GGX101025.

References

[1] John O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat", *Brain Research*, vol. 34, no. 1, pp. 171-175, 1971.

[2] John O'Keefe and Michael L. Recce, "Phase relationship between hippocampal place units and the EEG theta rhythm", *Hippocampus*, vol. 3, no. 3, pp. 317-330, 1993.

[3] "The 2014 Nobel Prize in Physiology or Medicine - Advanced Information". Nobelprize.org. Nobel Media AB 2014. Web. 9 Oct 2015. nobelprize.org/nobel_prizes/medicine/laureates/2014.

[4] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard Moser, "Microstructure of a spatial map in the entorhinal cortex", *Nature*, vol. 436, no. 7052, pp. 801-806, 2005.

[5] Mark C. Fuhs and David S. Touretzky, "A spin glass model of path integration in rat medial entorhinal cortex", *The Journal of Neuroscience*, vol. 26, no. 16, pp. 4266-4276, 2006.

[6] Neil Burgess, Caswell Barry, and John O'Keefe, "An oscillatory interference model of grid cell firing". *Hippocampus*, vol. 17, no. 9, pp.801-812, 2007.

[7] Jeff Orchard, Hao Yang, Xiang Ji, "Does the entorhinal cortex use the Fourier transform?". *Frontiers in Computational Neuroscience*, vol. 7, article 179, 2013.

[8] M. Schmidt , H. Lipson , "Distilling free-form natural laws from experimental data", *Science*, vol. 324, pp. 81-85, 2009.

[9] AG Floares, "Automatic reverse engineering algorithm for drug gene regulating networks", *IASTED International Conference on. Artificial Intelligence and Soft Computing*, pp 238-243, 2007.

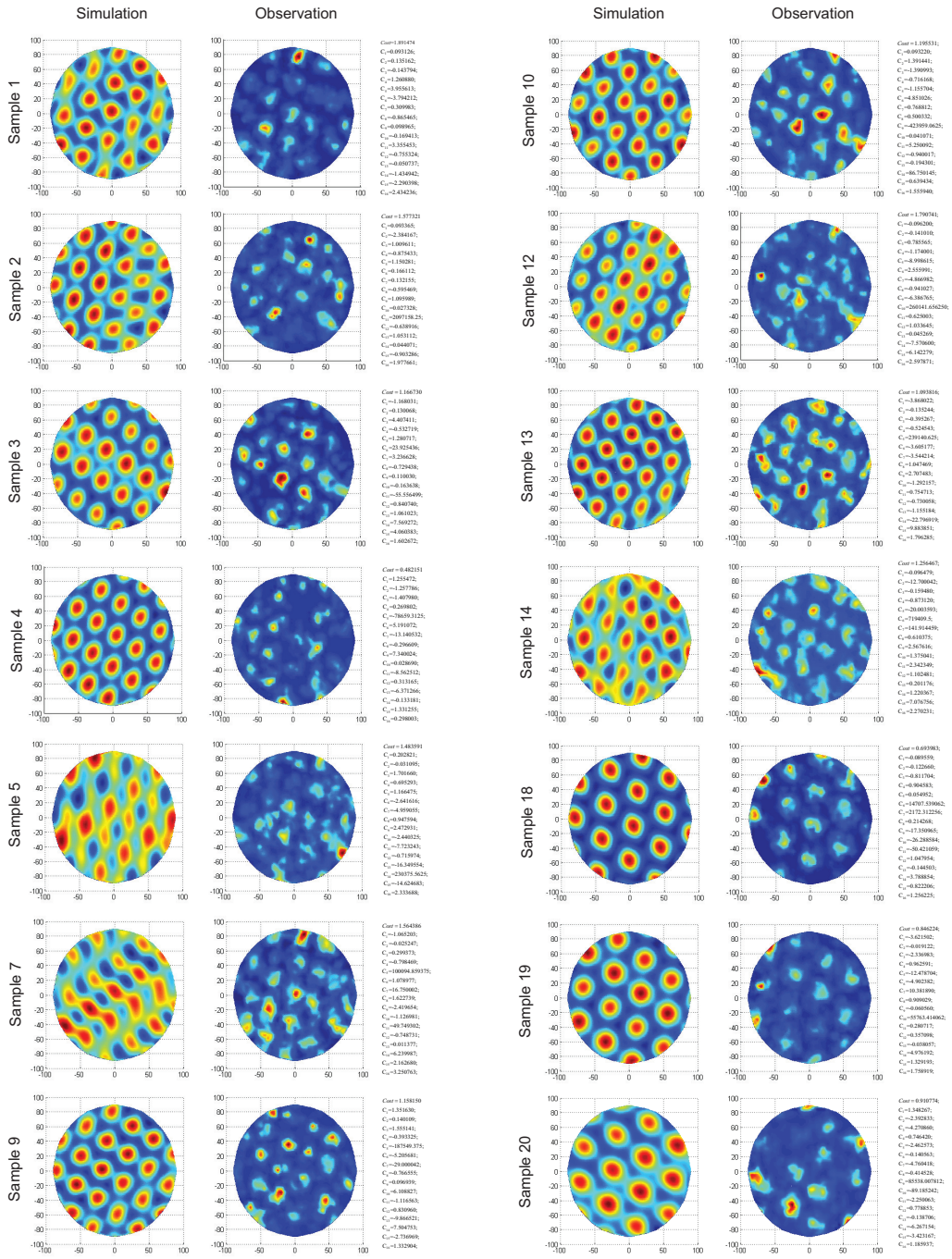


Fig. 7. Firing rate maps of test samples using Equation 7.

[10] L. Qian , H. Wang , ER. Dougherty, "Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering", *IEEE Transactions on Signal Processing*, vol. 56, pp. 3327-3339, 2008.

[11] Uday Kumar Chakraborty, "Static and dynamic modeling of solid oxide fuel cell using genetic programming", *Energy*, vol. 34, no. 6, pp. 740-751, 2009.

[12] Lin Wang, Bo Yang, XiuYang Zhao, YueHui Chen and Jun Chang, "Reverse extraction of early-age hydration kinetic equation from observed data of Portland cement", *SCIENCE CHINA Technological Sciences*, vol. 53, no. 6, pp. 1540-1553, 2010.

[13] Lin Wang, Bo Yang, Ajith Abraham, "Distilling middle-age cement hydration kinetics from observed data using phased hybrid evolution", *Soft Computing*, 2015, in press.

[14] Lin Wang, Bo Yang, Shoude Wang, Zhifeng Liang, "Building Image Feature Kinetics for Cement Hydration Using Gene Expression Programming With Similarity Weight Tournament Selection", *IEEE Transactions on Evolutionary Compuation*, vol. 19, no. 5, pp. 679- 693, 2015.

[15] Candida Ferreira, "Gene expression programming: a new adaptive algorithm for solving problems", *Com. Sys.*, vol. 13, no. 2, pp. 87-129, 2001.

[16] J. Kennedy and R. C. Eberhart, "A new optimizer using particle swarm theory", *In: Proc. 6th. Int. Symp. Micro Mach. Hum. Sci.*, pp. 39-43, 1995.

[17] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I. Moser, "Microstructure of a spatial map in the entorhinal cortex", *Nature*, pp. 801-806, 2005.

[18] Gaute T Einevoll, Felix Franke, Espen Hagen, Christophe Pouzat, and Kenneth D Harris, "Towards reliable spike-train recordings from thousands of neurons with multielectrodes", *Curr Opin Neurobiol*, vol. 22, no. 1, pp. 11-17, 2012.

[19] J.R. Koza, *Genetic programming: on the programming of computers by means of natural selection*, Cambridge, MA: MIT Press, 1992.

[20] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, 1975 (second edition: MIT Press, 1992)

[21] B.T. Zhang, H. Muhlenbein, "Balancing accuracy and parsimony in genetic programming", *Evol. Comput.*, vol.3, no. 1, pp. 17-38, 1995.

[22] *CUDA programming guide version 2.3.1*, NVIDIA, 2009.

[23] Lunjun Chen, *Optimal design for machinery: genetic algorithm*, Machinery Industry Press, Beijing, 2005. (in Chinese)

[24] Julija Krupic, Neil Burgess, John OKeefe, "Neural Representations of Location Composed of Spatially Periodic Bands", *Science*, vol. 337, pp. 853-857, 2012.