Evolving Spiking Neural Networks of Artificial Creatures Using Genetic Algorithm

Elahe Eskandari, Arash Ahmadi, Shaghayegh Gomar, Majid Ahmadi, Mehrdad Saif
University of Windsor, Windsor, Ontario, Canada

Abstract—**This paper presents a Genetic Algorithm (GA) based evolution framework in which Spiking Neural Network (SNN) of single or a colony of artificial creatures are evolved for higher chance of survival in a virtual environment. The artificial creatures are composed of randomly connected Izhikevich spiking reservoir neural networks. Inspired by biological neurons, the neuronal connections are considered with different axonal conduction delays. Simulation results prove that the evolutionary algorithm has the capability to find or synthesis artificial creatures which can survive in the environment successfully and also simulations verify that colony approach has a better performance in comparison with a single complex creature.**

I. Introduction

Despite recent advancements, researchers are still attempting to figure out wide range of unsolved problems about the most powerful organism of human body: the brain. For instance, in [1] a C.elegans artificial creature has been presented. In this case, since the C.elegans has a preliminary known specific neural structure there is no need for further investigation of the neural structure, whereas the approach which is presented here is to discover the most compatible structure among a large space of possible solutions. This is a step towards achieving a better understanding of real behavior in nature, therefore, a reservoir network is used.

A. Reservoir Computing

SNN is capable of producing variety dynamical behaviors. They can be organized for many applications such as multilayer feed forward or recurrent networks, like traditional neural networks. According to biological observations, however, a biological network can be composed of a randomly connected form within a large scale consistent structure. With this explanation, a new form of artificial neural network was introduced: Reservoir Computing (RC). This type of network is suitable especially for temporal input and output pattern processing. Principal specifications of reservoir computing model are [26-29]: - Layer of K input neurons are connected to the reservoir. - A main part consist of M randomly connected neurons. - A layer of readout neurons with trained connections from reservoir network. The main motivation of reservoir computing is a large solution space creation to discover the most compatible networks for special purposes.

B. Neuron Model

A variety of models have been developed for biological neurons with different degrees of precision and complexity. Choosing the model strongly depends on the application, which is a trade-off between precision and computational cost. From a precision viewpoint, the Hodgkin Huxley (HH) [14] model is a model with the highest accuracy in terms of biochemical modelling of the neurons, but it has a high computational cost. On the other hand, Integrate and Fire (IF) is recognized as one of the simplest models with highest computational efficiency [2]. This model is a low cost but inaccurate model which is incapable of producing complex dynamic of neurons such as bursting. IZ model, introduced by Izhikevich [15], presents a compromise between accuracy and computability, which has made it a widely accepted popular model. It is claimed to be as realistic as Hodgkin-Huxley model. This model is a 2-dimensional map-based model that is described by two differential equations:

$$\begin{cases} \frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I \\ \frac{du}{dt} = a(bv - u) \end{cases} \tag{1}$$

with the auxiliary after-spike resting equation as:

$$\text{if } v > 0 \; then \quad \begin{cases} v \to c \\ u \to u = u + d \end{cases} \tag{2}$$

where v is the membrane potential, u is the recovery variable and I represents the input current. When a small pulse of current is applied, the membrane potential raises. If the current be adequately strong, membrane voltage crosses its apex (30mV); after which the membrane potential and the recovery variable will reset according to the auxiliary reset equations. The higher applied current the higher frequency neuron has in its output train of spikes. Here a, b, c, and d are dimensionless variables and are described in [15]. The range of parameters in Izhikevich model can be seen in the TABLE I [30]. Since this model is computationally feasible and biologically acceptable, it is utilized here. In this work reservoir networks composed of spiking neurons are simulated. In continue, design of a single complex creature and its evolution process are described and it is examined if genetic algorithm is an appropriate method for evolution process of the single creatures.

TABLE I: Boundaries of Parameters in Izhikevich Model

	a	b	c	d
Min	0.002	0.1	-65	0.05
Max	0.1	0.3	-55	8

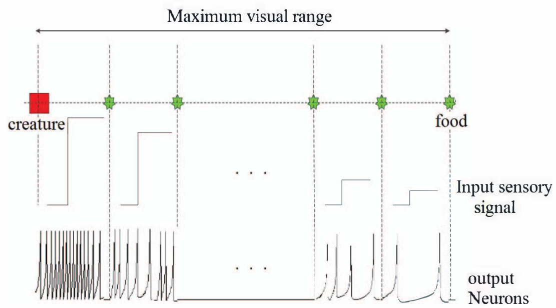

Fig. 1: Visual input coding scheme.

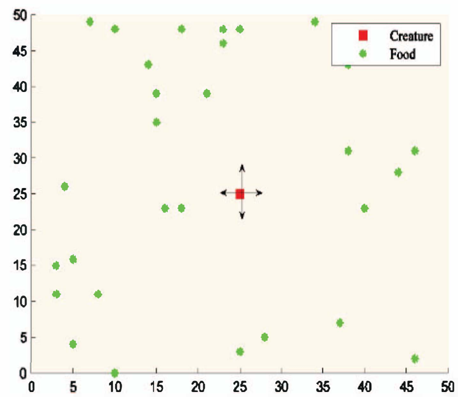

Fig. 2: A typical creature and environment.

II. Design And Evolution of Artificial Creatures

A. Single creature structure

Each creature network has an input connected to an output layer through a randomly connected IZ neurons with different axonal conduction delays between each couple of neurons. Please note, in this network, not only the connection between two neurons are random, but also the neurons type selection is random, too. In reality, in the mammalian nervous system, the axonal conduction delays are different depending on the type and spatial positioning of the neurons. In this study a random conduction delay has been assumed for each synaptic connection in range of 1ms to D=20 ms [16].

B. Single creature and environment interaction

The single biologically-inspired artificial creature is placed in a two-dimensional environment in which the creature must find randomly located foods. Before each movement, the creature receives input signals from the foods in its vision range. Strength of the stimuli signals depends on the distance from the food, where the strength of the stimuli signals linearly decreases with the distance from a food. In terms of neural activity, the stimuli signal is applied to the IZ model of the respective neurons as an input current, Fig. 1 shows the visual input coding scheme. As can be seen from this figure, if the creature is closer to the food, it receives stronger signal and subsequently related output neurons fire spikes with higher frequency

It should be noted that each creature has an initial energy level and by every movement the initial energy decreases and by eating the food, the energy level increases up to a specific level. Accordingly, each creature can survive in the environment until the energy level becomes zero. Input layer of the neural network creatures consist of N_i neurons for input signals reception and output layer of the network has N_o neurons for locomotion instructions and moving in the environment. Four directions of movement have been defined as: forward, right, left and back. Therefore, $N_i/4$ neurons are responsible for receiving input signals from each direction and $N_o/4$ neurons are responsible for locomotion in each direction. Fig.2 illustrates a typical creature and external environment. In this picture red square denotes creature and green points show distributed foods in the environment. One of the methods for information coding in neural systems by spikes is rate coding in which the population activity rate coding has been used to distinguish different directions by related neurons. This coding has been successfully applied in experiments on sensory or motor systems. In population activity, average number of firing in a population of neurons is calculated in a fixed time window as:

$$A(t) = \frac{1}{\Delta t}\frac{n_{act}(t;t+\Delta t)}{N_p} \tag{3}$$

where N_p is number of neurons in the population, $n_act(tt+\Delta t)$ is the number of spikes which are issued by all neurons in the population during the time window. Window time has a fixed length between t and t+t and t is an appropriate time interval [10]. In each time window the equation 3 is calculated for neurons of each direction named Ai(t) and then to distinguish the direction Ai(i=1,2..4) values are compared with each other. Ultimately, creature moves toward direction that the related neurons have maximum rate of fires. These fixed time windows consist of 600 time-steps. Each time step is 0.5 ms. Fig.3 shows flowchart of the details.

The neurons which are responsible for moving one direction may fire the same maximum of spikes that the neurons which are responsible for another direction. This situation can happen for any pairs of directions, that is right (or forward and left, back and right or back and left). One solution for this problem is injecting the noise into the spiking networks which is more representative of the real life systems. By this way, some degree of randomness is given to the network and the probability of identical maximum firing decreases. Even though the probability of such a situation is low, but in that case, an unfixed input stimulus can be randomly applied to the neurons before each movement [8].

To introduce noise into networks in this study we utilize a function, which produces random numbers between 0-20 mV. Therefore, the chance of having the same maximum firing for different directions is reduced dramatically. Also, it should be noted that the strength of the noise should be limited to avoid strong effects on creatures efficiency.

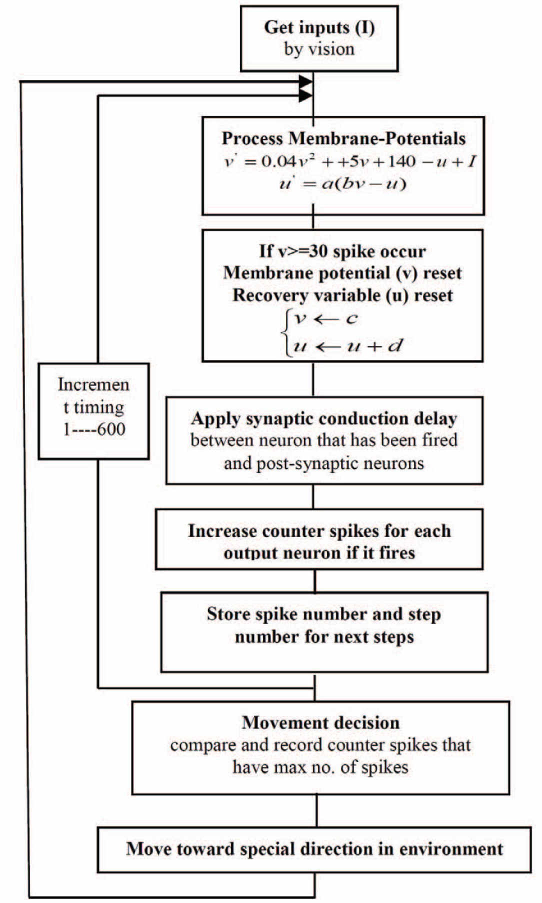

Fig. 3: Flowchart for movements of creature.

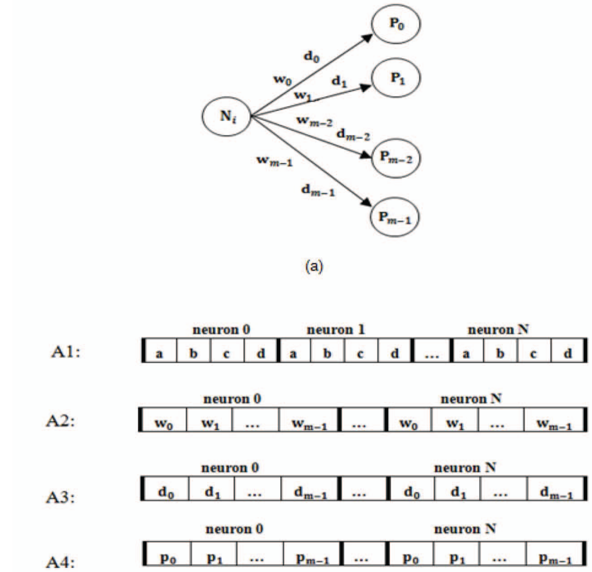

Fig. 4: a) A typical neuron and postsynaptic connections with weights and delays, b) A typical chromosome A1: Parameters of the Izhikevich model, A2: synaptic weights, A3: conduction delays, A4: post-synaptic neurons number.

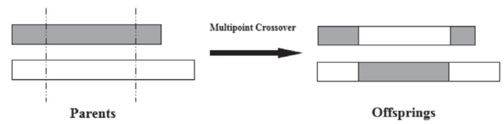

Fig. 5: Flowchart for movements of creature.

As can be seen from flowchart of Fig.4 , at first creature receives inputs from the environment by its vision neurons. In the second stage, a 600-step time window is considered to determine direction. Each time step is 0.5ms. So the creature can determine the direction in 300ms. Neural activities are calculated in sub steps during time window. After each time window the number of spikes of each output neuron has been counted and Ai(t) values are calculated and compared with each other. Therefore, creature moves to the direction that its related neurons have maximum rate of firing. Then creature moves to a new location and receives new input from environment and the process will be repeated.

C. Evolutionary Procedure

In the GA algorithm an initial random population of creatures is generated so that the neurons type, connections, weights values and axonal delays are selected randomly. The neural networks of the creatures are coded as chromosomes, as shown in Fig.4 Each chromosome consists of four parts: A_1, A_2, A_3 and A_4. Each part consists of N segments for N neurons of a creature. The first part, A_1, denotes a, b, c and d parameters of Izhikevich neuron model. Each segment of A2 shows weights of the connections between corresponding neuron and its postsynaptic neurons. Similarly, each segment of A_3 indicates delays of the connections between corresponding neuron and its postsynaptic neurons. Segment A_4 shows postsynaptic neurons number for related neuron of each segment. Each artificial creature, which its neural network is coded in chromosome, is placed in an environment to search and catch the randomly located foods in its periphery. Then fitness value is calculated for each creature as the number of foods it has found. Then next generation are produced by combination of the elites (15%), crossover (55%) and mutation (30%) of the initial population. Elites are the best chromosomes, which are directly transferred to the next generation. Because of long length of the chromosomes, five cut points are randomly chosen for crossover in each parent. Fig.5 shows a typical crossover with two cut points and Fig.6 illustrates a flowchart for the proposed evolutionary model. Selections are based on Roulette Wheel approach, more detailed information can be found in [27].

D. Simulation Result for Evolving of Single Creatures

A virtual environment and artificial creatures are designed and implemented in a C++ platform in which simulations are performed. At First, a population of 100 single artificial creatures were tested and evaluated by GA. Each single crea-

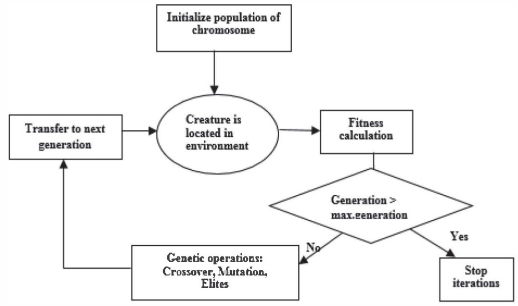

Fig. 6: Genetic Algorithm flowchart.

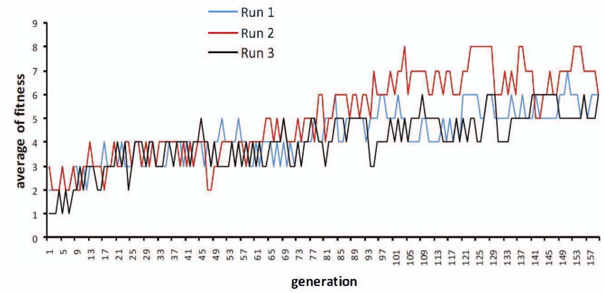

Fig. 7: The maximum fitness values of artificial creatures for three executions of the program in single creature approach.

ture has 108 neurons. The input layer of the creature's neural network consists of 36 neurons in which every 9 neurons are responsible for receiving visual input signals from foods in one special direction. Also, the output layer of this neural network is made of 36 neurons as every 9 neurons are related to each locomotion part. The rest of the neurons, internal cells, form reservoir part of the network. In this structure each neuron is connected to maximum of 4 postsynaptic neurons with different axonal conduction delays between every two neurons

The single creature is located in a 50*50 two-dimensional environment. 30 foods are randomly distributed in the environment. The initial energy level for each single creature has been assumed 675 units. The amount of energy is decreased by one unit in each time step and increased by catching a food to a certain level. Accordingly, each creature survives in the external environment until the energy level becomes zero. The creature's fitness is calculated based on the number of foods that this creature can find. GA is applied to evolve the population of artificial creatures. The average of artificial creatures' fitness values, which is representing the evolutionary procedure of them, under above conditions has been simulated for three executions in Fig.7 As is observable, this graph indicates that the average of fitness values grows in generations. Since GA is sensitive to primary randomly population, a paltry difference is expected among three states but all states are in agreement about progressing procedure. In continue, Fig.8(a-c) indicates the maximum fitness value of each generation for these states. These figures indicate that the best fitness is increased with progressing generations and evolving creatures; as it is shown in Fig.8 (a-c), the best creature could find 22, 23 and 20 foods of 30 randomly distributed foods in state 1, 2 and 3, respectively.

III. COLONY OF SIMPLE CREATURES

Evolving artificial single creature using GA algorithm was successful to reach evolved creatures that are capable of finding distributed foods in an environment. To enhance the performance of artificial creatures in foraging foods, inspired by nature, a colony approach is discussed and it is examined if GA is practical for this case. In previous scheme, the single creatures were rather complex and were made of 108 neurons, but in this section, colonies are made of a number of simple creatures. A population of 100 colonies that each one is made of 9 simple creatures with identical neural network are evolved by GA. In these colonies, each creature has a simple neural network, which consists of 12 neurons as: 4 neurons in the input layer for receiving input signal from 4 directions, 4 neurons in the output layer for movement in four directions and the main structure part of the creatures neural network is made of 4 neurons. To distinguish direction of the movement, a simple firing rate coding has been utilized for each creature so that one neuron is considered for each locomotion part. In every time window, the total numbers of spikes in the output neurons are compared with each other and artificial creature moves to the direction in which the firing rate is maximum. The initial energy level for each creature in the colony is assumed to be 75 units. At first, in the initial population, creatures neural networks are coded as chromosomes. Each colony of creatures is generated by coping chromosomes, whereas all members of each colony are genetically identical. In continue, each colony of simple creatures is located in the environment, in different places. Similar to the single creature approach, they search the peripherals in order to find distributed foods in the environment. The fitness value for each colony is calculated and evolving process continues by GA, in this approach some of the colonies are successful to find all 30 foods and survive. Fig.9 shows the number of successful colonies in each generation.

IV. COMPARISON AND DISCUSSION

Simulations results for a complex single creature and a colony of simple creatures are compared with each other under similar conditions in this section. For this, the total number of neurons must be equal to the number of neurons in a single complex creature in all simple creatures of a colony. Therefore, if a single creature is composed of N neurons with initial energy level of E, each simple creature must be composed of N/S neurons in the colonies, in which S is the total number of creatures in the colony. Furthermore, the total number of input (output) neurons of all simple creatures in a colony is equal to the number of input (output) neurons of the single complex creature, so that the numbers of neurons which are responsible for each inputs (output) direction are equal in these

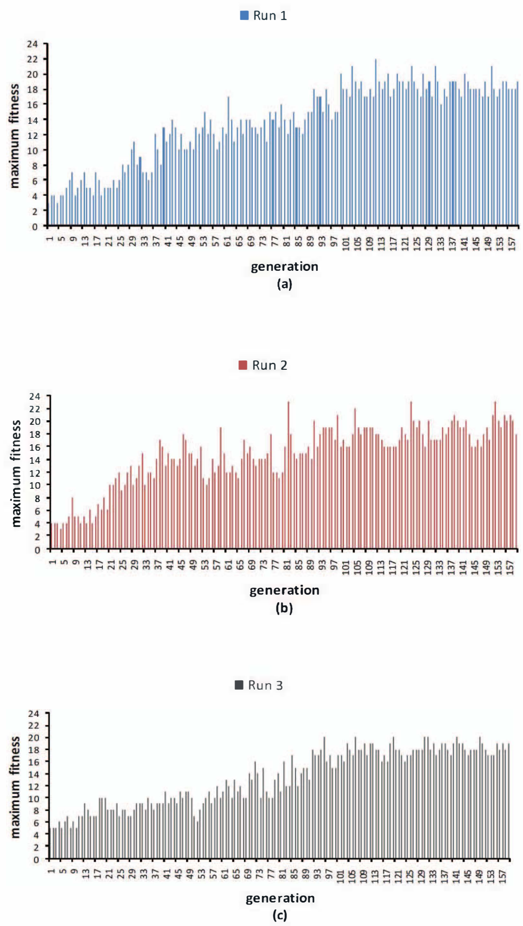

Fig. 8: The number of successful colonies with progressing generations for colony approach.

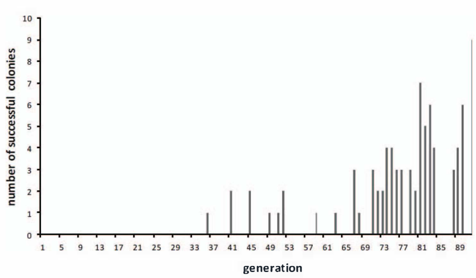

Fig. 9: The number of successful colonies with progressing generations for colony approach.

two approaches. These explanations are equated in 4-5: Eq 4 and 5

In addition, initial energy level for every simple creature must be E/S , also Fig./refFig.10 shows these explanations. More details are presented in TABLE II for $N = 108$, $S = 9$ and $E = 675$.

For the sake of simplicity, the single complex creature approach is called state A and the colony approach is state

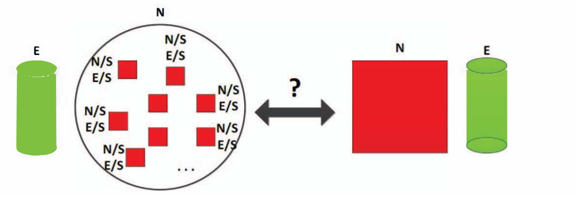

Fig. 10: A colony and a complex creature with equal conditions.

TABLE II: Comparison of The Single and Colony Approaches

Number of Neurons	Colony: Each Creature	Colony: Total of Creatures	Single Creature
Input layer	4	36	36
Output layer	4	36	36
Reservoir	4	36	36
Total	12	108	108
Initial Energy	75	675	675

B. In each generation fitness functions are calculated for all individuals. Fig.11(a) compares the average fitness in generations for two cases in one typical simulation running. As can be seen from this figure, the average of fitness in two states grows with generations similar to the previous sections; but the colony of creatures scheme is more successful and the fitness values are higher than state A. Further, the maximum fitness values have been shown for two states in Fig. 11(b). As it is observable from this figure, in state B, some colonies found all of 30 randomly distributed food sources in the environment by evolution. As it is mentioned before, the fitness function for each colony is equal to the number of foods that a colony or a creature can find in an environment. In state A, the evolution results for complex creatures were not considerably successful and the creatures could find a small number of foods in the environment while in the state B, creature colonies could find many of randomly distributed foods by evolution and even some of colonies were successful to catch all of the foods and survive. Fig. 11(c) shows the number of these successful colonies. All artificial creatures in one colony have identical Spiking Neural Networks. A typical neural network for a member of successful colony is shown in Fig.12, weights and delays of all synaptic connections are annotated in this graph. Izhikevich parameters for each neuron in this structure are also presented in Table III. For a better visual understanding of the creatures movement,Fig.13 illustrates a movement path for each member of a typical successful colony.

Although characteristics of the successful structures can be explained in the form of matrixes in which represent the connections, delays and weights, but comparison of these structures are difficult. Unfortunately, some well-known criteria such as correlation is not an appropriate in this case because there are patterns of similarity which cannot be detected using such criteria. Fig. 15 shows this limitation in an example. In this figure, two typical neural networks are shown.

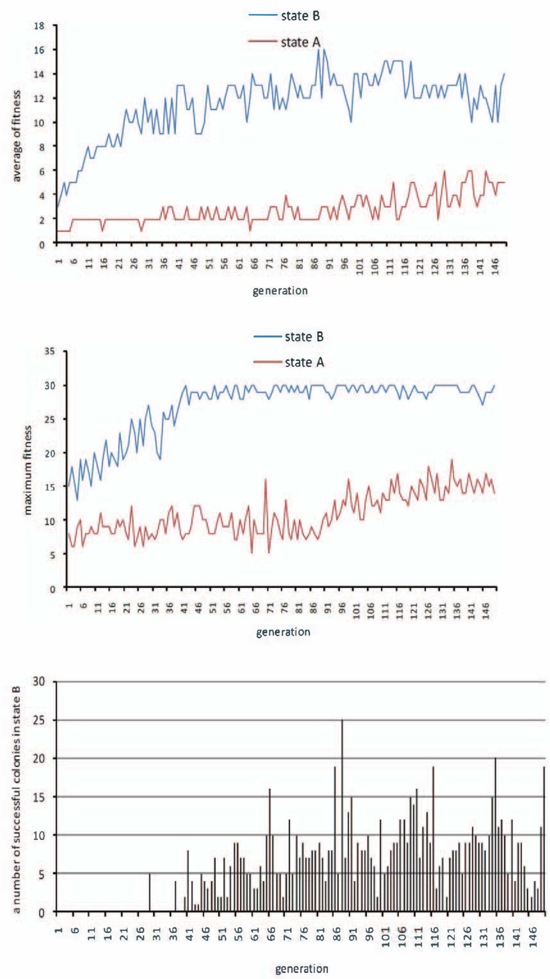

Fig. 11: a) Average fitness, b) maximum fitness, c) the number of these successful colonies with progressing generations for two approaches. State A shows results for single creature approach, state B shows results for colony approach.

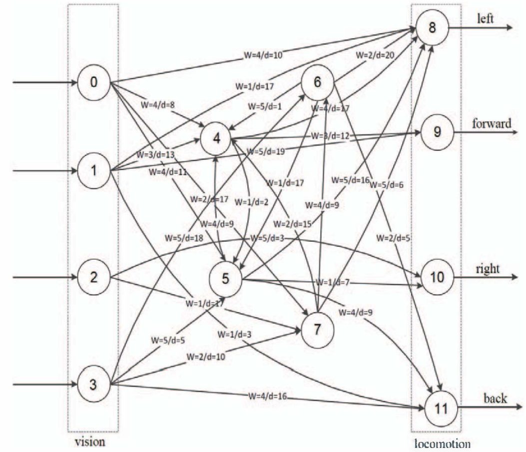

Fig. 12: Typical neural network of a member of successful colony.

TABLE III: Parameters of Izhikevich Model For Each Neuron In Successful Artificial Creature Colony At Fig.12

Neuron	a	b	c	d
0	0.0198	0.1815	-59.6722	0.7220
1	0.0866	0.1437	-56.4655	3.3161
2	0.0967	0.1974	-64.0912	2.0647
3	0.0212	0.2340	-55.3577	5.8897
4	0.0287	0.2818	-63.9609	7.7176
5	0.0948	0.1694	-63.4235	7.5771
6	0.0183	0.1716	-57.9565	2.5717
7	0.0357	01042	-57.8677	0.3819
8	0.0522	0.2395	-57.7838	5.1148
9	0.0194	0.2531	-64.4727	2.9580
10	0.0512	0.1837	-63.5873	1.4572
11	0.0519	0.2216	-62.4976	3.1060

Connections of these networks are indicated as matrixes so that if connection exists between two neurons, the corresponding value in the matrix is 1, otherwise 0. As can be seen, these neural networks have same structures but different matrixes because the positions of neuron 2 and 4 have been changed. Also this limitation is for other characteristics of successful structures, such as matrixes of weights, delay and Izhikevich neurons parameters. Therefore, finding the similarities among matrixes of experiment result which show similarities between successful creatures is a challenging task.

In order to have a better comparison between successful colonies structures a set of 3-dimensional graphs like Fig. 15, are utilized in this section. In this graph, an index number is considered for each successful colony and each plate represents characteristics of individuals of the colonies (please note that each colonys members are identical). Neural network characteristics of survived colonies in all generations are shown in Fig. 16 (a-e). This number is in chronological order of generations as the later generations have larger indexes. Since a generation might has more than one successful colonies, therefore, the index of a creature might not show the exact number of its generations but we definitely can say that higher indexes represent more evolved creatures. The Z axis shows this index. Fig. 18(a) and Fig. 18(b) show Izhikevich neurons parameters of the evolved creatures. In these figures each Z-plane corresponds one successful neural network. X-axis and Y-axis represent a and b parameters in Fig.16(a) and c and d parameters in Fig. 16(b), respectively. Since each colony member has 12 neurons, there are 12 corresponding dots in Z-plane. These 3-dimensional graphs (Fig.16(a) and Fig.16(b)) shows similarity among neurons of 100 successful creatures. Also, each Z-plane in Fig. 16(c), Fig. 16(d) and Fig. 16(e) indicates delays, weights and connections of the neurons for each successful creature. In Fig. 16(c), X-axis represents pre- synaptic neuron's number and Y-axis shows weight value which is in range of 0-5. Each neuron

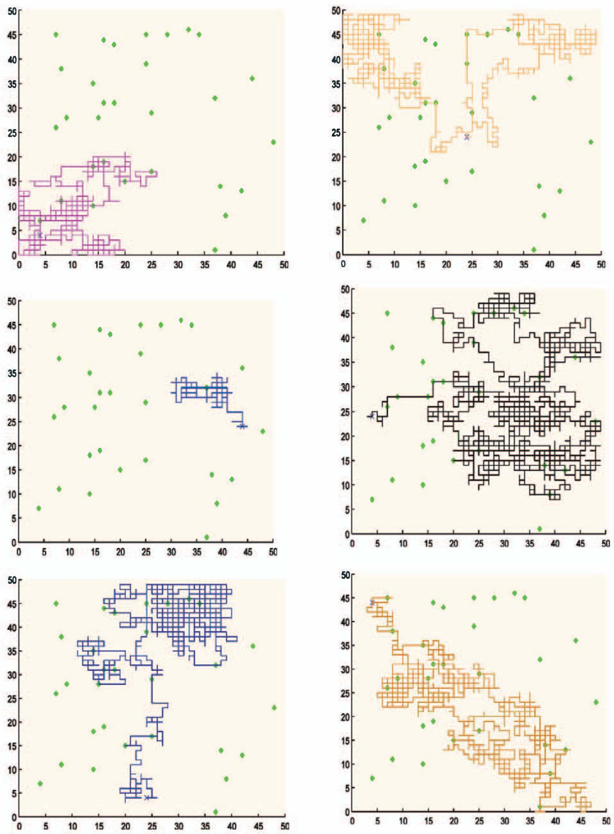

Fig. 13: Path example of creatures for one typical successful colony that consist of 9 simple creatures.

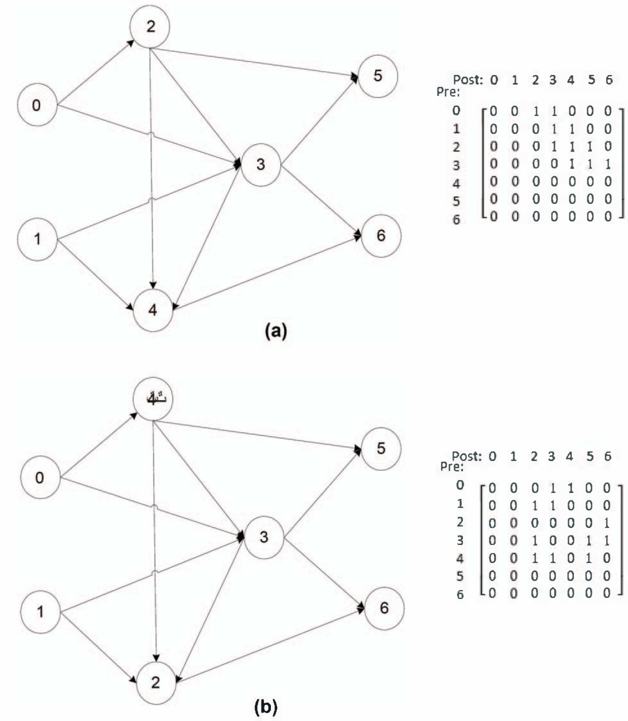

Fig. 14: Path example of creatures for one typical successful colony that consist of 9 simple creatures.

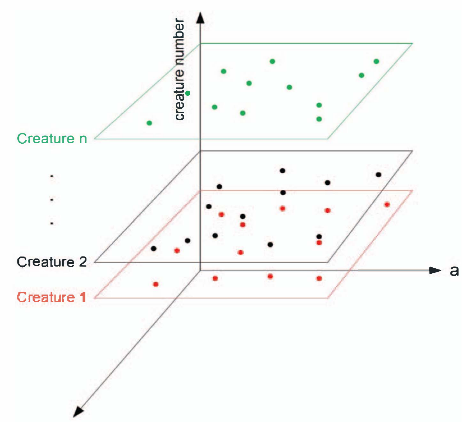

Fig. 15: 3-D graph for characteristic of successful creatures.

is connected to maximum of 4 post-synaptic neurons so in this figure, 4 weight values are indicated for every neuron of each creature. Considering this note, zero weights denote no connection between two neurons. Fig. 16(d) exhibits synaptic delays between the connections in neural networks of the successful creatures. X-axis represents pre-synaptic neuron's number and Y-axis shows delay value which is in range of 1-20 ms. Four delay values are indicated for each neuron of every creature's neural network. Fig.16(e) shows the connections so that X-axis illustrates presynaptic neuron's number and Y-axis exhibits postsynaptic neuron's number. There are four postsynaptic neurons for each presynaptic neuron in the X-Y plane. As is observable, these graphs present a strong similarity among successful creatures and a tending towards a specific structure of the neural network because of similar structures which can be extracted from these graphs.

V. Conclusion

A Genetic Algorithm was presented for evolving artificial life forms including single and colonies of creatures. These artificial creatures were composed of biological spiking neurons which have been connected randomly. GA could be able to discover the most compatible structures in the solution space. On the other hand, it was an appropriate tool to develop biological based creatures because of its biological nature as the results of simulations verified that the Genetic Algorithm is efficient for this purpose. Winner Artificial creatures or colonies of GA algorithm were discovered among all creatures or colonies that were located in a 2-D environment in which they could search to catch the distributed foods and survive. In addition, inspiring nature, colony approach is an efficient way; therefore, we discussed this scheme and compared it with a rather complex artificial creature under similar conditions. Accordingly, it is concluded that the colony approach is more successful by evolution so that some of colonies have searched environment efficiently and been able to find all

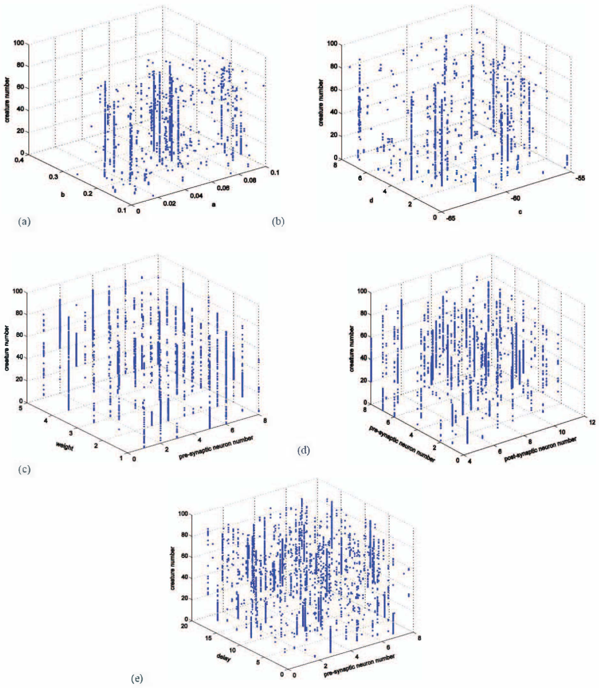

Fig. 16: a) parameters a and b of Izhikevich model, b) parameters c and d of Izhikevich model, c) weights, d) synaptic delays, e) connections of all neurons in a number of successful creature colonies

the randomly distributed food sources in the environment. Moreover, it is shown that, by evolution, neural networks of the successful creatures tend toward a specific neural network structure. We believe that this study can be a step forward to understand the morphology of artificial creatures which have biological nature regarding the use of spiking neural network. Also this paper suggests more complex artificial life examination by adding different part to these networks similar to different segments of the brain such as: vision, locomotion, hippocampus and communication in a future work. One of the main characteristics of this study is applying different axonal conduction delay between every two neurons in the neural network of artificial creatures, which it motivates us to enhance the artificial lives by STDP learning in future works.

REFERENCES

[1] Bailey, J.A., Wilcock, R., Wilson, P.R., and Chad, J.E. (2011). “Behavioral simulation and synthesis of biological neuron systems using synthesizable VHDL,”*Neurocomputing*, 74 (14-15), 2392- 2406.

[2] Brunel, N., and Van Rossum, M.C.W. (2007). “Lapicques 1907 paper: from frogs to integrate-and-fire,”*Biological Cybernetics*, 97, 337339.

[3] Brbisson, A.D. (2012). “Evolving Spiking Neural Network controllers for virtual animates that explore and forage in unknown environments.” MSc, Imperial College, London.

[4] Cliff, D., Harvey, I., and Husbands, P. (1992). “Issues in evolutionary robotics. ” School of Cognitive and Computing Sciences, University of Sussex.

[5] Dreyfus, G. (2005). “Neural Networks: Methodology and Applications. ” Berlin; New York: Springer.

[6] Coley, D. A. (1998).“An Introduction to Genetic Algorithms for Scientists and Engineers. ”*World Scientific* , University of Exeter.

[7] Davis, L. (1991).“Handbook of Genetic Algorithms. ”New York: Van Nostrand Reinhold.

[8] Su, D. (2010).“Neural networks for control of artificial life form. ” MSc, York University, Toronto.

[9] Floreano, D., and Mattiussi, C. (2001). “Evolution of spiking neural controllers for autonomous vision-based robots. ” In T. Gomi (Ed.), Evolutionary Robotics IV, Berlin: Springer.

[10] Gerstner, W.,and Kistler, W. (2002).“Spiking Neuron Models. ” Cambridge: Cambridge University Press.

[11] Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M., and Nolfi, S. (2007). “Self-Organized Coordinated Motion in Groups of Physically Connected Robots.” *IEEE Transactions on Systems, Man and Cybernetics* .37(1), 224239.

[12] Goldberg, D.E. (1989).“Genetic Algorithms in Search, Optimization and Machine Learning.” Reading MA: Addison- Wesley.

[13] Holland, J.H. (1998). “Emergence. ” Oxford: Oxford University Press.

[14] Hodgkin, A.L. and Huxley, A.F. (1952). “A quantitative description of membrane current and its application to conduction and excitation in nerve.” *Physiol J*, 117(4), 500544.

[15] Izhikevich, E. (2003). “Simple model of spiking neurons. IEEE Transactions on Neural Networks, 14(6), 15691572.

[16] Izhikevich, E. (2005). “Polychronization: computation with Spikes.”*Neural Computation*, 18(2), 245-282.

[17] Jin, X. (2010). “Parallel simulation of neural networks on spinnaker universal neuromorphic hardware.” PhD, Manchester University, Manchester.

[18] Judd, J.S. (1990). “Neural Network Design and the Complexity of Learning.” Cambridge: MIT Press.

[19] Kasabov, N. (2010). “To spike or not to spike: a probabilistic spiking neuron model.”*Neural Networks* ,23(1), 16-19.

[20] Mehra, P., and Wah, B.W. (1992). “Artificial Neural Networks: Concepts and Theory. CA: IEEE Comput Society Press.

[21] Colley, M., Callaghan, V., Hagras, H., Pounds-Cornish, A., and Clarke, G. (2002). “Evolving Spiking Neural Network controllers for autonomous robots.” Proceedings of the 2004 IEEE International Conference on Robotics and Automation (pp. 4620- 4626), New Orlean, IEEE.

[22] Mitchell,M. (1998).“An Introduction to Genetic Algorithms.” Cambridge: MIT Press.

[23] Moutarde, F. (2008). “A robot behavior-learning experiment using particle swarm optimization for training a neural-based animate.” Proceedings of IEEE symposium on the Artificial Life Conference (pp. 116123), Mines Paris Tech, IEEE.

[24] Oros, N., Steuber, V., Davey, N., Canamero, L., and Adams, R. (2009). “Evolution of bilateral symmetry in agents controlled by Spiking Neural Networks.” Proceedings of IEEE symposium on the Artificial Life Conference (pp. 116123),Nashville, IEEE.

[25] Paugam-Moisy H., and Bohte, S. (2010). “Computing with spiking neuron networks. In: Rozenberg G, Bck T, Kok J, editors.” Handbook of Natural Computing. Heidelberg: Springer Verlag.

[26] Paugam-Moisy, H., Martinez, R., and Bengio, S. (2008). “Delay learning and polychronization for reservoir computing.” Neurocomputing, 71(7), 11431158.

[27] Poli, R., Langdon, W.B., and McPhee, N.F. (2008). “A Field Guide to Genetic Programming. ”USA: Stanford University.

[28] Saggie-Wexler, K., Keinan, A., and Ruppin, E. (2006). “Neural processing of counting in evolved spiking and McCulloch-Pitts agents.” Artificial Life, 12(1), 1-16.

[29] Schrauwen, B., Verstraeten, D., and Campenhout, J.V. (2007). “An overview of reservoir computing: theory, applications and implementations.” Proceedings of the15th European Symposium on Advances in Computational Intelligence and Learning Conference(pp. 471482).

[30] Soares, G., Borges, H., Gomes, R., and Oliveira, G. (2010). “Emergence of neuronal groups on a self-organized spiking neurons network based on genetic algorithm.” Proceedings of the 2010 eleventh Brazilian symposium on the Neural networks (pp. 4348). Brazil, IEEE.

[31] Urbanczik, R., and Senn, W. (2009). “Reinforcement learning in populations of spiking neurons.” Nature Neuroscience, 12(3), 250-252.

[32] Vreeken, J. (2003). “Spiking Neural Networks: an introduction.” Artificial Intelligence Laboratory, Intelligent Systems Group, Utrecht University:Institute for Information and Computing Sciences. www.infoscience.epfl.ch