Memristor's characteristics: From non-ideal to ideal

Fan Sun(孙帆), Jing Su(粟静), Jie Li(李杰), Shukai Duan(段书凯), and Xiaofang Hu(胡小方)†

College of Artificial Intelligence, Southwest University, Chongqing 400715, *China*

(Received 2 May 2022; revised manuscript received 23 May 2022; accepted manuscript online 2 June 2022)

Memristor has been widely studied in the field of neuromorphic computing and is considered to be a strong candidate to break the von Neumann bottleneck. However, the non-ideal characteristics of memristor seriously limit its practical application. There are two sides to everything, and memristors are no exception. The non-ideal characteristics of memristors may become ideal in some applications. Genetic algorithm (GA) is a method to search for the optimal solution by simulating the process of biological evolution. It is widely used in the fields of machine learning, combinatorial optimization, and signal processing. In this paper, we simulate the biological evolutionary behavior in GA by using the non-ideal characteristics of memristors, based on which we design peripheral circuits and path planning algorithms based on memristor networks. The experimental results show that the non-ideal characteristics of memristor can well simulate the biological evolution behavior in GA.

Keywords: memristor, non-ideal characteristic, genetic algorithm, path planning

PACS: 84.35.+i, 84.37.+q, 87.19.lv **DOI:** 10.1088/1674-1056/ac7548

1. Introduction

Memristors are non-volatile analog memories whose resistance values (conductance value) change with external excitations.[1] Since the advent of the first memristor prepared by Hewlett and Packard (HP) in 2008,[2] memristors have been widely used in the field of neuromorphic computing.[3–7] The ion migration phenomenon in memristors is very similar to the neurotransmitter spreading process in neurosynapses and is widely used in neural networks for synaptic functions. Jo *et al.* proposed that synaptic weight regulation behavior could be simulated by manipulating the ion migration process in the memristor.[8]

The most significant advantage of memristors is that it has both non-volatile and the ability to perform matrix multiplication calculations in parallel. Matrix multiplication calculation is the core calculation in all kinds of neural networks. Therefore, memristors are considered to be devices with the characteristics of "memory computing integration" and have the potential to break the bottleneck of the traditional von Neumann architecture storage wall. Since memristors can greatly accelerate the forward inference process of neural networks, various research and applications of memristors in the field of neuromorphic computing are emerging. Wu *et al.* built an all-hardware composition of a store-and-calculate fusion computing architecture and deployed a convolutional neural network by integrating multiple memristor arrays to achieve recognition of handwritten digits.[9] Chen *et al.* revisited various properties of memristors and predicted the unobserved property, which is the pseudo-polarity reversibility property.[10]

Although memristors have great advantages in the field of neuromorphic computing, it is undeniable that they also have some drawbacks (non-ideal characteristics), such as nonlinearity, asymmetry, conductance drift, and fewer intermediate states,[11] which limit the further development of memristors in the field of neuromorphic computing. Researchers have compensated for the negative impacts of these drawbacks by improving the algorithms or designing special circuits.[12–17] However, the complexity has also been increased and brings additional energy consumption and time delays.

Everything has two sides, and memristors are no exception. It is interesting to study whether the non-ideal characteristics of memristors can become ideal in specific applications, which is the inspiration for this paper. Randomness and variation are the main tones among the many non-ideal characteristics of memristors. Genetic algorithm (GA) is a method that simulates the biological evolutionary process, using a computer language to describe phenomena such as genetic variation in biological evolution.[18–20] These phenomena are full of unpredictable variations, which are similar to the main tone of the non-ideal properties of memristors.

In this paper, the non-ideal characteristics of the memristor are used to restore the biological evolutionary behavior in GA, and the related circuit is designed, based on which the path planning algorithm based on the memristor network is implemented.

The rest of this paper is organized as follows. Section 2 introduces the theory related to memristors and GA. Section 3 describes the GA and path planning based on memristive networks. Section 4 contains the experimental examples and sim-

†Corresponding author. E-mail: huxf@swu.edu.cn

© 2023 Chinese Physical Society and IOP Publishing Ltd

ulation results. Finally, Section 5 outlines the conclusions.

2. Related theory

2.1. Memristive network

A typical memristor network consists of a memristor crossbar array and related circuits, as shown in Fig. 1.

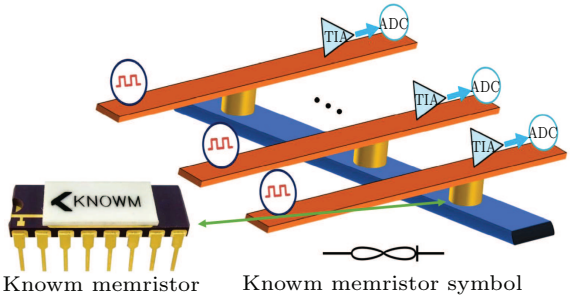


Fig. 1. Memristive network architecture diagram.

The memristors used in this paper are generated and manufactured by Knowm, which is a self-directed type of memristors.[21] The voltage pulses generated by the analog-to-digital converter (ADC) are able to change the conductance value of the memristor. The operation to increase the memristor conductance value is SET, and the operation to decrease the memristor conductance value is RESET. In addition, since the conductance value of the memristor can only express positive values, the characterization of a single message needs to be expressed using the conductance difference between the two memristors.[6] The transimpedance amplifier (TIA) and the digital-to-analog converter (DAC) are used to obtain the conductance value of the memristor. In addition, by converting the weights of the neural network to conductance values and combining them with Kirchhoff's law, the memristor crossbar array can perform matrix multiplication operations in parallel.

2.2. Genetic algorithm

GA is a heuristic algorithm inspired by Darwin's theory of natural evolution, which selects the best individuals to solve the problem by continuously updating iterations. The general process of GA is to encode and initialize the population for the problem, set the fitness function, and use crossover, selection and variation operators for evolution, as schematically shown in Fig. 2.

GA evaluates the different individuals in the population by means of a fitness function. A higher score means that the individual is better and also means that there is a greater chance of staying in the evolutionary process. In addition, genetic variation and crossover will result in the creation of new individuals in the population, which increases the diversity of individuals in the population. Through many iterations, the inferior individuals in the population will be gradually eliminated, and the superior individuals will be retained.

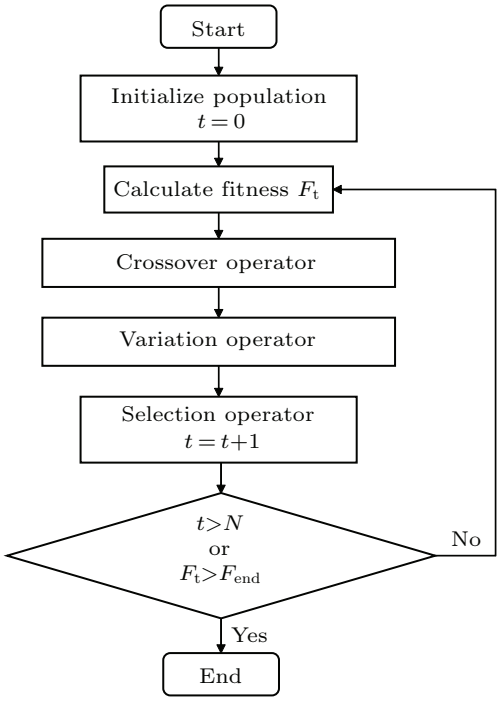


Fig. 2. Classical genetic algorithm process diagram.

3. Genetic algorithms and path planning based on memristive networks

The conductance value of the memristor can be programmed to change and be maintained for a long time, so the conductance value of the memristor can represent different information. In this paper, a 5×5 map is used as an example to map the different conductance values of the memristor to the coordinates in the map, and Fig. 3 shows how the conductance values of the memristor are mapped to the coordinates in the map. Each memristor in the 1×5 memristor array corresponds to a particular row in the map, by dividing the conductance values of the memristors into different levels (in this paper, the conductance values of the memristors are divided into five different levels), and the different levels correspond to the values of the lateral coordinates in the map.

The diversity of biological populations increases the probability of the emergence of good individuals in the population. In this paper, we simulate the diversity of biological populations by generating a large number of individuals through a random method. In the initialization process of the genetic algorithm, random pulses (random width, number, and amplitude) are applied to the memristors in the memristor network, making the conductance values of the memristors located at different levels, and the conductance values of the memristors represent a certain coordinate in the map, and the different coordinates are combined to form a path. It should be noted that the paths generated in the initialization phase are not necessarily connected, and the following equation is used to determine whether two coordinates are neighboring each

other on the map. If the two coordinates are not neighboring, it is necessary to insert a new coordinate between the two coordinates and repeat the determination using the equation until the path is in a connected state.

$$D = \max\left\{\text{abs}\left(x^{i+1} - x^{i}\right),\ \ \text{abs}\left(y^{i+1} - y^{i}\right)\right\}. \tag{1}$$

GA accompanies the evolution of biological populations, and meritocracy is an important criterion for the continuity of individuals in this evolutionary process. In this paper, different paths from the starting point to the end point correspond to different individuals in the biological population, and the different paths are evaluated by the fitness function, and the fitness function is given in the following equation.

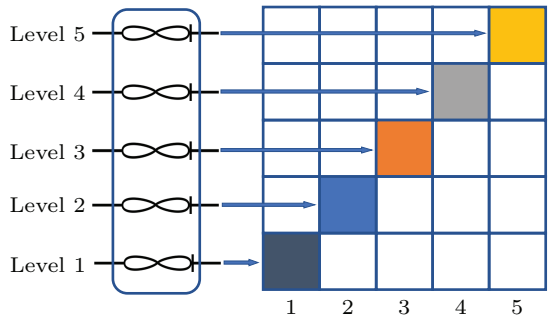


Fig. 3. Relationship between the conductance values of the memristor array and the map coordinate values.

Here, f_1 denotes the evaluation from the perspective of path distance, and a shorter distance means a better path; f_2 denotes the evaluation from the perspective of smoothness of the path, and a smoother path means a better path; α and β denote the weights of distance and smoothness, respectively. The fitness value of a path is taken into consideration in both aspects. In the iterative process, the fitness score of each individual will be calculated repeatedly, and the roulette algorithm is used to restore superiority and inferiority in biological evolution. A higher fitness means a higher probability that the individual will be retained in the iterative process.

Genetic variation is an important step in GA, which not only ensures the diversity of individuals but also gives the possibility for the emergence of even better individuals. Genetic variation possesses unpredictable and random properties, which are similar to the non-ideal characteristics of memristors. By applying random pulses to the memristor, it is able to make the conductance value of the memristor change randomly and thus generate different paths, which is consistent with the generation of new individuals by a mutation in traditional GA. We have

$$f_{\text{itness}} = \alpha \cdot f_1 + \beta \cdot f_2, \tag{2}$$

where f_1 and f_2 read

$$f_1 = \frac{1}{\sum_{i=1}^{N} a_n \sqrt{\left(x^{i+1} - x^{i}\right)^2 + \left(y^{i+1} - y^{i}\right)^2}}, \tag{3}$$

$$f_2 = \sum_{i=1}^{N-2} \arccos\left(\frac{b_i^2 + c_i^2 - a_i^2}{2 b_i c_i}\right). \tag{4}$$

The path planning algorithm based on the memristive network is shown in Table 1.

Table 1. The path planning algorithm based on the memristive network.

Algorithm 1 Path planning based on memristive networks
1: Initialize the conductance value of the memristive network
2: Path initialization based on conductance value
3: **for** iteration = 1 **to** N **do**
4: Randomly choose a memristor to apply random pulses
5: Generate new mutation paths
6: Calculate and record the fitness value
7: Iteration with Roulette Algorithm
8: **end for**
9: Find the best path according to the fitness value

Figure 4 shows the peripheral circuit structure of the path planning algorithm based on the memristor network. The circuit generates the specified pulses (width, polarity, number, and amplitude) by controlling the DAC, and the pulse stimulation causes a change in the conductance value of the memristor, and the current flowing through the memristor can be measured by the TIA and ADC, and the conductance value of the memristor can be obtained by using Ohm's law.

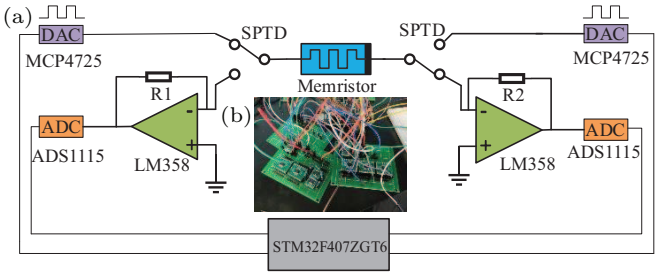


Fig. 4. (a) The circuit structure of the memristive network-based path planning algorithm. (b) Physical circuit system.

4. Experiment

The Knowm memristor mechanism depends on the movement of Ag^+ into the active layer of the device to change the resistance of the device, and the Ag^+ content in the channel determines the resistance of the device. In addition, the active layer of the Knowm memristor consists of Ge_2Se_3.[21]

Figure 5 shows the variation of the conductance value of the Knowm memristor under 30 SET pulses and 30 RESET pulses. The data for the tests are obtained by the circuit designed in Fig. 4.

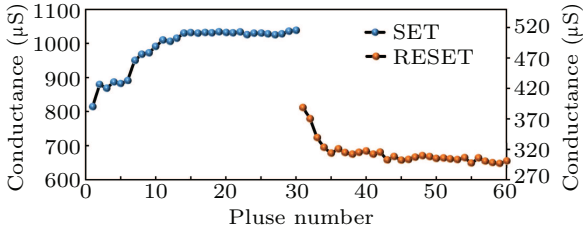


Fig. 5. Conductance changes of the memristor during 30 SET and 30 RESET pulse modulation.

Figure 6 shows the variation of the memristor conductance value under 500 random pulses (random width, number, and amplitude). Due to the non-ideal characteristics of the memristor, the variation of the memristor conductance value is completely random and unpredictable, which is the necessary condition for the memristor to be able to restore phenomena such as genetic variation in GA. The yellow dashed lines in Fig. 6 indicate the splitting lines of the different levels of the memristor conductance values.

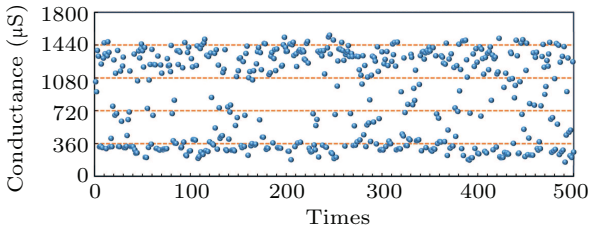


Fig. 6. Conductance changes of the memristor during 500 random pulses modulation.

The path planning in this paper takes a 5×5 map as an example, and before the experiment, the conductance value of the memristor is controlled at five different levels by applying different pulses to it. The best paths and the conductance values of the five amnesic resistors are shown in Fig. 7, which also illustrates that the peripheral circuit system constructed in this paper has the necessary conditions to find the optimal paths.

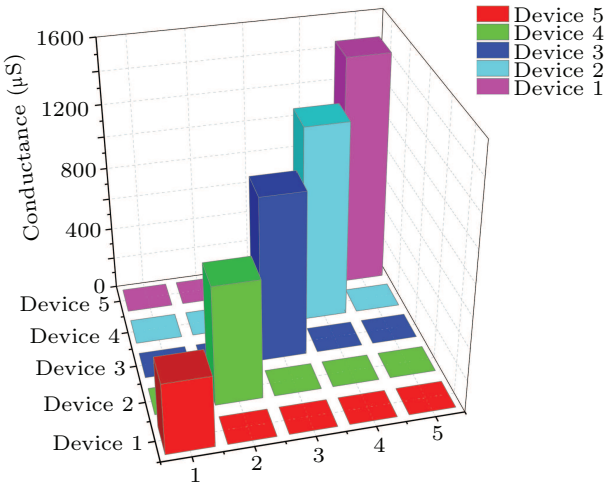


Fig. 7. The conductivity of the memristor network in the ideal path state.

The GA requires multiple iterations in the search for the optimal solution. We read the changes in the memristor conductance values during the iterations (four times in total) and plotted the corresponding paths.

In addition, the values of α and β in the fitness function are both 0.5.

Figure 8 shows the iterative process of the path with the starting and terminal points labeled. It should be noted that the above iterative process is plotted after reading the resistance value of the memristor through the circuit. During the iterative process, the paths gradually evolve from the initial complex to the simple direction, which shows that the path search algorithm based on the memristor network designed in this paper is effective.

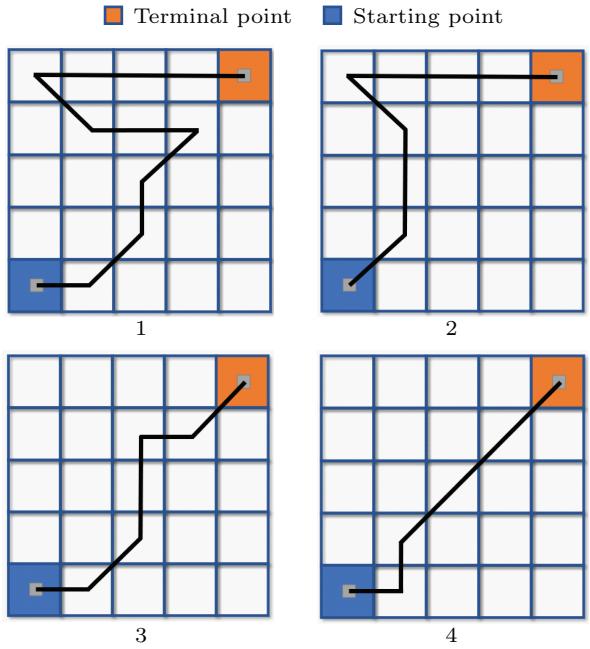


Fig. 8. Evolution of the coordinates corresponding to the memristor resistance value in the path planning algorithm.

5. Conclusion

In summary, we have studied the non-ideal characteristics of memristors, used the non-ideal characteristics of memristors to restore the basic operators in genetic algorithms, designed and implemented the relevant peripheral circuits, and implemented a path planning algorithm based on memristor networks on this basis. The simulation results show that the non-ideal characteristics of the memristor can simulate the biological evolutionary behavior in the GA, and the path planning algorithm based on the memristor network can search for a better path after several iterations. This shows that it is possible to make the non-ideal characteristics of the memristor into ideal characteristics by thinking backward. This study can provide a new way of thinking for the practical application of the non-ideal characteristics of the memristor.

Acknowledgements

Project supported by the National Natural Science Foundation of China (Grant Nos. 61976246 and U20A20227), the Natural Science Foundation of Chongqing, China (Grant No. cstc2020jcyj-msxm X0385), and the National Key R&D Program of China (Grant Nos. 2018YFB130660 and 2018YFB1306604).

References

[1] Chua L 1971 *IEEE Trans. Circuit Theory* **18** 507

[2] Strukov D B, Snider G S, Stewart D R and Williams R S 2008 *Nature* **453** 80

[3] Yoon J H, Wang Z, Kim K M, Wu H, Ravichandran V, Xia Q, Hwang C S and Yang J J 2018 *Nature* **9** 417

[4] Burr G W, Shelby R M, Sebastian A, *et al.* 2017 *Adv. Phys. X* **2** 89

[5] Yu S, Gao B, Fang Z, Yu H, Kang J and Wong H P 2013 *Adv. Mater.* **25** 1774

[6] Prezioso M, Merrikh-Bayat F, Hoskins B D, Adam G C, Likharev K K and Strukov D B 2015 *Nature* **521** 61

[7] Li C, Belkin D, Li Y, *et al.* 2018 *2018 IEEE International Memory Workshop (IMW)*, May 13–16, 2018, Kyoto, Japan, pp. 12–15

[8] Jo S H, Chang T, Ebong I, Bhadviya B B, Mazumder P and Lu W 2010 *Nano Lett.* **10** 1297

[9] Yao P, Wu H, Gao B, Tang J, Zhang Q, Zhang W, Yang J J and Qian H 2020 *Nature* **577** 641

[10] Chen L, He Z, Li C, Wen S and Chen Y 2020 *Int. J. Bifur. Chaos* **30** 2050172

[11] Xi Y, Gao B, Tang J, Chen A, Chang M F, Hu X S, Van Der Spiegel J, Qian H and Wu H 2020 *Proc. IEEE* **109** 14

[12] Yao P, Wu H, Gao B, *et al.* 2017 *Nat. Commun.* **8** 15199

[13] Cai Y, Tang T, Xia L, Cheng M, Zhu Z, Wang Y and Yang H 2018 *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)* January 22–25, 2018, Jeju Island, South Korea, pp. 117–122

[14] Ambrogio S, Narayanan P, Tsai H, *et al.* 2018 *Nature* **558** 60

[15] Li C, Li Y, Jiang H, *et al.* 2018 *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 27–30, 2018, Florence, Italy, pp. 1–4

[16] Ravichandran V, Li C, Banagozar A, Yang J J and Xia Q 2018 *Sci. China Inf. Sci.* **61** 1

[17] Wang Y, Wu S, Tian L and Shi L 2020 *Neurocomputing* **407** 270

[18] Katoch S, Chauhan S S and Kumar V 2021 *Multimedia Tools and Applications* **80** 8091

[19] Mirjalili S, Song D J, Sadiq A S and Faris H 2020 *Nature-Inspired Optimizers* Vol. 811 p. 69

[20] Zhi H and Liu S 2019 *J. Visual Commun. Image Represent.* **58** 495

[21] Campbell K A 2017 *Microelectron. J.* **59** 10