Double DQN for Chip-Level Synthesis of Paper-Based Digital Microfluidic Biochips

Fang-Chi Wu[1], Jian-De Li[2], Katherine Shu-Min Li[1], Sying-Jyan Wang[2], and Tsung-Yi Ho[3]
[1]Department of Computer Science and Engineering, National Sun Yat-Sen University
[2]Department of Computer Science and Engineering, National Chung Hsing University
[3]Department of Computer Science, National Tsing Hua University

***Abstract*—Paper-based digital microfluidic biochip (PB-DMFB) technology is one of the most promising solutions in biochemical applications due to the paper substrate. The paper substrate makes PB-DMFBs more portable, cost-effective, and less dependent on manufacturing equipment. However, the single-layer paper substrate, which entangles electrodes, conductive wires, and droplet routing in the same layer, raises challenges to chip-level synthesis of PB-DMFBs. Furthermore, current design automation tools have to address various design issues including manufacturing cost, reliability, and security. Therefore, a more flexible chip-level synthesis method is necessary. In this paper, we propose the first reinforcement learning based chip-level synthesis for PB-DMFBs. Double deep Q-learning networks are adapted for the agent to select and estimate actions, and then we obtain the optimized synthesis results. Experimental results show that the proposed method is not only effective and efficient for chip-level synthesis but also scalable to reliability and security–oriented schemes.**

Keywords—Paper-based digital microfluidic biochips, chip-level synthesis, Reinforcement learning, Double DQN

I. Introduction

Paper-based digital microfluidic biochips (PB-DMFBs) achieve "lab-on-paper" due to the paper substrate. The paper substrate makes PB-DMFBs more portable and cost-effective, easy-to-use, and less dependent on manufacturing equipment [1]-[3]. PB-DMFBs is useful in applications, including genomics, point-of-care diagnostics, and environment monitoring. Easy-to-use and low dependence on expensive manufacturing equipment allow efficient parallelization of biochemical experiments with PB-DMFBs even in a resources-limited region [1]-[3].

The manufacturing process of a PB-DMFB is outlined as follows [1]. First, the PB-DMFB design, including fluidic-level synthesis and chip-level synthesis, is determined by PC or cloud-based synthesis platform [4], [5]. Fluidic-level synthesis conducts resource scheduling, placement, and droplet routing for a given bioassay. Chip-level synthesis carries out control port assignment and conductive wire routing such that PB-DMFBs can control droplets. After the design phase, the PB-DMFB design is printed and surface-coated with dielectric and oil films. The final step is actuating sample and reagent droplets on the paper substrate with an integrated power system. Such a manufacturing process allows users to print out PB-DMFBs by themselves.

In chip-level synthesis of PB-DMFBs, the electrode array is designed to transport droplets. An electrode is activated by applying electric signal to the control port connected to the electrode through conductive wires. An electric field is generated at the activated electrode, so a nearby droplet will be attracted to the activated electrode. Heuristic methods for chip-level synthesis have been proposed [6], [7]. However, these methods tend to be inflexible, so they may not be able to deal with various requirements coming with the versatile applications. For example, diagnosis and fault tolerance require PB-DMFBs to be programmable so as to change the droplet routing [8]. The signal-based electrode addressing method [6] cannot be directly applied to achieve diagnosis and fault tolerance, since alternative droplet routing is not considered. On the other hand, some security-oriented applications require multiple designs in order to distribute, control, and protect the usage of PB-DMFB IPs [4], [9]. Therefore, a flexible chip-level synthesis method that is capable of generating multiple designs is necessary.

Double deep Q-learning network (DQN) is a reinforcement learning (RL) algorithm with Q-value estimations, which evaluates the action quality applied to environment [10]-[13]. Q-learning based algorithms have been applied to optimization problems [11], [13], [14]. In Q-leaning, an agent finds a series of actions with the maximum Q-values and obtains the optimized reward. The Q-learning process is iterative so that the agent can leverage experiences from the executed episodes to estimate and select high-quality actions.

In this paper, we propose the first RL based chip-level synthesis for PB-DMFBs. The contributions of this paper are summarized as follows.

- We adapt reinforcement learning, including an agent, an environment, actions, and experiences, to the chip-level synthesis in PB-DMFBs.
- Double DQN is adapted for an agent who selects and takes a series of actions to the biochip. Double DQN can tackle the potential overoptimistic estimations in RL.
- Non-Q-value action selection methods are adapted to provide better experiences in the early episodes, which avoids potential long converging process. Decay of Non-Q-value action selection methods is employed for the Q-value action selection to find comprehensive solutions in the middle and late episodes.

II. Preliminaries

A. Double DQN

Double DQN (deep Q-learning network) is a machine learning method based on reinforcement learning (RL) [10]-[13]. In the RL process, an agent selects and takes an action to the environment and then obtains the corresponding reward and experience as feedback. By collecting and replaying the experiences from the interactions to the environment, the agent can find a policy to gain better reward. Q-learning methods leverage Q-table to estimate the action quality instead of a model of the environment [10]. The agent in Q-learning repeats episodes which involve choosing an action a according to the current state s, taking action a, and observing the reward r and the new state s'. Such an iterative updating procedure converges to the optimal action function [10], [11].

978-3-9819263-5-4/DATE21/©2021 EDAA

Q-table is built for estimating action quality. However, the huge searching space of states and actions may result in difficulty of building Q-table. Furthermore, the delay between taking actions and the resulting rewards can be very long [11]. These challenges inspired the introduction of neural network to RL learning. Deep Q-learning is an RL learning method using a deep Q-learning network for estimations of action quality [11]. A deep Q-learning network is a multi-layered neural network which outputs the selected action with the given input state [11]. Deep Q-learning uses the same deep Q-learning network to select and evaluate actions, which results in overoptimistic value estimations away from the true optimal values [12], [13].

To tackle the overoptimistic estimations, van Hasselt et al. have proposed double DQN method [13]. By applying two estimators to Q-learning, the double DQN method decouples selection from evaluation. In other words, there are two deep Q-learning networks in the double DQN method. One determines the policy of selecting actions, and the other determines the action qualities.

B. Droplet Control in PB-DMFBs

Like conventional DMFBs, operations in bioassays (e.g., mixing, split, etc.) are carried out through transporting reagent and sample droplets along the preset paths [6], [7]. In order to transport droplets, an integrated power system activates electrodes printed on the paper substrate according to a given order. An activation sequence is the input signal sequence applied to a control port so that the connected electrode can generate electrical field. Three signal values may appear in a sequence: 1, 0, and X, which stand for logic high, logic low, and don't care values, respectively. A logic high value indicates that the electrode connected to the control port is activated to induce an electrical field at the electrode. A logic low value means the corresponding electrodes should not be activated according to the fluidic constraints [6], [7], which prohibits any undesired transportation of droplets. An X value can be assigned to either 1 or 0.

C. Constraints of Chip-Level Synthesis in PB-DMFBs

To avoid undesired transportations, the fluidic constraints and the electrical field interference constraint [6], [7] are abided by in the proposed method. The fluidic constraints, including static and dynamic fluidic constraints, impose sufficient distance between two droplets. In order to achieve this goal, it is required that the eight electrodes surrounding the activated electrode (i.e., forbidden region) should not be activated.

Electrical field interference is unique in PB-DMFBs since electrodes, conductive wires, and droplet routing are entangled in the same paper layer [6], [7]. If the activated conductive wire for a given droplet d_1 is so close to another non-target droplet d_2, an undesired transportation of d_2 may occur. Therefore, both activation time and location of the non-target droplet also have to be considered at the same time to prevent electrical field interference. To tackle electrical field interference, either the activation time condition or the location condition has to be eliminated.

III. Double DQN for Chip-Level Synthesis

In this paper, we propose a double DQN based method for chip-level synthesis of PB-DMFBs. The proposed method searches possible chip designs for the given bioassay to find the optimal solution that can be printed to perform the desired biochemical experiments. The problem formulation is given as follows.

Input: PB-DMFB specifications, including (1) chip size, (2) upper bound on control ports, and (3) droplet trajectories T for the bioassay.
Objective: (1) achieving 100% routability and minimizing manufacturing cost, including (2) wirelength and (3) number of used control ports.
Constraints: (1) fluidic constraints and (2) electrical field interference constraint.
Output: Chip-level synthesis results (conductive wire and electrode assignments).

A. Environment, Actions, and State

In reinforcement learning, an agent selects and takes a series of actions to an environment and obtains the experiences and rewards. As chip-level synthesis of PB-DMFBs, the environment is biochip layout including locations of electrodes. Electrodes to be routed are determined from the droplet trajectories. As long as a droplet passes through the location of an electrode, the electrode has to be routed in the following process. An action processes the electrode to be routed by assigning it to a clique. A clique consists of a control port and a set of electrodes whose activation sequences are compatible. Elements of a clique (including the electrodes and the control port) will be connected by conductive wires such that they can share the same activation sequences, which reduces the manufacturing cost significantly [6], [7].

When an action is applied to the environment, the reward to be returned is based on the information of the biochip layout and the taken action. For efficient learning, five features are extracted from a biochip layout as a state, including, the estimated wirelength, the number of electrodes in each clique, the number of overlapped bounding boxes, the number of fully overlapped bound boxes, and the average area for each electrode in each clique. The estimated wirelength is the routing cost to connect the electrodes and control ports, and it is calculated by the Manhattan distance between a pair of elements determined by the taken actions. The pair of element consists of the processed electrode of the taken action and the element with the shortest distance to the processed electrode.

The remaining features are extracted to address the routability issues. The number of electrodes in each clique may affect routability. To comply with the fluidic constraint, electrodes in a clique should be sparsely distributed in a chip. As a result, routing a clique with a large number of electrodes can be challenging. The bounding box of a clique is determined by the two elements with the longest distance. The region of the bounding box is the most possible region for wire routing. The numbers of partially overlapped and fully overlapped bounding boxes represent the congestion degree of the chip layout. The average area for each electrode in a clique is the ratio of bounding box area to the number of electrodes. Through extracting these features, the agent can learn the cost and congestion information from a chip layout. In addition to the five features, we also count the number of cliques to evaluate the state. A clique consists of electrodes controlled by a shared control port, so the number of cliques is equal to the number of used control ports.

B. Double DQN Synthesis Flow

The proposed double DQN synthesis flow is given in Fig. 1. For a given set of specifications, a fixed number (500 in our experiments) of episodes are performed to obtain the synthesis results and experiences. An episode involves interactions among the agent, the environment, and the experiences. The first step in an episode is initialization. After that, for each electrode to be routed, the agent selects and takes the action to the environment, and the corresponding new state and reward are returned from the environment. The states (including the original and the new one), the taken action, and the reward are store as an experience.

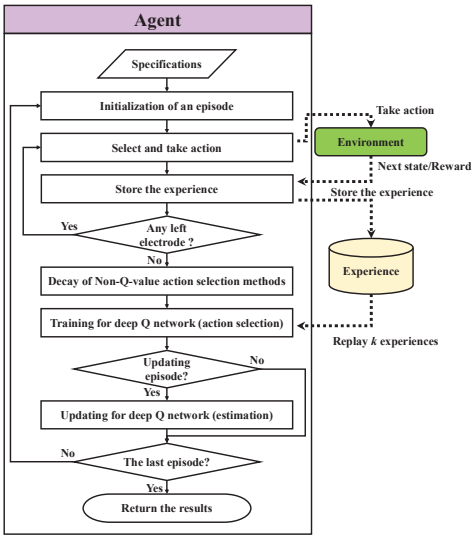


Fig. 1. Double DQN synthesis flowchart.

Once all the electrodes to be routed are processed, the agent deals with updates for action selections and the double DQN. For the action selection DQN, training is executed in each episode. The agent replays k experiences to train the action selection DQN for each episode, which enables the agent to perform the Q-value action selection through the trained DQN in the next episode. In contrast to the action selection DQN, the estimation DQN is updated at a fixed frequency (one update per 25 episodes). Such a periodic estimation DQN results in more stable and reliable learning [13].

The proposed double DQN leverages experience replay to train DQN. However, the inefficient experiences in the early episodes may result in long converging process. Thus, non-Q-value action selection methods are adapted to provide better experiences in the early episodes. Since the non-Q-value action selection methods do not consider routability issues caused by the taken actions, we need the Q-learning method to find comprehensive solutions with routability taken into account. Therefore, for the middle and late episodes, the probabilities of performing non-Q-value action selection have to be decreased. Therefore, decay of non-Q-value action selection methods is included in each episode to diminish the effect of non-Q-value action selection methods. More details about non-Q-value action selection methods and their decay will be discussed in III.C.

After performing all the episodes, the agent returns all the results. Please note that the agent in each episode constructs a possible chip-level synthesis result, which makes it possible to provide multiple high-quality (i.e., reward) designs for the given bioassay. This implies that the proposed DQN method is scalable to security-oriented applications that need multiple designs to control or protect the usage of biochip IP [4], [9].

C. The Action Selection by the Agent

The action selection flowchart is presented in Fig. 2. The first step is to exclude actions that violate any constraint. Any actions involving the forbidden region of the fluidic constraints will not be considered. For the electrical field interference constraint, the agent checks the following two conditions. (1) The action results in wire routing is close to any other droplet location, which can be checked by whether the bounding box caused by the action are fully covered by the forbidden regions or not. (2) The wire routing is activated when another droplet is nearby. If the above conditions are satisfied, the action will be excluded. After that, four action selection methods can select an action.

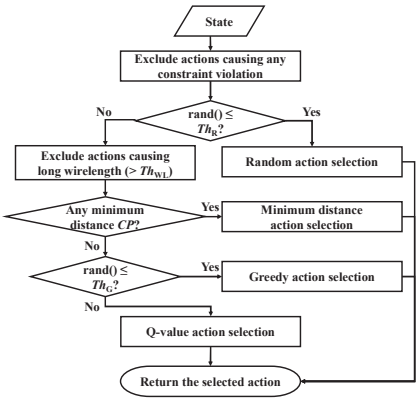


Fig. 2. The action selection flowchart of the agent.

As mentioned above, non-Q-value action selection methods are adapted to provide better experiences in the early episodes. Random action selection provides diversified experiences to the agent, which allows the agent to evaluate quality of the following actions. In PB-DMFBs, electrodes at chip boundary have the minimum distance to one of the control ports (*CP*s). Here, the minimum distance indicates the shortest Manhattan distance for all the pairs of electrodes and control ports. The agent should check whether the processed electrode has at least one control port with the minimum distance. If so, the agent selects one of the actions involving such a control port. A greedy action selection method is applied to select one of the qualified actions. The greedy action selection method selects the action that produces the control circuit (i.e., connected electrodes and the control port) with the shortest Manhattan distance to the processed electrode. If there is no such control circuit, the method selects the control port with the shortest Manhattan distance to the processed electrode will be selected.

Decay of the non-Q-value action selection methods is included in each episode. We leverage thresholds to diminish the effect of the non-Q-value action selection methods in the middle and late episodes. In Fig. 2, rand function returns a value ranging from 0 to 1. By decreasing the random selection threshold Th_R in each episode, the use of random action selection method is diminished until Th_R is no larger than 0.1.

TABLE I. EXPERIMENTAL RESULTS

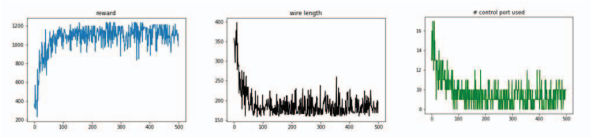
Circuit	Statistics			[6]		[9]		The proposed double DQN				
								States (agent)			Maze routing	
	Size	*#E*	*#CP*	*WL*	$\#CP_u$	*WL*	$\#CP_u$	WL_s	$\#CP_s$	*CPU(s)*	*WL*	$\#CP_u$
amino-acid-1	6x8	20	24	187	8	171.9	14.8	154	7	492.24	184	7
amino-acid-2	6x8	24	22	210	10	224	18.5	180	8	453.69	248	8
protein-1	13x13	34	48	411	14	328.5	15.2	370	13	646.23	441	13
protein-2	13x13	51	46	606	27	678.4	29.8	558	21	903.86	729	21

Initially, Th_R is 1, which indicates the action selection method in the first episode is the random selection method. Then, Th_R is multiple by a decay factor (0.965 in the experiments) in each episode. The probability of applying the greedy action selection method is reduced by decaying greedy selection threshold Th_G in the same way.

Q-value action selection is based on the deep Q-learning networks. The deep Q-learning network is a neural network with two fully connected layers. When the agent inputs a state representing the biochip layout to this network, the agent obtains Q-values for each action. As a result, the agent can select the action with the largest Q-value as the selected action [12], [13].

IV. EXPERIMENTAL RESULTS

The proposed double DQN is implemented with Python 3.7.6, TensorFlow 2.2.0, and Open AI Gym. It is executed on a PC with 1.6GHz Intel core i5-8250U CPU and 4GB RAM. In this paper, we assume that the space between two adjacent electrodes can accommodate three wires [4], [6]-[9]. The parameters for all the experiments are listed as follows. Four bioassays (listed in Table I) are used in the experiment. For each circuit, 500 episodes are performed. In each episode, k is 5, which means that the agent replays five experiences to train the action selection DQN. The estimation DQN is updated every 25 episodes by duplicating the current parameters of the action selection DQN. The decay factor of Th_R and Th_G are 0.965 and 0.985, respectively. Th_R and Th_G are multiplied by their respective decay factor until they are no larger than 0.1. Threshold Th_{WL} is used to exclude actions causing long wirelength as show in Fig. 2. We set Th_{WL} to be half of the long side of the biochip.



(a) (b) (c)

Fig. 3. Rewards of amino-acid-1 with 500 episodes. (a) Reward. (b) Wirelength. (c) Number of used control port.

Experimental results are summarized in Table I. *Size* is the size of the electrode array. *#E* is the number of electrodes to be routed. *#CP* is the upper bound on control ports. *WL* and $\#CP_u$ are wirelength and the number of used control ports. DRM [9] provides multiple designs for a given bioassay, and *WL* and $\#CP_u$ for DRM are the average numbers of all designs. WL_s and $\#CP_s$ provided by the agent are state estimated by the agent in the proposed method. *CPU* indicates the run time in seconds for the 500 episodes. We adapt Maze routing algorithm to process routing, and the optimized results for the number of used control ports is also included in Table I. In every case the Maze routing is finished in 0.5 seconds. All the results in Table I achieve 100% routability.

Experimental results show that the proposed double DQN is efficient for chip-level synthesis. The designs with the optimized number of used control ports usually have more routability issues during routing. As shown in Table I, taken actions from the agent can deal with the routability issues. Rewards and states of amino-acid-1 for the 500 episodes are shown in Fig. 3. The Q-learning process converges around the 100th episode. After that, the agent provides possible solutions with high rewards in the remaining episodes.

V. CONCLUSION

Paper-based digital microfluidic biochip technology is a promising solution for biochemical applications. In this paper, we proposed the first reinforcement learning based chip-level synthesis for PB-DMFBs. Experimental results have shown that the proposed method is not only effective and efficient for chip-level synthesis but also scalable to reliability and security–oriented schemes.

REFERENCES

[1] H. Ko, et al., "Active digital microfluidic paper chips with inkjet-printed patterned electrodes," *Advanced Materials*, Vol. 26, No. 15, pp.2335-2340, 2014.

[2] P. Wang, et al., "Development of a paper-based, inexpensive, and disposable electrochemical sensing platform for nitrite detection," *Electrochemistry Communications*, 74–78, 2017.

[3] N. Ruecha, et al., "Paper-based digital microfluidic chip for multiple electrochemical assay operated by a wireless portable control system." *Advanced Materials Technologies*, 2.3, 2017.

[4] J.-D. Li, et al., "Watermarking for Paper-Based Digital Microfluidic Biochips," accepted by *IEEE Internation Test Conference in Asia*, 2020.

[5] T.-M. Tseng, et al., "Cloud Columba: Accessible Design Automation Platform for Production and Inspiration." in *Proc. IEEE/ACM International Conference on Computer-Aided Design*. 2019.

[6] J.-D. Li, et al., "Congestion-and timing-driven droplet routing for pin-constrained paper-based microfluidic biochips," in *Proc. IEEE Asia and South Pacific Design Automation Conference*, 2016.

[7] Q. Wang, et al., "Integrated Control-Fluidic CoDesign Methodology for Paper-Based Digital Microfluidic Biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39.3 613-625, 2019.

[8] J.-D. Li, et al., "Test and diagnosis of paper-based microfluidic biochips," in *Proc. 34th IEEE VLSI Test Symposium*, pp.1-6, 2016.

[9] J.-D. Li, et al., "Digital rights management for paper-based microfluidic biochips." in *Proc. IEEE 27th Asian Test Symposium*, 2018.

[10] R. Sutton and A. Barto. Reinforcement Learning: An Introduction, *MIT Press*, 1998.

[11] V. Mnih, et al., "Playing atari with deep reinforcement learning." *arXiv preprint arXiv*:1312.5602, 2013.

[12] H. van Hasselt, Double Q-learning, in *Proc. Advances in Neural Information Processing Systems*, 23:2613–2621, 2010.

[13] H. van Hasselt, A. Guez, and D. Silver, Deep reinforcement learning with double q-learning, *arXiv preprint arXiv*:1509.06461, 2015.

[14] C.-Y. Chen and J.-L. Huang, "Reinforcement-Learning-Based Test Program Generation for Software-Based Self-Test," in *Proc. IEEE 28th Asian Test Symposium*, 2019.