

Evolving Spike-Train Processors

Juan Liu and Andrzej Buller

ATR Human Information Science Laboratories
2-2-2 Hikaridai, Keihanna Science City, Kyoto 619-0288 Japan
{juanliu; buller}@atr.jp

The research described in this paper was motivated by the idea to process purposefully given spike-trains using a cellular automaton (CA). CAs have three attractive features, namely massive parallelism, locality of cellular interactions, and simplicity of basic components (cells). However, the difficulty of designing a CA for a specific behavior causes limited interest in this computational paradigm. Automating the design process would substantially enhance the viability of CAs. Evolving CAs for purposeful computation is a scientific challenge undertaken to date by, among others, Mitchell *et al.* [1], Sipper *et al* [2] and de Garis *et al* [3].

In previous work [4], we designed a special 2-dimensional cellular automaton, called qCA, which can be used for purposeful manipulations on binary time-series (spike-trains). The proposed procedure of qCA synthesis is to decompose the problem into logic synthesis and routing. As for logic synthesis, we evolve a graph as a half-product for the qCA synthesis, called Pulsed Para-Neural Networks (PPNN). PPNN consists of functional nodes and directed edges representing pure delays, in which each node returns a pulse at clock t if it received one and only one pulse at clock $t-1$. The routing is a conversion of the obtained PPNN into an initial state of a qCA using a relatively trivial heuristics.

The desired PPNN is not assumed to be evolved from scratch. In this paper, we described an evolutionary method to find unknown delays of a given scheme so that the network could produce desired spike-trains. We investigated the scheme shown in Fig.1(a) as a test-bed of our method. The chromosomes of GA are represented as $V_i = (D_1, \dots, D_n) \in R^n$, where n is the number of delays to be found. The vectors (genotype) are converted into binary spike-trains (phenotype) via PPNN and then evaluated by a fitness function that compares the similarity of obtained and desired spike-trains. However, the fitness landscape is flat and punctuated with high spikes. And the small peaks around the global optimal point make the problem even harder. The idea to fuzzify and shift output signals resulted in a breakthrough. We converted the binary spike-trains produced by the evolved PPNNs into fuzzified time-series to differentiate spike-trains of the same Hamming distance. Then, we shifted the obtained spike-trains and took the best-fit value as the fitness of the chromosome so that the fitness value could be more consistent with the similarity of two spike-trains. The modified genotype-to-phenotype mapping procedure provided a smoother and more informative fitness landscape. Similar to memetic algorithms [5], our method is based on Lamarckian evolutionary process combining GA with a hill climbing technique to accelerate the convergence of the evolution. The experiment result is presented in Fig.1 (b), which confirms that the proposed approaches could improve the performance of the algorithm. It remains unclear whether additional fuzzification of signals circulating inside PPNN, or the fuzzification of node function, would result in higher evolvabil-

