A Modified Pulse Coupled Neural Network for Shortest Path Computation

Yongsheng SANG¹,²,†, Zhang YI³

¹ School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
² School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China
³ Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu 610065, China

Abstract

Shortest path computation is a classical combinatorial optimization problem. Neural networks, as a class of effective optimization approaches, have been used for processing path optimization problems for a long time. Pulse Coupled Neural Network (PCNN) is a very different neural network, which has also been proposed to compute shortest paths in recent years. In some existing PCNN models, the computational complexity is closely related to the length of actual shortest path. As a result, these PCNNs suffer from high computational cost for very long paths. This paper proposed a new PCNN model, called Dual Source PCNN (DSPCNN), which can improve the computational efficiency of PCNNs for shortest path problems. Two Autowaves produced by the DSPCNN: one comes from the Source neuron, and the other comes from the Goal neuron. Once the Autowaves from the two firing sources meet, the DSPCNN stops, and then the shortest path is found by backtracking the two Autowaves. Experimental results show that the DSPCNN is more efficient than some existing PCNNs.

Keywords: Dual Source Pulse Coupled Neural Network; Shortest Path; Autowave; Search Space

1. Introduction

The computation of shortest path is a well-known problem in network analysis, such as road networks in geographic information systems (GIS)[1], computer networks in modern communication systems[2], etc. In real GIS applications, an often demand is to query shortest paths between different locations on a road network. In some cases, the shortest path usually needs to be determined in real time. For example, the driver of ambulance in a hospital should determine the shortest path quickly to ensure the safety of patients. Moreover, large road networks are often involved in real applications, and the determination of shortest paths on large networks can be computationally very intensive. The same situation also exists in modern communication networks. Thus, it is necessary to study faster shortest path algorithms.

Related work. There are two main kinds of shortest path problems. One is single source shortest path problem, which is typically solved by the famous Dijkstra algorithm[3]. The other is the all source shortest path problem, which is due to Floyd algorithm[4]. The two algorithms can produce global optimization solution with an acceptable computational cost. However, they become inefficient in real applications...
involving large scale networks. Some new variations of shortest path problems have also been studied in recent years, for example, dynamic algorithm[5] for processing traffic changes, fuzzy shortest path algorithm[6], etc.

Neural networks are a class of critical methods to process complex problems, such as combinatorial optimization problems. Among them, Pulse Coupled Neural Network is a very active neural network in recent years. Based on the phenomena of synchronous pulse bursts in the cat visual cortex, Echhorn et al introduced the linking field network model[7]. When some modifications were introduced to the linking field network, a simplified PCNN is proposed for image processing. The PCNN has already been applied to many fields, such as image recognition[8] and motion matching[9]. Caulfield et al first introduce PCNN to the maze problem[10]. Their work is a clever approach to a difficult problem. The PCNN is modified so that the output pulses decay in time. The maze is explored by the PCNN and the shortest solution path is found by backtracking the decayed pulses. However, many neurons are needed in large networks since one pulse of the coupled neuron corresponds to a unit length of path. Reference [11] reported a delay PCNN model to search the shortest paths. Reference [12] and [13] proposed another two PCNN models to compute shortest paths. These modified PCNN models need fewer neurons than Caulfield's approach. However, the computational complexity of [11, 12, 13] relates to the length of the actual shortest path. Thus theses models usually involve much search space in some large scale applications. In order to meet the real-time demand in some real applications, this paper proposes a faster PCNN model, which can improve the computational efficiency significantly.

The rest of this paper is organized as follows. Section 2 gives some preliminaries. The DSPCNN model is described in Section 3. The algorithm is given in Section 4. Experimental results are given in Section 5. Conclusions are drawn in Section 6.

2. Preliminaries
In this paper, we mainly discuss point-to-point shortest path problem. The input to the preprocessing stage is an undirected graph \( G = (V, E) \) with \( n \) vertices and \( m \) edges, and nonnegative lengths \( \ell(e) \) for each edge \( e \). Another two inputs are a Source node \( s \) and a goal node \( g \). The goal of the proposed algorithm is to find a shortest path from \( s \) to \( g \). Let \( \text{dist}(s, g) \) denote the shortest-path length from \( s \) to \( g \) with respect to \( \ell \). In general, \( \text{dist}(s, g) = \text{dist}(g, s) \).

3. The DSPCNN Model
In this paper, to compute the point-to-point shortest path more efficiently, a Dual Source PCNN (DSPCNN) model is proposed. Unlike some existing PCNN models for solving the shortest path problem, this model can produce two Autowaves from two different firing sources. At \( t=t_0 \), the Source neuron and the Goal neuron fire and emit pulses simultaneously. Then, the two Autowaves propagate in parallel by their neighboring neurons at next instant till they meet together. In order to distinguish the two Autowaves, the Autowave propagating from Source neuron is denoted as \( P_s \), and the Autowave propagating from Goal
neuron is denoted as P_i. If a neuron fires on the stimulating of P_s Autowave, it outputs P_s pulses. If a neuron fires on the stimulating of P_g Autowave, it outputs P_g pulses. If a neuron fires on the stimulating of both P_s and P_g pulses, it indicates that the two Autowaves meet and the model should stop. In order to realize the pre-mentioned function, some modifications are introduced to the traditional PCNN model, which are shown in Fig.1.

Fig.1 shows the architecture of a single neuron N_i. A DSPCNN neuron consists of three parts: the receptive field, the modulation field, and the pulse generator. The receptive field contains two compartments: the feeding part F_i and the linking part L_i. In this model, the feeding input F_i is initialized as an external input I_i. The linking input is computed by a logical OR fashion, see Eqn (2), which promises that a neuron can receive linking input L_i = 1 if anyone of its neighbors fires. In the equation, N(i) denotes the maximal index id for the neighboring neurons of neuron N_i.

\[
F_i = I_i \quad \text{ (1) }
\]

\[
L_i = \bigcup_{k=1}^{N(i)} Y_k \quad \text{ (2) }
\]

In the modulation field, unlike some existing PCNN model for shortest path problem, the generating mechanism of the internal activity for DSPCNN neurons is given by Eqn (3). As can be seen, once a neuron receives stimulating from its neighbors, its internal activity would keep increasing by a constant \( \Delta U \). When the internal activity value of a neuron increases over its threshold, the neuron will fire and emit pulses, see Eqn (4). Because two Autowaves propagate in the network, the equation differentiate that the neuron is stimulated by which Autowave.

\[
U_i[t] = U_i[t-1] + \Delta U \left( \text{step}(L_i[t] \times F_i) \cup \text{step}(U_i[t-1]) \right) \quad \text{ (3) }
\]

In the modulation field, unlike some existing PCNN model for shortest path problem, the generating mechanism of the internal activity for DSPCNN neurons is given by Eqn (3). As can be seen, once a neuron receives stimulating from its neighbors, its internal activity would keep increasing by a constant \( \Delta U \). When the internal activity value of a neuron increases over its threshold, the neuron will fire and emit pulses, see Eqn (4). Because two Autowaves propagate in the network, the equation differentiate that the neuron is stimulated by which Autowave.
4. Computing Shortest Path Using DSPCNN

To compute shortest path for networks, whether it is true road network or artificial network, at first they are all transformed to a graph for further processing. For example, to process real road networks, each location is transformed into a graph node, and each segment of road is mapped into an edge in graph. The next step is to map the graph into DSPCNN model. Each node in the graph corresponds to a DSPCNN neuron, and each edge associated with a link between neurons. The cost of an edge can be viewed as an external input for the two neurons connected by the edge. In this paper, the created DSPCNN has the same topology structure with the original graph.

At time $t = 0$, Source neuron and Goal neuron fire simultaneously. Then the two Autowaves $P_s$ and $P_g$ from the two firing sources propagate to their neighbors. A variant Meeting is used to determine whether the two Autowaves meet together, and the meeting neuron is denoted by $N_m$. If $N_j$ fires on the stimulation of $N_i$, we call neuron $N_i$ is the precursor of neuron $N_j$.

A full algorithm of computing shortest path using DSPCNN model is given as follows:

Step1. Create a DSPCNN network according to the topology structure of the original network (graph).

Step2. Initialize the DSPCNN. Set the Source node $N_s$ and the Goal node $N_g$. Set $\Delta U$ as the minimal edge length. Set $V_\Theta$ as a very large constant, for example $V_\Theta = 100 \times m \times \max(\ell(e))$ (m is the number
of edges). External input $\ell(e_{ki})$ is set to the actual cost of the edge. For each neuron, we set $I_i = 1$, $Y[0] = 0$, $L_i[0] = 0$, $U_i[0] = 0$, $Y_i[0] = 0$, $\Theta_i[0] = V_{th}$, $\text{Meeting} = 0$.

**Step 3.** Run the network. For $N_s$ and $N_e$, set $Y_{N_s}[0] = 1$ and $Y_{N_e}[0] = 1$.

**Step 4.** For each stimulated neuron $N_i$,

1) Compute $L_i[t]$ according to Eqn (2);

2) Compute $U_i[t]$ according to Eqn (3);

3) Compute $Y_i[t]$ according to Eqn (4);

4) Compute $\Theta_i[t]$ according to Eqn (6) and update $\Theta_i[t]$ according to Eqn (5);

If $Y_i(t) == 1$ then record its precursor. If two Autowaves meet on the neuron, then set $N_m = N_i$ and $\text{Meeting} = 1$.

**Step 5.** Repeat Step 4 until condition $\text{Meeting} = 1$.

**Step 6.** Backtrack from $N_m$ to $N_s$ and $N_e$, combine the two paths and return it.

5. Experiments

**5.1. Illustrative Example**

At first, an illustrative example is given to show the effect of the proposed DSPCNN model. The PCNN in [13] is also tested by the query for comparison. A real road network dataset is employed in this experiment. The dataset is City of Oldenburg (OL) Road Network from [14], called OL dataset, which includes 6105 nodes and 7034 edges. We used a sample path query pair from node 1 (denote by S) to node 4300 (denoted by T) to test the search space and running time of the two models. This is a typical long path in the road network, which goes through diverse complex blocks including dense and sparse ones.
The result is shown in Fig. 2. The search result of the method [13] is shown in Fig. 1(a). As it can be seen, for searching this sample path, this method visited all of the nodes in the network. This is because the computational cost of the PCNN model in [13] is related to the path length. From the Fig. 1(b), we can see that the proposed DSPCNN only visited 2494 nodes for searching the path. Clearly, the search space of DSPCNN is far less than the PCNN in [13]. For search space, DSPCNN achieved 59.15% improvement to the method [13]. To search the path, the PCNN in [13] took 3.03 seconds, and DSPCNN took 0.36 second. As a result, DSPCNN achieved 88.12% improvement of search time. On the search quality, DSPCNN returned a shortest path same as that of method [13]. Thus, this experiment shows that DSPCNN can find shortest path with less search space, and it is a more efficient model than the PCNN in [13].

5.2. Performance Comparison

We also conduct a series of experiments on the OL dataset to evaluate the average search space and
running time of the DSPCNN. The DSPCNN and PCNN in [13] are both tested by 500 randomly generated queries of shortest paths. The comparison result of search space is shown in Fig.3. In the figure, for each query pair, the search space of DSPCNN is far less than that of PCNN in [13]. Fig.4 gives a comparison of running time between DSPCNN and the method in [13]. We record the CPU time for the 500 times of queries, and plot the time curves in ascending order of path length. From the figure, we can see that the DSPCNN is always faster than PCNN in [13]. The average query time of PCNN in [13] is 1.74 seconds, while the average response time of DSPCNN is 0.67 second. For the search space, the average number of the visited nodes of method [13] is 4658, and that of DSPCNN is 1276. Clearly, the proposed DSPCNN achieves higher computational efficiency than the method in [13]. In DSPCNN, because two Autowaves propagate in parallel, each Autowave travels half of the shortest path when they meet. As a result, the computational cost of this model only relates to half of the length of the real shortest path. That’s why this model achieves higher computational efficiency than the method [13].

6. Conclusions

This paper proposes a dual source PCNN model to compute shortest paths. By use of two Autowaves from two firing source (Source node and Goal node) propagating in parallel, the computational cost only relates
to half of the returned shortest path, and thus the proposed DSPCNN achieves higher efficiency than some traditional PCNN model for shortest path computation. Experiments show that the proposed method can search the global shortest path and involve lower search space, which can save running time significantly.

Acknowledgement

This work was supported by the National Science Foundation of China under Grants 60970013 and 60931160441.

References