

Dynamic Shortest Path Algorithm in Stochastic Traffic Networks Using PSO Based on Fluid Neural Network

Yanfang Deng, Hengqing Tong

School of Science, Wuhan University of Technology, Wuhan, China
Email: dengyf@whut.edu.cn, tonghq2005@whut.edu.cn

Received March 23rd, 2010; revised June 23rd, 2010; accepted September 10th, 2010

ABSTRACT

The shortest path planning issue is critical for dynamic traffic assignment and route guidance in intelligent transportation systems. In this paper, a Particle Swarm Optimization (PSO) algorithm with priority-based encoding scheme based on fluid neural network (FNN) to search for the shortest path in stochastic traffic networks is introduced. The proposed algorithm overcomes the weight coefficient symmetry restrictions of the traditional FNN and disadvantage of easily getting into a local optimum for PSO. Simulation experiments have been carried out on different traffic network topologies consisting of 15-65 nodes and the results showed that the proposed approach can find the optimal path and closer sub-optimal paths with good success ratio. At the same time, the algorithms greatly improve the convergence efficiency of fluid neuron network.

Keywords: Particle Swarm Optimization, Fluid Neuron Network, Shortest Path, Traffic Networks

1. Introduction

In recent years there has been a resurgence of interest in the shortest path problem in various transportation engineering applications [1]. In a distributed route guidance system, an in-vehicle computer is commonly used to calculate the optimal route in a large traffic network. Typically the recommended routes must be found within a very short time period (e.g., a few seconds). In a real-time automated vehicle dispatching system, new routes and schedules must be identified within a reasonable time after a customer requesting a service. Because the travel times are the basic input to the real-time routing and scheduling process and are dynamic in most urban traffic environments, there is an implicit requirement to use a minimum path algorithm repeatedly during the optimization procedure.

In the above applications, the shortest path problem, which is one of key technologies of distributed route guidance system, concerns with finding the shortest path from a specific origin to a specified destination in a given network while minimizing the total distance, time or cost associated with the path. This problem has been studied extensively in the fields of computer science, operation research, and transportation engineering [2] and so on.

The well-known algorithms including the Bellman's dynamic programming algorithm [3], the Dijkstra algorithm [4] and Bellman-Ford successive algorithm [5], are referred to as the standard shortest path algorithms. However, these traditional algorithms have major shortcomings: firstly, they are not suitable for networks with negative weights of the edges; secondly, the algorithms search only for the shortest route, but they cannot determine any other similar or non-similar short routes; thirdly, they exhibit high computational complexity. Therefore, the optimal shortest path algorithms tend to be too computationally intensive for real-time one-to-one applications in realistic traffic networks. Artificial neural networks (ANNs) have been examined to solve the shortest path problem relying on their parallel architecture to provide a fast solution [6,7]. However, ANN approach has several limitations such that they are less adaptable to topological changes in the network graph including the cost of the arcs. Moreover, the ANNs do not consider sub-optimal paths. Among other approaches for this problem, the powerful evolutionary programming techniques such as genetic algorithm (GA) [8] and tabu Search [9,10], which shows better performance compared to those of ANN approach and overcome the limitations mentioned

above, have considerable potential to be investigated in the pursuit for more efficient algorithms. PSO is such an evolutionary optimization technique, which can solve most of the problems solved by GA with less computation cost [11,12]. The most attractive feature of PSO is that it requires less computational bookkeeping and, generally, a few lines of implementation codes. Because of the specific algorithmic structure, PSO has been mainly applied to many continuous optimization problems with few attempts for combinatorial optimization problems [13-15]. Unlike the GA, the PSO algorithm has no complicated evolutionary operators such as crossover and mutation. In the PSO algorithm, the potential solutions, called as particles, are obtained by “flowing” through the problem space by following the current optimum particles. Generally speaking, the PSO algorithm has a strong ability to find the most optimistic result, but it has a disadvantage of easily getting into a local optimum. After suitably modulating the parameters for PSO algorithm, the rate of convergence can be speeded up and the ability to find the global optimistic result can be enhanced. The PSO algorithm’s search is based on the orientation by tracing P_b that is each particle’s best position in its history, and tracing P_g that is all particles’ best position in their history; therefore, it can rapidly arrive around the global optimum. However, because the PSO algorithm has several parameters to be adjusted by empirical approach, if these parameters are not appropriately set, the search will become very slow near the global optimum.

In this paper, the PSO algorithm based on the FNN model to search for the shortest path in stochastic traffic networks is proposed for the first time to our best knowledge. The technique makes full use of their advantages and uses the PSO algorithm to do global search in the beginning of stage, and then uses the FNN algorithm to do local search around the global optimum P_g . In particular, this hybrid algorithm will be used to train the FNN weights for function approximation and classification problems in convergent speed and generalization performance. Due to the fast global search feature of PSO, it improves greatly the efficiency of the convergence of the fluid neuron network, and decreases greatly the computation time of optimization path.

The paper is organized as follows. In Section 2, FNN model is introduced and briefly discussed. PSO algorithm and the particle encoding mechanism to solve shortest path problem in the traffic networks is presented in Section 3. The results from computer simulation experiments are discussed in Section 4. Section 5 concludes the paper.

2. FNN Model in the Traffic Networks

FNN [16], which based on continuous Hopfield neural network, has some merits, for example, its convergent process is as clear as the equilibrium process of fluid; its stable state is unique, which is determined by both the flow rate and the invariability in volume that the potential inside neurons must satisfy, corresponding to the Kirchoff’s node law in electrical circuits and the conservation properties of liquid in volume respectively. Moreover, it is unnecessary for FNN to solve the optimal problem by minimizing the energy function. Since the traffic network is very similar to FNN, it is reasonable to utilize FNN to handle the shortest path problem in traffic networks.

FNN has audio-visual properties of fluid as shown in **Figure 1**. The model is expressed as follows [17]:

$$\begin{aligned} \frac{du_i}{dt} &= \sum_{j \neq i} T_{ij} (s_j - s_i) - 2 |T_{ii}| u_i + I_i \\ T_{ii} &= - \sum_{j \neq i} T_{ij} \quad i, j = 1, 2, \dots, N \\ T_{ij} &= T_{ji} \geq 0 \quad i \neq j \\ s_i &= g(u_i) = a \cdot \left(\frac{1}{1 + e^{-bu_i - c}} \right) \end{aligned} \quad (1)$$

where each neuron is viewed as a vessel containing u-fluid, $u_i, s_i, -2 |T_{ii}| u_i$ and I_i is the volume, the level, the leakage and the external flow rate poured into ($I_i > 0$) or out ($I_i < 0$) of u-fluid in vessel i , respectively. The nonlinear function g characterizes the shape of the vessel. T_{ij} represents the capacity of the flow between both the vessels i and j , as a pipe connected between both the vessels and its diameter is controlled by the value of T_{ij} . $T_{ij} (s_j - s_i)$ is the actual flow rate of u-fluid from vessel j to vessel i , whose sign indicates the directions. When the system is closed, the system will reach a stable state, and then the total volume of the u-fluid in the system will be constant. Evidently FNN is quite similar to the traffic network. Certain amount of traffic flow going through the traffic network can be viewed as the same amount of fluid flowing through the vessels, that is, the neurons. Fluid naturally flows from a higher level to a

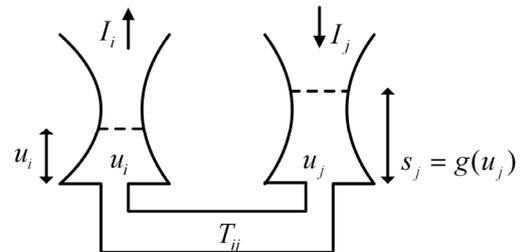


Figure 1. The fluid neural network system[16].

lower level, which is also the fundamental property of u-fluid. When iterations are enough, the height of every vessel is unchangeable and the system reaches a stable state. Of all links connected with a neuron the link having the largest volume of flow is the most probable route chosen by the fluid. By this way, from the Original node to the Destination node, the route with the largest volume of flow is the shortest path. A traffic network consists of nodes and links. Every node is represented by a neuron. The link cost corresponds to the weight between nodes. Specifically, the link cost could be the link travel time, congestion degree, synthesis index from information center, and the link length or road condition stored in database of in-vehicle computer. The computer produces a cost matrix, and then transfer it to T , in which T_{ij} is the connection degree between i and j . Using travel time as an example:

$$T_{ij} = \begin{cases} 0 & i \text{ cannot reach } j \text{ directly} \\ t_{im}^{-\beta} (KT + h\Delta\tau) & i \text{ can reach } j \text{ through } m \end{cases} \quad (2)$$

where $\beta > 0$, $t_{im}^{-\beta} (KT + h\Delta\tau)$ is the prediction of travel time of link m at time $KT + h\Delta\tau$. In traffic networks, because of the road condition, traffic volume and traffic management and control, the costs of one link of two directions are not always equal, that is $T_{ij} \neq T_{ji}$, thus the weight matrix is not symmetrical. $T_{ii} = -\sum_{j \neq i} T_{ij}$ cannot be

satisfied for one node too. Since there are no cost data from node i to i and there is no point in finding the shortest path from i to i , ignoring the leakage of (1), we obtain the following FNN for traffic networks:

$$\frac{du_i}{dt} = \sum_{j=1}^N T_{ij} (s_j - s_i) + I_i \quad i, j = 1, 2, \dots, N \quad (3)$$

As far as traffic networks are concerned, the convergence of FNN is somehow deteriorated in the global search. However, the local search has a good convergence. Therefore, the PSO algorithm is used to do global search in the beginning of stage, and then uses the FNN algorithm to do local search around the global optimum P_g , which decrease greatly computational time to find the optimal shortest path.

3. PSO Algorithm and Particle Encoding for Shortest Path Problem

From the beginning of 90's, new optimization technique researches using analogy of swarm behavior of natural creatures have been started, Eberhart and Kennedy developed PSO based on the analogy of swarm of bird and fish school [18]. PSO is initialized with a group of random particles and then searches for optimum solution by updating generations. In each iteration, each particle is

updated by following two "best" values. The first one is the best solution it has achieved so far and is called p_b and the second one is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population and is called g_b . Finally, each particle updates its velocity and positions using following equations:

$$\begin{aligned} velocity(k+1) &= w * velocity(k) + c1 * rand() * \\ &\quad (p_b - p(k)) + c2 * (g_b - p(k)) \quad (4) \\ p(k+1) &= p(k) + velocity(k+1) \end{aligned}$$

where velocity is the particle's velocity, p is the current particle's position, $rand()$ is a random number between (0, 1), $c1$ and $c2$ are learning factors. The inertia weight w is employed to control the impact of the previous history of velocities on the current velocity, thus to influence the trade-off between global and local exploration abilities of the flying points. The performance of each particle is measured according to a predefined fitness function, which is related to the problem being solved. In stochastic traffic networks, each particle is defined as a sequence of vertexes that represent a valid path and the fitness function is the cost of the path according to the cost of the edges.

The common PSO is either global version or local version of PSO. In global version, all other particles influence the velocity of a particle, while in the local version of PSO, selected number of neighbor particles affect the particle's velocity. In this paper, the global version of PSO is chosen to search the shortest path in traffic networks. At the same time, a constriction factor [19] is used, which can achieve the best performance while using velocity clamping. Thus, it is readily observed that the PSO is very simple to be implemented. However, the thorniest problem in applying PSO to the shortest path problem is how to encode a path in a traffic network into a particle in PSO. This encoding in turn affects the effectiveness of search process. The proposed path encoding algorithm for PSO is essentially based on indirect priority based encoding. The direct encoding scheme is not preferred since a random sequence of nodes is definitely not a good choice for path construction. But the priority-based encoding scheme is suitably modified to address the concerns raised earlier during path construction. Therefore, the pseudo-codes for implementation of priority-based encoding algorithm is presented.

Let N_{max} be the maximum number of nodes in the traffic networks. Let V_p^k be a partial path (corresponding to the position/priority vector of a particle) under growth, which contains $k+1$ nodes with the terminal node i^k ($k=0$ corresponds to the partial path with source node only), and $(k+1)$ th node is to be selected. Let x^k be the dynamic priority vector, which initially contains the

priority values (position vector of the particle), referred to by \mathbf{x} . Every time a node is added to the partial path, the corresponding position in x^k is given a large negative value ($-N_\infty$). Without loss of generality, node number 1 is taken as source node and destination node ID is N_{\max} . The implementations of the modified priority-based encoding along with gradual path construction process are summarized in the following steps:

Step 1:(Initialization) Let $k \leftarrow 0$, $V_p^k \leftarrow 1$ and $x^k \leftarrow x$; $t^k \leftarrow 1$ and $x^k(t^k) = -N_\infty$.

Step 2: (Termination test) If $t^k = N_{\max}$, or $k > N_{\max}$, go to Step 4; else $k \leftarrow k+1$ and go to Step 3.

Step 3: (Path extension) Select node t^k as the node with highest priority from among the nodes having direct links with node t^{k-1} if $(t^k - t^{k-1}) > -M$, where M is a positive integer. Set $V_p^k \leftarrow \{V_p^{k-1}, t^k\}$ and $x^k(t^k) = -N_\infty$.

Step 4: (Complete path) Return complete valid path V_p^k or return invalid path V_p^k if the terminal node is not the destination node.

In Step 2, if the number of iteration exceeds N_{\max} , it would mean either a valid path has not been found due to loops or the path does not terminate at the destination node in N_{\max} steps. In that case, the objective function evaluation of the corresponding particle is made to return a very low value as penalty. In Step 3, the nodes having direct links with the terminal node of the already grown partial path can be found from the network topology. The value of M should be judiciously decided based on network topology.

The quality of a particle (solution) is measured by a fitness function. Here, the fitness function is obvious as the goal is to find the minimal cost path. Thus, the fitness of i th particle is defined as:

$$f_i = \left(\sum_{j=1}^{N_i-1} C_{yz} \right)^{-1}, y = PP^i(j) \text{ and } z = PP^i(j+1) \quad (5)$$

where PP^i is the set of sequential node IDs for i th particle, N_i equal to number of nodes that constitute the path represented by i th particle, and C_{yz} is the cost of the link connecting node y and node z . Thus, the fitness function takes maximum value when the shortest path is obtained. If the path represented by a particle happens to be an invalid path, then its fitness is assigned a penalty value ($= 0$) so that the particle's attributes will not be considered by others for future search.

After the PSO algorithm is used for global search in the beginning of stage, the FNN algorithm is used for local search around the global optimum P_g , therefore, the convergence of traffic networks improve greatly.

4. The Simulation Results and Discussions

The proposed algorithm for the shortest path search is tested on random traffic networks as are shown **Figure 2** through computer simulations using Matlab on an Intel processor (T2050@1.60G). Population size is 30 and the particle's position vector which represents the node priorities is initialized with random integer values in the range $[-90,90]$ and the velocities in the range $[-10,10]$. Both learning factors (c_1 and c_2) are chosen to be 2.05, thus, constriction factor is 0.729 [19]. During the path construction process, the value of M is set to 4. The algorithm is set to run for a maximum of 600 iterations unless stated otherwise. The average CPU time required to achieve the results shown in **Figure 3**. From **Figure 3**, it can be found that average CPU time is less than 1 second for more than 60 nodes, which can achieve real-time search for the best dynamic path in the stochastic traffic networks. In addition to seeking solution for optimal path, it is also equally important to look for closer sub-optimal paths, especially, when the optimal path search is congesting. **Figure 4** shows the route failure ratio to achieve the optimal path in the **Figure 2** traffic networks. For 95% (and 90%) of the optimal path corresponding to 95% (and 90%) of the optimal path cost, the route failure

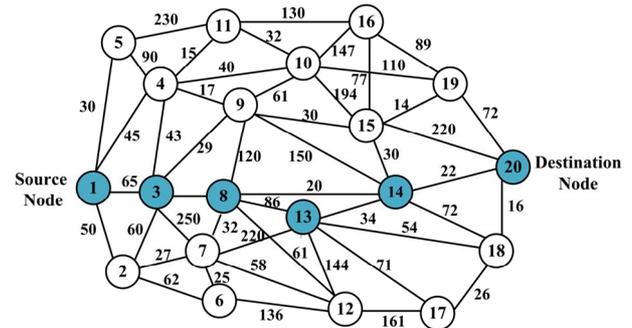


Figure 2. A typical 20-node random network where the weights of the connecting edges are also shown adjacent to the corresponding edges.

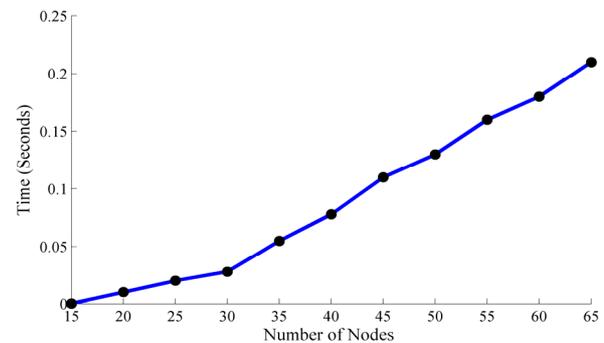


Figure 3. Convergence time with different nodes (population size = 30).

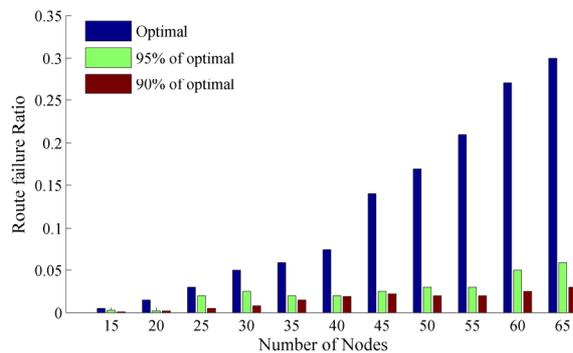


Figure 4. Route failure ratio to achieve optimal, 95% and 90% of optimal path (population size = 30).

ratio is more than 99%. When the nodes are more than 60, the failure ratio is less than 5%. This feature is highly beneficial in real-time automated vehicle dispatching system.

5. Conclusions

In this paper, a new method is proposed for solving the shortest path problem in stochastic traffic networks. The approach based on combining FNN model with PSO. The PSO search uses a modified indirect path-encoding scheme. This algorithm is simple and easy and can find quickly the shortest path without falling into local minimums, which will occur if energy function is used to solve the same problem. Since PSO and FNN are all parallel algorithms, it is very easy to run the new algorithm on parallel computer or even on neural computer, which will reduce the computational time drastically. As a further study, the parameter of PSO algorithm will further be optimized by chaotic operator for diversity of particle in order to further enhance the convergence time of the algorithm.

6. Acknowledgments

I would like to thank the experts who take their time to check the paper and give me valuable suggestions. At the same time, the project was supported by the National Natural Science Foundation of China (30570611, 60773210).

REFERENCES

- [1] L. Fu, D. Sun and L. R. Rilett, "Heuristic Shortest Path Algorithms for Transportation Applications: State of the Art," *Computers & Operations Research*, Vol. 33, No. 11, 2006, pp. 3324-3343. doi:10.1016/j.cor.2005.03.027
- [2] D. Wang, B. Zhang, N. Jiang, Y. W. Jing and S. Y. Zhang, "Traffic Dynamics Based on the Shortest Path Routing Strategy," *Proceedings of the 21st Annual International Conference on Chinese Control and Decision Conference*, Guilin, 17-19 June 2009, pp. 1106-1109.
- [3] R. Bellman, "On a Routing Problem," *Quarterly of Applied Mathematics*, Vol. 16, 1958, pp. 87-90.
- [4] E. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, Vol. 1, No. 1, 1959, pp. 269-271. doi:10.1007/BF01386390
- [5] A. Goldberg and T. Radzik, "A Heuristic Improvement of the Bellman-Ford Algorithm," *Applied Mathematics Letters*, Vol. 6, No. 3, 1993, pp. 3-6. doi:10.1016/0893-9659(93)90022-F
- [6] M. Ali and F. Kamoun, "Neural Networks for Shortest Path Computation and Routing Incomputer Networks," *IEEE Transactions on Neural Networks*, Vol. 4, No. 6, 1993, pp. 941-954. doi:10.1109/72.286889
- [7] R. Wang, S. Guo and K. Okazaki, "A Hill-Jump Algorithm of Hopfield Neural Network for Shortest Path Problem in Communication Network," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, Vol. 13, No. 6, 2009, pp. 551-558.
- [8] C. Ahn and R. Ramakrishna, "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 6, 2002, pp. 566-579. doi:10.1109/TEVC.2002.804323
- [9] J. Kuri, N. Puech, M. Gagnaire and E. Dotaro, "Routing Foreseeable Lightpath Demands Using a Tabu Search Meta-Heuristic," *Proceedings of Global Telecommunications Conference*, New York, 17-21 November 2002, pp. 2803-2807.
- [10] N. R. Raidl and B. A. Julstrom, "A Weighted Coding in a Genetic Algorithm for the Degree-Constrained Minimum Spanning Tree Problem," *Proceedings of the 2000 ACM Symposium on Applied Computing*, Como, 2000. doi:10.1145/335603.335888
- [11] E. Elbeltagi, T. Hegazy and D. Grierson, "Comparison among Five Evolutionary-Based Optimization Algorithms," *Advanced Engineering Informatics*, Vol. 19, No. 1, 2005, pp. 43-53. doi:10.1016/j.aei.2005.01.004
- [12] C. Mouser and S. Dunn, "Comparing Genetic Algorithms and Particle Swarm Optimisation for an Inverse Problem Exercise," *ANZIAM Journal*, Vol. 46, 2004, pp. 174-181.
- [13] A. Salman, I. Ahmad and S. Al-Madani, "Particle Swarm Optimization for Task Assignment Problem," *Microprocessors and Microsystems*, Vol. 26, No. 10, 2002, pp. 363-371. doi:10.1016/S0141-9331(02)00053-4
- [14] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu and Q. X. Wang, "Particle Swarm Optimization-Based Algorithms for Tsp and Generalized Tsp," *Information Processing Letters*, Vol. 103, No. 5, 2007, pp. 169-176. doi:10.1016/j.ipl.2007.03.010
- [15] M. F. Tasgetiren, Y.-C. Liang, M. Sevkli and G. Gencyilmaz, "A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in the Permutation Flowshop Sequencing Problem," *European Journal of Operational Research*, Vol. 177, No. 3, 2007, pp. 1930-1947. doi:10.1016/j.ejor.2005.12.024
- [16] G. Lei, "A Neuron Model with Fluid Properties for Solving Labyrinthian Puzzle," *Biological Cybernetics*,

- Vol. 64, No. 1, 1990, pp. 61-67. doi:10.1007/BF00203631
- [17] H. Wen and Z. Yang, "Study on the Shortest Path Algorithm Based on Fluid Neural Network of in-Vehicle Traffic Flow Guidance System," *Proceedings of the IEEE International Vehicle Electronics Conference*, Changchun, 6-9 September 1999, pp. 110-113.
- [18] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, Vol. 4, 1995, pp. 1942-1948.
- [19] M. Clerc, "The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization," *Proceedings of the IEEE Congress on Evolutionary Computation*, Washington, 1999, pp. 1951-1957.