Commercial Vehicle Scheduling with Time Window by a New Kind of Hopfield Neural Network

Arezoo Mohandessi1, Mehdi Ghatee2, and Gilbert S. Young1

1Computer Science Department, California State Polytechnic University, Pomona, CA, USA 91768
2Computer Science Department, Amirkabir University of Technology, Tehran, Iran 15875

Abstract- In this paper, we will present a Hopfield Neural Network (HNN) model to tackle the Vehicle Routing Problem with Flexible Time Windows (VRPFlexTW). The rationale of the proposed HNN model is to derive a new form of energy function to describe vehicle routing costs in the VRPFTW. In order to achieve a minimum total cost, we first study a standard mathematical model which contains all constraints in the VRPFTW. Then, a Hopfield Neural Network (HNN) model is designed to rapidly find the satisfactory solution of VRPFTW. Finally, we evaluate the proposed HNN by a number of Vehicle Routing Problem with Time Windows (VRPTW) benchmark instances. Our computational results show that the proposed HNN significantly reduces the computational time compared to standard mathematical methods.

Keywords: Vehicle routing problems, Time windows, Soft time windows, Flexible time windows, Hopfield Neural Network.

1 Introduction

A carrier companies’ intention is to transport the products to customers in a cost-effective and reliable manner with respect to customer service demands. In this setting, the problem is often described as the Vehicle Routing Problem (VRP). The VRP explores the optimal vehicle routing by minimizing the total travel cost incurred by a set of homogeneous vehicles that deliver customer demands, which has been widely used in many real-life applications, including bank deliveries, postal deliveries, and school bus routing (see Hashimoto et al. [4]).

In practice, time window constraints can be relaxed to a certain extent. One of the popular variants of the VRP is the Vehicle Routing Problem with Time Windows (VRPTW) [1]. The VRPTW investigates less expensive routes such that each customer is served within a predetermined flexible time window by a single vehicle. In particular, each vehicle starts and ends its routes at given depots and delivers a quantity not exceeding its capacity. Moreover, each vehicle is permitted to arrive before the opening of the time window and wait at no cost until service becomes possible. However, each vehicle is not permitted to arrive after the time window closes [3].

The original vehicle routing problem with time windows (VRPTW) is NP-hard [5]. In the literature, quite a few exact and heuristic optimization techniques have been developed to tackle the VRPTW. These heuristic algorithms explore least cost routes by minimizing the total cost of achieving the best result. Among those optimization techniques, the time window relaxation employs soft time windows in vehicle routing problem, named the Vehicle Routing Problem with Flexible Time Windows (VRPFlexTW). The soft time window assumes that some or all customer time windows are soft and can be violated by paying appropriate penalties (see Balakrishnan [5]). As a result, this approach allows vehicles to service customers in a flexible time, and assists the company to reduce the cost and the total travel time.

In this paper, we propose a Hopfield Neural Network (HNN) model to rapidly find the solution to the VRPFlexTW with lower and upper bound flexible time windows. The fundamental idea of the proposed HNN is to derive a new energy function describing the cost of vehicle routings in VRPFlexTW. We first analyze a mathematical model of the VRPFlexTW. Then, we design a Hopfield Neural Network model to find the satisfactory vehicle routing by minimizing the underlying energy function. Finally, we apply this approach to a set of real-life VRPFlexTW benchmark problems. Our computational results show that the proposed HNN, significantly reduce the computational time compared to standard mathematical methods.

2 Methodology

Figure 1 depicts the possible arrivals and their corresponding penalty cases at customer location. In the early servicing case, service at the customer starts between the flexible earliest time and the original earliest time. In the late servicing case, service takes place between the original latest time and the flexible latest time. Note that if the vehicle arrives early, have to wait at a customer’s location until the flexible time window is reached. In addition, they cannot serve after the customer flexible time window, as mentioned earlier in [3].
3 Model Formulation

A direct graph \( G = (V, E) \) is often used to describe the VRPFlexTW [3], consisting of the set \( V \) of nodes and the set \( E \) of edges. In the node set \( V \), the central depot is represented by the node with index 0, while the customer locations are the other nodes in \( V \). Figure 2 depicts a sample graph of five cities.

For each customer \( i \in V \), we have a positive demand \( q_i \), a time window \([W_i^s, W_i^f]\). For each node \( i \), a flexible time window \([W_i^s, W_i^f]\) is generated with respect to the length of the original time window, where \( W_i = W_i^s - (W_i^s - W_i^f) \) and \( W_i = W_i^s - (W_i^s - W_i^f) \).

Additionally, \( Q \) represents the capacity given for each vehicle \( v \in V \) where \( V \) denotes a homogeneous fleet.

Associated with each arc \((i, j) \in A\), \( t_{ij} \) and \( d_{ij} \) represent the travel time and the distance along that arc, respectively. A fixed cost \( C_f \) is incurred for using a vehicle. Time window violations, i.e., serving a customer within \([W_i^s, W_i^f]\) and \([W_i^s, W_i^f]\) are penalized by \( C_s \) and \( C_d \), for one unit of earliness and one unit of delay, respectively. Moreover, \( C_s \) is the cost paid for one unit of distance.

Under these assumptions, the mathematical model is formulated as follows:

\[
\begin{align*}
\text{Min } & c_i \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij} + c_f \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ij} + c_e \sum_{i=1}^{N} \sum_{j=1}^{N} e_{ij} + c_s \sum_{i=1}^{N} \sum_{j=1}^{N} h_{ij} \\
\text{Subject to: } & \\
\sum_{j=1}^{M} \sum_{i=1}^{N} x_{ij} = 1 & i=2, \ldots, N \quad (2) \\
\sum_{j=1}^{M} x_{ij} - \sum_{j=1}^{M} x_{ij} = 0 & j=2, \ldots, N, v=1, \ldots, M \quad (3) \\
\sum_{i=1}^{N} q_i \sum_{j=1}^{M} x_{ij} & \leq Q \quad (4)
\end{align*}
\]

In the above model, \( x_{ij} \) is equal to 1 if vehicle \( v \) serves node \( j \) immediately after node \( i \) and 0, otherwise. \( s_{ij} \) denotes the time that vehicle \( v \) starts serving node \( i \). Furthermore, \( e_{iv} \) and \( h_{iv} \) represent the earliness and the delay at node \( i \), in case it is served by vehicle \( v \), respectively. The objective (1) is to minimize the total cost, which consists of traveling costs, fixed costs of vehicles used for servicing, and penalty costs incurred for early and late servicing.

The constraints (2) and (3) guarantee that exactly one vehicle arrives and leaves a customer location. The constraints (4) ensure that the vehicle capacity is not exceeded. The constraints (5) and (6) indicate that each vehicle route starts and terminates at the depot. The constraints (7) represent the relationship between the starting time of service at a customer and the departure time of vehicle from its predecessor. The constraints (8) ensure that the service takes place at each customer with respect to the customer (flexible) time window. The constraints (9) and (10) link the earliness and the beginning of service; similarly, the constraints (11) and (12) link the delay and the
beginning of service. The constraints (13) indicate that there is no partial servicing.

4 Hopfield Neural Network

Hopfield Neural Network is beautifully simple device that can be used for storing memories as distributing pattern activities. It is a kind of energy based model because the properties derive from global energy function according to [6].

The global energy is the sum of many contributions. Each contribution depends on one connection weight and the binary states of two neurons:

\[ E(s) = \sum_{i,j} s_is_jw_{ij} + \sum_i s_i \theta_i \]  

(14)

See Hopfield and Tank [8] for details. Serpen [9] proved that the following iterative scheme converges to the local minima of the energy function:

\[
\frac{ds_i}{dt} = -\sigma(s_i(n))h_i(n)
\]

(15)

\[
\sigma(s_i(n))h_i(n) = \sum_j w_{ij}s_j(n) + I_i
\]

(16)

\[
s_{i}(n+1) = s_{i}(n) + \frac{ds_i}{dt}(n)
\]

4.1 Converting the Mathematical Model to Hopfield Neural Network Form

We encounter two different decision arguments, binary \( X_{ijv} \) and \( S_{iv} \), with integer value. In this problem, we will try to add one dimension in order to eliminate \( S_{iv} \). Then, we can add constraints to energy function. We will use 4 dimensions variable \( X_{ijvs} \) instead of \( X_{ijv} \), which means that \( X_{ijvs} = 1 \) depicts that vehicle \( v \) in time \( s \), start to service the customer in the city of \( i \), then it goes to city \( j \). The energy function with considering of objective (1) is:

\[
E = \sum_{v,i,j} W_{ijv}s_{ijv} + \sum_{v,i} c_{iv}X_{ijvs} \]

(17)

\[
e_{iv} = I_v - a_v = I_v - (s_{iv} + t_u) = I_v - \sum_{k=1}^{N} x_{ivos}(s_x + t_d)
\]

(18)

\[
h_{iv} = a_v - u_v = (s_{iv} + t_u) - u_v = \sum_{k=1}^{N} x_{ivos}(s_x + t_d) - u_v
\]

(19)

By assigning 17 and 18 in 16, an energy function can be achievable:

\[
E = \sum_{i} (c_{i} - c_{iv}X_{ijvs}) + \sum_{v,i,j} W_{ijv}s_{ijv}
\]

(20)

We changed constrains (2 – 13) to the fine expression as follows:

\[
E_s = \sum_{v,i,j} W_{ijv}(s_{ijv} - 1)^2
\]

(21)

\[
E_s = \sum_{v,i,j} W_{ijv}(s_{ijv} - 0)^2
\]

(22)

\[
E_s = \sum_{i} (c_{i} - c_{iv}X_{ijvs})
\]

(23)

Now we can create Hopfield energy function:

\[
E = \sum_{v,i,j} W_{ijv}s_{ijv} + \sum_{v,i} c_{iv}X_{ijvs}
\]

(24)

\[
\alpha x = \sum_{v,i,j} (s_{ijv} - 1)^2 + \theta_i
\]

(25)

\[
\alpha x = \sum_{v,i,j} (s_{ijv} - 0)^2
\]

(26)

\[
\alpha x = \sum_{i} (c_{i} - c_{iv}X_{ijvs})
\]

(27)

\[
\alpha x = \sum_{v,i} c_{iv}X_{ijvs}
\]

(28)

\[
\alpha x = \frac{\partial E}{\partial X_{ijvs}} \bigg|_{w_{ijv}}
\]

(29)

\[
\alpha x = \frac{\partial E}{\partial X_{ijvs}} \bigg|_{w_{ijv}}
\]

(30)

\[
\alpha x = \frac{\partial E}{\partial X_{ijvs}} \bigg|_{w_{ijv}}
\]

(31)

\[
\alpha x = \frac{\partial E}{\partial X_{ijvs}} \bigg|_{w_{ijv}}
\]

(32)

\[
\alpha x = \frac{\partial E}{\partial X_{ijvs}} \bigg|_{w_{ijv}}
\]

(33)

\[
\alpha x = \frac{\partial E}{\partial X_{ijvs}} \bigg|_{w_{ijv}}
\]

(34)

\[
\alpha x = \frac{\partial E}{\partial X_{ijvs}} \bigg|_{w_{ijv}}
\]

(35)

\[
\alpha x = \frac{\partial E}{\partial X_{ijvs}} \bigg|_{w_{ijv}}
\]

(36)

5 Results

In this section, we present the numerical results obtained with the proposed HNN on a set of real-life VRPTW problems. The experiments are carried out on a computer with (Intel Core i7-6700U) 2.53 GHz CPU and 12 GB of RAM.

5.1 Benchmark Examples

We apply the proposed HNN to the VRPTW benchmark problems in the R1-type collection [7], which contain 29 problem instances with 100 customers. Each instance has one
depot as the central location of the homogeneous fleet of vehicles, and the vehicle capacity is 200 units. In our experiments, the cost coefficients \((c_i, c_j, c_e, c_f)\) are set to \((2.0, 400, 0.5, 1.0)\). To find a minimum cost in this example, we applied the proposed Hopfield Neural Network. To come up with reasonable results, there was a need to find learning coefficient \(\sigma\) in (15). To tune up these parameters, we investigate the effect of different values, varying between 0 and 1, for \(\sigma\) on the efficiency of solutions. As indicated in Figure 4, for \(\sigma = 0.3\) the energy function converges to zero with the least number of iterations.

5.2 Comparison with the MM

Table 1 presents the Fitness [10] and the computational time of HNN on the 10 problem instances from the R1-type collection, in comparison with MM. Compared to MM, which consumes more than 4 hours to obtain the optimal solution, the proposed HNN significantly reduces the computational time to half one second with an acceptable cost.

<table>
<thead>
<tr>
<th>Problem</th>
<th>HNN Fitness</th>
<th>Time (s)</th>
<th>MM Fitness</th>
<th>Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>125.89</td>
<td>0.3969</td>
<td>124.86</td>
<td>&gt;4</td>
</tr>
<tr>
<td>2</td>
<td>177.66</td>
<td>0.3971</td>
<td>176.66</td>
<td>&gt;4</td>
</tr>
<tr>
<td>3</td>
<td>279.18</td>
<td>0.3962</td>
<td>278.98</td>
<td>&gt;4</td>
</tr>
<tr>
<td>4</td>
<td>368.95</td>
<td>0.3966</td>
<td>365.87</td>
<td>&gt;4</td>
</tr>
<tr>
<td>5</td>
<td>126.37</td>
<td>0.4083</td>
<td>125.34</td>
<td>&gt;4</td>
</tr>
<tr>
<td>6</td>
<td>188.02</td>
<td>0.3970</td>
<td>187.05</td>
<td>&gt;4</td>
</tr>
<tr>
<td>7</td>
<td>284.67</td>
<td>0.3984</td>
<td>284.66</td>
<td>&gt;4</td>
</tr>
<tr>
<td>8</td>
<td>374.88</td>
<td>0.4007</td>
<td>374.87</td>
<td>&gt;4</td>
</tr>
<tr>
<td>9</td>
<td>132.16</td>
<td>0.3994</td>
<td>131.04</td>
<td>&gt;4</td>
</tr>
<tr>
<td>10</td>
<td>132.55</td>
<td>0.3975</td>
<td>132.25</td>
<td>&gt;4</td>
</tr>
</tbody>
</table>

Table 1 Simulation Results Obtained by HNN and MM

Figure 5 illustrates the fitness values of HNN on the 10 problem instances from the R-type collection, in comparison with MM. According to figure 5, there is a very slight difference between the fitness values of HNN and MM.

6 Concluding Remarks

We have investigated the mathematical model of VRPFlexTW problem. Hopfield Neural Network was used to solve vehicle routing flexible time windows problem. Several numerical experiments were illustrated to confirm the effectiveness of Hopfield Neural Network. In their future research, the authors plan to accelerate the proposed Hopfield Neural Network by considering meta-heuristics.

7 References