Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Hopfield neural network based on clustering algorithms for solving green vehicle routing problem

Serap Ercan Comert[a*], Harun Resit Yazgan[a] and Gamze Turk[a]

[a]*Sakarya University, Industrial Engineering Department, Sakarya, Turkey*

CHRONICLE

Article history:
Received February 15 2022
Received in Revised Format
March 28 2022
Accepted June 22 2022
Available online
June, 22 2022

Keywords:
Green vehicle routing problem
Hopfield Neural Network
K-means clustering algorithm
K-medoids clustering algorithm

ABSTRACT

As a result of the rapidly increasing distribution network, the toxic gases emitted by the vehicles to the environment have also increased, thus posing a threat to health. This study deals with the problem of determining green vehicle routes aiming to minimize CO_2 emissions to meet customers' demand in a supermarket chain that distributes fresh and dried products. A new method based on clustering algorithms and Hopfield Neural Network is proposed to solve the problem. We first divide the large-size green vehicle routing problem into clusters using the K-Means and K-Medoids algorithms, and then the routing problem for each cluster is found using the Hopfield Neural Network, which minimizes CO_2 emissions. A real-life example is carried out to illustrate the performance and applicability of the proposed method. The research concludes that the proposed approach produces very encroaching results.

1. Introduction

Since the beginning of time, logistics has been a critical link for the worldwide circulation of products. Due to the significant impact of the COVID-19 pandemic on the world economy, logistics has become an essential role in preventing supply chain disruptions. Companies and organizations need to make various decisions to optimize and effectively manage their processes in distribution systems. One of the most critical decisions in the distribution process is the vehicle routing decisions made from the facilities to the customers. The Vehicle Routing Problem (VRP), first scientifically defined by Dantzig and Ramser in 1959, is the problem of determining minimum cost routes with a fleet of vehicles to meet customer demands. The VRP, which is one of the problems in the NP-hard class, has many different types according to its constraints (Dantzig & Ramser, 1959). In recent years, studies that consider an environmental effect and minimize costs in distribution systems have become widespread, and the concept of green logistics has developed. The green vehicle routing problem (GVRP), on the other hand, can be defined as a VRP that takes energy consumption into account in general. Reducing fuel consumption in transport activities will significantly impact greenhouse gas emissions (Erdoğan & Miller-Hooks, 2012). The GVRPs are generally considered as two ways. The first is aimed to minimize fuel consumption, CO_2 emissions, load amount, travel time in the general vehicle routing problem. The second is to minimize vehicle routes using either conventional and/or alternative fuels within a vehicle fleet.

There are different solution methods developed to solve GVRPs in the literature. The methods can be classified as an exact, a heuristic, and a metaheuristic. Since the VRP is in the NP-hard problem class, it is challenging and sometimes impossible to reach an optimal solution using exact methods if the size of the problem overgrows. For this reason, heuristic/meta-heuristic methods can be chosen to solve these problems, which usually have the potential to provide the best values (near-optimal) within reasonable solution time constraints (Lenstra & Rinnooy Kan, 1981). This study uses an artificial neural network (ANN) among the metaheuristic methods.

* Corresponding author
E-mail: serape@sakarya.edu.tr (S. E. Comert)

2022 Growing Science Ltd.
doi: 10.5267/j.ijiec.2022.6.002

ANNs are complex systems consisting of the connection of artificial neurons, similar to neurons, which are the basic unit of the human brain, with different topology and network models. Instead of giving a step-by-step method in the program, the neural network generates its own internal rules that make the association and organizes them by comparing the results with examples (Rojas, 1996). In general, ANNs are examined in two main groups: feedforward and feedback. For the feedforward networks, neurons are organized through input, hidden, and output layers. The neurons in each layer are associated with neurons in the next layer. However, the layers have no connections between each other. The feedback networks are also called iterative networks. The most important feature of these neural networks is the feedback connection, parallelism, collective analog mode, and dynamic memory. Each neuron collects the inputs of hundreds and thousands of other neurons to provide cascading output (Jain et al., 1996; Lippmann, 1987). Hopfields Neural Network (HNN), one of the feedback networks, is among the first heuristic approaches developed to solve combinatorial optimization problems in ANNs. HNN was first proposed in 1982 by JJ Hopfield to solve optimization problems by reducing programming complexity and quickly reaching steady states. Its high computation speed provides the most suitable solution for neural networks to calculate the shortest path in a much shorter time. The Hopfield model is the energy minimizing network and is helpful as a content-addressable memory for solving combinatorial optimization problems (Hopfield, 1982). For these reasons, HNN is preferred in this study.

A few studies on HNN focus on solving the Traveling Salesman Problem (TSP), the most basic form of vehicle routing problems, can be found in the literature. Hopfield & Tank (1985) firstly used the HNN to solve the TSP. This technique minimizes an energy function with some terms and parameters.

Some studies focused on developing the energy function, ensuring that the model produces feasible solutions (Van den Bout & Miller, 1988; Li et al., 2016). Selecting the initial parameter values is used more optimally (Kamgar-Parsi & Kamgar-Parsi, 1992; Lai & Coghill, 1992; Wu & Yang, 2004, Tan et al., 2005) to solve the TSP using HNN. Luo (2019) also improved Hopfield's energy function and proposed a genetic algorithm (GA) based on HNN for TSP. He used his proposed method on sample problems with 8 and 10 cities and compared the results with the classical HNN.

To obtain more effective solutions, some researchers hybridized HNN with some methods. Jolai & Ghanbari (2010) integrated Z-score and logarithmic data transformation techniques with HNN to solve the TSP. They compared their results on a sample problem considering ten cities. Oshima et al. (2015) proposed two methods to solve TSP by integrating the random cut and neuronal death method into HNN. Zhong et al. (2017) created a dynamic step-based continuous HNN and applied it on sample problems as 10 and 20 cities.

Abdel-Moetty (2010) applied HNN and the Ant colony algorithm (ACA) for the TSP using 15 cities and compared these results. Sarwar & Bhatti (2012) compared HNN and Lin and Kernighan heuristic for TSP by considering sample size as 5, 10, 20, and 50 customers.

This paper proposes a hierarchical approach that consists of a clustering algorithm and HNN to solve the GVRP. We first divide the large-size GVRP into clusters using the K-Means and K-Medoids algorithms, and then vehicle routes are determined for each cluster using the HNN to minimize CO_2 emissions. The proposed approach is also applied to a real-life example related to a distribution CVRP meeting the weekly demands of a supermarket chain.
The HNN results compared with cluster first route second approach proposed by Comert et al. (2018) using Friedman test.

Studies in the literature show that HNN effectively solves small-size TSP problems. It confirms that the literature survey has given up to 50 customers. In order to effectively solve real-life problems with more than 50 customers, a large-size problem can be transformed into a small-size problem using clustering algorithms. Therefore, our study's first contribution is to deal with HNN's large-scale problems based on K-means and K-medoids clustering algorithms. Additionally, this brief survey shows a significant gap in the literature related to applying the HNN for VRPs. The GVRP is a more comprehensive type of TSP created by adding specific constraints to it. As a result, it requires more processing than the TSP as a solution is difficult. Considering the speed and computational power features of neural network approaches in applications, we contribute to the literature by using the HNN for the large size GVRP.

The remainder of this paper is organized as follows. Section 2 presents the GVRP. The clustering algorithms used in this study are discussed in section 3. A general overview of HNNs and HNNs method for GVRP is mentioned in Section 4. We present real-life examples and the proposed method results in Section 5. Finally, conclusions and discussions of future research directions are presented in section 6.

2. Green Vehicle Routing Problem (GVRP)

The GVRP aims to find feasible routes for delivering products from one depot to customers by minimizing the CO_2 emissions with vehicles considering a certain capacity. The GVRP has many vehicles to meet the demands of customers. Vehicles that carries customer demands should not exceed the capacity constraint. The route of each vehicle starts and ends at the depot. Each customer must be visited only once by a single-vehicle. The problem aims to minimize CO_2 emissions.

The GVRP is defined in a complete graph G = (N, A) where N = {0, 1, …, n} is the set of customers and "0" represents the depot. A corresponds to the edges connecting vertices of N. The notation is provided in Table 1.

Table 1
Notation of GVRP

Sets	
C	: Set of customers
K	: Set of vehicles
N	: Set of all nodes, N="0"∪ C
Parameters	
Q	: Cargo capacity of the vehicles
$q(S)$	: Demand of customers in the subset S
$\boldsymbol{e_{ij}}$	: CO_2 emissions between customer i and j
Decision Variable	
$\boldsymbol{x_{ijk}}$=Binary decision variable, taking a value of 1 if vehicle k goes from customer i to customer j and 0 otherwise.	

We modified Toth & Vigo's (2002) mathematical model using the CO_2 emission model presented by Nie and Li (2013). The mathematical formulation of the GVRP is as follows:

Objective Function:

$$\text{Min } Z = \sum_{i=0}^{n}\sum_{j=0}^{n} e_{ij} \sum_{k=1}^{m} x_{ijk} \tag{1}$$

Constraints:

$$\sum_{i \in N}\sum_{k \in K} x_{ijk} = 1 \qquad \forall j \in C \tag{2}$$

$$\sum_{j \in N}\sum_{k \in K} x_{ijk} = 1 \qquad \forall i \in C \tag{3}$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0 \qquad \forall k \in K \quad \forall h \in C \tag{4}$$

$$\sum_{i \in C} q_i \sum_{j \in N} x_{ijk} \leq Q \qquad \forall k \in K \tag{5}$$

$$\sum_{j \in C} x_{0jk} = 1 \qquad \forall k \in K \tag{6}$$

$$y_i - y_j + n \sum_{k \in K} x_{ijk} \leq n - 1 \quad i \neq j \quad \forall i \in C \ \forall j \in C \tag{7}$$

Constraint (1) is an objective function that aims to minimize the overall CO_2 emission of the route. Constraint (2) and (3) guarantee that each customer is visited exactly once. Constraint (4) provides the flow constraint that the same vehicle enters and leaves the same depot in a route. Constraint (5) states that the maximum loading capacity of the vehicles is satisfied. Constraint (6) guarantees that each vehicle is used exactly once. Constraint (7) is used to eliminate sub-tours.

3. Solution Methodology

We propose a clustering algorithm and the HNN-based approach to solve large GVRP in a reasonable time since GVRP generalizes the TSP, which is NP-hard, and VRPs are generally much more difficult to resolve than TSPs of the same size. In our approach, we first divide the large-size GVRP into clusters using the K-Means and K-Medoids algorithms, and then vehicle routes are determined for each cluster using the HNN to minimize CO_2 emissions.

3.1. Clustering analysis

Cluster analysis divides data into groups according to the determined similarity criteria. The clustering algorithm aims to minimize the similarity among groups and maximize the similarity of the within groups. Clustering techniques are divided into hierarchical and non-hierarchical clustering techniques. In this study, we use K-Means and K-Medoids clustering algorithms which are a member of non-hierarchical clustering algorithms.

3.1.1. K-Means algorithm

K-Means, one of the oldest clustering algorithms, was developed by J.B. MacQueen in 1967 (MacQueen, 1967). The assignment mechanism of K-Means, one of the most widely used unsupervised learning methods, allows each data to belong to only one cluster. This method's main idea is that the central point represents the cluster (Han & Kamber, 2001). The Error Sum of Squares criterion (SSE) is most used to evaluate the K-Means clustering algorithm. The clustering result with the lowest SSE value gives the best results. The sum of the squares of the distances of the objects from the cluster's center points is calculated with Eq. (8) (Tan et al., 2006).

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x) \tag{8}$$

This criterion allows clusters to be determined densely and separately from each other. The algorithm tries to identify k groups that will reduce the error sum of squares function. The K-Means algorithm divides the data set consisting of n data into k clusters with the k parameter given to the algorithm by the user. Cluster similarity is measured by the mean value of the objects in the cluster, which is the gravity center of the cluster (Xu & Wunsch, 2005). The pseudocode for the K-Means algorithm is given in Algorithm 1.

Algorithm 1. The K-Means Algorithm

```
Begin
  Determine the cluster centroids
Do
  Calculate the distance between specified point and centroids
  Assign all objects to the group that has the closest centroid
  Recalculate the positions of the k centroids
Until convergence criteria are met
End begin
```

3.1.2. K-Medoids algorithm

Kaufman & Rousseeuw (1990) developed the K-Medoids clustering algorithm to eliminate the sensitivity of the K-Means clustering algorithm to noise and outline. There are many different variants of the K-Medoids algorithm. The first proposed K-Medoids algorithm is PAM (Partitioning Around Medoids). Although representative objects are often called centrotypes in the clustering literature, they are called medoids in the PAM algorithm. In the PAM algorithm, k randomly selected representative objects (Medoids) are chosen as the cluster's center. After k medoids are determined, k clusters are created by assigning each object to the closest medoids. Each medoid is replaced with a non-representative object in the following steps and shifted until improved clustering quality. Thus, the possibility of some data moving to the cluster center is eliminated (Kaufman & Rousseeuw, 1990). The pseudocode for the K-Medoids algorithm is given in Algorithm 2.

Algorithm 2. The K-Medoids Algorithm

```
Begin
  Determine k cluster number
  Choose k objects as initial medoids
Do
Assign the rest of the clusters to the closest medoids x
Calculate objective function
Randomly select y object that is not medoids
   If changing x and y minimizes the objective function
      Change x and y
   End if
Until convergence criteria are met
End begin
```

3.2. Hopfield neural networks

Hopfield neural network structures are single-layer and fully connected neural network structures used mainly for auto-association, developed by the physicist John Hopfield in 1982 (Hopfield, 1982).

All nodes in the HNN are fully interconnected. Each node is an input to every other node in the network. Connections from each node to itself can be thought of as a zero-weighted connection. HNN is a single-layer ANN with feedback and generally accepts bipolar (-1 or +1) or binary (0 or 1) inputs. Fig. 1 shows the HNN architecture (Rao, 1995).

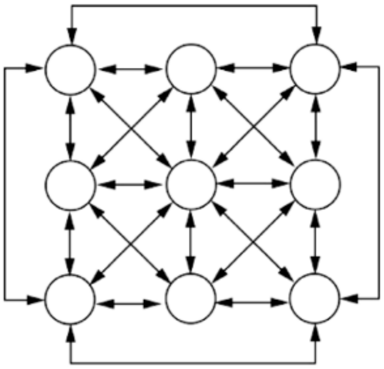


Fig. 1. HNN architecture

HNN is divided into discrete and continuous according to their output values. Although the working styles are the same, the activation function is different. The sign function is used in discrete HNN. However, the hyperbolic tangent function is used in continuous HNN. While discrete HNN is used for self-association, continuous HNN is among the first heuristic approaches developed for solving combinatorial optimization problems for ANN.

In HNN, each variable in the problem is represented by a node. The connections between variables are also characterized by their weight values. Positive weight values show mutually supportive relationships between variables, while negative weight values represent inhibitory relationships. Optimization of the neural network is the process of minimizing the objective function. In HNN, the weight remains unchanged throughout the process, and the current situation of the neuron is related to the last moment. For this reason, the energy function satisfies the stability of HNN. The energy function is as follows (Equation (9)):

$$\mathbf{E} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}v_i v_j - \sum_{j=1}^{N} v_j I_j \tag{9}$$

where N is the number of neurons, w_{ij} is the weight of interconnect between the output of the neuron j and the input of the neuron i, v_i is the output of the neuron i, I_j is the external input signal of the neuron j (Hopfield & Tank, 1985).

3.2.1. Hopfield neural network-based method for GVRP

The proposed method's steps for the GVRP are given as follows:

Step 1: The distance matrix among the customers' pairs is initialized.

Step 2: An initial transposition matrix of size *N*x*N* is defined to create the connection weights of the network. *N* represents the number of customers.

The transposition matrix of size *N*x*N* consists of 0 and 1 elements. Each row and column represent a particular city and position, respectively. The element "1" in the i^{th} position of row A indicates that the visit order of city A is *i*. The transposition matrix of the four cities routing problem representing HNN and the route defined as the matrix is shown in Fig.2.

$$\boldsymbol{V} = \begin{bmatrix} \mathbf{A{:}\,0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{B{:}\,0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{C{:}\,0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{D{:}\,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} = \mathbf{D \rightarrow B \rightarrow A \rightarrow C \rightarrow D}$$

Fig. 2. Transposition Matrix in the routing problem of 4-City

The first row of matrix *V* in Fig. 2 represents city A. Vector (0, 0, 1, 0) means that city A is the third city to visit. The second vector (0, 1, 0, 0) indicates that city B will be visited in the second position, the third vector (0, 0, 0, 1) indicates that city C will be visited in the fourth position and the last vector (1, 0, 0, 0) indicates that city D will be visited first.

Step 3: The energy function is created as in Equation (10) to store the connection of weights (Hopfield & Tank, 1985).

$$\boldsymbol{E = \frac{A}{2}\sum_{x}\sum_{i}\sum_{j\neq i} v_{x.i}v_{x.j} + \frac{B}{2}\sum_{i}\sum_{x}\sum_{y\neq x} v_{x.i}v_{y.i} + \frac{C}{2}\left(\sum_{x}\sum_{i} v_{x.i} - N\right)^{2} + \frac{D}{2}\sum_{x}\sum_{x\neq y}\sum_{i} e_{x.y}v_{x.i}(v_{y.i+1} + v_{y.i-1})} \qquad (10)$$

In Equation (10), positive constants are A, B, C, and D. It is crucial to choose constants correctly for the stability of HNN. *N* is the number of cities to be visited on route. $e_{x,y}$ represents the CO_2 emissions from city *x* to city *y*. $v_{x,i}$ is the output of the *i*th neuron in the neuron array corresponding to the *x*th city. The final state of *N*x*N* variables consists of the terms 0 or 1 or very close to 0 and 1. If we analyze the energy function, the first and second terms give row and column constraints, respectively. The first term indicates that no city is visited more than once at the same time. The second term represents that each city is precisely once. The third term takes into care that all cities are visited. The first three constraints of the energy function provide a stable combination matrix. The final sum provides the constraint for the shortest CO_2 emissions. This term's value is minimum when the total CO_2 emissions are the shortest. The value of D is essential for deciding between the time is taken for convergence and the solution's optimality. A low D value means closer to the optimum solution, but it takes longer for the HNN to converge. If the D value is high, it converges quickly, but the solution deviates from the optimum.

Step 4: The connection weights of the network are calculated by equation (11).

$$\boldsymbol{W_{xi.yj} = -A\delta_{xy}\left(1-\delta_{ij}\right) - B\delta_{ij}\left(1-\delta_{xy}\right) - C - Dd_{xy}\left(\delta_{j.i+1} + \delta_{j.i-1}\right)} \qquad (11)$$

where δ is the Kronecker's delta, which is defined as:

$$\delta_{xy} = \begin{cases} 1. & x = y \\ 0. & x \neq y \end{cases} . \delta_{ij} = \begin{cases} 1. & i = j \\ 0. & i \neq j \end{cases}$$

Step 5: The network parameter values are entered, and the energy function is iterated for the number of iterations determined using the hyperbolic tangent function, one of the activation functions, to obtain the stable state of the neural network. The hyperbolic tangent function is given in Equation (12).

$$\boldsymbol{v_{xi} = f(U_{xi}) = \frac{1}{2}[1 + \tan\mathrm{h}\left(\frac{U_{xi}}{U_0}\right)]} \qquad (12)$$

Here U_{xi} is the input of the neuron and U_0 is an initial value. U_{xi} calculates as Equation (13) (Wilson & Pawley, 1988).

$$\boldsymbol{U_{xi} = \sum_{i=1}^{N} w_{xi.yj}\, v_{yj} + I_{xi}} \qquad (13)$$

Step 6: As a result of the iterations, in the transposition matrix, the position corresponding to the maximum value of each column is set to 1, the others to 0.

Step 7: It is checked whether the created transposition matrix satisfies the constraints of the GVRP problem. If so, the following steps are continued. If not, go to Step 5.

Step 8: The route created according to the transposition matrix is divided by considering the customer demands and the vehicle capacity constraint, and the vehicles are assigned.

4. Experimental Results

4.1. A case study

In this study, the problem is to find the vehicle routes that minimize the total CO_2 emissions to meet the store demands of the vehicles leaving the main depot in a supermarket chain operating. The supermarket chain has one main depot and 78 stores. The location of the main depot and stores on the map is shown in Fig. 3. The CO_2 emissions of inter-store and stores between the main depot are calculated according to the Comprehensive Emission Model proposed by Nie & Li (2013). This problem includes a homogeneous fleet with a 40 pallets capacity. The vehicle fleet has two types of trucks: fresh and dry products for a distribution process. 78 stores are divided into two as daily and not daily stores. The daily stores demand products every weekday. Whereas not daily stores only demand products three days a week (Monday, Wednesday, and Friday). The number of daily and not daily stores are 66 and 12, respectively. The numbers of the daily and not daily stores are given in Table 2.

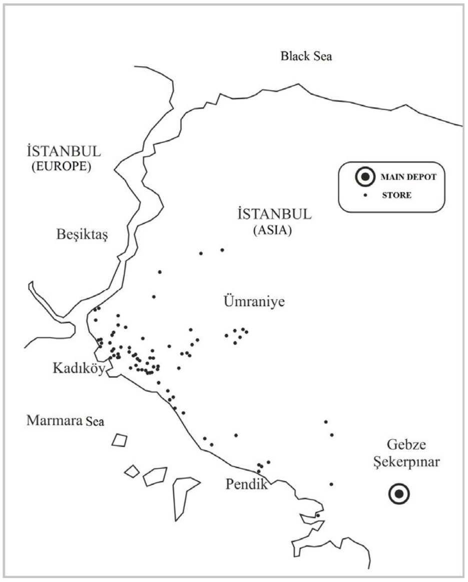


Fig. 3. Location of the main depot and stores

Table 2
Store type and it's number

Store Type	Store Number
Daily	1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,25,26,28,29,31,32,34,35,37,38,40,41, 43,44,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,72,73,75,76,78
Not daily	6,24,27,30,33,36,39,42,45,71,74,77

Days are classified as "busy" and "not busy" days. Distributing all stores is called a "busy day", whereas distributing only not daily stores is called a "not busy day". The problem is divided into four sub-problems based on the distribution days of the products.

1. Sub-problem: Distribution of Fresh Products on Busy Day
2. Sub-problem: Distribution of Dry Products on Busy Day
3. Sub-problem: Distribution of Fresh Products on not Busy Day
4. Sub-problem: Distribution of Dry Products on not Busy Day

20-week demand data are obtained considering the average of the previous quarterly demands of the stores. In Table 3, the demands of the sub-problems in the first week.

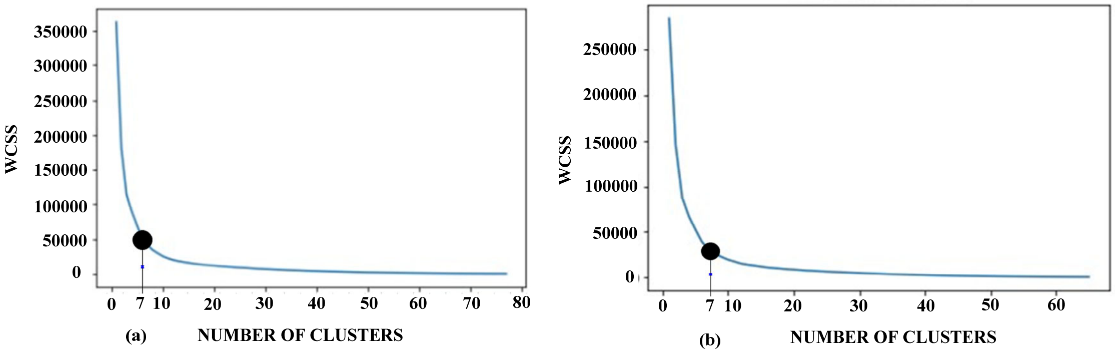


Fig. 4. The elbow graph of stores distributed on (a) busy days, (b) not busy days

Before using the proposed HNN to solve the problem, 78 and 66 stores are clustered with K-Means and K-Medoids clustering algorithms. First, the number of clusters (k) is determined for the K-Means and K-Medoids algorithms, as explained in Section 3. The Elbow method is used to determine the k number in this study. The elbow method is a method that deals with the percentage of variance, which is a function of the number of clusters. This method is the most preferred method that shows us the big picture. Within cluster sum of square (WCSS) is calculated by taking the sum of the squared distance of each point from the cluster centroid. The Elbow method states that the point where the amount of change in the WCSS decreases, the elbow point, is the optimum point (Salvador & Chan, 2003). The optimum k value is determined as seven with the Elbow method. The Elbow graph of stores distributed on busy days and not busy days are shown in Fig. 4.

Table 3
Demand amounts of sub-problems in first week (pallet)

Stores	"1"	"2"	"3"	"4"	Stores	"1"	"2"	"3"	"4"
1	5	8	5	8	40	5	8	5	8
2	7	12	7	12	41	5	8	5	8
3	9	14	9	14	42	3	6	0	0
4	7	12	7	12	43	5	8	5	8
5	9	14	9	14	44	5	8	5	8
6	3	6	0	0	45	3	6	0	0
7	7	12	7	12	46	5	8	5	8
8	9	14	9	14	47	7	12	7	12
9	7	12	7	12	48	7	12	7	12
10	5	8	5	8	49	5	8	5	8
11	5	8	5	8	50	9	14	9	14
12	7	12	7	12	51	9	14	9	14
13	5	8	5	8	52	5	8	5	8
14	5	8	5	8	53	5	8	5	8
15	7	12	7	12	54	9	14	9	14
16	5	8	5	8	55	5	8	5	8
17	5	8	5	8	56	5	8	5	8
18	7	12	7	12	57	7	12	7	12
19	5	8	5	8	58	5	8	5	8
20	5	8	5	8	59	7	12	7	12
21	9	14	9	14	60	7	12	7	12
22	5	8	5	8	61	5	8	5	8
23	5	8	5	8	62	7	12	7	12
24	3	6	0	0	63	9	14	9	14
25	5	8	5	8	64	5	8	5	8
26	5	8	5	8	65	5	8	5	8
27	3	6	0	0	66	5	8	5	8
28	5	8	5	8	67	5	8	5	8
29	5	8	5	8	68	9	14	9	14
30	3	6	0	0	69	5	8	5	8
31	5	8	5	8	70	7	12	7	12
32	5	8	5	8	71	3	6	0	0
33	3	6	0	0	72	5	8	5	8
34	5	8	5	8	73	7	12	7	12
35	5	8	5	8	74	3	6	0	0
36	3	6	0	0	75	5	8	5	8
37	5	8	5	8	76	9	14	9	14
38	5	8	5	8	77	3	6	0	0
39	3	6	0	0	78	7	12	7	12

"1": Sub-problem 1, "2": Sub-problem 2, "3": Sub-problem 3, "4": Sub-problem 4

Table 4 and Table 5 show the clustered stores distributed on busy days and not busy days, respectively.

Table 4
Clustered stores distributed on busy days

Clusters of K-Means Algorithm							Clusters of K-Medoids Algorithm						
#1	# 2	# 3	# 4	# 5	# 6	# 7	#1	# 2	# 3	# 4	# 5	# 6	# 7
17	49	3	10	60	7	1	30	1	4	72	7	58	13
18	50	13	11	63	8	2	45	2	5	73	8	59	15
19	51	14	12	64	9	4	49	3	6	75	9	60	16
20	58	15	68	65	66	5	50	14	10	78	66	61	21
21	59	16	69	77	67	6	51	17	11	35	67	62	24
22	61	24	70			30	52	18	12	37		63	26
23	62	26	71			52	53	19	68			64	27
25		27	73			53	54	20	69			65	29
28		29				54	55	22	70			77	32
31		32				56	56	23	71				34
33		34				57	57	25					38
36		35					76	28					40
38		37						31					41
39		40						33					43
41		43						36					44
42		47						39					47
44		48						42					48
45		72						46					
46		74						74					
55		75											
76		78											

Table 5
Clustered stores distributed on not busy days

Clusters of K-Means Algorithm							Clusters of K-Medoids Algorithm						
#1	# 2	# 3	# 4	# 5	# 6	# 7	#1	# 2	# 3	# 4	# 5	# 6	# 7
3	49	50	1	10	60	2	1	2	10	7	13	58	49
13	58	51	4	11	63	5	17	3	11	8	15	59	50
14	59	52	17	12	64	7	18	4	12	9	16	60	51
15	61	53	18	66	65	8	22	5	66	14	21	61	52
16	62	54	19	67		9	23	19	67	72	26	62	53
26		56	20	68		75	25	20	68	73	29	63	54
29		57	21	69		76	38	28	69	75	32	64	56
32			22	70		78	41	31	70	78	34	65	57
34			23	73			44	76			35		
35			25				46				37		
37			28				55				40		
40			31								43		
43			38								47		
47			41								48		
48			44										
72			46										
			55										

Each cluster given in Table 4 and Table 5 is solved with HNN, which the stages are given in section 4. The proposed HNN is coded in Python and run on a PC with an Intel Core I7 CPU 4.5 GHz processor and 16.00 GB RAM. Extensive computational experiments are carried out for deciding on the standard parameter setting of HNN. N is taken as the total number of customers. A and D are calculated as N*N and N/2, respectively. Step size is set to 0.0001. The neural network initial value (U_0) is taken as 0.1. The number of iterations is determined as 10000.

In this section, because of page limitations and the manuscript's readability, the obtained results that the best results on 30 runs considering first-week demand of distribution of fresh products on busy and not busy days (sub-problem 1 and 3) are given in Table 6 and 7 in detail, respectively. The results for 20 are summarized in Table 8. The transposition matrixes of the GVRP representing HNN are demonstrated in Appendix A.

Table 6
Results of HNN for sub-problem 1

	Cluster	Vehicle	Routes	Total Demand	Total CO_2 Emissions (ton)
HNN-C1	1	1	79-23-76-31-33-19-39-46-20-79	40	213003.7
		2	79-17-36-28-25-21-38-44-79	37	207161.8
		3	79-55-41-42-45-18-22-79	28	207386.5
	2	1	79-58-49-51-50-59-79	35	173683.4
		2	79-62-61-79	12	132790.3
	3	1	79-35-32-37-34-72-75-78-79	37	231877.4
		2	79-3-14-74-13-24-29-26-79	35	235697.1
		3	79-15-16-40-27-48-43-47-79	39	220643.1
	4	1	79-70-11-12-10-73-71-69-79	39	209858.1
		2	79-68-79	9	170313.1
	5	1	79-64-65-77-63-60-79	29	259514
	6	1	79-66-7-9-8-67-79	33	291419.6
	7	1	79-53-54-57-52-56-30-79	34	187164.6
		2	79-2-6-5-4-1-79	31	185816.5
			TOTAL	**438**	**2926329.1**
HNN-C2	1	1	79-54-53-51-50-49-79	37	182895.6
		2	79-76-45-55-30-56-52-57-79	37	220867.7
	2	1	79-20-14-74-3-36-23-18-79	37	227833.1
		2	79-25-28-19-33-31-39-46-2-79	38	208060.6
		3	79-1-22-42-17-79	18	193905.3
	3	1	79-69-70-12-10-11-68-79	38	191883.1
		2	79-4-5-6-71-79	22	179300.6
	4	1	79-73-78-75-72-35-37-79	34	248054.9
	5	1	79-66-7-9-8-67-79	33	291419.6
	6	1	79-64-65-58-77-62-61-59-79	37	281084
		2	79-60-63-79	16	74596.23
	7	1	79-32-34-13-24-47-15-43-79	37	229181.2
		2	79-48-27-40-16-21-38-44-79	39	216823.4
		3	79-41-29-26-79	15	213003.7
			TOTAL	**438**	**2958908.8**

HNN-C1: HNN with K-Means algorithm, HNN-C2: HNN with K-Medoids algorithm

Table 7
Results of HNN for sub-problem 3

	Cluster	Vehicle	Routes obtained by HNN	Total Demand	Total CO_2 Emissions (Ton)
HNN-C1	**1**	1	79-48-16-40-15-43-47-79	36	222440.6
		2	79-35-32-37-34-72-14-13-79	35	235697.1
		3	79-3-29-26-79	19	216598.7
	2	1	79-58-61-49-59-62-79	29	175705.6
	3	1	79-51-54-57-56-52-53-79	40	190310.3
		2	79-50-79	9	150540.6
	4	1	79-41-38-44-46-23-1-4-79	37	205139.6
		2	79-17-20-28-19-31-21-25-79	39	209858.1
		3	79-55-18-22-79	17	203791.5
	5	1	79-10-11-12-66-67-73-69-79	39	231877.4
		2	79-68-70-79	16	173234.0
	6	1	79-65-64-63-60-79	26	126274.3
	7	1	79-75-78-76-2-5-79	37	230079.9
		2	79-7-9-8-79	23	264457.1
			TOTAL	**402**	**2836004.8**
HNN-C2	**1**	1	79-23-25-38-44-41-22-55-1-79	40	201544.6
		2	79-18-17-46-79	17	200645.9
	2	1	79-2-3-76-31-19-28-79	40	207611.2
		2	79-20-4-5-79	21	185816.5
	3	1	79-70-11-12-66-10-67-69-79	39	199073.1
		2	79-68-79	9	170313.1
	4	1	79-7-9-8-73-75-79	35	304226.8
		2	79-78-72-14-79	17	217497.4
	5	1	79-21-16-48-43-15-47-79	40	220643.1
		2	79-34-37-32-35-40-13-26-29-79	40	241314.3
	6	1	79-65-58-61-59-62-63-79	38	137059.3
		2	79-60-64-79	12	95042.8
	7	1	79-51-53-57-52-56-54-79	40	191433.7
		2	79-49-50-79	14	156831.8
			TOTAL	**402**	**2729053.6**

HNN-C1: HNN with K-Means algorithm, HNN-C2: HNN with K-Medoids algorithm

Table 8
Total vehicles and CO_2 emissions of sub-problems

	Sub-problem 2				**Sub-problem 1**				**Sub-problem 4**				**Sub-problem 3**			
Wee	**HNN-C1**		**HNN-C2**		**HNN-C1**		**HNN-C2**		**HNN-C1**		**HNN-C2**		**HNN-C1**		**HNN-C2**	
k	**V**	**TE (Ton)**	**V**	**TE (Ton)**	**V**	**TE (Ton)**	**V**	**TE (Ton)**	**V**	**TE (Ton)**	**V**	**TE (Ton)**	**V**	**TE (Ton)**	**V**	**TE (Ton)**
1	14	2926329.	14	2958908.	21	4181433.	22	4399155.	20	2836004.	21	2729053.	14	3816540.	14	3974945.
2	15	3128997.	14	2949471.	21	4314672.	23	4616428.	20	2694676.	21	2505686.	13	3994493.	13	4086165.
3	13	2868135.	13	2885885.	22	4416456.	22	4492849.	20	2836454.	20	2729699.	14	3930906.	15	3900798.
4	14	3060467.	14	3033280.	22	4466112.	22	4540933.	21	2800504.	19	2907680.	14	3883244.	15	3822157.
5	14	2997330.	15	3183371.	23	4582050.	23	4579579.	21	2685913.	20	2779608.	13	3872909.	14	3891586.
6	15	3148095.	13	2969469.	22	4524755.	23	4661814.	21	2673780.	20	2728590.	13	3803466.	14	3945061.
7	13	2801874.	14	3026989.	21	4403424.	23	4685631.	20	2682767.	20	2897569.	13	3880801.	15	3870241.
8	14	3006767.	14	3049457.	21	4226595.	22	4196262.	20	2823871.	19	2907455.	14	3784157.	15	3702848.
9	13	2615339.	14	3000925.	22	4576658.	22	4553290.	21	2851957.	20	2718493.	14	3905039.	14	3904618.
10	13	2887458.	14	3027438.	21	4349274.	22	4619124.	22	2670185.	19	2684986.	13	3878961.	15	3788904.
11	15	3204941.	13	2942281.	22	4380281.	22	4507904.	20	2809941.	20	2930373.	14	3803944.	15	3825303.
12	15	3229881.	14	3013058.	23	4564300.	22	4534866.	21	2877572.	20	2950370.	14	4018745.	15	3925289.
13	14	2927677.	13	2839599.	22	4362980.	23	4649681.	19	2652210.	21	2823871.	13	3742618.	14	3952476.
14	13	2850160.	14	3038897.	23	4606766.	21	4409940.	21	2659175.	18	2714673.	13	3661464.	14	3562419.
15	14	3048334.	13	2859147.	21	4544752.	22	4561379.	21	2822523.	20	2948797.	14	3999436.	15	3902146.
16	13	2883189.	14	3024742.	22	4491052.	21	4347252.	21	2702765.	20	2908803.	13	3882331.	15	3948881.
17	14	2946101.	14	3029011.	22	4372866.	23	4631706.	22	2794212.	19	2740484.	14	3771996.	15	3676560.
18	13	2775563.	14	3020922.	22	4506780.	22	4494872.	22	2689508.	19	2763655.	13	3625008.	14	3768907.
19	15	3149893.	14	3031482.	21	4258950.	21	4302315.	21	2847014.	19	2742310.	14	3645707.	14	3795420.
20	14	2992387.	15	3162250.	21	4228842.	22	4450384.	21	2808143.	20	2942731.	14	3807960.	15	3928659.

VN: Total vehicle number, TE: Total CO2 emissions, HNN-C1: HNN with K-Means algorithm, HNN-C2: HNN with K-Medoids algorithm

5.2. Statistical analysis

In this section, the computational results obtained by the HNN-C1 and HNN-C2 are compared to the results obtained by the CFRS proposed by Comert et al. (2018). They used the same dataset using a different objective function which minimizes total distance. So, the results of 20 weeks were normalized to apply to the Friedman Test. Only sub-problem 1' Friedman test is given in detail. The rest of them are summarized in Table 11.

· *Comparison in terms of using K-means algorithm (C1)*

The null and alternative hypotheses are determined as follows:

H_0: There is no significant difference between the performance of the two methods (HNN-C1 and CFRS-C1) with a 95% confidence interval.
H_a: There is a significant difference between the performance of the two methods (HNN-C1 and CFRS-C1) with a 95% confidence interval.

Table 9 shows the Friedman test results. The Friedman test statistics are highly significant (p 0.05), so the null hypothesis can be accepted, and the H_0 hypothesis is accepted. In other words, there is no significant difference between the two methods (HNN-C1 and CFRS-C1).

Table 9
Friedman Statistics for a sub-problem 1 (C1)

Total *N*	20
Test Statistic	0.000
Degrees of freedom	1
***p*-value**	1.000

- *Comparison in terms of using K-medoids algorithm (C2)*

The null and alternative hypotheses are determined as follows:

H_0: There is no significant difference between the performance of the two methods (HNN-C2 and CFRS-C2) with a 95% confidence interval.
H_a: There is a significant difference between the performance of the two methods (HNN-C2 and CFRS-C2) with a 95% confidence interval.

Table 10 shows the Friedman test results. The Friedman test statistics are highly significant ($p \geq 0.05$), so the null hypothesis can be accepted, and the H_0 hypothesis is accepted. In other words, there is no significant difference between the two methods (HNN-C2 and CFRS-C2).

Table 10
Friedman Statistics for a sub-problem 1 (C2)

Total *N*	20
Test Statistic	0.800
Degrees of freedom	1
***p*-value**	0.371

The other sub-problem' Friedman test results, are given in Table 11.

Table 11
Friedman Test results of sub-problems

Sub-problem	Compared Algorithms	Responses
1	HHNC1 and CFRS-C1	No Difference
	HHNC2 and CFRS-C2	No Difference
2	HHNC1 and CFRS-C1	No Difference
	HHNC2 and CFRS-C2	No Difference
3	HHNC1 and CFRS-C1	No Difference
	HHNC2 and CFRS-C2	No Difference
4	HHNC1 and CFRS-C1	No Difference
	HHNC2 and CFRS-C2	No Difference

HNN-C1: HNN with K-Means algorithm, HNN-C2: HNN with K-Medoids algorithm, CFRS-C1: CFRS with K-Means algorithm, CFRS-C2: CFRS with K- Medoids algorithm

When comparing HNN and CFRS proposed by Comert et al. (2018) in terms of their performance, the statistical analysis illustrates no significant differences among them in the four sub-problems, as shown in Table 11. The authors know that HNN in solving VRP has not yet been addressed in the literature. Here, we compared the solution performance of HNN with CFRS proposed by Comert et al. (2018) and concluded that HNN obtained competitive results.

6. Conclusion

In this study, a newly proposed method based on combining K-Means and K-Medoids algorithms with HNN is developed to solve large-scale problems. The method is adopted for GVRP. Firstly, stores are grouped into clusters using K-means and K-medoids algorithms. Thus, we enable the solution of large-scale problems. Then, we route each cluster with HNN to minimize CO_2 emissions. The proposed HNN-based method is applied to a real-life example to illustrate its effectiveness. This real-life problem is related to a distribution GVRP meeting the weekly demands of a supermarket chain with 78 stores and one depot. The results obtained using HNN and CFRS proposed by Comert et al. (2018) are compared using the Friedman test. The test result shows the applicability and effectiveness of the developed HNN-based method in VRP. Future work may consider electric vehicle routing problems from alternative fuel vehicles with the proposed HNN and clustering algorithms-based method.

References

Abdel-Moetty, S. M. (2010). Traveling salesman problem using neural network techniques. *In The 7th International Conference on Informatics and Systems (INFOS).* Cairo, Egypt.

Comert, S.E., Yazgan, H.R., Kır, S., & Yener, F. (2018). A cluster first-route second approach for a capacitated vehicle routing problem: a case study. *International Journal of Procurement Management, 11*(4), 399-419.

Dantzig, G., & Ramser, J. (1959). The Truck Dispatching Problem. *Management Science, 6*(1), 80-91.

Erdoğan, S., & Miller-Hooks, E. (2012). A Green Vehicle Routing Problem. *Transportation Research Part E: Logistics and Transportation Review, 48*(1), 100–114.

Han, J., & Kamber, M. (2001). *Data Mining and Concepts Techniques*. San Francisco: Morgan Kaufmann Publishers Inc.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America, 79*(8), 2554–2558.

Hopfield, J.J., & Tank, D.W. (1985). Neural Computation of Decisions in Optimization Problems. *Biological Cybernetics, 52*, 141-152.

Jain, A.K., Mao, J., & Mohiuddin, K.M. (1996). Artificial Neural Networks: A Tutorial. *Computer, 29*(3), 31-44.

Jolai, F. & Ghanbari, A. (2010). Integrating data transformation techniques with Hopfield neural networks for solving travelling salesman problem. *Expert Systems with Applications, 37*(7), 5331-5335.

Kamgar-Parsi, B., & Kamgar-Parsi, B. (1992). Dynamical Stability and Parameter Selection in Neural Optimization. *In Proceedings International Joint Conference on Neural Networks* (Vol. 4, pp. 566-571).

Kaufman, L., & Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Son.

Lai, W.K., & Coghill, G.G. (1992). Genetic Breeding of Control Parameters for the Hopfield/Tank Neural Net. *In Proceedings International Joint Conference on Neural Networks* (Vol. 4, pp. 618-623).

Lenstra, J.K., & Rinnooy Kan, A.H.G. (1981). Complexity of vehicle routing and scheduling problems. *Networks, 11*(2), 221–227.

Li, R., Qiao, J., & Li, W. (2016, June). A Modified Hopfield Neural Network for Solving TSP Problem. *In 2016 12th World Congress on Intelligent Control and Automation (WCICA)* (pp. 1775-1780).

Lippmann, R.P. (1987). An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine, 4*(2), 4-22.

Luo, Y. (2019, July). Design and Improvement of Hopfield network for TSP, *In AICS 2019: Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science* (pp. 79–83).

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations*. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281–297).

Nie, Y., & Li, Q. (2013). An eco-routing model considering microscopic vehicle operating conditions. *Transportation Research Part B Methodological, 55*, 154–170.

Oshima, R., Uwate, Y., & Nishio, Y. (2015, December). Hopfield Neural Network with Synaptic Pruning for TSP, In *International Symposium on Nonlinear Theory and its Applications NOLTA2015* (pp. 105-108).

Rao, V. B. (1995). *Neural networks and fuzzy logic*. M&T Books, IDG Books Worldwide, Inc.

Rojas, R. (1996). *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin.

Salvador, S. & Chan, P. (2003). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms*. In 16th IEEE International Conference on Tools with Artificial Intelligence.* Boca Raton, FL, USA.

Sarwar, F., & Bhatti, A.A. (2012). Critical analysis of hopfield's neural network model for TSP and its comparison with heuristic algorithm for shortest path computation. *In Proceedings of 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST)* (pp. 111-114).

Tan, K.C., Tang, H., & Ge, S.S. (2005). On Parameter Settings of Hopfield Networks Applied to Traveling Salesman Problems. *IEEE Transactions on Circuits and Systems—I: Regular Papers, 52*(5), 994-1002.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining*. Addison Wesley.

Toth, P., & Vigo, D. (2002). *The vehicle routing problem, in: Discrete Mathematics and Applications*. Philadelphia.

Van den Bout, D.E., & Miller, T.K. (1988). A Travelling Salesman Objective Function that works. *In Proceedings IEEE International Conference on Neural Networks* (Vol. 2, pp. 299-303).

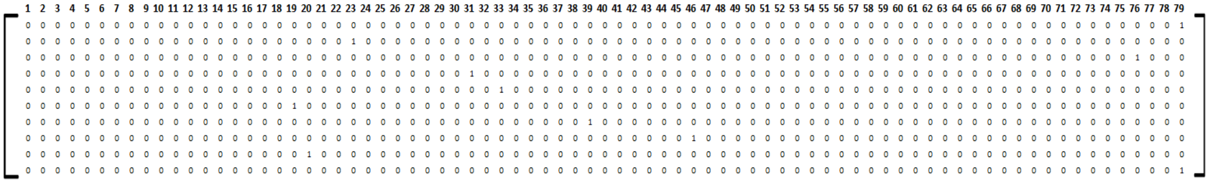
Wilson, G.V. & Pawley, G.S. (1988). On the stability of the Travelling Salesman Problem algorithm of Hopfield and Tank. *Biological Cybernetics, 58*, 63–70.

Wu, H., & Yang, Y. (2004, December). Application of continous hopfield network to solve the TSP. *In 8th International Conference on Control, Automation, Robotics and Vision Kunming (pp. 2258-2263).*

Xu, R. & Wunsch, D. (2005). Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks, 16*(3), 645-678.

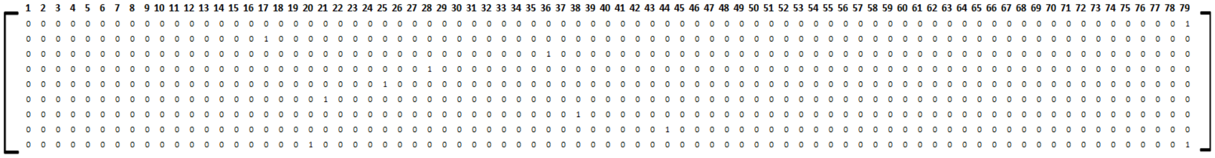
Zhong, C., Luo, C., Chu, Z., & Gan, W. (2017). A continuous hopfield neural network based on dynamic step for the traveling salesman problem. *In International Joint Conference on Neural Networks (IJCNN) (pp. 3318-3323).*

Appendix A. The transposition matrix of the GVRP representing HNN



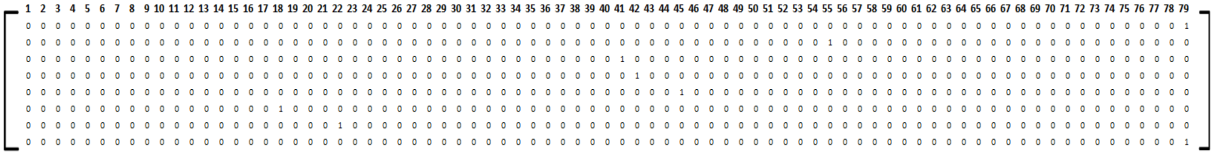
$$\left[\begin{array}{ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

Route of Vehicle 1 in Cluster 1 found by HNN-C1 = 79-23-76-31-33-19-39-46-20-79



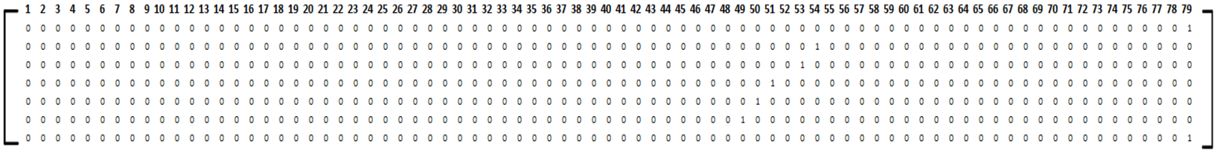
$$\left[\begin{array}{ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

Route of Vehicle 2 in Cluster 1 found by HNN-C1 = 79-17-36-28-25-21-38-44-79



$$\left[\begin{array}{ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

Route of Vehicle 3 in Cluster 1 found by HNN-C1 = 79-55-41-42-45-18-22-79



$$\left[\begin{array}{ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

Route of Vehicle 1 in Cluster 1 found by HNN-C2 = 79-54-53-51-50-49-79

$$\left[\begin{array}{ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

Route of Vehicle 2 in Cluster 1 found by HNN-C2 = 79-76-45-55-30-56-52-57-79