World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:3, 2011

A Subjectively Influenced Router for Vehicles in a Four-Junction Traffic System

Anilkumar Kothalil Gopalakrishnan

***Abstract*—**A subjectively influenced router for vehicles in a four-junction traffic system is presented. The router is based on a 3-layer Backpropagation Neural Network (BPNN) and a greedy routing procedure. The BPNN detects priorities of vehicles based on the subjective criteria. The subjective criteria and the routing procedure depend on the routing plan towards vehicles depending on the user. The routing procedure selects vehicles from their junctions based on their priorities and route them concurrently to the traffic system. That is, when the router is provided with a desired vehicles selection criteria and routing procedure, it routes vehicles with a reasonable junction clearing time. The cost evaluation of the router determines its efficiency. In the case of a routing conflict, the router will route the vehicles in a consecutive order and quarantine faulty vehicles. The simulations presented indicate that the presented approach is an effective strategy of structuring a subjective vehicle router.

***Keywords*—**Backpropagation Neural Network, Backpropagation algorithm, Greedy routing procedure, Subjective criteria, Vehicle priority, Cost evaluation, Route generation

I. Introduction

THIS This paper describes the application of a subjectively influenced vehicle router for vehicles in a four-junction traffic system. The aim is to show the feasibility of a subjective approach in a vehicle routing problems that have been established in the form of job schedulers through [1], [2], [3]. When compared to the computational complexity of the real world vehicle traffic problems [4]-[5], the proposed vehicle routing approach focuses on the theoretical simulation of a subjective vehicle router in a four-junction traffic system which may be used in vehicle routing application in future. In this context, the utilization of the parallel processing ability of the BPNN [6] and the significance of greedy algorithms[7]-[8] can be able to formulate a subjective vehicle router for generating reasonable vehicle routing solutions for a problem with a queue of n waiting vehicles in a four-junction traffic system. That is, the combined algorithmic structure has an ability to solve a vehicle routing problem with less computational complexity.

Since the neural networks alone cannot be adequately used to route vehicles in a vehicle routing problem, the application of a proper vehicle routing procedure along with the neural network is essential. Due to this reason, a combination of a 3-layer BPNN and a greedy vehicle routing procedure are used to generate results that are reasonable.

Anilkumar Kothalil Gopalakrishnan is a lecturer in the Department of Computer Science, PDCRL, Faculty of Science and Technology, Assumption University, Bangkok, Thailand 10240 (phone: +66-891351711; e-mail: anil@scitech.au.edu).

The priority order of vehicles assigned by the router depends on the given vehicle selection criteria called *subjective criteria* based on the user's requirements and is described in section IV. That is, the router identifies each waiting vehicle based on their priority values and the priority is assigned to vehicles waiting in front of each junction of the traffic system. Finding priorities of vehicles, especially in this case, has not been formally described and that is what this paper attempts to do. The subjective nature of the router may vary based on the needs of different users. The router handles a set of eight vehicles concurrently at a time through the junction nodes of the traffic system (see Fig. 1) and soon after that it routes the next set of eight successive vehicles from the queue of the junction without rendering any junction idle.

The dataset generated from the subjective criteria (of the user) for the initial training of the BPNN is called *seen data*. These seen dataset and the vehicle routing procedure are meant to carry the details of how the vehicle selection process happens and the vehicles are to be selected by the four junction nodes of the traffic system. Once the vehicle router is tuned with a proper vehicle selection criteria and greedy vehicle routing procedure for a problem with a set of vehicles (as many as eight vehicles can be waited at each junction at a time) it can act as a vehicle routing agent. In this case, the user is replaced by the router permanently. Furthermore, this router employs a greedy type algorithm which is, by their characteristic, quicker and it does not need to consider the details of all solution alternatives.

The greedy procedure selects vehicles from their junction node queue based on their priorities and then detects their routes (the router generates the junction routes of each vehicle in a random basis). Once the routes of each vehicle have been estimated, the router routes a set of waiting vehicles in a concurrent manner in order to minimize their overall junction clearing time. That is, the router distinguishes a set of eight vehicles (each junction node of the traffic system holds a set of two vehicles and there are a total of eight vehicles at a time) from their starting nodes based on their priorities. The router checks whether there are any route conflicts among the vehicles during their concurrent routing process. In the case of a route conflict, the router will route the conflict route vehicles in a consecutive order without causing any collision or violating traffic laws. The router is able to tolerate a faulty vehicle at a time by quarantining it inside the junction path of the traffic system without jeopardizing the entire vehicle flow through the system (the details of the vehicle fault tolerance of the router is described in section IX.

II. STRUCTURE OF THE FOUR-JUNCTION TRAFFIC SYSTEM

The performance of the vehicle router is based on the structure of a four-junction traffic system and is shown in Fig. 1, where, N_1, N_2, N_3 and N_4 are the four junction nodes of the traffic system and V_i, V_j, V_k, V_l, V_m, V_n, V_o and V_p are the indication of vehicles at entrance of the nodes of the traffic system at a time, $(i, j, k, l, n, o, p) \in 1,\ldots,n$). Similarly, the variables D_1, D_2, D_3 and D_4 are the distances between the junction nodes N_1 and N_2, N_1 and N_3, N_3 and N_4, and N_2 and N_4 respectively. The router is supposed to be situated at the centre of the four-junction traffic system (as shown in Fig.1) for controlling the junction nodes as well as the vehicles at a time and also to control the flow of selected vehicles from the junction nodes in a real time basis. The additional lounge space is reserved in each path of the traffic system is for storing faulty vehicles, called a *quarantine* area.

The router differentiates each vehicle from the junction nodes based on priorities (each nodes has two bidirectional entrances). The priority of a vehicle is assigned based on four parameters; *earliest-arrival*, *critical type*, *vehicle size* and *speed limit* (details of the vehicle priority detection is described in section IV).The main function of the router is to control each junction node of the system for facilitating a smooth flow of vehicles through the system based on their routes.

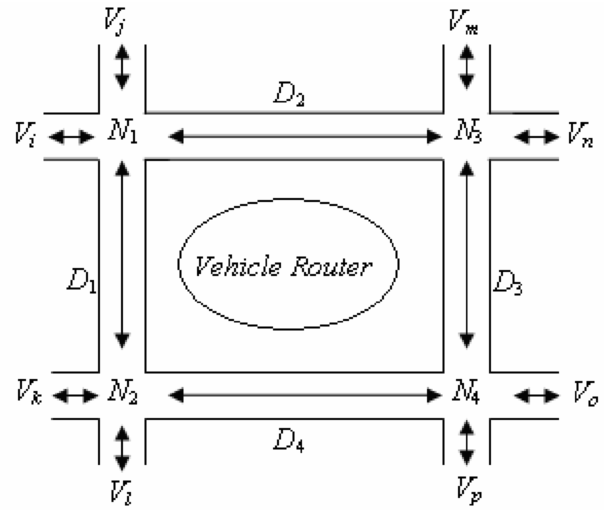

Fig. 1 Structure of the four-junction traffic system with vehicle router

Each junction node of the traffic system handles two vehicles at a time and there are a total of eight vehicles available at a time in front of the system. In order to achieve a reasonable overall junction clearing time for a set of eight vehicles at a time, each junction node vehicles must be prioritized based on the parameters. That is, at any given time, one of the two waiting vehicles from each junction node is selected by the router based on the priorities. Therefore, four vehicles in the traffic are allowed to route through the system at a time. Meanwhile, the router selects and routes the next four vehicles from the junction nodes concurrently without causing any delay. All non-conflict route vehicles are always routed in a concurrent manner through the system and conflict route vehicles are routed in a consecutive order without causing any collision or traffic law violations.

In order to estimate the priorities of vehicles, the router collects four parameters of each vehicle from their starting nodes. The four parameters of each vehicle are collected by the router on a real-time basis defined as follows:

- *A*, *earliest-arrival* of a vehicle is a numerical value indicating the arrival instant of a vehicle before it starts to cross its junction node.
- *C*, *critical type* of a vehicle is a numerical value indicating the critical nature of a waiting vehicle and it means that the vehicle is an ambulance or a fire engine.
- *V*, *vehicle-size* of a vehicle is a numerical value indicating whether the waiting vehicle is a small or a big.
- *S*, *speed limit* of a vehicle is a numerical value indicating the speed limit of a vehicle when it is in the system. The speed of each vehicle in the system is limited based on their size that is, if it is large, then a certain speed limit will be imposed on it by the router.

In order to apply the subjective router concept to a vehicle routing problem, the following conditions are applicable:

i. Junction nodes of the traffic system (N_1, N_2, N_3 and N_4) are independent to each other. Each junction node is allowed to handle only one vehicle at a time. That is, out of two waiting vehicles only one vehicle is allowed to cross the nodes at a time.
ii. It assumes that vehicles have different sizes and masses and there is a fixed space between each vehicle during their flow into the node junction system.
iii. There is a two-way passage at the entrance of each junction node that allows bidirectional flow of vehicles from their starting nodes. Besides, inside the system, there should be only one lane bidirectional path that allows only one vehicle to cover the entire route inside the system from its starting node to destination node at a time (see Fig.1).
iv. There are a total of eight vehicles allowed to wait in front of each junction node at a time.
v. Any vehicle should not be allowed to cross a junction node from where it is started (there is no Hamiltonian circuit is allowed inside the traffic system).
vi. The route of each vehicle is determined by the router based on their starting nodes.
vii. Each path inside the traffic system should have a quarantine area for holding faulty vehicles.

This routing technique conditions are not specifically defined with respect to the existing practical systems.

III. STRUCTURE OF THE VEHICLE ROUTER

The vehicle router has a combined structure of a 3-layer BPNN with topology 4-20-1(one input layer with four inputs, one hidden layer with twenty neurons, and an output layer with one output) and a greedy routing procedure. The performance of the BPNN is evaluated by a convergence test and the details of the convergence test are described in section V. At each time, from the vehicle priority list, the router will generate two more priority lists; the first list is max*_list*, which contains the descending order of the four most priority vehicles out of two waiting vehicles from each junction node. Similarly, the second priority list is the *min_list*, which contains the descending order of the four lower priority vehicles out of two waiting vehicles from each node. The structure described is as shown in Fig. 2 below.

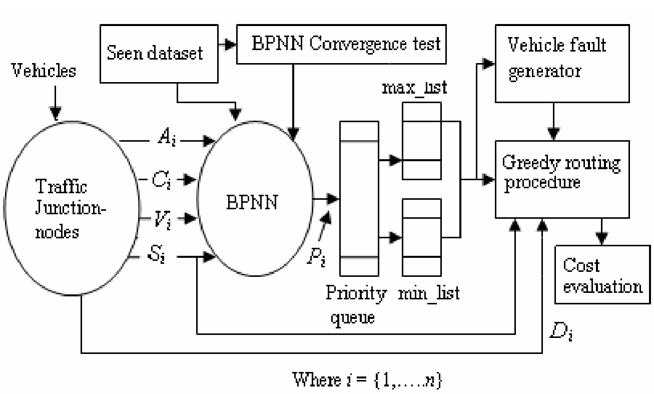

Fig. 2 Structure of the vehicle scheduler along with the traffic system

At the beginning of a routing process, the router selects a set of four vehicles from the *max_list* and routes each vehicle in a concurrent manner based on their assigned priorities and routes (the details of vehicle route generation are described in section VII). That is, all the four vehicles from the *max_list* are selected and routed from their respective starting nodes (the details of the junction node selection of vehicles are described in section VI).

The most priority vehicles would get the first consideration during their concurrent flow; that is, the first vehicle from the *max_list* will be allowed to route through the system without any conflict with the vehicles that come after it in the flow of traffic. Similarly, as soon as the last vehicle from the *max_list* starts its routing, the first vehicle from the *min_list* starts its routing concurrently with the last vehicle of the *max_list*; all the four vehicles from the *min_list* are selected and routed respectively. The router would also check to find out whether the last vehicle from the *min_list* has started its routing or not. If so, then the router would determine the over all junction clearing time of eight vehicles and evaluates that with a predefined cost value (the details of cost evaluation are described in section VIII) in order to validate the performance of the router. It also monitors whether there is any route conflict with adjacent vehicles before allowing them to route through the junction nodes of the system. If there is a route conflict, the router reroutes the affected vehicles consecutively through the system to ensure that the concurrent routing process never causes any node violations.

The four parameters used as inputs to the BPNN are, (1) A_i, earliest-arrival of vehicle *i*, (2) C_i, critical type of vehicle *i,* (3) V_i, vehicle-size of vehicle *i* and (4) S_i, speed limit of vehicle *i*. The BPNN also has one output: P_i, priority of vehicle *i* with four input variables and one output variable as shown in Fig. 3. In a typical 3-layer BPNN, the computation time will be asymptotically $\Theta(ih + ho)$, where *i, h*, and *o* are the number of input neurons, hidden neurons and the output neurons, respectively [6].

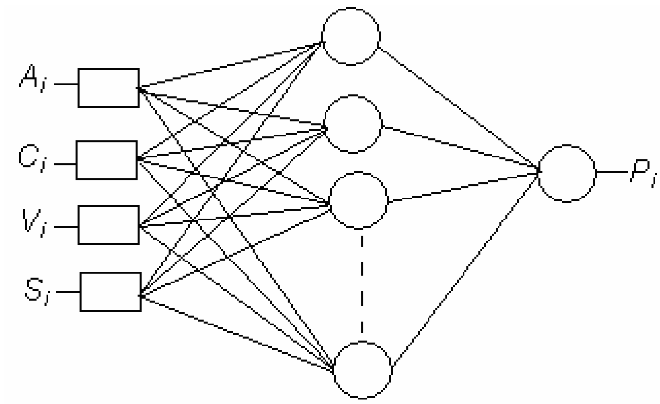

Fig. 3 A 3-layer BPNN with four inputs and an output

IV. Subjective Criteria for Vehicle Priority

In order to generate a seen dataset for the initial training of the BPNN of the router, there are five numerical values with their proper linguistic terms applied along with the parameters of each vehicle. The four parameters of a vehicle with their numerical values and linguistic terms are as follows:

a). A_i is the *earliest-arrival* of vehicle *i* with values: [0.1 (*very early*), 0.3 (*early*), 0.5 (*not early)*, 0.7 (*late*), and 0.9 (*very late*)].

b). C_i is the *critical type* of vehicle *i* with values: [0.1 (*very low*), 0.3 (*low*), 0.5 (*not low)*, 0.7 (*high*), and 0.9 (*very high*)].

c). V_i is the *vehicle-size* of vehicle *i* with values: [0.1 (*very small*), 0.3 (*small*), 0.5 (*not small)*, 0.7 (*large*), and 0.9 (*very large*)].

d). S_i is the *speed limit* of vehicle *i* with values: [0.1 (*very low*), 0.3 (*low*), 0.5 (*not low)*, 0.7 (*high*), and 0.9 (*very high*)].

e). The output of the BPNN, P_i is the *priority* of vehicle *i*, ranging from 0.01(*very low*) to 0.99 (*very high*).

Based on the numerical values and linguistic terms of the parameters of a vehicle, sample criteria for generating the seen dataset for finding vehicle priority are as listed below:

- Speed limit of a vehicle is inversely proportional to its size.
- A vehicle with *high* critical type value will achieve a *high* priority.
- A large/*very large* size vehicle never holds a *very high/high* critical type value.
- A vehicle with *very early/ early* earliest-arrival will get more priority than a vehicle with *late/very late* earliest-arrival value with respect to its critical type value.
- A vehicle with any earliest-arrival value, a *high/very high* critical type, a *very small/small/not small* vehicle-size and a speed limit value is based on the vehicle type and its priority will depend on the critical type value.

- A vehicle with any earliest-arrival, a *high/very high* critical type, a *large/very large* vehicle-size and a speed limit based on its vehicle-size; its priority will depend on its earliest-arrival value.
- A vehicle with a *very early/early* earliest-arrival, a *very low/low* critical type, any vehicle-size and a speed limit value based on its vehicle-size; will have a *very small/small* priority.
-A vehicle with a *late/very late* earliest-arrival, a *very low/low/not low* critical type, a *large/very large* vehicle-size and a speed limit based on its vehicle-size; will have priority depending on its critical type value.
-A vehicle with a *very early/early* earliest-arrival, a *very low/low* critical type, a *very large/large* vehicle-size, and a speed limit value based on its vehicle-size; will have priority that is inversely proportional to its critical type.
- A vehicle with a *very early/early* earliest-arrival, a *very low/low* critical type, a *very small/small* vehicle-size and a speed limit value based on its vehicle-size; will have a *very high / high* priority.
- A vehicle with a *not early/late/late* earliest-arrival, a *very low/low* critical type, a *very large/large* vehicle-size and any speed limit based on its vehicle-size; will have a *very small/small* priority.

A sample seen dataset with fifty inputs and their respective output data patterns based on the above subjective criteria is shown in Appendix.

V. Convergence Test of the BPNN

The initial data training of the BPNN depends on the size of the seen data and the topology of the network. Once the BPNN is trained until its MSE is 0.001, it is essential to ensure that the BPNN is free from problems such as 'over-fitting' and local maxima during its initial training process. The details of the convergence test of the BPNN are given here:

(1). Train the BPNN with the seen dataset by proper training parameters such as *learning rate* (α) and *momentum term* (β) until its MSE is reduced to a value 0.001.
(2). Select the input data pattern from the seen dataset after its training.
(3). Select the output data from the seen dataset after its training (say, Q) similarly to step (ii).
(4). Input the selected data pattern (from step ii) to the BPNN and find its output by the BPNN (say, Q').

A *similarity measure,* $S(Q, Q')$ is the convergence test of the BPNN and can be interpreted as follows: if $S(Q, Q')$ is above or equal to +0.99, then the selected BPNN is an acceptable one. Otherwise, the BPNN is considered as unacceptable and therefore repeat its training with different parameters and topologies until an acceptable net topology is seen.

A correlation coefficient statistics [9] is used to measure the similarity between two datasets of equal size and the results showed values between -1 and +1 on the basis of the datasets. The mathematical formulae of the correlation coefficient are described below. Let $S_{i,j}$ is the normalized similarity between two sets of attribute values X_i and X_j of datasets i and j. The formulation of $S_{i,j}$ is given as:

$$S_{i,j}=(\sum_{k=1}^{n}(X_{i,k}-\overline{X_i})*(X_{j,k}-\overline{X_j}))/[\sum_{k=1}^{n}(X_{i,k}-\overline{X_i})^2*\sum_{k=1}^{n}(X_{j,k}-\overline{X_j})^2]^{1/2} \quad (1)$$

Where

$$\overline{X_i}=1/n*(\sum_{k=1}^{n}X_{i,k}) \quad (2)$$

and

$$\overline{X_j}=1/n*(\sum_{k=1}^{n}X_{j,k}). \quad (3)$$

VI. Junction Node Selection

Each junction node of the system has a pair of a 2-way entrance and an internal path that allows only one vehicle, hence allowing four vehicles to cross based on their routes at a time. The router routes the first set of four most priority vehicles in a concurrent manner followed by the fourth vehicle and then selects the next set of four low priority vehicles from the list. Virtually this form of vehicle selection controls a set of eight vehicles at a time from the respective starting nodes as shown in TABLE I below.

TABLE 1
VEHICLES AND THEIR STARTING NODES

Vehicles	Start node
V_i AND V_j	N_1
V_k AND V_l	N_2
V_m AND V_n	N_3
V_o AND V_p	N_4

VII. Details of the Greedy Routing Procedure

The greedy routing procedure of the router includes four distinct sub procedures; *Route generation*, *Route conflict detection*, *Vehicle flow handling* and *Fault tolerance*. Details of these four sub procedures are described in the following sections.

A.. Vehicle Route Generation

The router generates routes for each vehicle randomly based on their starting nodes (staring nodes of vehicles shown in TABLE I). That is, the route of a vehicle is generated by creating random index values for the *vehicles_list* (an array that keeps the details of eight vehicles) and compares the index values with the *route_ list* (an array that keeps all the possible routes for a vehicle). As per the route generation criteria vehicles with Hamiltonian cycle route is forbidden. For example, routes of vehicle V_i/V_j can be: $N_1N_2N_4N_3$, $N_1N_3N_4N_2$, $N_1N_2N_4$, $N_1N_3N_4$, N_1N_2, or N_1N_3, where the meaning of the route $N_1N_2N_4N_3$ is, a vehicle from node N_1 travels from junction node N_1 to N_2 then N_4 and finally it exits the traffic junction through the node N_3 as shown in Table II.

TABLE II
VEHICLES AND THEIR POSSIBLE ROUTES FROM THEIR STARTING NODES

Vehicle	Possible Routes inside the system
V_i/V_j	$N_1N_2N_4N_3$, $N_1N_3N_4N_2$, $N_1N_2N_4$, $N_1N_3N_4$, N_1N_2, N_1N_3
V_k/V_l	$N_2N_1N_3N_4$, $N_2N_4N_3N_1$, $N_2N_4N_3$, $N_2N_1N_3$, N_2N_1, N_2N_4
V_m/V_n	$N_3N_4N_2N_1$, $N_3N_1N_2N_4$, $N_3N_4N_2$, $N_3N_1N_2$, N_3N_4, N_3N_1
V_o/V_p	$N_4N_2N_1N_3$, $N_4N_3N_1N_2$, $N_4N_2N_1$, $N_4N_3N_1$, N_4N_2, N_4N_3

B. Route Conflict Detection

Route conflict checking procedure dynamically checks the conflicted routes of all vehicles before they are allowed to route through the junction nodes. The route conflict detecting procedure as shown in Fig. 4, for example indicates that the routes of vehicles, V_i and V_k are $N_1N_2N_4$ and $N_2N_1N_3$ respectively; from these routes, the vehicles V_i and V_k are not allowed to start concurrent flow through the system because of their conflicted routes. That is, the router detects the conflicts of adjacent vehicles before their routing and then allows them to flow consecutively instead of concurrently. The following example shows how the router handles route conflicts of vehicles during the flow in the system. The routes of eight vehicles V_n, V_k, V_i, V_p, V_j, V_m, V_o and V_l are $N_3N_1N_2$, $N_2N_1N_3N_4$, $N_1N_2N_4$, $N_4N_2N_1$, $N_1N_2N_4$, $N_3N_4N_2N_1$, N_4N_3 and $N_2N_1N_3$, respectively as shown in Table III.

```
int conflict (int first_ route, int second_ route)
      {
if the second  digit of the first_route is equal to the
second digit of the second_route then return conflict
(1) else return no conflict (-1)

if the first digit of the first_route is equal to the
second digit of the second_route then return conflict
(1)  else return no conflict (-1)

if the second digit of the first_route is equal to the
first digit of the second_route then return conflict (1)
else return no conflict (-1)

 if the second digit of the first_route is equal to the
third digit of the second_route then return conflict
(1)  else return no conflict (-1)

if the third digit of the first_route is equal to the
second digit of the second_route then return conflict
(1)  else return no conflict (-1)

            ;            ;                 ;
 }
```

Fig. 4 Route conflict detecting procedure

TABLE III
ROUTE STATUS FROM CONFLICT CHECKING PROCEDURE

Vehicles	Route conflict?
$V_k - V_n$	Yes
$V_i - V_k$	Yes
$V_p - V_i$	Yes
$V_j - V_p$	Yes
$V_m - V_j$	Yes
$V_o - V_m$	Yes
$V_l - V_o$	No

C. Vehicle Flow Handling

The following terms are generated by the router during its routing process:

1. *Routing time* (T_r) is the amount of time that a given vehicle spends in the junction system from its starting node to the destination node of the system. It is assumed that all the vehicles in the traffic system travel at a constant speed. Hence the acceleration of each vehicle in the system is considered as zero. Therefore, the routing time, T_r of a vehicle can be calculated as:

$$T_r = (\text{Total distance traveled} / \text{Speed limit}). \quad (4)$$

For example, let $N_1N_3N_4$ is the route of vehicle, V_i and its routing time, $T_{r,i}$ can be calculated as:

$$T_{r,i} = (D_2 + D_3)/S_i, \quad (5)$$

where D_2 and D_3 are the distances between junctions N_1 and N_3, N_3 and N_4 respectively and S_i is the speed limit of the vehicle, V_i in the system.

2. *Total routing time* ($T_{r\text{-}total}$) is the sum of the routing time of all node vehicles of the traffic system at a time and is calculated as:

$$T_{r\_total} = \sum_{i=1}^{n} T_{r,i} \,. \quad (6)$$

3. *Overall junction clearing time* (T_{over}) is the over all time taken by the router to clear all vehicles from the four junction nodes of the traffic system at a given time. Assume that $T'_{r,i}$ and $T'_{r,(i+1)}$ are the routing times of vehicles i and (i+1) during their concurrent flow, the condition that is taken by the router for calculating the routing time, $T'_{r,(i+1)}$ of vehicle i is: *If there is a route conflict between vehicles i and* (i+1), *then* $T'_{r,(i+1)}$ *is* $T_{r,i} + T_{r,(i+1)}$ *otherwise* $T'_{r,(i+1)} = T_{r,(i+1)}$, where $T_{r,i}$ and $T_{r,(i+1)}$ are the routing times of vehicles i and (i+1) based on (4) and at this point $T'_{r,i}$ is equal to $T_{r,i}$.

Likewise, the router calculates the routing time of other vehicles during their concurrent flow. Therefore, the overall junction clearing time, T_{over} of the traffic system can be estimated as:

$$T_{over} = max\,(T'_{r,i}, T'_{r,(i+1)}, \ldots.., T'_{r,(i+n-1)}), \quad (7)$$

where n is the size of the vehicles in the list of the router at a time. Based on the conditions of the router, each junction node of the traffic system allows only one vehicle at a time. Hence the main job of the router is to control the four independent junction nodes (N_1, N_2, N_3 and N_4) of the traffic system in order to maintain a concurrent flow of vehicles without violating the vehicle routing conditions (routing conditions are described in the section II). That is, the router clears eight vehicles at a time from its junction nodes and comes up with a best overall junction clearing time. Fig. 5 shows the vehicle flow handling procedure for a set of n vehicles.

```
for (i=0; i < n-1; i++)
{           if there is a conflict between routes
of vehicles i and i+1 is true then
{1. Display routes and vehicles.
2. Calculate;
T'r,(i+1) = Tr,i + Tr,(i+1)  }
else
{
            1. Display vehicle routes
               and vehicles.
            2. T'r,(i+1) = Tr,(i+1) }}
             T'r,i = Tr,i
Find Tover; Tover = max (T'r,i, T'r,(i+1), …..,
T'r,(i+n)) and display Tover.
```

Fig. 5 Vehicle flow handling procedure

D. Fault Tolerance

Even though the detection and toleration of faulty vehicles is a complex process [4] and is beyond the scope of this paper, a simple fault tolerance ability of the router is described here. That is, the router is informed of presence of a faulty vehicle with its name during the vehicles routing process. Then the router quarantines the faulty vehicle without jeopardizing the flow of the entire traffic system, a random faulty vehicle is generated and it is handled by the router in the following way:

- If the faulty vehicle has no conflict with other vehicles, then quarantine it without causing a fault penalty and router estimates the overall junctions clearing time (T_{over}) as in (7).
- Incase of a conflict route, the faulty vehicle causes a fault penalty, T_f and it will be added to the overall junction clearing time of the traffic system. Details of the fault penalty calculation are described in the following section.

1. Fault Penalty Calculation

Assume a set of three vehicles V_i, V_j, and V_k with routes $N_1N_2N_4$, $N_2N_1N_3$ and N_4N_3 respectively. If V_i is failed during its traverse from nodes N_1 to N_2, then a fault penalty, $T_{f,i}$ is added to the routing time of the following vehicle. The faulty penalty of vehicle V_i can be calculated as:

$$T_{f,i} = T'_{r,i} + q, \quad (8)$$

where $T'_{r,i}$ is the routing time of vehicle V_i from N_1 to N_2 and q is the *quarantine delay* and is considered as 0.1. Similarly, if V_j is failed during its route from N_1 to N_3, then there is no fault penalty to be added to the routing time of vehicle V_k which has a non-conflict route with other vehicles.

VIII. Cost Evaluation

There is a cost evaluation applied along with the router procedure to evaluate the performance of the router. It is based on the average routing time and overall junction clearing time of a set of eight vehicles at a time. The details of the cost evaluation are described below:
The average routing time ($T_{r\text{-}avg}$) of a set of n vehicles can be calculated as:

$$T_{r\text{-}avg} = T_{r\text{-}total} / n. \quad (9)$$

From (7) and (9), the *cost value*, C of the router can be estimated from the differences between T_{over} and $T_{r\text{-}avg}$. The aim of this evaluation is to show that the router can provide a reasonable junction clearing time for a set of vehicles at a time. The cost value can be estimated as:

$$C = (T_{over} - T_{r\text{-}avg}). \quad (10)$$

If the cost value, C is zero, then the performance of the router is assumed to be *good-enough*. If C is a positive number, then the performance of the router is considered to be *reasonable*.

IX. Details of the Router Procedure

The details of the implementation of the router include the following distinct steps:
i. Generate a set of eight vehicles with there attributes.
ii. The backpropagation algorithm trains the BPNN for assigning priorities to vehicles based on their attributes and saves the priorities in their descending order into the priority queue. The convergence test measures the acceptability of the BPNN.
iii. Create two priority lists, max_list and min_list based on the priority queue:
a. max_list [] = {max(P_{Vi}, P_{Vj}), max(P_{Vk}, P_{Vl}), max(P_{Vm}, P_{Vn}), max(P_{Vo}, P_{Vp})}
b. min_list [] = {min(P_{Vi}, P_{Vj}), min(P_{Vk}, P_{Vl}), min(P_{Vm}, P_{Vn}), min(P_{Vo}, P_{Vp})}
iv. Generate junction route of each vehicle and calculate their routing time, T_r.
v. The vehicles whose priorities are in the max_list are allowed to flow concurrently from their starting nodes and sooner to the concurrent flow of vehicles from the min_list.
vi. The conflict checking procedure checks the routing conflict of vehicles during their concurrent flow and calculates their overall junction clearing time, T_{over}.
vii. If there is a faulty vehicle, then the router would quarantine it and re-route the following vehicles and the overall junction clearing time with fault penalty.
viii. Display the concurrent flow order of eight vehicles in a given time and indicates the performance of the router with the cost value.
ix. Go to step 1 for routing the next vehicle set.

A flow chart of the vehicle router is shown in Fig. 6.

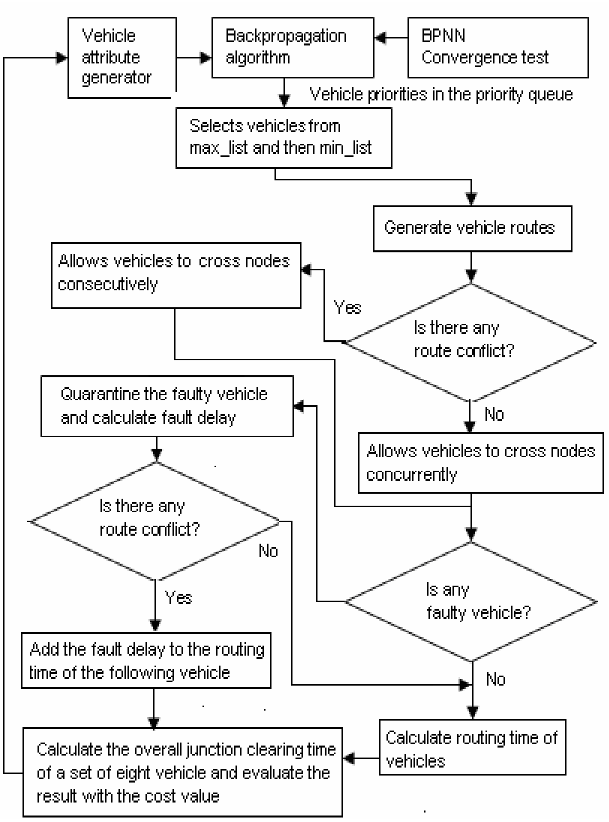

Fig. 6 Flow chart of the vehicle scheduler

X. Simulation of the Vehicle Router

The subjective vehicle router is written in C++ and supportive simulations are made to show the effectiveness of the router for the vehicle routing problem in the four-junction traffic system. For the purpose of the study, two simulations are shown: first, a routing problem with a set of eight vehicles and second, routing a problem with ten sets (each set has eight vehicles hence a total of eighty vehicles) of vehicles in the traffic system. In order to simplify the complexity of the simulation, the node distances D_i of the system is 0.1 where $i \in \{1,..,4\}$. Details of the simulations carried out are discussed below.

The priorities of eight vehicles along with their parameters and starting nodes are shown in TABLE IV. As per (1), the selected 3-layer BPNN of the router is acceptable with a similarity value of +0.998. A virtual indication of eight vehicles and their starting nodes is shown in Fig. 7. TABLE V and TABLE VI show the vehicles in *max_list* and *min_list* with their routes routing times. From these two priority lists, the priority order of vehicles can be determined as; V_2-V_8 -V_3 -V_5- V_6 -V_4 -V_1 -V_7, where V_1, V_2, V_3, V_4, V_5, V_6, V_7 and V_8 are the eight vehicles in the four junction nodes of the traffic system. TABLE VII shows the route conflict status of adjacent vehicles during their concurrent routing process. Similarly, from TABLE VII, it can be seen that vehicle V_8 is a faulted one. Hence it is quarantined and its fault penalty estimated as per (8) and is added to vehicle V_3.

TABLE IV
Vehicles and their priority order based on their attributes

Vehicle	A_i	C_i	V_i	S_i	P_i	Start node
V_1	0.1	0.2	0.8	0.2	0.1256	N_1
V_2	0.3	0.9	0.1	0.9	0.8198	N_1
V_3	0.4	0.8	0.4	0.8	0.7201	N_2
V_4	0.5	0.4	0.7	0.3	0.3224	N_2
V_5	0.9	0.6	0.2	0.8	0.4537	N_3
V_6	0.2	0.5	0.8	0.2	0.4525	N_3
V_7	0.7	0.1	0.5	0.6	0.0046	N_4
V_8	0.8	0.9	0.3	0.7	0.7898	N_4

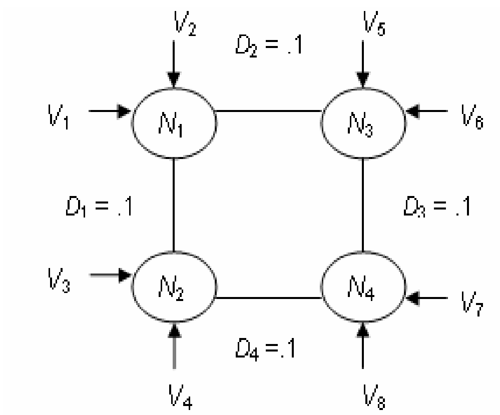

Fig. 7 Virtual indications of eight vehicles in their starting nodes

TABLE V
Vehicles in the *max_list* with their routes and route times

Vehicle	Route	Routing time (T_r)
V_2	$N_1N_2N_4$	0.2222
V_8	$N_4N_2N_1$	0.2857
V_3	N_2N_4	0.125
V_5	$N_3N_4N_2$	0.25

TABLE VI
Vehicles in the *min_list* with their routes and route times

Vehicle	Route	Routing time(T_r)
V_6	$N_3N_1N_2$	1
V_4	N_2N_4	0.333
V_1	$N_1N_3N_4$	1
V_7	$N_4N_2N_1$	0.333

The overall junction clearing time (T_{over}) taken by the router for eight vehicles is 1 and the average routing time ($T_{r\text{-}avg}$) is 0.4437. As per the cost evaluation based on (10), the performance of the router is *reasonable* with a value of +0.56. Fig. 8 and Fig. 9 show the graphical representations of eighty vehicles with their *starting time* vs. *junction clearing time* and their *starting time* vs. *average routing time* respectively. From the graphs, the best overall junction clearing time (T_{over}) taken by the router is 0.9004 and the average routing time ($T_{r\text{-}avg}$) is 0.372 for a vehicle set with a start time 20.3. Similarly, the worst overall junctions clearing time taken by the router is 3.56 ($T_{r\text{-}avg}$ is 1.168) for a vehicle set with start time 5.43. Concerning the cost evaluation, all the resulted schedules are *reasonable*. In order to simplify the complexity of the simulation, there are no faulty vehicles added in this section

TABLE VII
ROUTE CONFLICT STATUS

Vehicles	Route conflict?	Faulty Vehicle
V_2 - V_8	Yes	V_8
V_8 - V_3	Yes	V_8
V_3 - V_5	Yes	-
V_5 - V_6	No	-
V_6 - V_4	No	-
V_4 - V_1	No	-
V_1 - V_7	No	-

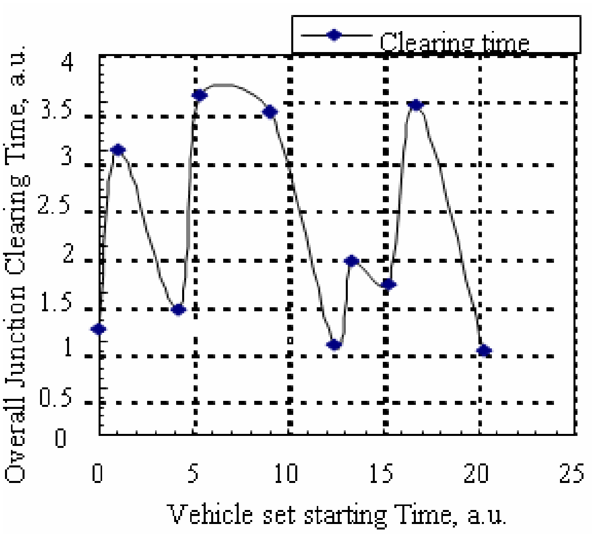

Fig. 8 A graph of eighty vehicles with their starting time vs. overall junction clearing time

XI. CONCLUSION

The subjective vehicle router has its ability in generating reasonable routing schedules by establishing proper neural network training paradigm, subjective criteria and cost evaluation. The router utilizes the customizable nature of the BPNN and the quick solution feature of the greedy algorithm. The term 'vehicle priority' of the router cannot be described formally; that is, it is not possible to define the priority of a vehicle in a normal way because that depends only on the given subjective criteria to the router. That is, the results of the router are biased towards certain objectives based on vehicle selection criteria.

The proposed router is flexible enough to adopt needs of various users and it functions like an intelligent vehicle routing agent for providing reasonable results. The configuration of the said router is most suitable in routing packets in a data transmission network and robotics applications as per the user's choice.

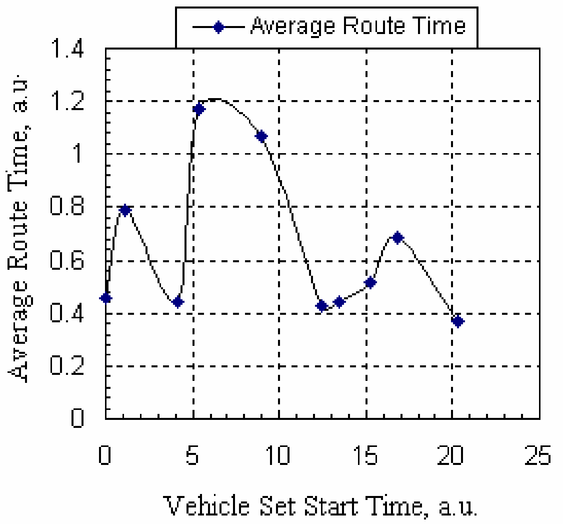

Fig. 9 A graph of eighty vehicles with their Starting time vs. Average routing time

TABLE A
A SEEN DATASET WITH FIFTY INPUTS AND THEIR RESPECTIVE OUTPUT DATA PATTERNS

A	*C*	*V*	*S*	*P*
0.1	0.1	0.7	0.1	0.07
0.1	0.9	0.3	0.9	0.82
0.1	0.3	0.7	0.1	0.21
0.1	0.3	0.1	0.9	0.3
0.1	0.1	0.9	0.3	0.12
0.1	0.1	0.5	0.9	0.07
0.1	0.5	0.1	0.9	0.48
0.1	0.9	0.7	0.1	0.81
0.1	0.7	0.1	0.9	0.67
0.1	0.3	0.9	0.3	0.24
0.1	0.7	0.3	0.7	0.67
0.3	0.9	0.1	0.9	0.81
0.3	0.1	0.7	0.1	0.05
0.3	0.1	0.3	0.5	0.53
0.3	0.7	0.1	0.9	0.67
0.3	0.3	0.7	0.5	0.19
0.3	0.7	0.5	0.7	0.65
0.3	0.9	0.7	0.5	0.79
0.3	0.9	0.1	0.7	0.81
0.3	0.7	0.1	0.7	0.63
0.5	0.1	0.7	0.1	0.041
0.5	0.3	0.5	0.3	0.18
0.5	0.1	0.3	0.9	0.039
0.5	0.7	0.3	0.9	0.57
0.5	0.3	0.9	0.3	0.21

TABLE A (Cont.)

A	*C*	*V*	*S*	*P*
0.5	0.1	0.5	0.3	0.04
0.5	0.3	0.5	0.7	0.25
0.5	0.5	0.7	0.3	0.46
0.5	0.9	0.5	0.7	0.83
0.7	0.1	0.7	0.1	0.031
0.7	0.9	0.5	0.7	0.78
0.7	0.1	0.9	0.1	0.034
0.7	0.7	0.1	0.7	0.64
0.7	0.9	0.3	0.5	0.78
0.7	0.3	0.9	0.3	0.19
0.7	0.3	0.1	0.7	0.22
0.7	0.1	0.5	0.7	0.03
0.7	0.5	0.1	0.7	0.41
0.7	0.9	0.3	0.7	0.83
0.9	0.7	0.5	0.7	0.63
0.9	0.1	0.3	0.9	0.021
0.9	0.9	0.7	0.3	0.75
0.9	0.7	0.1	0.9	0.54
0.9	0.5	0.1	0.9	0.33
0.9	0.1	0.7	0.1	0.016
0.9	0.9	0.1	0.9	0.79
0.9	0.5	0.3	0.7	0.38
0.9	0.1	0.7	0.3	0.016
0.9	0.3	0.5	0.7	0.18

REFERENCES

[1] K.G. Anilkumar and T. Tanprasert, “Neural Network Based Generalized Job-Shop Scheduler”, in *Proc. of* 2[nd] *IMT-GT Regional Conference on Mathematics, statistics and Applications*, Penang, Malaysia, 2006, pp. 53-58.

[2] K.G Anilkumar and T. Tanprasert, “Neural Network Based Greedy Job Scheduler”, in *Proc of National Computer Science and Engineering Conference (NCSEC* 2006*),* Thailand, 2006, pp. 257-262.

[3] K.G Anilkumar and T. Tanprasert, “Generalized Job-shop Scheduler Using Feed Forward Neural Network and Greedy Alignment Procedure”, in Proc*. of IASTED Conference on Artificial Intelligence and Applications*, AIA-2007, Austria, pp. 115-120.

[4] I. Okhrin and K. Richter, “The Real-Time Vehicle Routing Problem”, *Operation Research proceedings* 2007, Springer Berlin Heidelberg, 2007, pp. 141-146.

[5] D. Bertsimas, P. Chervi and M. Peterson, “Computational Approaches to Stochastic Vehicle Routing Problems”, *Journal of Transportation Science*, vol. 29, 1995, pp. 342-352.

[6] V. B. Rao and H. V. Rao, *Neural Networks & Fuzzy logic*, BPB Publications, Delhi, 1996.

[7] S. Stinson, *An Introduction to the Design and Analysis of Algorithms*, Cambridge Press, 1980, pp. 70-92.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, the MIT Press: McGraw-Hill, 2001, pp. 370-399.

[9] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, 5[th] edition, NJ: Prentice Hall, NJ, 2002, pp. 668-719.

Anilkumar Kothalil Gopalakrishnan received his Doctor of Philosophy in Computer Science from Assumption University, Bangkok, Thailand in 2008. Currently, he is a researcher (Parallel Distributed Computing Research Lab) and a lecturer in the Department of Computer Science, Assumption University. His research interests are in the fields of neural network computation, Intelligent Scheduling, Parallel Computing and Bioinformatics.