Article:

Amer, HM, Al-Kashoash, H, Hawes, M et al. (3 more authors) (2019) Centralized simulated annealing for alleviating vehicular congestion in smart cities. Technological Forecasting and Social Change, 142. pp. 235-248. ISSN 0040-1625

https://doi.org/10.1016/j.techfore.2018.09.013

Centralized Simulated Annealing for Alleviating Vehicular Congestion in Smart Cities

Hayder Amer[a,*], Hayder Al-Kashoash[b], Matthew Hawes[a], Moumena Chaqfeh[c], Andrew Kemp[d], Lyudmila Mihaylova[a]

[a] *Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, United Kingdom*
[b] *Technical Institute Qurna, Southern Technical University, Basra, Iraq*
[c] *College of Information Technology, UAE University, Al Ain 15551, United Arab Emirates*
[d] *Electronic and Electrical Engineering School, University of Leeds, Leeds LS2 9JT, United Kingdom*

Abstract

Vehicular traffic congestion is a serious problem arising in many cities around the world, due to the increasing number of vehicles utilizing roads of a limited capacity. Often the congestion has a considerable influence on the travel time, travel distance, fuel consumption and air pollution. This paper proposes a novel dynamic centralized simulated annealing based approach for finding optimal vehicle routes using a VIKOR type of cost function. Five attributes: the average travel speed of the traffic, vehicles density, roads width, road traffic signals and the roads' length are utilized by the proposed approach to

find the optimal paths. The average travel speed and vehicles density values can be obtained from the sensors deployed in smart cities and communicated to vehicles and roadside communication units via vehicular ad hoc networks. The performance of the proposed algorithm is compared with four other algorithms, over two test scenarios: Birmingham and Turin city centres. These show the proposed method improves traffic efficiency in the presence of congestion by an overall average of 24.05%, 48.88% and 36.89% in terms of travel time, fuel con-

*Corresponding author. Tel.: +44-114-222-5675
Email addresses: hmamer1@sheffield.ac.uk (Hayder Amer), hayderaam@stu.edu.iq (Hayder Al-Kashoash), m.hawes@sheffield.ac.uk (Matthew Hawes), moumena@uaeu.ac.ae (Moumena Chaqfeh), a.h.kemp@leeds.ac.uk (Andrew Kemp), l.s.mihaylova@sheffield.ac.uk (Lyudmila Mihaylova)

sumption and CO_2 emission, respectively, for a test scenario from Birmingham city in the UK. Additionally, similar performance patterns are achieved for the a test with data from Turin, Italy.

Keywords: Traffic congestion control, Simulated annealing, IoV applications, Multi-attribute decision making.

1. Introduction

1.1. Related Works

The increasing number of vehicles on city road networks has resulted in serious road traffic congestion problems which have an effect on the journey travel time, travel cost, fuel consumption and air pollution. The Victoria Transport Policy Institute has reported that the first most common cause of time delay, fuel consumption and wasted money in the United States is vehicle traffic congestion [1]. This has resulted in significant economic and productivity losses. It is predicted that this cost could rise to $121 billion in 2020 [1], [2]. Therefore, traffic congestion detection and avoidance mechanisms are two of the most pressing issues in Intelligent Transportation Systems (ITSs) [3].

The Internet of Things (IoT) [4], as a promising communication technology, enables various devices such as Wireless Sensor Nodes (WSNs), cell towers, mobile phones, Radio Frequency Identification (RFID) tags and Near Field Communication (NFC) devices to interact and cooperate with each other to attain a common goal [5]. The authors in [6] have suggested a model based on IoT in order to utilize the information and communications technologies which improve the smart city mobility. The framework includes a comprehensive system of the urban information that supports smart cities from sensor-based information collection level to data processing and management, Cloud-based integration of respective systems and services.

The authors in [7] have investigated the utilization of IoT technologies in order to approach particular city infrastructure problems and provide support to traffic operators such as the utilization of traffic sensing tools that includes wifi scanners and magnetic sensors technologies. This would enhance the collection of the traffic information and improve the entire mobility of the smart cities.

The Internet of Vehicles (IoV) is a part of the IoT, that predicts future vehicles as being connected, allowing the sharing of data to enhance traffic comfort and safety. IoV is a promising technology that can offer solutions for providing traffic control systems to monitor road conditions and travel journeys. The authors in [8] have suggested a framework based on IoV in order to support the collection of the local traffic information. This system utilizes a central station selection approach for the data penetration and the optimal traffic information transmission model in order to enhance the efficiency of the data communication in the network. Moreover, the weighted and undirected graph paradigm for IoV networks has been studied. The characteristic of GPS dataset has been analyzed and the network time-invariant has been verified.

The main part of the IoV is the Vehicular Ad hoc NETworks (VANETs) that include two main components. Firstly, Vehicle to Vehicle (V2V) communication systems that is On-Board Units (OBUs) installed in the vehicles themselves [9]. Secondly, RoadSide Units (RSUs) or Vehicle to Infrastructure (V2I) systems, consisting of magnetic and induction loop sensors, magnetometers, infrared sensors etc. The authors in [10] have discussed the operation of Vehicular Sensing Networks (VSNs) in a smart city. They have investigated the use of a trust evaluation technique in order to guarantee a reliable and safe communication among drivers in the VANETs environment. This helps to broadcast and combine reliable and valuable traffic data related to smart cities. Moreover, in [10] the significance of an accurate topological construction has been highlighted and they propose a secure selection model of the traffic information to enhance the available network capacity.

The enhancement of data communication of VANETs has been investigated in [11] where the complex network theory has utilized for the analysis and improvement of the data dissemination in VANETs. The complex characteristics of the vehicular network have been analyzed based on some communication parameters for VAENTs environment. Furthermore, the authors have proposed a clustering approach in order to station selection, a traffic allocation optimization model and an information source selection model.

1.2. Main contributions

In this work, a new centralized dynamic multi-objective optimization algorithm based on VANETs is presented. It integrates the simulated annealing (SA) algorithm with the VIKOR method [12], [13] as a cost function to formulate an approach called: centralized SA-VIKOR (CSA-VIKOR). The main goal of CSA-VIKOR is to provide the drivers with the optimal paths according to multiple criteria in order to meet the diverse navigation requirements of the drivers. This is achieved via optimization based on a multi-objective cost function. The result of which is that journeys that achieve the minimum Travel Time (TT), minimum Travel Distance (TD), minimum fuel consumption, minimum amount of emissions or a combination of the four.

In a previous work [14], an Improved Simulated Annealing TOPSIS (ISA-TOPSIS) algorithm is proposed to enhance the mobility in smart cities. ISATOPSIS is a decentralized method that can only reduce the local traffic congestion, by considering only two attributes (vehicles average speed and road length) in its optimization to provide optimal paths. This work differs in that five attributes have been utilized to form a VIKOR based cost function for use by a CSA algorithm. In addition, the advantage of the centralized algorithm structure is that information about the wider road network can be embedded into the system, unlike the ISATOPSIS method where only local traffic data is available. This means vehicles are re-routed sooner, meaning they are less likely to reach the congestion in the first place.

According to [13], a comparative analysis has revealed that VIKOR and TOPSIS methods use different normalizations and aggregating functions for ranking. The VIKOR method presents ranking preferences based on the their “closeness” to the ideal solution and introduces a compromise solution with an advantage rate. In comparison, the main idea of the TOPSIS method is that the selected preference should have the farthest Euclidean distance from the negative ideal solution and the shortest Euclidean distance from the positive ideal solution. The TOPSIS method proposes two source solutions but it does not regard the relative importance of the distances from these solutions [13]. In addition, the CSA-VIKOR utilizes data relating to the entire road network, unlike the ISATOPSIS method where only local traffic data is available.

The main contributions of this paper are:

1. A new centralized traffic congestion alleviation approach called CSA-VIKOR is proposed. This algorithm provides optimal routes that reduce the traffic congestion problem in smart cities, reducing global travel times, fuel consumptions and CO_2 emissions. CSA-VIKOR has the ability to escape from local minima or maxima and move toward a solution that is close from the global optima. This is because it allows transition to a worse solutions under certain conditions.
2. CSA-VIKOR is a centralized approach and can react to dynamic route optimization problem by aggregating real-time data from VANETs and effectively estimate alternative routes for the drivers. The advantage of the centralized algorithm structure in this case is that information about the wider road network can be embedded into the system. This differs to the work in [14], where only the local traffic information is used to optimize the route of each vehicle, reducing the potential improvement in travel time etc.
3. CSA-VIKOR can optimize vehicle paths based on multiple criteria by introducing a multi-attribute decision making (MADM) methodology called VIKOR method, allowing numerous criteria to be considered when optimizing routes for the vehicles.

The remainder of the paper is organized as follows: in Section 2 a literature review of related works is presented. in Section 3, details of the CSA-VIKOR algorithm are given. In Section 4, a performance evaluation is provided. Finally, conclusions are drawn in Section 5.

2. Related Works

The area of ITSs is considered as one of the most important applications for the VANETs [15]. Recently, researchers have shown an increased interest in developing efficient solutions for improving mobility in ITS [16]. One challenging task encountered in this area is finding the optimal navigation route within a reasonable period of time.

The vehicle path planning problem has been investigated by numerous studies using a variety of algorithms. For example, the Dijkstra and A^* algorithms have been used in [17–21]. The ant colony, genetic algorithm, hybrid heuristic and dynamic traffic assignment (DTA) methods have been used in [22–25]. Unfortunately, most of these studies did not consider the use of real time data to decrease the likelihood of congestion forming or they have no knowledge about the traffic state for use in re-routing vehicles to avoid the congestion.

In [26], a Multiple Attribute Decision Making (MADM) method called Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is presented. This mechanism assumes that both vehicles and RSUs are equipped with the VANETs to get the real-time traffic information and estimate the optimal route by using the TOPSIS method. However, the algorithm's performance is similar in some extent to the Dijkstra's algorithm [17] and it can only provide partially optimal results, which leads to the transfer of the congestion to other roads.

Route planning methods that consider real-time traffic information from VANETs have also been considered [27–30]. In [27] an Ant-based Vehicle Congestion Avoidance System (AVCAS) has been proposed. This system combines the predicted average travel speed with the segmentation of a city map to identify and alleviate the congestion. Then it uses the dynamic Dijkstra's algorithm to find different shortest paths to re-route the vehicles. In [28] and [29], two approaches have proposed to reduce the vehicular traffic congestion in smart cities. The first one is a centralized and the second one is a decentralized framework. Both solutions rely on re-routing based on the Dijkstra's algorithm or on the A^* algorithm to prevent congestion and improve traffic conditions. In [30] A Distributed Vehicular Traffic Re-Routing System for Congestion Avoidance (DIVERT) has been proposed. The K shortest path method have been utilized by the DIVERT to re-route the vehicles and avoid the congestion. However, congestions in these methods are transferred to the new routes as the algorithm only selects the shortest travel distance when re-routing vehicles.

In [31], the authors have proposed a centralized real-time path planning based on hybrid VANET where the real-time data have been collected using V2V, V2I and cellular systems. Then these data have been sent to the vehicular traffic server to be processed and evaluated. Once an accident or a congestion has happened the vehicular traffic server is responsible of finding an optimal alternative route by using the stochastic Lyapunov optimization method. However, this approach suffer from high delay time because it collects the real-time data from different communication systems and processes it at one place to find the optimal alternative path of the drivers. Additionally, the vehicular congestion in this approach is only considered when an accident has occurred.

An optimization solution has been proposed in [32] with the aim of determining a system-optimal traffic scattering that improves the traffic flow and ensures that the shortest path for the drivers is not increased past a given threshold. In this paper, a road network is provided with an Origin-Destination (OD) matrix. Additionally, the edge travel time is provided and the resulting models are linear. However, this work has not considered the real-time data and dynamic re-routing of vehicles when unpredicted traffic congestion has happened.

In recent works [33], [34], optimization models have been proposed in order to minimize the effects of vehicular emissions, i.e. the pollution routing problem (PRP). In the first work [33], the main goal is to calculate a set of paths with speeds over each edge of the paths at the same time in order to reduce the total amount of emissions, fuel consumption and environmental cost. The PRP has been formulated as a mixed-integer convex programming problem based on disjunctive convex programming and solved by using the branch-and-cut algorithm.

In the second work [34], the eco-system optimal dynamic traffic assignment (ESODTA) system has been proposed in order to determine optimal eco-routing or green paths to reduce the total vehicular emission. The ESODTA system has been formulated by using the simplified queuing model (SQM). An expanded space-time network is constructed and a bottleneck discharge rate is introduced as an additional variable. The problem is solved by a Lagrangian relaxation-based algorithm.

However, these works have not utilized any real-time data obtained from the VANETs via wireless access for vehicular environment (WAVE) protocols. Hence, when an unpredicted congestion happens due to an accident the methods from [33] and [34] can not be applied to re-route the vehicles in order to avoid the congestion. Additionally, in [33] and [34] a dynamic multi-objective optimization method is not considered and hence the method cannot find a trade off between different attributes to determine the optimal routes.

3. Framework description

V2V and V2I communication systems are the two main components in the VANETs. Both V2V and V2I communication systems can efficiently support the ITS applications by collecting the real-time traffic information such as position, speed, direction and an accident information [35]. As a result, these data can be utilized for traffic management, path planning and vehicle localization.

Here, it is assumed that both systems utilize Central Access Messages (CAMs) or beacon messages [36]. CAMs are broadcast packets sent periodically between V2V or V2I communication systems that focus on monitoring traffic and congestion alleviation [37]. Figure 1 illustrates the V2V and V2I systems. Individual RSUs monitor their assigned segment of the road network and have an overview of the average speeds and the density of vehicles on the roads. This information can then be communicated between RSUs, allowing them to have an overview of the state of the road network as a whole. As a result this allows them to select the optimal path to route vehicles to their destinations. The remainder of this

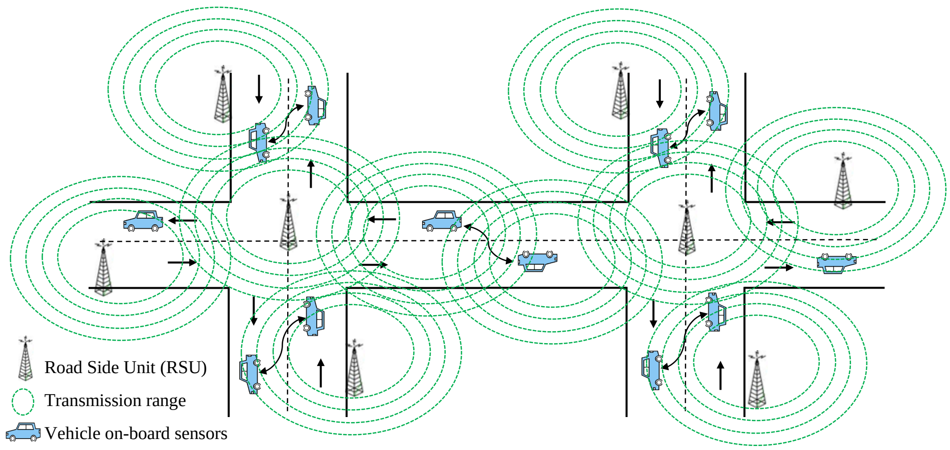


Figure 1: IoV road network infrastructure

section describes the proposed framework by specifying the real-time data collecting methodology, the road network model and the CSA-VIKOR algorithm.

3.1. Real-Time Data Collecting Methodology:

The proposed protocol relies on periodic messages for data dissemination. Each periodic message that is sent by a vehicle to the RSU monitoring its current location contains $\{roadID, averagespeed, position, denstiy, route, V_f\}$. Here, the $roadID$ gives the ID of the road where the vehicle is, $avergaespeed$ gives the vehicle's speed on the road, $position$ indicates the current location of the vehicle, $denstiy$ is the number of vehicle per unit distance on a road, $route$ represents the route along which the vehicle is travelling and V_f gives the speed limit of the road.

The RSU also sends periodic messages to other RSUs within its transmission range. Each RSU holds a road matrix A, which contains the average speeds, vehicle densities, road lengths, road widths and the number of traffic signals. The current average speeds are found from the speed measurements over the previous 5 seconds (one measurement a second). The density of the vehicles on a given segment of road is found using (5), which is given below. Reset parameters (roads length, roads width and number of traffic signals) are uploaded to the RSU during the off-line phase. Once new data is received by a RSU, the avoidance mechanism is triggered, in which congested roads are specified based on the Greenshield's traffic flow theory [38].

The speed ratio is defined to numerically represent the traffic state of a road as follows:

$$V_r = \frac{V_i}{V_f}. \tag{1}$$

Here, V_r is the speed ratio, V_i is the current average speed on the road and V_f is the speed limit of the road.

According to Greenshield's model [38], a linear relationship exists between speed and density, which has the following form:

$$V_r = 1 - D_r, \tag{2}$$

where D_r represents the density ratio and can be calculated as follows:

$$D_r = \frac{d_i}{d_j}. \tag{3}$$

In (3) d_i is the current density of vehicles on the road and d_j is the maximum jam density, which is computed as follows:

$$d_j = K\frac{L_i}{Avg_L}. \tag{4}$$

Note, K is the number of lanes on the road, L_i is the road length and Avg_L is the average vehicle length plus the minimum gap between two vehicles. This work assumes Avg_L is 6.2m, as is done in [39] and [40].

From (2) and (3) d_i can be calculated as:

$$d_i = d_j(1 - V_r) \tag{5}$$

and the following can be deduced (see Figure 2) [41]:

1. The road is under severe traffic congestion and the traffic density approaches d_j when the speed ratio V_r approaches zero.
2. The road is under free flow speed and the traffic density is low when V_r approaches 1.

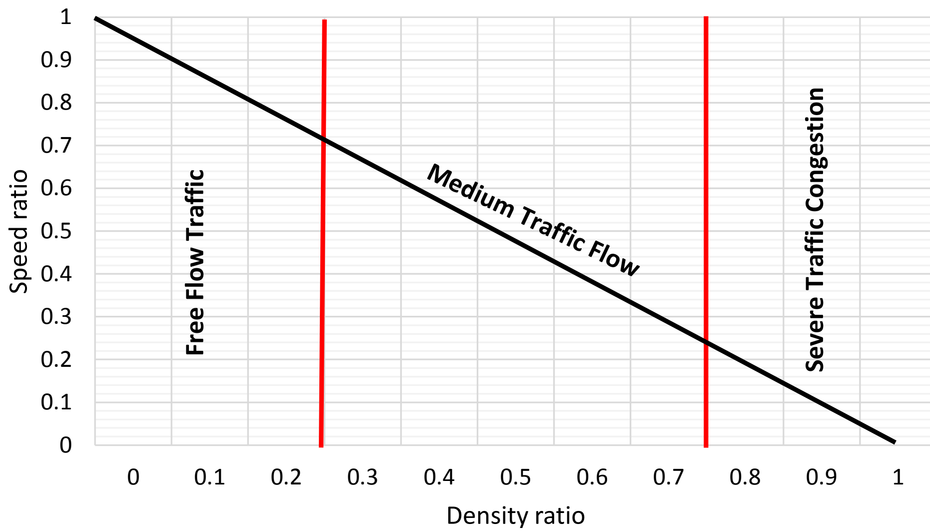


Figure 2: Traffic density-speed relationship [41]

As a result, the speed ratio can be used rather than the density ratio to numerically represent the traffic state. According to Figure 2, when the V_r is very low the traffic density will be very high and the road will be under severe traffic congestion. The proposed algorithm aims to improve traffic efficiency by detecting and avoiding congestion. Traffic congestion is detected when the speed ratio $V_r \leq 0.5$. This value has been chosen in order to avoid the severe congestion which is highlighted in Figure 2 when the speed ratio V_r become less than 0.5 that means the traffic density increased on the road segments and the congestion will be severe.

3.2. Road Network Model:

Suppose the network map consists of a set of intersections (Nodes) N and a set of edges $E = \{e_1, e_2, \ldots, e_i\}$ connected between the nodes. Then the road network can be modeled as a directed graph $G = (N, E)$. Suppose each intersection on the map contains n roads or alternatives, each of them having j routing metrics or attribute values and a weight vector which represents the importance (weight) of the attributes. The road network matrix A can be formulated as follows:

$$A = \begin{bmatrix} r_1(a_1) & r_2(a_1) & r_3(a_1) & \dots & r_m(a_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_1(a_n) & r_2(a_n) & r_3(a_n) & \dots & r_m(a_n) \\ w_1 & w_2 & w_3 & \dots & w_m \end{bmatrix} \tag{6}$$

where $r_j(a_i)$ represents the value of j^{th} routing metric (r_j attribute) for the i^{th} road (a_i alternative) for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$. In this work m=5 attributes are used, which are C_L, C_S, C_D, C_W, C_{TS}. Each of these attributes (or performance measures) are defined in the list given below. The weights $w = \{w_j \mid j = 1, 2, \dots, m\}$ represents the relative importance of the attributes being considered, which can be found as detailed in Section 3.5.

The five attributes (or performance measures) that are considered in the proposed optimization strategy are:

1. Road Length, $C_L = \{f_j(a_i)| \, i = 1, 2, \dots, n; \quad j = 1\}$, which is given as a normalized length in a directed graph G for each alternative in the matrix $\hat{A}$.
2. Average Speed, $C_S = \{f_j(a_i)| \, i = 1, 2, \dots, n; \quad j = 2\}$, which gives the normalized average velocities. The average velocities are found by aggregating the velocities of the vehicles (for the previous 5 seconds) entering the roads in $\hat{A}$ at a given time.
3. Density of vehicles on the road, $C_D = \{f_j(a_i)| \, i = 1, 2, \dots, n; \quad j = 3\}$, which is given as a normalized density in $\hat{A}$ and calculated using (5).
4. The road width, $C_W = \{f_j(a_i)| \, i = 1, 2, \dots, n; \quad j = 4\}$, which represents the number of lanes on the road in $\hat{A}$.
5. The set $C_{TS} = \{f_j(a_i)| \, i = 1, 2, \dots, n; \quad j = 5\}$, which represents the normalized number of crossing or signals on each of the roads in $\hat{A}$.

Here, as the units of each attribute are different (e.g. C_L is measured in meters, while C_S is measured in meters/second), processing all values for every routing metric into a comparability sequence is necessary. This is achieved as follows:

$$f_j(a_i) = \frac{r_j(a_i)}{\sqrt{\sum_{i=1}^{n}(r_j(a_i))^2}} \text{ where } i = 1, 2, \ldots, n; j = 1, 2, \ldots, m, \tag{7}$$

where $f_j(a_i) \in [0, 1]$ is the normalized value of j^{th} routing metric for the i^{th} road for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. The normalized road matrix $\hat{A}$ can be constructed as follow:

$$\hat{A} = \begin{bmatrix} f_1(a_1) & f_2(a_1) & f_3(a_1) & \ldots & f_m(a_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_1(a_n) & f_2(a_n) & f_3(a_n) & \ldots & f_m(a_n) \\ w_1 & w_2 & w_3 & \ldots & w_m \end{bmatrix}. \tag{8}$$

3.3. A Centralized Simulated Annealing VIKOR (CSA-VIKOR) Algorithm:

SA is a probabilistic method proposed by [42] that was originally inspired by the operation of annealing in metallurgy. The main idea of SA is to accept worse solutions under certain conditions in order to escape from local minima or maxima and move toward the global optima. The algorithm starts with an initial solution that is randomly generated. Each iteration then produces a new random solution based on the previous solution. A new solution which has a better cost (this is a measure of how suitable a solution is, see Section 3.4 for further details) is accepted. However, a solution with a worse cost than the previous solution is only accepted with a set transition probability.

This section presents a dynamic path planning method based on SA with a VIKOR cost function. The proposed approach is as follows:

- Off-line phase of route calculation: The SA algorithm starts with an off-line path computation, in which $C = \{c_1, c_2, c_3, \ldots, c_l\}$ represents a set of vehicles. Every c_l in C has a set of routes, that can take it from it's origin to its destination, that is generated from the normalized matrix $\hat{A}$. Every road a_i in the normalized road matrix $\hat{A}$ has a cost function constructed using the VIKOR method [12], [13]. Each route for vehicle c_l consists of a varying number of roads between the vehicles origin and destination. The cost function of each route is then given as sum of the costs associated with the individual road segments making up the route. The SA algorithm is outlined in Algorithm 1, where U_c means the current solution or an initial possible route. This initial path is generated randomly from the road matrix $\hat{A}$ as shown in Figure 3.

The temperature T is a random variable, which is fairly decreased with time and controlled by α which is a cooling rate factor. When the temperature parameter has a very high value, a new path U_n is calculated randomly from each intersection road matrix $\hat{A}$. The cost of the new path, $C(U_n)$, is then compared to the cost of the previous path, $C(U_c)$. When it is resolved to be a good solution by the superiority test (step 7 in Algorithm 1) the new solution or route is accepted. When the new cost is greater than or equal to the previous cost, the new route can still be accepted with a given transition probability (P_t in Algorithm 1). This expands the search space and helps to avoid local optimal solutions. When T approaches T_m, the path with the minimum cost value has a high probability of being accepted.

The path U_n in step number 7 of Algorithm 1 is constructed based on previous path U_c as follows:

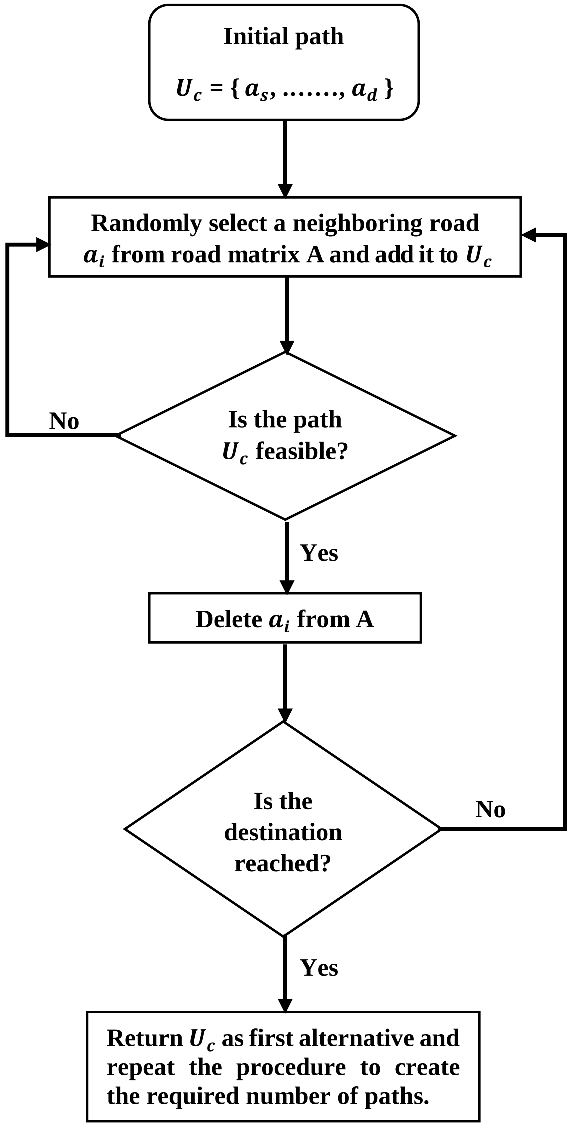


Figure 3: The procedure for generating a random initial path [14]

Algorithm 1 Simulated Annealing Algorithm

```
U_c = U_{c_0} current solution
T = T_0 an initial temperature
α = cooling rate
T_m is the minimum temperature
B_s = the best solution
B_s ← U_c
While T > T_m:
    Generate a random neighboring solution U_n from Â
        If C(U_n) < C(U_c) then
            Move to U_n
            Accept change B_s ← U_n
        Else If C(U_n) ≥ C(U_c) then
            Calculate ΔE = C(U_c) − C(U_n)
            Move to U_n with transition probability
            P_t = 1/(1 + exp(−ΔE/T))
            End
        T = αT
    End
Return B_s
```

1. Consider a current route $U_c = \{a_s, a_1, \ldots, a_i, a_{i+1}, \ldots, a_{l-1}, a_l, a_d\}$ where a_s is the road segment the vehicle starts on, a_d is the destination segment and a_i means the i-th road segment.
2. The procedure for generating a random neighboring solution or randomly perturbed route (see Figure 4) consists of the following three steps.
    (a) Two roads a_i and a_l are chosen randomly in the route U_c as a source and destination, respectively.
    (b) Generate a random route between a_i and a_l.
    (c) The route $U_c = \{a_s, a_1, \ldots, a_i, a_{i+1}, \ldots, a_{l-1}, a_l,$

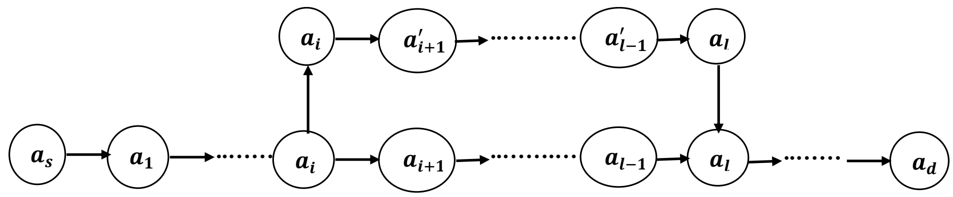


Figure 4: How to construct a new path U_n based on an initial path U_c [14]

$a_d\}$ is substitute by $(a_i,\ a'_{i+1},\ \ldots,\ a'_{l-1},\ a_l)$ to generate a new route $U_n = \{a_s, a_1, \ldots, a_i, a'_{i+1}, \ldots, a'_{l-1}, a_l, a_d\}$.

3. Test the continuity of the new route.
4. Repeat steps 2-3 if the route is not feasible. Otherwise, continue with SA as in Algorithm 1 and compare the cost of the new route to the previous route.

- On-line phase of route calculation: In this phase, the vehicles start driving through the city with the route created by the off-line path calculation. Once the congestion is detected, the on-line phase is triggered automatically to determine an alternative route as follows: Once new data becomes available, the RSU updates the road costs $RC_k = \{roadId, averagespeed, position, density, V_f\}$. Based on this data the road matrix $\hat{A}$ for the RSU is updated. Then the RSU will identify congested roads in $\hat{A}$ and generates a set of congested roads contained in the matrix CR. Hereafter, once vehicle v_k approaches an intersection it will send a query message that contains $msgQ_k = \{roadId, position, route\}$ to the RSU. This RSU will then evaluate the route for vehicle c_l. If the evaluation shows that the route will bypass a congested road, then c_l will keep traveling using the current path. Otherwise, the CSA-VIKOR at the RSU will be activated and reloaded with an updated search space. The alternative route will then be computed and transmitted to the vehicle. This allows the vehicle to then continue towards its destination, avoiding the congestion that has formed. Algorithm 2 shows the procedure of the congestion avoidance us-

Algorithm 2 Congestion avoidance using CSA-VIKOR approach

1: ***Input***:

C set of vehicles in the network, G Network graph, Beacon messages between RSUs and vehicles, A road matrix with different attributes

2: ***Output***:

A new optimal route that avoids the congested roads

3: ***At each RSU***:

For road $\in G$ do

Calculate the average vehicle speeds

Calculate the average vehicle density

Update matrix A (roadID, averagespeed, density)

Calculate normalized road matrix $\hat{A}$

End

4: ***Detect congested roads***:

For road $\in \hat{A}$ do

Calculate the velocity ratio V_r

If $V_r \leq 0.5$ then

Add the congested road into matrix CR

End

End

5: ***Find new route***:

For $c_l \in C$ do

Extract the vehicle route from beacon messages

For road $\in$ vehicle route do

If road $\in$ Congested Road Matrix then

Exclude the congested road from matrix $\hat{A}$

Get the current coordinate of the vehicles

Update road costs RC_K

Update matrix $\hat{A}$ for each RSU by using VIKOR method

Calculate new optimal route by using CSA-VIKOR

Send the new route to the vehicles;

End

End

End

ing CSA-VIKOR. This procedure is repeated every time vehicles approach an intersection and enter the transmission range of a RSU.

3.4. VIKOR Cost Function of CSA-VIKOR Algorithm:

In this paper the cost function is implemented using the VIKOR method [12]. VIKOR is an approach for solving multi-criteria optimization problems. It works by ranking solutions and picking the best alternative, by comparing each to the ideal (optimal solution based on conflicting criteria) and worst solution. This ranking is based on Lp-metric given in (9) [13]. This metric is termed the level of regret and is given by:

$$L_{p,i} = \left\{ \sum_{j=1}^{m} \left[w_j \frac{\left(f_j^* - f_j(a_i)\right)}{\left(f_j^* - f_j^-\right)} \right]^p \right\}^{\frac{1}{p}}, \quad 1 \leq p \leq \infty \quad , \tag{9}$$

where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. Here, $L_{1,i}$ is defined as the maximum group utility (S_i), and $L_{\infty,i}$ is defined as the minimum individual regret of the opponent (R_i), f_j^* and f_j^- represent the best and the worst values of all attributes, respectively.

The VIKOR method then ranks the alternative solutions as follows:

1. Find that best f_j^* and the worst f_j^- values, for each criteria being considered in the optimization.
2. Calculate S_i and R_i using:

$$S_i = L_{1,i} = \sum_{j=1}^{m} w_j \frac{\left(f_j^* - f_j(a_i)\right)}{\left(f_j^* - f_j^-\right)}, \tag{10}$$

$$R_i = L_{\infty,i} = \max_j \left[w_j \frac{\mid f_j^* - f_j(a_i) \mid}{\mid f_j^* - f_j^- \mid} \right], \tag{11}$$

where w_j is the weight of the j^th criterion, expressing the relative importance of criteria.

3. Find $Q_i, i = 1, 2, \ldots, n$, using:

$$Q_i = \alpha \frac{\left(S_i - S^*\right)}{\left(S^- - S^*\right)} + \left(1 - \alpha\right) \frac{\left(R_i - R^*\right)}{\left(R^- - R^*\right)}. \tag{12}$$

Here $S^* = \min_i S_i$, $S^- = \max_i S_i$, $R^* = \min_i R_i$, $R^- = \max_i R_i$, and α is a weighting term. A value of $\alpha = 0.5$ is selected to ensure that the compromise solution is stable within a decision making process.

4. The alternative solution can be ranked in descending order based on the values for S, R and Q. The interested reader can find further details about the VIKOR standard ranking procedure in [12] and [13].

However, in this paper the ranking procedure of VIKOR method has been avoided by using the SA optimization algorithm. The cost function of CSA-VIKOR method has been calculated by adding Q, S and R of each potential route as in (13) below. The best route is then chosen using the SA procedure in Algorithm 1. Here, μ_i, φ_i and β_i are the weight parameters of functions Q_i, S_i and R_i, respectively. They are chosen so that $\mu_i + \varphi_i + \beta_i = 1$ and μ_i, φ_i, $\beta_i > 0$, $\forall i \in n$. The values of μ_i, φ_i and β_i are selected by trial and error to satisfy the system objective and requirement given by:

$$Y_i^* = \mu_i Q_i + \varphi_i S_i + \beta_i R_i \quad Y_i^* \in [0,1] \quad \forall i = 1, \ldots, n. \tag{13}$$

3.5. Calculation of Attribute Importance Weights:

The selection of the weights that control the relative importance of each attribute under consideration has an impact on the final solution that is reached. Various methods for making this selection have been previously suggested [43]. In this paper, the Maximizing Deviation (MD) method is used to determine the weight vector $w = [w_1, w_2, \ldots, w_m]$ and to ensure the weight stability intervals of the multiple attributes.

First, the initial values of the weight vector are specified $w^* = [w_1^*, w_2^*, \ldots, w_m^*]$. According to [44], the weight vector w^* should be selected to maximize all deviation values (the differences between values of the same attribute) for all the attributes. The weight vector can then be formulated using a non-linear programming model:

$$\begin{aligned} \max_{w_j^*} F(w^*) &= \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{z=1}^{n} w_j^* d(f_j(a_i), f_j(a_z)), \\ \text{subject to} &\sum_{j=1}^{m} w_j^{*2} = 1, \quad 0 \le w_j^* \le 1, \end{aligned} \tag{14}$$

where the distance $d(f_j(a_i), f_j(a_z))$ between different values belonging to the same attribute is given by:

$$d(f_j(a_i), f_j(a_z)) = (f_j(a_i) - f_j(a_z))^2. \tag{15}$$

To solve (14), let

$$L(w_j^*, \lambda) = \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{z=1}^{n} w_j^* d(f_j(a_i), f_j(a_z)) + \frac{1}{2}\lambda\left(\sum_{j=1}^{m} w_j^{*2} - 1\right) \tag{16}$$

represent the Lagrangian function of the constrained optimization model, where λ is a Lagrange multiplier. The partial derivatives of L are estimated as

$$\frac{\partial L(w_j^*)}{\partial w_j^*} = \sum_{i=1}^{n}\sum_{z=1}^{n} d(f_j(a_i), f_j(a_z)) + \lambda w_j^* = 0, \quad \text{for} \quad 1 \le j \le m \tag{17}$$

and

$$\frac{\partial L(w_j^*)}{\partial \lambda} = \frac{1}{2}\left(\sum_{j=1}^{m} w_j^{*2} - 1\right) = 0. \tag{18}$$

From (17), it can be derived that

$$w_j^* = -\frac{\sum_{i=1}^{n}\sum_{z=1}^{n} d(f_j(a_i), f_j(a_z))}{\lambda} \quad \text{for} \quad 1 \le j \le m. \tag{19}$$

After substituting (19) into (18), it gives

$$\lambda = -\sqrt{\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\sum_{z=1}^{n} d(f_j(a_i), f_j(a_z)\right)^2}. \tag{20}$$

Replacing (20) into (19), it gives

$$w_j^* = \frac{\sum_{i=1}^{n}\sum_{z=1}^{n} d(f_j(a_i), f_j(a_z))}{\sqrt{\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\sum_{z=1}^{n} d(f_j(a_i), f_j(a_z)\right)^2}} \text{ for } 1 \le j \le m. \tag{21}$$

The final w_j is derived from the normalization of w_j^* as

$$w_j = \frac{w_j^*}{\sum_{j=1}^{m} w_j^*} = \frac{\sum_{i=1}^{n}\sum_{z=1}^{n} d(f_j(a_i), f_j(a_z))}{\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{z=1}^{n} d(f_j(a_i), f_j(a_z))} \text{ for } 1 \leq j \leq m. \tag{22}$$

4. Performance Evaluation

The proposed framework has been tested and evaluated through a vehicular network simulator Veins [45] which integrates the traffic simulator Simulator for Urban Mobility (SUMO) [46] with the network simulator OMNeT++ [47] to manage the mobility of vehicles and the communication between V2V or V2I communication systems. Two realistic maps have been imported from the Open Streets Map (OSM) tool [48] to evaluate and test the proposed method (the scenarios of Birmingham city in U.K. and Turin city in Italy) as shown in Figure 5 and 13.

The proposed algorithm has been implemented for different vehicular environments to optimize the traffic scenario. The CSA-VIKOR has been compared with the Original Dijkstra's Algorithm (ODA), Dynamic Dijkstra's Algorithm (D-DA), Dynamic A^* (D-A^*) algorithm, which were implemented as in [19] and the ISA-TOPSIS method which was implemented as in [14]. The simulation of all algorithms has been executed for ten independent Monte Carlo runs and then the average values of the obtained results were recorded. Five different performance measures have been considered in this performance evaluation:

- **Average Travel Time (ATT)**: average travel time of all vehicles.
- **Average Fuel Consumption (AFC)**: average fuel consumption taken by vehicles
- **Average CO_2 emission**: average CO_2 emission of all vehicles.
- **Average Travel Distance (ATD)**: average travel distance taken by vehicles.

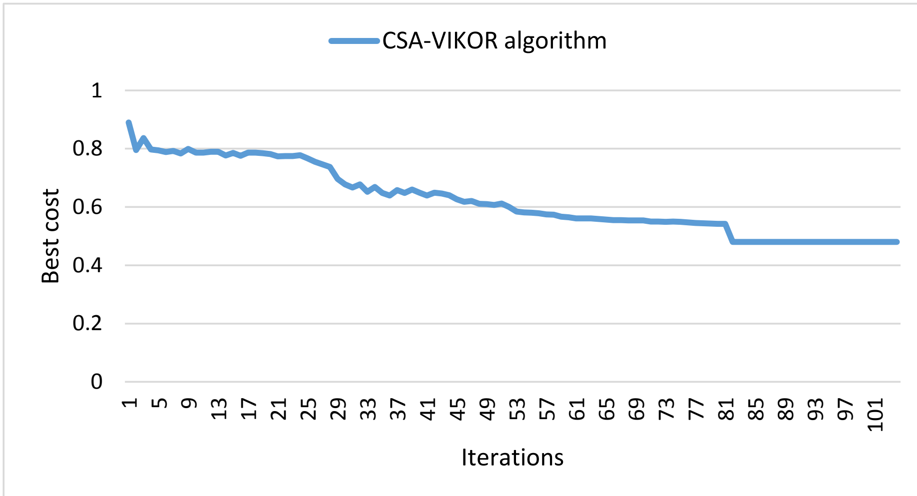


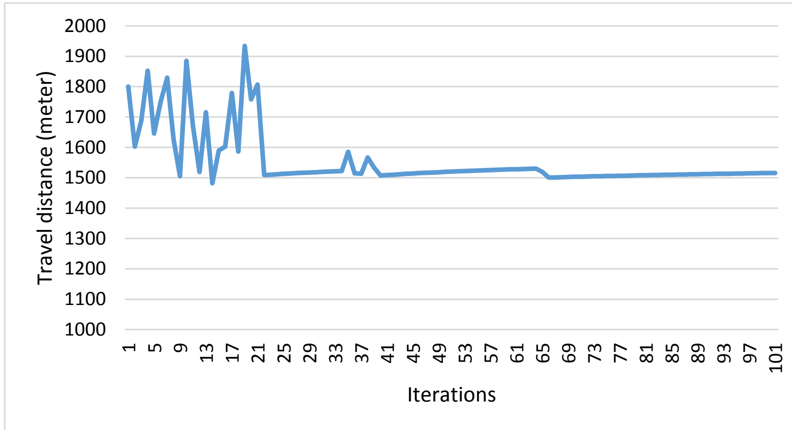
Figure 5: CSA-VIKOR convergence

- **Average Travel Speed (ATS)**: average travel speed of all vehicles.

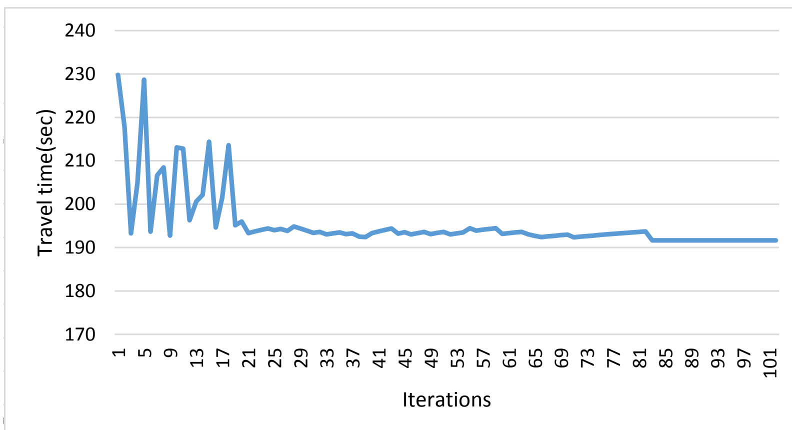
The EMIssions from Traffic (EMIT) model [49], is used to calculate fuel consumption and vehicle emissions. This is done using a statistical model and vehicle speeds and accelerations from the SUMO simulator.

Figure 5 shows the convergence of the CSA-VIKOR approach to the optimal cost. It is clear that the CSA-VIKOR continues to search for the solution until reaching convergence, which in this example is after 81 time steps. This algorithm shows a good search ability and finds the solution lading to the optimal route.

Figures 6 a and b show the convergence of the algorithm in terms of the travel distance and travel time, respectively. It is obvious that the CSA-VIKOR finds the travel distance and travel time at the same speed from 0 to 81 times. Therefore, we can deduce from Figures 6 a and b that the search speed of the algorithm is good enough to find the solution.



(a)



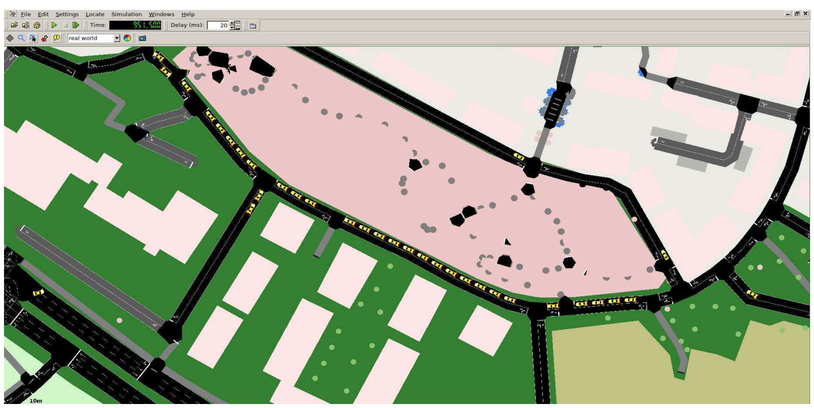
(b)

Figure 6: Convergence curves of CSA-VIKOR. (**a**) Travel distance convergence line; (**b**) Travel time convergence line.

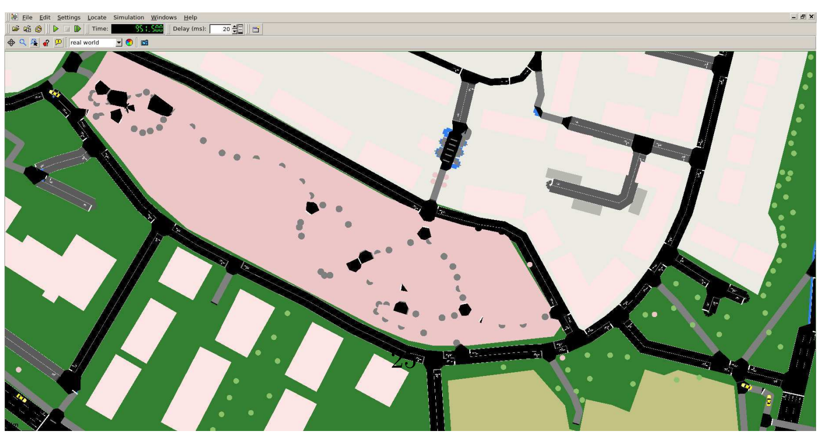
4.1. Birmingham City Scenario

In this Scenario, a realistic map scenario for Birmingham city centre has been imported from OSM tool into SUMO simulator. The area under consideration is shown in Figure 7. Figure 8a then shows a selected region where congestion forms if no dynamic re-routing is used when congestion is detected. When the proposed method is used this congestion is then avoided as shown in Figure 8b.

Figure 7: Birmingham city centre Map that is imported into SUMO



(a)



(b)

Figure 8: The traffic flow of vehicles with ODA and CSA-VIKOR. (**a**) The traffic flow with ODA appraoch; (**b**) Traffic flow using CSA-VIKOR approach.

Table 1: Simulation parameters as configured in the SUMO implementation of Birmingham scenario

Simulation parameters	**Value**
Map dimension	3.5 km$\times$2.5 km
Maximum allowed speed	32 m/s
Simulation time	1000 s
MAC/PHY	IEEE 802.11p
Max. transmission range	600 m
Number of vehicles	100-1000
T **off-line**	100 °C
α **off-line**	0.998
T **on-line**	35 °C
α **on-line**	0.992
μ_i	0.55
φ_i	0.225
β_i	0.225
Number of simulation runs	10 times
Confidence Level	95%

Table 1 shows the parameters that have been used in the simulation, where the vehicles speed have been chosen by the designer using U.K. road laws as a guide. The SA parameters (T and α) have been selected to give a suitable trade-off between computation time and amount of optimization. The other parameters have been chosen based on the OSM and Veins specification.

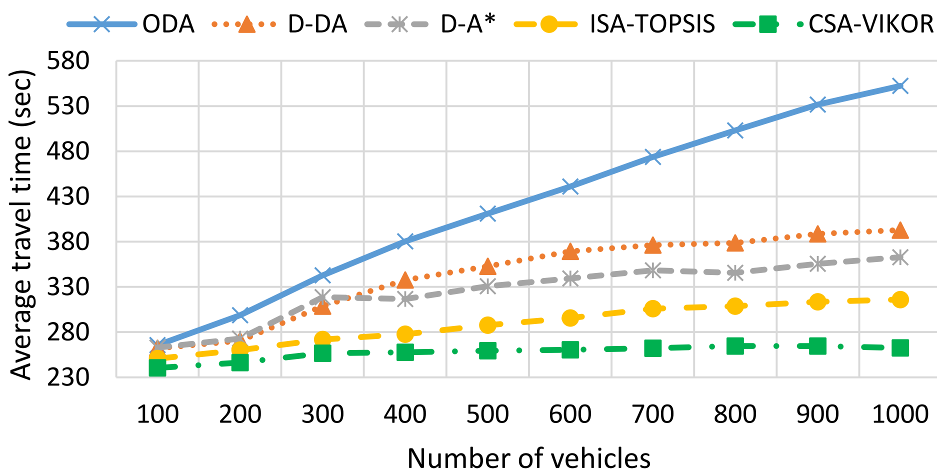


Figure 9: Average travel time

4.1.1. Average Travel Time

Figure 9 depicts the average travel time obtained by the five methods. It is clear that the average travel time has a direct relationship with the number of vehicles. As is expected, the average travel time increases as the number of vehicles increases. This is because the more vehicles there are on the roads, the greater the probability of traffic jams forming (as well as them being longer due to the increased number of vehicles), resulting in larger average travel times. This is illustrated by the results shown for ODA in Figure 9.

The D-DA and D-A^* algorithm obtain travel times that are more constant and lower than those of the ODA. This is due to the re-routing mechanism when the congestion is detected. The ATT of D-DA is slightly better than the D-A^* algorithm for low numbers of vehicles. However, when the number of vehicles starts increasing, the ATT of the D-A^* algorithm becomes much better compared with the D-DA. This is due to the fact that the D-DA selects the shortest route for all vehicles and due to the low number of vehicles in the sparse vehicle example, it does not move the congestion to another area. In contrast, in a dense vehicle scenario re-routing, many vehicles to the route with the shortest path will move the congestion from one area to another one.

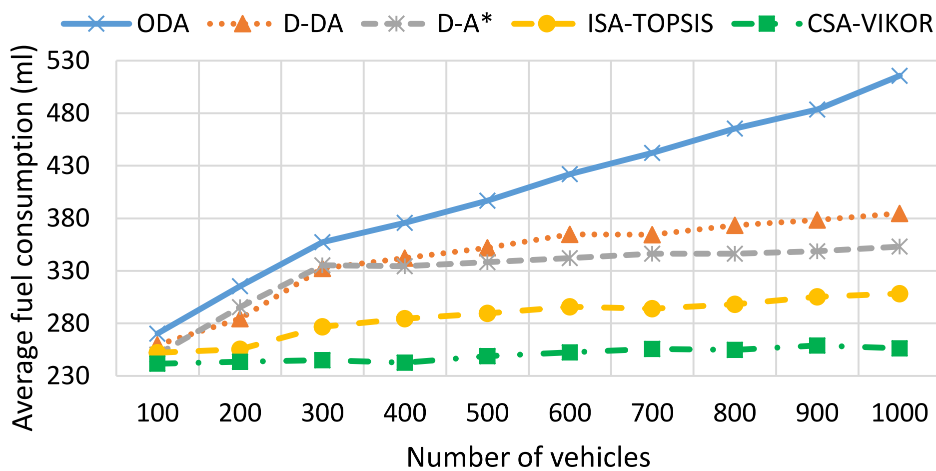


Figure 10: Average fuel consumption

It is clear that the CSA-VIKOR algorithm has significantly improved the average travel time as compared to the ODA, D-DA, D-A^* and ISA-TOPSIS, respectively. The reason is that the CSA-VIKOR algorithm routes the vehicles through the less congested path by using the MADM VIKOR method to consider more attributes (density of cars on the road, the traffic light at the roads and width of the roads) in its cost function. Additionally, the centralized approach CSA-VIKOR helps the RSUs to update most of the roads in the road matrix, giving an overview of almost all the roads on the map. The ISA-TOPSIS is seen to have better ATT as compared with the ODA, D-DA nad D-A^* algorithms. This is because the ISA-TOPSIS utilizes the real-time information in it is cost function. However, its performance is still relatively lower than the CSA-VIKOR algorithm. The reason is that the ISA-TOPSIS only has a local overview of roads on the map.

4.1.2. Average Fuel Consumption

Figure 10 illustrates the AFC obtained from all algorithms. It can be seen that the impact of taking the longest and shortest route on the traffic efficiency and the fuel consumption. The D-A^* algorithm selects the longest free flow route that helps to distribute the vehicle and decrease the congestion. However,

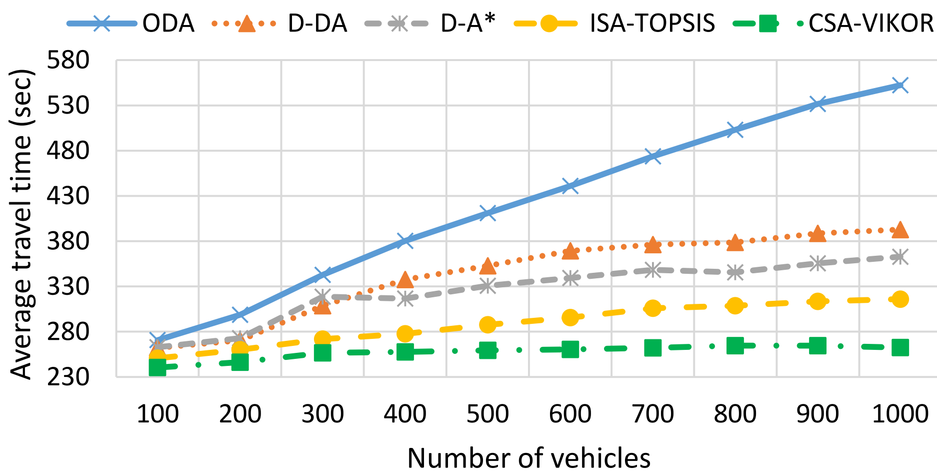


Figure 11: Average CO_2 emission

that leads to driving for long distances which in turn leads to the consumption of more fuel by the engine. On the other hand, in dense scenarios choosing the shortest path by ODA and D-DA leads to the generation of severe traffic congestion which in turn leads to the consumption of more fuel due to the higher numbers of vehicles waiting in the queues.

Despite the D-DA, D-A^* and ISA-TOPSIS algorithms having lower fuel consumption than ODA, their traffic efficiency is lower compared with the efficiency of the CSA-VIKOR algorithms. This is due to the shorter waiting time, better average speeds and an optimal path that is selected based on different navigation criteria by this algorithm. Additionally, both the ISA-TOPSIS and CSA-VIKOR pay attention to the congestion by using the real-time data from VANETs that helps to re-route the vehicles by selecting the optimal paths and avoiding the traffic jams. However, CSA-VIKOR shows better performance as compared to the ISA-TOPSIS. This is due to the fact that the ISA-TOPSIS can only avoid the local traffic congestion because it only updates a few roads in the vehicle transmission range.

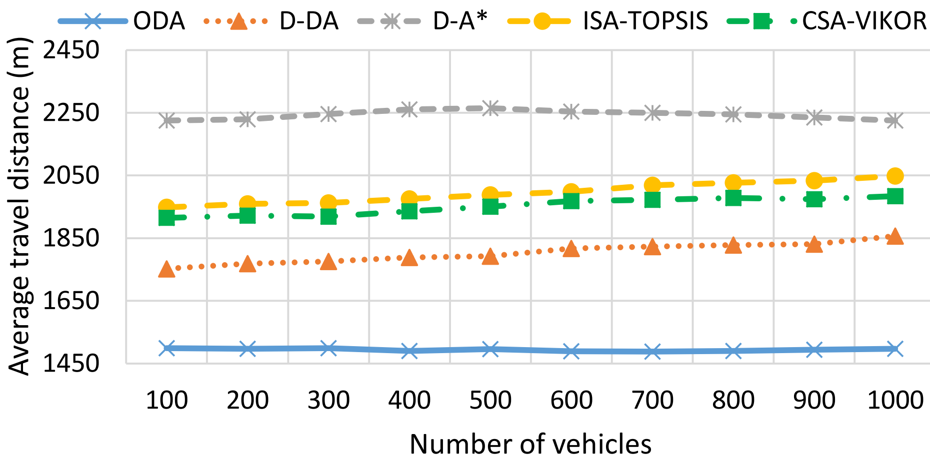


Figure 12: Average travel distance

4.1.3. Average CO_2 Emissions

Figure 11 shows the CO_2 emissions result obtained by the five algorithms. The results of CO_2 emissions are directly related to the results of fuel consumption. The longer the travel distance, the larger the waiting time, the more fuel that is consumed by the engine, the more CO_2 emissions there are. For large vehicle numbers and when there is congestion, there are longer waiting times on the roads, increasing the fuel consumption and CO_2 emissions.

It is clear from Figure 11 that CSA-VIKOR has the lowest average CO_2 emissions compared with the other algorithms. This is due to it gives the best average travel speeds by finding the optimal paths. The ISA-TOPSIS comes in the second place in terms of CO_2 emissions. Both D-DA and ODA have the worst CO_2 emissions due to a large amount of fuel being consumed by the vehicle on the routes being optimized by these methods.

4.1.4. Average Travel Distance

Figure 12 depicts the average travel length result for all of the algorithms. The D-A^* has a larger average travel distance than ODA, D-DA, ISA-TOPSIS and CSA-VIKOR. This is due to the fact that D-A^* chooses the routes with the longest travel distance (with the maximum average speed limit) but it has the

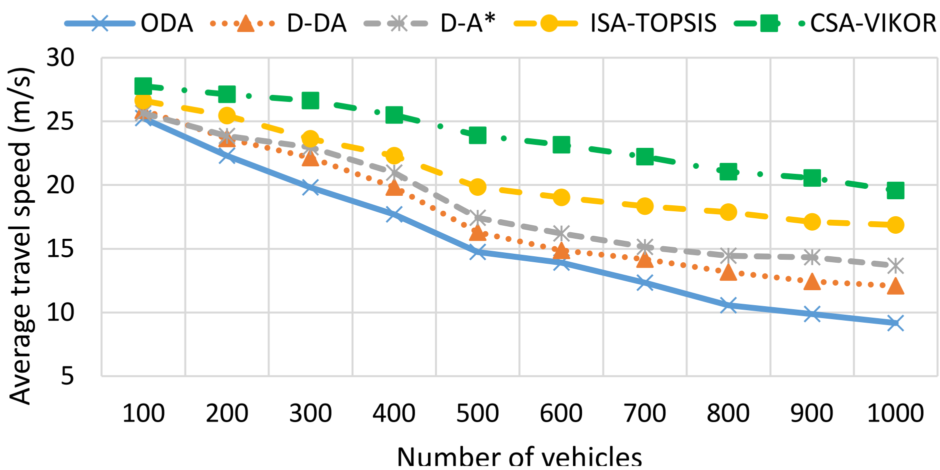


Figure 13: Average travel speed.

minimum travel time and distributes the vehicles on them to avoid generating further congestion.

It can also be seen that both ISA-TOPSIS and CSA-VIKOR can find a compromise by minimizing effectively the travel time, fuel consumption and CO_2 emissions due to their ability to consider multiple pieces of traffic information. However, this reduction leads to a slight increase in the travel distance compared to ODA and D-DA. This is a knock on effect of ISA-TOPSIS and CSA-VIKOR using real-time traffic information to re-route vehicles to avoid congestion that forms on the shorter routes.

The D-DA has slightly increased the travel distance as compared to ODA. This is because the vehicles are already on the shortest paths (as are the ones routed using ODA) and are then given an alternative route when congestion is detected. However, this has not aided the drivers in avoiding the congestion because it re-routes the vehicles with the shortest path and that leads to transfer the congestion from one area to another. The ODA has a constant travel distance as compared to the other algorithms, as there is no re-routing in this case. Note, this distance is fixed and is not affected by whether congestion occurs or not.

4.1.5. Average Travel Speed

Figure 13 speeds the average travel speed obtained by all of the algorithms. CSA-VIKOR has recorded the best average travel speed compared to the other methods for all vehicle numbers considered. This is due to the congestion avoidance method and providing the vehicles with the best alternative paths. Additionally, it is a centralized approach that gives drivers a comprehensive overview for almost all of roads on the map.

The CSA-VIKOR has recorded the best performance in terms of average travel speeds. This is because this optimization method can avoid the traffic congestion by using the traffic information for the wider region and considering five attributes in its optimization that it is not possible for the ODA, D-DA and D-A^* algorithms. The ISA-TOPSIS has better performance in terms of ATS as compared with the ODA, D-DA and D-A* algorithms. However, this method can only improve the local traffic congestion by utilizing the local traffic information.

Despite the D-DA and D-A^* having better performance compared to ODA, they have a relatively poor performance when compared to the proposed solution. This is because they avoid the congested roads by re-routing all vehicles along an alternative route based on a single attribute. Therefore, the congestion is transferred to the new routes. The ODA has the worst average travel speed. This due to a large number of vehicles being stuck in traffic congestion, due to there not being a re-routing mechanism. We can see the impact of travel speed on the traffic efficiency especially the fuel consumption and CO_2 as in Figures 10 and 11, respectively.

4.1.6. Average CPU Time

Figure 14 illustrates the average CPU time that has been estimated by combining the transmission time of real-time data with the elapsed time to find a new path and re-route vehicles to avoid congested roads. It is clear from Figure 14 that the ODA, D-DA and D-A^* require less CPU time as compared to ISA-TOPSIS and CSA-VIOKR algorithms. However, that has not really

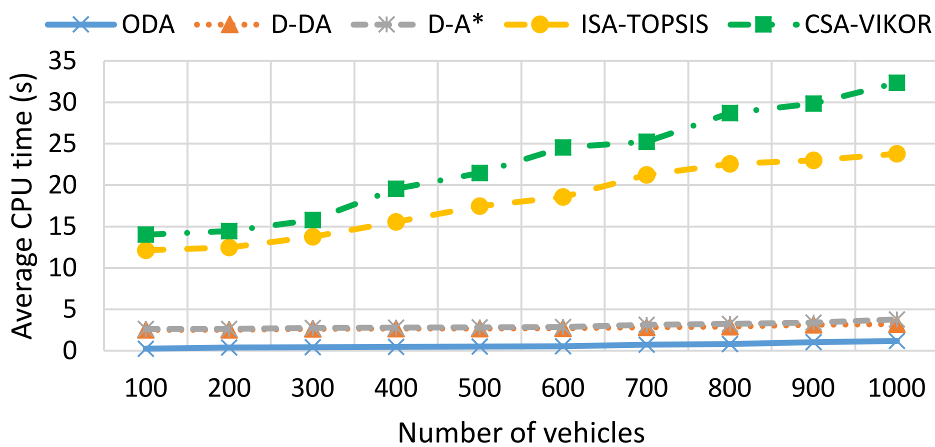


Figure 14: Average CPU time

helped the driver to avoid the congestion because they re-route drivers with the shortest path or longest free flow path to escape from the congestion and that leads to moving the congestion from one street to another one. ISA-TOPSIS has better CPU time as compared to CSA-VIKOR. This is due to the search of an alternative route has been done independently in each vehicle rather than waiting for all traffic states to be communicated by the RSUs. However, this only helps the driver to escape from the local traffic congestion. CSA-VIKOR has the largest CPU time because the optimization has been done centrally in each RSU to find the optimal path for all drivers. This helps to decrease the global travel time, fuel consumption as well as CO_2 emissions of all drivers.

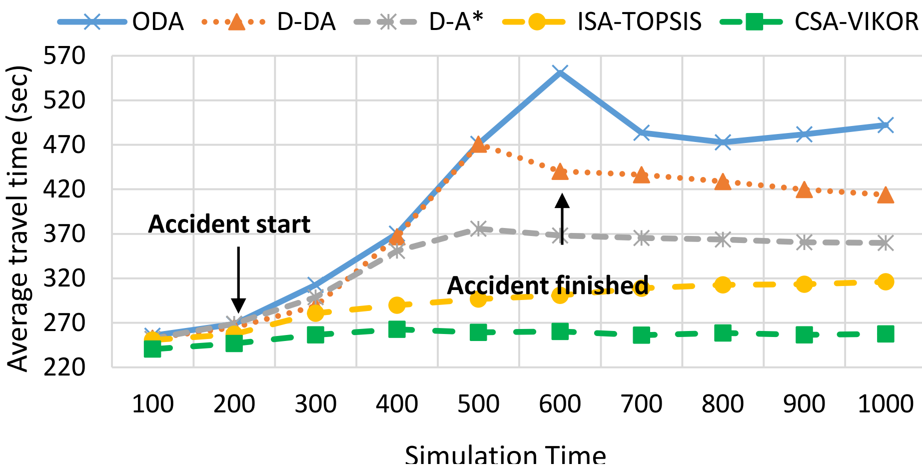


Figure 15: Average travel time of the accident scenario

4.1.7. Average Travel Time of Accident Scenario

Figure 15 represents an accident scenario, which is similar to a scenario considered in [27] in order to evaluate the performance of all of the algorithms in the presence of accidents. In this scenario, 1000 vehicles were involved and 500 vehicles are assigned to the five algorithms, with each algorithm being used to select the route of 100 vehicles. The remaining 500 vehicles are guided randomly to generate accidents and congestion on the some of the main roads in the map of Birmingham being considered. The accident starts after 200 seconds of simulation time and it is cleared after 600 seconds.

As can be seen from Figure 15, all vehicles that are routed through the shortest paths via ODA have a worse travel time as compared to the other algorithms. This is due to the lack of utilizing of real-time data and the lack of a mechanism to avoid the congested roads. The D-DA gives the second worst travel times. This is due to the dynamic re-routing of all of the vehicles to the shortest paths, which leads to the transfer of congestion to other roads. The D-A^* algorithm has a better reaction to the congestion as compared to ODA and D-DA because it routes the vehicles along the roads with the minimum travel times. However, its' performance is still less efficient than compared to ISA-

Figure 16: Turin city centre Map that is imported into SUMO

TOPSIS and CSA-VIKOR. This is due to the efficient reaction and utilization of real-time traffic information and congested roads by ISA-TOPSIS and CSA-VIKOR.

The CSA-VIKOR algorithm has the best performance in the presence of congestion as it utilizes the real-time data of almost all of the roads on the map while the ISA-TOPSIS can only utilize the local traffic information. This sometimes does not give the driver enough time to avoid the congested area. In addition, the CSA-VIKOR algorithm selected the optimal paths based on two real-time navigation criteria (average speeds and vehicle densities) and three static criteria (roads length, roads width and a number of traffic signals on the roads). This helps the driver to select the path with the fewer number of vehicles, higher average speeds, larger number of lanes and fewer traffic signals.

4.2. Turin City Scenario

Another realistic map scenario for Turin city centre in Italy has been imported from the OSM tool as shown in Figure 16.

In this Scenario, The parameters for the CSA-VIKOR algorithm are the same as the Birmingham city scenario and are summarized in Table 1 except

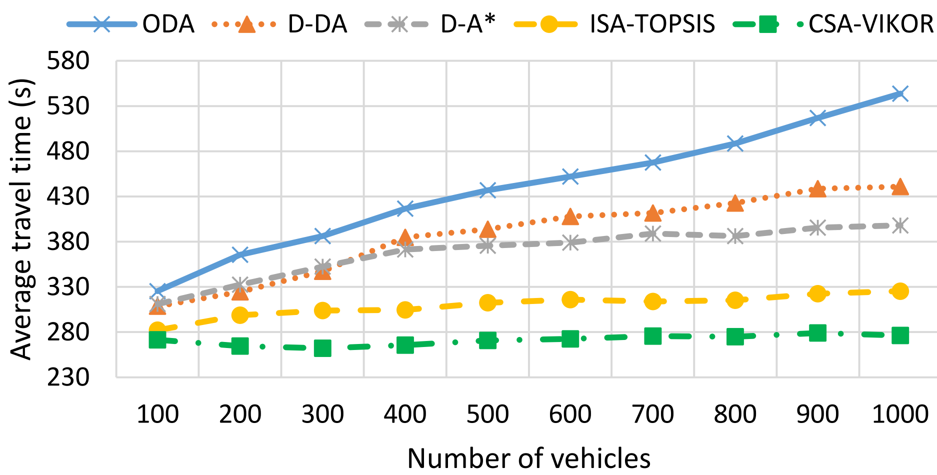


Figure 17: Average travel time

the map size that is 3.8 km×3.2 km and the maximum allowed speed that has been set up to 30 m/s.

Figures 17, 18 and 19 show the obtained results of average travel time, average fuel consumption and average travel speed of all of the algorithms being evaluated. They show a similar performance pattern as that has been obtained for the Birmingham city scenario. Moreover, they show that the CSA-VIKOR algorithm still has the best performance as compared to the other algorithms. In conclusion, the relative results of the algorithms have not been changed by changing the city under consideration. However, the absolute values of the average travel times, fuel consumption and travel speed levels have changed due to the difference in the size of roads and ring roads of Turin map used as compared to Birmingham map. The improved performance over the comparison algorithms is due to the fact that real-time traffic information has been used to continuously optimize travel time, travel speed, fuel consumption and CO_2 emissions.

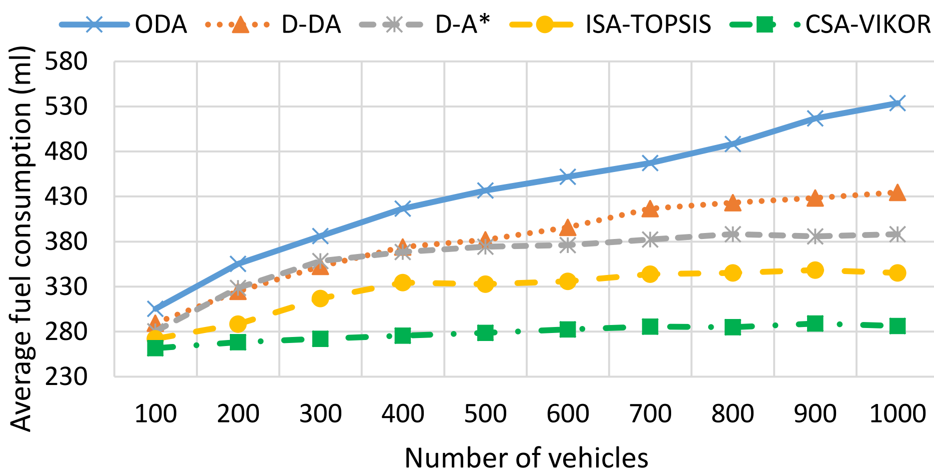


Figure 18: Average fuel consumption

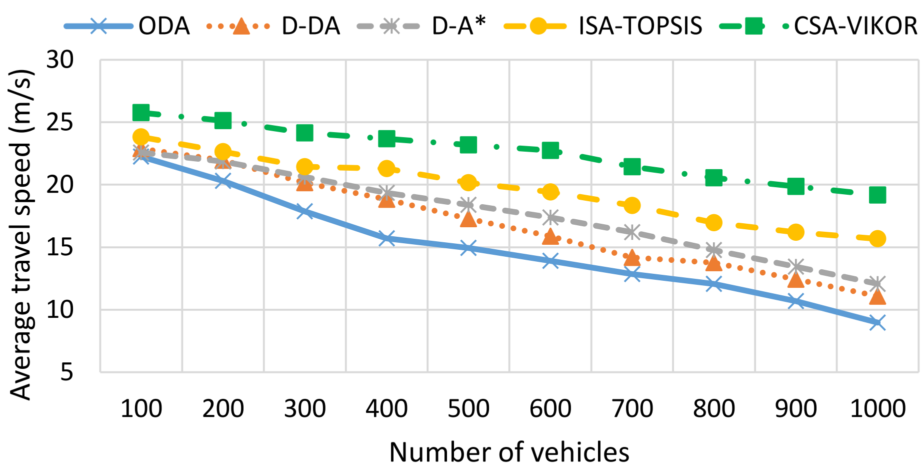


Figure 19: Average travel speed

5. Conclusion

As the number of vehicles on road networks increases so does the problem of congestion, making the development of congestion avoidance algorithms an important problem. In this paper, a new congestion alleviation method called a centralized simulated annealing VIKOR (CSA-VIKOR) algorithm has been proposed. The novelty of this work consists of the developed optimization algorithm based on a multi-objective cost function and dynamic route planning. The proposed method can lead to a reduction in global travel time, fuel consumption and CO_2 emissions. Simulation results from the Birmingham scenario show that the proposed approach has a better performance as compared to the original Dijkstra's algorithm, dynamic Dijkstra's algorithm, A^* algorithm and an improved simulated annealing TOPSIS algorithm. As reported from the Birmingham scenario, it is shown that the proposed approach improves traffic efficiency in the presence of congestion by an overall average of 24.05%, 48.88% and 36.89% in terms of travel time, fuel consumption and CO_2 emissions, respectively. Moreover, similar performance patterns were obtained for the Turin-based simulation.

Acknowledgment

The authors are grateful to the Iraqi Government, especially The Ministry of Higher Eduction and The Higher Committee For Education Development in Iraq. They would also like to acknowledge the support from the SETA project funded from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 688082.

References

[1] T. Litman, Congestion costing critique: Critical evaluation of the urban mobility, Report, Victoria Transport Policy Institute, USA, 2013.

[2] L.-W. Chen, P. Sharma, Y.-C. Tseng, Dynamic traffic control with fairness and throughput optimization using vehicular communications, IEEE Journal on Selected Areas in Communications 31 (9) (2013) 504–512.

[3] A. Kanungo, A. Sharma, C. Singla, Smart traffic lights switching and traffic density calculation using video processing, in: Proceedings of Recent Advances in Engineering and Computational Sciences (RAECS), 2014, IEEE, 2014, pp. 1–6.

[4] Z. Shelby, C. Bormann, 6LoWPAN: The Wireless Embedded Internet, John Wiley & Sons, 2009.

[5] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A Survey, Computer networks 54 (15) (2010) 2787–2805.

[6] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An information framework for creating a smart city through internet of things, IEEE Internet of Things journal 1 (2) (2014) 112–121.

[7] V. Kostakos, T. Ojala, T. Juntunen, Traffic in the smart city: Exploring city-wide sensing for traffic control center augmentation, IEEE Internet Computing (6) (2013) 22–29.

[8] J. Wang, C. Jiang, Z. Han, Y. Ren, L. Hanzo, Internet of vehicles: Sensing-aided transportation information collection and diffusion, IEEE Transactions on Vehicular Technology 67 (5) (2018) 3813–3825.

[9] M. Boban, R. Meireles, J. Barros, P. Steenkiste, O. K. Tonguz, Tvrtall vehicle relaying in vehicular networks, IEEE Transactions on Mobile Computing 13 (5) (2014) 1118–1131.

[10] J. Wang, C. Jiang, K. Zhang, T. Q. Quek, Y. Ren, L. Hanzo, Vehicular sensing networks in a smart city: Principles, technologies and applications, IEEE Wireless Communications 25 (1) (2018) 122–132.

[11] J. Wang, C. Jiang, L. Gao, S. Yu, Z. Han, Y. Ren, Complex network theoretical analysis on information dissemination over vehicular networks, in: Proceedings of Communications IEEE International Conference on (ICC) 2016, IEEE, 2016, pp. 1–6.

[12] C.-L. Hwang, K. Yoon, Multiple attribute decision making: methods and applications a state-of-the-art survey, Vol. 186, Springer Science & Business Media, 2012.

[13] S. Opricovic, G.-H. Tzeng, Compromise solution by mcdm methods: A comparative analysis of VIKOR and TOPSIS, European journal of operational research 156 (2) (2004) 445–455.

[14] H. Amer, N. Salman, M. Hawes, M. Chaqfeh, L. Mihaylova, M. Mayfield, An improved simulated annealing technique for enhanced mobility in smart cities, Sensors 16 (7) (2016) 1013.

[15] S.-h. An, B.-H. Lee, D.-R. Shin, A survey of intelligent transportation systems, in: Proceedings of Third International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), 2011, IEEE, 2011, pp. 332–337.

[16] C. Kan, J. Miles, Its handbook 2000, recommendations from the world road associations (piarc), Tech. rep., ISBN 1-58053-103-2 (2000).

[17] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische mathematik 1 (1) (1959) 269–271.

[18] V. T. N. Nha, S. Djahel, J. Murphy, A comparative study of vehicles' routing algorithms for route planning in smart cities, in: Proceedings of the First International Workshop on the Vehicular Traffic Management for Smart Cities (VTM), IEEE, 2012, pp. 1–6.

[19] S. Wang, S. Djahel, J. McManis, C. McKenna, L. Murphy, Comprehensive performance analysis and comparison of vehicles routing algorithms in smart cities, in: Proceedings of Global Information Infrastructure Symposium, 2013, IEEE, 2013, pp. 1–8.

[20] V. Jindal, H. Dhankani, R. Garg, P. Bedi, Maco: Modified aco for reducing travel time in vanets, in: Proceedings of the Third International Symposium on Women in Computing and Informatics, ACM, 2015, pp. 97–102.

[21] A. Faro, D. Giordano, Algorithms to find shortest and alternative paths in free flow and congested traffic regimes, Transportation Research Part C: Emerging Technologies 73 (2016) 1–29.

[22] P. E. Hart, N. J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics, 4 (2) (1968) 100–107.

[23] M. Dorigo, G. Di Caro, L. M. Gambardella, Ant algorithms for discrete optimization, Artificial life 5 (2) (1999) 137–172.

[24] H. Kanoh, K. Hara, Hybrid genetic algorithm for dynamic multi-objective route planning with predicted traffic in a real-world road network, in: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, ACM, 2008, pp. 657–664.

[25] S. Samaranayake, J. Reilly, W. Krichene, J.-B. Lespiau, M. L. Delle Monache, P. Goatin, A. Bayen, Discrete-time system optimal dynamic traffic assignment (SO-DTA) with partial control for horizontal queuing networks, in: Proceedings of the IEEE American Control Conference (ACC), 2015, IEEE, 2015, pp. 663–670.

[26] C. P. Chen, J. Zhou, W. Zhao, A real-time vehicle navigation algorithm in sensor network environments, IEEE Transactions on Intelligent Transportation Systems 13 (4) (2012) 1657–1666.

[27] M. R. Jabbarpour, A. Jalooli, E. Shaghaghi, R. M. Noor, L. Rothkrantz, R. H. Khokhar, N. B. Anuar, Ant-based vehicle congestion avoidance system using vehicular networks, Engineering Applications of Artificial Intelligence 36 (2014) 303–319.

[28] A. M. de Souza, R. S. Yokoyama, L. C. Botega, R. I. Meneguette, L. A. Villas, Scorpion: A solution using cooperative rerouting to prevent congestion and improve traffic condition, in: Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), IEEE, 2015, pp. 497–503.

[29] A. M. de Souza, R. S. Yokoyama, N. L. S. da Fonseca, R. I. Meneguette, L. A. Villas, Garuda: A new geographical accident aware solution to reduce urban congestion, in: Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), IEEE, 2015, pp. 596–602.

[30] J. Pan, I. S. Popa, C. Borcea, Divert: A distributed vehicular traffic re-routing system for congestion avoidance, IEEE Transactions on Mobile Computing 99 (2016) 1–1.

[31] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, F. Bai, Real-time path planning based on hybrid-vanet-enhanced transportation system, IEEE Transactions on Vehicular Technology 64 (5) (2015) 1664–1678.

[32] E. Angelelli, I. Arsik, V. Morandi, M. Savelsbergh, M. Speranza, Proactive route guidance to avoid congestion, Transportation Research Part B: Methodological 94 (2016) 1–21.

[33] F. Ricardo, H. Qie, S. Yongjia, A disjunctive convex programming approach to the pollution-routing problem, Transportation Research Part B 94 (2016) 61–79.

[34] L. Chung-Cheng, L. Jiangtao, Q. Yunchao, P. Srinivas, M. R. Nagui, Z. Xuesong, Eco-system optimal time-dependent flow assignment in a congested network, Transportation Research Part B 94 (2016) 217–239.

[35] F. D. Da Cunha, A. Boukerche, L. Villas, A. C. Viana, A. A. Loureiro, Data communication in VANETs: a survey, challenges and applications, Ph.D. thesis, INRIA Saclay (2014).

[36] E. Egea-Lopez, P. Pavon-Marino, Distributed and fair beaconing rate adaptation for congestion control in vehicular networks, IEEE Transactions on Mobile Computing 15 (12) (2016) 3028–3041.

[37] Y. Yao, L. Rao, X. Liu, Performance and reliability analysis of ieee 802.11 p safety communication in a highway environment, IEEE Transactions on Vehicular Technology 62 (9) (2013) 4198–4212.

[38] M. A. Chowdhury, A. W. Sadek, Fundamentals of intelligent transportation systems planning, Artech House, U.S., 2003.

[39] N. Parrado, Y. Donoso, Congestion based mechanism for route discovery in a V2I-V2V system applying smart devices and IoT, Sensors 15 (4) (2015) 7768–7806.

[40] Ministerio de Transporte de Colombia. Codigo de Trnsito de Colombia, `http://www.colombia.com/noticias/codigotransito/`, accessed: 2016-09-30.

[41] M. Chaqfeh, A. Lakas, A novel approach for scalable multi-hop data dissemination in vehicular ad hoc networks, Ad Hoc Networks 37 (2016) 228–239.

[42] S. Kirkpatrick, M. P. Vecchi, et al., Optimization by simmulated annealing, Science 220 (4598) (1983) 671–680.

[43] Y.-M. Wang, Y. Luo, Integration of correlations with standard deviations for determining attribute weights in multiple attribute decision making, Mathematical and Computer Modelling 51 (1) (2010) 1–12.

[44] Z. Wu, Y. Chen, The maximizing deviation method for group multiple attribute decision making under linguistic environment, Fuzzy Sets and Systems 158 (14) (2007) 1608–1617.

[45] C. Sommer, R. German, F. Dressler, Bidirectionally coupled network and road traffic simulation for improved IVC analysis, IEEE Transactions on Mobile Computing 10 (1) (2011) 3–15.

[46] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz, Sumo–simulation of urban mobility, in: Proceedings of the The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain, 2011.

[47] What is OMNeT++?., OMNet++discrete event simulator, opensimltd. web, https://omnetpp.org/, accessed: 2015-10-23.

[48] M. Haklay, P. Weber, Openstreetmap: User-generated street maps, Pervasive Computing, IEEE 7 (4) (2008) 12–18.

[49] A. Cappiello, I. Chabini, E. K. Nam, A. Lue, M. Abou Zeid, A statistical model of vehicle emissions and fuel consumption, in: Proceedings of the 5th IEEE International Conference on Intelligent Transportation Systems, 2002, IEEE, 2002, pp. 801–809.