FINDING LOWER BOUNDS OF LOCALIZATION WITH NOISY MEASUREMENTS USING GENETIC ALGORITHMS

TECHNICAL REPORT 675

FARHAN AHAMMED, JAVID TAHERI AND ALBERT ZOMAYA

MAY, 2011
Finding lower bounds of localization with noisy measurements using genetic algorithms

Farhan Ahammed∗†, Javid Taheri∗ and Albert Zomaya∗
∗School of Information Technologies
The University of Sydney
NSW 2006, Australia
{farhan.ahammed, javid.taheri, albert.zomaya}@sydney.edu.au
†NICTA Australia
Australian Technology Park
Level 5, 13 Garden Street
Eveleigh NSW 2015, Australia

Abstract—Vehicular Ad-Hoc Networks (VANETs) are wireless networks with mobile nodes (vehicles) which connect in an ad-hoc manner. Many vehicles use the Global Positioning System (GPS) to provide their locations. However the inaccuracy of GPS devices leads to some vehicles incorrectly assuming they are located at different positions and sometimes on different roads. VANETs can be used to increase the accuracy of each vehicle’s computed location by allowing vehicles to share information regarding the measured distances to neighbouring vehicles. This paper looks at finding how much improvement can be made given the erroneous measurements present in the system. An evolutionary algorithm is used to evolve instances of parameters used by the VLOCI2 algorithm, also presented in this paper, to find instances which minimises the inaccuracy in computed locations. Simulation results show a definite improvement in location accuracy and lower bounds on how much improvement is possible is inferred.

Index Terms—vehicular ad hoc networks, localization, GPS, distance measurements, location improve/refinement, evolutionary algorithm, localization lower bounds

I. INTRODUCTION

Wireless Sensor Networks (WSNs) and Vehicular Ad Hoc Networks (VANETs) are two types of wireless networks where the nodes are connected in an ad hoc manner. Situations may arise in both these networks where the locations of each node of the network is required for sufficient performance of the network. Localization is the name given to the class of algorithms which allow each node in the network to discover its location. Additionally, localization improvement algorithms can be implemented allowing the nodes to further refine their computed locations after aggregating other measured data.

VANETs are mobile networks where the nodes are vehicles. The vehicles are equipped with wireless communication devices allowing them to transmit and share real-time information. With this information vehicles and drivers will have up-to-date information regarding the state of traffic, allowing them to avoid congested and other abnormally affected areas. VANETs are dynamic with vehicles travelling at speeds up to, and in excess of, 100 km/h. This leads to ever-changing wireless connections between vehicles resulting in some dense (on some city roads) and sparse (on country roads) areas which change over time (some city roads are dense only during certain hours of the day).

Many vehicles are nowadays equipped with GPS devices and it is quite possible that most, if not all, vehicles will have these devices in the future. GPS devices are accurate to within 10 metres [1]—more than the length of most family cars—resulting in the GPS device reporting incorrect locations of its vehicles, sometimes placing them on the wrong road. Obtaining more accurate coordinates (position estimates) allows the vehicles to construct more precise models of their local traffic conditions.

With increased accuracy and better models, accidents can be prevented. Multi-car ‘pile-ups’ can be avoided if vehicles know immediately that other vehicles further in front are stopping suddenly or skidding. Although some sensors are already providing information about other vehicles directly in front and around the vehicle, VANETs can be used to provide information about vehicles further away. Increasing the accuracy of each vehicle’s computed location will improve safety as vehicles can make more informed decisions on how to react in dangerous situations.

VANETs can be used to improve on the position estimates provided by the GPS devices. Every vehicle
can use the VANET to provide their position estimate to all neighbouring vehicles within broadcasting range. The vehicles can also measure the distance between them and other vehicles using either specialised sensors/equipments [2], [3] or the wireless communication devices by employing radio frequency techniques such as time-of-arrival or received signal strength [4].

One of the most important issues facing the vehicles is that measured data will always be corrupted by noise. Coordinates provided by GPS devices, mentioned earlier, are accurate to within 10 metres. Data obtained by the distance measurement devices are also expected to not be exact. For this reason any localization and localization improvement algorithm should not be expected to produce exact results. The question, considered in this paper, is given the error present in the system, what is the lower bound of the achievable localization? Or what is the best level of accuracy that can be achieved given the erroneous data used to make the calculations?

To answer these questions, localisation improvement is viewed as an optimisation problem. Where the aim is, given the erroneous data each node has to work with, what is the best improvement of node location coordinates achievable. An evolutionary algorithm is the heuristic used in this paper to solve the optimisation problem.

An overview of previous work found in literature is presented in Section II. Section V introduces the notation and defines the problem addressed in this paper. The evolutionary algorithm is described in Section VI. Section VII and VIII present the results of the evolutionary algorithm and the simulations used to test the effectiveness of the instance found by the evolutionary algorithm. Discussions of the various results and concluding remarks are presented in Sections VII and VIII respectively.

II. RELATED WORK

There does not seem to be much work in literature addressing the idea of finding a lower bound in localization in wireless networks where erroneous data is present. There are localization algorithms designed to take advantage of some nodes that have GPS, or some other positioning functionality, to provide localization. Some algorithms use novel techniques while others use optimization algorithms.

Wang et al. used a Bayesian method analysis to explore the lower bound of localization uncertainty in sensor networks [5]. The aim of the sensor network considered by Wang et al. is to track the locations of objects within the sensing area. The lower bounds found are related to the position of the tracked object, not the sensors themselves. Multiple sensors are used to track a single object, where each of the sensors know of their exact location and the distance measurements they take are modelled as falling under a Gaussian distribution.

Shekofteh et al. use tabu search with simulated annealing to provide localization [6]. Each non-anchor node is capable of measuring the distance to anchor nodes. Their initial computed position is the centroid of the anchor nodes. A tabu optimisation search is used to modify the computed position to a more accurate estimate. The nodes take distance measurements which are modelled with Gaussian noise. A ‘noise factor’ is introduced to control the magnitude of the noise. Without initial coordinates, the flip ambiguity problem needs to be addressed. This is the situation where the nodes have constructed a graph of the network where the distance between each nodes is correct, but the actual positions of some of the nodes are not, which is due to the graph not being globally rigid [7]. A simulated annealing algorithm is used to fix this problem which occurs when the distance between actual and initial coordinates of the nodes is so large the tabu search although finds the closest position satisfying the distance requirements, it is in fact the wrong position. With 10% of the nodes designated anchor nodes, simulation results show the algorithm does provide localization to the non-anchor nodes.

Moore et al. addresses the issue of localization where the distance measurements are corrupted by noise [8]. They address the problem of discovering the coordinates of the nodes of a graph given the edge length, and with no initial coordinate information. Their algorithm constructs clusters of four nodes that each form complete graphs (K_4), called ‘robust quadrilaterals’. The clusters are then merged such that the probability of ambiguities is minimized. Their simulations produced location errors that fluctuated between values of 0.5 and 5.

Kannan et al. used the simulated annealing optimisation technique to provide localization in WSNs [9]. A set of anchors are present in the network with known location information, for which each node can measure the distance to. A subset of non-anchor nodes is chosen at random and each one is perturbed a certain distance from their current estimated position. The new configuration is evaluated based on the measured and estimated distance between each non-anchor node and every anchor node. Distance measurements are modelled as being corrupted by Gaussian noise. The severity of the noise is controlled by a noise factor. Their simulations produced good results, however it does suffer from ‘flip ambiguity’ in situations where nodes are placed on the same line. When the transmission range of the nodes are
set to almost \( \frac{1}{3} \) of the size of the region, and with 10\% anchor nodes, the location error reached values of 30\% of the region size.

Some alternative solutions [10]–[12] are applied in situations where a subset of nodes, usually termed anchor nodes or base stations, have knowledge of their own positions. The remaining nodes then communicate with the anchor nodes to determine their locations as they have no other method to estimate their locations. Similarly, Benslimane [13] addresses the situation where not all vehicles are equipped with GPS devices, or some cannot obtain data from their GPS devices and need to collaborate with the GPS-equipped vehicles to determine their locations.

Ouyang et al. considered the problem of using terrestrial TOA measurements to improve initial GPS measurements [14]. Upon receiving the initial GPS measurements, a ranging technique (e.g. TOA) is used to measure the distance to nearby reference stations. A corresponding matrix equation is constructed, for which the weighted least squares estimator is applied to. The Levenberg-Marquardt method is then used to iteratively find a solution with the GPS estimate used as the initial solution. The idea of obtaining a single GPS estimate and single distance measurements is common with this work. However Ouyang et al. assume that the anchor nodes, for which the distance is measured from, know of their exact positions. Their positions are not obtained using GPS. Their technique produces improvements of up to 30\%.

McGuire and Plataniotis [15] developed the Cramér-Rao, Weinstein-Weiss and Ziv-Zakai lower bounds for localization in wireless networks. In their models the nodes perform localization taking measurements from fixed base stations (anchor nodes) with known locations. The locations of the anchor nodes, regardless of how knowledge of their location is obtained, is deemed to be accurate, and is not modelled to contain error. The distance measurements are the only quantities modelled to contain noise.

While previous work has been done on providing localization in sensor and vehicular ad hoc networks, there does not seem to be any work on determining how much improvement can be achieved after a localization phase is completed. Not much work exists which assumes both the initial localization method and further distance measurements are error-prone or corrupted with noise. It is these situations that are addressed in this paper.

II. PROBLEM STATEMENT

Let \( G \) be a network of vehicles. The set of all vehicles in \( G \) is denoted \( V(G) \). The number of vehicles in the network is \( |V| \). Given any particular vehicle \( v_i \in V \), any other vehicle that \( v_j \) can send and receive messages from are deemed its neighbours. The set of all neighbours of \( v_i \) is denoted \( \text{nbrs}(v_i) \) and the number of neighbours of \( v_i \) is \( m_i = |\text{nbrs}(v_i)| \).

Each vehicle \( v_i \) is located at \( p_i = (x_i, y_i) \) (true location) and, due to inaccurate measurements, calculates that it is located at \( \hat{p}_i = (\hat{x}_i, \hat{y}_i) \) (its computed/estimated location). The words ‘location’ and ‘position’ are also used interchangeably. The location error for \( v_i \) is the distance between its true and computed location \( \delta_i = \| p_i - \hat{p}_i \| \). It is assumed every vehicle is equipped with a GPS device such that their initial position estimate is

III. GENETIC ALGORITHMS

Genetic algorithms is a class of Evolutionary Algorithms which aims to use a process similar to natural selection viewed in nature to find near-optimal solutions to optimisation problems.

Given an optimisation problem, a set, or population, of candidate solutions is randomly constructed. An objective function is applied to each solution to assess their fitness, allowing the solutions to be ranked in terms of how ‘well’ they satisfy the constraints of the optimisation problem. Based on the ranking, some solutions are chosen to be parents to produce children of the next generation who inherit features from their parents. Some children may be mutated, where certain aspects of those solutions are changed. The cycle repeats, producing the next generation.

Genetic algorithms are defined by the following components: Representation (definition of individuals or candidate solutions); Evaluation function (or fitness function); Population (a set of candidate solutions); Parent selection mechanism; Recombination (or crossover) to generate children and Mutation of some children.

An initial set of candidate solutions is created and the evaluation function is applied to all candidates. Based on the fitness of the individuals, some are selected to be parents and will be used to generate a new set of individuals (children). The crossover function used to generate the children ensures that characteristics of the parents are inherited by the children. The aim is that each generation inherits the favourable characteristics of its predecessors. Additionally, some children of each generation may be selected to undergo mutation where small random changes are made to the characteristics of the selected individuals.

The process of generating and mutating children to create a new population is repeated for a set number of generations, or until a satisfactory candidate solution is found.

IV. PROBLEM STATEMENT

Let \( G \) be a network of vehicles. The set of all vehicles in \( G \) is denoted \( V(G) \). The number of vehicles in the network is \( |V| \). Given any particular vehicle \( v_i \in V \), any other vehicle that \( v_j \) can send and receive messages from are deemed its neighbour. The set of all neighbours of \( v_i \) is denoted \( \text{nbrs}(v_i) \) and the number of neighbours of \( v_i \) is \( m_i = |\text{nbrs}(v_i)| \).

Each vehicle \( v_i \) is located at \( p_i = (x_i, y_i) \) (true location) and, due to inaccurate measurements, calculates that it is located at \( \hat{p}_i = (\hat{x}_i, \hat{y}_i) \) (its computed/estimated location). The words ‘location’ and ‘position’ are also used interchangeably. The location error for \( v_i \) is the distance between its true and computed location \( \delta_i = \| p_i - \hat{p}_i \| \). It is assumed every vehicle is equipped with a GPS device such that their initial position estimate is
a value \((\bar{x}_i^0, \bar{y}_i^0)\) where \(x_i - 10 < \bar{x}_i^0 < x_i + 10\) and \(y_i - 10 < \bar{y}_i^0 < y_i + 10\).

It is further assumed that every vehicle can take distance measurements between them and their neighbouring vehicles. The true distance between vehicles \(v_i\) and \(v_j\) is denoted \(d_{i,j}\), while the distance measured by \(v_i\) between itself and vehicle \(v_j\) is given as \(\hat{d}_{i,j} = \varepsilon \cdot d_{i,j}\), for some \(\varepsilon \in \mathbb{R}\). Since each vehicle is assumed to have their own measuring device, it is also assumed that \(\hat{d}_{i,j} \neq d_{i,j}\) because \(\hat{d}_{i,j}\) is the distance measured by \(v_i\) while \(d_{i,j}\) is the distance measured by \(v_j\).

A metric used to gauge the performance of the localisation algorithm of a network is the network location error (Definition 1). For each vehicle \(v_i\), its location error is already defined as \(\delta_i\). The network location error is the average of every vehicle’s location error.

**Definition 1 (Network Location Error).** Given a network of vehicles \(V\), where each vehicle \(v_i\) believes it is located at \(\hat{p}_i\), while its true location is \(p_i\). The location error of the network \((E_V)\) is defined as

\[
E_V = \frac{1}{n} \sum_{v_i \in V} \delta_i = \frac{1}{n} \sum_{v_i \in V} \|p_i - \hat{p}_i\|
\tag{1}
\]

Using the definition of network location error, the problem of localization can be formulated as shown in Definition 2.

**Definition 2 (Localization).** Given a network of vehicle \(V\), the goal of localization is for every vehicle \(v_i \in V\), located at \(p_i = (x_i, y_i)\), to compute its position \(\hat{p}_i = (x'_i, y'_i)\) such that the network location error is minimised.

The problem of Location Improvement (Definition 3) is referred to in this paper. The aim is to find a method of adjusting every vehicle’s current estimated position \(\hat{p}_i\) such that the location error is reduced.

**Definition 3 (Location Improvement).** Let \(V\) be a network of vehicles. Assume that every vehicle \(v_i\) has computed its own estimated position \(\hat{p}_i\). The problem of Location Improvement is to find a function \(\phi\) which modifies some or all of the vehicle’s estimated position such that the location error of the modified network has minimised.

**Definition 4 (VLOCI2).** VANET Location Improvement in 2 dimensions. A location improvement algorithm previously developed by Ahammed et al. [], which improves the computed locations of each vehicle where they are located on multiple lanes.

A. The optimisation problem

The problem addressed in this paper is to find a lower bound on how much the network location error can be reduced by given the level of erroneous data present in the system and used by the vehicles of the network.

Every vehicle \(v_i\) uses \(\phi\) to improve its location

\[
i, \hat{p}_1, \ldots, \hat{p}_N, \hat{d}_{i,1}, \ldots, \hat{d}_{i,N} \xrightarrow{\phi(w)} \hat{p}'_i
\]

the information given as input is its own vehicle ID \(i\), all its neighbours’ estimated positions \(\hat{p}_1, \ldots, \hat{p}_N\) and the measured distance to each of its neighbours \(\hat{d}_{i,1}, \ldots, \hat{d}_{i,N}\). The output is a new position \(\hat{p}'_i\).

The function \(\phi\), which is the VLOCI2 algorithm, is dependent on a weight function \(w\). The weight function is used to compute a weighted average because vehicles that are closer together can make more accurate distance measurements between each other.

The aim is to find a weight function that minimises the location error of the network (average distance between each vehicle’s estimated and true position).

V. THE EVOLUTIONARY ALGORITHM

This weight function is used to provide empirical evidence of a lower bound on the problem of location improvement. The lower bound will show that given the erroneous data present within the network, what is the lowest location error (that is, most accurate network model) that can be achieved.

The optimisation problem addressed in this paper is shown below

**Instance**

The weight function \(w\). The values of \(\{w_1, \ldots, w_k\}\) are to be found while \(\{\lambda_1, \ldots, \lambda_{k-1}\}\) are fixed

**Objective Function**

Let \(G\) be a graph. The objective value is computed by performing localization on \(G\) using the VLOCI2 algorithm. That is, the objective value is \(\sum_{i \in V(G)} \|\hat{p}'_i - \hat{p}_i\|\) where \(\hat{p}'_i = \phi(w)(i, \{\hat{p}_j, \hat{d}_{i,j}\})\)

**Goal**

Minimise the objective function

The graph \(G\) used will be designed to contain a large number of nodes, allowing the weight function to be tested on a large set of distance measurements obtained from neighbours.

A. Representation

It is the weight function \(w\) that the optimisation algorithm will search for. An instance is represented as a list of the \(\{w_1, \ldots, w_k\}\) values, with \(\lambda_i = 5i, k = 60\).
B. Objective Function

Given an instance \( I = w(x) \) the objective function is for each node of the network to apply the function \( \phi \), which uses the weight function \( w \), when running the localisation improvement algorithm. After all the nodes have updated their position estimates, The network location error becomes the objective value used to describe how well the instance improves the location improvement algorithm.

C. Population

The size of the population is of the form \( 2p \), where \( p \) is the number of parents. Each pair of chosen parents produces two children. The best \( p \) individuals in the population will be used as the parents to produce the next generation. The number of generations produced is 1000 before the evolutionary algorithm is stopped.

D. Parent selection

The population is sorted in decreasing order of \( f(F) \) and the first \( p \) individuals are used as parents. If the sorted set of individuals is \( \{ \xi_0, \xi_1, \ldots, \xi_{2p} \} \) then each pair of parents are selected at random according to the following non-uniform probability distribution:

\[
P(\xi_i) = \frac{h(\xi_i)}{\sum_{j=0}^{p} h(\xi_j)}
\]

where,

\[
h(\xi_i) = \begin{cases} h(\xi_{i+1}) \times 1.2, & 0 \leq i < p \\ 10, & i = p \\ 0, & i > p 
\end{cases}
\]

The constants above were chosen empirically. The fitter parents have a higher probability of being chosen.

E. Recombination

The single-point crossover technique is used to generate children from a pair of parents. Suppose the two parents chosen are

\[
p_1 = \{ w_1^1, w_2^1, \ldots, w_k^1 \} 
\]

\[
p_2 = \{ w_1^2, w_2^2, \ldots, w_k^2 \}
\]

To create the children, a random number \( 1 \leq \eta \leq k \) is first chosen. The children created will then have the form

\[
c_1 = \{ w_1^{\eta}, w_2^{\eta}, \ldots, w_k^{\eta} \} 
\]

\[
c_2 = \{ w_1^{\eta+1}, w_2^{\eta+1}, \ldots, w_k^{\eta+1} \}
\]

where

\[
w_i^{\eta} = \begin{cases} w_i^1, & i \leq \eta \\ w_i^2, & i > \eta 
\end{cases}
\]

\[
w_i^{\eta+1} = \begin{cases} w_i^1, & i \leq \eta \\ w_i^2, & i > \eta 
\end{cases}
\]

F. Mutation

Of all the children generated, each child is selected for mutation with probability 0.25. If a child is selected for mutation, one of the \( w_i \) in its weighted function is selected at random and is modified according to the following rule

\[
w_i' = w_i \times \varepsilon
\]

where \( \varepsilon \sim N(1, 25) \), the Gaussian distribution with mean 1 and standard deviation 5. The mean and standard deviation were chosen empirically.

G. Evolutionary algorithm overview

The evolutionary algorithm overview is designed to find an optimal solution to search for a weighted average function such that when used by the localization algorithm, will reduce the location error by a near-optimal minimum amount. The outline of the algorithm used is presented below:

1) Generate the graph used for evaluating instances
2) Set the estimated positions of all nodes to random values within \( \pm 10 \) of true position
3) Evaluate the population. Run the objective function (i.e. VLOC12). The new location error of the network becomes the objective value of each instance (weighted function).
4) Repeat (I) times:
   a) Generate children (recombine)
   b) Mutate selected children
   c) Set the estimated positions of all nodes to random values with to \( \pm 10 \) of true position
   d) Evaluate the population.

VI. Simulation results

Firstly the evolutionary algorithm was run to find an instance which minimises the location error, then simulations were performed to assess the effectiveness of the found instance.

A. Simulation parameters

1) Network model: The network itself was modeled using the Paramics Modeler v6.7.1 [16]. The vehicles travelled on a 4-lane highway where each lane was 3.65 metres wide. The length of the highway was 3km and the speed limit was 120 km/h, although some vehicles may drive faster than the speed limit due to the realistic nature of Paramics.

The communication range of the vehicles were set to 100 metres. The reasoning for this is because since the initial GPS coordinates are set to within 10 metres, when the distance measurement accuracy is set to a 10% accuracy, the distance measured to vehicles 100m...
away will result in a measurement within the range 100 ± 10 metres. For the vehicles further away, it would be the case their GPS coordinate alone will provide better accuracy. The weight function coefficients were chosen empirically.

When evaluating an instance, 4 hours of vehicles traveling across the highway was simulated. Once a vehicle reached the end of the highway, it did not re-enter. The rate at which the number of vehicles entered the highway was varied to allow for control on the vehicle density. The vehicle density is defined such that at 100%, 8000 vehicles are released per hour. The different densities tested in the simulations were 10% (i.e. 800 vehicles per hour), 15%, 20% and 25%. For the evaluation function, the simulation was set to have each density occurring for an hour.

The distance measurement accuracy was fixed at 10%. However the final instance found was tested on various levels of accuracy. The number of distance measurements each node takes (∆) should be based on the technique used to obtain distance measurements—smaller values of ∆ can be used with more accurate distance measuring devices. For the simulations, ∆ is constant, with five measurements taken each time to counter the variance in the measurements.

Each node runs the location improvement algorithm once, that is, the number of iterations I is set to one. When testing the final instance found, the number of iterations performed by each node is increased to assess the effectiveness of multiple iterations.

The simulation parameters for the evolutionary algorithm, network structure and evaluation function are summarised in Table I.

The parameters used for these tests are shown in Table II. Accurate distance measurements (i.e. α = 0) is used as a benchmark to compare with the cases when erroneous data is present in the system. In the simulations, each vehicle performs 10 iterations, this allows for the assessment on the number of iterations the location improvement algorithm can be restricted to.

B. Evolutionary algorithm results

The final instance (weight function) found is shown in Figure 1. The objective value of the fittest instance (average location error) in the population is used to determine the fitness of the population and Figure 2 shows how the population improved over time.

C. Weight function tests and results

Figures 3 and 4 shows how the weight function performs with differing levels of accuracy in distance measurements in the scenarios with the vehicle density set to 10% and 25% respectively.

Figure 5 shows the reduction in location error for each combination of vehicle density and distance measurement accuracy.

At the end of each tenth iteration, the percentage of vehicles that correctly computed to be in the correct lane is shown in Figure 6. The remaining vehicles thus placed themselves either on the adjacent lane, or further

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highway length (L)</td>
<td>3 km</td>
</tr>
<tr>
<td>No. lanes</td>
<td>4</td>
</tr>
<tr>
<td>Lane width</td>
<td>3.65 m</td>
</tr>
<tr>
<td>Speed limit</td>
<td>120 km/h</td>
</tr>
<tr>
<td>Dist. measurement accuracy (α)</td>
<td>10%</td>
</tr>
<tr>
<td>No. Dist. Meas. taken (∆)</td>
<td>5</td>
</tr>
<tr>
<td>Vehicle densities used</td>
<td>10%, 15%, 20%, 25%</td>
</tr>
<tr>
<td>No. iterations (I)</td>
<td>1</td>
</tr>
<tr>
<td>Sim. time per scenario</td>
<td>4 hrs, each veh. density per hr</td>
</tr>
<tr>
<td>No. Parents (p)</td>
<td>20</td>
</tr>
<tr>
<td>No. Generations</td>
<td>1000</td>
</tr>
<tr>
<td>Weight function intervals λ&lt;sub&gt;k&lt;/sub&gt;</td>
<td>5k, k = 1, . . . , 20</td>
</tr>
</tbody>
</table>

**TABLE I**
The simulation parameters used when running the evolutionary algorithm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highway length (L)</td>
<td>3 km</td>
</tr>
<tr>
<td>No. lanes</td>
<td>4</td>
</tr>
<tr>
<td>Lane width</td>
<td>3.65 m</td>
</tr>
<tr>
<td>Speed limit</td>
<td>120 km/h</td>
</tr>
<tr>
<td>Dist. measurement accuracy (α)</td>
<td>0%, 5%, 10%</td>
</tr>
<tr>
<td>No. Dist. measurements taken (∆)</td>
<td>5</td>
</tr>
<tr>
<td>Vehicle densities tested</td>
<td>10%, 15%, 20%, 25%</td>
</tr>
<tr>
<td>No. iterations (I)</td>
<td>10</td>
</tr>
<tr>
<td>Sim. time per scenario</td>
<td>1 hrs, each density per 25min</td>
</tr>
</tbody>
</table>

**TABLE II**
The simulation parameters used.

The parameters used for these tests are shown in Table II.

![Fig. 1. The weight function.](image-url)
VII. DISCUSSION AND ANALYSIS

A. The weight function

The weight function (Figure 1) decreases for values of \( x \geq 60 \). For values of \( 15 \leq x < 60 \) the weight function seems to decrease before increasing again.

However, the results show the weight function does cause the location error to reduce. With the accuracy of distance measurements set to 10\%, the location error after 10 iterations was reduced by 34\% when the vehicle density was 10\%, while the reduction was 50\% with the vehicle density set at 25\%.

B. Effects of distance measurement accuracy

For each vehicle density tested, the location error decreases as the distance measurement accuracy increases, as shown in Figure 5. When the vehicle density was at the highest, the location error reduced by more than 50\% for each of the distance measurement accuracy tested.

C. Effects of vehicle density

Figure 5 also shows how, regardless of the distance measuring accuracy, an increase in the vehicle density size results in a further reduction in location error. The figure also shows that the location error decreases after 10 iterations in all tested scenarios.

D. Number of iterations required

In each scenario tested, every node completed 10 iterations in the VLOCI2 algorithm. Figure 4 shows the location error strictly decreasing during the 10 iterations for when the vehicle density was at 25\%. In Figure 5 where the vehicle density is 10\%, the location error decreases during the 10 iterations when the distance measuring accuracy was 0\% and 5\%.

E. Lower bound on location error

Table III shows the lowest values the location error reached under the different tested conditions. The lowest value may not necessarily have occurred after the tenth iteration. This table demonstrates the lowest values achieved by the VLOCI2 algorithm with the found weight function.

Under the ideal scenario with \( \alpha = 0 \) the location error can be reduced to a value of 2.8 with the vehicle density at 25\%. For all but one of the scenarios tested, the location error had reached a minimum value that is less than the width of the lanes (3.56\text{m}).
Vehicles in correct/incorrect lanes

Δ Loc. Err. (10 iterations)

0%

25%

50%

Correct lane 1 lane off 2 lanes off 3 lanes off

Vehicle density

10%

15%

20%

25%

FIG. 6. The % of vehicles that computed to be in the correct lane, and those that were one, two or three lanes off.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Vehicle density</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>0%</td>
<td>3.0</td>
</tr>
<tr>
<td>5%</td>
<td>3.4</td>
</tr>
<tr>
<td>10%</td>
<td>3.7</td>
</tr>
</tbody>
</table>

TABLE III

THE LOWEST VALUES THE LOCATION ERROR REACHED UNDER THE DIFFERENT SCENARIOS.

VIII. CONCLUSIONS

The instance (weight function) found by the evolutionary algorithm, in conjunction with the VLOCI2 algorithm, is shown to improve the location error of VANETs vehicles receive an initial inaccurate GPS coordinate and use imprecise distance measurements to their neighbouring nodes to aid with calculations.

With accurate distance measurements, the location error can be reduced to values less than the width of a highway lane. In most cases, around 70% of all vehicles compute themselves to be in either the correct lane or an adjacent one. The weight function found by the evolutionary algorithm shows that with erroneous distance measurements and inaccurate GPS coordinates, the location error can at least be reduced to 2.8 metres.

REFERENCES


