A Multi-Objectif Genetic Algorithm-Based Adaptive Weighted Clustering Protocol in VANET
Mohamed Hadded, Rachid Zagrouba, Anis Laouiti, Paul Muhlethaler, Leila Azouz Saidane

To cite this version:

HAL Id: hal-01211442
https://hal.archives-ouvertes.fr/hal-01211442v2
Submitted on 25 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A Multi-Objective Genetic Algorithm-Based Adaptive Weighted Clustering Protocol in VANET

Mohamed Hadded *,†, Rachid Zagrouba *, Anis Laouiti †, Paul Muhlethaler ‡ and Leila Azouz Saidane *
*RAMSIS Team, CRISTAL Laboratory, 2010 Campus University, Manouba, Tunisia
†TELECOM SudParis, CNRS Samovar, UMR 5157, France
‡INRIA, BP 105. 78153 Le Chesnay Cedex, Paris-Rocquencourt, France
{mohamed.haddad, leila.saidane}@ensi.rnu.tn, rachid.zagrouba@cristal.rnu.tn, anis.laouiti@telecom-sudparis.eu,
paul.muhlethaler@inria.fr

Abstract—Vehicular Ad hoc NETworks (VANETs) are a major component recently used in the development of Intelligent Transportation Systems (ITSs). VANETs have a highly dynamic and portioned network topology due to the constant and rapid movement of vehicles. Currently, clustering algorithms are widely used as the control schemes to make VANET topology less dynamic for Medium Access Control (MAC), routing and security protocols. An efficient clustering algorithm must take into account all the necessary information related to node mobility. In this paper, we propose an Adaptive Weighted Clustering Protocol (AWCP), specially designed for vehicular networks, which takes the highway ID, direction of vehicles, position, speed and the number of neighboring vehicles into account in order to enhance the stability of the network topology. However, the multiple control parameters of our AWCP, make parameter tuning a non-trivial problem. In order to optimize the protocol, we define a multi-objective problem whose inputs are the AWCP’s parameters and whose objectives are: providing stable cluster structures, maximizing data delivery rate, and reducing the clustering overhead. We address this multi-objective problem with the Non-dominated Sorted Genetic Algorithm version 2 (NSGA-II). We evaluate and compare its performance with other multi-objective optimization techniques: Multi-objective Particle Swarm Optimization (MOPSO) and Multi-objective Differential Evolution (MODE). The experiments reveal that NSGA-II improves the results of MOPSO and MODE in terms of spacing, spread, ratio of non-dominated solutions, and inverse generational distance, which are the performance metrics used for comparison.

Keywords—VANET, Cluster Protocol, Ad hoc Networks, Multi-Objective Optimization, Pareto Front, NSGA-II, MOPSO, MODE.

I. INTRODUCTION

Vehicular Ad hoc NETworks (VANETs) were deployed to make communication between a set of vehicles possible using ad hoc wireless devices. Nowadays, these networks are used for a wide range of applications which can be divided into the following three categories: safety services, traffic management and user-oriented services. Vehicle To Vehicle communication (V2V) enables each vehicle to provide a warning in real time when a critical event is predicted. The warning message can be either through a seat vibration, tone or visual display or be either through a seat vibration, tone or visual display or when a critical event is predicted. The warning message can

The rest of the paper is organized as follows. In Section
2, we present related work. Section 3 presents our clustering
protocol called AWCP. Section 4 describes the AWCP Quality
of Service (QoS) problem and identifies the AWCP parameters
and performance criteria. Section 5 describes the optimization
methodology which consists in combining an NSGA-II and
a network simulator to determine the optimal parameters
of AWCP. Section 6 shows the simulation results and the
performance evaluation. Finally, conclusions and future work
are reported in Section 7.

II. RELATED WORK

Several studies focus on developing clustering protocols
for VANET, most of which are based on Mobile Ad hoc NETwork (MANET) clustering techniques. However, none of
the protocols proposed takes highways ID into consideration
when forming clusters formation in VANETs. As a result, these
protocols do not create a stable clustering architecture. Some
of these proposed protocols are described below.

In [8], the authors propose a lane-based clustering algo-

rithm designed to extend the cluster lifetime and reduce the
communication overhead. The cluster head is selected based
on the lane where most of the vehicles will flow. The authors
suppose that each vehicle knows its exact lane on the road
via a lane detection system and an in-depth digital street map
that includes lane information. A Lane Weight (LW) metric
is applied for each traffic flow in order to select the most
stable cluster head. The clustering algorithm involves only
the cluster formation phase where all vehicles are assumed
to follow a steady roadway and does not involve a cluster
maintenance phase where the vehicles change their directions
or lanes. A Multi-Head Clustering Algorithm was proposed
in [9]. This algorithm intends to create stable clusters and re-
duce re-clustering overhead by supporting single and multiple
cluster heads. In the cluster head election phase, all vehicles
that are in communication range of each other are organized
into clusters and one vehicle for each cluster is elected to
act as a Master Cluster Head (MCH). Then, some cluster
members from a cluster are selected to be Slave Cluster Heads
(SCHs). In order to form stable clusters, the authors have
imposed that all the vehicles in a cluster are moving in the
same direction. The authors proposed in [5] a multi-metric
algorithm for cluster head elections suitable for highway area
with the aim of achieving better results for network stability
as well as decreasing the dynamic nature of VANETs. In
addition to the position and the direction, this algorithm uses
a speed difference metric as a new parameter to increase the
cluster lifetime. The vehicles that are moving at high speed
are regrouped into one cluster while the vehicles moving at
low speed are grouped into another cluster. Several other
clustering algorithms designed for MANETs are also used in
VANETs and are frequently employed for comparison with
other VANET clustering protocols. For instance, the Lowest-
ID clustering algorithm [12] is based on electing a node with
the smallest ID as a cluster head, where each node has a fixed
ID. The Weighted Clustering Algorithm (WCA) [10] elects a
node to act as a cluster head based on a combined weight
which includes its average speed, and battery-life, the number
of its neighbors and their average.

However, the behavior of these clustering protocols is
highly influenced by small changes in the set of their config-
uration parameters such as cluster_size, hello_interval, election_interval, timeout_interval, etc.). Moreover, the authors
do not provide guidelines to tune and optimize them for
various mobility scenarios. Therefore, finding the best setting
of parameters for optimally configuring these protocols is
a major issue. Several mono- and multi-objective optimization
algorithm based approaches have been proposed in the
literature for optimally configuring communication protocols
in VANETs and MANET networks. For instance, Garcia-
Nieto et al. have used different meta-heuristic algorithms to
optimize the QoS of the AODV protocol [13] and a file transfer
protocol [16] in realistic VANET scenarios. In [14] and [15],
different multi-objective optimization algorithms are proposed
to find an optimal parameter set for broadcasting methods in
MANETs. Recently, Iturriaga et al. [17] presented a novel
parallel multi-objective local search to optimize the energy
efficient broadcasting algorithm by maximizing the coverage
and minimizing the energy, the broadcasting time and the
network resources. In this study, we propose a cluster protocol
named AWCP specifically designed for VANET which takes
mobility information into account in order to provide stable
clusters with a long lifetime. In addition, due to the high
number of possible configurations, we define a multi-objective
optimization problem where we apply the non-dominated
sorted genetic algorithm NSGA-II to find the optimal values
of the parameters of the AWCP protocol.

III. THE ADAPTIVE WEIGHTED CLUSTERING PROTOCOL

AWCP is an improvement of the WCA [10] protocol which
classifies vehicles into groups based on their highway ID and
direction. Since the clusters with vehicles which are moving
in different directions and with different highway IDs are
unstable, we impose that all the vehicles in a cluster have
the same highway ID and the same direction.

A. System Model

Our algorithm is based on the assumption that each vehicle
in a VANET can know its highway ID (see Figure 1) via a
digital road map and a positioning system, e.g. GPS (Global
Positioning System) or a GALLILEO receiver that also allows
it to obtain an accurate real-time three-dimensional geographic
position (latitude, longitude and altitude), direction, speed and
exact time. In the following sections, we describe how AWCP
elects a cluster head and maintains a stable cluster structure
able to mitigate the VANET issues related to network connec-
tion and to reduce the relative mobility between neighboring
vehicles with less communication overhead.

B. Cluster Head Election

Initially, all vehicles are in the Undecided State (US). To
divide the network into clusters, each vehicle changes its state
to Cluster Head Candidate (CHC) and it starts to broadcast a
HELLO message periodically for each interval HelloInterval
containing all of the necessary information (highway ID,
direction, position, speed) to its One-Hop neighbors (OH).

In order to form stable clusters, each vehicle only considers
neighbors moving on the same highway and in the same
direction, and ignores broadcasts from other vehicles. Upon
receipt of a HELLO message from all one-hop neighbors,
each vehicle $i$ will calculate its current weight $W_i$ using (1).
We have defined the election function $W_i$ based on the function defined in [10]. The most stable vehicle that can act as a Cluster Head (CH) is a vehicle which has the minimum average distance to the other vehicles in the cluster, the closest speed to the average speed and the maximum number of neighboring vehicles.

$$W_i = w_1 \cdot D_i + w_2 \cdot |\nu_i - \Delta_i| - w_3 \cdot N_i$$  \hspace{1cm} (1)

Where $D_i = (\sum_{j \in OH_i} dist(j,i))/N_i$ is the average distance between vehicle $i$ and its one-hop neighbors ($OHi$), $\nu_i$ is the speed of vehicle $i$, $\Delta_i = (\sum_{j \in OH_i} \nu_j)/N_i$ is the average speed of the vehicles, and $N_i$ is the number of one-hop neighbors of vehicle $i$. The corresponding weight factors are such that $\sum_{i=1}^{3} w_i = 1$. Then, each node $i$ will periodically broadcast an election beacon for each interval $Election\_Interval$ containing all of the necessary information for the CH election algorithm. The election beacon for vehicle $i$ contains its: ID, CH-ID which indicates the ID of the CH to which the node is attached, Highway ID, direction, and current Weight. Node $i$ then announces itself as a CH by assigning its own ID to the ID field of the election beacon. When a vehicle $i$ receives beacons, from its one-hop neighbors, it sorts its neighbor list $OHi$ according to the weights received in the beacons, and then it executes the cluster head election algorithm to change its status from CH to Cluster Member (CM), Cluster Gateway (CG) or remain CH. Figure 2 shows an example of 1-hop clusters formation on two highways by applying our cluster protocol.

The vehicle $i$ that has the minimum value of $W_i$ is elected as the CH. Then, all vehicles that are within transmission range of the CH become CMs or CGs and are not allowed to participate in another cluster head election procedure. The CH election algorithm terminates once all the vehicles either become a CH, CM or a CG. Algorithm 1 outlines the details of the CH nodes’ election. It is executed by each vehicle $i$ having at least one neighboring vehicle. In Algorithm 1, $i, j, x$ represent three vehicles which are moving in the same highway and on the same direction and are participating in the CH election process, while $S_i$ is the current state of vehicle $i$. In addition, $Cluster\_Size$ is the size of the cluster, $ITJ\_Interval$ is the time interval for a CH vehicle to broadcast the Invite-To-Join (ITJ) message, $PRE\_Interval$ is the time interval for a CM to signal its presence to its CH, while $CH\_Timeout\_Interval$ is the time interval for a vehicle to elect itself as a CH, if it did not receive or broadcast any messages or any ITJ messages during this period.

### C. Cluster maintenance

In VANETs, a vehicle can join or leave a cluster at any time. These two operations will have only local effects on the topology of the cluster if the vehicle is a CM. However, if the vehicle is the CH, it must hand over the responsibility to one of the very close cluster members before leaving the cluster. The first reason for that is to maintain the cluster structure even if the current CH leaves. The second reason is to avoid using the re-clustering algorithm and thus no re-clustering overhead is generated when the CH leaves the cluster. Then, the current CH will order the CM to switch to CH and switch its own state to CM.

1) **Join a Cluster:** The cluster head periodically broadcasts an ITJ messages to its one-hop neighbors. Once a US or CHC vehicle receives an ITJ message, and if it wishes to join the cluster, it will check the received signal strength. The US or CHC vehicle will consider the ITJ message to be valid if its signal strength is greater than the predefined threshold denoted by $Pr\_Threshold$. When receiving a valid ITJ message, the vehicle sends a Request-To-Join (RTJ) message including the vehicle’s highway ID, position, speed and direction. When the CH receives the RTJ message, it checks the direction of the requesting vehicle and, if it is in the same highway and moving in the same direction, the CH sends an acknowledgment (ACK) including its ID number. After the reception of the ACK, the corresponding vehicle becomes a CM of this cluster. Once a US vehicle becomes a CM, it is not allowed to participate in another cluster head election procedure. If a CM receives an ITJ message from another neighboring CH moving on the same highway and in the same direction, the vehicle will switch from the CM state to the CG state.

2) **Leaving a Cluster:** A vehicle remains in the CM state as long as it receives an ITJ message from its CH every $ITJ\_Interval$. When the CM vehicle does not receive an ITJ message from its CH during $CH\_Timeout\_Interval$, it considers that it has lost contact with the CH and thus switches its state to CHC. Each CH updates a timestamp field for each CM based on the PRE-MSG messages received. The CH removes a CM from its cluster members list if the difference between the current time and the last time stamp of the PRE-MSG message received from it is greater than $CM\_Timeout\_Interval$. The CH
Algorithm 1 Cluster head election

1: \( S_i \leftarrow CHC \)
2: \( OH_i \leftarrow \emptyset \)
3: \( i \) initializes \( timer_1 \)
4: while \( timer_1! = 0 \) do
5:  Upon reception of election beacon form vehicle \( j \), vehicle \( i \) will check:
6: if \( j \) is traveling in the same highway and in the direction then
7: receive and store \( W_j \) value
8: else
9: do nothing
10: end if
11: end while
12: while \( OH_i! = 0 \) and \( S_i == CHC \) do
13: The vehicle \( i \) sorts its \( OH_i \) list
14: \( v \) ← head of \( OH_i \)
15: if \( (i == v) \) then
16: \( S_i \leftarrow CH \)
17: for every \( ITJ \) second do
18: \( CH \) broadcasts an ITJ message
19: end for
20: \( i \) initializes \( timer_2 \)
21: while \( timer_2! = 0 \) do
22: if \( i \) receives an RTJ from another vehicle \( x \) then
23: if The current number of CM vehicles < \( \text{Cluster} \) Size then
24: \( i \) will send an ACK message to \( x \)
25: \( \text{end if} \)
26: \( \text{end if} \)
27: end while
28: else
29: \( i \) sends an RTJ message to \( v \)
30: \( i \) initializes \( timer_3 \)
31: while \( timer_3! = 0 \) do
32: if \( i \) receives an ACK from \( v \) then
33: \( S_i \leftarrow CM \)
34: \( CH - ID \leftarrow v \)
35: for every \( PRE \) Interval second do
36: \( i \) Sends a presence message to its CH \( v \)
37: end for
38: else
39: if \( i \) remains in the US or in the CHC state more than \( CH \) Timeout Interval seconds then
40: \( S_i \leftarrow CH \)
41: \( \text{end if} \)
42: \( \text{end if} \)
43: end while
44: \( \text{end if} \)
45: end while

will change its state to CHC, if its list of cluster members is empty.

3) Merging two or three clusters: When two or three CHs moving on the same highway and in the same direction receive an ITJ messages from each other with a signal strength greater than the predefined threshold \( Pr \) Threshold, only one of them will keep its CH responsibility while the others will switch to a CM. The CG between clusters becomes CM of the new cluster, and each CM whose CH has become a CM will remain a CM if it receives an ITJ message from the new CH, and will switch to CHC otherwise. The selection of a cluster head for merging clusters is done based on the weight \( W_i \).

IV. AWCP PARAMETERS AND PERFORMANCE CRITERIA

The performance of AWCP depends on the selection of the parameter settings that determine its behavior. For instance, the detection of topological changes can be adjusted by changing the \( Hello \) Interval parameter. We have defined a solution vector of real variables that can be fine tuned by using an optimization technique with the aim of obtaining QoS efficient AWCP configuration. Table I shows the parameters of AWCP and their variation ranges. These parameters are four timers, four counters and three weighting factors. The variation ranges of the four timers and the first two counters are set based on the clustering protocols proposed in the literature. The \( \text{Cluster} \) Size is the maximum number of vehicles in the cluster which should be less than \((R+l)2/(w+d)\), where \( R, l, w \) and \( d \) are respectively the transmission range, the number of road lanes, the standard length of the vehicles which is about \( 3m \) and the safety distance. \( P_{\text{min}} \) is the received signal strength where the distance between two vehicles is equal to the safety distance, where \( P_{\text{max}} \) is the received signal strength where the distance between two vehicles is equal to \( 3R/4 \).

**TABLE I. AWCP PARAMETERS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Lower bound</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello Interval</td>
<td>R</td>
<td>0.5</td>
<td>15</td>
</tr>
<tr>
<td>Election Interval</td>
<td>R</td>
<td>0.5</td>
<td>15</td>
</tr>
<tr>
<td>ITJ Interval</td>
<td>R</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>PRE Interval</td>
<td>R</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>CH_TIMEOUT Interval</td>
<td>R</td>
<td>2</td>
<td>45</td>
</tr>
<tr>
<td>CM_TIMEOUT Interval</td>
<td>R</td>
<td>3</td>
<td>45</td>
</tr>
<tr>
<td>Cluster Size</td>
<td>Z</td>
<td>1</td>
<td>((R+l)2/(w+d))</td>
</tr>
<tr>
<td>Pr Threshold</td>
<td>R</td>
<td>( P_{\text{min}} )</td>
<td>( P_{\text{max}} )</td>
</tr>
<tr>
<td>Distance Weight factor ((w_1))</td>
<td>R</td>
<td>0</td>
<td>1 - ( w_2 )</td>
</tr>
<tr>
<td>Neig Weight factor ((w_3))</td>
<td>R</td>
<td>0</td>
<td>1 - ((w_1 + w_2))</td>
</tr>
</tbody>
</table>

A given AWCP configuration is evaluated based on three of the most widely used QoS metrics in this area [5]: The Average Cluster Lifetime (ACL), which is the average time period from the moment when a vehicle becomes a CH, CM or CG to the time when it changes its state. The Control Packet Overhead (CPO), which is the rate of AWCP control packets used to form and maintain the cluster structures. And finally, the Packet Delivery Ratio (PDR), which is the ratio of the number of data packets that are correctly delivered to their destinations. Figure 3 shows the values of the three optimized objectives for different AWCP configurations. From this figure, it is clear that the performance of AWCP depends on the choice of the tuning parameters. Due to the conflicting nature of the objective functions and the large size of the search space, AWCP parameter tuning is an NP-hard problem due to the huge number of possible configurations [26]. Hence, we formulated AWCP parameter tuning as a multi-objective problem and we have proposed an optimization tool which consists in combining a non-dominated sorting genetic algorithm, version
2 (NSGA-II) [11] and a network simulator ns2 to determine the optimal parameters of AWCP.

V. NSGA-II BASED APPROACH FOR AWCP OPTIMIZATION

A. Overview of NSGA-II

Optimizing a group of conflicting objective functions is not a simple task. For simplicity, we assume that all objective functions should be minimized. In fact, the multiplication of some objective functions by -1 allows one to transform a maximization to minimization. Thus, the Multi-objective Optimization Problem (MOP) can be formulated as follows:

\[
(MOP) \quad \begin{cases} 
\min f_k(\mathbf{x}), \quad k = 1, \ldots, m \\
\text{subject to} \quad \text{lower}(x_i) \leq x_i \leq \text{upper}(x_i), \quad i = 1, \ldots, n
\end{cases}
\]

The vector \( \mathbf{x} = (x_1, \ldots, x_n)^T \in S \) is the vector of \( n \) decision variables. The \( \text{lower}(x_i) \) and \( \text{upper}(x_i) \) are respectively the lower and upper bounds of the variable \( x_i \). These bounds define the decision space \( S \). Let a minimization MOP be a solution \( \mathbf{x} \in S \) which dominates the solution \( \mathbf{x} \in S \) (it is denoted by \( \mathbf{x} < \mathbf{x} \)) if the following conditions are satisfied:

i) \( f_k(\mathbf{x}) \leq f_k(\mathbf{x}) \) \( \forall k \in \{1, \ldots, m\} \)

ii) \( \exists k \in \{1, \ldots, m\} \) such that \( f_k(\mathbf{x}) < f_k(\mathbf{x}) \)

The set of optimal solutions is composed of the non-dominated vectors, often called the Pareto front and also denoted \( \text{PF}_* = \{ \mathbf{x} \in S \mid \not\exists \mathbf{x} \in X, \mathbf{x} < \mathbf{x} \} \). In other words, the Pareto front is the set of compromise solutions. The goal of the multi-objective optimization is to find the Pareto front for a given problem. The NSGA-II algorithm [11] is often used to solve the multi-objective optimization problem. This method is a multi-objective version of the genetic algorithm in which the solutions explored are classified into Pareto-optimal fronts.

B. Proposed approach

The proposed approach is based on the NSGA-optimization tool, a network simulator and the ns2-trace analyzer (see Fig. 4). These three modules cooperate to determine the optimal AWCP configuration in different mobility scenarios. Firstly, the optimization tool generates a set of possible parameters which are transmitted to the network simulator. Thereafter, the simulations are launched and the trace file is built. This file is passed on to the third module (trace analyzer) which computes the values of the fitness functions. The calculated objective values are then transmitted to the optimization tool which evaluates and ranks the solutions according to these values. Then, the optimization tool runs its operations to regenerate another set of possible solutions. This process starts again, until the stop criterion is reached. Below, we describe the NSGA-II based optimization tool.

NSGA-II begins from an initial population (P) made up of solution vectors called “individuals”. At each iteration, an auxiliary population Q is formed by applying the crossover and mutation operators (lines 7 to 13). Then, both the current (P) and the new population (Q) are merged together to form

![Fig. 3. Distribution of solutions on the objective space.](image)

![Fig. 4. NSGA-II based approach for AWCP optimization](image)
one set of solutions \( R \), which will be sorted according to the non-domination and crowded comparison (line 15). For more details, one can see [11]. Finally, only the best individuals in \( R \) can be included in the next generation and will participate in the production step while the other individuals are deleted (lines 17 to 23). These steps are repeated until the maximum number of iterations is reached.

Each individual \( i \) in iteration \( l \) is encoded as a multi-dimensional vector \( \hat{x}_{i,l} = (x_{i,1}, \ldots, x_{i,N}) \). Each gene that encodes one AWCP parameter is defined by its type (real, integer), bounds and its precision \( p \). The initial population \( P_{l=0} = \{ \hat{x}_{i,0} : i = 1, \ldots, N \} \) is generated by randomly choosing the value of each gene in its variation range \( (\text{lower}(x_i), \text{upper}(x_i)) \).

\[
\hat{x}_{j,ltr=0} = \text{lower}(x_i) + \text{rand}[0,1] \ast (\text{upper}(x_i) - \text{lower}(x_i)) \\
\text{if } i = 1, \ldots, N \text{ and } j = 1, \ldots, N
\]

Where \( N \) is the population size, \( n \) is the vector’s dimension. Thereafter, the initial population is used by the circulated genetic operators to create a new population.

The crossover operator is one of the main parts of NSGA-II. The input of this operator consists of two solution vectors (known as parents), while the output is two child vectors, which have certain features from both parents [20] (see Figure 5). Because all the genes in each solution vector of the population are within their given intervals, the resulting vector should satisfy the formulated constraints in Section V. The two most used types of crossover operators are two-point crossover and uniform crossover. In this study, we found that the NSGA-II using uniform crossover outperforms the NSGA-II using two-point crossover in terms of the obtained children quality. In uniform crossover operator, a crossover mask \( \hat{p} = (x_i)_P \in \{0, 1\}^n \) is randomly computed, which determines from which parent vector each gene will inherit. Then, each gene \( i \) will be assigned to the first parent if \( x_i = 1 \), otherwise it will be assigned to the second parent. After recombination, the mutation operator is applied to randomly change some genes in an individual. This operator serves as a strategy to prevent solutions from being trapped in local optima. After mutation, if one or more of the genes in any new individual \( j \) are outside of their ranges, the individual \( \hat{x}^j \) is repaired according to the flowing rule:

\[
(x_i^j)_{1 \leq i \leq n} = \begin{cases} 
\text{lower}(x_i) + \frac{x_i^j + \text{lower}(x_i)}{2} & \text{if } x_i^j < \text{lower}(x_i) \\
\text{lower}(x_i) + \frac{x_i^j - \text{upper}(x_i)}{2} & \text{if } x_i^j > \text{upper}(x_i) \\
x_i & \text{otherwise}
\end{cases}
\]

Since the crossover and the mutation operator generate a list of new solution vectors, a set of ns-2 simulations are launched to compute the objective values.

VI. SIMULATION RESULTS AND PERFORMANCE EVALUATION

We carried out a set of experiments to prove the ability of NSGA-II coupled with the ns2 simulator to provide optimal performances, as well as its ability to fine tune the optimal values of the AWCP parameters. The optimization tool was implemented in Java while the simulation phase was carried out by running ns-2.34. Moreover, all our experiments were conducted using 2 desktop computers Intel Core i5 3.2GHz with 4 Gb of memory and O.S. Linux Ubuntu 12.04. In order to achieve the best optimal behavior of the AWCP protocol, several experiments on various VANET scenarios were necessary. In this section, we present the set of VANET scenarios used to obtain efficient QoS AWCP parameters and the experimental validation.

A. VANET scenarios

We generated a realistic VANET environment by selecting a real highway area from a digital map which took into account road directions, road intersection, highway bridges, and traffic rules. Figure 1 shows a metropolitan area from the Map of Tunis of size \( 4 \text{ km} \times 4 \text{ km} \) exported form OpenStreetMap (OSM) and edited using Java OpenStreetMap Editor (JOSM). Then SUMO [18] and MOVE [19] were used respectively to generate vehicle traffic scenarios and to simulate the area with vehicular traffic. To generate vehicular traffic by MOVE and SUMO, we defined for each direction a vehicle flow which described a swarm of vehicles. The parameters of each vehicle flow consisted of the maximum number of vehicles, the starting road and destination of the flow, the time to start and end the flow and the probabilities of turning to different directions at each junction (0.4 to go straight, 0.3 to turn left and 0.3 to turn right). Then the traffic traces generated by MOVE were used in the ns2 simulations. All the tests were performed on different VANET scenarios taking into account different vehicle densities and data loads: Low, Medium, High and Very High. The features of the VANET scenarios and the simulation parameters used in our experiments are summarized in Tables II and III.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Number of vehicles</th>
<th>Number of CBR sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low (S1)</td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>Medium (S2)</td>
<td>50</td>
<td>15</td>
</tr>
<tr>
<td>High (S3)</td>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>Very High (S4)</td>
<td>150</td>
<td>35</td>
</tr>
</tbody>
</table>

Fig. 5. Uniform crossover operator example
We can note that the closet configuration to the ideal objective three QoS metrics, since the \( \text{min-EUDT} \) vector (solutions of size \( z_{\text{ideal}} \) which contains the optimal value for each separate objective.

\[
(z_{\text{ideal}})^i_{1\leq j \leq k} = \min f_j(\overline{x}_i) ; i = 1, \ldots, \tau
\]

Table IV shows the solutions that give the best values for each AWCP QoS metric, which are the maximum ACL (max-ACL), maximum PDR (max-PDR), and minimum CPO (min-CPO), and the average values of the \( \tau \) non-dominated solutions obtained on the Pareto front. As shown in this table, in our case the ideal vector has three values : 94.06, 91.39, 3.82. Moreover, the Euclidean distance of each solution in the non-dominated set to the ideal objective vector is calculated and the solution with the smallest Euclidean distance is selected (min-EUDT). We can note that the closet configuration to the ideal objective vector (min-EUDT) presents the best trade-off between the three QoS metrics, since the min-EUDT configuration gives the best objective values for each QoS metric. The max-ACL configurations achieve a high cluster lifetime and have a high packet delivery performance but the clusters are formed and maintained with an excessive overhead (12.68%). The configuration that optimizes the PDR metric, max-PDR, delivers an important amount of data packets. However, it decreases the performance of the AWCP protocol in terms of ACL (79.71s). The configuration that creates clusters with the least overhead min-CPO, produces a significant reduction in the performance of AWCP in terms of ACL (45.81%) and it delivers a low packet delivery ratio although it has the advantage of fewer control messages.

The min-EUDT AWCP configuration found by NSGA-II which is the most balanced setting of parameters on the Pareto front is \( \text{Hello Interval}=0.78, \text{Election Interval}=0.16, \text{ITI Interval}=7.23, \text{PRE Interval}=9.16, \text{Pr Threshold}=7.23E-16, \text{CH Timeout Interval}=12.75, \text{CM Timeout Interval}=12.7, \text{Cluster Size}=50, W_1 = 0.716, W_2 = 0.204, \text{and } W_3 = 0.07. \)

<table>
<thead>
<tr>
<th>TABLE IV.</th>
<th>NSGA-II simulation results and optimized configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>ACL</td>
</tr>
<tr>
<td>max-ACL</td>
<td>94.06 s</td>
</tr>
<tr>
<td>max-PDR</td>
<td>79.71 s</td>
</tr>
<tr>
<td>min-CPO</td>
<td>45.81 s</td>
</tr>
<tr>
<td>NSGA-II avg</td>
<td>72.75 s</td>
</tr>
<tr>
<td>min-EUDT</td>
<td>90.02 s</td>
</tr>
</tbody>
</table>

C. NSGA-II Results Validation

In this section we present the results obtained by other multi-objective optimization approaches: Multi-Objective Differential Evolution (MODE) and Multi-Objective Particle Swarm Optimization (MOPSO) which are the most recently used to optimize communication in ad hoc networks presented in [21] and [22], respectively. The parameter settings of these optimization algorithms are shown in Table V.

<table>
<thead>
<tr>
<th>TABLE V.</th>
<th>Parameter settings of the optimization algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Parameter</td>
</tr>
<tr>
<td>MOPSO</td>
<td>Local Coefficient</td>
</tr>
<tr>
<td>Social Coefficient</td>
<td>( \phi_i )</td>
</tr>
<tr>
<td>Inertia Weigh</td>
<td>w</td>
</tr>
<tr>
<td>MODE</td>
<td>Crossover Probability</td>
</tr>
<tr>
<td>Mutation Factor</td>
<td>( \mu )</td>
</tr>
</tbody>
</table>

To demonstrate the distribution of non-dominated individuals on the objective space for each Multi-Objective Evolutionary Algorithm (MOEA), we have considered the two scenarios S1 and S3 as illustrative scenarios. Figure 9 depicts the Pareto-front obtained by gathering all the non-dominated solutions found in the 30 independent runs corresponding to these scenarios. The figure 9 shows that for scenario S1, NSGA-II offers 36.24% and 36.36% more non-dominated solutions than MOPSO and MODE, respectively. For the Scenario S3, it offers 38.24% and 54.41% more non-dominated solution than MOPSO and MODE, respectively. In addition, we note from the figure 9 that MODE has significantly failed to attain a wide non-dominated set both as well as it gives a poor distribution of non-dominated points. Although MOPSO has attain a small Pareto front compared to NSGA-II, it shows its ability to find a well-diversified non-dominated solutions set.

In order to compare better the performance of different MOEAs, we evaluate the Pareto fronts (PF) obtained by the three approaches in terms of spacing, spread, generation distance, Ratio of non dominated solutions, and computational time metrics. The goal from this comparison is to demonstrate the effectiveness of NSGA-II on different VANET scenarios. Table VI presents the average (and the standard deviation) of the four metrics as well as the computational time taken by each MOEA over 15 independent runs. This table shows that the NSGA-II is significantly better than the other two MOEAs in terms of both sparsity, spacing, inverse generational distance and the ratio of non-dominated solutions. The average number of non-dominated solutions found by NSGA-II in the 15 independent runs is 80.49%, 85.16%, 83.61% and
86.56% for the S1, S2, S3 and S4 scenarios, respectively. Therefore, the NSGA-II algorithm provides a wide range of non-dominated solutions in every run, whilst MOPSO and MODE give a small number of solutions along the Pareto front. Table VI also shows that all the MOEAs take almost the same computational time. This is due to the fact that all the algorithms have the same number of fitness function evaluations. It can be seen that the Pareto fronts obtained by NSGA-II are the best regarding the spacing and spread metrics on all the test scenarios except for the S4 scenario, where MOPSO is the best in terms of the spread metric. The lowest spacing in scenario S3 is found by NSGA-II with 51.7364.81% respectively better compared to MODE and MOPSO, and the largest spread is also found by NSGA-II (38% better, on average). Thus, the Pareto front solutions obtained by NSGA-II are better distributed with respect to the MODE and MOPSO. Similarly, in terms of inverse generational distance, NSGA-II had the best performance (both in terms of average value and standard deviation). Therefore, with respect to the performance metrics used for comparison, we can conclude that NSGA-II is the most suitable for the AWCP tuning problem. Moreover, the results show that MOPSO and MODE are both the second best with respect to spread, spacing metrics and inverse generational distance, and they are clearly the worst ones in terms of the ratio of non-dominated solutions

VII. CONCLUSION

Because of the rapidly changing topology and the lack of infrastructure, it is very challenging to deploy clustering methods in vehicular networks. In this paper, we focus on designing an adaptive and optimized clustering algorithm for vehicular networks, called AWCP, that takes into consideration the highway ID, direction, position, and speed information, in order to maximize cluster stability. However, due to the high number of feasible configurations of AWCP and the conflicting nature of its performance metrics, we defined a multi-objective optimization problem where the non-dominated sorted genetic algorithm NSGA-II is coupled with the ns2 simulator to find the optimal parameter values for the AWCP QoS metrics. The NSGA-II optimized configuration is validated by comparing it with the optimized MODE and MOSPO configurations on realistic VANET scenarios taken from the metropolitan area of Tunis (Tunisia). The experimental results show that the NSGA-II algorithm obtains well-distributed solutions over the Pareto front and presents the best results in terms of performance metrics. Thus, NSGA-II algorithm is more suitable for the AWCP parameter tuning problem.

Since the computational time required to perform 15 independent runs for all MOEAs in the S4 scenario is about 32 days, a parallel version of MOEAs running on multiple processors would allow larger populations and more generations to be used in this multi-objective optimization method while reducing the computational time required for very large scale VANET scenarios. Moreover, channel efficiency in VANETs could be improved by the development of a cross-layer architecture (MAC/AWCP) in which each cluster head is responsible for assigning bandwidth to all the members of its cluster.

REFERENCES


### TABLE VI. PERFORMANCE COMPARISON OF THE THREE MOEAS FOR THE S1, S2, S3 AND S4 SCENARIOS.

<table>
<thead>
<tr>
<th>VANET Scenario</th>
<th>MOEA</th>
<th>Spacing</th>
<th>Spread</th>
<th>Generational Distance</th>
<th>Ratio of Non-dom. Indiv.</th>
<th>Avg. Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>NSGA-II</td>
<td>4.1488621</td>
<td>1.3715102</td>
<td>65.0951358</td>
<td>17.3889108</td>
<td>5.4577908</td>
</tr>
<tr>
<td></td>
<td>MOPSO</td>
<td>8.7738382</td>
<td>5.6501962</td>
<td>64.6906121</td>
<td>16.7276727</td>
<td>7.8873787</td>
</tr>
<tr>
<td></td>
<td>MODE</td>
<td>5.5034728</td>
<td>3.7037692</td>
<td>41.0385453</td>
<td>9.0122561</td>
<td>9.0198743</td>
</tr>
<tr>
<td>S2</td>
<td>NSGA-II</td>
<td>3.5536077</td>
<td>1.7412408</td>
<td>83.3691357</td>
<td>2.8631863</td>
<td>2.9250822</td>
</tr>
<tr>
<td></td>
<td>MOPSO</td>
<td>5.7282304</td>
<td>3.0834036</td>
<td>50.3591110</td>
<td>17.6703528</td>
<td>4.5397505</td>
</tr>
<tr>
<td></td>
<td>MODE</td>
<td>6.0314481</td>
<td>2.8167532</td>
<td>53.7266384</td>
<td>4.8258113</td>
<td>4.1714498</td>
</tr>
<tr>
<td>S3</td>
<td>NSGA-II</td>
<td>2.6947245</td>
<td>0.4488186</td>
<td>63.2152538</td>
<td>11.0507085</td>
<td>3.19493041</td>
</tr>
<tr>
<td></td>
<td>MOPSO</td>
<td>4.0882696</td>
<td>4.6216437</td>
<td>31.3271774</td>
<td>13.0102978</td>
<td>6.2928839</td>
</tr>
<tr>
<td></td>
<td>MODE</td>
<td>4.4441424</td>
<td>2.1440006</td>
<td>47.0680982</td>
<td>10.9674119</td>
<td>4.85431004</td>
</tr>
<tr>
<td>S4</td>
<td>NSGA-II</td>
<td>2.9067278</td>
<td>0.4680387</td>
<td>56.3595601</td>
<td>13.9051238</td>
<td>4.0720004</td>
</tr>
<tr>
<td></td>
<td>MOPSO</td>
<td>3.7250376</td>
<td>0.5563667</td>
<td>51.0215217</td>
<td>23.5176434</td>
<td>6.3424167</td>
</tr>
<tr>
<td></td>
<td>MODE</td>
<td>6.9208950</td>
<td>1.4750501</td>
<td>68.4482836</td>
<td>3.5243498</td>
<td>4.7408295</td>
</tr>
</tbody>
</table>

* A bold font indicates the best metric value.

---


