Bandwidth Efficiency and Service Adaptiveness Oriented Data Dissemination in Heterogeneous Vehicular Networks

Penglin Dai, Member, IEEE, Kai Liu, Member, IEEE, Xiao Wu, Member, IEEE, Yong Liao, Member, IEEE, Victor Chung Sing Lee, Member, IEEE, and Sang Hyuk Son, Fellow, IEEE

Abstract—Heterogeneous network resources are expected to cooperate with each other to support efficient data services in vehicular networks. However, current data scheduling methods cannot efficiently exploit the benefit of heterogeneous wireless communication interfaces in vehicular networks. In this paper, we propose a software-defined network based service architecture, which enables the scheduler to manage heterogeneous network resources in a centralized way. We analyze the heterogeneity of both data items and networks in terms of data sizes and network features (e.g., cost, transmission rate, coverage, etc.), respectively. On this basis, we formulate a data broadcast and network interface selection (DBNIS) problem, which aims to minimize both the service delay and the network access cost. To tackle the problem efficiently, we propose a coding-assisted multiobjective evolutionary algorithm (CMOEA), which consists of two components: packet encoding and network interface selection. Specifically, for packet encoding, we first develop a packet-size based random linear encoding (PRLE) technique to improve bandwidth efficiency. Then, we theoretically analyze the performance bound of PRLE. For network interface selection, we propose a multiobjective algorithm for network interface selection to adaptively satisfy dynamic requirements with respect to service delay and network access cost by deriving a set of pareto-solutions. Finally, we build the simulation model and implement CMOEA for performance evaluation. The comprehensive simulation results demonstrate the superiority of CMOEA under a wide range of scenarios.

Index Terms—Heterogeneous vehicular networks, network coding, software defined network, multi-objective optimization.

I. INTRODUCTION

Vehicular networks are envisioned as the cornerstone of future intelligent transportation systems (ITSs). Particularly, efficient data dissemination is one of the fundamental technologies to support emerging ITSs, such as autonomous intersection control [1], lane reservation [2] and collision avoidance [3]. Nevertheless, current data dissemination mechanisms developed for vehicular communications can hardly satisfy ever-increasing service demands in vehicular networks. In particular, considering the highly dynamic topology of vehicular networks and the heterogeneous wireless communication interfaces, data services in vehicular networks are likely to suffer from intermittent connection and unpredictable delay [4].

In this work, we consider the heterogeneous vehicular network, which contains different wireless communication interfaces with different features, such as communication range, data transmission rate, capacity and network access cost, etc. Meanwhile, we consider the heterogeneity of data items in terms of data size for providing different types of services. In such a context, we investigate the problem of enhancing overall system performance in terms of service delay while minimizing overall network access cost. In particular, we consider to adopt bitwise exclusive-or (XOR) coding technique to enhance the efficiency of wireless bandwidth and reduce the service delay. Further, we consider that vehicles need to pay certain cost to access a network interface, such as payment for the data flow, power consumption, virtual credit, etc. For general purposes, we abstract them as the network access cost. On the other hand, different wireless communication interfaces have different features, such as network coverage, data transmission rate, etc. Therefore, one
of the primary targets is to minimize overall cost by adaptively selecting wireless communication interfaces based on different data service requirements. To sum up, the objectives of this work are to optimize overall system performance with respect to both the service delay and the network access cost.

However, it is challenging to achieve the above objectives due to unique features of heterogeneous vehicular networks. First, improper selection of wireless interfaces may result in frequent handover between wireless communication interfaces due to high mobility of vehicles and limited capacity of network resources, which may further cause unpredictable service delay and excessive network access cost. Second, it is non-trivial for designing an efficient coding technique when considering the heterogeneity of data sizes. Particularly, there might be a dilemma between reducing service delay and enhancing broadcast bandwidth efficiency. On the one hand, encoding more data items into a packet may simultaneously serve more vehicles and thus enhance the broadcast bandwidth efficiency. On the other hand, the size of an encoded packet is determined by the maximum size of its encoded data, and hence a packet is likely to have a larger size when more data items are encoded, which may cause longer service delay.

The main contributions of this work are outlined as follows.

- We consider the service scenario where data items are with different sizes and wireless interfaces are with different attributes, such as transmission rate, network coverage, network access cost, etc. On this basis, we present a software-defined network (SDN) based system architecture, where a controller at the control plane is able to collect global knowledge of the network and make scheduling decision for the devices in the data plane, including Roadside Units (RSUs), Base Stations (BSs), vehicles, etc.
- We formulate a multi-objective optimization problem called Data Broadcast and Network Interface Selection (DBNIS), which targets at minimizing both the service delay and the network access cost.
- We propose a coding-assisted multi-objective evolutionary algorithm (CMOEA), which consists of a data encoding strategy and a network interface selection scheme. Specifically, for data scheduling, we propose a packet-size based random linear encoding (PRLE) strategy, and analyze its performance by deriving the bound of the expected completion time. Further, we develop a problem-specific multi-objective algorithm for network interface selection (MA-NIS), to minimize both the service delay and the network access cost. Finally, we design a weight-based solution selection strategy to select the solution from the set of pareto-solutions, so as to adaptively balance the service delay and the network access cost based on particular application requirements.
- We build the simulation model by implementing a traffic simulator with the real-world map and a centralized scheduling module based on the proposed service architecture. Then, we implement the proposed CMOEA and give a comprehensive performance evaluation, which demonstrates the adaptiveness and the scalability of CMOEA under a variety of application scenarios.

The rest of this paper is organized as follows. Section II reviews the related work. Section III presents the system architecture. Section IV formulates the DBNIS problem. In Section V, we propose the CMOEA algorithm. In Section VI, we build the simulation model and evaluate the algorithm performance. Finally, Section VII summarizes this work and discusses future research directions.

II. RELATED WORK

Efficient data dissemination has been widely studied in vehicular networks. Great number of studies have investigated efficient Vehicle-to-Infrastructure (V2I)/Vehicle-to-Vehicle (V2V) communications via Dedicated Short Range Communication (DSRC). N. Wisitpongphan et al. [5] considered the broadcast storm problem in vehicular ad hoc networks (VANET). They analyzed the impact of broadcast storm in terms of message delay and packet loss rate and proposed a probabilistic and timer-based broadcast suppression to reduce the contention significantly at the Media Access Control (MAC) layer. X. Wu et al. [6] proposed a dynamic transmission delay based broadcast (DAYcast) protocol to improve transmission efficiency of the network. F. Ros et al. [7] focused on the problem of data dissemination of non-safety applications in the VANET. They developed a fully distributed adaptive algorithm called Acknowledge Broadcast from Static to Highly Mobile (ABSM), which aims to achieve high reliability and minimize the total number of retransmissions. Q. Xiang et al. [8] focused on the safety data dissemination in V2V communication. They proposed a packet-value-based safety data dissemination protocol (PVCast), which makes the dissemination decision for each packet based on its packet-value and effective dissemination coverage in order to satisfy the data preferences of all the vehicles. Distinguishing from the above work, in this study, we will further investigate the heterogeneous wireless communication feature of vehicular networks and exploit the benefit of heterogeneous network resources.

Network coding is widely applied to enhance the broadcast efficiency in vehicular networks. F. Liu et al. [9] theoretically analyzed network-coding based V2V communication in a two-way road network, which are classified into encountering phase and separated phase. They derived the probability mass function of dissemination completion time in encountering phase and data dissemination velocity in separated phase. Further, the simulation results confirmed the accuracy of the proposed model. K. Liu et al. [10] put the first effort on applying network coding in cooperative V2I/V2V communication environments. They proposed a network-coding-assisted scheduling algorithm to exploit the best joint effect of V2I and V2V communications. C. Wu et al. [11] designed a routing protocol in vehicular networks by employing interflow network coding to encode packets at the common backbone vehicles for different traffic flows. The proposed protocol can reduce the number of generated packets by 25% compared with conventional routing approaches. K. Liu et al. [12] investigated data broadcast via V2I communication by exploiting the vehicular caching and network coding for enhancing bandwidth efficiency of the RSU. A problem of cache-aided
data dissemination with network coding is formulated and it is proved to be NP-hard. Further, a memetic algorithm is proposed to efficiently solve the formulated problem. The above studies focused on improving data service performance in dynamic vehicular environments, where network coding based policies are designed to address problems such as intermittent connection, broadcast collision and packet loss, etc. Distinguishing from the above work, in this study, we adopt network coding to enhance bandwidth efficiency by considering the effect of data heterogeneity, such as data size, which brings us new challenges on designing coding policies.

Many studies have considered the heterogeneity of vehicular networks and investigated efficient data service approaches by integrating different network resources. To provide more reliable V2V services, C. Hung et al. [13] proposed the Mobility Pattern Aware Routing Protocol (MPARP) for heterogeneous vehicular networks by utilizing the bandwidth of the Wireless Metropolitan Area Network (WMAN) and VANET. D. Tian et al. [14] focused on the problem of the network selection of heterogeneous network resources for mobile users. They proposed a dynamic and self-adaptive method to guarantee the Quality of Service (QoS) satisfaction of users. P. Dong et al. [15] considered the problem of energy-efficient cluster management as a p-median problem in graph theory in heterogeneous vehicular networks. They proposed a centralized method to select proper cluster head to minimize the total transmission power. R. Atat et al. [16] proposed a cooperative technique for short range collaboration to complement long range wireless transmission for public safety systems. The proposed technique achieves much better performance compared with Long Term Evolution (LTE) and 802.11p on delivery delay of critical contents. The above studies focused on improving data service performance by compensating DSRC with alternative wireless communication interfaces. However, they did not consider the effect of different network access costs and the effect of dynamic application requirements on the data scheduling.

Recently, several studies have applied the SDN concept to manage heterogeneous network resources in vehicular networks. K. Zheng et al. [17] proposed a multi-layer architecture of a soft-defined heterogeneous vehicular network, which supports the dynamic nature of various applications while reducing the operating cost. Z. He et al. [18] applied the SDN concept to the heterogeneous vehicular network. In this paradigm, heterogeneous network resources are managed with a unified abstraction and an SDN-based communication solution is proposed to minimize communication cost in both single and multiple hop cases. Q. Zheng et al. [19] proposed a service architecture, which integrates SDN and radio resource virtualization into an LTE system. They formulated the problem as a partially observed Markov decision process. K. Liu et al. [20] described the first application of SDN concept in VANETs. A novel problem of cooperative data scheduling (CDS) is formulated, which is proved to be NP-hard. A centralized scheduling algorithm is proposed to tackle the problem based upon the described SDN architecture. In order to achieve delay-optimal solution, they proposed a virtualized radio resource scheduling scheme via stochastic learning. In this study, we present the detailed implementation of an SDN-based architecture, which is particularly designed for facilitating the scheduling by considering both the heterogeneity of network and the heterogeneity of data items in vehicular networks.

III. SYSTEM MODEL

In the concerned service scenario, vehicles would like to ask for a set of common interested data items, such as safety messages, traffic data, service information, etc. Data items are with different sizes due to different applications and categories. For instance, the size of safety-critical data (e.g. collision warning messages) is typically smaller than that of many value-added service data (e.g. infotainment messages). To improve broadcast efficiency, the bitwise XOR (⊕) operation is adopted at the server side to encode multiple heterogeneous data items into one packet. Even if the vehicle cannot decode out all data items immediately after receiving these packets, they will still cache the packets for improving decoding probability in the later services. Meanwhile, we consider the co-exist of different wireless communication interfaces in the system, which have different features such as network coverage, network access cost, transmission rate, etc. We assume that the network access cost is proportional to transmission rate and network coverage. For wireless interface with higher transmission rate, the vehicle can retrieve more data items in each time slot. The time slot is defined as the minimum time interval for accessing a network interface. Further, for wireless interface with larger radio coverage, the vehicle can be served with more stable network connection. In addition, due to limited bandwidth resources, we define the network capacity of one wireless interface as the maximum number of allowed accessing devices in one time slot. In this work, we aim at minimizing both the network access cost and the service delay in such a heterogeneous vehicular environment by selecting proper network interfaces for vehicles and accordingly, scheduling encoding packets for broadcasting via each wireless interface.
Solution 1 adopts HRF (Highest-rate-first) strategy for interface selection and MRF (Most Request First) for data scheduling. Specifically, HRF always selects the network interface with the highest transmission rate. Accordingly, \( n_1 \) always has higher priority to be assigned to vehicles. Only when \( n_1 \) reaches full capacity or it is not available to the vehicles, \( n_2 \) will be scheduled. On the other hand, MRF schedules the data item with the most pending requests for broadcasting. For example, by MRF, \( d_1 \) will be scheduled to broadcast via \( n_1 \) since it has two pending requests (from \( v_1 \) and \( v_2 \)). As a result, the service delay of \( v_1 \) is 2 since \( v_1 \) retrieves all data items after receiving \( d_1 \) and the network access cost of \( v_1 \) is 10 since \( v_1 \) has been assigned to \( n_1 \) in \( t_1 \) and \( t_2 \). Likely, we can compute the service delay and network access cost of \( v_2 \) is 5/3 and 10. Then, the total service delay and network access cost is 19/3 and 26. Solution 2 adopts the HRF strategy for interface selection as the same as Solution 1 and LRC (linear random coding) for data broadcast. Therefore, the interface selection of \( v_1 \) and \( v_2 \) in solution 2 is the same to Solution 1. For data broadcast, LRC randomly encodes multiple data items into one packet to serve multiple vehicles even if they need different data items. For example, the size of packet \( d_1 \oplus d_4 \) is determined by \( d_4 \), which is independently with the cache sets of both \( v_1 \) and \( v_2 \). As a result, the total service delay and network access cost is 17/3 and 26, respectively. Solution 3 further improves system performance by exploiting the benefits of both network coding and the features of different network interfaces. Specifically, it schedules \( v_3 \) to complete request in \( t_1 \) since no network is available to \( v_3 \) in \( t_2 \). Then, it schedules \( v_1 \) and \( v_2 \) to tune in to \( n_2 \) because it can satisfy the service requests of \( v_1 \) and \( v_2 \) with less network access cost. As a result, the total service delay and network access cost of Solution 3 is 5 and 14, respectively. With above scheduling, Solution 1 achieves the highest service delay and network access cost since it cannot exploit the cache information of vehicles and always chooses the interface with highest access cost. Compared with Solution 1, Solution 2 reduces service delay by improving broadcast efficiency but also achieves highest access cost since they adopt the same strategy for interface selection. By coordinating the behaviors of interfaces and vehicles, Solution 3 achieves both the lowest service delay and the least network access cost. Based on the above analysis, it is imperative to enhance overall system performance by coordinating between data scheduling and network assignment to vehicles based on the presented SDN architecture.

**IV. PROBLEM FORMULATION**

A. Preliminary

The set of data items is denoted by \( D = \{d_1, \ldots, d_{|D|}\} \) and the size of \( d_m \) (\( d_m \in D \)) is denoted by \( s(d_m) \). The set of vehicles is denoted by \( V = \{v_1, \ldots, v_{|V|}\} \). The matrix \( C_k \) denotes the set of encoded packets cached by \( v_k \). Each row in \( C_k \) represents an encoded packet denoted by the vector \( p = \{p(1), \ldots, p(|D|)\} \), where \( p(m) \) is a binary that indicates whether \( d_m \) is encoded \( (p(m) = 1) \) or not \( (p(m) = 0) \). Each packet \( p \in C_k \) is linearly independent with the set of \( C_k \setminus \{p\} \). Then, the rank of \( C_k \), denoted by \( r(C_k) \), is the number of...
encoded packets cached by \( v_k \). The set of wireless interfaces is denoted by \( N \). For each wireless interface \( n_i \in N \), the transmission rate is denoted by \( tr(n_i) \). Further, the network access cost and the network capacity of wireless interface \( n_i \) are denoted by \( e(n_i) \) and \( b(n_i) \), respectively. The primary notations are summarized in Table I.

### B. The DBNIS Problem

In this section, we formulate the DBNIS problem in detail. First, we introduce constraints of packet encoding in each time slot. Second, we compute the network access cost paid by each vehicle during service interval. Finally, we present the formulation of DBNIS problem, which is a multi-objective optimization problem.

First, the set of encoded packets to broadcast in each time slot should be determined in advance. Let \( p_{ij}^{l} \) denote the \( l \)th encoded packet broadcast by \( n_i \) in time slot \( t_j \), where \( p_{ij}^{l}(m) \) indicates whether \( d_m \) is encoded or not. Then, the packet size of \( p_{ij}^{l} \) is determined by the encoded data item with the maximum size, which is formulated by \( s(p_{ij}^{l}) = \max_{d_m \in D} \{ s(d_m) \cdot p_{ij}^{l}(m) \} \). The time taken for broadcasting \( p_{ij}^{l} \) equals the ratio of the packet size \( s(p_{ij}^{l}) \) and the transmission rate \( tr(n_i) \), namely, \( s(p_{ij}^{l})/tr(n_i) \). Then, the total time taken for broadcasting the set of encoded packets by \( n_i \) cannot exceed the length of one time slot \( TS \), expressed as follows:

\[
\sum_{l=1}^{k} s(p_{ij}^{l})/tr(n_i) \leq TS, i=1,2,\ldots,I, j=1,2,\ldots \tag{1}
\]

If the vector \( p_{ij}^{l} \) is linearly independent with the set of vectors in \( C_k \), that is, \( r(C_k \cup p_{ij}^{l}) = r(C_k) + 1 \), then \( p_{ij}^{l} \) is useful to \( v_k \) and \( p_{ij}^{l} \) will be added to \( C_k \). Otherwise, \( v_k \) will not retrieve \( p_{ij}^{l} \). Given the network interface \( n_k(t_j) \) assigned to vehicle \( v_k \) in time slot \( t_j \), the function \( g_k(p_{ij}^{l}) \) is used to denote whether the packet \( p_{ij}^{l} \) is retrieved by vehicle \( v_k \) or not.

\[
g_k(p_{ij}^{l}) = \begin{cases} 1, & r(C_k \cup p_{ij}^{l}) > r(C_k) \land n_k(t_j) = n_i \\ 0, & \text{otherwise} \end{cases} \tag{2}
\]

For a vehicle \( v_k \), the condition of completing the service request is that it can decode out all the data items. That is, the rank of matrix \( C_k \) is full \( (r(C_k) = |D|) \). Then, given the length of

---

**TABLE I**  
**SUMMARY OF NOTATIONS**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>The service interval</td>
</tr>
<tr>
<td>( TS )</td>
<td>The length of one time slot</td>
</tr>
<tr>
<td>( t_j )</td>
<td>The ( j )th time slot</td>
</tr>
<tr>
<td>( D )</td>
<td>The set of data items ( D = { d_1, \ldots, d_{</td>
</tr>
<tr>
<td>( s(d_m) )</td>
<td>The size of data item ( d_m )</td>
</tr>
<tr>
<td>( V )</td>
<td>The set of wireless interfaces ( V = { v_1, \ldots, v_M } )</td>
</tr>
<tr>
<td>( C_k )</td>
<td>The set of encoded packets cached by vehicle ( v_k )</td>
</tr>
<tr>
<td>( s_{dk} )</td>
<td>The service delay of vehicle ( v_k )</td>
</tr>
<tr>
<td>( p_{ij}^{l} )</td>
<td>The vector ( p_{ij}^{l} = [p_{ij}^{l}(1), \ldots, p_{ij}^{l}(</td>
</tr>
<tr>
<td>( N )</td>
<td>The set of wireless interfaces ( N = { n_1, \ldots, n_N } )</td>
</tr>
<tr>
<td>( e(n_i) )</td>
<td>The network access cost of the wireless interface ( n_i )</td>
</tr>
<tr>
<td>( b(n_i) )</td>
<td>The network capacity of the wireless interface ( n_i )</td>
</tr>
<tr>
<td>( tr(n_i) )</td>
<td>The transmission rate of the wireless interface ( n_i )</td>
</tr>
<tr>
<td>( N_{k}(t_j) )</td>
<td>The set of network interfaces covering ( v_k ) in ( t_j )</td>
</tr>
<tr>
<td>( n_k(t_j) )</td>
<td>The network interface that ( v_k ) is assigned to</td>
</tr>
</tbody>
</table>

Fig. 2. An example of the scheduling problem.
time slot $TS$ and the service time interval $[0,T]$, the service delay for vehicle $v_k$ (denoted by $sd_k$) is defined as the period from the time when the service request is submitted to the time when all data items are exactly retrieved ($r(C_k) = |D|$), which is formulated as follows:

$$sd_k = \min_{\forall j \in [1,T], \forall t > 0} \left\{ (j_1 - 1) \cdot TS + \sum_{l=1}^{t} s(p_{n_k(t_j)},j) / \text{tr} (n_k (t_j)) \right\} \left[ C_k \cup \left( \frac{1}{j_1} \sum_{l=1}^{t} g_k \left( p_{n_k(t_j),j} \right) \cdot p_{n_k(t_j),j} \right) \right] = |D|.$$ (3)

Where the expression before “$|$” represents the computation of elapsed time and the expression after “$|$” represents the condition of decoding out all data items.

Accordingly, one of the primary metrics, the average service delay, is defined as follows:

**Definition IV.1.** Average Service Delay (ASD) Given the time window $T$, during which all the service requests are satisfied, and the broadcast matrix is set to $P = [p_{ij}]_{I \times T \times L}$, then the average service delay (ASD) is defined as the mean value of service delay of all service requests, which is computed as follows:

$$f_1 = \frac{\sum_{k=1}^{\left| V \right|} sd_k / \left| V \right|}{\left| V \right|}.$$ (4)

On the other hand, when previous interface is disconnected, the vehicle will be assigned to a new wireless interface to continue retrieving pending data items. Let $N_k(t)$ denote the set of interfaces that covers $v_k$ in $t$, and $n_k(t)$ denote the interface that $v_k$ is assigned to in $t$, then $n_k(t) \in N_k(t)$. $N_k(t) = \emptyset$ represents that no interface is available to $v_k$ and $n_k(t) = 0$ represents that $v_k$ is not assigned to any network interface. Further, due to the limited network resources, the total number of devices that access to $n_i$ in each time slot cannot exceed the network capacity of $n_i$, expressed as follows:

$$\sum_{k=1}^{\left| V \right|} \{ \{n_k(t_j) = n\} \leq b(n), \forall n_i \in N, j = 1, \ldots T \}$$ (5)

where $\{n_k(t_j) = n\}$ equals 1 if $n_k(t_j)$ equals $n$, otherwise, it equals zero. When $v_k$ accesses $n_k(t_j)$ in $t_j$, it has to pay $c(n_k(t_j))$ cost. In order to comprehensively evaluate the system performance in terms of network access cost, we define average network access cost as the other primary metric, which is expressed as follows:

**Definition IV.2.** Average Network Access Cost (ANAC): given time window $T$, during which all service requests have been satisfied and the interface selection matrix is set to $IN = [n_k(t_j)]_{I \times T}$, then the average network access cost (ANAC) is defined as the mean value of network access cost paid by each vehicle during $[0,T]$, which is computed as follows:

$$f_2 = \frac{\sum_{k=1}^{\left| V \right|} \sum_{j=1}^{T} c(n_k(t_j))}{\left| V \right|}.$$ (6)

With the above analysis, we derive the formulation of the DBNIS problem as follows. Let $T$ be the time window during which all the service requests are satisfied. Given a solution $x = (P, IN)$, where $P = [p_{ij}]_{I \times T \times L} \text{ and } IN = [n_k(t_j)]_{I \times T}$, then the DBNIS problem is formulated as a two-objective optimization problem, which is expressed as follows:

Minimize $F(x) = (f_1(x), f_2(x))$

Subject to $\sum_{k=1}^{\left| V \right|} \{ \{n_k(t_j) = n\} \leq b(n), \forall n_i \in N, j = 1, \ldots T \}$

$p_{ij}(m) \in \{0,1\}, m = 1, \ldots, |D|, \forall p_{ij} \in P$

$n_k(t_j) \in N_k(t_j), \forall n_k(t_j) \in IN$ (7)

V. ALGORITHM DESIGN

In this section, we propose a coding-based multi-objective evolutionary algorithm (CMOEA) to solve the DBNIS problem. The CMOEA consists of two parts: PRLE strategy for packet encoding and MA-NIS for network interface selection. In particular, we first propose a packet-size based random linear encoding strategy and theoretically analyze its performance bound. Further, we develop a multi-objective evolutionary algorithm for implementing intelligent network interface selection.

A. PRLE Algorithm

In this section, we propose a packet-size based random linear encoding (PRLE) strategy, which is used for determining the set of encoded packets in each time slot. The pseudo-code of PRLE is shown in Algorithm V.1.

The procedure of proposed packet encoding algorithm is described as follows: firstly, the rest time $ts$ is initialized to the length of time slot $TS$ and the set of available encoded data items $D'$ is set to $D$ (lines 1–2 in Algorithm V.1). Secondly, for each $d_m$, we check whether it belongs to $D'$ or not. For each $d_m$, if $s(d_m) / tr(n_i) > ts$, then $d_m$ is removed from $D'$ (lines 5–9 in Algorithm V.1). Thirdly, we construct the encoded packets by randomly selecting the data items from $D'$ (lines 10–13 in Algorithm V.1). Fourthly, we compute the size of encoded packet $s(p^f) = \max_{d_m \in D} (p^f(m) \cdot s(d_m))$ and update $ts$ as $ts = ts - s(p^f) / tr(n_i)$ (lines 14–15 in Algorithm V.1). The procedures are repeated until $ts$ is no longer greater than zero.

In the following, we prove that the expected completion time of PRLE algorithm is less than that of any non-encoding broadcast scheme. First, we analyze the expected size of encoded packet.

**Theorem V.1.** Assuming that there are $M$ data items and the distribution of data size is with the probability density function of $h(x)$, then the expected size of encoded packet using bitwise random linear network coding is computed as follows:

$$E(S(p)) = \frac{M}{2^M} \int_0^\infty \left( \int_0^k h(x)dx + 1 \right)^{M-1} h(k)dk.$$ (8)
Proof: Let $p_M$ be the encoded packet generated from $M$ data items, where $p_M = \{p_M(1), p_M(2), \ldots, p_M(M)\}$ and $S(p_M) = \max_{m \in [1, M]} \{ p_M(m) S(d_m) \}$. Then the cumulative distribution function (CDF) of $S(p_M)$ is derived as follows:

$$ G(S(p_M) \leq k) = F\left( \max_{m \in [1, M]} \{ p(m) S(d_m) \} \leq k \right) $$

$$ = \prod_{m=1}^{M} \Pr (p(m) S(d_m) \leq k) $$

$$ = \prod_{m=1}^{M} \left\{ \Pr (S(d_m) \leq k | p(m) = 1) \right\} $$

$$ + \Pr (p(m) = 0) $$

$$ = \frac{1}{2^M} \left( \int_{0}^{k} h(x) dx + 1 \right)^{M-1} h(k) kdk \quad (9) $$

The probability density function of $S(p_M)$ is derived from the derivative of $G(S(p_M))$, which is computed by

$$ \frac{d}{dx} G(S(p_M) = k) = \frac{1}{2^M} \left( \int_{0}^{k} h(x) dx + 1 \right)^{M-1} h(k) \quad (10) $$

In the following, we analyze the relationship between $E(S(p))$ and $E(S(d_m))$. The expected size of a data item is computed by $E(S(d_m)) = \int_{0}^{\infty} h(k) kdk$. As it is difficult, if not possible, to derive a general theoretical upper bound of $E(S(p)) / E(S(d_m))$, we take a special example to show the relationship between $E(S(p))$ and $E(S(d_m))$. Assuming that the distribution of data size follows the Uniform distribution, then we can derive an upper bound of the ratio $E(S(p)) / E(S(d_m))$ as follows.

**Theorem V.2** Assuming that there are $M$ data items and the size of each data item independently follows the Uniform distribution with the probability density function $h(x) = \frac{1}{s_{\max} - s_{\min}}, \forall x \in [s_{\max}, s_{\min}]$, then the expected size of each encoded packet using bitwise random linear network coding is no more than twice of the expected size of any data item, that is, $E(S(p)) \leq 2E(S(d_m)), \forall m = 1, 2, \ldots, M$.

**Proof:** According to the expectation formula, the expected size of one data item $E(S(d_m))$, $\forall m = 1, 2, \ldots, M$ is computed as $s_{\max} + s_{\min}$. Further, the expected size of one encoded packet $p$ is computed as follows:

$$ E(S(p)) = \frac{M}{2^M} \int_{0}^{\infty} \left( \int_{0}^{k} h(x) dx + 1 \right)^{M-1} h(k) kdk \quad (11) $$

Then, we can get the analytic formula of $E(S(p_M))$ as follows:

$$ E(S(p_M)) = \left( \frac{M-1}{M+1} + \frac{1}{2^M(M+1)} \right) s_{\max} $$

$$ + \left( \frac{2}{M+1} - \frac{M+2}{2^M(M+1)} \right) s_{\min} \quad (12) $$

Since $\left( \frac{M-1}{M+1} + \frac{1}{2^M(M+1)} \right) \leq 1$ and $\left( \frac{2}{M+1} - \frac{M+2}{2^M(M+1)} \right) \leq 1$, the upper bound of $E(S(p_M))$ is derived as follows:

$$ E(S(p_M)) \leq s_{\max} + s_{\min} = 2E(S(d_m)) \quad (13) $$

**Theorem V.3** Assuming that there are $M$ data items and the size of data item follows the Uniform distribution with PDF $h(x) = \frac{1}{s_{\max} - s_{\min}}, \forall x \in [s_{\max}, s_{\min}]$, further, there are $V$ vehicles requesting for the set of common $M$ data items and randomly caching $L$ data items from the database. Let $Z_1$ denote the expected size of encoded packets broadcast by PRLE algorithm and $Z_2$ denote the expected size of data items broadcast by any non-encoding broadcast scheme, then $E(Z_1) \leq E(Z_2)$.

**Proof:** First, we derive $Z_1$ as follows. As the requests of vehicles are independent with each other, for clear exhibition, we consider the situation of an individual vehicle $v_k$. Let $P_m$ denote the total size of encoded packets that have to be received when the number of elements in $C_k$ equals $m$ and $Q_m$ denote the total size of broadcast encoded packets when $C_k$ increases from $m-1$ to $m$. Then, the relationship between $P_m$ and $Q_m$ is expressed as follows:

$$ P_m = Q_m + P_{m-1} \quad (14) $$

Then, the probability that $Q_m$ equals $\sum_{k=1}^{j} S(p_k)$ is computed by:

$$ \Pr(Q_m = \sum_{k=1}^{j} S(p_k)) = \left( 1 - \frac{1}{2^m} \right)^{j-1} \left( \frac{1}{2^m} \right)^{j} \quad (15) $$
Then, the expectation of $Q_m$ is derived as follows:
\[
E(Q_m) = \lim_{j \to \infty} \sum_{j=1}^{j} \left( \sum_{k=1}^{j} E(S(p_k)) \right) \left( 1 - \frac{1}{2^m} \right)^{j-1}
\]
\[
= \frac{E(S(p))}{1 - 1/2^m} \tag{16}
\]
As $Z_1$ is equivalent to $P_M$, then the upper bound of $E(Z_1)$ is derived as follows:
\[
E(Z_1) = E \left( \sum_{m=M-L+1}^{M} (P_m - P_{m-1}) + P_{M-L} \right)
\]
\[
= E \left( \sum_{m=M-L+1}^{M} Q_m \right)
\]
\[
= E \left( \sum_{m=M-L+1}^{M} \frac{S(p)}{1 - 1/2^m} \right)
\]
\[
\leq E(S(p)) E \left( L + \sum_{m=M-L+1}^{M} 2^{-m} \right)
\]
\[
\leq ME(d_m) \tag{17}
\]
For any non-encoding broadcast scheme, the optimal number of broadcast data items is equivalent to the number of commonly requested data items by $V$ requests, which is denoted by $Z_3$. The probability that $Z_3$ is less than $M - 1$ is computed as follows:
\[
\Pr(Z_3 \leq M - 1) = \frac{(M - 1)!}{M^L (M - L - 1)!} \left( \frac{M - 1}{M} \right)^M \tag{18}
\]
Since $\Pr(Z_3 \leq M) = 1$ and $\Pr(Z_3 > M - 1) = \Pr(Z_3 = M)$, then we have
\[
\Pr(Z_3 = M) = 1 - \Pr(Z_3 \leq M - 1) \tag{19}
\]
Since with increasing value of $V$, $\Pr(Z_3 \leq M - 1)$ is approximate to zero, we can acquire that $\Pr(Z_3 = M)$ is approximate to 1. Hence, the expected size of data items broadcast by any non-encoding broadcast scheme is estimated as $E(Z_2) \approx ME(d_m)$ and we have $E(Z_1) \leq E(Z_2)$.

Given certain wireless interface, the completion time of the service request is determined by the total size of broadcast data items. Therefore, we have proved that the expected completion time of PRLE algorithm is no more than that of any non-coding broadcast scheme. To acquire data encoding decision (denoted by broadcast matrix $P$), we perform the PRLE algorithm iteratively for each wireless interface.

### B. MA-NIS Algorithm

In this section, we propose a multi-objective algorithm for network interface selection (MA-NIS), which is used for adaptive network interface selection based on dynamic requirements on ASD and ANAC by different vehicular applications. We adopt the framework of the MOEA/D [24], whose basic idea is to decompose a multi-objective optimization problem into a number of multiple scalar subproblems and then use a population-based evolutionary method to optimize these subproblems simultaneously.

The decomposition method used in this paper is the Weight Sum [34], which decomposes the muti-objective in (7) into $|W|$ scalar optimization subproblems based on a set of generated weight vectors $W = \bigcup_{k=1}^{W} \{ w^k \}$. Therefore, for each weight vector $w^k = (w^k_1, w^k_2)$, $(w^k_1 + w^k_2 = 1)$, the objective of the $k$th corresponding subproblem is formulated as follows:
\[
g(x | w^k, z) = \sum_{i=1,2} w^k_i f_i(x), \text{ subject to } x \in \Omega \tag{20}
\]
where $\Omega$ is the decision space. The best solution of each subproblem is maintained for population evolution.

The procedure of the proposed multi-objective evolutionary mechanism consists of four steps. First, we prepare for the algorithm initialization, including solution encoding, non-dominate set, neighborhood set and initial population. Second, the new population is generated by the designed genetic operators, including selection, crossover and mutation operators. Third, the population of next evolution is updated by comparing the new population with parent population. Fourth, the whole procedure repeats until the stop criterion is satisfied. Once the iteration is terminated, the algorithm chooses the best solution from the non-dominate set based on the specific requirement on $f_1(x)$ and $f_2(x)$. In order to achieve optimal solution, the critical point of designing such a multi-objective algorithm is to make the detail implementation of each component adaptive to the DBNIS problem. In the following, we will describe the detail of each component. The pseudo-code of the proposed algorithm is shown in Algorithm V.2.

1) Initialization: In this part, we will initialize four components, chromosome encoding, non-dominate set, neighborhood set and initial population. First, we design a particular encoding form adaptive to the DBNIS problem. A solution is represented by a two-tuple $x = (P, IN)$, where $P$ is the data encoding decision already generated by the PRLE algorithm (seen in Section V-A) and $IN$ is network interface selection matrix for population evolution. For example, $IN = \begin{bmatrix} 1 & 2 \\ 3 & 3 \end{bmatrix}$ represents interface selection of two vehicles in two time slots. The first row $\begin{bmatrix} 1 & 2 \end{bmatrix}$ represents the first vehicle is assigned to interface $v_1$ in $f_1$ and $v_2$ in $f_2$.

Second, the non-dominate set $EP$ is set as an empty set, which is used for storing non-dominate solutions found so far. Let $x^n$ and $x^m$ denote two solutions, $x^n$ is dominated by $x^m$ if $f_i(x^n) \leq f_i(x^m), i = 1, 2$ and there is at least one $i \in \{1, 2\}$ such that $f_i(x^n) < f_i(x^m)$. We call a solution $x$ is a non-dominated solution if no solution $x'$ exists in the solution space which can dominate $x$.

Third, the neighborhood set $NB(k)$ of each $w^k$ is generated for offspring evolution. The closeness of two weight vectors $w^n$ and $w^m$ is evaluated by the Euclidean distance function $\sqrt{\sum_{i=1,2} (w^m_i - w^n_i)^2}$. We choose the $M$
closest weight vectors to \( w^k \) as its neighborhood set \( NB(k) \). The functionality of the neighborhood set \( NB(k) \) is to provide a set of solutions to its neighborhood sub-problems for optimizing the current solution to the \( k \)th subproblem. The implementation is shown in lines 4–9 in Algorithm V.2.

Fourth, in order to generate a set of initial feasible solutions, a probabilistic method is designed to initialize the population. Let \( x^k = (P, IN^k) \) denote the solution of the \( k \)th subproblem. First, for each weight vector \( w^k, n_k(t_j) \) of each vehicle \( v_k \), in each \( t_j \) is iteratively determined by comparing \( w^k \) and a random tmp generated from \([0,1]\). If \( w^k \) is larger than tmp, the vehicle \( v_k \) is assigned to the available interface with the highest transmission rate. Otherwise, \( v_k \) is assigned with the one with the lowest network access cost. Note that the interface is considered available to \( v_k \) if the network capacity is not full and it covers \( v_k \). The detail implementation of the population initialization is shown in lines 10–27 in Algorithm V.2.

2) Offspring generation: The new offspring solutions are generated by the designed genetic operators. As the traditional genetic operators will violate the constraints of DBNIS problem, we design special operators suitable for DBNIS problem, including crossover and mutation. For each \( w^k \), the two parent solutions \( x^m = (P, IN^m) \) and \( x^n = (P, IN^n) \) are randomly selected from the set of solutions corresponding to its neighborhood set \( NB(k) \). Let \( IN_j = (n_k(t_j))_{1 \times 1} \) denote the \( j \)th column of \( IN \). Then each column \( IN_j^* \) of \( IN^* \) in the new solution \( x^* \) is randomly selected from \( IN_j^m \) or \( IN_j^n \). Further, in order to increase the diversity of search process, a mutation operator is applied to each \( n_k(t_j) \) in \( IN^* \) with a predefined mutation probability \( \rho \). If the random tmp generated from \([0,1]\) is smaller than \( \rho \), then \( n_k(t_j) \) is selected to be mutated. Then, we randomly select an interface \( n_i \) from the set of available interfaces that covers \( v_k \) and check whether the capacity of \( n_i \) is full. If the condition is true, \( v_k \) is reassigned to \( n_i \). The detail implementation of crossover and mutation operators is shown in lines 31–43 in Algorithm V.2.

3) Update: For each \( w^k \), the solutions corresponding to its neighborhood set \( NB(k) \) and the non-dominate set \( EP \) should be updated based on the new solution \( x^* \). For each \( w^l \in NB(k) \), if \( q(x^*) | w^l | < q(x^* | w^l |) \), then \( x^l \) is replaced by \( x^* \). For each solution \( x \in EP \), if \( x \) is dominated by \( x^* \), then \( x \) is removed from \( EP \). If there is no solution \( x \in EP \) which can dominate \( x^* \), the solution would be added to \( EP \). The implementation of update is shown in lines 45–53 in Algorithm V.2.

4) Termination: The stopping criterion is defined as the condition that the predefined maximum iteration number is reached, which is set based on domain-knowledge. When the population evolution is terminated, the non-dominate set \( EP \) is output. In order to adaptively fulfill the given requirement on service delay and network access cost, a weight vector is given to determine the final solution. In particular, given \( w^l = (w_1^l, w_2^l) \) and \( w_1^1 + w_2^1 = 1, w_i \geq 0, i = 1, 2 \), we evaluate the priority of a solution \( x \in EP \) as \( \Psi(x) = \sum_{i=1}^{2} f_i(x) w_i^l \). The final solution \( x^* \) is chosen with the maximum value. In practice, the weight vector can be evaluated by certain evaluation method based on statistical information of particular application scenarios.

VI. PERFORMANCE EVALUATION

A. Setup

The simulation model is built based on the system architecture presented in Section III for performance evaluation. Specifically, the traffic simulator called Simulation of Urban Mobility (SUMO) [25] is adopted to simulate vehicle mobility and generate vehicle traces. The map is downloaded from OpenStreetMap, extracted from a 3 km \( \times \) 3 km area of the Hi-tech Zone, Chengdu, China. The control module based on C programming is implemented for enabling logically centralized scheduling, information collection, resource management. Further, the CMQEA, as well as the compared algorithms, is implemented by the MATLAB. The general procedures are described as follows. First, the algorithm takes the input parameters from the control module. Then, the algorithm is executed to optimize the scheduling decision. Finally, the algorithm output the scheduling decision to the controller, including wireless interface assignment and packet encoding strategy. To give a clear view of the developed simulation model, Fig. 3 illustrates the key functions and relationships of the three modules. In the default setting, the number of data items stored in the database is set to 70, which represents the set of common data items requested by each vehicle. The data items are characterized by different data sizes, which are uniformly generated from the interval \([1, 4]\). Then, the total number of vehicles simulated in the service region is set to 200. As the vehicles may initially cache encoded packets broadcast by wireless interfaces along their trajectories. The average number of packets cached by each vehicle is set to 24. Further, we set three
Algorithm V.2: Multi-objective algorithm for network interface selection (MA-NIS).

**Input:** Neighborhood size $M$, broadcast matrix $P$, and scheduling period $SP$

**Output:** the best solution $x^*$ found under a given selected weight

1. Set non-dominate set $EP$ as $\emptyset$
2. Generate a set of weight vector $W = \{w^1, \ldots, w^{|W|}\}$
3. Compute $\text{disk}(k, l)$ of $w^k$ and $w^l$, $\forall k, l = 1, \ldots, |W|$
4. for $k = 1$ to $|W|$ do
5. while $|NB(k)| < M$ do
6. $w^* = \arg \min \{\text{dist}(k, l) | \forall w^l \in W \and NB(k)\}$
7. $NB(k) \leftarrow NB(k) \cup w^*$
8. end while
9. end for
10. for $k = 1$ to $|W|$ do
11. Set $AC = [0]_{I \times SP}$
12. for $j = 1$ to $SP$ do
13. Set $I^* = I$
14. for each $v_i \in V$ do
15. Randomly generate a number tmp from the interval $[0, 1]$
16. if $\text{tmp} \leq w^k_i$ then
17. $n^k_i(t_j) = \arg \ max_{\forall n_i \in I} \{tr(n_i) | n_i \in N_i(t_j) \cap I^*\}$
18. else
19. $n^k_i(t_j) = \arg \ min_{\forall n_i \in I} \{e(n_i) | n_i \in N_i(t_j) \cap I^*\}$
20. end if
21. $AC(n^k_i(t_j)) = AC(n^k_i(t_j)) + 1$
22. if $AC(n^k_i(t_j))$ equals $b(n^k_i(t_j))$ then
23. $I^* \leftarrow I \and \{n^k_i(t_j)\}$
24. end if
25. end for
26. end for
27. $x^k = (P, IN^k)$
28. end for
29. while iteration < max_iter do
30. for $k = 1$ to $|W|$ do
31. Randomly select $w^m$ and $w^n$ from $NB(k)$
32. Randomly determine $IN^*_j$ as either $IN^*_j$ or $IN^*_j$, $j = 1, \ldots, SP$
33. for $j = 1$ to $SP$ do
34. for $l = 1$ to $V$ do
35. Randomly generate tmp from $[0, 1]$
36. Randomly select $n_i \in N_i(t_j)$
37. if $\text{tmp} \leq \rho$ and $AC(i, j) < b(n_i)$ then
38. $AC(i, j) = AC(i, j) + 1$
39. $AC(n^*_j(t_j), j) = AC(n^*_j(t_j), j) - 1$
40. $n^*_j(t_j) = n_i$
41. end if
42. end for
43. end for
44. $x^* = (P, IN^*)$

Algorithm V.2: Continued.

45. for $l = 1$ to $M$ do
46. if $\Phi(x^k | w^l) \leq \Phi(x^k | w^l)$ then
47. $x^k = x^*$
48. end if
49. end for
50. $EP = EP \setminus \{x | x \text{ is dominated by } x^*\}$
51. if no $x$ dominate $x^*$ then
52. $EP \leftarrow EP \cup \{x^*\}$
53. end if
54. end for
55. iteration = iteration + 1
56. end while
57. Compute $\Psi(x) = \sum_{i=1}^{2} f_i(x) w_i^*, \forall x \in EP$
58. $x^* = \arg \ min_{\forall x \in EP} \{\Psi(x)\}$

Types of network interfaces in the simulation: $n_1$, $n_2$, and $n_3$. The transmission rates of $n_1$, $n_2$ and $n_3$ are set as 10, 6, 4 unit of data size per time slot. Further, the network coverage of an interface is evaluated by the probability that the interface covers a vehicle, which are set to 0.9, 0.6 and 0.3 for $n_1$, $n_2$ and $n_3$, respectively. Further, according to the network setting, $n_1$ has the highest transmission rate and the largest network coverage, and hence it is set with the highest network access cost (i.e., 3). Further, $n_3$ has the lowest transmission rate and the smallest network coverage, and hence it is set with the lowest network access cost. The network performance of $n_2$ is between $n_1$ and $n_3$, therefore, its network access cost is set to 1. Further, the network capacities of three networks are all set to 250. Further, the length of time slot is set to 1 s, as it is sufficient for network interface switch in second order [26].

For algorithm implementation, the population size of CMOEA is set to 500, which is sufficient for maintaining the diversity of solutions in population evolution. Then, the maximum iteration number of CMOEA are set to 100, which takes enough times to search the pareto-optimal solutions for non-dominated set. The mutation probability is set to 0.01, which indicates that CMOEA will explore the search space by mutating each solution with 0.01 probability. Further, the selection weight vector is set to (0.55, 0.45), which assigns the weights of 0.55 and 0.45 on the ASD and the ANAC, respectively. In the present study, our parameter configurations are commonly used setting in the literatures, which can refer to [27]–[29]. We implement two network interface selection strategies and one commonly adopted data broadcast algorithms for performance evaluation. The two network interface selection strategies are Highest-Rate-First (HRF) and Lowest-Cost-First (LCF), which are priori to select network with the highest transmission rate and the lowest network access cost, respectively. The data broadcast mechanism is MRF [30], which broadcasts the data item with the most pending requests. Accordingly, we implement two comparison solutions MRF+HRF and MRF+LCF for performance evaluation. Two of the most important system metrics have been defined in (4) and (6), which are ASD and ANAC,
respectively. In addition, in order to evaluate the broadcast efficiency, we design another metric called broadcast productivity (BP), which is computed as follows. Let $b_k$ denote the minimum number of data items that vehicle $v_k$ has to receive for serving requests and $b_k'$ is the actual number of data items that vehicle $v_k$ has received. Then, the broadcast productivity is computed by \( \frac{\sum_{k=1}^{V} b_k / b_k'}{|V|} \).

B. Simulation Results

1) Effect of Traffic Workload: Figs. 4 and 5 show the ASD and the ANAC of the three algorithms under different traffic workloads. There are more service requests with an increasing number of vehicles. First, as shown in Figs. 4 and 5, when the vehicle number increases, the ASD and the ANAC of both HRF and LCF increases gradually. In contrast, CMOEA achieves much better performance on minimizing ASD and ANAC. The main reason is that the bandwidth efficiency of CMOEA is the highest among these three algorithms. As shown in Fig. 6, the broadcast productivity is close to 1, which is much higher than that of HRF and LCF. Therefore, more bandwidth resources are needed to complete the service, which causes the increasing of the ASD and the ANAC. Second, it is noted that as shown in Fig. 4, HRF performs better than LCF in terms of minimizing ASD. On the other hand, as shown in Fig. 5, LCF has lower ANAC than HRF. This is because HRF and LCF prefer to assign the interface with higher transmission rate and less network access cost, respectively. Therefore, for HRF, the average percentage of vehicles which accesses interface $n_1$ is 92.3%, which is much higher than that of LCF (25.9%). Accordingly, compared with LCF, HRF can achieve less ASD by acquiring higher network performance, but come with the cost of paying higher network access cost, which leads to higher ANAC. The above results further verify that the two objectives (i.e., minimizing both ASD and ANAC) are in conflict with each other, and it is nontrivial for CMOEA to optimize the overall system performance.

2) Effect of Cached Packet Number: Figs. 7 and 8 show the ASD and the ANAC of the three algorithms under different number of cached packets. With the increasing number of cached packets, both the ASD and ANAC of all algorithms decrease gradually. Therefore, vehicles can be satisfied without more bandwidth resources, so that it can achieve the lowest ASD and ANAC under various traffic workloads. On the other hand, with the increasing of vehicle number, the BP of both HRF and LCF reduces by about 6% and 16%, respectively. It indicates that more than 6% and 16% of broadcast data items by HRF and LCF are not helpful for satisfying extra requests when the vehicle number increases from 150 to 350. Therefore, more bandwidth resources are needed to complete the service, which causes the increasing of the ASD and the ANAC. On the other hand, as shown in Fig. 5, LCF has lower ANAC than HRF. This is because HRF and LCF prefer to assign the interface with higher transmission rate and less network access cost, respectively. Therefore, for HRF, the average percentage of vehicles which accesses interface $n_1$ is 92.3%, which is much higher than that of LCF (25.9%). Accordingly, compared with LCF, HRF can achieve less ASD by acquiring higher network performance, but come with the cost of paying higher network access cost, which leads to higher ANAC. The above results further verify that the two objectives (i.e., minimizing both ASD and ANAC) are in conflict with each other, and it is nontrivial for CMOEA to optimize the overall system performance.

Fig. 4. Average service delay under different traffic workloads.

Fig. 5. Average network access cost under different traffic workloads.

Fig. 6. Broadcast productivity under different traffic workloads.

Fig. 7. Average service delay under different number of cached packets.

Fig. 8. Average network access cost under different number of cached packets.
outperforms other two algorithms on both ASD and ANAC in all the scenarios. Particular, the performance gap between CMOEA and other two algorithms becomes larger when there are more cache packets. It can be explained as follows. First, as shown in Fig. 9, the BP of HRF and LCF decreases, which indicates that the broadcast data item of HRF and LCF can serve fewer vehicles as these data item have already cached by vehicles. Second, the BP of CMOEA is near to the optimal even when more data items are cached by vehicles, which indicates that CMOEA can schedule effectively by well exploiting the cached packets of vehicles. The above results show the effectiveness of CMOEA under different cache packet numbers.

3) Effect of the Service Workload: Figs. 10 and 11 show the ASD and the ANAC of the three algorithms under different number of requested data items. With an increasing service workload, both the ASD and the ANAC of three algorithms increase gradually. It is because that the larger number of requested data items indicate that more data items are demanded by vehicles. More broadcast data items and more bandwidth resources are required by vehicles for retrieving all requested data items, which causes longer service delay and higher network access cost. Particularly, as shown in Fig. 12, the BP of two comparative algorithms increases not so significant with the increasing number of requested data items, which makes little contributions to reducing the ASD and the ANAC. On the other hand, CMOEA can generate effective packets more easily since more data items are unretrieved by vehicles. Therefore, CMOEA maintains the best performance and outperforms other two algorithms on both the ASD and the ANAC in all service scenarios, which shows the scalability of CMOEA against service workload.

4) Effect of the Weight Vectors: In this part, we evaluate the adaptiveness of CMOEA under different weight vectors $w_k = (w_{k1}, w_{k2}), k = 1, 2, \ldots$. Specifically, $w_{k1}$ ranges from 0.35 to 0.75 with interval 0.05, which indicates that the assigned
weight for ASD is getting higher. Fig. 13 shows the ASD and the ANAC of CMOEA under different numbers of requested data items. Each black rectangular dot represents the result under a specific weight vector. According to Fig. 13, we have the following observations. First, when $w_k^i$ increases, the preference of selecting network interface with high transmission rate increases. Therefore, the ASD of CMOEA decreases and the ANAC of CMOEA increases. Clearly, assigning higher weight on one metric will adversely impact the performance of the other one. Second, the variation of two metrics at the borders is not so obvious because at beginning, the changing of weight on one part is too small to affect the scheduling preference on network interface selection. Third, both the points of LCF and HRF are above the curve of CMOEA in all cases in Fig. 13, which indicates that we can always choose a point in the curve which has both shorter ASD and lower ANAC compared with other algorithms. It is because the designed multi-objective evolutionary mechanism can effectively search the Pareto-optimal front in the solution spaces. The above analysis demonstrates that CMOEA is able to provide adaptive solutions to strike the balance between service delay and network access cost based on given application requirements.

VII. SUMMARY AND FUTURE WORK

In this paper, we present an SDN-based architecture for data services in heterogeneous vehicular networks. The SDN controller periodically collects the information of all devices, including vehicles, RSUs and BSs, and then it makes the scheduling decisions based on the global knowledge. Further, we consider the network heterogeneity and the data heterogeneity when formulating the problem of DBNIS, which aims to minimize both the ASD and the ANAC. Further, to improve the broadcast efficiency, we develop the PRLE algorithm to encode multiple packets with heterogeneous data sizes and give a specific example to show that the expected completion time of PRLE is no more than that of any non-encoding broadcast scheme. Then, to adaptively satisfy the dynamic requirement on the ASD and the ANAC, we design the CMOEA algorithm by combing PRLE with a particularly designed multi-objective evolutionary mechanism, including multi-objective decomposition, solution encoding, population initialization and genetic operators. By adjusting the weight vectors, CMOEA can adaptively select the solution from the non-dominate set to strike the best balance between the ASD and the ANAC. Lastly, we build the simulation model and give comprehensive performance evaluation. The simulation results demonstrate the adaptiveness and scalability of CMOEA in a wide range of scenarios.

In the future work, the multi-hop V2I/V2V communication will be investigated to further enhance system scalability. In addition, we will validate the system performance in a more realistic environment by incorporating real trajectories of vehicles and considering lower layer features of wireless and mobile communications.

REFERENCES

Kai Liu (S’07–M’12) received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2011. From December 2010 to May 2011, he was a Visiting Scholar with the Department of Computer Science, University of Virginia, Charlottesville, VA, USA. From 2011 to 2014, he was a Postdoctoral Fellow with Singapore Nanyang Technological University, Singapore. He is currently an Assistant Professor with the College of Computer Science, Chongqing University, Chongqing, China. His research interests include mobile computing, vehicular networks, and intelligent transportation systems.

Penglin Dai (S’15–M’17) received the B.S. degree in mathematics and applied mathematics and the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2012 and 2017, respectively. He is currently an Assistant Professor with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. His research interests include intelligent transportation systems and vehicular cyber-physical systems.

Xiao Wu (S’05–M’07) received the B.Eng. and M.S. degrees in computer science from Yunnan University, Yunnan, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2008. He is currently a Professor with the Southwest Jiaotong University, Chengdu, China. He is also the Assistant Dean of School of Information Science and Technology and the Head of Department of Computer Science and Technology, Southwest Jiaotong University. He is currently an Assistant Professor with the Department of Software, Chinese Academy of Sciences, Beijing, China. From 2003 to 2004 and 2007 to 2009, he was a Research Assistant and a Senior Research Associate with the City University of Hong Kong, respectively. From 2006 to 2007, he was a Visiting Scholar with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, and a Visiting Associate Professor with the School of Information and Computer Science, University of California, Irvine, Irvine, CA, USA, from 2015 to 2016. His research interests include multimedia information retrieval, image/video computing, and data mining. He was the recipient of the second prize of the Natural Science Award of the Ministry of Education, China, in 2016.

Yong Liao (M’14) received the Ph.D. degree from Chongqing University, Chongqing, China, in 2014. He is currently a Research Associate and the Deputy Director with the Center of Communication and TT&C, Chongqing University, Chongqing, China. He has authored more than 60 journal and conference papers. His current research interests include high-speed mobile communication, 5G and future communication, and aerocraft TT&C and communication.

Victor Chung Sing Lee (M’92) received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 1997. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests include vehicular networks, real-time databases, and performance evaluation. He is a member of the ACM and the IEEE Computer Society. He has been the Chairman of the IEEE, Hong Kong Section, Computer Chapter in 2006–2007.

Sang Hyuk Son (M’85–SM’98–F’13) received the B.S. degree in electronics engineering from Seoul National University, Seoul, South Korea, the M.S. degree from KAIST, Daejeon, South Korea, and the Ph.D. degree in computer science from the University of Maryland, College Park, College Park, MD, USA. He is the President of Daegu Gyeongbuk Institute of Science and Technology, Daegu, South Korea. He has been a Professor of Computer Science Department with the University of Virginia, and WCU Chair Professor with Sogang University. He has been a Visiting Professor with KAIST, City University of Hong Kong, Ecole Centrale de Lille in France, and Linkoping University and University of Skovde in Sweden. His research interests include cyber physical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He has authored or coauthored more than 340 papers and edited/authored 4 books in these areas. His research has been funded by the Korean Government, National Research Foundation, National Science Foundation, DARPA, Office of Naval Research, Department of Energy, National Security Agency, and IBM. He is a member of the Korean Academy of Science and Technology and the National Academy of Engineering of Korea. He has served on the editorial board of the ACM Transactions on Cyber Physical Systems, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and Real-Time Systems Journal. He is a founding member of the ACM/IEEE CPS Week, and serving as a member of the steering committee for the IEEE RTCSA and Cyber Physical Systems Week. He was the recipient of the Outstanding Contribution Award from the Cyber Physical Systems Week in 2012.