Deep Deterministic Policy Gradient to Minimize the Age of Information in Cellular V2X Communications

Zoubeir Mlika, Member, IEEE, and Soumaya Cherkaoui, Senior Member, IEEE

Abstract—This paper studies the problem of minimizing the age of information (AoI) in cellular vehicle-to-everything communications. To provide minimal AoI and high reliability for vehicles’ safety information, non-orthogonal multiple access is exploited. We reformulate a resource allocation problem that involves half-duplex transceiver selection, broadcast coverage optimization, power allocation, and resource block (RB) scheduling. First, to obtain the optimal solution, we formulate the problem as a mixed-integer nonlinear programming problem and then study its NP-hardness. The negative result of NP-hardness motivates us to design efficient sub-optimal solutions. Consequently, we model the problem as a single-agent Markov decision process (MDP). The MDP model helps in solving the problem efficiently using fingerprint deep reinforcement learning (DRL) techniques such as deep-Q-network (DQN) methods. Nevertheless, applying DQN is not straightforward due to the curse of dimensionality implied by the large and mixed action space that contains discrete RB scheduling decisions and continuous power and coverage optimization decisions. Therefore, to solve this mixed discrete/continuous problem efficiently simply and elegantly, we propose a decomposition technique that consists of first solving the discrete subproblem using a matching algorithm based on state-of-the-art stable roommate matching and then solving the continuous subproblem using DRL algorithm that is based on deep deterministic policy gradient (DDPG). We validate our proposed method through Monte Carlo simulations where we show that the decomposed matching and DRL algorithm successfully minimizes the AoI and achieves almost 66% performance gain compared to the best benchmarks for various vehicles’ speeds, transmission power, or packet sizes. Further, we prove the existence of an optimal value of broadcast coverage at which the learning algorithm provides the optimal AoI.

Index Terms—Cellular vehicle-to-everything, 5G cellular vehicle-to-everything, non-orthogonal multiple access, age of information, resource allocation, deep reinforcement learning, deep deterministic policy gradient.

I. INTRODUCTION

THE vehicle-to-everything (V2X) ecosystem is one of the key verticals in fifth-generation (5G) networks [1]. It enables vehicles to communicate with each other, with road infrastructure, and with other road users. It is an instrumental enabler for smart cities and intelligent transportation systems (ITS) [2]. Future generations of cellular networks (5G and beyond) are expected to meet the connectivity requirements of the automotive industry, including the V2X ecosystem, by providing the necessary wireless technology.

Many use cases for the automotive vertical can be supported, as outlined in the third generation partnership project (3GPP) Release 15 (3GPP Rel-15), including fleet management and infotainment [3]. In Rel-16, other advanced services are investigated by new radio (NR) that are grouped into four categories: extended sensors, vehicle platooning, advanced driving, and remote driving [4].

We can broadly organize the different use cases into three groups of services: (i) safety services, (ii) non-safety services, and (iii) infotainment services. The main use case of V2X communication belongs to the first group of services, which aims to improve road safety through efficient and timely information exchange between vehicles, with road users, and/or with road infrastructure, to avoid road accidents. Important use cases of the safety services include vehicle detection, vehicle platooning, remote driving, to name a few.

Safety services often require ultra-reliable, low latency communication (URLLC) and timely information exchange. For example, for a platoon of vehicles with a minimum distance of 2 meters between vehicles, the end-to-end latency requirements should be about 25 milliseconds to support a speed of 100 km/h. In contrast, for a remote driving application, the end-to-end latency should not exceed 5 milliseconds [1]. In addition to the delay or latency metrics, the age of information (AoI) [5] plays a key role in V2X communication [6] to ensure timely information exchange. In a V2X ecosystem, to provide safety services, it is necessary that the information exchanged between vehicles is fresh. The freshness of information can be measured by the AoI that plays a paramount importance in vehicular networks.

AoI captures delay from an application layer perspective and is defined as the time between the generation of information and its successful delivery [5]. AoI includes not only delay but also inter-delivery time, which is a more important and vital performance metric for safety services in vehicular networks. The inter-delivery time is indeed taken into account by AoI since the latter keeps increasing as long as the information has not been updated.

The objective of this paper is to develop algorithmic solutions that successfully minimize the average AoI.
V2X communication can support a direct transmission mode (LTE) have been considered. In 3GPP Rel-16, NR cellular V2X communication can support a direct transmission mode called sidelink [1], [7], and a transmission mode over the cellular network, which together provides seamless connectivity to vehicles through a unified radio.

Resource allocation for V2X side-link communication can be performed in Mode 1 or Mode 2 [1]. The first mode is centralized where the roadside units (RSU) schedule the resources for V2X side-link communication, while the second mode is distributed where vehicles autonomously allocate network-configured resources for V2X side-link communication. Thanks to the promising solution of the cellular-based V2X network that provides high cellular coverage and very low latency even in high mobility scenarios [8], we adopt, in this paper, the centralized resource allocation approach. Despite all the benefits that the cellular-based V2X network can provide, optimizing AoI to support safety services in a V2X ecosystem is a challenging problem from an algorithmic perspective.

Due to the broadcast nature of vehicular communications, it is difficult to guarantee a minimum AoI for security packets and the solution to this problem has not been well answered in the literature, especially from optimization and algorithmic point of view. Minimizing AoI is even more complex when the orthogonal multiple access (OMA) technique is used because OMA prevents multiple vehicles from broadcasting over the same resource due to interference [8]. Therefore, for better AoI and better spectral efficiency, the power-domain non-OMA (NOMA) technique [9] is proposed as a potential solution for beyond 5G networks [8]. Thus, multiple vehicles (a NOMA group), can broadcast their safety information simultaneously over the same resource [10]. To decode the information of each vehicle, various multi-user detection algorithms have been proposed, including the well-known successive interference cancellation (SIC). Despite the difficulty of performing decoding in a large NOMA group (due to large processing times), the powerful computation capabilities of advanced integrated circuit chips allow the complexity of SIC to be reduced and the delays incurred to be disregarded [8].

The use of NOMA in vehicular networks to optimize AoI remains challenging due to (i) broadcasting and (ii) half-duplex modes of operation. Due to the half-duplex, when multiple vehicles broadcast their information in the same resource block (RB), then no vehicle is capable of receiving the correct information from within its coverage area. Further, broadcasting under the NOMA technique brings even more severe interference to other vehicles, pedestrians, and infrastructure due to limited spectrum resources. To solve the above two issues, wise RBs scheduling and power allocation (to correctly decode the received information in a non-orthogonal manner, the transmission power of each transmitter vehicle should be carefully allocated) and careful broadcast coverage range optimization of each vehicle are required. All of the above decisions should be optimized while guaranteeing up-to-date information transmission to minimize the AoI of the vehicles. Therefore, it is necessary to rethink the state-of-the-art resource allocation and propose new algorithmic techniques to optimize the vehicles’ coverage and resources to minimize the AoI in a NOMA-based vehicular network while ensuring different constraints such as i) a half-duplex vehicle does not transmit and receive safety information over the same RB, ii) a maximum transmission power. Note that the optimization of vehicle coverage is not well studied in the literature, although it is argued that grouping vehicles into clusters is particularly important for reducing interference and thus minimizes AoI [11], [12]. Therefore, the considered problem statement is the minimization of AoI using the NOMA technique in half-duplex vehicular networks while deciding the coverage areas of vehicles, scheduling their safety information, and allocating the transmission power of each NOMA group.

A. Related Works

In [8], Di et al. considered the resource allocation problem in V2X communication networks using NOMA. Specifically, the problem is formulated using mathematical programming to maximize the number of successfully decoded signals, which involved three decision-making: (i) transceiver selection in a half-duplex scenario, (ii) frequency allocation, and (iii) power allocation. The authors first have shown the NP-hardness of the problem by reducing the vertex cover problem to a special case of their problem, then, using matching theory, they proposed centralized semi-persistent scheduling (SPS)-based approach, as in Mode 1, to handle the first two decision making and designed a rotation-based matching algorithm using the multidimensional stable roommate problem. Next, they proposed an iterative power control scheme based on a distributed implementation. Our work can be seen as an extension of this work but B. Di et al. did not optimize AoI and their complexity analysis does not apply to our case.

In [13], Sodhro et al. proposed a machine learning-driven mobility management method for industrial network in box (NIB) applications to intelligently allocate the resources with high energy efficiency. The authors also proposed a novel architecture of 6G-based intelligent quality-of-experience (QoE) and quality-of-service (QoS) optimization in industrial NIB. A 6G NIB framework in association with LTE networks is further studied and an energy-efficient use case for 6G industrial NIB is presented. The authors used artificial intelligence and mathematical optimization to derive their framework. The authors showed, using real-time datasets, MATLAB and CVX-OPT, that the proposed mobility management method provides high energy efficiency and better QoE and QoS in 6G-based industrial NIB.

In [14], Sodhro et al. proposed a joint cross-layer approach for the development of green, sustainable, reliable, and smart 5G-based ITS. The cross-layer based reliability optimization approach combines four layers, including physical layer (energy efficiency), medium access control layer (battery-
aware), network layer (reliability), and application layer (signaling and transmission), by adopting modulation level monitoring, node’s duty cycle, routing/processing, and particular application selection. Energy and reliability optimization problems were formulated and real-time datasets were used in MATLAB to evaluate the performance.

In [15], Sodhro et al. studied the problem of reliable channel modeling in fog computing vehicular networks and they proposed reliable and delay-tolerant channel model with better QoS and they optimized QoS in terms of mobility, reliability, and packet loss. For fog computing intra-vehicle networks, the authors proposed, based on artificial intelligence, a reliable and interference-free mobility management algorithm to provide efficient communication, computation, cooperation, and storage. Next, for fog computing inter-vehicle networks, they proposed a reliable and efficient multi-layer framework. Real-time datasets and the convex optimization framework of MATLAB, as well as the ANOVA platform, were used to evaluate the proposed solution. Further, the authors discussed some challenges and possible solutions for inter city vehicular communications including mobility management, interference mitigation, power control, and routing.

In [16], Sodhro et al. studied the problem of maximizing the reliability and connectivity and minimizing the packet loss in software-defined Internet of vehicles networks. The authors first proposed, based on the key features of the wireless channel, a system model for reliability and connectivity optimization in software-defined Internet of vehicles. Then, a stable and scalable link optimization algorithm, that adopts the key characteristics of the reliability and connectivity optimization in vehicular networks, is proposed and compared to baseline methods. Next, a novel vehicular connectivity, and reliability framework is proposed that comprises different steps including vehicular connectivity to the Internet, collecting and monitoring the vehicle’s data, and connecting the vehicles for reliable, and connectivity optimized communication. Further, a smart city use case was studied.

Despite the interesting works of Sodhro et al. in [13]–[16], the authors did not optimize the AoI and thus the formulated problem in our work is mainly different.

In [12], Abdel-Aziz et al. studied the problem of power minimization in vehicle-to-vehicle (V2V) communication networks by considering the tail distribution of AoI. The authors used extreme value theory and queueing theory to study extreme AoI events that go beyond the average AoI analysis. It is assumed that the vehicles are grouped by an RSU into clusters based on their geographic locations. The problem is formulated as a time-averaged power minimization with probabilistic AoI constraints in both the deterministic and Markovian arrival cases and is solved using Lyapunov stochastic optimization. The authors formulated their problem differently from ours and did not focus on optimizing the half-duplex transceiver selection, coverage optimization, and resource allocation. A Manhattan mobility model was adopted in the Monte Carlo simulations.

In [17], Chen et al. considered a Manhattan grid V2V network and formulated a single-agent Markov decision process (MDP) that aims to improve the expected long-term performance of all vehicles. An RSU is assumed to make decisions, as in the Mode 1 case discussed earlier, for vehicles by allocating frequency bands and scheduling packets over time. It is assumed that the RSU can choose the groups of vehicles based on geographic locations. Due to the high mobility and variation of vehicle traffic, the resource allocation decision space can grow exponentially. Therefore, the authors proposed an MDP decomposition technique using the Q-function decomposition. Using long short-term memory from the recurrent neural network and deep reinforcement learning (DRL), a proactive AoI-aware algorithm is proposed to optimize the AoI in a decentralized manner. However, the constraints of the studied problem are different from our constraints such as the half-duplex transceiver selection, and coverage optimization. Evaluations were conducted using the Python programming language with TensorFlow framework where the algorithms were trained in an offline manner.

In [18], Samir et al. studied AoI minimization in a network of unmanned aerial vehicles (UAVs). The problem formulated is in the form of a mixed-integer non-linear program that aims to optimize the UAVs’ trajectories as well as scheduling decisions while guaranteeing minimum throughput constraints. By modeling the problem as an MDP, the authors proposed a DRL approach based on the deep deterministic policy gradient (DDPG) algorithm. A complexity analysis of the proposed solution was also given. However, the paper did not consider the coverage optimization and the half-duplex transmission (transceiver selection) of the vehicles. The authors used Monte Carlo simulations and trained their proposed solutions using tensor processing units.

In [19], Peng et al. studied the AoI performance of SPS in a V2X communication network. Due to packet collisions and half-duplex effects of the SPS-based MAC layer protocol, the AoI is shown to deteriorate. The authors proposed a collaboratory approach based on piggyback mechanisms in which vehicles act cooperatively to inform each other on potential collisions. However, the paper only considers the minimization of AoI from a MAC layer perspective, and no algorithmic solution was proposed. Theoretical performance analysis in the scenarios of both static and dynamic vehicular traffic flows was presented and its convergence was studied. Monte Carlo simulations were performed in a freeway scenario.

In [20], Liang et al. studied the problem of spectrum sharing and power allocation in vehicular networks. The objective is formulated, using MDP, to maximize the total link throughput between the vehicle and the infrastructure while guaranteeing the URLLC requirements of V2V links. The authors proposed, using proper reward design, a multi-agent MDP that is solved in a distributed manner using fingerprint-based DQN methods. The proposed solution was simulated using Monte Carlo methods and was shown to allow the vehicles to learn from interactions with the vehicular environment and figure out a clever cooperative strategy in a distributed way. However, the paper did not optimize the AoI.

The application of DQN techniques to solve the resource allocation problem in wireless networks with applications to vehicular networks is discussed in [21]. However, fingerprint-
receiver selection, coverage optimization, and wireless resource allocation. We solve the joint problem of half-duplex transceiver selection, power allocation, and RBs scheduling. NOMA is used to wisely use the limited ecosystem to minimize the average AoI while applying the continuous control DRL algorithm finds the discrete actions by assigning the RBs to vehicles using state-of-the-art stable roommate matching techniques. Next, the continuous control DRL algorithm uses the deep deterministic policy gradient (DDPG) to find the transmission power levels and broadcast coverage ranges.

We show through extensive simulations that the decomposition-based matching and DRL algorithm successfully minimizes the AoI and achieves better AoI compared to benchmarks.

We identify important research gaps that were not solved by previous works. In particular, previous work (i) has optimized only access latency; however, we believe that optimizing the AoI metric is more important in vehicular networks due to the need to provide up-to-date information for safety-critical applications. Or, (ii) the works have focused on the MAC perspective of AoI in vehicular networks, but we believe that considering AoI optimization from an algorithmic perspective is more relevant to solving the intrinsic difficulty of the problem. Or, (iii) the works did not optimize the coverage along with the half-duplex transceiver selection to minimize the AoI.

We show in Fig. 1 the position of our work in comparison to related works that include resource allocation based on deep reinforcement learning (DRL) methods that optimize jointly the coverage optimization, the half-duplex transceiver selection, power allocation, and RBs scheduling.

In this paper, we fill these research gaps by studying a V2X ecosystem to minimize the average AoI while applying the NOMA technique. NOMA is used to wisely use the limited wireless resources by different vehicles in a non-orthogonal manner. We solve the joint problem of half-duplex transceiver selection, coverage optimization, and wireless resource allocation (i.e., time slots scheduling and power allocation) according to the following methodology. Specifically, the joint problem involves complex mixed decisions that include discrete decisions (half-duplex selection and resource block allocation) and continuous decisions (coverage optimization and NOMA power allocation). To solve this challenging problem efficiently, the discrete-based decisions problem is modeled as a matching problem according to [8] and solved using state-of-the-art methods (though [8] did not optimize AoI). However, the continuous-based decision problem, which cannot be discretized, is modeled as a Markov decision process (MDP) and solved using deep reinforcement learning (DRL) methods based on deep deterministic policy gradient (DDPG).

To obtain an optimal solution to this joint problem using off-the-shelf solvers (for small-scale networks at least due to the high computational complexity of solver-based solutions), it is formulated as a mixed-integer non-linear program (MINLP). Next, we provide a rigorous NP-hardness analysis of the problem. Note that, in general, a particular problem is MINLP does not necessarily imply its NP-hardness. The most directly related work to ours is [8] in which the authors did not consider AoI optimization, only proposed simple heuristic and greedy methods, and studied the NP-hardness of the problem only in the case of frequency allocation, i.e., when the system involved single frequency channel, it is not obvious to generalize the NP-hardness proof presented in [8].

We rigorously prove its NP-hardness.

We model the AoI minimization problem as a single-agent Markov decision process.

We solve the AoI minimization problem using a decomposition technique that first solves a matching subproblem and then solves an MDP problem. The matching algorithm finds the discrete actions by assigning the RBs to vehicles using state-of-the-art stable roommate matching techniques. Next, the continuous control DRL algorithm uses the deep deterministic policy gradient (DDPG) to find the transmission power levels and broadcast coverage ranges.

We show through extensive simulations that the decomposition-based matching and DRL algorithm successfully minimizes the AoI and achieves better AoI compared to benchmarks.

B. Notations

The transmission link between $i$ and $j$ is denoted in subscript as $i \rightarrow j$. We use boldface letters to denote multidimensional vectors, i.e., $\mathbf{x} := [x_{i \rightarrow j}]$ is a 3D matrix with indices $i$, $j$, and $t$, and $\mathbf{p} := [p_{i}^{(t)}]$ is a 2D matrix of indices $i$ and $s$. Calligraphic letters denote sets, e.g., the set of the first $n$ positive integers is denoted by $A = \{1, 2, \ldots, n\}$. The calligraphic letter with parentheses $\mathcal{O}(\cdot)$ denotes the big-O notation. In general, the superscript denotes time index and is given in parentheses (e.g., $x^{(s,t)}$ denotes the variable $x$ at time instant $s$ of the period $t$) to remove any confusion about exponentiation. The binary logarithm of a variable $x$ is denoted by $\lg(x)$. Other notations are defined when used.

Fig. 1. A flow chart of the position of our work in comparison to the related works. Our work is represented with solid arrows whereas other related works are represented with dashed arrows.
II. MODEL

A. System Model

We consider a cellular-based V2X communication network in which there are \( m \) half-duplex vehicles denoted by the set \( \mathcal{V} \) and a road-side unit (RSU). Time is discrete and is divided into time slots of duration \( \tau \) seconds each (\( \tau \) is on the order of a few milliseconds). Each set of \( k \) consecutive time slots is denoted by \( \mathcal{S} \) and forms what we call a transmission period. We consider a time window of \( n \) transmission periods denoted by the set \( \mathcal{T} \). In this work, for simplicity, we consider a single carrier system model in which a time slot simply represents a resource block (RB). The multi-carrier system model is a simple extension that does not affect our theoretical analysis of the problem. Each RB has a bandwidth of \( \omega \) Hertz.

At the beginning of each transmission period \( t \), each vehicle \( i \) generates its safety information, or simply, a safety packet of size \( \zeta_i^{(t)} \) bits, and must transmit it to all other vehicles using broadcasting.\(^2\) The vehicles communicate with each other via side-link communication where the RSU acts as a central controller that makes decisions for the vehicles. Multiple vehicles can transmit on the same RB using NOMA. Indeed, the use of NOMA improves spectral efficiency and ensures low latency and high reliability for vehicular applications [8]. However, due to half-duplex communication and broadcasting, a vehicle cannot transmit and receive on the same RB.

In addition, the transmission power must be carefully allocated to achieve the promised gains of NOMA. In summary, once each vehicle \( i \in \mathcal{V} \) has generated its safety information, it must make four decisions: (i) choose its role as transmitter or receiver, (ii) choose a set of receiving vehicles by choosing its coverage range (in the case of a transmitter), (iii) assign a set of time slots during which it will transmit, and (iv) allocate, for each chosen time slot \( s \), a transmission power \( p_i^{(s)} \) that must not exceed a maximum transmission power \( \mathcal{T}_i \). In the considered transmission scenario, all decisions are made by the RSU in a centralized way as it is done in [8], [17]. An example of our system model is given in Fig. 2.

B. Mathematical Model

The main objective is to improve the freshness of safety information in vehicular networks. Information freshness is measured using the well-known AoI metric. When vehicle \( i \in \mathcal{V} \) transmits to vehicle \( j \in \mathcal{V} \), the AoI of link \((i,j)\) up to the transmission period \( t \) is denoted by \( \Delta_{i\rightarrow j}^{(t)} \). We use the arrow notation here to denote the transmission direction between vehicles. Based on our discrete-time model and the periodic nature of safety information in vehicular networks, the AoI \( \Delta_{i\rightarrow j}^{(t)} \) denotes the number of transmission periods that have elapsed up to the transmission period \( t \) since the last successful transmission between \( i \) and \( j \). The smaller the \( \Delta_{i\rightarrow j}^{(t)} \), the better the freshness of the information. Note that the use of discrete transmission periods as units of AoI is considered in [22], [23].

To mathematically define \( \Delta_{i\rightarrow j}^{(t)} \), we introduce the binary variable \( x_{i\rightarrow j}^{(t)} \) to denote the successful transmission between vehicles \( i \) and \( j \). In other words, \( x_{i\rightarrow j}^{(t)} = 1 \) if and only if vehicle \( j \) successfully receives the safety information generated by vehicle \( i \) during the transmission period \( t \). That said, the AoI of the link \((i,j)\) can be recursively calculated as follows:

\[
\Delta_{i\rightarrow j}^{(t)} := \begin{cases} 
1, & \text{if } x_{i\rightarrow j}^{(t-1)} = 1, \\
1 + \Delta_{i\rightarrow j}^{(t-1)}, & \text{otherwise}.
\end{cases}
\]

That is, the AoI of the safety packet \( \zeta_i^{(t)} \) of the link \((i,j)\) has the following two properties: (i) it increases by one as long as it is not successfully received by \( j \) in the previous transmission period, and (ii) it goes back to one when it is successfully received by \( j \) in the previous transmission period. Thus, the AoI, \( \Delta_{i\rightarrow j}^{(t)} \), can be simply given by the formula:

\[
\Delta_{i\rightarrow j}^{(t)} = 1 + (1 - x_{i\rightarrow j}^{(t-1)}) \Delta_{i\rightarrow j}^{(t-1)}.
\]

Here we see that when \( x_{i\rightarrow j}^{(t-1)} = 1 \), meaning that the information was received in \( t - 1 \) (and thus it is up-to-date), then the AoI is equal to 1 (the minimum value). However, when \( x_{i\rightarrow j}^{(t-1)} = 0 \), meaning that the information was not received previously (and thus it is outdated), then the AoI increases by 1.

When vehicle \( i \in \mathcal{V} \) communicates with vehicle \( j \in \mathcal{V} \) at time slot \( s \) during the transmission period \( t \), the signal-to-interference-plus-noise ratio (SINR) is given by:

\[
y_{i\rightarrow j}^{(s,t)} := \frac{g_i^{(s)} h_{i\rightarrow j}^{(s,t)}}{1 + I_{i\rightarrow j}^{(s,t)}}.
\]

In (3), \( g_i^{(s)} \) denotes the normalized power gain of the channel, i.e., \( g_i^{(s)} = |h_i^{(s,t)}|^2 / N_0 \) where \( h_i^{(s,t)} \) is the channel coefficient which includes large and small scale fading and \( N_0 \) is the noise power over the considered bandwidth \( \omega \). The interference term in the denominator, \( I_{i\rightarrow j}^{(s,t)} \), is calculated according to the NOMA principle in which multiple vehicles (including \( i \)) transmit to vehicle \( j \). Specifically, the interference \( I_{i\rightarrow j}^{(s,t)} \) at the \( j \)th receiving vehicle comes from the transmitting vehicles (different from \( i \)) that are scheduled at time slot \( s \) of the transmission period \( t \) and transmit to vehicle \( j \). Vehicle \( j \) then applies SIC to decode the received information. It is well-known that the highest channel decoding order should be applied in this uplink-like NOMA scenario [24]. In other words, the interference \( I_{i\rightarrow j}^{(s,t)} \) received by vehicle \( j \) comes from all vehicles other than \( i \) that have lower channel power gains.

Let \( \mathcal{I}_{i\rightarrow j}^{(t)} \) be the set of vehicles other than \( i \) that transmit to vehicle \( j \) at time slot \( s \) of the transmission period \( t \) such that \( g_{i'\rightarrow j}^{(s,t)} > g_{i\rightarrow j}^{(s,t)} \) for all \( i' \in \mathcal{I}_{i\rightarrow j}^{(t)} \), i.e., \( \mathcal{I}_{i\rightarrow j}^{(t)} = \{i' \in \mathcal{V} \setminus \{i\} : g_{i'\rightarrow j}^{(s,t)} > g_{i\rightarrow j}^{(s,t)}\} \). Therefore, the interference term \( I_{i\rightarrow j}^{(s,t)} \) is given mathematically as follows:

\[
I_{i\rightarrow j}^{(s,t)} := \sum_{i' \in \mathcal{I}_{i\rightarrow j}^{(t)}} p_{i'}^{(s)} g_{i'\rightarrow j}^{(s,t)} x_{i'\rightarrow j}^{(t)}.
\]

For example, in Fig. 2 the set \( \mathcal{I}_{X\rightarrow j}^{(s,t)} \) for a given pair \((s,t)\) is given by vehicle \( W \) where \( X \) denotes the red vehicle to which \( V \) and \( W \) are transmitting to.
Based on the SINR, the achievable rate (based on Shannon theorem) on link \((i, j)\) at time slot \(s\) of transmission period \(t\) can be obtained mathematically as follows:

\[
R_{i \rightarrow j}^{(s,t)} = w t \log(1 + y_{i \rightarrow j}^{(s,t)}).
\]

The resource allocation problem considered is called the vehicular age minimization problem (VAMP). Its objective is to minimize the average AoI of all vehicles while associating vehicles with each other, scheduling transmission to appropriate RBs, and allocating transmission powers. To solve VAMP, first, we formulate it as a mixed-integer nonlinear program (MINLP). This formulation is useful for obtaining the optimal solution using off-the-shelf solvers. Then, we study initialization of variables prior to the first transmission period. Finally, we propose a single-agent MDP formulation that provides an online suboptimal solution using DRL techniques.

VAMP can be formulated as the following MINLP:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in V} \sum_{j \in V} \sum_{t \in T} \Delta_{i \rightarrow j}^{(t)}, \\
\text{subject to} & \quad y_{i}^{(s,t)}, x_{i \rightarrow j}^{(t)} \in [0, 1], \forall i, j, s, t, \\
& \quad \Delta_{i \rightarrow j}^{(0)} = 1, x_{i \rightarrow j}^{(0)} = 1, y_{i}^{(0)} = 1, \forall i, j, \\
& \quad \Delta_{i \rightarrow j}^{(t)} = 1, y_{i}^{(t)} = 1, \forall i, j, t, \\
& \quad \Delta_{i \rightarrow j}^{(t)} = 1 + (1 - x_{i \rightarrow j}^{(t-1)}) \Delta_{i \rightarrow j}^{(t-1)}, \forall i, j, t, \\
& \quad \Delta_{i \rightarrow i}^{(t)} = 0, x_{i \rightarrow i}^{(t)} = 0, \forall i, t, \\
& \quad \sum_{s \in S} y_{i}^{(s,t)} R_{i \rightarrow j}^{(s,t)} \geq y_{i}^{(t)} x_{i \rightarrow j}^{(t)}, \forall i, j, t, \\
& \quad x_{i \rightarrow j}^{(t)} \leq \sum_{s \in S} y_{i}^{(s,t)}, \forall i, j, s, t, \\
& \quad y_{i}^{(s,t)} \leq \sum_{j \in V} x_{i \rightarrow j}^{(t)}, \forall i, j, s, t, \\
& \quad y_{i}^{(s,t)} x_{i \rightarrow j}^{(t)} + y_{j}^{(s,t)} x_{j \rightarrow i}^{(t)} \leq 1, \forall i, j, j', s, t, \\
\end{align*}
\]

In (6), the objective function given in (6a) calculates the overall AoI of each pair of vehicles \((i, j)\) according to (2). Constraints (6b)-(6f) illustrate the optimization variables, their initialization, and the evolution of the AoI. Here, \(\Delta_{i \rightarrow j}^{(t)}\) represents the AoI of link \((i, j)\) up to transmission period \(t\) and is an integer variable whose lower bound is 0 and upper bound is \(n + 1\) (see (1)). The variable \(y_{i}^{(s,t)}\) is binary that is equal to 1 if and only if vehicle \(i\) is scheduled to transmit at time slot \(s\) of transmission period \(t\). The variable \(x_{i \rightarrow j}^{(t)} = 1\) if and only if vehicle \(j\) successfully receives the safety information generated by vehicle \(i\) during the transmission period \(t\). The initialization of variables prior to the first transmission period is given in constraints (6d). The evolution of AoI is given by constraints (6e). Constraints (6f) indicate that vehicle \(i\) cannot transmit to itself and therefore must have an AoI equal to 0 and \(x_{i \rightarrow j}^{(t)} = 0\).

Constraints (6g) guarantee that if vehicle \(i\) transmits to vehicle \(j\) during transmission period \(t\), then the total achievable throughput on the allocated RBs is at least the corresponding packet size. The relationship between the variables \(x_{i \rightarrow j}^{(t)}\) and \(y_{i}^{(s,t)}\) is given in constraints (6h) and (6i). That is, (i) if vehicle \(i\) successfully delivers its safety packet to vehicle \(j\) during transmission period \(t\), then there must exist at least one time slot \(s\) at which \(i\) has been scheduled, and (ii) if vehicle \(i\) has never successfully delivered its safety packet to any vehicle during transmission period \(t\), then it must not be scheduled at any time slot \(s\). Due to the NOMA technique and half-duplex communication, constraints (6j) ensure that vehicle \(i\) cannot transmit to the vehicle \(j\) and receive from vehicle \(j'\) (including \(j\)) in the same time slot. Similarly, constraints (6k)
guarantee that vehicle \( j \) cannot receive from vehicle \( i \) and transmit to vehicle \( i' \) (including \( i \)) in the same time slot. Constraints (6i) limit the number of vehicles scheduled at time slot \( s \) to \( m \) to reduce the decoding complexity in SIC. Finally, constraints (6m) ensure that the transmission power of vehicle \( i \) is nonnegative and does not exceed its maximum value.

Clearly, VAMP as formulated in (6) is nonlinear. Note that, although constraints (6e), (6j), and (6k) can be easily linearized by introducing auxiliary binary variables, constraints (6g) cannot be linearized due to the logarithm in (5) as well as the expression of SINR and interference (3) and (4), respectively. Therefore, it is very challenging to solve VAMP optimally in polynomial-time. This conclusion is general and is often found in the literature. To justify this conclusion rigorously, we provide formal proof that VAMP is indeed NP-hard even for a restricted version. As mentioned earlier, a similar problem was studied in [8] that considered a single transmission period and optimized a different objective function. The authors studied the NP-hardness of the problem and showed by a restriction that the formulated problem is NP-hard. However, their proof is only valid for multi-carrier vehicular networks because they showed that the graph coloring problem is reducible to the subcarrier allocation problem. Therefore, the question of whether the single carri er problem as formulated in (6) is NP-hard or not remains open.

C. NP-Hardness

In this subsection, we consider a restricted version of VAMP in which there are only two transmission periods and the OMA technique is used (i.e., the maximum number of vehicles that can be grouped together is \( m = 1 \)). The problem in the NOMA case is more general and therefore more difficult than this restricted case. For this restricted case, to minimize the AoI, one must maximize the number of successfully received safety packets in the first transmission period. Indeed, as shown in (1), the AoI up to the first transmission period is constant but the AoI up to the second transmission period depends on the number of packets successfully received in the first transmission period. The restricted version of VAMP will be called the VAMP*. We prove the following result:

**Lemma 1:** VAMP* is NP-hard.

**Proof:** Once we show that VAMP* is NP-hard, then, by restriction [25], VAMP is NP-hard too. We reduce, in polynomial-time, the maximum independent set (MIS) problem [25] to VAMP*.

Given a graph with \( \ell \) vertices as an instance of MIS, we construct, in polynomial-time, an instance of VAMP such that MIS is solved if and only if VAMP* is solved. The vertices of the graph correspond to the vehicles while its edges correspond to the time slots in the second transmission period. An additional vehicle \( i^* \) is created. Thus, we create a total of \( \ell + 1 \) vehicles denoted by \( \{1, 2, \ldots, \ell\} \cup \{i^*\} \). We consider a simple star network scenario (an uplink-like scenario) in which all vehicles in \( \{1, 2, \ldots, \ell\} \) wish to transmit to \( i^* \), i.e., we can simply choose the channel gain between the vehicles in \( \{1, 2, \ldots, \ell\} \) to be very small and the transmission power of vehicle \( i^* \) to be zero. Unless otherwise specified, the other network parameters (e.g., transmission power, noise power, etc.) are normalized to one. It remains to construct the channel gains between vehicle \( i \) in \( \{1, 2, \ldots, \ell\} \) and vehicle \( i^* \). The channel coefficient of each link \((i, i^*)\) at time slot \( s \) is set to 1 if \( s \) is incident to vertex \( i \) and is set to 0 otherwise. In this way, the achievable rate between vehicle \( i \) and \( i^* \) at a time slot is either 1 or 0. The size of the safety packet, \( \zeta_i \), of vehicle \( i \) is chosen to be equal to the degree of vertex \( i \). Therefore, for vehicle \( i \) to satisfy its safety packet requirements, it must be scheduled at all time slots that are incident to it in the corresponding graph. This construction is clearly performed in polynomial-time.

On the one hand, assume that MIS is solved with its given instance. That is, assume that an independent set \( I \) of maximum cardinality is found. Then, we can schedule each vehicle \( i \in I \) at the time slot incident to it. In doing so, we obtain a maximum number of successfully received packets since the cardinality of \( I \) is maximal and the size of the corresponding safety packet is equal to the degree of each vertex. Further, since \( I \) is an independent set, each time slot is used by at most one vehicle. Thus, VAMP* is solved.

On the other hand, assume that VAMP* is solved with its created instance. That is a solution to VAMP* in which a maximum number of successfully received safety packets is found. Then, since each scheduled vehicle respects the size of its safety packet, it must therefore be scheduled at all time slots incident to it. This set of scheduled vehicles forms an independent set since each time slot is used by at most one vehicle. Moreover, this independent set is of maximum cardinality since the number of successfully received safety packets is maximum.

Putting all this together, we showed that MIS reduces to VAMP* in polynomial-time. This shows that VAMP* is NP-hard and thereby proves the lemma.

In the sequel, we present the proposed solution to solve VAMP. First, we start by presenting a modified (adapted to specifically solve VAMP) version of the benchmark algorithm proposed in [8] to find the RBs scheduling. Next, we present our proposed DRL algorithm based on the DDPG method to find the transmission power and the association of the vehicle.

Due to the broadcast nature of vehicular networks as well as for practical implementation reasons, the vehicle association optimization as formulated in (6) is transformed into vehicle coverage optimization. In other words, vehicle \( i \) communicates with vehicle \( j \) at time slot \( s \) if and only if the distance between the two vehicles at time slot \( s \), \( d_{i-j}^{(s)} \), is at most \( r_i \), i.e., \( d_{i-j}^{(s)} \leq r_i \). Here, \( r_i \) is an optimization variable associated with vehicle \( i \) and it denotes its communication range, i.e., vehicle \( i \) can transmit to all vehicles \( j \in \mathcal{N}_i \), where \( \mathcal{N}_i \) is the set of neighbours of vehicle \( i \) defined as follows:

\[
\mathcal{N}_i := \{ j \in \mathcal{V} : d_{i-j}^{(s)} \leq r_i \} \setminus \{i\}. \tag{7}
\]

For example, in Fig. 2 the set \( \mathcal{N}_W \) is given by the two red vehicles in the broadcasting range of \( W \).

III. THE DEEP REINFORCEMENT LEARNING SOLUTION

As discussed earlier, VAMP is NP-hard and is MINLP. Machine learning, in particular DRL, has received a lot of
attention for solving NP-hard problems in wireless and vehicular networks. The flexibility in the design of the reward function and the action and state spaces is what makes DRL approaches attractive.

To effectively solve VAMP, we propose a DRL-based approach using MDP. Note that the MDP formulation could be helpful in exploring dynamic programming methods in future. Due also to the structure of the recursive relation found in the AoI function, exploring dynamic programming is appealing. The design of dynamic programming algorithms will be left for our future work and we only focus here on designing DRL solutions. First, we model VAMP as a single-agent MDP in which the RSU, as a DRL agent, interacts with the vehicular environment and makes decisions accordingly. The MDP is defined mathematically by the tuple $\mathcal{M} = (S, A, T, R)$, where $S$ represents the set of states, $A$ represents the set of actions, $T$ represents the transition function, and $R$ represents the reward function. In the following, we explicitly describe each element of MDP $\mathcal{M}$.

$S$: a state $s(t)$ corresponds to a transmission period $t$. The DRL agent, once in state $s(t)$, learns the status of the underlying vehicles at transmission period $t$. This includes vehicles’ positions, directions, speeds, communication ranges, and transmission powers, the channel gains between vehicles, the current AoI of each pair of vehicles, and the current transmission period $t$. The terminal state $s(n)$ corresponds to the final transmission period $n$ including the status of the underlying vehicles.

$A$: an action is given by the matrix $A = [a_1, a_2, \ldots, a_m]$ of size $m \times 3$ where $a_i = [a_{i1}, a_{i2}, a_{i3}]$ for $i \in \{1, 2, \ldots, m\}$. The action $a_i$ of vehicle $i \in \{1, 2, \ldots, m\}$ is a row vector in which the variable $a_{i1}$ belongs to the real interval $[0, \bar{r}]$ that represents the communication coverage selected by the DRL agent for vehicle $i$, where $\bar{r}$ is an upper-bound for the communication range of each vehicle. Here, $a_{i1} = 0$ means that the corresponding vehicle is selected to be a receiving vehicle in the corresponding transmission period. The variable $a_{i2}$ belongs to the set $\{0, 1, 2, \ldots, k\}$ and represents the time slots selected by the DRL agent for vehicle $i$. Finally, the variable $a_{i3}$ belongs to the real interval $[0, \bar{v}]$ which represents the power levels among which the DRL agent would select the transmission power for vehicle $i$.

$T$: the transition from one state to another is deterministic. In other words, once the DRL agent is in state $s(t)$ and its chosen action is $A(t)$, it transitions to the next state $s(t+1)$ (to the next transmission period) with probability 1 and observes the updates corresponding to the next state.

$R$: the reward is what the DRL agent seeks to maximize throughout its interaction with the environment. A straightforward approach to designing the reward is to choose the objective function in (6a) since we seek to minimize the average AoI. Nevertheless, this reward function is not Markovian because the reward in a certain transmission period does not depend on the immediate state of the environment and on the chosen action. Rather, it depends on the history of the actions that the DRL agent has taken, i.e., the AoI between a pair of vehicles $(i, j)$, at transmission period $t$, clearly depends on whether or not the two vehicles have successfully delivered their safety packets in previous transmission periods. To cope with such subtlety, the reward function needs to be rethought. It is shown in [26] that maximizing the number of successfully delivered packets in each transmission period would help in minimizing the average AoI. Therefore, the reward function is chosen in proportion to the number of successfully delivered safety packets in each period:

$$ R(s(t), A(t)) := r(t) = \frac{\eta(t)}{m^2}, \quad (8) $$

where $\eta(t) := 2 \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{V}} x_{i\rightarrow j}^{(t)} - m^2$ and the denominator $m^2$ is only used for normalization.

The rationale behind the choice of this reward function is that when all vehicles broadcast to each other (i.e., $x_{i\rightarrow j}^{(t)} = 1$ for all $i$ and $j$), which is the preferred outcome, the reward will take on the maximum possible value which is 1. If the DRL agent maximizes the reward function in (8), then it will maximize the number of successfully delivered safety packets. According to to [26], the DRL agent will therefore be able to reduce the average AoI across transmission periods.

The previously defined action space of MDP $\mathcal{M}$ consists of a mixture of discrete and continuous actions. Discrete action spaces make the learning procedure difficult due to the exponentially large number of possible actions. Exploring such a discrete action space will therefore be difficult for the DRL agent. In the case of VAMP, the number of discrete actions depends on the correspondence between the time slots $\{0, 1, 2, \ldots, k\}$ and the vehicles $\{1, 2, \ldots, m\}$, which can be up to $O(k^m)$. To overcome this issue, we employ a continuous control strategy for DRL, called deep deterministic policy gradient (DDPG) [18], [27].

Before describing the DDPG method in detail, we show how the time slots are matched to the vehicles and thus solving the discrete decisions making problem. To do so, we follow a benchmark algorithm proposed in [8].

### A. Matching Vehicles to Time Slots

For clarity, we describe the original algorithm proposed in [8] and include all necessary modifications and details. The main idea of the algorithm is to maximize the number of successfully received packets in each transmission period by applying a two-sided matching technique. We call this algorithm greedy matching (GM). If a vehicle is matched to a time slot, it is considered as a transmitting vehicle in that time slot, and otherwise, it is considered as a receiving vehicle in that time slot.

A two-sided matching $\Psi$ is defined mathematically as a mapping from $\mathcal{V} \cup \mathcal{S} \cup \mathcal{B}$ to $\mathcal{V} \cup \mathcal{S} \cup \mathcal{B}$, where $\Psi(i) \subseteq \mathcal{S}$, for $i \in \mathcal{V}$ and $\Psi(s) \subseteq \mathcal{V}$, for $s \in \mathcal{S}$. The vehicles that are matched to the same time slot are called matching peers, which refer to an extension of the notion of roommates in the original stable roommate matching problem [28]. A pair of vehicles $(i, i')$ is called a forbidden pair if the distance between the two
vehicles at time slot $s$ is upper-bounded by $r$, i.e., $d_{i \rightarrow i'}^{(s)} \leq r$. GM uses the notion of preference relation which we define in the following.

**B. Preference Relation**

Each vehicle establishes a preference relationship for all other vehicles. It must choose its corresponding matching peers in such a way that it is positioned away from them so that their communication ranges do not overlap. This reduces potential collision, which is defined in [8] as the cross-influence brought by multiple transmitting vehicles on the conflicting receiving vehicles. The cross-influence brought by any two (distinct) transmitting vehicles $i$ and $i'$ in time slot $s$ on the corresponding receiving vehicles is given as [8]:

$$c_{i \rightarrow i'}^{(s)} := \begin{cases} (2r - d_{i \rightarrow i'}^{(s)})^2, & \text{if } 2r > d_{i \rightarrow i'}^{(s)}, \\ e, & \text{otherwise,} \end{cases}$$

(9)

where $-0.1 < e < 0$ is a sufficiently small constant. If a time slot is matched to only one vehicle, then the cross-influence exists and is equal to $e$. Therefore, the average cross-influence brought by vehicle $i$ when matched to time slot $s$ is given by [8]:

$$Q_i^{(s)} := \frac{1}{1 + |\Psi(s)|} \sum_{i' \in \Psi(s)} c_{i \rightarrow i'}^{(s)}.$$  

(10)

We can see from (10) that, if $i$ is the only vehicle matched to time slot $s$, then the average cross-influence is equal to $e$. More simply, the cross-influence is a way to measure how far two vehicles are. If they are far-away apart then the interference will be small and vice versa.

Based on the definition of the cross-influence, vehicle $i$ can construct its preference relationship. Specifically, vehicle $i$ prefers time slot $s$ to time slot $s'$ if (i) its average cross-influence that it brought at time slot $s$ is less than the one that it brought at time slot $s'$, i.e., $Q_i^{(s)} < Q_i^{(s')}$, (ii) there is no forbidden pair at time slot $s$, and (iii) time slot $s$ does not match more than $\overline{m}$ vehicles. Accordingly, the vehicles can define their feasible preference relations. GM goes through two phases: In the first phase, it finds a feasible solution, and in the second phase, it rotates the matching peers to further reduce their average cross-influence while guaranteeing feasibility.

**C. Phase 1: Finding a Feasible Solution**

A greedy approach is used in GM to find an initial solution [8]. Let $\Psi$ represent the current matching. If a not-yet-matched vehicle $i$ can form a forbidden pair with vehicle $i' \in \Psi(s)$, then vehicle $i$ cannot be matched to time slot $s$. We associate to each vehicle $i$ a set $M_i$ of available matched time slots (the set of time slots that do not form forbidden pairs with $i$), defined as follows:

$$M_i := \{ s \in S_{\text{matched}} : \Psi(s) \cap F_i^{(s)} = \emptyset \},$$

(11)

where $S_{\text{matched}}$ defines the set of time slots already-matched in the current matching $\Psi$ and $F_i^{(s)}$ defines the set of vehicles that form forbidden pairs with vehicle $i$ at time slot $s$. In the case where $M_i = \emptyset$, i.e., vehicle $i$ is not allowed to match to an already-matched time slot, it matches to a not-yet-matched time slot $s'$, that is, $\Psi(s') = \emptyset$. Otherwise, vehicle $i$ matches to time slot $s_i^* \in M_i$ such that the average cross-influence brought by vehicle $i$ is as small as possible, i.e.,

$$s_i^* := \arg\min_s Q_i^{(s)}.$$  

(12)

Once all vehicles or time slots are matched, the second phase begins.

**D. Phase 2: Rotating the Matching**

Consider a matching $\Psi$ with $L \geq 2$ matched vehicles returned from the first phase. A single matched vehicle from $L := \{1, \ldots, L\}$ is denoted by $\sigma(\ell)$ for $\ell \in L$. We obtain the following original matching sequence:

$$\{ \{\sigma(1), \Psi(\sigma(1))\}, \ldots, \{\sigma(L), \Psi(\sigma(L))\} \}.$$  

(13)

An example of a matching for $L = 3$ is given by $\{1, (1), (2), (2), (3), (3)\}$. A rotation sequence with respect to $\ell, \ell \in L$, is defined as follows [8]:

$$R_\ell := \{ \{\sigma(i), \Psi(\sigma(\text{mod}_L(\ell + i)))\} : i \in L \},$$  

(14)

where the function $\text{mod}_L(x)$ denotes the modulo operator that returns the remainder of the division of $x$ by $L$. Note that $R_L$ consists of the original matching sequence given in (13). For example, with the matching sequence $\{ \{1, (1), (2), (2), (3), (3)\} \}$, we obtain $R_1 = \{ \{1, (2), (2), (3), (3), (1)\} \}$, and $R_2 = \{ \{1, (3), (2), (1), (3), (2)\} \}$, and $R_3 = \{ \{1, (1), (2), (2), (3), (3)\} \}$

Given a rotation sequence $R_\ell, \ell \in L \setminus \{1\}$, we define a rotation matching with respect to $R_\ell$ as follows [8]:

$$\Psi_{L, R_\ell} := (\Psi \setminus R_L) \cup R_\ell.$$  

(15)

The equation (15) means that each vehicle $i$ is re-matched to the time slot defined in the rotation sequence $R_\ell$, that is, vehicle $i$ is initially matched to time slot $\Psi(\sigma(i))$, and after the rotation matching, it will be matched to time slot $\Psi(\sigma(\text{mod}_L(\ell + i)))$. To illustrate this, let us take a look at the example below: $\Psi_{1,2,3, R_1} = (\Psi \setminus R_3) \cup R_1$. That is we transform $\Psi$ from the matching $\Psi(1) = \{1\}, \Psi(2) = \{2\}, \Psi(3) = \{3\}$ to the matching $\Psi_{1,2,3, R_1}(1) = \{2,3\}, \Psi_{1,2,3, R_1}(2) = \{3\}, \Psi_{1,2,3, R_1}(3) = \{1\}$ by removing $R_3$ from $\Psi$ and adding $R_1$ to it.

For $\ell \in L \setminus \{1\}$, a rotation matching $\Psi_{L, R_\ell}$ is valid if any vehicle $i \in L$ does not form a forbidden pair with a matching peer in $\Psi_{L, R_\ell}(\Psi_{L, R_\ell}(i)) \setminus \{i\}$. A valid rotation matching is optimal if it achieves the smallest average cross-influence, i.e.,

$$\ell^* = \arg\min_{\ell \in L} \sum_{i \in L} \sum_{s \in \Psi_{L, R_\ell}(i)} Q_i^{(s)}.$$  

(16)

Finally, a matching $\Psi$ is called $L$-rotation stable if all rotation matchings in $\Psi$ are optimal, i.e., no rotation matching with $L \leq \overline{L}$ can further reduce the average cross-influence of $\Psi$, where $\overline{L}$ is used to control the complexity of the GM algorithm.

With that said, the GM algorithm proceeds as follows. It is implemented in the RSU, which collects the position, distance,
and speed information of the vehicles. Then it finds a feasible matching as described in the first phase. The second phase of GM contains several iterations. Each iteration consists of an optimal rotation matching that the RSU executes after selecting a random rotation sequence. The algorithm terminates when no rotation sequence further reduces the total cross-influence. GM is presented in pseudo-code in Algorithm 1.

Algorithm 1 The GM Algorithm

| Input: | Vehicles and time slots. |
| Output: | An $L$-rotation stable matching $\Psi$. |

//Initialization:
1. Let the current matching be $\Psi$.
2. Define $S' = \{1, 2, \ldots, k\}$ as the not-yet-matched time slots.
3. for $s = 1, k$ do
   4. for $i = 1, m$ do
      5. $F_i^{(s)} = \{i' \in \mathcal{V} : d_i(\psi) \leq r\}$.
   6. end for
   7. end for

//The first phase:
8. for $i = 1, m$ do
   9. Find the set of matched time slots $M_i$ as in (11).
10. if $M_i = \emptyset$ and $S' \neq \emptyset$ then
11. Select a random time slot $s_i^*$ from $S'$.
12. else
13. Select time slot $s_i^*$ according to (12).
14. end if
15. $\Psi(i) = s_i^*$.
16. $S' = S' \setminus \{s_i^*\}$.
17. end for

//The second phase:
18. while the average cross influence can be reduced do
19. Select a random rotation sequence $R_{t+1}$.
20. Find the optimal rotation matching $\Psi_{L,R_{t+1}}$.
21. $\Psi = \Psi_{L,R_{t+1}}$.
22. end while

The complexity of GM algorithm is upper-bounded by the complexity of the second phase in line 18. The complexity of the second phase is given by $\mathcal{O}(mT^2)$ where $L$ is the maximum length of a rotation sequence. In our vehicular network scenario, the value of $L$ is fixed and is much smaller than $m$, thus the complexity $\mathcal{O}(mT^2)$ is considered polynomial. The overall complexity of GM is thus given by $\mathcal{O}((Tm)^2)$, where $T$ is the maximum number of iterations needed for the while loop in line 18 of Algorithm 1 to terminate. The space complexity of GM is $\mathcal{O}(mk)$ since we need to store the matching $\Psi$ and the set of matched slots as given in (11).

To solve VAMP, the RSU (the DRL agent) applies the DDPG method (will be described shortly) in conjunction with GM. Since GM performs time slots/vehicles matching, thus the action space of MDP $\mathcal{M}$ defined earlier is modified to contain only coverage selection decisions and power allocation decisions. Note that, as we saw earlier, vehicle $i$ chooses an action $a_i = \{a_{i1}, a_{i2}, a_{i3}\}$ for $i \in \{1, 2, \ldots, m\}$. We modify this action by encoding it as $[a_{i1}, a_{i3}]$ since the action $a_{i2}$ corresponds to the time slots/vehicles matching already chosen by GM.

The interaction between the DRL agent and the vehicular network environment is given in the system block in Fig. 3. The proposed DDPG-based DRL algorithm is called DDPG-GM and is implemented centrally in the RSU. It proceeds in $E$ episodes as follows. For each episode, the DRL agent observes the vehicular environment and makes decisions accordingly. The decision making is divided into two parts: GM actions (i.e., action $a_{i2}$ for each vehicle $i$) and DDPG actions (i.e., actions $[a_{i1}, a_{i3}]$ for each vehicle $i$). For GM actions, once the DRL agent observes the current state $s(t)$, it applies the GM matching algorithm to find the correspondence between time slots and vehicles. Note that, we can encode the resulting time slots/vehicles matching in the current state, i.e., the current state includes not only the elements of the state space $S$ but also the resulting time slots/vehicles matching. Then, for DDPG actions, we apply an actor-critic, off-policy, model-free algorithm that operates on continuous action spaces [27]. DDPG combines the advantages of DQN (including off-policy learning, replay buffer, and target Q-network updates) and the policy gradient to efficiently make continuous decisions. The DRL agent, to perform DDPG actions as shown in Fig. 3, creates two deep neural networks (DNNs), one is called the actor-network and the other the critic network. The role of the actor-network is to learn the policy of the DRL agent by making decisions about the broadcast communication coverage of each vehicle as well as the transmission power of each transmitting vehicle. On the other hand, the role of the critic network is to learn the value of the Q-function for each state/action pair. Once the DDPG-GM algorithm has observed the environment and chosen its actions $A(t)$, the reward function $R(s(t), A(t))$ is evaluated to find the number of successfully delivered safety packets. Then the environment transitions to the next state and the new time slots/vehicles matching is obtained, which is encoded in the new state $s(t+1)$. A replay buffer is then used to record the transition tuple $(s(t), A(t), s(t+1), r(t))$. Once the replay buffer contains a sufficient number of tuples, a mini-batch of a certain size is sampled from it to perform the policy gradient. Finally, the actor and critic parameters are updated by minimizing some loss functions. The DDPG-GM algorithm is illustrated in Algorithm 2 and some details about it will follow.

The DDPG-GM algorithm is divided into two main phases: an initialization phase and a learning phase. During the initialization phase, the creation of DNNs is performed. First, a critic network $Q(s, A|\theta^Q)$ with weights $\theta^Q$ is created for each state $s$ and action $A$. Then, an actor-network $\pi(s|\theta^\pi)$ with weights $\theta^\pi$ is created for each state $s$. Two target networks are also created that correspond to the critic and actor networks. The target critic network is denoted as $Q'$ and has weights $\theta^{Q'}$, which are initially set to $\theta^Q$ and the target actor-network is denoted as $\pi'$ and has weights $\theta^{\pi'}$, which are initially set to $\theta^\pi$. Finally, the replay buffer is initialized to contain empty transitions, and a $E \times 1$ reward vector $r_{avg}$ is initialized to zero to contain the average rewards of the last 50 episodes.

During the learning phase, DDPG-GM iterates over $E$ episodes. For each episode $e$, the vehicular network is reset,
including the positions, speeds, etc. of the vehicles. Then the GM algorithm is applied to obtain the time slots/vehicles matching solution. This solution is encoded with the other parameters of the vehicles to obtain the current state of the environment. Then a random process is initialized to perform the action exploration in DDPG and the reward that will take into account the reward accumulated so far is initialized to zero. From this point, DDPG-GM iterates through the transmission periods until the second-to-last transmission period and it performs action exploration and policy gradient. An action \( \mathbf{A}(t) \) is selected using the actor-network as follows:

\[
\mathbf{A}(t) = \pi(s(t)|\theta^\pi),
\]

where \( t \) denotes the transmission period. The action is then pre-processed to obtain a feasible action to VAMP and the reward function is invoked to compute the number of successfully delivered safety packets as in (8). Then the accumulated reward \( r_e \) is updated and the next state \( s(t+1) \) is obtained by first applying the GM algorithm to find the corresponding time slots/vehicles matching during \( t+1 \), and then observing the vehicular environment. Once the transition tuple \((s(t), \mathbf{A}(t), r(t), s(t+1))\) is found, it is stored in the replay buffer. As soon as the replay buffer is not empty, a mini-batch training step is performed by sampling a tuple of random \( T \) transitions from the replay buffer. This mini-batch of transitions is trained by minimizing the minimum square error of the loss function:

\[
\frac{1}{T} \sum_t (v(t) - Q(s(t), \mathbf{A}(t)|\theta^Q))^2,
\]

where \( v(t) = r(t) + \gamma Q(s(t+1), \pi'(s(t+1)|\theta^\pi')|\theta^Q') \) and \( \gamma \) denotes the discount value. The minimization performed in (18) updates the weights \( \theta^Q \) of the critic network. Then the weights \( \theta^\pi \) of the actor network are updated by performing the gradient descent method as follows:

\[
\frac{1}{T} \sum_t \nabla_{\theta^\pi} Q(s(t), \mathbf{A}(t)|\theta^Q)|_{s=s(t), \mathbf{A}=\pi(s(t)|\theta^\pi)} \nabla_{\theta^\pi} \pi(s|\theta^\pi)|_{s(t)}.
\]

Finally, the target network weights \( \theta^Q' \) and \( \theta^\pi' \) are updated using the hyperparameter \( \tau \), the previous state is updated to contain the next state, and the average reward of the last 50 episodes is computed.

IV. Simulation Results

In this section, we validate the proposed DDPG-based resource allocation method in a simulated vehicular network. The simulation setup is based on the highway case detailed in 3GPP TR 37.885 [29]. We used the Monte Carlo simulation method to perform all the simulations. We used the Julia programming language [30] with the Flux.jl [31] machine learning library package to perform deep learning. The training is conducted in an offline fashion but the validation is done in an online fashion where the vehicles and network parameters change in real-time. We used a desktop computer with 16 GB of RAM and with a 2.6 GHz Intel i7 processor and NVIDIA GeForce RTX 2070 Super graphic card.

We consider a multi-lane highway with a total length of 2 km where each lane has a width 4 meters. There are a total of six lanes—three for the forward direction (vehicles move from right to left) and three for the backward direction (vehicles move from left to right). The vehicles are generated in the vehicular environment according to a spatial Poisson process. The vehicle speed determines the vehicle density, and
the average distance between vehicles (in the same lane) is set to $2.5s_i v$ [8] where $v$ is the absolute vehicle speed. The speed of a vehicle depends on the lane it is in: the $i$th forward lane (top to bottom with $i \in \{1, 2, 3\}$) is characterized by the speed of $60 + 2(i - 1) \times 10 \text{ km/h}$, while the $i$th backward lane (top to bottom with $i \in \{1, 2, 3\}$) is characterized by the speed of $100 - 2(i - 1) \times 10 \text{ km/h}$. Unless otherwise specified, the important simulation parameters are given in table I.

We train two different vehicular networks. The first one is composed of $m = 4$, vehicles (a 4-sized vehicular network) while the second network is composed of $m = 10$ vehicles (a 10-sized vehicular network). The DNNs are created and trained in the Julia programming language [30] using Flux.jl [31] machine learning library. The actor DNN consists of an input layer of size $s_{\text{actor}} = 2m^2 k + m^2 + 3mk + 4m + 2$ and an output layer of size $a_{\text{actor}} = 2m$ and two fully connected hidden layers containing 500 and 300 neurons respectively. The rectified linear unit activation function (relu) given by $\text{max}(0, x)$ is used in each layer except in the last one in which the hyperbolic tangent function (tanh) given by $(\exp(2x) - 1)/(\exp(2x) + 1)$ is used. The critic DNN consists of an input layer of size $s_{\text{actor}} + a_{\text{actor}}$ and an output layer of size 1 and two fully connected hidden layers containing 500 and 300 neurons respectively. The relu activation function is used in each layer except in the last one where no activation function is used. The actor and critic DNNs are trained with the ADAM optimizer [33] with a corresponding learning rate of $2e-4$ and $1e-4$. The training lasts 1000 episodes. The target update hyperparameter $\tau$ is set equal to $\tau = 1e-3$ and the discount factor $\gamma$ is set to $\gamma = 0.99e-2$. The replay buffer has size of 3e6 and the batch size is 64. The random noise exploration parameter is set to $1e-1$. The channel coefficients are time-varying in each time slot of length 25 ms and a transmission period has 100 ms length.

We summarize the parameters of the actor and critic DNNs in table II, where the size of the input layer depends on the trained vehicular network (either a 4-sized vehicular network or a 10-sized vehicular network).

In the literature, we mentioned that very few papers study the AoI optimization problem in a vehicular network. Further, no previous work studied the objective of minimizing the average AoI by optimizing vehicles coverage, resource allocation (RBs and power), and half-duplex transmitter selection. Our work can be seen as an extension of [8] where the authors proposed how to solve the half-duplex transmitter selection problem but without optimizing the AoI. We adapted the

| Parameter                                | Value     |
|------------------------------------------|-----------|
| Carrier frequency                        | 2 GHz     |
| Bandwidth per RB                         | 100 kHz   |
| Vehicle antenna height                   | 1.5 m     |
| Vehicle antenna gain                     | 3 dBi     |
| Vehicle receiver noise figure            | 9 dB      |
| Shadowing distribution                   | Log-normal |
| Fast fading                              | Rayleigh fading |
| Pathloss model                           | LOS in WINNER + B1 [32] |
| Shadowing standard deviation             | 3 dB      |
| Road configuration                       | Highway road configuration [29] |
| Vehicle drop model                       | Spatial Poisson process |
| Number of transmission periods $n$       | 200       |
| Safety packets sizes $\xi$               | [3, 15] KB |
| Maximum coverage $\varpi$                | 200 m     |
| Maximum transmission power $\mathcal{P}$ | 1 Watt    |
| Noise power $N_0$                        | $-114 \text{ dBm}$ |
proposed algorithm in [8] and build on it to design a DRL method that solves a challenging joint discrete/continuous optimization problem to minimize the AoI. In the simulations results, we compared our proposed solutions to three baseline approaches that we adapted and implemented to solve VAMP: two are based on the NOMA technique and one is based on the OMA technique. The main idea of all baselines comes from [8] and they solve the time slots/matching problem using the GM algorithm. All baselines randomly select the coverage areas of the different vehicles in the interval $[0, C]$. All baselines are centralized and implemented inside the RSU. They are called MAX-GM, RND-GM, and OMA-GM. In the OMA-GM, every time slot is used by at most one vehicle to meet OMA constraints and the vehicles transmit with their maximum transmission power $\overline{p}$. In the MAX-GM and RND-GM, every time slot can be used by any vehicle under the NOMA technique. However, in the MAX-GM, vehicles transmit with their maximum transmission power $\overline{p}$, while in the RND-GM, vehicles transmit with random transmission powers in the interval $[0, \overline{p}]$.

From Fig. 4, we conclude that:

- The rewards increase as the number of episodes increases, which illustrate the convergence of the proposed DDPG-GM algorithm;

- The accumulated rewards do not vary a lot despite the highly dynamic nature of the network. This illustrates the stability of our solutions.

In Fig. 5 and Fig. 6, we illustrate the performance of DDPG-GM when varying the size of the vehicular network as well as the size of the safety packets. We conclude:

\begin{table}[h]
\centering
\caption{Actor and Critic Parameters}
\begin{tabular}{|c|c|}
\hline
Actor Parameter & Value \\
\hline
Size of the input layer & 210 or 1062 \\
Size of the output layer & 8 or 20 \\
Number of hidden layers (HLs) & 2 \\
Number of neurons of first HL & 500 \\
Number of neurons of second HL & 300 \\
Activation function of the first HL & relu \\
Activation function of the second HL & relu \\
Activation function of the output layer & tanh \\
Target update factor $\tau$ & 1e-3 \\
Learning rate & 2e-4 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Critic Parameters}
\begin{tabular}{|c|c|}
\hline
Critic Parameter & Value \\
\hline
Size of the input layer & 218 or 1082 \\
Size of the output layer & 1 \\
Number of HLs & 2 \\
Number of neurons of the first HL & 500 \\
Number of neurons of the second HL & 300 \\
Activation function of the first HL & relu \\
Activation function of the second HL & relu \\
Activation function of the output layer & nothing \\
Target update factor $\tau$ & 1e-3 \\
Learning rate & 1e-4 \\
\hline
\end{tabular}
\end{table}
As the size of the safety packets increases, the AoI increases (although the increase is only marginal) as it becomes more difficult to satisfy their requirements. Thus, fewer safety packets will be successfully broadcast to their destinations, and thus the time between the generation of safety packets and their successful delivery will become large, which increases AoI;

- DDPG-GM manages to learn a better AoI compared to other algorithms and is, therefore, able to deliver more safety packets by optimizing the coverage areas and transmission powers of each vehicle;

- The AoI is higher when the size of the vehicle network is large due to the half-duplex and broadcast nature of V2X communications. Indeed, when there are more vehicles in the network, if any of them broadcast their safety packets, then the interference degrades the performance and thus fewer safety packets will be successfully delivered;

- The OMA technique has the largest AoI, due to the inefficient use of time slots resources.

Fig. 7 and Fig. 8 illustrate the performance of DDPG-GM when the size of the vehicular network and the maximum coverage $\tau$ vary. We have the following conclusions:

- DDPG-GM has the lowest AoI;

- For DDPG-GM, when the maximum coverage $\tau$ increases up to a certain value, the AoI decreases because a larger $\tau$ implies a larger communication coverage and thus more safety packets will be successfully received;

- Beyond a certain value of coverage, the communication coverage of each vehicle increases further, resulting in multiple overlapping communication areas. The improvements of the AoI of the DDPG-GM algorithm is then stopped and it reaches a floor value (and may even start to increase slightly).

These conclusions are particularly true for all algorithms in the case of a large network of vehicles. In the case of a smaller vehicular network, we have the following conclusions:

- The decrease in AoI is faster, especially for the baselines algorithms. This is due to the sparsity present in the network. In other words, when the number of vehicles is small, choosing a small $\tau$ makes the communication area of each vehicle empty and the AoI will therefore be high. Thus, when $\tau$ increases, the decrease in the AoI will be significant;

- In the case of DDPG-GM, due to the efficient learning procedure, the achieved AoI is already minimal even for a small $\tau$, so its decrease when the latter increases are marginal;

- We can conclude that increasing the maximum coverage beyond a certain point is unnecessary because the decrease in the AoI is marginal in the DDPG-GM algorithm;

- The OMA-based algorithm obtains the highest AoI, which is mainly due to the inefficient use of time slots resources;

- The use of the NOMA technique is important for improving the performance of vehicular networks. We further conclude that the DDPG-GM algorithm succeeds in learning better AoI compared to other algorithms thanks to NOMA and the use of the policy gradient-based DRL and DQN approaches.

In Fig. 9 and 10, we illustrate the performance of DDPG-GM when varying the size of the vehicular network and the maximum transmission power $p$. We conclude the following:

- For all algorithms, the AoI decreases slightly as the maximum transmission power increases since it becomes easier to meet the safety packets requirements with higher transmission powers;

- Comparing MAX-GM and RND-GM, we find that the AoI is minimal with MAX-GM because RND-GM uses random power allocation in each NOMA group;

- The OMA-based algorithm achieves the highest AoI which is mainly due to the inefficient use of time slots resources. This illustrates the importance of the NOMA technique in vehicular resource allocation;

- DDPG-GM manages to achieve better AoI than the other algorithms thanks to NOMA and the use of DRL;

- The AoI is higher when the size of the vehicle network is large; which is due to the half-duplex and broadcast nature of V2X communications. Indeed, when there are more vehicles in the network and many of them broadcast their safety packets, the interference becomes high, which affects the performance of the algorithms. Thus,
fewer safety packets will be successfully delivered, and consequently, the AoI increases.

In Fig. 11, we illustrate the impact of vehicle speeds on the AoI by considering a 10-sized vehicular network. We consider two groups of speeds $v_1$ and $v_2$. The $v_1$ speed is given by $[90, 110, 130, 130, 110, 90]$ and the $v_2$ speed is given by $[40, 60, 80, 80, 60, 40]$ where the $i$th component of $v_j$ ($j = 1$ or $2$) represents the $i$th lane and the vehicles belonging to this lane move with the corresponding speed. We have the following conclusions:

- DDPG-GM always has the best AoI compared to all algorithms and regardless of the speeds;
- As the vehicles move faster, minimizing the AoI becomes much more difficult in such a highly dynamic network;
- The AoI corresponding to $v_1$ is higher than the one corresponding to $v_2$ which is much slower than $v_1$.

V. Conclusion, Limitations and Future Works

In this paper, we studied the age of information (AoI) minimization problem in the context of a vehicle-to-everything (V2X) ecosystem by deploying the non-orthogonal multiple access (NOMA) technique. The problem involves coverage optimization, half-duplex transceiver selection, power allocation, and resource blocks (RBs) scheduling. We formulated the problem as a mixed-integer non-linear problem and studied its NP-hardness. Due to the NP-hardness result of the problem as well as the non-linear optimization formulation, classical optimization techniques are not feasible. To solve this complex problem using reinforcement learning (RL) techniques, we formulated it as a single-agent Markov decision process (MDP). Due to the curse of dimensionality and the mixed nature of the action space that contains discrete actions corresponding to the RBs scheduling and transceiver selection and continuous actions corresponding to coverage selection and power allocation, the application of fingerprint-based deep RL (DRL) approaches, such as the deep-Q-network (DQN) methods, is not feasible. To effectively solve the problem, we proposed a decomposition-based greedy matching and deep deterministic policy gradient (DDPG) algorithm. Specifically, based on previous work, we proposed a stable roommate matching algorithm to solve the RBs scheduling problem. Then the DDPG method is used to learn the continuous coverage and power decisions. The DDPG method is based on recent advances in DQN, including off-policy learning, replay buffer, and batch normalization. We showed that the proposed decomposition-based algorithm outperforms baseline algorithms and successfully provides low AoI in different vehicular network scenarios.

The main limitations of our work that we will address in future works are the following. First, the multi-agent DDPG algorithm could be investigated to solve the continuous resource allocation subproblem and thus implement a fully distributed approach. Second, modeling and solving the entire discrete/continuous resource allocation problem using DRL approach is a challenging task that should be carefully studied in future work where for example the clip and count-based proximal policy optimization could be investigated. Third, a joint method rather than a decomposition-based one would be interesting to investigate in future. Finally, dynamic programming approach can be a perfect candidate solution for the problem that is worth studying.
