Adaptive Delivery for High Definition Map Using A Multi-Arm Bandit Approach

Dawei Chen *, Haoxin Wang †, and Kyungtae Han *
* InfoTech Labs, Toyota North America R&D, Mountain View, CA, USA
† Department of Computer Science, Georgia State University, GA, USA
Email: dawei.chen1@toyota.com, haoxinwang@gsu.edu, kyungtae.han@toyota.com

***Abstract*—A high definition (HD) map is a key technology that enables autonomous driving, which has the characteristics of frequent updates and low latency requirements. Edge computing provides an efficient way to deliver the HD map to autonomous vehicles, which deploys the edge servers at the edge of the network and shortens the transmission distance. The edge-assisted HD map delivery is generally done by the wireless transmission between edge servers, like roadside units (RSU), and vehicles. However, the transmission channel status, like the transmission rate, is fragile and easily influenced by the speed of vehicles, the weather, and the number of connections of RSU. A proper HD map delivery is needed to meet a time deadline over different channel conditions. This work firstly utilizes the love-of-variety-based method to model the different versions of the HD maps with different data sizes. Then, an adaptive upper confidence bound based multi-arm bandit method is proposed to choose the appropriate version of the HD map under the different wireless communication statuses. The simulation results show the effectiveness of our proposed method, which achieves the best total accumulative rewards and the least regret compared with the baseline methods.**

I. Introduction

Having stepped into the era of information technology, there are enormous artificial intelligence-based autonomous devices, technologies, and services coming into being, and one important branch is the autonomous vehicle or the intelligent vehicle. According to the National Highway Traffic Safety Administration, the levels of vehicle automation can be categorized into six classes, which are distinguished by the extent of autonomy. Currently, the performance of autonomous vehicles can just meet the requirements between level 2 and 3, and both of which require the driver must be ready to take back control at any time. Aimed at achieving a higher automation level, one effective way is to utilize the high definition (HD) map. Unlike the traditional map, HD map is represented with a high degree of precision and resolution, which is as fine as 10-20 centimeters or better.

HD map is made up of various information and resources, such as drivable paths, lane marks, the priority of lanes, traffic lights and crosswalk to lane association, adjacent objects, and street furniture, which is represented in a high degree of resolution and precision, generally in the centimeter level. For practical autonomous driving use cases, HD map is the indispensable key for the advanced driver assistance system (ADAS). Intuitively, the contents of HD map can be roughly categorized into two classes: dynamic objects (such as pedestrians and vehicles) and static objects (such as traffic signs and lights). According to the definition of Automotive Edge Computing Consortium (AECC), the composition of HD map is layered, which can be represented by the highly dynamic layer, transient dynamic layer, transient static layer, and permanent static layer, as is illustrated in the upper part of Fig. 1 [1].

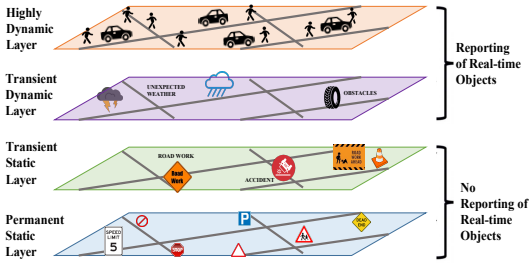

Fig. 1. HD map structure.

In the current map, it is common to find the gaps between actual circumstances and the map, which take days to be corrected. However, for autonomous driving, any delayed update for a HD map can result in dangerous or even fatal accidents without human intervention [2]. Therefore, a HD map must be updated in a timely manner. To deliver a HD map in a time-efficient way, edge computing paradigm can be an effective solution, which geographical-distributively deploys edge nodes within the network. In this way, the transmission distance can be reduced. The overall latency for the HD map delivery can be remarkably reduced accordingly [3].

However, the edge-assisted HD map delivery is generally done by the wireless transmission between an edge server, like a roadside unit (RSU), and vehicles. The transmission rate is fragile and easily influenced by the speed of vehicles, the weather, and the current number of connections of RSU. With different channel conditions, to complete HD map delivery within the time deadline, this work proposes a multi-arm bandit-based adaptive HD map selection scheme to choose the proper version of HD map to be delivered so that the deadline can be met anyway.

One of the challenges is how to model the different versions of HD map and quantify the corresponding data sizes. In this work, we propose a love-of-variety-based method to connect the number of sensors used for HD map generation and the HD map data size. When the variety is low, the HD map data is small as well since the information contained is less, which

is in accordance with the low transmission rate case. Likewise, when the variety is high, the number of utilized sensors is large, so the HD map data is large and in accordance with the high transmission rate case. To the best of our knowledge, there is no existing literature that models the HD map using love-of-variety-based approach.

Having modeled the different versions of HD map with different data sizes, how to select the proper one under a different communication status needs to be considered. Since communication channel varies from time to time, an adaptive upper confidence bound (AUCB) based-multi-arm bandit (MAB) method is proposed to solve such a sequence decision problem [4]. There is some existing literature investigating the MAB methods on autonomous driving applications. [5] proposes a restless MAB method to find a scheduling scheme for the edge server to minimize the traveled distance of autonomous vehicles. [6] proposes a multi-agent MAB algorithm for RSUs to learn the caching strategy for maximizing the accumulated cache utility over the time horizon. [7] proposes a MAB-based quality aware and cost-aware vehicle selection scheme to dynamically select suitable vehicles to reduce task replications. Also, there are some existing work focusing on the the low-latency and accurate HD map generation. [8] proposes a reinforcement learning-based data source selection scheme for efficient HD map distribution in vehicular named data networking scenarios. [9] proposes an HD map update algorithm, which utilizes only reliable information and considers the geometric characteristics of landmarks in a number of crowdsourced data. However, no existing literature considers an adaptive HD map selection for HD map delivery under different wireless channel conditions.

To summarize, the main contributions of this paper are as the following:

- In this paper, we propose a love-of-variety-based method to model the different versions of HD map with different data sizes. To the best of our knowledge, there is no existing literature doing in this way.
- To address the HD map version selection problem regarding different wireless communication status, an AUCB algorithm is proposed.
- Simulation results show the effectiveness of our proposed method, which achieves the best total accumulative rewards and the least regret compared with the baseline methods.

II. System Model and Problem Formulation

In this section, we firstly introduce the love-of-variety-based HD map modeling in subsection II-A. Then, the communication modeling between an edge server and vehicles is described in subsection II-B. Finally, the formulation of the proper HD map selection problem with delivery time deadline is given in subsection II-C.

A. High Definition Map Model

HD map is inevitable for autonomous driving, which is overlaid with various information, such as road signs, lane markings, pedestrian, and vehicle locations. Basically, the HD map is generated from the crowdsourcing data captured by the on-vehicle sensors, such as Radar, LiDar, camera, GNSS, IMU, and ultrasonic sensors. Intuitively, when the number of sensors is larger, the data size of generated HD map is huger accordingly, since the information collected by the different kinds of sensors is addictive. To model the different versions of HD map, we propose a love-of-variety-based approach.

We assume the HD map generator can perform one type of data in a specific time slot, and the generator has a demand for multiple varieties of sensor data over time to extract the entire information. Therefore, a time vector $\boldsymbol{t} = t_{i\in\mathcal{N}}$ can be used to describe the data analysis process over different sensors, where t_i indicates the HD map generator is executing a specific kind of sensor data during a fixed time slot, i is the time slot index, and $\mathcal{N}$ is the total time slots index set. Apart from the utilization of diverse sensors, the data size of the HD map is also related to computation time slot t_i. With the increase of time consumption, more data can be analyzed, and the data size of obtained information that contributes to the formation of the HD map will be larger accordingly. Therefore, the utility function that indicates the data size of the generated HD map can be defined as $u(t_i)$, which is a strictly increasing and concave function and meets the condition $u(0) = 0$, as is suggested in [10]. The general utility function can be defined as

$$u(t_i) = \frac{1}{1-\rho}\left[(a+t_i)^{1-\rho} - a^{1-\rho}\right] + bt_i, \tag{1}$$

where $a \geq 0$, $b \geq 0$, and $0 < \rho < 1$ are constant coefficients. Intuitively, the data size of the obtained HD map from computation over different types of sensors are additive. Overall, the aggregated utility of HD map data generator is $\sum_{i\in\mathcal{N}} u(t_i)$.

Mathematically, to involve variety, one key challenge is how to evaluate or quantify the computation ability of the HD map generator to hand over computation from one kind of sensor data x to another kind of sensor data y, or how to quantify the diversity of sensor data computed within a certain time period. Besides, since the HD map generator needs diverse sensor data to obtain more accurate and effective information for HD map, quantifying the willingness of exchanging among multiple sensors is also challenging. In order to solve this problem, we introduce elasticity, whose definition is shown below.

Definition 1. *For two variables x and y, the x-elasticity of y is defined as $\epsilon_x^y = -\frac{\partial y}{\partial x}\frac{x}{y}$.*

The interpretation of elasticity is that the percentage change in y is in response to the percentage change in x. If the value of elasticity is larger, it means y is more sensitive to the change of x. To quantify the willingness of exchanging among multiple sensors, the definition of relative love-of-variety (RLV) is given as the following.

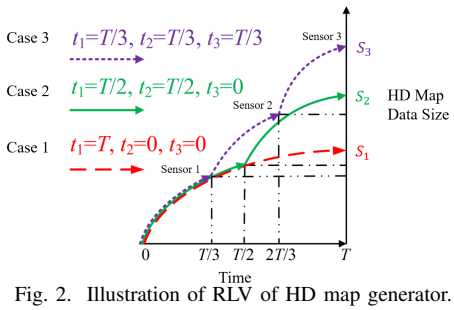

Fig. 2. Illustration of RLV of HD map generator.

Definition 2. *The HD map generator's relative love-of-variety is the elasticity of the marginal utility with respect to the computation time slot t_i, which is described by*

$$r(t_i) = \epsilon_{t_i}^{u'} = -\frac{u^{''} t_i}{u'} > 0. \tag{2}$$

Obviously, from Definition 2, the value of RLV reflects whether the HD map generator is willing to exchange different sensor data in consecutive time slots for achieving a higher marginal utility [11], as is described in Fig. 2. Also, we associate the relative love-of-variety with data size, which is $s = r \times f$, where f is a constant denoting the data size parameter. For case 3, the HD map generator changes consumption among sensor data at the end of time slot $\mathcal{N}/3$ and achieves the largest HD map data size s_3. For case 2, the HD map generator consumes two types of sensor data within time $\mathcal{N}$ and achieves data size s_2. While the HD map generator in case 1 keeps the same type of sensor data throughout time interval and achieves the lowest data size s_1.

B. Communication Model

In this work, we adopt time-division medium access (TDMA) technology as the communication protocol between the vehicle and RSU. Without loss of generality, for other protocols, similar approaches can be easily extended. Besides, it is assumed that the vehicles within the same RSU coverage are allocated an orthogonal sub-channel and the interference brought by neighbor users can be ignored. For the specific autonomous vehicle, the transmission rate at each time slot can be described as $\nu_i = B \log_2 \left(1 + \frac{ph}{N_0}\right)$, where B is the sub-channel bandwidth allocated to the autonomous vehicle, p is transmission power, h is channel gain between the vehicle and RSU at time slot t_i, and N_0 is the Gaussion noise. Intuitively, the required time for HD map transmission at each time slot can be characterized as $d_i = \frac{s_i}{\nu_i} = \frac{s_i}{B \log_2 \left(1 + \frac{ph}{N_0}\right)}$. Suppose the span of each time slot t_n is the deadline for the HD map delivery, therefore, with the proper selection of s_i, the delivery time d_n should be less or equal to t_n.

C. Problem Formulation

We consider a time period T containing n time slots $\{t_1, ..., t_i, ..., t_n\}$. Within each time slot, the autonomous vehicle needs the corresponding HD map being delivered from the RSU. There are total k versions of HD map and the set of available HD map is denoted as $S = \{s_1, ..., s_k\}$. At each time slot t_i, the map s_{ij} is selected, where $j \in [1, k]$. As is described in the previous subsection, the different version of HD map is made up by a different number of sensors, accordingly, the utility or quality of HD map r_j and the data size s_i will be different as well. For each HD map delivery time slot, the purpose is to get the highest map quality while meeting the deadline of transmission completion time, i.e., within the time slot. Therefore, the objective function of this problem can be defined as follows,

$$\begin{aligned} &\max_j \sum_{i=1}^{n} r_{ij}, \\ &s.t.\ C1 : \frac{s_{ij}}{\nu_i} \leq t_i, \\ &C2 : s_j \in S. \end{aligned} \tag{3}$$

The first constraint denotes that the transmission time of the selected HD map cannot exceed the time limitation, which is the duration of a time slot. The second constraint describes the selected HD map should belong to the available HD map set. Such a submodular problem with cardinality constraint is approved NP-hard [6]. To solve this problem, we propose a multi-arm bandit based-solution.

III. Method

In this section, the traditional MAB methods, greedy, and ϵ greedy, are introduced in subsection III-A and the proposed AUCB is described in subsection III-B

A. Traditional Action-Value Methods

To solve problem (3), a MAB based-method can be a solution. Basically, MAB problem is a class of problems that allocate a limited set of resources between alternative choices in a way so that the expected rewards can be maximized. Here, we define each action a_t is the choice of the HD map version s_j. The rewards R_t will be the corresponding utility of the selected HD map. The expected value followed by an arbitrary action a is the expected reward given that arm a is selected, which can be denoted as $q_*(a) = \mathbb{E}[R_t | A_t = a]$. In other words, the purpose of our proposed method is to select the HD map version a with the highest utility and meet the transmission deadline at the same time.

To solve this problem, the challenge is the tradeoff between exploration and exploitation [12]. Since the reward distribution of each arm is not prior-known for the vehicle, therefore, the final policy will be learned based on each observation. We denote the estimated value of action a at time slot t as $Q_t(a)$. The closeness of $Q_t(a)$ and $q_*(a)$ is desired. Here, exploration means we need to choose the specific version of HD map many times so that the reward can be estimated more accurately or the reward statistics are more precise. The exploitation here means the current HD map version with the highest reward should be selected so that the long-term rewards can be achieved.

To maximize the long-term rewards, the challenge is to estimate the value of actions and use estimates to make action selection decisions. One thing to note here is the true value of

Algorithm 1 ϵ-greedy algorithm for MAB

1: Input: Action space $\mathcal{A}$; Maximum iteration steps $t_{\max}$; Exploring probability ϵ.
2: **for** a=1:k **do**
3: Initialize $Q(a) = 0$ and $R_0 = 0$;
4: **end for**
5: **for** t=1:n **do**
6: **if** probability is $1 - \epsilon$ **then**
7: Select an arm based on (4);
8: **else**
9: Select a random arm in an equal probability way;
10: **end if**
11: Accumulate the reward R_t;
12: Update Q_t by $Q_{t-1} + \frac{1}{t}(R_{t-1} - Q_{t-1})$;
13: t=t+1;
14: **end for**
15: Output: the optimal HD map selection scheme.

an action is the mean reward when the action is chosen. An intuitive way to estimate this is to use the received rewards: $Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i \times \mathbb{1}_{A_t=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_t=a}}$, where $\mathbb{1}$ is the selection variable that is 1 if the action is taken and 0 if it is not. The numerator means the sum of rewards when a is taken prior to the current time slot t. In addition, the denominator is the number of times to take action a prior to the current time slot t.

The typical solution of MAB for the high exploitation is the greedy method, which means at each time slot, the arm with the highest reward will be selected and the probability to select the other is zero. The corresponding selection scheme can be written as

$$A_t = \arg\max_a Q_t(a). \tag{4}$$

Here, $\arg\max_a$ means the action a maximizes the estimated reward value. Obviously, the greedy method always exploits the current knowledge to maximize the concurrent reward.

The greedy method mentioned above selects the arm at each time slot in an exploitation way, which is disadvantageous for maximizing the long-term rewards. Because with a limited number of arms observed, the rewards and corresponding statistics of other unobserved ones cannot be obtained, which may contain higher rewards than existing observations [13]. To make a balance between exploitation and exploration, one way is to introduce a probability to select the other arm, which is called ϵ-greedy algorithm. This method behaves greedily most of the time, but with the small probability ϵ, it selects randomly from among all the actions with equal probability, which is independent of the action-value estimation. The details are shown in Algorithm 1.

Step 9 in Algorithm 1 is the incremental implementation update method, which helps compute the average of observed rewards in a computation-efficient way. Originally, the average rewards are calculated by $Q_t = \frac{R_1+R_2+...+R_{t-1}}{t-1}$, where Q_t describes the estimation of action value after it has been chosen $t-1$ times. However, in this manner, the requirements for memory and computation would increase over time as more observations were obtained. Since each additional reward needs extra memory to store and additional computation to calculate the numerator. To overcome this, we take the update scheme as step 9 in Algorithm 1. A simple derivation is shown below. Given the $t-1$-th estimation Q_{t-1} and reward R_{t-1}, the updated average of all the n rewards can be obtained by (8).

$$\begin{aligned} Q_n &= \frac{1}{t-1}\sum_{i=1}^{t-1} R_i = \frac{1}{t-1}\left(R_{t-1} + \sum_{i=1}^{t-2} R_i\right) \\ &= \frac{1}{t-1}\left(R_{t-1} + (t-2)\frac{1}{t-2}\sum_{i=1}^{t-2} R_i\right) \\ &= \frac{1}{t-1}\left(R_{t-1} + (t-2)Q_{t-1}\right) \\ &= \frac{1}{t-1}\left(R_{t-1} + (t-1)Q_{t-1} - Q_{t-1}\right) \\ &= Q_{t-1} + \frac{1}{t}\left(R_{t-1} - Q_{t-1}\right). \end{aligned} \tag{5}$$

Obviously, this method only requires memory for Q_t and t, and small computation for each new observed reward.

B. Adaptive Upper-Confidence-Bound Method

The greedy method focuses on the best-performed arm at representation but sometimes the other arms may be better. ϵ-greedy introduces a probability to try the non-greedy actions but with no preference for those that are nearly greedy or particularly uncertain. It would be better to select among the non-greedy actions according to their potential for actually being optimal, taking into account both how close their estimates are to being maximal and the uncertainties in those estimates. In addition to our empirical reward estimates, we need an upper confidence bound to describe the largest plausible mean of each arm. One way to solve this is the upper-confidence-bound (UCB) algorithm [14]. For the UCB method, to construct the confidence interval, the utilization of Hoeffding's inequality and Chernoff bound is inevitable, which are defined as follows.

Theorem 1. *(Hoeffding's inequality) Given independent random variables $\{X_1, ..., X_m\}$ where the range of each variable is $a_i \le X_i \le b_i$, we have*

$$\mathbb{P}\left(\frac{1}{m}\left(\sum_{i=1}^{m} X_i - \sum_{i=1}^{m}\mathbb{E}[X_i]\right) \ge \epsilon\right) \le e^{\left(\frac{-2\epsilon^2 m^2}{\sum_{i=1}^{m}(b_i-a_i)^2}\right)}. \tag{6}$$

In this way, Hoeffding's inequality provides an upper bound on the probability that the sum of bounded independent random variables deviates from its expected value by more than a certain amount. Traditionally, central limit theorem guarantees are useful for large sample sizes, but if the number of samples is small, it is not suitable anymore. Therefore, the Chernoff bound can be an alternative way to deploy.

Algorithm 2 Proposed AUCB algorithm

Input: The transmission rate r_i; Action space $\mathcal{A}$; Maximum iteration steps $t_{\max}$; Exploring probability ϵ; The span of each time slot g.
for t=1:n **do**
  Calculate the transmission time T_i of each HD map version based on current transmission rate r_i.
  if $T_i < g$ **then**
    The HD map version will be put into action space;
  else
    Remove the HD map version in the action space;
  end if
  for a=1:k **do**
    Initialize $Q(a) = 0$ and $R_0 = 0$;
    Select an arm based on the policy (7);
    Accumulate the reward R_t;
    Update Q_t by $Q_{t-1} + \frac{1}{t}(R_{t-1} - Q_{t-1})$;
    t=t+1;
  end for
end for
Output: the optimal HD map selection scheme.

Theorem 2. *(Chernoff bound) Given the independent random variable X, the bound in terms of its moment-generating function is given by $\mathbb{P}(\mathbb{X} \geq \epsilon) \leq \frac{\mathbb{E}[e^{tX}]}{e^{\epsilon t}}$.*

Then, using the Hoeffding's inequality and Chernoff bound, the action selection method of AUCB is described as

$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right], \tag{7}$$

where $\ln t$ is the natural logarithm of t, $N_t(a)$ denotes the number of times that action a is selected prior to time t, and the number $c > 0$ is a constant that controls the degree of exploration.

Obviously, AUCB adopts a square root term to measure the uncertainty or variance of the action's value. Mathematically, what is maximized in (7) is a sort of upper bound on the possible values of action a, where c is the one to determine the confidence level. On the one hand, the uncertainty is presumably reduced for each time the arm a is selected. This is because the denominator $N_t(a)$ increments and the uncertainty term decreases accordingly. On the other hand, each time when the arm that is not a is selected, the t increases but the denominator $N_t(a)$ remains the same, resulting in the increasing uncertainty estimation. The natural logarithm in the numerator decides the increase will be smaller and smaller over time but the value is bounded. Therefore, all of the actions will be traversed but the selection probability for the arms with low estimation values or high-frequency selections will decrease over time [15]. The overall proposed AUCB based-method is shown in Algorithm 2.

Now, for the algorithms, how to evaluate their performances needs to be considered. One standard approach is to compare the algorithm's cumulative rewards to the best-arm benchmark $q^*(a) = \max_a Q(a) \times T$. The regret can be defined as the following.

Definition 3. *The regret of the algorithm at round T is defined as:*

$$R(T) = q^*(a) \times T - \sum_{t=1}^{T}(a_t). \tag{8}$$

With this definition, the regret bound of AUCB can be defined.

Theorem 3. *The regret bound for the UCB algorithm is, for $T \geq 1$,*

$$R(T) \leq \sum_i 4c^2\delta_i^{-1}\log(T) + \frac{2c^2}{c^2-1}\delta_i, \tag{9}$$

where $\delta_i = \max_j q^*(a)_j - q_i$ is the optimality gap of an arm.

IV. Simulation results

In this section, the simulation results are introduced. The experiments are conducted on MATLAB platform. The total play rounds are set as 2,000. The parameter ϵ is set as 0.01 for ϵ greedy algorithm. c is set as 3 for the proposed AUCB algorithm. The RLV-based utility function is defined as $u(t_i) = 2 \times (1 + t_i)^{0.5} - 1$, by setting $a = 0$, $b = 1$, and $\rho = 0.5$. During the time period, the channel bandwidth B is randomly chosen from 10 MHz, 20 MHz, 30 MHz, 50 MHz, and 100 MHz. The transmission power p is 23 dBm. The Gaussian noise N_0 is set as -96 dBm. The HD map data size parameter f is set as 20. The duration of each time slot t_i is 0.1s. The number of arms k is set as 5, which means we have 5 different versions of HD map.

The total accumulative rewards and regret are illustrated in Fig. 3 and Fig. 4, respectively. In Fig. 3, from the perspective of long-term rewards, the proposed AUCB achieves the highest value, followed by the ϵ greedy and greedy algorithms, respectively. This is because the greedy method focuses on the best-performed arm at representation but sometimes other actions may be better. ϵ-greedy introduces a probability to try non-greedy actions but with no preference for those that are nearly greedy or particularly uncertain, which results in a better performance compared with the greedy algorithm. For the proposed AUCB algorithm, it adopts a square root term to measure the uncertainty or variance of the action's value, which helps to make a better trade-off between exploration and exploitation. With the wireless channel constraint, the proposed AUCB has a high probability to excavate a better arm to play instead of choosing the current best-performed one. This can also be verified in Fig. 4. The regret shows the closeness between the optimal HD map chosen policy and the actual policy. Therefore, a lower regret value indicates the performance of the algorithm is better. Likewise, since the better trade-off between exploration and exploitation, the proposed AUCB algorithm achieves the lowest regret value, followed by the ϵ greedy algorithm. The greedy algorithm obtains the highest regret value because it is actually conducted in a pure exploitation way.

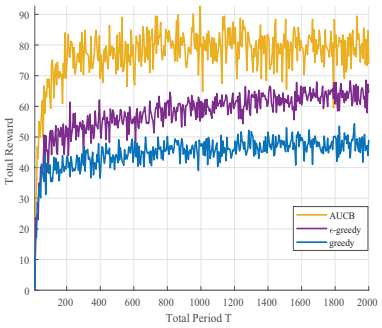

Fig. 3. Total cumulative rewards comparison.

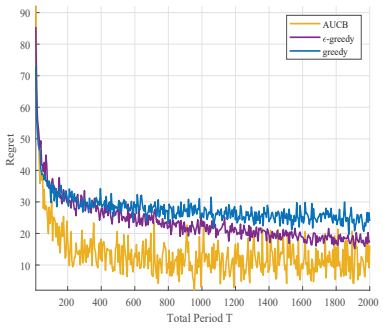

Fig. 4. Regret comparison.

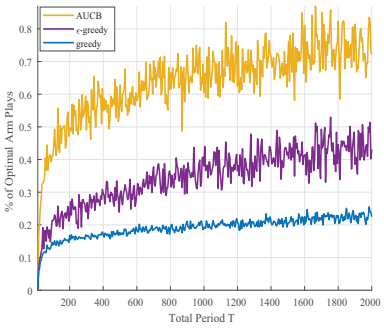

Fig. 5. Optimal arm selection probability.

In addition, the results of the probability of optimal arm playing are shown in Fig. 5, which is in accordance with the previous results for rewards and regrets. With a higher probability allocated to the exploration, the chances to find the optimal arm are increased accordingly. Therefore, the proposed AUCB achieves 80% probability to choose the best proper HD map for delivery, which meets the time deadline and obtains more information (reflected by a higher relative love-of-variety and rewards) simultaneously.

Besides, from Figs. 3, 4, and 5, we can find one thing is common, which is that the curve of AUCB fluctuates the most while the greedy one is more steady compared with the others. This is because greedy is performed in a pure exploitation way and with no exploration, which makes it more stable. Although ϵ greedy makes an improvement by introducing the probability ϵ for exploration, the value is usually small so it behaves steadily compared with the proposed AUCB. While in terms of the proposed AUCB, the term $\ln t$ is introduced for the arm selection policy. All of the actions will be traversed but the selection probability for the arms with low estimation values or high-frequency selections will decrease over time, resulting in the obvious fluctuation.

V. Conclusion

As a latency-sensitive application, HD map delivery needs to be performed in a timely manner for enabling autonomous driving. This work discusses the problem of edge-assisted HD map delivery with adaptive version selection under different wireless channel statuses. To solve this problem, we firstly propose a love-of-variety method to model the different versions of HD maps that have different data sizes. Then, an AUCB method is proposed to learn the optimal policy for HD map selection under a certain wireless condition. The simulation results show the effectiveness of our proposed method, which achieves a higher total accumulative reward, a lower regret value, and a higher optimal arm selection probability, compared with baseline methods. For the future work, we will explore the impact of factors on the performances, such as the bandwidth and required computation power. In addition, this work currently consider one vehicles that requires HD map. We would like to consider a scenario that a group of vehicles share the same link with edge server that need HD map under different communication status.

References

[1] Automotive Edge Computing Consortium White Paper. (2020, May) Operational Behavior of a High Definition Map Application. [Online]. Available: https://aecc.org/wp-content/uploads/2020/07/Operational_Behavior_of_a_High_Definition_\\Map_Application.pdf

[2] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1697–1716, Jun. 2019.

[3] D. Chen, Y.-C. Liu, B. Kim, J. Xie, C. S. Hong, and Z. Han, "Edge computing resources reservation in vehicular networks: A meta-learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5634–5646, March 2020.

[4] Y. Xu, B. Kumar, and J. D. Abernethy, "Observation-free attacks on stochastic bandits," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 550–22 561, Dec. 2021.

[5] M. Li, J. Gao, L. Zhao, and X. Shen, "Adaptive computing scheduling for edge-assisted autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5318–5331, March 2021.

[6] X. Xu, S. Gao, and M. Tao, "Distributed online caching for high-definition maps in autonomous driving systems," *IEEE Wireless Commun. Lett.*, vol. 10, no. 7, pp. 1390–1394, March 2021.

[7] Y. Qian, Z. Zuo, and Y. Hao, "Online vehicle selection for task replication via bandit learning," in *2021 IEEE 45th Annual Computers, Software, and Applications Conference*, Madrid, Spain, July 2021.

[8] F. Wu, W. Yang, J. Lu, F. Lyu, J. Ren, and Y. Zhang, "Rlss: A reinforcement learning scheme for hd map data source selection in vehicular ndn," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10 777–10 791, Jul. 2021.

[9] K. Kim, S. Cho, and W. Chung, "Hd map update for autonomous driving with crowdsourced data," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1895–1901, Feb. 2021.

[10] Y. Zhao, H. Wang, H. Su, L. Zhang, R. Zhang, D. Wang, and K. Xu, "Understand love of variety in wireless data market under sponsored data plans," *IEEE J. Select. Areas Commun.*, vol. 38, no. 4, pp. 766–781, February 2020.

[11] Y. Zhao, H. Su, L. Zhang, D. Wang, and K. Xu, "Variety matters: a new model for the wireless data market under sponsored data plans," in *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, Phoenix, AZ, USA, Jun. 2019, pp. 1–10.

[12] T. B. de Oliveira, A. L. Bazzan, B. C. da Silva, and R. Grunitzki, "Comparing multi-armed bandit algorithms and q-learning for multiagent action selection: a case study in route choice," in *2018 International Joint Conference on Neural Networks*, Rio de Janeiro, Brazil, July 2018.

[13] A. Ferdowsi, S. Ali, W. Saad, and N. B. Mandayam, "Cyber-physical security and safety of autonomous connected vehicles: Optimal control meets multi-armed bandit learning," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7228–7244, July 2019.

[14] P. Dai, Z. Hang, K. Liu, X. Wu, H. Xing, Z. Yu, and V. C. S. Lee, "Multi-armed bandit learning for computation-intensive services in mec-empowered vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7821–7834, April 2020.

[15] K. Xiong, S. Leng, X. Chen, C. Huang, C. Yuen, and Y. L. Guan, "Communication and computing resource optimization for connected autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12 652–12 663, October 2020.