Efficient Neural Nets for V2X Sequential Dynamic User Allocation

Ha Manh Bui [1]

Abstract

We study the Vehicle-to-everything (V2X) Sequential Dynamic User Allocation problem, i.e., how to allocate new vehicles to multiple wireless Access Point (AP)s s.t. can maximize the total throughput of all AP. This problem has been formalized by the Combinatorial Multi-Armed Bandit setting and solved by classical algorithms (e.g., greedy and Combinatorial Upper Confidence Bound-Resource Allocation). However, such classic approaches still have a low allocation performance. Recently, the Neural Combinatorial Bandits (NCB) has shown promising results by enhancing allocation performance. That said, this method needs to re-optimize a Neural Network model with all of the cumulative training data for every step, causing a high response latency and limiting the scalability in real-time V2X applications. To tackle this challenge, we propose an efficient Neural Bandits algorithm that only needs to update model parameters with the latest sample. We show our proposed method has a competitive resource allocation with NCB and outperforms them in terms of inference speed.

1. Introduction and Related work

Vehicle-to-everything (V2X) has become a hot research topic recently by the development of self-driving cars and the emergence of the 5th generation mobile network. It is a communication between a vehicle and related entities, including Vehicle-to-infrastructure and Vehicle-to-vehicle. Due to a massive amount of information signals (e.g., speed, traffic condition, direction, location, traffic incidents, etc.), and the high mobility, the V2X communication channel state information becomes outdated rapidly and its application is required **high efficiency** (i.e., ultra-low latency) and **high reliability** (Christopoulou et al., 2023).

One resource allocation problem in computer networking that is related to V2X could be Sequential Dynamic User Allocation (SDUA) (Gupta et al., 2022), in which at each time, given multiple wireless Access Point (AP)s with different and unknown numbers of existing vehicle users, we need to allocate new incoming vehicles to these APs s.t. can maximize the total throughput of all AP. This problem can be viewed as an exploration and exploration trade-off. To solve this, Gupta et al. (2022) has formalized under the Combinatorial Multi-Armed Bandit (CMAB) setting (Chen et al., 2013; 2016), and then used classical bandit algorithms, including greedy and Combinatorial Upper Confidence Bound-Resource Allocation (CUCB-RA). However, such algorithms have low reliability (i.e., allocation performance) by still having high regret and low reward.

Method	High reliability	High efficiency
Classic	✗	✓
NCB	✓	✗
Ours	✓	✓

Table 1. A comparison between methods in terms of reliability with allocation performance (regret & reward) and efficiency (latency).

Recently, with the development of Neural Networks, Hwang et al. (2023) has shown Neural Combinatorial Bandits (NCB) approach that outperforms aforementioned classical bandit algorithms with a significantly lower regret and higher reward. That said, the NCB algorithm needs to optimize a Neural Network model with all of the cumulative training data for every step, causing a high latency and limiting the scalability in real-time V2X SDUA application (Christopoulou et al., 2023; Gao et al., 2019).

Motivated by the NCB algorithm and toward reducing the gap between resource allocation and efficiency performance aspects (see Tab. 1), in this project:

1. We reimplement the SDUA experiment in Gupta et al. (2022), and we additionally shows the result of NCB (Hwang et al., 2023) in this setting.
2. We show NCB is not efficient, and we propose an efficient algorithm that can achieve the **ultra-low latency** performance in V2X.
3. We empirically show our proposed method has a competitive result with NCB in terms of resource allocation performance, and outperforms them in terms of efficiency.

[1]Department of Computer Science, Johns Hopkins University, Baltimore, Maryland, USA. Correspondence to: Ha Manh Bui <hb.buimanhha@gmail.com>.

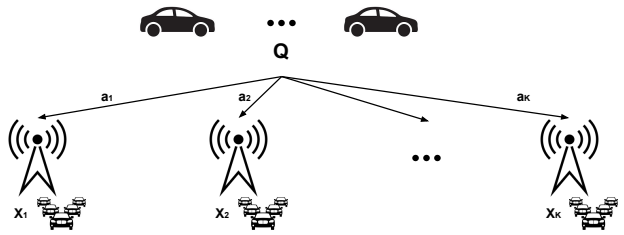

Figure 1. SDUA problem by allocating Q vehicle users to K wireless APs s.t. can maximize the total throughput of all AP.

2. Methods

Notation and the SDUA problem. We consider the Sequential Dynamic User Allocation (Gupta et al., 2022), with K wireless Access Points in Fig. 1. At each round, there are X_k unknown existing users in each AP k, and Q new incoming vehicle users. Our goal is allocate this Q vehicle users to K APs by action vector $\mathbf{a} = (a_1, \cdots, a_K)$, $\mathbf{a} \in \mathbb{N}^K$, s.t. can maximize the total throughput of all AP, i.e.,

$$\max_{a_k} \sum_{k=1}^{K} 0.2(X_k + a_k)e^{-0.2(X_k+a_k)}, \text{ s.t. } \sum_{k=1}^{K} a_k = Q, \tag{1}$$

where each user has a fixed traffic load of 0.2 and consider the well-known ALOHA protocol (Abramson, 1970) for each AP. Note that we assume all the users of an AP will leave when the round ends, so a_k in the current round will not affect X_k in the future rounds (Gupta et al., 2022).

SDUA under CMAB setting. Let consider an online version, at each time t, we allocate $a_{k,t}$ to k s.t. $\sum_{k=1}^{K} a_{k,t} = Q$. Then, we observe feedback reward $f_k(a_{k,t}, X_{k,t}), \forall k$, where $X_{k,t}$ is a random variable reflects the random fluctuation of the generated reward, i.e., $X_{k,t} \overset{\text{i.i.d.}}{\sim} \mathbb{P}(X_k)$. From Eq. 1, an action at time t is a vector of K arms, i.e., $\mathbf{a}_t = (a_{1,t}, \cdots, a_{K,t})$ and we can define the expected total reward at t as

$$r(\mathbf{a}_t, \mathbb{P}(\mathbf{X})) = \mathbb{E}[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})], \tag{2}$$

where $\mathbb{P}(\mathbf{X}) = (\mathbb{P}(X_1), \cdots, \mathbb{P}(X_k))$. Let us denote the action caused by policy π be $\mathbf{a}_t^\pi$.

Under CMAB setting and motivated by the Combinatorial Upper Confidence Bound (Chen et al., 2013), Zuo & Joe-Wong (2021) has proposed CUCB-RA. Specifically, given an offline (α, β)-approximation Oracle $\mathcal{O}$, which output $\mathbf{a}_t^{\mathcal{O}}$ s.t. $p(r(\mathbf{a}_t^{\mathcal{O}}, \mathbb{P}(\mathbf{X})) \geq \alpha \sup_{\mathbf{a}_t} r(\mathbf{a}_t, \mathbb{P}(\mathbf{X}))) \geq \beta$. Then, we can measure the performance of π by the approximation regret for T rounds by

$$Reg_\alpha^\pi(T, \mathbb{P}(\mathbf{X})) = \alpha T \sup_{\mathbf{a}_t} r(\mathbf{a}_t, \mathbb{P}(\mathbf{X})) - \sum_{t=1}^{T} r(\mathbf{a}_t^\pi, \mathbb{P}(\mathbf{X})). \tag{3}$$

Algorithm 1 ϵ-greedy with offline oracle $\mathcal{O}$

Input: Budget Q, Oracle $\mathcal{O}$.
for $(k, a) \in \mathcal{S}$ **do**
  $T_{k,a} \leftarrow 0$ {total number of times arm (k, a) is played}.
  $\hat{\mu}_{k,a} \leftarrow 0$ {empirical mean of $f_k(a, X_k)$}.
end for
for $t = 1 \rightarrow \infty$ **do**
  $\mathbf{a}_t \leftarrow \mathcal{O}((\hat{\mu}_{k,a})_{(k,a)\in\mathcal{S}}, Q)$.
  Shuffle $\mathbf{a}_t$ with probability ϵ.
  Take allocation $\mathbf{a}_t$, observe feedback $f_k(a_{k,t}, X_{k,t})$'s.
  for $k \in [K]$ **do**
    $T_{k,a_{k,t}} \leftarrow T_{k,a_{k,t}} + 1$.
    $\hat{\mu}_{k,a_{k,t}} \leftarrow \hat{\mu}_{k,a_{k,t}} + (f_k(a_{k,t}, X_{k,t}) - \hat{\mu}_{k,a_{k,t}})/T_{k,a_{k,t}}$.
  end for
end for

Classical algorithms. Consider a discrete case, i.e., $\mathcal{A}_d = \{0, 1, \cdots, N-1\}$, $|\mathcal{A}_d| = N = Q + 1$, $\mathbf{a}_t \in \{\mathbf{a}_t | a_{k,t} \in \mathcal{A}_d, \sum_k a_{k,t} = Q\}$, the algorithm is based on the set base arm $\mathcal{S} = \{(k, a) | k \in [K], a \in \mathcal{A}_d\}$, $|\mathcal{S}| = KN$. For each $(k, a) \in \mathcal{S}$, let the expected reward of playing (k, a) be

$$\mu_{k,a} = \mathbb{E}_{X_{k,t} \overset{\text{i.i.d.}}{\sim} \mathbb{P}(X_k)} [f_k(a, X_{k,t})], \mu = (\mu_{k,a})_{(k,a)\in\mathcal{S}}. \tag{4}$$

Then, the expected total reward of $\mathbf{a}_t$ and μ is

$$r(\mathbf{a}_t, \mu) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] = \sum_{k=1}^{K} \sum_{a\in\mathcal{A}_d} \mu_{k,a} \mathbb{I}\{a_{k,t} = a\}. \tag{5}$$

Following Eq. 5, we can show the classical Greedy and Upper Confidence Bound algorithms that achieve $\mathcal{O}(\log(T))$ regret (Zuo & Joe-Wong, 2021), to balance the trade-off between exploration and exploitation (Sutton & Barto, 2018; Chen et al., 2013). We summarize the Greedy method in Alg. 1 and CUCB-RA method in Alg. 2.

Algorithm 2 CUCB-RA with offline oracle $\mathcal{O}$

Input: Budget Q, Oracle $\mathcal{O}$.
for $(k, a) \in \mathcal{S}$ **do**
  $T_{k,a} \leftarrow 0$ {total number of times arm (k, a) is played}.
  $\hat{\mu}_{k,a} \leftarrow 0$ {empirical mean of $f_k(a, X_k)$}.
end for
for $t = 1 \rightarrow \infty$ **do**
  for $(k, a) \in \mathcal{S}$ **do**
    $\rho_{k,a} \leftarrow \sqrt{\frac{3 \ln t}{2 T_{k,a}}}$ {confidence radius}.
    $\bar{\mu}_{k,a} \leftarrow \hat{\mu}_{k,a} + \rho_{k,a}$ {upper confidence bound}.
  end for
  $\mathbf{a}_t \leftarrow \mathcal{O}((\bar{\mu}_{k,a})_{(k,a)\in\mathcal{S}}, Q)$.
  Take allocation $\mathbf{a}_t$, observe feedback $f_k(a_{k,t}, X_{k,t})$'s.
  for $k \in [K]$ **do**
    $T_{k,a_{k,t}} \leftarrow T_{k,a_{k,t}} + 1$.
    $\hat{\mu}_{k,a_{k,t}} \leftarrow \hat{\mu}_{k,a_{k,t}} + (f_k(a_{k,t}, X_{k,t}) - \hat{\mu}_{k,a_{k,t}})/T_{k,a_{k,t}}$.
  end for
end for

Neural Network algorithm. Leveraging the development of Neural Network, Hwang et al. (2023) has recently proposed Neural Combinatorial Bandits (NCB) that can outperform the aforementioned classical algorithms. Specifically, based on CUCB-RA, NCB use a Neural Network NN_θ, parameterized by θ to map from previous context X_{t-1} to approximate directly the empirical mean of $f_k(a, X_{k,t})$. Then, after every step t, it updates θ based on cumulative data, i.e., $\{(X_j, \mu_j)\}_{j=0}^{t}$ by using Stochastic Gradient Descent with Mean Square Error (Sutton & Barto, 2018). We summarize NCB algorithm in Alg. 3.

Algorithm 3 NCB with offline oracle $\mathcal{O}$

Input: Budget Q, Oracle $\mathcal{O}$, Context $X_0 \leftarrow 0$.
for $(k, a) \in \mathcal{S}$ **do**
  $T_{k,a} \leftarrow 0$ {total number of times arm (k, a) is played}.
end for
for $t = 1 \rightarrow \infty$ **do**
  $\hat{\mu} \leftarrow \text{NN}_\theta(X_{t-1})$ {compute empirical mean}.
  for $(k, a) \in \mathcal{S}$ **do**
    $\rho_{k,a} \leftarrow \sqrt{\frac{3 \ln t}{2 T_{k,a}}}$ {confidence radius}.
    $\bar{\mu}_{k,a} \leftarrow \hat{\mu}_{k,a} + \rho_{k,a}$ {upper confidence bound}.
  end for
  $\mathbf{a}_t \leftarrow \mathcal{O}((\bar{\mu}_{k,a})_{(k,a)\in\mathcal{S}}, Q)$.
  Take allocation $\mathbf{a}_t$, observe feedback $f_k(a_{k,t}, X_{k,t})$'s.
  for $k \in [K]$ **do**
    $T_{k,a_{k,t}} \leftarrow T_{k,a_{k,t}} + 1$.
  end for
  $\theta \leftarrow \theta - \nabla_\theta \mathbb{E}_{\{(X_j,\mu_j)\}_{j=0}^{t}} [\text{NN}_\theta(X_j) - \mu_j]^2$
  {update NN to estimate empirical mean}.
end for

Our proposed method. The NCB algorithm, however, needs to update the Neural Network model with all of the cumulative data after every step. This is at the price of a slow speed for a SDUA algorithm, limiting the scalability in real-time V2X applications in the real world. To tackle this challenge, we propose Alg. 4, which only updates the Neural Network model with the latest data after every step (colored by blue). This reduces significantly the computational cost of the previous NCB algorithm, and we will empirically show that it still has a competitive result with NCB in the next following section.

3. Experiments

Experimental setting. Following Gupta et al. (2022), we compare methods by extracting the data from a real-world V2X-SDUA dataset. Specifically, we use the RF Jamming Dataset for Vehicular Wireless Networks (Pajevic et al., 2022), and we choose $K = 4$ APs, i.e., $\{91, 92, 94, 95\}$ on the 3-rd floor of Building 3 on campus, and record their associated users from 13:00 to 16:00 on March 2, 2015. We set $Q = 8$, i.e., allocate 8 new users to the four aforementioned APs. Then, we compare Greedy Alg. 1 with $\epsilon = 0$ and $\epsilon = 0.1$, CUCB-RA Alg. 2, NCB Alg. 3, and our Alg. 4 on three criteria, including cumulative regret in Eq. 3 (lower is better), cumulative reward in Eq. 2 (higher is better), and latency for each t step (lower is better). Source code and data are provided in the attached file of this report.

Algorithm 4 Our proposed algorithm with offline oracle $\mathcal{O}$

Input: Budget Q, Oracle $\mathcal{O}$, Context $X_0 \leftarrow 0$.
for $(k, a) \in \mathcal{S}$ **do**
  $T_{k,a} \leftarrow 0$ {total number of times arm (k, a) is played}.
end for
for $t = 1 \rightarrow \infty$ **do**
  $\hat{\mu} \leftarrow \text{NN}_\theta(X_{t-1})$ {compute empirical mean}.
  for $(k, a) \in \mathcal{S}$ **do**
    $\rho_{k,a} \leftarrow \sqrt{\frac{3 \ln t}{2 T_{k,a}}}$ {confidence radius}.
    $\bar{\mu}_{k,a} \leftarrow \hat{\mu}_{k,a} + \rho_{k,a}$ {upper confidence bound}.
  end for
  $\mathbf{a}_t \leftarrow \mathcal{O}((\bar{\mu}_{k,a})_{(k,a)\in\mathcal{S}}, Q)$.
  Take allocation $\mathbf{a}_t$, observe feedback $f_k(a_{k,t}, X_{k,t})$'s.
  for $k \in [K]$ **do**
    $T_{k,a_{k,t}} \leftarrow T_{k,a_{k,t}} + 1$.
  end for
  $\theta \leftarrow \theta - \nabla_\theta [\text{NN}_\theta(X_t) - \mu_t]^2$
  {update NN to estimate empirical mean}.
end for

Allocation performance. Fig. 2 and Fig. 3 show the cumulative regret and cumulative reward respectively over $T = 2000$ steps. Firstly, we observe that the CUCB-RA algorithm has a better performance than the greedy-based approaches by having a lower regret and higher reward. Secondly, the recent NCB outperforms classical methods when the step increases. Thirdly, our proposed approach has a competitive result with NCB in terms of allocation performance. Finally, the cumulative regret figure confirms the Theorem that these algorithms achieve $\mathcal{O}(\log(T))$ regret.

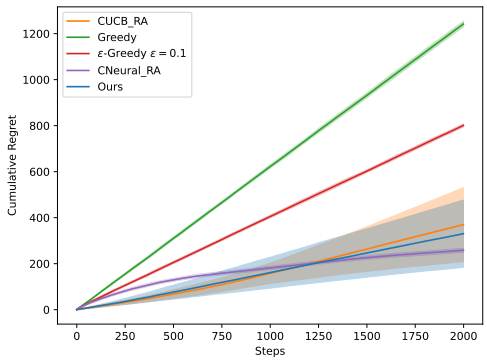

Figure 2. Cumulative regret with the Sequential Dynamic User Allocation setting across 10 different runs.

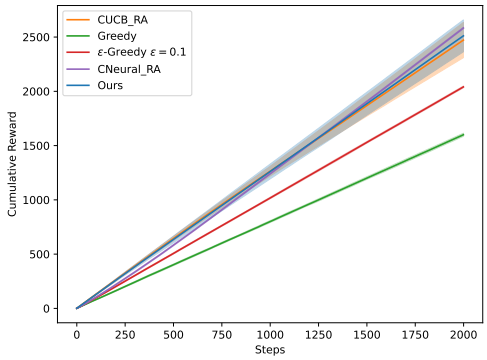

Figure 3. Cumulative reward with the Sequential Dynamic User Allocation setting across 10 different runs.

Efficiency performance. Fig. 4 compares the efficiency between algorithms in terms of latency for each step t. Although NCB outperforms classical methods in terms of allocation performance, it suffers from high latency. This latency is monotonic increasing by Alg. 3 needs to update cumulative data after every step. Conversely, our method only updates the latest data, resulting in much lower latency than NCB, and almost as efficient as classic algorithms.

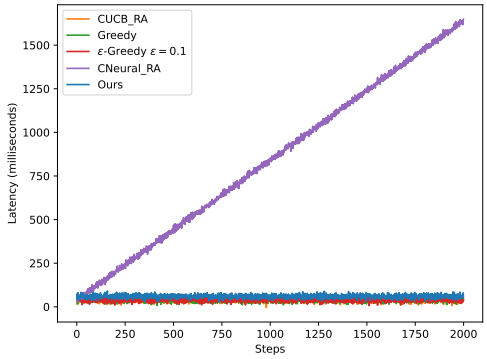

Figure 4. Latency performance with the Sequential Dynamic User Allocation setting across 10 different runs.

4. Conclusion

In this project, we reimplement the SDUA experiment, then compare greedy-based, CUCB-RA, and NCB methods. We observe that NCB performs better than classical approaches regarding allocation performance. However, NCB is not efficient. Therefore, we propose an efficient version that balances allocation and efficient performance. We show the proposed method works well in practice with the SDUA experiment. We hope this project will be a foundation for researchers to develop efficient Neural Nets for real-time resource allocation V2X application in the real world.

References

Abramson, N. The aloha system: Another alternative for computer communications. In *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*. Association for Computing Machinery, 1970.

Chen, W., Wang, Y., and Yuan, Y. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Chen, W., Wang, Y., Yuan, Y., and Wang, Q. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 2016.

Christopoulou, M., Barmpounakis, S., Koumaras, H., and Kaloxylos, A. Artificial intelligence and machine learning as key enablers for v2x communications: A comprehensive survey. *Vehicular Communications*, 2023.

Gao, J., Khandaker, M. R. A., Tariq, F., Wong, K.-K., and Khan, R. T. Deep neural network based resource allocation for v2x communications. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019.

Gupta, S., Zuo, J., Joe-Wong, C., Joshi, G., and Yağan, O. Correlated combinatorial bandits for online resource allocation. In *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2022.

Hwang, T., Chai, K., and Oh, M.-H. Combinatorial neural bandits. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Pajevic, L., Karlsson, G., and Fodor, V. Crawdad kth/campus, 2022. URL `https://ieee-dataport.org/collections/crawdad`.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Zuo, J. and Joe-Wong, C. Combinatorial multi-armed bandits for resource allocation, 2021.

Acronyms

AP Access Point. 1–3

CMAB Combinatorial Multi-Armed Bandit. 1, 2

CUCB-RA Combinatorial Upper Confidence Bound-Resource Allocation. 1–4

NCB Neural Combinatorial Bandits. 1, 3, 4

SDUA Sequential Dynamic User Allocation. 1–4

V2X Vehicle-to-everything. 1, 3, 4