Combinatorial Multi-Armed Bandit for Sequential Resource Allocation

Ha Manh Bui [1]

Abstract

We study the Sequential Resource Allocation (SRA) problem, i.e., how to find an optimal decision of a fixed budget to minimize the allocation cost over time. Under the Combinatorial Multi-Armed Bandit (CMAB) setting, this problem has been formalized and solved by the Combinatorial Upper Confidence Bound Resource Allocation (CUCB-RA) algorithm. That said, previous works mainly focus on theoretical guarantees without empirical contributions. Motivated by this approach, in this project, we will reimplement and extend empirical results for different applications in healthcare and computer networking domains. After that, we propose a density CUCB-RA algorithm to enhance uncertainty quality and robust performance under distribution shifts. Finally, we show our current result with Neural Combinatorial Bandits on the healthcare dataset.

1. Introduction

The SRA has become a major problem in plenty of real-world domains, e.g., economics, healthcare, computer science, etc. This SRA problem can be converted to the CMAB setting (Chen et al., 2013; 2016), in which given K arms, our goal is to select a super-arm, i.e., an action vector $\mathbf{a} = (a_1, \cdots, a_K)$, with a constraint on the budget Q to maximize the expected total reward, i.e.,

$$\max_{(a_1,\cdots,a_K)} \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right], \text{ s.t. } \sum_{k=1}^{K} a_k \leq Q, \quad (1)$$

where $f_k(a_k, X_k)$ represent the reward function of arm k-th at budget allocation a_k with a random noise $X_k \overset{\text{i.i.d.}}{\sim} \mathbb{P}(X_k)$.

For instance, in the healthcare application in Fig. 1, assume that there are K different hospitals and $X_k \overset{\text{i.i.d.}}{\sim} \mathbb{P}(X_k)$ available Intensive Care Unit (ICU) beds in the hospital k-th. Everyday, assume X_k are unobserved, and we need to find a strategy to allocate Q COronaVIrus Disease (COVID) patients to K hospitals by an action vector $\mathbf{a} = (a_1, \cdots, a_K)$,

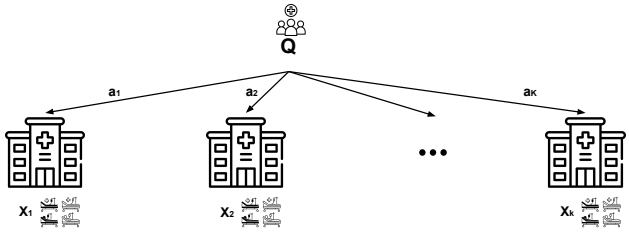

Figure 1. SRA in Healthcare by allocating Q COVID patients to K hospitals s.t. can maximize the number of rescued patients.

where $\sum_{k=1}^{K} a_k = Q$, such that can maximize the number rescued patients $\mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right]$.

Under the CMAB approach, Zuo & Joe-Wong (2021) has proposed the CUCB-RA algorithm to balance between exploration and exploitation, then proved regret bounds for it with an offline (α, β)-approximation Oracle $\mathcal{O}$. However, this work does not provide any empirical results. Gupta et al. (2022) is an extension of this direction with experiments on computer networking domains. Yet, this paper does not publish the source code for re-implementation.

Motivated by the CUCB-RA algorithm and toward reducing the gap between theoretical and empirical aspects, in this project:

1. We reimplement the computer networking experiments (Gupta et al., 2022).
2. We additionally conduct a healthcare application in the mentioned example with COVID patients allocation.
3. Following the CUCB-RA algorithm and based on density approach (Bui & Liu, 2023), we propose a safety version that can improve the uncertainty and robustness when $\mathbb{P}(X_k)$ is shifted over the time t.
4. Motived by the recent Neural network approaches (Wen et al., 2015; Hwang et al., 2023), we show our current result with Neural Combinatorial Bandits (Hwang et al., 2023) in the healthcare example.

2. The CUCB-RA algorithm

Notation and CMAB setting. Let consider an online version, at each time t, we allocate $a_{k,t}$ to k s.t. $\sum_{k=1}^{K} a_{k,t} \leq Q$. Then, we observe feedback reward $f_k(a_{k,t}, X_{k,t}), \forall k$,

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

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 in CMAB, 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 $r(\mathbf{a}_t, \mathbb{P}(\mathbf{X})) = \mathbb{E}[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})]$, 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 Combinatorial Upper Confidence Bound (CUCB) (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

$$Reg_{\alpha}^{\pi}(T, \mathbb{P}(\mathbf{X})) = \alpha T \sup_{\mathbf{a}_t} r(\mathbf{a}_t, \mathbb{P}(\mathbf{X})) - \sum_{t=1}^{T} r(\mathbf{a}_t^{\pi}, \mathbb{P}(\mathbf{X})). \tag{2}$$

Consider a discrete case, i.e., $\mathcal{A}_d = \{0, 1, \cdots, N-1\}$, $|\mathcal{A}_d| = N \leq Q+1$, $\mathbf{a}_t \in \{\mathbf{a}_t | a_{k,t} \in \mathcal{A}_d, \sum_k a_{k,t} \leq Q\}$, 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

$$\mu_{k,a} = \mathbb{E}_{X_{k,t} \overset{\text{i.i.d.}}{\sim} \mathbb{P}(X_k)} [f_k(a, X_{k,t})], \mu = (\mu_{k,a})_{(k,a) \in \mathcal{S}}. \tag{3}$$

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

$$r(\mathbf{a}_t, \mu) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] = \sum_{k=1}^{K} \sum_{a \in \mathcal{A}_d} \mu_{k,a} \mathbb{I}\{a_{k,t} = a\}. \tag{4}$$

Following Eq. 4, we can show an Upper Confidence Bound (UCB) algorithm that achieves $\mathcal{O}(\log(T))$ regret by the following Theorem (Zuo & Joe-Wong, 2021):

Theorem 2.1. *For the CUCB-RA algorithm on an online DRA problem with a bounded smoothness constant $B \in \mathbb{R}^+$, let the reward gap $\Delta_{\mathbf{a}} = \max\left(0, \alpha \cdot \sup_{\mathbf{a}_t} r(\mathbf{a}_t, \mathbb{P}(\mathbf{X})) - r(\mathbf{a}, \mathbb{P}(\mathbf{X}))\right)$, $\Delta_{\min}^{k,a} = \inf_{\mathbf{a} \in \mathcal{A}_d^k : a_k = a, \Delta_a > 0} \Delta_a$, $\Delta_{\max}^{k,a} = \sup_{\mathbf{a} \in \mathcal{A}_d^k : a_k = a, \Delta_a > 0} \Delta_a$, $\Delta_{\min} = \min_{(k,a) \in \mathcal{S}} \Delta_{\min}^{k,a}$, $\Delta_{\max} = \max(k,a) \in \mathcal{S} \Delta_{\max}^{k,a}$, then we have:*

1. *If $\Delta_{\min} > 0$, the distribution-dependent bound is*

$$Reg_{\alpha,\beta}(T, \mathbb{P}(X)) \leq \sum_{(k,a) \in \mathcal{S}} \frac{48 B^2 Q \ln T}{\Delta_{\min}^{k,a}} + 2BKN + \frac{\pi^3}{3} KN \Delta_{\max}, \tag{5}$$

2. *and the distribution-independent bound is*

$$Reg_{\alpha,\beta}(T, \mathbb{P}(X)) \leq 14B\sqrt{QKNT \ln T} + 2BKN + \frac{\pi^3}{3} KN \Delta_{\max}. \tag{6}$$

Proof. The proof is based on Hoeffding's inequality with two standard properties (1) Monotonicity and (2) 1-Norm Bounded Smoothness of $r(\mathbf{a}, \mu)$. The full proof is provided in Apd. A of Zuo & Joe-Wong (2021)'s paper. □

Algorithm 1 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

From Theorem 2.1, we can easily apply the UCB algorithm to balance the trade-off between exploration and exploitation with confidence bound (Sutton & Barto, 2018; Chen et al., 2013). We summarize the CUCB-RA algorithm in Alg. 1.

3. Experiments

3.1. Re-implementation SRA in computer networking

We firstly implement and compare results of the greedy, ϵ-greddy, and CUCB-RA for the following SRA settings:

3.1.1. Dynamic User Allocation

We consider the Dynamic User Allocation (Gupta et al., 2022), with K wireless Acess Point (AP) and Q new incoming users at each round. Let X_k denote the number of existing users in each AP k and $a_k \in \mathbb{N}$ denote the number of new users allocated to it. Our goal is to maximize the total throughput of all AP, i.e.,

$$\max_{a_k} \sum_{k=1}^{K} 0.2(X_k + a_k) e^{-0.2(X_k + a_k)}, \text{ s.t. } \sum_{k=1}^{K} a_k = Q. \tag{7}$$

We extract the data from a real-world dataset (Pajevic et al., 2022). Specifically, $K = 4$ 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. Fig. 2 shows the cumulative regret (left)

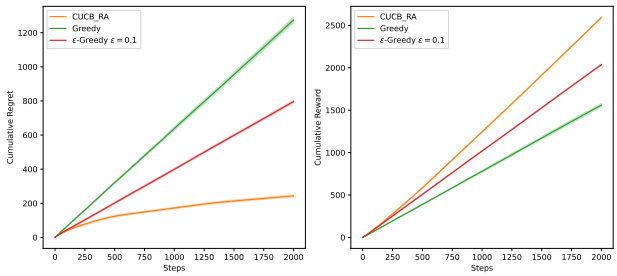

Figure 2. Cumulative Regret (left) and cumulative reward (right) with the Dynamic User Allocation setting across 10 different runs.

and cumulative reward (right) 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 cumulative regret figure confirm Theorem 2.1 that CUCB-RA achieves $\mathcal{O}(\log(T))$ regret.

3.1.2. Online Server Assignment

We consider the Online Server Assignment (Gupta et al., 2022), with K independent job streams with unknown expected job arrival rates λ_k and the realized job arrival rate $X_k \overset{\text{i.i.d.}}{\sim} \mathbb{U}(\lambda_k - 0.1, \lambda_k + 0.1)$. There are Q additional servers to be assigned with the number of additional servers allocated to stream k is $a_k \in \mathbb{N}$. Assume the service rate of all servers as 1, and our goal is to minimize the average expected waiting time of all job streams, i.e.,

$$\min_{a_k} \sum_{k=1}^{K} \frac{1}{1 - \frac{X_k}{a_k+1}} \frac{X_k}{\sum_{k=1}^{K} X_k}, \text{ s.t. } \sum_{k=1}^{K} a_k \leq Q. \quad (8)$$

We simulate the data with $K = 4$, $\lambda = \{0.2, 0.4, 0.6, 0.8\}$, and $Q = 8$. Fig. 3 shows the cumulative regret (left) and reward (right) respectively over $T = 5000$ steps. Unlike the previous setting, the greedy algorithm performs slightly better than the CUCB-RA by having a lower cumulative regret and higher reward. That said, CUCB-RA still has a better performance than ϵ-greedy. In addition, CUCB-RA also achieves $\mathcal{O}(\log(T))$ regret, confirming the result from Theorem 2.1.

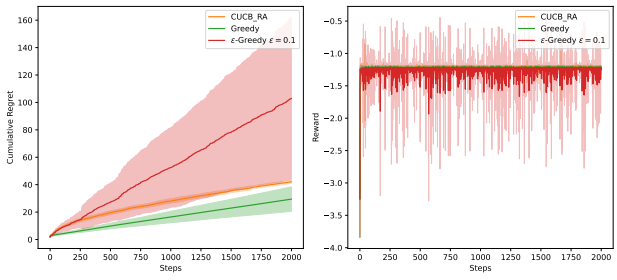

Figure 3. Cumulative Regret (left) and reward (right) with the Online Server Assignment setting across 10 different runs.

3.1.3. Online Water Filling

We consider the Online Water Filling (Boyd & Vandenberghe, 2004), with K communication channels and Q unit power, needs to be assigned in Orthogonal frequency-division multiplexing systems. Let X_k represent the floor above the baseline at which power can be added to the channel and $a_k \in \mathbb{R}^+$ represents the power allocated to channel k. The goal is to maximize the total throughput, i.e.,

$$\max_{a_k} \sum_{k=1}^{K} \log\left(X_k + a_k\right), \text{ s.t. } \sum_{k=1}^{K} a_k \leq Q. \quad (9)$$

We simulate the data with $K = 4$, for each channel k, we set the expectation $\mu_k = \mathbb{E}_k[X_k] \overset{\text{i.i.d.}}{\sim} \mathbb{U}(0.8, 1.2)$, and the realization of $X_k \overset{\text{i.i.d.}}{\sim} \mathbb{U}(\mu_k - 0.1, \mu_k + 0.1)$. We set $Q = 1$ with discretization granularity 0.2, i.e, $\mathcal{A}_d = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Similar to the Dynamic user allocation setting, Fig. 4 shows CUCB-RA achieves the best performance by having the lowest cumulative regret and highest cumulative reward in this Online water filling setting over $T = 5000$ steps. Additionally, it also achieves $\mathcal{O}(\log(T))$ regret, continuing confirm the result from Theorem 2.1.

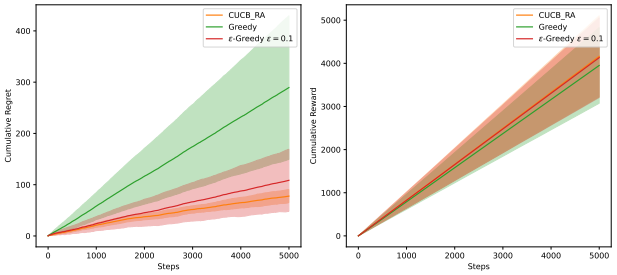

Figure 4. Cumulative Regret (left) and cumulative reward (right) with the Online Water Filling setting across 10 different runs.

3.2. SRA in healthcare domain

We additionally consider a new setting with the COVID patients allocation (Parker et al., 2020), as mentioned in the Introduction section. Specifically, assume the number of available ICU beds X_k at the hospital k-th are unobserved, and we need to allocate $a_k \in \mathbb{N}$ COVID patients to hospital k-th to maximize the number of rescued patients. Following Parker et al. (2020), we can define the objective function by the number of overflows (i.e., #patients - #beds) in the

hospital, i.e.,

$$\min_{a_k} \sum_{k=1}^{K} \max\left(0, a_k - X_k\right), \text{ s.t. } \sum_{k=1}^{K} a_k = Q. \tag{10}$$

We deploy algorithms in the real world by using a Johns Hopkins COVID dataset (Parker et al., 2020), including $K = 22$ different Texas hospitals, X_k available ICU beds are recorded from April 12, 2020 to September 19, 2020. For every day, we set Q is sampled from a Normal distribution, i.e., $Q \overset{\text{i.i.d.}}{\sim} \mathcal{N}(400, 100)$. Similar to the previous experiments, from Fig. 5, we also observe that the CUCB-RA has the best performance by having the lowest logarithm regret and higher reward by saving around 200 COVID patients at the final day.

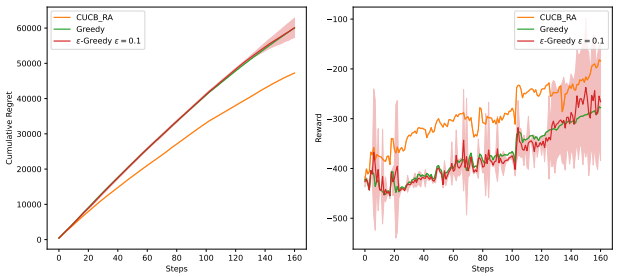

Figure 5. Cumulative Regret (left) and reward (right) with the COVID patients allocation setting across 10 different runs.

4. Future works

4.1. Improving CUCB-RA under distribution shifts

The CUCB-RA in Alg. 1 are assuming that the random noise is sampled independent and identically, i.e., $X_k \overset{\text{i.i.d.}}{\sim} \mathbb{P}(X_k)$. However, this is not practical, e.g., the number of available ICU beds can suddenly be changed by some unobserved reasons like other pandemics (nonstationary, distribution shifts). And if we follow the CUCB-RA, in a long time, it can exploit sub-optimal super-arm.

To avoid this issue, we propose a density CUCB-RA, which is motivated by the idea of using the density function to adjust the confidence intervals (Bui & Liu, 2023). Alg. 2 summarizes our algorithm, where the blue color represents our edits when compared to the CUCB-RA in Alg. 1. In particular, after every step, we will estimate a density function $p_\alpha(X_{k,t}|X_{k,t-j})$ on $\mathbf{X}$, with sliding window size j, and parameterized by α. Then, in the next step, the conditional likelihood from this density function will be used to adjust the exploration rate to control the upper confidence bound. Intuitively, we can see that with a high-density value, the exploration rate will be low. And, when there are distribution shifts, with a low-density value, the exploration rate will be high. This can help to avoid the exploiting sub-optimal super-arm issue of CUCB-RA in Alg. 1.

Algorithm 2 Density 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 $t = 1 \to \infty$ **do**
  for $(k, a) \in \mathcal{S}$ **do**
    $\rho_{k,a} \leftarrow \sqrt{\frac{3 \ln t}{2 T_{k,a}}}$ {confidence radius}.
    $\bar{\mu}_{k,a} \leftarrow \hat{\mu}_{k,a} + \frac{1}{p_\alpha(X_{k,t-1}|X_{k,t-j})}\rho_{k,a}$ {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
  Estimate density $p_\alpha(X_{k,t}, \cdots, X_{k,t-j})$.
end for

We compare our proposed algorithm with others in the Online Water Filling setting from Sec. 3.1.3. We use this dataset because we know the true underlying function $\mathbb{P}(\mathbf{X})$, so we can shift it easily. Specifically, we set $T = 10000$ steps, from time $t = 0 \to 5000$, we use the same setting which mentioned in Sec. 3.1.3. Then, at time $t = T/2 = 5000$, we suddenly change the expectation of arm 3-th to $\mu_3 = 10$. Fig. 6 shows our Density CUCB-RA outperforms other methods and can avoid the exploiting sub-optimal arm issue. Since it only adjusts the exploration rate, it still achieves $\mathcal{O}(\log(T))$ regret from Theorem 2.1.

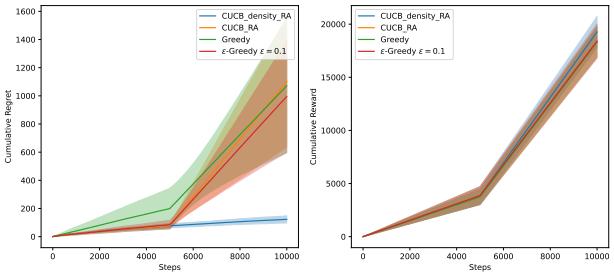

Figure 6. Cumulative Regret (left) and cumulative reward (right) with the Online Water Filling under distribution shifts setting across 10 different runs.

4.2. Contextual combinatorial Neural Bandit for SRA

So far, we can see that the CUCB-RA algorithm generally has a better performance, however, does not significantly

outperform greedy-based approaches. Motivated by the recent Neural Combinatorial Bandits (Hwang et al., 2023), which use a Neural Network with a context input to approximate reward function and also based on UCB algorithm, we finally show our current results on SRA in the healthcare domain. From Fig. 7, we can see that the Neural Network approach has not worked well yet by still quite similar to the CUCB-RA algorithm. The reason for this under-expected performance could be (1) the over-fitting reason and (2) we are considering SRA setting, which requires a constraint budget Q, so we must modify their initial algorithm a bit. We will leave this direction for our future work.

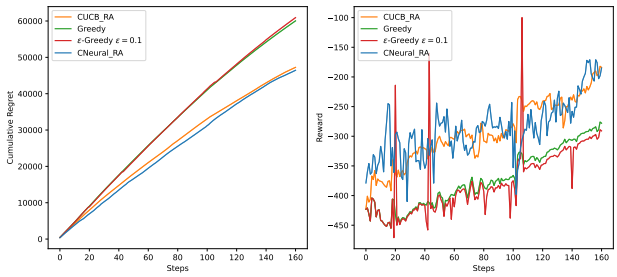

Figure 7. Cumulative Regret (left) and reward (right) with the COVID patients allocation setting with a Neural Net approach.

5. Conclusion

In this project, we reimplement the CUCB-RA algorithm on computer networking applications, then compare them with the greedy-based methods. We also extend for COVID patients allocation in the real-world healthcare domain. We observe that CUCB-RA generally performs better than greedy-based approaches. Additionally, we propose a density CUCB-RA algorithm, which can improve performance under distribution shifts. In future work, we will continue to discover provable Neural Bandit approaches to enhance SRA performance. We hope this project will be a foundation to develop safe and reliable Combinatorial Neural Bandit algorithms to help people in high-stake SRA applications in the real world.

References

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

Bui, H. M. and Liu, A. Density-softmax: Scalable and distance-aware uncertainty estimation under distribution shifts, 2023.

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.

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.

Parker, F., Sawczuk, H., Ganjkhanloo, F., Ahmadi, F., and Ghobadi, K. Optimal resource and demand redistribution for healthcare systems under stress from covid-19, 2020.

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

Wen, Z., Kveton, B., and Ashkan, A. Efficient learning in large-scale combinatorial semi-bandits. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

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

Acronyms

AP Acess Point. 2

CMAB Combinatorial Multi-Armed Bandit. 1, 2

COVID COronaVIrus Disease. 1, 3–5

CUCB Combinatorial Upper Confidence Bound. 2

CUCB-RA Combinatorial Upper Confidence Bound Resource Allocation. 1–5

ICU Intensive Care Unit. 1, 3, 4

SRA Sequential Resource Allocation. 1–5

UCB Upper Confidence Bound. 2, 5