Combinatorial Multi-Armed Bandit for Sequential Resource Allocation

Ha Manh Bui

Johns Hopkins University

Problem and motivation

- The Combinatorial Multi-Armed Bandit (CMAB) was firstly introduced by Chen et al. (2013).
- Unlike the standard Bandit, at time t, we need to select a **supper arm**, i.e., an action vector $(a_{1,t}, \cdots, a_{K,t}) \in \mathbb{R}^K, a_{i,t} \in \{0,1\}$, $\forall i \in [K]$, s.t. can maximize the expected total reward.
- Under CMAB setting, Zuo & Joe-Wong (2021) has extend for Sequential Resouce Application (RA) application with budget Q as follows

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

  where $f_k(a_k, X_k)$ represent the reward function of arm k-th at budget allocation a_k with a random noise $X_k \overset{\text{i.i.d.}}{\sim} \mathbb{P}(X_k)$.
- Then, they proposed the Combinatorial Upper Confidence Bound Resource Allocation (CUCB-RA) algorithm with application in computer networking (Gupta et al., 2022), however, they do not publish the code.
- In this project, we (1) **reimplement** their experiments in computer networking applications, (2) **extend** to the **healthcare domain**, and (3) **in future work, can discover new algorithms** based on CUCB-RA.

The CUCB-RA Algorithm

- Recall, our goal is

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

- **CUCB-RA Algorithm** (Zuo & Joe-Wong, 2021). Consider the discrete case, i.e., $A_d = \{0, 1, \cdots, N-1\}$, $|A_d| = N \leq Q+1$, $\mathbf{a}_i \in \{\mathbf{a}_i | a_{k,t} \in A_d, \sum_k a_{k,t} \leq Q\}$, the algorithm is based on the set base arm $S = \{(k,a) | k \in [K], a \in A_d\}$, $|S| = KN$.
- For each $(k,a) \in S$, let the expected reward of playing (k,a) be

$$\mu(k,a) = \mathbb{E}_{X_{k,t} \sim \mathbb{P}(X_k)}\left[\sum_k f_k(X_{k,t}, a_{k,t})\right], \mu = (\mu_{k,a})_{(k,a)\in S}. \quad (2)$$

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

$$r(\mathbf{a}_t, \mu) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] = \sum_{k=1}^{K} \sum_{a \in A_d} \mu_{k,a} \mathbb{I}\{a_{k,t} = a\}. \quad (3)$$

- Based on Eq. (3), we can apply the UCB (Sutton & Barto, 2018) algorithm that achieves $mat\hbar cal O(\log(T))$ regret, where T is the number of rounds played, to balance the trade-off between exploration and exploitation with confidence bound.

Algorithm 1 CUCB-RA with offline oracle $\mathcal{O}$

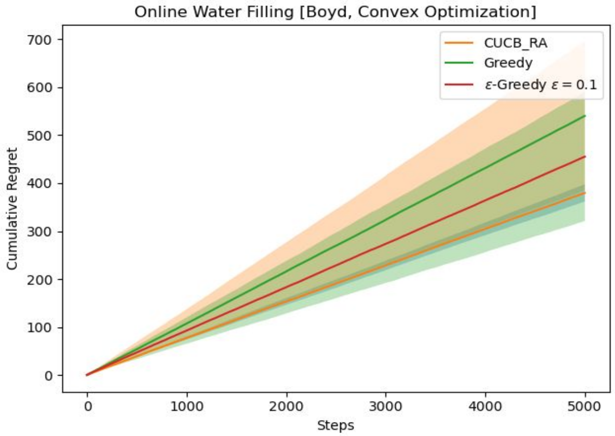
Input: Budget Q, Oracle $\mathcal{O}$.
for $(k,a) \in \mathcal{S}$ **do**
  $T_{k,a} \leftarrow 0$ {total number of times arm (k,a) is played}.
  $\hat{\mu}_{k,a} \leftarrow 0$ {empirical mean of $f_k(a, X_k)$}.
end for
for $t = 1 \rightarrow \infty$ **do**
  for $(k,a) \in \mathcal{S}$ **do**
    $\rho_{k,a} \leftarrow \sqrt{\frac{3\ln t}{2T_{k,a}}}$ {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

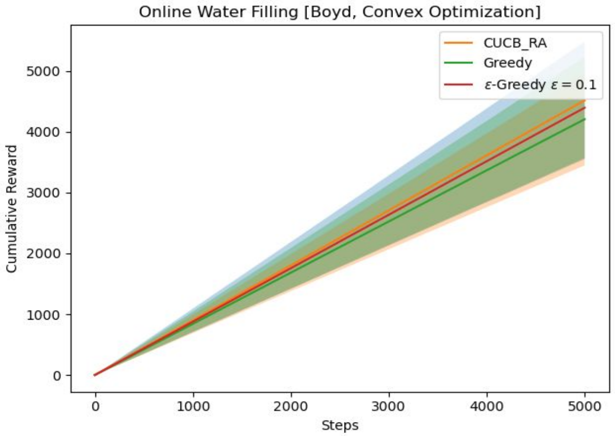
(1) Results in Computer Networking

- **Online Water Filling** (Boyd & Vandenberghe, 2004). We consider 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.,

$$\max_{a_k} \sum_{k=1}^{K} \log(X_k + a_k), \text{ s.t. } \sum_{k=1}^{K} a_k \leq Q. \qquad (4)$$

- **Simulation**. Set $K = 4$, $Q = 1$, $T = 5000$, $\mathbb{E}[X_k] \sim \mathbb{U}(0.8, 1.2)$, and $X_k \sim \mathbb{U}(\mathbb{E}[X_k] - 0.1, \mathbb{E}[X_k] + 0.1)$.
- **Observation**. CUCB-RA outperforms Greedy and ϵ-Greedy across 10 random seeds.





(2) Application in Healthcare domain

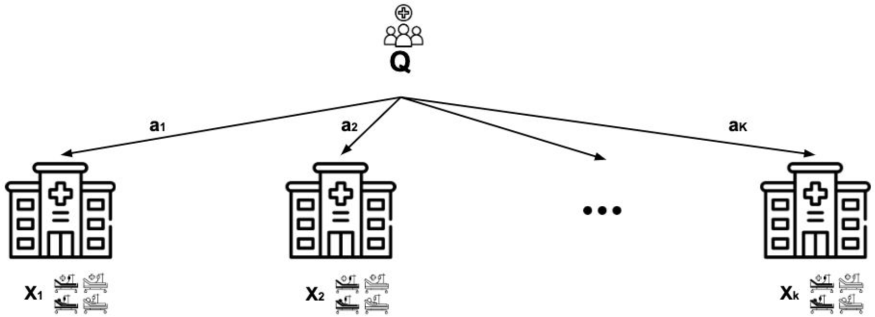


Figure 1. Sequential Resource Allocation in Healthcare domain.

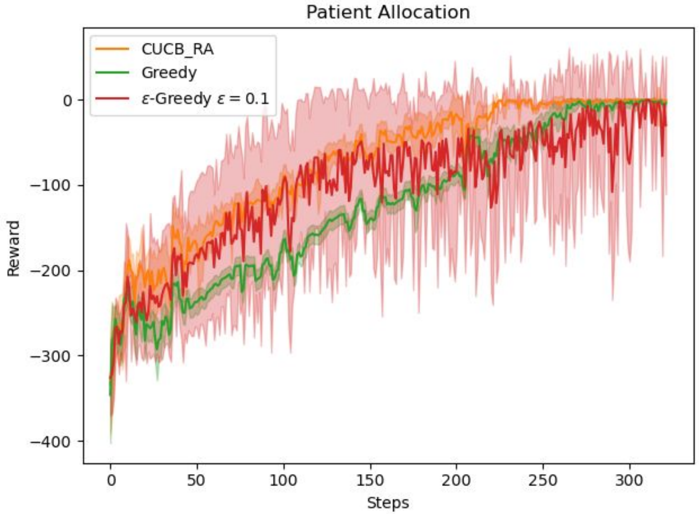
- Consider K different hospitals, at each time t, there are $X_{k,t} \sim \mathbb{P}_{X_k}$ available ICU beds in the hospital k-th.
- Assume at time t, $X_{k,t}$ are unobserved, and we need to find a strategy to allocate Q COVID patients to K hospitals by an action vector $(a_{1,t}, \cdots, a_{K,t})$, $\sum_{k=1}^{K} a_{k,t} = Q$, s.t. can maximize the number of rescued patients $\mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right]$.
- Following Parker et al. (2020), we can define the reward function by the number of overflow (#patient - #bed) in the hospital, i.e.,

$$f_k(a_{k,t}, X_{k,t}) = -max(0, a_{k,t} - X_{k,t}). \quad (5)$$

(2) Results in Healthcare domain

- We deploy on a small real-world dataset from https://jhu-covid-optimization.github.io/covid-data/ (Parker et al., 2020), including $K = 23$ different hospitals across $T = 163$ days, each day, there are $X_{k,t}$ available of ICU beds, and $Q \sim mathcalN(400, 100)$ patients.
- **Observation**. Similar to Computer Networking experiments, CUCB-RA outperforms Greedy and ϵ-Greedy across 10 random seeds.

TSA ID	TSA AREA	2020-04-12	2020-04-13	2020-04-14	2020-04-15	2020-04-16	2020-04-17	2020-04-18	2020-04-19
A.	Amarillo	77	78	67	54	92	89	83	57
B.	Lubbock	93	89	91	86	99	109	95	73
C.	Wichita Falls	20	18	19	19	21	18	18	23
D.	Abilene	36	36	27	13	23	26	20	23
E.	Dallas/Ft. Worth	623	603	612	745	723	745	655	765
F.	Paris	36	32	27	36	36	31	25	33
G.	Longview/Tyler	103	103	95	103	84	103	80	64
H.	Lufkin	35	33	28	26	22	26	18	21
I.	El Paso	73	67	86	82	93	79	89	85
J.	Midland/Odessa	42	37	45	45	41	38	43	51
K.	San Angelo	13	15	18	20	22	22	22	15
L.	Belton/Killeen	56	46	57	56	57	77	56	42
M.	Waco	26	21	26	23	23	21	21	12
N.	Bryan/College St	15	26	27	22	22	25	22	27
O.	Austin	144	156	164	174	178	169	173	172
P.	San Antonio	351	333	342	326	320	345	310	299
Q.	Houston	403	341	401	318	328	387	385	394

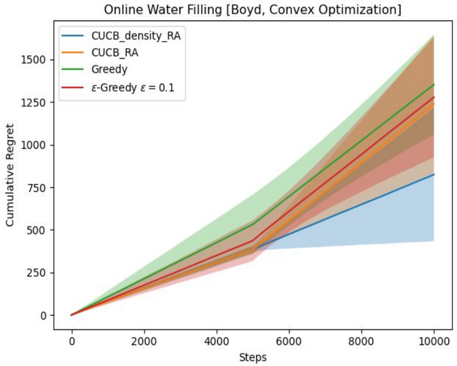


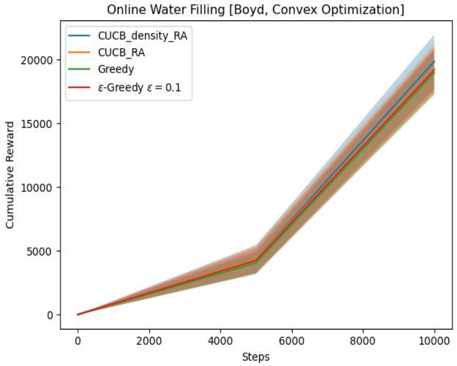
(3) CUCB-Density-RA Algorithm under distribution shifts

- **Motivation**:
  - We are assuming that the random noise $X_k \overset{\text{i.i.d.}}{\sim} \mathbb{P}(X_k)$, but it is not practical, e.g., the number of available ICU beds can suddenly be changed by some unobserved reasons like other pandemics (distribution shifts).
  - So, the previous CUCB-RA algorithm can exploit sub-optimal super arms.
- **Our idea**. Improve the CUCB-RA algorithm by adjust exploration rate to control the upper confidence bound, i.e.,
  - With high density, the exploration rate will be low.
  - With low density, the exploration rate will be high
- **Obervation**. Our CUCB-Density-RA version can **improve the performance under distribution shifts**.

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 $t = 1 \to \infty$ **do**
  for $(k, a) \in \mathcal{S}$ **do**
    $\rho_{k,a} \leftarrow \sqrt{\frac{3 \ln t}{2T_{k,a}}}$ {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



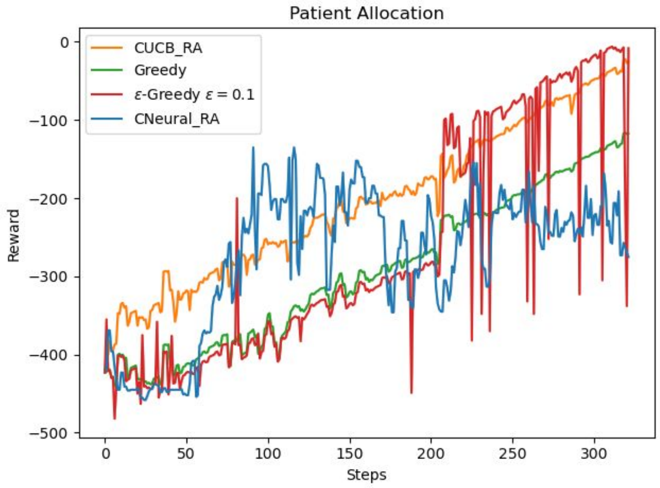


(3) New algorithms based on CUCB-RA

- **Motivation**:
  - The CUCB-RA does not really outperforms Greddy-based methods.
  - Try **Combinatorial Neural Bandits** (Hwang et al., 2023), which use **Neural Networks** to approximate reward function and also based on **CUCB algorithm**.
- But, we need to modify the algorithm to work in CUCB-RA, and it doesn't work yet...

Algorithm 1 Combinatorial Neural UCB (CN-UCB)

Input: Number of rounds T, regularization parameter λ, norm parameter B, step size η, network width m, number of gradient descent steps J, network depth L.
Initialization: Randomly initialize $\boldsymbol{\theta}_0$ as described in Section 3.1 ~~and~~ $\cancel{\mathbf{Z}_0 = \lambda \mathbf{I}}$
for $t = 1, ..., T$ **do**
  Observe $\{\mathbf{x}_{t,i}\}_{i \in [N]}$
  Compute $\hat{v}_{t,i} = f(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1})$ and $u_{t,i} = \hat{v}_{t,i} + \cancel{\gamma_{t-1} \|\mathbf{g}(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1})/\sqrt{m}\|_{\mathbf{Z}_{t-1}^{-1}}}$ for $i \in [N]$, Q
  Let $S_t = \mathbb{O}_S(\mathbf{u}_t \cancel{+ \mathbf{e}_t}, Q)$
  Play super arm S_t and observe $\{v_{t,i}\}_{i \in S_t}$
  ~~Update~~ $\cancel{\mathbf{Z}_t = \mathbf{Z}_{t-1} + \sum_{i \in S_t} \mathbf{g}(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1}) \mathbf{g}(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1})^\top / m}$
  Update $\boldsymbol{\theta}_t$ to minimize the loss in Eq.(4) using gradient descent with η for J times
  ~~Compute γ_t and e_{t+1} described in lemma 1~~ $\nabla_\theta \mathbb{E}_{(x_k, v_k)} \{ f(x_k, \theta), v_k \}^2$
end for



Summary

- We consider the Combinatorial Multi-Armed Bandit for Sequential Resource Allocation.
- We **reimplemnt** CUCB-RA and Greedy-based algorithm on Computer Networking dataset.
- We **extend to healthcare application**, and observe CUCB-RA performs quite well.
- We **study new algorithm** called CUCB-Density-RA version that **can improve the performance under distribution shifts**.
- Next, we want to try with Neural Networks approaches, e.g., **Combinatorial Neural Bandits** (Hwang et al., 2023) to enhance the exploration and exploitation performance.

Thank you for your attention!

References

- Boyd, S. and Vandenberghe, L. Convex Optimization. Cambridge University Press, 2004.
- 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.
- Zuo, J. and Joe-Wong, C. Combinatorial multi-armed bandits for resource allocation, 2021.
- Gupta, S., Zuo, J., Joe-Wong, C., Joshi, G., and Ya˘gan, 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.
- Sutton, R. S. and Barto, A. G. Reinforcement Learning: An Introduction. The MIT Press, second edition, 2018.
- 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.
- Hwang, T., Chai, K., and Oh, M.-H. Combinatorial neural bandits. In Proceedings of the 40th International Conference on Machine Learning, 2023.