IMPERIAL

GINO-Q: Learning an Asymptotically Optimal Index Policy for Restless Multi-armed Bandits

Gongpu Chen, Soung Chang Liew, Deniz Gündüz

Imperial College London & The Chinese University of Hong Kong
AAAI2026, Jan 2026

Restless Multi-armed Bandits (RMABs)

Problem Statement

- M Markov arms, each corresponding to an MDP $\mathcal{B}_i = (\mathcal{S}_i, \mathcal{A}_i, r_i, p_i)$, where
  - state space $\mathcal{S}_i$, a finite set
  - action space $\mathcal{A}_i = \{0, 1\}$
  - reward function $r_i(s, a)$
  - transition kernel $p_i(s_i^{t+1} | s_i^t, a_i^t)$
- all arms evolve independently based on their actions

$$P(s_1^{t+1}, \cdots, s_M^{t+1} | s_1^t, \cdots, s_M^t, a_1^t, \cdots, a_M^t) = \prod_{i=1}^{M} p_i(s_i^{t+1} | s_i^t, a_i^t)$$

- the arms are weakly coupled by a constraint: $\sum_{i=1}^{M} a_i^t = N$ for all t
  $\rightarrow$ only $N < M$ arms can be activated at each time
- Objective:

$$\max_{\{a_i^t : i \in [M], t \geq 1\}} \lim_{T \to \infty} \frac{1}{T} \mathbb{E} \left[\sum_{t=1}^{T} \sum_{i=1}^{M} r_i \left(s_i^t, a_i^t \right) \right]$$

Restless Multi-armed Bandits (RMABs)

An MDP with large space and action spaces

An RMAB can be formulated as an MDP with

- joint state space $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_{\mathsf{M}}$
- joint action space $\mathcal{A} = \{(\mathsf{a}_1, \mathsf{a}_1, \cdots, \mathsf{a}_{\mathsf{M}}) : \sum_{\mathsf{i}=1}^{\mathsf{M}} \mathsf{a}_{\mathsf{i}} = \mathsf{N}\}$

Restless Multi-armed Bandits (RMABs)

An MDP with large space and action spaces

An RMAB can be formulated as an MDP with

- joint state space $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_M \quad \Rightarrow \quad$ grow exponentially with M
- joint action space $\mathcal{A} = \{(a_1, a_1, \cdots, a_M) : \sum_{i=1}^{M} a_i = N\} \quad \Rightarrow \quad$ a combinatorial space of size $\binom{M}{N}$

Example: A moderate-scale RMAB with $M = 100, N = 25, |\mathcal{S}_i| = 10$. Then

$$|\mathcal{S}| = 10^{100}, |\mathcal{A}| = \binom{100}{25} = 2.4 \times 10^{23}$$

Hard to solve using standard RL methods!

SOTA solutions: Whittle-index-based RL methods $\rightarrow$ Learn the Whittle index policy in model-free setting

Whittle Index Policy

Decouple the RMAB across arms

- Relaxation

$$\sum_{i=1}^{M} a_i^t = N, \forall t \quad \rightarrow \quad \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} a_i^t\right] = N$$

- Lagrange Multiplier method

$$\max_{\{a_i^t\}} \inf_{\lambda} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{M}\left[r_i\left(s_i^t, a_i^t\right) - \lambda a_i^t\right]\right] + N\lambda$$

- Decomposition (for any fixed λ)

$$J_i(\lambda): \max_{\{a_i^t : t\geq 1\}} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\left[r_i\left(s_i^t, a_i^t\right) - \lambda a_i^t\right]\right]$$

Whittle Index Policy

Decompose the RMAB into single-arm problems

Each single-arm problem $J_i(\lambda)$ is an MDP associated with the arm $\mathcal{B}_i$

$$J_i(\lambda): \max_{\{a_i^t : t \geq 1\}} \lim_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} \left[r_i\left(s_i^t, a_i^t\right) - \lambda a_i^t\right]\right]$$

- state space $\mathcal{S}_i$, a finite set
- action space $\mathcal{A}_i = \{0, 1\}$
- reward function $r_i(s, a) - \lambda a$ $\quad \rightarrow \quad$ λ can be interpreted as the cost of action 1, called service charge
- transition kernel $p_i(s_i^{t+1} | s_i^t, a_i^t)$

Whittle Index Policy
Definition and Indexability

Whittle index policy

- Assign a Whittle index to each state of each arm:

$$\mathsf{WI}_i(s) = \inf\{\lambda : \mathsf{Q}_i(s, 1, \lambda) = \mathsf{Q}_i(s, 0, \lambda)\}.$$

  where $\mathsf{Q}_i(s, a, \lambda)$ is the Q-function of $\mathsf{J}_i(\lambda)$
  (Interpretation: $\mathsf{WI}_i(s)$ is the infimum service charge that makes the two actions equally optimal)
- At each time, select the top N arms with the highest Whittle indices

The Whittle index policy is only applicable to indexable RMABs

Indexability

- Denote by $\mathcal{E}_i(\lambda)$ the set of states in which action 0 is optimal for problem $\mathsf{J}_i(\lambda)$
- An arm $\mathcal{B}_i$ is considered indexable if $\mathcal{E}_i(\lambda)$ expands monotonically from the empty set to the entire state space $\mathcal{S}_i$ as λ increases from $-\infty$ to ∞.
- An RMAB is indexable if all its arms are indexable.

Whittle-index-based learning methods

Idea and limitation

Basic idea: Learn the Whittle index for each state of each arm in the model-free setting, and select the top N arms with highest indices at each time

- If an arm is indexable, there is a unique λ that satisfies

$$Q_i(s, 1, \lambda) = Q_i(s, 0, \lambda) \tag{1}$$

- learn a λ satisfying (1) for each state s of each arm, use it as the Whittle index

Limitations:

- The Whittle index policy is only applicable to indexable RMABs—NOT all RMABs are indexable
- Verifying indexability requires full knowledge of the system and considerable analytical effort

Existing Whittle-index-based learning methods assume the RMAB is always indexable, which is NOT true.

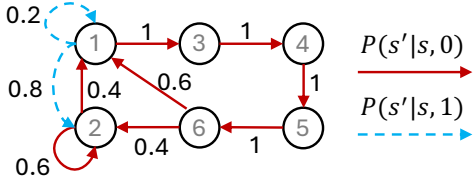
What if using Whittle-index-based methods in non-indexable RMABs?

Whittle-index-based learning methods

Counter example

A non-indexable arm

- Transition probabilities as shown in the right figure
- Reward function defined as
  $\mathsf{r}_\mathsf{i}(1,1) = -10, \mathsf{r}_\mathsf{i}(1,0) = -4, \mathsf{r}_\mathsf{i}(2,1) = \mathsf{r}_\mathsf{i}(2,0) = 4$
  and $\mathsf{r}_\mathsf{i}(\mathsf{s},1) = 0, \mathsf{r}_\mathsf{i}(\mathsf{s},0) = -2$ for $\mathsf{s} \in \{3,4,5,6\}$.

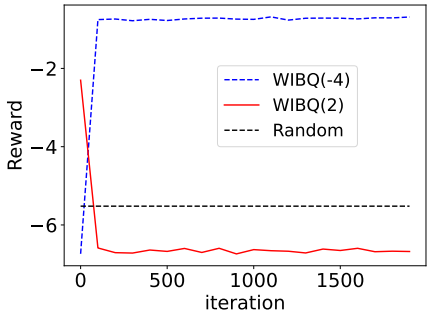


The arm is NOT indexable because

- state $1 \in \mathcal{E}_\mathsf{i}(\lambda)$ for $-4 \leq \lambda \leq 2$ and
- state $1 \notin \mathcal{E}_\mathsf{i}(\lambda)$ for $\lambda < -4$ and $\lambda > 2$

When learning the Whittle index for state 1, the algorithm may randomly converge to either 2 or -4

The performance can be arbitrarily poor!



It's necessary to develop a method that does not require indexability

Gain Index Policy

Decouple the RMAB across arms

- Relaxation

$$\sum_{i=1}^{M} a_i^t = N, \forall t \quad \rightarrow \quad \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{M} a_i^t\right] = N$$

- Lagrange Multiplier method

$$\max_{\{a_i^t\}} \inf_{\lambda} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{M}\left[r_i\left(s_i^t, a_i^t\right) - \lambda a_i^t\right]\right] + N\lambda,$$

- Equivalent transform:

$$\text{Relaxed RMAB: } \inf_{\lambda} \max_{\{a_i^t\}} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{M}\left[r_i\left(s_i^t, a_i^t\right) - \lambda a_i^t\right]\right] + N\lambda,$$

Gain Index Policy

Decompose the RMAB into single-arm problems

Each single-arm problem $J_i(\lambda)$ is an MDP associated with the arm $\mathcal{B}_i$

$$J_i(\lambda): \max_{\{a_i^t: t\geq 1\}} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\left[r_i\left(s_i^t, a_i^t\right) - \lambda a_i^t\right]\right]$$

- state space $\mathcal{S}_i$, a finite set
- action space $\mathcal{A}_i = \{0, 1\}$
- reward function $r_i(s, a) - \lambda a$ $\quad\rightarrow\quad$ λ can be interpreted as the cost of action 1, called service charge
- transition kernel $p_i(s_i^{t+1}|s_i^t, a_i^t)$

The Bellman optimality equation:

$$V_i(s, \lambda) + g_i(\lambda) = \max_{a\in\{0,1\}}\left\{r_i(s, a) - \lambda a + \sum_{s'\in\mathcal{S}_i} p_i(s'|s, a)V_i(s', \lambda)\right\},\ s \in \mathcal{S}_i,$$

Gain Index Policy

Decompose the RMAB into Single-arm Problems

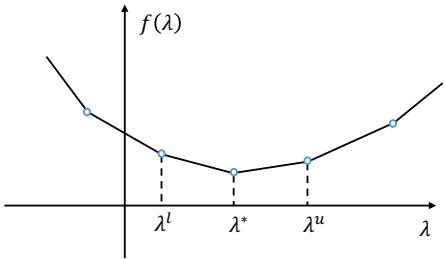
The relaxed RMAB problem reduces to

$$\inf_{\lambda} \mathsf{f}(\lambda) \triangleq \sum_{\mathsf{i}=1}^{\mathsf{M}} \mathsf{g}_{\mathsf{i}}(\lambda) + \mathsf{N}\lambda.$$

Denote by λ^* the optimal solution to the relaxed RMAB problem.

Lemma

For any RMAB with bounded reward functions $\{\mathsf{r}_{\mathsf{i}}\}_{\mathsf{i}\in[\mathsf{M}]}$, $\mathsf{f}(\lambda)$ is a piecewise linear and convex function. In addition, there always exists a bounded λ^* that achieves the minimum value of $\mathsf{f}(\lambda)$.



Gain Index Policy
Definition and Properties

For any $J_i(\lambda)$, the Q-function (state-action value function) is:

$$Q_i(s, a, \lambda) \triangleq r_i(s, a) - \lambda a + \sum_{s' \in \mathcal{S}_i} p_i(s'|s, a) V_i(s', \lambda).$$

- We express V_i and Q_i as functions of λ to highlight their dependence on λ.

Definition (Gain index policy)

For each arm $i \in [M]$ and each state $s \in \mathcal{S}_i$, a gain index is defined as:

$$W_i(s) \triangleq Q_i(s, 1, \lambda^*) - Q_i(s, 0, \lambda^*). \tag{2}$$

Then the gain index of the i-th arm at time t is given by $W_i(s_i^t)$. The gain index policy activates the N arms with the largest N gain indices, with ties broken arbitrarily.

Gain Index Policy

Definition and Properties

Theorem (Asymptotical Optimality, informal)

If N/M is fixed, then the gain index policy is asymptotically optimal in the following sense:

$$\lim_{M\to\infty} \frac{1}{M} G_M^{\text{ind}} = \lim_{M\to\infty} \frac{1}{M} G_M^{\text{opt}},$$

where G_M^{ind} and G_M^{opt} denote the cumulative reward of the RMAB under the gain index policy and the optimal policy, respectively.

Remarks:

- The gain index policy tends to be optimal when the number of arms is large enough
- Applicable to all RMABs $\rightarrow$ Does NOT require indexability
- Computing gain indices requires knowledge of the system

How to learn gain indices in the model-free setting?

GINO-Q

Learn the gain indices via Q-learning

Goal: Learn the gain index for each state s of each arm:

$$W_i(s) \triangleq Q_i(s, 1, \lambda^*) - Q_i(s, 0, \lambda^*).$$

- need to learn λ^* and the associated Q-functions
- λ^* is the minimizer of the convex function $f(\lambda)$

GINO-Q

Estimate the gradient of $f(\lambda)$

Definition (Auxiliary MDP)

The auxiliary MDP associated with arm $\mathcal{B}_i$ is defined as $\mathcal{M}_i = (\mathcal{S}_i, \mathcal{A}_i, c, p_i)$, where the state space $\mathcal{S}_i$, action space $\mathcal{A}_i$, and transition kernel p_i are identical to those of $\mathcal{B}_i$. The cost function is given by $c(s, a) = a$ for all $s \in \mathcal{S}_i$ and $a \in \mathcal{A}_i$.

For any policy π, the average cost h_i^π is determined by

$$D_i^\pi(s, a) + h_i^\pi = a + \sum_{s', a'} p_i(s'|s, a)\pi(a'|s')D_i^\pi(s', a').$$

$D_i^\pi(s, a)$ denotes the state-action value function of $\mathcal{M}_i$ under policy π.

$$\frac{dg_i^\pi}{d\lambda} = -h_i^\pi, \ \forall i \in [M]. \tag{3}$$

Let π_i^λ denote the optimal policy for problem $J_i(\lambda)$, then the derivative of function $f(\lambda)$ is given by

$$f'(\lambda) = \sum_{i=1}^{M} \frac{dg_i(\lambda)}{d\lambda} + N = N - \sum_{i=1}^{M} h_i^{\pi_i^\lambda}. \tag{4}$$

GINO-Q

A Three-timescale stochastic approximation method

- Slow timescale: stochastic gradient-decent

$$\lambda^{t+1} = \lambda^t - \theta^t(N - \sum_{i=1}^{M} h_i^t).$$

- Medium timescale: Q learning

$$Q_i(s_i^t, a_i^t) \leftarrow (1-\beta_i^t)Q_i(s_i^t, a_i^t) + \beta_i^t[r_i(s_i^t, a_i^t) - \lambda^t a_i^t + \max_a Q_i(s_i^{t+1}, a) - g_i^t].$$

- Fast timescale: SARSA learning to estimate $f'(\lambda)$

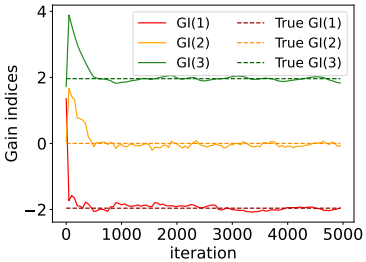
$$D_i(s_i^t, a_i^t) \leftarrow (1-\alpha_i^t)D_i(s_i^t, a_i^t) + \alpha_i^t[a_i^t + D_i(s_i^{t+1}, a_i^{t+1}) - h_i^t].$$

The timescales are controlled by the stepsizes:

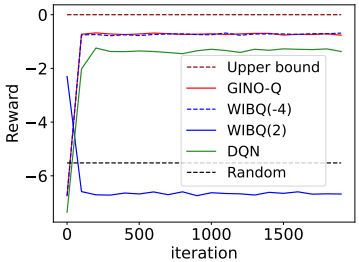
$$\alpha_i^t = \frac{C_1}{t}, \beta_i^t = \frac{C_2}{t\sqrt{\log t}}, \theta^t = \frac{C_3}{t \log t}\mathbf{1}\{t(\mathrm{mod}\, C_4) = 0\}$$

Experiments

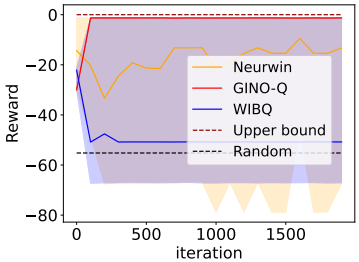
Group 1: A Non-indexable RMAB



Gain index over training



Reward over training
($\mathrm{M} = 10, \mathrm{N} = 7$)

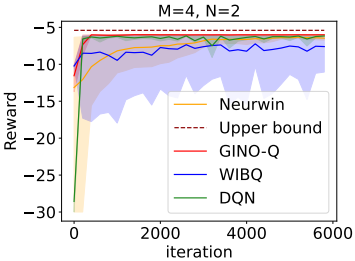


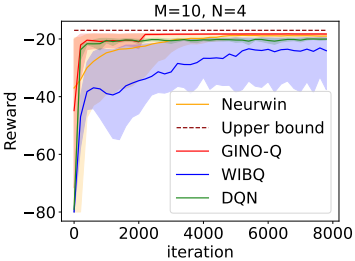
Reward over training
($\mathrm{M} = 100, \mathrm{N} = 70$)

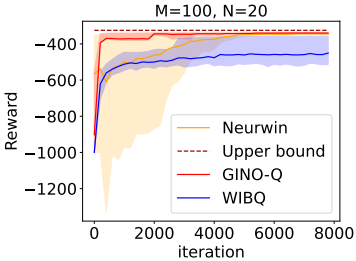
- GINO-Q indeed learns the gain indices
- While Whittle-index-based methods can perform poorly, GINO-Q always learn a near-optimal policy

Experiments

Group 2: Channel Allocation



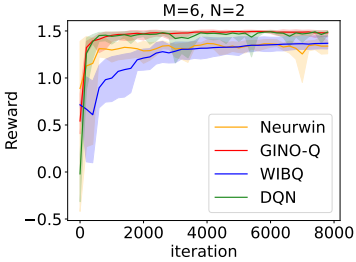


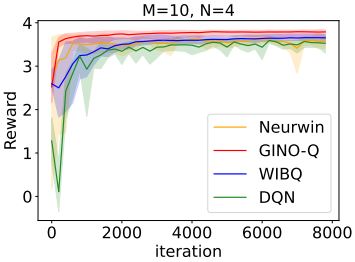


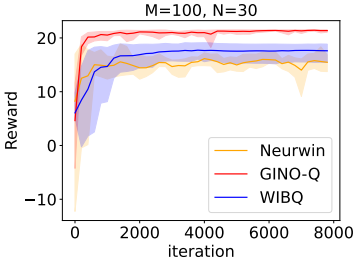
- Standard RL methods is infeasible when M is large
- GINO-Q achieves the best performance and converges fast

Experiments

Group 3: Patrol Scheduling







- Even in indexable RMABs, GINO-Q performs better than Whittle-index-based methods

Conclusion

- RMAB is a widely used model for resource allocation, network scheduling, public health intervention, and more
- Standard RL methods are impractical due to exponential state spaces and combinatorial action spaces
- Most existing approaches rely on the Whittle index policy, which assumes indexability—a condition often unmet in practice
- In non-indexable RMABs, Whittle-index-based methods can perform arbitrarily poorly
- We propose **GINO-Q**, a learning algorithm based on the gain index policy, which does **not** require indexability
- GINO-Q is asymptotically optimal and achieves SOTA performance in non-asymptotical settings

IMPERIAL

Thank you

GINO-Q: Learning an Asymptotically Optimal Index Policy for Restless Multi-armed Bandits
AAAI2026, Jan 2026