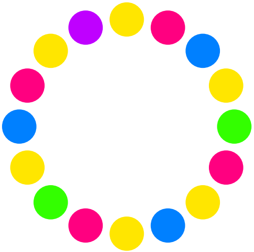
Recharging Bandits

Bobby Kleinberg **Cornell University**

Joint work with Nicole Immorlica.



NYU Machine Learning Seminar **New York, NY** **24 Oct 2017**

Prologue

Can you construct a dinner schedule that:

- never goes **2** days without macaroni and cheese
- never goes **3** days without pizza
- never goes **5** days without fish?

Answer: Impossible. For $N \geq 60$,

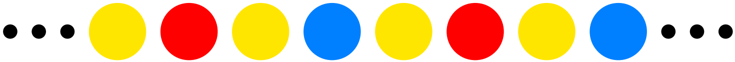
$$\lfloor N/2 \rfloor + \lfloor N/3 \rfloor + \lfloor N/5 \rfloor > N.$$

Prologue

Can you construct a dinner schedule that:

- never goes **2** days without macaroni and cheese
- never goes **4** days without pizza
- never goes **5** days without fish?

Answer: Possible.



Prologue

Can you construct a dinner schedule that:

- never goes **2** days without macaroni and cheese
- never goes **3** days without pizza
- never goes **100** days without fish?

Answer: Impossible.

Prologue

Can you construct a dinner schedule that:

- never goes **2** days without macaroni and cheese
- never goes **5** days without pizza
- never goes **100** days without fish
- never goes **7** days without tacos?

Answer: Impossible.

Prologue

Can you construct a dinner schedule that:

- never goes **2** days without macaroni and cheese
- never goes **5** days without pizza
- never goes **100** days without fish
- never goes **7** days without tacos?

Answer: Impossible.

Prologue

Can you construct a dinner schedule that:

- never goes **2** days without macaroni and cheese
- never goes **5** days without pizza
- never goes **100** days without fish
- never goes **7** days without tacos?

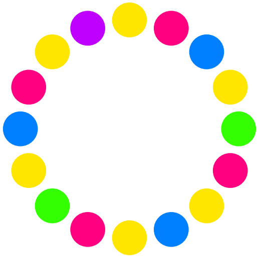
Answer: Impossible.

Prologue

The Pinwheel Problem

Given $g_1, \ldots, g_n$, can $\mathbb{Z}$ be partitioned into $S_1, \ldots, S_n$ such that S_i intersects every interval of length g_i?

E.g., $(g_1, \ldots, g_5) = (3, 4, 6, 10, 16)$

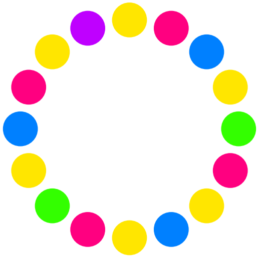


Prologue

The Pinwheel Problem

Given $g_1, \ldots, g_n$, can $\mathbb{Z}$ be partitioned into $S_1, \ldots, S_n$ such that S_i intersects every interval of length g_i?

What is the complexity of this decision problem?



Prologue

The Pinwheel Problem

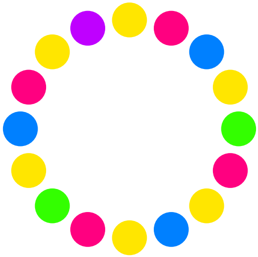
Given $g_1, \ldots, g_n$, can $\mathbb{Z}$ be partitioned into $S_1, \ldots, S_n$ such that S_i intersects every interval of length g_i?

What is the complexity of this decision problem?

It belongs to PSPACE.

No non-trivial lower bounds known.

Later in this talk: PTAS for an optimization version.



The Multi-Armed Bandit Problem

Stochastic Multi-Armed Bandit Problem: A decision-maker ("gambler") chooses one of n actions ("arms") in each time step.

Chosen arm yields random payoff from unknown distrib. on $[0,1]$.

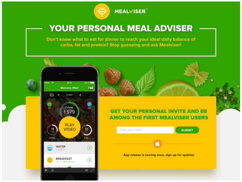
Goal: Maximize expected total payoff.

Recharging Bandits

In many applications, an arm's expected payoff is an increasing function of its "idle time."

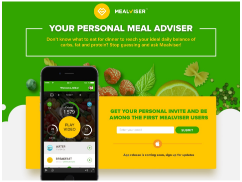
Recharging Bandits

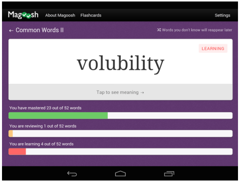
In many applications, an arm's expected payoff is an increasing function of its "idle time."



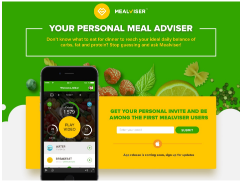
Recharging Bandits

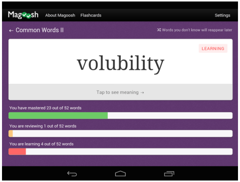
In many applications, an arm's expected payoff is an increasing function of its "idle time."





In many applications, an arm's expected payoff is an increasing function of its "idle time."





Recharging Bandits

In many applications, an arm's expected payoff is an increasing function of its "idle time."

Recharging Bandits

- Pulling arm i at time t, when it was last pulled at time s, yields random payoff with expectation $H_i(t-s)$.
- H_i is an increasing, concave function; $H_i(t) \leq t$.

Recharging Bandits

In many applications, an arm's expected payoff is an increasing function of its "idle time."

Recharging Bandits

- Pulling arm i at time t, when it was last pulled at time s, yields random payoff with expectation $H_i(t-s)$.
- H_i is an increasing, concave function; $H_i(t) \leq t$.

Concavity assumption implies free disposal: in step t, pulling i is better than doing nothing because

$$H_i(u-t) + H_i(t-s) \geq H_i(u-s).$$

In many applications, an arm's expected payoff is an increasing function of its "idle time."

Recharging Bandits

- Pulling arm i at time t, when it was last pulled at time s, yields random payoff with expectation $H_i(t-s)$.
- H_i is an increasing, concave function; $H_i(t) \leq t$.

With known $\{H_i\}$: a special case of deterministic restless bandits.

General case is PSPACE-hard [Papadimitriou & Tsitsiklis 1987].

Which reinforcement learning problems have a PTAS?

In many applications, an arm's expected payoff is an increasing function of its "idle time."

Recharging Bandits

- Pulling arm i at time t, when it was last pulled at time s, yields random payoff with expectation $H_i(t-s)$.
- H_i is an increasing, concave function; $H_i(t) \leq t$.

Plan of attack:

1. Analyze optimal play when $\{H_i\}$ are known.
2. Use upper confidence bounds + "ironing" to reduce the case when $\{H_i\}$ must be learned to the case when they are known.

Recharging Bandits

In many applications, an arm's expected payoff is an increasing function of its "idle time."

Recharging Bandits

- Pulling arm i at time t, when it was last pulled at time s, yields random payoff with expectation $H_i(t-s)$.
- H_i is an increasing, concave function; $H_i(t) \leq t$.

Plan of attack:

1. Analyze optimal play when $\{H_i\}$ are known.
2. Use upper confidence bounds + "ironing" to reduce the case when $\{H_i\}$ must be learned to the case when they are known.

Greedy $\frac{1}{2}$-Approximation

Greedy algorithm: always maximize payoff in current time step.

Greedy/OPT ratio can be arbitrarily close to 1/2

- $H_1(t) = 1 - \varepsilon, \quad H_2(t) = t.$
- Greedy always pulls arm 2.
- "Almost-OPT" pulls arm 1 for $T \gg 1$ time steps, then arm 2.
- Net payoff $(2 - \varepsilon)T + 1$ over $T + 1$ time steps.

Greedy $\frac{1}{2}$-Approximation

Greedy algorithm: always maximize payoff in current time step.

Greedy/OPT is never less than 1/2

- Imagine allowing the algorithm (but not OPT) to pull two arms per time step.
- At each time, supplement the greedy selection with the arm selected by OPT, if they differ.
- This at most doubles the payoff in each time step.
- Net payoff of supplemented schedule $\geq$ OPT.
  (free disposal property)

Rate of Return Function

For $0 \leq x \leq 1$, let $R_i(x)$ denote maximum long-run average payoff achievable by playing i in at most x fraction of time steps.

$$R_i(x) = \sup \left\{ \frac{1}{T} \sum_{j=1}^{\ell} H_i(t_j - t_{j-1}) \;\middle|\; \begin{array}{l} T < \infty,\ \ell \leq x \cdot T, \\ 0 = t_0 < t_1 < \cdots < t_\ell \leq T \end{array} \right\}.$$

Rate of Return Function

For $0 \leq x \leq 1$, let $R_i(x)$ denote maximum long-run average payoff achievable by playing i in at most x fraction of time steps.

$$R_i(x) = \sup \left\{ \frac{1}{T} \sum_{j=1}^{\ell} H_i(t_j - t_{j-1}) \;\middle|\; \begin{array}{l} T < \infty,\ \ell \leq x \cdot T, \\ 0 = t_0 < t_1 < \cdots < t_\ell \leq T \end{array} \right\}.$$

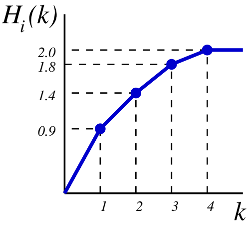
Fact: R_i is piecewise-linear with breakpoints $R_i(\frac{1}{k}) = \frac{1}{k} H_i(k)$.

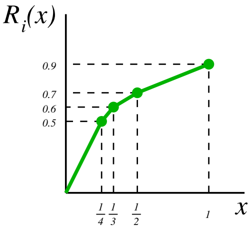
Rate of Return Function

For $0 \le x \le 1$, let $R_i(x)$ denote maximum long-run average payoff achievable by playing i in at most x fraction of time steps.

$$R_i(x) = \sup \left\{ \frac{1}{T} \sum_{j=1}^{\ell} H_i(t_j - t_{j-1}) \;\middle|\; \begin{array}{l} T < \infty,\ \ell \le x \cdot T, \\ 0 = t_0 < t_1 < \cdots < t_\ell \le T \end{array} \right\}.$$

Fact: R_i is piecewise-linear with breakpoints $R_i(\frac{1}{k}) = \frac{1}{k} H_i(k)$.





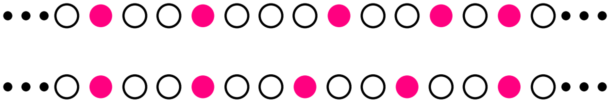
Rate of Return Function

For $0 \leq x \leq 1$, let $R_i(x)$ denote maximum long-run average payoff achievable by playing i in at most x fraction of time steps.

$$R_i(x) = \sup \left\{ \frac{1}{T} \sum_{j=1}^{\ell} H_i(t_j - t_{j-1}) \;\middle|\; \begin{array}{l} T < \infty,\ \ell \leq x \cdot T, \\ 0 = t_0 < t_1 < \cdots < t_\ell \leq T \end{array} \right\}.$$

Fact: R_i is piecewise-linear with breakpoints $R_i(\frac{1}{k}) = \frac{1}{k} H_i(k)$.

Proof sketch: The optimal sequence $0 = t_0 < \cdots < t_\ell \leq T$ has at most two distinct gap sizes, $\lfloor \frac{1}{x} \rfloor$ and $\lceil \frac{1}{x} \rceil$.

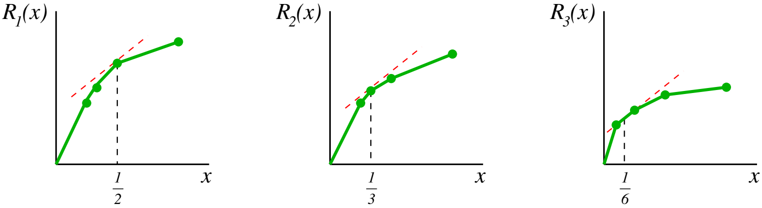


Concave Relaxation

The problem

$$\max \left\{ \sum_{i=1}^{n} R_i(x_i) \;\middle|\; \sum_i x_i \leq 1, \forall i\, x_i \geq 0 \right\}$$

specifies an upper bound on the value of the optimal schedule.

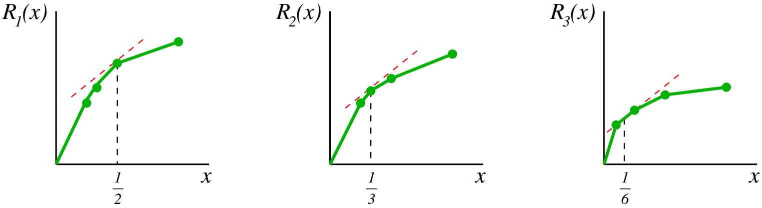


Concave Relaxation

The problem

$$\max \left\{ \sum_{i=1}^{n} R_i(x_i) \;\middle|\; \sum_i x_i \leq 1, \, \forall i \; x_i \geq 0 \right\}$$

specifies an upper bound on the value of the optimal schedule.



Mapping $(x_1, \ldots, x_n)$ to a schedule: pinwheel problem!

Independent Rounding

First idea: every time step, sample arm i with probability x_i.

Then $\tau_i =$ delay of arm $i = t_j(i) - t_{j-1}(i)$ is geometrically distributed with expectation $1/x_i$.

Rounding scheme gets $x_i \cdot \mathbb{E} H_i(\tau_i)$ whereas relaxation gets $R_i(x_i) = x_i H_i(1/x_i) = x_i \cdot H_i(\mathbb{E}\tau_i)$.

Fact: if H is concave and non-decreasing and Y is geometrically distributed then $\mathbb{E} H(Y) \geq \left(1 - \frac{1}{e}\right) H(\mathbb{E} Y)$.

Independent Rounding

First idea: every time step, sample arm i with probability x_i.

Then $\tau_i =$ delay of arm $i = t_j(i) - t_{j-1}(i)$ is geometrically distributed with expectation $1/x_i$.

Rounding scheme gets $x_i \cdot \mathbb{E} H_i(\tau_i)$ whereas relaxation gets $R_i(x_i) = x_i H_i(1/x_i) = x_i \cdot H_i(\mathbb{E}\tau_i)$.

Fact: if H is concave and non-decreasing and Y is geometrically distributed then $\mathbb{E} H(Y) \geq \left(1 - \frac{1}{e}\right) H(\mathbb{E} Y)$.

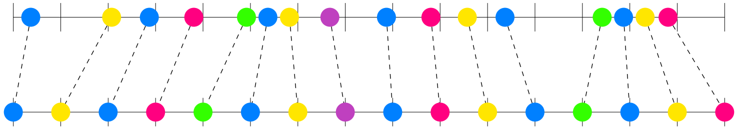
To do better, need rounding scheme that reduces variance of τ_i.

Interleaved Arithmetic Progressions

Second idea: round continuous-time schedule to discrete time.

In continuous time, pull i at $\{\frac{r_i+k}{x_i} \mid k \in \mathbb{N}\}$ where $r_i \sim_{\mathsf{Unif}} [0, 1)$.

Map this schedule to discrete time in an order-preserving manner.

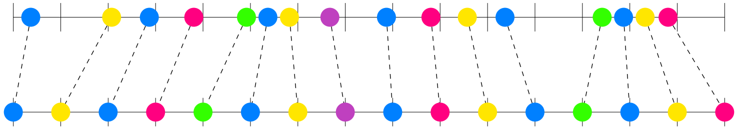


Interleaved Arithmetic Progressions

Second idea: round continuous-time schedule to discrete time.

In continuous time, pull i at $\{\frac{r_i+k}{x_i} \mid k \in \mathbb{N}\}$ where $r_i \sim_{\mathsf{Unif}} [0, 1)$.

Map this schedule to discrete time in an order-preserving manner.



Between two pulls of i, we pull j either $\lfloor x_j/x_i \rfloor$ or $\lceil x_j/x_i \rceil$ times.

$$\tau_i = 1 + \sum_{j \neq i} Z_j$$

$\{Z_j\}$ are independent, each supported on 2 consecutive integers.

Convex Stochastic Ordering

Definition

If X, Y are random variables, the *convex stochastic ordering* defines $X \leq_{\mathrm{cx}} Y$ if and only if $\mathbb{E}\phi(X) \leq \mathbb{E}\phi(Y)$ for every convex function ϕ.

Convex Stochastic Ordering

Definition

If X, Y are random variables, the *convex stochastic ordering* defines $X \leq_{\mathrm{cx}} Y$ if and only if $\mathbb{E}\phi(X) \leq \mathbb{E}\phi(Y)$ for every convex function ϕ.

Lemma

If X is a sum of independent Bernoulli random variables and Y is Poisson with $\mathbb{E}Y = \mathbb{E}X$ then $X \leq_{\mathrm{cx}} Y$.

Convex Stochastic Ordering

Definition

If X, Y are random variables, the *convex stochastic ordering* defines $X \leq_{\mathrm{cx}} Y$ if and only if $\mathbb{E}\phi(X) \leq \mathbb{E}\phi(Y)$ for every convex function ϕ.

Lemma

If X is a sum of independent Bernoulli random variables and Y is Poisson with $\mathbb{E}Y = \mathbb{E}X$ then $X \leq_{\mathrm{cx}} Y$.

$$\begin{aligned} \tau_i &= 1 + \textstyle\sum_{j \neq i} Z_j \leq_{\mathrm{cx}} 1 + \mathrm{Pois}(\tfrac{1}{x_i} - 1) \\ x_i \cdot \mathbb{E}H_i(\tau_i) &\geq x_i \cdot \mathbb{E}H_i(1 + \mathrm{Pois}(\tfrac{1}{x_i} - 1)) \end{aligned}$$

Approximation Ratio for Interleaved AP Rounding

Fact 1: If H is concave and non-decreasing and Y is Poisson, then

$$\mathbb{E}H(1+Y) \;\geq\; (1-\tfrac{1}{2e})H(1+\mathbb{E}Y)$$

Fact 2: If H is concave and non-decreasing and Y is Poisson with $\mathbb{E}Y \geq m$, then

$$\mathbb{E}H(1+Y) \;\geq\; \left(1-\frac{1}{\sqrt{2\pi m}}\right)H(1+\mathbb{E}Y)$$

Conclusion: Interleaved AP rounding is

- a $1-\frac{1}{2e} \approx 0.816$ approximation in general
- a $1-\delta$ approximation for "small arms" to whom the concave relaxation assigns $x_i < \delta^2$

PTAS for Recharging Bandits

Let $\varepsilon > 0$ be a small constant. Two easy cases . . .

1. All arms are big. Every arm that gets pulled in the optimal schedule is pulled with frequency ε^2 or greater.

   Then the optimal schedule uses only $1/\varepsilon^2$ arms. Brute-force search takes polynomial time.

2. All arms are small. If the optimal concave program solution has $x_i < \varepsilon^2$ for all i, then randomly interleaved arithmetic progressions get $1 - \varepsilon$ approximation.

Combine the cases using "partial enumeration". For $p = O_\varepsilon(1) \ldots$

Outer loop: iterate over p-periodic schedules of arms and gaps.

Inner loop: fit small arms into gaps using interleaved AP rounding.

PTAS Difficulties

- Gaps in the p-periodic schedule may not be equally spaced.
  - For each small arm choose just one congruence class (mod p) of "eligible gaps."
  - Bin-pack small arms into congruence classes.

PTAS Difficulties

- Gaps in the p-periodic schedule may not be equally spaced.
  - For each small arm choose just one congruence class (mod p) of "eligible gaps."
  - Bin-pack small arms into congruence classes.
  - Works if $x_i < \varepsilon^2/p$ for small arms while $x_i \geq 1/p$ for big arms.

PTAS Difficulties

- Gaps in the p-periodic schedule may not be equally spaced.
  - For each small arm choose just one congruence class (mod p) of "eligible gaps."
  - Bin-pack small arms into congruence classes.
  - Works if $x_i < \varepsilon^2/p$ for small arms while $x_i \geq 1/p$ for big arms.
  - Eliminate intermediate arms by finding $k \leq 1/\varepsilon$ such that arms with $x_i \in (\varepsilon^{4(k+1)}, \varepsilon^{4k}]$ contribute less than $\varepsilon \cdot \mathrm{OPT}$.
  - Conclusion: $\#$ of big arms $\leq (1/\varepsilon)^{O(1/\varepsilon)}$.

PTAS Difficulties

- Gaps in the p-periodic schedule may not be equally spaced.
- Why can we assume big arms are scheduled with period $p = O_\varepsilon(1)$?
  - We need existence of a p-periodic schedule that matches two properties of OPT
    1. rate of return from big arms
    2. amount of time left over for small arms
  - Existence proof is surprisingly technical; omitted.
  - Conclusion $p = (\#\text{big})/\varepsilon^2$ suffices.

PTAS Difficulties

- Gaps in the p-periodic schedule may not be equally spaced.
- Why can we assume big arms are scheduled with period $p = O_\varepsilon(1)$?
  - We need existence of a p-periodic schedule that matches two properties of OPT
    1. rate of return from big arms
    2. amount of time left over for small arms
  - Existence proof is surprisingly technical; omitted.
  - Conclusion $p = (\#\text{big})/\varepsilon^2$ suffices.
- Grand conclusion: PTAS with running time $n^{(1/\varepsilon)^{(24/\varepsilon)}}$.

PTAS Difficulties

- Gaps in the p-periodic schedule may not be equally spaced.
- Why can we assume big arms are scheduled with period $p = O_\varepsilon(1)$?
  - We need existence of a p-periodic schedule that matches two properties of OPT
    1. rate of return from big arms
    2. amount of time left over for small arms
  - Existence proof is surprisingly technical; omitted.
  - Conclusion $p = (\#\text{big})/\varepsilon^2$ suffices.
- Grand conclusion: PTAS with running time $n^{(1/\varepsilon)^{(24/\varepsilon)}}$.
- Remark: the 0.816-approximation runs in time $O(n^2 \log n)$.

Recharging Bandits: Regret Minimization

Now suppose $\{H_i\}$ are not known, must be learned by sampling.

Recharging Bandits: Regret Minimization

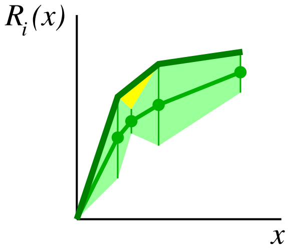
Now suppose $\{H_i\}$ are not known, must be learned by sampling.

Idea: divide time into "planning epochs" of length $\phi = O(n/\epsilon)$.

In each epoch ...

1. Compute $\bar{H}_i(x)$, an upper confidence bound on $H_i(x)$, $\forall i$.
2. Run approx alg. on $\{\bar{H}_i\}$ to schedule arms within epoch.
3. Update empirical estimates and confidence radii.

Main challenge: Although H_i is concave, $\bar{H}_i$ may not be.



Recharging Bandits: Regret Minimization

Now suppose $\{H_i\}$ are not known, must be learned by sampling.

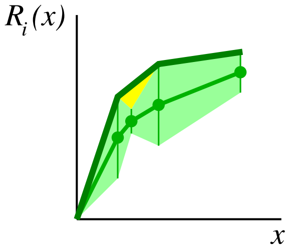
Idea: divide time into "planning epochs" of length $\phi = O(n/\epsilon)$.

In each epoch . . .

1. Compute $\bar{H}_i(x)$, an upper confidence bound on $H_i(x)$, $\forall i$.
2. Run approx alg. on $\{\bar{H}_i\}$ to schedule arms within epoch.
3. Update empirical estimates and confidence radii.

Main challenge: Although H_i is concave, $\bar{H}_i$ may not be.

Solution: Work with $\bar{R}_i$ and "iron" the non-concavity, without disrupting the approximation guarantee.



Recharging Bandits: Regret Minimization

Now suppose $\{H_i\}$ are not known, must be learned by sampling.

Idea: divide time into "planning epochs" of length $\phi = O(n/\epsilon)$.

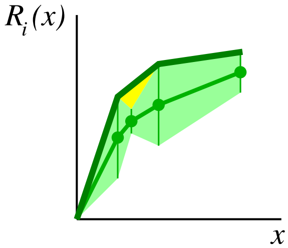
In each epoch ...

1. Compute $\bar{H}_i(x)$, an upper confidence bound on $H_i(x)$, $\forall i$.
2. Run approx alg. on $\{\bar{H}_i\}$ to schedule arms within epoch.
3. Update empirical estimates and confidence radii.

Approx alg. is almost black box.

Can plug in greedy, interleaved AP rounding, or PTAS.

Approx. factor reduced by $1 - \varepsilon$, plus $O\left(n \log(n) \sqrt{T \log(nT)}\right)$ regret.



Recharging Bandits: Regret Minimization

Now suppose $\{H_i\}$ are not known, must be learned by sampling.

Idea: divide time into "planning epochs" of length $\phi = O(n/\epsilon)$.

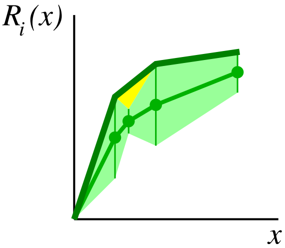
In each epoch ...

1. Compute $\bar{H}_i(x)$, an upper confidence bound on $H_i(x)$, $\forall i$.
2. Run approx alg. on $\{\bar{H}_i\}$ to schedule arms within epoch.
3. Update empirical estimates and confidence radii.

Approx alg. is almost black box.

Can plug in greedy, interleaved AP rounding, or PTAS.

Approx. factor reduced by $1 - \varepsilon$, plus $O\left(n \log(n) \sqrt{T \log(nT)}\right)$ regret.



Summary

Recharging bandits: A model for learning to schedule recurring tasks (interventions) whose benefit increases with latency.

Approximation algorithms:

- simple greedy ($\frac{1}{2}$);
- rounding concave relaxation using interleaved arithmetic progressions ($1 - \frac{1}{2e}$);
- partial enumeration and concave rounding ($1 - \varepsilon$).

Nice connections to pinwheel problem in additive combinatorics.

Open Questions

1. Pinwheel problem
    1. Complexity? (Could be in P. Could be PSPACE-complete.)
    2. Is $(g_1, \ldots, g_n)$ always feasible if $\sum_i g_i^{-1} \leq 5/6$?
    3. Is $(g_1 + 1, \ldots, g_n + 1)$ always feasible if $\sum_i g_i^{-1} \leq 1$?

Open Questions

1. Pinwheel problem
   1. Complexity? (Could be in P. Could be PSPACE-complete.)
   2. Is $(g_1, \ldots, g_n)$ always feasible if $\sum_i g_i^{-1} \leq 5/6$?
   3. Is $(g_1 + 1, \ldots, g_n + 1)$ always feasible if $\sum_i g_i^{-1} \leq 1$?
      Best result in this direction: increase $g_i + 1$ to $g_i + g_i^{1/2+o(1)}$.
      [Immorlica-K. 2017]

Open Questions

1. Pinwheel problem
   1. Complexity? (Could be in P. Could be PSPACE-complete.)
   2. Is $(g_1, \ldots, g_n)$ always feasible if $\sum_i g_i^{-1} \leq 5/6$?
   3. Is $(g_1 + 1, \ldots, g_n + 1)$ always feasible if $\sum_i g_i^{-1} \leq 1$?
      Best result in this direction: increase $g_i + 1$ to $g_i + g_i^{1/2+o(1)}$.
      [Immorlica-K. 2017]
2. Reinforcement learning: What other special cases admit PTAS?

Open Questions

3. Applications: extend recharging bandits model to incorporate domain-specific features such as . . .
    1. (fighting poachers) Strategic arms with endogenous payoffs. [Kempe-Schulman-Tamuz '17]
    2. (invasive species removal) Externalities between arms. Movement costs.
    3. (education) Payoffs with more complex history-dependency. [Novikoff-Kleinberg-Strogatz '11]