Microsoft Research

Each year Microsoft Research hosts hundreds of influential speakers from around the world including leading scientists, renowned experts in technology, book authors, and leading academics, and makes videos of these lectures freely available.

2016 © Microsoft Corporation. All rights reserved.
Recharging Bandits

Nicole Immorlica, Microsoft Research

Joint work with Bobby Kleinberg
multi-armed bandits.

A decision-maker ("gambler") chooses one of $n$ actions ("arms") in each time step.

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

Goal. Maximize expected total payoff.

Measured in regret, i.e., the difference between gambler’s payoff and payoff of best fixed arm.
recharging bandits.

In many applications, an arm’s expected payoff is an increasing function of its “idle time.”
recharging bandits.

Definition. In recharging bandits, pulling arm $i$ at time $t$ when it was last pulled at time $s$ yields a payoff with expectation $H_i(t-s) \in [0, t-s]$.

Assumption. The function $H_i(\cdot)$ is
- increasing: rewards accumulate over time,
- concave: pulling $i$ is better than doing nothing.

Goal. Maximize expected total payoff.
Known payoffs. Analyze optimal play when expected payoffs $H_i(\cdot)$ are known.

Interlude. Scheduling songs with specific frequencies.

Unknown payoffs. Use upper-confidence bounds plus "ironing" to reduce to known payoffs case.
greedy.

Maximize expected payoff in current step.

Performance.
- **Negative result**: Ratio of Greedy to OPT can be arbitrarily close to $1/2$.
- **Positive result**: Ratio of Greedy to OPT is never less than $1/2$, i.e., Greedy is a 2-approximation.
negative result.

If we run for $T + 1$ time steps,
- **Greedy**: Always play Beatles for payoff of $T + 1$,
- **OPT**: Play Nirvana for $T$ time steps, then Beatles, for a payoff of $(1 - \epsilon)T + T + 1 = (2 - \epsilon)T + 1$.

$$H_{\text{nirvana}}(t) = 1 - \epsilon \quad H_{\text{beatles}}(t) = t$$
greedy.

Maximize expected payoff in current step.

Performance.
- **Negative result:** Ratio of Greedy to OPT can be arbitrarily close to $1/2$.
- **Positive result:** Ratio of Greedy to OPT is never less than $1/2$, i.e., Greedy is a 2-approximation.
positive result.

**Enhanced Greedy**: pulls up to 2 arms per time step.
- Construct the greedy schedule.
- In time step $t$, if greedy pulls a different arm than OPT, pull that arm as well.
negative result.

If we run for $T + 1$ time steps,
- **Greedy**: Always play Beatles for payoff of $T + 1$,
- **OPT**: Play Nirvana for $T$ time steps, then Beatles, for a payoff of $(1 - \epsilon)T + T + 1 = (2 - \epsilon)T + 1$.

$$H_{\text{nirvana}}(t) = 1 - \epsilon \quad \text{and} \quad H_{\text{beatles}}(t) = t$$
positive result.

**Enhanced Greedy**: pulls up to 2 arms per time step.
- Construct the greedy schedule.
- In time step $t$, if greedy pulls a different arm than OPT, pull that arm as well.
positive result.

Enhanced Greedy, pulls up to 2 arms per time step.
- Construct the greedy schedule.
- In time step $t$, if greedy pulls a different arm than OPT, pull that arm as well.

Proof. (of 2-approximation)

$$2 \times \text{Greedy} \geq \text{Enhanced Greedy} \quad \text{(by Greedy rule)}$$
positive result.

Enhanced Greedy, pulls up to 2 arms per time step.
- Construct the greedy schedule.
- In time step $t$, if greedy pulls a different arm than OPT, pull that arm as well.

Proof. (of 2-approximation)

$$2 \times \text{Greedy} \geq \text{Enhanced Greedy} \geq \text{OPT}$$

(by concavity)
improved approximation.

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

\[
R_i(x) = \sup \left[ \frac{1}{T} \sum_{j=1}^{l} H_i(t_j - t_{j-1}) \right]
\]

where \( T < \infty, l \leq xT, t_0 = 0 < t_1 < \cdots < t_l \).
improved approximation.

Lemma. $R_i(x)$ is concave and piecewise linear with breakpoints $R_i\left(\frac{1}{k}\right) = \left(\frac{1}{k}\right) \cdot H_i(k)$ for integers $k$.

Proof. The optimal sequence has at most two distinct gap sizes, $\left\lfloor \frac{1}{x} \right\rfloor$ and $\left\lceil \frac{1}{x} \right\rceil$: 

\[
\begin{array}{cccccccc}
\square & \square & \square & \square & \square & \square & \square & \square \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots
\end{array}
\]
improved approximation.

Lemma. $R_i(x)$ is concave and piecewise linear with breakpoints $R_i\left(\frac{1}{k}\right) = \frac{1}{k} \cdot H_i(k)$ for integers $k$.

Proof. The optimal sequence has at most two distinct gap sizes, $\lfloor \frac{1}{x} \rfloor$ and $\lceil \frac{1}{x} \rceil$: 

```
  . . .
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] ...
```

```
  . . .
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] ...
```
improved approximation.

Concave relaxation. Find optimal frequencies \( \{x_i\}_{i=1}^n \).

\[
\max \left\{ \sum_{i=1}^n R_i(x_i) \mid \sum_{i} x_i \leq 1, \forall i, x_i \geq 0 \right\}
\]
improved approximation.

Lemma. $R_i(x)$ is concave and piecewise linear with breakpoints $R_i(\frac{1}{k}) = \left(\frac{1}{k}\right) \cdot H_i(k)$ for integers $k$.

Proof. The optimal sequence has at most two distinct gap sizes, $[\frac{1}{x}]$ and $[\frac{1}{x}]$: 

\[ \ldots \square \square \square \square \square \square \square \square \square \square \square \square \square \ldots \]
improved approximation.

Concave relaxation. Find optimal frequencies $\{x_i\}_{i=1}^n$.

$$\max \left\{ \sum_{i=1}^{n} R_i(x_i) \mid \sum_{i} x_i \leq 1, \forall i, x_i \geq 0 \right\}$$
improved approximation.

Concave relaxation. Find optimal frequencies \( \{x_i\}_{i=1}^n \).

\[
\max \left\{ \sum_{i=1}^n R_i(x_i) \mid \sum_{i} x_i \leq 1, \forall i, x_i \geq 0 \right\}
\]

\( R_1(x_1) \)  \( R_2(x_2) \)  \( R_3(x_3) \)
improved approximation.

Concave relaxation. Find optimal frequencies $\{x_i\}_{i=1}^n$.

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

![Graphs](R_1(x_1), R_2(x_2), R_3(x_3))
improved approximation.

Concave relaxation.
Finds optimal frequencies $\{x_i\}_{i=1}^n$.

Rounding.
Given target frequencies $\{x_i\}_{i=1}^n$, find a schedule that pulls arm $i$ at least once every $(1/x_i)$ steps.
Known payoffs. Analyze optimal play when expected payoffs $H_i(\cdot)$ are known.

Interlude. Scheduling songs with specific frequencies.

Unknown payoffs. Use upper-confidence bounds plus “ironing” to reduce to known payoffs case.
pinwheel scheduling.

Can you construct an infinite playlist that plays
- Nirvana at least every 2 songs
- Beatles at least every 3 songs
- Pink Floyd at least every 5 songs?

No! In the first 30 songs, you’d need
- \( \frac{30}{2} = 15 \) Nirvana songs
- \( \frac{30}{3} = 10 \) Beatles songs
- \( \frac{30}{5} = 6 \) Pink Floyd songs

which is 31 > 30 songs in total.
pinwheel scheduling.

Can you construct an infinite playlist that plays
- Nirvana at least every 2 songs
- Beatles at least every 4 songs
- Pink Floyd at least every 5 songs?

Yes.
pinwheel scheduling.

Can you construct an infinite playlist that plays
- Nirvana at least every 2 songs
- Beatles at least every 3 songs
- Pink Floyd at least every 100 songs?

No, even though \( \frac{1}{2} + \frac{1}{3} + \frac{1}{100} < 1 \).
pinwheel scheduling.

Classic problem. Given $x_1, \ldots, x_l$, can the integers be partitioned into sets $S_1, \ldots, S_l$ such that set $S_i$ intersects every interval of length $1/x_i$?

Conjecture. If $\sum_{i=1}^l x_i \leq 5/6$, then yes!

But we only need an approximate optimization version of this problem to round our relaxation.
Concave relaxation.
Finds frequencies \( \{x_i\}_{i=1}^n \) that maximize \( \sum_i R_i(x_i) \).

Independent rounding.
At each time step, play arm \( i \) with probability \( x_i \).

Analysis. Delay \( \tau_i \) of arm \( i \) geometrically distributed with expectation \( 1/x_i \).
- Rounding gets \( x_i \cdot E[H_i(\tau_i)] \).
- Relaxation gets \( R_i(x_i) = x_i \cdot H_i(E[\tau_i]) \).
A \((1 - 1/e)\) factor of relaxation as \( H_i(\cdot) \) is concave.
improved rounding.

Interleaved rounding.

- In continuous time, pull $i$ at $\left\{ \frac{r_i + k}{x_i} \mid k \in \mathbb{N} \right\}$

where $r_i \sim U[0,1]$ is a random offset.
improved rounding.

Interleaved rounding.
- In continuous time, pull $i$ at $\left\{ \frac{r_i+k}{x_i} \mid k \in \mathbb{N} \right\}$

where $r_i \sim U[0,1]$ is a random offset.
improved rounding.

Interleaved rounding.

- In continuous time, pull \( i \) at \( \left\{ \frac{r_i + k}{x_i} \mid k \in \mathbb{N} \right\} \)
  where \( r_i \sim U[0,1] \) is a random offset.
- Map to discrete time by preserving order.
improved rounding.

Interleaved rounding.
- In continuous time, pull $i$ at $\left\{ \frac{r_i + k}{x_i} \middle| k \in \mathbb{N} \right\}$
  where $r_i \sim U[0,1]$ is a random offset.
- Map to discrete time by preserving order. This is a lower variance rounding scheme.
improved rounding.

Analysis:
- Let $Z_i$ be the # of pulls of arm $j$ between two pulls of arm $i$. Delay $\tau_i$ of arm $i$ is

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

- $Z_j$ are independent with mean $\frac{x_j}{x_i}$, and are supported on $\left[ \frac{x_j}{x_i} \right]$ and $\left[ \frac{x_j}{x_i} \right]$. 
improved rounding.

Defn. $X$ is less than $Y$ in the convex stochastic order, written $X \leq_{cx} Y$, if $\mathbb{E}[\phi(X)] \leq \mathbb{E}[\phi(Y)]$ for every convex function $\phi$.

Lemma. If $X$ is a sum of independent Bernoullis and $Y$ is Poisson with $\mathbb{E}[X] = \mathbb{E}[Y]$, then $X \leq_{cx} Y$.

$$\tau_i = 1 + \sum_{j \neq i} Z_j \leq_{cx} 1 + \text{Pois} \left( \frac{1}{x_i} - 1 \right) = 1 + Y_i$$
improved rounding.

Thm. Interleaved rounding is a \((1 - \frac{1}{2e})\)-approx.

Prf. Contribution to rounding is \(x_i \cdot E[H_i(\tau_i)]\); contribution to concave relaxation is \(x_i \cdot H_i\left(\frac{1}{x_i}\right)\).

\[E[H_i(\tau_i)] \geq E[H_i(1 + Y_i)]\]  
(convex stochastic ordering)
improved rounding.

Thm. Interleaved rounding is a \((1 - \frac{1}{2e})\)-approx.

Prf. Contribution to rounding is \(x_i \cdot E[H_i(\tau_i)]\); contribution to concave relaxation is \(x_i \cdot H_i\left(\frac{1}{x_i}\right)\).

\[
E[H_i(\tau_i)] \geq E[H_i(1 + Y_i)] \\
\geq (1 - \frac{1}{2e})H_i(1 + E[Y_i])
\]

(fact about concave functions of Poisson RVs)
improved rounding.

Thm. Interleaved rounding is a \((1 - \frac{1}{2e})\)-approx.

Prf. Contribution to rounding is \(x_i \cdot \mathbb{E}[H_i(\tau_i)]\); contribution to concave relaxation is \(x_i \cdot H_i\left(\frac{1}{x_i}\right)\).

\[
\mathbb{E}[H_i(\tau_i)] \geq \mathbb{E}[H_i(1 + Y_i)] \\
\geq (1 - \frac{1}{2e})H_i(1 + \mathbb{E}[Y_i]) \\
= (1 - \frac{1}{2e})H_i\left(\frac{1}{x_i}\right)
\]
improved rounding.

Interleaved rounding.

- A \( \left(1 - \frac{1}{2e}\right) \approx 0.8 \) factor of concave relaxation.
- For “small arms” to whom the relaxation devotes at most a \( \epsilon^2 \) fraction of time steps, get a \( (1 - \epsilon) \) factor!
improved rounding.

PTAS. A \((1 - \epsilon)\) factor for any constant \(\epsilon > 0\).
- Guess “big arms” and find an optimal schedule for them by brute force, leaving some gaps.
- Use relaxation to schedule “small arms” in gaps.

Running time is \(n^{\frac{1}{\epsilon}}\), vs \(O(n^2 \log n)\) for other alg.
improved rounding.

PTAS. A \((1 - \epsilon)\) factor for any constant \(\epsilon > 0\).
- Guess "big arms" and find an optimal schedule for them by brute force, leaving some gaps.
- Use relaxation to schedule "small arms" in gaps.

Running time is \(n^{\left(\frac{1}{\epsilon}\right)\frac{24}{\epsilon}}\), vs \(O(n^2\log n)\) for other alg.
outline.

Known payoffs. Analyze optimal play when expected payoffs $H_i(\cdot)$ are known.

Interlude. Scheduling songs with specific frequencies.

Unknown payoffs. Use upper-confidence bounds plus “ironing” to reduce to known payoffs case.
regret minimization.

Unknown payoffs. Now payout functions $H_i(\cdot)$ are not known, but must be learned by sampling.

Goal. A learning algorithm that gets payoff

$$\alpha \cdot \text{OPT} - O\left(n \log n \sqrt{\frac{\log n T}{T}}\right)$$

where OPT is the payoff of the optimal schedule that knows the payout functions of the arms.
regret minimization.

Minor challenge. Schedules optimize over time; choosing today’s arm considering tomorrow’s reward.

Idea. Divide time into “planning epochs” and
- Compute upper confidence bound $\overline{H}_i(x)$ on $H_i(x)$
- Run approximation algorithm of choice on $\overline{H}_i(x)$
- At end of epoch, update empirical estimates
regret minimization.

Major challenge. Although $H_i(\cdot)$ is concave, upper confidence bound $\overline{H}_i(\cdot)$ might not be concave, breaking approximation guarantees.

Idea. Define properties relating an algorithm’s approximation guarantee in this augmented reality to its regret bound.
Augmented Reality. Algorithm works with hallucinations of rewards and delays.
- Nature chooses hallucinated rewards $\bar{H}_i(d_i)$, i.e., upper confidence bounds on true reward $H_i(d_i)$.
regret minimization.

**Augmented Reality.** Algorithm works with hallucinations of rewards and delays.
- Nature chooses hallucinated rewards $\bar{H}_i(d_i)$.
- Algorithm chooses hallucinated delays $\tau(t)$, allowing it to group together similar delays to reduce variance of reward estimates.
regret minimization.

Defn. An \((\alpha, \beta)\)-strong algorithm satisfies:
- \(\alpha\)-approximate optimality. The hallucinated reward is good enough.

\[
E \left[ \sum_t \bar{H}_i(t)(\tau(t)) \right] \geq \alpha \sum_t H_i^*(t)(d^*(t))
\]

- hallucinated reward of hallucinated delay
- true reward of optimal schedule
**regret minimization.**

**Defn.** An \((\alpha, \beta)\)-strong algorithm satisfies:

- \(\alpha\)-approximate optimality.  
The hallucinated reward is good enough.

- \(\beta\)-approximate attribution.  
The hallucination rewards are close to accurate.

- \(\beta\)-approximate sampling.  
The hallucinated delays are close to actual delays.
regret minimization.

Reduction. Divide time into epochs.
- At start of epoch, run recharging bandit algorithm using upper confidence bounds as hallucinated rewards to determine schedule.
- Use hallucinated delays to collect training data.
- At end of epoch, use training data to update upper confidence bounds.
regret minimization.

**Thm.** For any \((\alpha, \beta)-\text{strong}\) algorithm satisfying [technical conditions], reduction produces a recharging bandit algorithm with expected reward

\[
(1 - \epsilon)\alpha\beta^3 \text{OPT} - \frac{\text{poly}(n)}{1-\beta} \sqrt{\frac{2 \ln (nT)}{T}}
\]

**Prf.** Follows from a bunch of messy algebra implied by conditions of an \((\alpha, \beta)-\text{strong}\) algorithm.
regret minimization.

Thm. PTAS is a \( ((1 - O(\epsilon)), (1 - O(\epsilon)) \)-strong alg.

Idea. Work with concave relaxation and iron out non-concavity without disrupting approximation.
regret minimization.

Thm. PTAS is a $(1 - O(\epsilon)), (1 - O(\epsilon))$-strong alg.

Idea. Work with concave relaxation and iron out non-concavity without disrupting approximation.
regret minimization.

Thm. PTAS is a \( (1 - O(\varepsilon)), (1 - O(\varepsilon)) \)-strong alg.

Idea. Work with concave relaxation and iron out non-concavity without disrupting approximation.
Absence makes the heart grow fonder.

Recharging bandits. The payout of an arm is an increasing concave function of the delay since it was last pulled.

Approximations. Fast constant approximations, PTAS.

Black-box learning. Use upper-confidence bounds plus “ironing” to reduce to known payoffs case.
future directions.

Applications. Extend techniques to handle domain-specific features such as
- **Fighting poachers**: arms react strategically to plan.
- **Invasive species removal**: externalities between arms, movement costs.
- **Education**: payoffs with more complex history dependencies.
regret minimization.

Thm. For any \((\alpha, \beta)\)-strong algorithm satisfying [technical conditions], reduction produces a recharging bandit algorithm with expected reward

\[
(1 - \epsilon)\alpha\beta^3 \text{OPT} - \frac{\text{poly}(n)}{1-\beta} \sqrt{\frac{2 \ln (nT)}{T}}
\]

Prf. Follows from a bunch of messy algebra implied by conditions of an \((\alpha, \beta)\)-strong algorithm.
regret minimization.

Major challenge. Although $H_i(\cdot)$ is concave, upper confidence bound $\overline{H}_i(\cdot)$ might not be concave, breaking approximation guarantees.

Idea. Define properties relating an algorithm’s approximation guarantee in this augmented reality to its regret bound.
improved rounding.

Thm. Interleaved rounding is a \((1 - \frac{1}{2e})\)-approx.

Prf. Contribution to rounding is \(x_i \cdot E[H_i(\tau_i)]\); contribution to concave relaxation is \(x_i \cdot H_i\left(\frac{1}{x_i}\right)\).

\[
E[H_i(\tau_i)] \geq E[H_i(1 + Y_i)] \\
\geq (1 - \frac{1}{2e})H_i(1 + E[Y_i])
\]

(fact about concave functions of Poisson RVs)
Augmented Reality. Algorithm works with hallucinations of rewards and delays.
- Nature chooses hallucinated rewards $\bar{H}_i(d_i)$.
- Algorithm chooses hallucinated delays $\tau(t)$, allowing it to group together similar delays to reduce variance of reward estimates.
regret minimization.

Thm. Greedy is a \(\left(\frac{1}{2}, \left(1 - \frac{1}{n}\right)\right)\)-strong algorithm.

Prf. Recall Enhanced Greedy plays both the greedy and optimal arms in time steps where they differ.

\[
2 \times \text{Greedy} \geq \overline{H}(\text{Enhanced Greedy}) \\
\geq H(\text{Enhanced Greedy}) \\
\geq \text{OPT}
\]

Implying \(\left(\frac{1}{2}\right)\)-approximate optimality (rest easy).
regret minimization.

Thm. PTAS is a \((1 - O(\epsilon)), (1 - O(\epsilon))\)-strong alg.

Idea. Work with concave relaxation and iron out non-concavity without disrupting approximation.
regret minimization.

**Thm.** PTAS is a $\left(1 - O(\epsilon)\right), \left(1 - O(\epsilon)\right)$-strong alg.

**Idea.** Work with concave relaxation and iron out non-concavity without disrupting approximation.