# Talk 4
# bounds on stochastic bandits & idea of Markovian bandits

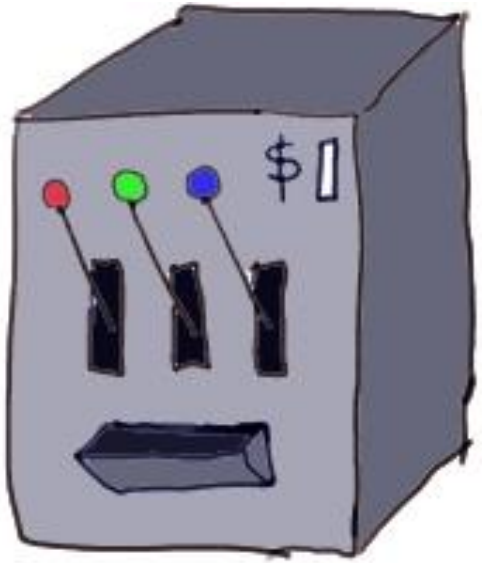
by Claudio and Mathias

# short recall

- **Multiarmed bandit**: «sequentially decide for one of K arms to pull»

# short recall

- **Multiarmed bandit**: «sequentially decide for one of K arms to pull»
- 3 classes of multiarmed bandits:
  - **stochastic bandit**: each arm fixed distribution static during all rounds
  - **adversarial bandit**: bandit allowed to change payouts in each round
  - **Markovian bandit**: state of the activated arm is allowed to change after usage (in «Markovian style»)



# stochastic bandits

| Machine number | 1 | 2 | . . . | i | . . . | K |
|---|---|---|---|---|---|---|

# stochastic bandits

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Machine number | 1 | 2 | . . . | i | . . . | K |
| distribution | $\nu_1$ | $\nu_2$ | . . . | $\nu_i$ | . . . | $\nu_K$ |
| mean of distribution $mean(\nu_i) = \mu_i$ | $\mu_1$ | $\mu_2$ | . . . | $\mu_i$ | . . . | $\mu_K$ |

# Regret

$K \geq 2$ arms,

$X_{i,t}$ reward of arm i in time step t,

$I_t \in \{1, \ldots, K\}$ selected arm in time step t,

# Regret

| | |
|---|---|
| $K \geq 2$ | arms, |
| $X_{i,t} \sim \nu_i$ | reward of arm i in time step t, |
| $I_t \in \{1, \ldots, K\}$ | selected arm in time step t, |

# Regret

$K \geq 2$ arms,

$X_{i,t} \sim \nu_i$ reward of arm i in time step t,

$I_t \in \{1, \ldots, K\}$ selected arm in time step t,

The **regret** after n plays:

$$R_n = \max_{i=1,\ldots,K} \sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}$$

# Regret

$K \geq 2$ arms,

$X_{i,t} \sim \nu_i$ reward of arm i in time step t,

$I_t \in \{1, \ldots, K\}$ selected arm in time step t,

The **regret** after n plays:

$$R_n = \max_{i=1,\ldots,K} \sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}$$

The **pseudo-regret** after n plays:

$$\bar{R}_n = \max_{i=1,\ldots,K} \mathbb{E}\left[\sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}\right]$$

# Regret: stochastic bandits

The **pseudo-regret** after n plays:

$$\bar{R}_n = \max_{i=1,\dots,K} \mathbb{E}\left[\sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}\right]$$

# Regret: stochastic bandits

The **pseudo-regret** after n plays:

$$\bar{R}_n = \max_{i=1,\dots,K} \mathbb{E}\left[\sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}\right]$$

$$= n\,\mu^* - \sum_{t=1}^{n} \mathbb{E}\left[\mu_{I_t}\right]$$

where $\mu^* := \max_{i=1,\dots,K} \mu_i$

# Regret: stochastic bandits

The **pseudo-regret** after n plays:

$$\bar{R}_n = n\,\mu^* - \sum_{t=1}^{n} \mathbb{E}\left[\mu_{I_t}\right]$$

# Regret: stochastic bandits

The **pseudo-regret** after n plays:

$$\bar{R}_n = n\,\mu^* - \sum_{t=1}^{n} \mathbb{E}\left[\mu_{I_t}\right]$$

$$= \left(\sum_{i=1}^{K} \mathbb{E}[N_n(i)]\,\mu^*\right) - \mathbb{E}\left[\sum_{i=1}^{K} N_n(i)\,\mu_i\right]$$

# Regret: stochastic bandits

The **pseudo-regret** after n plays:

$$\bar{R}_n = n\,\mu^* - \sum_{t=1}^{n} \mathbb{E}\left[\mu_{I_t}\right]$$

$$= \left(\sum_{i=1}^{K} \mathbb{E}[N_n(i)]\,\mu^*\right) - \mathbb{E}\left[\sum_{i=1}^{K} N_n(i)\,\mu_i\right]$$

$$= \sum_{i=1}^{K} \Delta_i \mathbb{E}[N_n(i)]$$

where

$N_n(i) :=$ number of times arm i pulled up to time n

$\Delta_i := \mu^* - \mu_i$ = deviation of ith mean from the best mean

# UCB - strategy

- Exploration – exploitation dilemma
- «optimism in face of uncertainty»
  - «plausible» environment
  - consistent with data
  - take most «favourable» environment

# Upper confidence bound theorem

- «No free lunch»-principle
- Assumption: there is a convex function $\psi$ on $\mathbb{R}$ with $\lambda \geq 0$:
  - $\log \mathbb{E}\left[e^{\lambda(X-\mathbb{E}[X])}\right] \leq \psi(\lambda)$
  - $\log \mathbb{E}\left[e^{\lambda(\mathbb{E}[X]-X)}\right] \leq \psi(\lambda)$
- For example: if $X \in [0,1]$, take $\psi(\lambda) = \frac{\lambda^2}{8}$

# Upper confidence bound theorem

- Legendre-Fenchel transform/convex conjugate
- $\psi^*(\lambda) = \sup_{\lambda \epsilon \mathbb{R}}(\lambda\varepsilon - \psi(\lambda))$
- Example: $\psi(x) = e^x \;\Rightarrow\; \sup_{y \epsilon \mathbb{R}} xy - e^x$

  $\Rightarrow y = \log(x) \;\Rightarrow\; \psi^*(x) = x\log(x) - x$
- Example: $\psi(x) = \frac{1}{p}|x|^p \;\Rightarrow\; \psi^*(x) = \frac{1}{q}|x|^q$

  where $1 < p, q < \infty$, $\frac{1}{p} + \frac{1}{q} = 1$
- Example: $\psi(\lambda) = \frac{\lambda^2}{8} \;\Rightarrow\; \sup_{\lambda \epsilon \mathbb{R}} \lambda\varepsilon - \frac{\lambda^2}{8} \;\Rightarrow \varepsilon - \frac{\lambda}{4} = 0$

  $\Rightarrow\; \lambda = 4\varepsilon \;\Rightarrow \psi^*(\varepsilon) = 2\varepsilon^2$

# Lower bound theorem

- reward distribution $X_{i,t} \sim Bernoulli(p, q)$, $p, q \in [0,1]$
- strategy statisfies
  - $\mathbb{E}[N_n(i)] = o(n^a)$
  - $\Delta_i > 0$
  - $a > 0$
- then $liminf_{n\to\infty} \frac{\bar{R}_n}{\log(n)} \geq \sum_{i:\Delta_i>0} \frac{\Delta_i}{kl(\mu_i, \mu^*)}$

# Comparing upper and lower bound

- $kl(\mu_i, \mu^*) = \mu_i \log\left(\frac{\mu_i}{\mu^*}\right) + (1-\mu_i)\log\left(\frac{1-\mu_i}{1-\mu^*}\right)$
- $kl(\mu_i, \mu^*) \leq \frac{(\mu_i - \mu^*)^2}{\mu^*(1-\mu^*)}$
- $\Rightarrow \quad liminf_{n\to\infty} \frac{\bar{R}_n}{\log(n)} \geq \sum_{i:\mu^*-\mu_i>0} \frac{\mu^*-\mu_i}{kl(\mu_i, \mu^*)}$

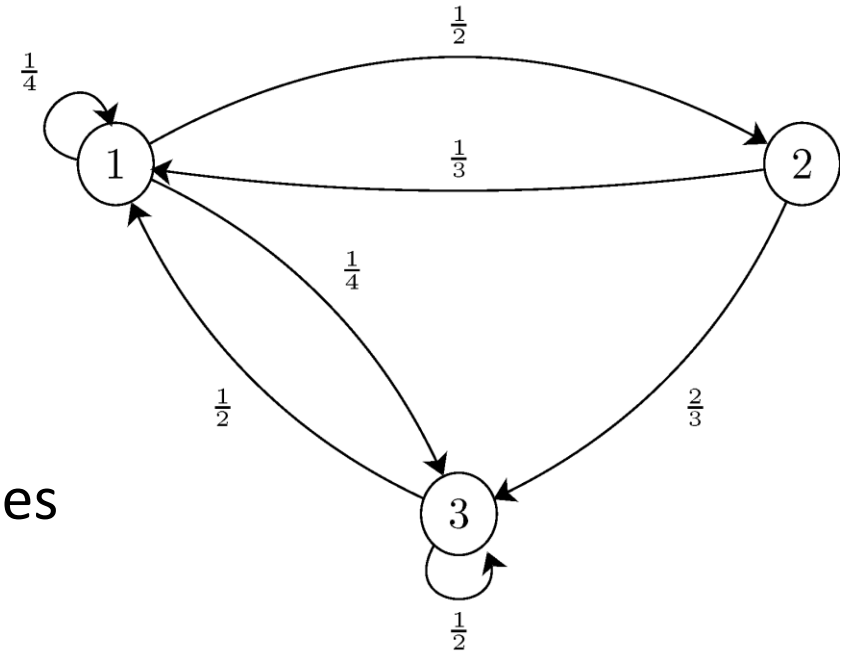
$$\geq \sum_{i:\mu^*-\mu_i>0} \frac{\mu^* - \mu_i}{2(\mu_i - \mu^*)^2} \mu^*(1-\mu^*) = \sum_{i:\mu^*-\mu_i>0} \frac{\mu^*(1-\mu^*)}{\mu^* - \mu_i}$$

# Comparing upper and lower bound

- $\bar{R}_n \leq \sum_{i:\mu^*-\mu_i>0} \left( \frac{2\alpha}{\mu^*-\mu_i} \log(n) + \frac{\alpha}{\alpha-2} \right)$
- $liminf_{n\to\infty} \frac{\bar{R}_n}{\log(n)} \geq \sum_{i:\mu^*-\mu_i>0} \frac{\mu^*(1-\mu^*)}{(\mu^*-\mu_i)}$

# Bandit process

- Bandit process: process of a single machine/arm
- 2 possible actions of the machine:
  - **continue**:
    - produces reward $r(x_t)$, $x_t$ state of machine at time t
    - the state changes to $x_{t+1}$ according to Markov dynamics $x \to y$ with probability $P(x, y)$
  - **freeze:**
    - produces no reward
    - state does not change
- **states** of the machine follow a **Markov process**
  → transition to other states with given transition probabilities



# Markovian bandit

- **Markovian bandits**:
  - collection of K bandit processes
  - at each time t: *exactly one* machine continued, *all others* stay frozen

# Markovian bandit

- **Markovian bandits**:
  - collection of K bandit processes
  - at each time t: *exactly one* machine continued, *all others* stay frozen

- Objective to optimize: **$\beta$-discounted reward** ($i_t$: arm pulled at time t)

$$\mathbb{E}\left[\sum_{t=0}^{\infty} r_{i_t}\left(x_{i_t}(t)\right)\beta^t\right], \qquad 0 < \beta < 1$$

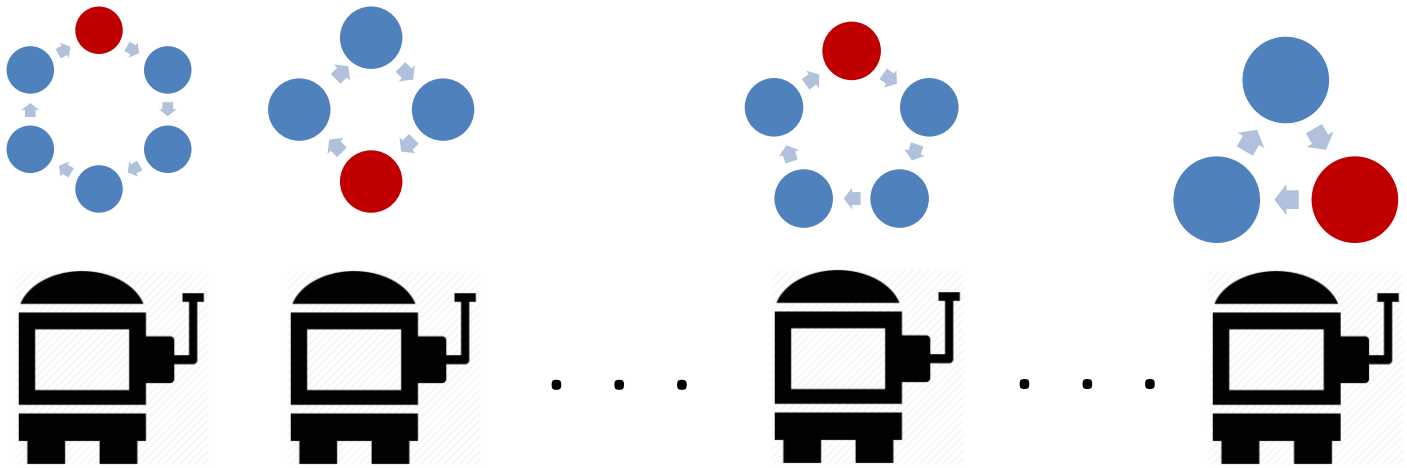
$\beta$: discounting factor, balances exploration-exploitation trade-off, keeps sum finite

# Markovian bandits

| Machine number | 1 | 2 | . . . | i | . . . | K |
|---|---|---|---|---|---|---|

# Markovian bandits

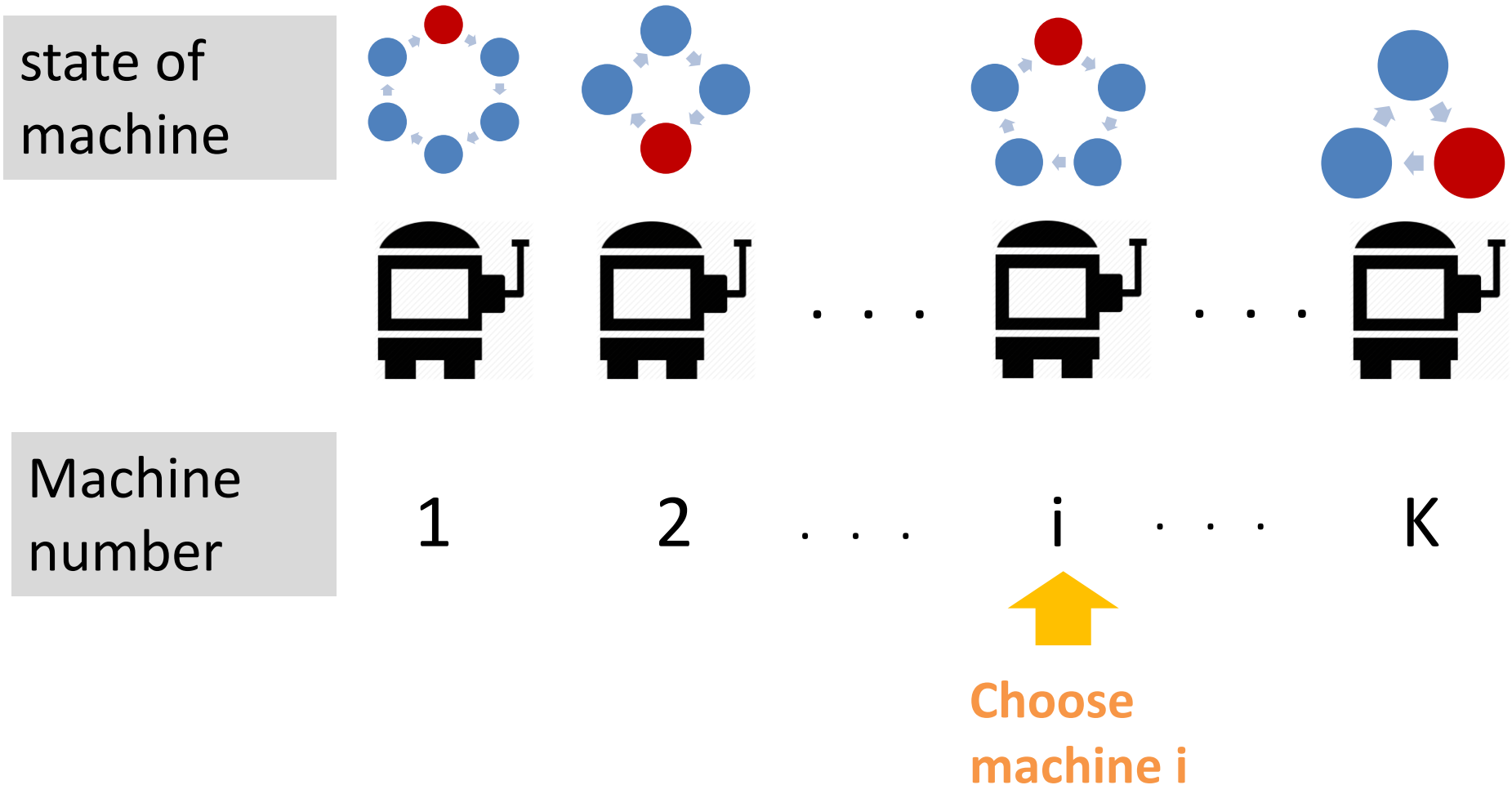
state of machine



Machine number

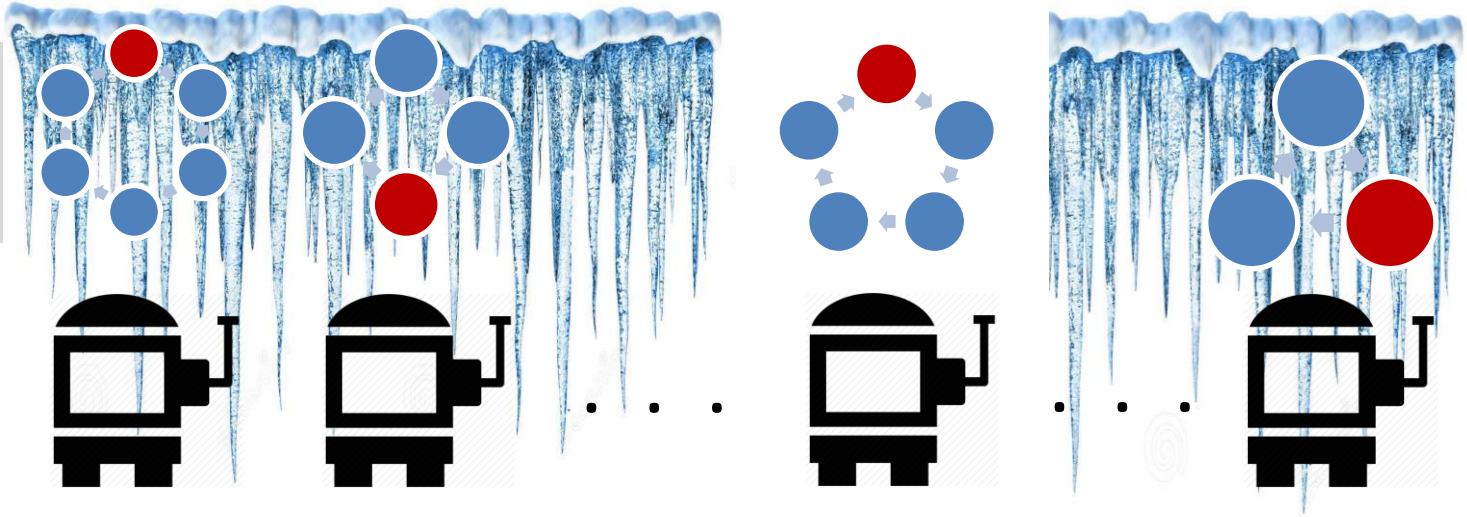
1 2 . . . i . . . K

# Markovian bandits



# Markovian bandits

state of machine



Machine number

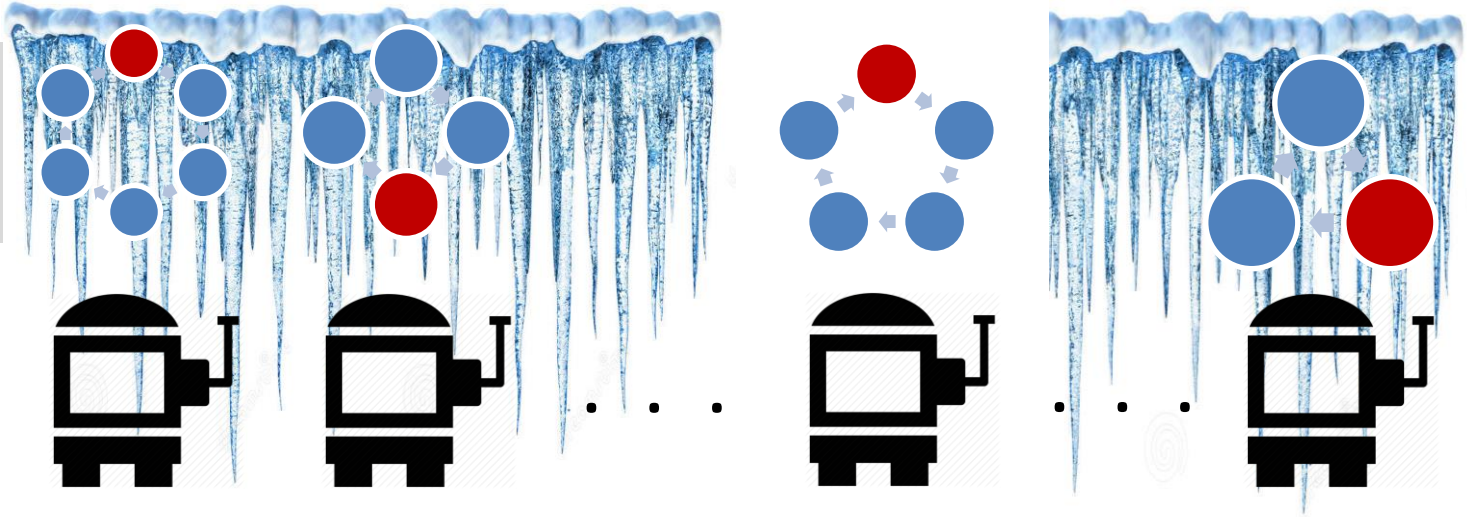
1 2 . . . i . . . K

Other machines stay frozen
→ states dont change

Choose machine i

# Markovian bandits

state of machine



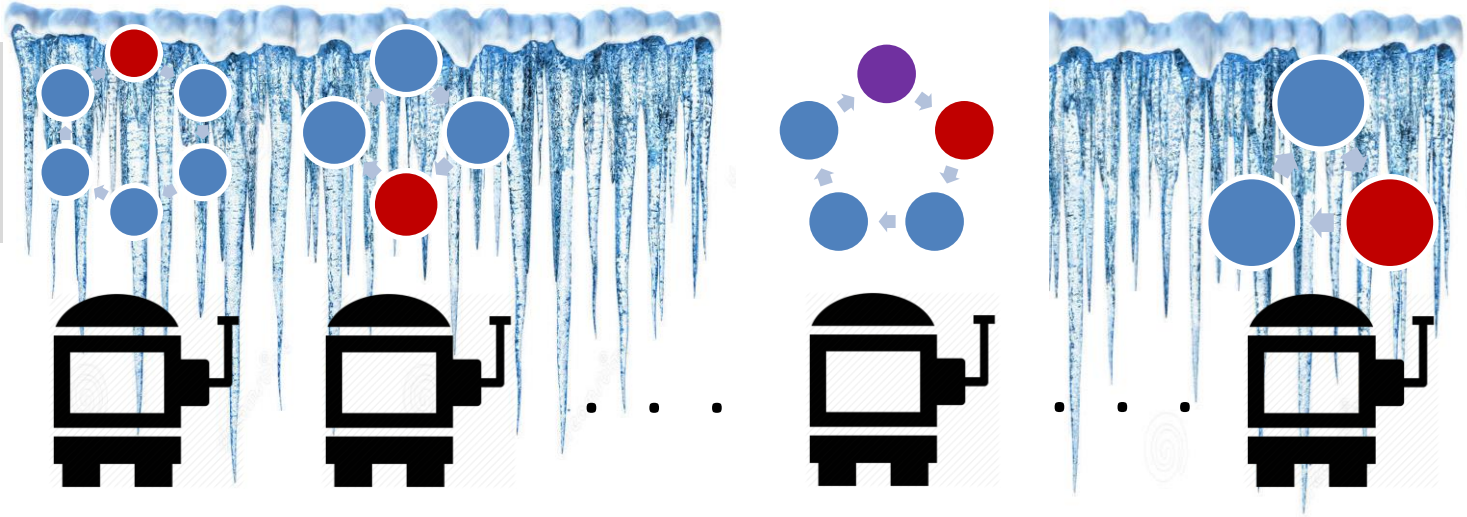
Machine number

1 2 . . . i . . . K

Produces reward $r(x_t)$

# Markovian bandits

state of machine



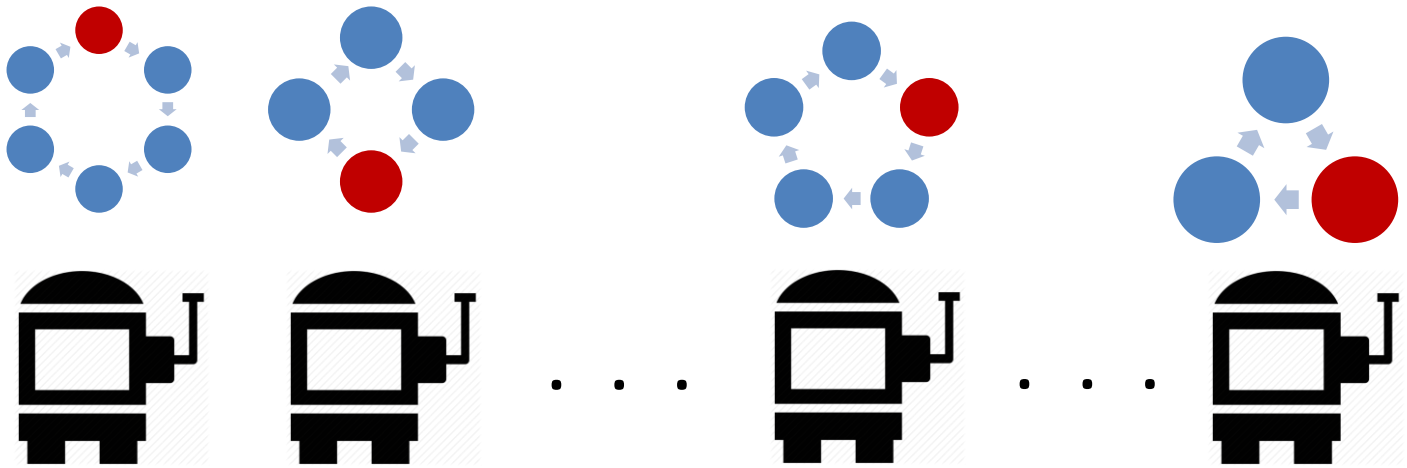
Machine number

1 2 . . . i . . . K

**Change State of machine i**

# Markovian bandits

state of machine



Machine number

1 2 . . . i . . . K

# Prolem: How to maximize?

Objective: $\beta$-discounted reward ($i_t$: arm pulled at time t)

$$\mathbb{E}\left[\sum_{t=0}^{\infty} r_{i_t}\left(x_{i_t}(t)\right)\beta^t\right], \qquad 0 < \beta < 1$$

- Possible Solution: **Dynamic Programming Solution** → computationally very expensive
- Better Solution: **Forward Indcution** → can show that optimal solution of this type → computationally much cheaper

# Solution: Forward induction

- **Idea**: Allow the number of $\tau$ steps over which we look ahead at each stage, to depend on how system evolves while these steps are taking place → i.e. stopping time!

# Car journey example

- **Problem**:
  - choose route for journey by car
  - Several different possible routes, all of *same length* which itersect at various points
  - Objective: choose route which minimizes time taken for journey

# Car journey example

- **Problem**:
  - choose route for journey by car
  - Several different possible routes, all of *same length* which itersect at various points
  - Objective: choose route which minimizes time taken for journey
- **Model**:
  - Markov decision process
  - Distance covered so far = «time» variable
  - Time take to cover each successive mile = negative reward
  - position=state
  - Action space = roads to continue at crossroads

# Car journey example

- **Suboptimality**:
  - 1st stage in forward induction → find route $\zeta_1$ and distance $\sigma_1$ along $\zeta_1$ from the start point st. average speed in traveling distance $\sigma_1$ along $\zeta_1$ maximized
  - **Suboptimal route example:**

    Start with short stretch of highway, followed by very slow section **vs.** choose a trunk road which permits good steady average speed (over all faster than the other)
  - **Trouble**:

    irrevocable decisions have been take at each cross-roads → there are alternative routes that are avilable at a stage but aren't available later on if they are not chosen→ **irrevocable decisions**

# Optimality of forward induction

- Forward induction optimal if decisions not irrevocable
- **Irrevocable**:

  any alternative that is avilable at any stage and is not chosen, my be chosen at a later stage (with exactly the same sequence of rewards (appart from discount factor))
- **Optimality for Markovian bandits**:

  decisions made not irrevocable, i.e. any alternative not chosen is available in every upcomming time (because of freezing)

  → Forward induction **optimal for Markovian bandits**

# Forward induction maximizations

- 2 maximizations:
  - **Inner maximization**:
    - Given: decision rule for taking a sequence of decisions
    - Maximize: choose stopping time $\tau$ to maximize the conditional expected reward rate
  - **Outer maximization**:
    - Given: stopping time $\tau$ from the inner maximization
    - Maximize: choose decision rule to maximize the result of the inner maximization for that decision rule

# Forward induction process

- **Resulting Procedure**:
  - At t=0:
    - Given: initial state of process
    - Select: decision rule and stopping time $\tau_1$ and follow for next $\tau_1$ steps
  - For t=1,2,…
    - Given: info accumulated so far
    - Select: new decision rule and stopping time $\tau_{t+1}$ by conditioning on info accumulated so far and follow it for next $\tau_{t+1}$ steps

# Gittins Index Theorem

**Maximal expected discounted reward**:

→Obtained by always continuing the bandit having gratest Gittins index

$$G_i(x_i) = \sup_{\tau \geq 1} \frac{\mathbb{E}[\sum_{t=0}^{\tau-1} r_i(x_i(t))\beta^t \mid x_i(0) = x_i]}{\mathbb{E}[\sum_{t=0}^{\tau-1} \beta^t \mid x_i(0) = x_i]}$$

where $\tau$ is a stopping time.

# Gittins Index

Discounted reward up to $\tau$

Discounted time up to $\tau$

$$G_i(x_i) = \sup_{\tau \geq 1} \frac{\mathbb{E}[\sum_{t=0}^{\tau-1} r_i(x_i(t))\beta^t \mid x_i(0) = x_i]}{\mathbb{E}[\sum_{t=0}^{\tau-1} \beta^t \mid x_i(0) = x_i]}$$

where $\tau$ is a stopping time.

# Gittins Index: Adventage

- find best strategy by only computing Gittins indexes of all arms
- Gittins index of one arm **doesn't depend on other arms** → huge **computational savings** compared with dynamic programming
- only need to solve **K «one-dimensional» problems** in each time step

# Example: Single Machine Scheduling

- **Problem**: n jobs to be scheduled on one machine
- job i has processing time $t_i$ and positive reward $r_i$
- If job 1 processed immediately before job 2, then:

$$r_1\beta^{t_1} + r_2\beta^{t_1+t_2} > r_2\beta^{t_2} + r_1\beta^{t_2+t_1}$$

$$\Leftrightarrow \quad G_1 = (1-\beta)\frac{r_1\beta^{t_1}}{1-\beta^{t_1}} > (1-\beta)\frac{r_2\beta^{t_2}}{1-\beta^{t_2}} = G_2$$

- Total discounted reward is maximized by choosing always job with biggest $G_i$

# Example: Single Machine Scheduling

- Obtaind by calculation: **best strategy** → always choose biggest $\frac{r_i \beta^{t_i}}{1-\beta^{t_i}}$
- The same obtained with **Gittins index**:

$$G_i(x_i) = \sup_{\tau \geq 1} \frac{\mathbb{E}[\sum_{t=0}^{\tau-1} r_i(x_i(t))\beta^t \mid x_i(0) = x_i]}{\mathbb{E}[\sum_{t=0}^{\tau-1} \beta^t \mid x_i(0) = x_i]}$$

$$= \frac{r_i \beta^{t_i}}{1+\beta+\cdots\beta^{t_i-1}}$$

optimal stopping time $\tau = t_i$