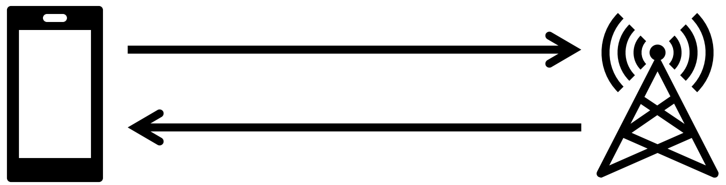
Online Rate Selection In Nonstationary Wireless Channels

Haoyang Zhang

Problem Specification – Rate Selection



Input: a set of possible rates
$0 \leq r_1 < r_2 < \cdots < r_N$

Unknown: corresponding success rates
$1 \geq \theta_1(t) > \theta_2(t) > \cdots > \theta_N(t) \geq 0$

Observation: success/fail signals
$y_i(t) \sim Bernoulli(\theta_i(t))$

Goal: find a strategy to maximize overall throughput
$$\max_{i(t)} \sum_t r_{i(t)} \theta_{i(t)}(t)$$

Gupta, Harsh, Atilla Eryilmaz, and Rayadurgam Srikant. "Low-Complexity, Low-Regret Link Rate Selection in Rapidly-Varying Wireless Channels." *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018.

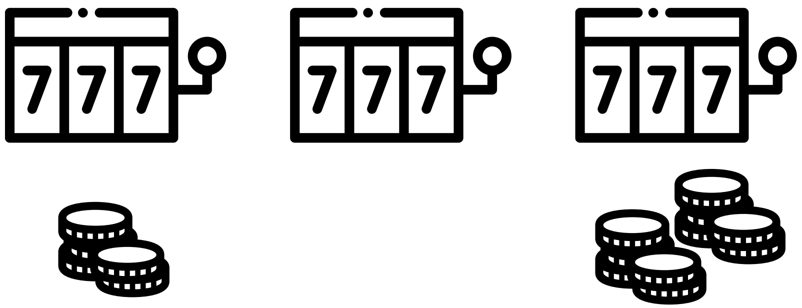
Problem Modeling – Multi-Armed Bandits

Def. $x_i(t) = r_i y_i(t)$

Unknown: corresponding success rates
$1 \geq \theta_1(t) > \theta_2(t) > \cdots > \theta_N(t) \geq 0$

Observation: rewards
$x_i(t) \sim r_i \times Bernoulli(\theta_i(t))$

Goal: find a strategy to maximize overall reward
$$\max_{i(t)} \sum_t x_{i(t)}(t)$$



Icons made by Freepik from www.flaticon.com

Proposed Method

Goal: find a strategy to maximize accumulative reward.

$$\max_{i(t)} \sum_{t} x_{i(t)}(t)$$

A balance between exploration and exploitation.

1. An efficient way to identify the best arm in a long-term stationary environment.
2. A proper constrain to exploitation in order to adapt to the time-variant environment.

Proposed Method – Lotka-Volterra

Goal: identify the best arm in a long-term stationary environment.

$$\max_{i(t)} \sum_t x_{i(t)}$$

Def. P_i = the probability of choosing the i-th arm.

Essentially, it is competition between P_i given the total resource is limited

$$\sum_i P_i = 1$$

Proposed Method – Lotka-Volterra

Goal: identify the best arm in a long-term stationary environment.

$$\max_{i(t)} \sum_t x_{i(t)}$$

Lotka-Volterra competition dynamics, "winner-take-all".

$$\frac{dz_i}{dt} = z_i \left(a_i - \sum_{j=1}^{N} c_j z_j \right) \qquad \frac{dP_i}{dt} = P_i \left(\mu_i - \sum_{j=1}^{N} \mu_j P_j \right)$$

z_i: Population (probability) of choosing i-th arm

a_i: natural growth rate, defined as $\mu_i = r_i \times \theta_i$

c_j: competitive factor, defined as $\mu_j = r_j \times \theta_j$

Niiyama, Tomoaki, et al. "Lotka-Volterra competition mechanism embedded in a decision-making method." *arXiv preprint arXiv:1907.12399* (2019).

Proposed Method – Lotka-Volterra

A long-term stationary case: $r = (0.9, 0.5, 0.3, 0.1), \theta = (0.3, 0.7, 0.8, 1)$

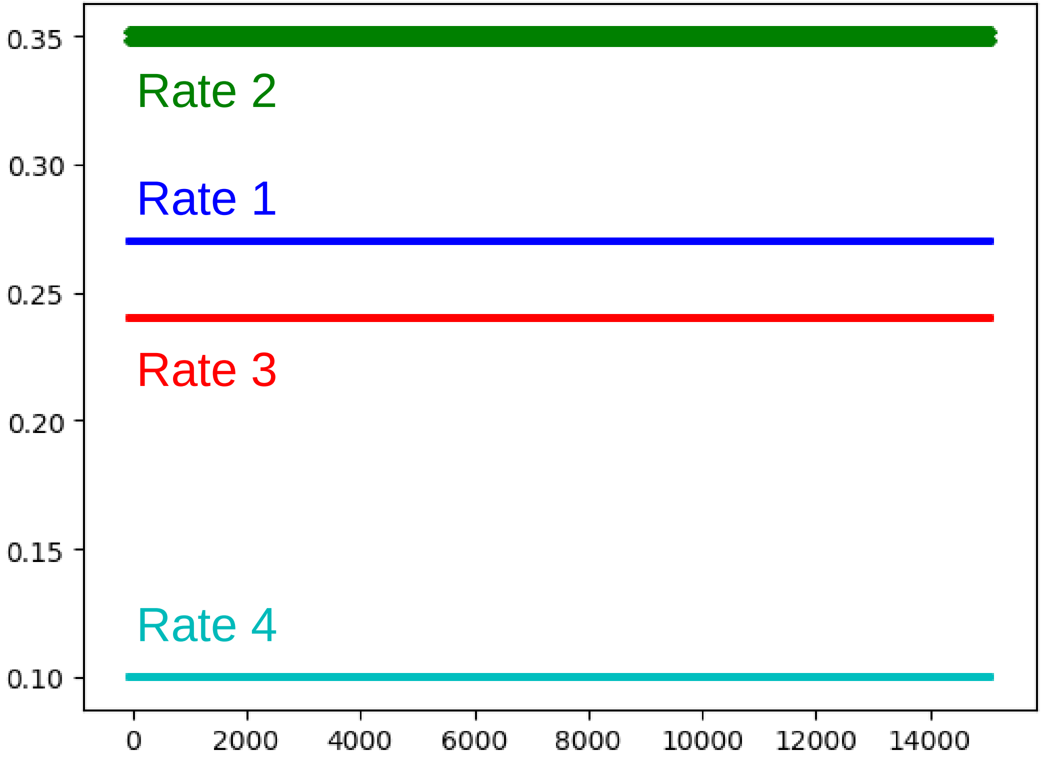


Fig.1 The true expectation throughput of each rate over time.

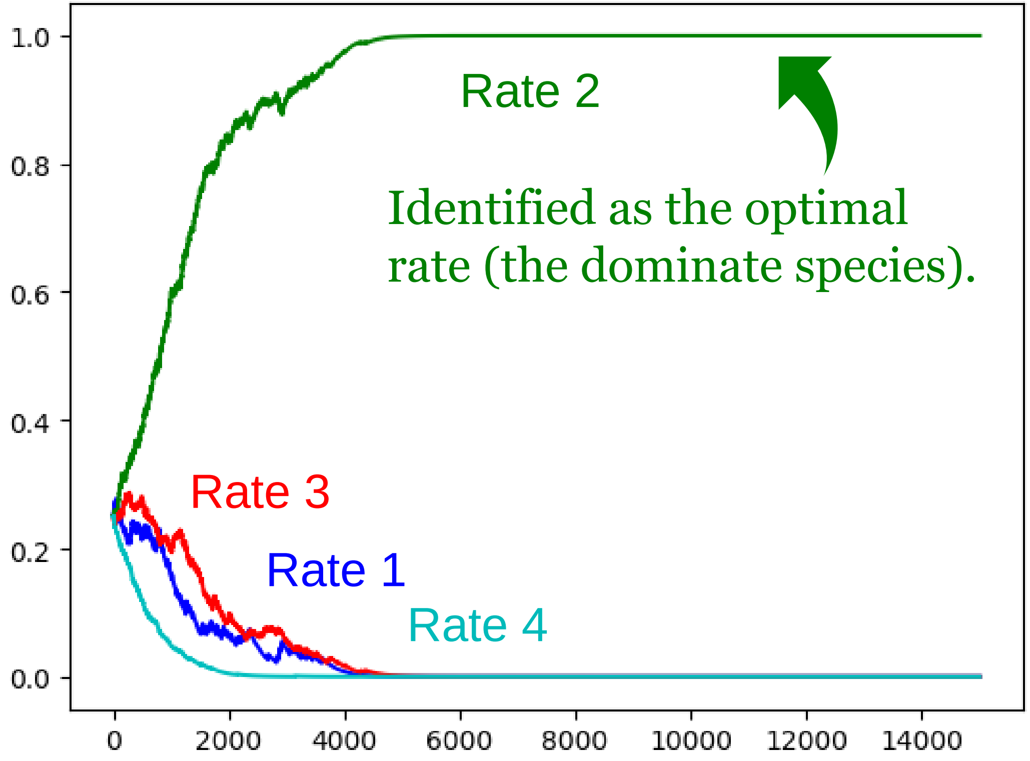


Fig.2 The probability of choosing each rate over time.

Github : https://github.com/endlesstory0428/RU2019F-CS530

Proposed Method – Crowding Effect

Goal: A proper constrain to exploitation

Lotka-Volterra competition dynamics with nonlinear crowding effect

$$\frac{dz_i}{dt} = z_i\left(a_i - \sum_{j=1}^{N} c_j z_j\right) - d_i z_i^{1+\delta}$$

z_i: Population (probability) of choosing i-th arm
a_i: natural growth rate, defined as θ_i
c_j: competitive factor, defined as θ_j
d_i: crowding effect factor, hyperparameter for this model
δ: nonlinear factor, hyperparameter for this model

This effect guarantees multi-species coexistence.

Gavina, Maica Krizna A., et al. "Multi-species coexistence in Lotka-Volterra competitive systems with crowding effects." *Scientific reports* 8.1 (2018): 1198.

Proposed Method – Crowding Effect

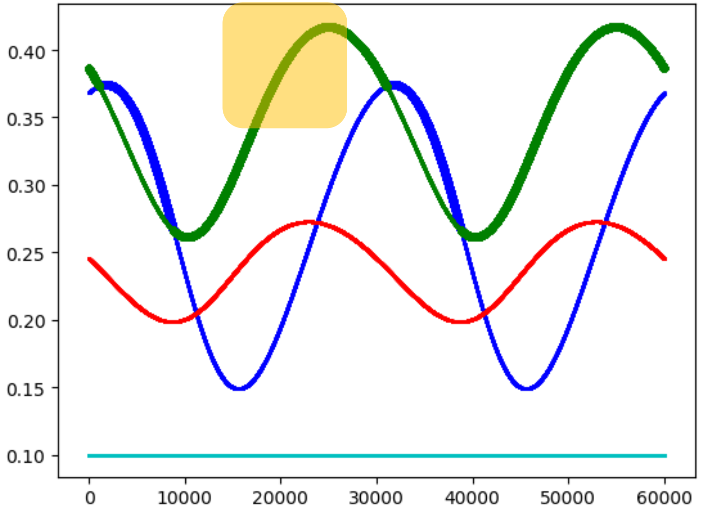


Fig.3 The true expectation throughput

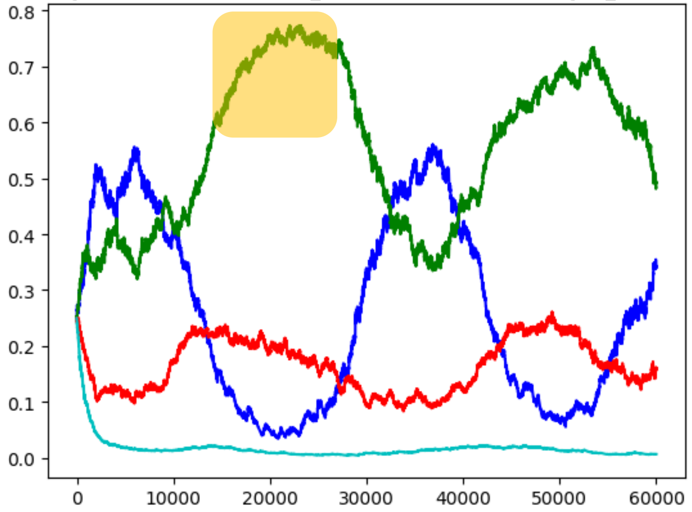


Fig.4 The probability of choosing a rate

When a rate provides the highest throughput in a short period, it will dominate. Namely, we will exploit it as much as possible. The others will be limited to a small number for future exploration.

Github : https://github.com/endlesstory0428/RU2019F-CS530

Proposed Method – Crowding Effect

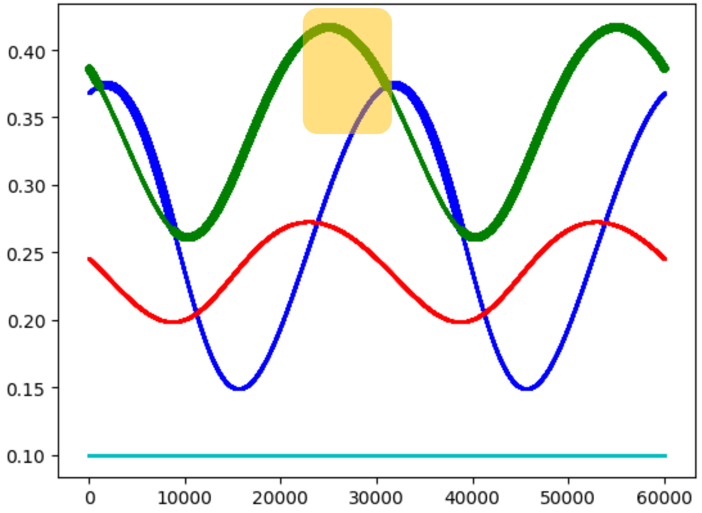


Fig.3 The true expectation throughput

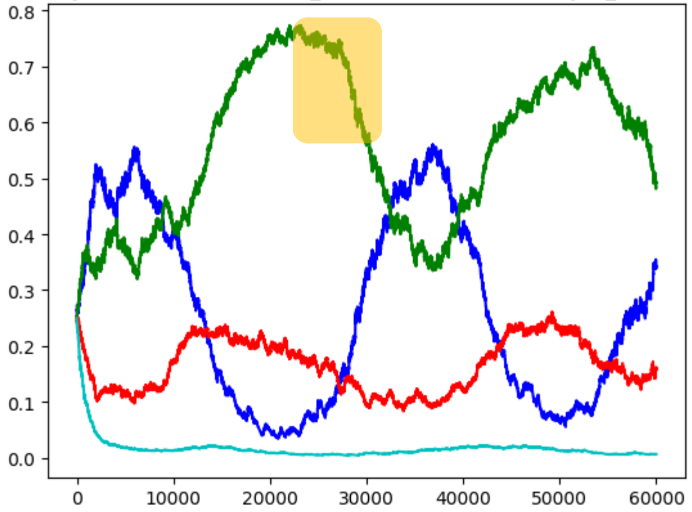


Fig.4 The probability of choosing a rate

Once its success rate θ_i is reduced, its natural growth rate $\mu_i = r_i \times \theta_i$ will also decrease, and due to the crowding effect, it will no longer be able to hold such a large population (probability of being selected). Namely, in this case, we will shift our focus to exploring potentially better choices.

Github : https://github.com/endlesstory0428/RU2019F-CS530

Proposed Method – Crowding Effect

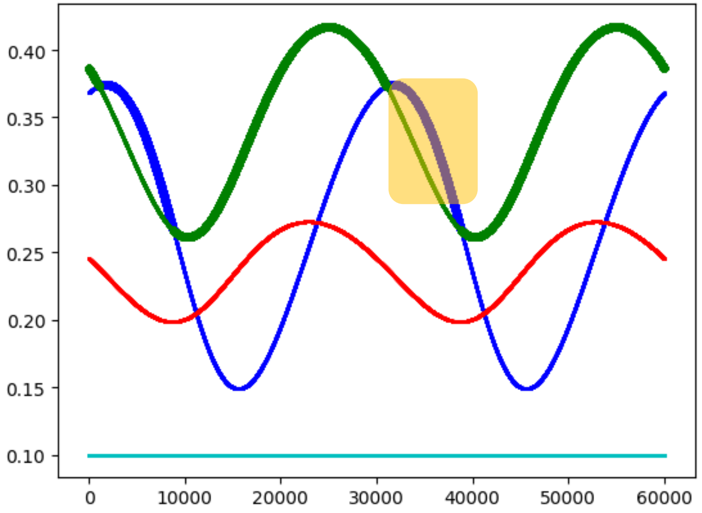


Fig.3 The true expectation throughput

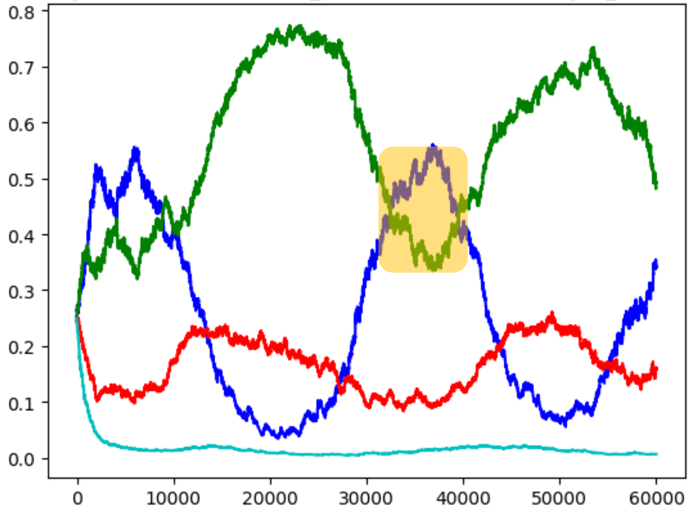


Fig.4 The probability of choosing a rate

When two (or more) rates are approximately equivalent in terms of throughput, we will keep collecting new data for future decision making.

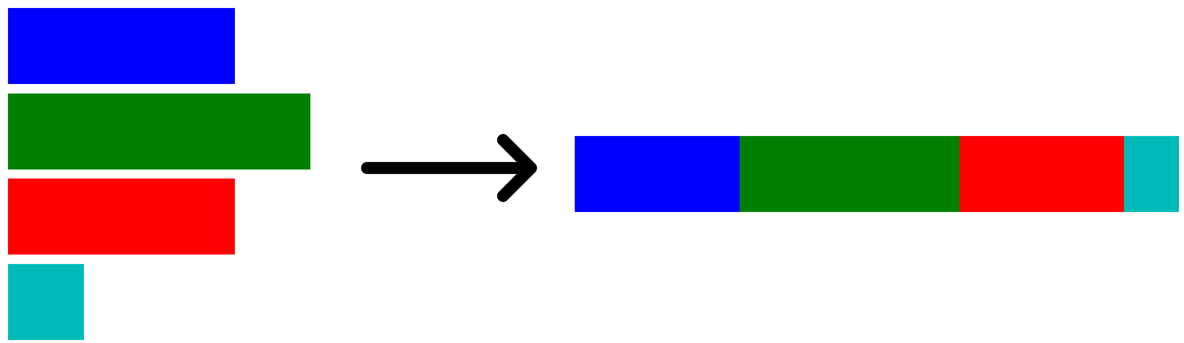
Note: when collecting new data, we are biased to a presumably better choice (blue one in this case).

Github : https://github.com/endlesstory0428/RU2019F-CS530

Proposed Method – Summary

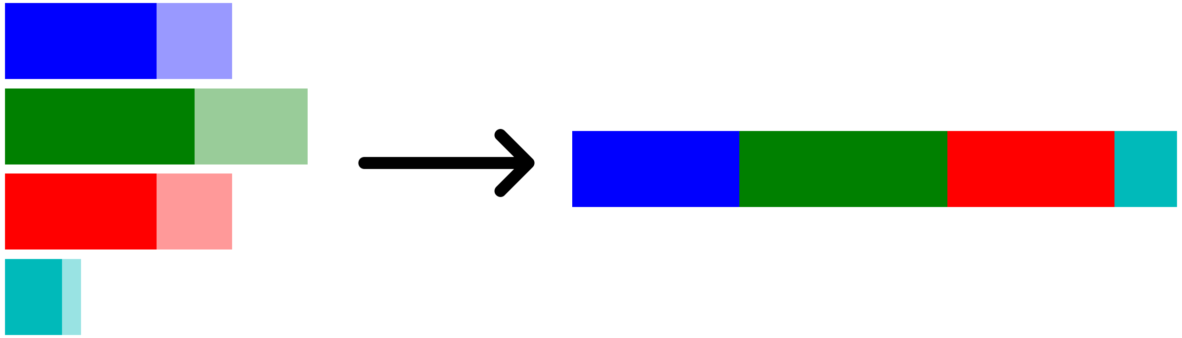
Rate Selection	Multi-Armed Bandits	Lotka-Volterra
Possible rates Success rates	Expectation of rewards	Natural growth rates
Success/fail signals	Realization of rewards	Temporary population growth
Probability of being selected		Population of a species
Identifying the optimal rate	Identifying the optimal arm	Identifying the dominate (fittest) species
Maximize throughput over time	Maximize accumulative rewards	The survival of the fittest
Exploitation and exploration		Dynamics of an ecosystem

Algorithm



```
init population[0:N] = 1
while True:
     probability = population / sum(population)
     i = selecting an index with probability
     reward = observing transmission outcome
     dPopulation = -b * d * power(population, 1 + delta)
     factor = b * reward
     dPopulation[i] += factor / (1 – factor) * sum(population)
     population += population + dPopulation
```

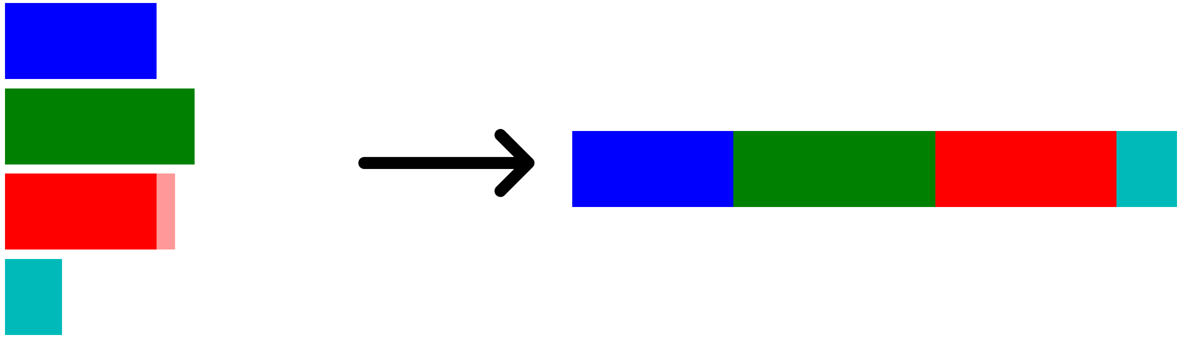
Algorithm



```
init population[0:N] = 1
while True:
     probability = population / sum(population)
     i = selecting an index with probability
     reward = observing transmission outcome
     dPopulation = -b * d * power(population, 1 + delta)
     factor = b * reward
     dPopulation[i] += factor / (1 - factor) * sum(population)
     population += population + dPopulation
```

$$dz_i = dt(z_i\left(a_i - \sum_{j=1}^{N} c_j z_j\right) - d_i z_i^{1+\delta})$$

Algorithm



```
init population[0:N] = 1
while True:
     probability = population / sum(population)
     i = selecting an index with probability
     reward = observing transmission outcome
     dPopulation = -b * d * power(population, 1 + delta)
     factor = b * reward
     dPopulation[i] += factor / (1 – factor) * sum(population)
     population += population + dPopulation
```

$$dz_i = dt(z_i\left(a_i - \sum_{j=1}^{N} c_j z_j\right) - d_i z_i^{1+\delta})$$

Experiment

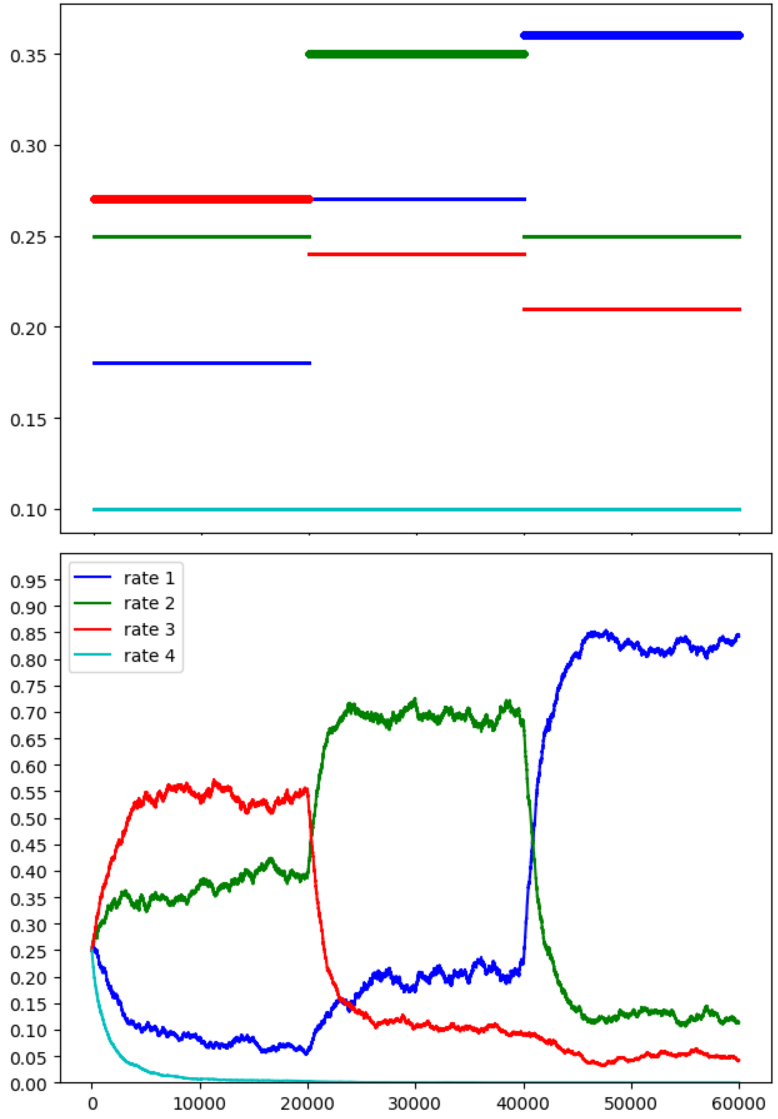


Fig.5 The probability of choosing a rate

b controls the speed of adapting to environment.

A large *b* may cause jitters.

$b = 0.01$
$\text{d} = 0.9$
$\delta = 0.2$

$b = 0.003$
$\text{d} = 0.9$
$\delta = 0.2$

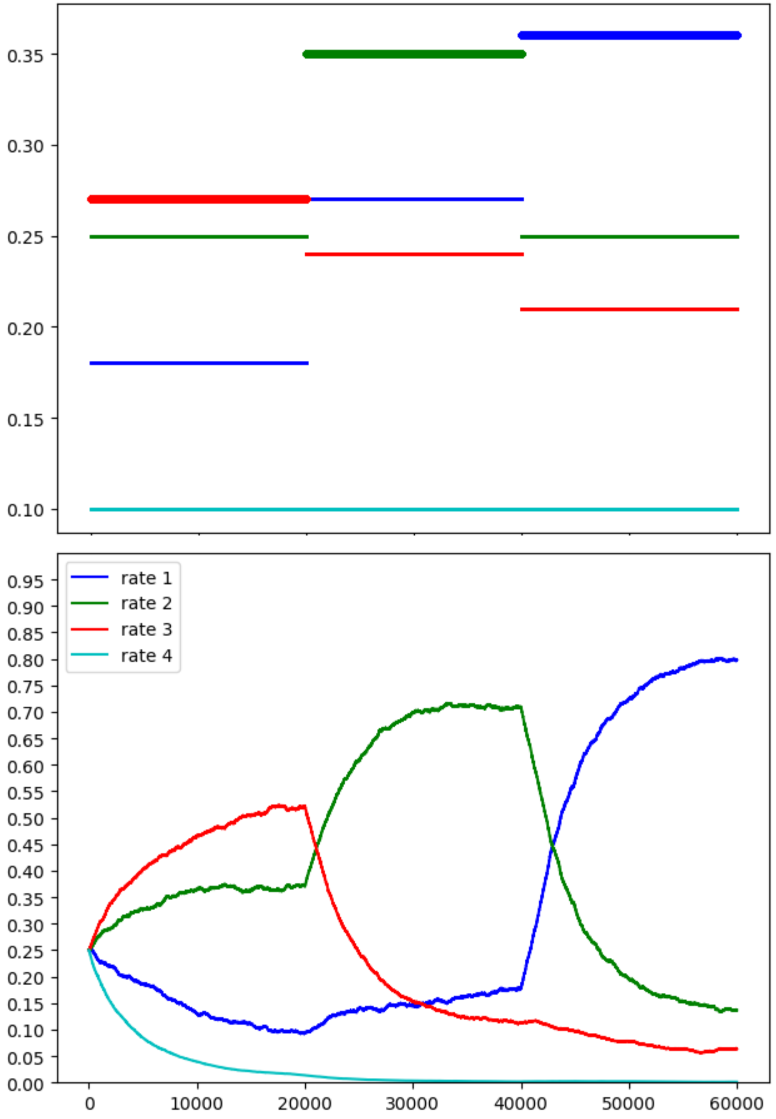


Fig.6 The probability of choosing a rate

Github : https://github.com/endlesstory0428/RU2019F-CS530

Experiment

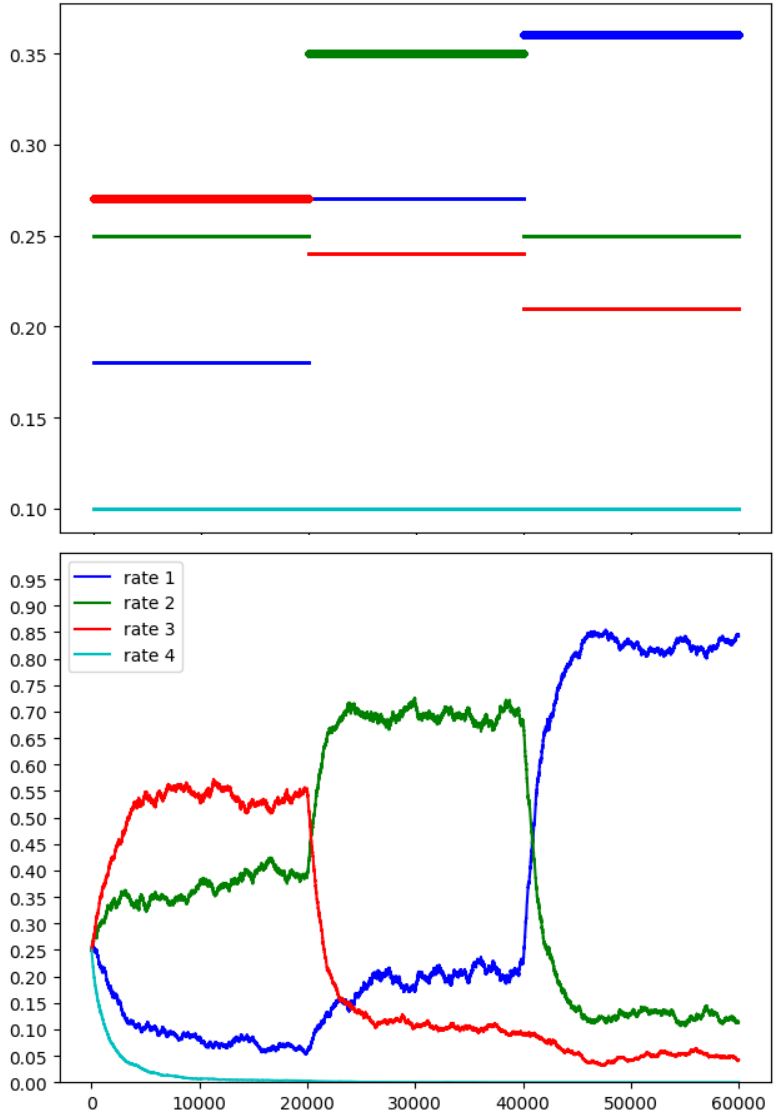


Fig.5 The probability of choosing a rate

δ controls the strength of pushing probability to $\frac{1}{N}$.

A small δ may cause "extinction" of some rates.

$b = 0.01$
d = 0.9
$\delta = 0.2$

$b = 0.01$
d = 0.9
$\delta = 0.5$

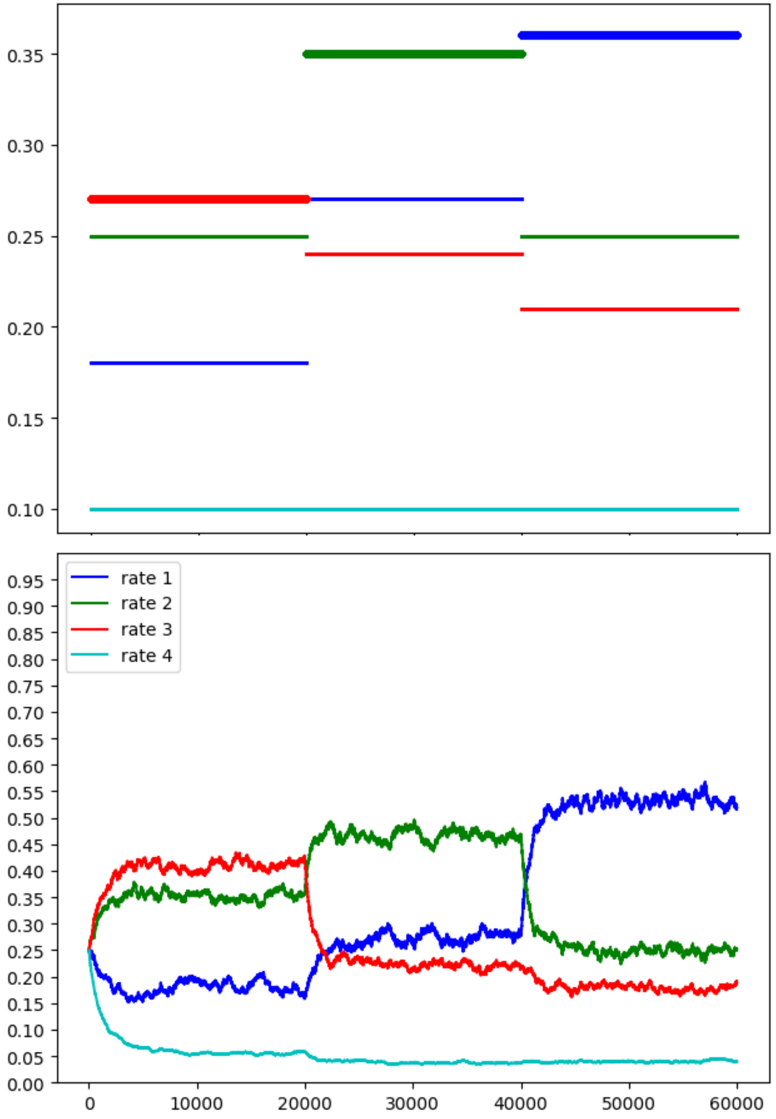


Fig.7 The probability of choosing a rate

Github : https://github.com/endlesstory0428/RU2019F-CS530

Experiment

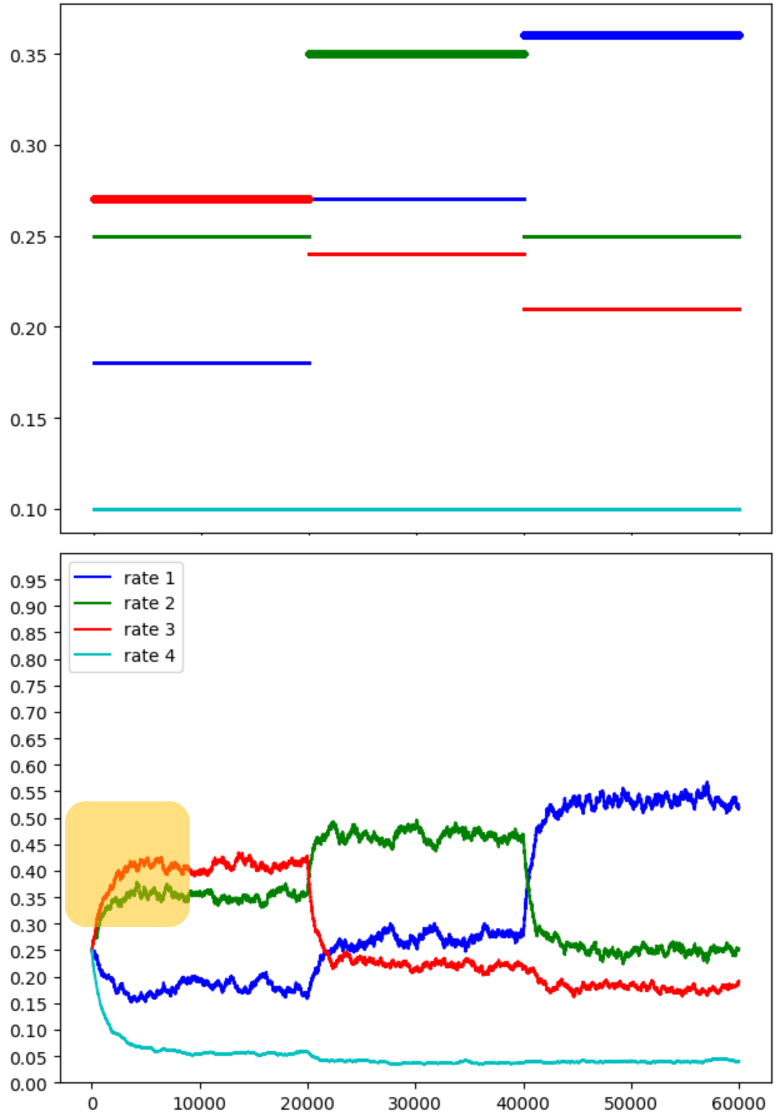


Fig.7 The probability of choosing a rate

d influences the size of convergence population.

Population ≈ 1

Population $\approx 10^5$

$b = 0.01$
$\mathrm{d} = 0.9$
$\delta = 0.5$

$b = 0.01$
$\mathrm{d} = 0.001$
$\delta = 0.5$

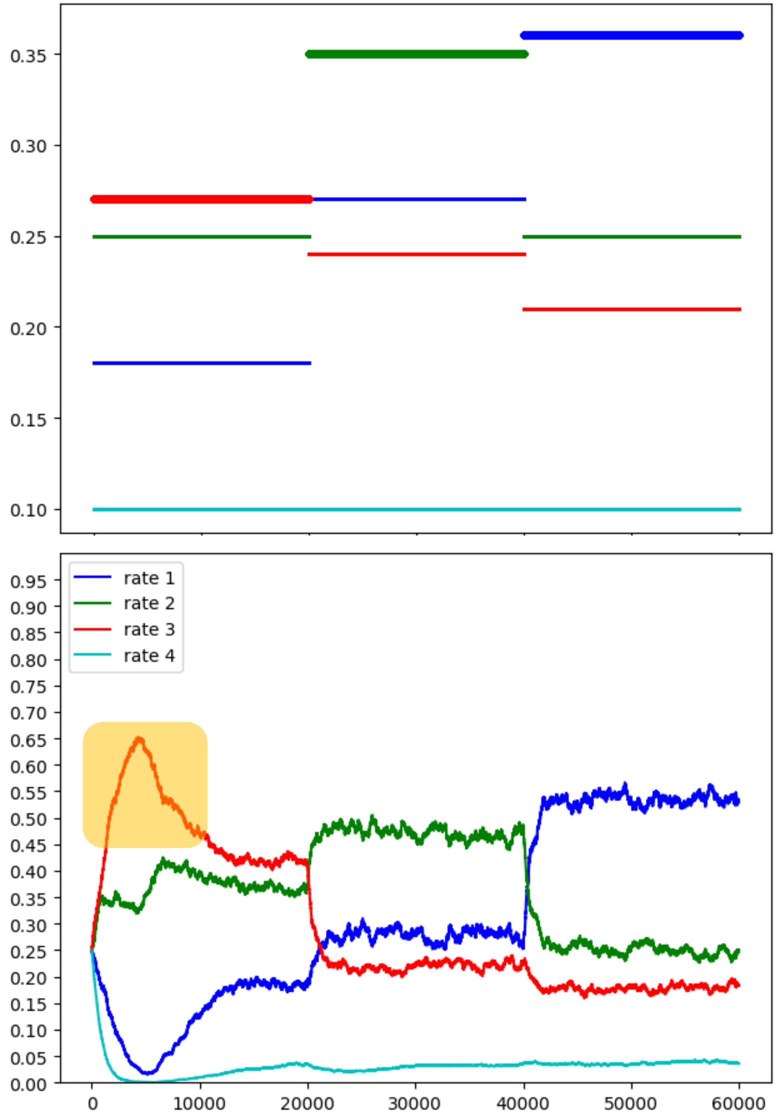


Fig.8 The probability of choosing a rate

Github : https://github.com/endlesstory0428/RU2019F-CS530

Experiment

Randomly select 4 rates: 73%
Randomly select first 3 rates: 90%
Keep selecting Rate 1: 86%
Keep selecting Rate 2: 92%
Keep selecting Rate 3: 91%
Keep selecting Rate 4: 22%

Our algorithm: 94%

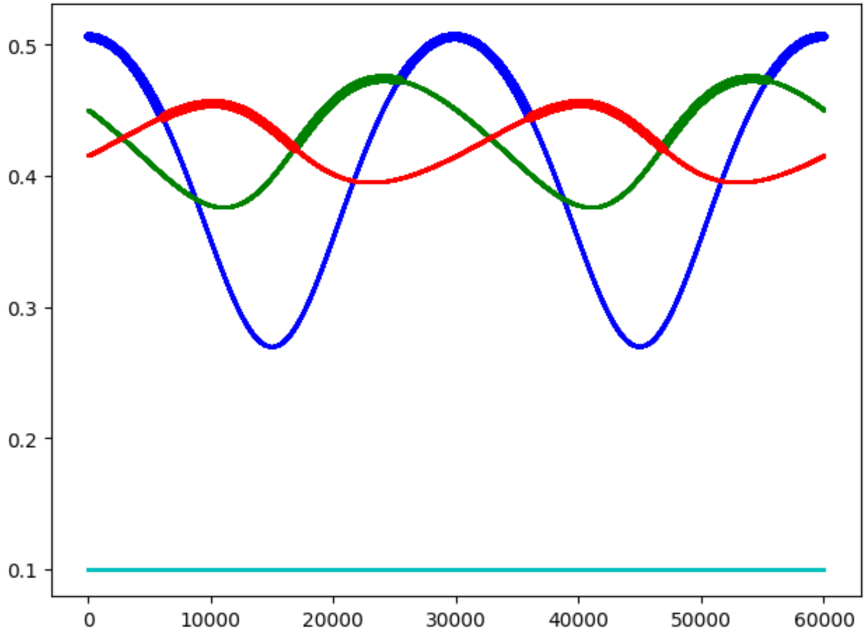


Fig.8 The true expectation throughput

$b = 0.01$
$d = 0.1$
$\delta = 0.2$

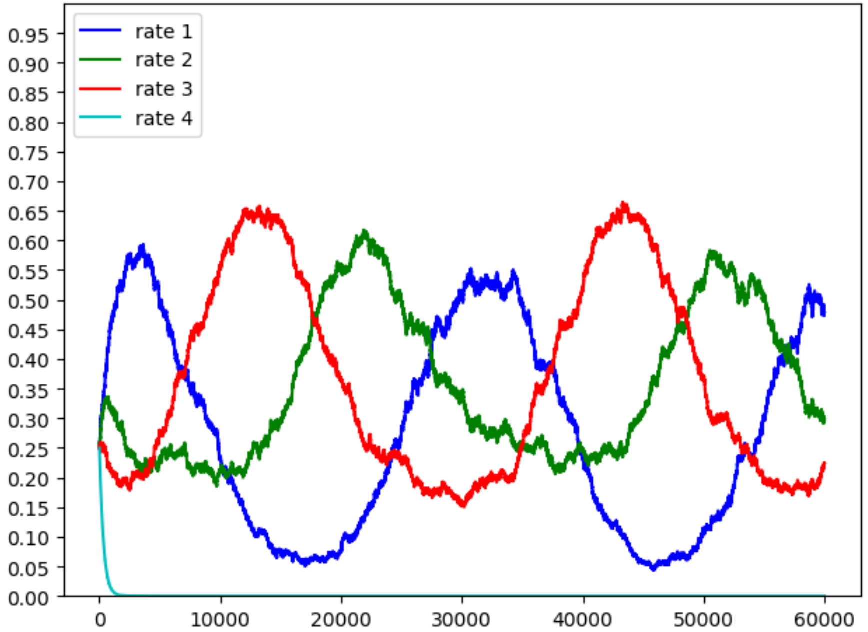


Fig.10 The probability of choosing a rate

Github : https://github.com/endlesstory0428/RU2019F-CS530

Future Work

Joint Bandit Problem

If we believe $r_i\theta_i$ is too small to be optimal because of our estimation $\widehat{\theta}_i$, then $\forall j > i, \widehat{\theta}_j < \widehat{\theta}_i$, which may help us eliminate some choices.

Meta-learning

Choosing hyper-parameters b, d, δ can be considered as a meta-level selection problem. We can apply this framework to it, but how to define "rewards"?