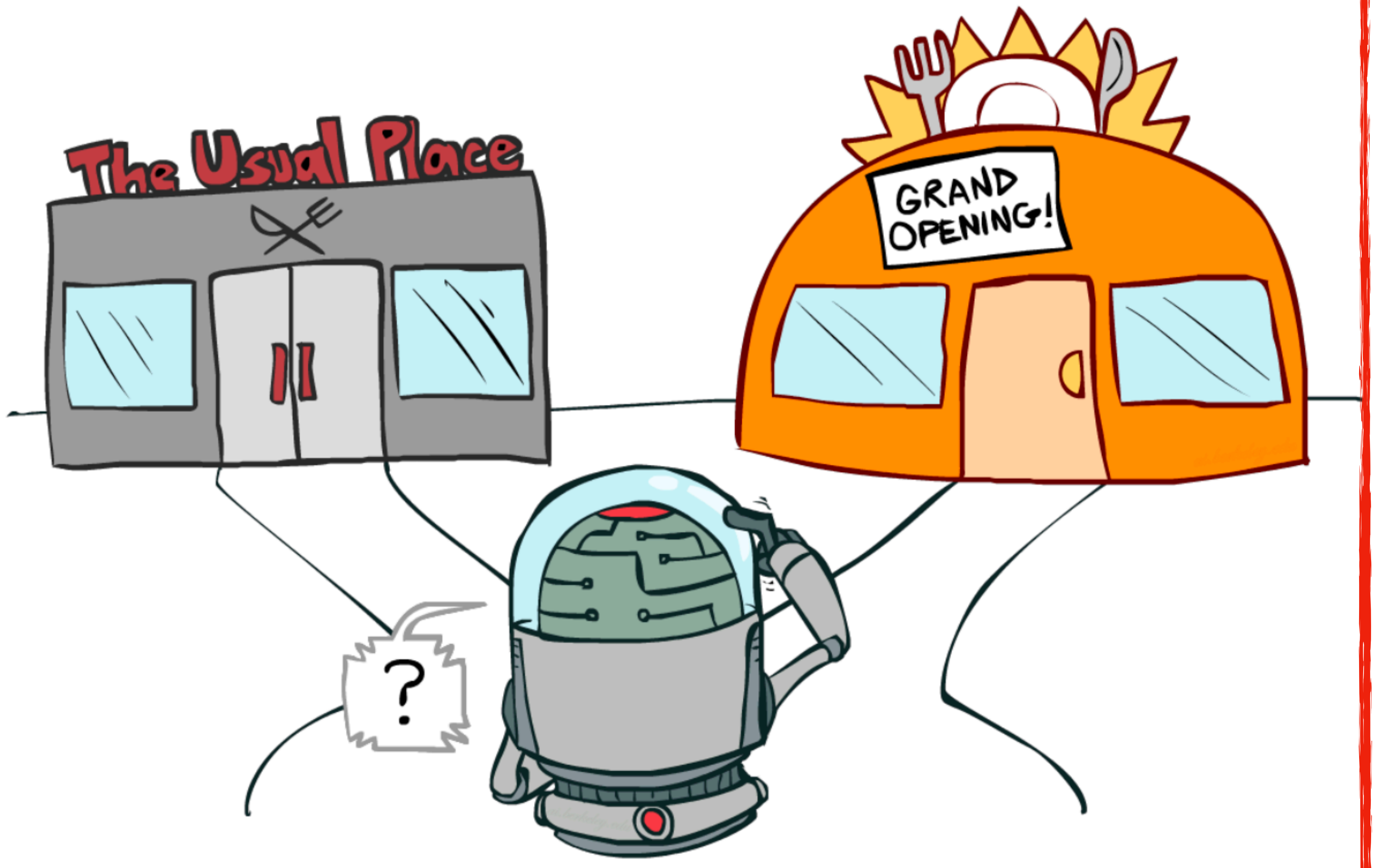
Dealing with Uncertainty: Part 1

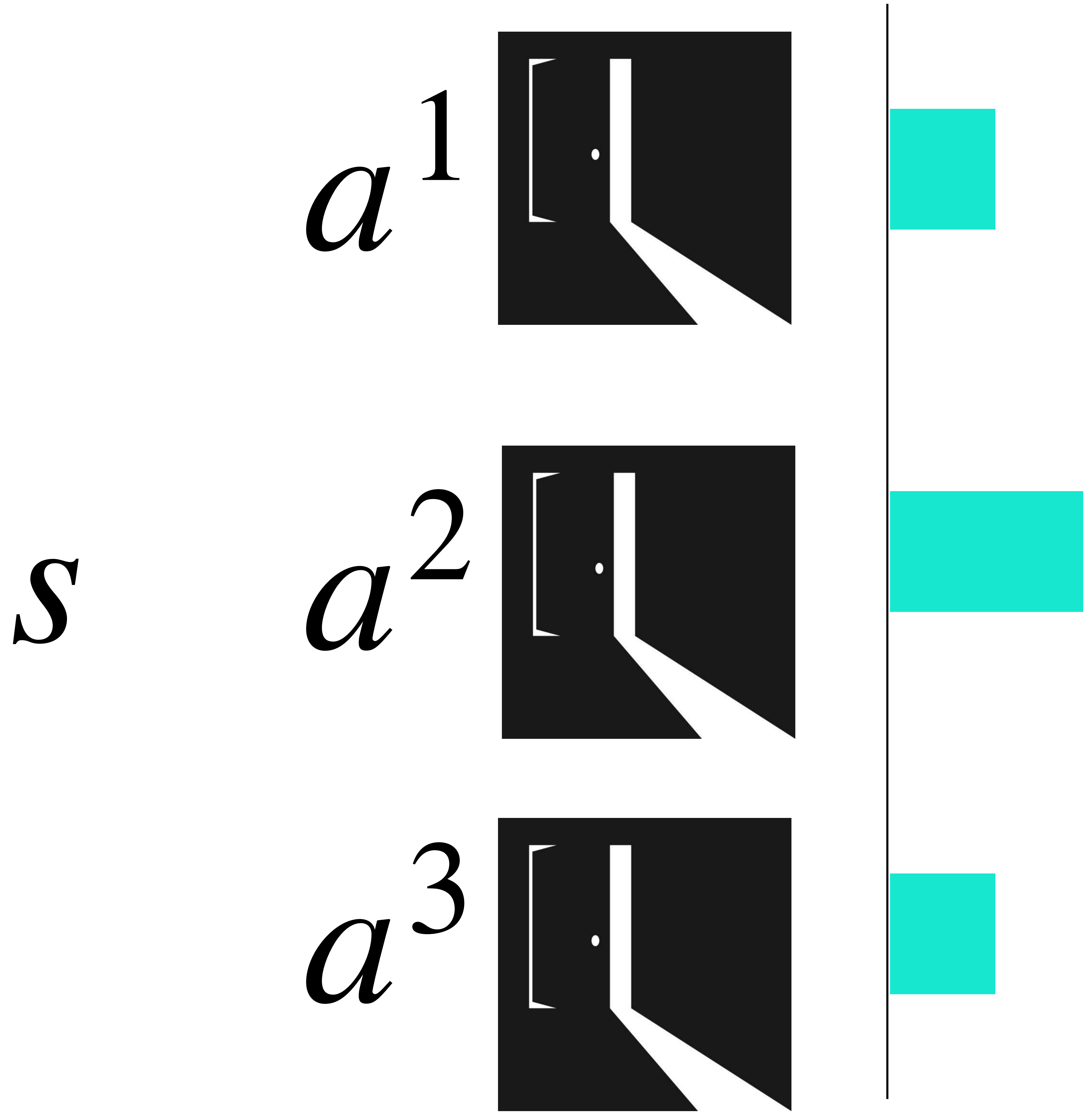
Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Two Ingredients of RL



Exploration Exploitation



Estimate Values $Q(s, a)$


Uncertainty

Types of uncertainty

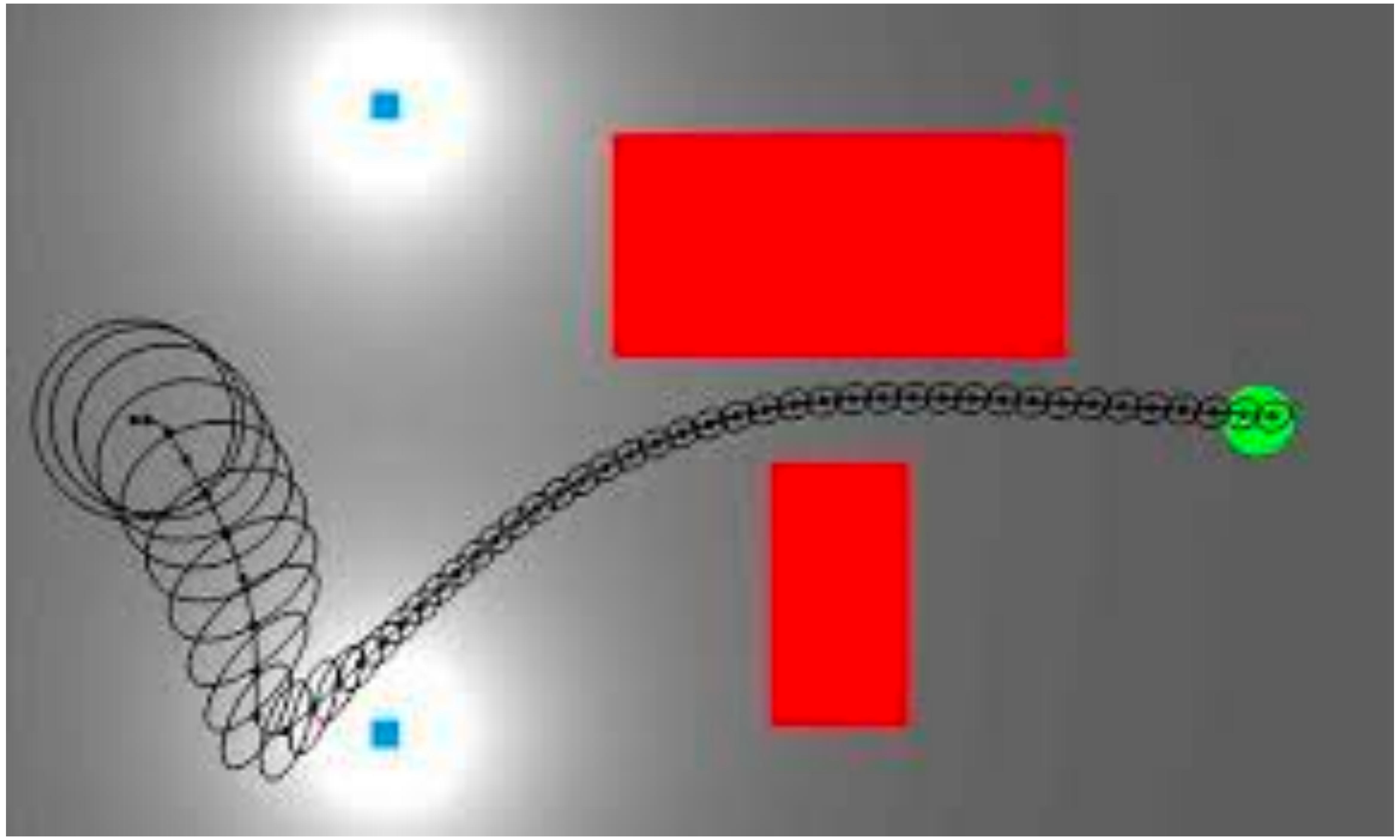
Aleatoric uncertainty

(Can't change this uncertainty)

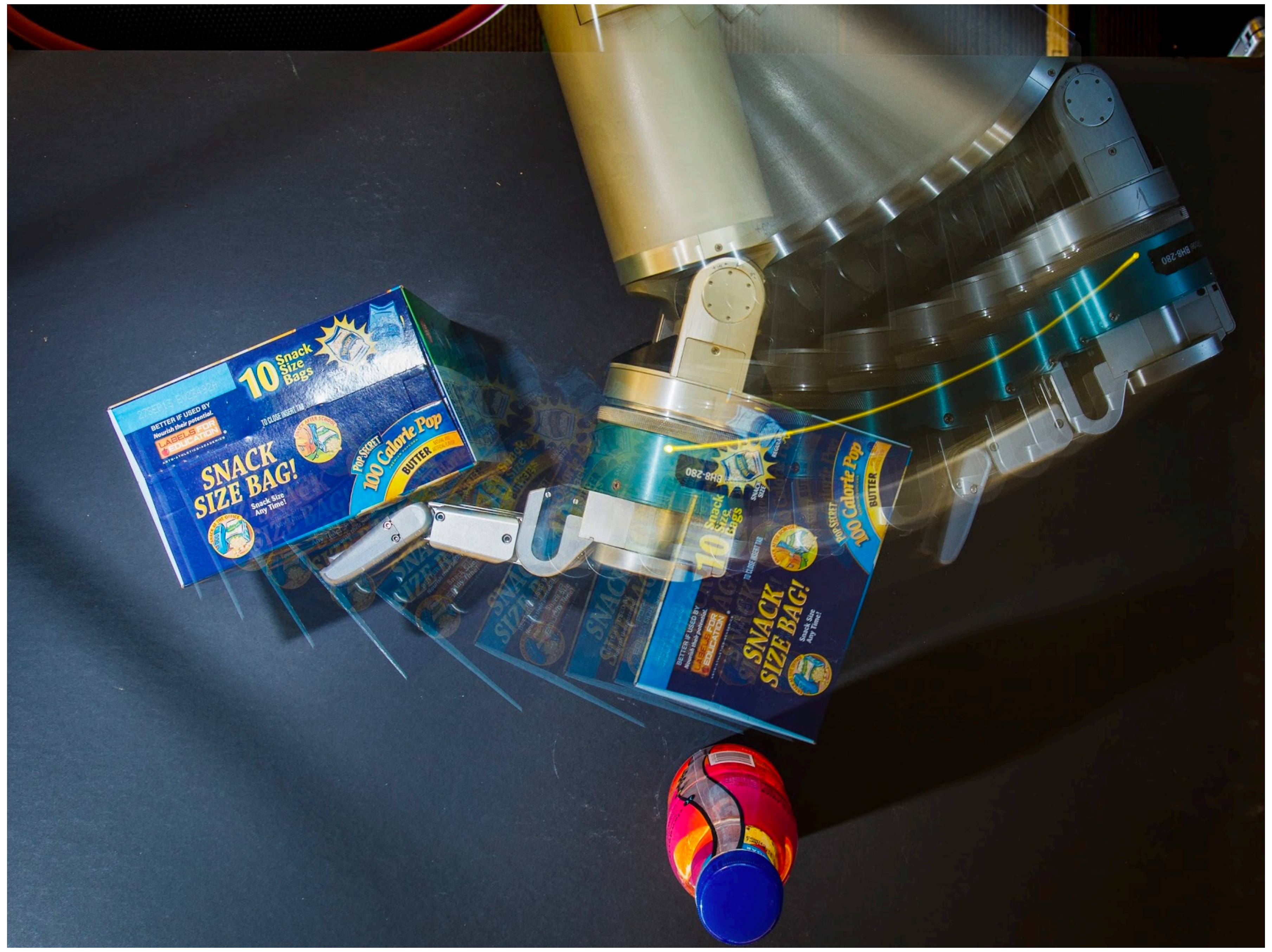
Epistemic uncertainty

(Acquire knowledge!)

Epistemic Uncertainty



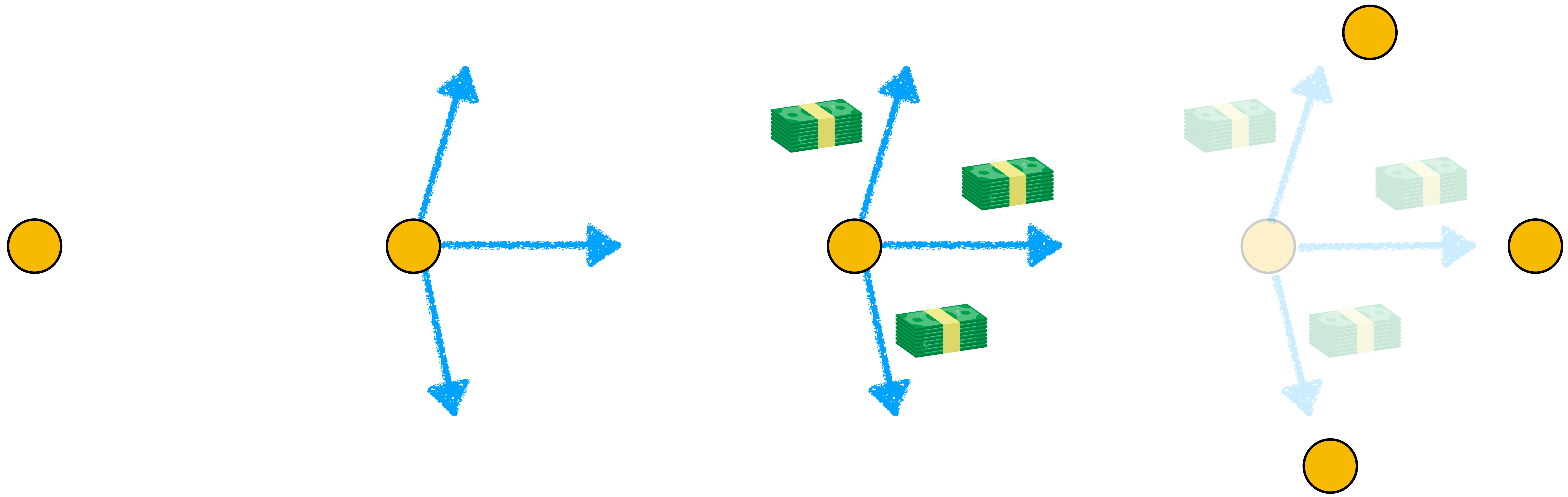
Uncertain about state



Uncertain about transitions

Can be uncertain about any of these things!

$$< S, A, C, \mathcal{T} >$$



What do we want to do about uncertainty?

Pure Exploration	Optimally explore / exploit	Pure Exploitation
Collapse uncertainty as quickly as possible	Take information gathering steps, but be robust along the way	Be robust against uncertainty
20 questions	Life!	UAV flying in wind

Activity!

Categorize the following robot applications!

0 — 5 — 10

Pure Exploration — Optimally explore / exploit — Pure Exploitation

Self-driving through an intersection

Assistive manipulation via shared autonomy

UAV autonomously mapping a building

Grasping an object on the top-shelf

Off-road driving through terrain

Think-Pair-Share

Think (30 sec): Categorize the following robotics application from 0 (pure exploration) to 10 (pure exploitation)

Pair: Find a partner

Share (45 sec): Partners exchange ideas

Self-driving through an intersection

Assistive manipulation via shared autonomy

UAV autonomously mapping a building

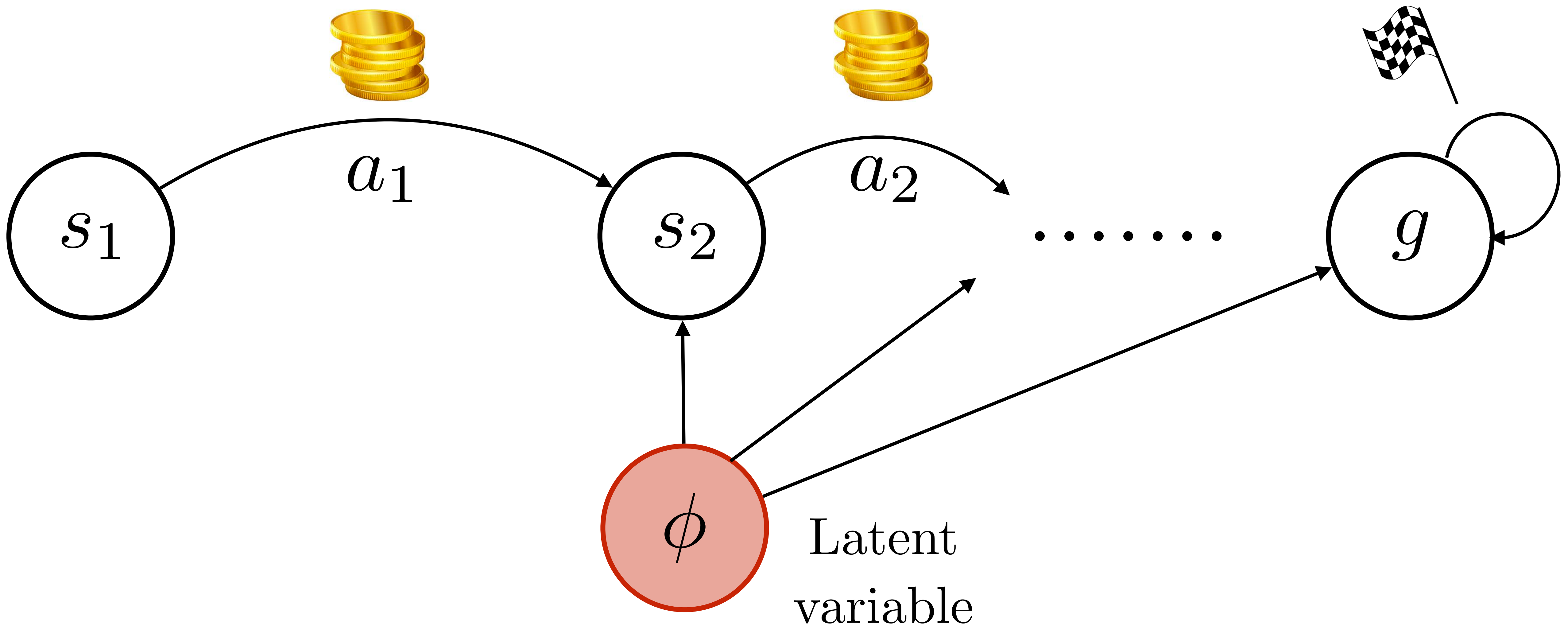
Grasping an object on the top-shelf

Off-road driving through terrain

But what is the *optimal* exploration-exploitation algorithm?

Belief Space Planning

Can frame optimal exploration / exploitation as Belief Space Planning



State: $s \in \mathcal{S}$

(fixed latent variable) $\phi \in \Phi$

Transition: $P(s'|s, a, \phi)$

Prior: $P(\phi)$

Bayes Optimality:

The Holy Grail

GAME OVER!

Belief Space Planning is NP-Hard at best, undecidable at worst

Need to relax our problem!

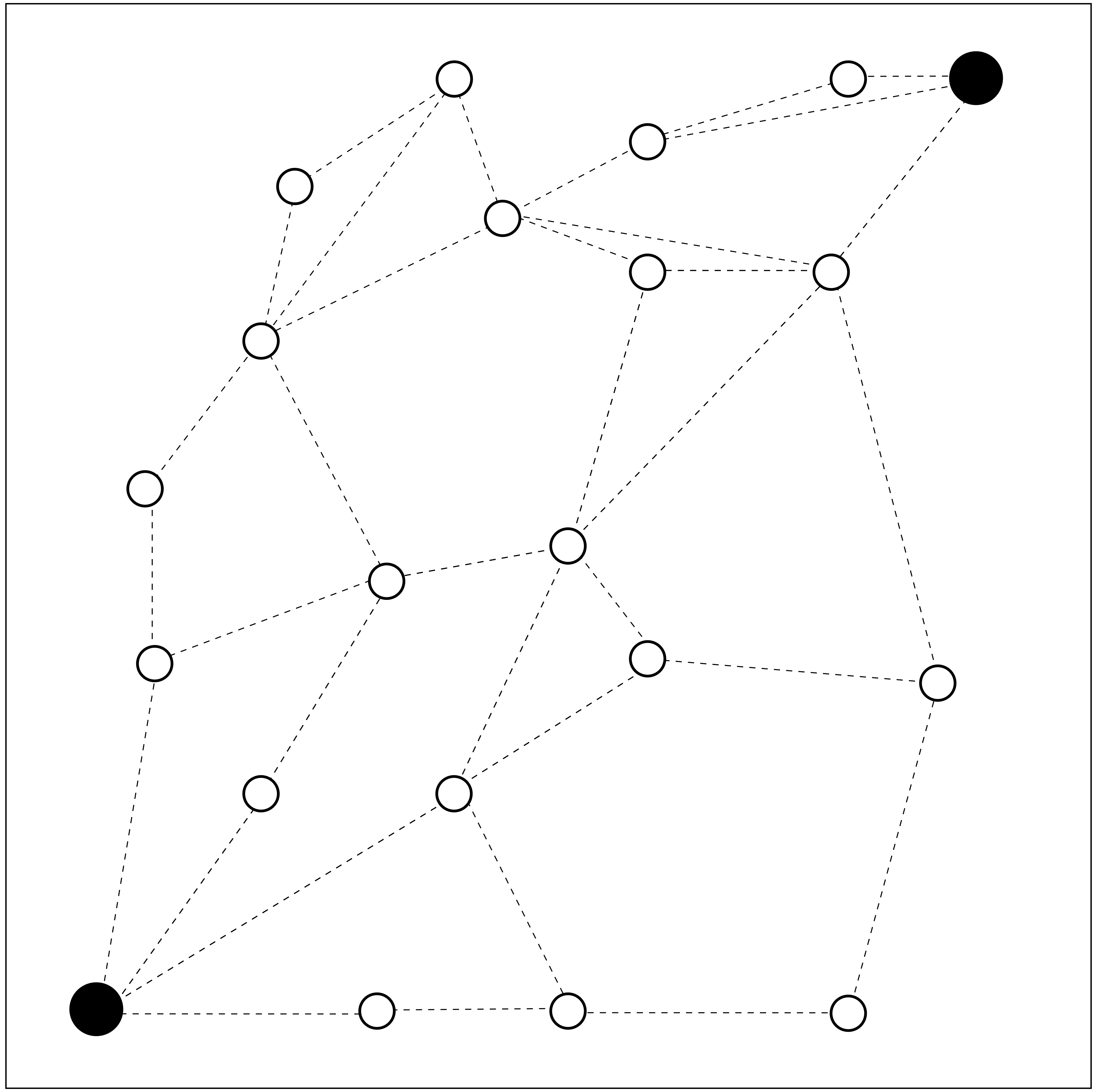
A Tale of Relaxations

Optimism in the Face of Uncertainty (OFU)

The Lazy Shortest Path Problem

Let's say you have a graph where you don't know the cost of edges. (Can be 0 or 1)

Find the shortest path while minimizing number of edges queried



An *really simple* algorithm

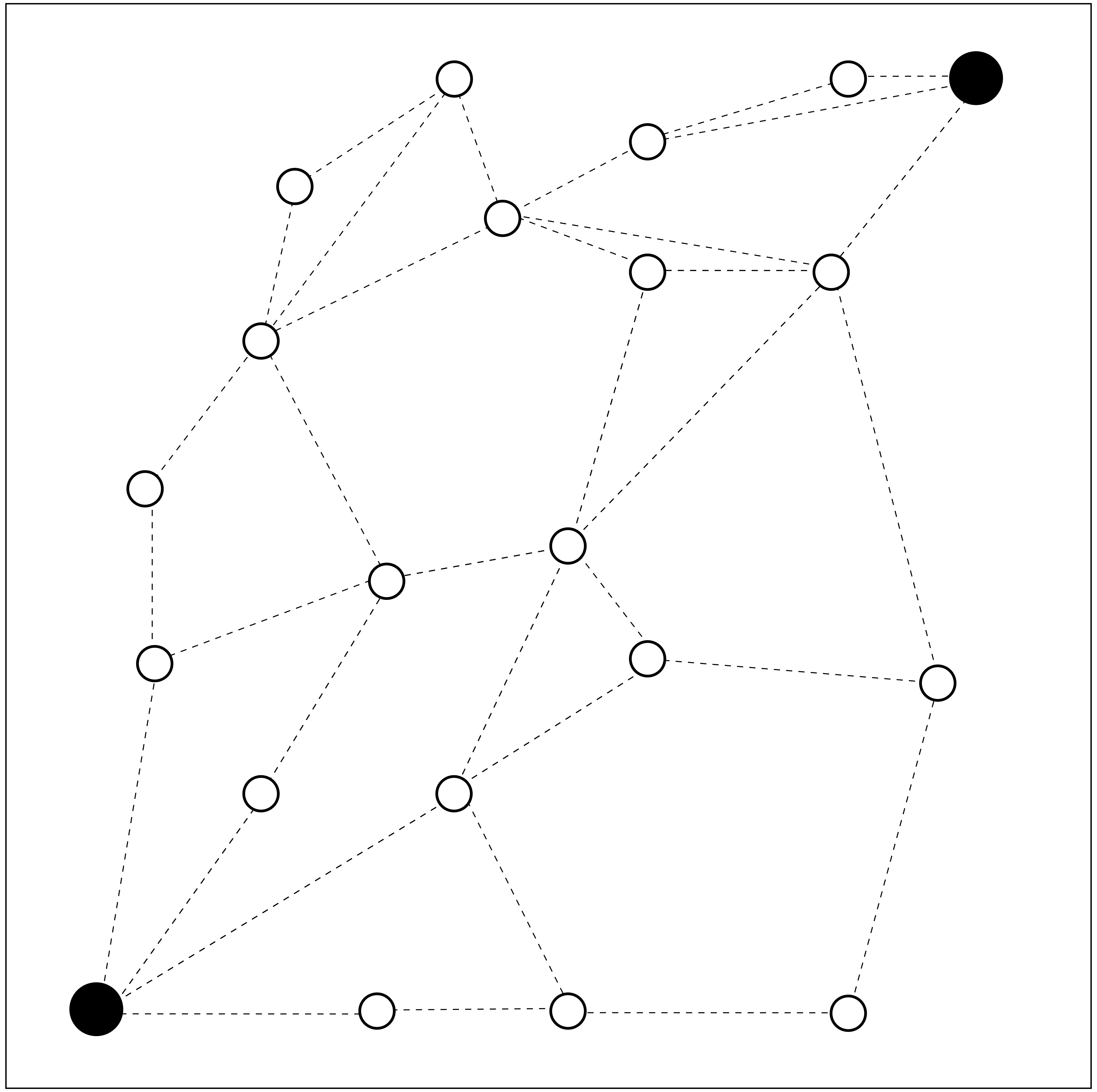
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



An *really simple* algorithm

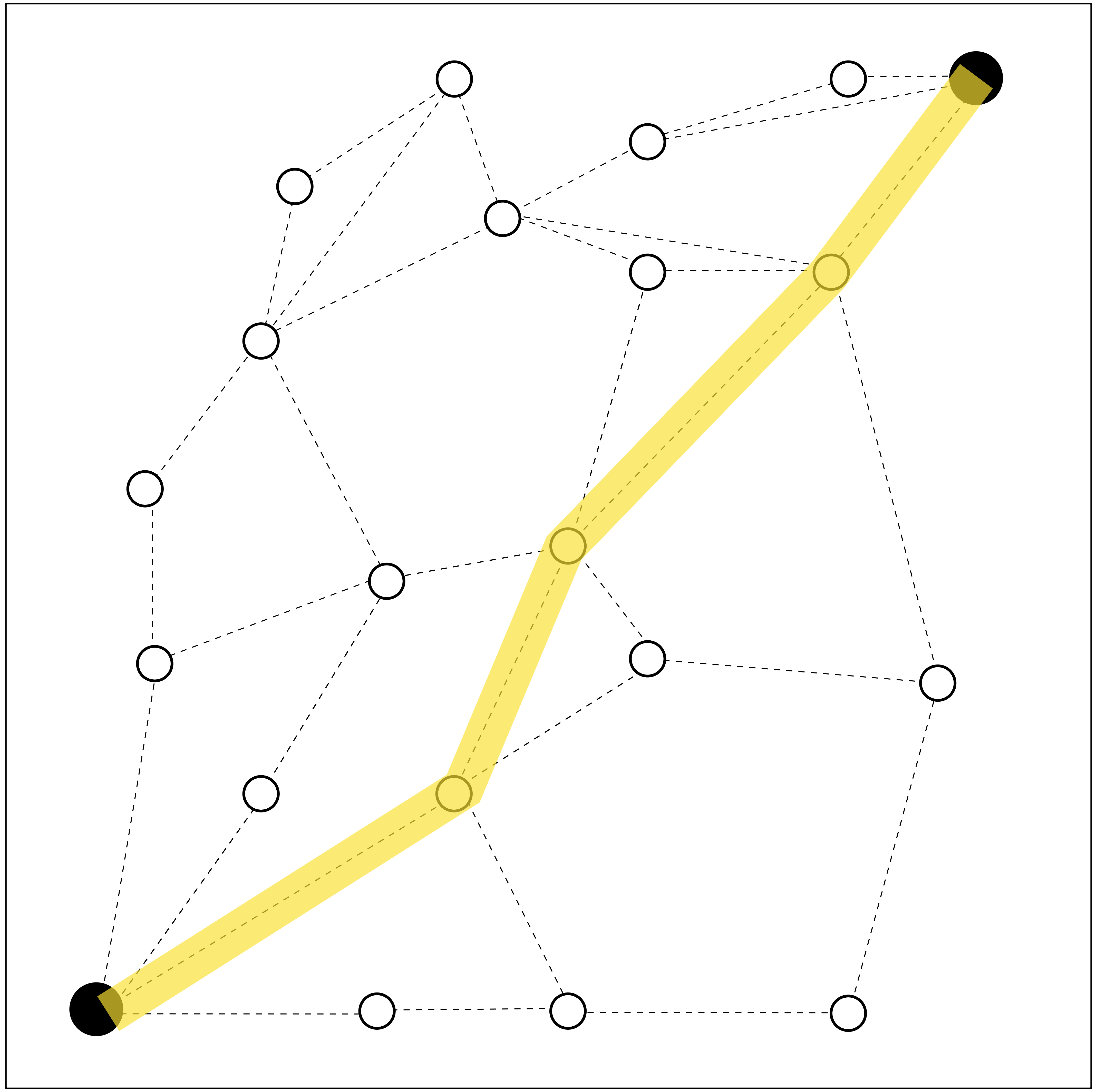
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



An *really simple* algorithm

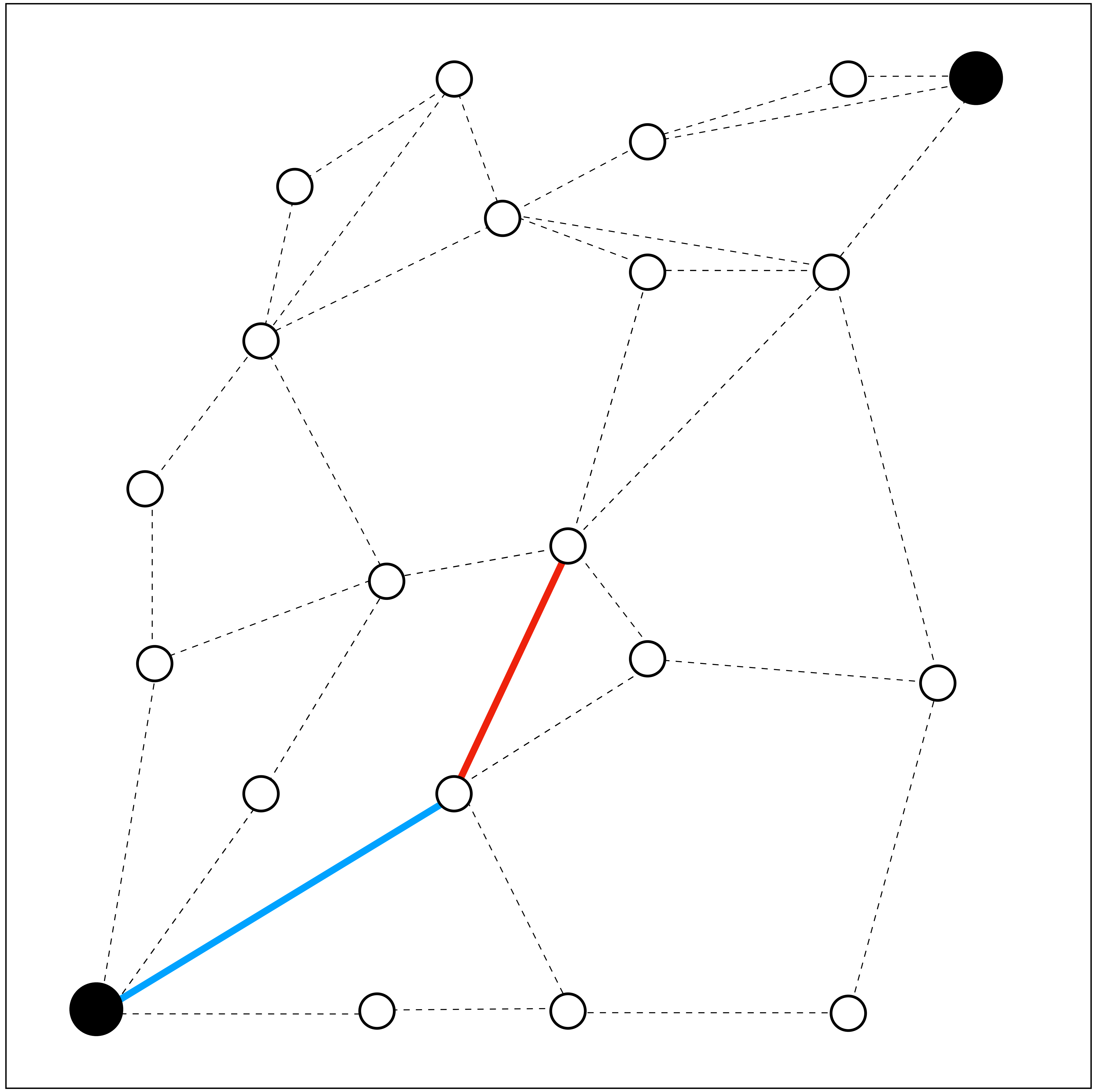
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



An *really simple* algorithm

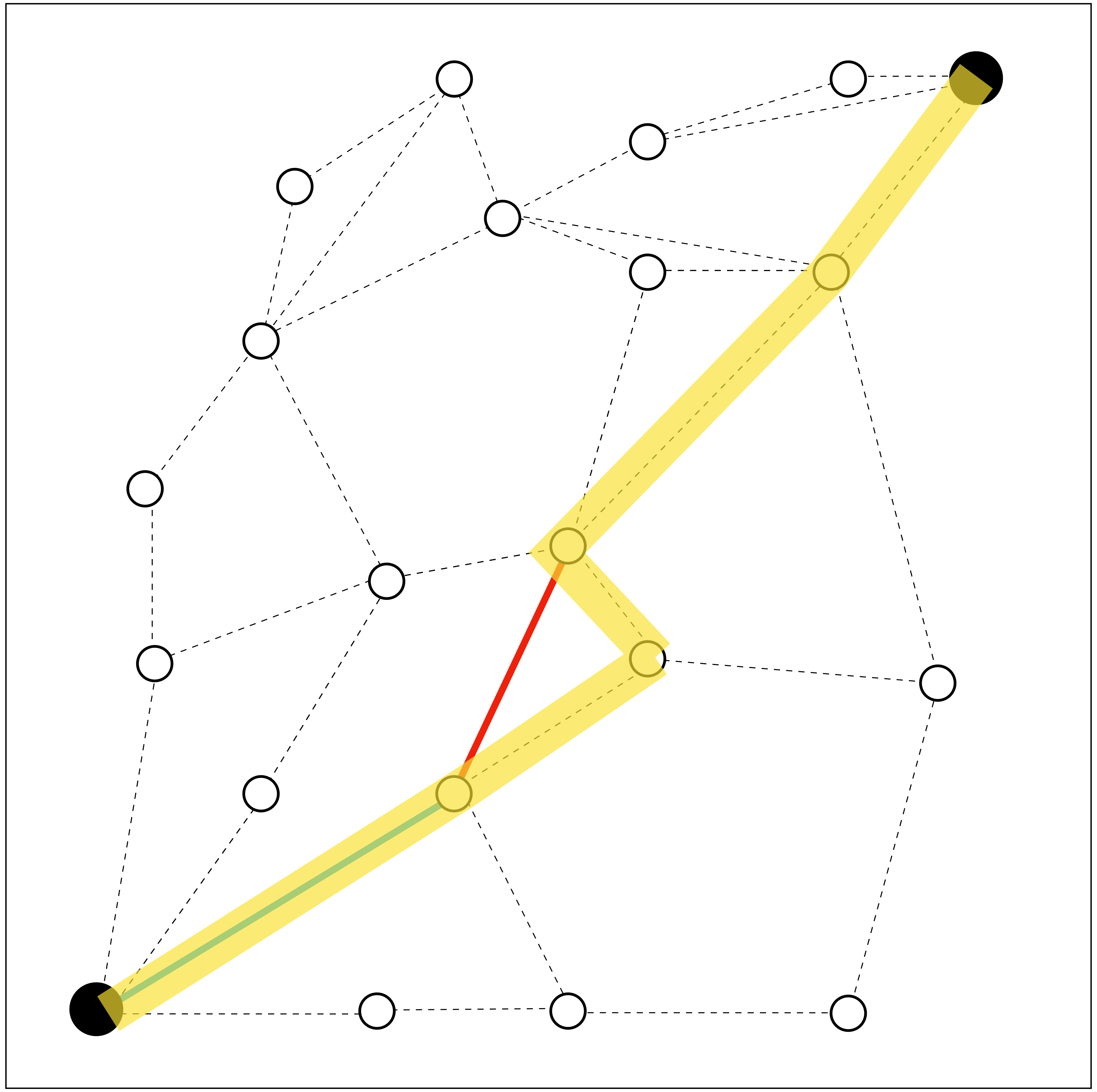
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



An *really simple* algorithm

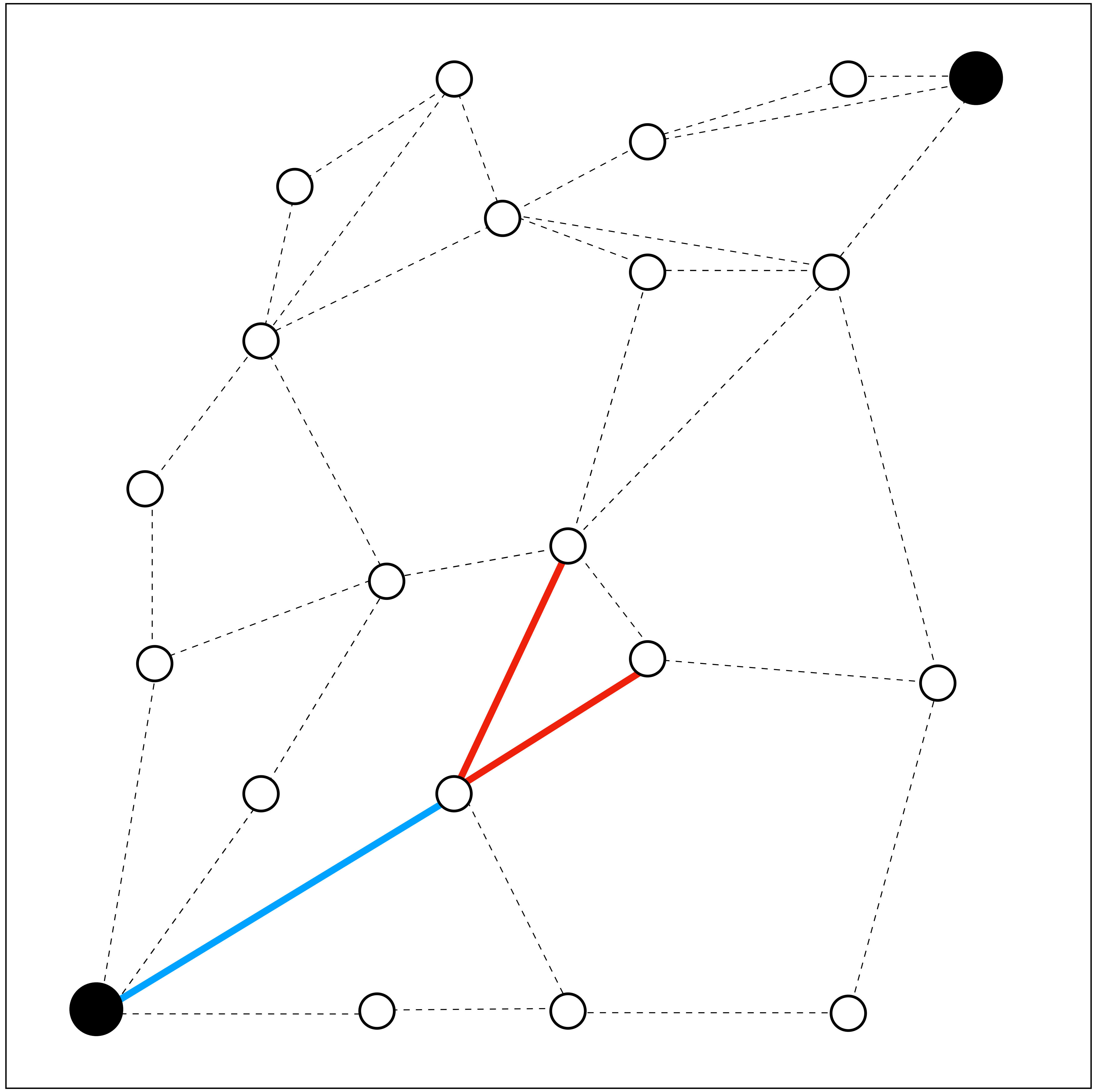
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



An *really simple* algorithm

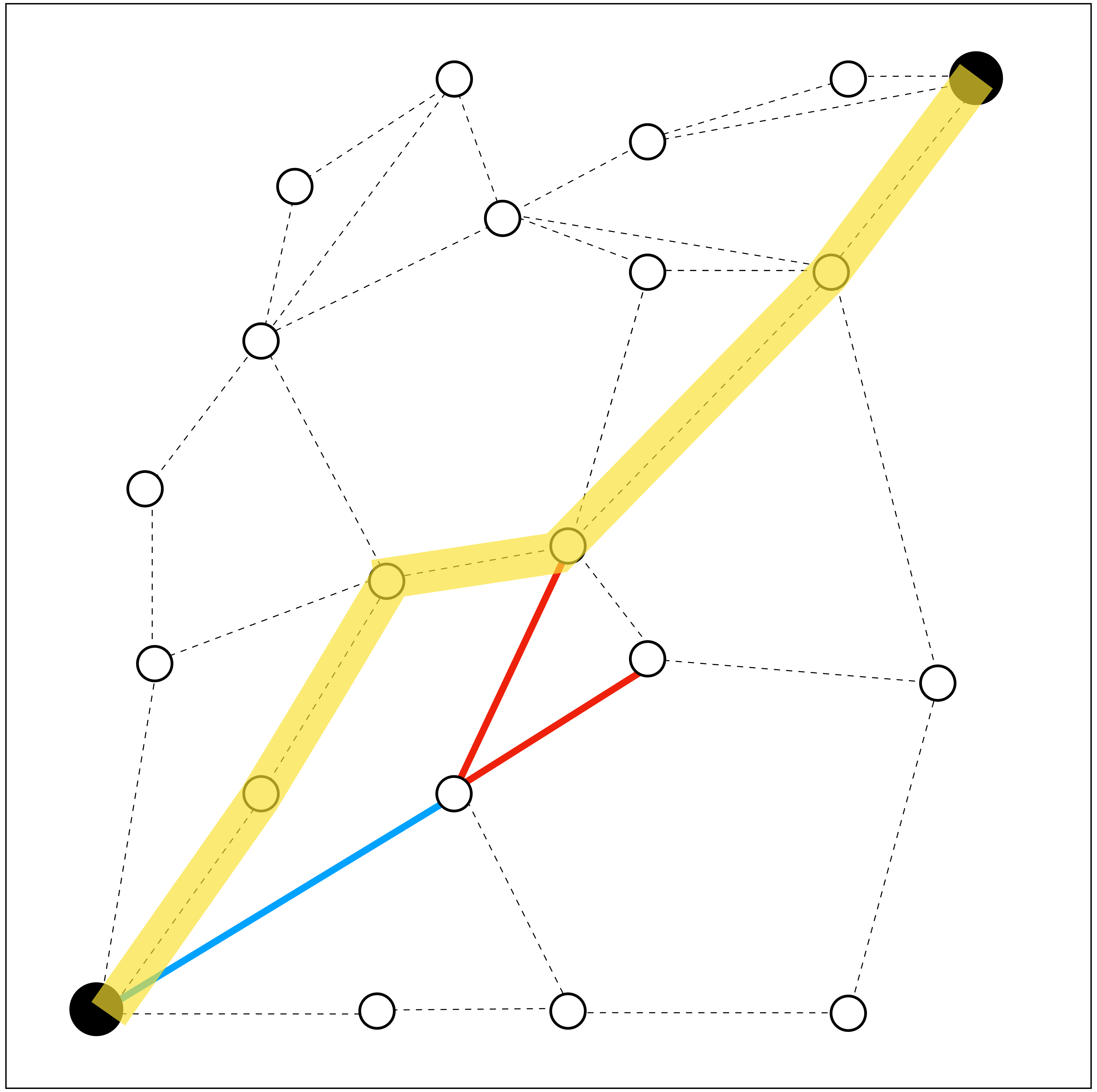
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



An *really simple* algorithm

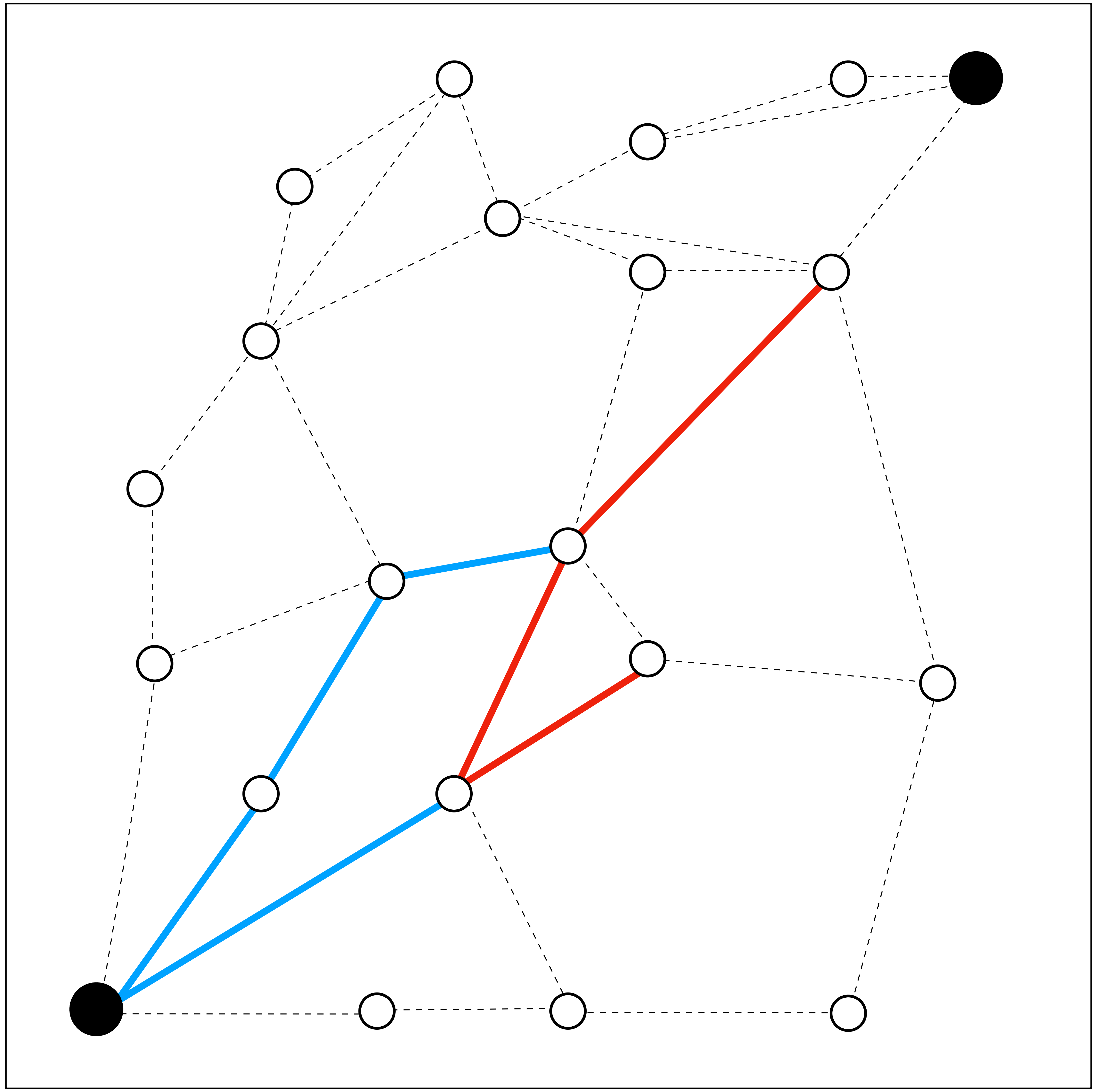
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



An *really simple* algorithm

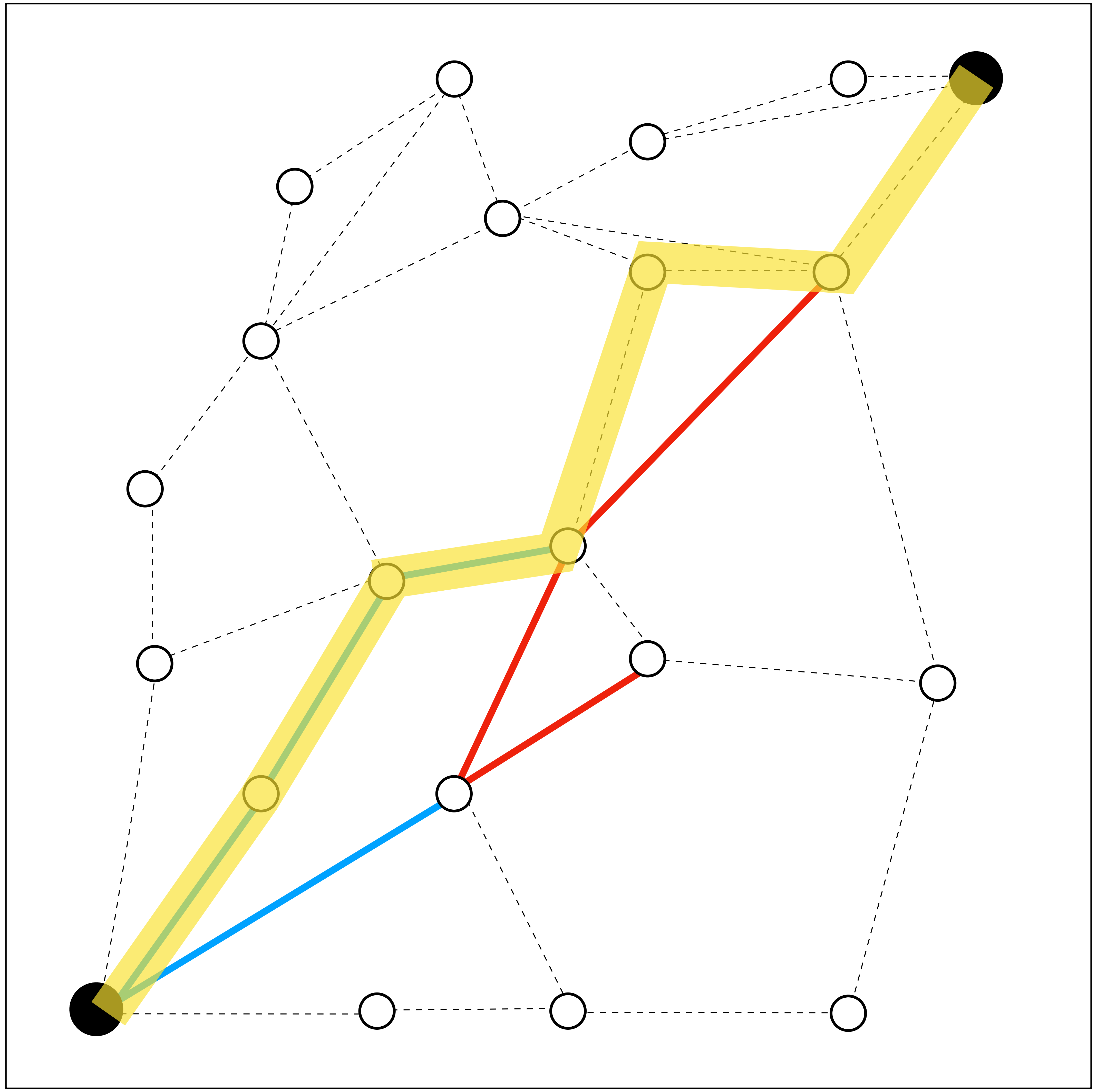
Optimistically initialize all cost(edge) = 0

Repeat till shortest feasible path found:

Find the shortest path

Evaluate shortest path

Update costs



An *really simple* algorithm

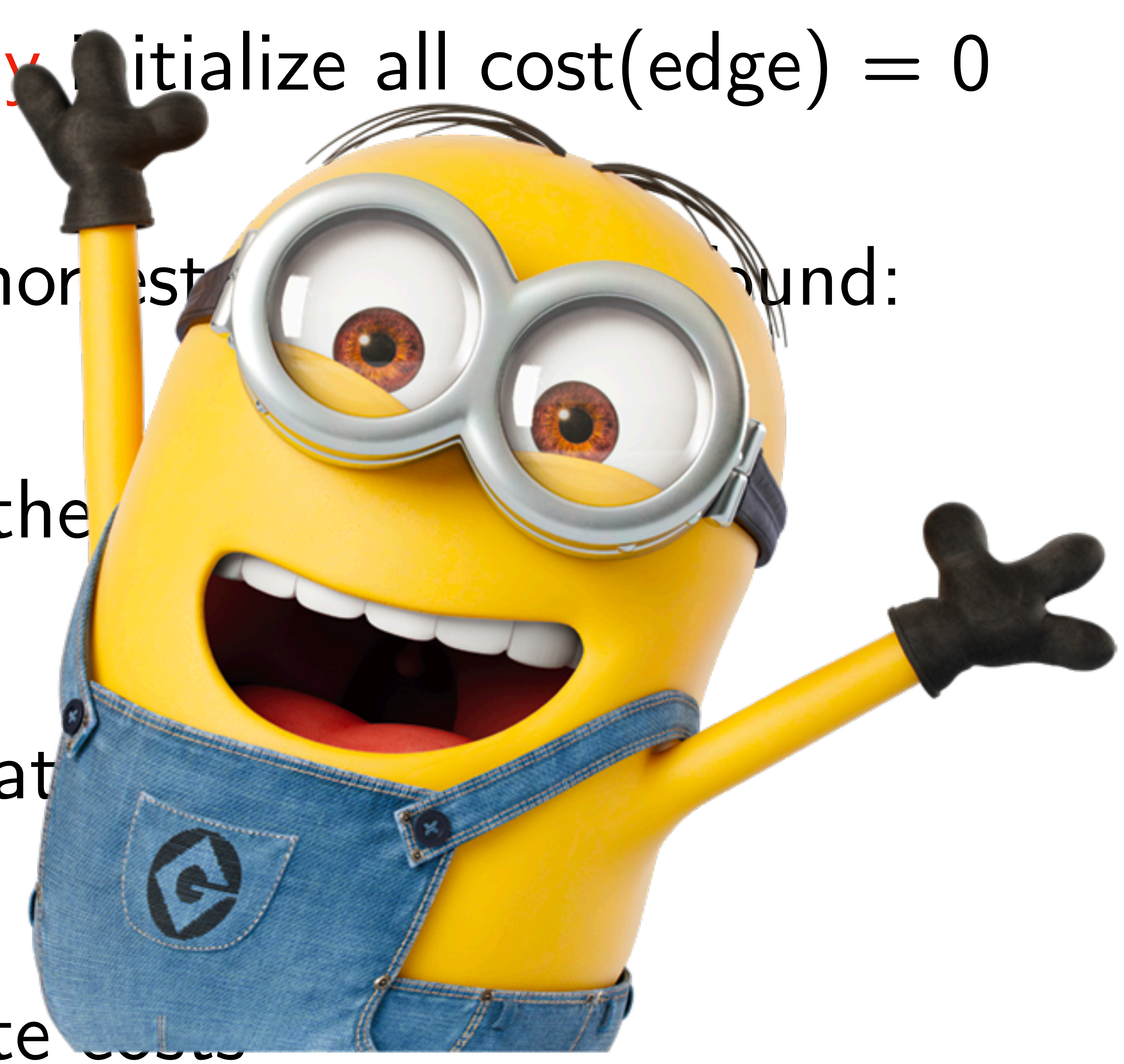
Optimistically itialize all cost(edge) $= 0$

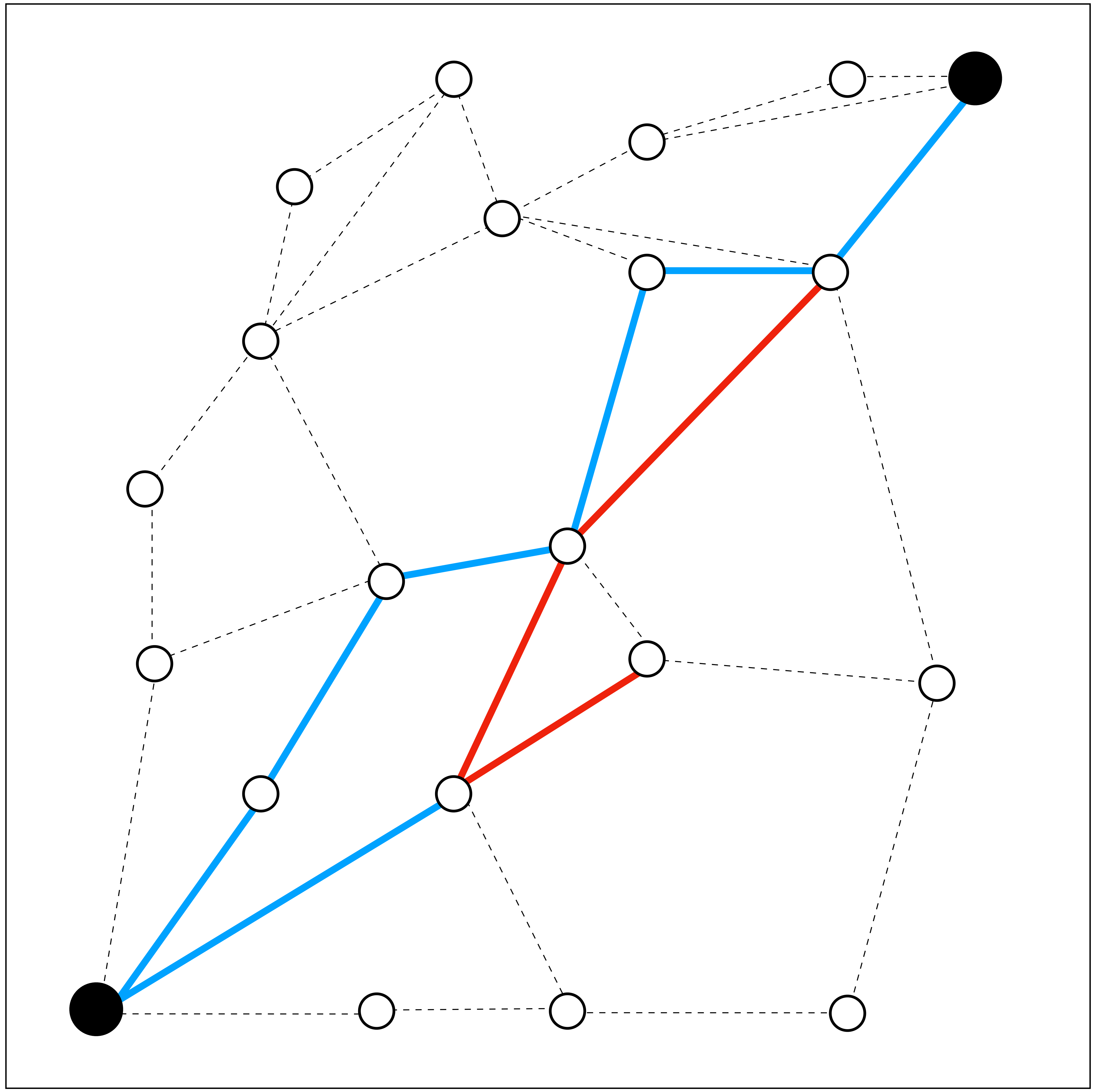
Repeat till shor est ound:

Find the

Evaluat

Update





Many questions ...

Why do we care about minimizing edge queries?

What can we prove about this algorithm?

Principle of Optimism in the Face of Uncertainty (OFU)

One of two things will happen:

1. Either we are correct and done!
2. Or we were wrong and eliminated a candidate option

Optimism in the Face of Uncertainty

Path 1

Path 2

Path 3

Path 4

⋮

Path N

Sort paths by ascending cost

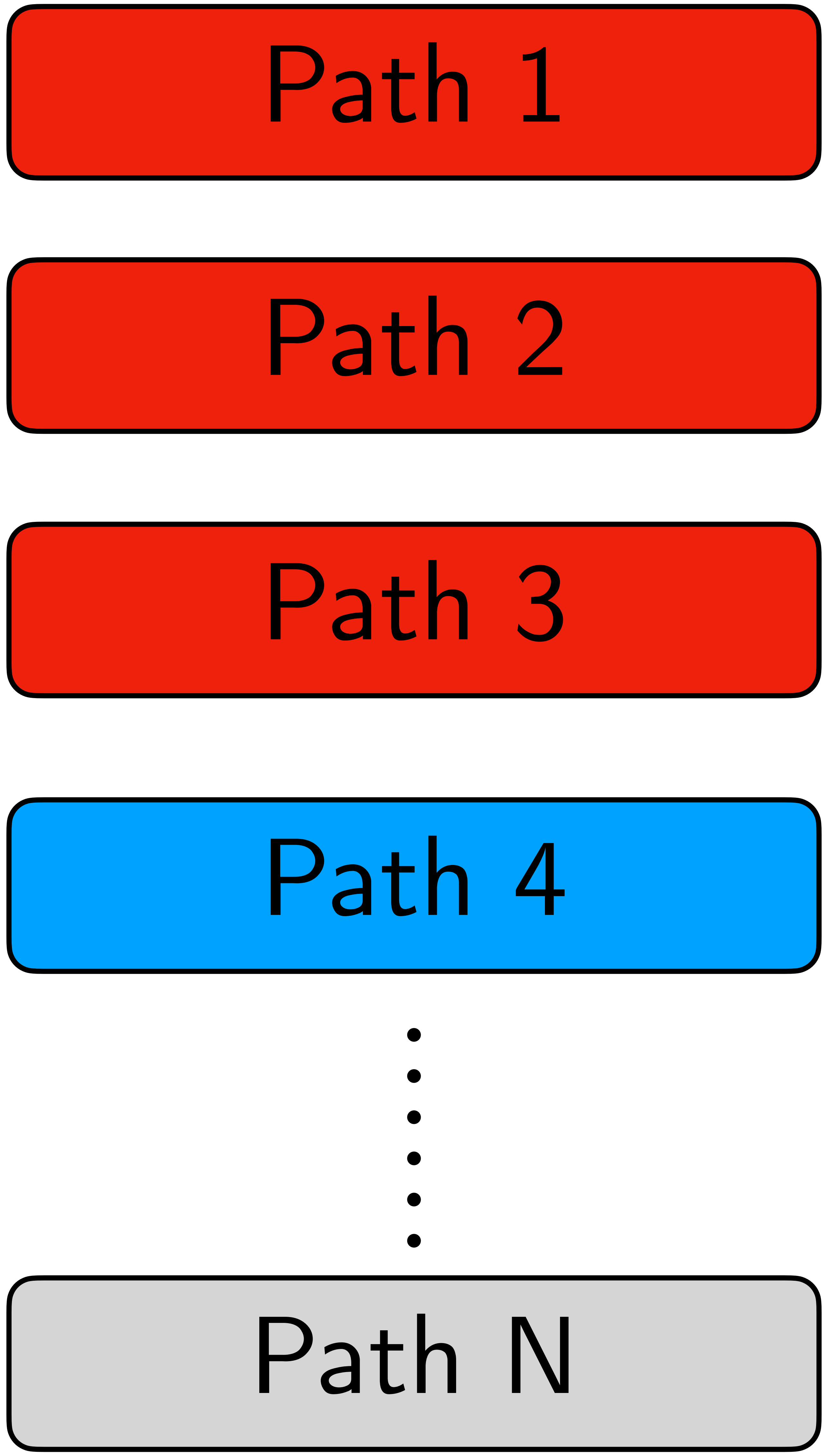
Optimism in the Face of Uncertainty

- Path 1
- Path 2
- Path 3
- Path 4
- ⋮
- Path N

Sort paths by ascending cost

Keep checking each path

Optimism in the Face of Uncertainty



Sort paths by ascending cost

Keep checking each path

At most check K paths till you find the shortest one

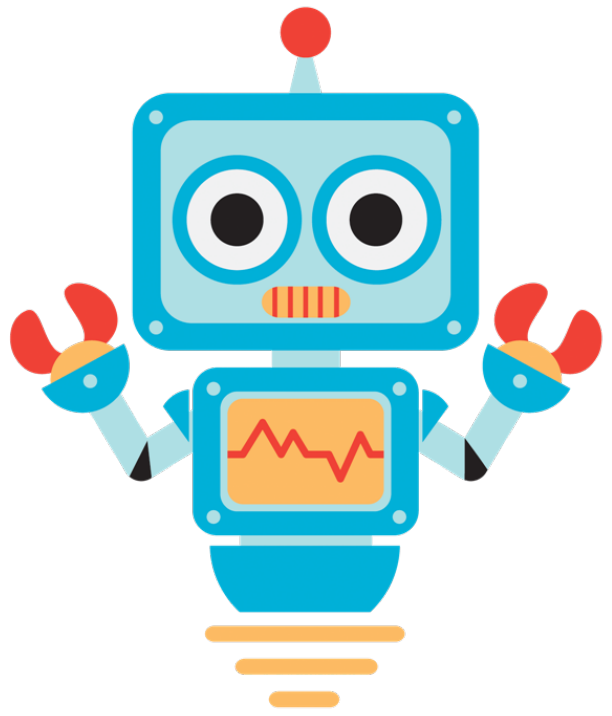
Optimal strategy given no other information

A more general instance: R-MAX

- Let's say we are tasked with exploring an unknown MDP
- Optimistically initialize the MDP
  - Assume all unknown state actions transition to "heaven" and get maximum reward indefinitely R_{max}
- Repeat forever
  - Solve for the optimal policy given current model. Execute policy
  - If you visit a state K number of times, update model to use empirical transition and reward function
- Can prove that you act optimally in all but a fixed set of N steps (PAC-MDP guarantee)

What if each evaluation is stochastic?

Doors



a^1 ?

a^2 ?

a^3 ?

⋮

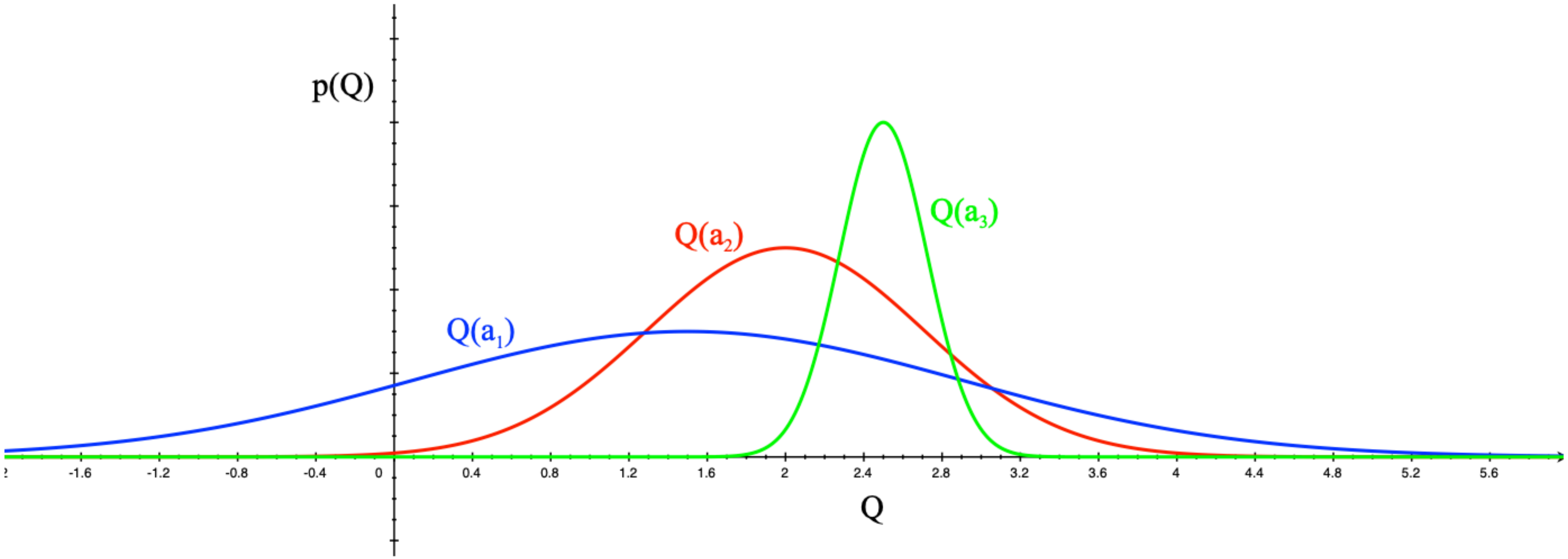
+100

+1

-1000


Doors
Round 1
Round 2
Round 3
a^1
a^2
a^3
+100
+1
-1000

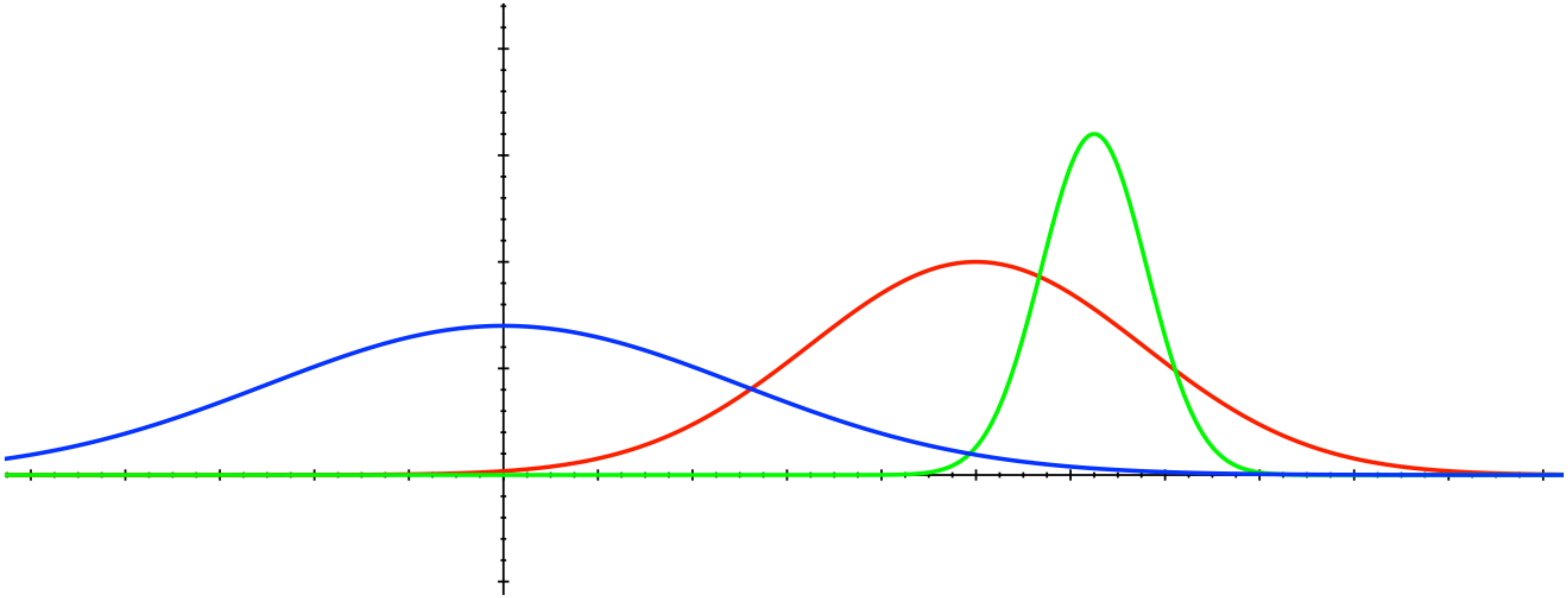
Optimism in the Face of Uncertainty



- Which action should we pick?
- The more uncertain we are about an action-value
- The more important it is to explore that action
- It could turn out to be the best action

Credit: David Silver

Optimism in the Face of Uncertainty



- After picking blue action
- We are less uncertain about the value
- And more likely to pick another action
- Until we home in on best action

Upper Confidence Bound

- Estimate an upper confidence $\hat{U}_t(a)$ for each action value
- Such that $Q(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$ with high probability
- This depends on the number of times $N(a)$ has been selected
  - Small $N_t(a) \Rightarrow$ large $\hat{U}_t(a)$ (estimated value is uncertain)
  - Large $N_t(a) \Rightarrow$ small $\hat{U}_t(a)$ (estimated value is accurate)
- Select action maximising Upper Confidence Bound (UCB)

$$a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, \hat{Q}_t(a) + \hat{U}_t(a)$$

Credit: David Silver

Upper Confidence Bound

Theorem (Hoeffding's Inequality)

Let $X_1, ..., X_t$ be i.i.d. random variables in [0,1], and let $\overline{X}_t = \frac{1}{\tau}\sum_{\tau=1}^{t} X_\tau$ be the sample mean. Then

$$\mathbb{P}\left[\mathbb{E}\left[X\right] > \overline{X}_t + u\right] \leq e^{-2tu^2}$$

- We will apply Hoeffding's Inequality to rewards of the bandit
- conditioned on selecting action a

$$\mathbb{P}\left[Q(a) > \hat{Q}_t(a) + U_t(a)\right] \leq e^{-2N_t(a)U_t(a)^2}$$

Credit: David Silver

Upper Confidence Bound

- Pick a probability p that true value exceeds UCB
- Now solve for $U_t(a)$

$$e^{-2N_t(a)U_t(a)^2} = p$$

$$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

- Reduce p as we observe more rewards, e.g. $p = t^{-4}$
- Ensures we select optimal action as $t \to \infty$

$$U_t(a) = \sqrt{\frac{2\log t}{N_t(a)}}$$

Credit: David Silver

Upper Confidence Bound

Value of action

$$a_t = \operatorname*{argmax}_{a \in \mathcal{A}} Q(a) + \boxed{\sqrt{\frac{2 \log t}{N_t(a)}}}$$

How many times did you try action?

Exploration Bonus

Can prove that it is no-regret ($\lim_{t \to \infty} \frac{\log t}{t} = 0$)

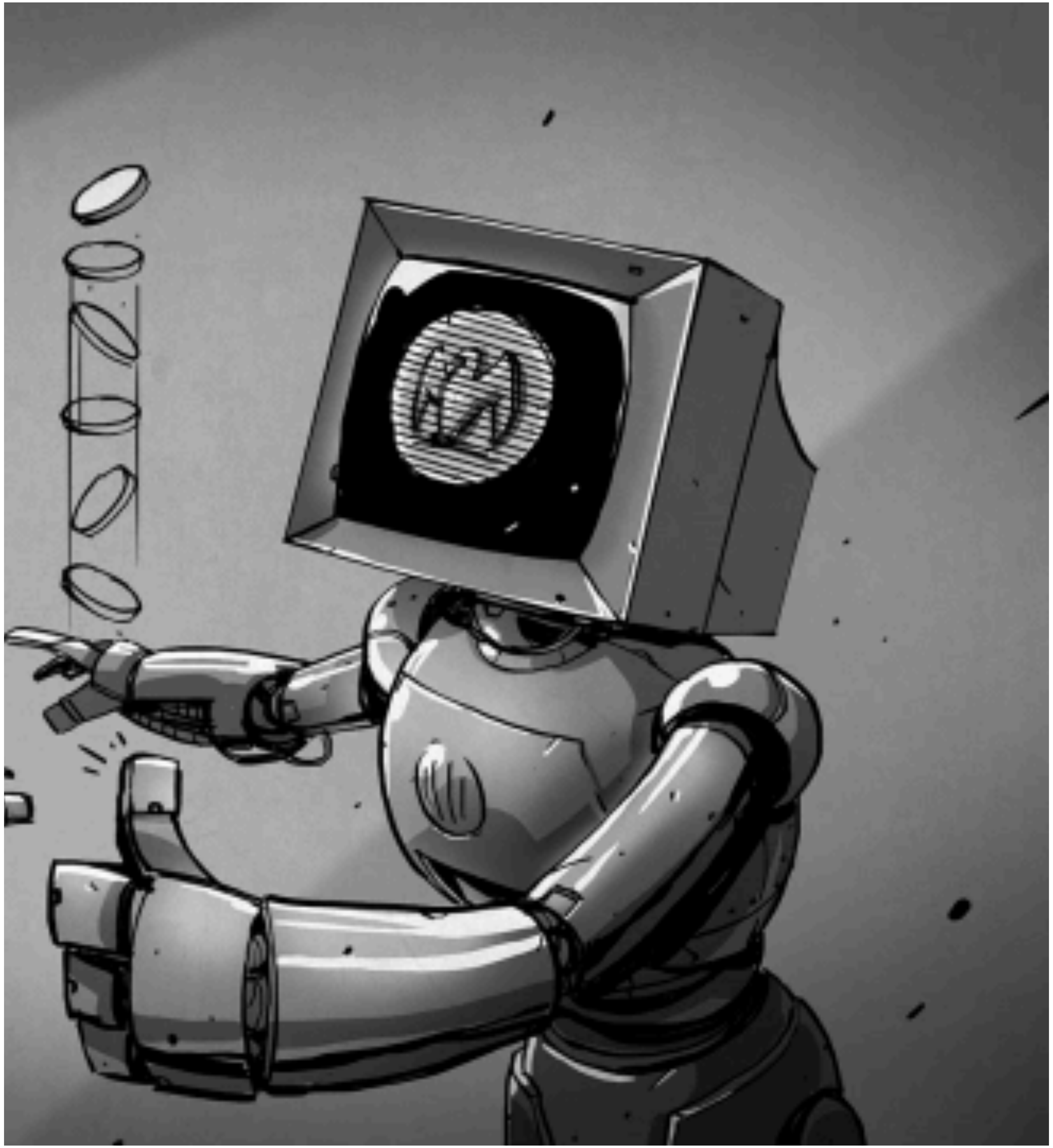
Credit: David Silver

How can we apply this to RL?

Add an exploration bonus to the reward function!

$$r^+(s,a) = r(s,a) + \sqrt{\frac{2\log n}{N(s,a)}}$$

What if we have a really good prior knowledge?



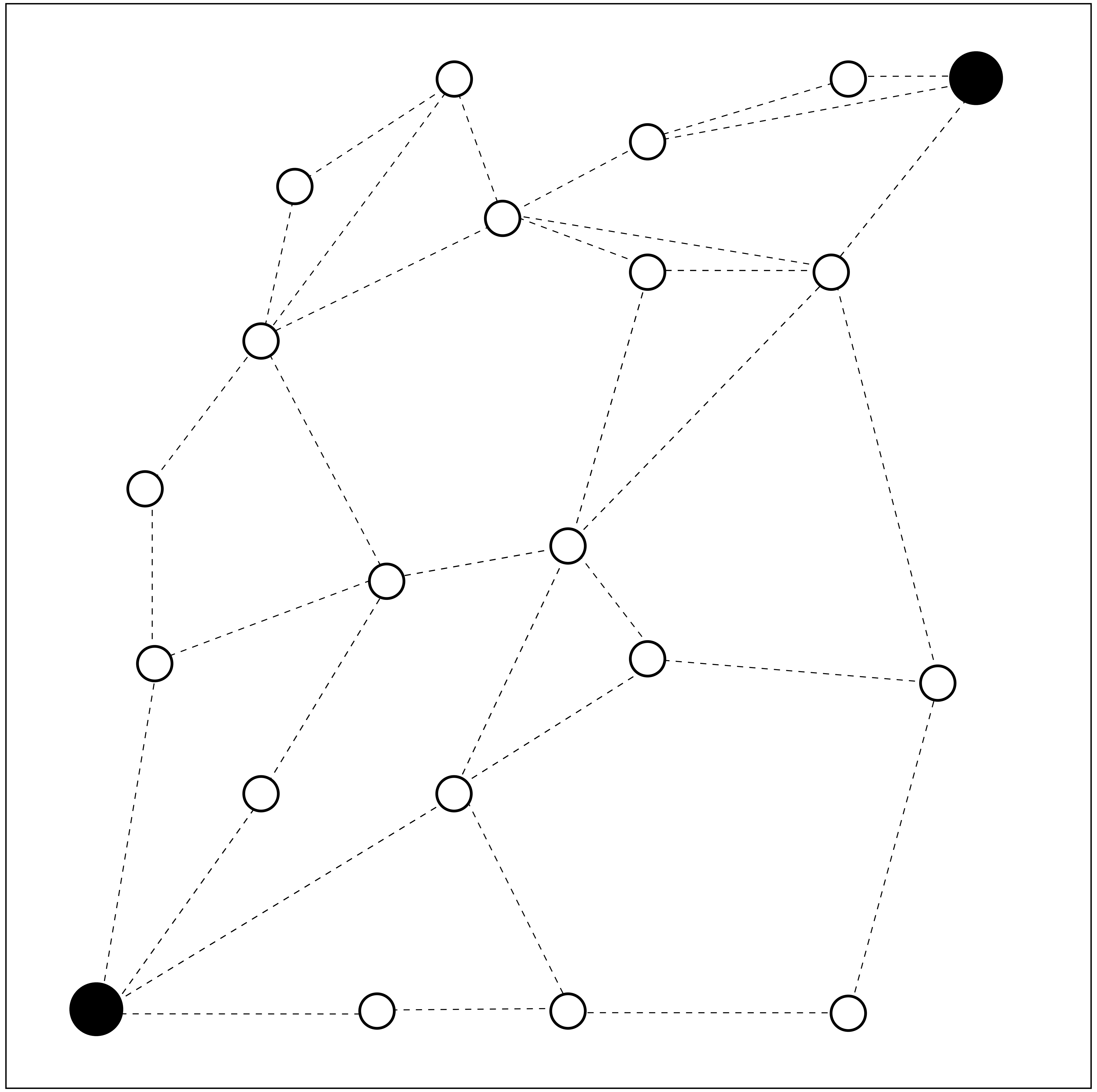
Posterior Sampling

The Online Shortest Path Problem

You just moved to Cornell and are traveling from office to home.

You would like to get home quickly but you are uncertain about travel times along each edge

Suppose we had a prior on travel time for each edge (Mean θ_e, Var σ_e)

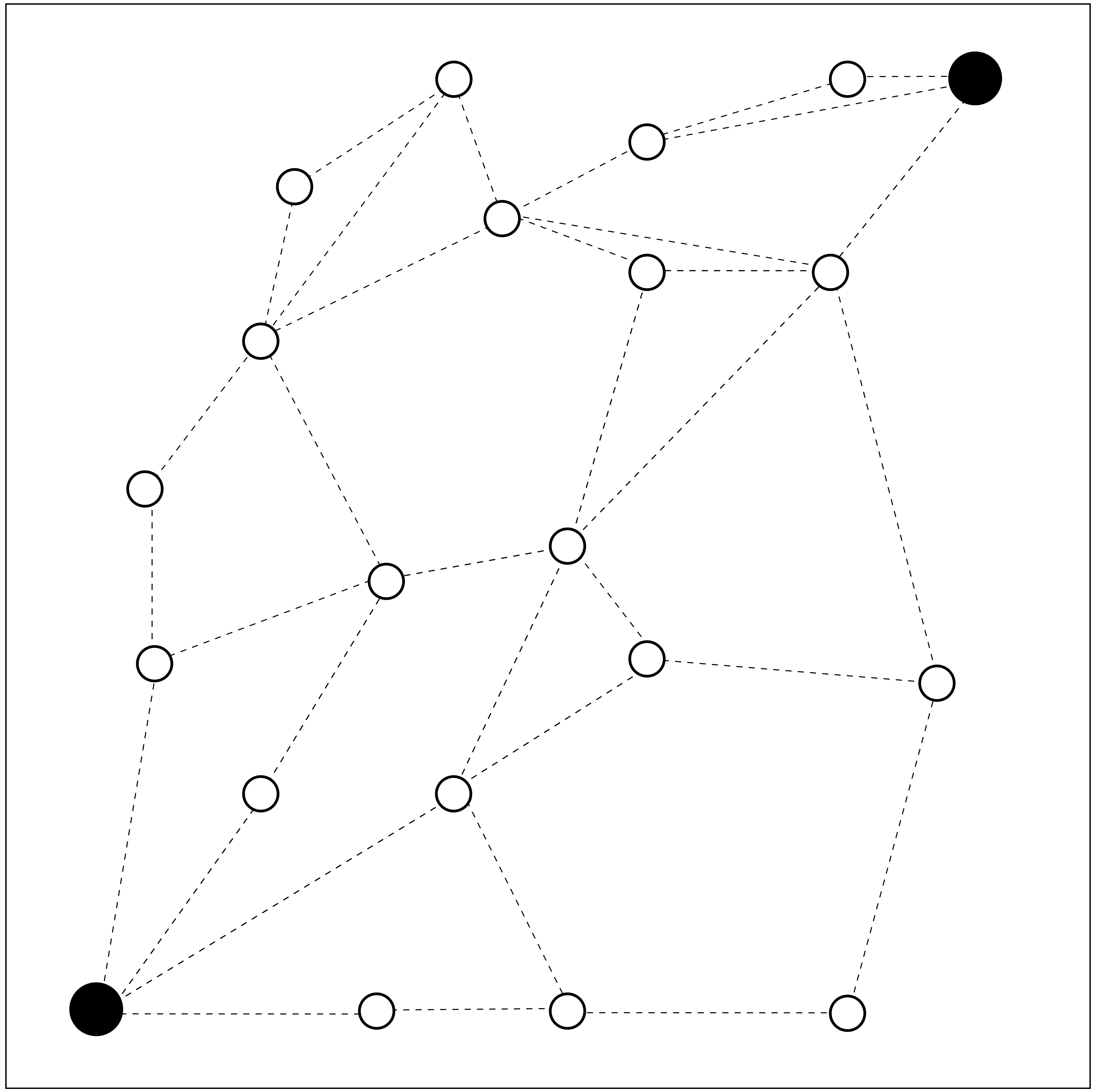


Can we apply UCB?

You just moved to Cornell and are traveling from office to home.

You would like to get home quickly but you are uncertain about travel times along each edge

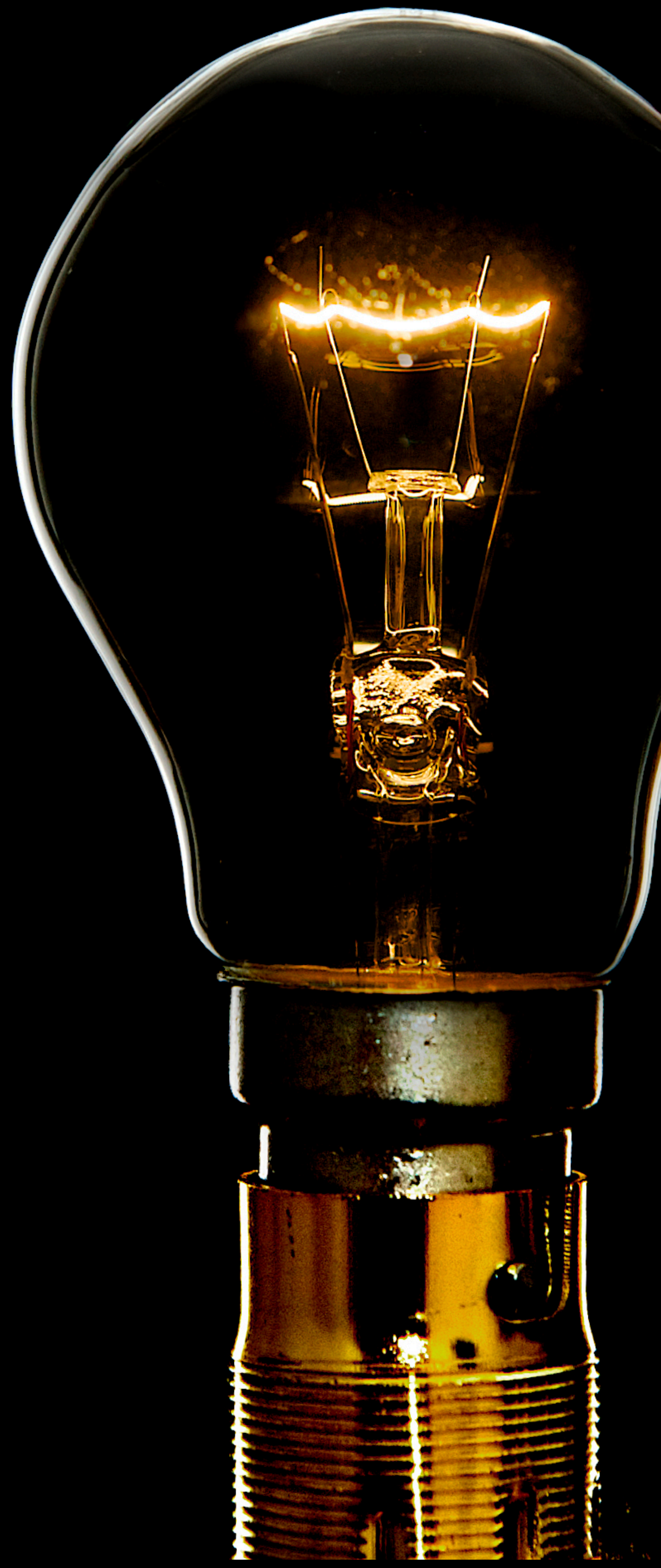
Suppose we had a prior on travel time for each edge (Mean θ_e, Var σ_e)



UCB is a nightmare!

Hard to compute upper confidence bounds for arbitrary distributions

Have to “tune” exploration bonus, too much and we will over explore



What if ...

... we just sampled travel times from our prior and solved the shortest path?

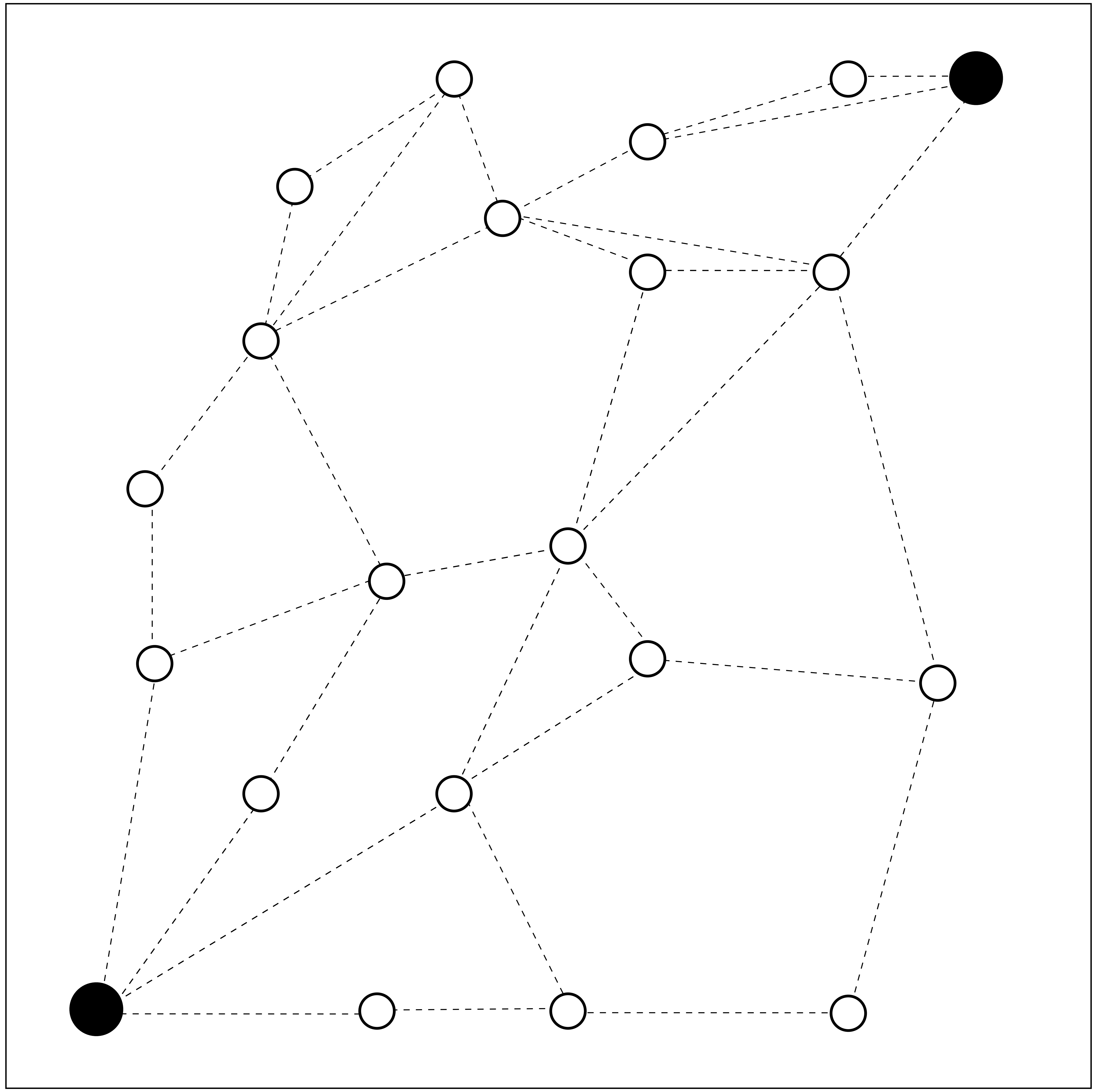
A suspiciously simple algorithm

Repeat forever:

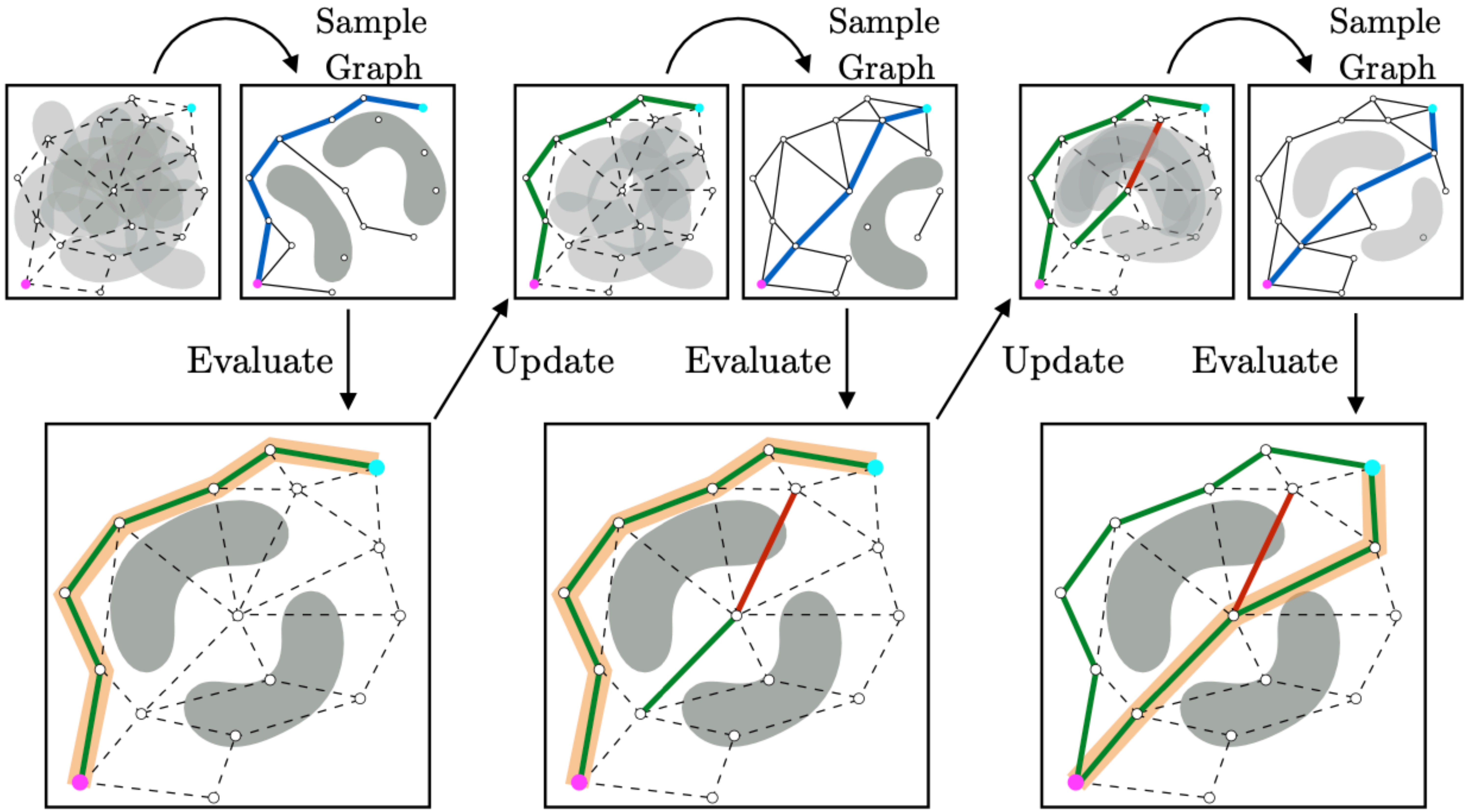
Sample edge times from posterior

Compute shortest path

Travel along path, and update posterior



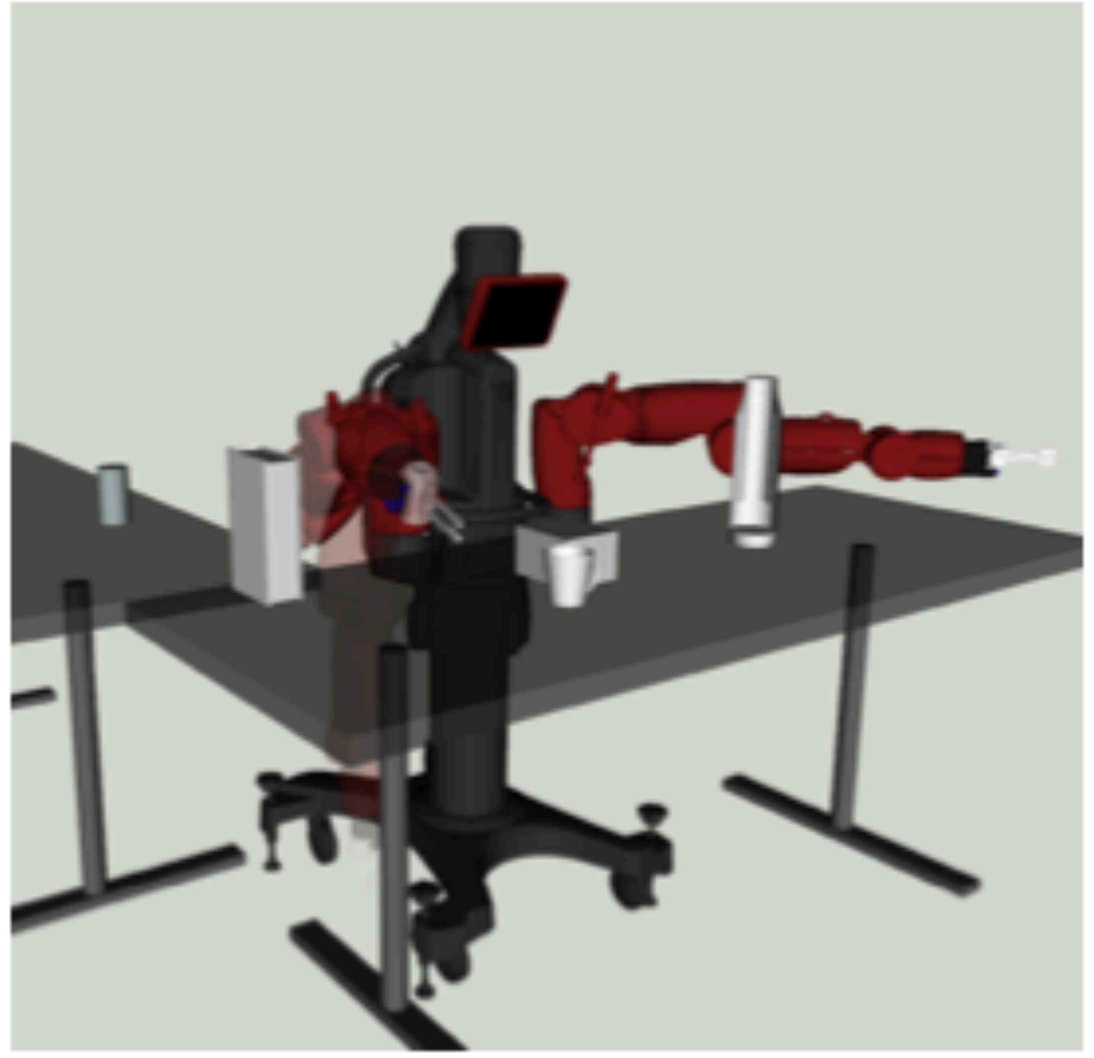
Posterior Sampling for Motion Planning

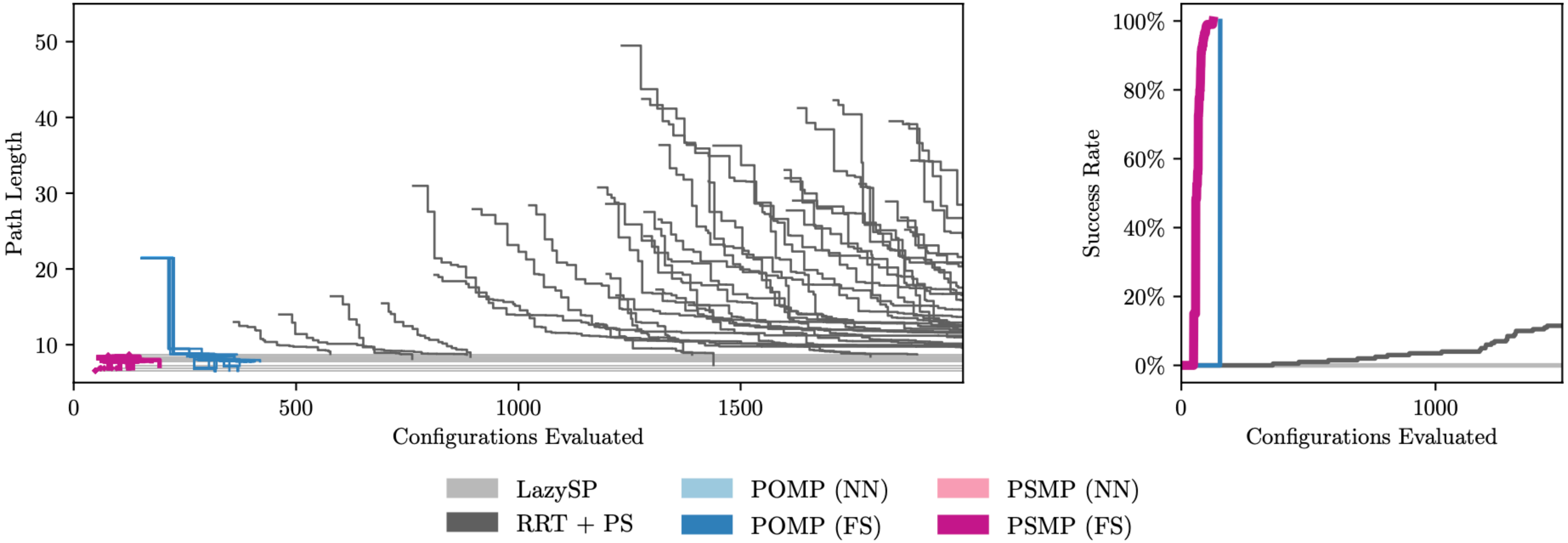


Posterior Sampling for Anytime Motion Planning on Graphs with Expensive-to-Evaluate Edges

Brian Hou, Sanjiban Choudhury, Gilwoo Lee, Aditya Mandalika, and Siddhartha S. Srinivasa

Posterior Sampling for Motion Planning





Posterior Sampling for Anytime Motion Planning on Graphs with Expensive-to-Evaluate Edges

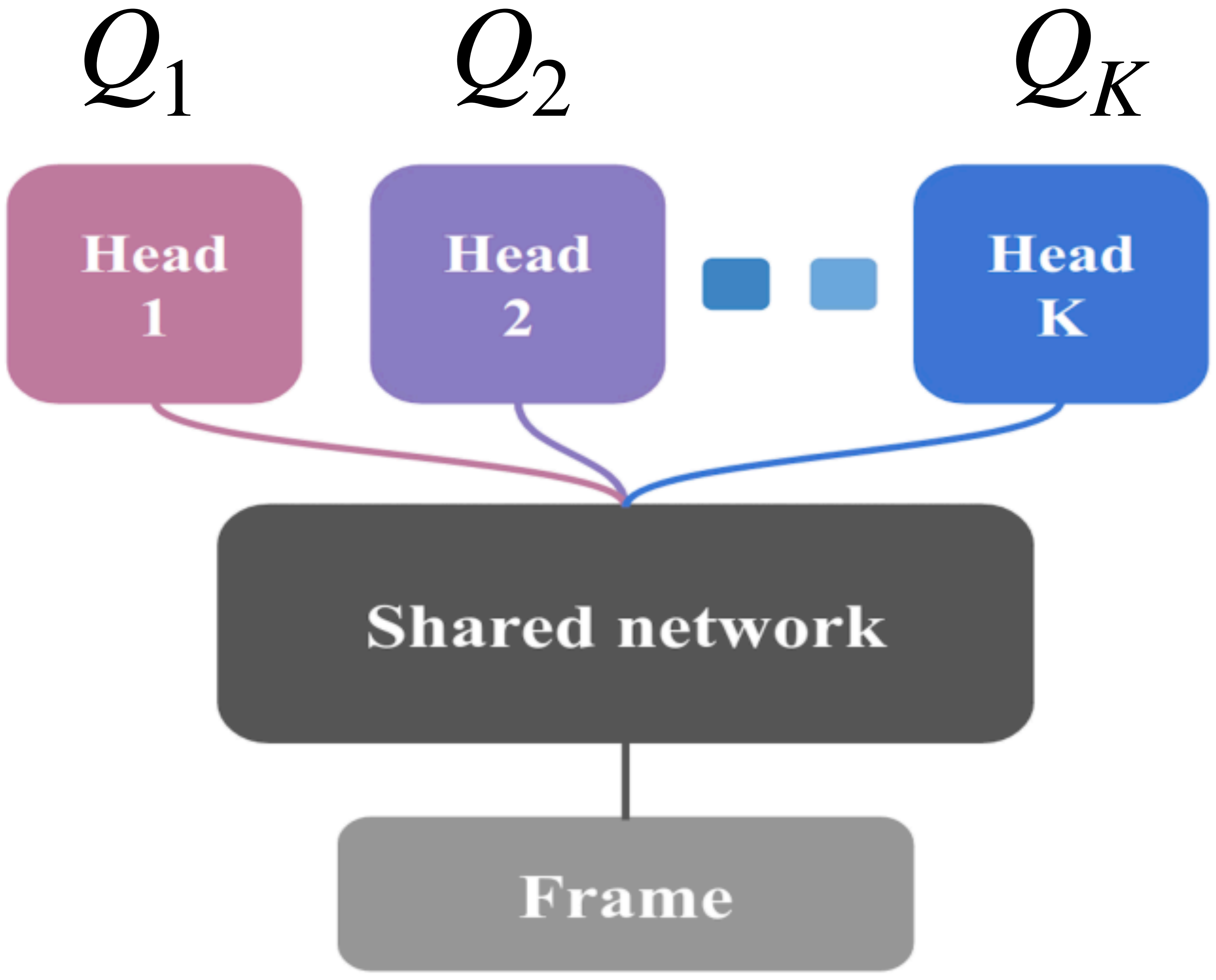
Brian Hou, Sanjiban Choudhury, Gilwoo Lee, Aditya Mandalika, and Siddhartha S. Srinivasa

Posterior Sampling for Reinforcement Learning

1. sample Q-function Q from $p(Q)$
2. act according to Q for one episode
3. update $p(Q)$

Deep Exploration via Bootstrapped DQN

Ian Osband[1,2], Charles Blundell[2], Alexander Pritzel[2], Benjamin Van Roy[1]
[1]Stanford University, [2]Google DeepMind
{iosband, cblundell, apritzel}@google.com, bvr@stanford.edu



Bootstrapped Q Network

Posterior Sampling for Reinforcement Learning

1. sample Q-function Q from $p(Q)$
2. act according to Q for one episode
3. update $p(Q)$

Atari

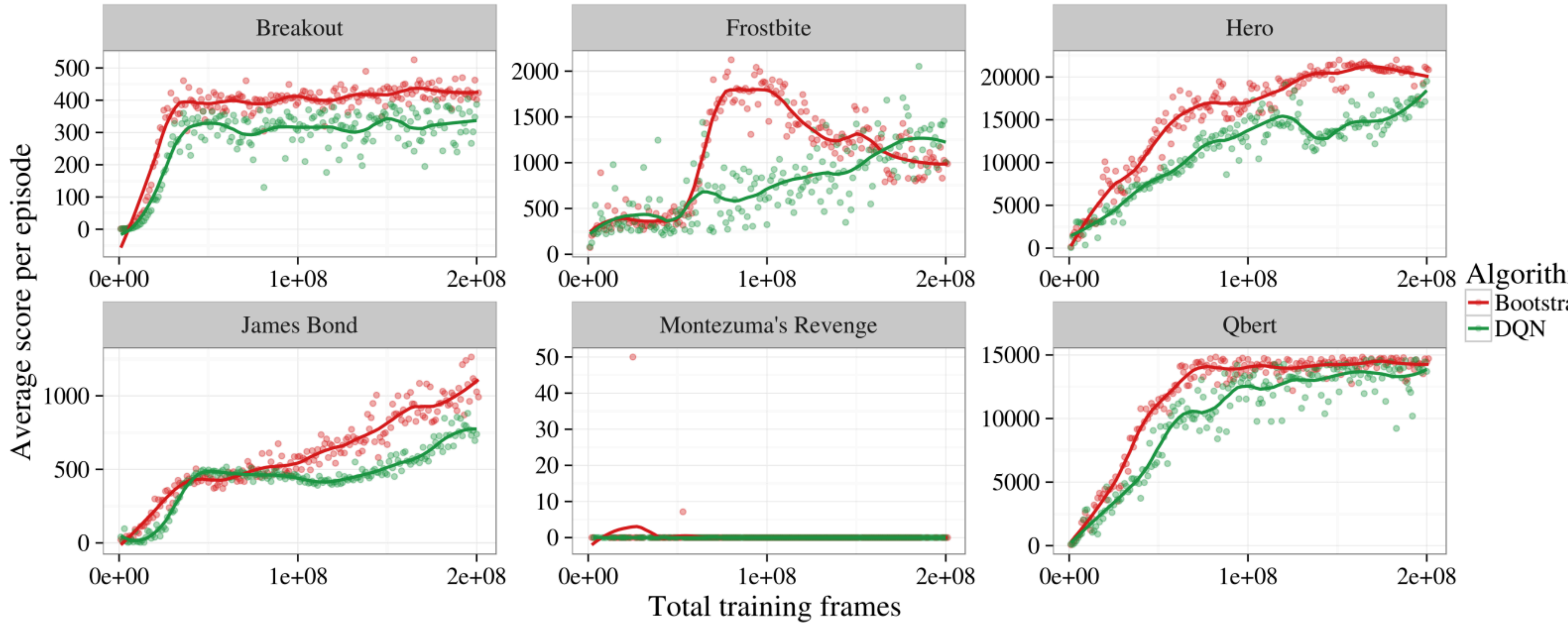
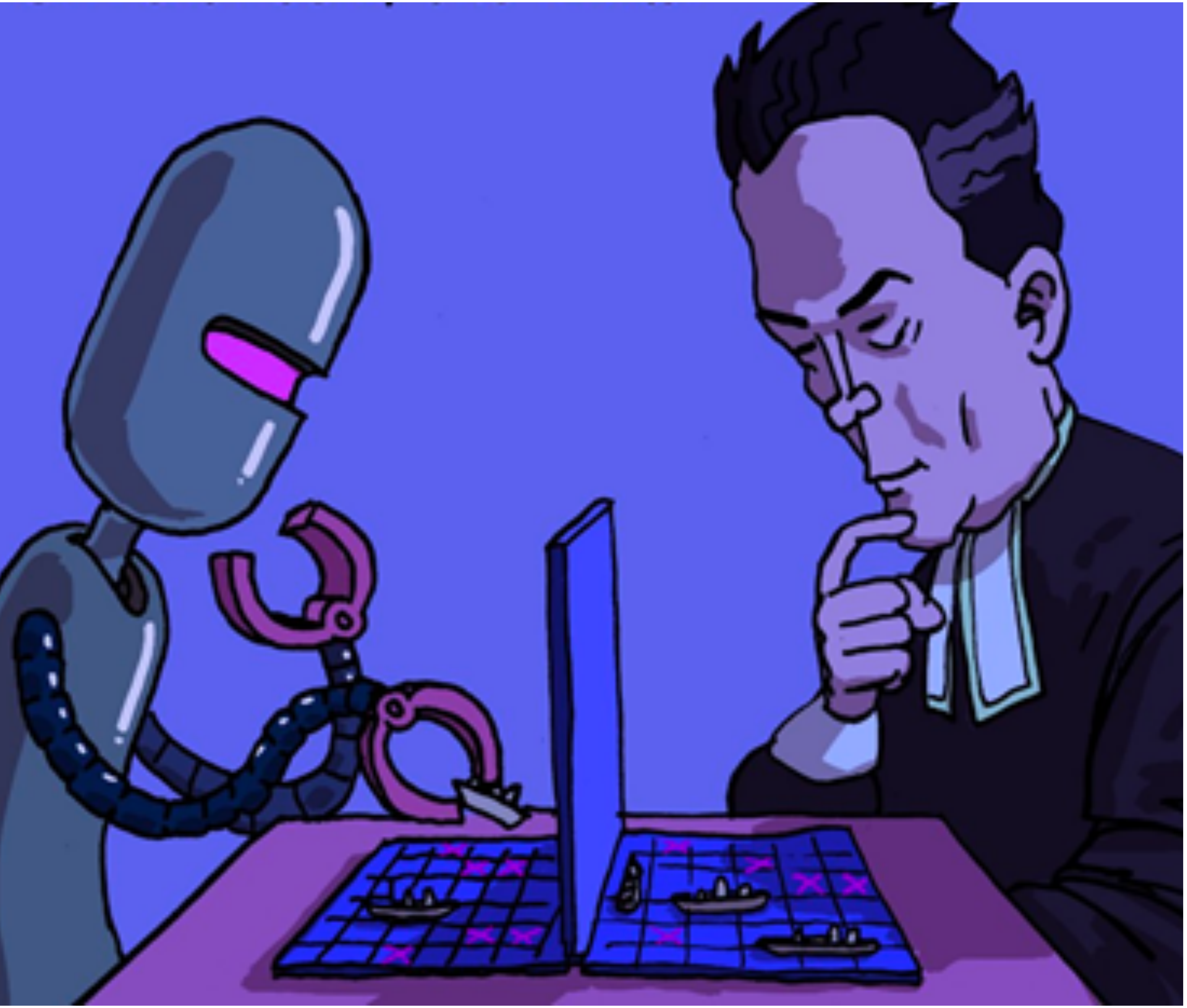


Figure 6: Bootstrapped DQN drives more efficient exploration.

Why does work better than taking random actions?

What if we wanted to explore as optimally as possible using prior information?



Information Gain

20 Questions

Let's say you have a set of hypotheses

$$\{\theta_1, \theta_2, \ldots, \theta_n\}$$

and a set of tests

$$\{t_1, t_2, \ldots, t_n\}$$

Given a prior over hypotheses $P(\theta)$

Find the minimal number of tests to identify hypothesis

20 Questions

Let's say you have a set of hypotheses

$$\{\theta_1, \theta_2, \ldots, \theta_n\}$$

and a set of tests

$$\mathcal{T} = \{1, \ldots, N\}$$

NP-HARD

Given a prior over hypotheses $P(\theta)$

Find the minimal number of tests to identify hypothesis

A simple algorithm

Greedily pick the test that maximizes information gain

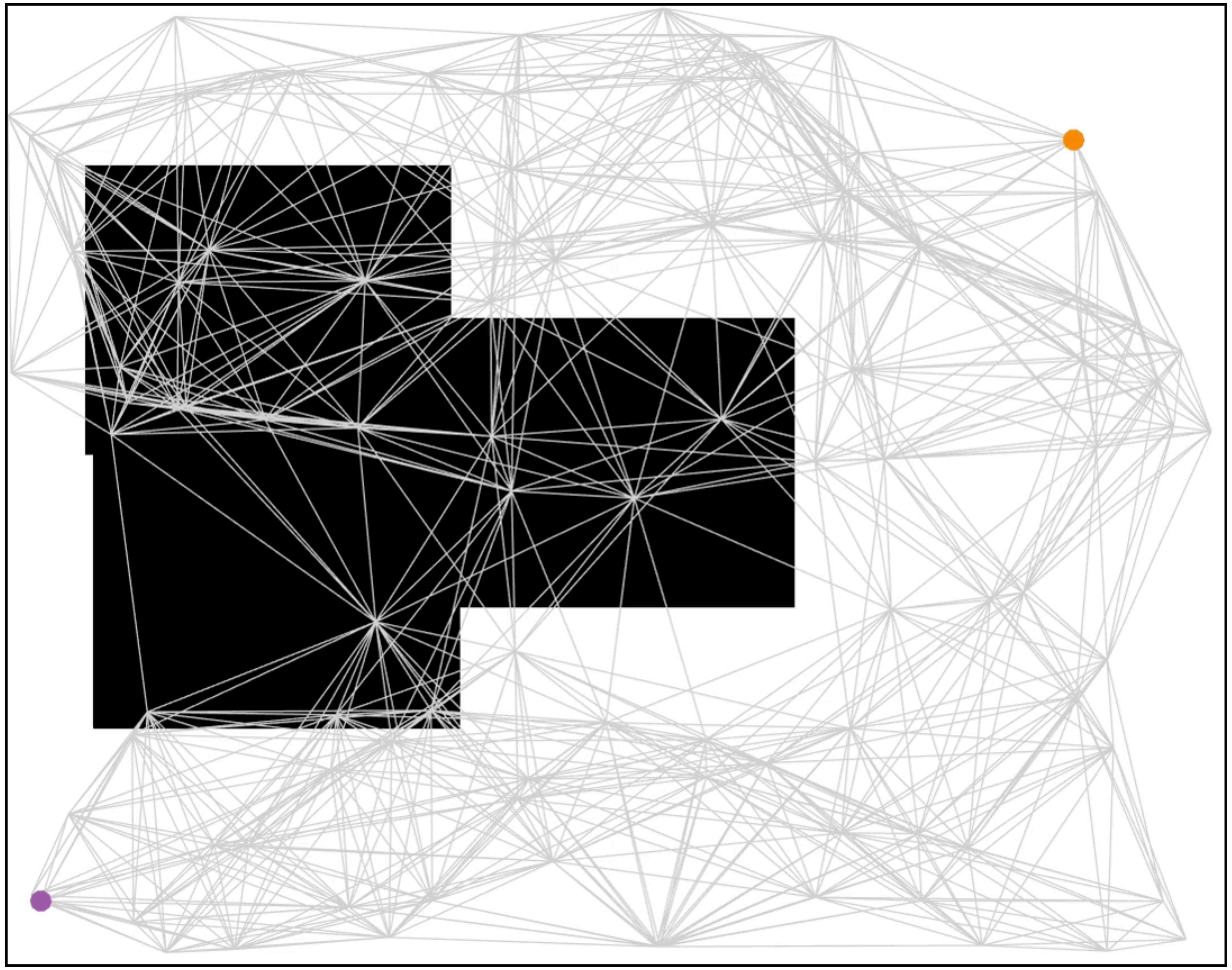
$$\max_t H(\theta) - \mathbb{E}_o H(\theta \mid t, o)$$

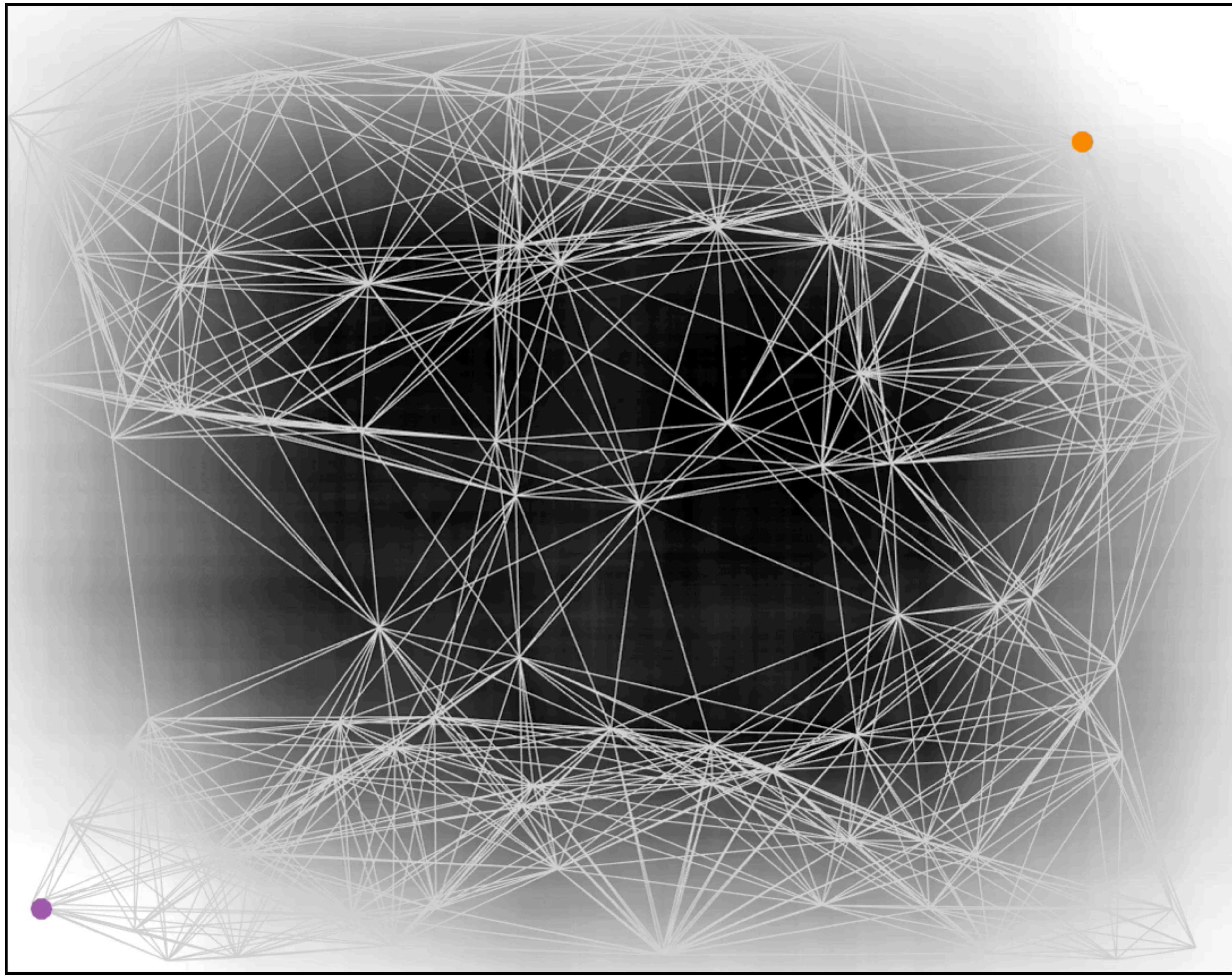
Entropy   Posterior entropy

This is near-optimal!

Optimal edge evaluation for shortest path

[CJS+ NeurIPS'17] [CSS IJCAI'18]



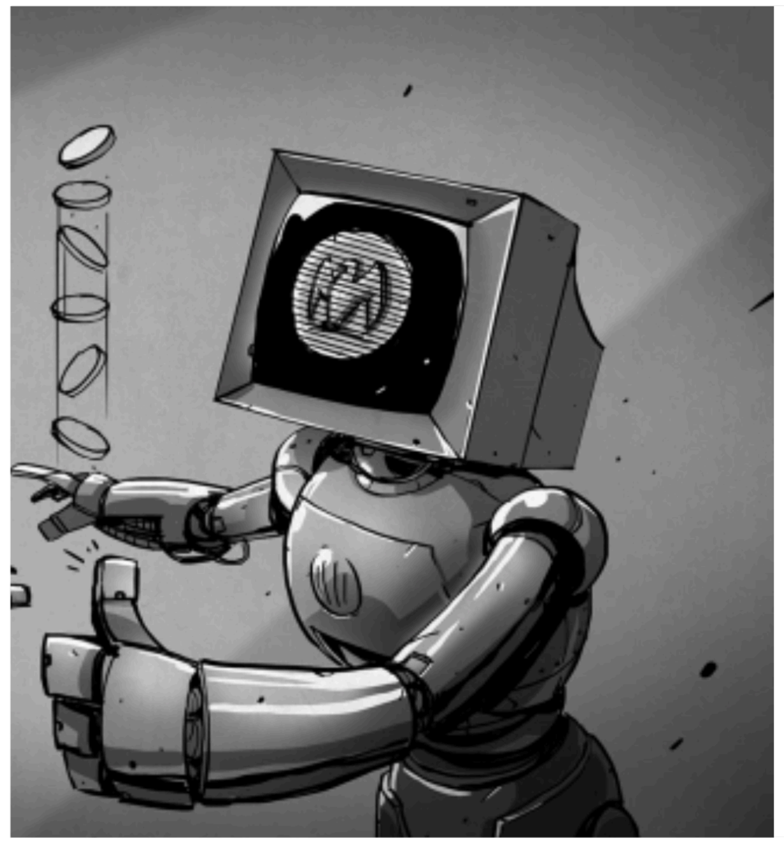


tl;dr

Belief Space Planning is NP-Hard at best, undecidable at worst

Need to relax our problem!

Optimism in the Face of Uncertainty (OFU)



Posterior Sampling

Information Gain