
lyft
UBER
lyft
UBER
lyft

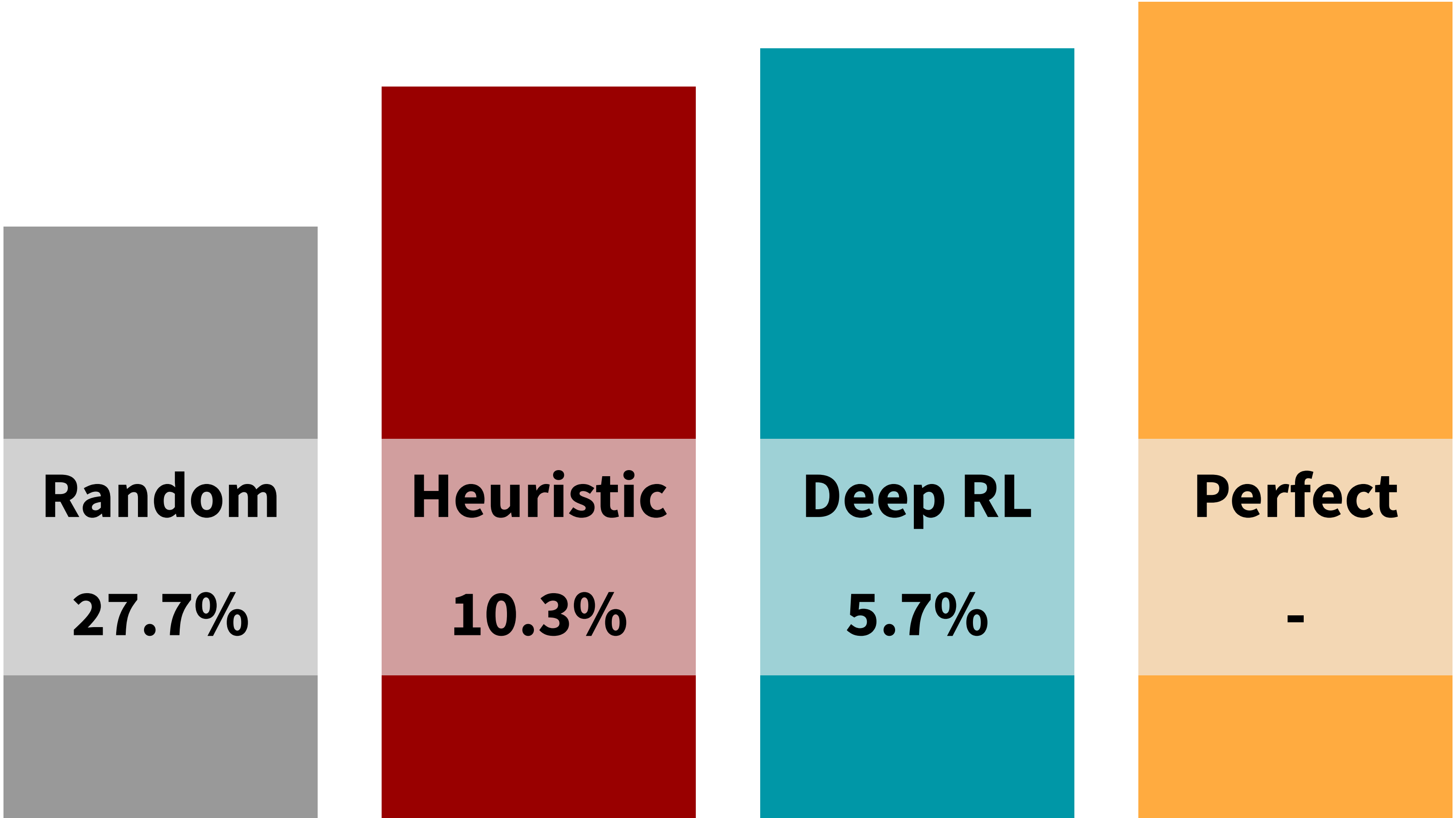
Autonomous Vehicles

Controlling Autonomous Electric Fleets

Deep RL + OR
Policies + Bound

Controlling Autonomous Electric Fleets

Some success so far



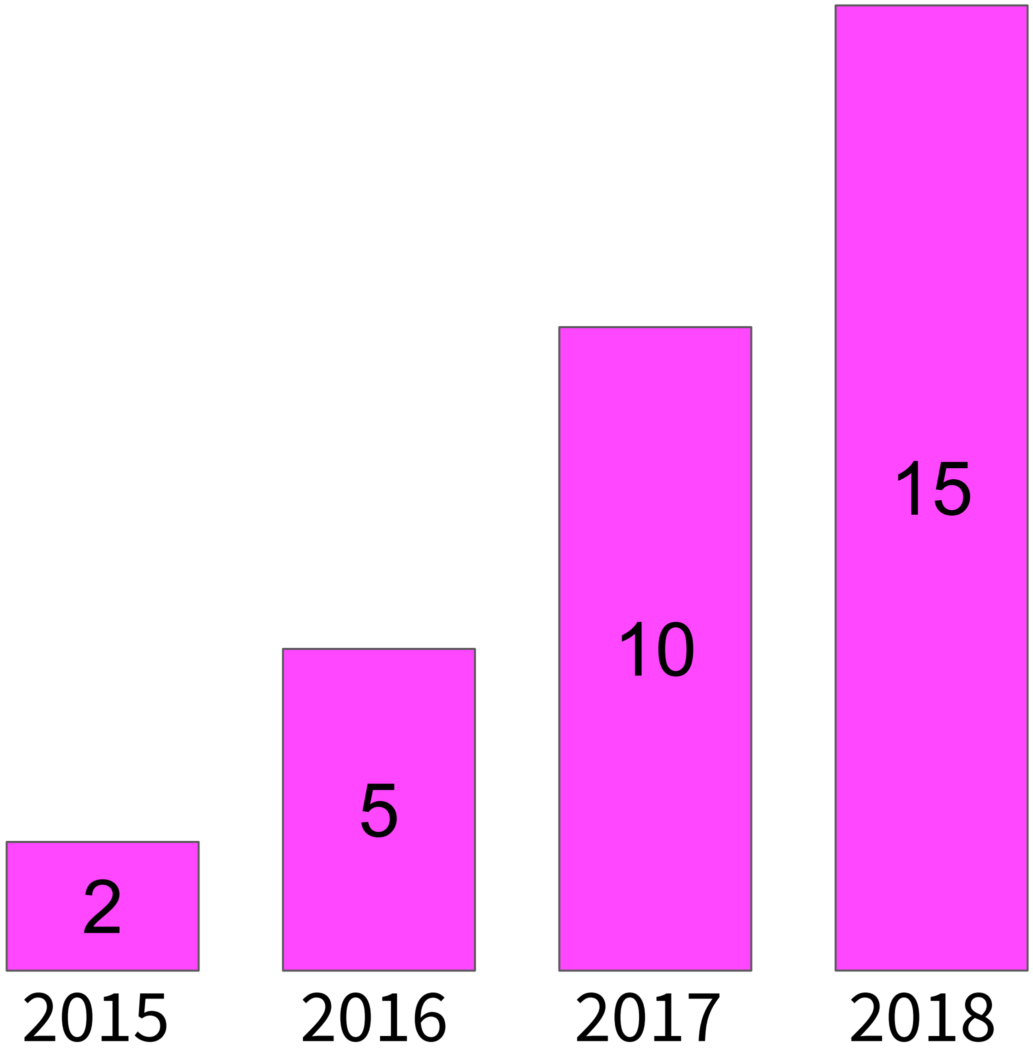
Control of Autonomous Electric Fleets for Ridehail Systems

ND Kullman, JE Mendoza, JC Goodson, M Cousineau

Background

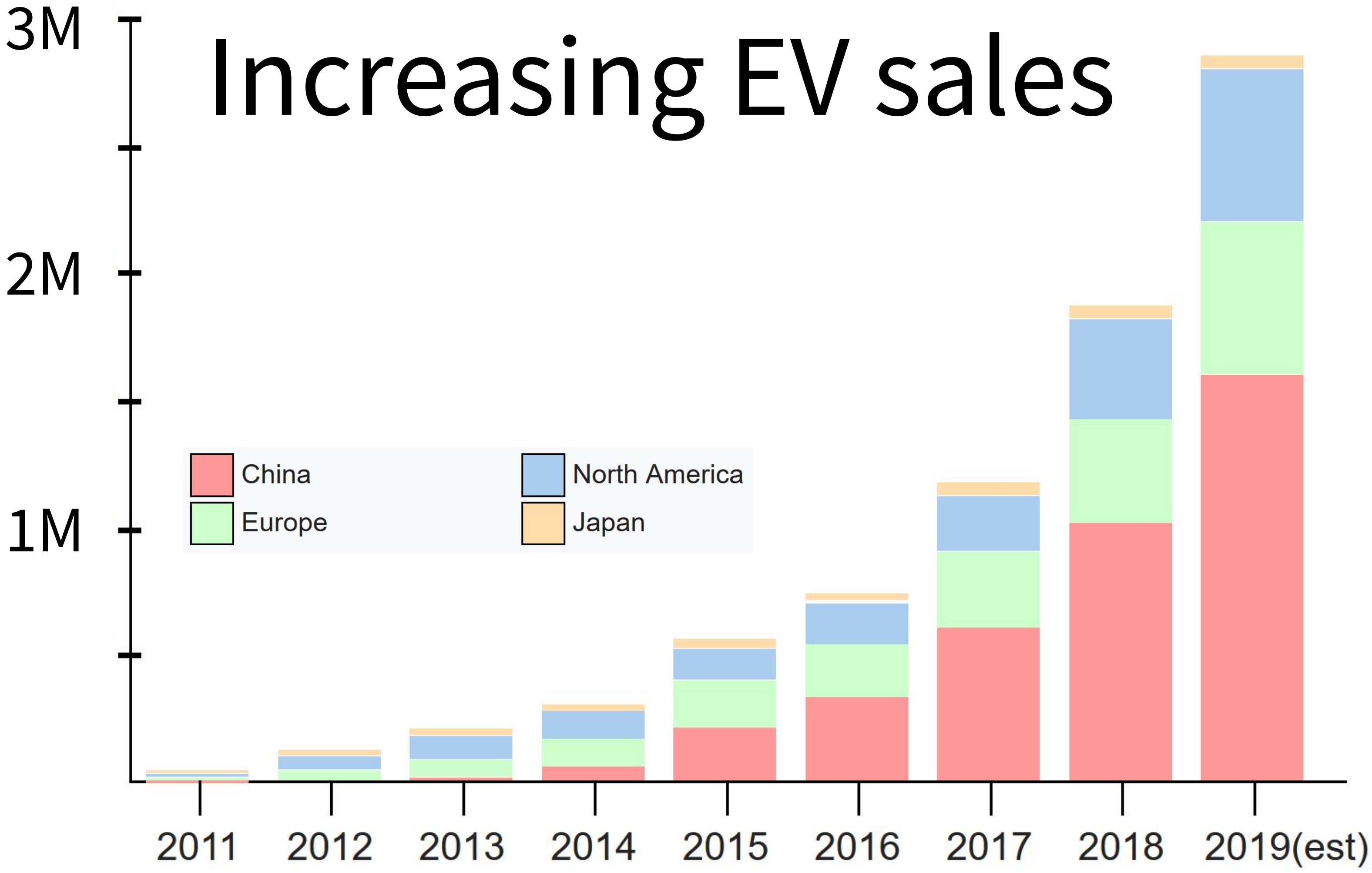
Ridehail explosion

NYC ridehail trips
(M, avg/mo)




400 ppm
YOU ARE HERE
CO_2
800 kYA
Today

Increasing EV sales



Autonomy imminent

Autonomy imminent

"From our standpoint... we think people [will] not need to touch the wheel... sometime probably around, I don't know, second quarter next year…"

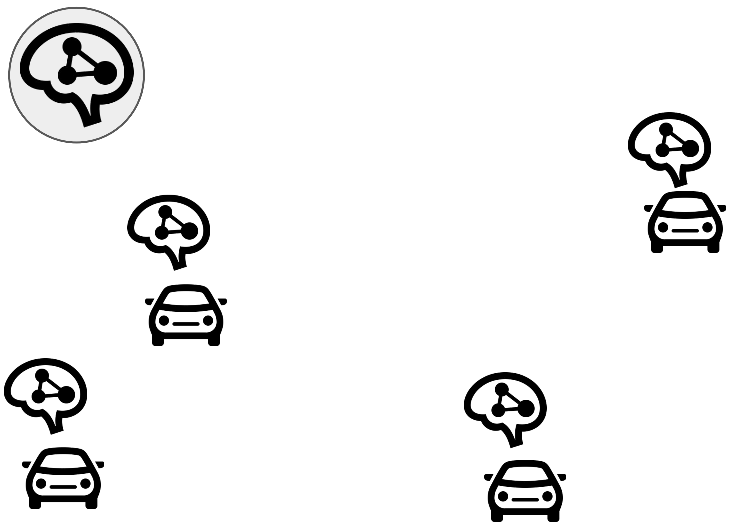
With **regulatory approval** in some places starting around **end of 2020**

Elon Musk, CEO Tesla

Autonomy imminent

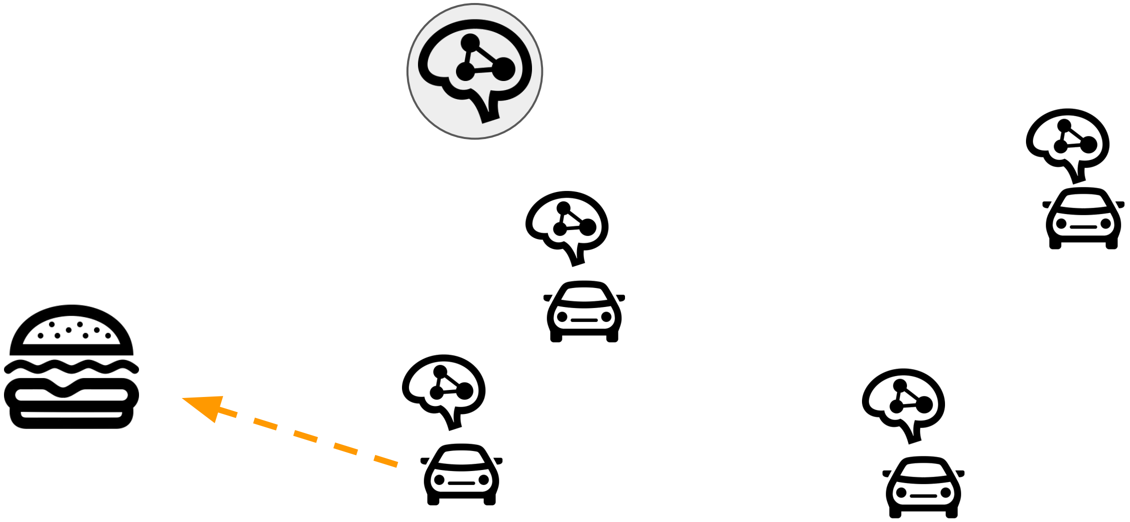
Control of AEV Fleets is Centralized

Today



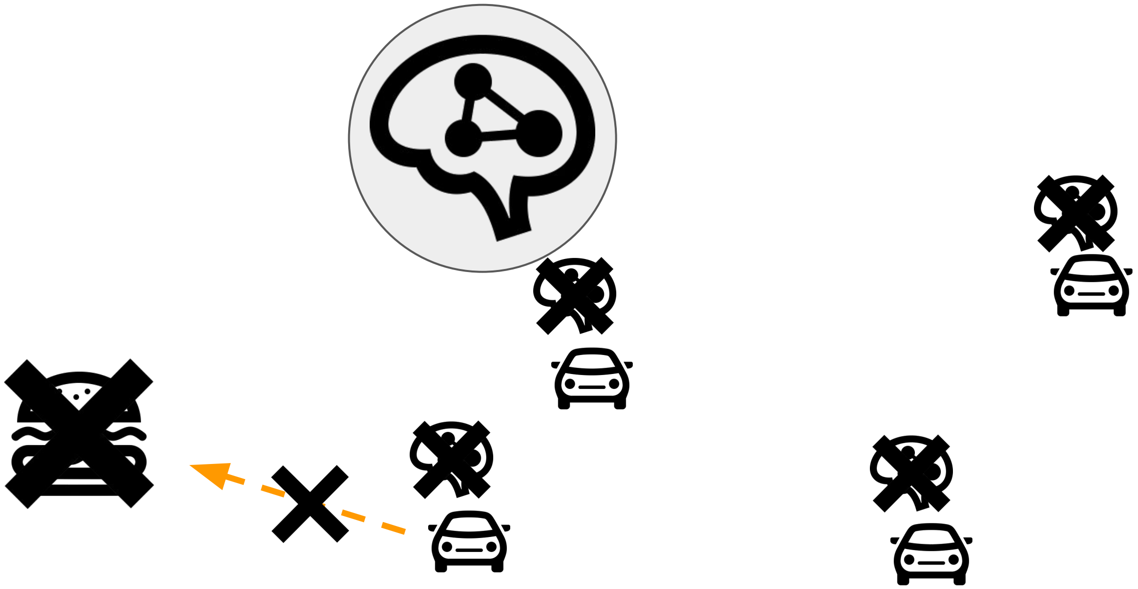
Control of AEV Fleets is Centralized

Today



Control of AEV Fleets is Centralized

w/ AEVs



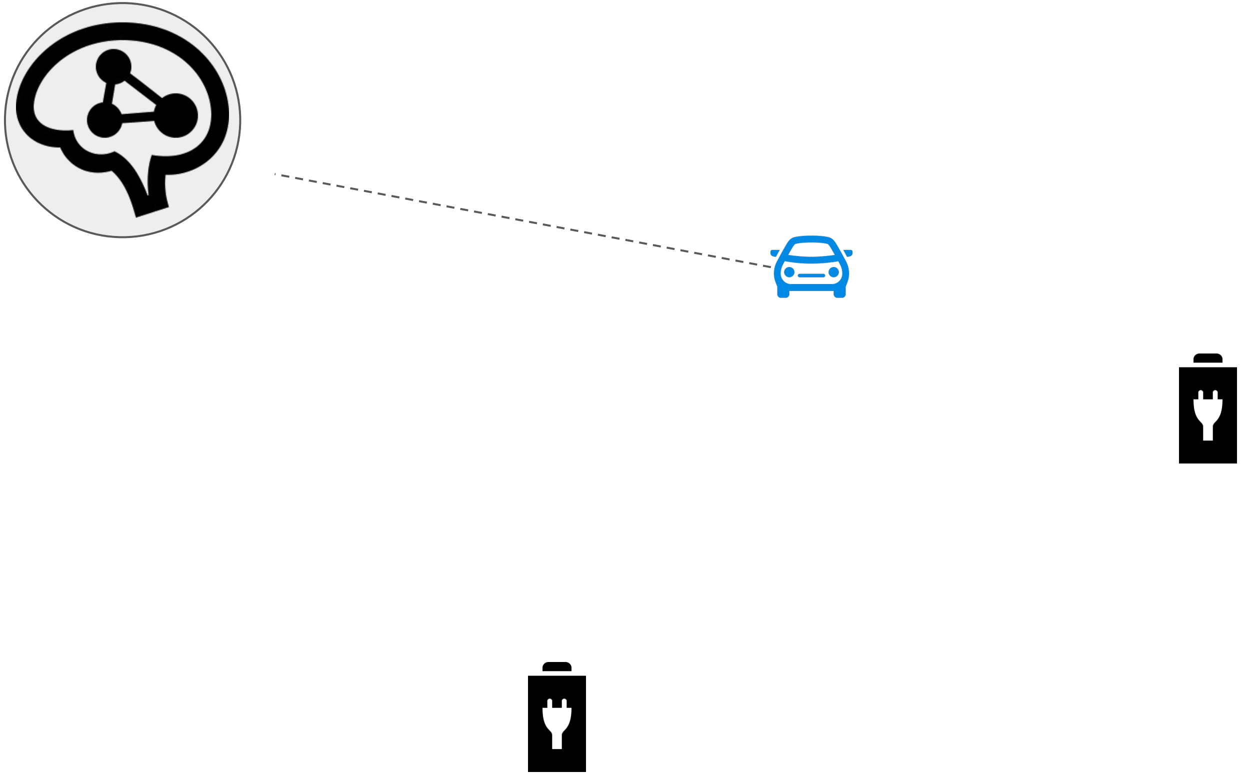
How to Control a Fleet of AEVs

How to Control a Fleet of AEVs

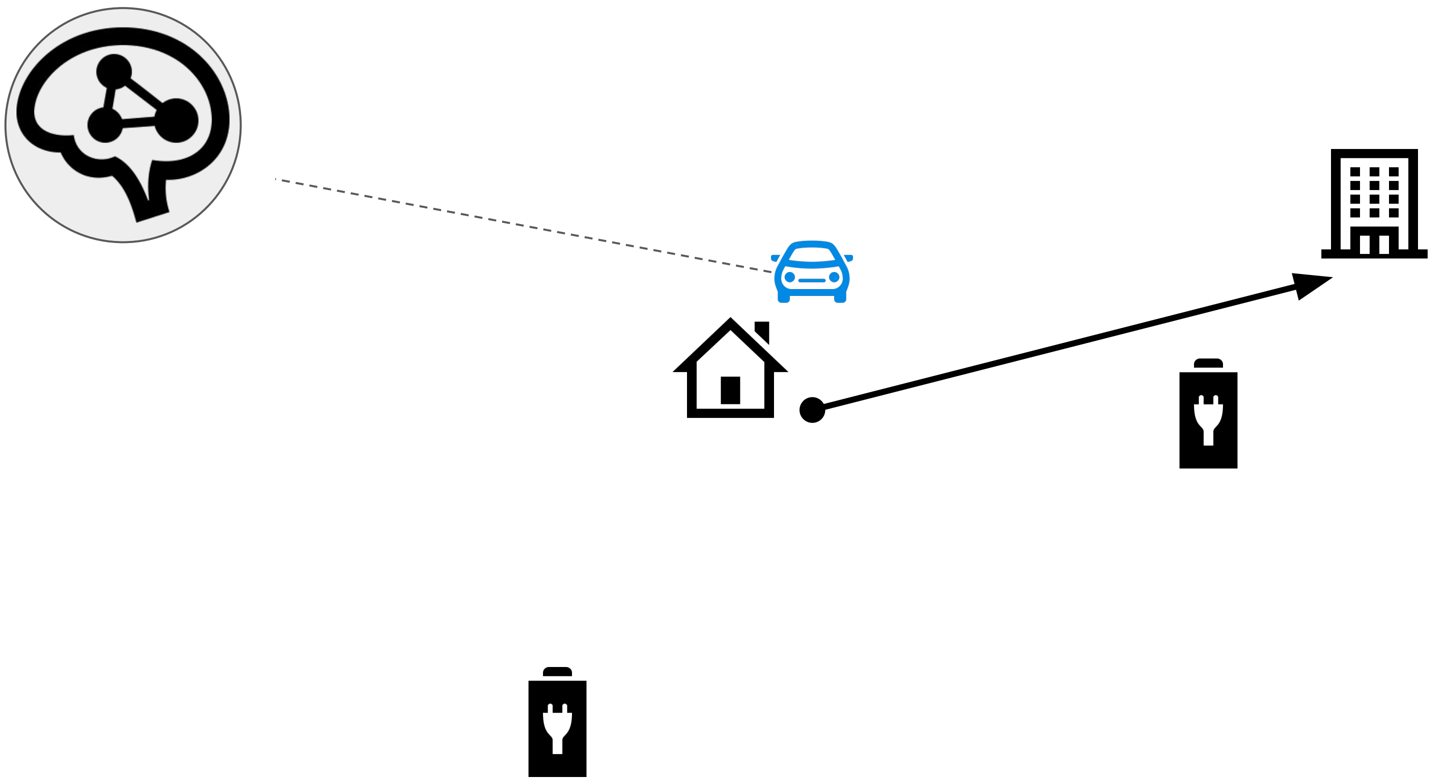
Assign vehicles to requests

Recharge/reposition vehicles

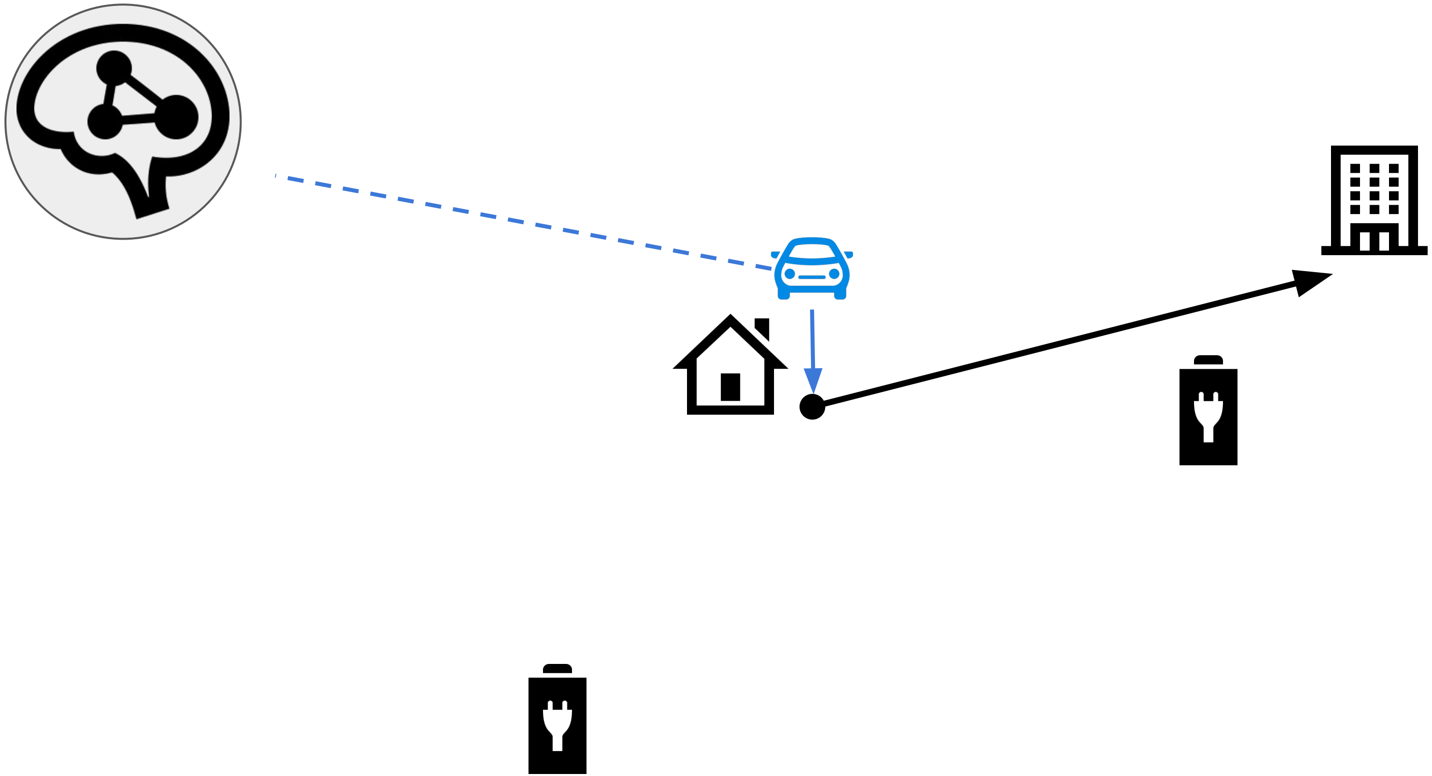
How to Control a Fleet of AEVs



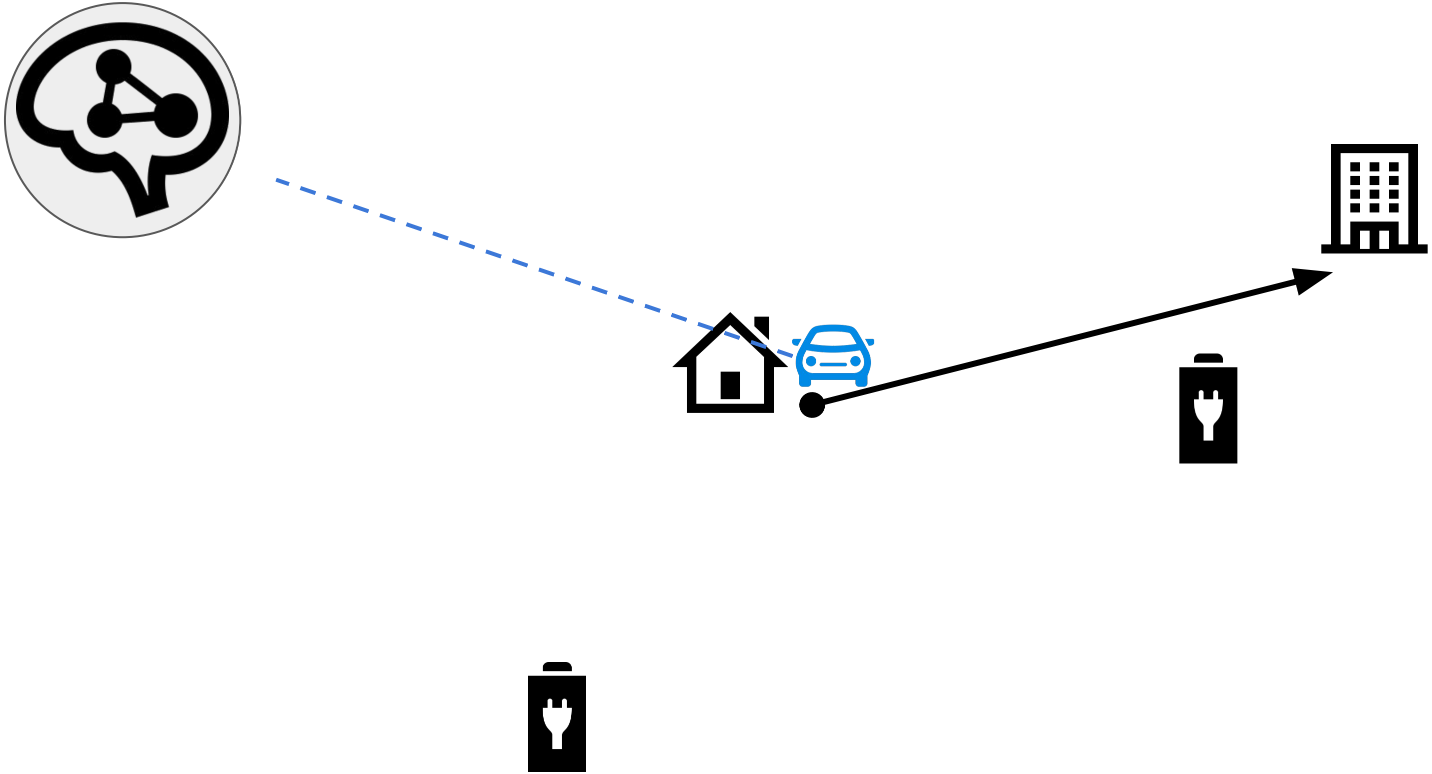
How to Control a Fleet of AEVs



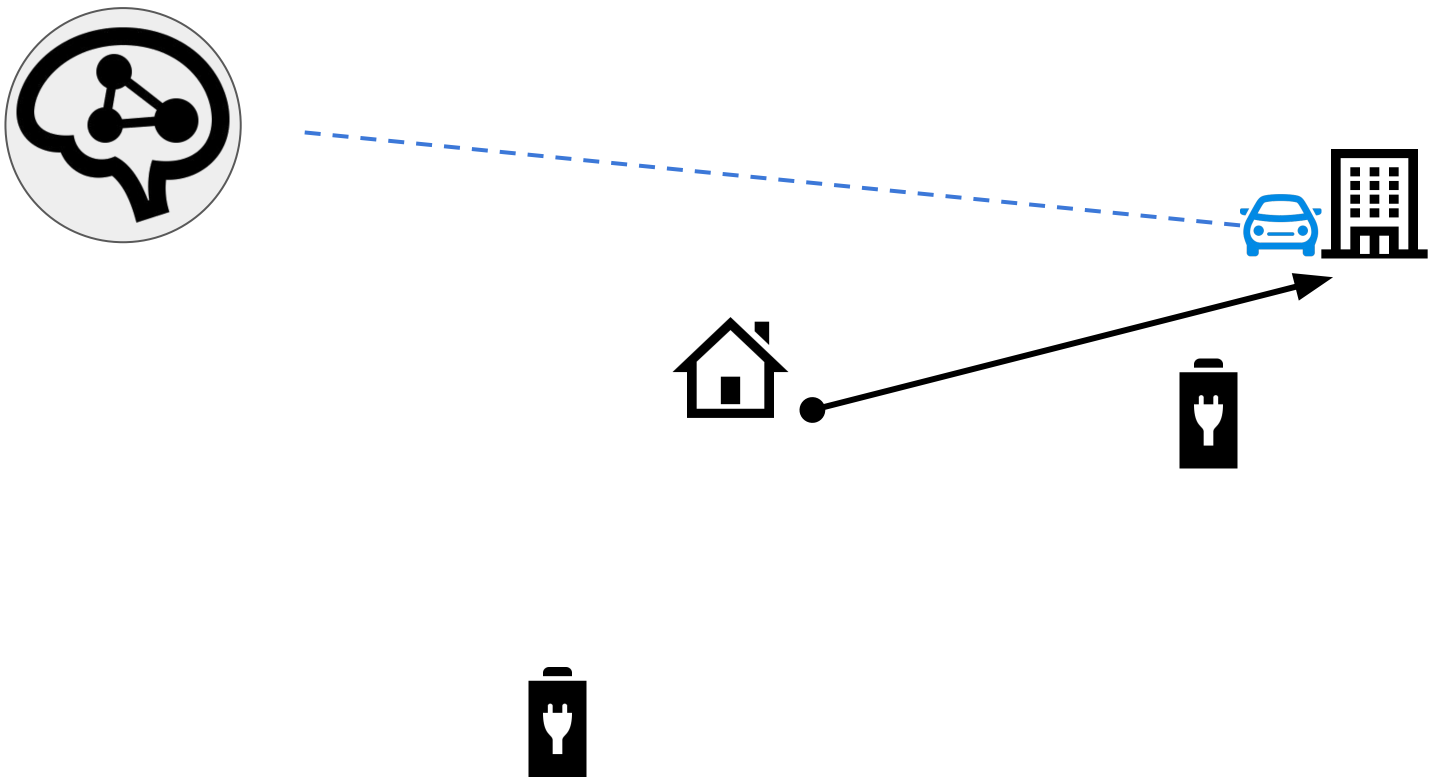
How to Control a Fleet of AEVs



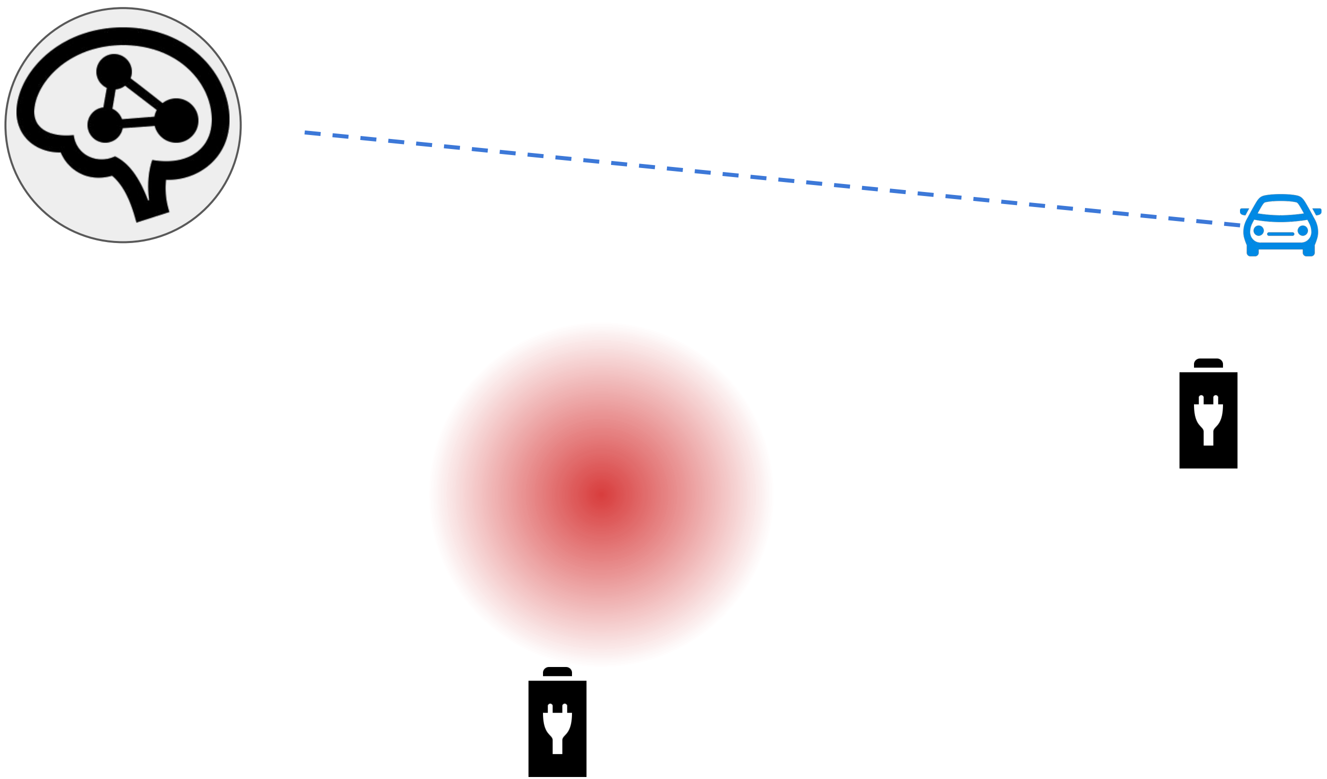
How to Control a Fleet of AEVs



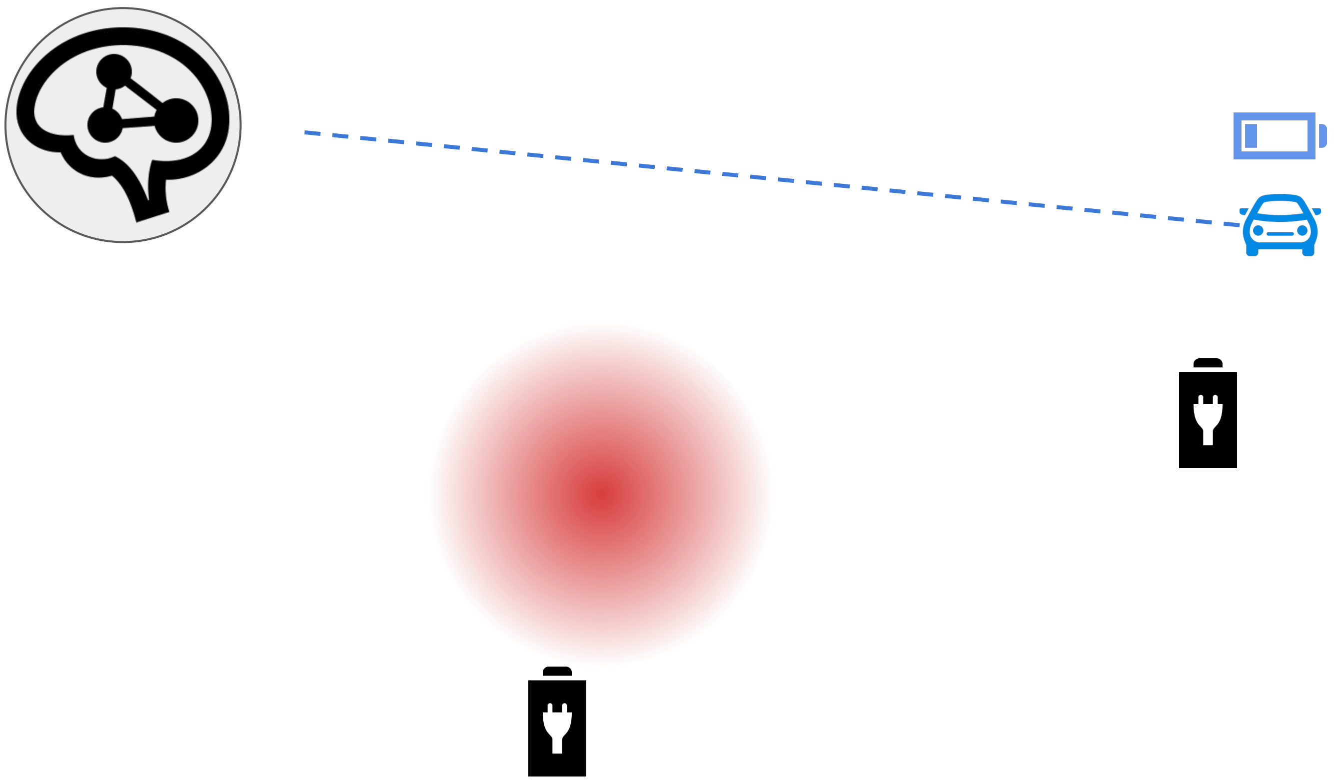
How to Control a Fleet of AEVs



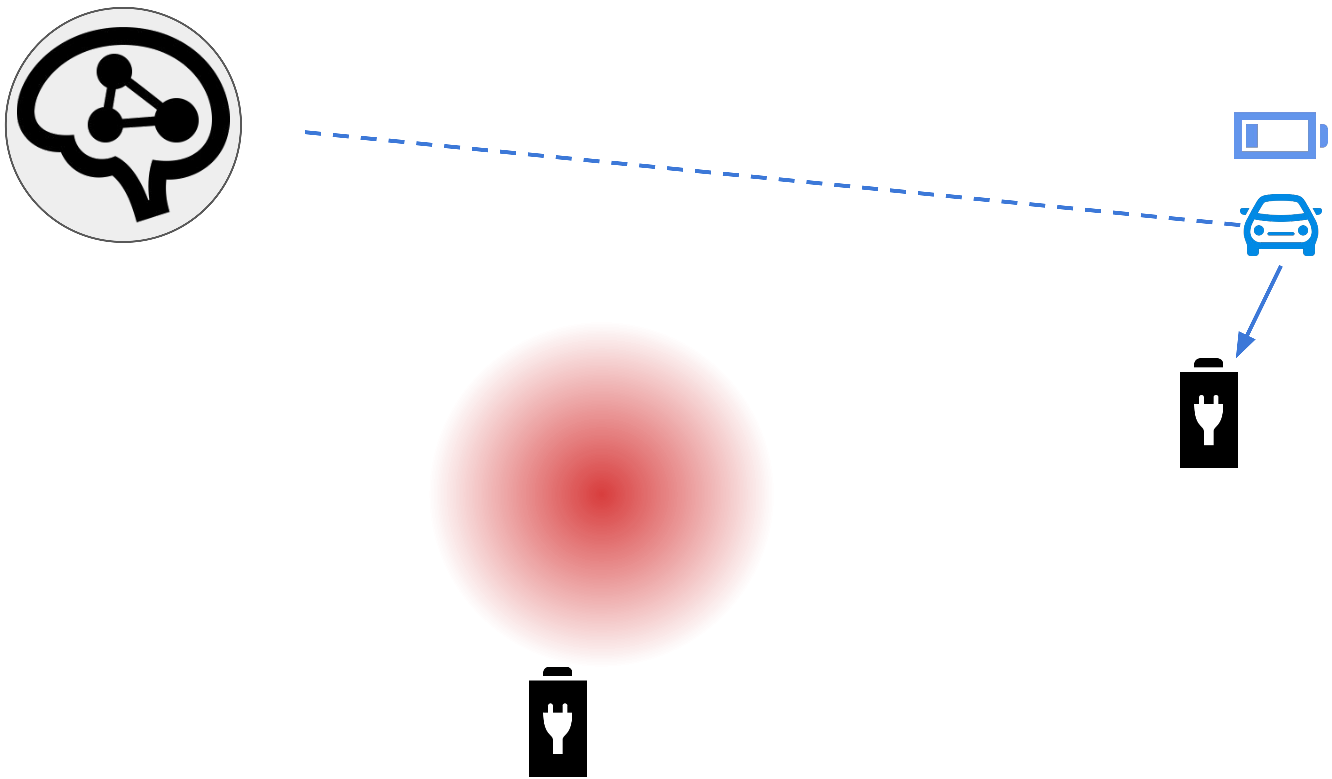
How to Control a Fleet of AEVs



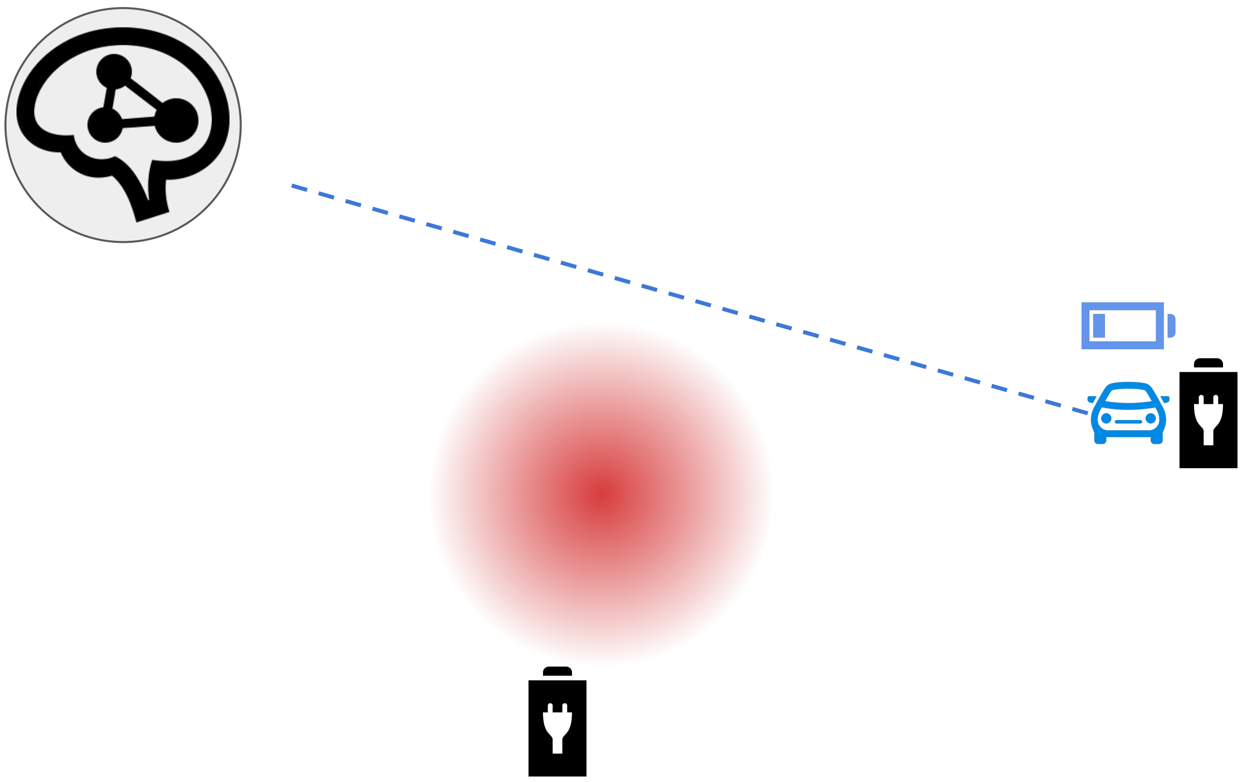
How to Control a Fleet of AEVs



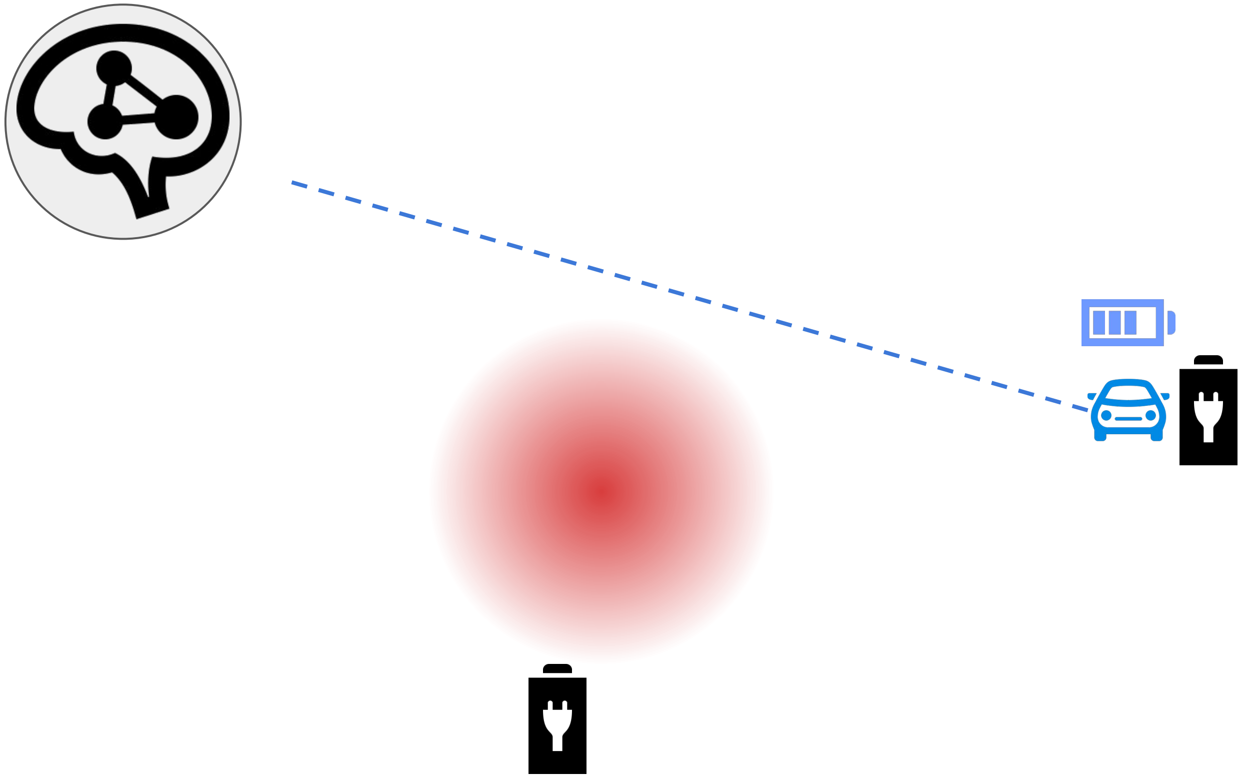
How to Control a Fleet of AEVs



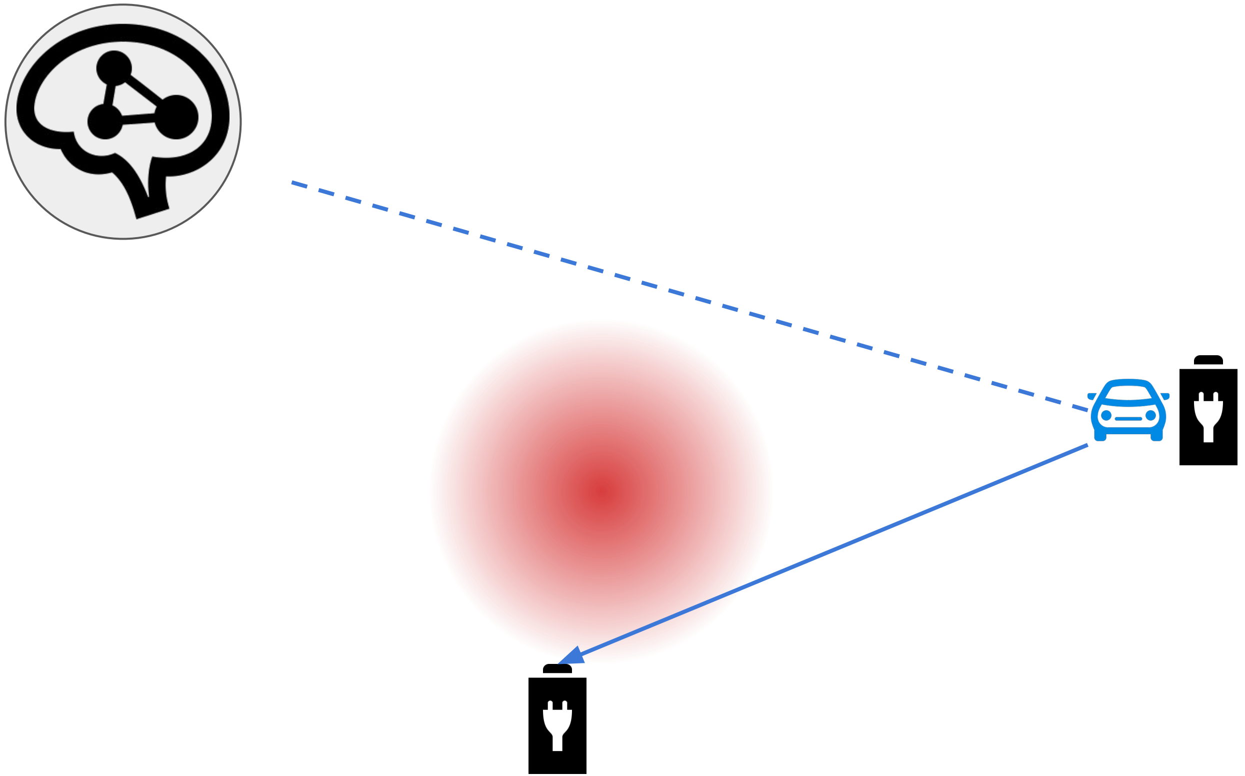
How to Control a Fleet of AEVs



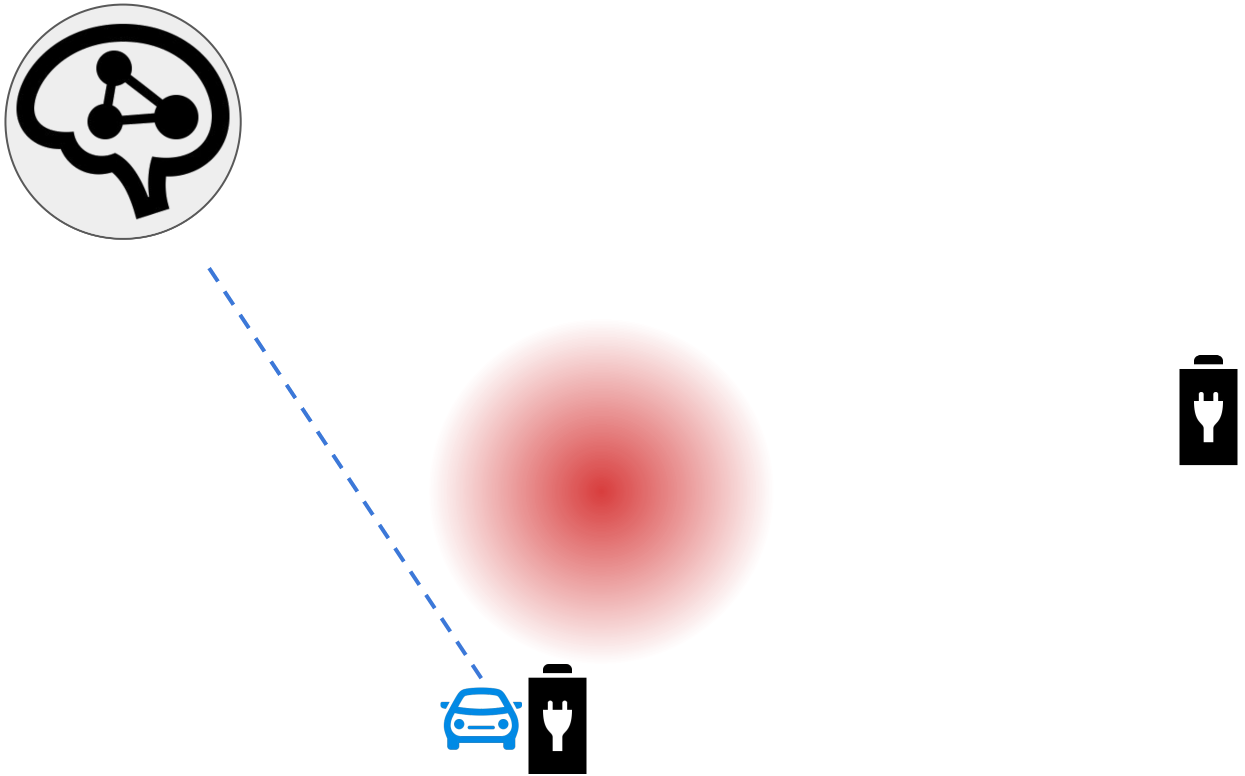
How to Control a Fleet of AEVs



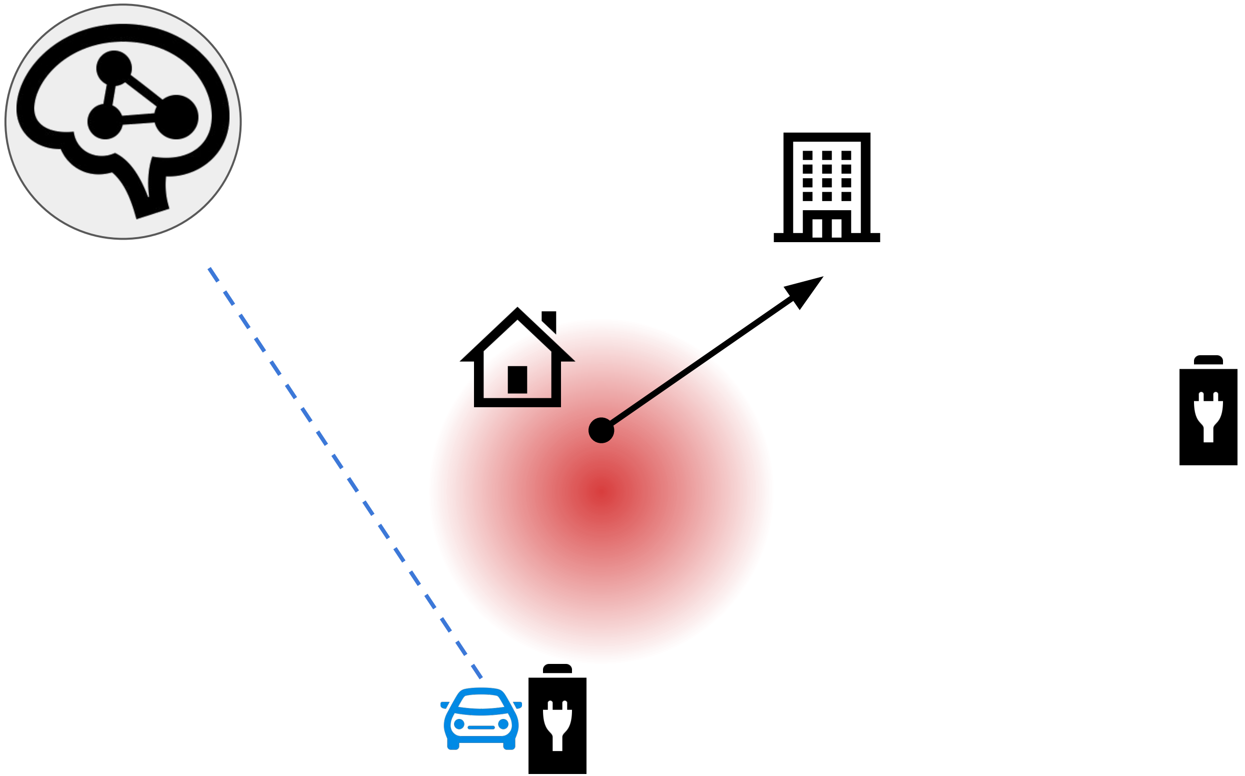
How to Control a Fleet of AEVs



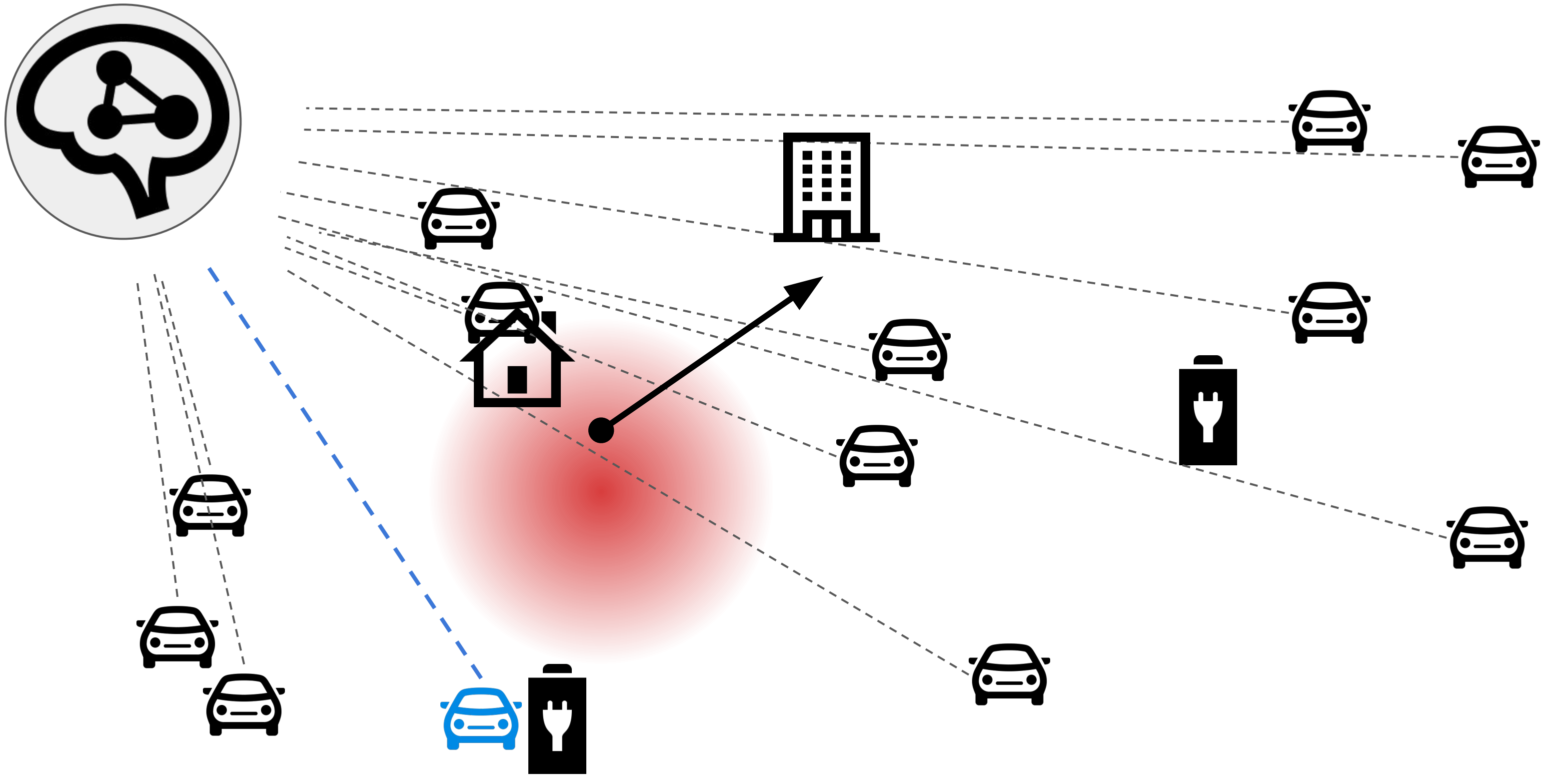
How to Control a Fleet of AEVs



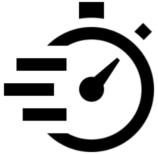
How to Control a Fleet of AEVs



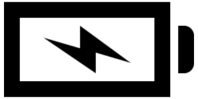
How to Control a Fleet of AEVs



How to Control a Fleet of AEVs



Real time decision making

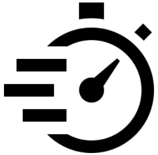


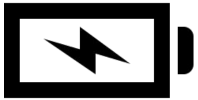
Electrification constraints

Realistic instances

How to Control a Fleet of AEVs

Past Studies





How to Control a Fleet of AEVs

Past Studies

Al-Kanj et al. (2018)	✔ -	✔	X*

How to Control a Fleet of AEVs

Past Studies

Al-Kanj et al. (2018)	✔-	✔	X*
Holler et al. (2018)	✔	X	✔-

How to Control a Fleet of AEVs

Past Studies

Al-Kanj et al. (2018)	✔ -	✔	X*
Holler et al. (2018)	✔	X	✔ -
Bertsimas et al. (2019)	✔ -	X	✔ +

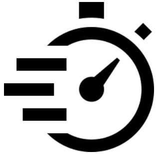
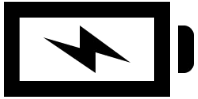
How to Control a Fleet of AEVs

Past Studies

Al-Kanj et al. (2018)	✔-	✔	X*
Holler et al. (2018)	✔	X	✔-
Bertsimas et al. (2019)	✔-	X	✔+
Hyland & Mahmassani (2018)	✔	X	✔-

How to Control a Fleet of AEVs

Past Studies

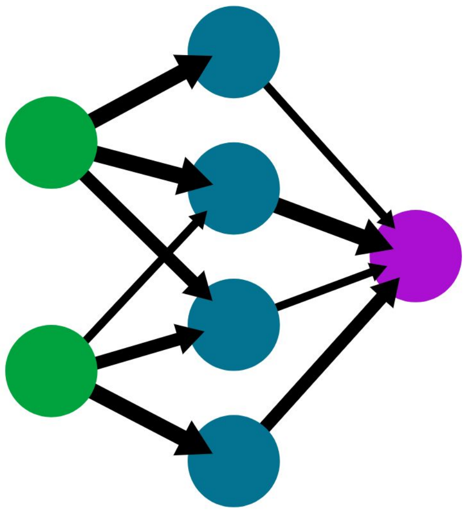
			
Al-Kanj et al. (2018)	✓-	✓	X*
Holler et al. (2018)	✓	X	✓-
Bertsimas et al. (2019)	✓-	X	✓+
Hyland & Mahmassani (2018)	✓	X	✓-
Kullman et al. (2019)	✓	✓	✓

Model & Methods

Methods combine Deep RL and OR

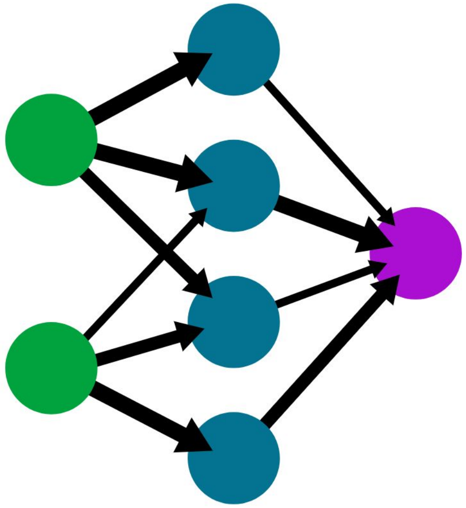
Methods combine Deep RL and OR

Dynamic problem

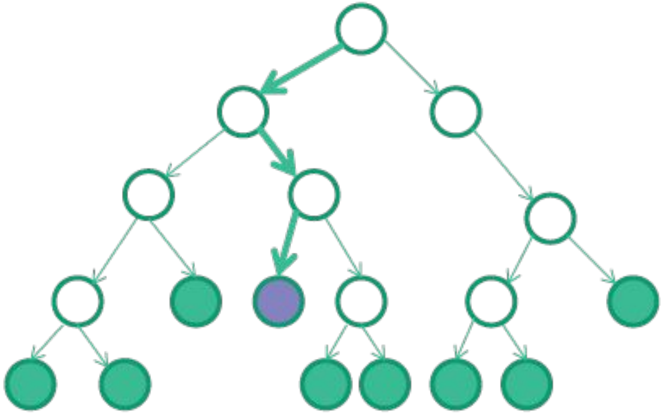


Methods combine Deep RL and OR

Dynamic problem

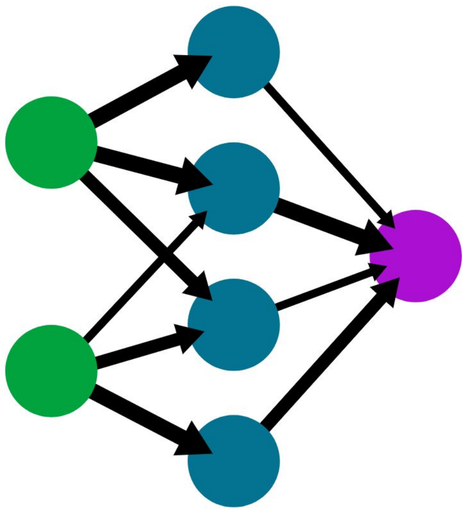


Static Bound



Methods combine Deep RL and OR

Dynamic problem

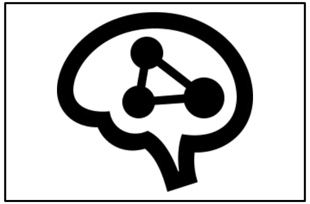


Static Bound

Dynamic Problem Model: MDP

Dynamic Problem Model: MDP

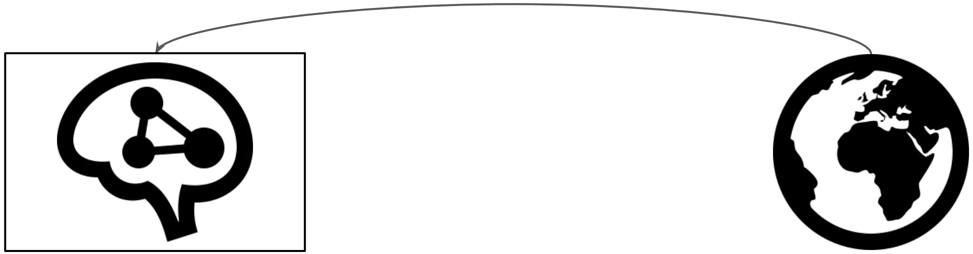
Agent



Environment

Dynamic Problem Model: MDP

State



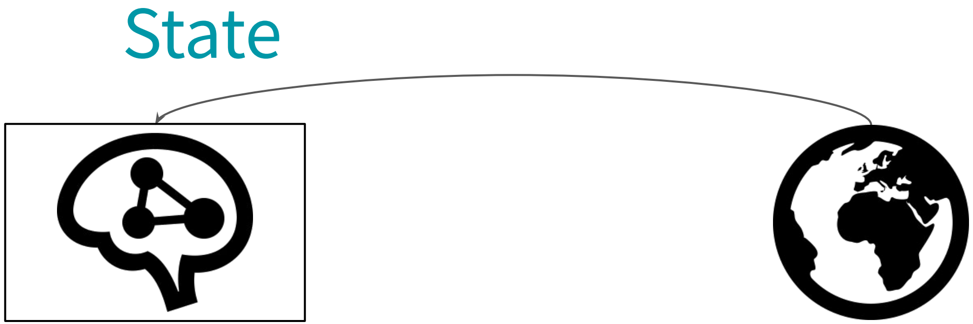
Time

Vehicles':

- Positions
- Charges
- Scheduled jobs

Request

Dynamic Problem Model: MDP



Initial:

Time

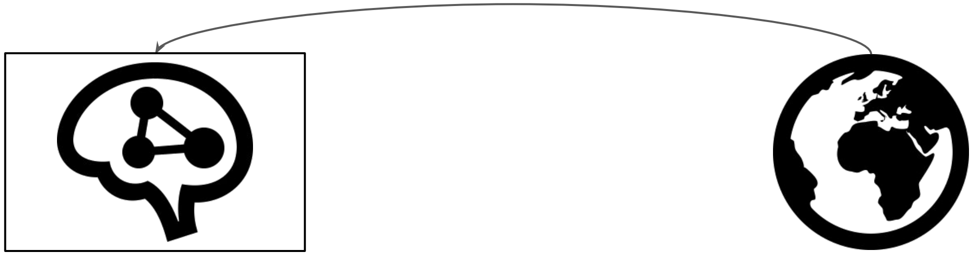
Vehicles':

- Positions
- Charges
- Scheduled jobs

Request

Dynamic Problem Model: MDP

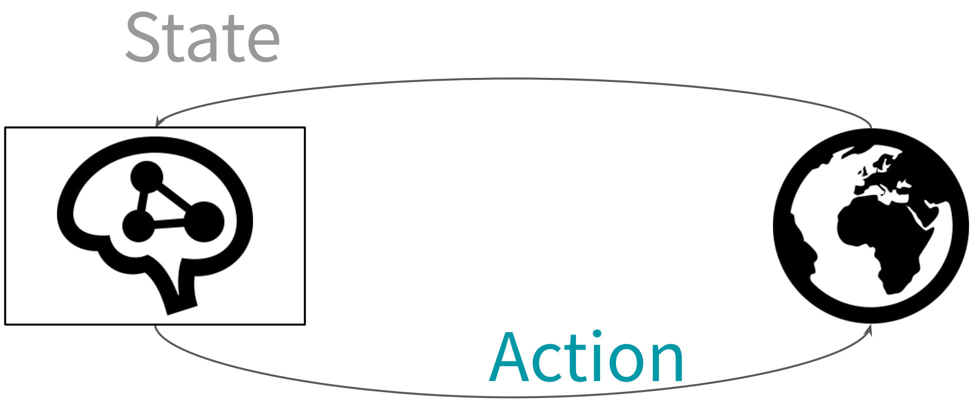
State



Initial:

Time	**0**
Vehicles':	
Positions	**At charging stations**
Charges	$\mathbf{q_{init}}$
Scheduled jobs	**Idle indefinitely**
Request	∅

Dynamic Problem Model: MDP



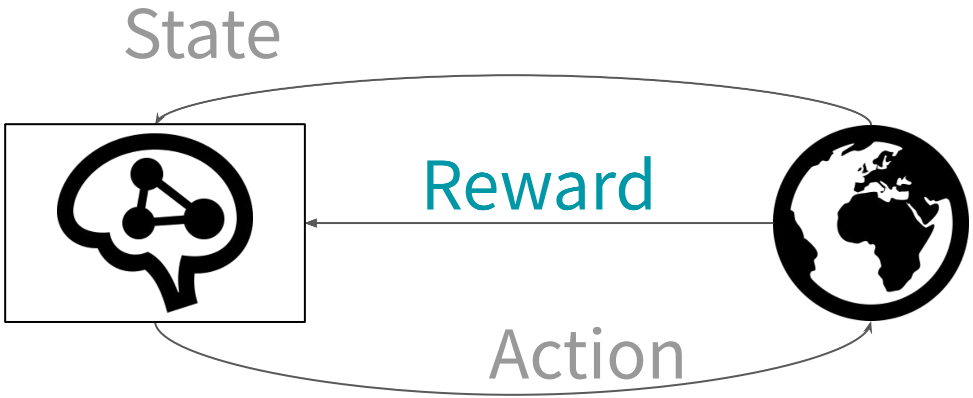
Assign vehicle to new request

For each vehicle:

New reposition/recharge instructions

Depends on state

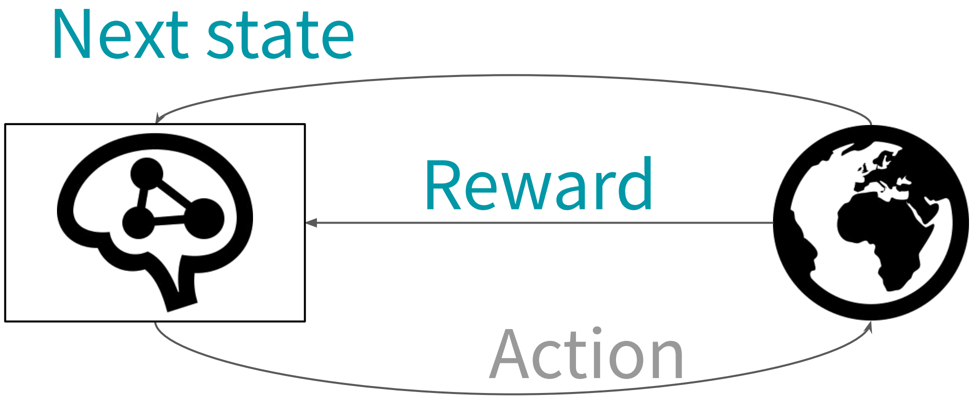
Dynamic Problem Model: MDP



Value of request, if served

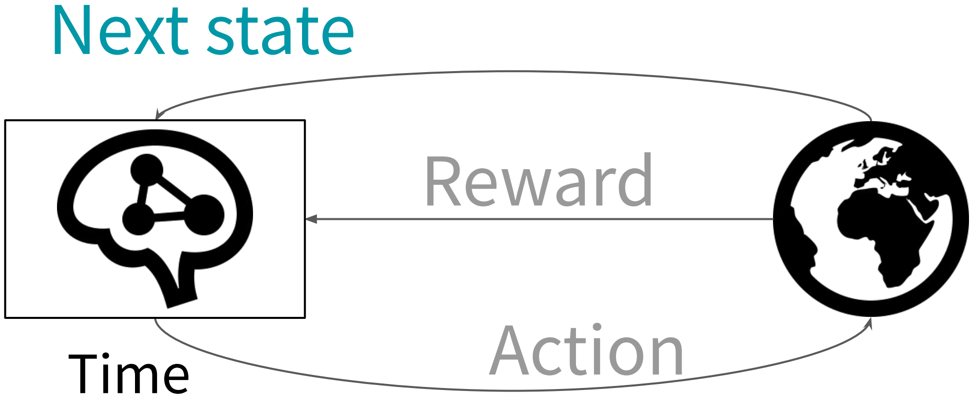
Fixed + distance-dependent

Dynamic Problem Model: MDP



Update state

Dynamic Problem Model: MDP



Time

Vehicles':

Positions

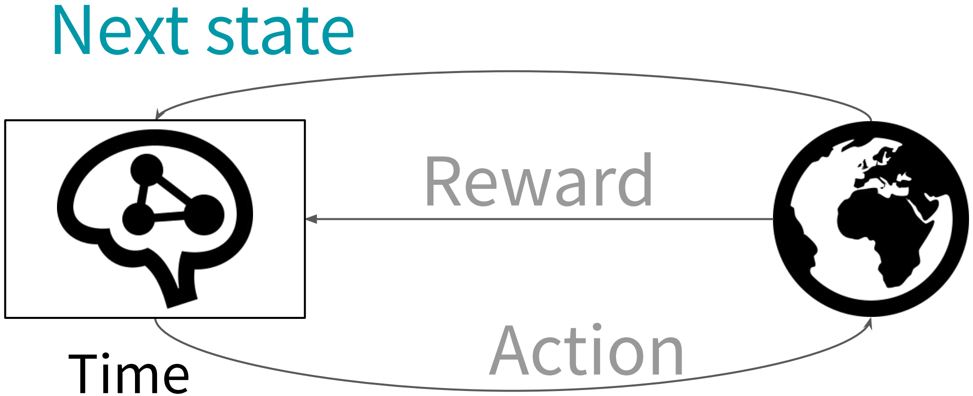
Charges

Scheduled jobs

Terminal:

Request

Dynamic Problem Model: MDP



Time

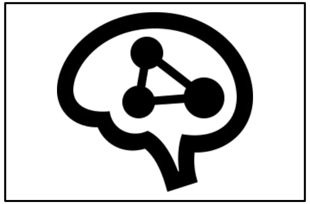
Vehicles':

- Positions
- Charges
- Scheduled jobs

Request

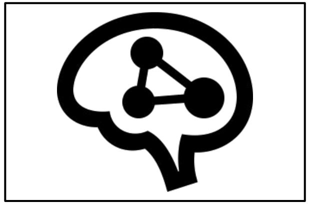
Terminal: Time expired

Dynamic Problem Model: MDP



Objective

Dynamic Problem Model: MDP



Objective

Find agent/policy maximizing E[sum of rewards]

Agents

Agents

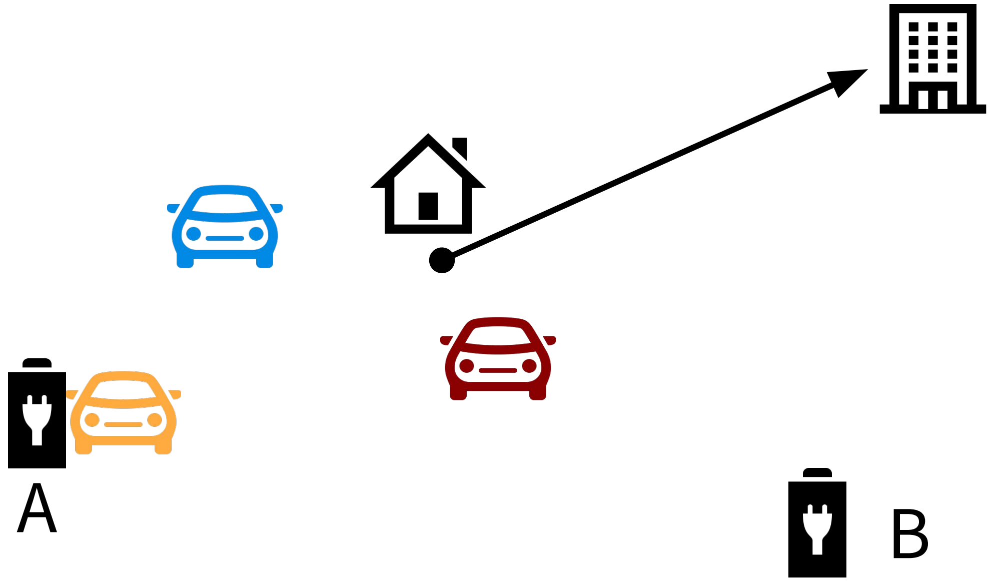
1. Random

Agents

1. Random
2. Nearest (heuristic)

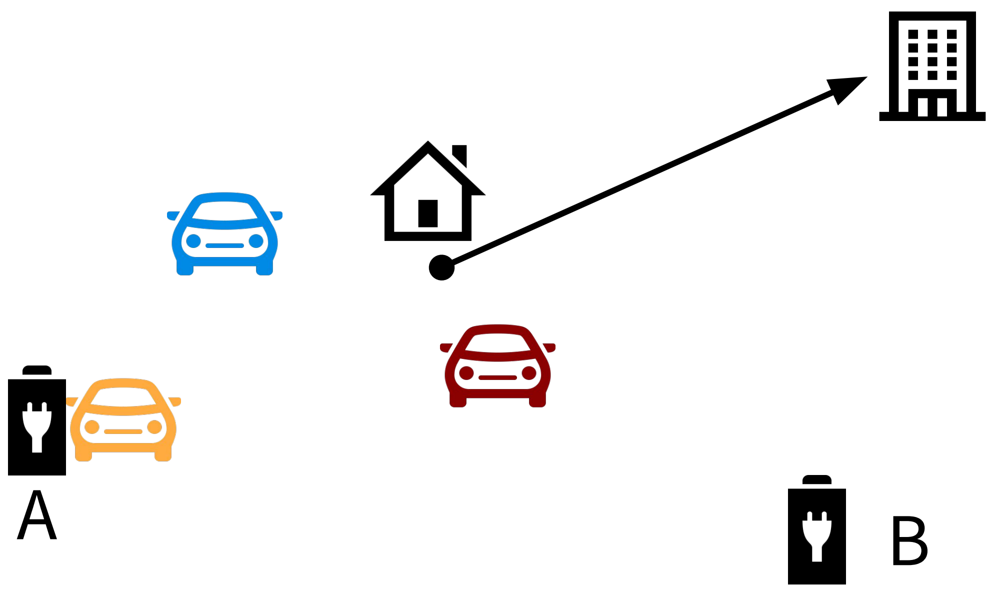
Agents

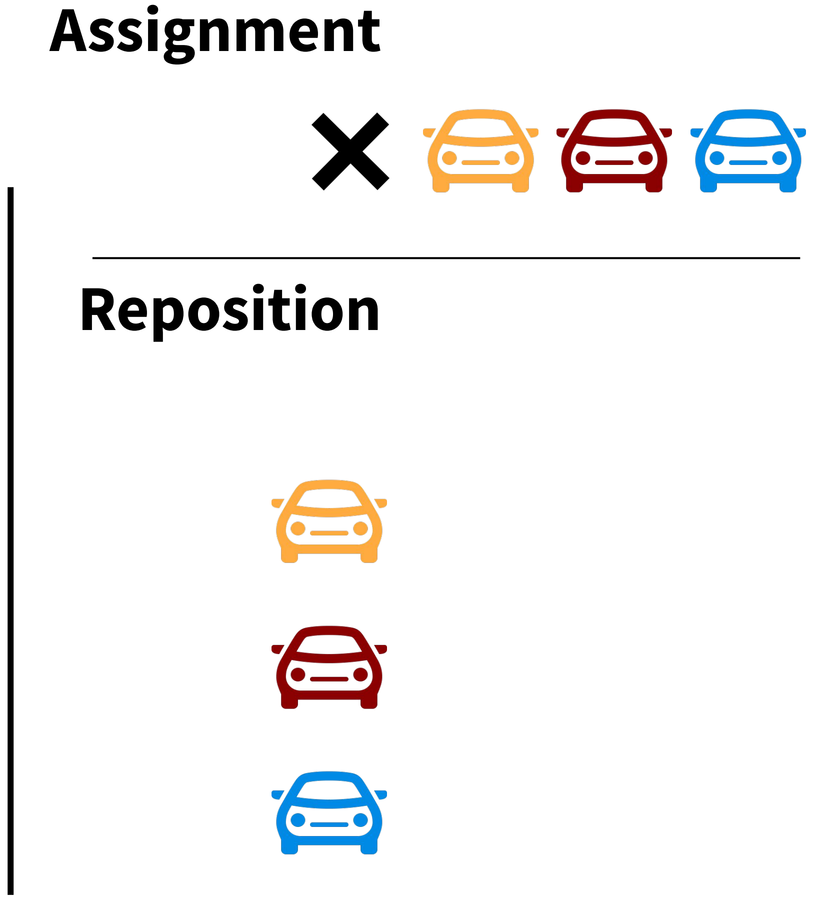
1. Random
2. Nearest



Agents

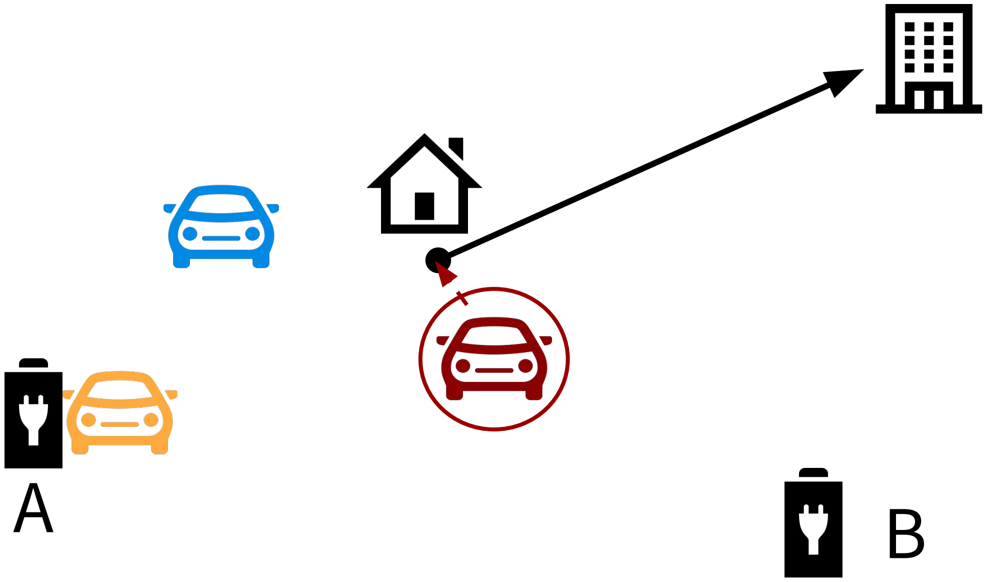
1. Random
2. Nearest

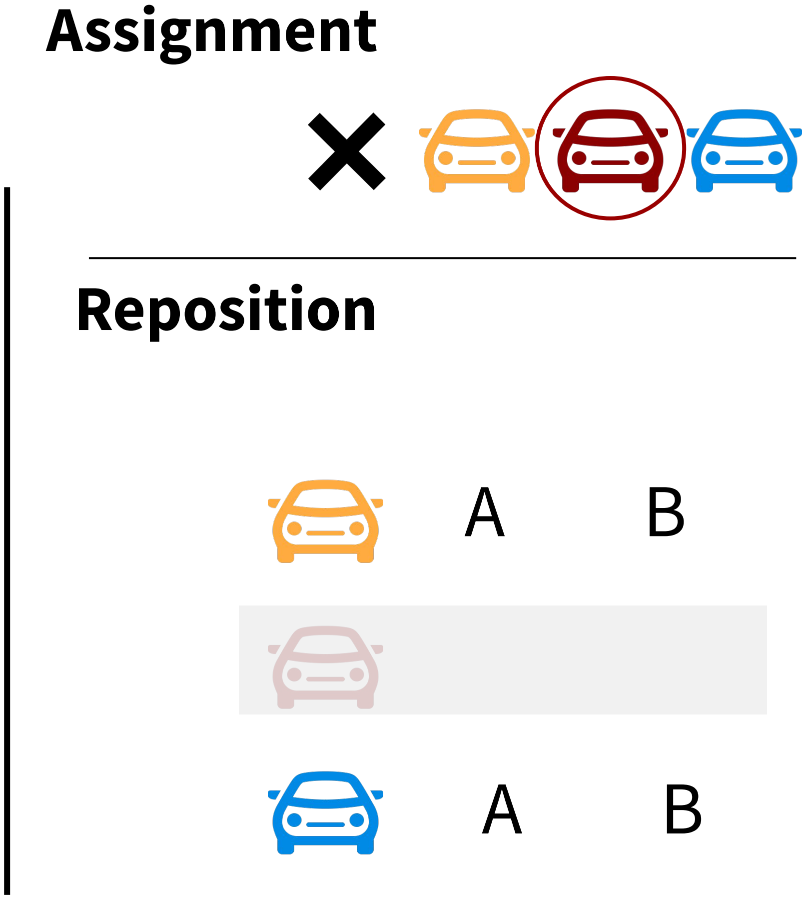




Agents

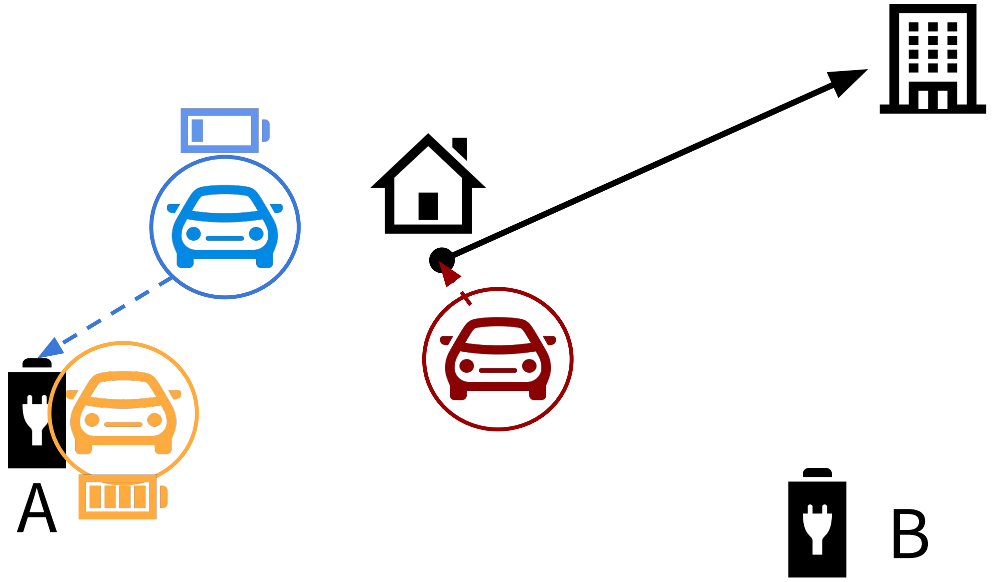
1. Random
2. Nearest



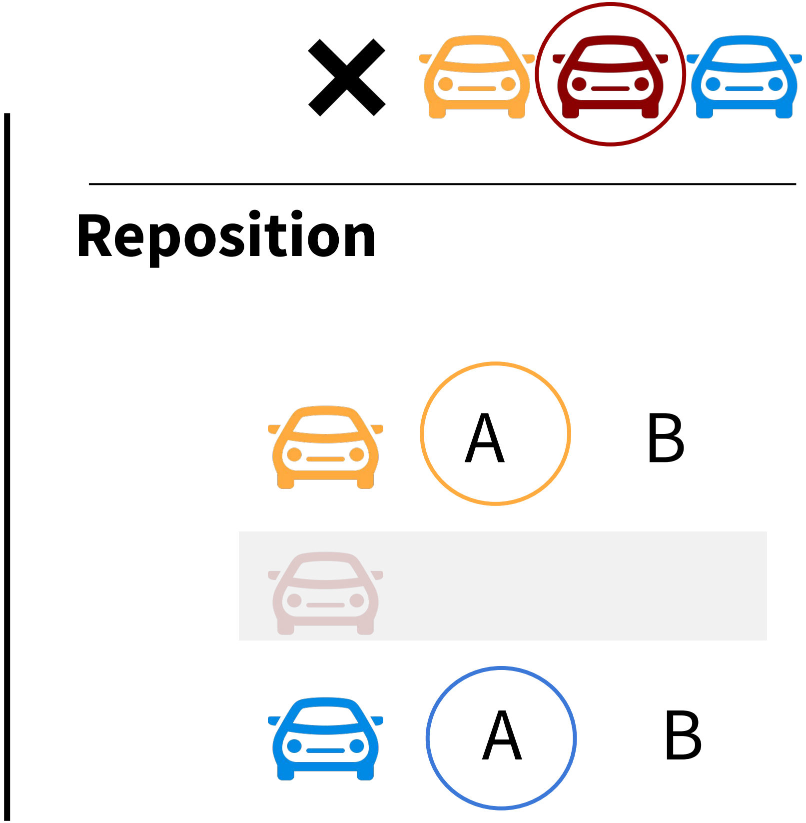


Agents

1. Random
2. Nearest



Assignment

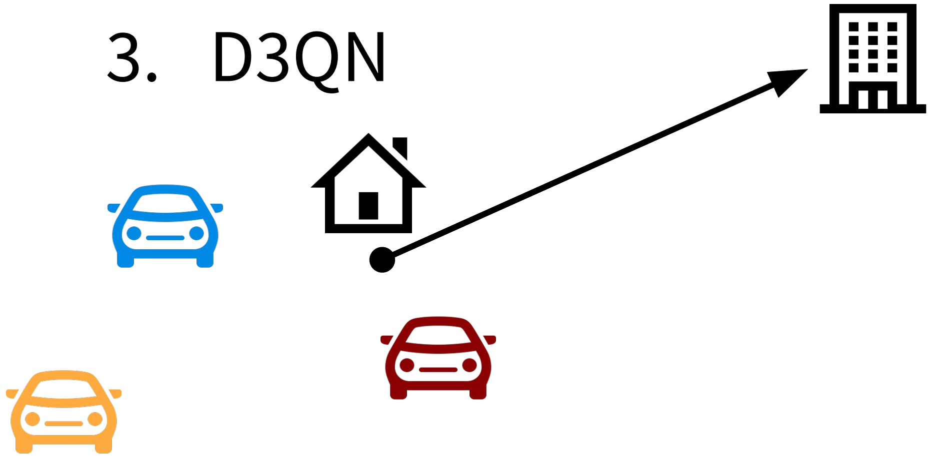


Agents

1. Random
2. Nearest
3. D3QN

Agents

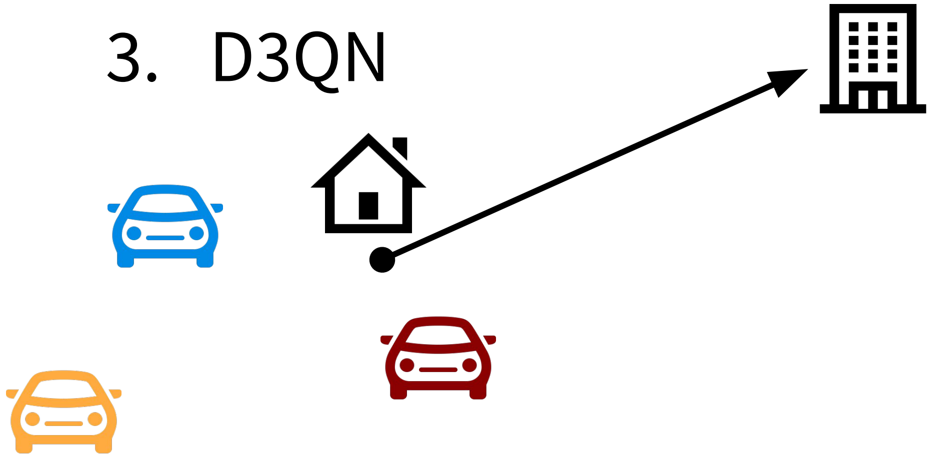
1. Random
2. Nearest
3. D3QN

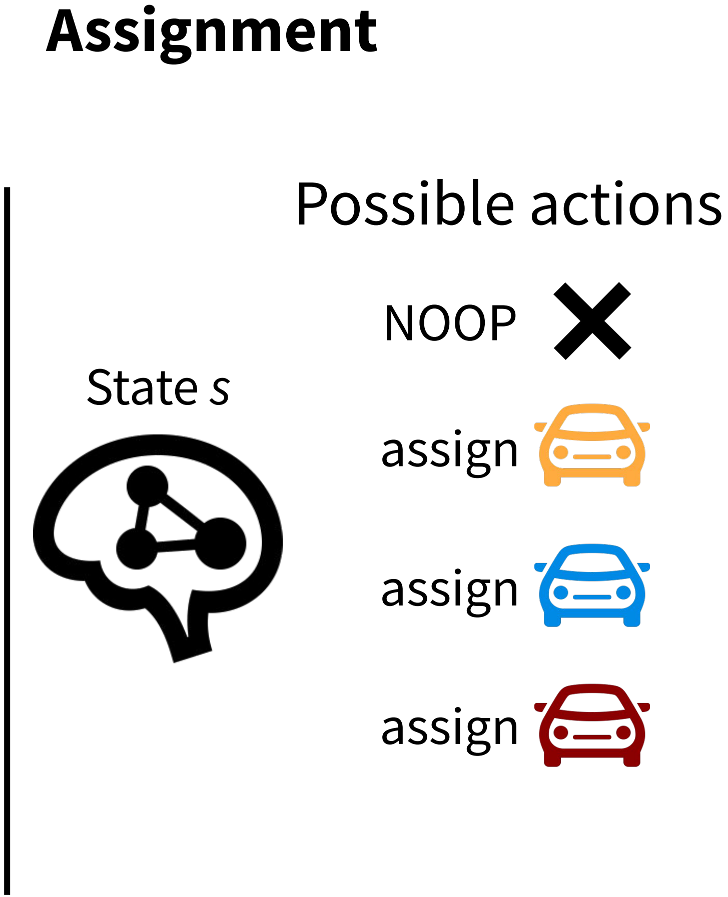


Assignment

Agents

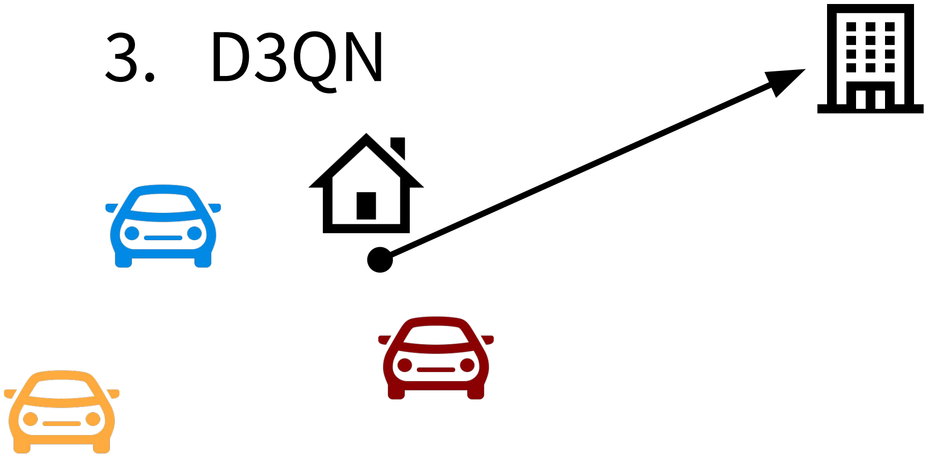
1. Random
2. Nearest
3. D3QN



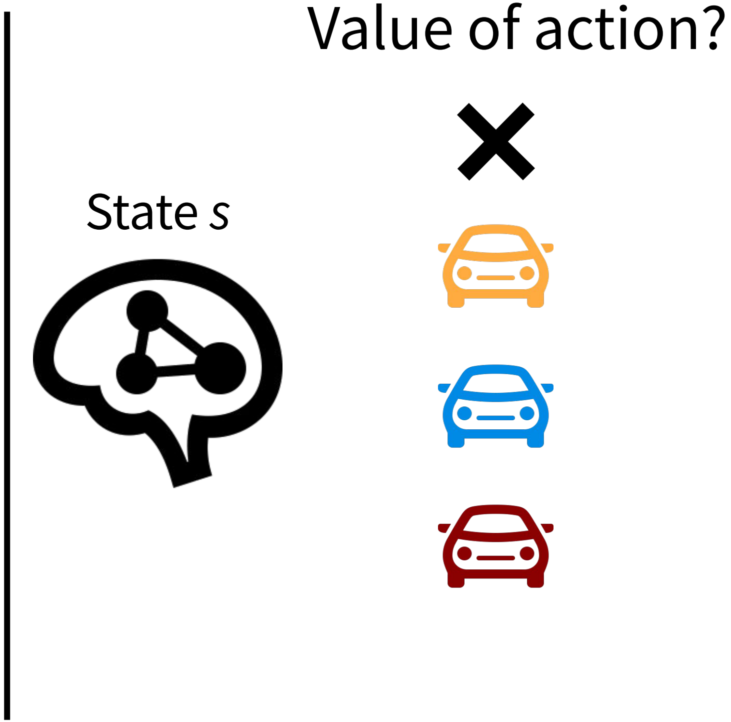


Agents

1. Random
2. Nearest
3. D3QN

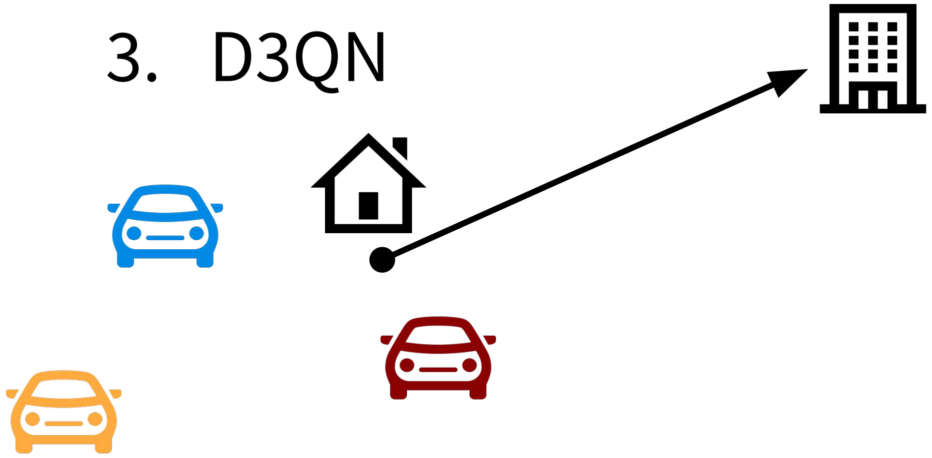


Assignment



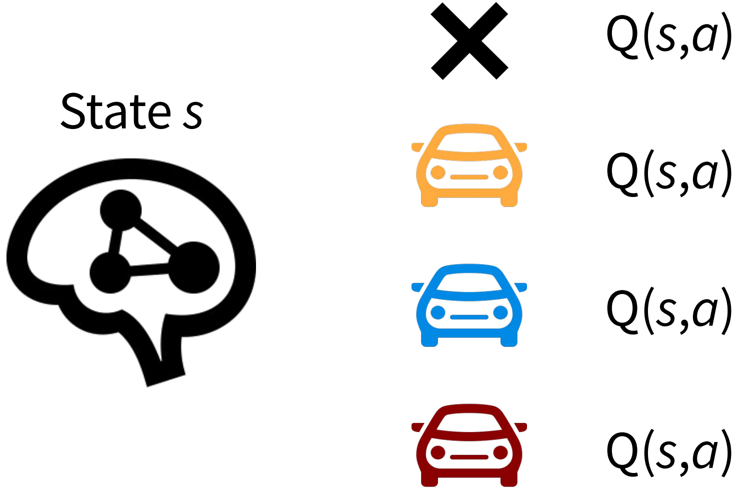
Agents

1. Random
2. Nearest
3. D3QN



Assignment

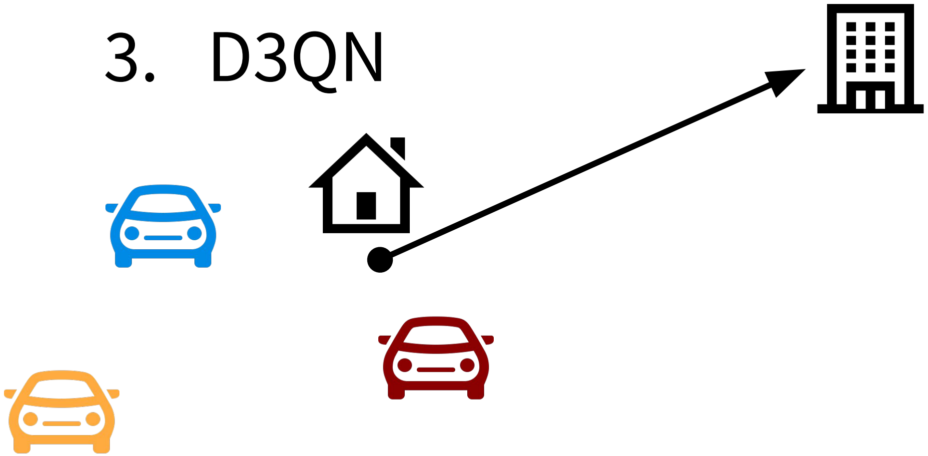
Value of action?



Immediate reward + reward-to-go

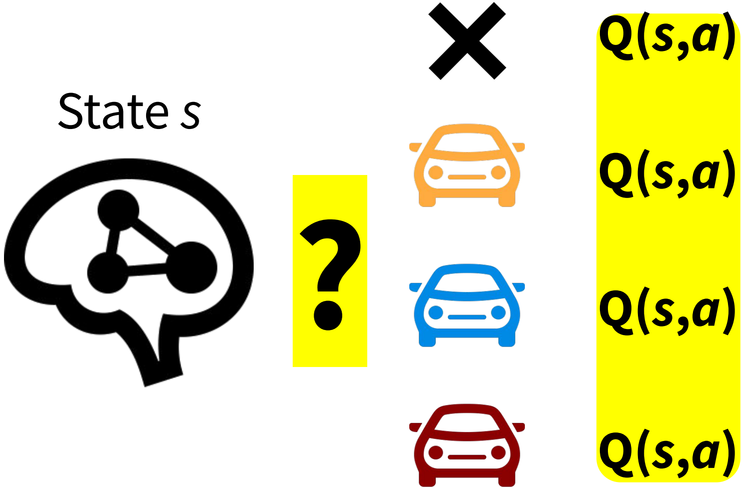
Agents

1. Random
2. Nearest
3. D3QN



Assignment

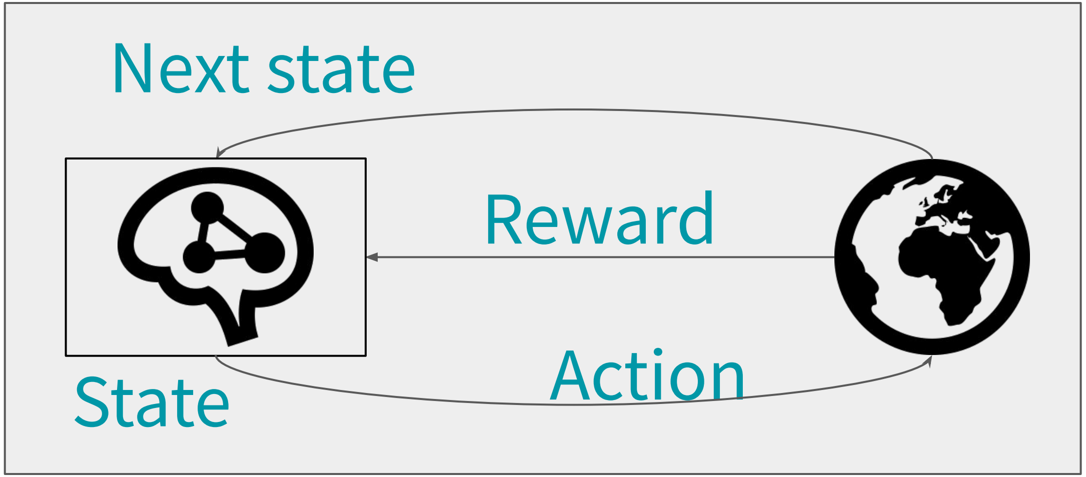
Value of action?



How to learn Q(,)?

Agents

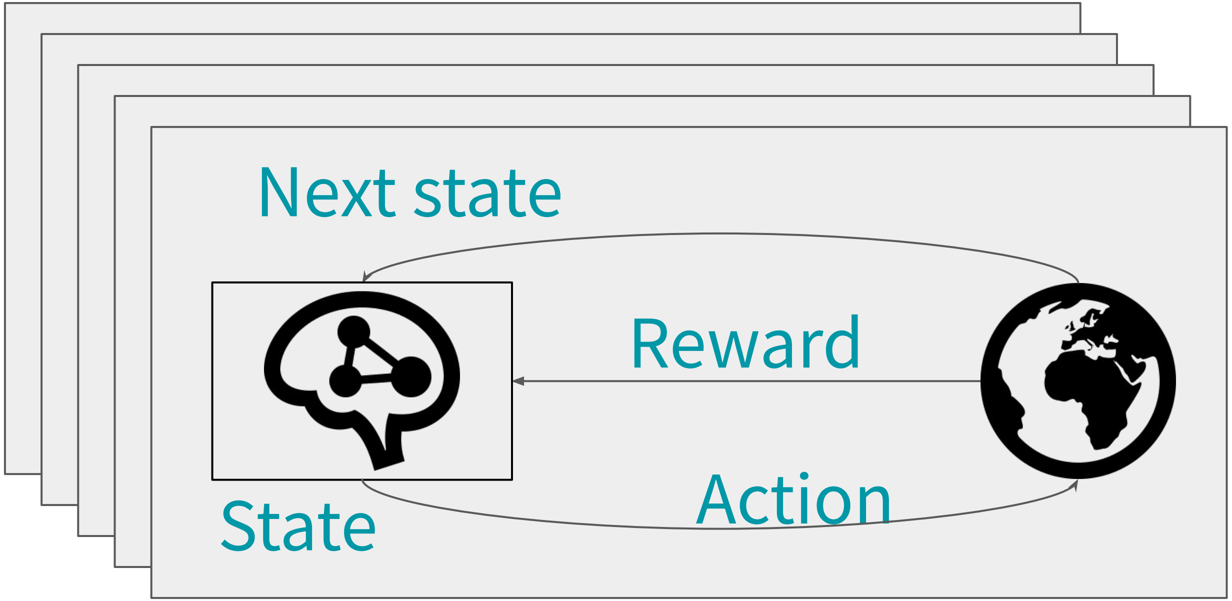
1. Random
2. Nearest
3. D3QN



How does it learn?

Agents

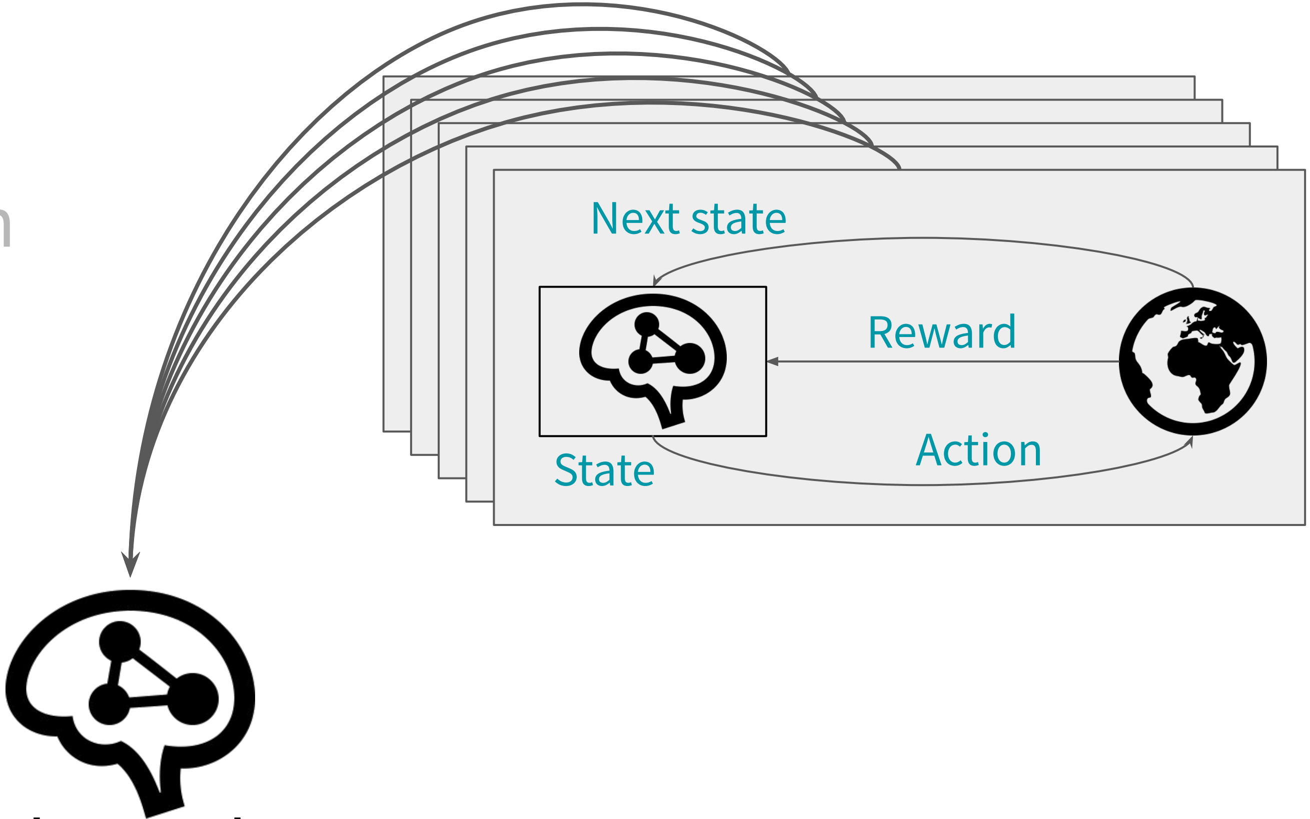
1. Random
2. Nearest
3. D3QN



How does it learn?

Agents

1. Random
2. Nearest
3. D3QN



How does it learn?

Agents

1. Random
2. Nearest
3. D3QN

Architecture

Agents

1. Random
2. Nearest
3. D3QN

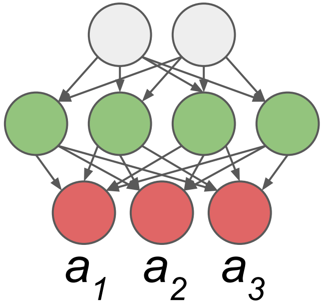
Architecture

DQN

Input s

"Hidden"

$Q(s, a)$



Agents

1. Random
2. Nearest
3. D3QN

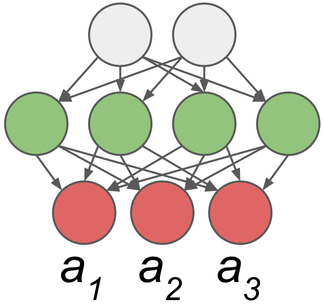
Architecture

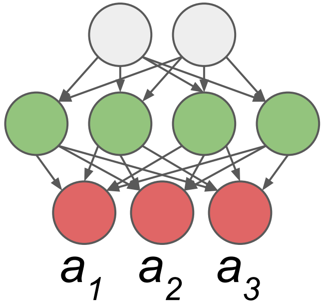
DDQN

Input s

"Hidden"

Q(s, a)





Primary

Selection

Target

Evaluation

Agents

1. Random
2. Nearest
3. D3QN

Architecture

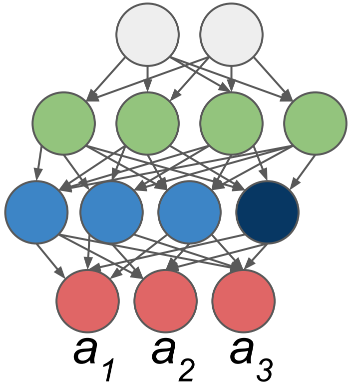
D3QN

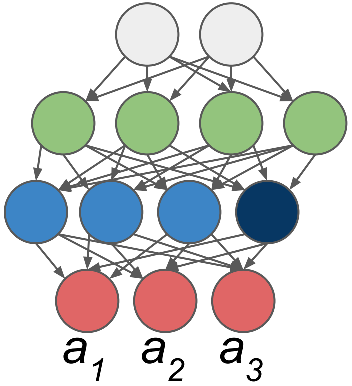
Input s

"Hidden"

A(*s*,*a*) + V(*s*)

$Q(s, a)$



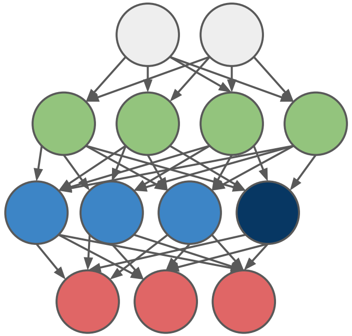


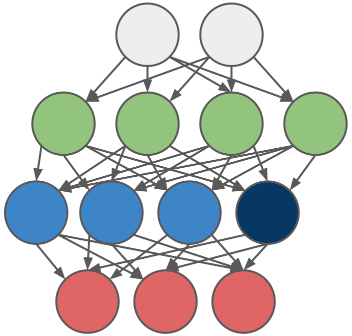
Agents

1. Random
2. Nearest
3. D3QN**

Architecture

D3QN

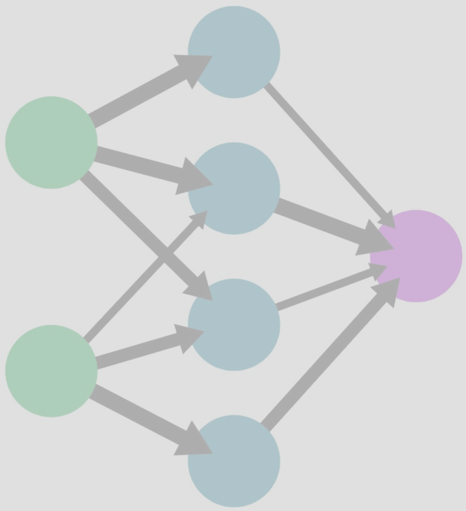




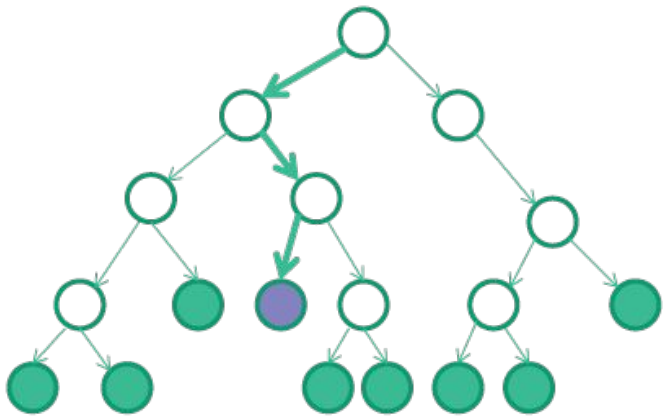
Currently just responsible for vehicle-request pairing

Methods combine Deep RL and OR

Dynamic problem



Static Bound



Static Problem

Static Problem

Perfect Information

Perfect Info: OR tools

Perfect Info: Benders decomposition

Perfect Info: Benders decomposition

Master problem

Subproblem

Perfect Info: Benders decomposition

Master problem

Assign requests to vehicles

Time feasibility

Subproblem

Perfect Info: Benders decomposition

Master problem

Assign requests to vehicles

Time feasibility

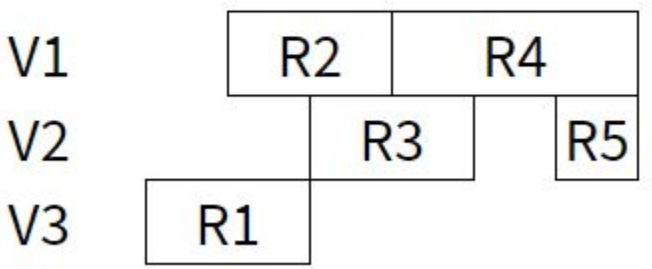
Subproblem

Energy feasibility

Charging decisions

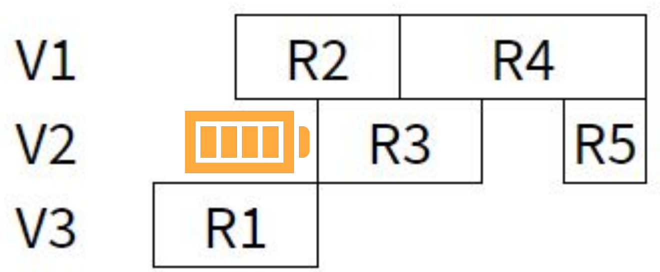
Perfect Info: Benders decomposition

Master problem



Perfect Info: Benders decomposition

Master problem

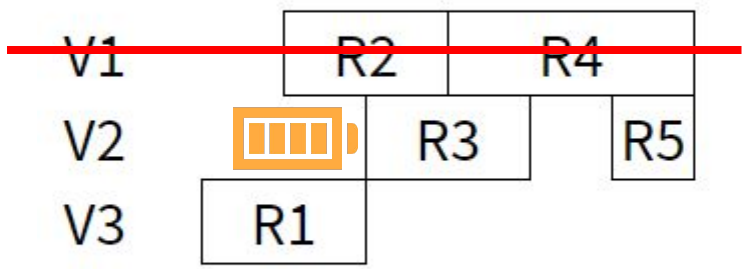


Subproblem

FRVCP: Froger et al. (2018)

Perfect Info: Benders decomposition

Master problem



Subproblem

FRVCP: Froger et al. (2018)

Data & Empirical Results

Data: Manhattan

Data: Manhattan

Trips:

Yellow Taxi + Green Taxi + Ridehail (2017)

Business mornings

100 requests / hour

Data: Manhattan

Trips:

- Yellow Taxi + Green Taxi + Ridehail (2017)

CSs:

- All current and planned CSs

Data: Manhattan

Trips:

- Yellow Taxi + Green Taxi + Ridehail (2017)

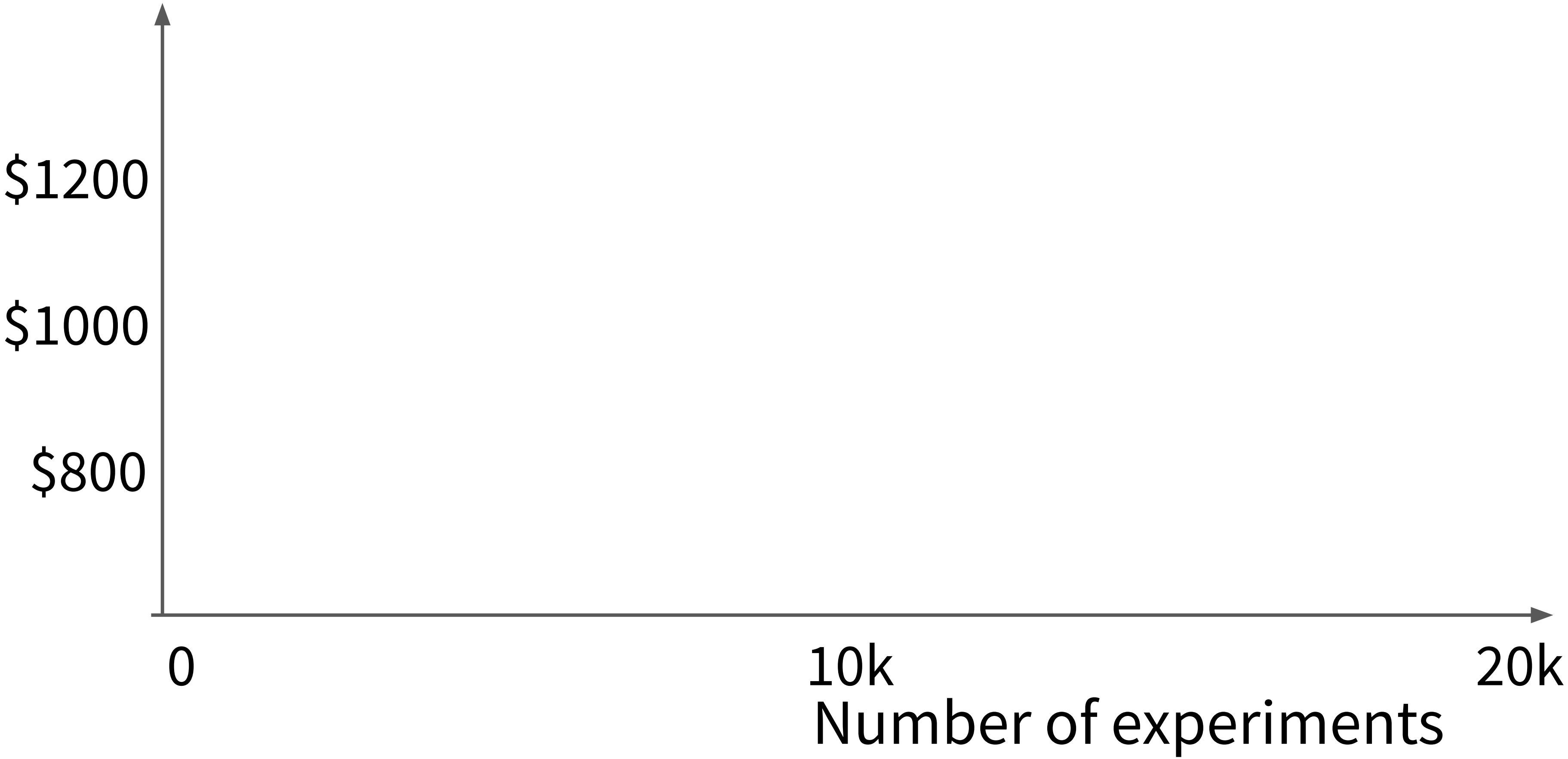
CSs:

- All current and planned CSs

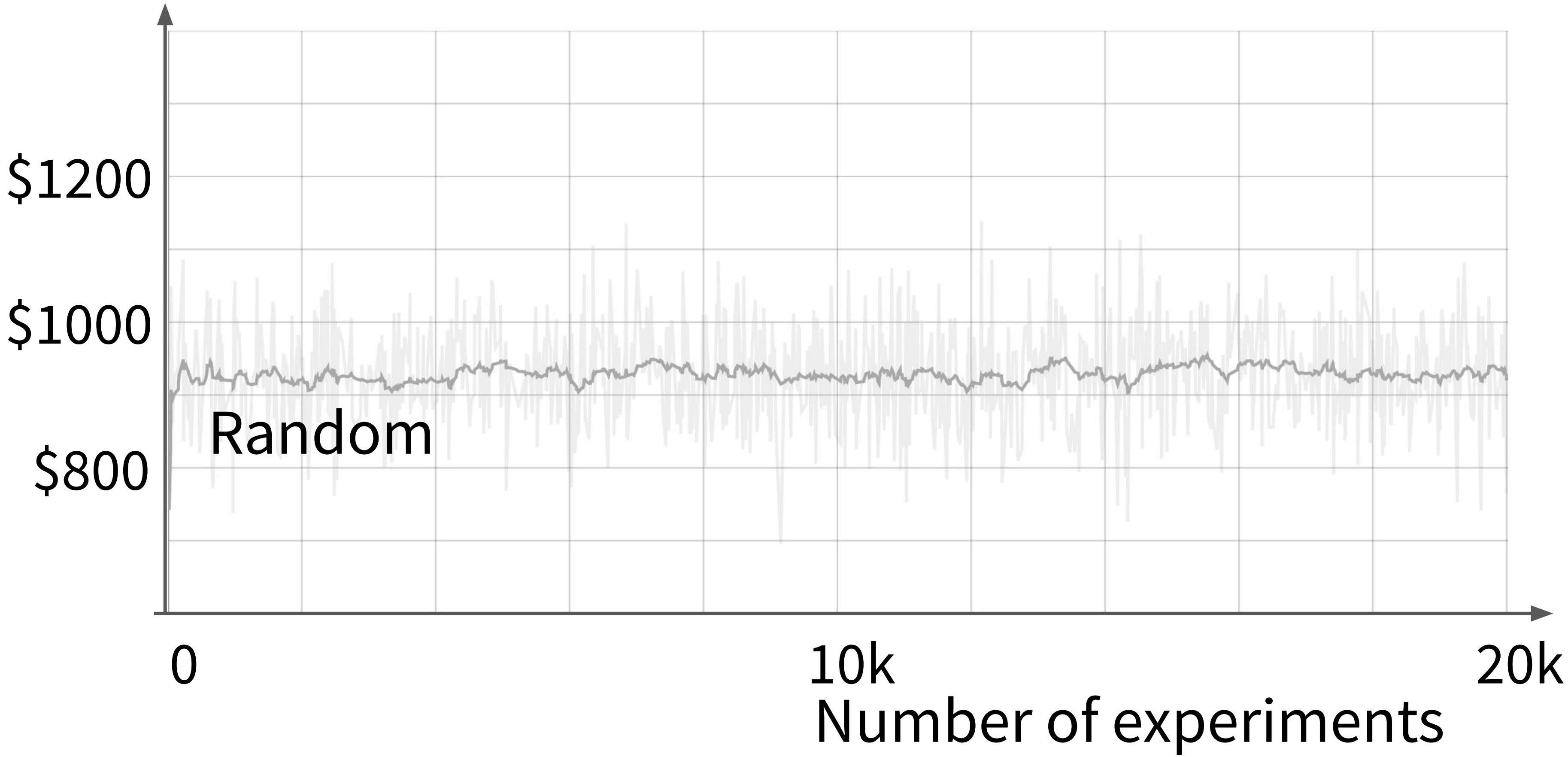
Vehicles:

- Mid-range Tesla Model 3
- Travel at ~avg NYC taxi speed
- *~40 vehicles*

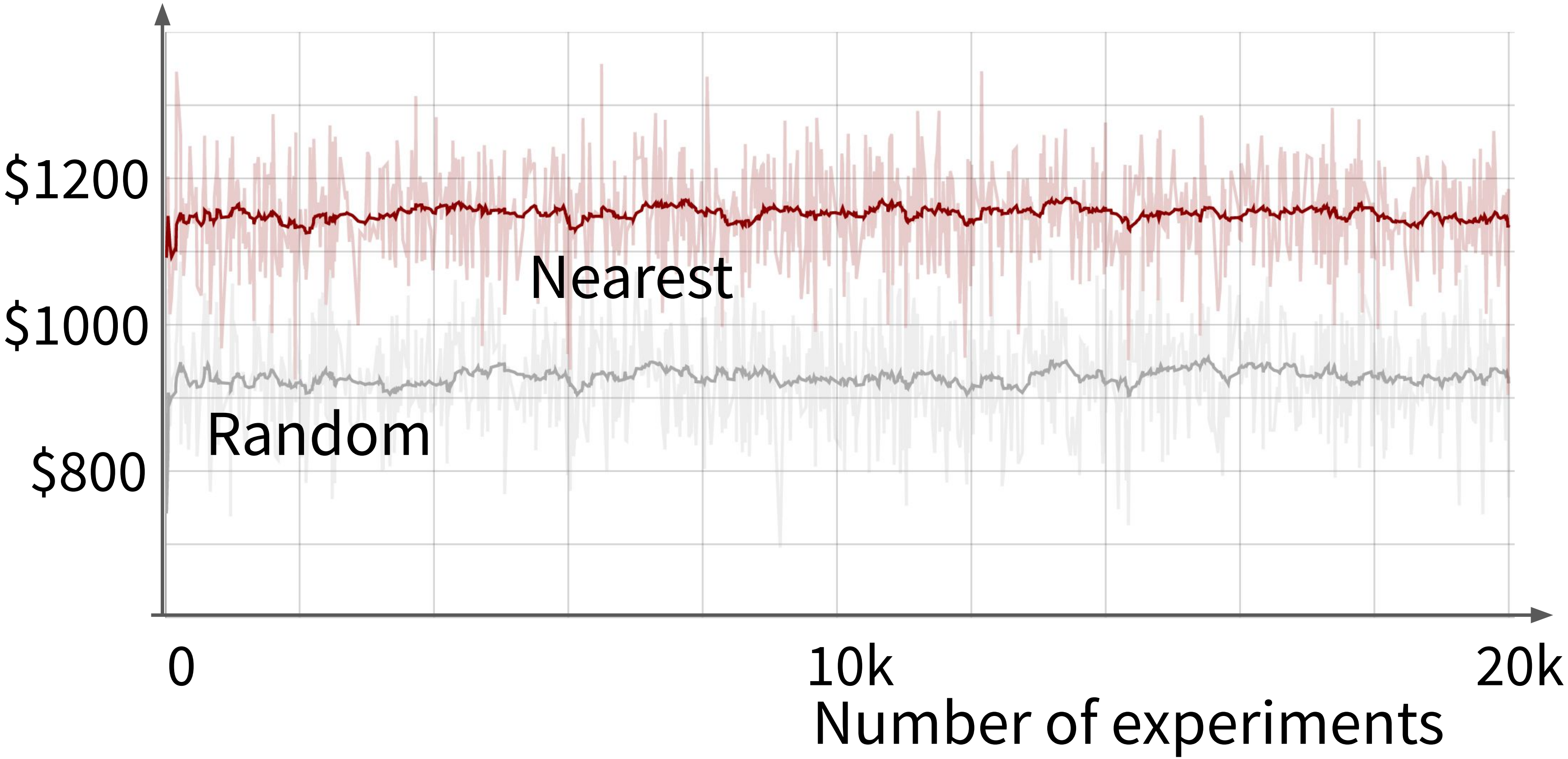
Results



Results

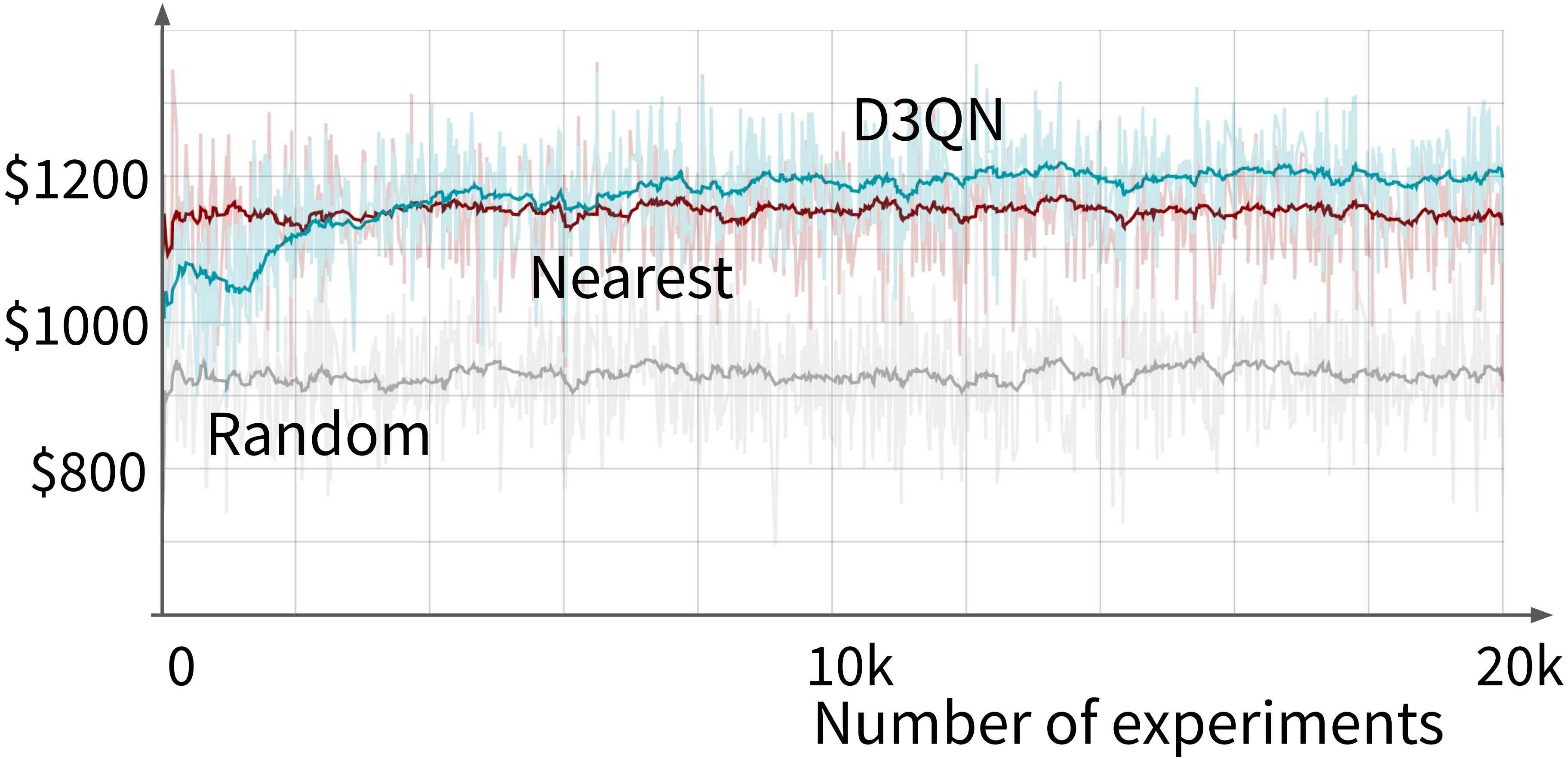


Results

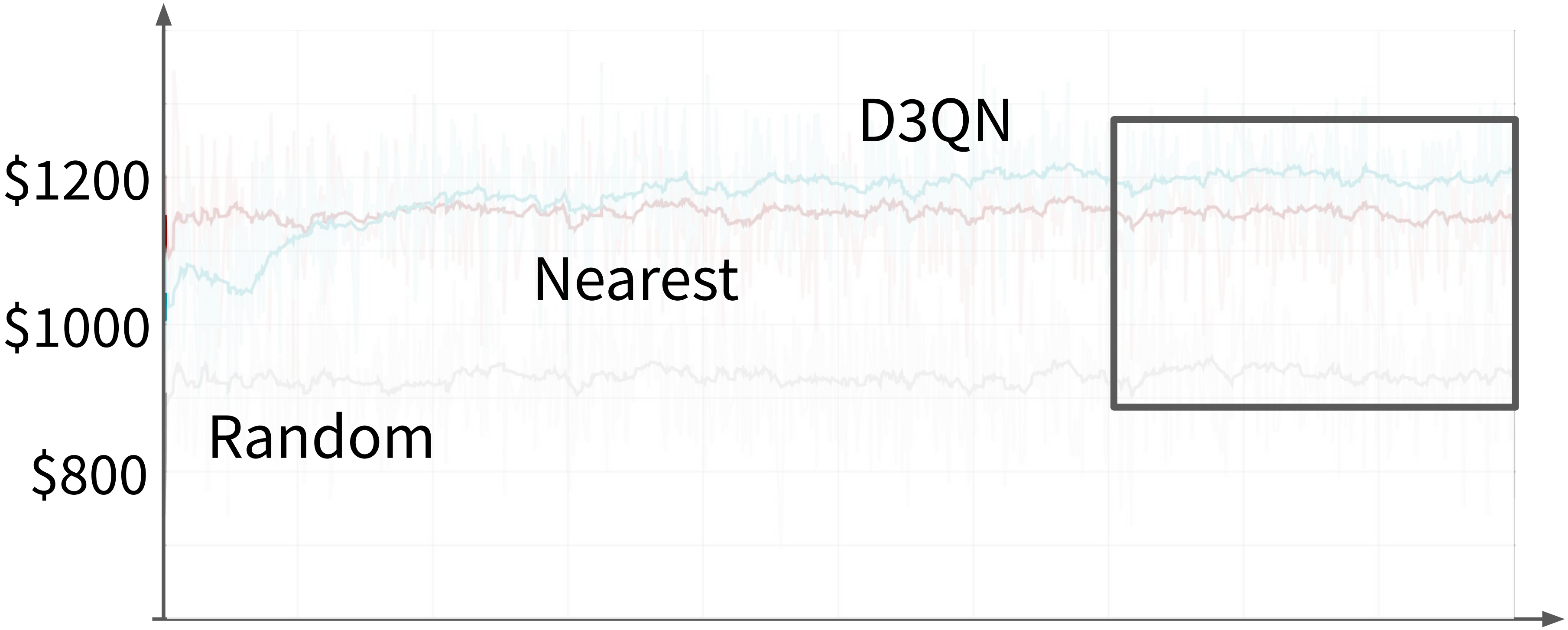


$1200
$1000
$800
Nearest
Random
0
10k
20k
Number of experiments

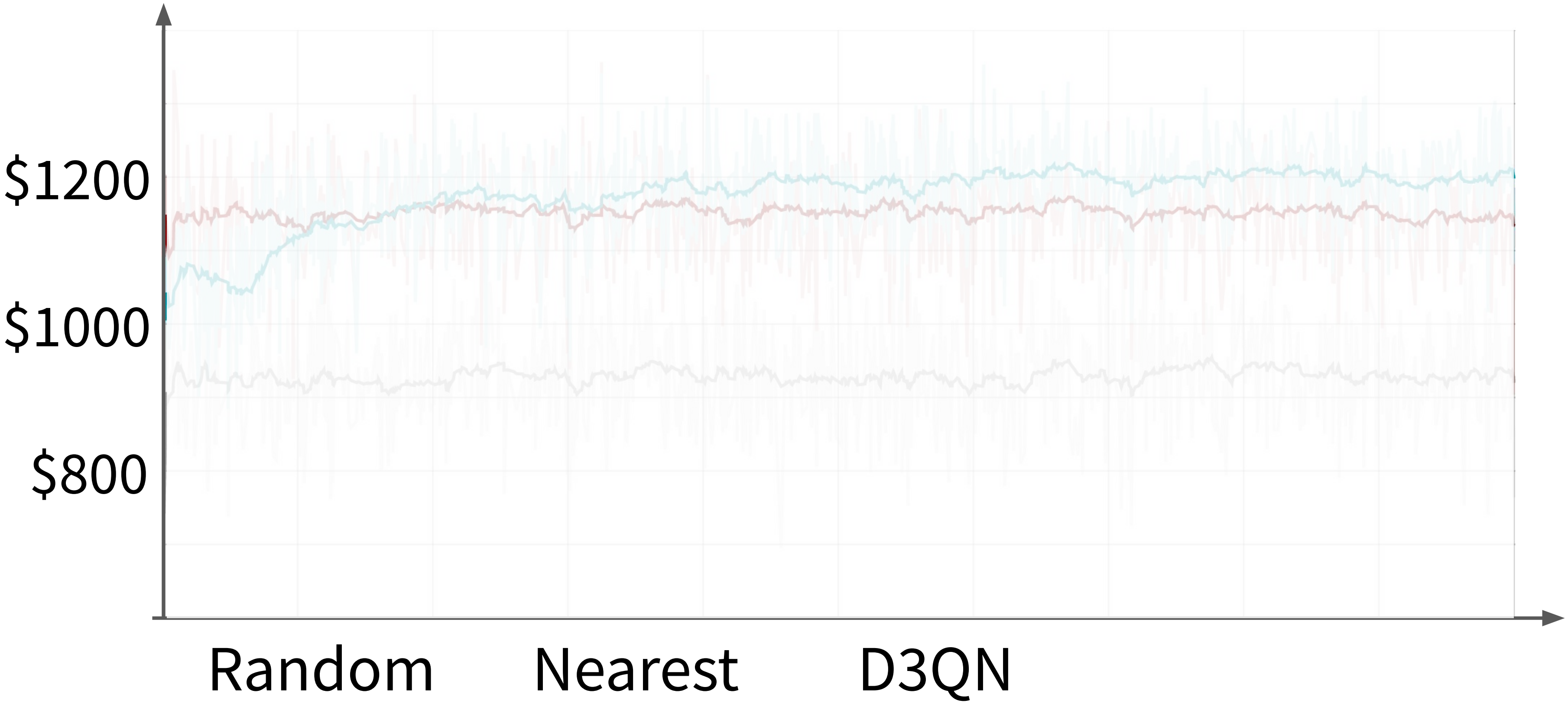
Results


D3QN
Nearest
Random
$1200
$1000
$800
0
10k
20k
Number of experiments

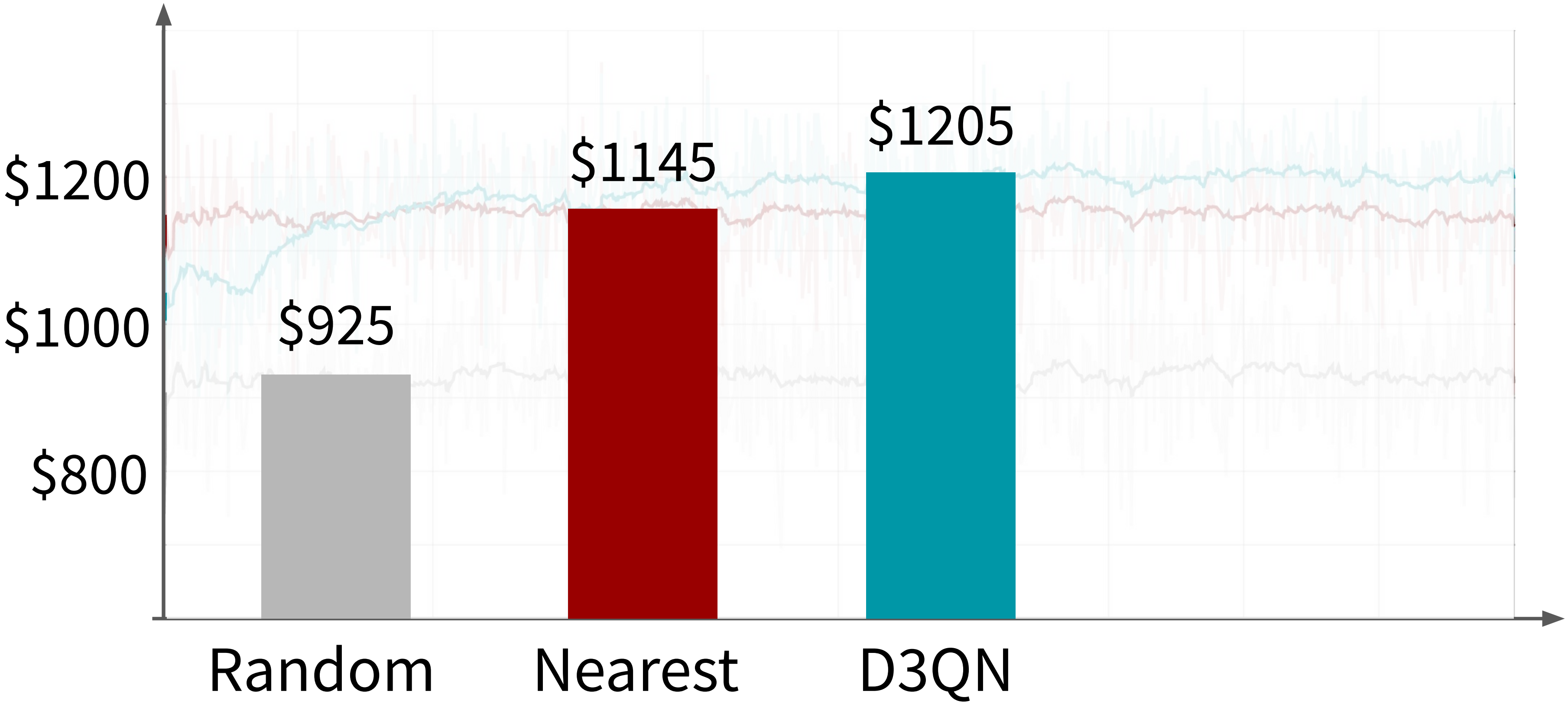
Results



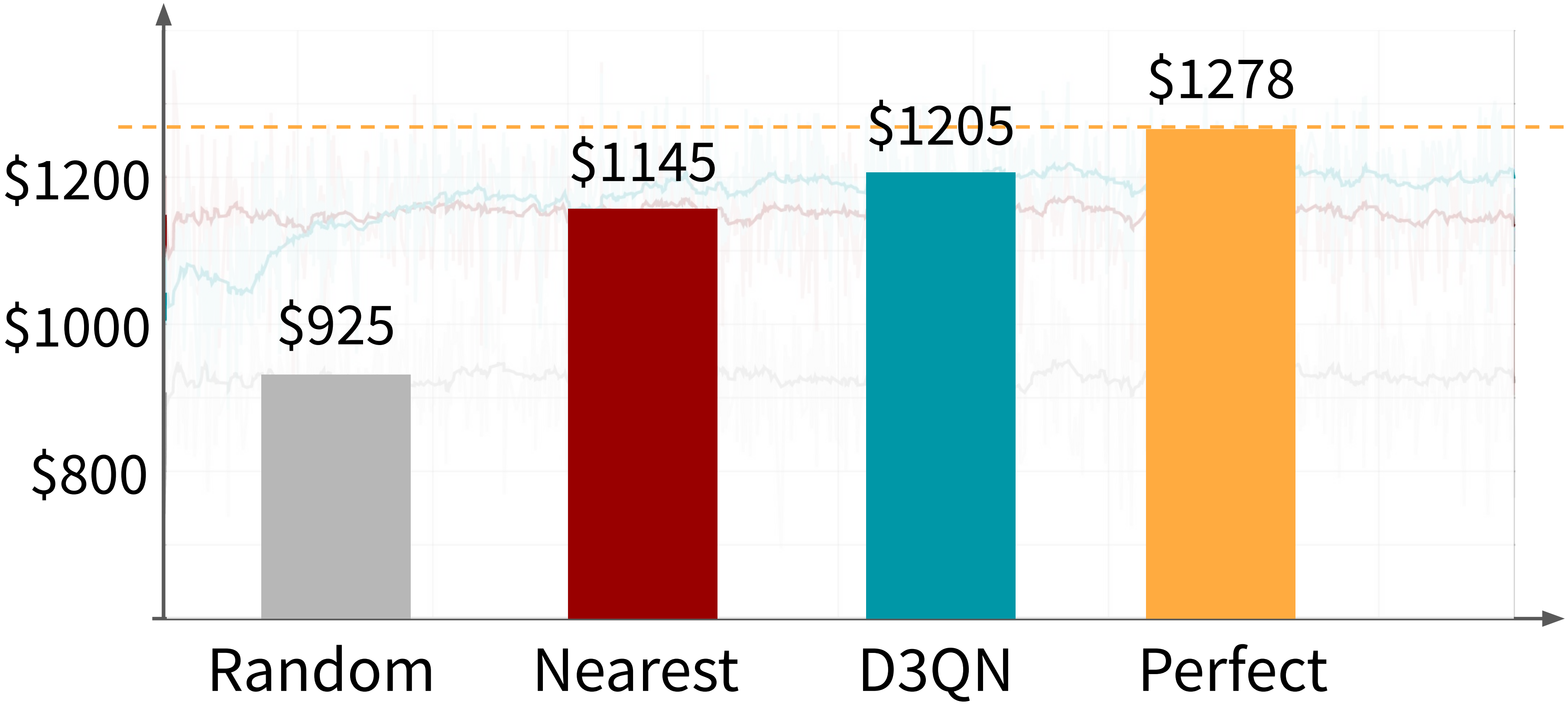
Results



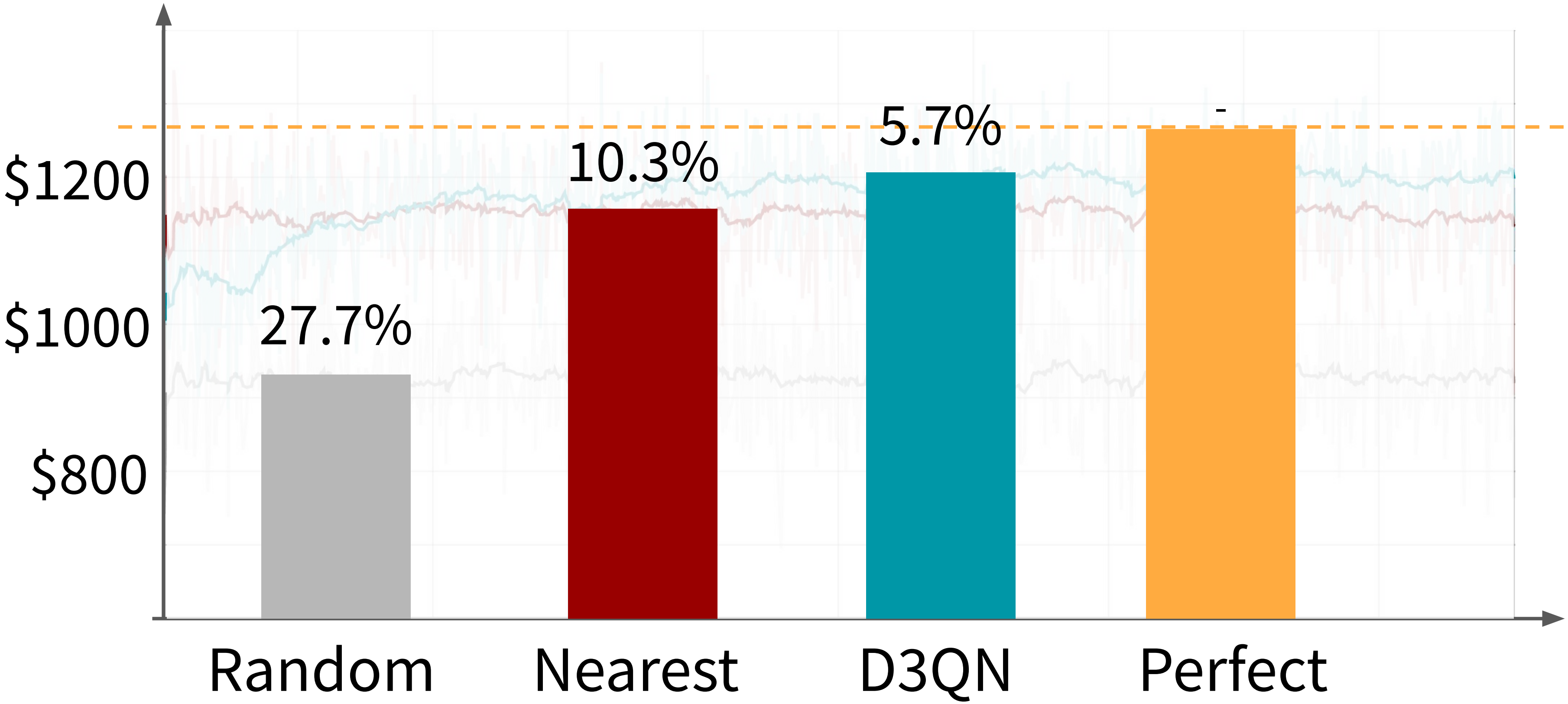
Results



Results


$1200
$1000
$800
$925
$1145
$1205
$1278
Random
Nearest
D3QN
Perfect

Results



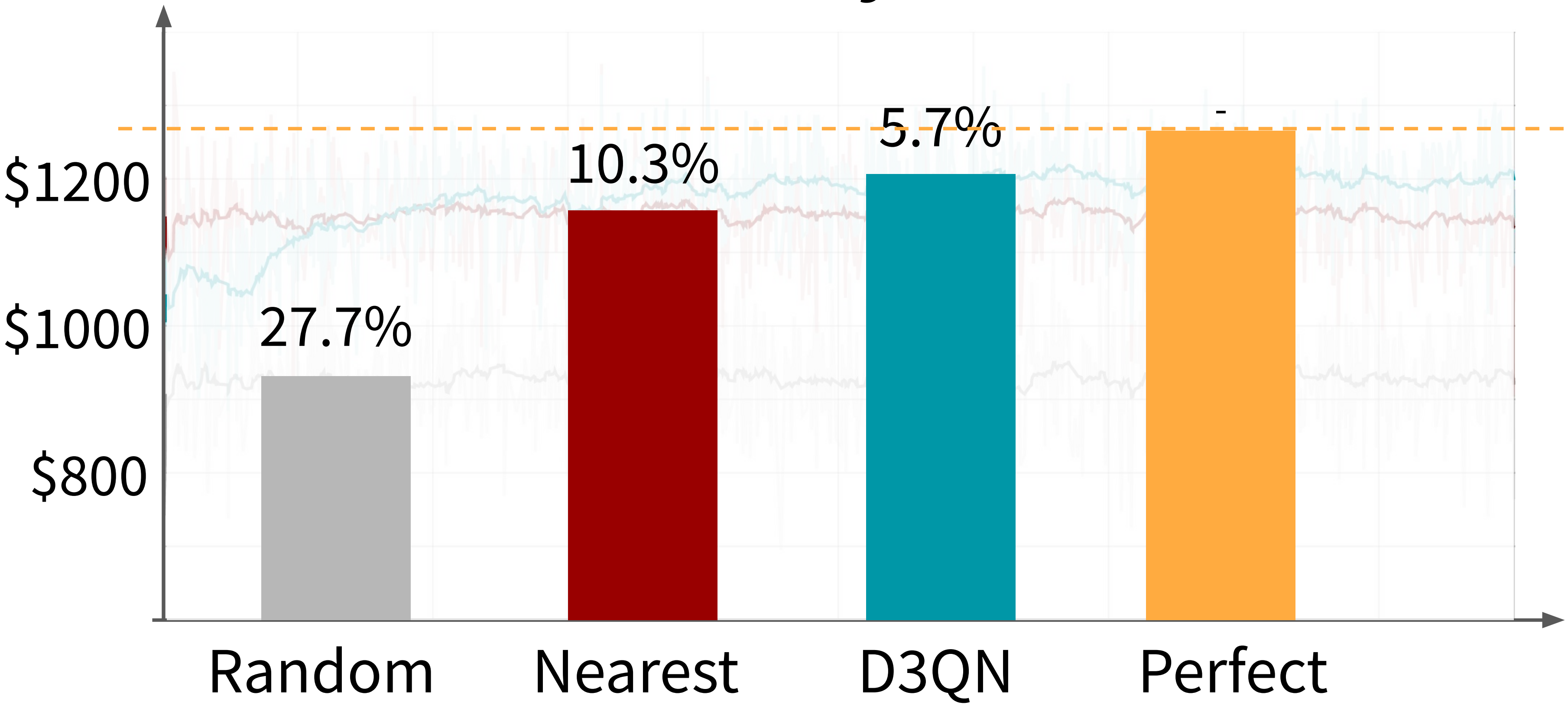
Concluding Remarks

TODO

Bigger instances

Implement “Full Control” agent

Thank you



nicholas.kullman@etu.univ-tours.fr