Strictly CONFIDENTIAL

The end of Airline Revenue Management as we know it?

(Deep) Reinforcement Learning for Revenue Management

AGIFORS Symp 2017, London
October 2017
Speakers: Rodrigo ACUNA AGOST and Thomas FIIG

Credits

RM Expertise + AI Expertise = *(Deep) Reinforcement Learning for Revenue Management*

Amadeus Airlines R&D + Innovation and Research

Nicolas BONDOUX

Quan NGUYEN

Thomas FIIG

Rodrigo ACUNA-AGOST

amadeus

Motivation – limitations of RMS

RMS assumptions

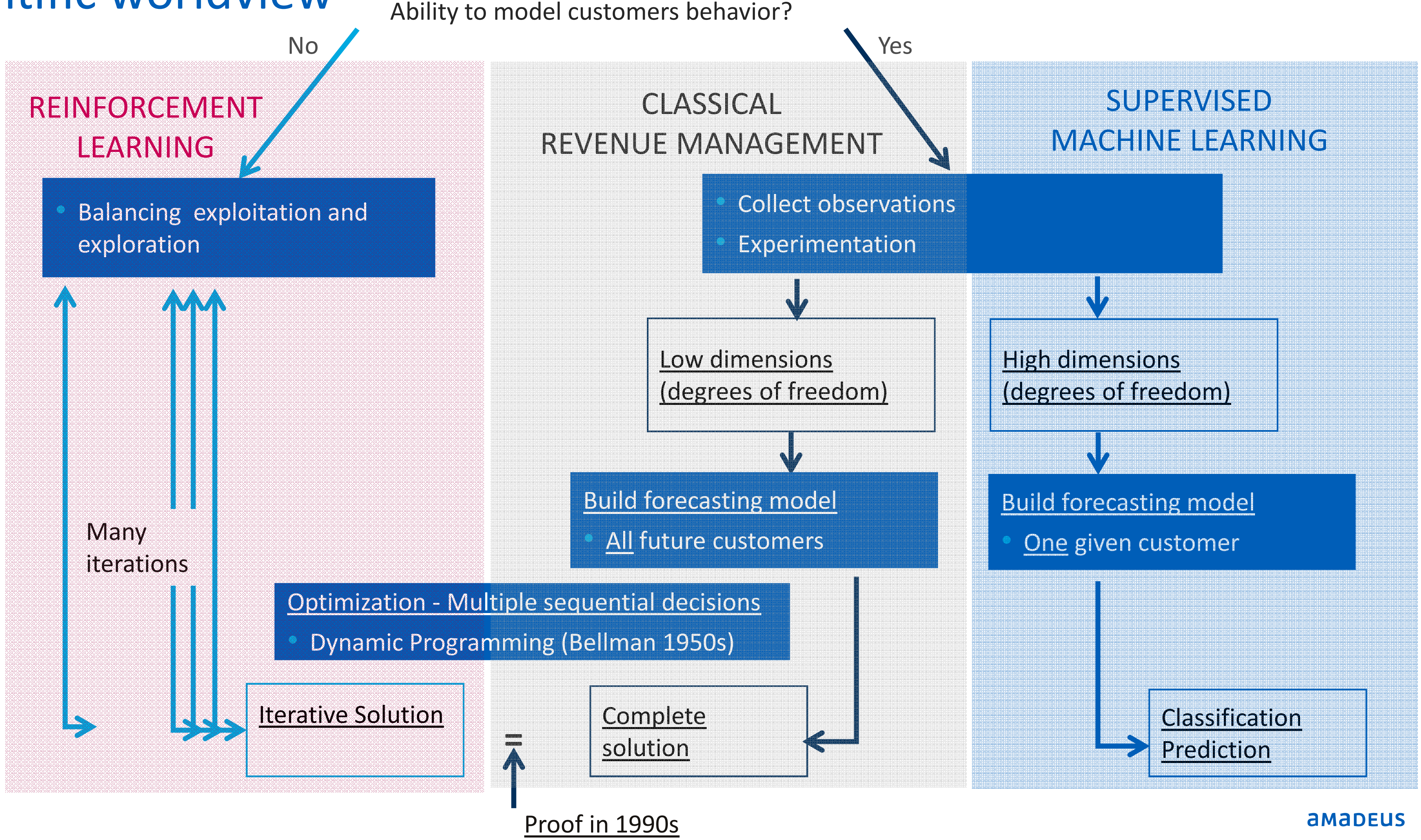
- RMS assumes that the future is accurately described by the past:
  - Issue with change in business environment (new competitors)
  - Issue with shift in demand and willingness to pay
  - Issue with change in customer behavior (for example: arrival pattern)

- RMS assumes that customers are rational:
  - However, customers are irrational, influenced by psychological factors (framing, etc.).
  - There is no model for irrationality.

- RMS assumes monopoly:
  - Competitors offers are accounted for implicitly by how they affect customers behavior. This corresponds to a monopoly seen from RMS

- RMS assumes that a model exist that describes “world”. For general offers this is impossible:
  - Increased complexity of offers (seat + ancillaries)
  - Complex products (flexibility, time to think, etc.), bundles of ancillaries; are difficult to price.
  - Interactions between the prices of ancillaries, bundles, fare families, etc.

Scientific worldview

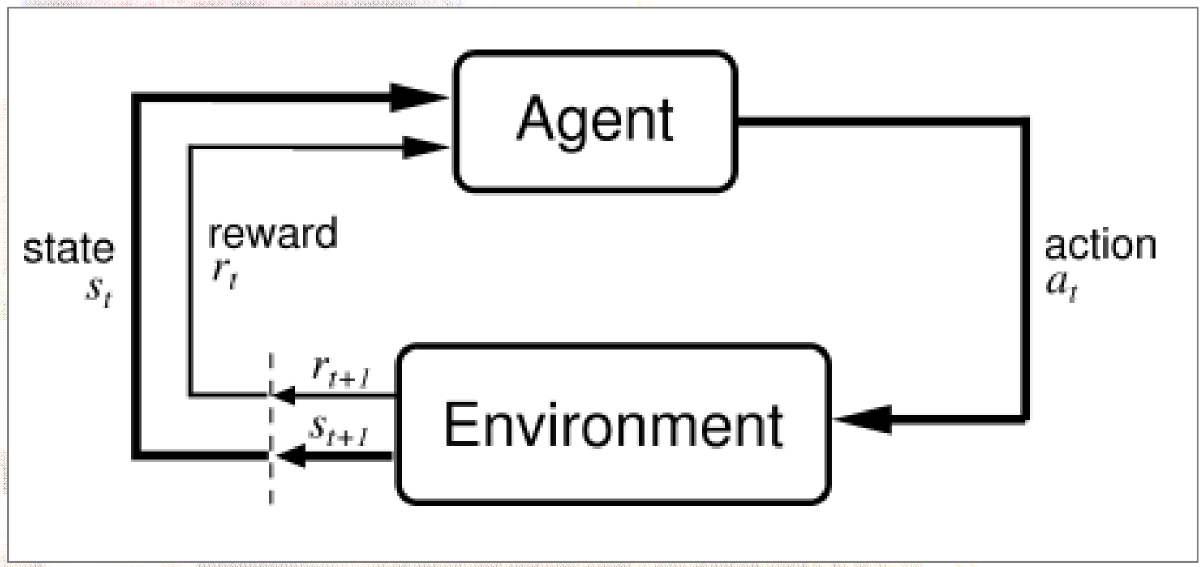


Application of RL

Self-driving cars

How it works?

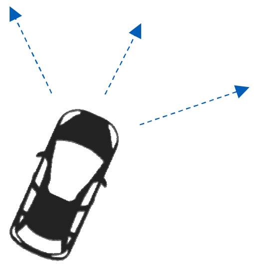
(1) Actions
(2) State
(3) Reward



Actions: { Left, no-change, Right }

State: { Information of Sensors }

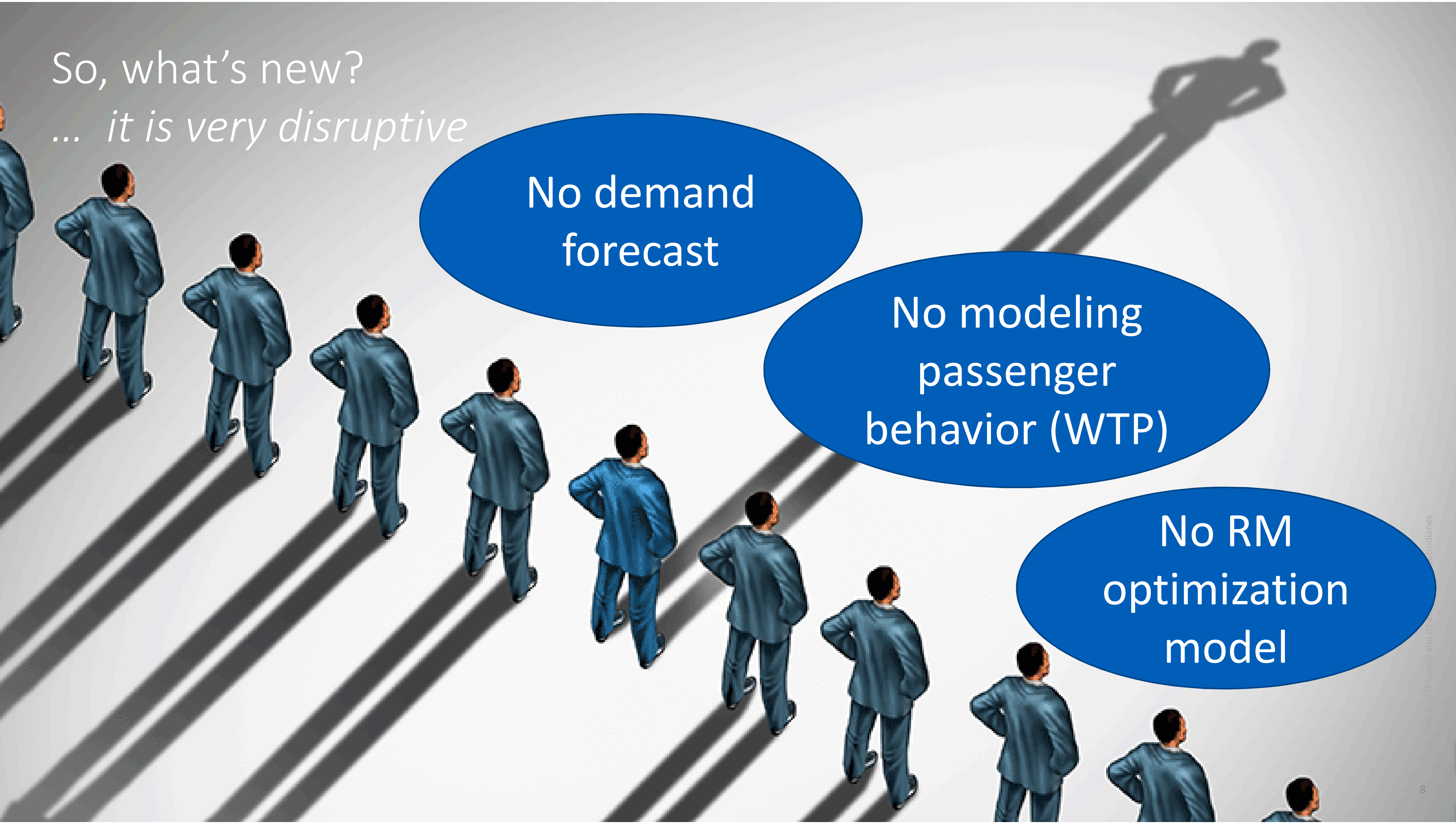
Reward = stay alive as long as possible
(Alive = no crash)



Sensors …

So, what’s new?

… it is very disruptive



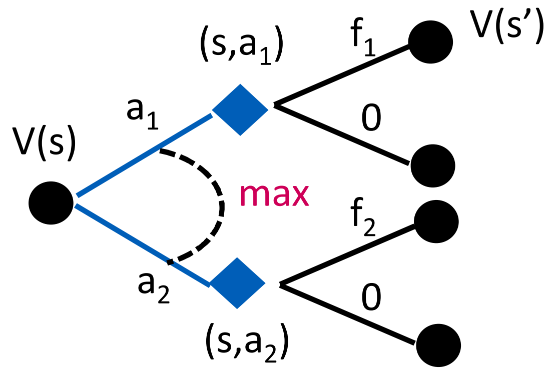
Reinforcement Learning

Mathematical details*)

Bellman (1950s)

$$V(t,x) = Max_f\left[(1-P(f))V(t+1,x) + P(f)\big(f + V(t+1,x-1)\big)\right]$$

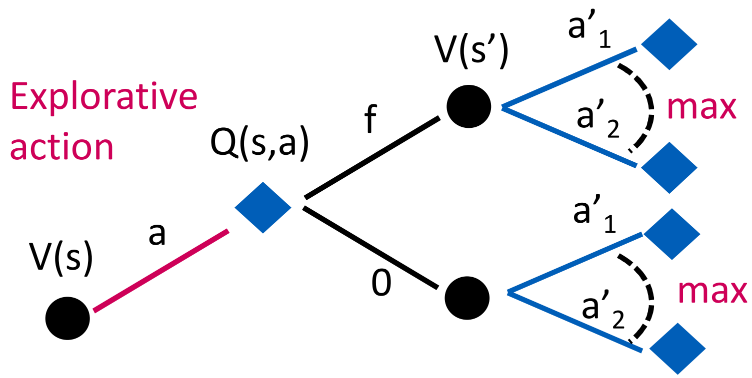
$$V(s) = Max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + V(s')]$$



Q-learning Watkins (1989)

$$Q(s,a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + V(s')]$$

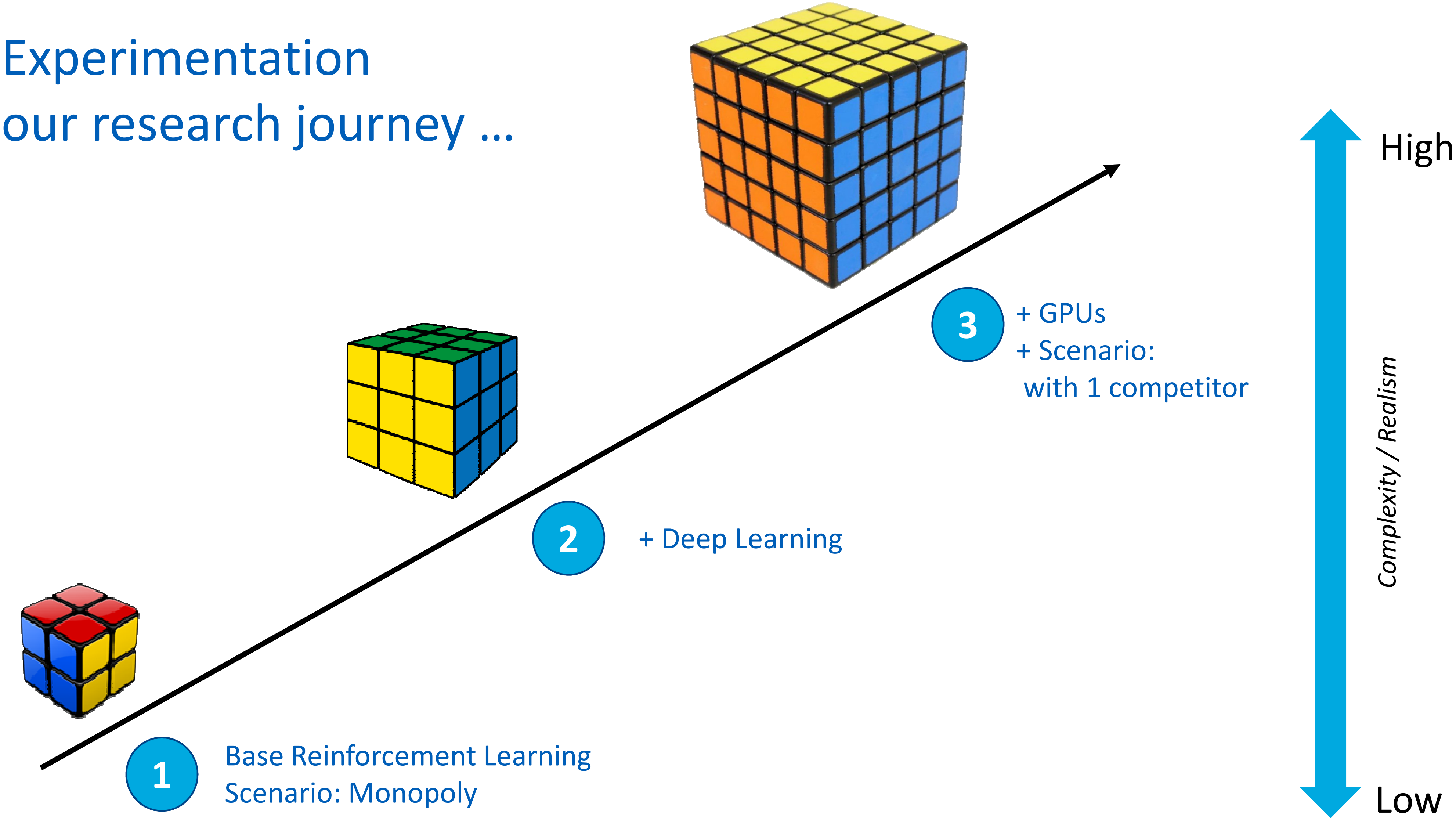
$$\boxed{V(s) = Max_a\, Q(s,a)}$$



$$Q(s,a) \leftarrow [1-\alpha]Q(s,a) + \alpha[f + V(s')\,]$$

*) Reinforcement Learning, Sutton, Brato, 1998

Experimentation
our research journey …



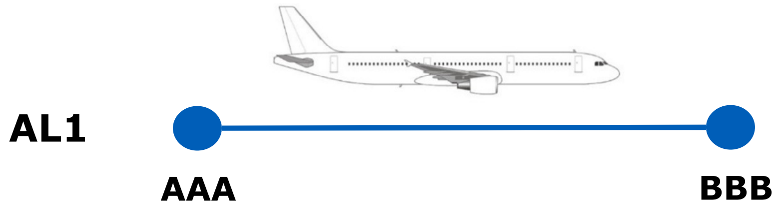
Experiments 1

Base Reinforcement Learning in a Monopoly

Simulation set-up
Cap = 10
DCP = 20
Fare classes = 3
Fenceless fare structure

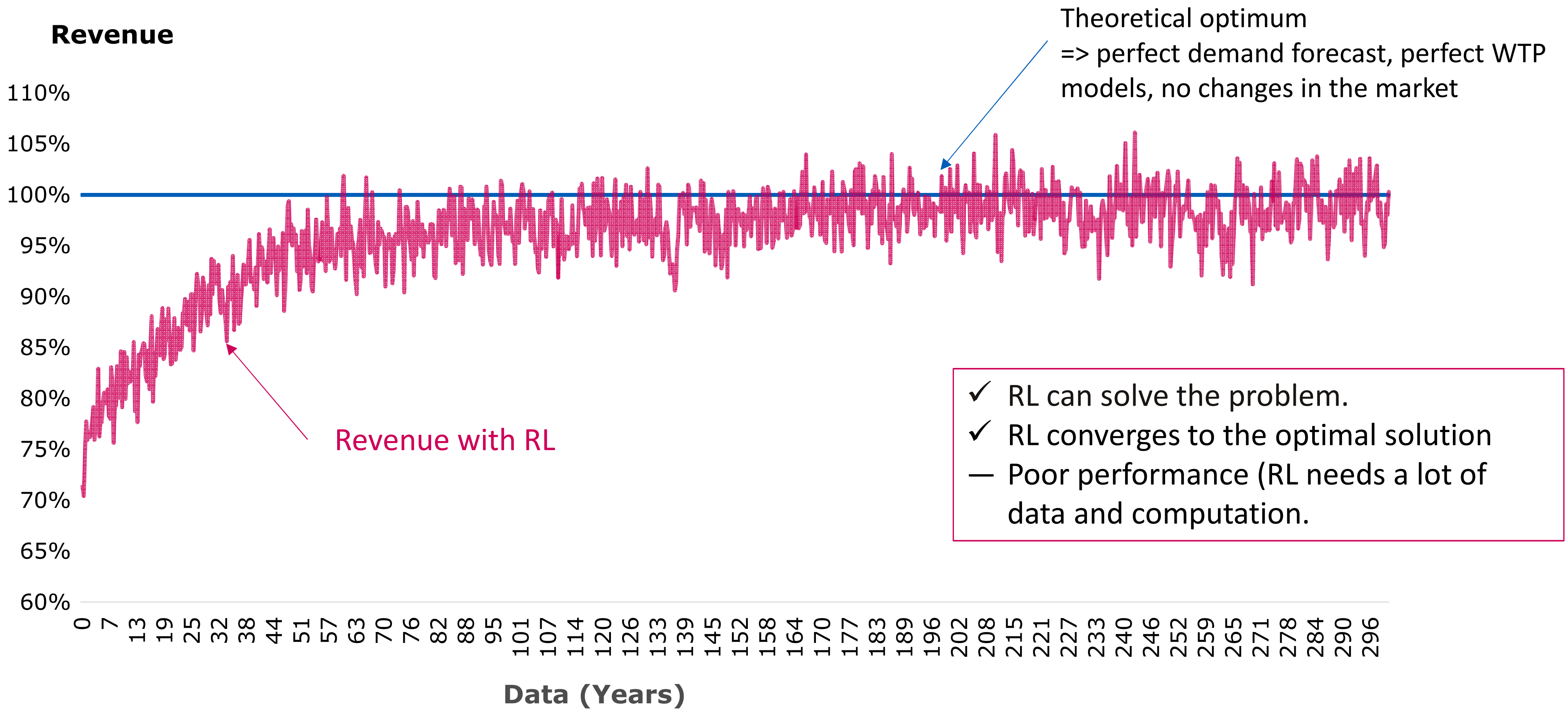
RMS basecase
- AL1: Dynamic Programming
- Two customer segments with different frat5
- Forecaster = Q-forecasting

Reinforcement Learning
- No Forecaster or Optimizer
- AL1: Q –learning
- State (t,x)
- Action: f1,f2,f3, closed



Experiments 1

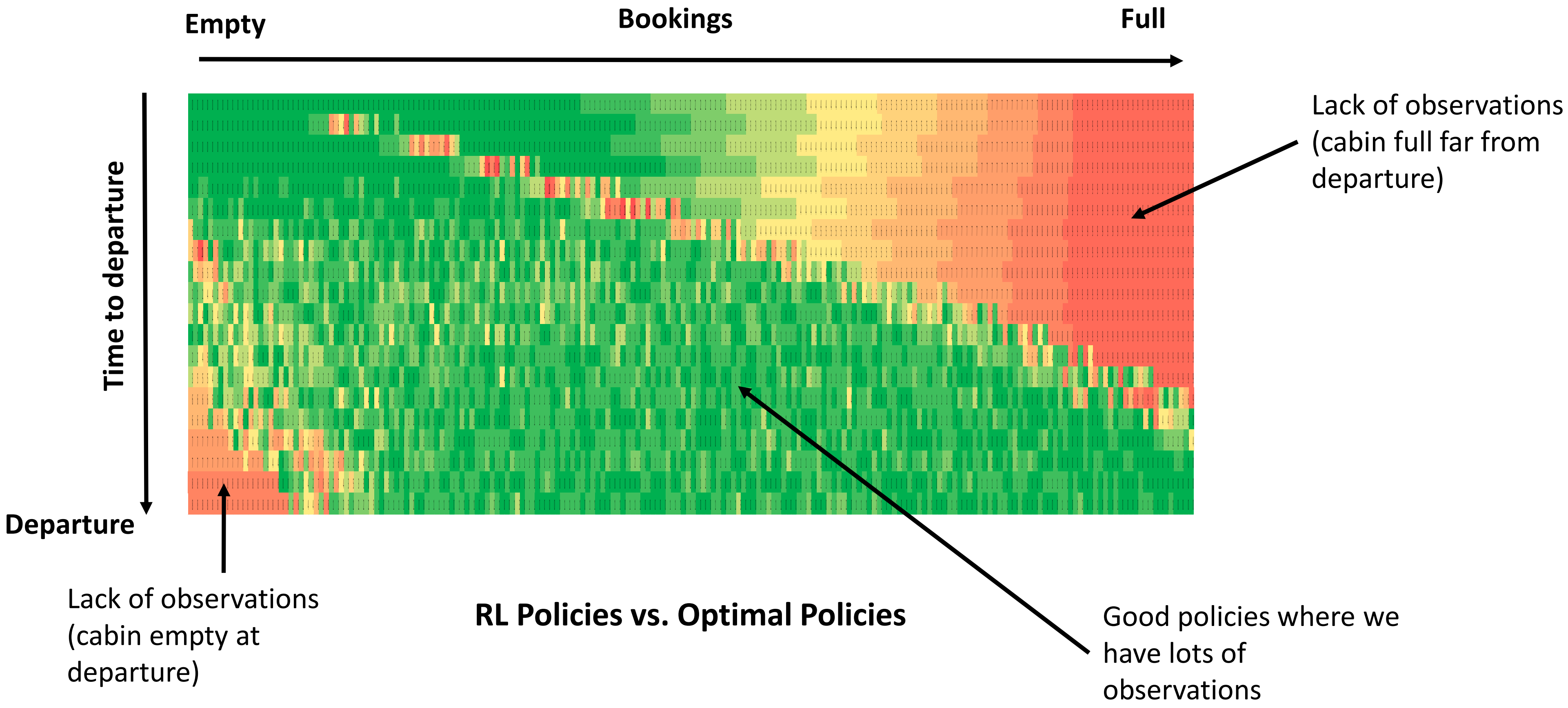
Base Reinforcement Learning in a Monopoly



- ✓ RL can solve the problem.
- ✓ RL converges to the optimal solution
- — Poor performance (RL needs a lot of data and computation.

Experiments 1

Base Reinforcement Learning in a Monopoly

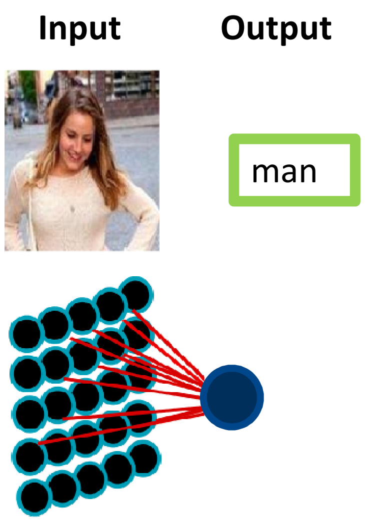


RL Policies vs. Optimal Policies

Deep Reinforcement Learning 2

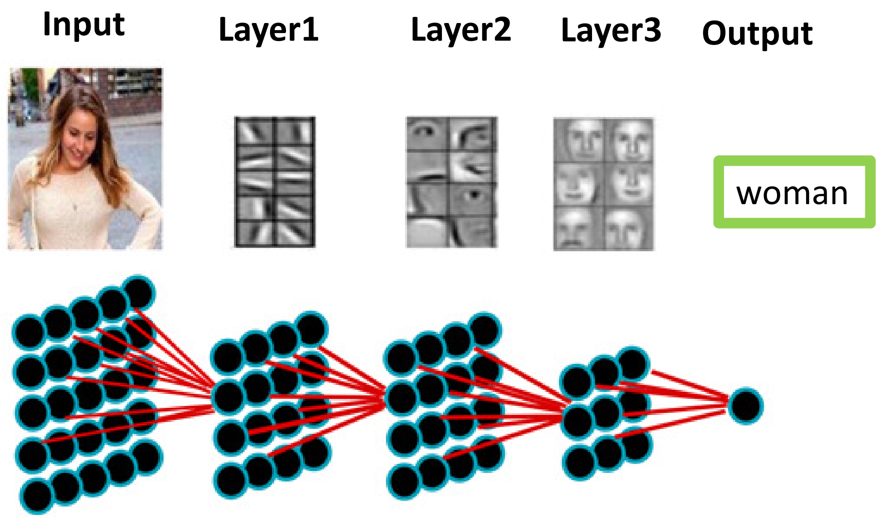
Deep Neural Network

Classical Artificial Neural Networks



Accuracy:
worse than other ML methods

Deep Neural Network



Accuracy:
better than other ML methods

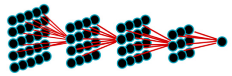
Deep Reinforcement Learning 2

Deep Neural Network as function approximation

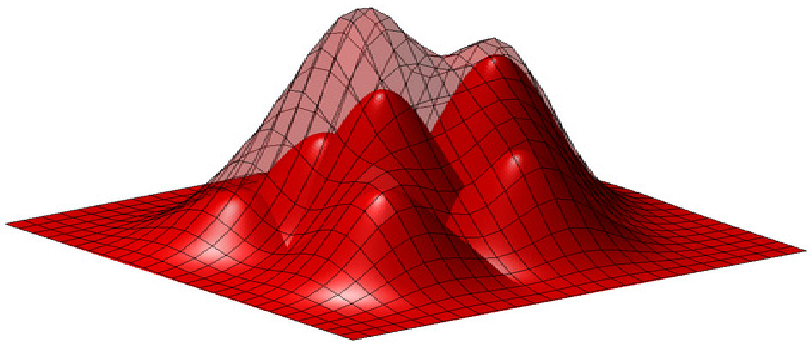
What is **function approximation**?

$$(s, a) \longrightarrow Q(s, a)$$

$$(s, a) \xrightarrow{\theta(x, y) = \alpha x + \beta y} Q(s, a, \theta) = \alpha s + \beta a$$



$(s, a) \longrightarrow Q(s, a,$  $)$

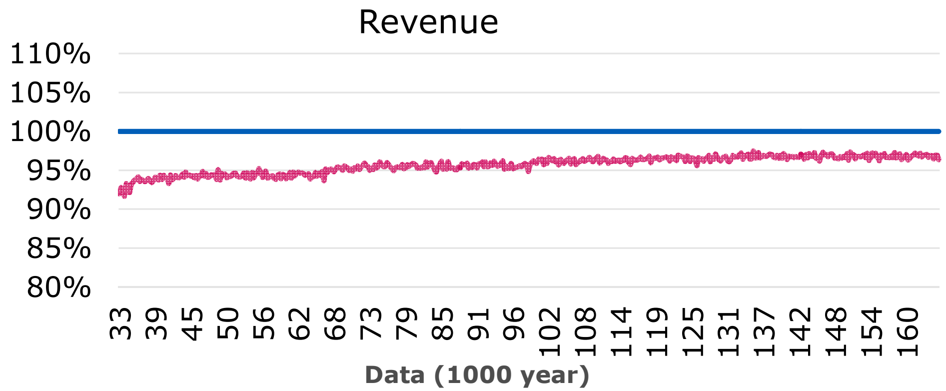


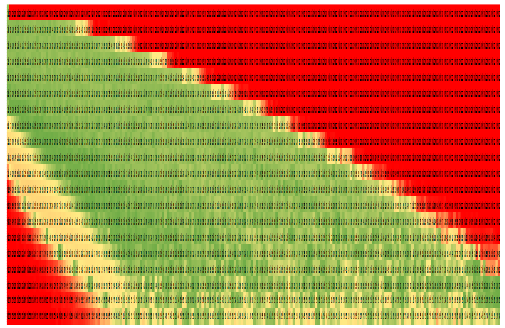
Experiments 2

+ Deep Learning

Normal RL

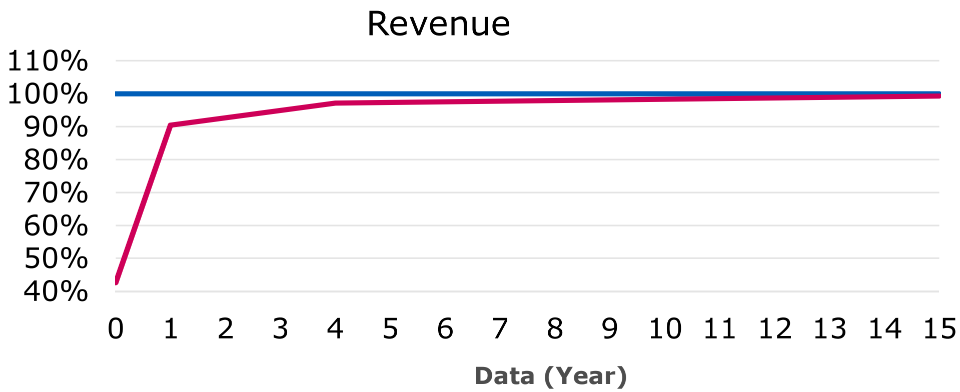
135kY of flight data ~ **97%** of opt. rev

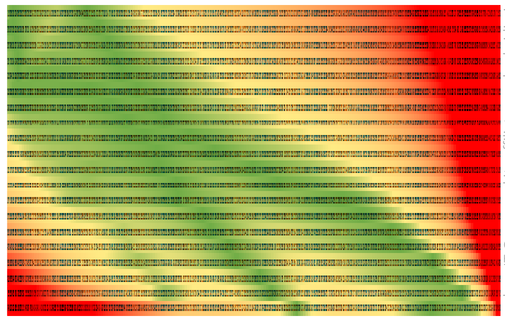




Deep RL

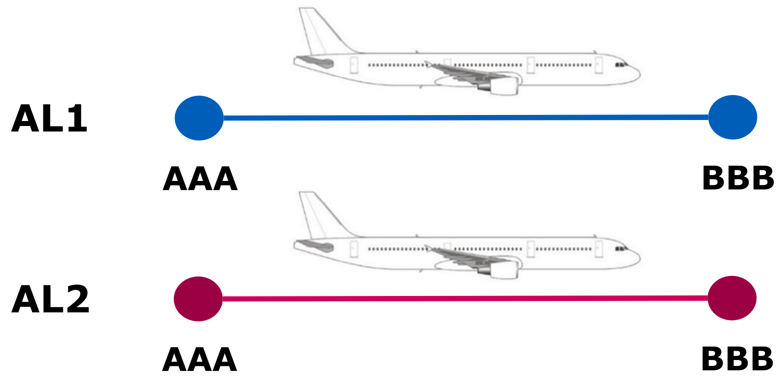
5-15Y of flight data ~ **99%** of opt. rev





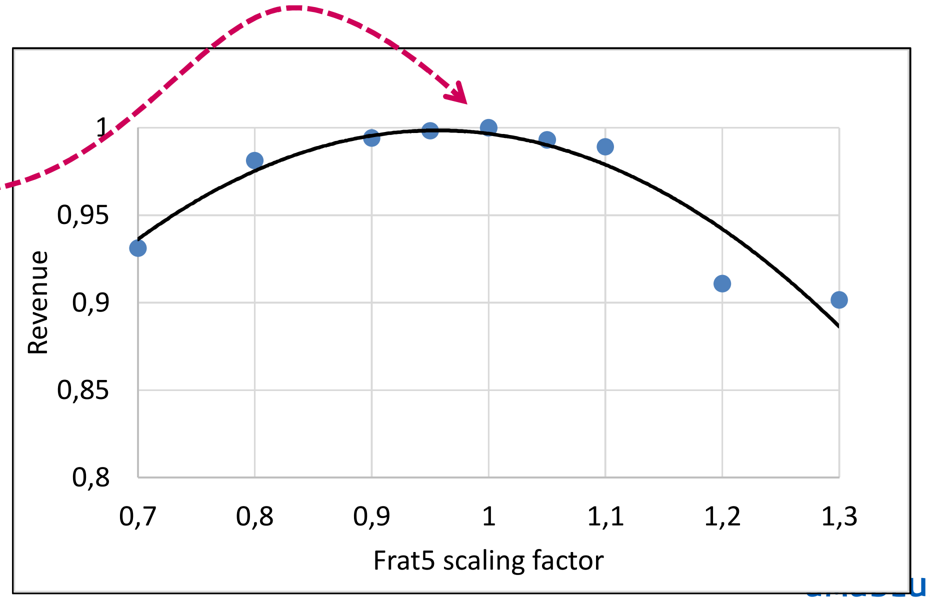
Experiments 3

Reinforcement Learning in Duopoly



Simulation set-up
Cap =50
DCP=20
Fare classes = 10
Fenceless fare structure

RMS basecase
- AL1: Dynamic Programming
- A2: AT80
- Two customer segments with different frat5
- Estimated frat5 (optimal revenue)
- Forecaster = Q-forecasting

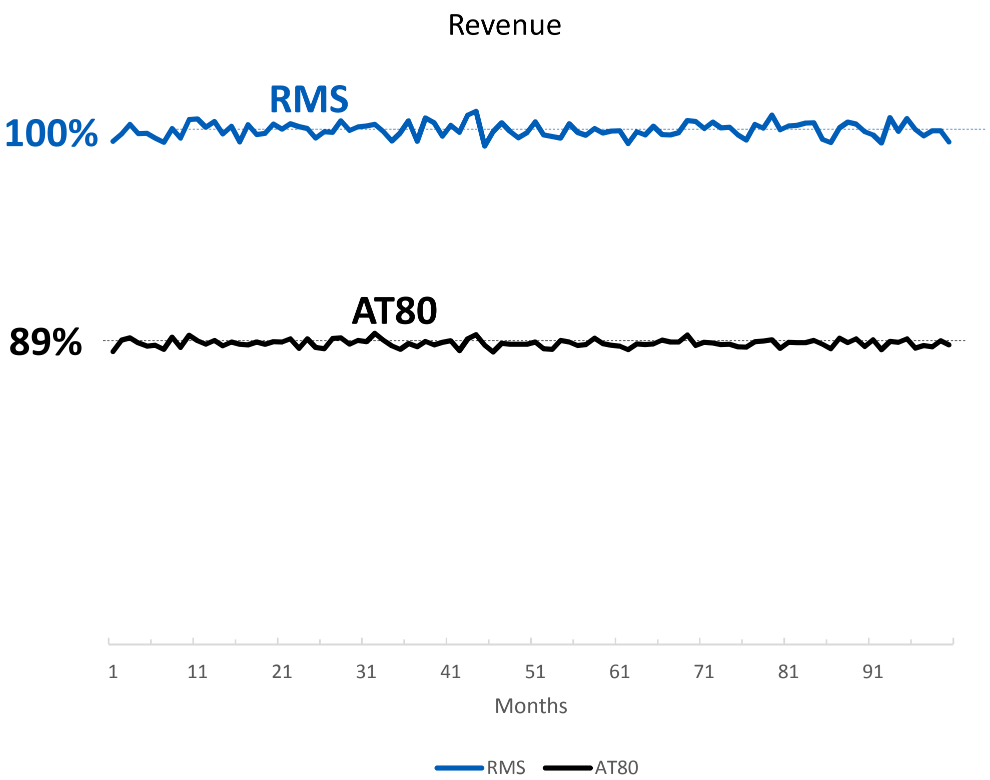


Reinforcement Learning
- No Forecaster or Optimizer
- AL1: Deep RL
- State (t,x)
- Action: f1,f2,f3,…,f10, closed

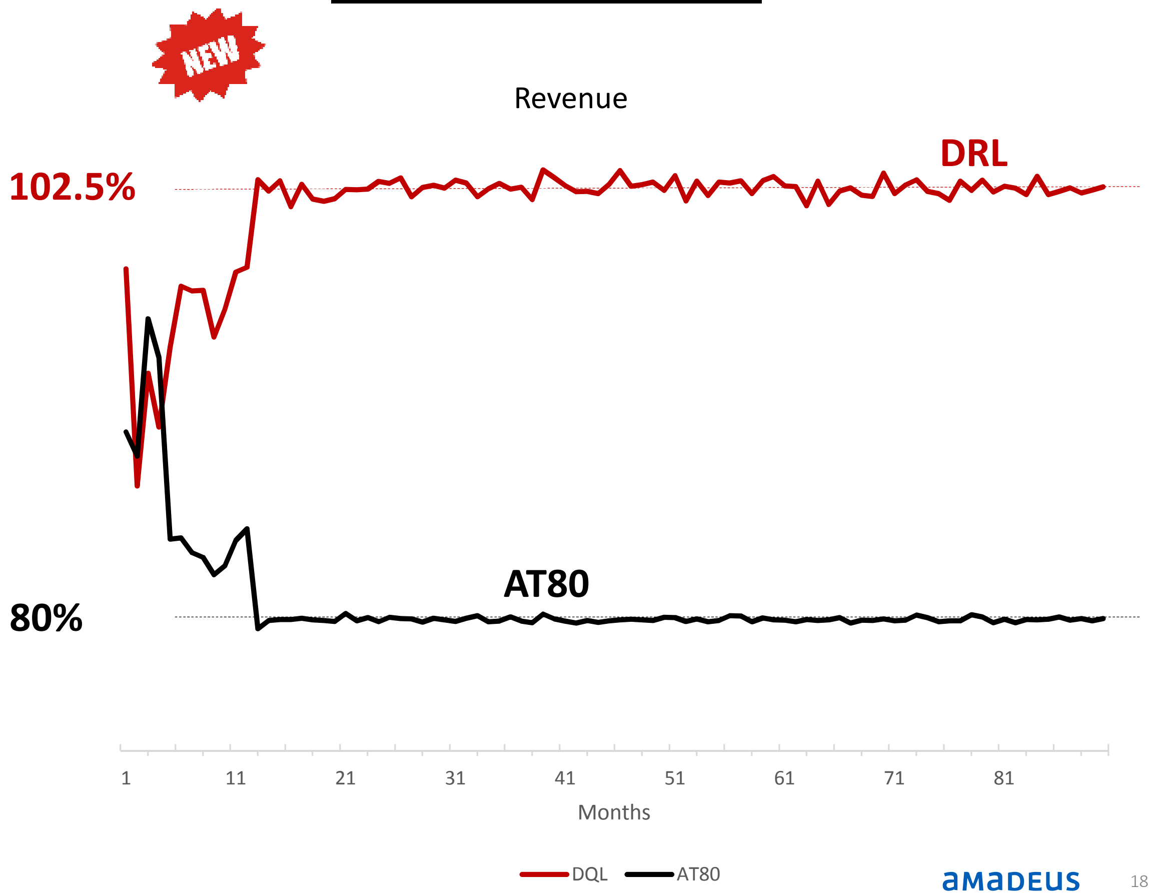
Experiments 3

One Competitor + GRUs

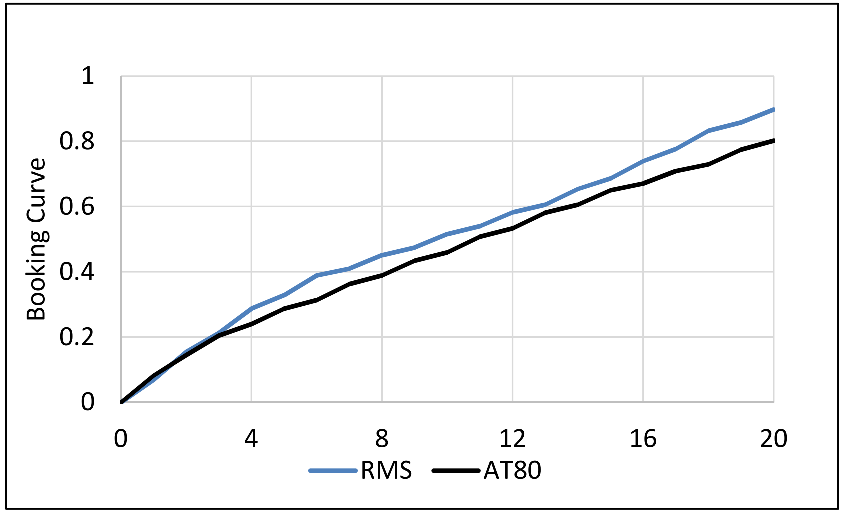
RMS vs Competitor

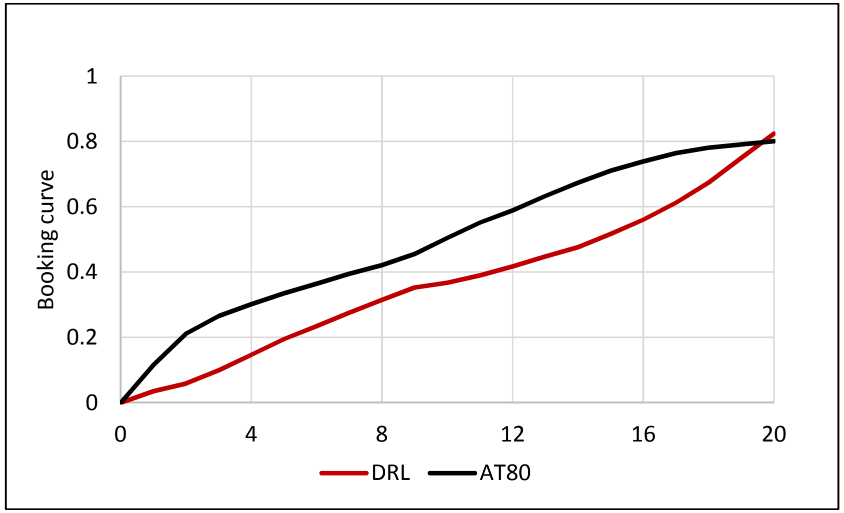


DRL vs Competitor



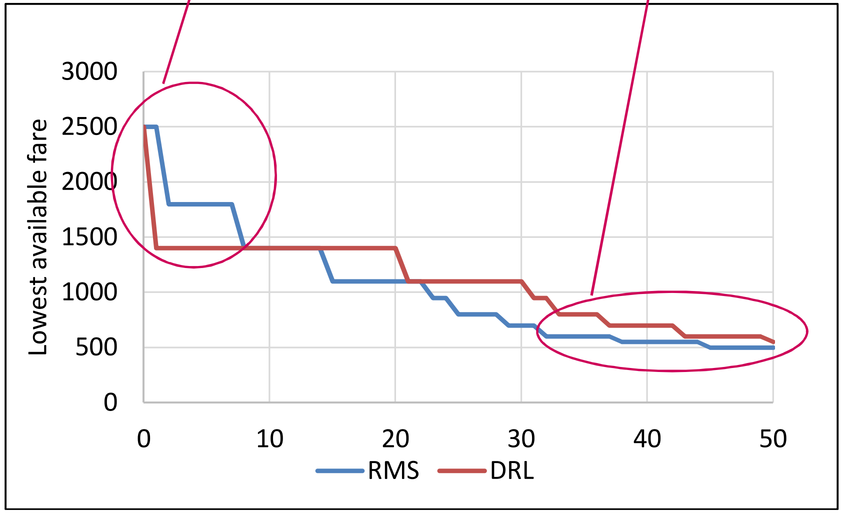
Why is RL better? 3





DRL open classes close to departure

DRL closes classes far from departure

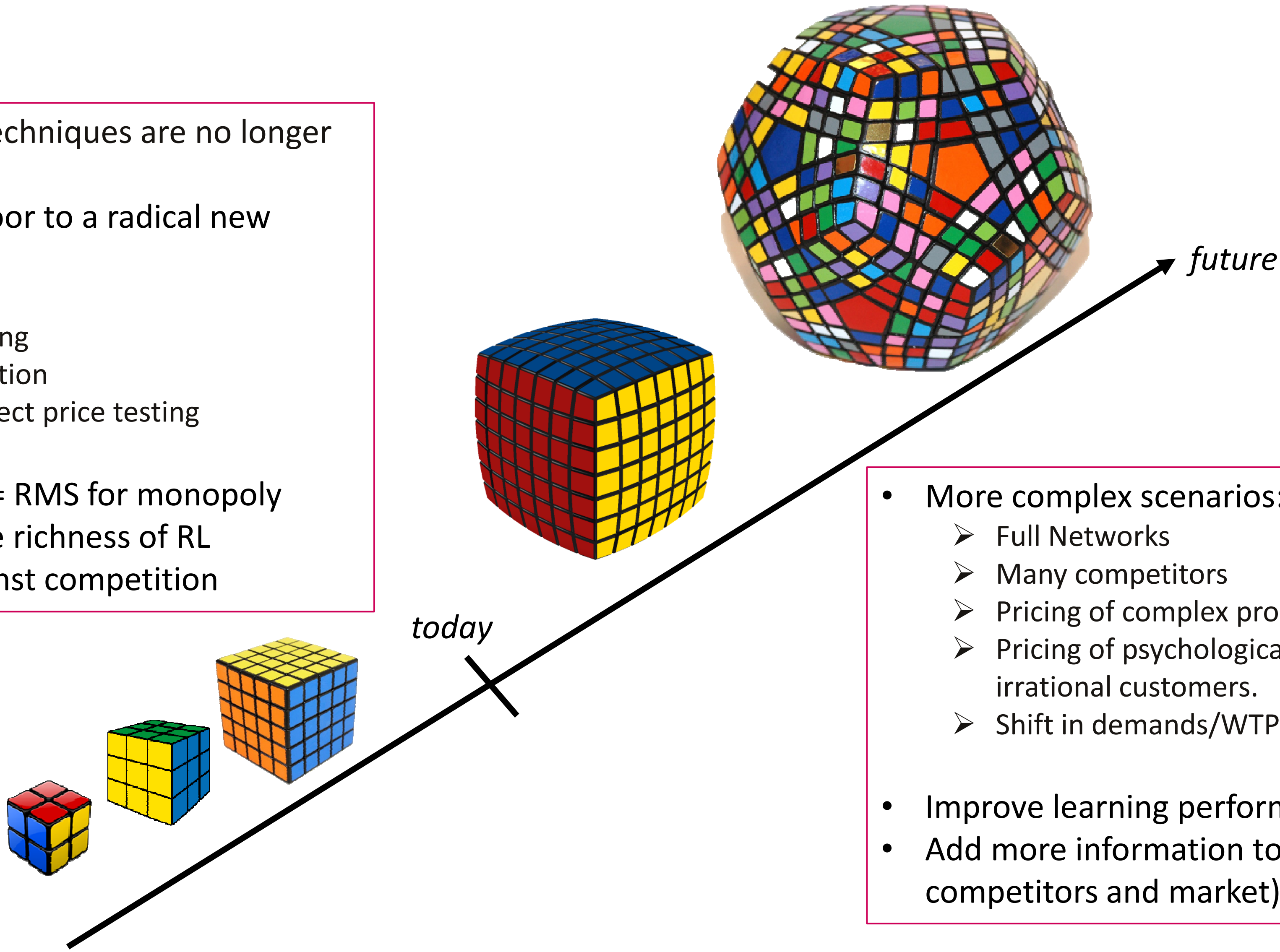


- Remember RMS were optimal
- DRL produces higher revenue by understanding the competitive game and swamping the competitors with low yield passengers.

Conclusion

- Classical RMS techniques are no longer sufficient.
- RL opens the door to a radical new approach:
  - ➢ Model free
  - ➢ No forecasting
  - ➢ No optimization
  - ➢ Leans by direct price testing

- Shown that RL = RMS for monopoly
- We discover the richness of RL
- Beats RMS against competition



- More complex scenarios:
  - ➢ Full Networks
  - ➢ Many competitors
  - ➢ Pricing of complex product
  - ➢ Pricing of psychological factors - irrational customers.
  - ➢ Shift in demands/WTP

- Improve learning performance
- Add more information to the state (eg., competitors and market)