Transparent and Interpretable Control by Reinforcement Learning Agents

An Empirical Study on Linear Function Approximators for Isolated Intersections

分かりやすく解釈可能な強化学習エージェント 単独交差点制御のための線形関数適用の実証実験

東京大学 生産技術研究所 大口研究室（交通制御工学）

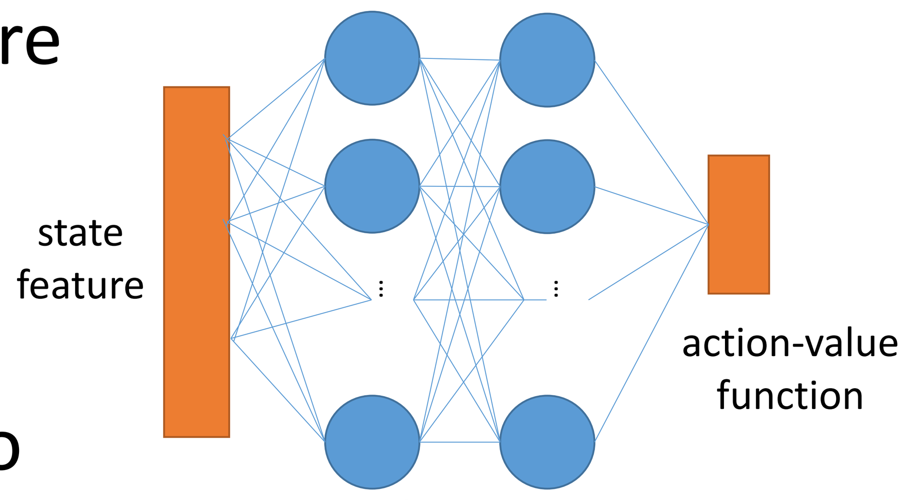
http://www.transport.iis.u-tokyo.ac.jp/

Sahachaisaree S., Han T., Oguchi T.

1) Background

- Reinforcement learning (RL) is a machine learning paradigm used to solve sequential decision-making problems; e.g., urban traffic signal control.

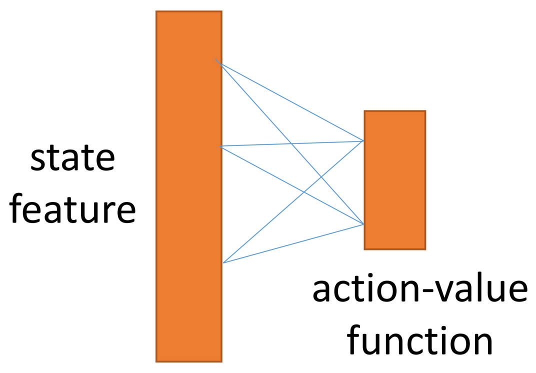
- Value-based RL agents are mostly represented by complex function approximators (FA's).



- Their parameters are too numerous and cannot be interpreted.

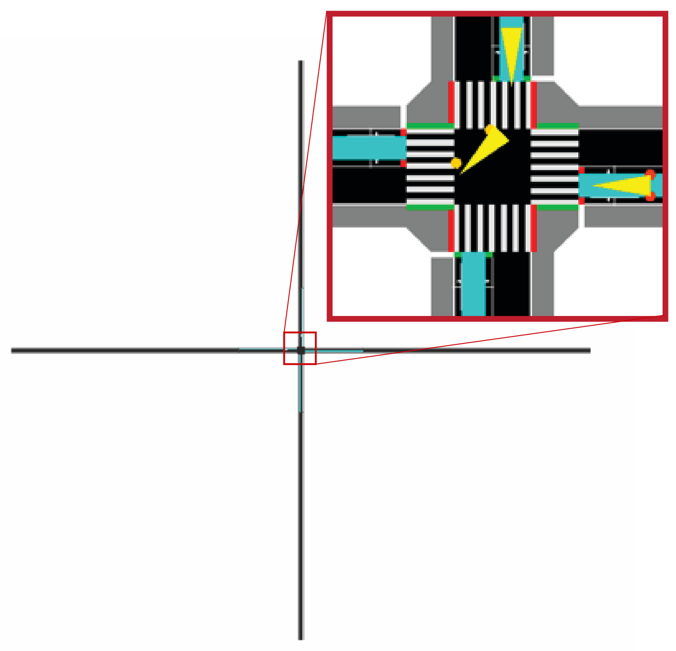
Research question ?

Could RL agents that are using the simplest linear FA's reach the same performance level of those using more complex FA's?



2) Control problem setup

- Following the research question, a control problem is setup for numerical experiments.
  - Intersection of 2 two-lane roads with no turning bays
  - Total vehicle arrival rate is at a moderate level, far from the saturation level.
  - 100-m detection range from the stop line for each approach lane



- Based on the simulated traffic at intersection, a Markov decision problem (sequential decision-making problem) is formulated.
  - State features : lane queue length *[veh]* and one-hot encoded active stage *[-]*
  - Reward function : sum of queue lengths *[veh]*
  - Discrete actions : N-S green or W-E green with action interval of 10 seconds in the simulation

3) Proposed techniques and experiments design

I. Initialise the linear FA with **informed initial weights**

II. Use a **least-squares solution** instead of stochastic gradient descent (SGD)

$$
\mathbf{w}_{\pi_0}^{info} =
\begin{array}{c c}
\phi_1, \phi_2 & \\
\begin{bmatrix} +1 & -1 \\ -1 & +1 \\ -1 & +1 \\ +1 & -1 \\ -1 & +1 \\ +1 & -1 \end{bmatrix} &
\begin{array}{l} x_{\phi_1} \\ x_{\phi_2} \\ x_{W,queue} \\ x_{N,queue} \\ x_{E,queue} \\ x_{S,queue} \end{array}
\end{array}
$$

Informed initial weights

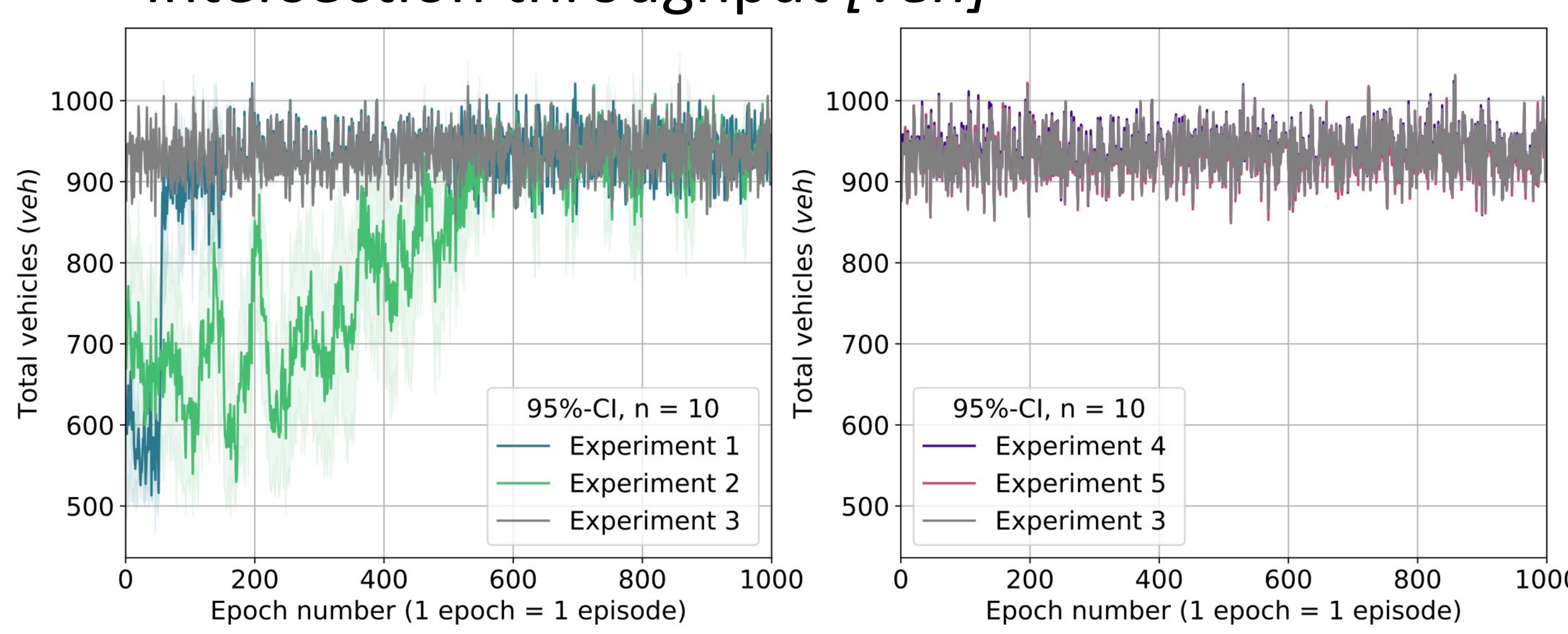
- An artificial neural network (ANN) with one 32-neuron hidden layer is used as the more complex FA as a counterpart to linear FA.

# exp	FA	Informed initial weights	Learning method
1	ANN	Not possible	SGD
2	Linear	No	SGD
3	Linear	Yes	SGD
4	Linear	Yes	Least-squares
5	Linear	Yes	No learning

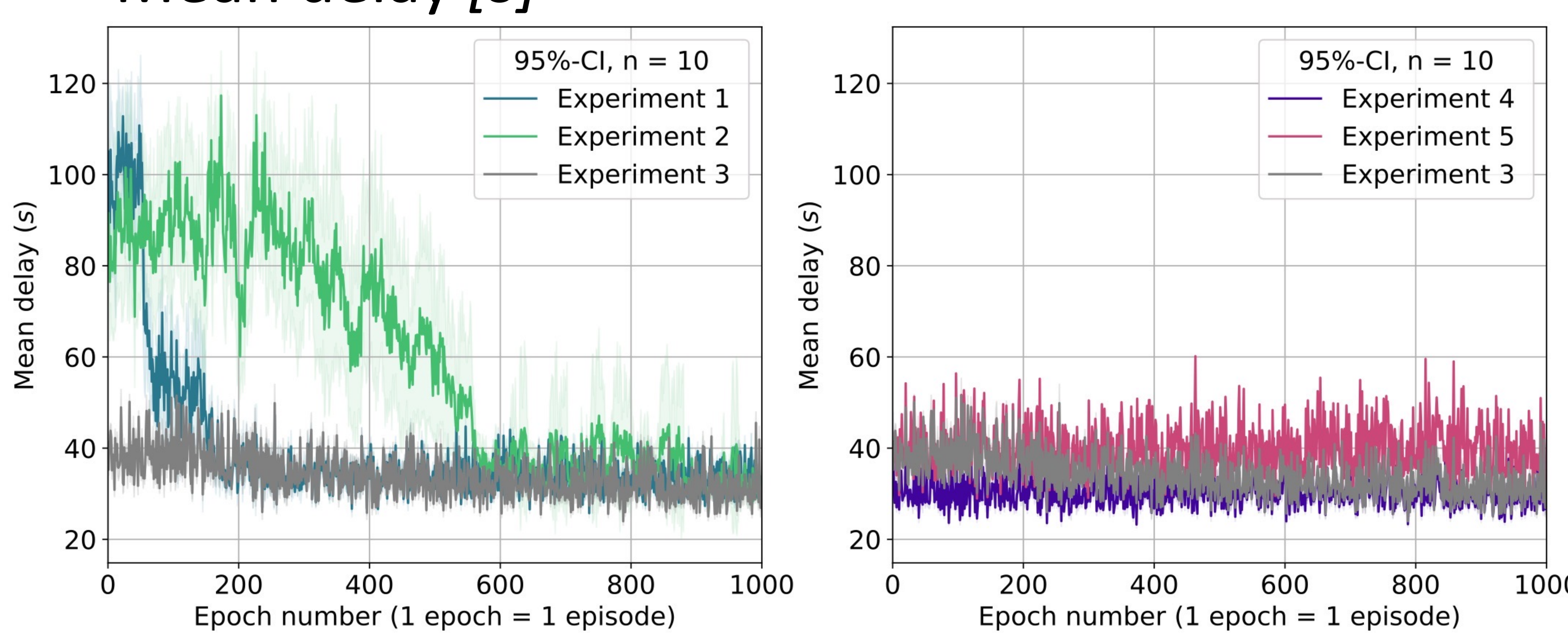
*each experiment is repeated 10 times

4) Result and conclusion

Intersection throughput *[veh]*



Mean delay *[s]*



epoch number ∝ learning progress

- ☑ **Same performance level**
- ☑ **Lower number of weight parameters**
  256 parameters → 12 parameters
- ☑ **Each individual numerical parameter allows numerical comprehension**
- ☑ **Efficient learning by the two proposed techniques**