ADAPTIVE POWER MANAGEMENT OF ENERGY HARVESTING SENSOR NODES USING REINFORCEMENT LEARNING

A Comparison of Q-Learning and SARSA Algorithms

適応的電力制御を行う環境発電駆動センサノードの強化学習戦略の比較評価

SWoPP 2017

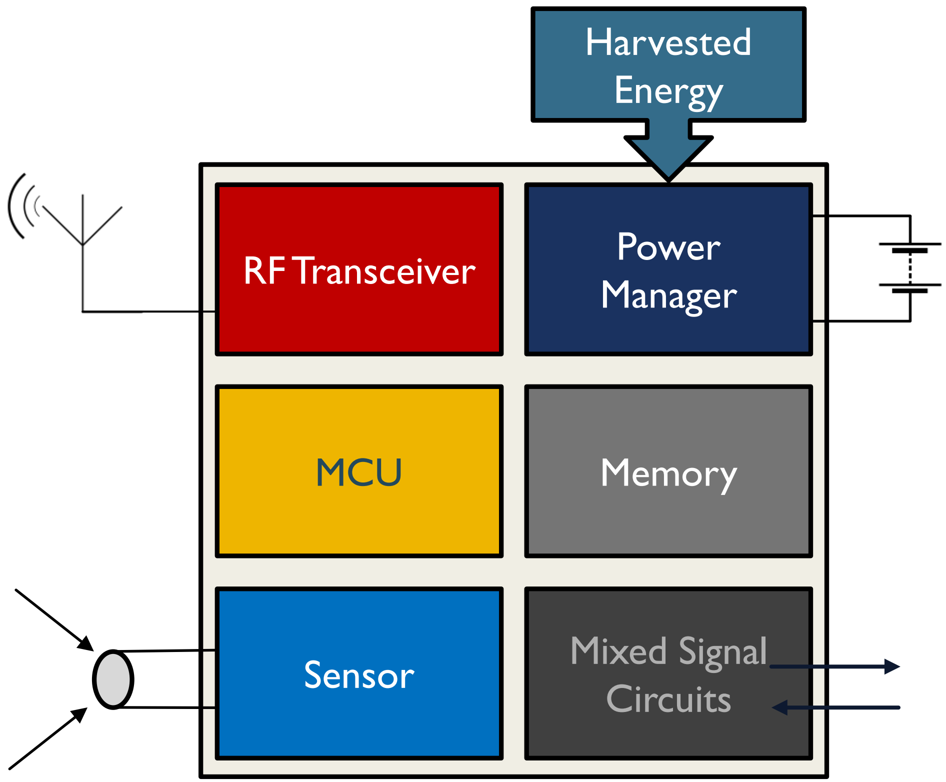
SHASWOT SHRESTHAMALI
MASAAKI KONDO
HIROSHI NAKAMURA

THE UNIVERSITY OF TOKYO

INTRODUCTION

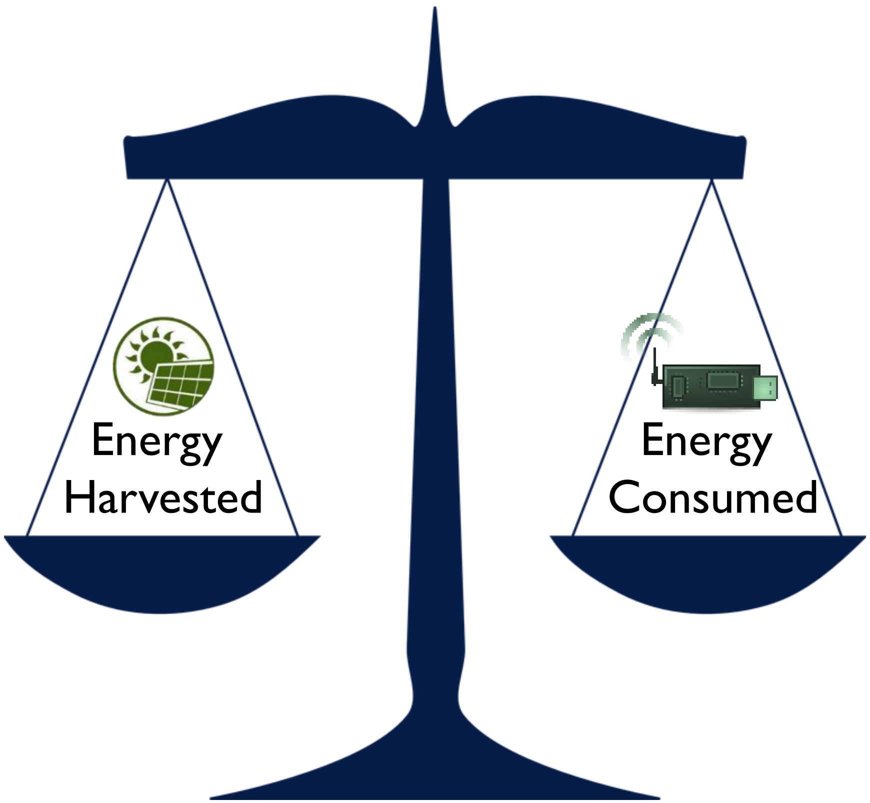
- Use Reinforcement Learning (RL) for power management in Energy Harvesting Sensor Nodes (EHSN)
  - Adaptive control behavior
  - Near-optimal performance
- Comparison between different RL algorithms
  - Q-Learning
  - SARSA

ENERGY HARVESTING SENSOR NODE CONCEPT



- **CONSTRAINTS**
  - Sensor node has to be operating at ALL times
  - Battery cannot be completely depleted
  - Battery cannot be overcharged (exceed 100%)
  - Battery size is finite
  - Charging/discharging rates are finite

OBJECTIVE: NODE-LEVEL ENERGY NEUTRALITY

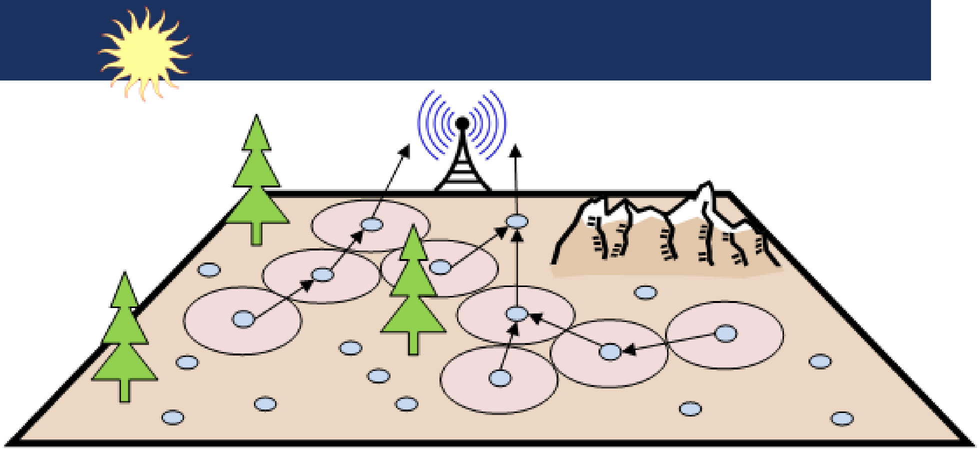


- We want to use **ALL** the energy that is harvested.
- One way of achieving that is by ensuring **node level energy neutrality –** the condition when the amount of energy harvested equals the amount of energy consumed.
- Autonomous Perpetual operation can be achieved

CHALLENGES

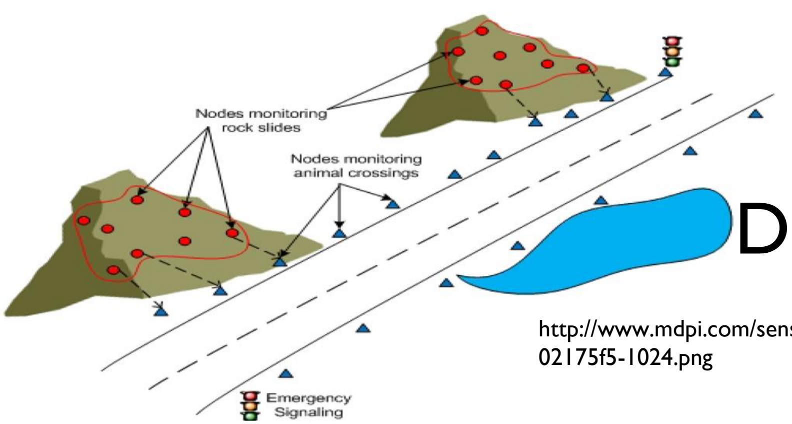
Environmental Sensor Networks – P.I. Corke et. al.

MOVING SENSORS



https://sites.google.com/site/sarmavrudhula/home/research/energy-management-of-wireless-sensor-networks

DIFFERENT ENVIRONMENTS



DIFFERENT SENSORS

http://www.mdpi.com/sensors/sensors-12-02175/article_deploy/html/images/sensors-12-02175f5-1024.png

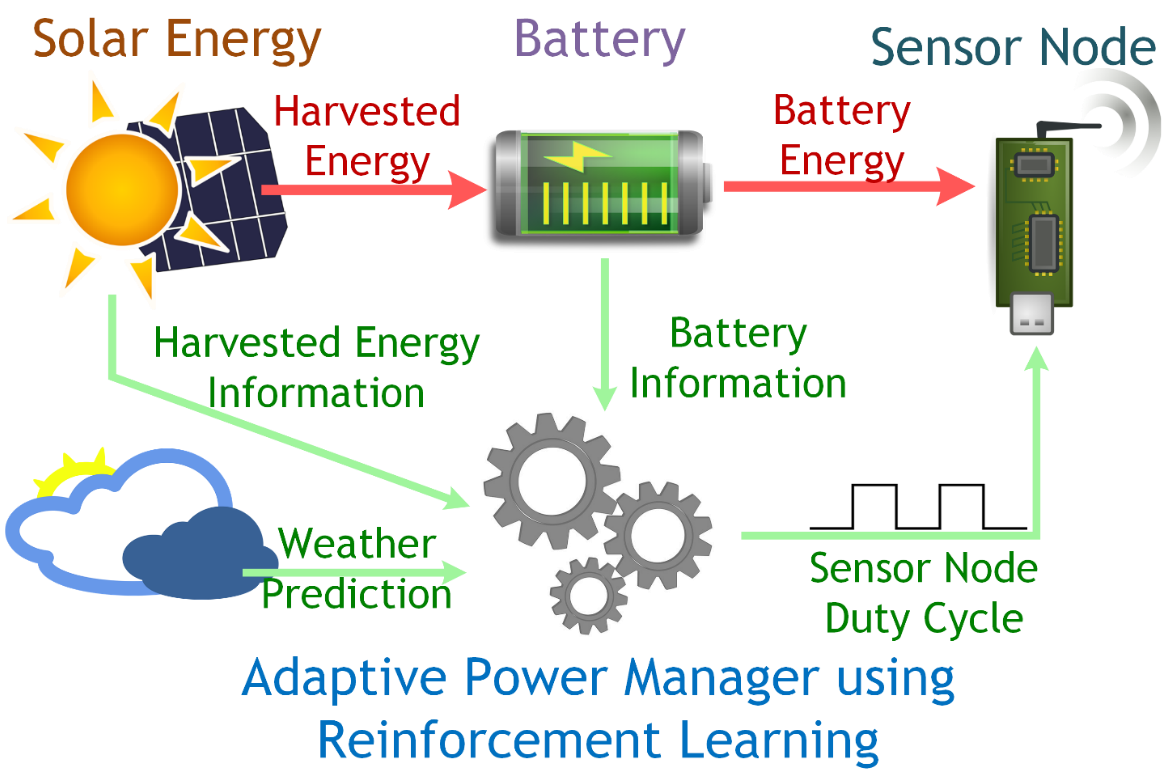
SOLUTION

Preparing heuristic, user-defined contingency solutions for all possible scenarios is **impractical**.

We want a **one-size-fits-all** solution sensor nodes that are capable of:

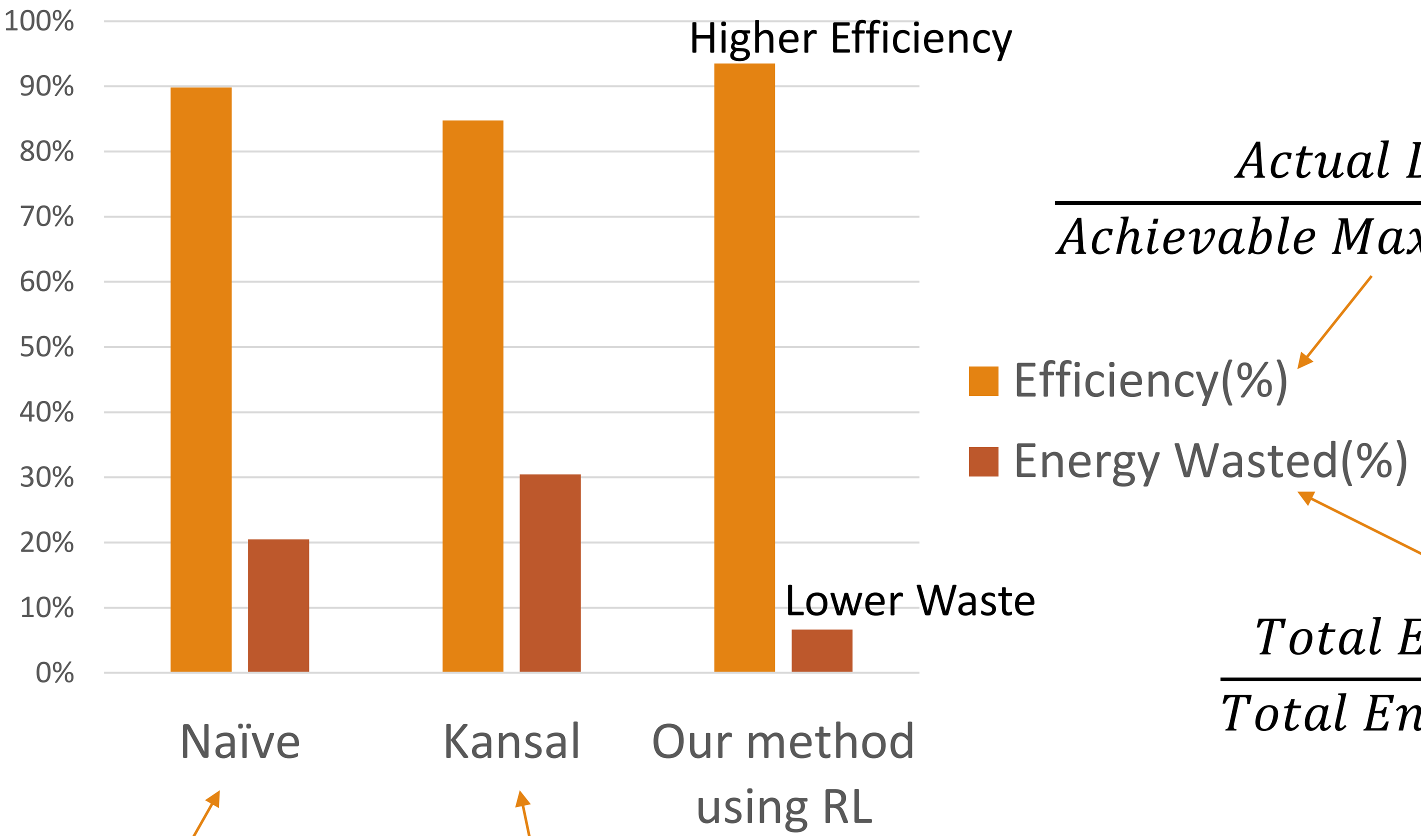
- **autonomously learning** optimal strategies
- **adapting** once they have been deployed in the environment.

SOLUTION



- Use RL for adaptive control
- Use a solar energy harvesting sensor node as a case example

Q-Learning Results (ETNET 2017)



$$\frac{Actual\ Duty\ Cycle}{Achievable\ Maximum\ Duty\ Cycle}$$

$$\frac{Total\ Energy\ Wasted}{Total\ Energy\ Harvested}$$

Duty Cycle is proportional to battery level

Fix duty cycle for present day by predicting total energy for next day

$$Energy\ Waste = Energy\ Harvested - Node\ Energy - Charging\ Energy$$

Q-Learning (ETNET 2017)

- Demonstrated that RL approaches outperform traditional methods.
- **Limitations**

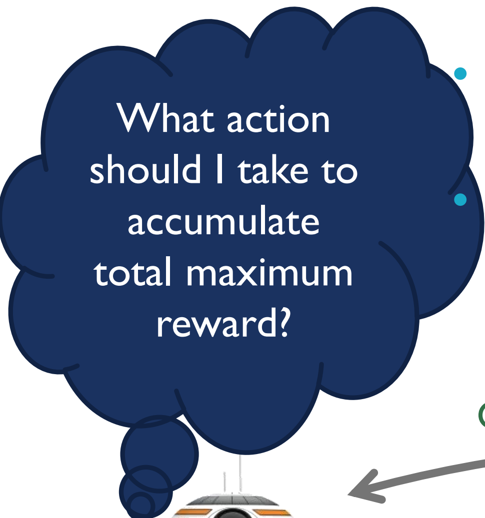
- State explosion
  - 200 x 5 x 6 states
  - Q-table becomes too large to train using random policy
- Long training times
  - Required 10 years worth of training
- Reward function did not reflect the true objective of energy neutrality.

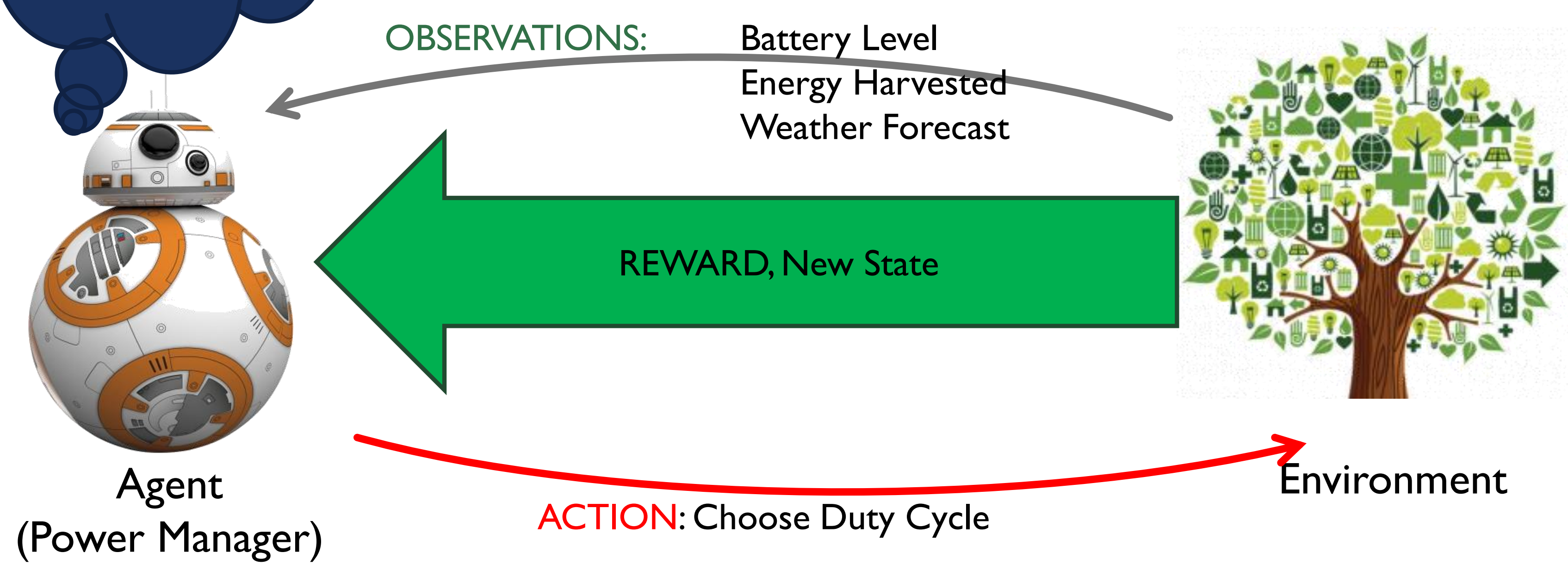
REINFORCEMENT LEARNING

IN A NUTSHELL

REINFORCEMENT LEARNING

- Type of Machine Learning based on experience rather than instruction
- Map situations (states) into actions – and receive as much reward as possible





REINFORCEMENT LEARNING

- IMPORTANT CONCEPTS
  - Q-VALUE
  - ELIGIBILITY TRACES

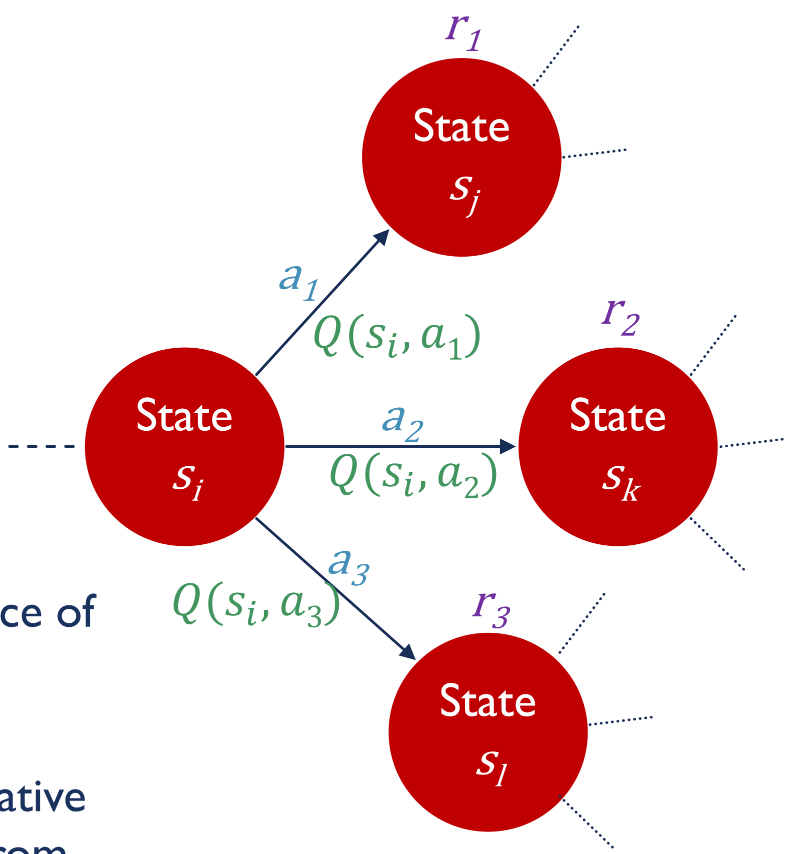
Q-VALUE

- To give a measure of the "goodness" of an action in a particular state, we assign each state-action pair a Q-value:
  Q(state, action)

- Learned from past (training) experiences.

- Higher Q-value → better the choice of action for that state.

- Q(s,a) value is the *expected* cumulative reward that you can get starting from state *s* and taking action *a*



Q-VALUE

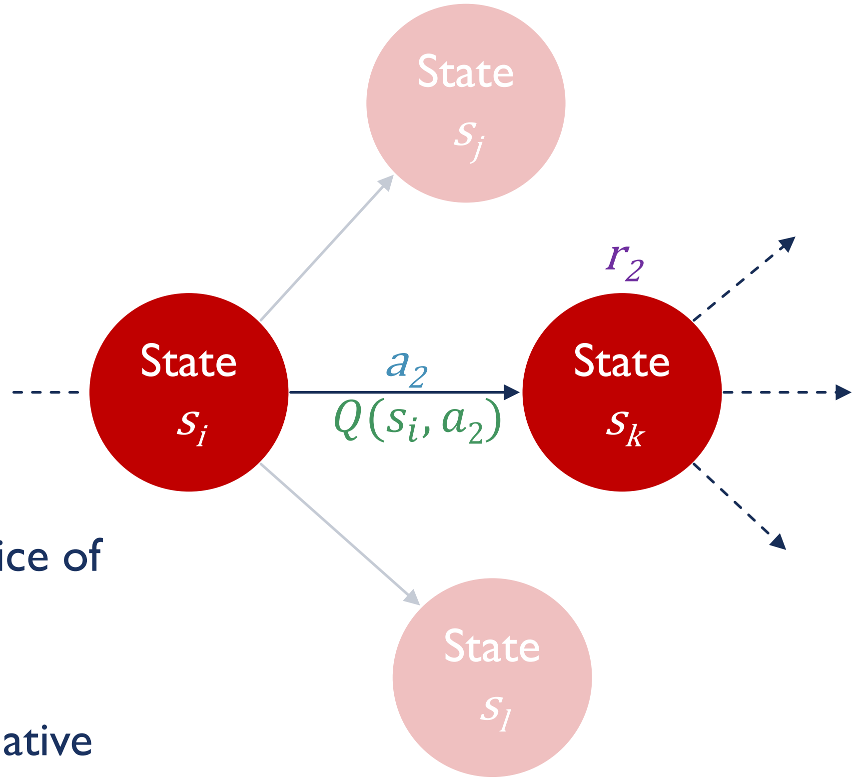
- To give a measure of the "goodness" of an action in a particular state, we assign each state-action pair a Q-value:

  Q(state, action)

- Learned from past (training) experiences.

- Higher Q-value → better the choice of action for that state.

- Q(s,a) value is the *expected* cumulative reward that you can get starting from state *s* and taking action *a*



LEARNING Q-VALUES

TO FIND $\boldsymbol{Q(s_k, a_k)}$

- Start with arbitrary guesses for $Q(s_k, a_k)$
- Update $Q(s_k, a_k)$ incrementally towards the *target* value (Bootstrapping)

- General Update Rule

$$NewEstimate \leftarrow OldEstimate + StepSize[Target - OldEstimate]$$
$$NewEstimate \leftarrow (1 - StepSize) \times OldEstimate + StepSize \times Target$$

$$Q(s_k, a_k) \leftarrow (1 - \alpha)Q(s_k, a_k) + \alpha \times Target$$

Target → ?

SARSA VS Q-LEARNING

- Agent starts at state s_k and takes some action a_k according to policy π.
- Receives a reward r_k and is transported to new state s_{k+1}.

SARSA

- The agent *considers* taking the next action a_{k+1}.
- The Q-value $Q(s_k,a_k)$ is then updated.

$$Q^{\pi}(s_k, a_k) \leftarrow (1-\alpha)Q^{\pi}(s_k, a_k) + \alpha[r_k + \gamma Q^{\pi}(s_{k+1}, a_{k+1})]$$

Q-LEARNING

- The agent *assumes* the next action will be the action with the highest Q-value.
- The Q-value $Q(s_k,a_k)$ is then updated.

$$Q(s_k, a_k) \leftarrow (1-\alpha)Q(s_k, a_k) + \alpha[r_k + \gamma \max_a Q(s_{k+1}, a)]$$

- ε-greedy policy is used i.e. random actions are taken with probability ε to allow exploration.

SARSA VS Q-LEARNING

SARSA

$$Q^{\pi}(s_k, a_k) \leftarrow (1-\alpha)Q^{\pi}(s_k, a_k) + \alpha[r_k + \gamma Q^{\pi}(s_{k+1}, a_{k+1})]$$

$$NewEstimate \leftarrow (1 - StepSize) \times OldEstimate + StepSize \times Target$$

Q-Learning

$$Q(s_k, a_k) \leftarrow (1-\alpha)Q(s_k, a_k) + \alpha[r_k + \gamma \max_a Q(s_{k+1}, a)]$$

SARSA VS Q-LEARNING

SARSA

- On-policy learning:
  - updates the policy it is using during training
- Update is carried out by *considering* the next action to be taken
- Faster convergence but requires an initial policy.
- Easier to integrate with function approximation

Q-Learning

- Off-policy learning:
  - final learned policy is the same regardless of training methods
- Assumes the best actions will always be taken
- Takes longer to converge
- Difficult to integrate with linear function approximation

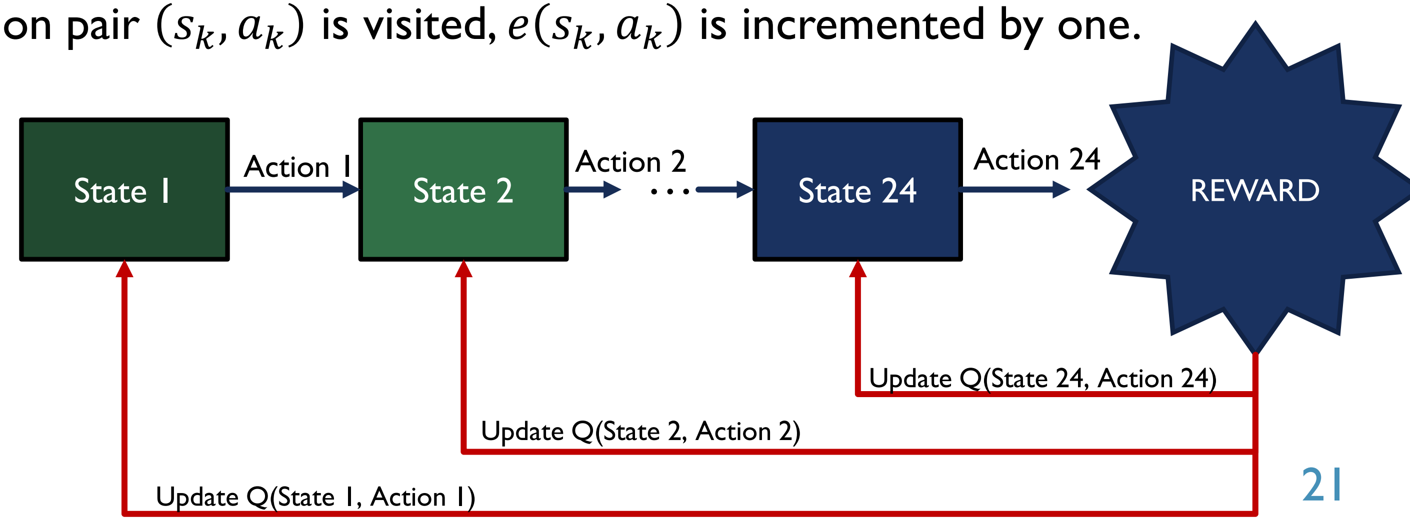
	SARSA	Q-Learning
Choosing Next Action	ε-greedy policy	ε-greedy policy
Updating Q	ε-greedy policy	Greedy policy

ELIGIBILITY TRACES

- In our model, one action is taken every hour. The reward is awarded at the end of 24 hours. A single action cannot justify the reward at the end. A series of 24 state-action pairs are responsible for the reward.

- To update the Q-values of the *appropriate* state-action pairs, we introduce a memory variable, $e(s, a)$, called the **eligibility trace**.

- $e(s, a)$ for ALL state-action pairs decays by λ at every time step.
- If the state-action pair (s_k, a_k) is visited, $e(s_k, a_k)$ is incremented by one.



SARSA(λ) AND Q-LEARNING (λ)

- SARSA(λ) – integrate eligibility traces with SARSA algorithm
- Q(λ) – integrate eligibility traces with Q-Learning algorithm
- λ, $0 < \lambda < 1$, is the strength with which Q-values of early contributing state-action pairs are updated as a consequence of the final reward.

ADAPTIVE POWER CONTROL USING REINFORCEMENT LEARNING ALGORITHMS

- SARSA(λ) – SARSA with eligibility traces
- SARSA
- Q(λ) – Q-Learning with eligibility traces
- Q-Learning

STATE DEFINITION

$$\text{State at } t_k = \left(S_{dist}(k), S_{batt}(t_k), S_{eharvest}(t_k), S_{day}(t_k) \right)$$

Distance from energy neutrality, $S_{dist}(t_k)$	**Battery,** $S_{batt}(t_k)$	**Harvested Energy,** $S_{eharvest}(t_k)$	**Weather Forecast,** $S_{day}(t_k)$
- 20000 mWh	Low (< 20%)	0 mWh	Very little sun
- 19000 mWh	Mid (20% to 80%)	0 to 100 mWh	Overcast
⋮	High (> 80%)	100 mWh to 500 mWh	Partly Cloudy
0 mWh		500 mWh to 1000 mWh	Fair
⋮		1000 mWh to 1500 mWh	Sunny
19000 mWh		1500 mWh to 2000 mWh	Very Sunny
20000 mWh		> 2000 mWh	

ACTION SPACE

- Choose duty cycle of the sensor node

$$A = a(t_k) \in \{1,2,3,4,5\}$$

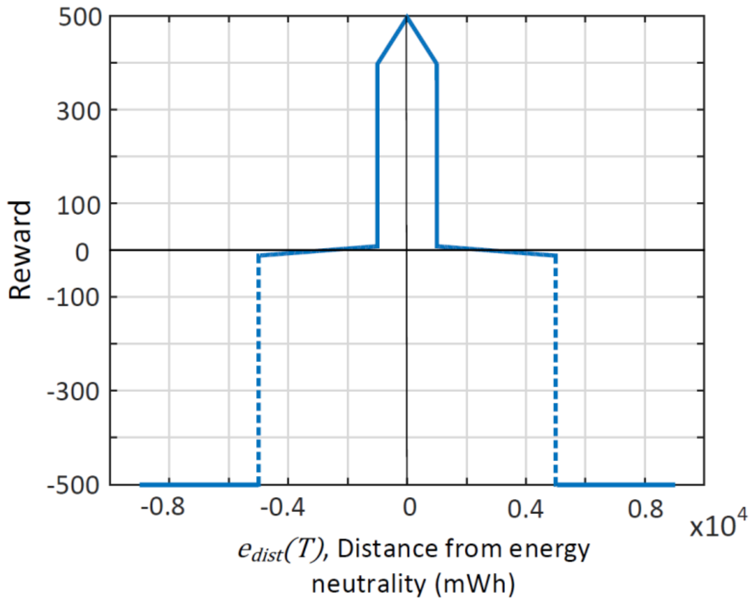
ACTION $a(t_k)$	DUTY CYCLE (%)	ENERGY CONSUMED PER HOUR (mWh)
1	20	100
2	40	200
3	60	300
4	80	400
5	100	500

REWARD FUNCTION

- Awarded at the end of an episode (day).
- Each episode consists of 24 one-hour epochs.
- We want the net energy difference between initial and final battery levels to be zero.

- Use a reward scheme that depends on **Energy Neutral Performance** (ENP) at the end of the episode $(t_k = T)$.

- Energy Neutral Performance can be defined here as
  - $|Initial\ battery\ level - Final(current) battery\ level|$



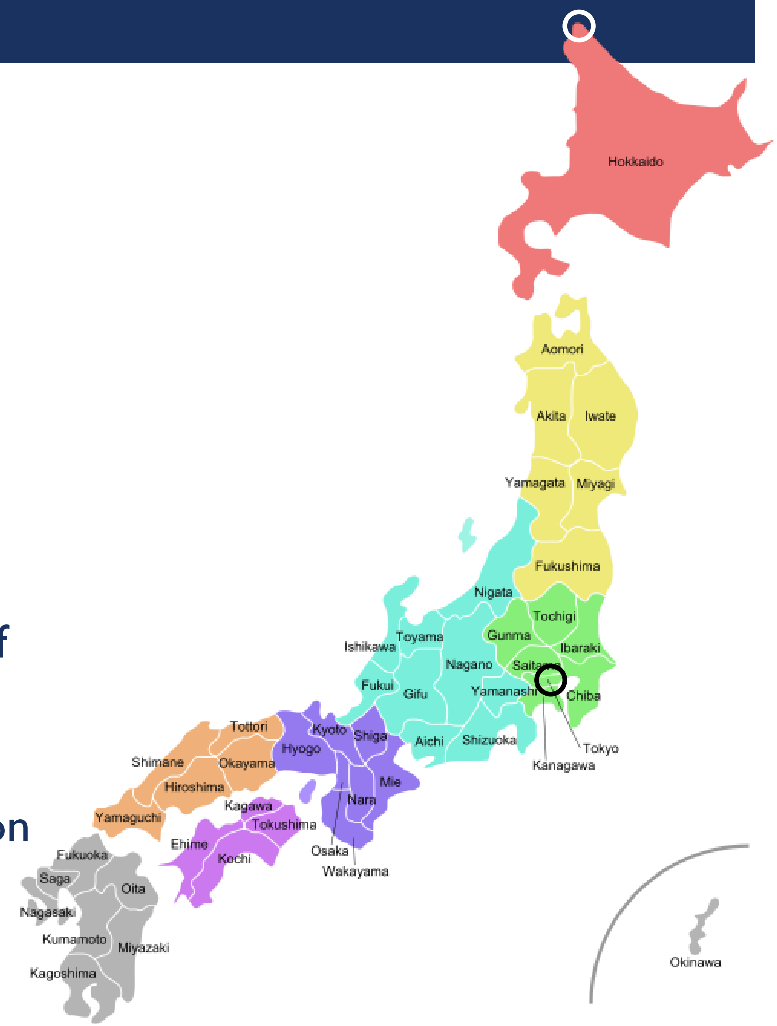
TRAINING AND TESTING

- **Training:**
  Tokyo, Year 2010

- **Testing:**
  Tokyo, Year 2010/2011
  Wakkanai, Year 2010/2011

- Wakkanai has a much colder climate than that of Tokyo and received much lesser solar radiation.

- We observe the adaptive behavior of our solution when the location of implementation is different from the location of its training

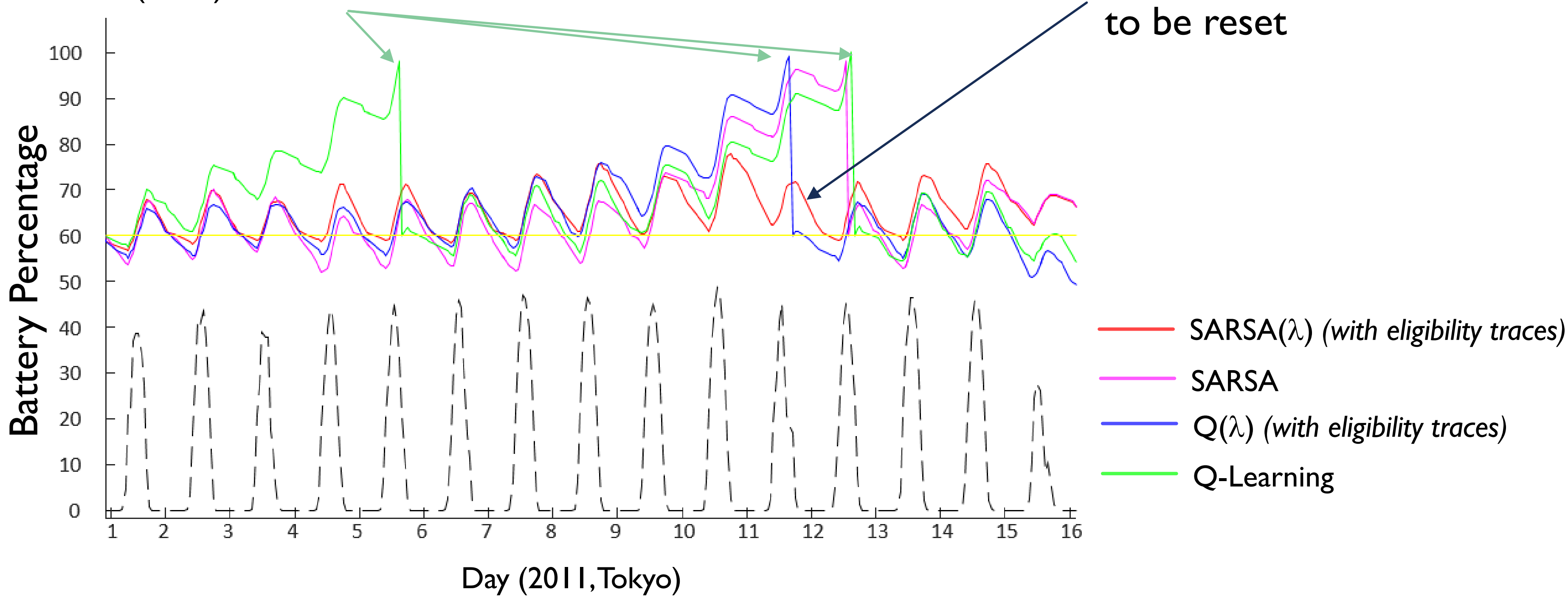


RESULTS

SARSA VS Q-LEARNING

Battery overflows and has to be reset to initial level (60%)

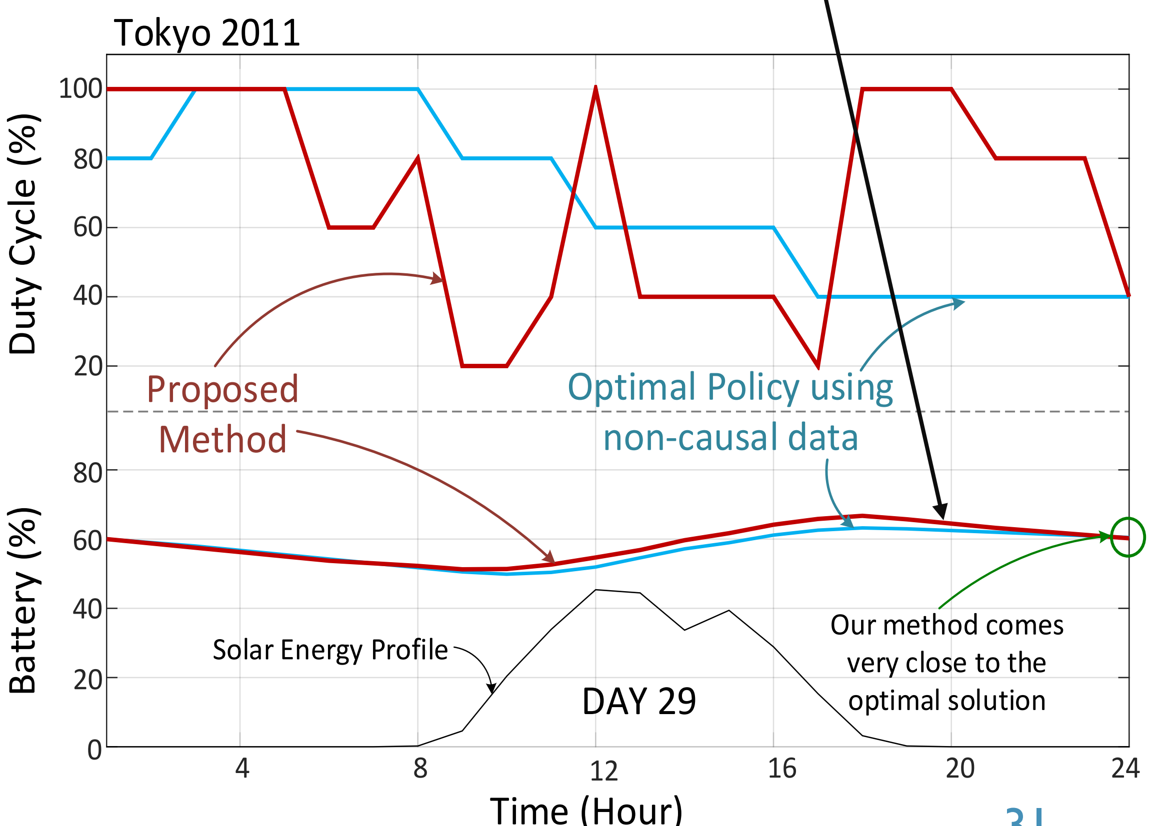
SARSA(λ) battery profile does not need to be reset



ENERGY NEUTRAL OPERATION

- SARSA(λ)compared with Optimal Policy
- Optimal Policy
  - Theoretical upper limit
  - Calculated using future information and linear programming techniques

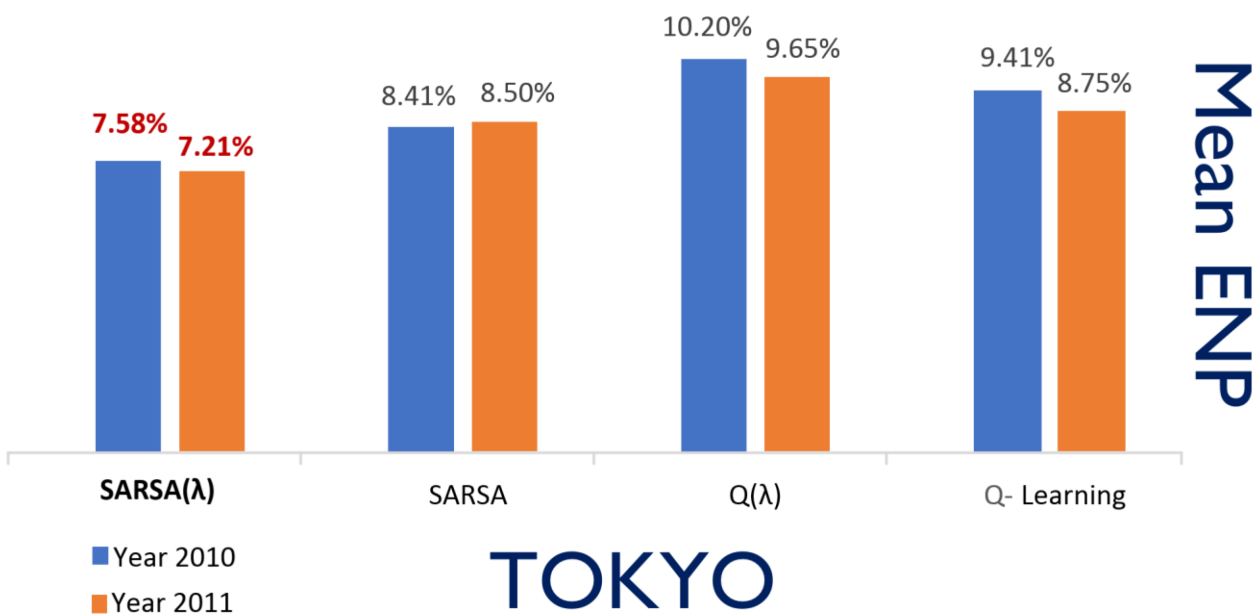
Battery profiles for SARSA and Offline Policy are very similar



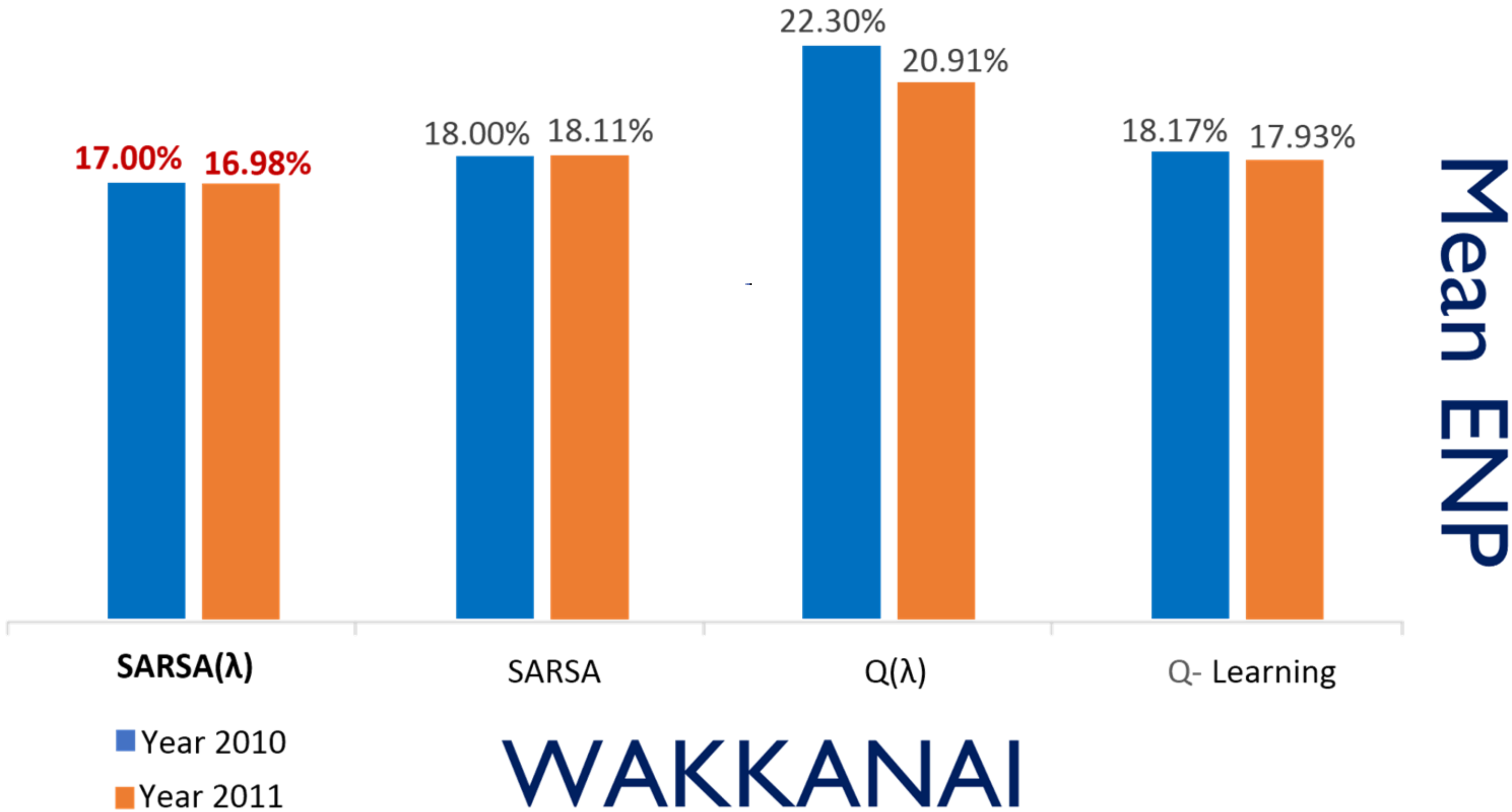
SARSA VS Q-LEARNING

- Every day the battery is reset to initial battery level
- ENP (as a percentage of maximum battery capacity, B_{MAX}) is observed at the end of each day of the year.

$$ENP = |Battery\ at\ 00{:}00\ - Battery\ at\ 23{:}59|$$
$$ENP = |60\%\ of\ B_{MAX} - Battery\ at\ 23{:}59|$$



SARSA VS Q-LEARNING



WAKKANAI

OBSERVATIONS

- SARSA(λ) – **BEST PERFORMANCE**.
- Q(λ) – **WORST PERFOMANCE**.
  - The “high” learning rate causes Q-values to oscillate with large amplitudes and the policy cannot converge.
  - A lower learning rate shows better performance but at expense of longer learning times.
- SARSA methods have a generally robust performance as compared to Q-Learning.

- Using eligibility traces with SARSA enhances the performance.

SUMMARY

- Adaptive Control is achieved by using SARSA RL methods .
  - Results from SARSA RL are near optimal.

- SARSA(λ) outperforms Q-Learning methods.

For further details about our work using SARSA(λ), please refer to our paper to be presented in EMSOFT 2017 and published in ACM TECS Journal.

Adaptive Power Management in Solar Energy Harvesting Node using Reinforcement Learning

THANK YOU FOR LISTENING

ANY COMMENTS OR QUESTIONS ARE WELCOME

shaswot@hal.ipc.i.u-tokyo.ac.jp

This work was partially supported by JSPS KAKENHI Grant Number 16K12405.