aiko

autonomous space missions.

www.aikospace.com

14-17 June 2021 | European workshop on On-Board Data Processing

Scheduling Downlink Operations using Reinforcement Learning

Luca Romanelli[1] (Speaker), Alessandro Benetton[1], Mattia Varile[1]

[1]AIKO Autonomous Space Missions, Torino, Italy

European Workshop on On-Board Data Processing

Summary

- AIKO objective
- Problem definition and RL approach
- Design choices
- Results & future work

Applying RL in space

- Our goal was to design and train an agent to make decisions autonomously without telling it how to do so
  - This is Reinforcement Learning

- 2 key questions:
  - What kind of decisions does the agent make?
    - This defines the goal of the RL agent
  - How can the agent do this?
    - Through interaction with the environment

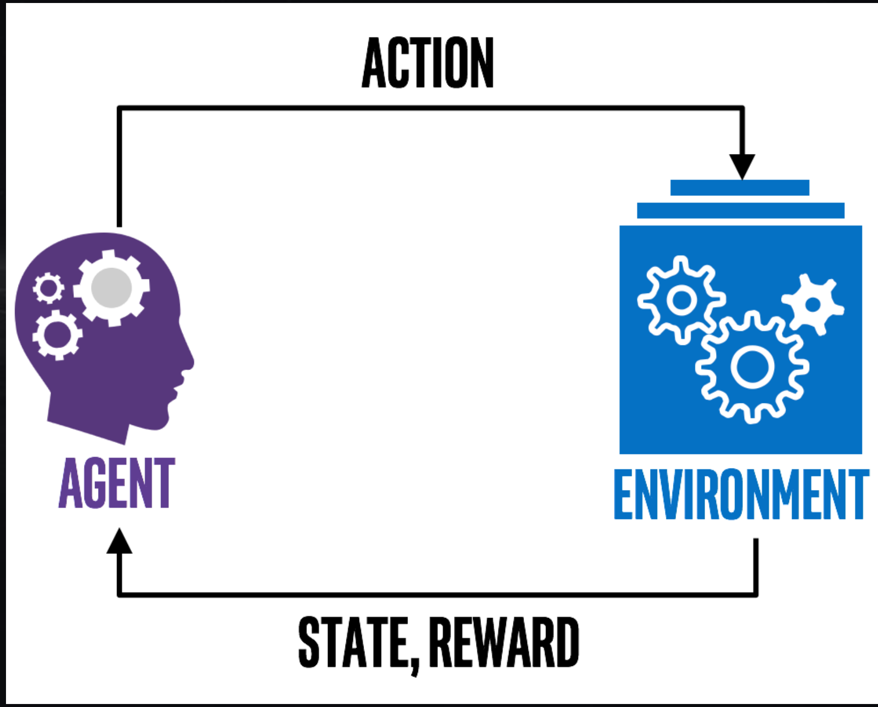
What is the goal?

- Goal: optimize the throughput and the efficiency of downlink operations

- What does the agent have to learn?
  - How to schedule satellite packets that need to be downloaded to ground to improve the outcome based on:
    - The type of data being downloaded
    - The resource utilization
    - The downlink capacity

How can the agent learn?

- The agent will learn through interaction with the environment
- The interaction is modeled as a Markov Decision Process
  - State space
  - Action space
  - Reward function

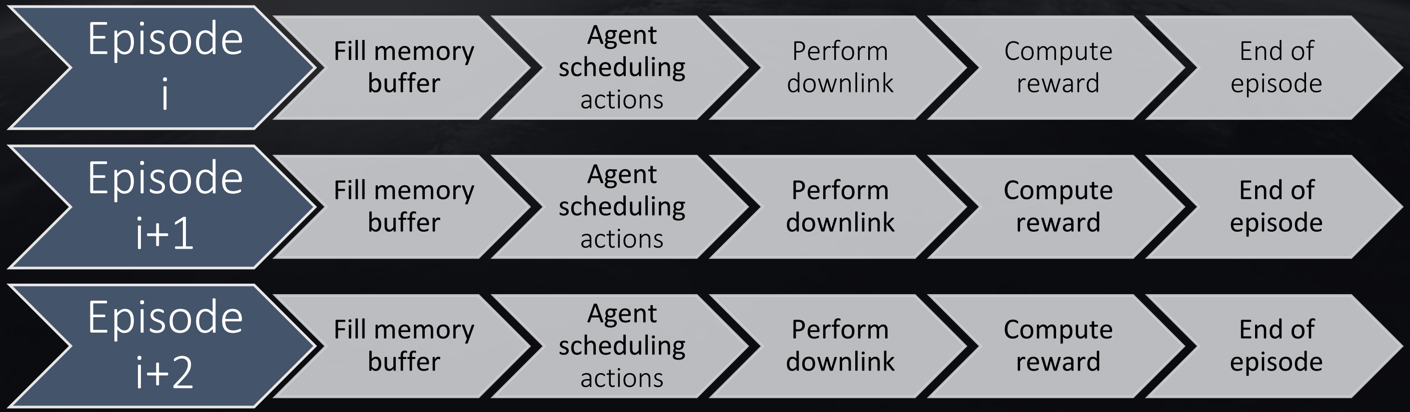


- The environment can be the real world or a simulator
  - Before we deploy a learning agent in space, we need to do experiments and train the agent in a simulation environment

The scheduling task

- The task is though as episodic
  - Each episode consists of scheduling packets stored in the buffer to be downloaded within a downlink operation

- Agent-Environment interaction:



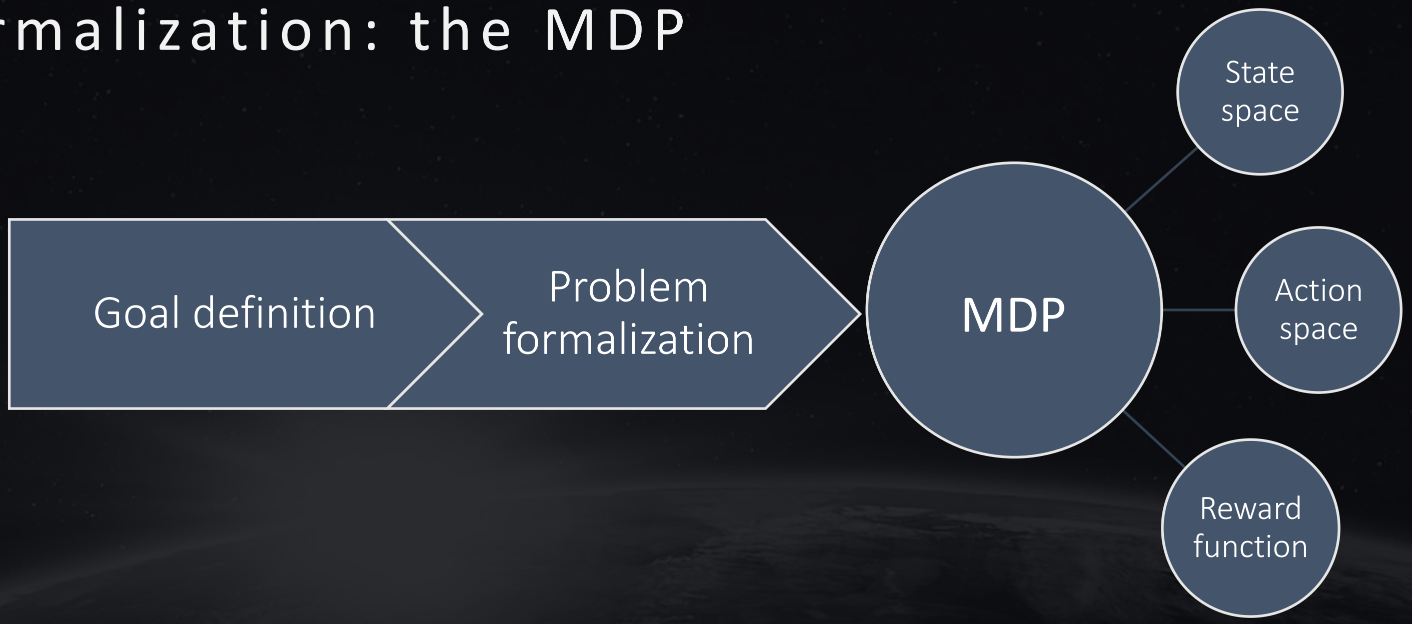
COP: the knapsack problem

- Given a set of items determine the number of each item to include in a collection so that:
  - the total weight is less than or equal to a given limit
  - the total value is as large as possible

- The online version of the problem is more challenging due to the uncertainty with which the items arrive
  - The problem is stochastic

aiko

Problem formalization: the MDP



	Classic/Offline	Online
State space	• Priority and length of the current packet to be scheduled • Total sum of priority and length residual inside the buffer • Residual downlink capacity	• Priority and length of the current packet to be scheduled • Max downlink capacity • Current used downlink capacity

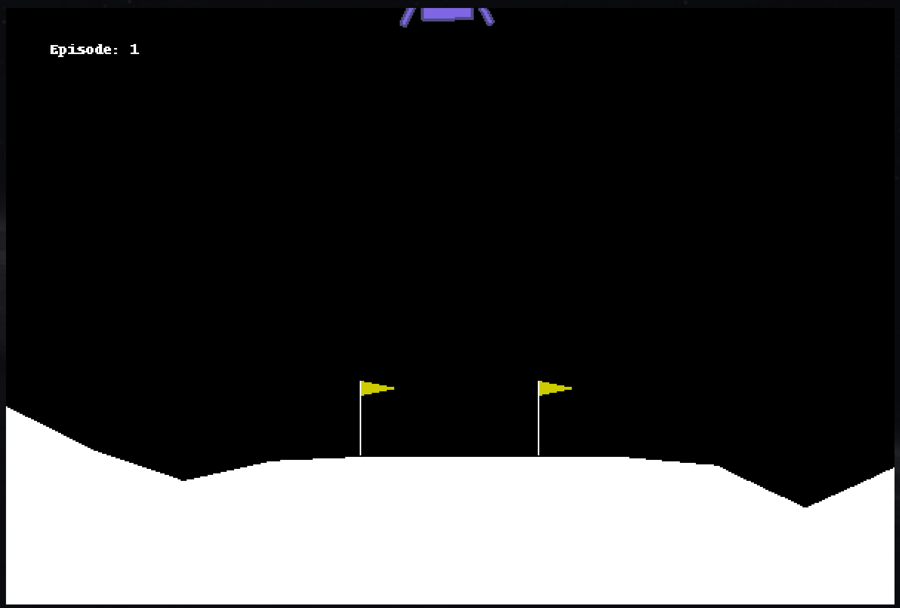
Problem formalization: the MDP

	Classic/Offline	Online
Action space	• Schedule or not schedule the current packet of the sequence	• Accept or reject the current packet available
Reward function	• A positive reward proportional to the priority of the packet when it's scheduled • A negative reward at the end of the episode proportional to the residual downlink capacity	• A positive reward proportional to the priority of the packet when it's scheduled • A negative reward when the agent schedules a packet that doesn't fit the current downlink capacity

Environment implementation

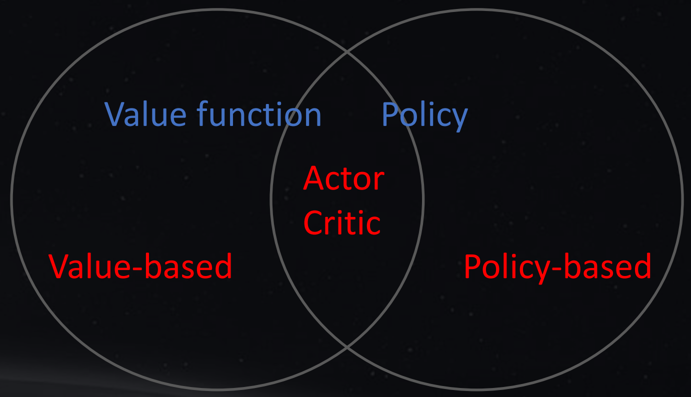
- Based on the Openai-gym interface
  - The most common toolkit for training RL algorithms
  - Almost all RL frameworks are compatible with this interface

- It consists of three main blocks
  - *init*: the initialization of the environment
  - *reset:* the beginning of an episode
    - Responsible for returning the first observation of the environment
  - *step*: the update of the environment after an agent's action
    - Responsible for returning the next state and reward to the agent



Algorithm choice

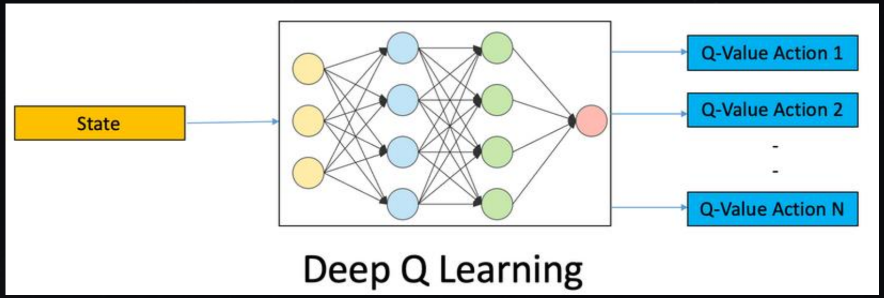
- What to learn?
  - The value function (Q-function)
    - value-based algorithms
  - The policy
    - policy-based algorithms
  - Both the policy and the value function
    - actor-critic algorithms



- Function approximation
  - Deep Neural Networks provide a powerful tool to generalize over new unseen observations

DQN agent

- Approximates the Q-value function using a deep Q-network
  - The number of input and output nodes are related to the state space and the action space

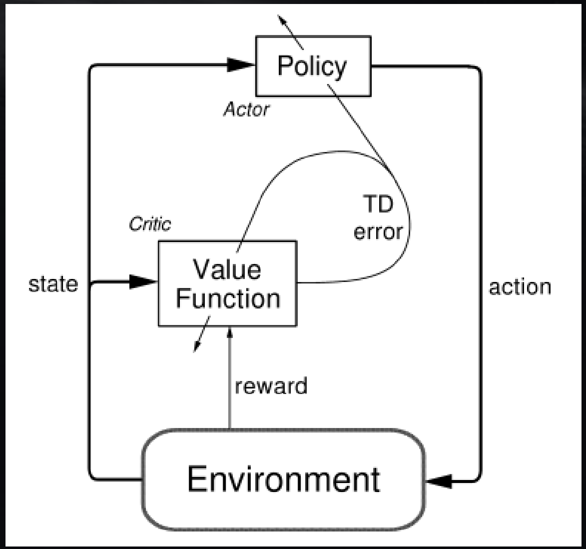


- Experience replay technique
  - The agent's experience is stored inside a replay memory and sampled randomly during the training

PPO agent

- Policy gradient algorithm
  - It aims at modelling and optimizing the policy directly
  - It can learn stochastic policy

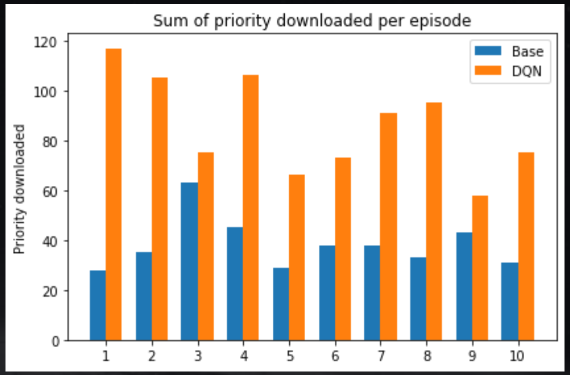
- Actor-Critic method
  - Actor network
    - Parametrized policy which select actions
  - Critic network
    - Approximated value function that criticizes the actions taken by the actor

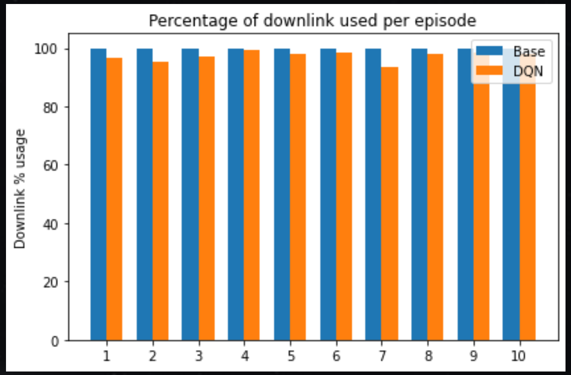


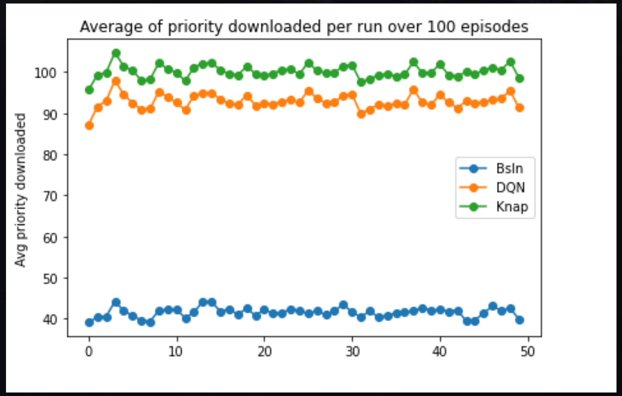
aiko

Simulation results – Offline problem

- Environment setup
  - Buffer of 100 packets to be downloaded with random values for length and priority
  - Random value of the downlink capacity in each episode
- DQN hyperparameters
  - 2 hidden layers with 64 units each
  - Mean squared TD-error
- Comparison with dynamic programming approach
  - 50 runs with 100 episodes each

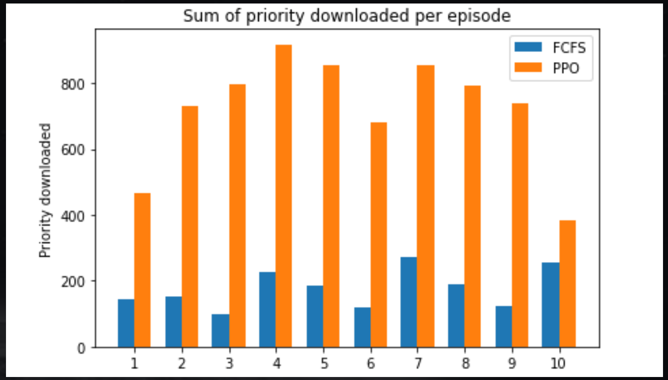


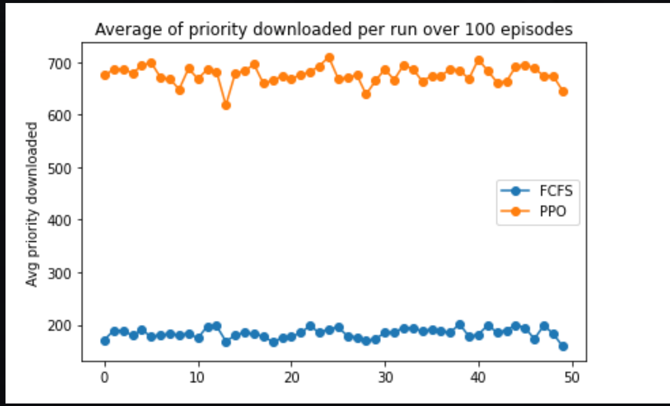




Simulation results – Online problem

- Environment setup
  - Windows of 50 packets which are generated online with random values for length and priority
  - Random value of the downlink capacity in each episode
- PPO hyperparameters
  - Same network architecture for both the actor and the critic
  - 2 hidden layers with 64 units each
- Average results over 50 runs with 100 episodes each





Future improvements and next steps

- Improvements:
  - Add complexity to the scenario
    - e.g. add stochasticity to the downlink operations

- Next steps:
  - Apply the RL solution to different optimization problems