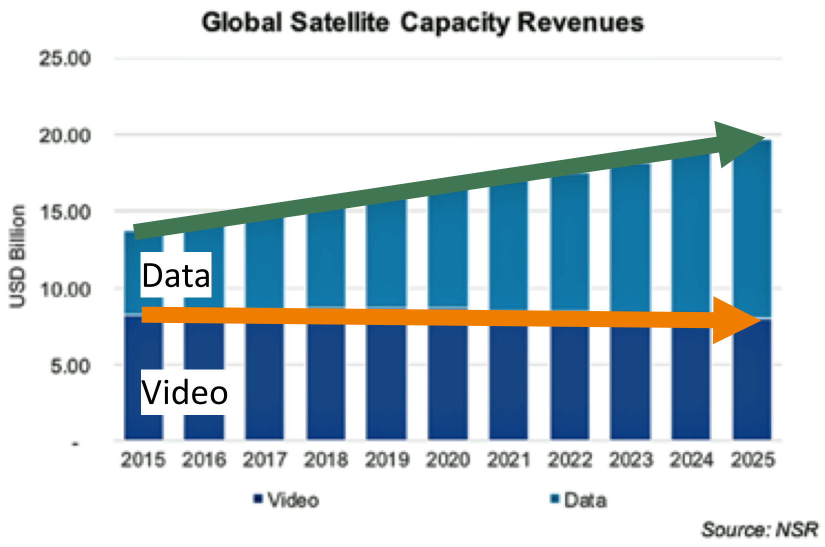
Deep Reinforcement Learning for Continuous Power Allocation in Flexible High Throughput Satellites

Juan Jose Garau Luis (garau@mit.edu), **Markus Guerster, Inigo del Portillo, Edward Crawley, Bruce Cameron**
Massachusetts Institute of Technology

June 26th 2019

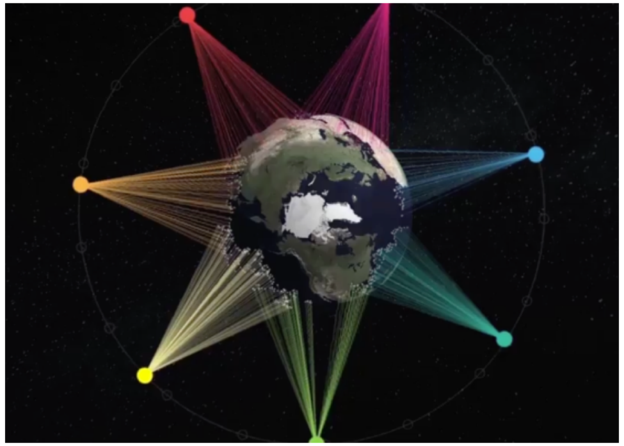
IEEE Cognitive Communications for Aerospace Applications Workshop
Ohio Aerospace Institute, Cleveland, Ohio

The next generation of communication satellites



- Satellite communications **demand** is estimated to **duplicate** by 2025, with **data** being the main business
- Demand will become **bidirectional** and more **fluctuating**
- **New entrants** include in-flight applications and cruise ships

- Spot beams, phased arrays, and digital processors will provide **increased flexibility** to new systems
- Future constellations will have more than 20,000 **fully-dynamic spot beams**
- The power and bandwidth, the frequency plan, and the pointing and shape of **each beam** will be **individually configurable**



O3b mPower

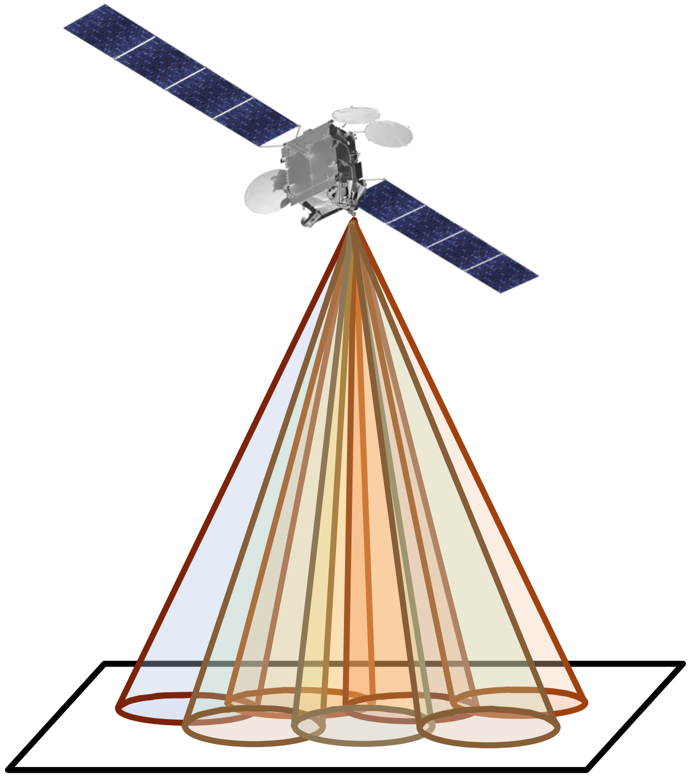
Dynamic resource allocation problem

- Satellite operators face the challenge of **automating** their **resource allocation** strategies to exploit this new flexibility and turn it into a larger service capacity

- The problem is **complex**: the solution space is **high-dimensional**, **non-convex** [1], and **NP-hardness** has been proved [2]

- Previous studies have examined **metaheuristic** algorithms [1-4], which are **not easily operable** under **real-time constraints**

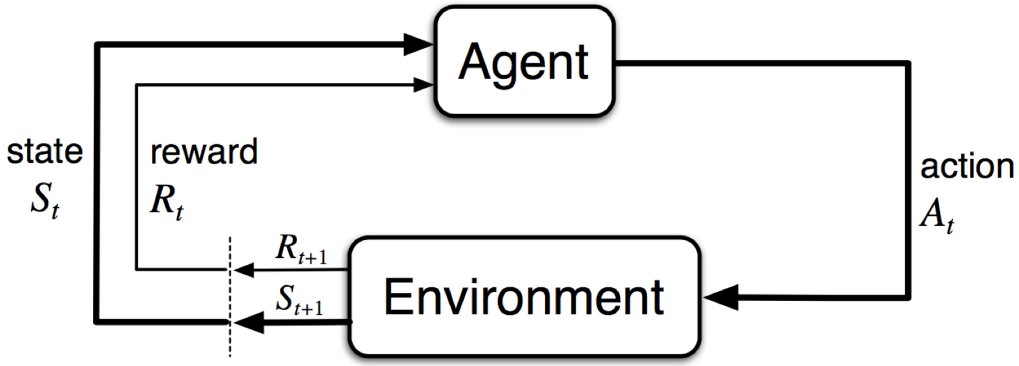
- Two recent studies [5, 6] have applied **discretized** Deep Reinforcement Learning (**DRL**) approaches, **challenging** when dimensionality is high



- We propose a **DRL architecture** based on **continuous** variables to allocate **power,** working within time and dimensionality **constraints**

Reinforcement Learning

- Typical **Reinforcement Learning (RL)** setup is composed of **five elements** [7]



- Goal is to find a **policy** that maps each state into an action to **maximize** cumulative discounted **reward**

$$\boxed{\pi(a_t|s_t)}$$

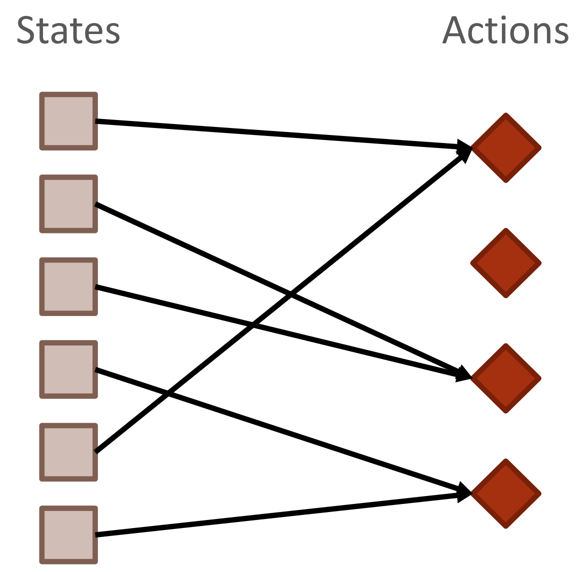
Policy

$$\boxed{G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k}$$

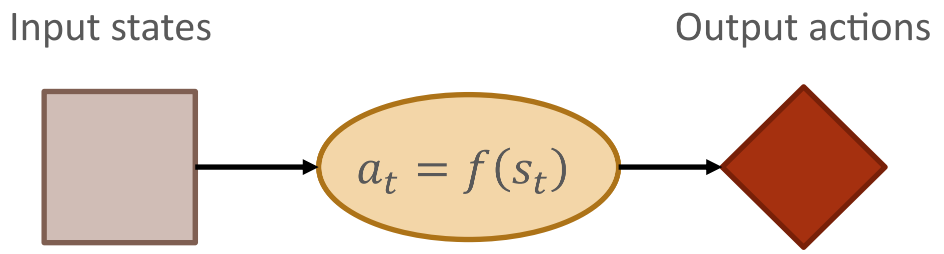
Cumulative discounted reward

Deep Reinforcement Learning

- When the number of different states and actions is **small**, computing **tabular policies** is preferred



- When dimensionality is **high** or states/actions are intrinsically **continuous**, computing a **tabular** policy is **impractical**
- Optimizing an **approximator function** is chosen **instead**



- **Deep Reinforcement Learning** consists of the use of **neural networks** as function approximators in a RL setup

Problem formulation

- Our objective is allocating power to each beam to **minimize** the **Unmet System Demand** (USD) and overall **power** consumption

$$
\begin{aligned}
\underset{P_{b,t}}{\text{minimize}} \quad & \sum_{t=1}^{T} \left[USD_t(P_{b,t}) + \beta \sum_{b=1}^{N_b} P_{b,t} \right] \\
\text{subject to} \quad & P_{b,t} \leq P_b^{max}, \quad \forall b \in \mathcal{B}, \forall t \in \{1, ..., T\} \\
& \sum_{b=1}^{N_b} P_{b,t} \leq P_{tot}, \quad \forall t \in \{1, ..., T\} \\
& P_{b,t} \geq 0, \quad \forall b \in \mathcal{B}, \forall t \in \{1, ..., T\}
\end{aligned}
$$

Power per beam

Maximum power per beam

Total satellite power

Minimum power per beam

- The **Unmet System Demand** accounts for the amount of **demand** that is **not satisfied** (also used in [2, 3]):

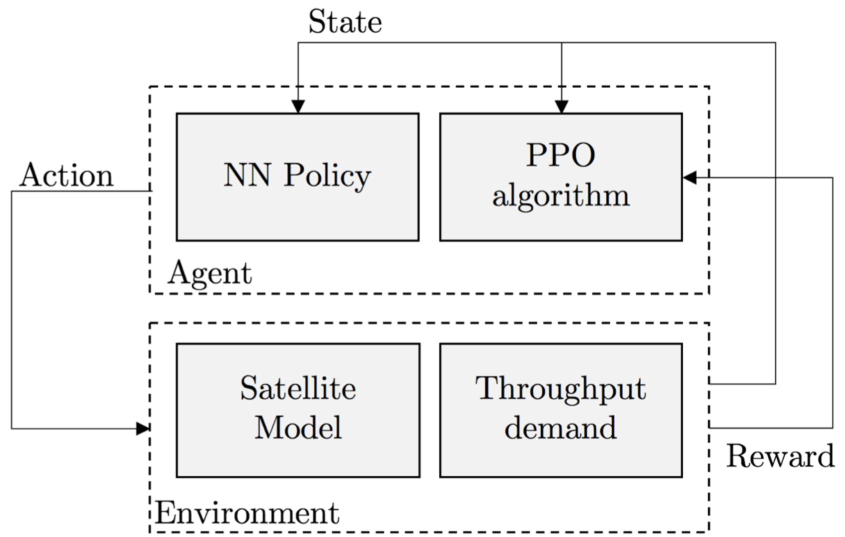
$$USD_t = \sum_{b=1}^{N_b} \max[D_{b,t} - R_{b,t}(P_{b,t}), 0]$$

Demand per beam

Data rate per beam

Proposed architecture

- DRL architecture based on a satellite communications **model**, a neural network **policy**, and the Proximal Policy Optimization (**PPO**) [8] algorithm as policy improvement method



- The **state** is composed by the demand of the current timestep and the demand and optimal power of the two previous timesteps

$$s_t = \{\mathcal{D}_t, \mathcal{D}_{t-1}, \mathcal{D}_{t-2}, \mathcal{P}^*_{t-1}, \mathcal{P}^*_{t-2}\}$$

- The **action** is the power allocated per beam

$$a_t = \{P_{b,t} \,|\, b \in \{1, ..., N_b\}, 0 \leq P_{b,t} \leq P_b^{max}\}$$

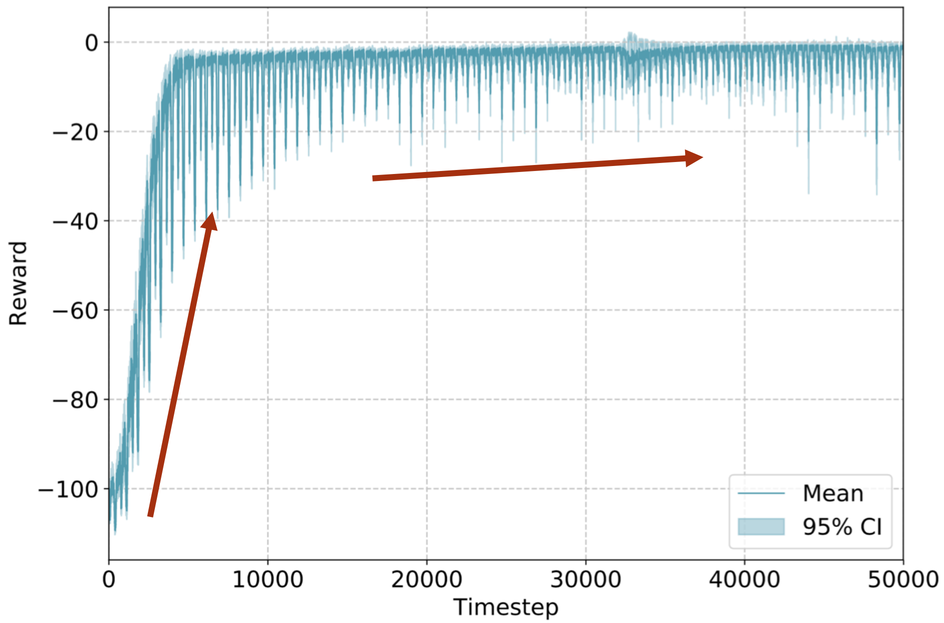
- The **reward** is a weighted combination of the USD and the power MSE

$$r_t = \frac{\alpha \sum_{b=1}^{N_b} \min(R_{b,t} - D_{b,t}, 0)}{\sum_{b=1}^{N_b} D_{b,t}} - \frac{\sum_{b=1}^{N_b} (P_{b,t} - P^*_{b,t})^2}{\sum_{b=1}^{N_b} P^*_{b,t}}$$

- The **policy network** chosen is a Multilayer Perceptron (MLP), a fully-connected network

Results

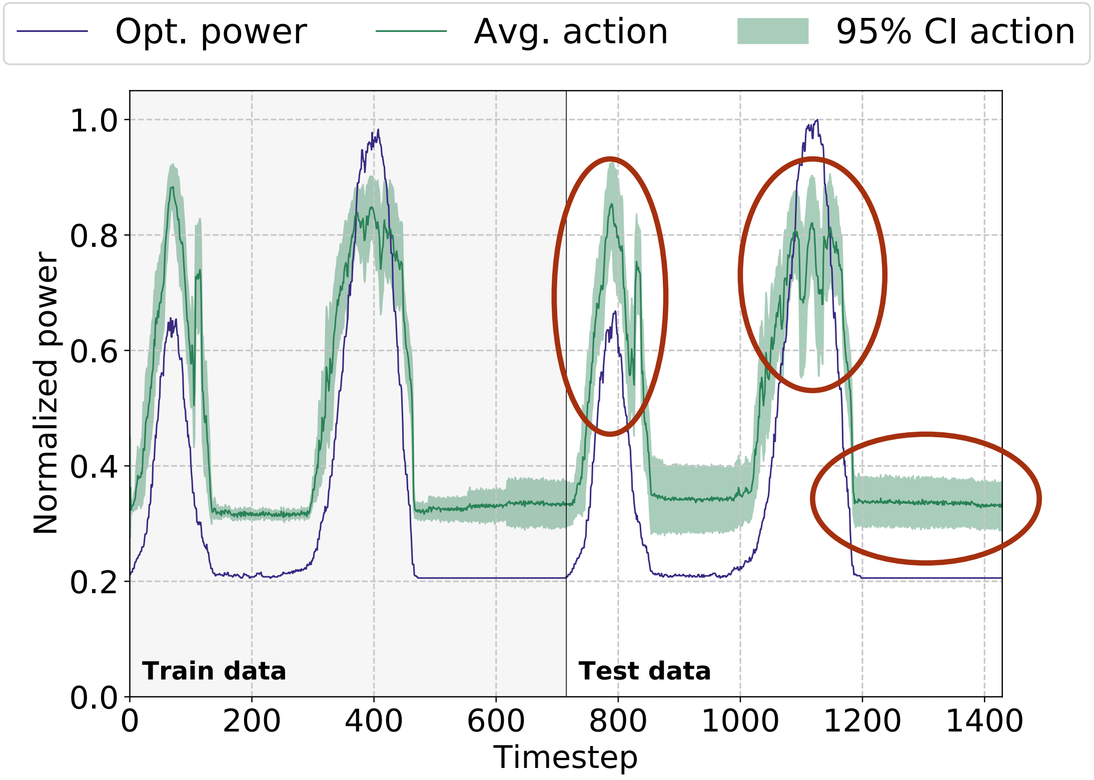
- 30-beam GEO satellite located over North America
- Time series, provided by SES, with demand samples every 2 minutes throughout 48 hours
- First 24 hours taken as training data, policy evaluated on last 24 hours
- Results averaged over 10 simulations



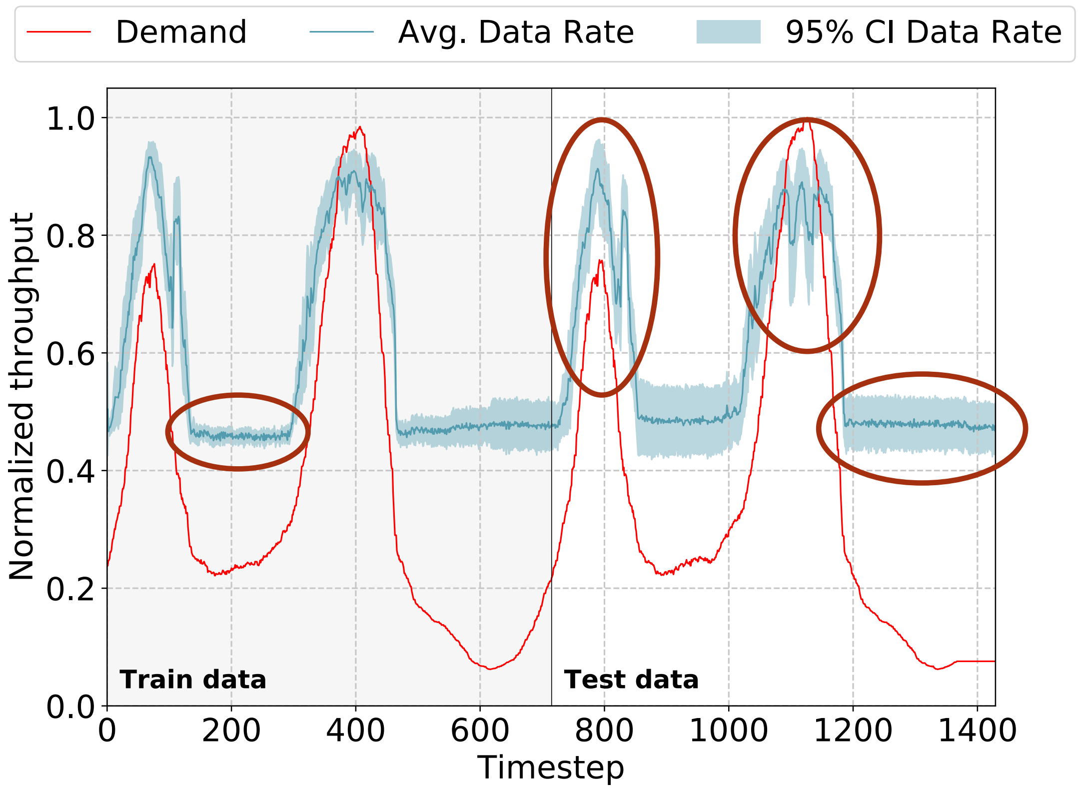
Training reward sequence

- The agent quickly **learns** that **increasing** mean **power** is better to serve customers
- After ~5,000 iterations the policy **saturates** and starts learning **frequency** components

Results



Power allocated by the policy vs. optimal power



Data rate result of the policy vs. demand

Extended results after the paper

Performance summary table – Test data

	MLP	LSTM	GA 125 it.	GA 500 it.
Agg. demand	1	1	1	1
Avg. USD	0.0093	0.0116	0	0
Opt. energy	1	1	1	1
Output energy	1.35	1.41	1.22	1.05
Exec. time (s)	0.019	0.020	25.6	98.9

- The policy, on average, serves 99% of the demand
- Spends 35% more power than necessary

- A Long Short Term Memory network (LSTM) does not necessarily improve the results

- GA generally achieve zero USD and better power results
- 1,300 times slower than DRL
- Hard to scale

Conclusions and future work

In this study we have…

- Proposed a **Deep Reinforcement Learning** architecture for **power allocation** using **continuous** state and action spaces
- Simulated a **30-beam satellite** with a dynamic resource management engine based on our architecture
- Achieved a **~1,300 times speed increase** with respect to metaheuristics while offering comparable quality solutions

Next steps include…

- **Refining** the architecture, since the policy presents some **suboptimalities** in terms of power allocation (35% extra power compared to GA)
- Working on the **generalizability** (robust to diverse data) and **scalability** (systems with more beams) of the policy
- Increasing the **complexity** of the problem by adding **new** optimization **variables** (e.g. frequency plan)

References

[1] G. Cocco, T. De Cola, M. Angelone, Z. Katona, and S. Erl, “Radio resource management optimization of flexible satellite payloads for DVB- S2 systems,” IEEE Transactions on Broadcasting, 64(2):266-280, 2018

[2] A. I. Aravanis, B. Shankar, P. D. Arapoglou, G. Danoy, P. G. Cottis, and Bjorn Ottersten, “Power allocation in multibeam satellite systems: a two-stage multi-objective optimization,” IEEE Transactions on Wireless Communications, 14(6):3171-3182, 2015

[3] A. Paris, I. del Portillo, B. G. Cameron, and E. F. Crawley, “A genetic algorithm for joint power and bandwidth allocation in multibeam satellite systems,” in 2019 IEEE Aerospace Conference, 2019

[4] F. R. Durand, and T. Abrão, “Power allocation in multibeam satellites based on particle swarm optimization,” International Journal of Electronics and Communications, 78:124-133, 2017

[5] P. V. Rodrigues Ferreira et al., “Multiobjective reinforcement learning for cognitive satellite communications using deep neural network ensembles,” IEEE Journal on Selected Areas in Communications, 36(5):1030-1041, 2018

[6] X. Hu, S. Liu, R. Chen, W. Wang, and C. Wang, “A deep reinforcement learning-based framework for dynamic resource allocation in multibeam satellite systems,” IEEE Communications Letters, 22(8):1612-1615, 2018

[7] R. S. Sutton, and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv preprint, arXiv:1707.06347, 2017

THANK YOU!

Contact e-mail: garau@mit.edu