Reinforcement learning algorithms for managing a steel slab yard

Elisa Savio[1], Laura Meneghetti[1], Pasquale Claudio Africa[1], Gianluigi Rozza[1], Loris Busolini[2], Matteo Sandri[2], Davide Armellini[2], Paolo Borzone[2]

[1]Mathematics Area, mathLab, SISSA, International School of Advanced Studies, Trieste, Italy
[2]Danieli Automation S.p.A., Buttrio, Udine, Italy

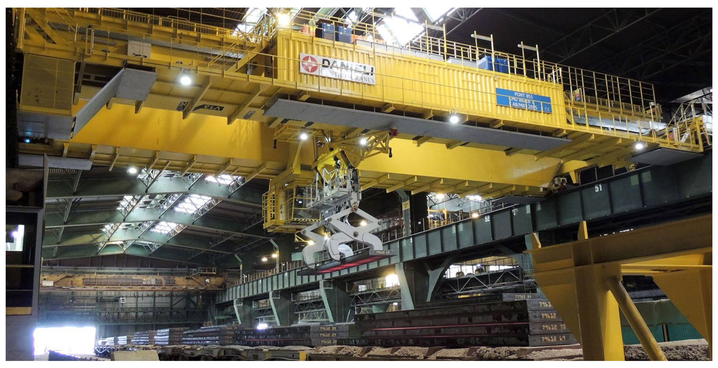
Introduction

We study the applicability of reinforcement learning algorithms to the management of a steel slab yard; the project is inspired by a model developed by Danieli, industry leader in the supply of completely automatic yards with Electric Overhead Travelling (EOT) cranes, in which products are moved automatically without operators' intervention. In particular we are investigating the applicability of Proximal Policy Optimization (PPO).

The real setting

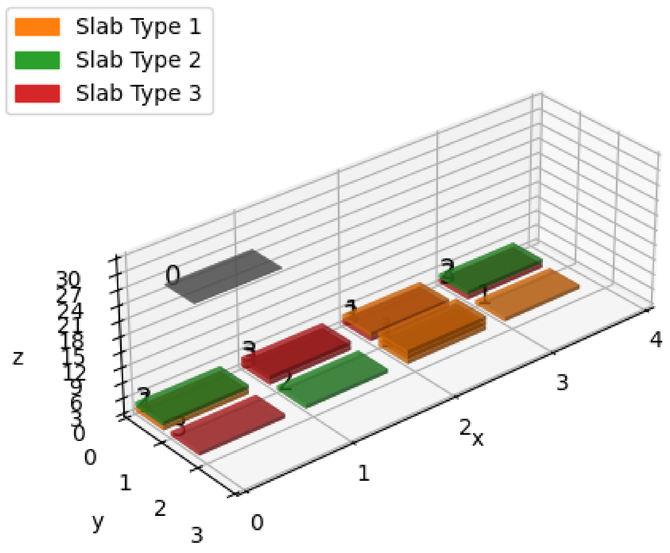
A slab yard is a large warehouse that stores steel slabs. Its management must reduce the slabs' relocations while correctly feeding the delivery zone. The problem is complicated by the high number of variables present:

- number of cranes,
- number of buffer zones,
- number of steel grades,
- changes in casting productivity,
- changes in the rolling mill production schedule.



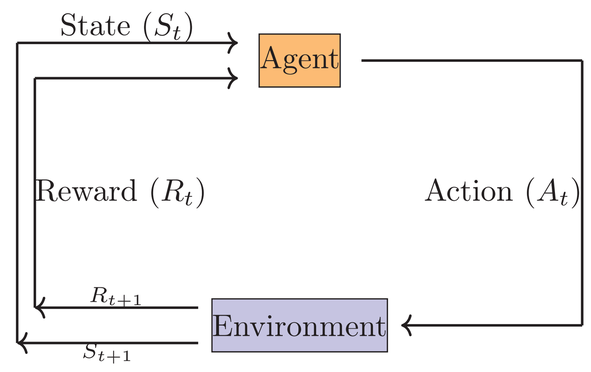
The simplified setting

We focus on the case of a single crane (the grey cell in the representation below) that unloads the products from an incoming train ($y = 0$), stores them in the yard and loads a roller table ($y = 2$) according to a scheduled feeding sequence. These operations have to be completed by the cranes in the lowest time possible, and so the best configuration for positioning each slab has to be found, both in terms of space and time.



RL formulation of the problem

- **Environment**: rectangular three-dimensional matrix, where each (x, y, z)-coordinate identifies a slab; the empty positions are identified by a 0, while each slab is identified by its type $(1, 2, \dots, n_{\text{types slabs}})$.
- **Agent**: the crane.
- **State**:
  - $n_{\text{slabs per box}}$ matrices of dimension $x_{\text{dim}} \cdot y_{\text{dim}}$, each representing the filling of the z-level of the slab yard, for $z = 0, \dots, n_{\text{slabs per box}} - 1$;
  - one $x_{\text{dim}} \cdot y_{\text{dim}}$ matrix of zeros, except for the (x, y) position in which is the crane, where we put the value $n_{\text{types slabs}}$;
  - a one-dimensional vector storing the desired output sequence.



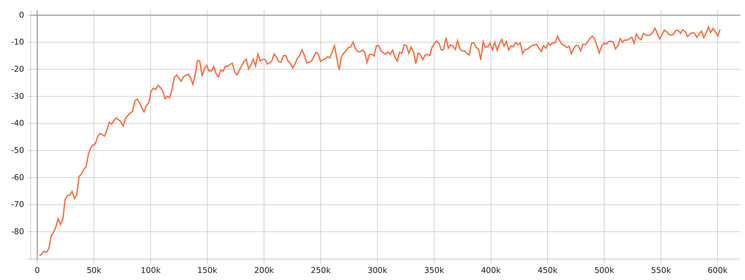
- **Actions**: represented by a four-dimensional vector $(x_{\text{start}}, y_{\text{start}}, x_{\text{target}}, y_{\text{target}})$: the crane moves at maximum height, firstly in x-direction, then in y-direction, hence it goes down in order to load the slab in position $(x_{\text{start}}, y_{\text{start}})$. Then it is lifted, and the same operations are repeated to reach the box $(x_{\text{target}}, y_{\text{target}})$, where it unloads the slab.
- **Rewards**:

$$\begin{cases} -1 & \text{if the crane performs an invalid action} \\ +0.5 - e^{\text{time}} & \text{if the crane delivers a correct slab on the output roller table} \\ -0.3 - e^{\text{time}} & \text{if the crane delivers a wrong slab on the output roller table} \\ -e^{\text{time}} & \text{for any other admissible action} \end{cases}$$

where *time* is the time needed by the crane to perform the movement.

The environment is constructed using the Gymnasium interface [3], so it is standardized to be used with almost any RL algorithm.

Ongoing studies and future developments



Mean of the reward (y-axis) every 2048 timesteps (x-axis) of the learning process.

We apply Proximal Policy Optimization (version implemented by Stable Baselines3 [1]) to the above problem.
The following issues are currently under consideration:

- finding the optimal values of the hyperparameters used by the algorithm (learning rate, clipping factor, discount factor, ...);
- investigating how the learning process changes depending on the value of the above hyperparameters;
- finding the best structure of the Actor Network and the Critic Network;
- representing the slab yard as an image, in such a way to structure the Actor Network and the Critic Network as Convolutional Neural Networks.

References and Acknowledgements

[1] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[3] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis. Gymnasium, Mar. 2023.

This work is partially supported by an industrial Ph.D. grant sponsored by Danieli Automation S.p.A.