RL in Elevator Dispatch

Course Project Mid-term Report
Reinforcement Learning and Decision Making Under Uncertainty,
University of Neuchatel, Spring 2025

William Dan*
University of Bern
william.dan@students.unibe.ch

27 April 2025

1 Current Progress

I choose to implement event-driven elevator group simulator [Wei+20]. The current progress can be seen in the public repository on github [Dan25].

2 Elevator Dispatch Simulator

There are 3 types of event corresponding to a step: passenger spawning, door opening and door closing. The core logic is shown in algorithm 1[1]. The state space and action space are shown in the table 1[1] and table 2[2]. Note that the state space is slightly different from that in the initial proposal.

Algorithm 1 Event-Driven Elevator Group Simulator

procedure ADVANCEUNTILNEXTEVENT
    $t_{\text{spawn}} \leftarrow$ NextSpawnTime()
    $t_{\text{car}} \leftarrow$ NextCarEventTime()
    $\Delta t, evt \leftarrow$ EarlierOf(t_{spawn}, t_{car})
    AdvanceClockAndCars(Δt)
    if *evt* = SPAWN **then**
        SpawnPassengers()
    end if
    if *evt* = DOOR_OPEN **then**
        HandleArrival(CurrentCar)
    end if
    if *evt* = DOOR_CLOSE **then**
        FinalizeDoorClose(CurrentCar)
    end if
    return SnapshotReward(), evt
end procedure

*University of Bern, 3012 CH-Bern, Switzerland.

Table 1. MDP state representation for an N-floor, M-car elevator group

Symbol	Shape	Value / Type	Description
$\bar{B}$	$N \times M \times 2$	$\mathbb{R}_{\geq 0}$	*Hall-call matrix* after direction replication and truncation; $\bar{b}_{i,j}$ is cumulative (or binary) wait of callers at floor i in the travel direction of car j.
A	$N \times M$	$\{0, 1\}$	*Car-call matrix*; $a_{i,j} = 1$ iff a passenger in car j wishes to alight at floor i.
p	M	$\{1, \ldots, N\}$	Discrete current floor index of each car.
d	M	$\{-1, 0, 1\}$	Travel direction of each car (-1 down, 0 idle, $+1$ up).

The four tensors are stacked as $(\bar{B}, A, P, D)$, making a 3-D array of shape $(N, M, 5)$ which is then flattened to a feature vector of length $5NM$ for the neural policy.

Table 2. MDP action space

Symbol	Domain / Encoding	Cardinality	Interpretation
$a = (i, j)$	$i \in \{1, \ldots, N\},\ j \in \{1, \ldots, M\}$	$\lvert\mathcal{A}\rvert = N \times M$	"Dispatch car j to stop next at floor i" (subject to non-reversal and capacity constraints).

3 Metric Design

At the end of each step, the environment will return a reward. The reward is relevant to the waiting time of all the passengers in the hall, the riding time of all the passengers in the cars.

Passenger clocks.

$$\forall p \,:\, w_p(t) = t - \tau_p^{\text{req}} \quad r_p(t) = t - \tau_p^{\text{board}}$$

Instantaneous cost.

$$C(t) = \sum_{p \in \mathcal{W}(t)} \big[w_p(t)\big]^2 + \sum_{p \in \mathcal{B}(t)} \big[r_p(t)\big]^2$$

Simulator reward (per event).

$$R(t) \;=\; -\,C(t)$$

4 traditional approaches

4.1 FIFO

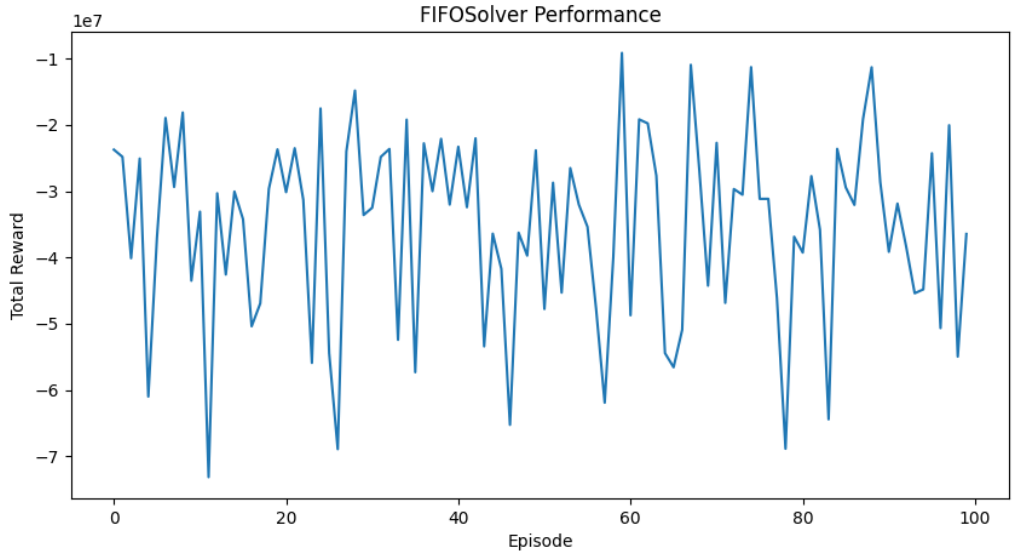

Figure 1. FIFO performance: average reward = -3.5 * 1e7

4.2 LOOK

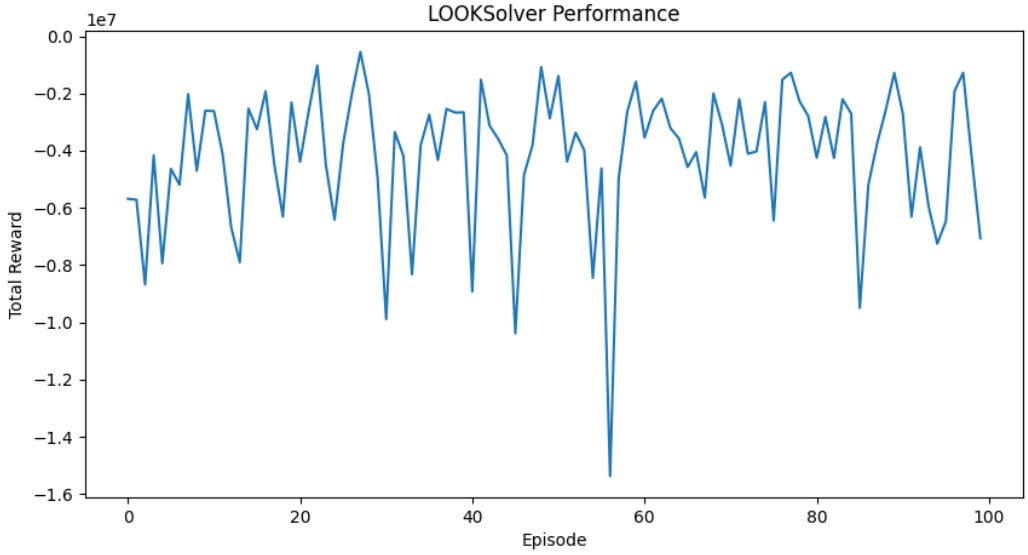

Figure 2. FIFO performance: average reward = -4.1 * 1e6

5 RL approaches

5.1 Policy Gradient with Baseline

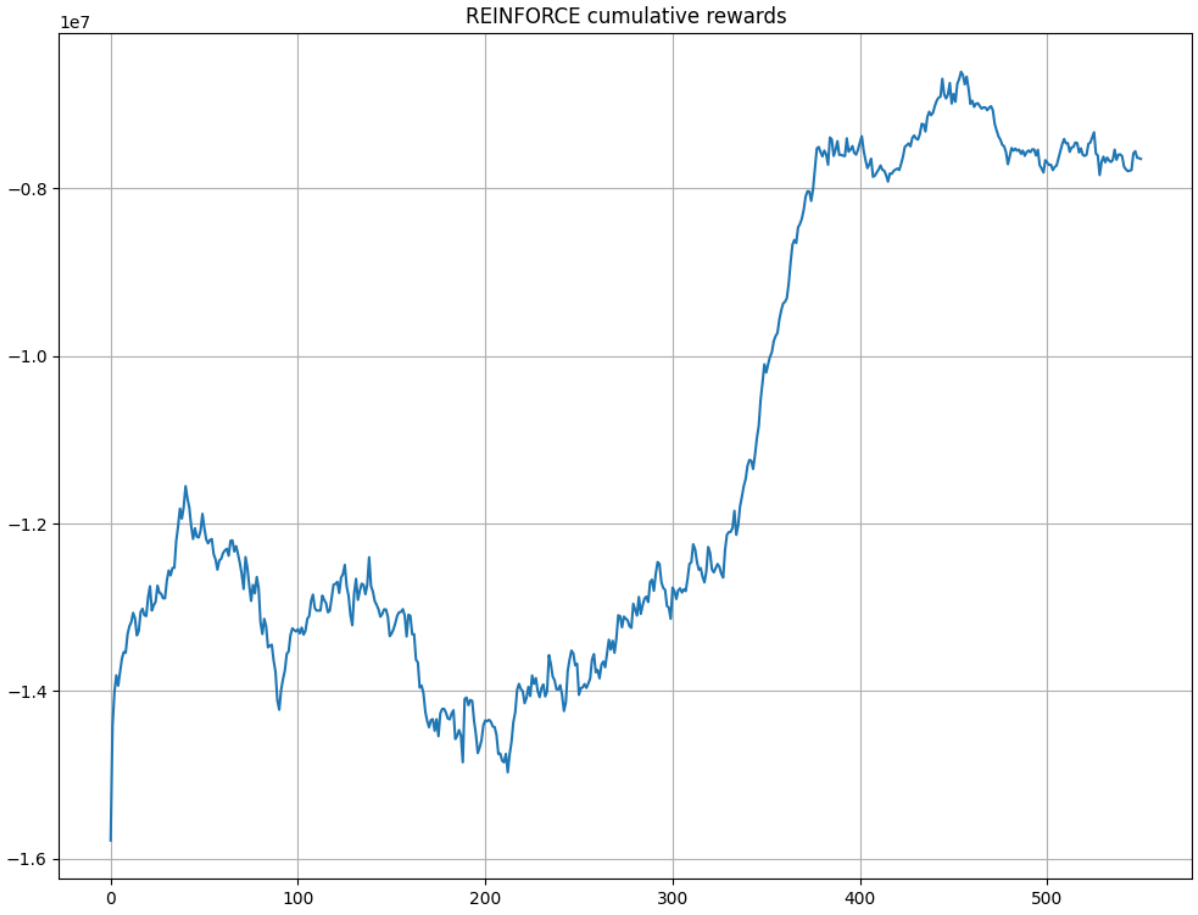

Figure 3. PG training result

Figure 4. PG performance: avg reward = -7.3 * 1e6

5.2 1-step Actor-Critic

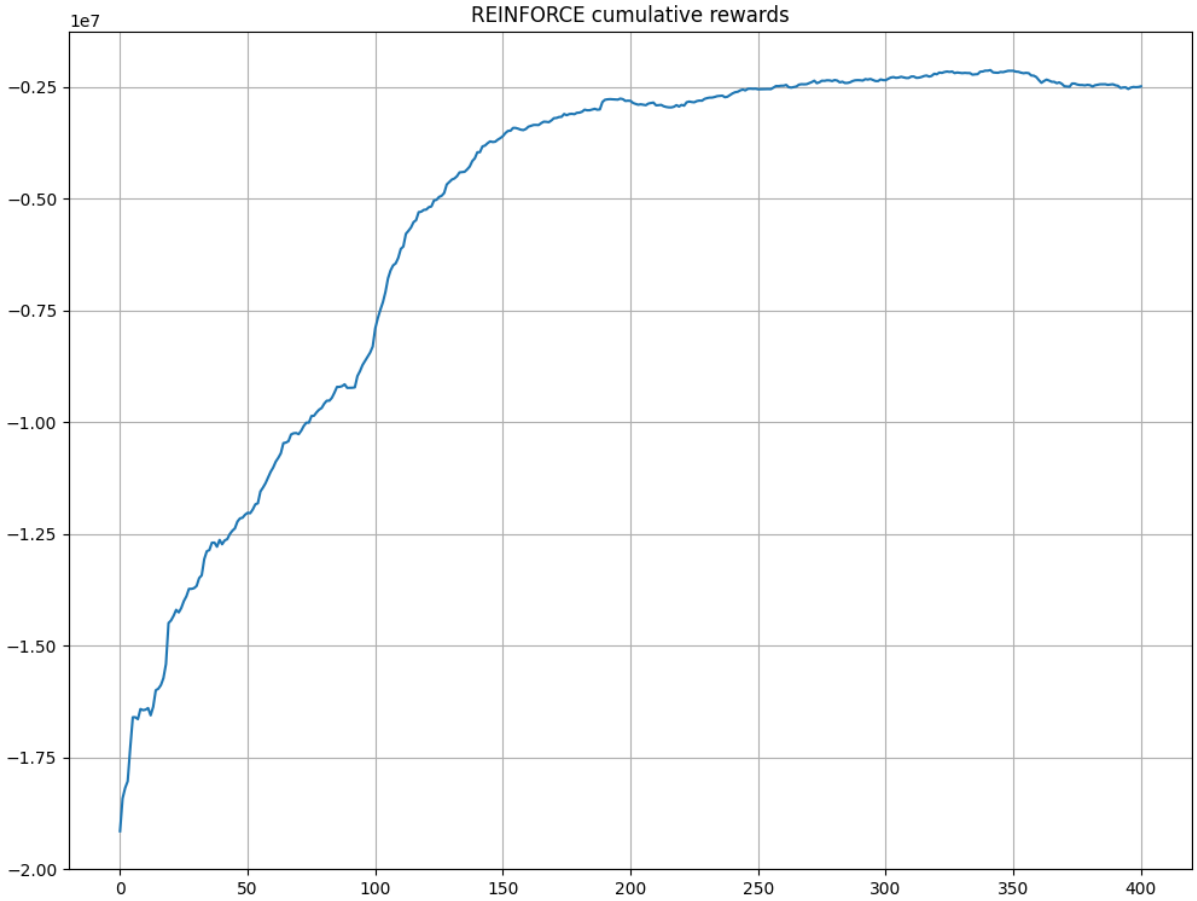

Figure 5. AC training result

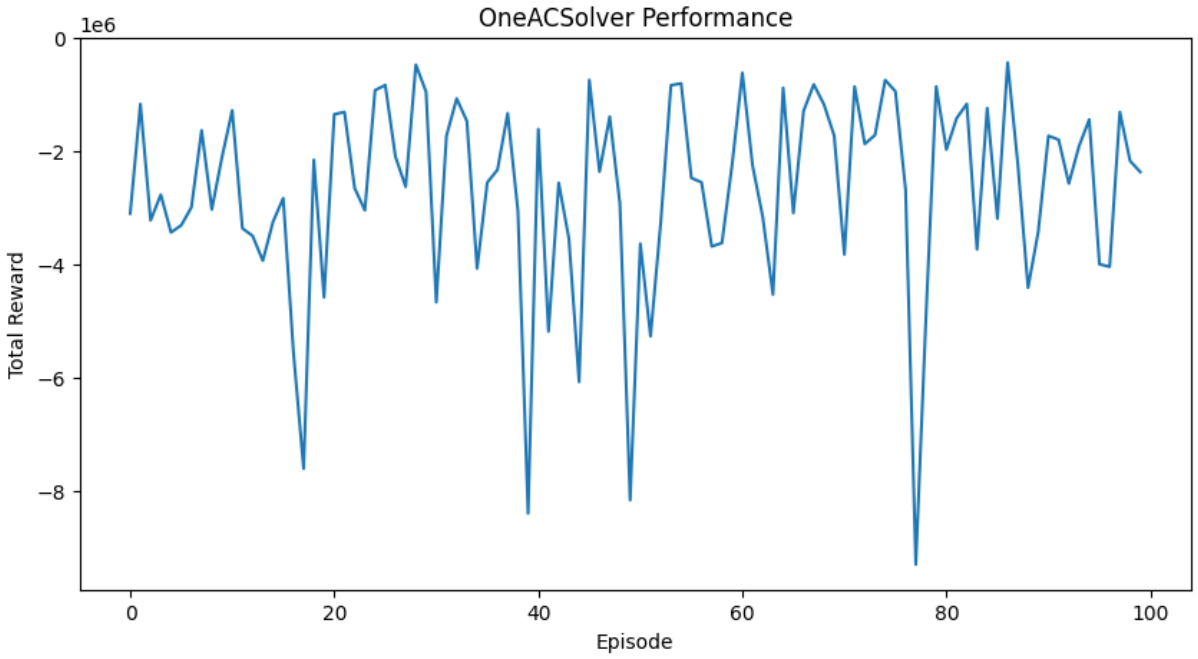

Figure 6. AC performance: avg reward = -2.6 * 1e6

6 TBD

6.1 add feature of Elevator Capacity

Currently, we assume all cars have infinite capacity, which is not true in the real case. I will implement the feature in the future if there's time.

6.2 more Metrics to unleash the power of RL

Currently, we only have waiting time and riding time as metrics. In other words, we focus on the efficiency. In this regard, RL approaches cannot show significant advantage over traditional approaches. I will design more metrics such as energy cost (namely pursuing shortest traveling distance), the maximal waiting time (minimize it) etc.

6.3 specific passengers spawning distribution

Currently, I use two exponential distribution. One is for choosing the next spawning time. The other is for deciding the number of passengers generated this time. I want to try some different types of distribution to see what will happen.

References

[Dan25] W. Dan. *elevator-gym. An RL gym environment for the elevator-dispatch problem*. Version latest. If you use this software, please cite it. Apr. 27, 2025. URL: https://github.com/william-dan/rl-elevator.

[Wei+20] Q. Wei, L. Wang, Y. Liu, and M. M. Polycarpou. "Optimal Elevator Group Control via Deep Asynchronous Actor-Critic Learning". In: *IEEE Trans. Neural Networks Learn. Syst.* 31.12 (2020), pp. 5245–5256. DOI: 10.1109/TNNLS.2020.2965208.