In-Station Train Dispatching: A PDDL+ Planning Approach

Matteo Cardellini,[1] Marco Maratea,[1] Mauro Vallati,[2] Gianluca Boleto,[1] Luca Oneto[1]

[1] DIBRIS, University of Genova, Italy, firstname.lastname@unige.it
[2] School of Computing and Engineering, University of Huddersfield, UK, m.vallati@hud.ac.uk

In-Station Train Dispatching Problem

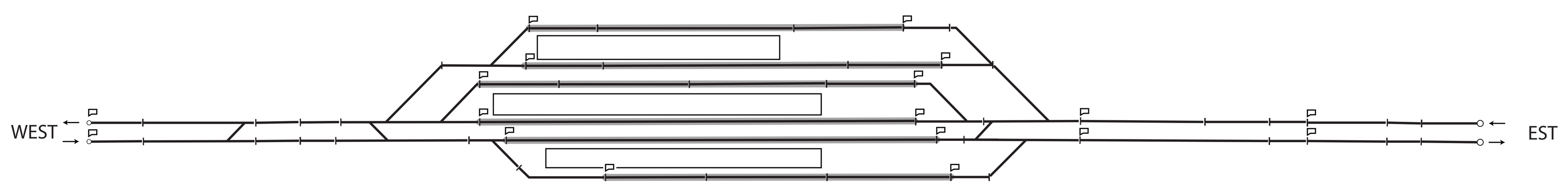

Figure 1: Schematic representation of part of the Italian railway station we use in our experimental analysis

Main components of a station:

- *track segments*: the minimal observable rail units.
- *entry points* and *exit points*: allow the train to enter or leave the station.
- *itineraries*: sequences of connected track segments manually grouped by experts of the specific railway station.
- *platforms*: set of track segments in which a train can stop for embark/disembark goods or passengers.
- *timetable*: timings in which trains are expected to arrive at the controlled station.

Constraints to the movements of train in the station:

- A track segment can be occupied by a single train at the time.
- A train is required to *reserve* an itinerary, and this can be done only if the itinerary is currently not being used by another train.
- While a train is navigating the itinerary, the track segments left by the train have to be released in order to avoid congesting the station.

Types of trains in station:

- *Transit/Stop*: the train is moving through the controlled station from one entry point to an exit point without making a stop.
- *Origin*: the train is initially at a platform and it is required to leave the controlled station via a specified exit point.
- *Destination*: the train enters the station via a given entry point, and after reaching a platform it will end its trip.

Problem: Given a railway station infrastructure, the position and arrival times of trains at the station, find a route for every train that allows to respect the provided timetable, as much as possible.

Control flow with PDDL+ constructs

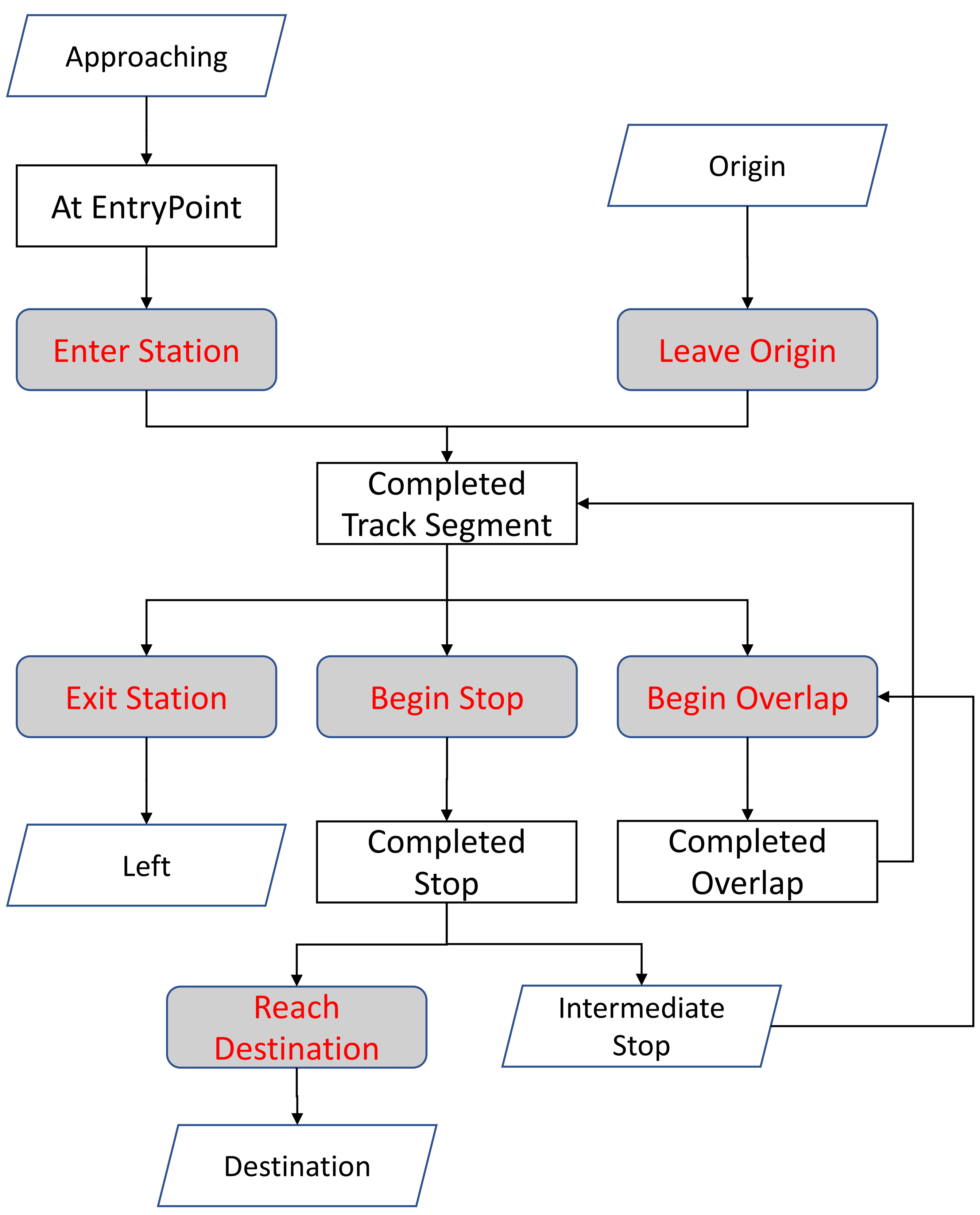

Figure 2: Valid state transitions for a train in a controlled railway station. Parallelograms indicate train states, grey rectangles indicate operators, and white rectangles indicate events. Edges indicate valid sequences. Processes are omitted for readability.

Forward Heuristic Search for Train Dispatching

The planning engine ENHSP was specialised in three ways:

- **Adaptive Delta**: in the proposed model, it is possible to know a priori when events will be triggered. So there is no need to use a fixed execution and planning delta, but the delta value can be adjusted according to when the next event will be triggered, or an action will be available.
- **Specialised Heuristic**: following the traditional A* settings, the cost of a search state s is calculated as $f(s) = g(s) + h(s)$. In our specialisation, $g(s)$ is calculated as the elapsed modelled time from the initial state to s. The $h(s)$ is a heuristic calculated as
$$h(s) = \sum_{t \in N(s)} (\text{reachTime}(t) + \text{penalisation(t)})$$
where $N(s)$ is the set of trains of the given problem that did not yet achieve their goals at s. The reachTime method measures the time that the considered train needs to reach its final destination. The penalisation method gives a very high penalisation value P for each goal that has not yet been satisfied at state s.
- **Constraints**: The idea behind such constraints is to avoid situations where trains are left waiting for long periods of time, occupying valuable resources. The times limited are: (a) the time spent by a train in the station (b) the time passed from the arrival of the train in the station and (c) the time spent stopping at a platform. The maximum times are calculated a priori, according to historical data, and depends on the structure of the station.

Evaluation

We tested the proposed PDDL+-based approach by exploiting the real-world data coming from the Italian railway network and provided by RFI.

Validation. It consisted in successfully assessing that the PDDL+ formulation is capable of representing how the addressed in-station train dispatching problem is currently faced by the human operators. Validation consisted of: (i) comparison with what would be expected in a common sense solution, by the visual inspection of the generated plans by a domain expert, and (ii) automated validation of the historic human-made plans using the planner's validator.

Minimisation of Delays. This evaluation focused on the benefits that the proposed approach can deliver when dealing with delayed trains. Results are:

- comparing it to human-made plans, our approach would have never caused delays but, instead, it would usually have been able to reduce them.
- running simulations and artificially increasing the trains' delays on their schedule, our proposed approach was able to plan the movement of trains in order to absorb 28% of the delay in entrance to the station.

Increment of the Railway Station Capacity. A stress test was performed with the aim of understanding if the proposed approach can lead to an increment of the railway station capacity. An increasingly large number of synthetically generated trains were added to the simulation. Results show that adding up to 25% more trains does not result in a reduced capacity of our approach of mitigating the injected delay. Consequently, the proposed approach seems to be very robust.

Importance of the Domain-Specific Extension. Figure 3 shows the impact of the designed domain-specific extensions on ENHSP:

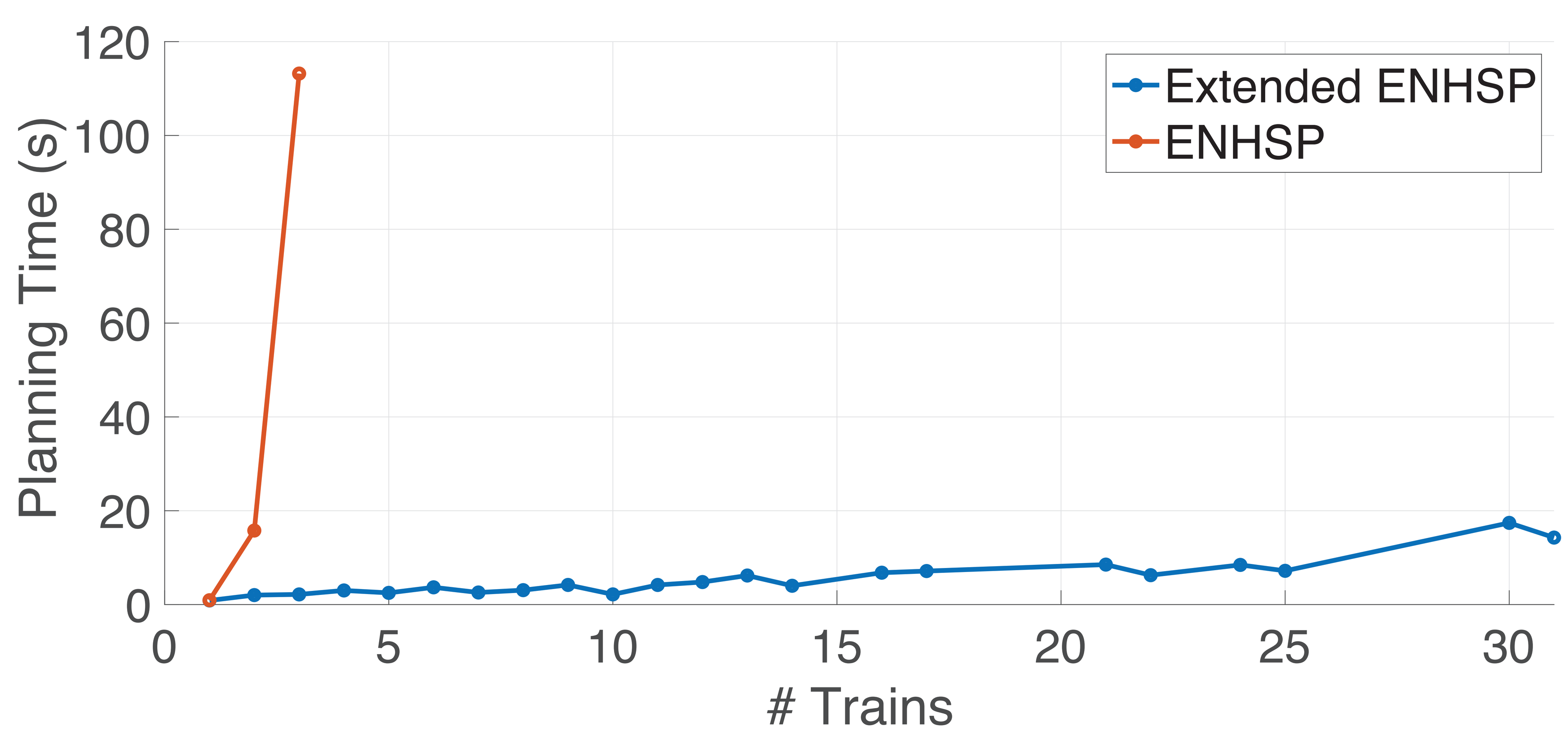

Figure 3: The CPU planning time of the ENHSP planning with extensions vs without