Intelligent and adaptive asset management model for railway sections using the iPN method

Ali Saleh[a,b,*], Rasa Remenyte-Prescott[c], Darren Prescott[c], Manuel Chiachío[a,b]

[a]*Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071, Spain.*
[b]*Dept. Structural Mechanics & Hydraulics Engineering, University of Granada, 18071, Granada, Spain.*
[c]*Resilience Engineering Research Group, The University of Nottingham Science Road, Nottingham, University Park, NG7 2RD, United Kingdom.*

Abstract

The maintenance strategy in railway transportation is crucial in ensuring safety, availability, and reducing operating costs. However, finding the optimal maintenance plan that takes into account the complex relationships between railway assets can be a challenging task. To address this challenge, this study introduces an Intelligent Petri Net (iPN) model to effectively consider the maintenance and operation of railway sections with a focus on optimising ballast maintenance. The iPN model merges Petri net (PN) with Reinforcement Learning (RL) to create a model that is able to simulate and learn at the same time. The model is able to use diverse information, including usage, degradation rates, maintenance effectiveness, fault probabilities, and maintenance time, to simulate and learn at the same time. By considering the interconnections between these factors, the model found that reducing unnecessary maintenance actions increases the age of railway sections and leads to higher net profits. The study also introduced a method to reduce computational effort by dividing the PN into subnets and another method to make learning faster by using multiple RL environments. In conclusion, the developed iPN model presents a promising solution for optimising ballast maintenance within railway operation.

Keywords: Petri net, Reinforcement learning, Q-learning, railway, maintenance modelling, degradation models

1. Introduction

Railway systems serve as the backbone of modern transportation, facilitating the movement of goods and people across vast distances with efficiency and reliability. However, the seamless functioning of these intricate networks relies on a complex interplay of various components, each of which demands special attention. One such important element is the ballast, the layer of crushed stones beneath the tracks that provides stability, distributes load, and facilitates water drainage. The optimization of ballast maintenance

*Corresponding authors. e-mail: alisaleh@ugr.es

is an endeavor of paramount significance, as it directly influences the safety, performance, and sustainability of the entire railway system.

The settlement of ballast and other underlying layers, such as the formation, causes the deterioration of railway track geometry. As the track geometry degrades, the track becomes uneven and both ride quality and safety are affected. Track maintenance can address degraded track geometry but poor maintenance planning can lead to levels of degradation that are sufficiently high for the railway to become unfit for purpose. At best, this can lead to high monetary losses due to downtime and corrections, and at worst, to safety risks including fatalities and injuries to the public, users, and workforce. An example of catastrophic failure is the Potters Bar train derailment, which resulted in 7 fatalities, 76 injuries, and a £3,150,000 fine for Network Rail [1]. This is why renewal and maintenance planning in the railway is a critically-important decision-making problem for engineers and a crucial topic for ongoing research.

Railway track settlement is directly affected by ballast quality. Good ballast should be free of dust, dirt, or small particles [2]. Ballast is fouled when it contains small particles, and this can lead to several undesirable consequences. Fouling impedes the water drainage path, which results in a saturated subgrade and permanent deformation of the soil, and can lead to wet beds that increase rail and sleeper degradation rates. Ballast fouling also impacts the distribution of loads, affecting the settlement within the subgrade, and degrading the vertical track geometry profile. Causes of ballast fouling include the breakdown of ballast particles due to dynamic forces from traffic and maintenance activities. Once the ballast becomes highly fouled, maintenance actions are no longer effective and renewal should take place. Before renewal, it is important to avoid unnecessary maintenance actions because of the ballast breakdown that they induce. This can be achieved by following a condition-based maintenance strategy, which calls for maintenance only when needed. However, the challenge is to know when and what type of maintenance is essential for the different known system conditions.

The connection between the operational life of the ballast and maintenance actions is apparent due to how they are interrelated. Maintenance actions, such as tamping or stoneblowing, are carried out to correct the rail's vertical geometry profile, but they also have an impact on the ballast's condition. These maintenance actions might lead to an increase in the rate of degradation and a reduction in the ballast's lifespan. This reduction happens because the ballast can only go through a certain number of maintenance actions before it becomes highly fouled and needs replacement. Consequently, a thoughtful assessment is required to determine whether restoring the vertical geometry in each situation is preferable or if postponing such actions would be a better choice to extend the ballast's lifespan. The objective of this study is to identify the most suitable course of action, whether it involves tamping, stoneblowing, renewal, or no intervention, based on the specific conditions and the track's maintenance history while reducing maintenance costs.

Several studies have considered the problem of optimizing the ballast maintenance [3–5]. The basis for estimating the appropriate time for condition-based maintenance (CBM) interventions in railway is the track degradation modeling [6]. Degradation models can be classified into physics-based, data-driven, and hybrid models [7]. Physics-based models, which estimate degradation based on mechanical properties of track components, offer adaptability to different traffic conditions and materials, particularly in the early stages of the life cycle when historical data is limited [7]. However, these models are deterministic and often overlook input uncertainties [8]. To address this limitation, hybrid approaches have been proposed, such as the elastoplastic physics-based model combined with a sequential model that considers parameter uncertainty [7]. On the other hand, empirical models, particularly stochastic models, have been used to address uncertainties in railway asset degradation behavior and model parameter estimation. These models utilize stochastic processes, such as the Weibull distribution, Wiener process, Gamma process, and Petri net (PN), to represent degradation rates of track components and predict track deterioration [9–13]. Furthermore, Markov chain approaches have been employed to model the outcomes of transformations between different states of track degradation, incorporating maintenance operations and discretized levels of degradation progression [14–16]. Recent advancements in artificial intelligence and machine learning have also gained attention in railway transportation, with applications in predictive maintenance, condition monitoring, and track fault prediction [17, 18].

Among the mentioned approaches, PNs are typically regarded as powerful modelling tools for degradation and maintenance modelling due to their ability to account for resource availability, concurrency, and synchronisation, which are common aspects that underline the majority of the asset management models, along with their adequacy for dealing with highly multidimensional and heterogeneous input variables [19, 20]. However, ordinary PNs do not have learning capabilities, and that limits the capacity to autonomously adapt the resulting maintenance schedule to the changing nature of the influencing conditions. Several efforts were made to enrich the PN model with learning capabilities to allow it to be used for optimisation problems. Plausible Petri nets [21], which are based on Bayesian learning, are effective in making the PN self-adaptive, but they are limited to homogeneous and low-dimensional variables. Possibilistic Petri nets [22] and fuzzy Petri nets (FPN) [23, 24] can be viewed as knowledge representation formalisms, but their intelligence is limited to adjusting fuzzy production rules using soft-computing techniques. More recently, a novel methodology referred to as iPNs has been proposed and utilizes Reinforcement learning (RL) to enable and optimize decision-making and to upgrade the PN to an intelligent system [25]. The iPN method best suits decision-making problems since RL, especially temporal difference learning, can be viewed as a more general extension of dynamic programming (DP) that does not require a complete model of the environment [26].

In this study, the iPN method is used to optimize condition-based maintenance and renewal of railway ballast. Based on data from a typical European track, the method has been formulated and integrated into the iPN model. The formulas cover factors affecting ballast condition, the impact of maintenance on ballast condition and its degradation rate, and the effect of ballast condition on rail vertical geometry profile and the rate of rail faults. The case of a railway system comprised of multiple sections has been considered, with the decisions of each section evaluated independently based on its condition and other pertinent factors affecting its state. This study incorporates in its objective function the costs associated with life-cycle, maintenance and renewal, travel, delay, and traffic disruption, either directly or indirectly. Distinctive features are presented by this study compared to prior work that is presented in the literature review (Section 2). It is the first endeavor to employ the PN for detailing the operation and maintenance intricacies of the ballast and other railway components, simultaneously utilizing RL to optimize maintenance protocols. In earlier research, the emphasis was singular, either on the modeling facets or the optimization components; however, a sophisticated model combined with optimization was lacking. The result was an optimal maintenance strategy that balances safe and good condition of sections while reducing maintenance frequency to decrease costs and increase the ballast's section lifespan. The average lifespan increased from 29.5 to 42.5 years while reducing the probability of being in a *Super-red* condition from 0.04% to 0.007%. Importantly, the developed model is not restricted to the study of the railway track sections for which it was developed. It can be used to study the asset management of other railway track by adjusting input parameters and rewards, without any need for the alteration of the core features of the PN model or the associated analysis.

The *i*PN method, as outlined in [25], proposes a technique for integrating multiple decisions when simultaneous decision-making is required. This approach entails incorporating multiple RL agents with centralised decision-making, leading to convergence towards an optimal policy. However, the combination of multiple decisions results in an exponential increase in the action space, making the learning process more complex [27]. In order to mitigate this challenge, the method described in this study avoids the combination of decisions and instead treats each section as a separate RL environment with its own unique elements. This approach reduces the dimensions of both states and actions, as it eliminates the need to combine states or actions. Furthermore, it enables agents to learn from one another when they are pursuing similar goals under comparable conditions. This approach has been applied to a railway with 10 sections, resulting in 2830 states and 4 actions per state, as opposed to $3.29 \cdot 10^{24}$ states and 10^5 combinatorial actions per state. Additionally, it enables the consideration of similar states from different sections as equivalent states with equivalent actions, resulting in a reduction of the actually trained states to 283.

The current study also addresses the methodological challenge of reducing the complexity of PN models for industrial applications in transportation, which often result in high computational costs. While previous literature has proposed various techniques for reducing PN complexity, each of these approaches has its own limitations. Reduction based on defined rules is effective in avoiding logical errors but may not be sufficiently general for all types of PN structures [28–31]. Symmetrical reduction of PN can only be applied when symmetries are present [30]. Reduction through the use of algebraic equations is only applicable when the PN has specific properties such as redundant transitions, redundant places, or place agglomerations

[32]. Proposing reduced models and inferring their parameters so that the results at key outputs are similar to those of the original PN requires additional computation for the inference process [20]. To overcome these limitations, this study provides a systematic method for reducing PN complexity by decomposing it into subnets with reduced computational cost while preserving its structure and functionality. This method can be applied to any type of PN and requires minimal additional rules, leading to a 3 times reduction in computational time for the considered PN case.

The rest of the paper is structured as follows. Section 2 reviews the maintenance policies, the undertaken problems, the objective functions, and the methods used in the railway maintenance field. Section 3 presents the underlying foundations of Q-learning and Petri nets, along with an overview of the iPN model and the proposed method for decomposing a PN into multiple subnets. Section 4 introduces a technique for decomposing the RL environment into multiple environments and a method for sharing experience between RL agents within the context of the iPN. Section 5 details the creation of an operation and maintenance intelligent PN model for a railway with multiple sections. The results of the railway case study are presented in Section 6, followed by a discussion of the results in Section 7. Finally, Section 8 provides concluding remarks.

2. Literature Review on railway maintenance

2.1. Maintenance policies

A maintenance policy is a decision made by managers based on maintenance models to ensure the proper functioning of a system [15]. Maintenance policies can be categorized into three types: preventive, corrective, and improvement [6]. The objective of the corrective policy is to enhance the asset's inherent reliability, maintainability, or safety while preserving its original function. It involves repairing or replacing failed parts to quickly restore equipment [33]. However, corrective maintenance is costly and increases safety risks due to unexpected failures. Despite the implementation of preventive maintenance, unexpected failures can still occur, leading to the need for corrective maintenance. Preventive maintenance reduces failures through inspections and repairs [34]. It includes predetermined maintenance and CBM [6]. Predetermined maintenance involves regular inspections and repairs, while CBM utilizes real-time data for proactive maintenance. Predetermined maintenance can reduce the probability of disruption and system failure but can result in additional unnecessary maintenance actions. On the other hand, CBM can avoid unnecessary maintenance actions while ensuring safety and economic benefits. Predictive maintenance, a form of CBM, utilizes data analysis and predictive modeling to identify potential issues before they occur [6]. However, it requires significant resources for data collection and analysis.

2.2. Undertaken problems

The planning process in maintenance management aims to address crucial decisions regarding maintenance intervals for track segments and the allocation of necessary resources [17]. It involves ensuring the availability of required resources, determining appropriate actions, sequencing tasks, and identifying the necessary skills for maintenance operations [17]. The role of a planner, as explained by [35], includes assessing the scope of maintenance tasks, identifying the required expertise and craft, estimating the duration of tasks, and specifying the necessary parts and tools. Moreover, the planning function encompasses various aspects, such as task identification, complexity assessment, workforce estimation, spare parts and materials identification, and tool requirement determination [36]. The objective of the planning process is to make important choices regarding the timing of maintenance intervals for track segments and the allocation of necessary maintenance resources. In the railway industry, decision-making involves planning and scheduling activities, such as budgeting, quality prediction, project definition, project prioritization, possession allocation, timetabling, maintenance scheduling, and performance evaluation and feedback.

Maintenance planning involves several key aspects. One aspect is determining the timing of maintenance interventions based on accurate track condition prediction, which requires considering track geometry and track structure indices [37, 38]. However, relying solely on track geometry variables may not provide an accurate prediction of track condition [3]. To enhance maintenance planning, additional factors such as ballast fouling and geometry degradation should be considered when identifying maintenance needs [3]. Decision support systems and optimization models have been proposed to assist in the planning process. For instance, [39] developed a stochastic degradation model for condition-based maintenance (CBM) planning, while [40] optimized the number of tamping interventions considering track degradation and recovery. Furthermore, the setup cost of tamping equipment can be incorporated into the cost function [41].

Maintenance action identification and prioritization are crucial steps in maintenance planning. Railway infrastructure maintenance can be based on predetermined schedules or condition-based approaches [6]. Various optimization models have been proposed to determine the optimal maintenance limit intervals for different track quality indicators, taking into account preventive and corrective maintenance costs as well as potential train delays [42]. Furthermore, optimization models have been developed to decide whether immediate or postponed maintenance should be conducted based on factors such as reliability functions, associated costs, and identified defects [43]. Decision-making frameworks incorporating multi-attribute utility theory have also been used to prioritize maintenance tasks based on estimated conditions and multiple factors [44].

The scheduling of inspection intervals plays a crucial role in ensuring track safety and reliability while managing maintenance and inspection expenses. Optimization models have been proposed to determine inspection intervals based on safety risks and maintenance costs [45]. Rescheduling of inspection intervals has also been explored to mitigate the disruption caused by inspection scheduling and improve decision-making [46]. The close relationship between inspection scheduling and maintenance scheduling emphasizes the significance of inspection intervals in railway track maintenance planning and execution.

Possession scheduling is another important aspect of maintenance planning. Possession refers to the closure of specific sections of railway tracks for maintenance or repair work. Effective possession scheduling is essential for safe and efficient railway operations [47]. The optimization of possession scheduling can be approached from different perspectives. Some studies focus on fixed train timetables and aim to determine the best possession time for maintenance activities [48]. Others consider fixed possession times and seek to optimize the train timetable around those periods [49]. Additionally, simultaneous possession and train timetable scheduling models have been developed to optimize both train operations and maintenance activities [49]. Integrating maintenance activities and optimizing vehicle routing and crew scheduling can lead to cost savings and improved efficiency [50]. Furthermore, equipment logistics, such as transporting machinery and equipment to maintenance locations, need to be carefully planned and scheduled [51].

2.3. Objective functions

Objective functions of planning and scheduling play a crucial role in optimizing the efficiency and sustainability of railway transportation systems. One significant aspect considered in these objective functions is the life-cycle cost, which encompasses various expenses associated with the entire lifespan of railway transportation [6]. These expenses include maintenance and replacement costs, building expenses, track utilization fees, and costs related to the final stages of the system's operational life.

To enhance decision-making regarding new construction and the maintenance and replacement of track components, decision-makers utilize life-cycle cost analysis [52]. This analysis takes into account both measurable expenses such as construction, maintenance, and renewal, as well as intangible factors like quality deterioration, traffic delays, safety concerns, and environmental impacts [52]. By considering these various aspects, decision-makers can optimize investment strategies and ensure the long-term sustainability and efficiency of railway transportation systems.

Researchers have identified four components that encompass the overall expenses associated with a track and its rolling stock over its lifetime [53]. These components include construction costs, operational aspects (such as capacity loss, fuel or energy consumption, environmental impact, accident risk, and socio-economic implications), maintenance expenses, and costs incurred at the end of the track and rolling stock's life. Studies have established the life-cycle cost of railway tracks by considering both measurable and non-measurable expenses, such as maintenance, renewal activities, penalties due to track quality issues, customer losses, and damage caused by subpar quality [9, 53].

Another important factor in planning and scheduling costs is the maintenance cost. A commonly used approach is to assign a fixed cost per activity or time unit, which forms the basis for estimating the costs [41]. For example, Gustavsson [41] proposed an improved linear programming model for scheduling tamping operations on ballasted tracks, incorporating unit maintenance cost and the cost of maintenance occasions. Daddow et al. [54] utilized a similar cost formulation to calculate the cost of each unit tamping action, while Vale et al. [40] focused on reducing the number of tamping actions. Moreover, Letot et al. [10] considered a fixed cost for the tamping machine.

Renewal costs are another important aspect of track maintenance. These costs can be classified into two categories: component renewal and full track renewal [11]. Researchers have proposed optimization frameworks to determine the optimal balance between track unavailability and life-cycle cost (LCC) [11]. These frameworks consider factors such as the unitary cost of renewal work, residual value of track components, and potential savings from grouping track segments. Integrated methodologies have been developed to account for equipment preparation, setup expenses, and predetermined expenditures associated with each renewal activity throughout the lifespan of a component [55].

Possession cost is a significant factor to consider in maintenance operations. Previous studies have proposed different approaches to address possession costs and their impact on overall maintenance costs. One approach involves assigning hourly costs to account for the time required for possession in order to carry out maintenance activities [56]. Another method utilizes fixed estimated possession costs per maintenance action [57]. Train cancellations can also be taken into consideration when estimating possession costs [58]. The objective is to minimize the overall maintenance cost while taking possession costs into account.

In summary, objective functions for planning and scheduling of railway transportation systems encompass a wide range of costs, including life-cycle costs, maintenance costs, renewal costs, and possession costs. By considering these costs and optimizing decision-making processes, the efficiency, sustainability, and overall performance of railway transportation systems can be enhanced.

2.4. Search algorithms and simulation methods

A suitable approach for solving the railway track maintenance planning and scheduling (RTMP&S) problem involves considering decision-making levels, decision variables, track condition data, objectives, and constraints. Linear and integer programming methods are commonly used in RTMP&S because they can handle both continuous and integer decision variables [6]. Depending on the characteristics of the decision variables, linear or nonlinear programming can be applied. For instance, integer programming is suitable for determining maintenance actions or resource allocation.

In the case of single objective function models, mixed-integer linear programming methodologies that combine continuous and integer variables are commonly employed [6]. Commercial solvers like CPLEX, Gurobi, or FICO Xpress are often used to solve these optimization problems. In addition to these methodologies, various heuristics and metaheuristics have been employed to provide faster satisfactory solutions. These include decomposition-based heuristics, multiple neighborhood search heuristics, solution frameworks based on Lagrangian relaxation, iterative approaches with greedy and local search algorithms, tabu search heuristics, and customized metaheuristic algorithms [50, 59–62].

For multi-objective function models, researchers have proposed various methodologies to optimize conflicting or non-conflicting objectives. These methodologies consider maintenance-related unavailability, life cycle cost of track components, maintenance expenses, costs due to train delays, train maintenance planning, timetabling, and selection of maintenance strategies. Multi-objective optimization techniques like weighted sums and Pareto optimality are used to identify optimal solutions [16].

While linear programming dominates in RTMP&S, there is a growing interest in utilizing non-linear programming. Non-linear formulations and search techniques such as the steepest gradient and improved genetic algorithms have been employed for maintenance scheduling [63]. Other research directions include the utilization of Model Predictive Control techniques at various levels, which involve methods like pattern search, transformation into Mixed-Integer Linear Programming, Dantzig-Wolfe decomposition, and gradient-free algorithms [12].

Researchers have shown a growing interest in integrating simulation models with optimization engines for RTMP&S problems in recent years. Discrete event simulation is the predominant method used and offers advanced capabilities to address the complexities associated with real-world maintenance planning and scheduling problems [34]. Additionally, alternative approaches such as Monte Carlo simulation have been employed to represent deterioration and restoration of track geometry, providing insights into maintenance costs and optimal timing for interventions [42].

One widely used approach for optimizing maintenance in the railway industry is the Markov decision process (MDP) [64]. MDP methods capture the stochastic nature of the railway system, incorporating uncertainties and variability into maintenance decision-making. Probabilistic transitions between states in MDP models allow decision-makers to account for degradation, failures, and repairs, resulting in more accurate maintenance optimization. MDP-based dynamic programming excels in handling large-scale maintenance optimization problems in the railway industry. Efficient algorithms like value iteration and policy iteration compute optimal policies and value functions for complex systems. This scalability is crucial for considering numerous components and subsystems within railway infrastructure. MDP-based approaches also facilitate the development of robust and adaptive maintenance strategies. By updating the value function and policy based on changing system conditions, decision-makers can dynamically adapt maintenance strategies to factors such as traffic patterns, weather conditions, and component aging. Reinforcement learning (RL) is a promising approach for addressing complex railway industry problems. RL has been applied to rail maintenance and renewal planning, optimizing costs and risk reduction [65]. It has also been used for railway alignment optimization, minimizing construction costs while satisfying alignment constraints [66]. RL-based methods have been utilized for dynamic maintenance policies in multi-component systems with degradation and random shocks [67]. Additionally, RL combined with digital twin technology has enhanced railway maintenance efficiency, reducing maintenance activities and defects [68].

RL outperforms linear programming, non-linear programming, and MDP in railway maintenance and planning. It effectively handles uncertainties and variability, optimizing maintenance through probabilistic transitions and degradation probabilities in PN simulations. RL scales well, deriving optimal policies for complex systems. Its adaptability enables dynamic updates to maintenance strategies. Integration with advanced techniques like deep deterministic policy gradients and digital twin technology automates optimization, reduces activities, and minimizes defects. RL proves valuable for railway track maintenance planning and scheduling.

3. Methodology

This section provides the methodological background and techniques proposed in this paper.

3.1. Basics about Reinforcement Learning

RL is a goal-oriented machine learning field that teaches an *agent* the correct decisions by trial and error. Single-agent RL methods can be formulated through a Markov decision process (MDP), which is described by a tuple of $\langle \mathcal{S}, \mathcal{A}, P_d \rangle$; where $\mathcal{S}$ is the set of the *states* of the *environment*, $\mathcal{A}(s)$ is the set of *actions* available at *state* s, and P_d represents the *dynamics* of the MDP [26]. The *dynamics* is defined as $P_d = \Pr\{S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a\}$, which is the probability of obtaining *reward* r and *state* s' by taking an *action* a at *state* s. At each time step, t, the *agent* receives a *state* of the environment, S_t, and takes an *action*, A_t, following a *policy* $\pi(a|s)$, which controls the probability of taking an action a being at state s. This results at the next time step, $t+1$, in an immediate *reward*, R_{t+1}, and a change in the *state*, S_{t+1}. The goal of the agent is to find the optimal policy that maximises the long-run rewards, not the immediate reward. Long run rewards coming after a time step t are called the *expected return*, (G_t), and can be calculated as:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k \tag{1}$$

where $\gamma \in [0, 1]$ is a discount rate parameter to prevent $G_t \to \infty$ when $T \to \infty$ (known as a *continuous task problems*). On the contrary, in *episodic task problem*, the terminating time step T is a finite number, thus G_t can be calculated by choosing $\gamma = 1$.

Having a complete model of the environment dynamics, P_d, is not always feasible. Thus, model-free RL by temporal-difference learning (TDL) is widely used due to its simplicity and the minimal amount of computation [26]. In TDL, the value of each state-action pair is called the Q-Value and the whole set of Q-Values represents the outcomes of the Q-function, $q_\pi(s, a)$. Q-Values are updated in TDL method as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[G_t - Q(S_t, A_t)] \tag{2}$$

In this equation the Q-Value is updated toward a target value, which is G_t, and $\alpha \in [0, 1]$, which is the learning rate, represents how much change will be made toward this target. G_t can be calculated in several ways. Using a one-step bootstrapping technique to calculate G_t as $G_t = R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$ results in the *Q-learning* method, which is one of the earliest and most famous TDL methods [69]. Another way is to calculate it based on Equation 1, and this results in one of the incremental implementations of the Monte-Carlo RL (MCRL) method [26]. MCRL is good to use at the beginning of the learning process since it does not depend on the unconverged values of the successor states, and Q-learning is better to be used later on because it is an off-policy TDL method that allows exploration at the successor states without affecting the previous ones. In this paper, the MCRL method is used at the beginning of the learning process and the Q-learning at the end of it. To reach an optimal policy, the *ε-greedy* strategy can be used as follows:

$$A_t = \begin{cases} \arg\max_a Q(S, a), & \text{with probability (1-}\varepsilon\text{)} \\ A \in_R \mathcal{A}(s), & \text{with probability } \varepsilon \end{cases} \tag{3}$$

where $\varepsilon \in [0, 1]$ is the exploration rate. Choosing the action with the highest Q-Value is called exploitation and choosing an action randomly is called exploration. It is important to keep a balance between both steps because exploitation helps in getting more rewards and evaluating the Q-Values based on the policy that appears to be the best. On the other hand, exploration allows exploring other actions that may result in higher Q-Values than the already explored actions.

3.2. The Intelligent Petri net method

Petri nets (PN) are directed graphs with two types of nodes, which are the *places*, represented by circles and the *transitions*, represented by rectangles. The number of *tokens*, which are depicted as black dots, contained in each of the places represents the state of the system. A PN is defined as a 5-tuple $\langle \mathbf{P}, \mathbf{T}, \mathbf{F}, \mathbf{M}_0, \mathbf{W} \rangle$ [70], where $\mathbf{P} = \{p_1, p_2, \ldots, p_{n_p}\}$ is the set of places, $\mathbf{T} = \{t_1, t_2, \ldots, t_{n_t}\}$ is the set of transitions, $\mathbf{F} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$ is the set of arcs, $\mathbf{W} : \mathbf{F} \to \mathbb{N}_{>0}$ is the set of weights function, and $\mathbf{M}_0 : \mathbf{P} \to \mathbb{N}_{>0}$ is the number of tokens in each place initially, which is the initial markings.

The architecture of the PN can be summarised in the incidence matrix, $\mathbf{A} \in \mathbb{N}^{n_p \times n_t}$, which is the subtraction of the *backward incidence matrix* $\mathbf{A}^- = \left[a^-_{ij}\right]$ from the *forward incidence matrix* $\mathbf{A}^+ = \left[a^+_{ij}\right]$, where a^-_{ij} coincide with $\mathbf{W}(p_i, t_j)$, which is the weight of the arc from place p_i to transition t_j, and a^+_{ij} coincide with $\mathbf{W}(t_j, p_i)$, which is the weight of the arc from transition t_j to place p_i. The dynamics of the PN are controlled by the state of each transition, which manages the flow of tokens. Each transition has a set of input places, ${}^{\bullet}\mathbf{P}_t$, referred to as the *pre-set places*, and output places, $\mathbf{P}^{\bullet}_t$, referred to as the *post-set places*. According to the *firing rule* in ordinary PNs, a transition, t_j, is said to be enabled once the markings of all its pre-set places are equal or greater than the weights of its pre-set arcs ($\mathbf{M}(p) \geq \mathbf{W}(t_j, p)\ \forall p \in {}^{\bullet}\mathbf{P}_{t_j}$). Every enabled transition has the ability to fire, and this consumes tokens from its pre-set places and produces tokens in its post-set places equal to the weights of the arcs connecting the places to the transition. This operation can be done for all transitions together in an efficient way using the *state equation* defined by:

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{A}^T \mathbf{u}_k \tag{4}$$

where k is the time step and $\mathbf{u} = [u_1, u_2, \ldots, u_{n_t}]^T$ is the firing vector. More rules can be added to deal with the complexity of dynamic systems. In timed PN (TPN), a transition can't fire after it is enabled until a given delay, τ, passes. The value of τ can be deterministic or given by a probability density function, thus the PN is referred to as stochastic Petri Net (SPN). For high-level PN (HLPN), the logic flow is used in a wider manner by using flexible definitions of arc types, and tokens, along with transition firing rules to extend the basic formalism [71]. The HLPN definitions used in this paper are the *inhibitor arc*, which ends with a small empty circle, and the *reset arc*, which ends with a small filled circle. The *inhibitor arc* is connected from a place to a transition, and it disables the transition if the place has tokens equal to or more than the weight of the arc. The *reset place* is connected from a transition to a place, and it changes the marking of the place to a value equal to the weight of the arc once the transition fires [20].

Besides, function nodes, which are nodes with rhombus shapes are defined to perform some necessary calculations for the PN model. This definition allows modeling continuous aspects within the PN model, which is a discrete even model. A function node can come alone or after a transition. If a function is not connected to any transition, it is executed every change of state; whereas, if it comes after a transition, it is executed only when the transition fires.

To give the Petri net the ability to choose an optimum action, the intelligent PN (iPN) is used [25]. This variant introduces a finite set $\mathbf{G} = \{g_1, g_2 \ldots, g_{n_g}\}$, named *action groups*, to the PN tuple to incorporate RL in decision making. Each action group, g_i, is composed of a finite set of transitions, $T_{g_i} \subseteq \mathbf{T}$, that represent decisions within a RL environment. Accordingly, RL selects which of the transitions will be enabled based on the rules described in [25]. It is important in this approach to distinguish between the RL states and the PN states. RL states are extracted from the RL environment and PN states are based on the markings of the PN.

3.3. Decomposing the PN into multiple subnets

This section proposes a method of decomposing the PN into multiple subnets without losing any functionality of the original net in order to reduce the computational cost. For any PN, the computational cost lies in getting the firing vector, $\mathbf{u}$; whereas updating the state according to the state equation (Equation 4) is just a matrix multiplication, which is not computationally expensive. Calculating the firing vector requires checking the enabling conditions and then the firing conditions before assigning the firing state for every transition. If a PN is modeling multiple system functions, it will be known by the PN designer that groups of transitions will not be utilized for specific system states. Accordingly, it is possible to avoid checking

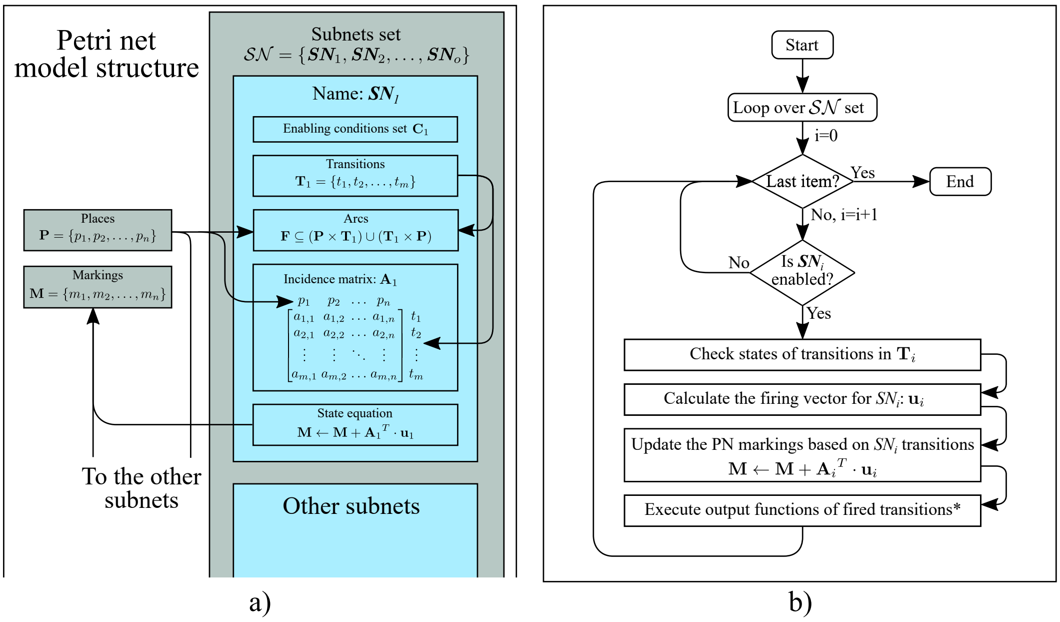


Figure 1: Panel a): PN structure. Panel b): Flowchart explaining the algorithm to update the state of the PN, when the PN is decomposed into multiple subnets.

these transitions to reduce computational costs without affecting the results because it is known that they will be disabled either way. To do so, this paper proposes the decomposition of the PN into subnets with each net having a set of conditions that enables it. Then, transitions in each subnet will be checked only after the subnet is enabled. In this new implementation, the places and their markings will be kept in the main net while all other information will be in the subnets. Accordingly, the PN structure is described by the tuple $\langle \mathbf{P}, \mathcal{SN}, \mathbf{M}_0 \rangle$ with $\mathcal{SN} = \{\boldsymbol{SN}_1, \boldsymbol{SN}_2, \ldots, \boldsymbol{SN}_i, \ldots, \boldsymbol{SN}_o\}$ being the set of subnets as shown in Figure 1a. Then, any subnet, $\boldsymbol{SN}_i$, will be described by a tuple $\langle \mathbf{T}_i, \mathbf{G}_i, \mathbf{F}_i, \mathbf{W}_i, \mathbf{C}_i \rangle$, with $\mathbf{G}$ being the set of the action groups that exist only if iPN is used, and $\mathbf{C}_i$ the set of conditions that enable the subnet. Since places remain common and are stored in the main net, every subnet will have an incidence matrix based on the connections between its own transitions and the places of the main net. This can be built in the same way as for an ordinary PN, but while considering only the transitions of the subnet. Based on this, the dynamics of the system will be described based on some additional rules shown below, and the process to update the state of the PN by calculating its markings is shown in the flowchart of Figure 1b:

- a subnet is enabled if it satisfies all its enabling conditions.
- if the subnet is enabled, all its transitions (and action groups in case of iPN) are checked, the firing vector of the subnet is calculated, and the state equation (Equation 4) is applied to update the markings of the main net based on the firing vector and incidence matrix of the subnet.

Remark. *The incidence matrices of subnets will contain many zero columns because the transitions of each* *subnet do not have connections with all the places. One might think that distributing places on the subnets or creating the incidence matrices of the subnets based only on the existing connections would be helpful to avoid these unnecessary connections and improve computational efficiency. However, once the places are distributed, there will be nothing to connect the subnets together, and this will require defining additional rules to solve this issue, which can add complication and computational cost. On the other hand, creating the incidence matrices based only on the active connections will change the dimensions of these matrices. This is like avoiding some places in each subnet, so it will be necessary to store the set of avoided places in each subnet to update only the markings of the included places every time the state equation (Equation 4) is applied inside the subnet. This will result in additional computational costs that will be in most cases greater than the cost of multiplying by columns of zeros.*

4. Extension of the iPN method for complex environments

This section provides a description of how the RL environment is divided into multiple environments to reduce the combinatorial state-action spaces. Also, it explains how experience can be shared among agents of different environments in the scope of the iPN and through action groups. These two ideas make the learning process faster and reduce computational costs.

4.1. Dividing the RL environment to multiple environments

Multi-agent Reinforcement learning (MARL) methods are concerned with the cases of multiple agents interacting in the same environment. These methods can be cooperative, where the agents try to achieve a common goal, or competitive, where they try to compete to see who achieves more. In some cases, a mixed environment can be created, where agents form groups, cooperating within each group and competing against other groups. In this study, the main focus is on cooperative MARL methods to optimise systems with multiple tasks.

MARL differs from single-agent RL in that the environmental state and reward function that each agent receives is a function of the joint actions of all agents. For this, each agent has to consider other agents' actions in addition to the environment. The process of taking multiple decisions is usually modeled through a stochastic game [72], also known as a Markov game [73]. A stochastic game is a multi-decision extension of the MDP and can be described by the tuple $\langle \mathcal{S}, \mathcal{A}, P_d, \mathcal{R} \rangle$; where $\mathcal{S}$ is the set of the *states* of the *environment*, $\mathcal{A} = \mathcal{A}^1 \times \ldots \times \mathcal{A}^n$, where $\mathcal{A}^i$ is the set of agent i *actions*, $P_d = \Pr\{S_{t+1} = s' | S_t = s, \mathbf{A}_t = \mathbf{a}\}$ is the transition probability function from state s to state s' in the next state while taking the joint action $\mathbf{a}$, and $\mathcal{R} = \mathcal{R}^1, \ldots, \mathcal{R}^n$, with $\mathcal{R}^i = \Pr\{R_{t+1} = r' | S_t = s, \mathbf{A}_t = \mathbf{a}, S_{t+1} = s'\}$ is the reward probability function for agent i after transitioning from state s to state s' while taking the joint action $\mathbf{a}$. Accordingly, each agent will have a Q-Value function of the state and the joint action, and reaching a globally optimum policy requires coordination between agents [74]. However, considering joint action results in an exponential increase of the action space at each state [27]. Besides, trying to optimise several agent's policies in one problem requires the definition of an environment that considers the important aspect for every agent, which results in a great increase in the state space. The huge state-action spaces make the computational costs extremely high for already complex problems.

The necessity to use MARL methods is when agents cooperate in the same environment [27]. Indeed, there exist some problems where it is possible to divide the environment into several sub environments, but without being able to divide the problem into multiple independent optimisation problems. This happens if the conditions and decisions of each environment do not directly affect the rewards of other environments, but can affect the transition probabilities or other aspects. In this paper, the RL environment is decomposed into multiple environments with each one having its own states, available actions, reward functions, and a single agent. Thus, the problem drops back to single-agent RL, but with multiple interacting environments. Environments can intersect to keep some information commonly available to all the agents, and in this way, agents can cooperate explicitly.

4.2. Sharing agents experience through similar action groups

A key aspect of problems with multiple agents is the allowance for experience sharing between agents that are solving similar tasks to learn faster and better [74]. This still applies to the case of multiple agents optimising multiple environments if these environments share similar characteristics.

Multiple environments may require the same decisions at similar conditions if these environments are similar. For example, if two agents are optimising the maintenance of two identical components while considering each component as a separate environment of a system, it is expected that the two components require the same action if they were in the same conditions. This means that the two components can follow the same policies for the same decisions. In the RL formulation, the policy is directly related to the Q-Values. Thus, a way to exchange information between agents of similar environments is to assign the same Q-Values for similar actions. By doing this, any update in the Q-Value of an action in any environment will cause the update of similar actions in the other environments.

In the case of iPN, actions are equivalent to enabling transitions inside an action group. Each time an action group is enabled, the agent receives a representation of the environmental state. If the state is new, it is created automatically with its available actions in that environment, and by this, the RL environment is populated by the states and actions. To link the Q-Values of similar actions, the following steps can be performed:

- let a similar group set, $\mathcal{SG} = \{g_1, g_2, \ldots, g_n\}$, be a set of action groups that require similar policies and represent similar decisions.

- all the action groups should have the same number of ordered transitions that represent *similar actions*. Thus, every action will have similar actions in similar action groups.

- if a decision is required in g_l and the state, S, is new, create a state in the environment of each action group in $\mathcal{SG}$, and not only in the environment of g_l.

- any action, $A_{l,i} \in g_l$ will have the same Q-Value as its similar actions in the other action groups: $Q(S, A_{1,i}) = \ldots = Q(S, A_{l,i}) = \ldots = Q(S, A_{n,i})$

- any time $Q(S, A_{l,i})$ is updated, all the Q-Values of similar actions are updated.

5. Case study: optimising the maintenance of ballast in multiple railway track sections for optimal railway operations.

In this section, an iPN has been developed to model various aspects of a railway consisting of ten railway sections while optimizing the maintenance of its ballast. Each section has a length of three position keys, where a position key is a 220-yard length of track known as a Poskey. The track speed of the considered section is less than 20 MPH, the annual usage is 20 Equivalent Million Gross Tonnage (EMGT), and all sleepers are of small concrete type.

The maintenance of the sections is assumed to be carried out through two types of maintenance actions: condition-based maintenance and opportunistic maintenance. For condition-based maintenance, the decision is taken after updating the condition of the section following each inspection, which is assumed to be periodic every six months. On the other hand, opportunistic maintenance decisions can be made for a section when maintenance equipment is available on-site to perform maintenance for any other section.

Once a decision is made to repair a section based on its condition, the maintenance team prepares for the maintenance, travels to the site, and performs the required maintenance. If the equipment is available on-site, a decision to repair other sections can be made, referred to as opportunistic maintenance. This maintenance approach allows repairing multiple sections at once to save preparation and travel costs, even if these sections do not urgently require maintenance based on their conditions.

The maintenance actions for sections are performed in series, starting from the first section and ending with the last one, assuming that only one maintenance team will perform the required work. Each section can be in one of five conditions denoted as 'E' (Excellent), 'VG' (Very-good), 'G' (Good), 'P' (Poor), and 'SR' (Super-red). These conditions reflect the safety and stability of the railway, and they are directly related to the standard deviation of the vertical geometry profile, as will be seen in the next section. Experts suggest that the "super-red" condition should be avoided at all costs. This is because it poses a significant safety risk to railway operations and can lead to speed restrictions, causing train delays and potential fines. It is, therefore, crucial to take preventative measures to minimize the likelihood of encountering this condition, as it can have negative impacts on both safety and efficiency.

The problem is to find the optimal maintenance decision for each condition of each section. Two policies, named policy A and policy B, were proposed as base cases to find the optimal maintenance decisions. Policies A and B allow maintenance to be performed each time the 'Very-good' and the 'Good' conditions are reached respectively. Also, both policies follow the same sequence of maintenance actions, which is 7 tamping actions, 3 stoneblowing actions, and then renewal. The third policy is optimised by using RL, with the details of the RL inputs as described in Section 5.3. For this study, it is assumed that tamping becomes ineffective after 7 actions, and stoneblowing becomes ineffective after 3 actions. Thus, the maximum number of allowed tamping and stoneblowing actions are 7 and 3 actions respectively. Besides, it is not allowed to perform tamping after a stoneblowing action is performed. This will prevent stoneblowing actions from being before tamping to follow what is done in reality. Before introducing the PN model, the following section provides an introduction to railway modelling and the formulas used.

5.1. Track degradation and maintenance modelling

The railway consists of several interacting assets, mainly Plain Line (PL) track and Switches and Crossings (S&Cs). These are made up of components that have different degradation, inspection, and maintenance mechanisms. The track is made up of the rail to provide guidance and a smooth running surface, sleepers to support the rail at the correct gauge and inclination and to transmit loads to ballast, rail pads to provide electrical insulation and distribute loads on the sleepers, ballast to support sleepers at the correct level, spread forces into the formation, and allow surface water drainage, formation to support ballast and collect water to the drainage system, subgrade, which is the natural layer where all other parts are built on, and the drainage system to convey water away from the track. The primary focus of the case study is on the ballast and rail, but other parts are considered if they are linked to these two parts.

The ballast is good when it is composed of crushed angular hard rocks and stones, free of dirt and dust, uniformly graded, and not prone to cementing action [2]. The degradation mechanism of ballast is called fouling, which occurs when small particles build up within the ballast. Causes of ballast fouling can be the ballast breakdown, sleeper wear, and the infiltration from the surface, underlying granular layers, or subgrade [2]. Ballast fouling can result in a saturated subgrade and wet beds because it impedes water drainage [75]. This leads to differential track settlement because of the uneven distribution of loads. The ballast can be maintained either by tamping, stoneblowing, or renewal techniques, which restores the track geometry to a better condition [76]. Ballast tamping is the common form of correcting the track geometry. It is done using specialised trains which lift the rail with the sleepers to the target level. Then, tamping tines are inserted and vibrated to squeeze the ballast under the gap, recovering the correct level of the rail. This process causes significant breakage of ballast particles, which can result in highly fouled ballast. When this is the case, tamping can no longer be effective and stoneblowing can be considered since it causes much less breakage of ballast and can be used even if the ballast is fouled [77]. Stoneblowing is performed using trains that lift the rail and the sleepers to the target level. Then, they insert tubes that use compressed air to blow a measured quantity of clean ballast into the gap below the sleepers, leaving the rail at the correct level after the tubes are removed. The only disadvantage of stoneblowing is that it is slower and more expensive than tamping [77]. Thus, a rule of thumb is to perform tamping operations until the ballast becomes highly fouled and to use stoneblowing after that [77]. However, after reaching a critical level of ballast fouling, maintenance actions become less effective, and renewal should be performed. Renewal can be performed by cleaning the ballast and reusing a portion of it mixed with new ballast, or by performing a complete replacement of the old ballast.

Consequently, maintenance activities impact the quality of ballast, leading to accelerated degradation rates and settlement. The extent of track settlement can be quantified through the utilisation of specialised trains, such as the *New Measurement Train* operated by *Network Rail* (NR), which employs laser scanning technology to assess changes in track geometry every 0.2 metres as the train progresses along the track. The train's journey is affected by variations in the track profile, with long and smooth undulations having minimal impact on train safety and comfort and are thus disregarded in assessments. The most commonly used metric for measuring track settlement is the vertical standard deviation (SD) of a set of measurements taken for each Poskey along the track [78]. This is due to the vertical geometry being the most prone to degradation and having the greatest influence on ride quality and maximum permissible speed. On a typical European track, a vertical SD of less than 5.2, 7.4, 8.3, 9.9, or ∞ is classified as "Excellent," "Very Good," "Good," "Poor," and "Super-red," respectively, for track speeds below 20 MPH.

The vertical geometry profile of the track is expected to improve as a result of maintenance activities performed on the ballast. However, as the quality of the ballast deteriorates, the ability of maintenance actions to correct the rail level becomes diminished. As the fouling index of the ballast increases, small particles fill the voids between rocks resulting in denser ballast. Maintenance activities may temporarily create voids between rocks, but the resistance to loads becomes weak, and as a result, small particles tend to quickly fill these voids once the track is subjected to loads [77]. This in turn makes the maintenance actions less effective.

The irregularities of the track surface have a substantial effect on the incidence of rail faults, in addition to the safety and quality of the ride [79]. As trains traverse the rail network, they exert substantial forces on the rails, which can result in a wide spectrum of defects and faults [80]. Rail corrugations, for instance, can impair the quality of the ride and accelerate the degradation of many track and vehicle components [79]. On the other hand, head wear can decrease the rail-wheel interface area and reduce the grip for braking and accelerating, thereby increasing the likelihood of faults [81]. In the event of a break, increased forces are imposed on the surrounding parts, and speed restrictions may be necessary to maintain safety, leading

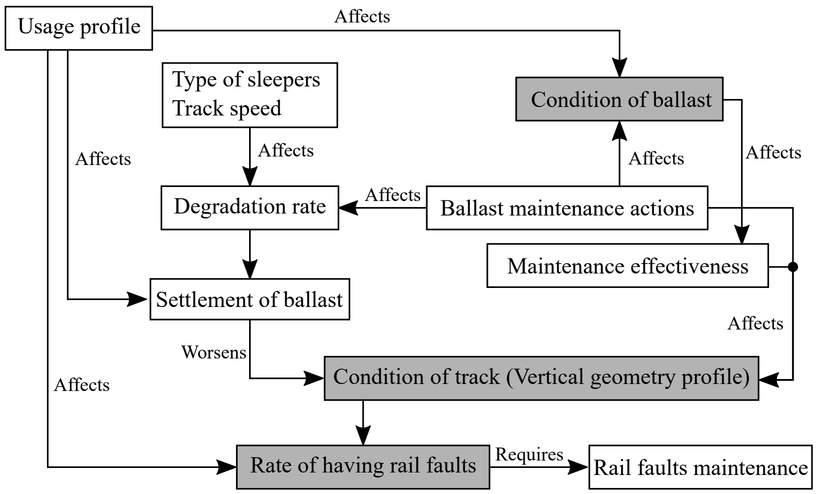


Figure 2: Summary of the railway relations that are considered in this study.

to delays and financial losses [82]. It is crucial to maintain appropriate vertical geometry to ensure the safe and efficient operation of the rail network and to minimise the risk of rail faults. Correction of rail faults may involve grinding or welding if the fault is not severe, while replacement of the rail may be required in more severe cases.

The relationships between railway infrastructure components, such as the track condition expressed in terms of the vertical geometry profile and the degradation of the ballast, are summarised in Figure 2. A degradation in track condition due to ballast settlement leads to an increase in the frequency of faults in the rail and necessitates more frequent maintenance activities. Conversely, while maintenance of the ballast can improve track condition, it can also result in ballast fouling and faster degradation, reducing the effectiveness of maintenance efforts. Thus, it is imperative to have a comprehensive understanding of these interrelated factors for effective railway infrastructure management and the maintenance of safe and efficient rail operations. To do so, models were created for the degradation rate of the track, the ballast maintenance effectiveness, and the rate of rail faults and their maintenance frequencies based on data from a typical European track. The data covered track geometry, fault and maintenance records of a European national rail network operator over a period of approximately 8 years, and also covered all uses from freight to passenger services and all levels of track speeds. Although the data are treated confidentially here, representative models were chosen to demonstrate the techniques introduced in this paper.

Several factors were tested to determine which of them affect the degradation rate and it was found that the type of sleepers, the speed of the track, and the maintenance history have an impact. A stochastic model was built relating the degradation rate (mm/EMGT) to these three factors, proposing a Weibull distribution for each entry. The study considered a track speed of less than 20 MPH with small concrete sleepers, and Table 1 summarises the parameters of the Weibull distributions for each maintenance history required for

Table 1: Shape parameter, β, and size parameter, η, of Weibull distributions for the degradation rate [m/EMGT] of small concrete sleepers with track speeds 5-20MPH and different maintenance histories.

After:	1^{st} Renewal	1^{st} tamp	3^{rd} tamp	5^{th} tamp	6^{th} tamp	7^{th} tamp	1^{st} stoneblowing
η	1.93E-04	2.15E-04	2.22E-04	2.06E-04	2.21E-04	2.37E-04	2.12E-04
β	1.03E+00	8.18E-01	1.05E+00	1.17E+00	7.25E-01	1.00E+00	6.49E-01

Table 2: The vertical geometry SD [mm] after each maintenance activity representing the maintenance effectiveness

Renewal	Tamping				StoneBlowing		
1^{st}	1^{st}	3^{rd}	5^{th}	6^{th}	1^{st}	5^{th}	7^{th}
0	1	1.5	3	3.5	1	2.5	3.5

modelling the degradation rate. The values of β and η are of the order of magnitude expected for this type of track and the observed variation in these values with changing maintenance history is, to some extent, due to the variation in the use of low speed track, which includes, for example, sidings and access to depots used by freight trains. This model allows the calculation of the SD of the track after a certain usage by using the following formula:

$$SD^* = SD + DR \cdot (U^* - U) \tag{5}$$

where DR represent the degradation rate that can be sampled from the distributions of Table 1 and U represent the usage in EMGT. The variables with and without asterisks represent the current and the last state respectively.

Conversely, ballast maintenance actions have the potential to improve the SD of the track. Nevertheless, the frequency of maintenance actions increases the rate of settlement as previously discussed. In light of this, Table 2 presents some assumed values that depict the impact of each maintenance action on the SD of the track in relation to the maintenance history of the track.

The analysis revealed a strong correlation between the vertical geometry SD of the track and the rate of faults, leading to the creation of a model that calculates the normalised rate of each rail fault as a function of the SD. The rate of faults increases with the length and usage of the track, thus normalisation was deemed necessary. Data recordings of various rail faults including Squat, Tache Ovale, Bolt Hole, Weld, Other, Rolling Contact Fatigue (RCF), Wheel burn, Lipping, Side Wear, Head Wear, Corrugation, and Unknown faults were analysed. To calculate the rate of each fault, the data was stacked and lines were fitted to the data, starting with squats and adding faults one by one, simplifying the decisions within the model.

Algorithm 1 outlines the procedure for determining the rate of occurrence of faults within each stacked group. To begin, a list of faults, denoted as FL, is established, and a set of faults stacked groups, denoted

Algorithm 1 Calculation of the probability of having a fault and its type

Define List of faults, FL = ["Squat", "Tache Ovale", "Bolt Hole", "Weld", "Other, "Rolling Contact Fatigue (RCF)", "Wheel burn", "Lipping", "Side Wear", "Head Wear", "Corrugation", "Unknown"]. Define stacked sets based on the order of FL are defined as: $FS_1 = \{FL[1]\}$, $FS_i = FS_{i-1} \cup \{FL[i]\}$ $\forall i \in \{2, \ldots, 12\}$.

The rate of having a fault from each of the lists $FS_i \forall i \in \{1, \ldots, 12\}$ can be calculated based on the following equation:

$$FR_i[\text{/poskey/EMGT}] = A_i \cdot SD^3 + B_i \cdot SD^2 + C_i \cdot SD \tag{6}$$

with A_i, B_i, and C_i for each set can be found in Table 3.

Function 1 – Fault type based on SD:

---

choose $R \in_R [0, 1]$. Then, $R \to R/(L \cdot \Delta U)$, with L and ΔU being the length in Poskeys and usage in EMGT respectively.

if $R < FR_{12}(SD)$ **then** ▷ having fault is probable because FS_{12} contain all faults

    for $i \in \{1, \ldots, 11\}$ **do**

        if $R < FR_i(SD)$ **then**

            Return FL_i ▷ fault i from list FL

    Return FL_{12} ▷ unknown fault

else

    Return $\emptyset$ ▷ no fault

Function 2 – Correction type based on the fault type:

Maintenance types list is MT =["rerail", "weld", "grind or other"]

any fault type, i, has 3 stacked probabilities, $SP_{i,1}$, $SP_{i,2}$, and $SP_{i,3}$, that stands for the elements of MT respectively and stored in Table 4.

To know which maintenance type corresponds to the fault, choose $R \in_R [0, 1]$

for $j \in \{1, 2, 3\}$ **do**

    if $R \leq SP_{i,1}$ **then**

        Return: MT_j

Table 3: Parameters for the calculation of the fault rate of different sets. $FL_{1,\ldots,12}$ are the stacked sets of faults defined in Algorithm 1

FS	A	B	C	FS	A	B	C
1	7.64E-05	-6.45E-04	3.44E-03	7	1.12E-04	-3.87E-04	4.93E-03
2	7.85E-05	-6.55E-04	3.73E-03	8	7.34E-05	1.31E-04	3.87E-03
3	6.90E-05	-5.45E-04	3.56E-03	9	1.66E-04	-3.48E-04	4.61E-03
4	9.38E-05	-7.96E-04	4.85E-03	10	1.61E-04	-1.98E-04	4.28E-03
5	1.18E-04	-6.94E-04	5.22E-03	11	1.60E-04	-1.92E-04	4.27E-03
6	1.13E-04	-5.46E-04	5.07E-03	12	1.70E-04	-2.31E-04	4.33E-03

Table 4: Stacked probabilities for each maintenance action based on fault type. FL is the list of faults defined in Algorithm 1

FL	Rerail	Weld	Grind or other	FL	Rerail	Weld	Grind or other
1	0.328	0.954	1	7	0.267	0.874	1
2	0.722	0.963	1	8	0.029	0.134	1
3	0.9	0.919	1	9	0.044	0.338	1
4	0.519	0.904	1	10	0.213	0.752	1
5	0.641	0.918	1	11	0.706	0.765	1
6	0.508	0.786	1	12	0.464	0.63	1

as FS, is constructed based on the elements of FL. For instance, if the first, second, and third elements of FL are Squat, Tache Ovale, and Bolt Hole respectively, then FS_1 consists of only Squats, FS_2 encompasses Squats and Tache Ovale, and FS_3 encompasses Squats, Tache Ovale, and Bolt Hole.

Subsequently, a third-order polynomial function is utilised to model the fault rate of each stacked group. The parameters of the fitted functions are listed in Table 3. Using these rates, the probability of encountering a fault after a specified usage over a certain track length can be calculated as depicted in the first function of Algorithm 1.

The second function of Algorithm 1 presents a method for sampling a correction for the fault from the available options. This function is based on the values in Table 4, which are derived from the stacked rates of correction methods for each type of fault. For instance, for the RCF, which is the sixth element of FL list, the rate of performing "Rerail", "Welding", and "Grinding and other" methods are 0.508, (0.789-0.508), and (1-0.786) respectively, as demonstrated in Table 4. These rates serve as sampling probabilities for the correction methods.

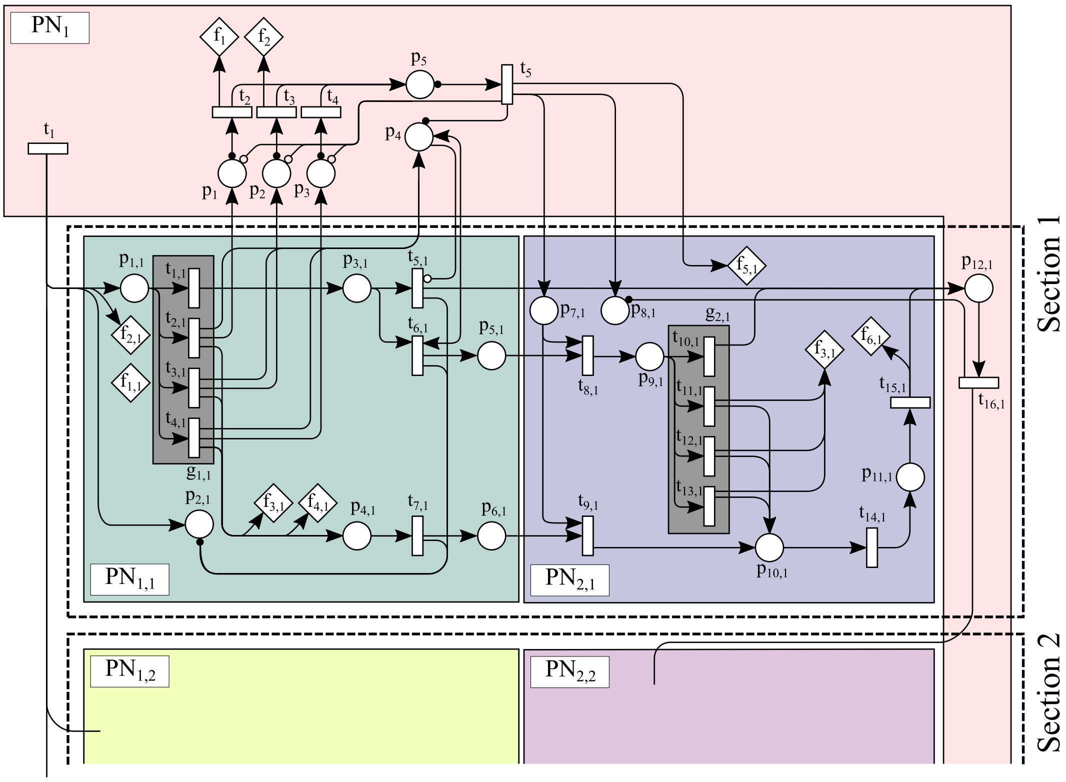


Figure 3: The iPN for modelling and optimising the railway maintenance and operation.

5.2. Railway iPN model

The aim of the railway iPN model is to create an expert decision support system (DSS) that helps in finding an optimal maintenance strategy for the ballast and rail taking into consideration the working conditions of the railway. Figure 3 shows the subnets that form the PN model, where node descriptions are provided in Table 5 and function descriptions in Figure 4. The railway is composed of 10 identical sections, with each section being modeled by two subnets. The names of the subnets of an arbitrary section $i \in \{1, 2, \ldots, 10\}$ are $PN_{1,i}$ and $PN_{2,i}$. An additional subnet called PN_1 models the common activities for all sections. Figure 3 does not show the subnets of sections $2, \ldots, 10$ due to lack of space. However, these subnets are identical to the ones of the 1$^{\text{st}}$ section. As with the subnet names, nodes that are in the common Subnet have one-number subscripts while the ones in the other subnets have two-number subscripts with the 2$^{\text{nd}}$ number referring to the ID of the section. The decomposition method provided in Section 3.3 can be used to reduce the computational costs as explained. If the decomposition method is used, the conditions to enable subnet $PN_{1,i}$ or $PN_{2,i}$ is to have a token in place $p_{1,i}$ or $p_{8,i}$ respectively, and PN_1 is always enabled without the need for any conditions.

Initially, all places are unmarked, time is equal to 0, and all the sections are in Excellent state. The information about each section is not represented by the PN places but is calculated through the functions described in Figure 4 as explained below. Function $f_{1,i}$ does not have any input arcs, so it runs every time the state of the PN changes. This function updates the condition, calculates the probability and the cost of having a fault, and calculates the reward based on the condition of the section. The dynamics of the problem start with transition t_1, which is a timed transition representing the inspection. This transition fires every 0.5 yrs. to mark $p_{1,i}$ and $p_{2,i}$ and execute function $f_{2,i}$. The function $f_{2,i}$ updates the available actions by excluding the non-available actions from action group $g_{1,i}$. These actions are *no-action*, *tamping*, *stoneblowing*, and *renewal*, represented by transitions $t_{1,i}$, $t_{2,i}$, $t_{3,i}$, and $t_{4,i}$ in $g_{1,i}$, respectively. After 7 tamping actions, tamping cannot be chosen for the ballast due to the reached fouling level, and after 3 stone blowing actions, the only option becomes renewal. Thus, based on the maintenance history, available actions are updated.

By marking $p_{1,i}$, action group $g_{1,i}$ becomes enabled, representing the need to make a maintenance decision. Accordingly, the RL agent selects one of the available transitions from $g_{1,i}$. If $t_{1,i}$ is chosen, $p_{3,i}$ will be marked to indicate that no maintenance decision was made. On the other hand, if $t_{2,i}$, $t_{3,i}$, or $t_{4,i}$ is triggered, $p_{4,i}$ will be marked. $f_{3,i}$ and $f_{4,i}$ will be executed, and p_1, p_2, or p_3 will be marked to indicate that the preparation for tamping, stoneblowing, or renewal, respectively, has commenced.

Each maintenance action consists of two distinct steps: preparation and travel, followed by the actual maintenance itself. The duration for executing each of these steps is determined using the functions denoted as $f_{3,i}$ and $f_{4,i}$ (Figure 4). Function $f_{3,i}$ is responsible for updating the actual maintenance time, influencing the progression of maintenance time transition ($t_{15,i}$), and aggregating RL rewards based on the associated maintenance costs. Meanwhile, function $f_{4,i}$ pertains to the adjustment of maintenance preparation time, governing the timing of preparation time transitions ($t_{2,i}$, $t_{3,i}$, $t_{4,i}$), and accumulating RL rewards linked to the preparatory expenses.

A single maintenance preparation can effectively address the repair needs of multiple sections when they require the same type of maintenance. Transitions t_2, t_3, and t_4 model the durations for maintenance preparations: t_2 for tamping, t_3 for stoneblowing, and t_4 for renewal. Places p_1, p_2, and p_3 indicate that preparations are underway. All preparation actions must be completed in order for the maintenance of the first section to commence. The initiation of maintenance is represented by transition t_5. As depicted in the PN model, this process involves inhibiting transition t_5 with places p_1, p_2, and p_3, ensuring that maintenance cannot start until all preparation tasks are finished.

Following the commencement of maintenance (triggered by the firing of t_5), the option to repair sections not previously selected for repair becomes available. This can be seen as a form of opportunistic maintenance. An advantage of deciding to repair a section on-site is the utilization of available maintenance trains to repair additional sections, thus saving costs associated with preparation and travel. To indicate the availability of maintenance trains, functions f_1 and f_2 are executed upon the firing of transitions t_2 and t_3, respectively, signifying the availability of tamping or stoneblowing trains.

The actual maintenance comes after finishing the preparation and reaching the site. It starts after transition t_5 is fired to mark $p_{7,i}$ and $p_{8,i}$. If $p_{5,i}$ is marked, $p_{7,i}$ enables $t_{8,i}$ indicating that a decision not to repair section i was taken. Then, $t_{8,i}$ fires to allow a decision through action group $g_{2,i}$ regarding the opportunistic maintenance. If the opportunistic decision was not to repair section i, $t_{10,i}$ fires to mark $p_{12,i}$ indicating that this section is finished; whereas, if a maintenance action was decided, $f_{3,i}$ will be executed and $p_{10,i}$ will be marked indicating that the maintenance can start. On the other hand, if the section was already decided to be maintained, $p_{6,i}$ and $p_{7,i}$ will allow $t_{9,i}$ to fire, which marks $p_{10,i}$, indicating the possibility to proceed directly in performing the maintenance. The time taken to perform maintenance is modeled by transition $t_{14,i}$ while its effectiveness is modeled by $f_{6,i}$. It is assumed that maintenance is only done during non-working hours of the train, which means that maintenance can be a number of interrupted intervals. Working hours can cause further degradation of the non-reached parts of the section before the maintenance ends. This is why $f_{6,i}$ is not directly executed by $t_{14,i}$ but by $t_{15,i}$, allowing for $f_{1,i}$ to account for degradation that occurs during the maintenance period. The firing of $t_{15,i}$ marks $p_{12,i}$, indicating that this section is finished. After the i^{th} section is finished, $t_{16,i}$ fires to mark $p_{7,i+1}$ and $p_{8,i+1}$, and the maintenance of the next section starts.

5.3. RL inputs

To optimize track maintenance, multiple optimization problems are tackled by breaking down the track into separate RL environments. Each section of the track is treated as an individual environment, complete with its own agent, states, actions, and value function. As these environments make up the same system, each one impacts the transition probability function of the others. Reward functions are specific to each environment and are solely dependent on the agent's decisions within that environment. The optimization problems are episodic and terminate when the decision to renew the section ballast is made, as this represents a new investment and a fresh start. The goal of the agent is to increase rewards, which equate to the cost function and are defined in terms of revenues and expenses. This approach requires the agent to maximize the use of the section's ballast before renewal.

5.3.1. Definition of the rewards function (cost function)

Rewards in RL play a crucial role in guiding the RL agent towards optimizing the maintenance strategy. In the context of the railway industry, these rewards can be expressed in monetary terms, as the railway companies aim to ensure financially efficient operation while ensuring appropriate levels of safety. The rewards can be either positive, representing revenues, or negative, representing costs. However, due to the commercial sensitivity of costs and revenues, it is not possible to represent rewards in terms of actual monetary values. Therefore, the rewards are presented in unitless forms while maintaining their realistic values relative to each other through consultation with railway experts.

Table 5: The description of the iPN nodes.

Node	Description
	$PN_{1,i}$
$p_{1,i}$	taking maintenance decision
$t_{1,i}, t_{2,i}, t_{3,i}, t_{4,i}$	no-action, tamping, stoneblowing, and renewal decisions respectively
$p_{2,i}$	subnet key
$p_{3,i}, p_{4,i}$	no-action or a maintenance action is chosen respectively
$t_{5,i}$	opportunistic maintenance is not possible, reset the PN key
$t_{6,i}, p_{5,i}$	opportunistic maintenance is possible, reset the PN key
$t_{7,i}, p_{6,i}$	Perform maintenance after finishing the preparation, reset the PN key
	$PN_{2,i}$
$p_{7,i}$	site is reached
$p_{8,i}$	subnet key
$t_{8,i}, p_{9,i}$	no maintenance was decided, check for opportunistic maintenance
$t_{9,i}$	maintenance was decided, proceed
$t_{10,i}, t_{11,i}, t_{12,i}, t_{13,i}$	no-action, tamping, stoneblowing, and renewal decisions respectively (opportunistic)
$p_{10,i}$	ready to perform maintenance
$t_{14,i}$	models the time taken to finish repairing the section, which is controlled by function $f_{3,i}$, and allows for $f_{1,i}$ to consider the effect of further degradation
$p_{11,i}, t_{15,i}$	models the maintenance effectiveness and update condition
	PN_1
t_1	inspection, with delay equal to 0.5
$(p_1, t_2), (p_2, t_3), (p_3, t_4)$	tamping, stoneblowing, and renewal preparation respectively with the delay of transitions controlled by function $f_{4,i}$
p_4	there exist a section that will be maintained
p_5	one or more preparations are finished
t_5	site is reached, ready for doing maintenance
$p_{12,i}, t_{16,i}$	section is finished move to the next section

$f_{1,i}$	**if** $t_S^* > t_S$ **then** ($t_S^* = t$ is the current time and t_S is the last time the section was updated) Calculate the usage, U^*, at the current time, t_S^*, based on the usage rate: $U^* = 20t_S^*$ Get SD^* using Equation 5. Get, R_c, the continuous reward between the two states of the section based on Algorithm 2. Accumulate the reward of the last RL state: $R_t \to R_t + R_c$ Update the variables: $U = U^*, SD = SD^*, t_S = t_S^*$
$f_{2,i}$	Update the available actions for $g_{1,i}$ according to the following rules: tamping is not allowed after 7 tamps or after stoneBlowing. stoneblowing is not allowed more than 3 times
$f_{3,i}$	sample the output rate, OR [yrds/hr.], from $\mathcal{W}(1.28, 249.27)$ for tamping and $\mathcal{W}(1.30, 237.26)$ for stoneblowing Convert OR to [Poskeys/hr.]:$OR \to OR/220$ Calculate the maintenance time, t_M [yrs.], assuming 2,080 working hours per year: $t_M = OR \cdot L/2080$ Get the actual maintenance costs based on the section's length, C_m Accumulate the reward of the last RL state: $R_t \to R_t - C_m$ Update the maintenance history
$f_{4,i}$	The time to prepare for maintenance and reach the site, t_p, will be equal to: 1 night if the condition is *super-red*, 1 week if the condition is *poor*, 2 weeks if the condition is *good*, and 1 month otherwise. Accumulate the reward of the last RL state: $R_t \to R_t - C_p$
$f_{5,i}$	Update the available opportunistic maintenance actions for $g_{2,i}$ based on $f_{2,i}$ rules while considering the prepared transitions
$f_{6,i}$	Update SD based on the maintenance effectiveness (Table 2) Update DR from the distributions in Table 1 If the maintenance action is a renewal, terminate the old RL episode and start a new one
f_1	Indicate that the preparation for tamping maintenance is done
f_2	Indicate that the preparation for stoneblowing maintenance is done

Figure 4: Description of the functions used in the iPN model.

The study examines the direct and indirect effects of RL decisions on the rewards function. To construct the reward function, various effects are considered, including maintenance and renewal costs, preparation and travel costs, possession costs, delay costs, catastrophe costs, and revenues. Directly influenced by the RL agent's decisions are the costs associated with ballast maintenance, preparation, and travel. Ballast maintenance costs are incurred on a per-section basis, while preparation and travel costs are paid once to address multiple sections that undergo maintenance at the same time in close locations. The RL agent can strategically choose to repair multiple sections simultaneously, known as opportunistic maintenance, in order to minimize the expenses related to preparation and travel.

Indirectly affected by the RL decisions are the costs that depend on the condition of the track. When the track's condition deteriorates, the likelihood of rail faults increases, resulting in higher maintenance costs required to address these faults [79]. These costs are incurred for each track section, as outlined in the effects summary provided in Table 6. It is important to note that this study does not specifically focus on optimizing decisions related to this particular maintenance type; instead, it considers it as part of the overall costs influenced by decisions concerning ballast maintenance. Consequently, the expenses associated with travel and preparation for this maintenance type are included within the broader maintenance costs, rather than being treated separately.

Additionally, a degraded track condition can result in increased delay, possession, and catastrophe costs, while simultaneously decreasing rail revenues. Therefore, the reward function incorporates these costs and revenues as a function of the track's condition, which are represented as condition-based rewards, r_c, as illustrated in Figure 5. It is widely recognized that as the railway condition deteriorates, precautionary speed restrictions should be imposed to minimize the risk of failures. However, in severe cases of track degradation, speed restrictions alone may not be sufficient to mitigate the risks, which could potentially lead to catastrophic outcomes. Furthermore, deteriorated tracks may require urgent maintenance during operational hours, leading to increased possession costs. On the other hand, poor rail conditions directly impact revenue generation. Deteriorated rail infrastructure reduces operational efficiency, leading to slower trains, longer travel times, and unreliable services. These obstacles discourage potential customers and erode the trust of existing passengers, resulting in reduced ridership and decreased revenue. It should be noted that excluding revenues from the reward function even if the revenues are not affected by the condition may lead the agent to make decisions to prematurely renew sections and terminate the episode. Thus, including revenues is crucial to motivate the RL to continue the episode in a logical manner.

Condition-based rewards capture the difference between revenues and costs at different track conditions, as depicted in Figure 5. Positive r_c are associated with favourable rail conditions, where revenues exceed costs. However, as costs and losses surpass revenues, as observed in the "Poor" and "Super-red" conditions, r_c become negative. These negative rewards serve as an indicator of the adverse impact that these conditions could have on overall profitability. Importantly, these rewards are calculated per Poskey and usage,

Action	Type	Effect	Decision
Ballast maintenance	tamping	-1000/section	Chosen by RL agent
	stoneblowing	-2000/section	
Ballast preparation and travel	tamping	-1000	
	stoneblowing	-2000	
Ballast renewal	-	Start a new episode	
Rail maintenance (includes preparation and travel)	rerail	-1500/section	Affected by the track condition, which is related to the RL decisions
	welding	-300/section	
	grinding	-100/section	
Delay	-	Form r_c, which is function of the condition (Figure 5)	
Possession	-		
Catastrophe occurrence	-		
Revenues	-		

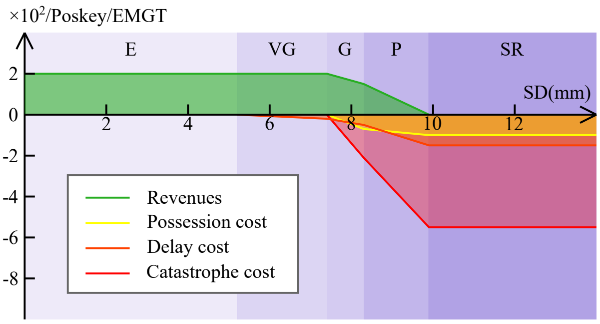
Table 6: Breakdown of input costs and revenues (unitless) influencing rewards (cost function).

accumulating continuously based on the usage. Conversely, other rewards are constant and calculated only once per occurrence of the action. In summary, the reward function comprises constant rewards specific to each action and continuous condition-based rewards (r_c) that are determined by the track's condition, calculated per usage and Poskey, which are all summarized in Table 6.

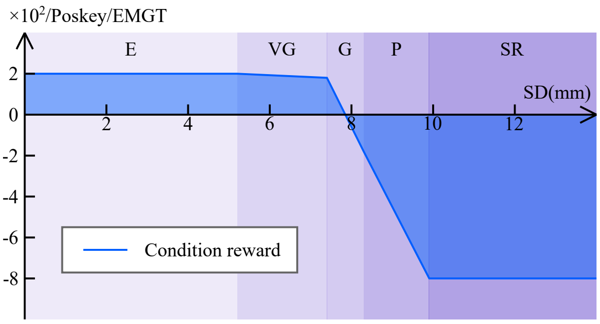
5.3.2. Definition of the environment

The environment of each section is defined in terms of the important features that the RL agent needs for taking decisions. Thus, the factors describing the environment are chosen to be:

- the condition of the section, which depends on the value of the SD[mm] and is classified by the intervals [0,4], (4,5.2], (5.2,6.5], (6.5,7.4], (7.4,8.3], (8.3,9.9], and (9.9,∞). These intervals are named E_1, E_2, VG_1, VG_2, G, P, and SR respectively.
- the settlement rate, which is represented by the rate of change in the SD [mm/EMGT]. It is divided into two groups depending on whether the degradation rate is less than or greater than 0.2. These groups are denoted as slow and fast and represented by letters S and F respectively.
- the maintenance history, which is represented by the last maintenance type and the number of times this maintenance was performed previously. The maintenance history is named by a letter and a number with the letter representing the types and the number representing the number of previous actions. The letters T, SB, and R stand for the tamping, the stoneblowing, and the renewal actions



(a) The revenues and costs that are a function of the condition, which form up the condition-based reward, r_c.



(b) Condition-based reward, r_c, which is the summation of revenues and costs that are a function of the condition.

Condition	SD (mm)	Revenues	Costs			Condition reward (r_c)
			Possession	Delay	Catastrophe	
Excellent (E)	0	200	0	0	0	200
Very Good (VG)	5.2	200	0	0	0	200
Good (G)	7.4	200	0	20	0	180
Poor (P)	8.3	150	70	50	210	-180
Super Red (SR)	9.9	0	100	150	550	-800

(c) Summary of revenues, costs, and rewards that are a function of condition at different condition thresholds.

Figure 5: Visualization of revenues and costs influenced by rail conditions, depicted as condition-based rewards, r_c. The accompanying table presents corresponding values at each condition threshold in terms of SD, specifically for a track speed range of 5-20 MPH on a typical European track.

Algorithm 2 Calculation of the continuous rewards existing between two states of a section.

Inputs: Degradation rate DR, initial usage U, final usage U^*, initial SD, final SD^*.
Get, C_T, the set of condition thresholds that are crossed between SD and SD^*: $\{c_{T1}, c_{T2}, \ldots, c_{Tn}\}$.
Get the usage of the section when crossing each of the conditions thresholds values by rearranging Equation 5: $U_i = (1/DR)c_{Ti} + (U - SD/DR)\forall i \in 1, \ldots, n$.
Divide the degradation between U and U^* into $n + 1$ intervals: $[U, U_1], [U_1, U_2], \ldots, [U_n, U^*]$.
Initialize the continuous reward, $R_c = 0$
for $i \in \{1, \ldots, n+1\}$ **do** ▷ for all the intervals.
    Calculate the average standard deviation $\overline{SD}$ to do the calculations based on it.
    Check if a fault will occur using function 1 of Algorithm 1.
    Check the probability of having a fault using function 1 in Algorithm 1.
    Sample the correction method of the fault using function 2 in Algorithm 1.
    calculate the usage, $\triangle U$, during this interval.
    Calculate the cost for correcting the fault, c_f.
    Get the condition reward, r_c, using Figure 5.
    Accumulate the continuous reward: $R_c \rightarrow R_c + (r_c + c_f) \cdot L \cdot \triangle U$.

respectively. For example, T_2 stands for 2 previous tamping actions.

- the PN state, which is represented by the markings of places $p_{1,i}$ and $p_{7,i}$ for each section. This details which type of decision is required.

5.3.3. Algorithm tuning and parameter scaling

The Q-learning method uses the bootstrapping effect, which means that Q-Values are updated based on the values of the successor states. At the beginning of the learning process, all the Q-Values start with random numbers, which makes these updates far from their actual values. This can give an advantage to the Monte-Carlo RL method over the Q-learning at the beginning of the learning process. However, Q-learning is an off-policy method that outperforms the Monte-Carlo or other on-policy methods by being able to explore the environment without affecting the Q-Values updates [26]. For this, the learning process was divided into two parts, with the Monte-Carlo RL method being used in the first one, and the Q-learning in the second one.

The problem of optimising the policy for each section is considered an episodic task. However, the number of episodes cannot be used to control the duration of the learning process because each environment for each section has its own episode counter. A common variable that is shared among all environments is the time. This variable has an effect on the number of episodes in each environment without affecting their conditions, so it can be used as an arbitrary variable to control the duration of the learning process. For this, the duration of the learning process is chosen to be equal to 6×10^7 yrs., with 2×10^7 yrs. using the Monte-Carlo RL method and 4×10^7 yrs. using the Q-learning method. The model and its formulas are affected by the time difference and not by the time, so the accumulation of time throughout the learning process is just a way to increase the number of episodes, but it does not have any physical meaning or effect.

Three parameters, which are the discount rate γ, the learning rate α, and the exploration rate ε, should be specified for the RL methods. The discount rate is assigned a value $\gamma = 1$ to avoid being biased to early returns. The learning rate, α, is controlled by the following formula:

$$\alpha = \begin{cases} 1/n_u & \text{if } n_u < 1000 \\ 10^{-3} & \text{if } n_u > 1000 \end{cases} \tag{7}$$

where n_u is different for each state-action pair and it represents the number of times its Q-Value is updated. For n_u less than 1000, the formula ensures that the Q-Value is equal to the average of all the previous expected returns, and neglects the effect of initial values of the Q-Values [25].

$$\begin{aligned} \sigma_\varepsilon(t) &= a + b\exp(-c \cdot t), \text{ with:} \\ a &= \varepsilon_{\min} \\ b &= \varepsilon_{\max} - \varepsilon_{\min} \\ c &= \ln[(\varepsilon'_{\min} - a)/b]/t_e \\ \varepsilon'_{\min} &= v(\varepsilon_{\min} - \varepsilon_{\max}) + \varepsilon_{\max} \end{aligned} \tag{8}$$

Equation 8, which is based on Equation 12 in [25], presents an exponential decay function with easily adjustable parameters that is utilized to regulate the decay of ε. The parameters of this function are the end of the decay process, t_e, the maximum, $\varepsilon_{\max}$, and minimum, $\varepsilon_{\min}$, values of the controlled variable, and a parameter called v. $v = 1 - \epsilon$ indicates how close the practical minimum is to the actual minimum. The argument of the function is t, and for a range of $t \in [0, t_e]$ the output of the function decays from $\varepsilon_{\max}$ to $\varepsilon_{\min}$.

Exponential decay parameters are assigned for each part of the learning process. For the Monte-Carlo RL method part, $\varepsilon_{\max}$, $\varepsilon_{\min}$, t_e, and v are chosen as 1, 10^{-4}, $0.95 \times 2 \times 10^7$, and 0.99 respectively, with the argument, t being equal to the time ($t =$time). On the other hand, for the Q-learning method part, $\varepsilon_{\max}$, $\varepsilon_{\min}$, t_e, and v are chosen as 0.2, 10^{-3}, $0.9 \times 4 \times 10^7$, and 0.99 respectively, with the argument, t passed as, $t =$time-2×10^7 to shift the argument to the starting point of the Q-learning part.

Since all the sections share the same characteristics, it is expected that they have the same optimal policy. To make the learning process faster, the RL agents were allowed to share the experience by updating the Q-Values in all environments once a Q-Value of similar state-action pair is updated in any environment.

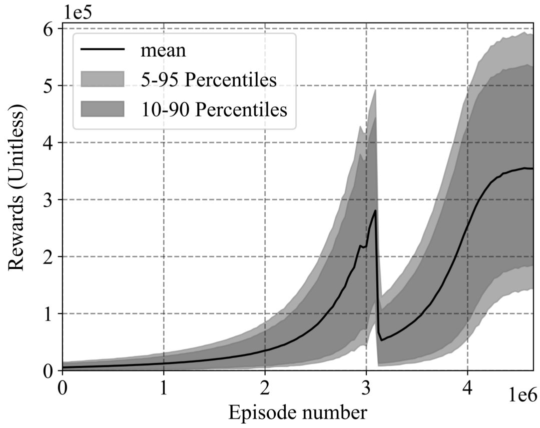


Figure 6: The variation of rewards as a function of the episode number.

6. Results

This section shows the results of the different simulations performed to find the best policy for the operation and maintenance of the railway sections. Other simulations were performed to test the idea of dividing the PN into multiple subnets. To do so, the PN proposed in Section 5.2 is simulated two times, one while dividing the PN into multiple subnets, and another without dividing it. Each of the two simulations was performed for multiple lifetimes such that the total duration of the accumulated lifetimes is 5000 yrs. in each simulation. Besides, no optimisation was done in these simulations and a random policy was used to select transitions from action groups. As a result, the time taken for the simulation with and without the subnet rules was 201 and 613 seconds respectively. This indicates a 3 times reduction in the computational cost between the two simulations.

Figure 6 shows the variation of the total reward as a function of the learning process. The learning process is divided into short intervals to be able to plot the mean and other measures for each of the intervals. It can be seen that the total reward increases until the end of the Monte-Carlo RL part of the learning process, then drops and continues increasing after the Q-learning starts. The end of the learning process shows a stable curve which indicates that the policy is no longer changing. The decisions of the final RL policy are summarised in Figure 7. The left part of the Figure shows the decisions for normal maintenance while the right part is for opportunistic maintenance. The 11 rows show the possible maintenance history states while the columns describe the condition of the section and settlement rate. The different decisions are described by the colours in the legend above the figure, where the white areas represent the unexplored states. For example, the decision shown in the red square is tamping and it corresponds to the state described by the section being in good condition with a slow settlement rate and 4 previous tamping actions. This figure shows that a sequence of 7 tamping actions followed by 3 stoneblowing actions and a renewal action is good

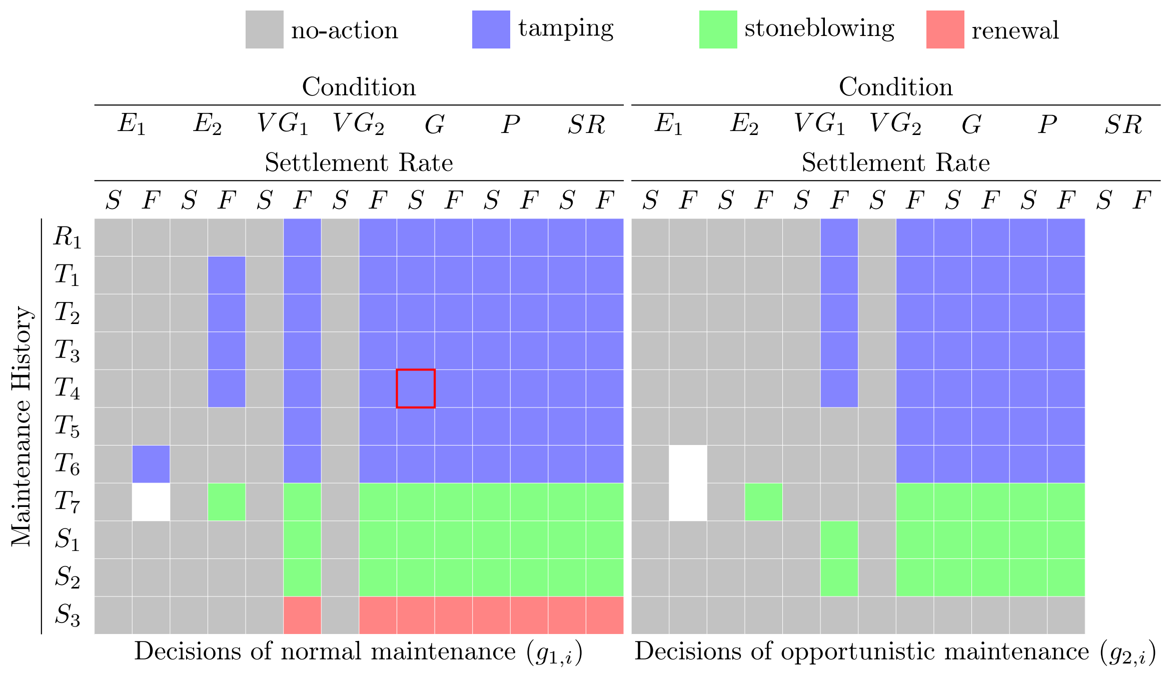


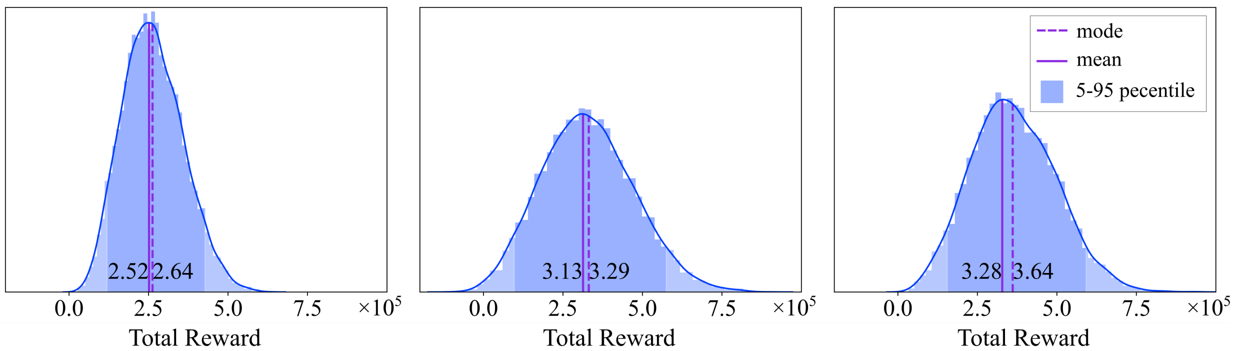
Figure 7: The final RL policy, which is described by the optimal actions at each of the RL states. The acronyms are defined in Section 5.3.2, white areas are unexplored states, and the red square is an indication of an example explained in the text.

for all track conditions since no stoneblowing action can be found above row T_7 and no stoneblowing action can be found above row S_3. This means that the RL policy does not differ from policies A and B described in the introduction of Section 5 in terms of the type of maintenance action. However, the decision regarding the need for maintenance is shown to be dependent on all the features of the RL environment because the no-action decisions are scattered in different zones. It can be seen that the need for maintenance increases as the condition of the section worsens especially in the cases of a fast settlement rate. Also, the distribution of actions over states is similar for opportunistic and normal maintenance actions.

The decisions shown in Figure 7 are reflected in the total rewards, the percentage of time spent in each condition, the distribution of maintenance actions over time and the section condition. Table 7 shows the average percentage of time spent in each condition when following each of the policies. Comparing percentages of time spent in each condition is preferable to using total absolute durations spent in each of the conditions per episode in order to avoid unfair conclusions. This is because absolute durations can be misleading due to episodes terminating with the ballast's life, rather than the rail's service life. Rail operators should maintain the rail according to its service life, not the life of the ballast. By using percentages, conditions can be reflected relative to the duration of the rail's service life, and can indicate probabilities of each state. It can be noted that Policy B increases the probability of being in the Poor and Super-red conditions while the RL policy was able to reduce the probability of being in the Super-red

Table 7: The average percentage of time spent in each condition for each of the simulated policies.

	Excellent	Very good	Good	Poor	Super-red
Policy A	87.33	11.75	0.53	0.33	0.04
Policy B	57.19	35.43	5.06	1.59	0.70
RL policy	66.17	30.97	2.40	0.34	0.007



a) Policy (A) b) Policy (B) c) RL policy

Figure 8: The distributions of rewards for the three considered policies.

condition to almost 0 while having a very low probability of being in the Poor condition. Policy A showed the ability to have the highest percentage of the Excellent condition but without being able to have the lowest percentage in the Super-red condition.

Figure 8 shows the distribution of rewards of the considered policies. The distributions are plotted because the problem is stochastic, and comparing based on one value can be misleading in such cases. The figure shows that Policy A resulted in the minimum rewards of all the policies, but it ensures that the rewards of an episode are always greater than 0, whereas Policy B results in rewards greater than Policy A, but can result in negative rewards. On the other hand, the RL policy ensures positive rewards whilst also ensuring the maximum rewards of all policies in terms of the mean and mode.

Figure 9 shows the distribution of maintenance actions over the condition of the section. It can be seen that the maintenance actions are concentrated in the Very-Good condition for Policy A, in the Good condition for Policy B, and in different conditions for the RL Policy. The figure also shows that there are very few actions taken in the Super-Red condition for Policy A and RL policy, whereas Policy B has a significant number of actions taken in this condition. Figure 10 shows the distribution of maintenance actions over the age of the section. For the three policies, tamping is followed by stoneblowing then by renewal. The renewal action is an indication of the end of life of the section. Policy A resulted in the shortest life with an average equal to 29.5 yrs, followed by the RL Policy with an average of 42.5 yrs., then by Policy B with an average equal to 45.2 yrs.

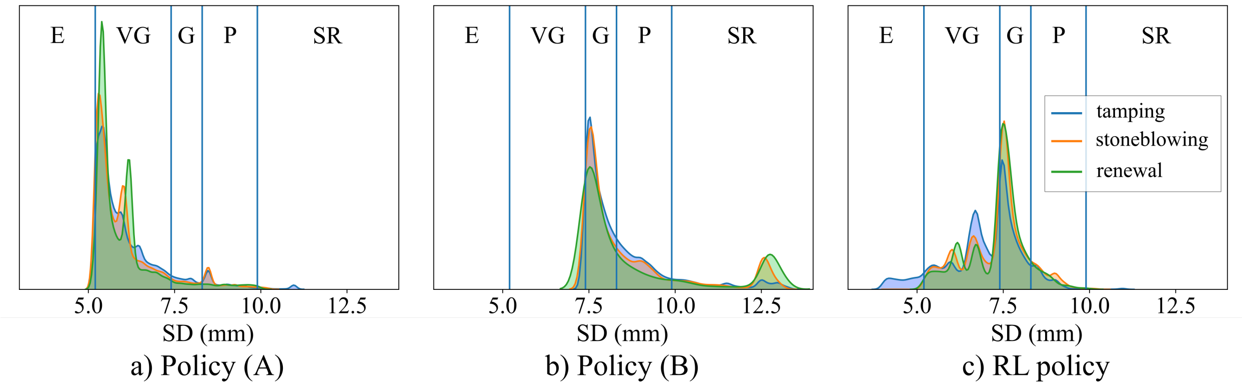


Figure 9: The distributions of the maintenance actions over the condition of the section for the three considered policies.

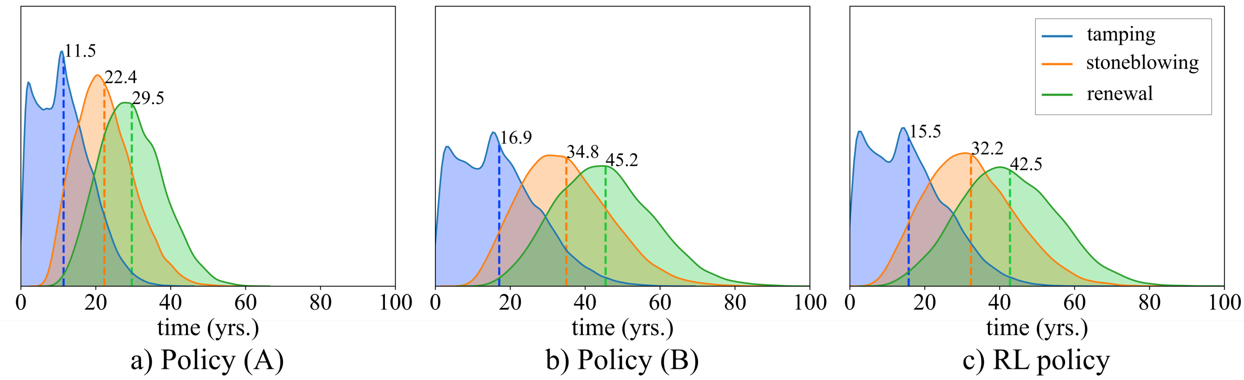


Figure 10: The distributions of the maintenance actions over the age of the section for the three considered policies.

7. Discussion

Regarding the division of the PN to multiple subnets, comparing the random simulations for the PN case, which is presented in Section 5.2, revealed a three-fold decrease in computational costs despite some parts of the PN being computed in parallel. The subnet rules can result in greater reduction if more subnets were in series because, for parallel subnets, all of the transitions are important to be checked at the same state, which weakens the effect of unchecking unimportant transitions because they will be few.

Regarding the optimisation problem, the total rewards shown in Figure 6 indicate that the optimal RL policy was reached since stable results are seen by the end of the learning process. The drop in the total rewards shown when the Q-learning started is due to the change in the exploration rate. Since the Q-learning is an off-policy RL method, increasing the exploration rate does not cause any diversion to the Q-Values and keeps their updates correct [26], but it allows the RL agent to discover other decisions that may be better than those already explored.

The results displayed in Figure 7 depict the decisions made by the RL agent for condition-based and opportunistic maintenance actions, which together constitute the final policy. Notably, the zones where the agent chose to perform opportunistic maintenance are a subset of those where condition-based maintenance is performed. This implies that the costs saved by avoiding the preparation and travel involved in opportunistic maintenance did not influence the agent's decision. The optimal policy, therefore, advocates for repairing a section only if it requires maintenance based on its condition, and not on the availability of equipment. Following this policy means that opportunistic maintenance is only triggered if there is a change in the section's condition between the inspection and the arrival at the site. In such cases, the decision to repair will be made on-site, which makes it opportunistic, but the decision will be due to the change in condition rather than the availability of equipment on site. Pursuing this policy ensures that maintenance actions are conducted only when required, given that each action has an impact on the ballast's life by crushing its particles and making it more susceptible to fouling, thereby shortening its lifespan. In addition, the policy tries to delay ballast maintenance as much as possible to extend its usage period until it reaches a highly fouled state where it can no longer be utilized. By postponing ballast maintenance, the overall maintenance costs can be reduced, as the ballast can be used for a longer duration before replacement becomes necessary.

Table 7 shows that the final RL policy was the best in avoiding the Super-red condition but without having the maximum percentage in the Excellent condition. Since the goal is to increase the net profit of each section, it is not important in which condition the section stays the most, but rather the effect of being in each of the conditions. As shown in Figure 5, the Super-red condition has very high negative rewards that make avoiding it more important than being in Excellent condition. In addition to that, keeping the section in the best condition all the time may result in performing additional maintenance actions that require more costs and may result in performing maintenance before it is needed. This can result in reducing the remaining useful life as explained in the previous paragraph, which shortens the age of the section as shown in Figure 10a, and decreases the total rewards gained per episode as shown in Figure 8a. Besides, postponing the maintenance can increase the age of the section as shown in 10b, but it can lead to having negative consequences due to being in undesired conditions as shown in Figure 8b. On the other hand, the RL policy was able to increase the rewards to a mean and mode better than the other two policies without having any episodes with negative rewards by taking the maintenance decision just before reaching any negative consequences based on different features. This required having the actions distributed over the condition of the section as shown in Figure 9c in contrast to Policies A and B which show a concentration of actions in specific zones. This shows that the decision to perform maintenance actions is not only a function of the section condition but also of the settlement rate and the maintenance history.

The RL policy outperforms both policies A and B in terms of rewards and in terms of avoiding the Super-Red condition. It can be concluded from the results that the effect of the maintenance action on the remaining useful life was more important than their costs. The RL agent was trying to avoid the maintenance action until the condition becomes unacceptable in order to use the section for producing revenues as much as possible before the section moves to the next stage, which is the after-maintenance stage. Each maintenance can be seen as a new beginning that has its own revenues before the losses start, so performing another maintenance before all the revenues after the first maintenance are harvested was like losing them. At the same time, the RL agent was successfully able to avoid the risk of being in the Super-Red condition as can be seen from Figure 9 and Table 7. This was due to the intelligent strategy shown in Figure 7, which considered the condition and the settlement rate. The figure shows that even if the condition was Excellent but the settlement is fast, the decision was to perform the maintenance for some occasions. This decision may result in a great reduction in the age of the section, but it also results in avoiding any risk of being in the Super-Red condition.

The outcomes presented in this paper are highly influenced by the reward functions that have been assigned. These functions, which are based on estimations and expert opinions, describe the costs and revenues involved, allowing for the calculation of the net profit. However, the accuracy of the simulations and results could be improved if more precise information regarding the costs and revenues associated with railway transport were available. This could potentially result in changes to the findings.

Nevertheless, the paper provides a reliable method for optimizing operation and maintenance based on the currently available data. However, to enhance the paper's findings further, it would be beneficial to incorporate the impact of maintenance activities of all railway components, such as infrastructure, superstructure, signalling, and catenary. By including this level of detail, the model's accuracy and precision can be significantly improved. Such a comprehensive analysis could be addressed in future studies.

An improvement can be made to the methodology, which is using *function approximation* RL methods, e.g. Deep Reinforcement learning. This can help in avoiding the discretisation process of the states and considering continuous states instead. For example, the condition of the section can still be a feature that describes the condition of the environment, but it will take the SD as a continuous variable argument instead of dividing the condition into several groups based on SD. This wipes out the need to use thresholds, which can result in further improvements in terms of taking the decisions at more specific states instead of having the same decisions for wide intervals of values.

In addition, the use of prognostic methods can be important in predicting the remaining useful life of the section based on its current state. This can be included as a feature in the RL environments to improve decision-making and can be more realistic than including the settlement rate which is difficult to measure in real-life applications.

8. Conclusions

An iPN model was created for the maintenance and operation of railway sections while focusing on optimising the maintenance of the ballast. This model is able to find the optimal maintenance strategy that can reduce the risk of being in undesired conditions while increasing revenues and decreasing costs.

This paper also proposes several ideas to improve the computational efficiency of the model. A method to divide the PN into several subnets was proposed and found to be successful in reducing computational costs. Besides, each section of the railway was considered a separate environment that has its own RL elements. This allows the RL agents to focus only on the important aspects when taking the decisions of each section and neglecting unnecessary information, which reduces the number of RL states. This, in turn, facilitates experience sharing between RL agents relating to sections of similar characteristics.

The model was applied to a practical problem and it shows the ability to reach an optimum maintenance strategy. The results show that it is crucial to avoid unnecessary maintenance actions because they can reduce the ballast age. This is because tamping and stoneblowing actions play a direct role in ballast fouling, which requires replacement once it becomes highly fouled. At the same time, the maintenance should be done before any risk of reaching a bad condition in order to avoid downtime or safety risks. A maintenance plan that gives the optimum decision as a function of various features of the railway section was found. This was able to avoid undesired conditions while increasing the age of each section and increasing the net profits per life of each section.

Acknowledgements

This paper is part of the ENHAnCE ITN project (`https://www.h2020-enhanceitn.eu/`) funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957. The authors would also like to thank the funding provided by the National Research and Science Council of Spain under grant agreements No. RTI2018-101841-B-C21 and PLEC2021-007798. The first author would like to acknowledge the University of Nottingham (UK) for hosting while conducting this study.

References

[1] Judiciary of England and Wales, Office of rail regulation v network rail infrastructure limited., Technical Report, Judiciary of England and Wales, 2012. URL: `https://www.judiciary.uk/wp-content/uploads/JCO/Documents/Judgments/office-rail-regulation-network-rail-sentencing-remarks-04042012.pdf`.

[2] E. T. Selig, J. M. Waters, Track geotechnology and substructure management, Thomas Telford, 1994.

[3] J. Sadeghi, M. Motieyan-Najar, J. Zakeri, B. Yousefi, M. Mollazadeh, Improvement of railway ballast maintenance approach, incorporating ballast geometry and fouling conditions, Journal of Applied Geophysics 151 (2018) 263–273.

[4] Y. Guo, V. Markine, G. Jing, Review of ballast track tamping: Mechanism, challenges and solutions, Construction and Building Materials 300 (2021) 123940.

[5] M. Daddow, X. Zhang, H. Qiu, Z. Zhang, Y. Liu, A mathematical model for ballast tamping decision making in railway tracks, Civil Engineering Journal 6 (2020) 2045–2057.

[6] M. Sedghi, O. Kauppila, B. Bergquist, E. Vanhatalo, M. Kulahci, A taxonomy of railway track maintenance planning and scheduling: A review and research trends, Reliability Engineering & System Safety 215 (2021) 107827.

[7] J. Chiachío, M. Chiachío, D. Prescott, J. Andrews, A knowledge-based prognostics framework for railway track geometry degradation, Reliability Engineering & System Safety 181 (2019) 127–141.

[8] A. Falamarzi, S. Moridpour, M. Nazem, A review of rail track degradation prediction models, Australian Journal of Civil Engineering 17 (2019) 152–166.

[9] L. F. Caetano, P. F. Teixeira, Optimisation model to schedule railway track renewal operations: a life-cycle cost approach, Structure and Infrastructure Engineering 11 (2015) 1524–1536.

[10] C. Letot, I. Soleimanmeigouni, A. Ahmadi, P. Dehombreux, An adaptive opportunistic maintenance model based on railway track condition prediction, IFAC-PapersOnLine 49 (2016) 120–125.

[11] L. F. Caetano, P. F. Teixeira, Availability approach to optimizing railway track renewal operations, Journal of Transportation Engineering 139 (2013) 941–948.

[12] Z. Su, A. Jamshidi, A. Núñez, S. Baldi, B. De Schutter, Integrated condition-based track maintenance planning and crew scheduling of railway networks, Transportation Research Part C: Emerging Technologies 105 (2019) 359–384.

[13] J. Andrews, D. Prescott, F. De Rozières, A stochastic model for railway track asset management, Reliability Engineering & System Safety 130 (2014) 76–84.

[14] D. Prescott, J. Andrews, Investigating railway track asset management using a markov analysis, Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit 229 (2015) 402–416.

[15] S. Sharma, Y. Cui, Q. He, R. Mohammadi, Z. Li, Data-driven optimization of railway maintenance for track geometry, Transportation Research Part C: Emerging Technologies 90 (2018) 34–58.

[16] S. Bressi, J. Santos, M. Losa, Optimization of maintenance strategies for railway track-bed considering probabilistic degradation models and different reliability levels, Reliability engineering & system safety 207 (2021) 107359.

[17] J. D. Campbell, J. V. Reyes-Picknell, H. S. Kim, Uptime: Strategies for excellence in maintenance management, CRC Press, 2015.

[18] M. Dell'Orco, M. Ottomanelli, L. Caggiani, D. Sassanelli, New decision support system for optimization of rail track maintenance planning based on adaptive neurofuzzy inference system, Transportation research record 2043 (2008) 49–54.

[19] A. Saleh, D. Prescott, R. Remenyte, M. Chiachio, An optimized asset management petri net model for railway sections (2023).

[20] M. Chiachío, A. Saleh, S. Naybour, J. Chiachío, J. Andrews, Reduction of petri net maintenance modeling complexity via approximate bayesian computation, Reliability Engineering & System Safety 222 (2022) 108365.

[21] M. Chiachío, J. Chiachío, D. Prescott, J. Andrews, A new paradigm for uncertain knowledge representation by plausible Petri nets, Information Sciences 453 (2018) 323–345.

[22] J. Lee, K. Liu, W. Chiang, Modeling uncertainty reasoning with possibilistic Petri nets, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 33 (2003) 214–224.

[23] B. Pei-Ming, Learning capability in fuzzy Petri nets based on bp net [j], Chinese Journal of Computers 5 (2004).

[24] M. M. Hanna, A. Buck, R. Smith, Fuzzy Petri nets with neural networks to model products quality from a cnc-milling machining centre, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 26 (1996) 638–645.

[25] A. Saleh, M. Chiachío, J. F. Salas, A. Kolios, Self-adaptive optimized maintenance of offshore wind turbines by intelligent petri nets, Reliability Engineering & System Safety (2022) 109013.

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

[27] L. Busoniu, R. Babuska, B. D. Schutter, Multi-agent reinforcement learning: An overview, Innovations in multi-agent systems and applications-1 (2010) 183–221.

[28] A. Senderovich, A. Shleyfman, M. Weidlich, A. Gal, A. Mandelbaum, To aggregate or to eliminate? optimal model simplification for improved process performance prediction, Information Systems 78 (2018) 96–111. URL: https://www.sciencedirect.com/science/article/pii/S0306437916306329. doi:https://doi.org/10.1016/j.is.2018.04.003.

[29] A. Senderovich, A. Rogge-Solti, A. Gal, J. Mendling, A. Mandelbaum, S. Kadish, C. Bunnell, Data-driven performance analysis of scheduled processes, Intl. Conf. on Business Process Management, Springer (2016) 35 â 52. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85123627763&partnerID=40&md5=83e184e57570ad913561efc6fef53a85, cited by: 1.

[30] F. M. BÃ¸nneland, J. Dyhr, P. G. Jensen, M. Johannsen, J. Srba, Stubborn versus structural reductions for petri nets, Journal of Logical and Algebraic Methods in Programming 102 (2019) 46–63. URL: https://www.sciencedirect.com/science/article/pii/S235222081830035X. doi:https://doi.org/10.1016/j.jlamp.2018.09.002.

[31] V. Padma, A study on flexible manufacturing systems using petri net reduction, in: IOP Conference Series: Materials Science and Engineering, volume 1130, IOP Publishing, 2021, p. 012060.

[32] B. Berthomieu, D. Le Botlan, S. Dal Zilio, Counting petri net markings from reduction equations, International Journal on Software Tools for Technology Transfer 22 (2020) 163–181.

[33] E. I. Basri, I. H. A. Razak, H. Ab-Samat, S. Kamaruddin, Preventive maintenance (pm) planning: a review, Journal of Quality in Maintenance Engineering 23 (2017) 114–143.

[34] M. Shafiee, J. D. Sørensen, Maintenance optimization and inspection planning of wind energy assets: Models, methods and strategies, Reliability Engineering & System Safety 192 (2019) 105993.

[35] R. D. Palmer, Maintenance planning and scheduling handbook, McGraw-Hill Education, 2013.

[36] S. Duffuaa, K. Al-Sultan, Mathematical programming approaches for the management of maintenance planning and scheduling, Journal of Quality in Maintenance Engineering 3 (1997) 163–176.

[37] J. S. Lee, I. Y. Choi, I. K. Kim, S. H. Hwang, Tamping and renewal optimization of ballasted track using track measurement data and genetic algorithm, Journal of Transportation Engineering, Part A: Systems 144 (2018) 04017081.

[38] M. Movaghar, S. Mohammadzadeh, Intelligent index for railway track quality evaluation based on bayesian approaches, Structure and Infrastructure Engineering 16 (2020) 968–986.

[39] C. Meier-Hirmer, A. Senee, G. Riboulet, F. Sourget, M. Roussignol, et al., A decision support system for track maintenance, Computers in railways X (2006) 217–226.

[40] C. Vale, I. M. Ribeiro, R. Calçada, Integer programming to optimize tamping in railway tracks as preventive maintenance, Journal of Transportation Engineering 138 (2012) 123–131.

[41] E. Gustavsson, Scheduling tamping operations on railway tracks using mixed integer linear programming, EURO Journal on Transportation and Logistics 4 (2015) 97–112.

[42] H. Khajehei, A. Ahmadi, I. Soleimanmeigouni, A. Nissen, Allocation of effective maintenance limit for railway track geometry, Structure and Infrastructure Engineering 15 (2019) 1597–1612.

[43] T. Satiennam, T. Phanyakit, Fuzzy multi-attribute decision making for the selection of a suitable railway track maintenance plan: A case study in thailand, GEOMATE Journal 17 (2019) 96–104.

[44] M. S. Kovačević, M. Bačić, I. Stipanović, K. Gavin, Categorization of the condition of railway embankments using a multi-attribute utility theory, Applied Sciences 9 (2019) 5089.

[45] L. Podofillini, E. Zio, J. Vatn, Risk-informed optimisation of railway tracks inspection and maintenance procedures, Reliability Engineering & System Safety 91 (2006) 20–35.

[46] M. H. Bin Osman, S. Kaewunruen, A. Jack, Optimisation of schedules for the inspection of railway tracks, Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit 232 (2018) 1577–1587.

[47] T. Lidén, M. Joborn, An optimization model for integrated planning of railway traffic and network maintenance, Transportation Research Part C: Emerging Technologies 74 (2017) 327–347.

[48] A. R. Albrecht, D. M. Panton, D. H. Lee, Rescheduling rail networks with maintenance disruptions using problem space search, Computers & Operations Research 40 (2013) 703–712.

[49] D. Arenas, P. Pellegrini, S. Hanafi, J. Rodriguez, Timetable rearrangement to cope with railway maintenance activities, Computers & Operations Research 95 (2018) 123–138.

[50] H. Khajehei, M. Haddadzade, A. Ahmadi, I. Soleimanmeigouni, A. Nissen, Optimal opportunistic tamping scheduling for railway track geometry, Structure and Infrastructure Engineering 17 (2021) 1299–1314.

[51] R. Santos, P. F. Teixeira, Heuristic analysis of the effective range of a track tamping machine, Journal of infrastructure systems 18 (2012) 314–322.

[52] C. Esveld, C. Esveld, Modern railway track, volume 385, MRT-productions Zaltbommel, 2001.

[53] M. Sasidharan, M. Burrow, G. Ghataora, A whole life cycle approach under uncertainty for economically justifiable ballasted railway track maintenance, Research in Transportation Economics 80 (2020) 100815.

[54] M. Daddow, X. Zhang, H. Qiu, Z. Zhang, Impact of unused life for track sections and available workforce in scheduling tamping actions on ballasted tracks, KSCE Journal of Civil Engineering 21 (2017) 2403–2412.

[55] C. Dao, R. Basten, A. Hartmann, Maintenance scheduling for railway tracks under limited possession time, Journal of Transportation Engineering, Part A: Systems 144 (2018) 04018039.

[56] T. Lidén, T. Kalinowski, H. Waterer, Resource considerations for integrated planning of railway traffic and maintenance windows, Journal of rail transport planning & management 8 (2018) 1–15.

[57] G. Budai, D. Huisman, R. Dekker, Scheduling preventive railway maintenance activities, Journal of the Operational Research Society 57 (2006) 1035–1044.

[58] M. Forsgren, M. Aronsson, S. Gestrelius, Maintaining tracks and traffic flow at the same time, Journal of Rail Transport Planning & Management 3 (2013) 111–123.

[59] X. Luan, J. Miao, L. Meng, F. Corman, G. Lodewijks, Integrated optimization on train scheduling and preventive maintenance time slots planning, Transportation Research Part C: Emerging Technologies 80 (2017) 329–359.

[60] C. Zhang, Y. Gao, L. Yang, Z. Gao, J. Qi, Joint optimization of train scheduling and maintenance planning in a railway network: A heuristic algorithm using lagrangian relaxation, Transportation Research Part B: Methodological 134 (2020) 64–92.

[61] A. Consilvio, A. Di Febbraro, R. Meo, N. Sacco, Risk-based optimal scheduling of maintenance activities in a railway network, EURO journal on transportation and logistics 8 (2019) 435–465.

[62] A. Bakhtiary, J. A. Zakeri, S. Mohammadzadeh, An opportunistic preventive maintenance policy for tamping scheduling of railway tracks, International Journal of Rail Transportation 9 (2021) 1–22.

[63] J. Zhao, A. Chan, M. Burrow, Reliability analysis and maintenance decision for railway sleepers using track condition information, Journal of the Operational Research Society 58 (2007) 1047–1055.

[64] L. C. Sancho, J. A. Braga, A. R. Andrade, Optimizing maintenance decision in rails: A markov decision process approach,

ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 7 (2021) 04020051.

[65] R. Mohammadi, Q. He, A deep reinforcement learning approach for rail renewal and maintenance planning, Reliability Engineering & System Safety (2022) 108615.

[66] T. Gao, Z. Li, Y. Gao, P. Schonfeld, X. Feng, Q. Wang, Q. He, A deep reinforcement learning approach to mountain railway alignment optimization, Computer-Aided Civil and Infrastructure Engineering 37 (2022) 73–92.

[67] N. Yousefi, S. Tsianikas, D. W. Coit, Reinforcement learning for dynamic condition-based maintenance of a system with individually repairable components, Quality Engineering 32 (2020) 388–408.

[68] J. Sresakoolchai, S. Kaewunruen, Railway infrastructure maintenance efficiency improvement using deep reinforcement learning integrated with digital twin based on track geometry and component defects, Scientific Reports 13 (2023) 2439.

[69] C. J. Watkins, P. Dayan, Q-learning, Machine learning 8 (1992) 279–292.

[70] T. Murata, Petri nets: Properties, analysis and applications, Proceesings of the IEEE 77 (1989) 541–580.

[71] K. Jensen, G. Rozenberg, High-level Petri nets: theory and application, Springer Science & Business Media, 2012.

[72] L. S. Shapley, Stochastic games, Proceedings of the national academy of sciences 39 (1953) 1095–1100.

[73] M. L. Littman, Markov games as a framework for multi-agent reinforcement learning, in: Machine learning proceedings 1994, Elsevier, 1994, pp. 157–163.

[74] L. Busoniu, R. Babuska, B. De Schutter, A comprehensive survey of multiagent reinforcement learning, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38 (2008) 156–172.

[75] R. Lal, Encyclopedia of soil science, 11, CRC Press, 2006.

[76] J. Litherland, J. Andrews, A petri net asset management framework for railway switches and crossings, Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit (2022) 09544097221110970.

[77] B. Aursudkij, A laboratory study of railway ballast behaviour under traffic loading and tamping maintenance, Ph.D. thesis, University of Nottingham Nottingham, UK, 2007.

[78] K. Tzanakakis, Track Settlements, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 147–148. URL: `https://doi.org/10.1007/978-3-642-36051-0_20`. doi:`10.1007/978-3-642-36051-0_20`.

[79] S. Kaewunruen, M. Ishida, S. Marich, Dynamic wheel–rail interaction over rail squat defects, Acoustics Australia 43 (2015) 97–107.

[80] R. Smith, Railway fatigue failures: an overview of a long standing problem, Materialwissenschaft und Werkstofftechnik: Entwicklung, Fertigung, Prüfung, Eigenschaften und Anwendungen technischer Werkstoffe 36 (2005) 697–705.

[81] A. Wilson, RAIL DEFECTS HANDBOOK, 2019.

[82] S. Kumar, Study of rail breaks: associated risks and maintenance strategies, Luleå tekniska universitet, 2006.