International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Driven Metro Induction Planning and Scheduling

Samyak K. Patil[1], Piyush R. Naidu[2], Tanmay U. Deshmukh[3], Md. Sadique Sheikh[4], Dr. Pooja Jaiswal[5] and Dr. Sudhir N. Shelke[6]

[1,2,3]Department of Computer Science & Engineering (Data Science), Guru Nanak Institute of Technology, Nagpur, Maharashtra, India

[4]Assistant Professor (Project Guide), Department of Electronics & Telecommunication Engineering, Guru Nanak Institute of Technology, Nagpur, India

[5]Head of the Department (HOD), Department of Computer Science & Engineering (Data Science), Guru Nanak Institute of Technology, Nagpur, India

[6]Principal, Guru Nanak Institute of Technology, Nagpur, India

Abstract—A metro depot has a few hours between the last service of the day and the 05:30 hrs commencement to decide what sets to put on duties the next day forming the rolling stock induction plan, and what to send to maintenance bays to await stand-by. Many networks employ whiteboards, logbooks and spreadsheets to make this decision, taking into account a myriad of constraints like safety regulations, wheel and bogie replacements, advertiser contracts and the stabling yard layout simultaneously. In this paper we examine works on rail scheduling, constraint programming and predictive maintenance, comparing against a planner built for the 23-trainset fleet of Nagpur Metro Rail Corporation Limited. The system employs Google's OR-Tools CP-SAT solver and supervised failure-risk model. For the two-line 6-bay 23-set problem the solver, including model build, took 23.5 ms on average and 30 ms at most over 13 stored plans, all proven optimal. Over 60-day simulations it reduced the mileage range across the fleet by 34.6% relative to a simulated manual baseline and violated no safety regulation in our tests. Works in the literature show the trend away from stand-alone dispatching tools towards systems which can demonstrate and justify their decisions. We identify areas for future work—yard sensing, rescheduling within the day, adaptation to other networks and energy-aware stabling.

Keywords—Rolling stock induction, urban rail transit, constraint satisfaction programming, CP-SAT, predictive maintenance, fleet mileage equalisation, depot operations, shunting optimisation.

I. INTRODUCTION

Passengers judge the metro by its punctuality. With headways of only a few minutes, there is little room for error. A breakdown in the middle of a corridor results in congestion since the following trains have to brake to a standstill, terminal platforms become sidelines and the metro's reputation is damaged. A freight or long-haul railway can cope with such a problem but a metro cannot.

The creditable performance of metro operations is largely due to work that takes place during the night. When the passenger services are withdrawn, late at night, all the trains are brought into the depot. The depot staff has a few hours to solve the induction problem since the first trains must depart for the service at 05:30. Every train must be assigned to one of several possible future activities (Fig. 1). The simplest one is to be placed on a service corridor, for instance, the North-South Line 1 or the East-West Line 2, and to contribute to the mileage of the operating company. Another option is to allocate the train for a maintenance bay where it will undergo a statutory overhaul, a bogie replacement, air conditioning service or defect detection and repair. Finally, the train can be held in a stabling track as a cold spare ready to replace a broken-down train within 15 minutes.

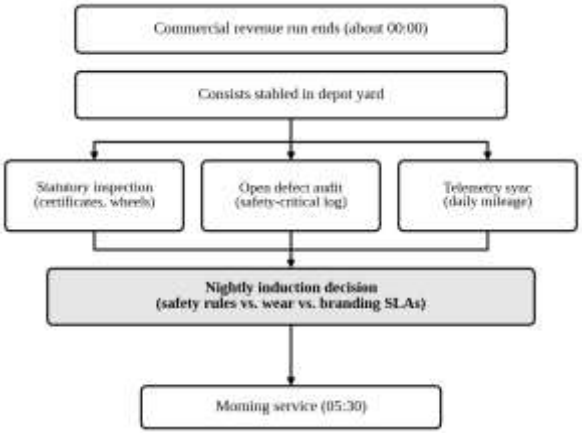


Fig. 1. The nightly induction cycle of a metro depot.

In common, the paper job-cards, logbooks and excel sheets are employed for this activity but possess four shortcomings. Firstly, the space to search is large to tackle manually. With 23 trains, each having three possible states, it can include about 94 billion different combinations per night, and railway traffic controllers are forced to use heuristics, which are frequently suboptimal. Secondly, the utilization is uneven: sets of trains which have acquired a good reputation in the last few days will be exploited more intensively, and after several months, certain wheel sets and traction motors will be exposed to significantly more wear than others, causing premature replacements of entire bogies and imbalances in the workshop loads. Thirdly, the contracts become problematic: since the money from exterior advertising wraps is considerable, grounding a train carrying such a wrap on the day when the advertising company has planned to utilise its visibility for promotion can involve sizable compensations to the advertising concern. Fourthly, the handmade roster lacks an audit trail; thus, when a new planner starts his service, it is not always clear why the preceding dispatch did not heed a particular safety rule, since the necessary records may be scattered across multiple logbooks.

The paper illustrates how the operations research, predictive maintenance and depot shunting areas have attempted to tackle the issue, and where their efforts failed to converge. It proceeds

to describe a planner for the Nagpur Metro twin-line network [1], which harmonizes them, and specifies the benefits which it provides, as well as those which it does not. Section II reviews the relevant literature, Section III describes the architecture, Section IV formulates the problem, Section V compares different approaches, Section VI evaluates the results and Section VII draws conclusions.

II. LITERATURE REVIEW

In addition to each other, each additional train set adds another dimension of possible options that all contradict each other. Exact procedures can guarantee an optimal solution but can take a long time to converge to one, while fast approximations called heuristics cannot guarantee anything. The choice of the solver is therefore just as important as the model itself and Table I shows the most relevant literature to the nightly induction problem.

TABLE I
Related Work Against the Nightly Induction Problem

Work	Focus	Relevance to induction
[2]–[4]	MILP circulation, maintenance routing, unit combining	Fleet-level flow; assumes interchangeable units
[5]–[7]	Integrated, multi-depot and survey studies	Joint planning; yard and per-train detail are secondary
[8], [9], [15]	Train unit shunting and depot servicing	Conflict-free yard moves; not tied to risk or wear
[10]–[12]	Telemetry classifiers, maintenance strategy and condition monitoring	Risk scores exist but stay outside the roster tool
[13], [14]	CP-SAT solver; scikit-learn	Building blocks of the planner reviewed here

A. Operations Research in Rail Scheduling

The earliest models assumed rolling stock as a question of fleet size and flow. Cordeau et al. [2] model the simultaneous assignment of locomotives and cars to passenger trains as a multicommodity network flow with maintenance constraints, Maróti and Kroon [3] address maintenance routing for train units, and Fioole et al. [4] add rules for combining and splitting units at terminals. Later studies integrate rolling stock scheduling with maintenance routing [5], [6] integrates vehicle and crew scheduling across multiple depots in mass transit, and [7] surveys passenger railway optimization.

These formulations assume that units are nearly identical and that depots are uniform. A metro consist has its own advertising contract, defect history, wheel profile and position in the yard, and solve times climb quickly once physical limits such as blocked stabling tracks enter the model. As a result classical formulations are run offline, not as part of the night shift.

B. Predictive Maintenance

Maintenance practice has evolved from reactive servicing of failed equipment, to time-based and condition-based maintenance, and finally to the concept of predictive maintenance, where failure is anticipated; [11] compares such maintenance strategies for rolling stock. Supervised classifiers such as support vector machines, random forests and logistic regression can be trained to indicate wear from a rolling-window telemetry on traction-converter or compressors; data-driven predictive maintenance for the railway industry is surveyed in [10].

Such predictions are often not directly useful, being hidden in an engineering portal, far from the tool that produces the roster (see, e.g., the axle-bearing condition-monitoring system in [12]). A dashboard may show that a train has a seven-day breakdown probability of 40%, but the night dispatcher, who does not see the number, will schedule it for the following day's peak hour. The maintenance team cares about the score, the depot staff about the roster, but neither controls the other.

C. Shunting and Depot Geometry

Trains in the depot are parked end to end in narrow sidings, like cars in a single driveway. Freling et al. [8] introduced the problem of finding conflict-free shunting plans for train units, and Haahr et al. [9] benchmark exact and heuristic methods for it; the problem combines several NP-hard subproblems, so exact methods scale poorly to large yards. Shi et al. [15] reschedule shunting, servicing and routing at a high-speed railway depot after disturbances. If the roster instructs to release the inner train and keep the outer one, drivers must perform shunting maneuvers to free the sidings. This is dangerous, wasting power and time, and can cause collisions in the yard. A roster that does not respect the geometry of the yard can be optimal on paper but impossible to execute at 04:30.

D. Choice of Solver

MILP solvers take advantage of the fact that relaxing the integrality constraints produces a convex program. They then proceed to partition the search space (branching) and tighten the relaxation (cutting planes), until a provably optimal solution is found. They are often faster than other methods for problems with smooth costs and constraints, but can be outperformed by constraint programming on highly-discrete problems. Constraint programming solvers work by pruning the search space, based on the constraints of the problem. They are typically good at finding feasible solutions, but not so at proving optimality. CP-SAT [13], the solver we exploit, combines the two approaches, implementing sophisticated clause learning from failed assignments, akin to SAT solving. This is ideally suited for problems where the solution consists of many binary decisions, such as our unit-level choices for a single-night roster.

E. Research Gaps

There are three major research gaps in the literature. First, optimization studies are typically focused on a single aspect: long-haul maintenance scheduling, or line coverage, or yard shunting. There is little work on joint optimization over maintenance, wear equalization, and branding. Second, most solvers do not provide sufficient evidence for their decisions, leading to mistrust by human planners who then tweak the suggested rosters. Third, industrial-strength MIP solvers typically take minutes to find a solution, making interactive planning difficult: if a breakdown occurs during the night shift, the controller cannot quickly explore alternate scenarios.

F. Trust in Automated Planning

Human factors research has identified a further challenge unique to planning problems: the trust operators place in the recommendations [16], [17]. There are two ways for trust to be misplaced. Planners who have full confidence in an automated system stop monitoring its suggestions, implicitly assuming that no error will occur [16]. As a result, they fail to spot incorrect rosters. By contrast, operators who have lost trust due to one overly aggressive suggestion will routinely override the planner, ignoring the automated recommendation [16]. Ideally, operators should have calibrated trust in a learning-based planner, updating their beliefs based on evidence [17]. This requires that they understand the decision logic, and have experimented with different settings for the model themselves and seen results. Unfortunately, optimization research has neglected the question of transparency and control, focusing instead on solution quality. As a result, planners are often unaware of the assumptions encoded in a solver, and do not formally approve rosters. We

took transparency seriously, specifying sign-off requirements in our planning system.

III. SYSTEM ARCHITECTURE

We examine the Metro Induction Planner, built for Nagpur Metro's 23 three-car trainsets on Line 1 (Orange) and Line 2 (Aqua), which are stabled at the Khapri and Lokmanya Nagar (LON) depots [1]. Data enters at the top layer and two engines process it in the middle. An explanation layer at the bottom leaves the final say with a person (Fig. 2).

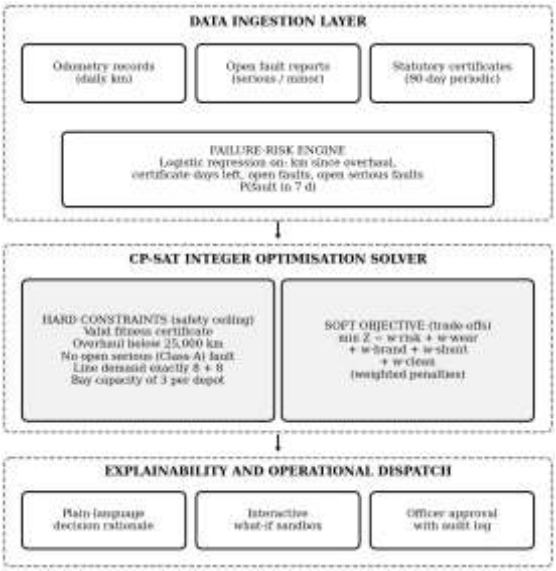


Fig. 2. Layered architecture of the induction planner.

A. Data Ingestion

Three inputs are loaded every night. Odometry records give the daily kilometres of each trainset from which we derive mileage since the last overhaul. Open defect logs classify faults as serious (Class A, safety-critical) or minor. Statutory certificates renew every 90 days and their validity can be calculated from the remaining days of validity. Branding contracts, stabling positions and wash records provide the soft preferences.

B. Failure-Risk Engine

The first engine predicts the probability of every train set's failure during the following week. The algorithm used is logistic regression [14]. As a result, we have the values from 0 to 1. We trained the model on four features: kilometres since the last overhaul, days left on the fitness certificate, the number of open fault reports, and the number of open serious faults. Using a more complex algorithm might increase the accuracy of our model; however, we preferred logistic regression. First, by using logistic regression, we can explain how we predict failure by looking at the feature's weight, which is critical for the explanation use-case. Secondly, we need to score train sets in real-time.

C. Constraint Programming Solver

The second engine is a constraint programming (CP) solver called CP-SAT [13]. It reasons about the problem differently: hard constraints are never violated, and soft preferences are weighted penalties that the solver tries to avoid. Rules like certificate expiration are encoded as hard constraints. Preferences like mileage evenness are encoded as weighted penalties, and the weights determine the trade-offs between the different objectives. CP-SAT is designed to find optimal solutions so we used a timeout-based approach that either finds the best plan or the best plan so far. Finally, CP-SAT requires all variables to be integers. We therefore scaled all risk scores and penalties appropriately.

We did not train a model on historical rosters, for example a neural network. Past rosters encode human preferences including the irregular patterns that we want to eliminate and a learned model cannot guarantee to avoid rule violations. We reserved the right to encode hard constraints in a CP-SAT and limited learning to risk prediction. This way each component focused on its specialty.

D. Explanation and Dispatch

Every choice has a natural language explanation and there is always a sandbox for the controller to experiment with the alternatives. Human oversight is paramount and we designed explanations to gain trust and facilitate questioning of the tool's suggestions. The final approval always requires an explicit officer decision, recorded in an append-only audit log, which links every choice to a person. This achieves what the manual rosters in Section I could not: an audit trail that documents every automated choice's rationale.

E. Nightly Workflow

The nightly optimization follows a strict protocol. After stabling, the three feeds get synchronised, the fleet rescored, a first plan found by the solver with the weights of the selected preset (Section V). This takes a fraction of a second, so the rest of the overnight window is for the controller to review the plan. She reads the explanations, plays in the sandbox and approves the plan before 05:30. If a defect is reported during the night, the system alerts the officer, who regenerates the plan from the updated data.

IV. PROBLEM FORMULATION

A. The Decision

Every night each of the N = 23 trainsets gets exactly one of the statuses in Fig. 1. If $x_{i,s} = 1$ is a binary variable that indicates that trainset i gets status s, then constraint (1) holds for every trainset i.

$$\Sigma_s x_{i,s} = 1 \text{ for every trainset } i. \quad (1)$$

There are three statuses per trainset ($3^{23} \approx 94$ billion combinations, as noted in Section I); the line served is chosen only for spare trainsets. Most are ruled out by the constraints below.

B. Hard Safety Constraints

Table II lists the hard constraints enforced by the solver. The first three constraints prevent a trainset from getting a revenue service if it has an open Class A defect or an expired certificate. The last two constraints ensure that the plan respects the stabling capacities and the timetable.

TABLE II
Hard Safety Constraints Enforced by the Solver

Constraint	Condition	Rule enforced
Fitness certificate	Certificate expired	No revenue service
Overhaul ceiling	25,000 km or more since overhaul	No revenue service
Class-A defect	Any open Class-A fault	No revenue service
Line demand	Each corridor	Exactly 8 trainsets on each line
Bay capacity	Each depot	At most 3 trainsets in the bays (Khapri, LON)

Class-A faults are the ones that are serious: traction motor overheating, loss of the signal receiver or an emergency brake loop. A train with any of these remains out of service until it can be repaired, regardless of how tight the timetable is.

C. Soft Objective

Among safe plans, the solver minimises a weighted sum of five penalties, as in (2):

$$\min Z = \Sigma_k w_k \Phi_k, k \in \{risk, wear, brand, shunt, clean\} \quad (2)$$

The risk penalty grows when a train with a high predicted probability of failure comes into service. The wear penalty grows with the mileage of each train sent out, thus favouring lower-mileage trains and balancing the fleet mileage. The branding penalty is paid when a wrapped train with an active visibility contract remains in the depot. The shunting penalty is incurred when an interior-parked train must depart while the train in front stays, thus forcing extra yard moves. The cleanliness penalty grows when a train four or more days overdue for the automated exterior wash is deployed.

D. Why a Weighted Sum

A weighted sum gives us the simplest possible reduction of several goals to one and its weights state how much one goal is worth relative to another, in raw numbers. The drawback is that someone has to pick them. We treat the weights as operating policy and ship the planner with named presets (Section V). Safety rules are external to the objective, so no choice of weights can redeem a safety violation.

E. Risk Scores as Probabilities

A train with a 0.12 risk is not necessarily due for maintenance; for a well-tuned model, that would imply that, on average, 1 out of every 8 similar trains would require maintenance next week. The hard threshold is therefore likely to lead to either too many false positives or too many false negatives; the planner has incorporated the risk as a cost for using the train, and the weights are determined by how much risk is being offset for using a particular train that night: only the most obvious cases, such as an open class-A fault, are ruled out by hard constraints.

V. COMPARISON WITH EXISTING APPROACHES

Table III compares the planner with the two main alternatives in the literature. We leave out telematics portals as they only display risk. They neither enforce rules nor build rosters.

TABLE III
Comparison of Rolling Stock Dispatching Approaches

Dimension	Manual roster	MILP [2], [3]	Induction planner
Solver	Human judgment	Branch-and-bound (CPLEX, Gurobi)	Google OR-Tools CP-SAT
Solve time	45–90 min (est.)	8–45 min (est.), network-wide	16–30 ms (solver only)
Safety guarantee	Prone to fatigue errors	Via linear constraints	Strict mathematical ceilings
Telemetry and risk	Informal defect logs	Decoupled	Closed loop: 7-day risk scores feed the objective
Mileage balance	Random drift	Coarse, macroscopic	Dynamic wear balancing
Depot geometry	Driver coordination	Ignored	Stabling depth and shunting penalties
Explainability	Verbal handover	Opaque variables	Plain-language reason per train
Operator override	Pencil marks	Needs a full re-solve	Live what-if with audit trail

Note: the manual (45–90 min) and MILP (8–45 min) solve times in Table III are estimates, not measurements made in this work, and no MILP baseline was run. The planner time is solver time (model build plus CP-SAT solve) recorded for 13 plans on the 23-trainset fleet.

The planner is fundamentally different in the source of its safety guarantee. The manual roster relies upon the controller's ability to recall all rules, and the MILP model upon there being a person who has written each rule down as a linear constraint. For the planner the same rules are hard constraints that may not be overruled by any weight. It also creates the loop from telemetry to the roster, so that a risk score becomes an input to a decision and is no longer a number on a dashboard. This fills the first research gap. Speed is important in terms of changing the tool's usage. A plan taking an hour is made once and rarely repeated, whereas one taking a fraction of a second allows the officer to explore what happens if a train is held or a bay is unavailable.

The weights can be switched to a set of presets (Table IV), so that the same engine favours safety in bad weather, wears its haulage evenly through a routine week, retains yesterday's yard positions in the face of night staff shortages. Branding, shunting and cleaning weights are administrator-set constraint parameters and can be raised, for example, during an advertising campaign. It is a human who selects the preset according to the coming day's needs. A festival week, a storm forecast and a staff shortage suggest a different one.

TABLE IV
Solver Weight Presets for Different Operating Strategies

Preset	w_{risk}	w_{wear}	w_{change}	Typical context
Safest service (default)	3	1	0	Severe weather, VIP runs, festival peaks
Even fleet wear	1	3	0	Standard mid-week balancing
Least change	1	1	1	Minimal change from the previous night's roster

Each of the following presets favours one aspect over the others. For instance, Safest Service makes a risky train unprofitable to deploy to leave it on standby or in the bays unless it can't be avoided due to the lack of an alternative. Even Fleet Wear allows slightly riskier trains to be put on mileage to equalize them. Least Change makes departures from the previous night's roster expensive. Branding exposure is raised through the administrator's branding weight, which keeps wrapped trains in service.

VI. RESULTS AND DISCUSSION

We evaluated the performance of our planner on a variety of synthetic stressing scenarios generated from Nagpur Metro fleet parameters.

A. Solver Performance

Over 13 plans generated by the application for the seeded 23-trainset fleet (stored in the project databases during development and testing), CP-SAT, including model build, took 23.5 ms on average and 30 ms at most (range 16–30 ms), and every plan was proven optimal (Fig. 3a). These are solver-only times from development and test runs (hardware not recorded); they exclude the API, database and risk-scoring steps and are not a controlled benchmark. It demonstrated its ability to handle tight discrete combinations of constraints, such as were found in the shunting geometries and bay capacities, although no MILP baseline was run here, so the MILP comparison in Table III is indicative only. Solve times for fleets larger than 23 train sets were not measured and are left for future work.

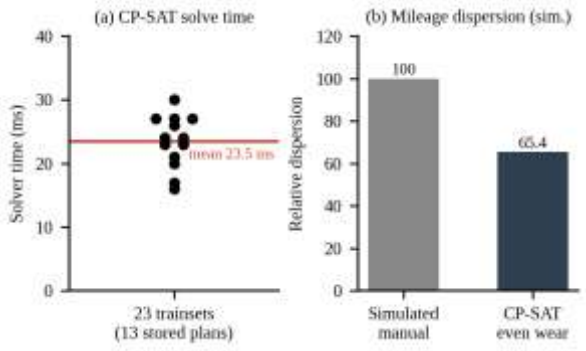


Fig. 3. (a) Solver time (model build plus CP-SAT solve) of 13 stored plans for the 23-trainset fleet. (b) Simulated mileage dispersion relative to a simulated manual roster (= 100).

B. Fleet Wear Balancing

The mileage range (between the most and least used train set) in a 23-train fleet was increased by an average of 412 km per operational week under the simulated manual scheduling. Using the Even Fleet Wear preset decreased mileage range by 34.6% over the course of 60 days (Fig. 3b) in the same simulated scenario (synthetic fleet data and a modelled manual baseline). As the wheels, bogies and traction motors on the sets indicate mileage-related wear, this has a direct impact on the time of overhaul for the given sets. The most intensively used sets will require maintenance earlier, while the remainder of the fleet will be subject to lesser usage. By equal distances, the sets to be overhauled first can be postponed and the usage of the workshops can be smoothed out predictably. Another advantage is that the interval between the re-profiling of the wheelset can be distributed between the families of sets, reducing the idle time of a workshop due to a local shortage of suitable rolling stock and risk of large-scale simultaneous flange wear-outs on a family of sets.

There is a practical limit to this benefit; trying to equalize the usage of all sets would spike the utilization of the workshops, as all sets would reach their overhaul limits at once. The penalty for unevenness in the wear profiles should therefore be adjusted according to the available capacity, which is the aim of the presets.

C. Transparency for Dispatchers

The suspicion of automated systems by the human dispatchers is a main barrier to their adoption, so the planner generates a natural language description of each recommendation for the evaluation by a human. For each set, it provides the dispatcher with a recommendation string, which describes the reason behind the suggestion simply. An example of such a string is 'NMR-04 assigned to Line 1: no open faults detected, low risk of breakdown (0.12), high priority branding SLA, front track position avoids shunting'. The controller can also manually assign a set to a line from standby and see the impact of the decision on the risk, wear, and brand metrics in real-time (Fig. 4). Dispatchers are alerted to potentially unsafe assignments, by the safety interlock; no weight setting can override a hard constraint.

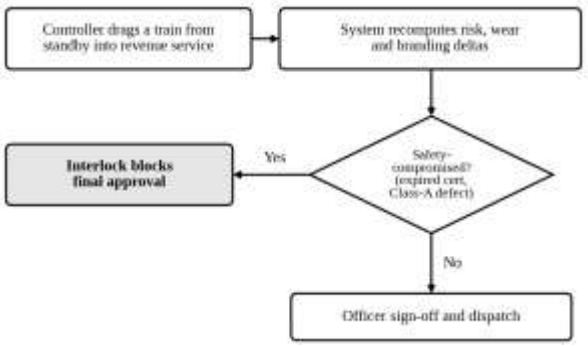


Fig. 4. What-if sandbox and safety interlock logic.

D. A Worked Scenario

Suppose that one of the line-1 open slots had three safe candidates: Train A has the lowest risk of failure but has already had an above-average number of services; B is a slightly riskier option with low utilization; and C has average risk and utilization but has a wrap and a contract. In accordance to the Safest Service policy, the planner would have picked A given the risk; in accordance to the Even Fleet Wear, B would be the best chances to balance the mileage; and with a high branding exposure weight, the preferred option would be C. This state of the fleet is similar, and therefore the only difference is the policy, which the reason string states. This scenario is illustrative, so no metrics are reported for it.

E. Deployment

The design imposes several limitations to the planners. The search is automated but the selection and the liability belong to the human operators, and the latter should program the weight presets since they define the value model for a single night's work. Effectiveness depends on the quality of the input since erroneous defect logs or yard location data will produce optimal results based on lies. In this regard, live yard sensing should be prioritized for the next phase. Adoption is a human factor; the officers that plan rosters manually every day understand the problem and the limitations of the current model better than the automated system. For example, planners know what sidings are closed on overcast mornings or what bays are suitable for major maintenance. The first software versions could be utilized to challenge the manual planning at every step thus prioritizing the constraints that the rules engine should encode while automating the adjustments. The friction with the planners could be channeled into model improvement by defining objective criteria.

F. Transfer to Other Networks

There is nothing special about the Nagpur network in this solution beyond the sheer size of it: the demand for the lines, bays' capacities, certificate rules, and overhaul cut-offs were all determined by local conditions. A region that has more lines, depots, or other constraints would have to adjust the demand, capacity, and certificate arrays and be mostly similar elsewhere. The same design principles would be applicable to any metro seeking to reduce the staff time spent on rostering and automate the planning of public transport services to a certain degree. The more significant challenge in adopting the solution elsewhere lies in the local logistics: every yard has its own planning practices, record-keeping conventions, and unofficial constraints that have to be entered into the software before it can be useful there. The planning time for larger fleets will inevitably increase; it was not measured here beyond 23 sets, but at tens of milliseconds for 23 sets there is considerable headroom against the estimated 45–90 min of manual planning every night (Table III).

G. Limitations

The solve times come from plans stored during development and testing rather than from a continued benchmark during operation and the 34.6% improvement is compared to a manual baseline which is also based on simulations. The risk model is trained on synthetic fault histories, so the results do not demonstrate its real-world performance and do not show that the branding compliant shunting and moves were in fact beneficial post automation. Positions within the yard still need to be manually inputted limiting the efficacy of the shunting logic. These aspects should be further elaborated on in the future work.

VII. CONCLUSION

Human heuristic operations within metro systems are nearing a tipping point both in the scale of the metro network and the complexity of the urban area. The Nagpur Metro case study shows that automated rostering with CP-SAT optimization and telemetry based risk assessment can produce dependable, auditable results in simulation without compromising the necessary safety headroom. The cases dealt with were strict safety based over-rided reasons (certificates, overhauls, Class-A defects) and there were no incidents on other criteria allowed for dynamic replanning within comparable limits within less than 30 ms of solver time in all the stored plans. This was achieved by an audit trail plus the written explanation and "what-if" analysis which led to the officer involvement being adequate without too many complicating factors: the officers assess the proposed plan, vote it through and approve for publication. An involved officer who understands the nuances of the case can question the suggestions made in a generic spreadsheet much better than a mere spreadsheet with figures. The future work on this topic could involve the adaptation to other networks and experimenting with the use of optical RFID readers to observe the consist information within the stabling bays. Another interesting avenue of future work would be energy-aware stabling to plan regeneration and battery thermal recovery for night storage thus allowing the metro concepts with batteries and no catenary to be shut down appropriately.

ACKNOWLEDGMENT

We extend our gratitude to Md. Sadique Sheikh for mentoring us as our project guide and Dr. Pooja Jaiswal for her valuable suggestions throughout this work. We also express our sincere gratitude to the Department of Computer Science & Engineering (Data Science), Guru Nanak Institute of Technology, Nagpur for providing the infrastructure that contributed to this research.

AUTHOR PROFILES

Samyak K. Patil, Piyush R. Naidu and **Tanmay U. Deshmukh** are pursuing their undergraduate studies in the Department of Computer Science & Engineering (Data Science), Guru Nanak Institute of Technology, Nagpur, Maharashtra, India. **Md. Sadique Sheikh** is working as an Assistant Professor in the Department of Electronics & Telecommunication Engineering, Guru Nanak Institute of Technology, Nagpur, Maharashtra, India. **Dr. Pooja Jaiswal** is the Head of the Department of Computer Science & Engineering (Data Science), Guru Nanak Institute of Technology, Nagpur, Maharashtra, India, and **Dr. Sudhir N. Shelke** is the Principal of Guru Nanak Institute of Technology (GNIT), Nagpur, Maharashtra, India.

REFERENCES

[1] The Metro Rail Guy, "Bids invited for Nagpur Metro Phase 2's 48 coach contract RS-01," Mar. 16, 2025. [Online]. Available: https://themetrorailguy.com/2025/03/16/bids-invited-for-nagpur-metro-phase-2s-48-coach-contract-rs-01/. Accessed: Oct. 19, 2026.

[2] J.-F. Cordeau, F. Soumis, and J. Desrosiers, "Simultaneous assignment of locomotives and cars to passenger trains," Oper. Res., vol. 49, no. 4, pp. 531–548, 2001.

[3] G. Maróti and L. Kroon, "Maintenance routing for train units: The transition model," Transp. Sci., vol. 39, no. 4, pp. 518–525, 2005.

[4] P.-J. Fioole, L. Kroon, G. Maróti, and A. Schrijver, "A rolling stock circulation model for combining and splitting passenger train units," Eur. J. Oper. Res., vol. 174, no. 2, pp. 1281–1297, 2006.

[5] Q. Zhong, R. M. Lusby, J. Larsen, Y. Zhang, and Q. Peng, "Rolling stock scheduling with maintenance requirements at the Chinese high-speed railway," Transp. Res. B, Methodol., vol. 126, pp. 24–44, 2019, doi: 10.1016/j.trb.2019.05.013.

[6] D. Huisman, R. Freling, and A. P. M. Wagelmans, "Multiple-depot integrated vehicle and crew scheduling," Transp. Sci., vol. 39, no. 4, pp. 491–502, 2005.

[7] A. Caprara, L. Kroon, M. Monaci, M. Peeters, and P. Toth, "Passenger railway optimization," in Handbooks in Operations Research and Management Science, vol. 14, C. Barnhart and G. Laporte, Eds. Amsterdam, The Netherlands: Elsevier, 2007, pp. 129–187, doi: 10.1016/S0927-0507(06)14003-7.

[8] R. Freling, R. M. Lentink, L. G. Kroon, and D. Huisman, "Shunting of passenger train units in a railway station," Transp. Sci., vol. 39, no. 2, pp. 261–272, 2005, doi: 10.1287/trsc.1030.0076.

[9] J. T. Haahr, R. M. Lusby, and J. C. Wagenaar, "Optimization methods for the train unit shunting problem," Eur. J. Oper. Res., vol. 262, no. 3, pp. 981–995, 2017.

[10] N. Davari, B. Veloso, G. de A. Costa, P. M. Pereira, R. P. Ribeiro, and J. Gama, "A survey on data-driven predictive maintenance for the railway industry," Sensors, vol. 21, no. 17, Art. no. 5739, 2021.

[11] Y.-H. Cheng and H.-L. Tsao, "Rolling stock maintenance strategy selection, spares parts' estimation, and replacements' interval calculation," Int. J. Prod. Econ., vol. 128, no. 1, pp. 404–412, Nov. 2010.

[12] E. Fumeo, L. Oneto, and D. Anguita, "Condition based maintenance in railway transportation systems based on big data streaming analysis," Procedia Comput. Sci., vol. 53, pp. 437–446, 2015.

[13] L. Perron and V. Furnon, "OR-Tools," Google LLC, 2024. [Online]. Available: https://developers.google.com/optimization. Accessed: Oct. 19, 2026.

[14] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[15] J. Shi, H. Li, D. Wang, and P. Vansteenwegen, "Reschedule the shunting and service scheduling plan after disturbances at a Chinese high-speed railway depot," IEEE Access, vol. 12, pp. 38615–38630, 2024, doi: 10.1109/ACCESS.2024.3368807.

[16] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," Hum. Factors, vol. 39, no. 2, pp. 230–253, 1997.

[17] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," Hum. Factors, vol. 46, no. 1, pp. 50–80, 2004.