

NRCan bridges the practicability gap in log transportation optimization

A. Abdellaoui, I. El Hallaoui, L. Benabbou, F. Aubé, M. Amazouz

G-2026-49

September 2026
Revised: October 2026

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Citation suggérée : A. Abdellaoui, I. El Hallaoui, L. Benabbou, F. Aubé, M. Amazouz (Septembre 2026). NRCan bridges the practicability gap in log transportation optimization, Rapport technique, Les Cahiers du GERAD G- 2026-49, GERAD, HEC Montréal, Canada. Version révisée: Octobre 2026

Suggested citation: A. Abdellaoui, I. El Hallaoui, L. Benabbou, F. Aubé, M. Amazouz (September 2026). NRCan bridges the practicability gap in log transportation optimization, Technical report, Les Cahiers du GERAD G-2026-49, GERAD, HEC Montréal, Canada. Revised version: October 2026

Avant de citer ce rapport technique, veuillez visiter notre site Web (<https://www.gerad.ca/fr/papers/G-2026-49>) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Before citing this technical report, please visit our website (<https://www.gerad.ca/en/papers/G-2026-49>) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026
– Bibliothèque et Archives Canada, 2026

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026
– Library and Archives Canada, 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Télec. : 514 340-5665
info@gerad.ca
www.gerad.ca

NRCan bridges the practicability gap in log transportation optimization

Abdelhakim Abdellaoui ^{a, c}

Issmail El Hallaoui ^{a, c}

Loubna Benabbou ^{b, c}

François Aubé ^d

Mouloud Amazouz ^d

^a *Department of Mathematics and Industrial Engineering, Polytechnique Montreal, Montréal (Qc), Canada, H3T 1J4*

^b *Department of Management Science, University of Quebec at Rimouski (UQAR), Rimouski (Qc), Canada, G5L 3A1*

^c *GERAD, Montréal (Qc), Canada, H3T 1J4*

^d *Natural Resources Canada (NRCan), CanmetENERGY-Varenes, Varennes (Qc), Canada, J3X 1P7*

abdelhakim.abdellaoui@polymtl.ca

issmail.elhallaoui@gerad.ca

loubna.benabbou@uqar.ca

September 2026

Revised: October 2026

Les Cahiers du GERAD

G–2026–49

Copyright © 2026 Abdellaoui, El Hallaoui, Benabbou, Aubé, Amazouz

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : Log transportation accounts for up to 40% of operating costs in the Canadian forestry sector (Audy et al. 2023), yet optimization tools for transportation planning have seen little adoption in practice. Planners set them aside because they omit critical operational constraints, cannot evolve as business constraints change, and offer no way to recover when disruptions invalidate the plan. Natural Resources Canada (NRCan) designed and validated an end-to-end decision support system (DSS) for large-scale log-truck routing and scheduling. At its core, the system brings together three analytics and optimization contributions. The first is an exact and comprehensive formulation of the multi-period routing and scheduling problem on a time-space network, solved at industrial scale through a tailored relax-and-fix/fix-and-optimize decomposition. The second is an implicit constraint learning framework that embeds trained machine learning predictors into the model as linear constraints, so that plans self-adjust to evolving planner preferences without manual re-engineering. The third is a dynamic reoptimization algorithm that combines local branching with reinforcement learning to recover near-optimal schedules under disruptions such as road closures, truck breakdowns, or travel delays, while staying close to the original plan. Deployed as a web-based platform within NRCan, the system is estimated, in its first industrial validation with a major Canadian forest company, to generate approximately \$8 million in annual savings and to avoid 4,800 tonnes of CO₂ per year. National adoption is projected to yield up to \$377 million in annual fuel savings and to avoid more than 500,000 tonnes of CO₂ per year, directly supporting Canada’s net-zero objectives.

Keywords : Decision support system; log transportation; routing and scheduling; machine learning; reinforcement learning; forestry

Acknowledgements: The authors are grateful to Natural Resources Canada, CanmetENERGY-Varenes, for funding and supporting this research, and for providing the operational data and field expertise that made this work possible. The authors used Claude to assist with language editing. All AI-assisted text was reviewed, revised, and verified by the authors, who take full responsibility for the final content.

1 Introduction

The Canadian forestry sector stands as one of the country’s most vital economic sectors. According to The State of Canada’s Forests: Annual Report 2024 (Natural Resources Canada 2024), nominal GDP from forest-related activities reached \$33.4 billion in 2024. Beyond its economic significance, the sector supports tens of thousands of jobs and plays a central role in Canada’s sustainable development agenda (Natural Resources Canada 2024). Yet despite its strategic importance, the industry faces mounting operational pressures. With nearly 226 million hectares of managed forest land (Natural Resources Canada 2024), harvesting operations span an immense geographic scale and require extensive transportation over long distances. Road accessibility is further complicated by seasonal constraints, as winter and spring conditions periodically restrict or degrade transportation routes. At the same time, growing regulatory pressure on environmental sustainability adds another layer of complexity to an already challenging operational setting (Abdellaoui et al. 2026b).

At the heart of these challenges lies log transportation, which represents up to 40% of total operating costs for a typical Canadian forest company (Audy et al. 2023). Given this economic weight, optimizing transportation is not merely an operational improvement but a strategic priority. The same lever also serves an environmental goal: every kilometer of travel eliminated reduces greenhouse gas emissions (Bektas and Laporte 2011), aligning operational gains with sustainability objectives. Driven by this dual motivation, the forestry sector is increasingly adopting digitization and advanced analytics to transform its logistics operations. At the national level, this movement is led by NRCan, which launched the *Digital Forest Value Chain Optimization* (DFVCO) initiative to support the industry’s operational efficiency and long-term profitability through optimization and data-driven analytics. The present work was developed within this initiative. It began with a diagnosis rather than a model: extensive discussions with Canadian forest companies revealed a clear and consistent picture. Practitioners had tested optimization tools before and largely set them aside, not because the mathematics was wrong, but because the tools were disconnected from operational reality (Weintraub et al. 1996, Andersson et al. 2008). Many other approaches never left the research stage and were validated only on small instances rather than on industrial-scale cases (El Hachemi et al. 2013, Amrouss et al. 2016, Melchiori et al. 2022). In both cases, the same limitations recur: existing solutions omit critical constraints, cannot adapt as business constraints evolve, and provide no way to recover when disruptions invalidate the established plan.

This operational reality points to a structural limitation. Optimization models rely on a predefined set of hard-coded constraints intended to represent the operational environment. In practice, real-world routing and scheduling problems rarely conform exclusively to these formally modeled constraints. Planners routinely follow additional unwritten conventions, context-dependent habits, resource-specific compatibilities, and tacit feasibility restrictions that evolve over time (Hewitt and Frejinger 2020, Pawlak and Krawiec 2017). Incorporating such evolving constraints typically requires modeling and operations research expertise that operational planners do not possess (Bessiere et al. 2017). As a result, mathematically optimal solutions are frequently rejected or manually adjusted before execution, introducing inefficiencies and inconsistencies that erode planner trust and, ultimately, the adoption of optimization-based solutions (Bayani et al. 2024).

Beyond this adaptability challenge, forestry operations are also exposed to real-time disruptions. Log transportation planning is typically organized in two levels: tactical planning defines medium-term schedules from static inputs such as road networks, demand forecasts, and supply availability, while operational planning executes these schedules day by day. In practice, however, unexpected events such as equipment breakdowns, road closures due to weather, demand fluctuations, or sudden changes in forest accessibility frequently render the established plan infeasible and force planners to revise it (Amrouss et al. 2016). Recovering from such disruptions requires rapid reoptimization that preserves feasibility while minimizing deviations from the tactical plan, a challenge that is both computationally demanding and largely underexplored in the log transportation literature (Audy et al. 2023).

To address these interconnected challenges, this paper presents an end-to-end DSS named **FOROP-T**, conceived as a human-in-the-loop analytics tool rather than a replacement for the planner in charge of log transportation planning. The system is structured around four integrated components, each targeting a distinct source of the practicability gap. The *kernel component* generates tactical plans for the log-truck routing and scheduling problem (*LTRSP*) by solving a large-scale mixed-integer linear program through tailored decomposition techniques. The *specific component* enforces the operational constraints particular to each forest company, allowing a single system to serve multiple partners. The *implicit component* learns the unwritten feasibility constraints that emerge and evolve over time, enabling the DSS to adjust itself without manual intervention. The *dynamic component* makes the system resilient to unforeseen events by rapidly reoptimizing plans while staying close to the original schedule.

The system is deployed as a web-based platform within NRCan and validated in real industrial operations with a Canadian forestry partner. The platform keeps the planner in the loop at every stage, and every manual adjustment feeds back into the system as data. In this initial testing, the system generated an estimated \$8 million in direct annual savings and a reduction of 4,800 tonnes of CO₂ emissions per year. At national scale, adoption is projected to yield up to \$377 million in annual fuel savings and to avoid more than 500,000 tonnes of CO₂ emissions, directly supporting Canada’s net-zero objectives (Government of Canada 2024).

This paper shows how optimization and analytics can overcome the practical limitations that have long kept log transportation tools from being adopted. Our DSS is built to meet three requirements: to evolve with new business needs, to withstand unforeseen events, and to stay easy to maintain, without complex customization. Meeting them is what bridging the practicability gap means. The paper makes four contributions:

1. A log transportation DSS whose *kernel component* generates tactical plans by solving the large-scale, multi-period *LTRSP* in practical runtimes, capturing the known constraints of the Canadian forest industry.
2. An *implicit component* that embeds trained machine learning predictors into the *LTRSP* as linear constraints, letting plans adapt to evolving planner preferences without manual re-engineering.
3. A *dynamic component* that combines local branching and reinforcement learning to recover feasible schedules after a disruption, while deviating minimally from the tactical plan.
4. An evaluation of the industrial validation on real data, measuring its financial, environmental, and operational impact and discussing managerial insights and transferability.

The individual methodological contributions were developed as the collaboration progressed and submitted to peer-reviewed journals: the *kernel component* (Abdellaoui et al. 2026b), the implicit constraint learning component (Abdellaoui et al. 2026a), and the dynamic reoptimization component (Abdellaoui et al. 2026c). The present paper combines them into an integrated, end-to-end DSS. To the best of our knowledge, this is the first work to design and validate at industrial scale a log transportation DSS that simultaneously captures the complete range of known forestry constraints, learns implicit planner constraints from historical plans, and recovers from unforeseen disruptions. The remainder of the paper is organized as follows. Section 2 describes the context and challenges of log transportation in Canada, and Section 3 presents the **FOROP-T** architecture. Sections 4, 5, and 6 detail the *kernel*, *implicit*, and *dynamic components*. Section 7 reports the DSS deployment, Section 8 presents its measured impact, and Section 9 concludes.

2 Log transportation context and challenges in Canada

Every day, harvesting crews cut and process timber into logs stored at roadside sites. From there, log trucks transport the wood to processing mills such as sawmills, pulp mills, and panel plants, where it is transformed into lumber, paper, and engineered wood products. As illustrated in Figure 1,

this chain spans five interdependent stages: wood harvesting, inventory management at supply sites, log-truck transportation, yard and wood management at receiving mills, and final processing. Log transportation sits at the center of this chain, the critical link that keeps mills supplied and harvesting crews productive. Coordinating this system goes far beyond a standard vehicle routing problem.

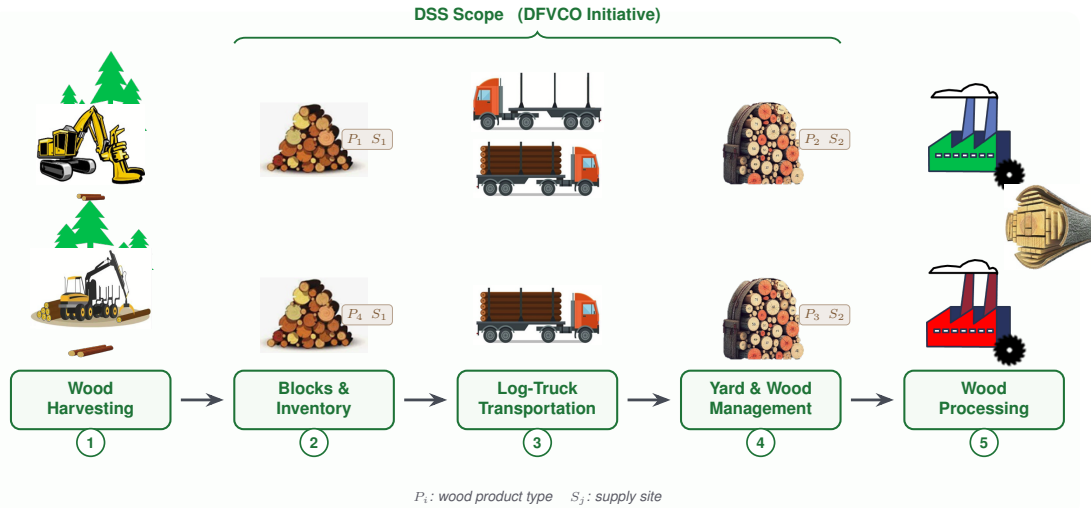


Figure 1: The Canadian forest value chain: from wood harvesting and inventory management at supply sites, through log-truck transportation, to yard management and wood processing at destination mills.

Trucks must be matched to specific products, synchronized with shared loaders, and delivered to mills on time-slotted receiving schedules (Abdellaoui et al. 2026b). Driver work-hour limits and seasonal road conditions further constrain the solution space (Audy et al. 2023). In most companies, the resulting weekly plan is built by hand under time pressure, with a planner reconciling these constraints through spreadsheets and phone calls (Abdellaoui et al. 2026b). This complexity is only the starting point. The constraints planners follow keep evolving, as new contracts, seasonal conditions, and informal practices emerge, so established plans are routinely overridden by hand (Pawlak and Krawiec 2017). Furthermore, disruptions such as road closures, breakdowns, or demand changes invalidate the plan during execution (Amrouss et al. 2016).

These observations, confirmed through our engagement with log transportation planners, point to three requirements a planning tool must meet to be adopted, rather than set aside like earlier ones. First, it must be complete, capturing the well-known operational constraints of forestry logistics. Second, it must be flexible, letting planners activate, deactivate, and extend constraints as business constraints evolve, without complex customization or analytical expertise. Third, it must be dynamic: when a disruption invalidates the plan, the tool should repair it.

We designed **FOROP-T** around these three requirements. Most existing tools chase the best solution as fast as possible. That is not what makes a tool stick in the field. **FOROP-T** instead aims to deliver near-optimal plans quickly, treating practicability as a first-class objective. It is built to be adopted: easy to use, easy to deploy, and designed as human-in-the-loop analytics that learn from each planner interaction. This is how we bridge the practicability gap: not through better analytics alone, but through a system planners actually want to use. The next section presents **FOROP-T** and its architecture.

3 FOROP-T architecture

To address these challenges, we designed an end-to-end DSS. As illustrated in Figure 2, its architecture works as follows: a set of operational inputs feeds a central database, from which optimization and analytics algorithms generate a routing and scheduling plan. In detail, the system integrates four tightly coupled components. The *kernel component* generates the tactical routing and scheduling plan by solving a large-scale optimization model. The three remaining components then adapt this plan: the *specific component* enforces the operational constraints particular to each industrial partner, the *implicit component* learns the unwritten constraints that planners apply in practice, and the *dynamic component* recovers feasible plans when disruptions occur during execution.

At the center of the platform lies a database that serves as its backbone. It connects to any standard data source, such as SQL Server, Excel, CSV files, or a company’s enterprise systems, to retrieve inputs such as fleet configurations, forest block inventories, mill demand targets, and road network parameters, and it stores the output of every algorithm for traceability and continuous learning. The database also supports real-time feeds, receiving live updates on truck positions, delivery status, and disruption events that serve as input to the *dynamic component*.

The planner interacts with the system through a web application, retaining full authority to review, edit, and approve every plan. To further lower the barrier to use, the platform embeds an *LLM-powered conversational agent* (Figure 2) that lets users populate forms and configure parameters through natural language. The next section details each component and its interaction with the others.

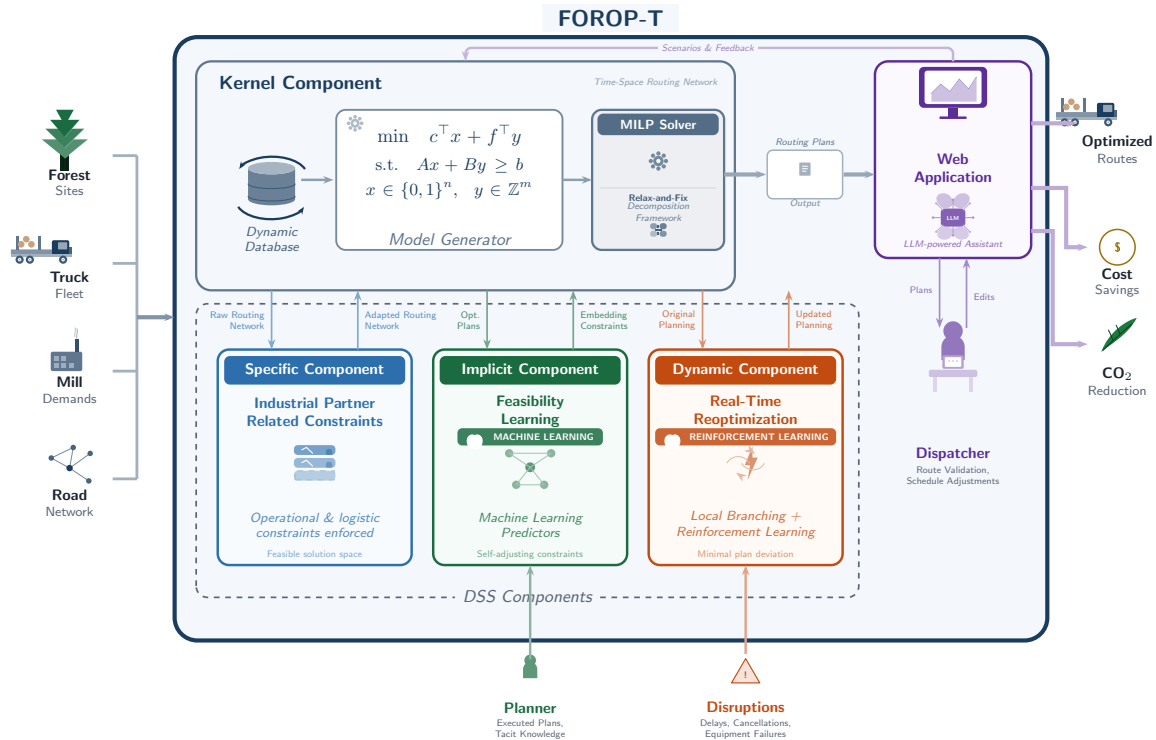


Figure 2: Overview of the proposed DSS for log transportation planning, integrating an optimization kernel, implicit constraint learning, and dynamic reoptimization within a web-based platform.

4 The kernel component: an exact model solved in practice

Each week, a planner assigns a fleet of log trucks, owned by several independent contractors, to loaded and empty movements between active harvest blocks and processing mills. A feasible plan must satisfy mill demand by species and product type, respect receiving windows, and synchronize trucks with shared loaders. It must further accommodate full and partial loads, comply with driver work-hour limits and relay handovers, observe product and client priorities, and account for a road network whose accessibility varies seasonally (Abdellaoui et al. 2026b). To our knowledge, no exact optimization model had previously integrated all of these requirements within a single formulation (Rönnqvist et al. 2015, Audy et al. 2023).

We aim for a generic and scalable DSS applicable across the Canadian forestry sector. Rather than tailoring a model to a single company, we formulate the operational constraints encountered throughout the sector and allow each company to enable only those relevant to its operations. As illustrated in Figure 3, the *kernel component* turns operational inputs into a solved plan through four stages. A queueing-driven timing module first derives the discretization interval Δ^* that sets the temporal granularity. The routing and scheduling stage then builds the time-space network $G = (N, A)$, a directed acyclic graph on which partner-specific constraints are applied by pruning. The modeling stage formulates the comprehensive mixed-integer program, which minimizes hauling and waiting cost subject to the operational constraints of forestry logistics. The solving stage finally applies a *Relax&Fix – Fix&Optimize* decomposition to produce a near-optimal plan within practical runtimes. The complete $\mathcal{LTRSP}$ formulation is given in Abdellaoui et al. (2026b).



Figure 3: Overview of the kernel component: a queueing-driven time interval determines the granularity of a time-space network, on which the optimization model is formulated and solved using a Relax & Fix – Fix & Optimize decomposition (Abdellaoui et al. 2026b).

The $\mathcal{LTRSP}$ is defined on a time-space network in which each node denotes a location at a given time interval. This representation reflects an essential property of forestry logistics: temporal and spatial decisions are inseparable. A truck reaching a harvest block before its loader becomes available incurs waiting time, whose cost is comparable to that of travel. Embedding time within the network captures this waiting explicitly, together with truck-loader synchronization, mill operating hours, and the remaining temporal constraints that govern feasibility.

The network structure poses a modeling question: how finely should time be discretized? The choice is important, since waiting at shared loaders accounts for a significant share of operational cost (Abdellaoui et al. 2026b). A short interval between successive trucks induces queueing, whereas a long interval leaves loaders underutilized. We therefore determine the interval analytically from historical loading data, using the mean loading time and its variability. The temporal discretization of the model is consequently grounded in observed data rather than fixed a priori.

Company-specific constraints are incorporated, whenever possible, without altering the mathematical formulation. Incompatibilities among vehicles, harvest blocks, and mills, along with the additional restrictions produced by the *specific component* (Figure 2), are enforced by eliminating the corresponding infeasible arcs prior to optimization. Only requirements that cannot be expressed through such preprocessing are added as explicit constraints, which preserves the generic structure of the model.

A representative weekly instance comprises millions of binary variables and constraints. When it is solved with a recent exact formulation from the literature (Melchiori et al. 2022), the optimality gap closes only slowly, even after preprocessing: as shown in Figure 4, it remains above 65% after 35 minutes and still exceeds 48% at the one-hour time limit, far beyond the response time acceptable in practice. The solver spends its first minutes in the LP root-relaxation phase, during which no integer incumbent yet exists, and progresses thereafter only in coarse steps as new incumbents are found.

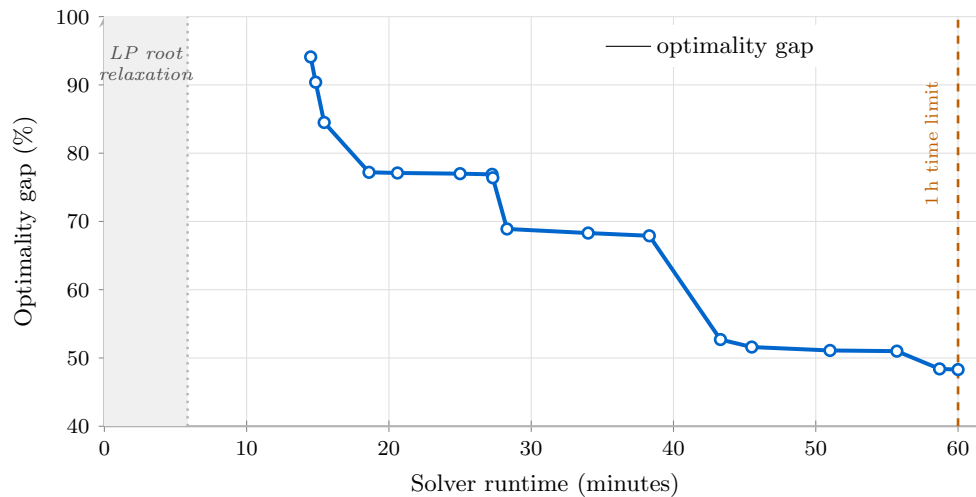


Figure 4: Evolution of the optimality gap as a function of solver runtime on a representative large instance.

In practice, planners require a high-quality plan as quickly as possible to review, adjust, and validate it before the weekly schedule is finalized. To this end, we combine a state-of-the-art CPLEX solver with a tailored *Relax&Fix – Fix&Optimize* decomposition (Abdellaoui et al. 2026b). The method exploits two properties of the problem. First, the linear relaxation is consistently tight and yields strong bounds from the outset. Second, the principal difficulty lies in obtaining an initial feasible solution, after which the incumbent improves rapidly until tailing-off begins. The decomposition fixes the decisions taken in previously optimized windows, solves the current window exactly, and relaxes future windows to accelerate the recovery of a feasible solution. It subsequently revisits selected windows to refine the incumbent.

Validated on one full year of real weekly instances, this decomposition delivers plans within 2% of optimality in under twenty minutes, which brings solving time within the operational window planners can accommodate. Detailed comparisons with baseline methods on twenty instances are given in Appendix A. Further details of the formulation and solution method are provided in Abdellaoui et al. (2026b).

5 The implicit component: learning the constraints planners apply

The *kernel component* produces near-optimal plans, but planners frequently revise them before execution. These revisions are not arbitrary. They reflect implicit operational constraints, unwritten conventions, and context-dependent preferences that no formal model has captured (Bessiere et al. 2017, Hewitt and Frejinger 2020). Such constraints are common in forestry, where operating conditions evolve continuously as seasonal road conditions change, mill specifications are updated, and informal coordination practices between drivers and loaders develop over time. Encoding these constraints requires modeling and operations research expertise that planners rarely possess. Left without an alternative, planners adjust plans by hand. Their judgment is valuable, but manual edits introduce errors, break global optimality, and accumulate inconsistencies over the planning week. This recurring cycle of override and correction gradually erodes confidence in the tool and undermines its adoption. We reverse this dynamic: rather than treating overrides as noise, we treat them as the most informative data the system receives, closing a human-in-the-loop cycle in which each manual edit becomes a learning signal.

We address this challenge through **CLIF**, a machine-learning-based framework for learning implicit constraints, outlined in Figure 5 and detailed in Abdellaoui et al. (2026a). The framework builds on a simple observation: systematic discrepancies between an optimal plan X^* and the executed plan $\tilde{X}$ reveal implicit feasibility constraints absent from the original model. We encode these discrepancies through a feasibility function $\tilde{f} : \mathcal{A} \rightarrow \{0, 1\}$ over the set of network arcs $\mathcal{A}$, where $\tilde{f}(a) = 1$ when arc a appears in executed plans and 0 otherwise, and learn an approximation $\hat{f}_\theta \approx \tilde{f}$ from historical pairs $\{(X_t^*, \tilde{X}_t)\}$ of optimal and executed plans.

Implicit constraints in forestry take several forms within the routing and scheduling network. Some are linear, expressible as weighted combinations of arc variables. Others are relational, encoding interactions among vehicles, locations, and time slots. Others are conditional, of an *if-then* type that activates only in specific contexts. Because the form of a hidden rule is unknown in advance, we do not commit to a single machine learning predictor. Linear regression recovers linear separability in arc usage, decision trees capture threshold-based and conditional structures (Bayani et al. 2024), and a graph neural network exploits the relational geometry of the routing graph by propagating information across neighboring arcs (Scarselli et al. 2009, Cappart et al. 2023).

We combine the three predictors through a confidence-based stacking procedure. The rationale follows the no-free-lunch principle: no single learner dominates across all rule types, and the nature of the active rule is unknown at deployment. Consequently, stacking assigns each arc-level prediction to the most confident learner, the one predicting closest to 0 or 1. The learned function is then embedded exactly into the routing and scheduling model $\mathcal{LTRSP}$ as a set of linear constraints (Grimstad and Andersson 2019), yielding the augmented model $\mathcal{LTRSP}_{\text{AUG}}$, which jointly enforces classical routing and scheduling feasibility and the recovered implicit feasibility:

$$\begin{aligned}
 \min_{x, \hat{x}, \Delta} \quad & \sum_{a \in \mathcal{A}} c_a x_a + \lambda \sum_{a \in \mathcal{A}} \Delta_a \\
 \text{s.t.} \quad & A^\top X \geq b, \\
 & \hat{x}_a = \hat{f}_\theta(\mathcal{G}, X)_a, \\
 & \Delta_a \geq |x_a - \hat{x}_a|,
 \end{aligned} \tag{1}$$

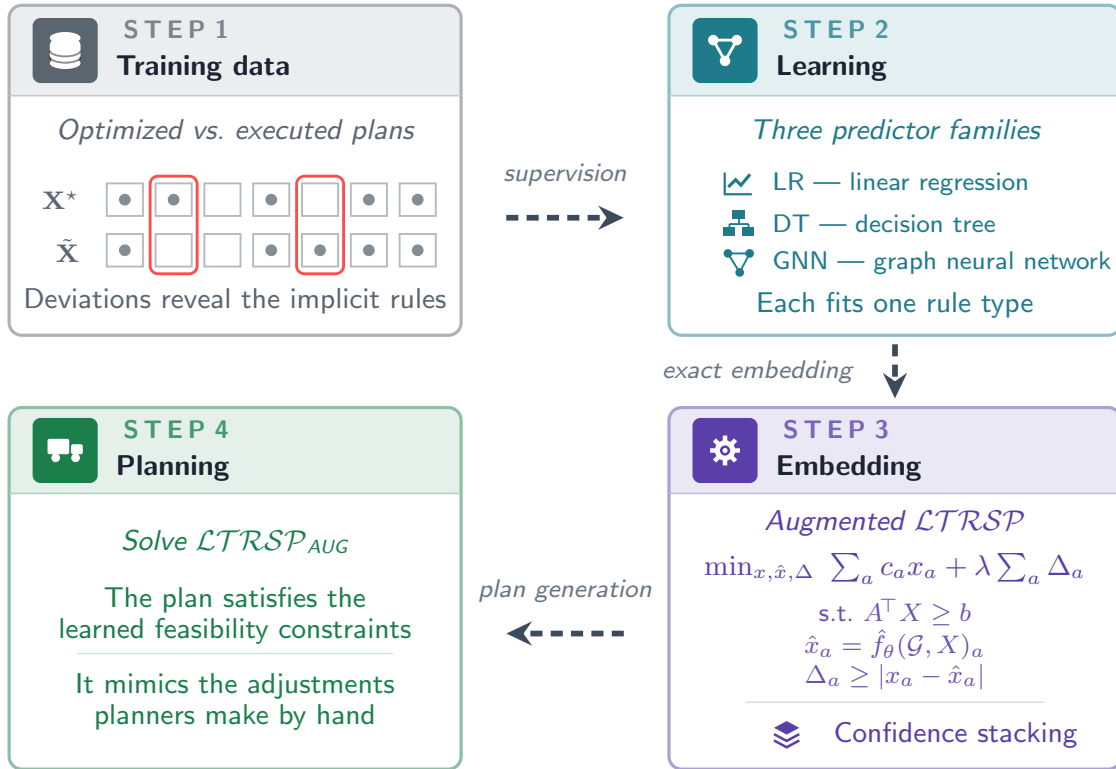


Figure 5: Overview of the implicit component: deviations between optimal and executed plans are learned by predictors and embedded into the $\mathcal{LTRSP}$ as linear constraints (Abdellaoui et al. 2026a).

where x_a indicates whether arc a is used and c_a is its cost, $A^\top X \geq b$ gathers the original routing and scheduling constraints, and $\hat{x}_a = \hat{f}_\theta(\mathcal{G}, X)_a$ is the prediction of the learned function on the routing graph $\mathcal{G}$ that is then linearized. The linearization depends on the predictor and is described in Abdellaoui et al. (2026a). Here Δ_a measures the deviation of x_a from that prediction, and λ balances routing and scheduling cost against alignment with the learned feasibility region.

Within **FOROP-T**, **CLIF** operates on the partner’s weekly instances, capturing the discrepancies between the optimal and the adjusted plan week after week. It follows the operational rhythm of the partner: the planner produces the tactical plan each Friday, and **CLIF** is trained the day before, so its learned constraints are already embedded when the next plan is generated. The weekly plans used for learning are the optimal plans with the adjustments the planner applies before execution. Changes occurring during execution are stochastic events rather than implicit constraints, and are handled by the *dynamic component*. The implicit constraints planners apply are numerous. Representative examples include restrictions preventing certain vehicles from serving a given block, homebase policies forbidding dispatch to specific blocks, a mutual-exclusion rule barring a vehicle group from serving two block groups at once, mill service cut-off times, and a morning co-service restriction between two mills.

CLIF aligns the optimal solution of $\mathcal{LTRSP}_{AUG}$ with these constraints, according to what it has learned. We trained it on past weeks in which planners applied the five constraints above, and tested it on twenty new weekly instances by solving $\mathcal{LTRSP}_{AUG}$ with the learned constraints embedded. Figure 6 reports these results in detail. Across all five constraints, every predictor reaches full satisfaction. Confidence-based stacking reaches it first, and at the lowest cost, under two percent of routing and scheduling cost, a modest price for plans planners accept without revision. It is therefore the

predictor retained within **FOROP-T**: by delegating each prediction to its most reliable learner, it accommodates a wider range of constraint types than any single model. Detailed comparisons with baseline methods are given in Appendix B.

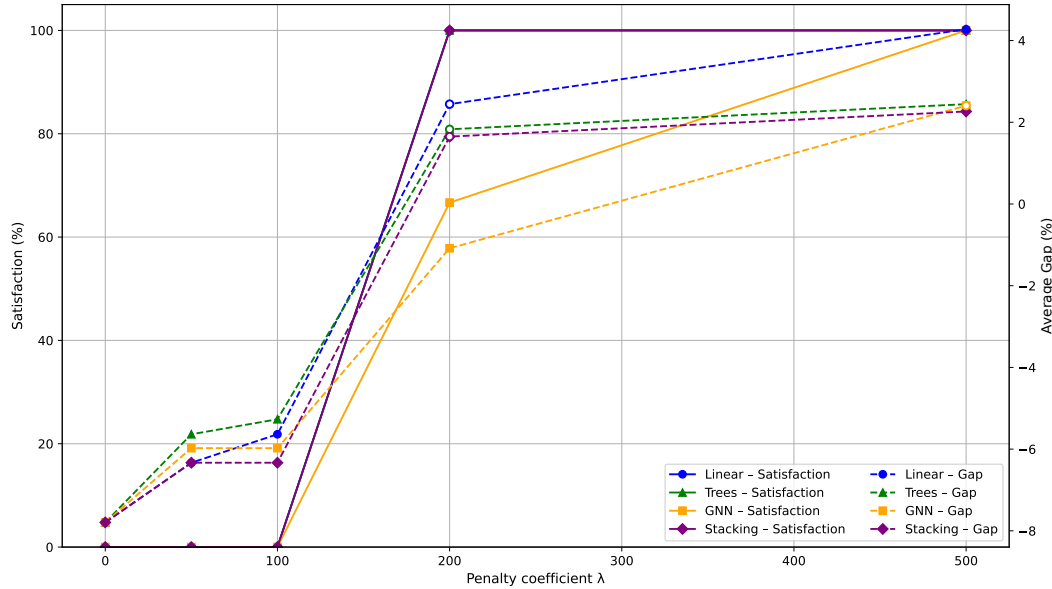


Figure 6: Satisfaction rate and objective gap versus the penalty coefficient λ for the combined rule set, evaluated on twenty weekly plans. The satisfaction rate is the fraction of plans that honor every implicit constraint, and the objective gap is the additional routing and scheduling cost incurred once the learned constraints are fully satisfied. Stacking reaches full satisfaction at a lower λ than any individual learner while maintaining the smallest gap throughout.

The *implicit component* sustains the long-term robustness of the DSS. By recovering the constraints that planners apply in daily operations, $\mathcal{LTRSP}_{AUG}$ produces plans that remain operationally credible and adapt to changing conditions without any manual reformulation of the model. This adaptability emerged as the feature most valued by Natural Resources Canada and its partner companies. A common cause of abandonment in the sector is reliance on rigid models that fail to track how operations evolve; by observing, learning, and respecting the constraints planners themselves apply, this component builds the trust that sustained adoption requires. Further details are given in Abdellaoui et al. (2026a).

6 The dynamic component: recovering from disruptions

The *kernel* and *implicit components* govern, respectively, the quality and adaptability of a planned schedule, but neither addresses what happens once execution begins. Canadian forestry operations face frequent and often severe disruptions: road closures from weather or maintenance, truck breakdowns, travel delays on degraded roads, and sudden demand fluctuations (Amrouss et al. 2016). Such events render parts of the weekly plan infeasible within hours of its release, and slow recovery translates directly into financial loss and unmet demand. When a disruption occurs, planners resolve it manually through phone calls and ad hoc rerouting and rescheduling. Manual recovery is demanding, sacrifices global optimality, and rarely preserves the plan structure around which drivers, loaders, and mill operators have already organized their days.

To close this gap, we developed a dynamic reoptimization framework that receives a disruption, characterizes its impact on the routing and scheduling network, and produces an updated schedule as quickly as possible while minimizing deviation from the original tactical plan. As illustrated in Figure 7 and detailed in Abdellaoui et al. (2026c), the framework combines the following elements: a disruption

characterization module that translates operational events into model modifications, a local branching formulation that controls the trade-off between cost and plan stability, and a reinforcement learning agent that adaptively tunes the search neighborhood from disruption severity and solver feedback. Rather than returning a single repaired plan, this component produces a menu of recovery options, a Pareto front spanning the cost–stability trade-off. This design responds to a requirement expressed by the planners, who asked for several alternatives to select from according to the operational context.

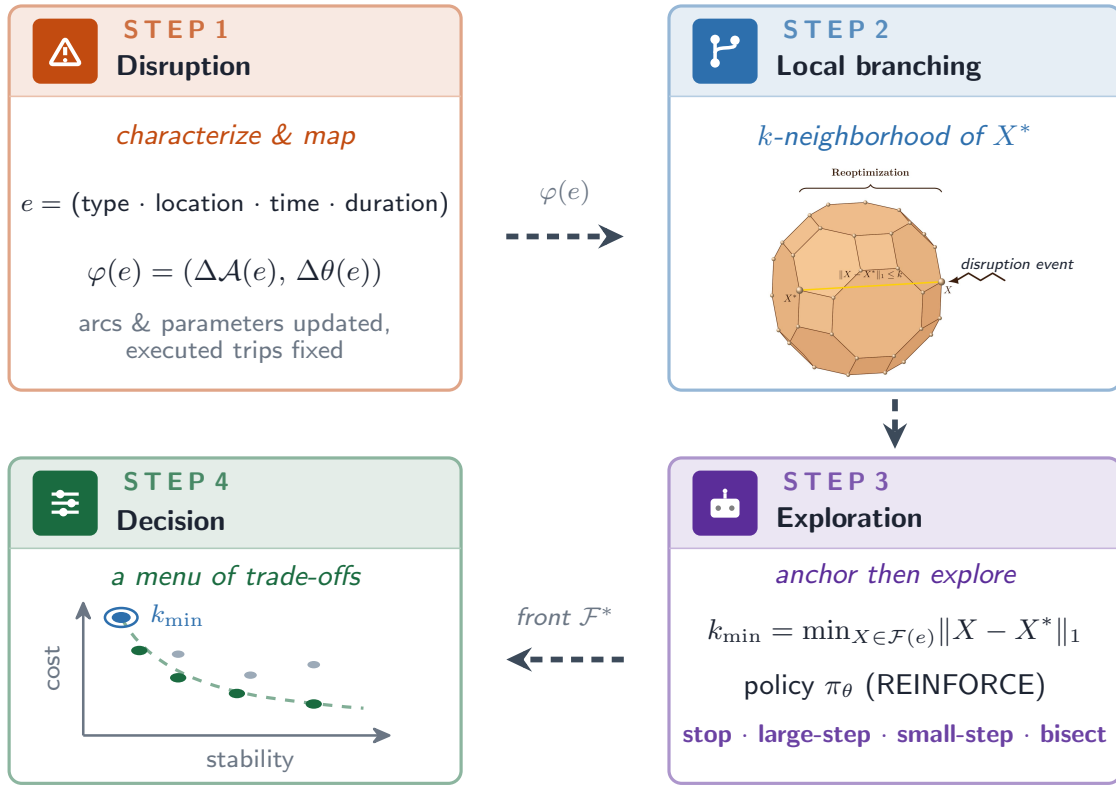


Figure 7: The dynamic component: disruptions are mapped, reoptimized by local branching with an RL-driven radius k , and presented as a stability–cost Pareto front (Abdellaoui et al. 2026c).

When a disruption event e occurs at time τ_e during execution, the *Disruption Characterization* translates it into a pair of model modifications: $\varphi(e) = (\Delta\mathcal{A}(e), \Delta\theta(e))$ where $\Delta\mathcal{A}(e)$ encodes changes to the routing and scheduling network, such as arc removals for blocked roads or disabled vehicles and arc additions for delayed departures, and $\Delta\theta(e)$ captures parameter changes such as revised demand, adjusted supply, or altered vehicle availability. Table 1 summarizes the four disruption categories the framework models.

All routing and scheduling decisions for trips completed or in progress at time τ_e are fixed to their executed values, keeping the recovered plan consistent with the physical state of the fleet. As mentioned before, reoptimization in forestry must balance two competing objectives: limiting the cost impact of the disruption while preserving as much of the original plan as possible.

We formalize this trade-off through a local branching constraint (Fischetti and Lodi 2003) that bounds the Hamming distance between the recovered plan X and the original tactical solution X^* , where x_{atv} equals 1 if arc a is used by vehicle v on day t :

$$\min_{X \in \mathcal{F}(e)} z(X) \quad \text{s.t.} \quad \sum_{(a,t,v)} |x_{atv} - x_{atv}^*| \leq k, \quad (2)$$

Table 1: Characterization of disruption events: required information and impact on the $\mathcal{LTRSP}$ model.

Event type	Description	Required information	Model impact
Road or site closure	Temporary shutdown of a forest road or site.	Time, location, duration.	$\Delta\mathcal{A}(e)$: movements crossing the closed segment are removed for this duration.
Travel delay	Slowdown caused by weather or road degradation.	Time, expected delay, location, truck concerned.	$\Delta\theta(e)$: travel time is increased and the shift propagates to later trips.
Truck or loader breakdown	Equipment out of service during a repair.	Time, equipment concerned, repair time, current location.	$\Delta\mathcal{A}(e)$: movements of the affected truck are removed until the repair is completed.
Demand or supply variation	Change in mill demand or forest supply.	Time, mill or site concerned, product, size of the change.	$\Delta\theta(e)$: the demand or supply requirement is revised accordingly.

where $z(X)$ denotes the $\mathcal{LTRSP}$ routing and scheduling objective evaluated on the plan X , and $\mathcal{F}(e)$ is the feasible region after applying $\varphi(e)$ and fixing the decisions taken before τ_e . The parameter k sets the size of the search neighborhood. A small k keeps the recovered plan close to the original, but limits cost improvement, whereas a large k allows broader exploration at the price of stability. Varying k therefore traces the menu of cost–stability trade-offs we seek. Exploration starts by computing $k_{\min}$, the smallest k for which problem (2) remains feasible, and then increases k from there. Building this menu by solving the local branching model at every value of k is impractical, since each solve is costly and most values yield no improvement. We therefore treat reoptimization as a Markov decision process and learn a policy that navigates k -space. The approach extends the RL-guided local branching of Liu et al. (2022) from single-solution improvement to the discovery of a Pareto front, with a redesigned state, action set, and reward. The policy selects promising values beyond $k_{\min}$, skips unproductive regions, and stops when returns diminish. The agent learns two things at once: where to look and when to stop. It tries new values of k beyond those already tested. Discovering a new non-dominated plan is rewarded, spending a solve for nothing is penalized, and the agent stops on its own once further solves are unlikely to pay off, which is what makes the recovery fast enough to be usable in the field. The policy is trained by reinforcement learning (Williams 1992).

Training follows the operational rhythm of our partners. The planner releases the tactical plan on Friday; over the weekend, the RL policy is trained on that instance using disruption scenarios designed with the partner to reproduce the events observed in practice, so the policy is ready when perturbations arise during the week. Training takes about 28 hours on average, well within the available window. Tested on disruptions spanning all categories across the weekly instances, the trained policy returns on average five recovery plans within a fifteen-minute budget. Detailed comparisons with baseline methods are given in Appendix C. From a practical standpoint, this component relieves planners of a significant burden. Recovering from a disruption by hand could take more than thirty minutes and yield a single feasible plan, with no assurance that its cost was reasonable, a balance even experienced planners struggle to strike.

This *dynamic component* keeps the human in the loop. First, the planner supplies the characteristics of the disruption, its time, location, and expected duration, which the platform can also retrieve automatically from the real-time feed. Second, the analytics return a set of Pareto-optimal options. The planner, who knows the drivers, the roads, and the realities of the day, then makes the final decision according to the operational context. Further details are provided in Abdellaoui et al. (2026c).

7 FOROP-T deployment

The complete DSS, integrating all components, is deployed as a web-based platform accessible to operational planners without any requirement for programming or optimization expertise. The system is currently deployed at Natural Resources Canada and is being tested by a major forest company in New Brunswick. Test trials and discussions are underway with an additional forest company in Quebec, extending the reach of the system to different operational contexts, fleet structures, and forest network configurations.

As illustrated in Figure 8, the web application serves as the human-machine interface through which planners engage with the full planning process. From a single platform, users trigger the optimization algorithms, visualize the generated plans on an interactive map, modify individual route assignments from field knowledge or last-minute changes, and approve the final schedule for execution. This workflow replaces the previous practice of assembling plans by hand each morning, compressing a process that once took most of a morning into a short review-and-approve step. The system is therefore not a black box that replaces human judgment: planners retain full authority over every routing decision, and their edits are checked for feasibility, then fed back into the learning loop, keeping the *implicit component* current with evolving preferences. An embedded *conversational agent* powered by a large language model assists planners in configuring disruption events, populating solver parameters, and navigating the system through natural language.

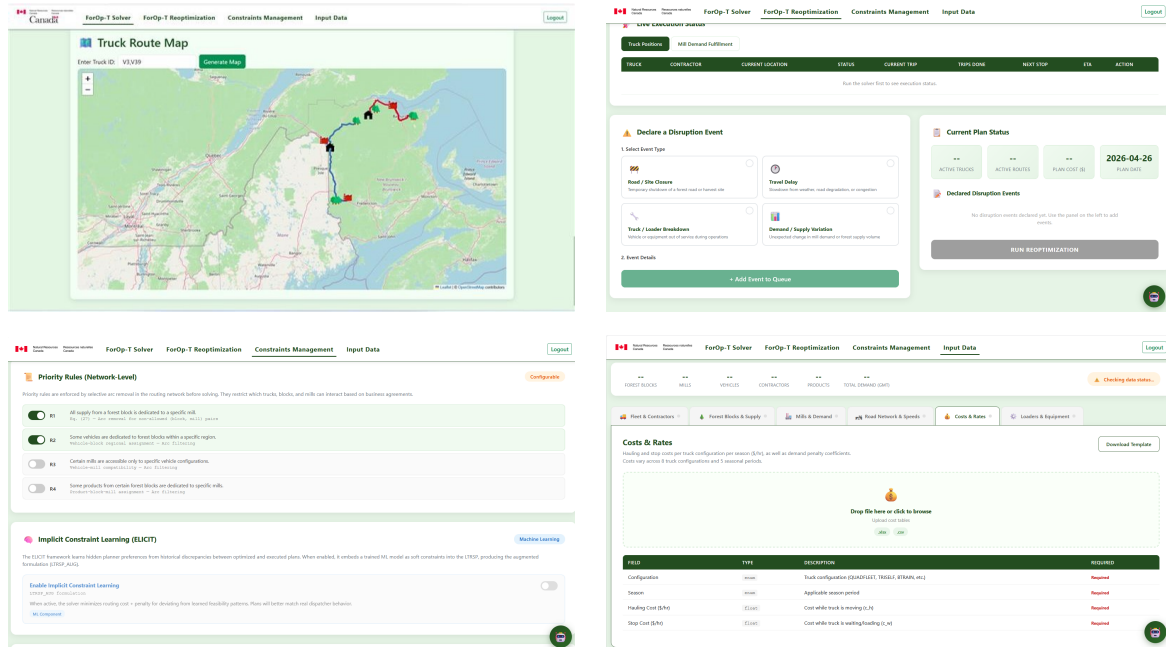


Figure 8: Screenshots of the FOROP-T decision support system interface.

FOROP-T components do not operate in isolation; they follow the weekly rhythm of the partner, as shown in Figure 9. On Thursday night, the *implicit component* is retrained on the plans executed during the previous weeks, so that the constraints it applies reflect current practice. On Friday, the planner runs the kernel with those constraints embedded, reviews the proposed plan, and releases it for the coming week. Over the weekend, the recovery policy of the *dynamic component* is trained on that plan, and is ready before execution begins. From Monday onward, each disruption triggers reoptimization and returns recovery options within minutes.

Since early 2026, the DSS has been deployed within NRCan's IT infrastructure and requires a single CPLEX license. It can equally be deployed within a forest company's own infrastructure.

NRCan provides planner training after deployment. A development is underway that connects the DSS to internal enterprise systems, in particular invoicing and accounting, so that contractor billing is generated automatically from executed routes, removing a significant source of administrative work after each planning cycle.



Figure 9: The weekly planning cycle: preparation from Thursday night to the weekend, then execution from Monday, with the dynamic component active throughout.

8 Impact and managerial insights

As mentioned, the framework was first deployed within NRCan's IT infrastructure, where planners can move from building plans by hand each morning to validating system-generated ones. Producing a weekly plan once took several hours. It now takes under one hour, including the planner's adjustments and analysis. Because the plan already incorporates the implicit constraints learned from past practice, this review is lighter, and the few edits planners still make feed the next learning cycle, so the system keeps closing the gap between the proposed and the executed plan. During execution, the DSS returns a menu of recovery plans within fifteen minutes, a task that planners previously found difficult and time-consuming. By absorbing these demanding tasks, the system frees planners for analysis, follow-up, and the decisions that require their judgment.

To quantify the financial gain, we compared **FOROP-T** against the New Brunswick company's historical manual planning over one full year of weekly instances. For each week, we ran the DSS on the same input data the planner had used, that is, the same harvest blocks, mill demands, fleet availability, and road conditions, and measured the total distance traveled by the resulting plan against the distance of the manual plan actually executed that week. Averaged over the year, this controlled comparison on identical instances showed a 35% reduction in the total distance traveled, which translates into \$8 million in annual savings and avoids 4,800 tonnes of CO₂ emissions per year. Scaled nationally on the basis of the total volume harvested in Canada, adoption across the sector is estimated to yield up to \$377 million in annual fuel savings and to avoid more than 500,000 tonnes of CO₂ each year, directly supporting Canada's net-zero objectives.

The structure of these savings is operationally informative. Decomposing the \$8 million saving between the two terms of the objective attributes about 85% to hauling and 15% to waiting. The waiting cost is dominated by truck loading and unloading times, which depend on the number of loaded trips required to meet demand. Since that number is essentially fixed for a given weekly demand, waiting leaves little room for improvement under any planning method. The hauling cost, by contrast, is precisely where centralization and backhauling act, by minimizing traveled distance, which is consistent with the 35% distance reduction reported above.

The analysis of these results, together with discussions with planners and forest-company management, points to three managerial insights. First, the hauling-dominated structure of the savings makes them robust. Hauling distances follow the planning decisions directly, whereas waiting times are exposed to operational stochasticity such as loader availability, synchronization delays, and queueing. A large share of the estimated saving is thus structurally achievable, independently of day-to-day variability. Second, fleet reduction is the dominant environmental lever: a smaller active fleet lowers fixed

operating costs and emissions in proportion, positioning the tool as an instrument of sustainability and not only of cost minimization. Third, centralized planning requires aligned incentives. Contractors may perceive fleet reduction as a loss of utilization and resist the transition. Since centralization generates savings at the company level, part of these can be redistributed to contractors through an increased hauling rate, in a proportion specific to each company’s contracts and realized savings.

Propagating this impact beyond the initial deployment, NRCan presented the DSS in April 2026 at the annual assembly of the Quebec Forest Producers Federation (FPFQ), which represents over 162,900 private forest owners across Quebec (FPFQ 2026). In September 2026, NRCan also presented the project to the general public in a French-language interview on Radio-Canada, the national public broadcaster (Radio-Canada 2026).

9 Conclusion

This paper reported on a partnership between Polytechnique Montréal, and Natural Resources Canada to bring optimization and analytics into the daily practice of log transportation planning. The obstacle was not the mathematics but its practicability: earlier tools had been dropped because they did not fit how operations evolve and how disruptions unfold. **FOROP-T** was designed to meet planners on that ground.

The DSS components work together toward this end: an exact and comprehensive formulation of the routing and scheduling problem, the learning of implicit constraints from executed plans, and fast recovery under disruption. Delivered through a platform that keeps the planner in control, **FOROP-T** has been adopted by NRCan and deployed within its IT infrastructure: it has been tested and validated by a forest company in New Brunswick, with trials underway at a second company in Quebec. It has also been presented by NRCan to the Quebec Forest Producers Federation. Its first test and industrial validation already point to considerable financial and environmental gains, and these results suggest that the practicability gap can be bridged by integrating optimization and analytics around the planners’ practical needs, which can earn their trust and support adoption.

Several questions remain open. The national projection awaits confirmation as the ongoing trials mature, and the framework’s transfer to other forest companies, with different fleets and contractual arrangements, will test its generality in practice. System connectivity is another direction: automating the data feeds, and streaming real-time truck positions and disruption events, would let it react without manual input. More broadly, the same three difficulties, incomplete models, constraints that evolve, and disruptions in execution, arise in many planning domains beyond forestry, where a similar human-in-the-loop design may prove equally valuable.

A Kernel component: Comparison with baseline methods

This appendix compares the *Relax&Fix – Fix&Optimize* (R&F–F&O) decomposition used by the *kernel component* with direct solution of the model by a commercial solver and with a recent exact formulation in the literature. All results reported here are taken from Abdellaoui et al. (2026b), where the experimental study is described in full.

A.1 Experimental setting

The experiments use 20 weekly instances built from historical data of the industrial partner. Each instance involves between 11 and 18 mills, 19 to 46 harvest blocks, 36 to 88 trucks operating from 17 to 30 home bases, and three or four product types, with a weekly demand between 3,820 and 23,500 green metric tonnes. The instances are grouped by the number of arcs in the time-space network, which drives the computational effort: five small (fewer than 30,000 arcs), nine medium (30,000 to

50,000), and six large (more than 50,000, up to 86,959). An anonymized subset of these instances is publicly available at <https://github.com/abdelabde/LTRSP-Benchmark>.

Three baselines are considered. The first two solve the complete model directly with CPLEX 22.1.1, with default and with tuned parameters, until an optimality gap of 2% is proven. The third is the formulation of Melchiori et al. (2022), a recent exact model for log-truck routing and scheduling. We reimplemented it, verified that it reproduces the objective values and model sizes reported by its authors, and extended it to a multi-period horizon, loader capacities, and penalized unmet demand, so that it applies to the same instances (Abdellaoui et al. 2026b). It is solved by CPLEX with a one-hour limit. Solution quality is measured by the gap to the best known solution (Gap BK) and by the gap to the best lower bound proven by CPLEX (Gap LB). All runs were performed on an Intel Core i7-8700 processor at 3.20 GHz with 64 GB of memory.

A.2 Results

Table 2 reports the results by instance category. Over the 20 instances, R&F-F&O reduces the average runtime from 27.3 minutes (default CPLEX) and 20.5 minutes (tuned CPLEX) to 11.1 minutes, with a lower average gap to the best known solution (0.19% against 0.56%). The reduction is largest on large instances, where the average runtime decreases from 40.4 minutes with tuned CPLEX to 17.2 minutes, and the longest run over all instances decreases from 78.6 to 19.0 minutes. With respect to the proven lower bound, R&F-F&O is within 1.3% of optimality on average and within 2.3% on every instance. This behavior is consistent with the structure of the model, whose linear relaxation is nearly integral, whereas finding a first feasible plan accounts for about a quarter of the direct solver runtime. R&F-F&O reduces this effort by relaxing the integrality of future time windows.

Table 2: Average runtime and solution quality of R&F-F&O and the baseline methods, by instance category. Gap BK is the gap to the best known solution, and Gap LB the gap to the best lower bound proven by CPLEX. The formulation of Melchiori et al. (2022) is solved with a one-hour limit.

Category	Inst.	CPLEX (default)		CPLEX (tuned)		R&F-F&O			Melchiori et al. (2022)
		Time (min)	Gap BK (%)	Time (min)	Gap BK (%)	Time (min)	Gap BK (%)	Gap LB (%)	Gap BK (%)
Small	5	12.75	0.54	7.38	0.59	5.03	0.45	1.34	35.8
Medium	9	23.02	0.53	14.63	0.57	10.52	0.09	1.27	41.2
Large	6	45.87	0.61	40.38	0.56	17.16	0.14	1.23	44.3
All	20	27.31	0.56	20.54	0.57	11.14	0.19	1.28	40.78

The formulation of Melchiori et al. (2022) returns, after one hour, plans whose cost exceeds the best known solution by 35% to 45%.

The time discretization of the network is another contribution of the *kernel component*. Rather than being fixed a priori, the discretization interval is derived from a queueing analysis of loader operations based on the observed loading times, which yields plans that remain robust to variations in service times during execution. The derivation and its sensitivity analysis are detailed in Abdellaoui et al. (2026b).

B Implicit Component: Comparison with baseline methods

This appendix compares **CLIF**, the learning framework of the *implicit component*, with the two approaches from the literature on which it builds. All results reported here are taken from Abdellaoui et al. (2026a), where the experimental study is described in full.

B.1 Experimental setting

The data cover 48 months of weekly routing and scheduling decisions of the industrial partner, on networks drawn from more than 900 forest blocks and 50 mills and served by a heterogeneous fleet of about 110 trucks. From this history, 190 weekly instances were extracted and split chronologically into 150 training instances, 20 validation instances, and 20 test instances, so that no information from the test period is used during training. For each instance, the data consist of a pair formed by the optimal plan of the routing and scheduling model and the corresponding executed plan.

Five rules were considered, corresponding to the constraints planners report applying most frequently: vehicles excluded from a given block (R_1), a homebase that may not dispatch to certain blocks (R_2), a vehicle group that may not serve two block groups on the same day (R_3), a cut-off time for serving certain mills (R_4), and a restriction allowing only one of two mill groups to be served in the morning (R_5).

Four variants of the augmented model are compared. The two baselines embed a single predictor: the linear regression of Hewitt and Frejinger (2020) and the decision tree of Bayani et al. (2024). The two variants proposed in Abdellaoui et al. (2026a) embed, respectively, a graph neural network and the confidence-based stacking of the three predictors, the latter being the variant retained in **FOROP-T**. Each variant is evaluated with two measures on the test instances. The satisfaction rate is the share of test instances whose plan satisfies all active rules. The objective gap is the relative increase in cost of that plan with respect to the optimal plan of the model in which the rules are written explicitly as constraints, that is, the cost of learning the rules instead of knowing them. Six experiments were conducted, one for each rule and one with the five rules active simultaneously, with the penalty coefficient λ varied over a predefined range.

B.2 Results

For a sufficiently large λ , every variant reaches a satisfaction rate of 100% on every rule: each predictor identifies, on unseen weeks, the decisions affected by the rules. The variants therefore differ through the objective gap, reported in Table 3. Confidence-based stacking attains the lowest gap on every rule and in the combined experiment, tied with the graph neural network on rules R_1 and R_5 . The difference is largest on rule R_3 , where the stacked model reaches 1.33%, against 3.39% for linear regression and 4.11% for the decision tree. With the five rules active simultaneously, the gap of the stacked model is 1.64%, against 2.44% and 1.82% for the two baselines. A single value $\lambda \approx 200$ achieves full satisfaction on all five rules with a gap below 1.7%, so that the coefficient does not need to be recalibrated from week to week.

Table 3: Objective gap (%) of the augmented model on the test instances, by rule and by embedded predictor. All variants reach a satisfaction rate of 100% on every rule. The stacked model attains the lowest gap in all six experiments, tied with the graph neural network on R_1 and R_5 .

Rule	Baselines from the literature		Abdellaoui et al. (2026a)	
	Linear regression (Hewitt and Frejinger 2020)	Decision tree (Bayani et al. 2024)	Graph neural network	Stacking
R_1	0.63	0.68	0.08	0.08
R_2	0.61	0.54	0.54	0.15
R_3	3.39	4.11	3.39	1.33
R_4	0.41	0.41	0.08	0.07
R_5	0.43	0.77	0.04	0.04
R_1 to R_5 combined	2.44	1.82	2.34	1.64

Analysis of the stacked model shows that no single predictor dominates: in the combined experiment, none is selected on more than 45% of the active arcs. Since the form of the rules applied by planners is not known in advance, committing to a single predictor family, as in Hewitt and Frejinger

(2020) and Bayani et al. (2024), risks a mismatch between the predictor and the structure of the rules. Stacking avoids this choice by selecting, for each decision, the most confident predictor within the optimization itself.

C Dynamic component: comparison with baseline methods

This appendix compares the reinforcement learning policy of the *dynamic component* with two baseline strategies for exploring the deviation–cost trade-off after a disruption. All results reported here are taken from Abdellaoui et al. (2026c), where the experimental study is described in full.

C.1 Experimental setting

The experiments use the 20 weekly instances of Appendix A, with the same grouping into five small, nine medium, and six large instances. For each instance, the tactical plan is first computed without disruption. Disruption scenarios are then produced by a generator that draws ten events over the five-day horizon, about two per day, among four types: road or site closures, truck or loader breakdowns, travel delays, and demand or supply variations. The policy of each instance is trained for 50 epochs and evaluated on one additional scenario generated with a different seed. The evaluation scenario is the same for all strategies, which gives 200 recovery problems (20 instances and 10 events) solved by each strategy.

The three strategies share the same local branching formulation and the same starting point. After a disruption, the smallest deviation $k_{\min}$ that restores feasibility is computed exactly, and the corresponding plan, the most stable recovery option, is the first point of the front. The strategies also share the same computational components, namely the memory of values already evaluated, the elimination of intervals where no improvement is possible, and the warm start of each solve. They differ only in the rule that selects the next value of the neighborhood size k . The first baseline is a fixed grid of ten equally spaced values between $k_{\min}$ and the upper limit $k_{\max}$, solved in increasing order. The second is a dichotomic search, which repeatedly splits the interval with the largest cost reduction between two known plans. The proposed strategy selects k with a policy trained by reinforcement learning from solver feedback and features of the disruption.

Three measures are reported. The hit rate is the share of solver calls that produce a new non-dominated plan, that is, the share of the computational effort converted into an option for the planner. The cost deviation is the relative difference between the cost of the cheapest recovered plan and the cost of the plan before disruption. The hypervolume is the area of the deviation–cost plane dominated by the recovered front, normalized by a reference region that depends only on the event, so that its value lies between 0 and 1. It increases with both the number of plans and their spread, and therefore distinguishes a front of well-separated options from a front of similar plans.

C.2 Results

Table 4 reports the results by instance category. The learned policy attains the highest hit rate on each of the 20 instances, with an average of 70.6%, against 48.2% for the dichotomic search and 33.1% for the fixed grid. Since the three strategies operate under the same budget and share the same components, this difference is due to the selection rule alone. The fixed grid, which explores the interval from $k_{\min}$ upward, often exhausts its budget before reaching the largest neighborhoods, where the cheapest plans lie, and ends 8.05% above the pre-disruption cost.

The learned policy also attains the highest average hypervolume, 0.310, against 0.264 for the dichotomic search and 0.170 for the fixed grid. The hit rate and the hypervolume therefore measure different properties and are reported together. The advantage of the policy does not decrease with network size, and its hit rate is highest on the large instances.

Table 4: Average hit rate (η), cost deviation of the cheapest recovered plan (Δz), and normalized hypervolume (HV) of the three exploration strategies, by instance category, over ten disruption events per instance and with a budget of 15 minutes per event.

Category	Inst.	Fixed grid			Dichotomic search			Learned policy		
		η (%)	Δz (%)	HV	η (%)	Δz (%)	HV	η (%)	Δz (%)	HV
Small	5	31.7	6.90	0.164	47.7	5.89	0.262	68.1	5.91	0.294
Medium	9	33.1	8.30	0.180	49.2	6.72	0.277	67.7	6.73	0.324
Large	6	34.2	8.63	0.160	47.1	6.14	0.245	77.1	6.18	0.302
All	20	33.1	8.05	0.170	48.2	6.34	0.264	70.6	6.36	0.310

References

- Abdellaoui, A., Boufous, A., El Hallaoui, I., Benabbou, L., Aubé, F. & Amazouz, M. Learning implicit feasibility constraints for real-world routing and scheduling: Application to log transportation. arXiv preprint arXiv:2606.05353 (2026).
- Abdellaoui, A., El Hallaoui, I., Benabbou, L., Aubé, F. & Amazouz, M. Real-world, large scale, multi-period log truck routing and scheduling: Application to Canadian forestry. arXiv preprint arXiv:2607.16978 (2026).
- Abdellaoui, A., El Hallaoui, I., Benabbou, L., Aubé, F. & Amazouz, M. Learning to recover: Adaptive local branching with reinforcement learning for log-truck routing and scheduling under disruptions. arXiv preprint arXiv:2607.20901 (2026).
- Amrouss, A., Gendreau, M., & Gendron, B. (2016). Managing unforeseen events in forest transportation. *Journal of Science and Technology for Forest Products and Processes*, 6(1), 35–44.
- Andersson, G., Flisberg, P., Lidén, B., & Rönnqvist, M. RuttOpt—a decision support system for routing of logging trucks. *Canadian Journal of Forest Research*, 38, 1784–1796 (2008).
- Audy, J., Rönnqvist, M., D’Amours, S. & Yahiaoui, A. Planning methods and decision support systems in vehicle routing problems for timber transportation: a review. *International Journal of Forest Engineering*, 34, 143–167 (2023).
- Bayani, M., Adulyasak, Y., & Rousseau, L.M. (2024). Learning and modeling implicit constraints in optimization models through decision trees. *Les Cahiers du GERAD*, 711, 2440.
- Bektas, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8), 1232–1250.
- Bessiere C, Koriche F, Lazaar N, O’Sullivan B (2017) Constraint acquisition. *Artificial Intelligence* 244:315–342.
- Cappart Q, Chételat D, Khalil EB, Lodi A, Morris C, Veličković P (2023) Combinatorial optimization and reasoning with graph neural networks. *Journal of Machine Learning Research* 24(130):1–61.
- El Hachemi, N., Gendreau, M. & Rousseau, L.-M. A heuristic to solve the synchronized log-truck scheduling problem. *Computers & Operations Research*, 40(3), 666–673 (2013).
- Fischetti, M. & Lodi, A. Local branching. *Mathematical Programming*, 98(1), 23–47 (2003).
- Fédération des producteurs forestiers du Québec (FPFQ). About FPFQ. <https://www.foretprivee.ca/a-propos/>. Accessed: 2026.
- Government of Canada. Net-Zero Emissions by 2050. <https://www.canada.ca/en/services/environment/weather/climatechange/climate-plan/net-zero-emissions-2050.html>, 2024.
- Grimstad B, Andersson H (2019) ReLU networks as surrogate models in mixed-integer linear programs. *Computers & Chemical Engineering* 131:106580.
- Hewitt, M., & Frejinger, E. (2020). Data-driven optimization model customization. *European Journal of Operational Research*, 287(2), 438–451.
- Liu, D., Fischetti, M. & Lodi, A. Learning to search in local branching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4), 3796–3803 (2022).
- Melchiori, L., Nasini, G., Montagna, J. & Corsano, G. A mathematical modeling for simultaneous routing and scheduling of logging trucks in the forest supply chain. *Forest Policy and Economics*, 136, 102693 (2022).
- Natural Resources Canada. The State of Canada’s Forests: Annual Report 2024. <https://natural-resources.canada.ca/sites/admin/files/documents/2025-07/StateofForestReport-2024-EN.pdf>, 2024.

- Pawlak TP, Krawiec K (2017) Automatic synthesis of constraints from examples using mixed integer linear programming. *European Journal of Operational Research* 261(3):1141–1157.
- Radio-Canada. Transport du bois : un nouvel outil plus efficace. Côté-à-côté, ICI Première, September 25, 2026. <https://ici.radio-canada.ca/ohdio/premiere/emissions/cote-a-cote/segments/rattrapage/2487526/transport-bois-un-nouvel-outil-plus-efficace>.
- Rönnqvist, M., D’Amours, S., Weintraub, A., Jofre, A., Gunn, E., Haight, R., Martell, D., Murray, A. & Romero, C. Operations research challenges in forestry: 33 open problems. *Annals of Operations Research*, 232, 11–40 (2015).
- Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. *IEEE Transactions on Neural Networks* 20(1):61–80.
- Weintraub, A., Epstein, R., Morales, R., Seron, J., & Traverso, P. A truck scheduling system improves efficiency in the forest industries. *Interfaces*, 26(4), 1–12 (1996).
- Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3), 229–256 (1992).