Optimizing Fuel Consumption and Pollutant Emissions in Truck Routing with Parking Availability Prediction and Working Hours Constraints

March 2022

A Research Report from the National Center for Sustainable Transportation

Filipe Vital, University of Southern California

Petros Ioannou, University of Southern California

National Center for Sustainable Transportation

METRANS Transportation Consortium
USC | CSULB

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. NCST-USC-RR-22-12	**2. Government Accession No.** N/A	**3. Recipient's Catalog No.** N/A
4. Title and Subtitle Optimizing Fuel Consumption and Pollutant Emissions in Truck Routing with Parking Availability Prediction and Working Hours Constraints		**5. Report Date** March 2022 **6. Performing Organization Code** N/A
7. Author(s) Filipe de Almeida Araujo Vital, https://orcid.org/0000-0001-5987-5993 Petros Ioannou, PhD, https://orcid.org/0000-0001-6981-0704		**8. Performing Organization Report No.** N/A
9. Performing Organization Name and Address University of Southern California METRANS Transportation Consortium University Park Campus, VKC 367 MC:0626 Los Angeles, California 90089-0626		**10. Work Unit No.** N/A **11. Contract or Grant No.** Caltrans 65A0686 Task Order 044 USDOT Grant 69A3551747114
12. Sponsoring Agency Name and Address U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE, Washington, DC 20590 California Department of Transportation Division of Research, Innovation and System Information, MS-83 1727 30th Street, Sacramento, CA 95816		**13. Type of Report and Period Covered** Final Research Report (March 2020 – December 2021 **14. Sponsoring Agency Code** USDOT OST-R

15. Supplementary Notes
DOI: https://doi.org/10.7922/G2S75DPP

16. Abstract
The transportation sector is responsible for a significant part of the U.S.'s greenhouse emissions, with a considerable amount being generated by medium- and heavy-duty trucks. However, when it comes to the trucking industry, 'green' routing studies do not consider other critical practical factors, like working hours regulations and parking availability. Due to parking shortages, routes and schedules that do not account for parking availability may lead to last-minute changes that make them more polluting than expected. Similarly, working hours regulations influence the timing of required rest stops, which may force drivers to deviate from initially selected routes and schedules with negative consequences to fuel consumption and emissions. This study addresses a variant of the shortest path and truck driver scheduling problem under parking availability constraints which focuses on optimizing fuel consumption and emissions by controlling the truck's travel speed and accounting for time-dependent traffic conditions. As it is impossible to be absolutely certain about the future parking availability of any location during planning, the case of stochastic parking availability was also studied. When studying the trade-offs between prioritizing emissions reduction or trip duration, it was found that although focusing on emissions reduction can increase trip duration significantly, this impact is greatly reduced when considering scenarios with limited parking availability. The problem formulation was further extended to model drivers' possible recourse actions when unable to find parking and the ensuing costs. This formulation was used to study how the solutions are affected by the level of information provided to drivers. It was found that ignoring uncertainty in parking availability results in inconsistent performance even when restricting parking to periods when probability of finding parking is high. Furthermore, results might not reflect the intent of the cost function used, e.g., minimizing illegal parking events and/or the priority assigned to emissions reduction. Giving drivers full information about the probability of finding parking at any time/location significantly improves performance and reduces illegal parking-related risks, but also substantially increase problem complexity and computation time. Using full information regarding parking availability but restricting the parking times to high availability time-windows can reduce complexity while maintaining consistent, although reduced, performance.

17. Key Words Hours of Service Regulations, Truck Driver Scheduling, Fuel Consumption Optimization, Parking-aware Planning	**18. Distribution Statement** No restrictions.		
19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 82	**22. Price** N/A

NCST

About the National Center for Sustainable Transportation

The National Center for Sustainable Transportation is a consortium of leading universities committed to advancing an environmentally sustainable transportation system through cutting-edge research, direct policy engagement, and education of our future leaders. Consortium members include: University of California, Davis; University of California, Riverside; University of Southern California; California State University, Long Beach; Georgia Institute of Technology; and University of Vermont. More information can be found at: ncst.ucdavis.edu.

Disclaimer

Acknowledgments

This study was funded, partially or entirely, by a grant from the National Center for Sustainable Transportation (NCST), supported by the U.S. Department of Transportation (USDOT) and the California Department of Transportation (Caltrans) through the University Transportation Centers program. The authors would like to thank the NCST, the USDOT, and Caltrans for their support of university-based research in transportation, and especially for the funding provided in support of this project.

NCST

Optimizing Fuel Consumption and Pollutant Emissions in Truck Routing with Parking Availability Prediction and Working Hours Constraints

A National Center for Sustainable Transportation Research Report

March 2022

Filipe Vital, Department of Electrical and Computer Engineering, University of Southern California

Petros Ioannou, Department of Electrical and Computer Engineering, University of Southern California

NCST

[page intentionally left blank]

TABLE OF CONTENTS

List of Tables

List of Figures

Optimizing Fuel Consumption and Pollutant Emissions in Truck Routing with Parking Availability Prediction and Working Hours Constraints

EXECUTIVE SUMMARY

According to the U.S. Environment Protection Agency (EPA, 2018), the U.S. Transportation sector is responsible for 28% of the US's greenhouse gas emissions, 23% of which are caused by medium- and heavy-duty trucks. This means that 6.4% of all greenhouse gas emissions in the U.S. are generated by trucks. Furthermore, this issue is not particular to the U.S.A. The European Union faces a similar problem, with almost 5% of their CO_2 emissions originating from heavy-duty vehicles (Gregor, 2018). Considering the continuous growth of the trucking industry, it is clear the importance of developing more efficient ways reduce trucks' pollutant emissions. However, when it comes to the trucking industry, `green' routing studies do not consider other important practical factors, like working hours regulations and parking availability. Due to parking shortages, routes and schedules that do not account for parking availability may lead to last-minute changes that make them more polluting than expected. Similarly, working hours regulations influence the timing of required rest stops, which may force drivers to deviate from initially selected routes and schedules with negative consequences to fuel consumption and emissions. Several fuel and emissions optimization problems have been treated in the literature, and we have developed regulation-compliant and parking-aware truck scheduling methods in previous projects, but the intersection of these problems is still a research gap. Currently, we still lack a model able to generate solutions with reduced environmental cost, yet accounting for practical constraints. This project's objective is to integrate fuel and emissions optimization, parking information, and regulation-aware scheduling to develop models able to better describe the practical constraints faced by drivers in realistic scenarios. We study methods for long-haul truck planning that generate regulation-compliant, parking-aware and environmentally friendly routes and schedules.

We extend the shortest path and truck driver scheduling problem model developed in a previous project (Vital & Ioannou, 2021a) to include controllable travel speed and time-dependent speed limits, and use a non-linear speed dependent fuel consumption model to optimize fuel consumption and pollutant emissions. When studying the trade-offs between prioritizing emissions reduction or trip duration, we found that although focusing on emissions reduction can increase trip duration significantly, this impact is greatly reduced when considering scenarios with limited parking availability. The scenarios studied showed reductions of up to 5-8% on average CO_2 emissions, which come at the cost of increases on average trip duration and average trip cost. However, results showed a large variance. Time-dependent instances with high priority set to reducing emissions showed an average increase in trip duration higher than 40% when parking is abundant, but lower than 20% when parking is scarce. At times prioritizing emissions might be too costly, but this cost is also influenced by the region's parking infrastructure conditions. These results illustrate the importance of improving

the models used to evaluate the impact of any policy and investment decisions. The proposed model can help estimate the level of emissions reduction that can be expected for different regions and types of vehicles, at what cost, and how they are affected by the region's truck parking infrastructure. We also present a cost lower bound that combines HOS requirements with information on optimal speeds for particular cost functions, and can be used to significantly speed-up problem solution in deterministic scenarios.

Afterwards, we considered the case of stochastic parking availability, as opposed to the deterministic time-windows considered initially. The resource-constrained shortest path formulation was further extended to model drivers possible recourse actions when unable to find parking and the ensuing costs. We used this formulation to study how the solutions are affected by the level of information provided to drivers. We found that ignoring uncertainty in parking availability results in inconsistent performance even when restricting parking to periods when probability of finding parking is high. Furthermore, results might not reflect the used cost function's intent, e.g., minimizing illegal parking events and/or the priority assigned to emissions reduction. Giving drivers full information about the probability of finding parking at any time/location significantly improves performance and reduces illegal parking-related risks, but also substantially increases problem complexity and computation time. Using full information regarding parking availability but restricting the parking times to high availability time-windows can reduce complexity while maintaining consistent, although reduced, performance.

Truck parking is a critical issue in the USA, and it can have a significant impact on the environment and industry costs. Integrating truck parking information in the planning process can mitigate this issue by recommending safer, more efficient itineraries to drivers. Although, in general, parking availability is uncertain and we cannot guarantee a parking space, by including this uncertainty in the model users can better manage the risks.

Introduction

According to the U.S. Environment Protection Agency (EPA, 2018), the U.S. Transportation sector is responsible for 28% of the US's greenhouse gas emissions, 23% of which are caused by medium- and heavy-duty trucks. This means that 6.4% of all greenhouse gas emissions in the U.S. are generated by trucks. Furthermore, this issue is not particular to the U.S.A. The European Union faces a similar problem, with almost 5% of their CO_2 emissions originating from heavy-duty vehicles (Gregor, 2018). The European Commission proposed, in 2018, targets for the reduction of emissions in new heavy-duty vehicles, showing that there is a growing concern with the topic (Gregor, 2018). Similar measures have already been adopted by the state of California as an effort to improve its fleet's efficiency and curb CO_2 emissions. Although California was able to reach its total emissions reduction targets early by pushing for the usage of renewable energy and greener technologies, the emissions caused by the transportation sector keep rising, and heavy-duty vehicles still count for around 8% of the state's CO_2 emissions (Barboza & Lange, 2018). Considering the continuous growth of the trucking industry, it is clear the importance of developing more efficient ways of using the trucks, trying to reduce their emissions as much as possible.

The problem of trying to minimize the fuel/energy consumption and pollutants emissions in the transportation sector is not new. Several studies have approached this topic both for passenger vehicles and trucks. A survey on 'green' vehicle routing can be found in (Eglese & Bektas, 2014), which gives an overview of the types of models used for fuel consumption and emissions, and of the different variants of the vehicle routing problem which involve environmental factors. Multiple models have been developed to estimate the fuel consumption and pollutants emissions based on different factors and targeting the usage on problems of different scales. Some models consider only the average speed of the car, as the one used in (Van De Hoef et al., 2015), but more precise models may consider the vehicle load (Zhang et al., 2015), road incline, and if the vehicle is accelerating, decelerating or cruising (Demir et al., 2011). Reference (Demir et al., 2011) presents a comparison of different fuel consumption models. These models are then used to give an environmental aspect to transportation problems. These problems can be divided based on their time-dependency (time-dependent traffic conditions or not), choice of decision variables (route, number of vehicles, travel speed, departure time, etc) and choice of cost function (only environmental factors or multi-objective). The problem presented in (Zhang et al., 2015) considers the impact of load and speed on carbon emissions and fuel consumption. Simulation results showed fuel consumption reductions of up to 18%. The traffic conditions are not considered, so the proposed model is time-independent. Also, the speed is taken as a parameter of the road, not as a decision variable for the model, so the model can choose to use a faster or slower road when convenient, but it cannot tell the driver to drive slower than the road 'regular' speed. In (Van De Hoef et al., 2015), the speed is used as a decision variable to optimize fuel consumption during the trip and to facilitate the organization of truck platoons. Although the treated problem is a routing problem, the fuel consumption optimization is only performed after a shortest path has been chosen for each vehicle. At this point it turns into a scheduling problem, as the route is given. Other variants of the problem consider time-dependent traffic conditions on the road networks (Franceschetti et al., 2013) or the impact of

different idling options used when resting (Koç et al., 2016). Several studies targeted the benefits of truck platoons (Van De Hoef et al., 2015), (Agriesti et al., 2018; Tsugawa, 2014; Tsugawa et al., 2016), showing that it is possible to achieve fuel consumption reductions in the range of 5%-15% depending on platoon speed, gap between trucks and position of the truck inside the platoon.

The problem is that most of these fuel/emissions-efficient models do not address important practical constraints of the trucking industry, i.e. working hours or Hours-of-Service (HOS) regulations and parking availability. These factors are particularly important to long-haul truck drivers, which are usually not the focus of these studies. In (Koç et al., 2016), the author considered the working hour regulations and included an environmental cost based on the emissions generated by truck idling depending on the equipment installed in the truck and the one available at the parking location. However, this model did not consider the possibility to optimize the emissions/fuel by controlling the speed of the vehicle, and also did not consider traffic conditions. In most cases, the studies focused on the trucking industry which consider working hours regulations and, to a certain extent, parking focus on the monetary costs directly accrued by the trucking company. Thus the objective function usually considers only the total trip duration (Asvin Goel, 2012; Kok et al., 2011), or a function of the total trip duration and the working time or travel distance (Rancourt et al., 2013). There are studies on the combined routing and scheduling problem, and studies focused only on scheduling, but it is the scheduling problem the one responsible for the practical feasibility of the solutions when subject to regulations and parking availability. Some methods allow the driver to rest at any point during the route (Asvin Goel & Irnich, 2017; Asvin Goel & Vidal, 2014; Kok, Meyer, et al., 2010; Rancourt et al., 2013; Xu et al., 2003), not considering the need for an appropriate rest location, others only allow the driver to rest at truck stops and/or client locations (Asvin Goel, 2012; Koç et al., 2016; Kok et al., 2011). Most studies do not consider time-dependent travel times, however there are still some that do (Kok et al., 2011; Kok, Hans, et al., 2010; Shah, 2008).

A part of this problem which is still overlooked most of the time is the issue of parking availability. Most models assume that any valid parking location will always be free, which is unrealistic as appropriate truck parking is an issue both in the U.S.A. (U.S. Department of Transportation, 2015) and in Europe (SETPOS Consortium, 2009). In (Vital & Ioannou, 2019, 2020), Vital and Ioannou studied the problem of including both parking availability information and HOS regulations in the planning of long-haul transportation, but those studies do not cover fuel consumption and emissions. Several variants of the two sides of this problem, fuel/emissions optimization and scheduling with working hours regulations, have been studied. Currently, what we lack is a model that can integrate both sides. A model able to generate solutions with reduced environmental cost, but that are still feasible in practice. In this project, we address this research gap by extending the shortest path and truck driver scheduling problem under parking availability constraints (Vital & Ioannou, 2021a), which focused on the working hours regulations and parking availability constraints, to consider the impact of traffic conditions and different travel speeds in the fuel consumption and pollutants emissions of the trucks, as well as how uncertainties in the parking availability affect the problem's solutions.

This report is organized as follows: Section Related Work reviews related work. Section USA's Hours of Service Regulations describes the HOS regulation considered. Section Problem Description describes the problem being studied. Section Model presents the model used to represent the problem. Section Dynamic Programming Formulation and Rollout Algorithm describes the dynamic programming formulation used to solve the problem. Section Parking Availability Uncertainty extends the model to consider stochastic parking availability. Section Experiments presents the experiments and results. Section Feasibility of Commercialization and possible model extensions discusses the system's commercialization potential. Section Conclusion presents the conclusion.

Related Work

Time-dependent Shortest Path Problem

The time-dependent shortest path problem was first studied by Cooke and Halsey (1966) (Cooke & Halsey, 1966), who extended bellman's equations to time-dependent networks and presented an dynamic programming solution for the discrete time problem. Since then, polynomial time solutions have been proposed for networks with the FIFO (first in, first out) property, i.e., one cannot arrive earlier at the end of an arc by departing later. (Kaufman & Smith, 2007) proves that, in problems where the network has the FIFO property the complexity of labeling algorithms for time-dependent networks is the same as for static networks. The FIFO assumption holds in practice for many networks, including transportation networks. Most algorithms proposed for the time-dependent shortest path problem are based on the Dijkstra and A* algorithms often studied for the static problem (Dean, 2004; Dell'Amico et al., 2008; Delling, 2011; Ferone et al., 2017; Nannicini et al., 2008; Ziliaskopoulos & Mahmassani, 1993). (Ziliaskopoulos & Mahmassani, 1993) proposed an algorithm to calculate simultaneously all the shortest paths from all nodes to a given destination node and for every discrete time step in a network with time-dependent arc costs. They presented a label correcting method that uses a bottom-up dynamic programming approach to calculate shortest paths. The problem does not assume FIFO networks. In (Dean, 2001), Dean studies the theoretical properties of time-dependent shortest path problems, and presents serial and parallel algorithms for the problem of calculating the earliest arrival time at one or more nodes in FIFO networks. In (Zhao et al., 2008), Zhao generalizes the A* algorithm for time dependent networks. The algorithm correctness is guaranteed if the used time-dependent estimator functions satisfy the proposed sufficient conditions. The landmark based ALT algorithm is also extended to the time-dependent case. Landmarks with precalculated optimal travel times to every node (at every time) are used to estimate lower bounds for the travel times from every node to the destination. These lower bounds satisfy the sufficient conditions proposed and are used in the A* algorithm to guide the exploration of the search space, improving performance. In (Nannicini et al., 2008), Nannicini et al. present a bidirectional search method that is also based on ALT algorithms. In (Delling, 2011), Delling presents a time-dependent version of the SHARC algorithm, which uses preprocessing routines based on highways hierarchies (Sanders & Schultes, 2006) and arc-flags (Lauther, 2004) to speed-up a Dijkstra-based algorithm.

Another important consideration is whether waiting at nodes is allowed. In (Orda & Rom, 1990), Orda & Rom showed that if waiting is allowed, then a shortest path can be found in polynomial time even without the FIFO assumption. However, if waiting is not allowed and the network does not have the FIFO property, the problem is NP-hard (Orda & Rom, 1990). Later, Foschini et al studied in detail the complexity of the arrival time function and of algorithms searching for a minimum delay path (Foschini et al., 2011). Omer and Poss (Omer & Poss, 2020) proposed a polynomial time algorithm for the case when wait times are allowed at all nodes, but when those wait times are not considered in the cost function. The algorithm calculates the shortest paths while iteratively increasing the maximum allowed total wait time. The authors used FIFO network and piece-wise linear travel time assumptions to prove that an optimum solution can be found while testing only a finite number of total wait times. The length of the sequence of wait times that need to be considered depends on the total number of breakpoints of all travel time functions. The sequence of wait times to be tested is chosen so that shortest paths passing through each node can be written as a concatenation of a shortest path found in the previous iteration, the wait time increase for the current iteration, and a path to the destination without waits. The problem addressed does not consider time-windows, the only constraint is on total waiting time.

(Huang et al., 2017) uses a time-delay neural network with the same topology as the road network to calculate the shortest path to a given destination node when travel time between nodes is defined by piecewise constant functions. The time complexity depends on the product of the number of time-windows needed to describe the travel time functions, and on the shortest path's arrival time at the destination node (in time steps). This is an interesting line of research, as, compared to methods based on Dijkstra and A*, the time complexity is not as affected by the network size.

Most studies focus on minimizing trip duration, arrival time or driving time. However, in practice, those are not the only relevant objectives. For example, the transportation companies might want to minimize fuel consumption, emissions, safety risks, or monetary costs. These types of objective functions may not satisfy FIFO assumptions and can be more problematic to deal with, as studied in (Orda & Rom, 1991). (Orda & Rom, 1991) showed that in some time-dependent minimum weight path problems there is no finite optimal path, and proposed conditions for the existence of finite optimal paths. (Cooper & Cowlagi, 2018) study path planning under time-dependent cost functions modeled as a spatiotemporal scalar field. As an example, they mention that minimizing a weighted sum of travel duration and exposure to traffic might be useful in reducing emissions-related health risks to long-haul truck drivers. They study the effects of allowing waiting under scalar fields defined by linear combinations of Gaussian functions and propose local conditions to prune search trees used by graph search algorithms. In (He et al., 2020), He et al. study problem variants where a subset of nodes has penalties for waiting or where there is a limit on the total waiting time at a subset of nodes. They proved that some variants are NP-Hard and proposed polynomial time algorithms for the ones that are not. In (Cai et al., 1997), Cai et al. consider node-dependent upper-bounds on the waiting time at each node.

The work on time-dependent shortest paths problems with constraints is limited. The most common constraints considered are applied to the waiting time, as in (Cai et al., 1997; He et al., 2020), or total trip duration (Wenting & Xiaoqiang, 2007). (Sherali et al., 2003) addresses time-dependent label-constrained shortest path problems which restrict the structure of acceptable paths. Besides the usual setup for shortest path problems, each arc is ascribed a label, and the acceptable label sequences are defined by a `language'. For example, the labels can represent travel modes, such as walk, drive, and bus, and the language can specify that only paths using at most 2 labels (travel modes) are acceptable. Although variants with time-window constraints are common for the static shortest path problem that is not the case for the time-dependent version. In (Spliet et al., 2018), Spliet et al. studied a problem variant with time-window constraints in the context of a routing problem. Spliet et al. presented an exact labeling algorithm and a heuristic tabu search algorithm for the shortest path problem with a capacity constraint, time-dependent travel times, time window constraints on both the nodes and on the arcs, and linear node costs. The labeling algorithm was based on the algorithm proposed by (Ioachim et al., 1998) for a similar variant, where the travel time is not time-dependent, but nodes have time-windows and time-dependent costs. (Mayerle et al., 2020) studied the time-dependent shortest path problem under time-window and hours-of-service regulation constraints, where the solution represents not only a path, but also a schedule specifying for how long the driver must rest at each location. They define a state-graph where each node is a certain stoppage configuration and present a Dijkstra-based algorithm with a pruning heuristic to find good solutions. A related problem was studied in (Kok et al., 2011), where Kok et al. presented an integer linear programming formulation for the time-dependent truck driver scheduling problem. Although their formulation considers a fixed path, it can be used as a post-processing step for shortest path or vehicle routing problems under time-window and hours-of-service regulation constraints.

Truck Driver Scheduling Problem

The inclusion of HOS rules in scheduling algorithms, the truck driver scheduling problem (TDSP), was approached in many studies in recent years (Archetti & Savelsbergh, 2009; A. Goel, 2010; Asvin Goel, 2012; Asvin Goel & Kok, 2012; Koç et al., 2016; Vital & Ioannou, 2019). Multiple regulations have been considered, including ones from the United States (Asvin Goel & Kok, 2012), Europe (A. Goel, 2010) and Canada (Asvin Goel & Rousseau, 2011). Furthermore, it is often studied as part of a vehicle routing and truck driver scheduling problem (VRTDSP) (Gaddy et al., 2018; Asvin Goel & Vidal, 2014; Koç et al., 2018; Kok et al., 2011; Kok, Hans, et al., 2010; Rancourt et al., 2013), which is a variant of the vehicle routing problem (VRP) that accounts for HOS rules, and, less commonly, it is studied in the context of shortest path problems (SPP) (Drexl & Prescott-Gagnon, 2010; Asvin Goel & Irnich, 2017; Mayerle et al., 2020). Besides the particular methods used, the differences between problems treated in the literature usually relate to the following aspects: regulation considered, optimality of the solutions, parking restrictions, cost function, and main problem (TDSP, VRP or SPP). We are most interested in how they approached parking restrictions and path planning.

Parking restrictions

Although truck parking is currently a critical issue, it is often overlooked in the literature, with many methods not even restricting parking to appropriate facilities. In (Archetti & Savelsbergh, 2009), Archetti et al. considered the problem of determining whether a sequence of n full truckload transportation requests is feasible given a set of HOS regulations and pick-up time-windows. The proposed method allows drivers to park anywhere and finds a feasible schedule in $O(n^3)$ time. In (A. Goel, 2010), Goel considered a similar problem using the European regulations, and in (Asvin Goel & Kok, 2012), presented an algorithm to find feasible schedules to visit n locations using the US regulations in $O(n^2)$ time. However, these methods assumed that drivers could park anywhere, which is not valid in practice. This assumption is also present in (Drexl & Prescott-Gagnon, 2010; Asvin Goel & Irnich, 2017; Asvin Goel & Vidal, 2014; Kok, Meyer, et al., 2010; Rancourt et al., 2013). In (Asvin Goel, 2012), Goel presented a mixed integer programming (MIP) formulation and a dynamic programming algorithm for the TDSP that restricts parking to client locations and calculates a schedule with minimum trip duration. Rest areas were modeled as clients with zero service time and unbounded time-windows. Similar MIP models were used in (Koç et al., 2016; Kok et al., 2011; Vital & Ioannou, 2019), focusing on different aspects of the problem but keeping parking restricted to appropriate facilities. In (Kok et al., 2011), Kok et al. addressed the issue of traffic congestion by considering time-dependent travel times and proposed a heuristic approach to integrate the TDSP model into a VRP method. In (Koç et al., 2016), Koç et al. approached the environmental impact caused by truck idling and how it is affected by the truck's equipment and rest areas' infrastructure. The drivers can only park at rest areas, which have different types of infrastructure available. Early arrival is allowed at client locations, but it does not count as off-duty time. The cost function accounts for the type of idling used in each stop given the equipment installed on the truck and the infrastructure available at each rest area. This method was later used as a base for a VRTDSP algorithm with the same focus (Koç et al., 2018). In (Vital & Ioannou, 2019), Vital and Ioannou approached the issue of truck parking availability and US HOS rules for long trips. Their model considered a single client trip, which hinders drivers' ability to plan consecutive trips. Parking was restricted to rest areas, and parking availability was modeled as time-window constraints for each rest area. Each rest area's availability time-windows take effect only if a stop is scheduled for that particular location. Due to the focus on parking availability issues, the model assumed that parking is unavailable outside of the delivery time-window and did not allow early arrival at the client or rest areas. As short-term staging due to warehouse or terminal hours is a source of truck parking demand (Cambridge Systematics, 2019; U.S. Department of Transportation, 2015), we see the restriction on early arrivals (also included in our model) as an important distinction when considering truck parking shortages. This study is the only one that considered time-dependent parking availability in the TDSP. Nevertheless, as (Vital & Ioannou, 2019) addresses only the scheduling problem, it does not account for alternative paths or parking locations that require a detour to be reached. This limitation motivates the other aspect of our work: path planning.

Path planning

The inclusion of parking constraints and HOS regulations when determining the shortest path between locations is relevant not only to individual drivers that need to plan their itineraries, but also to carriers and other stakeholders that need to estimate operational costs and allocate resources. Hence why we are interested in the shortest path problem with resource constraints (SPPRC) that lies between the TDSP and the VRTDSP. VRTDSP methods assume that the shortest path between any two clients is known (and independent of the current status of the HOS constraints), and use TSDP algorithms to calculate the cost of each route generated. The rest areas considered in these problems are located along these known shortest paths. If the driver is allowed to rest anywhere or only at client locations, this assumption does not affect the route cost. However, when parking is restricted and rest areas are considered, the minimum cost path between two clients will depend on the location of every reachable rest area and the HOS constraints' status at the departure time from the client. The inclusion of parking availability constraints makes it even more important to consider alternative paths and rest areas. When parking is scarce at the usual routes, it may be cost-effective to take a slightly longer path if it has better parking conditions. Failing to consider how parking availability and HOS constraints affect the shortest path between clients may cause planners to underestimate the trip's duration and cost. This inaccuracy can upset operations planning as well as fair driver remuneration (depending on how wages are determined). The issue is aggravated when drivers lack the flexibility to adjust their route, as some of the drivers surveyed in (Sun et al., 2013). In this case, the driver is limited to taking a sub-optimal route, further increasing the difference between estimated and actual trip cost and duration.

The shortest path problem with resource constraints (SPPRC) often appears in column generation solutions to the VRP (Costa et al., 2019) and several approaches have been proposed for its variants (Asvin Goel & Irnich, 2017; Horváth & Kis, 2016; Irnich & Desaulniers, 2005; Lozano et al., 2016; Pugliese & Guerriero, 2013). The SPPRC is often solved through dynamic programming-based labeling algorithms, applying tailored dominance rules and bound estimates to identify and discard inferior paths. SPPRC formulations and algorithms are tailored to their own problem variants and may not be directly applicable to other problems. Hence the need to develop tailored methods for the SPPRC in the context of HOS regulations and parking availability constraints. However, the number of studies using SPPRC formulations in the context of HOS-compliant planning is very limited. In (Drexl & Prescott-Gagnon, 2010), Drexl and Prescott-Gagnon present a SPPRC formulation to the problem of finding HOS-compliant routes and schedules, and propose exact and heuristic labeling algorithms. In (Asvin Goel & Irnich, 2017), Goel and Irnich propose an exact method for the VRTDSP using a branch and price algorithm where a SPPRC is used to generate HOS-compliant routes and their costs. An auxiliary network is used to model drivers' possible activities, but parking locations are not considered. Even though they consider HOS regulations, both (Asvin Goel & Irnich, 2017) and (Drexl & Prescott-Gagnon, 2010) assume that drivers may stop and rest anywhere on a route. This limitation is partially addressed in (Mayerle et al., 2020), where Mayerle et al. study the impact of Brazilian regulations in the planning of long-haul full truckload shipments. Differently from (Drexl & Prescott-Gagnon, 2010; Asvin Goel & Irnich, 2017), this study is not aimed at deciding

which clients to visit and in what order for a VRTDSP, but at how changes to HOS rules affect the best path to reach a client. They use a labeling algorithm and pruning heuristics to optimize the path a truck takes to reach a single client, while scheduling stops at allowed locations to satisfy regulations. Their model includes some time-restrictions to all rest stops by restricting departure times at the beginning of each work day, as well as the start time of lunch breaks. However, they also overlook the question of whether those parking locations will be available at the desired times. In addition, it shares the same single client limitation as (Vital & Ioannou, 2019).

USA's Hours of Service Regulations

The USA HOS regulation restricts for how long drivers can drive/work, and how long they should rest before being allowed to drive again. We refer to the off-duty periods required by the regulation based on their minimum duration: *breaks* (0.5 h), *daily rests* (10 h) and *weekly rests* (34 h). The USA HOS regulation can be summarized as follows (Federal Motor Carrier Safety Administration, 2021):

- 11-hour Driving Time Limit: A driver may drive at most 11 hours between 2 consecutive *daily rests*.
- 14-Hour Elapsed Time Limit: A driver cannot drive after 14 hours have elapsed since the last *daily rest* ended.
- Rest Breaks: A driver must take a *break* after 8 cumulative hours of driving time. Recent changes in the regulation allow this constraint to be satisfied by any non-driving period of 30 consecutive minutes.
- 60-Hour Limit: A driver cannot drive after having been on duty for 60 hours in any period of 7 consecutive days. The 7 days period can be reset by taking a *weekly rest*.

We do not consider the sleeper berth provision, which allows *daily rests* to be split. And, for the 60-hour limit, instead of restricting the on-duty time over any period of 7 consecutive days, the on-duty time between two consecutive *weekly rests* was restricted to 60 hours.

Problem Description

In this project, we first address a `green' variant of the SPTDSP-PA (shortest path and truck driver scheduling problem with parking availability constraints) (Vital & Ioannou, 2021a). This variant differs mainly by the inclusion of fuel consumption in the objective function, the usage of travel speed as a decision variable (in order to control fuel consumption), and for considering time-dependent traffic conditions. Then, to make the model more realistic, section Parking Availability Uncertainty extends it to include uncertainty in parking availability. In practice, it is impossible to be certain about the future parking availability of any location during planning. Therefore, we include this uncertainty in the model and study its effect on the solutions depending on the information provided to drivers/planners.

The problem consists of planning the path and schedule for a truck starting at an origin location and visiting an ordered list of clients, where the last client is referred to as the destination. Each

client has a fixed non-negative service time, and time-window constraints restricting the vehicle's arrival time. The schedule must comply with HOS (Hours-of-Service) regulations, which impose restrictions on how long the driver can work or drive without resting, and the minimum duration of rests (rests of different durations satisfy different restrictions). Drivers can rest only at rest areas, but arrival time at rest area nodes is also subject to time-window constraints (representing parking availability). The problem is solved over a simplified road network that includes only the main routes the truck can take between two consecutive client locations, and the rest areas around them. The simplified road network is defined as an acyclic directed graph $G = (V, A)$, where V is the set of nodes of the graph and A is the set of edges. The vehicle consumes fuel when driving or idling. We consider the consumption model for diesel trucks defined in section Consumption Model. The driving consumption rate is described by a non-linear speed-dependent function, and the idling consumption rate is taken as constant. Each road section $(i, j) \in A$ has a fixed length d_{ij} and a time-dependent allowed speed range $\left[s_{ij}^{-}(t), s_{ij}^{+}(t)\right]$, thus setting the allowed travel time to $\left[\frac{d_{ij}}{s_{ij}^{+}}, \frac{d_{ij}}{s_{ij}^{-}}\right]$. The average travel speed can be adjusted within the allowed range to control the travel time and energy consumption. We assume that the speed profiles are defined such that all edges satisfy FIFO assumptions when considering only one of the speed limits.

During long trips, HOS regulations require drivers to rest along the way. Rest stops are restricted to rest areas and their minimum durations are defined by the regulation. We do not allow for rests to be taken at client locations. However, note that service times longer than 30min can reset the 8h driving limit constraint despite counting as on-duty time for other constraints. Each parking location has a set of time-windows representing the intervals when parking spaces are expected to be available. These time-windows restrict the vehicle's arrival time. The vehicle is not allowed to arrive early and wait. The regulation sets a minimum duration for the rest stops, but it does not set a maximum duration, so the driver is allowed to extend the stay when convenient. Similarly, each client has a set of time-windows constraints and a service time, which define when the truck can arrive at the client and the duration of stay. However, drivers cannot extend the service time at the client. As rest areas are not required stops, the graph G is built so that rest areas can be bypassed. Clients are mandatory stops, so all considered routes go through the client nodes.

As this project addresses the issue of emissions reduction, we want this to be reflected in our objective function. We take as objective function a linear combination of trip duration and fuel consumption. The trip duration term accounts for driver wages and operational costs (excluding fuel), whereas the fuel term accounts for fuel and emissions costs. The emissions costs can be seen both as some kind of carbon pricing, or simply the level of importance attached to reducing emissions as opposed to reducing trip duration. Consider the following cost function for a decision of duration δ:

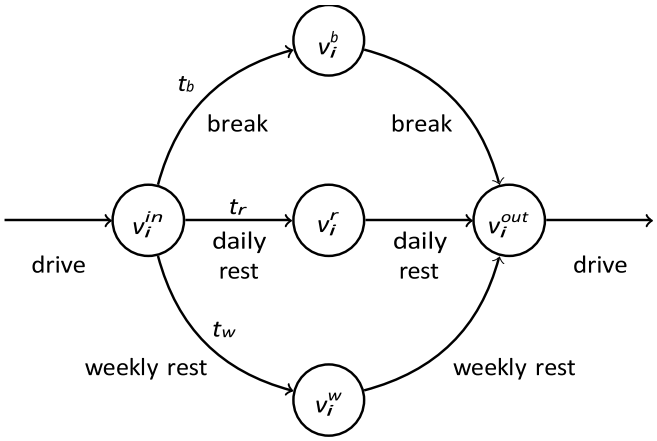
$$g(\delta) = \begin{cases} \alpha\delta + \beta\mu_e\zeta(\mu_e/\delta), & if\ driving \\ (\alpha + \beta\gamma + \theta)\delta, & o.w. \end{cases} \quad (1)$$

where α is the trucks hourly operational cost (excluding fuel) and β is the cost per unit of fuel/energy. For non-driving decisions, γ is the hourly idling fuel consumption, θ represents hourly costs incurred while stopped from sources other than idle fuel consumption and operational costs. For driving decisions, μ_e is the length of the road segment considered, and $\zeta(v)$ is the fuel consumption per unit of distance. This cost function considers both time and fuel related costs, and their relative importance can be adjusted using the parameters α, β, and θ.

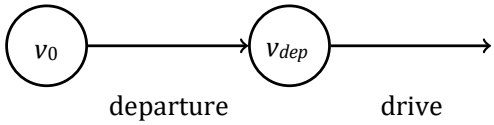
Model

The problem was formulated as a resource constrained shortest path problem as in (Vital & Ioannou, 2021a). The formulation uses an auxiliary network to explicitly model drivers' activities. Time and the counters associated with the different HOS regulations are treated as resources. Time is subject to time-window constraints at client and rest area nodes. Each HOS-related resource has a different upper limit, and depending on the activity being performed, these resources can increase (by the activity's duration), keep their current value, or be reset to zero. For example, a resource tracking driving time is not affected by edges representing service time at a client; a resource tracking the elapsed time since the last 10h rest (daily rest) will increase during a 30min stop (break), but will be reset to zero during a daily rest.

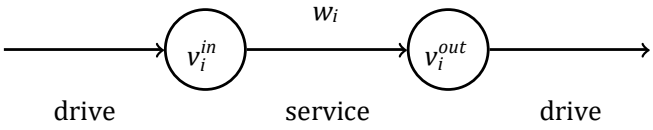
At the origin, client locations and rest areas, drivers perform non-driving activities. These nodes are expanded according to the subnetworks in Figure 1, forming an extended network $G' = (V', A')$ that includes non-driving activities explicitly in the graph. Each edge has its activity indicated below the arrow. Edges that have a fixed duration have their duration indicated above the arrow. The incoming/outgoing edges of the subnetwork are the incoming/outgoing edges of the node being expanded. The duration of edges representing rest extensions (edges marked with δ's in Figure 1a) was already controllable in (Vital & Ioannou, 2021a). When speed optimization is included, the duration of driving edges will be controllable within a given range, and in the time-dependent case, the range of allowed speed varies with time.



(a) Sub-network used to expand rest area nodes.



(b) Sub-network used to expand the origin node.



(c) Sub-network used to expand client nodes. w_i is the service time.

Figure 1. Sub-networks used to model non-driving activities.

System Equations

We consider the system's state as being a vector $x_k = (v_k, \theta_k)$, where $\theta_k = (\eta_k^0, \eta_k^b, \eta_k^r, \psi_k^r, \psi_k^w)$ containing the truck's current location ($v_k \in V'$) and the current resource values (θ_k). The resources are responsible for tracking the HOS restrictions, and arrival time at each node. The resources used are:

- η^0: Arrival time at current location
- η^b: Accumulated driving time since last *break*
- η^r: Elapsed time since last *daily rest*
- ψ^r: Accumulated driving time since last *daily rest*
- ψ^w: Accumulated on-duty time since last *weekly rest*

The evolution of the system is described by $x_{k+1} = f(x_k, u_k)$, where x_k is the current state, x_{k+1} is the next state, and u_k is the decision taken. The decision u_k is composed by an edge $e_k = (v_k, v_{k+1}) \in A'$, with length μ_k, and a duration δ_k included in e_k's allowed duration set. When dealing with edges related to driving, this set is defined by the length and allowed speed values of e_k. The function $f(x_k, u_k)$ defines how each element of x_k is affected by a decision u_k, and the resulting next state x_{k+1}. As different activities have different impacts on each

resource, each edge of the extended network has an activity assigned to it. Table 1 shows how the resources are updated depending on the activity. The functions f^d, f^s, f^b, f^r, f^w and f^0 describe the update rules for activities *drive*, *service*, *break*, *daily rest*, *weekly rest* and *departure*, respectively. Figure 1 shows how the activities are assigned to each edge. Note that η^b's and f^b's definitions differ from (Vital & Ioannou, 2020) due to recent changes in the regulation. Now the 8h limit is applied to driving time instead of elapsed time, and any non-driving period longer than 30 minutes can satisfy this constraint.

Table 1. Resource Extension Functions

	f^d	f^s	f^b	f^r	f^w	f^0
$\eta^0_{k+1} =$	$\eta^0_k + \delta_k$					
$\eta^b_{k+1} =$	$\eta^b_k + \delta_k$	$\begin{cases} 0, & if\ \delta_k > t_b \\ \eta^b_k, & o.w. \end{cases}$	0			η^b_k
$\eta^r_{k+1} =$	$\eta^r_k + \delta_k$			0		η^r_k
$\psi^r_{k+1} =$	$\psi^r_k + \delta_k$	ψ^r_k		0		ψ^r_k
$\psi^w_{k+1} =$	$\psi^w_k + \delta_k$		ψ^w_k		0	ψ^w_k

Consumption Model

The fuel consumption depends on the activity being considered, so we separate the model in driving, and idling. The model parameters considered are listed in Table 2.

Driving

We used the model presented by Wang and Rakha in (Wang & Rakha, 2017). More specifically, the parameters used are the ones for a convex model of a Freightliner/FLD 120, year 2001, labeled as ``HDDT8'' in their paper. The model first estimates the vehicle's power demand due to resistance forces acting on the vehicle, then estimates the consumption rate based on the power demand. We consider the average travel speed over each road section, and terms relative to acceleration and road grade were omitted. This model characterizes fuel consumption as a second-order polynomial function of the power demand, as follows:

$$P_D(v) = \left(\frac{\rho A C_D}{25.92} v^2 + mgC_R(c_1 v + c_2)\right)\frac{v}{3600\eta_d} \quad (2)$$

$$\zeta_D(v) = (\alpha_0 + \alpha_1 P_D(v) + \alpha_2 P_D(v)^2)\frac{3600}{v} \quad (3)$$

where $P_D(v)$ represents the power demand (kW), and $\zeta_D(v)$ represents the fuel consumption per distance (L/km). C_D is the drag coefficient (unitless). C_R, c_1 and c_2 are the rolling resistance parameters (unitless), η_d is the driveline efficiency (unitless), α_0, α_1 and α_2 are vehicle-specific model coefficients calibrated in (Wang & Rakha, 2017) using empirical data. The

air density (kg/m^3) is given by ρ, and the acceleration due to gravity is given by g. The terms v, m and A represent the truck's speed (km/h), mass (kg) and frontal area (m^2), respectively.

Idling

When `idling', we consider a fixed consumption rate F_I (L/h). Service time at clients is considered idling time.

Table 2. Model Parameters

Parameter	Description	Value
C_D (Wang & Rakha, 2017)	coefficient of drag	0.78
C_R (Wang & Rakha, 2017)	coefficient of rolling resistance	1.25E-3
c_1 (Wang & Rakha, 2017)	coefficient of rolling resistance	0.0328
c_2 (Wang & Rakha, 2017)	coefficient of rolling resistance	4.575
η_d (Wang & Rakha, 2017)	driveline efficiency	0.94
m **(**kg**)** (Wang & Rakha, 2017)	truck's total mass	3.6E4
A **(**m^2**)** (Wang & Rakha, 2017)	truck's frontal area	10
α_0 (Wang & Rakha, 2017)	vehicle-specific model coefficient	2.16E-3
α_1 (Wang & Rakha, 2017)	vehicle-specific model coefficient	7.98E-5
α_2 (Wang & Rakha, 2017)	vehicle-specific model coefficient	1.0E-8
F_I**(**L/h**)** (U.S. Department of Energy, 2015)	idling fuel consumption	3
g**(**m/s^2**)** (Wang & Rakha, 2017)	gravity	9.8066
ρ**(**kg/m^3**)** (Sripad & Viswanathan, 2017)	air density	1.2256
β_d**(**kg/L**)** (Argonne National Laboratory, 2020; U.S. Energy Information Administration, 2016)	CO_2 emission factor for diesel	3.13

Dynamic Programming Formulation and Rollout Algorithm

Let $J(x_k)$ be the minimum cost to go from state x_k to the destination, and X_d the set of feasible states at the destination node. This cost-to-go function is defined as:

$$J(x_k) = \begin{cases} 0, \text{ if } x_k \in X_d \\ \min\limits_{u \in U(x_k)} g(x_k, u) + J\big(f(x_k, u)\big), \text{o.w.} \end{cases} \quad (4)$$

where $g(x_k, u)$ is the cost accrued by decision u at state x_k, and $U(x_k)$ is the set of decisions u for which $f(x_k, u)$ is a feasible state. A state is considered feasible if all resources are within their respective feasible ranges. If $U(x_k)$ is empty, we say that the destination cannot be reached from x_k and $J(x_k)$ is infinite. The choice of $g(\cdot)$ determines what is being minimized. In this project, we use the cost function (1), which combines fuel and trip duration costs.

Although any node has only a finite number of outgoing edges, the decision space $U(x_k)$ can have uncountably many elements if the allowed duration set of one or more of these edges is a continuous interval. In order to mitigate this issue, we first propagate the constraints of each node to all upstream nodes. This reduces the feasible space at each node and the decision space to be considered for each decision. During execution, the algorithm uses the preprocessed feasible ranges to generate a reduced decision space, which is then discretized, generating a finite set of decisions. Nevertheless, due to the curse of dimensionality, this approach does not scale well for large instances. Using a coarse decision space discretization can bring significant improvements to computation time, but will also cause the cost to deteriorate. Therefore, we use a rollout algorithm (Bertsekas, 2017) to find suboptimal solutions while keeping the computational demand in check. The general idea is to use the cost obtained from applying a base policy as an approximate cost function, then use this approximation to generate a one-step lookahead policy. One-step lookahead policies choose the decision that minimizes the following expression:

$$\min_{u_k \in U(x_k)} g(x_k, u_k) + \tilde{J}\big(f(x_k, u_k)\big) \tag{5}$$

where $\tilde{J}(x_k)$ is the approximated cost-to-go of state x_k. Let the policy π be a function that returns a feasible decision $\pi(x_k) \in U(x_k)$ for every state x_k. $J_\pi(x_k)$ is the cost-to-go when the policy π is used to take decisions at every state, and it can be described as:

$$J_\pi(x_k) = \begin{cases} 0, \text{ if } x_k \in X_d \\ g\big(x_k, \pi(x_k)\big) + J_\pi\Big(f\big(x_k, \pi(x_k)\big)\Big), \quad \text{o.w.} \end{cases} \tag{6}$$

In this project, we used $\tilde{J}(x) = J_\pi(x)$, where π is the policy generated by solving the problem with a coarser discretization of the decision space. The strategy used to propagate constraints is included in Section Constraint Propagation and Feasible Decision Space. Section Graph Preprocessing describes how the graphs were preprocessed to reduce issues with short links. Section Analytical Solutions and Section Cost Lower Bound show, respectively, analytical solutions and cost lower bounds that can be used to speed-up the algorithm.

Constraint Propagation and Feasible Decision Space

Consider the following expression describes how the states are updated:

$$x_{i+1} = f(x_i, u_i),\ u_i \in U_i(x_i) \subset U_i \tag{7}$$

Let F_i represent the set of feasible states at node v_i. We define $U_i(x_i)$ as:

$$U_i(x_i) = \{u \in U_i |\ f(x_i, u) \in F_{i+1}\} \tag{8}$$

When choosing the decisions to test, we can either sample U_i and check the feasibility of each decision or calculate the feasible decision space with an inverse function $f^{-1}(F_{i+1}, x_i)$ that returns the elements of U_i that can generate a next state in F_{i+1}. In this case, F_{i+1} refers to the set of feasible next states, i.e., $\bigcup_{j,\,(v_i,v_j)\in A} F_j$. As most edges update the resources by adding its duration to the current resource, in general this operation consists of shifting the intervals representing the constraint for each resource, then taking the intersection between all of them,

e.g. if the next node has a time-window [10,15] and the current time is 5, then the decision duration must be in the interval [5,10] to be feasible. Different resources will generate different intervals, and feasible decisions must satisfy all of them.

Originally, F_i represents only the feasibility regarding the local constraints at node v_i, however, if we consider constraints from other nodes, we may be able to reduce F_i, and consequently reduce $U_i(x_i)$. Each node's local constraints can be propagated downstream and upstream to reduce other nodes' feasible spaces.

Forward Propagation

Let $\mathcal{F}^*\left(F_i,\ F_j, U_i(\cdot)\right)$ represent a function that returns which states in F_j can be reached from F_i, i.e.,

$$\mathcal{F}^*\left(F_i,\ F_j, U_i(\cdot)\right) = \left\{x_j \in F_j | \exists x_i \in F_i,\ \exists u \in U_i(x_i),\ f(x_i, u) = x_j\right\} \tag{9}$$

The set $\mathcal{R}_j^*$ of states that can be reached at node v_j is given by:

$$\mathcal{R}_j^* = \bigcup_{i,\,(v_i,v_j)\in A} \mathcal{F}^*\left(\mathcal{R}_i^*, F_j, U_i(\cdot)\right) \tag{10}$$

$\mathcal{R}_j^*$ can be overly complex due to the coupling between resources, so we try to approximate it by propagating the constraints for each resource separately. Let $F_i^{(r)}$ be the projection of F_i on the axis representing resource r, and $f^{(r)}$ the component of f that defines the evolution of resource r. Let $\mathcal{F}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right)$ be a function that returns which values of resource r can be reached at node v_j starting from a state in F_i, defined as follows:

$$\mathcal{F}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right) = \left\{x_j^{(r)} \in F_j^{(r)} | \exists x_i^{(r)} \in F_i^{(r)},\ \exists u \in U_i(x_i),\ f^{(r)}\left(x_i^{(r)}, u\right) = x_j^{(r)}\right\} \tag{11}$$

Let $\hat{\mathcal{R}}_j$ approximate $\mathcal{R}_j^*$ as follows:

$$\begin{aligned} \mathcal{R}_{i,j} &= \prod_r \left(\mathcal{F}^{(r)}\left(\hat{\mathcal{R}}_i,\ F_j, U_i(\cdot)\right)\right) \\ \hat{\mathcal{R}}_j &= \prod_r \bigcup_{j,\,(v_i,v_j)\in A} \mathcal{R}_{i,j}^{(r)} \end{aligned} \tag{12}$$

where $\mathcal{R}_{i,j}$ is the approximation accounting only for the constraints of upstream node v_i, and $\mathcal{R}_{i,j}{}^{(r)}$ is its projection on the axis representing resource r. Note that, for a given v_i , if $\exists r$ such that $\mathcal{F}^{(r)}\left(\hat{\mathcal{R}}_i, F_j, U_i(\cdot)\right) = \emptyset$, then $\mathcal{R}_{i,j} = \emptyset$. That is, if states from v_i cannot satisfy the constraints for 1 or more resources, then v_i will not be counted when calculating the reachable states at v_j. Furthermore, the edge (v_i, v_j) can be removed from the problem. At the origin node we have that $\mathcal{R}_0^* = \hat{\mathcal{R}}_0$, where $\hat{\mathcal{R}}_0$ is the set of possible initial states.

Backward Propagation

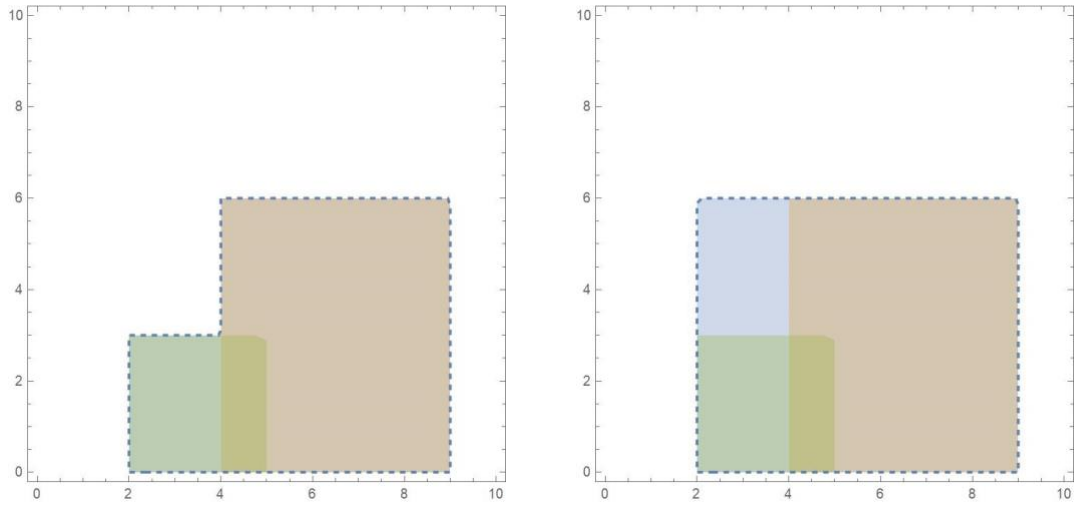
Backward propagation follows the same general idea as forward propagation. Let $\bar{F}_i$ be the reduced feasible space, i.e. the set of states $x_i \in F_i$ able to generate feasible solutions given downstream constraints. Like how we calculated $U_i(x_i)$, we need a function $\mathcal{B}\left(F_i,\ F_j, U_i(\cdot)\right)$ that can calculate the values of x_i at node v_i that can lead to at least one feasible state x_j at one of the successors v_j , i.e,

$$\mathcal{B}\left(F_i, F_j, U_i(\cdot)\right) = \left\{x_i \in F_i | \exists u \in U_i(x_i),\ f(x_i, u) \in F_j\right\} \tag{13}$$

However, this function is hard to compute and generates complex regions that will require more space to store, and more time to check during execution. Therefore, we calculate separate regions for each resource and use it to generate an approximate reduced feasible state space $\hat{F}_i$ as follows:

$$\begin{aligned} \mathcal{C}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right) &= \left\{x_i^{(r)} \in F_i^{(r)} | \exists u \in U_i(x_i),\ f^{(r)}\left(x_i^{(r)}, u\right) \in F_j^{(r)}\right\} \\ \hat{F}_{i,j} &= \prod_r \left(\mathcal{C}^{(r)}\left(F_i,\ \hat{F}_j, U_i(\cdot)\right)\right) \\ \hat{F}_i &= \prod_r \bigcup_{j,\,(v_i,v_j)\in A} F_{i,j}^{(r)} \end{aligned} \tag{14}$$

As in $\mathcal{R}_{i,j}$, $\hat{F}_{i,j}$ is the empty set if any resource constraint cannot be satisfied. In this case, the edge (v_i, v_j) can be removed from the graph as it cannot appear in a feasible solution. We do the forward propagation before the backward, so, at the destination node v_n, we have that $\bar{F}_n = \hat{F}_n = \hat{\mathcal{R}}_n$. For example, if a node v_j has a time-window $[10,15]$ and the edge (i,j) can have a duration in the interval $[2,5]$, then v_i must be visited in the time-window $[10-5,15-2] = [5,13]$. If a different edge (v_i, v_k) generated a propagated time-window of $[7,17]$ on v_i, we would consider the union of both time-windows, i.e., $[5,17]$. Then we would take the intersection of v_i's original time-window, say $[0,15]$, and the time-windows obtained from propagating downstream constraints to obtain an estimated feasible time-window of $[5,17]$. Note that the interval $[5,15]$ can be divided into an interval feasible for paths passing through v_i, $[5,13]$, and one feasible for paths through v_k, $[7,15]$. The same can happen to other resource constraints. Therefore, it is possible that a state in $\hat{F}_i$ only satisfies the time-window for a certain path but satisfies the HOS resource constraints only for a different path. As all constraints are satisfied by some path, the state is included in $\hat{F}_i$, but, in practice, that state cannot generate feasible successors. So, we have that $\hat{F}_i$ might contain states that cannot satisfy downstream solutions, i.e., $\bar{F}_i \subseteq \hat{F}_i \subseteq F_i$. Figure 2 shows a 2D example of the difference between reduced feasible state space $\bar{F}_i$ and its approximation $\hat{F}_i$. The blue region in Figure 2b belongs to $\hat{F}_i$, but not to $\bar{F}_i$.



(a) Blue dashed line: correct feasible space. (b) Blue dashed line: approximate feasible space. Blue region: infeasible.

Figure 2. The green and brown regions are examples of possible feasible regions in a 2D space. The figures show how the (a) exact and (b) approximate feasible spaces are calculated.

Propagating the resources

We separate the resource extension functions according to how they affect the resource being updated. The resource extension functions either add a value to the resource (ADD), maintain the current resource value (NoEff), or set the resource value to 0 (RESET). Let $e = (v_i, v_j)$ be an edge, $[\delta_e^-, \delta_e^+]$ be edge e's possible durations defined in U_i. In the case of static networks, we use U_i directly instead of $U_i(x)$. Let $[\eta_i^-, \eta_i^+]$ be the feasible values for resource r at node v_i. The approximate propagation functions described previously are defined as follows for the 3 types of REF:

Forward Propagation

ADD: $\mathcal{F}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right) = [\eta_i^- + \delta_e^-, \eta_i^+ + \delta_e^+] \cap [\eta_j^-, \eta_j^+]$

NoEff: $\mathcal{F}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right) = [\eta_i^-, \eta_i^+] \cap [\eta_j^-, \eta_j^+]$

RESET: $\mathcal{F}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right) = \{0\} \cap [\eta_j^-, \eta_j^+]$

Backward Propagation

ADD: $\mathcal{C}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right) = [\eta_j^- - \delta_e^+, \eta_j^+ - \delta_e^-] \cap [\eta_i^-, \eta_i^+]$

NoEff: $\mathcal{C}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right) = [\eta_i^-, \eta_i^+] \cap [\eta_j^-, \eta_j^+]$

RESET: $\mathcal{C}^{(r)}\left(F_i,\ F_j, U_i(\cdot)\right) = \{0\} \cap [\eta_j^-, \eta_j^+]$

When a resource's feasible range is a set of disjoint intervals, the functions above can be applied to each interval separately and we take the union of the resulting sets. Note that, in these REFs, the decision's duration is directly used to update the resource values. When energy/fuel consumption is included as a resource, the update value will be a function of the duration, so the propagation function will depend on the consumption model used.

Reduced Decision Space

The reduced decision space is generated following the same idea.

$$\bar{\mathrm{U}}_{\mathrm{i,j}}\left(\mathrm{x_i}, \mathrm{U_i}(\cdot), \hat{\mathrm{F}}_{\mathrm{j}}\right) = \left\{\mathrm{u} \in \mathrm{U_i}(\mathrm{x_i}) \middle| \mathrm{f}(\mathrm{x_i}, \mathrm{u}) \in \hat{\mathrm{F}}_{\mathrm{j}}\right\} = \bigcap_{\mathrm{r}} \left\{\mathrm{u} \in \mathrm{U_i}(\mathrm{x_i}) \middle| \mathrm{f}^{(\mathrm{r})}(\mathrm{x_i}, \mathrm{u}) \in \hat{\mathrm{F}}_{\mathrm{j}}^{(\mathrm{r})}\right\} \quad (15)$$

$$\bar{\mathrm{U}}_{\mathrm{i}}\left(\mathrm{x_i}, \mathrm{U_i}(\cdot), \left\{\hat{\mathrm{F}}_{\mathrm{j}}\right\}\right) = \bigcup_{\mathrm{j},\, (\mathrm{v_i}, \mathrm{v_j}) \in \mathrm{A}'} \bar{\mathrm{U}}_{\mathrm{i,j}}\left(\mathrm{x_i}, \mathrm{U_i}(\cdot), \hat{\mathrm{F}}_{\mathrm{j}}\right) \quad (16)$$

Let η_i be the current value of resource r. The other symbols are defined as in the previous section.

ADD: $\bar{U}_{i,j}^{(r)}\left(x_i, U_i(\cdot), \hat{F}_j\right) = \left\{u \in U_i(x_i) | f^{(r)}(x_i, u) \in \hat{F}_j^{(r)}\right\} = \left[\eta_j^- - \eta_i, \eta_j^+ - \eta_i\right] \cap [\delta_e^-, \delta_e^+]$

NoEff: $\bar{U}_{i,j}^{(r)}\left(x_i, U_i(\cdot), \hat{F}_j\right) = \begin{cases} \emptyset, if\ \eta_i \notin \hat{F}_j^{(r)} \\ [\delta_e^-, \delta_e^+],\ o.w. \end{cases}$

RESET: $\bar{U}_{i,j}^{(r)}\left(x_i, U_i(\cdot), \hat{F}_j\right) = \begin{cases} \emptyset, if\ 0 \notin \hat{F}_j^{(r)} \\ [\delta_e^-, \delta_e^+],\ o.w. \end{cases}$

Analytical Solutions

At nodes where the only possible next stop is the destination it is possible to analytically define the best decision so that the algorithm does not need to search over the remainder of that search tree branch. Naturally, the decision depends on the cost function and constraints being considered in the problem. Although we focus on diesel trucks, the problem formulation is very similar to the one needed to model battery electric trucks (BET). Therefore, this section considers the more general case used for BETs. This case can also be used if a limit is imposed on fuel consumption or emissions (set recharge rates to zero, and use emissions limit as battery capacity), or if fuel capacity and refueling time need to be considered (non-zero recharge rates, treating fuel capacity as battery capacity). The RCSPP formulation and REFs for BETs can be found in (Vital & Ioannou, 2021b). The solutions for diesel trucks without emissions limits or refueling concerns can be obtained by ignoring the battery constraint.

Consider the following cost function for a decision of duration δ:

$$C(\delta) = \begin{cases} \alpha\delta + \beta\mu_e\zeta(\mu_e/\delta), & if\ driving \\ (\alpha + \beta\gamma + \theta)\delta, & o.w. \end{cases} \quad (17)$$

, where α is the trucks hourly operational cost (excluding fuel/energy) and β is the cost per unit of fuel/energy. For non-driving decisions, γ is the hourly idling fuel/energy consumption, θ represents hourly costs incurred while stopped from sources other than idle energy consumption and operational costs. For driving decisions, μ_e is the length of the road segment considered, and $\zeta(v)$ is the fuel/energy consumption per unit of distance. This cost function considers both time and energy/fuel related costs, and their relative importance can be adjusted using the parameters α, β, and θ. In this section, we study the optimal decisions for the last driving and rest extension decisions.

Last driving decision

$$\frac{dC}{d\delta} = \alpha + \beta\mu_e \frac{d\zeta(v)}{d\delta} = \alpha + \beta\mu_e \frac{d\zeta(v)}{dv}\frac{dv}{d\delta} = \alpha - \beta\mu_e \frac{d\zeta(v)}{dv}\frac{\mu_e}{\delta^2} = \alpha - \beta v^2 \frac{d\zeta(v)}{dv} = 0 \quad (18)$$

Cost is minimum for $\tilde{\delta} = \frac{\mu_e}{\tilde{v}}$, such that $\tilde{v}$ is the root of $v^2 \frac{d\zeta(v)}{dv} = \frac{\alpha}{\beta}$. Assuming that $\zeta(v)$ is a convex function, and, consequently, $\frac{d\zeta}{dv}$ is monotonically non-decreasing, we can say that $v^2 \frac{d\zeta(v)}{dv}$ is strictly increasing over $(\max(0, v'), \infty)$, where v' satifies $\frac{d\zeta(v')}{dv} = 0$. As α and β are positive, $\tilde{v}$ is unique. The function $v^2 \frac{d\zeta(v)}{dv}$ does not depend on the edge, so $\tilde{v}$ can be calculated beforehand. Let $\left[\underline{\delta}, \overline{\delta}\right]$ be δ's domain, the optimal decision is given by:

$$\delta = \begin{cases} \underline{\delta}, & if \quad \tilde{\delta} < \underline{\delta} \\ \overline{\delta}, & if \quad \tilde{\delta} > \overline{\delta} \\ \tilde{\delta}, & o.w. \end{cases} \quad (19)$$

Last rest extension

Let ρ be the recharge rate at the current location, B_0 the battery charge on arrival at the current node, $\delta_0 \in \left[\underline{\delta_0}, \overline{\delta_0}\right]$ the rest extension to be chosen, and $\delta_\ell \in \left[\underline{\delta_\ell}, \overline{\delta_\ell}\right]$ the duration of the decision at the following edge, which is the last driving edge. The cost from the rest node to the destination can be written as $C(\delta_0, \delta_\ell) = (\alpha + \beta\gamma + \theta)\delta_0 + \alpha\delta_\ell + \beta\mu_e\zeta(\mu_e/\delta_\ell)$. Assume that, due to the destination node's resource constraints and the current state's resource values, $\delta_0 + \delta_\ell \in \left[\underline{D}, \overline{D}\right]$. The optimization problem being solved at the last rest decision can be described as:

$$\min_{\delta_0, \delta_l} C(\delta_0, \delta_l) = (\alpha + \beta\gamma + \theta)\delta_0 + \alpha\delta_l + \beta\mu_e\zeta(\mu_e/\delta_l) \quad (20)$$

s.t.:

$$\mu_e\zeta(\mu_e/\delta_l) - \delta_0\rho - B_0 \leq 0 \quad (21)$$

$$\underline{D} \leq \delta_0 + \delta_l \leq \overline{D} \quad (22)$$

$$\underline{\delta_0} \leq \delta_0 \leq \overline{\delta_0} \quad (23)$$

$$\underline{\delta}_l \leq \delta_l \leq \overline{\delta}_l \tag{24}$$

, where (21) guarantees that the battery charge is non-negative when arriving at the destination. (22) restricts the time to reach the destination, and can be related to both HOS and time-window constraints. (23) and (24) restrict the domains of δ_0 and δ_ℓ to the reduced decision space, which is affected by all constraints and the current state. Consider the following definitions:

$$\begin{aligned}
H(v) = &\quad v^2 \frac{d\zeta(v)}{dv} \\
P(v) = &\quad \frac{\mu_e \zeta(v) - B_0}{\rho} \\
\nabla C = &\quad \begin{bmatrix}(\alpha + \beta\gamma + \theta) & (\alpha - \beta H(v))\end{bmatrix} \\
\nabla g_1 = &\quad \begin{bmatrix}(-\rho) & (-H(v))\end{bmatrix} \\
\tilde{v}, &\quad H(\tilde{v}) = \frac{\alpha}{\beta} \\
\hat{v}, &\quad H(\hat{v}) = -\frac{\theta}{\beta} - \gamma \\
\check{v}, &\quad H(\check{v}) = \rho \\
\breve{v}, &\quad H(\breve{v}) = 0 \\
v^*, &\quad H(v^*) = \frac{\alpha\rho}{\alpha + \beta(\rho + \gamma) + \theta}
\end{aligned}$$

, where g_1 represents constraint (21). $H(v)$ and $P(v)$ are auxiliary functions defined to simplify the notation and represent, respectively, the derivative of the energy consumption with respect to δ_ℓ and the minimum feasible δ_0 given δ_ℓ. The v's with different accents are values used in the solution that can be calculated offline. $\tilde{v}$, $\hat{v}$, and v^* represent, respectively, the speeds at which the cost gradient ∇C is perpendicular to (23), (22), and (21). $\check{v}$ and $\breve{v}$ are the speeds at which (21) is parallel to (22) and (23), respectively. Note that, given a distance μ_e, each v also defines a duration δ_ℓ, e.g. $\hat{\delta}_\ell = \frac{\mu_e}{\hat{v}}$. The accents on the δ's indicate which v generate them. First, consider the case when (21) is not active (e.g., diesel trucks). The optimum point is given by:

$$\delta_0 = \begin{cases} \underline{\delta}_0, & if \quad \underline{\delta}_0 + \underline{\delta}_\ell \geq \underline{D} \\ \min\left(\overline{\delta}_0, \underline{D} - \underline{\delta}_\ell, \max\left(\underline{\delta}_0,\ \underline{D} - \hat{\delta}_l,\ \underline{D} - \overline{\delta}_\ell\right)\right), & o.w. \end{cases}$$

$$\delta_\ell = \begin{cases} \underline{\delta}_\ell, & if \quad \underline{\delta}_0 + \underline{\delta}_\ell \geq \underline{D} \\ \underline{D} - \delta_0, & if \left(\underline{\delta}_0 + \underline{\delta}_\ell < \underline{D}\right) \wedge \left(\delta_0 \neq \underline{\delta}_0\right) \\ \min\left(\overline{\delta}_\ell,\ \max\left(\underline{\delta}_\ell,\ \tilde{\delta}_l,\ \underline{D} - \underline{\delta}_0\right)\right), & o.w. \end{cases}$$

If the point (δ_0, δ_ℓ) satisfies (21), then it is optimal. Otherwise, it means that (21) must be active. In this case, we can define 7 candidate points and the sufficient conditions for them to be the optimum. The candidate points are given by the point along g_1 with minimum cost and

the points where (21) intersects other constraints, and the conditions conditions are derived from each point's KKT conditions. Table 3 presents the candidate solutions and their conditions. Feasibility is a basic necessary condition for any solution, and was thus omitted from the table. $P^{-1}(v)$ refers to the inverse of $P(v)$ over the domain $v \in [\breve{v}, \infty)$. The points x_2 and x_3, representing the candidates where (21) and one of the constraints forming (22) intersect, might be computationally expensive to calculate, so we can leave testing them for last. We can also use approximate solutions instead of solving it exactly. Note that the conditions are generated from speeds that can be calculated beforehand. Therefore, we may be able to directly eliminate some candidate solutions based on δ_ℓ's domain.

Table 3. Solution Candidates

Point	Condition
$x_1 = (P(v^*), \delta_\ell^*)$	-
$x_2 = \left(\overline{D} - \delta_2, \delta_2\right),\ P(\mu_e/\delta_2) = \overline{D} - \delta_2$	$\check{\delta}_\ell < \delta_2 \le \delta_\ell^*$
$x_3 = \left(\underline{D} - \delta_3, \delta_3\right),\ P(\mu_e/\delta_3) = \underline{D} - \delta_3$	$\delta_\ell^* \le \delta_3 \le \hat{\delta}_\ell$
$x_4 = \left(\underline{\delta}_0, P^{-1}\left(\underline{\delta}_0\right)\right)$	$\tilde{\delta}_\ell \le P^{-1}\left(\underline{\delta}_0\right) \le \delta_\ell^*$
$x_5 = \left(\overline{\delta}_0, P^{-1}\left(\overline{\delta}_0\right)\right)$	$\delta_\ell^* \le P^{-1}\left(\overline{\delta}_0\right) \le \check{\delta}_\ell$
$x_6 = \left(P\left(\mu_e/\underline{\delta}_\ell\right), \underline{\delta}_\ell\right)$	$\underline{\delta}_\ell \ge \delta_\ell^*$
$x_7 = \left(P\left(\mu_e/\overline{\delta}_\ell\right), \overline{\delta}_\ell\right)$	$\overline{\delta}_\ell \le \delta_\ell^*$

Cost Lower Bound

Let $A^d \subset A'$ represent the set of all arcs with driving as their assigned activity. For every node pair (p, q) such that there is a directed path from p to q , let $\mathcal{D}(p,q)$, $\mathcal{D}_d(p,q)$, and $\mathcal{D}_\ell(p,q)$ be, respectively, the minimum travel time (including service time), minimum driving time and minimum travel distance between nodes p and q with all resource, time-window and HOS constraints relaxed:

$$\mathcal{D}(p,q) = \begin{cases} \min\left(\Delta_{pq}\right), & if\,(p,q) \in A' \\ \min\limits_{(p,k)\in A'} \left(\min\left(\Delta_{pk}\right) + \mathcal{D}(k,q)\right), & o.w. \end{cases}$$

$$\mathcal{D}_d(p,q) = \begin{cases} 0, & if\,(p,q) \in A' \setminus A^d \\ \min\left(\Delta_{pq}\right), & if\,(p,q) \in A^d \\ \min\limits_{(p,k)\in A'} \left(\mathcal{D}_d(p,k) + \mathcal{D}_d(k,q)\right), & o.w. \end{cases}$$

$$\mathcal{D}_\ell(p,q) = \begin{cases} 0, & if\,(p,q) \in A' \setminus A^d \\ \mu_{pq}, & if\,(p,q) \in A^d \\ \min\limits_{(p,k)\in A'} \left(D_\ell(p,k) + D_\ell(k,q)\right), & o.w. \end{cases}$$

If there is no directed path from p to q , then $\mathcal{D}(p,q) = \mathcal{D}_d(p,q) = \mathcal{D}_\ell(p,q) = \infty$.

Let $D_{HOS}(d,\psi)$ represent the minimum duration of a HOS-compliant trip with d driving hours and initial resource vector ψ, assuming the driver can rest anywhere, and without considering service time and time-window constraints, i.e. if a driver were at the beginning of an empty straight road with length equivalent to d driving hours where he/she can rest anywhere, given an initial resource vector ψ, how long would he/she take to reach the end of the road without breaking the HOS regulations. A method to calculate $D_{HOS}(d,\psi)$ is described in (Vital & Ioannou, 2021a). Let $\mathcal{D}_s(p,q)$ be the service time required between nodes p and q. If the objective were simply to minimize trip duration, the lower bound $\mathcal{L}_{dur}$ can be calculated as:

$$\mathcal{L}_{dur}(p,q,\psi) = D_{HOS}(\mathcal{D}_d(p,q),\psi) + \mathcal{D}_s(p,q)$$

However, when considering a combination of trip duration and energy/fuel consumption or emissions as the objective function, the lower bound generated using only the duration term ($\alpha\mathcal{L}_{dur}(p,q,\psi)$) is too loose and not as useful. Therefore, we need a lower bound on the fuel consumption/emissions.

Bound 1

Idling cost: Let γ the energy/fuel consumption rate when idle (resting or service). A lower bound on the idling cost is given by:

$$\mathcal{L}_{idl1}(p,q,\psi) = (\beta\gamma + \theta)\big(D_{HOS}(\mathcal{D}_d(p,q),\psi) - \mathcal{D}_d(p,q) + \mathcal{D}_s(p,q)\big) \qquad (25)$$

$\mathcal{D}_s$ is fixed as client visits are mandatory. $\mathcal{D}_d$ considers the minimum driving time of each edge, and $D_{HOS}(d,\psi) - d$ is monotonically increasing in d (required rest time cannot decrease when driving time increases), so $\mathcal{L}_{idl1}$ is a lower bound on idling cost. Note that if the cost/consumption parameters for rest and service time are different, the term $\mathcal{D}_s(p,q)$ will appear separately multiplying its own parameter.

Driving consumption: Let v_{min} be the minimum travel speed allowed in the network. We assume that the fuel consumption per time $FC(v)$ is monotonically increasing in the range of speeds used in the problem, as is the case for the model we use. Therefore, $FC(v_{min})$ gives a lower bound on the energy/fuel consumption rate when driving. A lower bound on the consumption due to driving is given by:

$$\mathcal{L}_{f\_dr1}(p,q) = 3600 \cdot FC(v_{min})\mathcal{D}_d(p,q)$$

An alternative is using the minimum travel distance $\mathcal{D}_\ell(p,q)$ and the speed v_ℓ that minimizes the fuel consumption per distance, $\zeta(v)$, (or the nearest feasible speed) to generate a energy/fuel consumption lower bound.

Cost: Consider the cost function defined in (1). A cost lower bound is given by:

$$\mathcal{L}_{cost1}(p,q,\psi) = \alpha\mathcal{L}_{dur}(p,q,\psi) + \beta\mathcal{L}_{f\_dr1}(p,q) + \mathcal{L}_{idl1}(p,q,\psi)$$

Note that the driving energy/fuel consumption bound is calculated using the minimum travel speed, whereas the idling cost and trip duration bounds are calculated using the maximum travel speed. Therefore, this bound is not tight.

Bound 2

When calculating analytical solutions in Last driving decision, we showed how to calculate the optimal speed based on energy/fuel and duration costs, and consumption model. We now use this information to refine the lower bound.

Driving time: Bound 1 used a driving time considering the maximum travel speed. However, depending on the cost function, the cost increase due to fuel consumption at higher speeds may exceed savings due to shorter trip duration. Optimal solutions are expected to tend towards using the optimal speed $\tilde{v}$ (limited by possible increases in required rest time). With this in mind, we scale the driving time so that it represents the travel time at the optimal speed (or the nearest feasible speed).

$$\begin{aligned} v_t = &\quad max(min(\tilde{v}, v_{max}), v_{min}) \\ \widetilde{\mathcal{D}}_d(p,q) = &\quad \mathcal{D}_d(p,q)\frac{v_{max}}{v_t} \end{aligned}$$

This scaling assumes that all edges have the same speed limits and optimum speed. An alternative (but still assuming that all edges have the same optimum speed) would be to use the length of the minimum length path, $\mathcal{D}_\ell(p,q)$, to estimate a lower bound on the driving cost when traveling with speed v_t. A more general approach would be to, when building the graph, calculate v_t for each edge, and store in each edge the travel time and cost associated with v_t. The stored costs can be used to calculate a minimum cost path and its driving time. In both alternatives, the minimum cost (we refer to it as $\mathcal{L}_{dr\_cost}(p,q)$) can be used as a lower bound on the driving related costs (due to both emissions and duration) and we would require only to complement it with a lower bound on the idling costs (due to both emissions and duration).

It is important to remember that, due to HOS regulations, increasing driving time may end up increasing required rests. The extra rest time caused by driving time scaling is given by:

$$\Lambda = D_{HOS}\big(\widetilde{\mathcal{D}}_d(p,q), \psi\big) - \widetilde{\mathcal{D}}_d(p,q) - \big(D_{HOS}(\mathcal{D}_d(p,q), \psi) - \mathcal{D}_d(p,q)\big)$$

Trip duration and fuel consumption are calculated following the same ideas as Bound 1 but using the scaled driving time and correcting trip duration and idling time to remove the extra rest time.

Trip Duration: The trip duration is calculated as follows:

$$\mathcal{L}_{dur2}(p,q,\psi) = D_{HOS}\big(\widetilde{\mathcal{D}}_d(p,q), \psi\big) - \Lambda + \mathcal{D}_s(p,q)$$

Idling cost: The idling cost lower bound is given by the same expression as (25) due to the rest time correction, i.e.,

$$\mathcal{L}_{idl2}(p,q,\psi) = \mathcal{L}_{idl1}(p,q,\psi).$$

Driving consumption: Energy/fuel consumption due to driving is given by:

$$\mathcal{L}_{f\_dr2}(p,q) = 3600 \cdot FC(v_t)\widetilde{\mathcal{D}}_d(p,q)$$

Cost: A cost lower bound is given by:

$$\mathcal{L}_{cost2}(p,q,\psi) = \alpha\mathcal{L}_{dur2}(p,q,\psi) + \beta\mathcal{L}_{f\_dr2}(p,q) + \mathcal{L}_{idl2}(p,q,\psi)$$

Note that while $\mathcal{L}_{dur2}(p,q,\psi) \geq \mathcal{L}_{dur}(p,q,\psi)$ and $\mathcal{L}_{f\_dr2}(p,q) \geq \mathcal{L}_{f\_dr1}(p,q)$, $\mathcal{L}_{dur2}$ and $\mathcal{L}_{f\_dr2}$ are consistent with respect to the travel speed used for their calculation, and use a speed that minimizes cost (not accounting for mandatory rests). As the rest (idling) time is kept as the one from the minimum duration path, the rest time is the minimum feasible. Decreasing $\mathcal{L}_{dur2}$ would imply that one or more edges are using a speed greater than the optimal, causing an increase in fuel consumption costs that exceeds the savings in trip duration costs. Similarly, decreasing $\mathcal{L}_{f\_dr2}$, would cause an increase in trip duration costs, and increase overall cost. Therefore, $\mathcal{L}_{cost2}$ is a lower bound. Each term is not a lower bound for the value it approximates, but they are calculated so that they generate a cost lower bound. If the driving cost lower bound $\mathcal{L}_{dr\_cost}(p,q)$ is calculated directly, then the cost lower bound is given by:

$$\mathcal{L}_{cost2}(p,q,\psi) = \mathcal{L}_{dr\_cost}(p,q) + (\alpha + \gamma\beta + \theta)\underbrace{\left(D_{HOS}(\mathcal{D}_d(p,q),\psi) - \mathcal{D}_d(p,q) + \mathcal{D}_s(p,q)\right)}_{idling\ time}$$

Graph Preprocessing

In the approximate dynamic programming algorithm used, we store the decision and cost for several states at each node. Therefore, having a large number of intermediate nodes between rest areas increases both the number of decisions needed to reach the destination and the storage space required by the algorithm. Furthermore, when optimizing travel speed to reduce fuel consumption, the precision with which speed can be adjusted depends on the time resolution used in the decision space, but also on the length of any given edge. If an edge is too short, any change in duration might generate a travel speed outside of the allowed range. In order to reduce the number of nodes in the graph, we use a stop-based graph based on the road network and remove short edges between nearby rest areas (e.g., only consider rest areas that are at least 2h away from the current node). By stop-based graph we mean a graph that directly links possible stop locations (origin, rest areas, clients), analogous to customer-based graphs used for vehicle routing problems. However, the graph is not complete as each location is connected only to locations that were downstream in the original road network. As clients are mandatory stops and have a fixed order, nodes are not directly connected to nodes downstream of the next client. It can be seen as generating the stop-based graph based on the subnetworks connecting each pair of consecutive clients, as opposed to using the whole network directly. Figure 3 shows a graph representing a road network, whereas Figure 4 shows the stop-based graph that would be generated from that network. As our experiments set the same speed profile for all edges, each edge (i,j) of the stop-based graph was generated using the length of the minimum distance path between nodes i and j in the road network and setting the same speed profile used in the road network. We assume that a stop-based graph is known or can be obtained by the user, and do not cover the specifics of its construction for general

networks. Algorithms to construct customer-based graphs for time-dependent road networks were proposed in (Ben Ticha et al., 2021).

Given a stop-based graph, we remove edges that have distance or minimum travel time shorter than chosen limits, except when one of the edge's nodes is a client, the origin, or the destination. In our experiments, the time and distance limits were set to 2h and 100km, respectively. In addition, as HOS regulations limit driving time, edges with minimum travel time greater than 8h were also removed. Although it is possible for the fastest path between locations to vary with time in time-dependent networks, we assume that edge lengths (distance) are fixed in the stop-based graph.

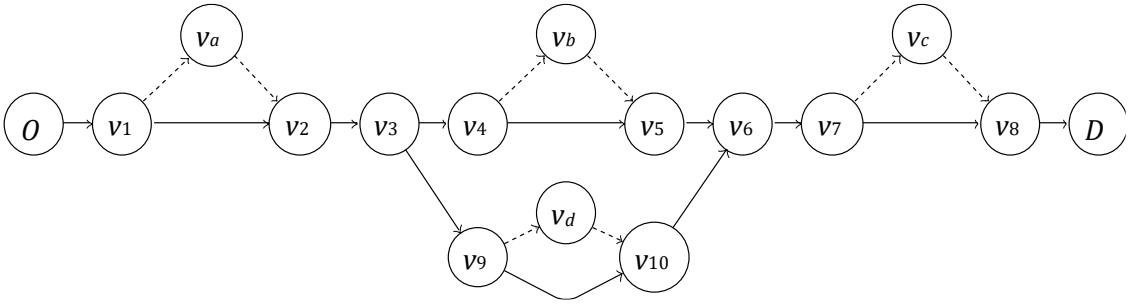


Figure 3. Example graph focusing on the road network. Focuses on rest area (nodes with letter indexes) placement along main roads. Easy to visualize but has a large number of intermediate nodes (nodes with number indexes).

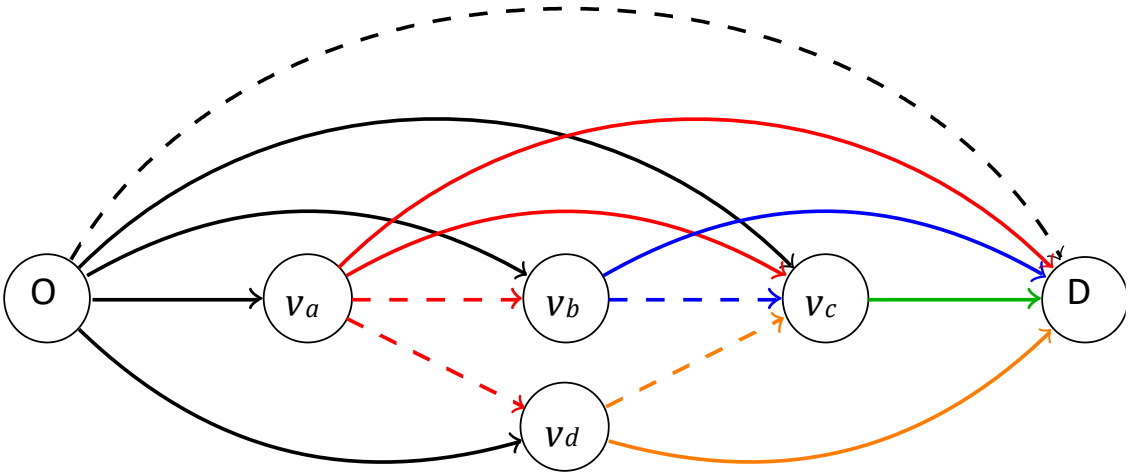


Figure 4. Stop-based graph generated from Figure 3 to focus on the connection between possible stops (rest areas, clients, origin, destination). Each possible stop is directly connected to downstream stops satisfying predetermined conditions. Dashed arrows exemplify edges that could be removed for being too short or too long.

Modifications for time-dependent networks

The dynamic programming formulation presented in section Dynamic Programming Formulation and Rollout Algorithm already represents the decision space as a function of the state ($U(x_k)$), so it is general enough to represent the time-dependent case and does not need any modification. However, the analytical solutions, lower bounds and the methods presented for constraint propagation are described for time-independent networks and need some clarification.

- **Constraint Propagation:** Although travel time affects all resources, due to the FIFO assumption, we focus the modifications on the time resource. Let the function $a_{ij}(\eta_i)$ represent the arrival time at node j when departing node i at time η_i, and we use an under-bar to indicate when the minimum speed is being considered, and an over-bar to indicate when the maximum speed is being considered. During forward propagation, we replace $[\eta_i^- + \delta_e^-, \eta_i^+ + \delta_e^+]$ by $\left[\underline{a}_{ij}(\eta_i^-), \overline{a}_{ij}(\eta_i^+)\right]$. During backward propagation, we replace $[\eta_i^- - \delta_e^+, \eta_i^+ - \delta_e^-]$ by $\left[\underline{a}_{ij}^{-1}(\eta_i^-), \overline{a}_{ij}^{-1}(\eta_i^+)\right]$, the superscript ` -1 ' refers to the inverse function. As we assume that the speed profiles satisfy FIFO assumptions, both $\underline{a}_{ij}$ and $\overline{a}_{ij}$ are strictly increasing and have unique inverses. For the other (non time) resources, we simply take δ_e^- and δ_e^+ as the lower and upper bounds for the travel time at any time instant. Another possibility is to use $[\eta_i^-, \eta_i^+]$ to calculate the range of possible travel times for these departure/arrival times. However, this method can only provide a better range if the interval $[\eta_i^-, \eta_i^+]$ is narrow and does not span a wide range of possible travel times.
- **Analytical Solutions:** The analytical solution for the last driving decision, described in section Last driving decision, is not affected by the time-dependent travel time as the departure time is fixed and known. The last rest extension decision is affected by the change in travel time, so the one described in section Last rest extension is not valid anymore. In this case, we did not calculate a new solution, and opted to stop using analytical solutions for those cases.
- **Cost Lower Bound:** Similar to the case of HOS constraints propagation, we use the upper and lower speed bounds over the whole planning period to calculate the bounds described in section Cost Lower Bound.

Parking Availability Uncertainty

The model presented in section Problem Description considered that parking availability could be predicted with certainty. However, in practice, there is a certain level of uncertainty in any prediction, and the longer the prediction horizons the less certain we can be about any prediction. Therefore, we now model parking availability in a probabilistic way. The previous formulation represented parking availability as time-windows at each rest area. Two possible ways of extending this formulation are:

- **Stochastic Time-Windows:** we assume that there is a continuous time interval within which parking is guaranteed, but we are unsure of the exact start and end times. The

time-window's start and end times are given by random variables with known distribution. Parking availability is defined indirectly, depending on whether the arrival time falls within that interval or not. So, we would need to consider the probability of arriving after the start of the time-window, but before its end. The deterministic model could be seen as an approximation using the expected values of the time-window's limits, or values that satisfy some confidence level. This model would ignore the small occupancy variations that can occur. For example, overnight, most of the parking spaces are taken by long-term parking. However, unless ALL parking spaces are used for long-term parking, there will still be some trucks leaving on occasion.

- **Stochastic Parking Availability:** directly model parking availability as a random variable with a time-dependent probability distribution, i.e., at any time t, there is a probability $p_i(t)$ of rest area i having an available parking space. In this approach, the probability of finding parking can be calculated directly, without worrying about how the distributions of time-windows' limits interact. The small variations that occur even at high occupancy periods can be modeled by a very small, but non-zero, probability of finding parking during that period. The deterministic model can be seen as time-windows defined by the intervals at which $p_i(t)$ exceeds a given threshold.

We take the second approach, modeling stochastic parking availability directly. A new binary component w, representing whether parking is available at the current location (Yes:1, No:0), is added to the state definition. This component can be used to control the actions available to drivers at rest areas, e.g., if parking is available ($w = 1$), the driver needs to choose for how long to rest, if the rest area is full, the driver needs to revise the trip plan and decide whether to search for nearby alternative parking locations or to continue driving. The new state contains the following information:

- Current node (v)
- Time when node was visited (η^0)
- Accumulated driving time since last break (η^b)
- Elapsed time since last daily rest (η^r)
- Accumulated driving time since last daily rest (ψ^r)
- Accumulated on-duty time since last weekly rest (ψ^w)
- Parking availability (w)

We define the update rule for w as $w_{i+1} = f^{(w)}(x_i, u_i, \omega) =$, where ω is a binary random variable characterized by a probability distribution $P(\cdot\,|x_i, u_i)$. As w aims to model the parking availability at rest areas, it is set to default values at other locations as needed. The dynamic programming formulation presented before can be updated as follows:

$$J^*(x) = \min_{u \in U(x)} E_\omega \{ g(x,u,\omega) + J^*\big(f(x,u,\omega)\big) \}$$
$$= \min_{u \in U(x)} \sum_{\omega=0}^{1} P(\omega|x,u) \Big(g(x,u,\omega) + J^*\big(f(x,u,\omega)\big) \Big) \quad (26)$$

It is important to note that as we are considering parking availability to be stochastic, it might be impossible to guarantee parking at all times. As long as the probability of finding parking at the visited locations is not 1, it is possible for a driver to try to park at every single location without success until the next location is too far to be reached without exceeding HOS constraints. Therefore, the model must include what happens in those cases. If the driver exceeds the HOS limits, the truck's monitoring equipment might automatically shut down the truck, and the driver would be stuck somewhere inconvenient for a while. The driver might face legal penalties, maybe a fine or license suspension. If the driver stops at a road shoulder or highway ramp, there is an associated risk of causing accidents or being fined. In any case, the model must consider that such scenarios are possible, what actions can be taken and what are their consequences/costs.

Recourse Actions

We consider two possible ways for a driver to react when unable to find parking at the current location: reroute and try to rest at a downstream facility, or look for an alternative parking location in the surrounding region. Essentially, the driver needs to decide whether it is feasible to stop later or if they need to stop right away. The deterministic model included three types of actions at rest areas, each one representing an off-duty period that resets the counter for a certain set of regulations. In the stochastic model, if parking is available, the same set of actions is used, but when parking is unavailable, we consider that all 3 rest actions are prohibited. Instead, we include the actions *search* and *exit*:

- **Exit:** represents the action of leaving the rest area and heading to the next location without resting, and is connected to the exit node of the rest area in the problem's graph representation.
- **Search:** represents the action of looking for an alternative parking option nearby, and it leads to the entrance node of an alternative parking location. The alternative parking location will behave the same way as a regular parking location, except by the fact that penalty costs will be incurred for its usage.

Figure 5 shows a diagram of the actions available at rest areas after including the recourse actions. The *search* action's duration can be used as a time penalty that forces drivers to adjust the rest of the trip, and it can have cost penalties included in it. Essentially, it is any action that will lead to having a location to rest without driving to another one of the facilities included in the graph. For the purpose of calculating fuel consumption, searching is treated as driving at a user-defined speed. In our experiments, we only apply penalties for the *search* action if its duration exceeds the driver's remaining allowed driving time. In this case, we assume that the driver would need to park at a location even worse than usual in order to avoid HOS violations, hence incurring some extra penalties (both a fixed penalty and one proportional to the excess

duration). In the *alternative parking locations*, we do not include time penalties as that was already considered in the *search* edge, but they may have both fixed and variable costs assigned to them (on top of the usual time and fuel consumption costs). Note that we want to model the fact that drivers can react to the lack of parking and we use these generic actions to do so. Whether drivers will drive around for a while looking for parking and then park at a road shoulder, or will make use of some service to arrange for appropriate parking, depends on the options available in the region, drivers/companies preferences and the risks/costs involved with each option. One could even include multiple sets of alternative parking locations and recourse actions, e.g., one for looking for a road shoulder to park at, and another for using an expensive service that offers guaranteed parking or driver replacement.

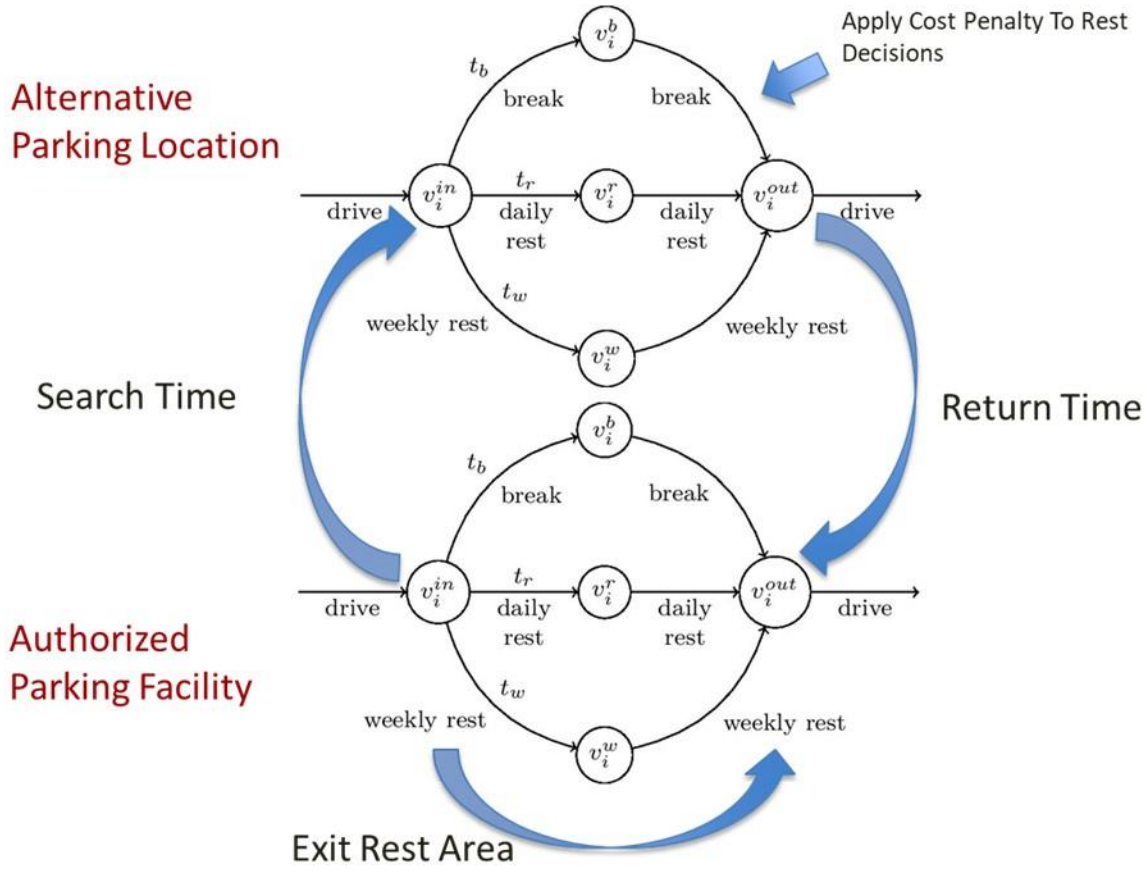


Figure 5. Subgraph representing the actions that can be taken at rest areas after inclusion of recourse actions and alternative parking locations.

Policy

Our objective is to give drivers and planners good recommendations about how to plan their trips. The policy obtained takes the current state of the system and outputs the decision that minimizes a certain cost (or whatever estimate we have of that cost). As can be seen in (26), in the stochastic case we optimize an expected value of the cost function, so the policy cost is a single value representing that expectation. However, it might be interesting for the user to

visualize more information about this cost. As shown in (6), to calculate the policy cost we need to simulate its effect on the system by recursively applying the policy. When doing this, we generate a decision tree describing how the system evolves under that policy given an initial state. The tree branches out at decisions leading to rest areas, as the state when arriving at the rest area depends on the random variable representing parking availability. As parking availability can only take binary values, this system is easier to simulate than systems where the random variables can take a large or infinite number of values (e.g., a continuous interval). Although that decision tree is readable only for very small problems, we can use it to calculate the probability distribution of any cost function, regardless of if it was actually considered during the policy generation or not. The decision tree gives us the possible paths and their probabilities, so if we know how to calculate the target information given the path, we can calculate the probability distribution of that information. For example, we can generate a policy that minimizes the estimated trip duration, then simulate that policy to find out how it affects the likelihood of using alternative parking, the probability distribution of emissions, etc. The time it takes to calculate that information depends on how fast the policy function can be evaluated and how many states need to be visited, so it will increase if the trip is longer or has many rest stops (rest stops cause branching).

Imperfect Information

Studying this question of simulating the cost/effects of a given policy raised another concern. Similar to how lookahead policies optimize an approximate cost, which is usually different from its own policy cost, policies may be calculated based on information or assumptions that do not match reality. For example, companies that disregard parking difficulties might generate policies based on the assumption that all rest areas are available 24/7. Even if that policy is calculated without using approximations and is the optimal policy for a scenario satisfying the assumption, it can show significantly different results when applied to the real world. Therefore, we think it is interesting to study how imperfect information affects different policies. This can give us some insight on the value of information and severity of cost misestimation.

For clarity, we redefine the policy used, differentiating between the model used by the planner and the ones used for simulation, where the models define the probability distributions, cost functions and state transition functions considered. Let $\pi_p(x_k)$ be a one-step lookahead policy calculated using a model p, and $J_{\pi_p,s}(x_k)$ be the cost of applying policy π_p when a model s is taken as the ``world model'' for simulation.

$$\pi_p(x_k) = argmin_{u \in U(x_k)} E_{\omega_p}\left\{g_p(x_k, u, \omega_p) + \tilde{J}\left(f_p(x_k, u, \omega_p)\right)\right\}$$

$$= argmin_{u \in U(x_k)} \sum_{\omega_p=0}^{1} P_p(\omega_p | x_k, u)\left(g_p(x_k, u, \omega_p) + \tilde{J}\left(f_p(x_k, u, \omega_p)\right)\right)$$

$$J_{\pi_p,s}(x_k) = \begin{cases} 0, \text{ if } x_k \in X_d \\ E_{\omega_s}\left\{g_s(x_k, \pi_p(x_k), \omega_s) + J_{\pi_p,s}\left(f_s(x_k, \pi_p(x_k), \omega_s)\right)\right\}, \text{o.w.} \end{cases}$$

As in the deterministic case, we use a rollout algorithm, so the approximate cost function is the policy cost of a base policy. Our base policy is the optimal policy over a coarsely discretized decision space. However, now it is important to emphasize that, as the simulation model might be unknown to the planner, the planning model is the one used to calculate the base policy's cost, i.e.:

$$\tilde{J}(x_k) = \begin{cases} 0, \text{ if } x_k \in X_d \\ \min_{u \in \tilde{U}(x_k)} E_{\omega_p}\left\{ g_p(x_k, u, \omega_p) + \tilde{J}\left(f_p(x_k, u, \omega_p) \right) \right\}, \text{o.w.} \end{cases}$$

, where $\tilde{U}(x_k)$ is the coarsely discretized decision space. This project's experiments consider the following cases:

- **Full Information (Scenario 0):** drivers have full access to parking availability probability distributions, and arrival time at rest areas is not restricted.
- **Time-window + Deterministic (Scenarios 1-3):** drivers assume that parking is guaranteed within certain time-windows generated from experience or data. As in the deterministic case, drivers can only arrive at rest areas within the given time-windows.
- **Time-window + Uncertainty (Scenarios 4-6):** drivers have access to the parking availability probability distribution within given time-windows. However, arrival at rest areas is still limited to within these time-windows.

Scenarios 1-3 were implemented while trying to limit modifications to the system's basic routines. They use a deterministic view of the world, so they were implemented by bypassing the usage of probability distributions when calculating the policies. Modifications to the current implementation of cost functions and policies are required to facilitate the decoupling between the information used during planning and the ``world model'' used for simulation/evaluation.

Experiments

The following sections present the results of experiments performed on static and time-dependent networks. The marginal operational cost per hour does not account for fuel costs and is set to \$54.77 (similar to \$54.71 found in (Murray & Glidewell, 2019)). The emission coefficient is set to 3.13 kg CO_2/L (0.44 from production (Argonne National Laboratory, 2020) and 2.69 from combustion (U.S. Energy Information Administration, 2016)), diesel prices were set to 1\$/L, and the emission costs are set to $18e^{-3}\$/kgCO_2$ (California Air Resources Board, 2021). The relative importance of reducing emissions is controlled by applying a penalty multiplier to the emission cost. The deterministic scenarios were also used to compare the effectiveness of the cost lower bounds presented in section Cost Lower Bound, and the results were included in section Stochastic Scenarios. We implemented our algorithm in Python 3.8, and all experiments were run on an Intel Core i7, 2.6 GHz CPU with 32 Gb of RAM. We would like to note that the obtained running times could be reduced by implementing the algorithm in faster languages, such as C, C++ or Java, however, this is not the focus of this work.

Static Network

The results presented were obtained using the lower bound described in section Bound 2. The rollout policy used a sampled decision space with 0.5h sampling interval, and the base policy is the optimal solution when using a sampled decision space with 5h sampling interval and stopping if a solution is within 5% of the cost lower bound.

Figure 6 presents the improvements obtained in CO_2 emissions when the penalty multiplier is increased under different parking availability conditions. We see up to 5-7% decrease in CO_2 emissions, with the smaller improvements happening in scenarios with narrow time-windows. Figure 7 and Figure 8 show the effects of different penalty multipliers on the average trip duration and nonpenalized trip cost, respectively. Although there are some outliers with more than 20% increase in trip duration or cost, the average increases in duration and cost are under 15% and 7%, respectively, in all scenarios. Figure 9 presents the average running time varies with the penalty multiplier. The instances placing a higher priority on emissions reductions show significantly longer running times. This is possibly due to the bounds on trip duration being tighter than the bounds on the multi-objective cost accounting for both duration and emissions. Nevertheless, as the penalty multiplier does not affect feasibility, solving faster instances first and using their solutions to generate upper bounds for slower instances might improve performance. This approach is tested in the experiments over time-dependent networks (section Time-Dependent Network). Table 4 to Table 7 present the results for each network separately.

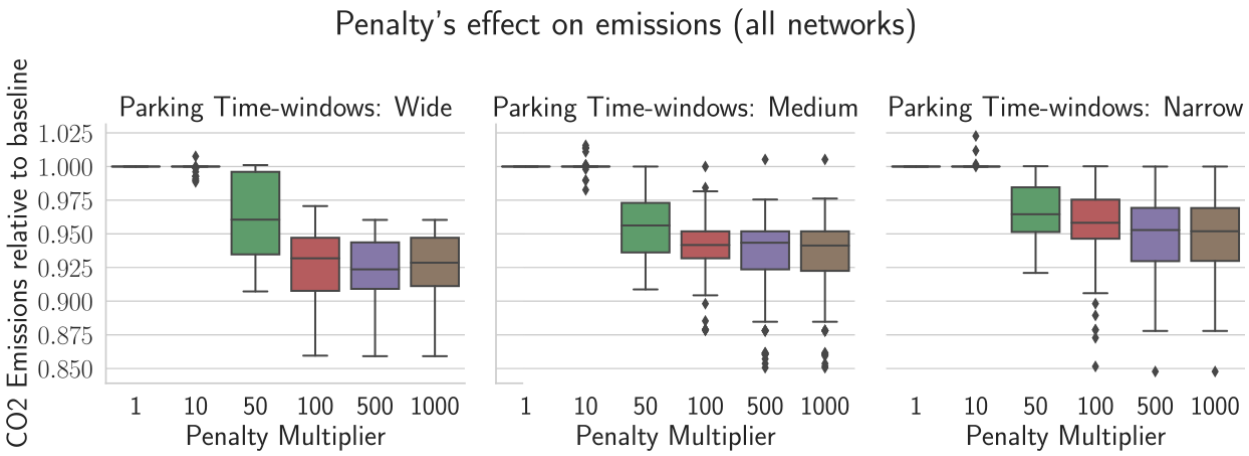


Figure 6. CO_2 emissions as a fraction of the baseline emission. The baseline emission for each scenario is the value obtained with penalty multiplier of 1.

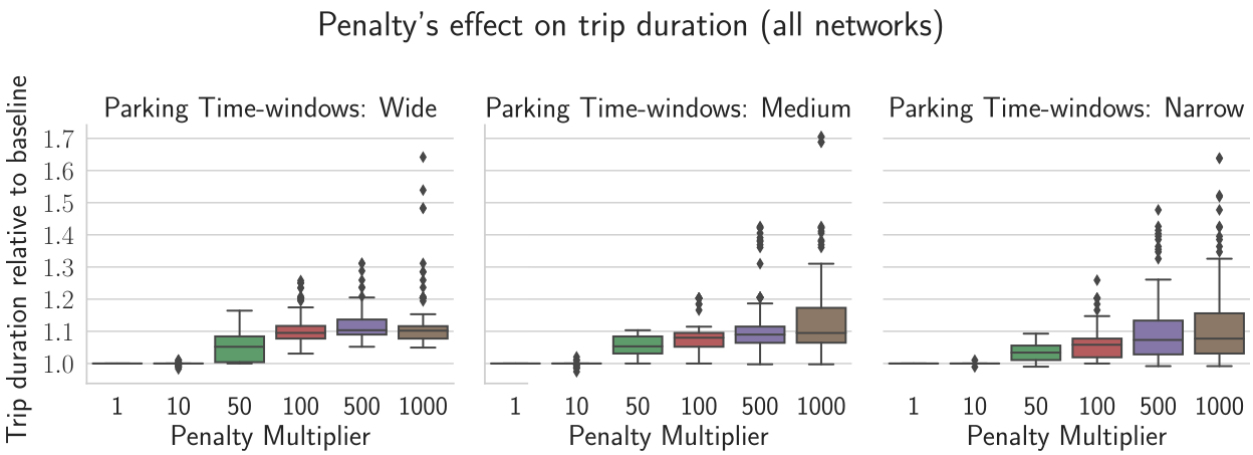


Figure 7. Trip duration as a fraction of the baseline duration. The baseline duration for each scenario is the value obtained with penalty multiplier of 1.

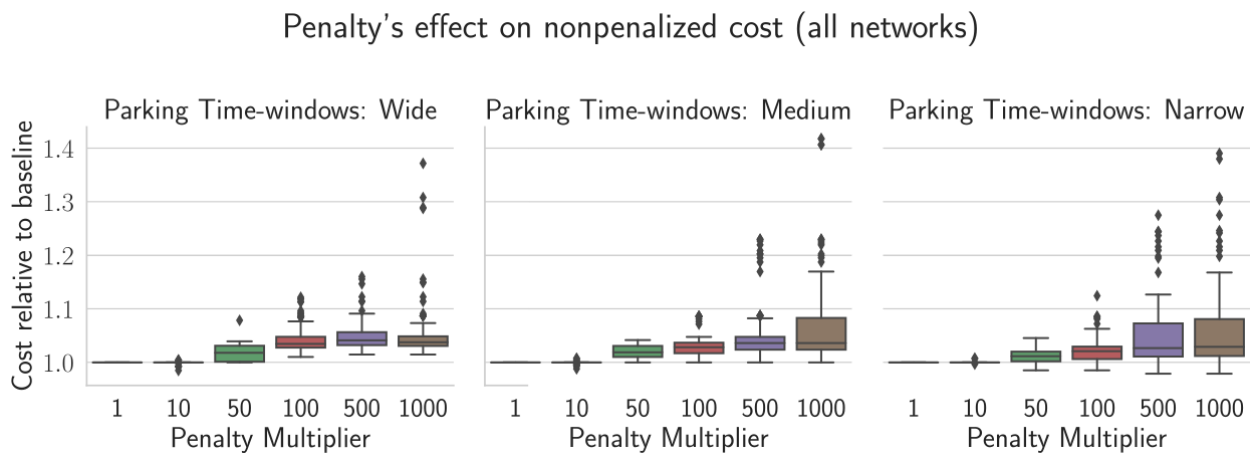


Figure 8. Nonpenalized cost as a fraction of the baseline cost. The baseline cost for each scenario is the value obtained with penalty multiplier of 1.

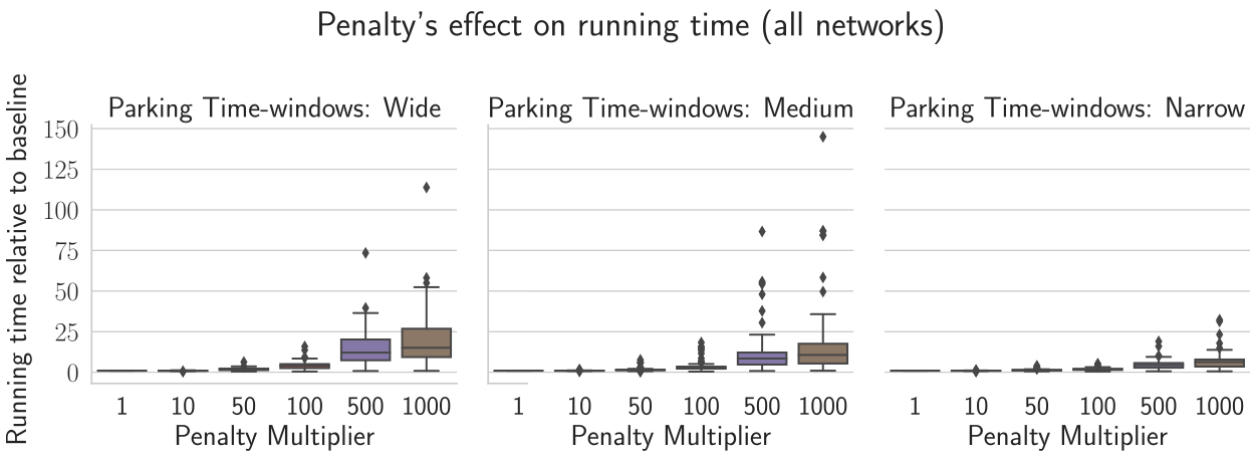


Figure 9. Running time as a fraction of the baseline cost. The baseline running time for each scenario is the value obtained with penalty multiplier of 1.

Time-Dependent Network

Including time-dependent travel times increases problem complexity so an extra heuristic was included to improve performance. As can be seen in the static network experiment results, the problem is significantly faster to solve when lower penalty values are used, so we solved the instances with small penalty first and used the solutions to generate upper bounds for instances with larger penalty. The solution for penalty 1 instances were used to generate upper bounds for the instances with penalty 10, which were used to improve instances with penalty 50, and so on. The results presented were obtained using the lower bound described in section Bound 2 and do not include the time spent solving smaller instances. The rollout policy used a sampled decision space with 0.2h sampling interval, and the base policy is the optimal solution when using a sampled decision space with 2h sampling interval and stopping if a solution is within 1% of the cost lower bound.

Figure 10 presents the improvements obtained in CO_2 emissions when the penalty multiplier is increased under different parking availability conditions. As in the static case, we see up to 5-7% decrease in CO_2 emissions, with the smaller improvements happening in scenarios with narrow time-windows. On the other hand, the impact of higher penalty values on trip duration and cost is substantially larger than in the static case. Although Figure 11 shows that emissions reduction comes at the cost of significant increases in trip duration, we can also notice that when parking availability is scarce the increase in trip duration is not as severe. This behavior is also reflected in the costs shown in Figure 12. Whereas scenarios with wide and medium time-windows showed average cost increases of up to 30%, scenarios with narrow time windows showed an average cost increase of around 12% when the penalty multiplier is set to 1000. These results illustrate the significance of the impact of parking availability conditions and HOS regulations in the cost-benefit analysis of prioritizing emissions reduction during planning.

Figure 13 shows the increases in running time relative to the instances with penalty equal to 1. Even though the solutions of instances with low penalty are being used as upper bounds for the

instances with high penalty, there is still a significant increase in the average running time. Possible reasons for this are: the lower bounds are tighter when lower penalties are used; the significantly larger increases in the penalty multiplier from 100 to 500 and from 500 to 1000 (compared to 1-10, 10-50) make it so the solutions used as upper bounds are not as efficient as for scenarios with closer penalties. Nevertheless, the faster running times for low penalty solutions make it convenient to solve a low penalty instance first and use its solution as an upper bound for high penalty instances. Table 8 to Table 11 present the results for each network separately.

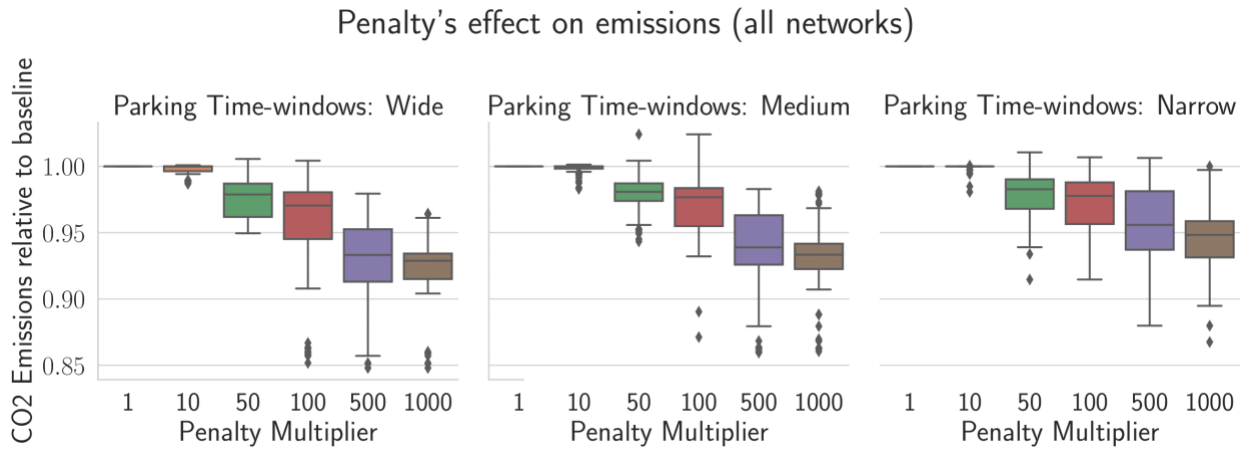


Figure 10. CO_2 emissions as a fraction of the baseline emission. The baseline emission for each scenario is the value obtained with penalty multiplier of 1.

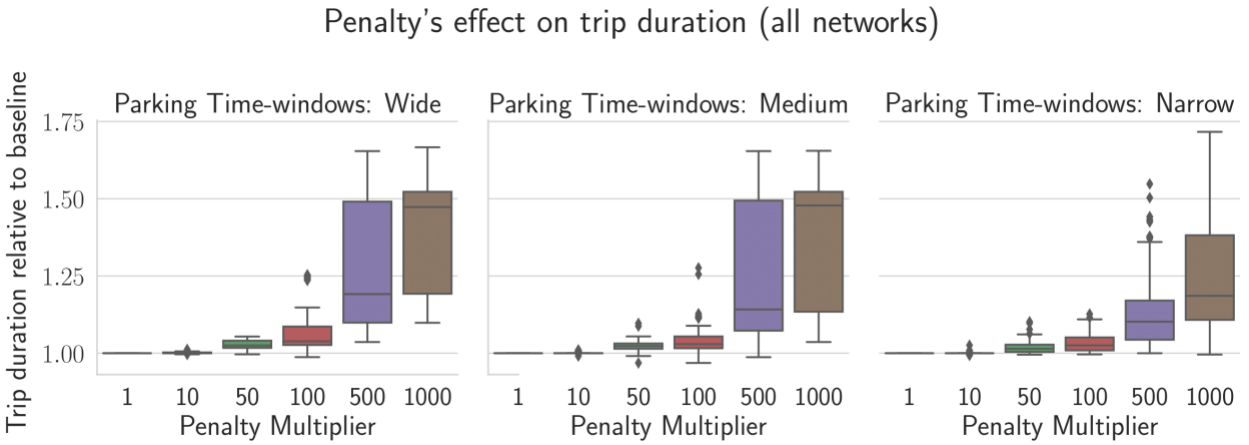


Figure 11. Trip duration as a fraction of the baseline duration. The baseline duration for each scenario is the value obtained with penalty multiplier of 1.

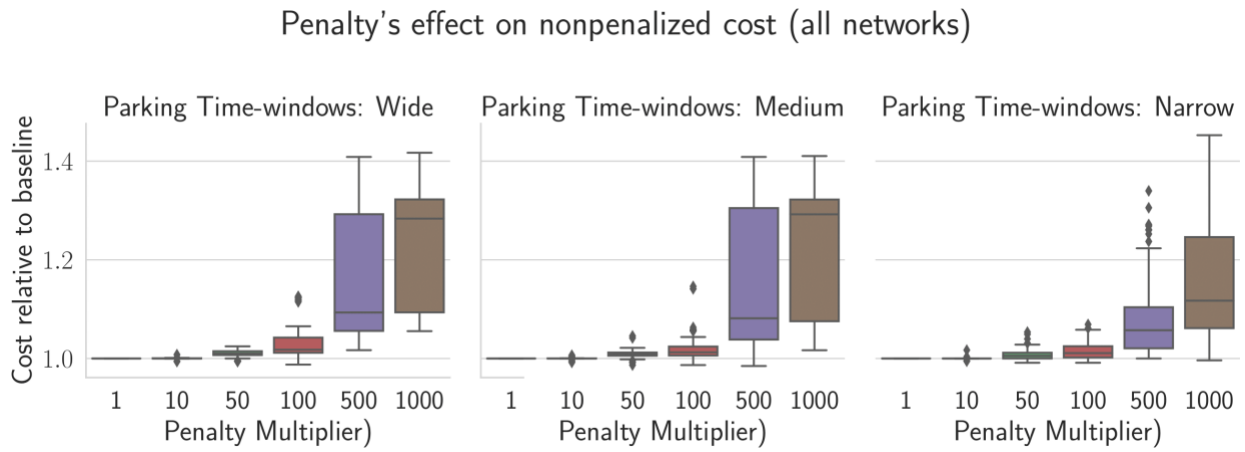


Figure 12. Nonpenalized cost as a fraction of the baseline cost. The baseline cost for each scenario is the value obtained with penalty multiplier of 1.

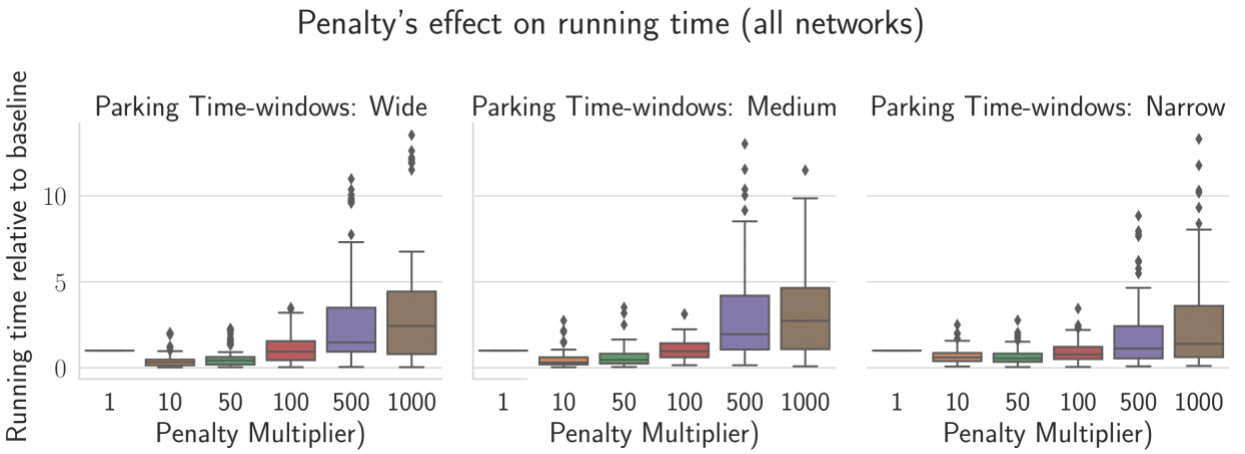


Figure 13. Running time as a fraction of the baseline cost. The baseline running time for each scenario is the value obtained with penalty multiplier of 1.

Uncertain Parking Availability

Our experiments used the same probability distribution for all locations. The parking availability probability distribution was set so that the likelihood of finding parking is low at night and high during the day. This behavior reflects drivers' usual complaints regarding difficulty to find parking for overnight rests. In practice, historical data should be used to define a probability distribution for each rest area.

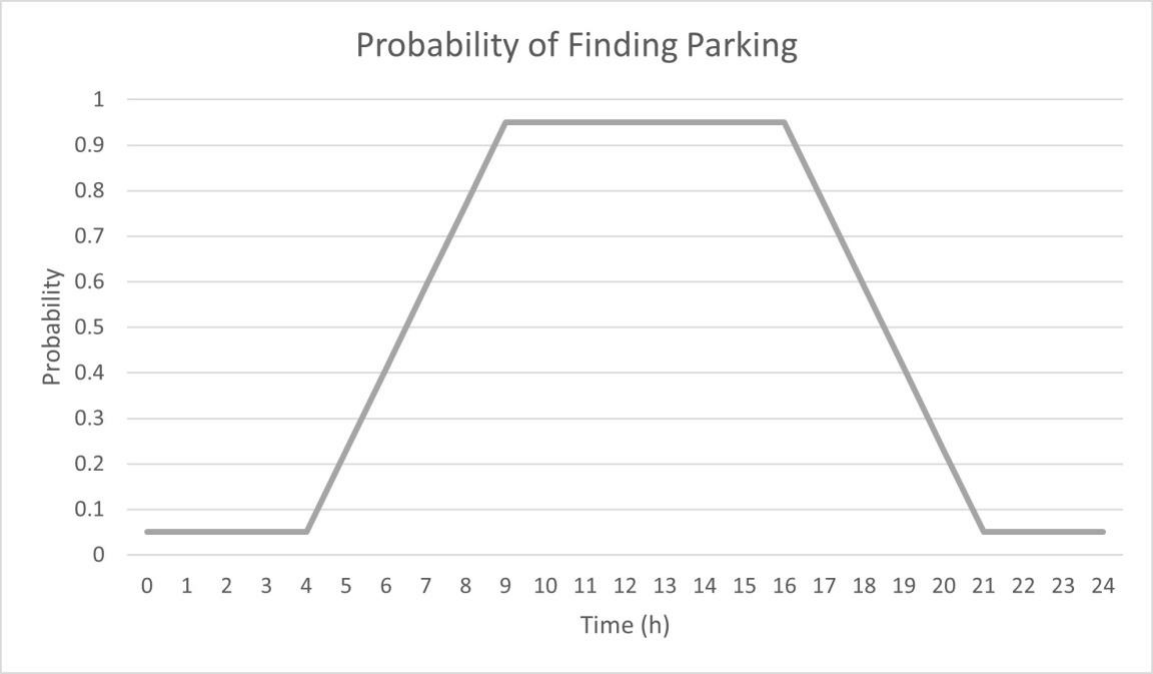


Figure 14. Function used to define the probability of finding parking at a given rest area and time.

Due to the increased complexity, the minimum travel time between rest areas was increased to 4h. In addition, experiments were run only in a subset of the networks used in the deterministic experiments. The objective function includes an hourly penalty of $10/h when parked illegally, and a fixed penalty of $100 per illegal parking event. We set the search when parking is unavailable to 0.5h. We apply a fixed penalty of $100 if the driver has less than 0.5h of remaining driving time, plus a variable penalty of $300/h applied on the difference between the search time and the remaining drive time. Even when parking at unauthorized locations, drivers still need some time to search for a less risky one. Therefore, we apply this penalty to represent both the increased risk of parking at a worse location due to lack of time, and the risk of not being able to find a location to park before exceeding regulations' limits. As described in section Imperfect Information, the scenarios used vary according to the information available to drivers/planners and the weight given to CO_2 emissions. In Scenario 0, drivers know the probability of finding parking at every location and time. For scenarios 1-3, drivers assume that parking is always available within given time-windows. These time-windows are defined as time intervals when the probability of finding parking is equal to or larger than a certain threshold. Scenarios 1, 2 and 3 use, respectively, 0.95, 0.77, and 0.59 as thresholds. Scenarios 4, 5 and 6 use the same time-windows as scenarios 1, 2 and 3, but drivers know that parking is not guaranteed and know the probability of finding parking at each time within those windows.

Results

Figure 15 presents the emissions, trip duration, and trip cost of one of the test networks. When full information is used (scenario 0), the results reflect well the intent of the cost functions used. Scenarios with larger CO_2 penalties result in lower emissions even at the cost of higher

trip cost and duration, and worse results have decreasing probabilities. Scenarios 1 to 3 generate somewhat disorderly probability distributions with average performance inferior to scenario 0 and larger variances. In these scenarios, the CO_2 penalty is unable to control the CO_2 emissions as effectively. For example, in scenario 2 of Network 5, the average CO_2 emissions with penalty 50 is larger than with penalty 1.

Figure 16 presents the probability distributions for parking-related performance measures for each test network for scenarios 0 to 3. As for Figure 15, scenario 0 has a significantly better performance, with decreasing probabilities for worse outcomes and similar behavior for all networks. On the other hand, scenarios 1 to 3's performance is less consistent and more sensitive to the network.

Figure 17 and Figure 18 present the same performance measures, but comparing scenarios 4 to 6 with scenario 0. In all these scenarios the driver has full information about the probability of finding parking. However, in the scenarios 4 to 6, drivers are restricted to the same time-windows considered in scenarios 1 to 3. We can see that by informing drivers about the parking uncertainty and including this information in the planning algorithm, the performance is significantly less affected by the time-windows used. The drop in performance is not as pronounced, and the results are less sensitive to scenario parameters, keeping the exponential-like shape of the distributions found for scenario 0. Although restricting drivers' decisions to these time-windows can negatively impact performance, it significantly reduces problem complexity, thus reducing running time. One possible approach it to use policies generated from narrow time-window scenarios as base policies for the rollout algorithm. Table 12 to Table 15 present the expected values for the performance measures used in Figure 15 to Figure 18 for each network, scenario and CO_2 penalty value.

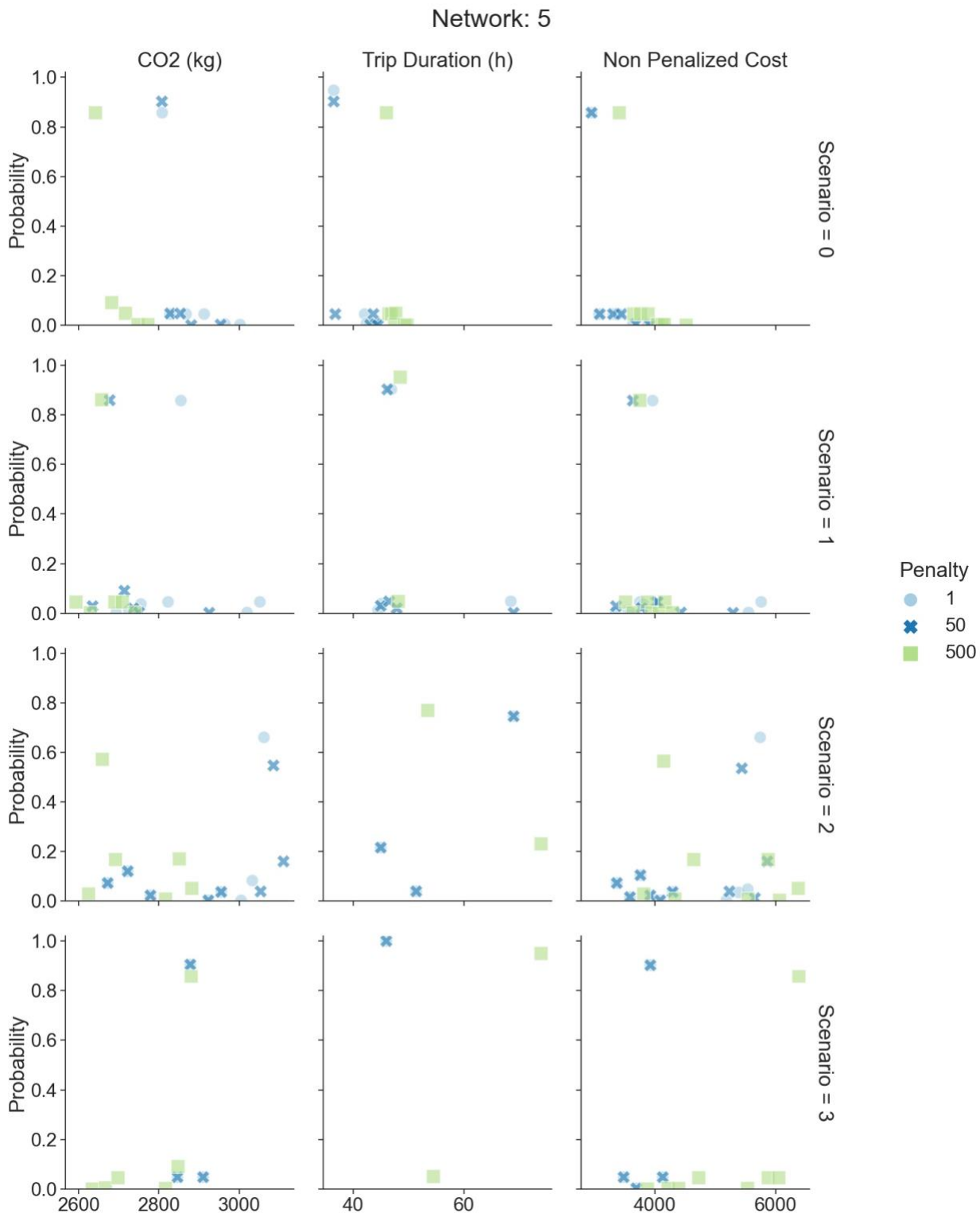


Figure 15. Probability distribution of CO_2 emissions, trip duration and trip cost (with CO_2 penalty parameter set to 1) of the decision policies obtained for network 5.

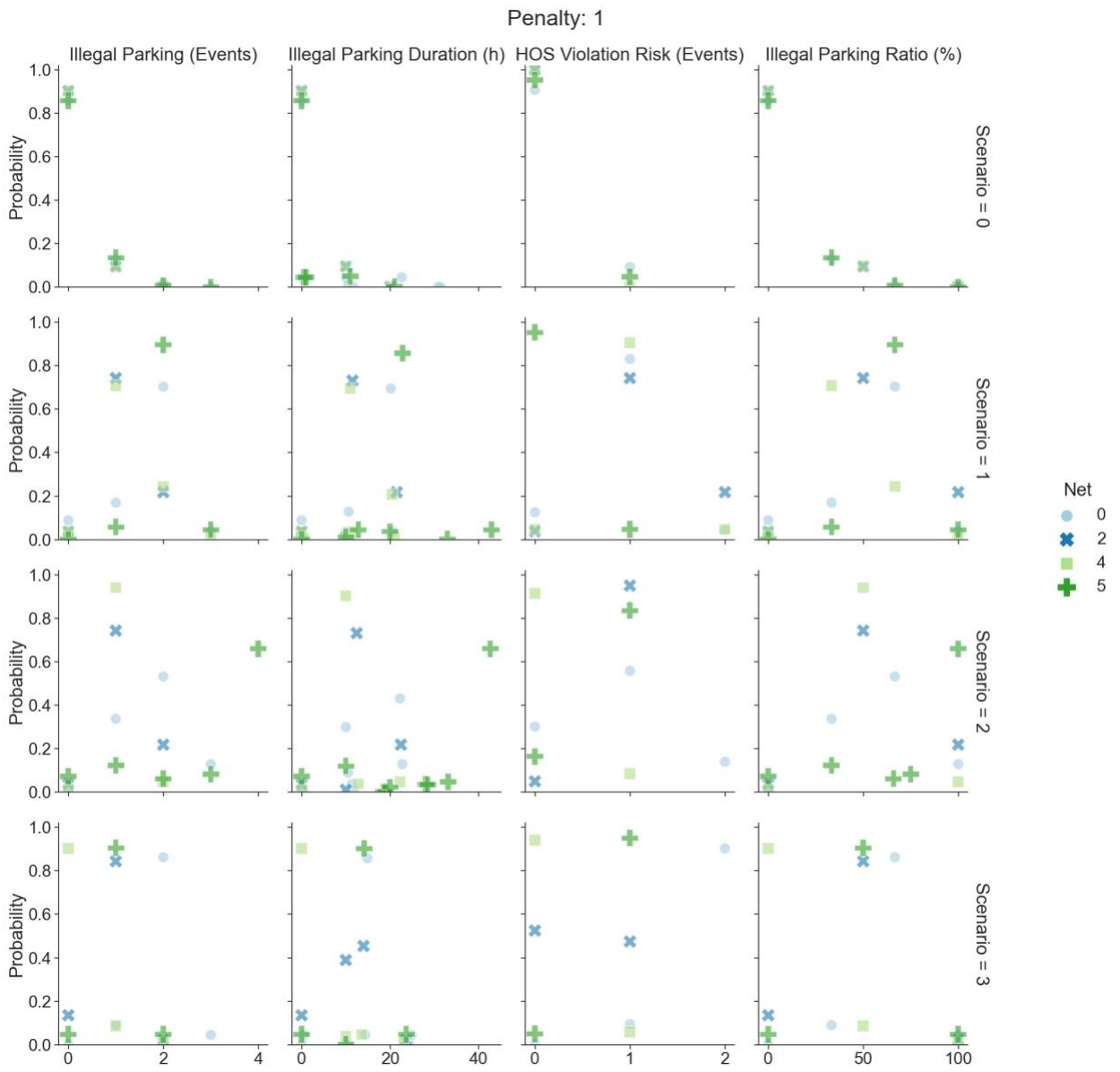


Figure 16. Probability distribution performance measures related to illegal parking for networks 0,2,4 and 5, with CO_2 penalty set to 1.

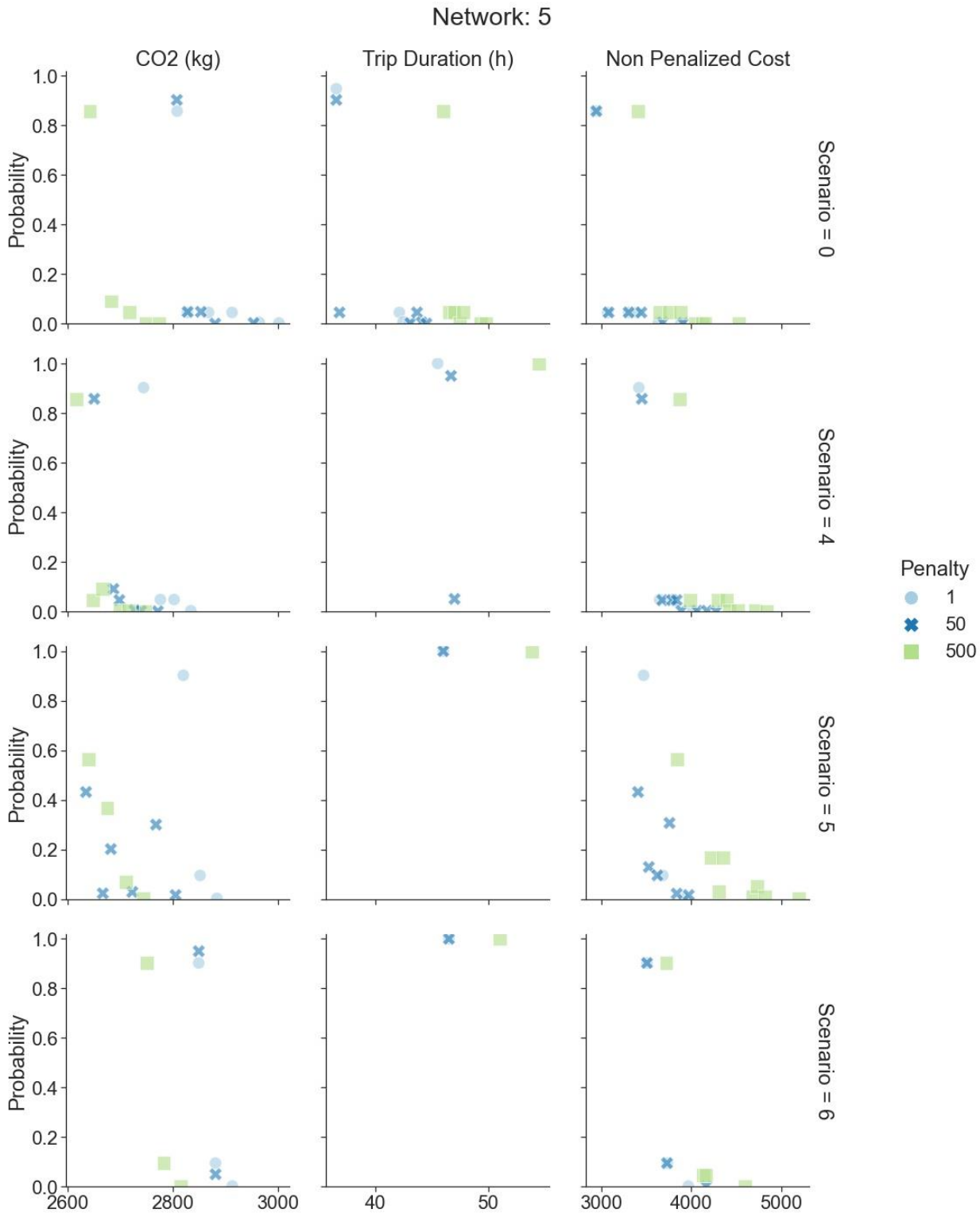


Figure 17. Probability distribution of CO_2 emissions, trip duration and trip cost (with CO_2 penalty parameter set to 1) of the decision policies obtained for network 5.

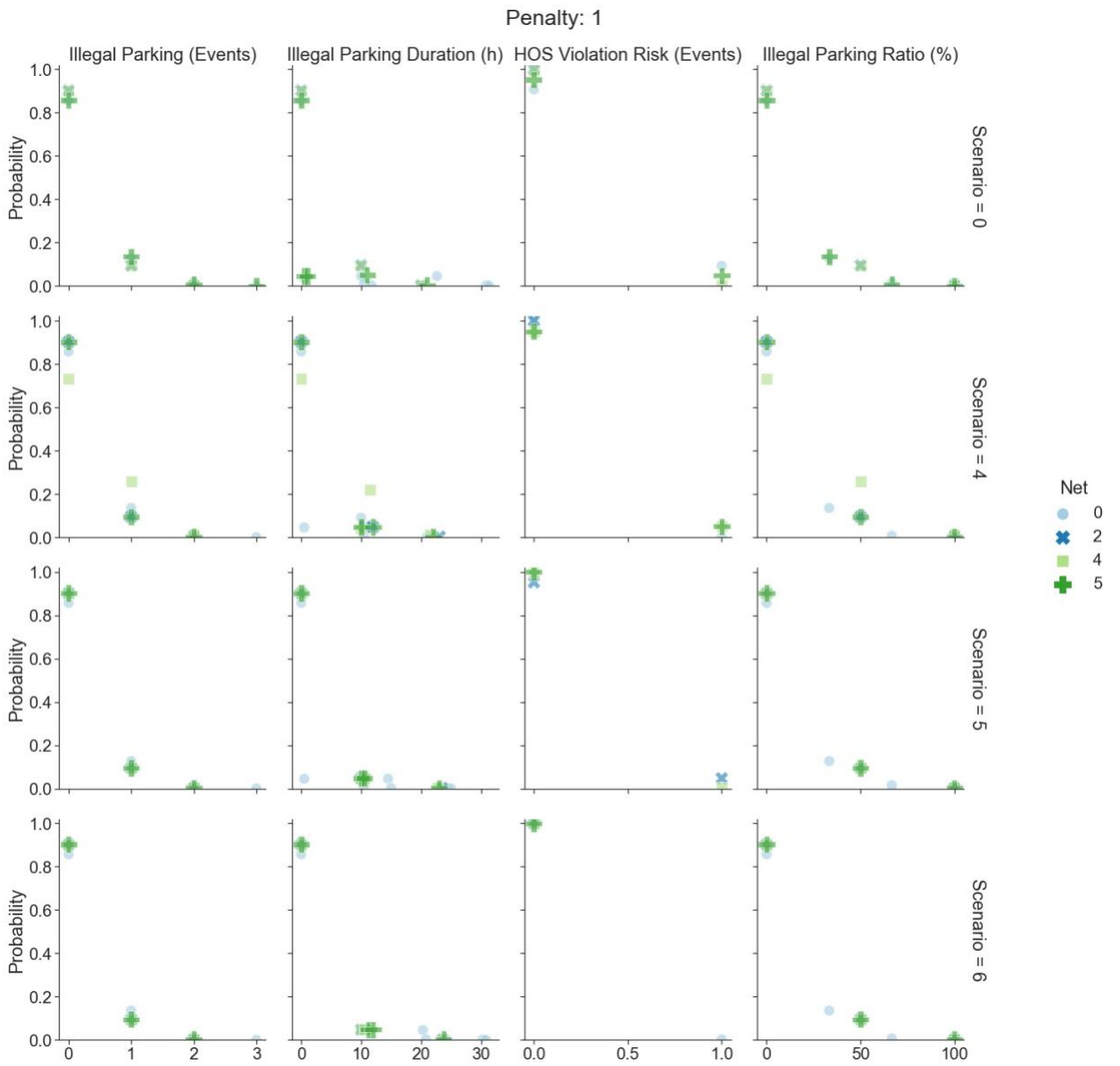


Figure 18. Probability distribution performance measures related to illegal parking for networks 0, 2, 4 and 5, with CO_2 penalty set to 1.

Feasibility of Commercialization and possible model extensions

In this section, to assess the feasibility of commercialization, we discuss the project's strengths and shortcomings. We begin by recalling the project objectives, followed by system requirements, potential benefits, and the disadvantages that could hinder implementation.

Objectives

The project objective is to integrate eco-routing, parking information, and regulation-aware scheduling to develop models able to better describe the practical constraints faced by drivers in realistic scenarios. Such a model can be used not only by individual drivers as a trip planning aid but also by policymakers to analyze the impacts of policy or investment decisions (e.g., changes in environmental policy, parking infrastructure, road network, and HOS regulations) on truck routes, schedules, costs, and emissions.

Requirements

Parking Data: The main requirement for any implementation of this system is for truck parking availability data to be available. Ideally, real-time parking availability should be available for all parking facilities being considered. Currently, truck parking data is available only in a limited number of parking facilities used as testbeds for intelligent truck parking systems and in some private truck stops with reservation systems. It might be feasible to test this system in regions where a sufficient number of parking facilities provide availability information, but this could significantly limit drivers' parking options. Another option would be to use rough estimates based on surveys or user experience to guide planning (e.g., avoid stopping at region A after 6 PM) and gradually update these guidelines according to user feedback.

Consumption Models: In our experiments, we used consumption models found in the literature. For better emissions estimates, the system would need consumption models for the main truck models in operation and models to be used for less common models, or a reasonable/convenient way to estimate models for particular vehicles as needed.

Map Information: The system would need road networks information, including information on which routes can be accessed by trucks, speed limits, historical data on average travel speed, and the ability to estimate future travel speeds (possibly accounting for planned events). This kind of information should already be available in today's GPS guidance systems.

Benefits

Illegal Truck Parking: Truck parking is a critical issue in the USA, and it can have a significant impact on the environment and industry costs. Integrating truck parking information in the planning process can mitigate this issue by recommending safer itineraries to drivers. Although, in general, parking availability is uncertain, and we cannot guarantee a parking space, by including this uncertainty in the model, users can better manage the risks. In scenarios where parking availability is deterministic, e.g., reservation systems, a scenario that guarantees parking is generated.

Emission Reduction: The scenarios studied showed reductions of up to 5-8% on average CO_2 emissions, which come at the cost of increases on average trip duration and average trip cost. However, results showed a large variance, with some instances having more than 40% increase in trip duration and cost, especially when considering time-dependent speed limits. These results show that, at times, prioritizing emissions might be too costly and illustrate the importance of improving the models used to evaluate the impact of any policy and investment decisions. The proposed model can help estimate the level of emissions reduction that can be expected for different regions and types of vehicles, at what cost, and how they are affected by the region's truck parking infrastructure.

Disadvantages

Cost Increase and Adoption: Safer and environmentally friendly itineraries can significantly increase trip duration (and, consequently, cost), hindering voluntary adoption of any such system unless some incentive is provided. The proposed model provides a way to generate regulation and parking-aware routes with a user-defined level of importance given to emissions and estimate the suggested routes' cost, duration, and emissions. However, we do not address how to encourage drivers and companies to adopt such a system. That said, policymakers could use the model to estimate their decisions' impact on the trucking industry and truck parking demand.

System-Level Effects: In this project, we consider the problem of planning trips for individual vehicles, and we do not account for the effects that each driver's decisions would have on the overall parking availability. This can be seen as assuming that a relatively small number of drivers use the system. If adopted on a large scale without taking system-wide effects into account, similar recommendations could be given to many drivers and end up having adverse effects. Therefore, the proposed model could be used for simulations or small-scale tests, but further research is required to make it adequate for large-scale deployment.

Scalability: The problem treated is very complex, requiring decisions on route, schedule, and travel speeds, depending on traffic conditions, HOS regulations, and parking availability information. Therefore, as trip length, duration, and the number of routes and parking facilities considered increase, the problem turns intractable fairly quickly. This problem can be alleviated by limiting the number of routes considered using heuristics or user knowledge, and grouping nearby parking facilities as a single location. Besides network size, the resolution used for decisions also has a significant impact on problem complexity. Our experiments use a resolution of 0.5h for decisions on trips with average durations between 28h and 60h, resulting in average running times ranging from 1s to 330s. This shows that, for similar-sized problems, there is still some margin to increase the resolution, network size, or relax some of the heuristics used. Especially if we consider that, as the interval between rests is relatively long, drivers could generate a schedule with a lower resolution right before the trip and let the system refine it while driving. However, running time sharply increases when longer trips are considered, especially for trips where on-duty time reaches the 60h limit, and 34h rests need to be scheduled. Another issue is that speed optimization is sensitive to link lengths and decision space's time resolution. Speed cannot be controlled if the link's minimum travel time is too

short, as any increase in travel time might push the speed beyond the minimum limit. The network can be preprocessed to merge short links, but fuel consumption calculation will be affected when the speed information for different links is merged into a single link. As the consumption model is nonlinear, the average travel time over the concatenated/merged link will generate a different consumption than the sum of the links' consumptions (when using their own average speeds). As merging links also helps improve algorithm performance by decreasing the number of decisions required, using the original links and increasing the resolution so that speed can be controlled would slow the algorithm down substantially. Therefore, we lose some accuracy in the emission estimates due to scalability issues.

Conclusion

The main issues for commercialization are the lack of truck parking availability data, scalability, and the need to study the systemwide effects that would be caused by a large-scale deployment. The current state of the USA's truck parking infrastructure is unfavorable for systems that rely on parking availability information such as the one developed in this project. Due to the limited availability of truck parking data, usage would be limited to regions and routes with facilities able to provide data, or to applications where it is acceptable to use rough estimates based on user experience or surveys. For example, if used to estimate the impact of policy decisions, it might suffice to perform periodic surveys of the parking conditions in order to determine reasonable simulation parameters. Furthermore, even in the absence of parking data, drivers might appreciate the option of defining time periods when they would rather avoid certain parking facilities based on past experience, even if that is not as accurate as data provided by intelligent parking systems. We believe that, currently, the potential for commercialization is low as a standalone system, but parts of it could be gradually integrated into existing routing/scheduling/planning systems for small-scale experimentation. We recommend as a primary direction for future research studying how large-scale usage of this type of planning algorithm would affect truck parking demand, and ways to coordinate decisions so as to avoid adverse effects.

Conclusion

In this study, we addressed a variant of the shortest path and truck driver scheduling problem under parking availability constraints which focuses on optimizing fuel consumption and emissions by controlling the truck's travel speed and accounting for time-dependent traffic conditions. As it is impossible to be absolutely certain about the future parking availability of any location during planning, we also studied the case of stochastic parking availability.

When studying the trade-offs between prioritizing emissions reduction or trip duration, we found that although focusing on emissions reduction can increase trip duration significantly, this impact is greatly reduced when considering scenarios with limited parking availability. We also present a cost lower bound that combines HOS requirements with information on optimal speeds for particular cost functions, and can be used to significantly speed-up problem solution in deterministic scenarios, both static and time-dependent.

The resource-constrained shortest path formulation was further extended to model drivers possible recourse actions when unable to find parking and the ensuing costs. We used this formulation to study how the solutions are affected by the level of information provided to drivers. We found that ignoring uncertainty in parking availability results in inconsistent performance even when restricting parking to periods when probability of finding parking is high. Furthermore, results might not reflect the intent of the cost function used, e.g., minimizing illegal parking events and/or the priority assigned to emissions reduction. Giving drivers full information about the probability of finding parking at any time/location significantly improves performance and reduces illegal parking-related risks, but also substantially increase problem complexity and computation time. Using full information regarding parking availability but restricting the parking times to high availability time-windows can reduce complexity while maintaining consistent, although reduced, performance.

References

Agriesti, S., Gandini, P., Marchionni, G., Paglino, V., Ponti, M., & Studer, L. (2018). Evaluation approach for a combined implementation of day 1 C-ITS and truck platooning. *IEEE Vehicular Technology Conference*, *2018-June*, 1–6. https://doi.org/10.1109/VTCSpring.2018.8417876

Archetti, C., & Savelsbergh, M. (2009). The Trip Scheduling Problem. *Transportation Science*, *43*(4), 417–431. https://doi.org/10.1287/trsc.1090.0278

Argonne National Laboratory. (2020). *GREET.Net 2020* (v. 1.3.0.13656).

Barboza, T., & Lange, J. H. (2018). *California hit its climate goal early — but its biggest source of pollution keeps rising*. LA Times. https://www.latimes.com/local/lanow/la-me-adv-california-climate-pollution-20180722-story.html

Ben Ticha, H., Absi, N., Feillet, D., Quilliot, A., & Van Woensel, T. (2021). The Time-Dependent Vehicle Routing Problem with Time Windows and Road-Network Information. *SN Operations Research Forum*, *2*(1), 4. https://doi.org/10.1007/s43069-020-00049-6

Bertsekas, D. (2017). *Dynamic Programming and Optimal Control, Vol. I* (4th Editio). Athena Scientific.

Cai, X., Kloks, T., & Wong, C. K. (1997). Time-varying shortest path problems with constraints. *Networks*, *29*(3), 141–150. https://doi.org/10.1002/(sici)1097-0037(199705)29:3<141::aid-net2>3.0.co;2-h

California Air Resources Board. (2021). *California Cap-and-Trade Program: Summary of California-Quebec Joint Auction Settlement Prices and Results* (Issue February 2021). https://ww2.arb.ca.gov/sites/default/files/2020-08/results_summary.pdf

Cambridge Systematics. (2019). *Nevada Truck Parking Implementation Plan - Task 5: Draft Recommendations*.

Cooke, K. L., & Halsey, E. (1966). The shortest route through a network with time-dependent internodal transit times. *Journal of Mathematical Analysis and Applications*, *14*(3), 493–498. https://doi.org/10.1016/0022-247X(66)90009-6

Cooper, B. S., & Cowlagi, R. V. (2018). Path-planning with waiting in spatiotemporally-varying threat fields. *PLOS ONE*, *13*(8), e0202145. https://doi.org/10.1371/journal.pone.0202145

Costa, L., Contardo, C., & Desaulniers, G. (2019). Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, *53*(4), 946–985. https://doi.org/10.1287/trsc.2018.0878

Dean, B. C. (2001). Shortest Paths in FIFO Time-Dependent Networks: Theory and Algorithms. *Japanese Journal of Applied Physics*, *23*(7), L490–L492. https://doi.org/10.1143/JJAP.23.L490

Dean, B. C. (2004). Algorithms for minimum-cost paths in time-dependent networks with waiting policies. *Networks*, *44*(1), 41–46. https://doi.org/10.1002/net.20013

Dell'Amico, M., Iori, M., & Pretolani, D. (2008). Shortest paths in piecewise continuous time-dependent networks. *Operations Research Letters*, *36*(6), 688–691. https://doi.org/10.1016/j.orl.2008.07.002

Delling, D. (2011). Time-Dependent SHARC-Routing. *Algorithmica*, *60*(1), 60–94. https://doi.org/10.1007/s00453-009-9341-0

Demir, E., Bektaş, T., & Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, *16*(5), 347–357. https://doi.org/10.1016/j.trd.2011.01.011

Drexl, M., & Prescott-Gagnon, E. (2010). Labelling algorithms for the elementary shortest path problem with resource constraints considering EU drivers' rules. *Logistics Research*, *2*(2), 79–96. https://doi.org/10.1007/s12159-010-0022-9

Eglese, R., & Bektas, T. (2014). Chapter 15: Green Vehicle Routing. In *Vehicle Routing* (pp. 437–458). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611973594.ch15

EPA. (2018). *Fast Facts. U.S.Transportation Sector Greenhouse Gas Emissions 1990 –2016*. *July*.

Federal Motor Carrier Safety Administration. (2021). *Title 49 Part 395, HOURS OF SERVICE OF DRIVERS*. Electronic Code of Federal Regulations. https://ecfr.federalregister.gov/current/title-49/subtitle-B/chapter-III/subchapter-B/part-395

Ferone, D., Festa, P., Napoletano, A., & Pastore, T. (2017). Shortest paths on dynamic graphs: A survey. *Pesquisa Operacional*, *37*(3), 487–508. https://doi.org/10.1590/0101-7438.2017.037.03.0487

Foschini, L., Hershberger, J., & Suri, S. (2011). On the complexity of time-dependent shortest paths. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 327–341. https://doi.org/10.1137/1.9781611973082.27

Franceschetti, A., Honhon, D., Van Woensel, T., Bektaş, T., & Laporte, G. (2013). The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, *56*, 265–293. https://doi.org/10.1016/j.trb.2013.08.008

Gaddy, A. C., Hernandez, S., & Nurre, S. (2018). Incorporating Truck Parking and Hours of Service (HOS) into a Truck Routing Heuristic. *Transportation Research Board 97th Annual Meeting*.

Goel, A. (2010). Truck Driver Scheduling in the European Union. *Transportation Science*, *44*(4), 429–441. https://doi.org/10.1287/trsc.1100.0330

Goel, Asvin. (2012). The minimum duration truck driver scheduling problem. *EURO Journal on Transportation and Logistics*, *1*(4), 285–306. https://doi.org/10.1016/j.cor.2011.12.016

Goel, Asvin, & Irnich, S. (2017). An Exact Method for Vehicle Routing and Truck Driver Scheduling Problems. *Transportation Science*, *51*(2), 737–754. https://doi.org/10.1287/trsc.2016.0678

Goel, Asvin, & Kok, L. (2012). Truck Driver Scheduling in the United States. *Transportation Science*, *46*(3), 317–326. https://doi.org/10.1287/trsc.1110.0382

Goel, Asvin, & Rousseau, L.-M. (2011). Truck driver scheduling in Canada. *Journal of Scheduling*, *15*(6), 783–799. https://doi.org/10.1007/s10951-011-0249-6

Goel, Asvin, & Vidal, T. (2014). Hours of Service Regulations in Road Freight Transport: An Optimization-Based International Assessment. *Transportation Science*, *48*(3), 391–412. https://doi.org/10.1287/trsc.2013.0477

Gregor, E. (2018). *EU Legislation in Progress: CO2 emission standards for heavy-duty vehicles*.

He, E., Boland, N., Nemhauser, G., & Savelsbergh, M. (2020). Time-Dependent Shortest Path Problems with Penalties and Limits on Waiting. *INFORMS Journal on Computing*, ijoc.2020.0985. https://doi.org/10.1287/ijoc.2020.0985

Horváth, M., & Kis, T. (2016). Solving resource constrained shortest path problems with LP-based methods. *Computers and Operations Research*, *73*, 150–164. https://doi.org/10.1016/j.cor.2016.04.013

Huang, W., Yan, C., Wang, J., & Wang, W. (2017). A time-delay neural network for solving time-dependent shortest path problem. *Neural Networks*, *90*, 21–28. https://doi.org/10.1016/j.neunet.2017.03.002

Ioachim, I., Gélinas, S., Soumis, F., & Desrosiers, J. (1998). A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, *31*(3), 193–204. https://doi.org/10.1002/(SICI)1097-0037(199805)31:3<193::AID-NET6>3.0.CO;2-A

Irnich, S., & Desaulniers, G. (2005). Shortest Path Problems with Resource Constraints. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.), *Column Generation* (pp. 33–65). Springer US.

Kaufman, D. E., & Smith, R. L. (2007). Fastest Paths in Time-Dependent Networks for Intelligent Vehicle-Highway Systems Application. *I V H S Journal*, *1*(1), 1–11. https://doi.org/10.1080/10248079308903779

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). A comparison of three idling options in long-haul truck scheduling. *Transportation Research Part B: Methodological*, *93*, 631–647. https://doi.org/10.1016/j.trb.2016.08.006

Koç, Ç., Jabali, O., & Laporte, G. (2018). Long-haul vehicle routing and scheduling with idling options. *Journal of the Operational Research Society*, *69*(2), 235–246. https://doi.org/10.1057/s41274-017-0202-y

Kok, A. L., Hans, E. W., & Schutten, J. M. J. (2011). Optimizing departure times in vehicle routes. *European Journal of Operational Research*, *210*(3), 579–587. https://doi.org/10.1016/j.ejor.2010.10.017

Kok, A. L., Hans, E. W., Schutten, J. M. J., & Zijm, W. H. M. (2010). A dynamic programming heuristic for vehicle routing with time-dependent travel times and required breaks. *Flexible Services and Manufacturing Journal*, *22*(1–2), 83–108. https://doi.org/10.1007/s10696-011-9077-4

Kok, A. L., Meyer, C. M., Kopfer, H., & Schutten, J. M. J. (2010). A Dynamic Programming Heuristic for the Vehicle Routing Problem with Time Windows and European Community Social Legislation. *Transportation Science*, *44*(4), 442–454. https://doi.org/10.1287/trsc.1100.0331

Lauther, U. (2004). An extremely fast, exact algorithm for finding shortest paths in static networks with geographical background. *Geoinformation Und Mobilität-von Der Forschung Zur Praktischen Anwendung*, *22*, 219–230.

Lozano, L., Duque, D., & Medaglia, A. L. (2016). An exact algorithm for the elementary shortest path problem with resource constraints. *Transportation Science*, *50*(1), 348–357. https://doi.org/10.1287/trsc.2014.0582

Mayerle, S. F., De Genaro Chiroli, D. M., Neiva de Figueiredo, J., & Rodrigues, H. F. (2020). The long-haul full-load vehicle routing and truck driver scheduling problem with intermediate stops: An economic impact evaluation of Brazilian policy. *Transportation Research Part A: Policy and Practice*, *140*(December 2018), 36–51. https://doi.org/10.1016/j.tra.2020.07.021

Murray, D., & Glidewell, S. (2019). *An Analysis of the Operational Costs of Trucking: 2019 Update* (Issue November).

Nannicini, G., Delling, D., Liberti, L., & Schultes, D. (2008). Bidirectional A* search for time-dependent fast paths. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5038 LNCS*, 334–346. https://doi.org/10.1007/978-3-540-68552-4_25

Omer, J., & Poss, M. (2020). A polynomial algorithm for minimizing travel time in consistent time-dependent networks with waits. *Networks*, *September*, net.21994. https://doi.org/10.1002/net.21994

Orda, A., & Rom, R. (1990). Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM*, *37*(3), 607–625. https://doi.org/10.1145/79147.214078

Orda, A., & Rom, R. (1991). Minimum weight paths in time-dependent networks. *Networks*, *21*(3), 295–319. https://doi.org/10.1002/net.3230210304

Pugliese, L. D. P., & Guerriero, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, *62*(3), 183–200. https://doi.org/10.1002/net.21511

Rancourt, M.-E., Cordeau, J.-F., & Laporte, G. (2013). Long-Haul Vehicle Routing and Scheduling with Working Hour Rules. *Transportation Science*, *47*(1), 81–107.

Sanders, P., & Schultes, D. (2006). Engineering Highway Hierarchies. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 4168 LNCS* (pp. 804–816). https://doi.org/10.1007/11841036_71

SETPOS Consortium. (2009). *Background Information and Considerations for Secure Truck Parking*.

Shah, V. D. (2008). *Time dependent truck routing and driver scheduling problem with hours of service regulations* [Northeastern University]. http://hdl.handle.net/2047/d10016995

Sherali, H. D., Hobeika, A. G., & Kangwalklai, S. (2003). Time-Dependent, Label-Constrained Shortest Path Problems with Applications. *Transportation Science*, *37*(3), 278–293. https://doi.org/10.1287/trsc.37.3.278.16042

Spliet, R., Dabia, S., & Van Woensel, T. (2018). The Time Window Assignment Vehicle Routing Problem with Time-Dependent Travel Times. *Transportation Science*, *52*(2), 261–276. https://doi.org/10.1287/trsc.2016.0705

Sripad, S., & Viswanathan, V. (2017). Performance Metrics Required of Next-Generation Batteries to Make a Practical Electric Semi Truck. *ACS Energy Letters*, *2*(7), 1669–1673. https://doi.org/10.1021/acsenergylett.7b00432

Sun, Y., Toledo, T., Rosa, K., Ben-Akiva, M. E., Flanagan, K., Sanchez, R., & Spissu, E. (2013). Route choice characteristics for truckers. *Transportation Research Record*, *2354*, 115–121. https://doi.org/10.3141/2354-12

Tsugawa, S. (2014). Results and issues of an automated truck platoon within the energy ITS project. *IEEE Intelligent Vehicles Symposium, Proceedings*, *Iv*, 642–647. https://doi.org/10.1109/IVS.2014.6856400

Tsugawa, S., Jeschke, S., & Shladover, S. E. (2016). A Review of Truck Platooning Projects for Energy Savings. *IEEE Transactions on Intelligent Vehicles*, *1*(1), 68–77. https://doi.org/10.1109/TIV.2016.2577499

U.S. Department of Energy. (2015). *Long-Haul Truck Idling Burns Up Profits*. https://afdc.energy.gov/files/u/publication/hdv_idling_2015.pdf

U.S. Department of Transportation. (2015). *Jason's Law Truck Parking Survey Results and Comparative Analysis* (Issue August).

U.S. Energy Information Administration. (2016). *Carbon Dioxide Emissions Coefficients*. https://www.eia.gov/environment/emissions/co2_vol_mass.php

Van De Hoef, S., Johansson, K. H., & Dimarogonas, D. V. (2015). Fuel-optimal centralized coordination of truck platooning based on shortest paths. *Proceedings of the American Control Conference*, *2015-July*, 3740–3745. https://doi.org/10.1109/ACC.2015.7171911

Vital, F., & Ioannou, P. (2021a). Scheduling and shortest path for trucks with working hours and parking availability constraints. *Transportation Research Part B: Methodological*, *148*, 1–37. https://doi.org/10.1016/j.trb.2021.04.002

Vital, F., & Ioannou, P. (2021b). Effects of Working Hour Regulations and Parking Shortages on Truck Electrification. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 3360–3365. https://doi.org/10.1109/ITSC48978.2021.9564810

Vital, F., & Ioannou, P. (2019). Long-Haul Truck Scheduling with Driving Hours and Parking Availability Constraints. *2019 IEEE Intelligent Vehicles Symposium (IV)*, *June*, 620–625. https://doi.org/10.1109/IVS.2019.8814011

Vital, F., & Ioannou, P. (2020). Truck Routing under Rest Area Parking Constraints. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–6. https://doi.org/10.1109/ITSC45102.2020.9294253

Wang, J., & Rakha, H. A. (2017). Fuel consumption model for heavy duty diesel trucks: Model development and testing. *Transportation Research Part D: Transport and Environment*, *55*, 127–141. https://doi.org/10.1016/j.trd.2017.06.011

Wenting, H., & Xiaoqiang, C. (2007). Time-Varying Shortest Path Problems with Perishable Product and Constraints. *2007 International Conference on Service Systems and Service Management*, 1–5. https://doi.org/10.1109/ICSSSM.2007.4280255

Xu, H., Chen, Z.-L., Rajagopal, S., & Arunapuram, S. (2003). Solving a Practical Pickup and Delivery Problem. *Transportation Science*, *37*(3), 347–364. https://doi.org/10.1287/trsc.37.3.347.16044

Zhang, J., Zhao, Y., Xue, W., & Li, J. (2015). Vehicle routing problem with fuel consumption and carbon emission. *International Journal of Production Economics*, *170*, 234–242. https://doi.org/10.1016/j.ijpe.2015.09.031

Zhao, L., Ohshima, T., & Nagamochi, H. (2008). A * Algorithm for the time-dependent shortest path. *WAAC08: The 11th Japan-Korea Joint Workshop on Algorithms and Computation*, *April*.

Ziliaskopoulos, A. K., & Mahmassani, H. S. (1993). Time-Dependent , Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications. *Transportation Research Record*, *1408*, 94–100.

Data Summary

Products of Research

The data generated are simulation data presented in plots and tables in the final report.

Data Format and Content

The data is presented as tables and plots in the final report, and consists of average emissions, durations and costs, as well as experiments' average running times.

Data Access and Sharing

The data is included in the final report.

Reuse and Redistribution

The data will be published as part of the final report.

Appendix A Experiment Results

Static Networks

Detailed Results

Table 4. Static Network Scenarios: Avg. CO_2 Emissions (Kg)

Penalty		1	10	50	100	500	1000
Network	Time-Window						
0	Wide	2927	2927	2896	2767.1	2767.1	2767.1
0	Medium	2927	2927	2886.4	2767.1	2767.1	2767.1
0	Narrow	2927	2927	2833.6	2791	2791.9	2791.9
1	Wide	4126.6	4128	4051.7	3919.2	3813.9	3811.2
1	Medium	4131.2	4131.2	3993.2	3965.3	3949.9	3962.1
1	Narrow	4161.3	4161.3	4056.2	4033.1	4023.6	4037.8
2	Wide	2849.7	2849.7	2765	2632.5	2632	2632
2	Medium	2849.7	2849.7	2697.4	2632.5	2632	2632
2	Narrow	2849.7	2849.7	2711	2688.1	2687.4	2687.9
3	Wide	4147.1	4138.1	3925.5	3668.2	3658.5	3654.3
3	Medium	4154.8	4143.3	3900.7	3903.5	3583.9	3580.8
3	Narrow	4203.6	4203.6	4090.6	3993.1	3773.7	3759
4	Wide	2797.3	2797.3	2633.7	2609.1	2609.1	2609.1
4	Medium	2797.3	2797.3	2646.1	2633.2	2625.8	2625.8
4	Narrow	2834.6	2835	2782.9	2771.9	2761.4	2753.5
5	Wide	2467	2467	2272.1	2167.8	2166.3	2166.3
5	Medium	2467	2467	2272.1	2180.6	2179.1	2179.1
5	Narrow	2467	2467	2321.7	2235.1	2263.5	2270.1
6	Wide	3853.2	3836	3788.4	3589.2	3512.7	3683.1
6	Medium	3816	3838.7	3702.7	3688.3	3684.6	3684.6
6	Narrow	3889	3893.5	3797.5	3741	3764.5	3764.5
7	Wide	2973.7	2973.7	2856.4	2830.6	2830.6	2830.6
7	Medium	2969.1	2969.1	2858.8	2830.6	2830.6	2830.6
7	Narrow	3006.2	3006.2	2907.7	2907.7	2880.4	2872.7
8	Wide	3785.8	3785.8	3457.9	3447.2	3457.6	3448.8
8	Medium	3751.8	3751.1	3557.6	3551.3	3504.3	3479.5
8	Narrow	3711.2	3719.4	3623.2	3646.9	3599.8	3584.7
9	Wide	2826.2	2826.2	2817.6	2633.6	2661.7	2661.7
9	Medium	2826.2	2826.2	2733.7	2633.6	2661.7	2638.7
9	Narrow	2826.2	2826.2	2690.5	2692.2	2686.4	2673.3

Table 5. Static Network Scenarios: Avg. Duration (h)

Penalty		1	10	50	100	500	1000
Network	Time-Window						
0	Wide	30.6	30.6	31	33.1	33.1	33.1
0	Medium	30.6	30.6	31.1	33.1	33.1	33.1
0	Narrow	30.6	30.6	31.9	32.5	32.6	32.6
1	Wide	48.4	48.4	49.3	51.2	53.3	53.9
1	Medium	48.3	48.3	50	50.5	51.2	51
1	Narrow	51.5	51.5	52.9	53.3	53.8	53.5
2	Wide	30.1	30.1	31.1	32.9	32.9	32.9
2	Medium	30.1	30.1	32	32.9	32.9	32.9
2	Narrow	30.1	30.1	31.8	32.3	32.3	32.3
3	Wide	49.3	49.4	53.2	59.9	61	67.6
3	Medium	49.1	49.3	53.1	53.2	68.1	68.2
3	Narrow	50	50	51.9	54.1	69.4	70.8
4	Wide	29.7	29.7	32.2	33.1	33.1	33.1
4	Medium	29.7	29.7	32	32.4	32.7	32.7
4	Narrow	34.7	34.8	35.3	35.7	36.7	37.3
5	Wide	26.9	26.9	29.4	32.4	32.5	32.5
5	Medium	26.9	26.9	29.4	32.2	32.3	32.3
5	Narrow	26.9	26.9	28.8	31.2	32.2	32.5
6	Wide	46.2	45.9	46.8	50.6	52	48.9
6	Medium	46.5	46.2	48.2	48.7	49.2	49.2
6	Narrow	49.2	49.2	50.3	51.3	50.2	50.2
7	Wide	30.9	30.9	32.6	32.6	32.6	32.6
7	Medium	31	31	32.5	32.6	32.6	32.6
7	Narrow	32.6	32.6	33.3	33.3	34.7	35.2
8	Wide	45.9	45.9	50.3	50.7	51	51.4
8	Medium	46.4	46.4	48.9	49.3	50.9	52.6
8	Narrow	50.1	50.1	50.7	51.2	54	54.7
9	Wide	29.9	29.9	30	32.4	32.7	32.7
9	Medium	29.9	29.9	31.1	32.4	32.7	39.9
9	Narrow	29.9	29.9	31.7	31.7	32.1	35.7

Table 6. Static Network Scenarios: Avg. Nonpenalized Cost ($)

Penalty		1	10	50	100	500	1000
Network	Time-Window						
0	Wide	2664.3	2664.3	2676.5	2748.4	2748.4	2748.4
0	Medium	2664.3	2664.3	2679.7	2748.4	2748.4	2748.4
0	Narrow	2664.3	2664.3	2702.8	2723.5	2729.3	2729.3
1	Wide	4042.4	4042.1	4067.4	4128.6	4208	4238.4
1	Medium	4041.3	4041.3	4088.7	4105.1	4139.3	4132.4
1	Narrow	4226.7	4226.7	4264.4	4282.7	4305.4	4291.9
2	Wide	2608.7	2608.7	2638.3	2692	2692.9	2692.9
2	Medium	2608.7	2608.7	2661.1	2692	2692.9	2692.9
2	Narrow	2608.7	2608.7	2656.3	2673.8	2675.2	2678.1
3	Wide	4098.7	4103.8	4240.2	4519.2	4574.4	4933.4
3	Medium	4093	4097.4	4224.9	4233.3	4941.2	4945.6
3	Narrow	4155.8	4155.8	4222.2	4310.3	5072.9	5146.7
4	Wide	2571.3	2571.3	2653	2694	2694	2694
4	Medium	2571.3	2571.3	2646.2	2661.6	2676.4	2676.4
4	Narrow	2859.3	2862.2	2874.7	2891.3	2940.9	2969.5
5	Wide	2308.6	2308.6	2379.8	2507.3	2510	2510
5	Medium	2308.6	2308.6	2379.8	2500.6	2503.4	2503.4
5	Narrow	2308.6	2308.6	2360.9	2464.9	2529	2545.2
6	Wide	3828.7	3808.5	3844	3982.7	4035.2	3920.2
6	Medium	3835.1	3825.9	3887.1	3912.1	3939.9	3939.9
6	Narrow	4009.9	4008.7	4035.4	4070.8	4021.9	4021.9
7	Wide	2698.2	2698.2	2746.8	2738.1	2738.1	2738.1
7	Medium	2699.4	2699.4	2744.6	2738.1	2738.1	2738.1
7	Narrow	2798.2	2798.2	2804.8	2804.8	2872.6	2894.7
8	Wide	3792.9	3792.9	3919.7	3937.8	3960.1	3980.6
8	Medium	3806.8	3806.9	3880.1	3898.5	3971.3	4057.2
8	Narrow	3994.9	3997.7	3998.3	4037.5	4173.6	4205.7
9	Wide	2592	2592	2594	2663.9	2688	2688
9	Medium	2592	2592	2626.3	2663.9	2688	3077.9
9	Narrow	2592	2592	2646.7	2644	2664.6	2856.3

Table 7. Static Network Scenarios: Avg. Running Time (s)

Penalty		1	10	50	100	500	1000
Network	Time-Window						
0	Wide	0.4	0.4	1.1	2.2	8.8	13.3
0	Medium	0.3	0.3	0.3	0.9	3.6	5.6
0	Narrow	0.2	0.2	0.2	0.3	1	1.5
1	Wide	1.6	1.6	2.8	7.5	32.4	40.3
1	Medium	1.1	1.1	1.6	3.6	12.3	15.8
1	Narrow	0.4	0.5	0.7	0.9	3.2	4.3
2	Wide	0.2	0.2	0.6	1.6	6.4	9.4
2	Medium	0.1	0.1	0.2	0.7	2.5	3.9
2	Narrow	0.1	0.1	0.1	0.2	0.6	0.9
3	Wide	3.7	3.4	6.1	10	22.1	26.9
3	Medium	5.1	5.2	6.5	12.7	27.5	27.4
3	Narrow	1.7	1.7	2.1	3.3	6.4	7.2
4	Wide	2	1.6	1.4	1.5	3.3	3.4
4	Medium	1.2	1	0.8	1	1.8	2
4	Narrow	0.8	0.7	0.6	0.6	0.9	0.9
5	Wide	0.4	0.3	0.5	1.8	5.8	7.1
5	Medium	0.3	0.4	0.5	1	3.2	4.4
5	Narrow	0.3	0.3	0.4	0.5	1.4	2
6	Wide	12.4	10.9	16.1	33.1	96.1	126.9
6	Medium	6	6	9.2	16.3	38.5	50.3
6	Narrow	1.7	1.7	2.8	4.2	9	12
7	Wide	0.2	0.2	0.5	0.6	2	2.7
7	Medium	0.2	0.2	0.3	0.4	0.9	1.3
7	Narrow	0.2	0.2	0.2	0.2	0.4	1
8	Wide	3.4	2.7	3.6	6.6	14.5	16.8
8	Medium	1.8	1.7	2.3	3.6	7.5	7.7
8	Narrow	0.9	0.9	1	1.4	2.3	2.6
9	Wide	0.5	0.5	0.9	2.1	9	11.5
9	Medium	0.3	0.3	0.4	1.1	3.6	5.1
9	Narrow	0.3	0.3	0.3	0.4	1	1.5

Lower Bound Impact

The experiments for both lower bounds used sampling intervals of 0.5h and 5h for the decision spaces of the rollout policy and base policy, respectively. Figure 19 shows the running time obtained using each lower bound. Bound 2 showed significant performance improvements. Figure 20 shows the ratio between the running time from bounds 2 and 1 on the left, and the

ratio between the objective function values on the right. Half of the instances showed a reduction of at least 76% in running time.

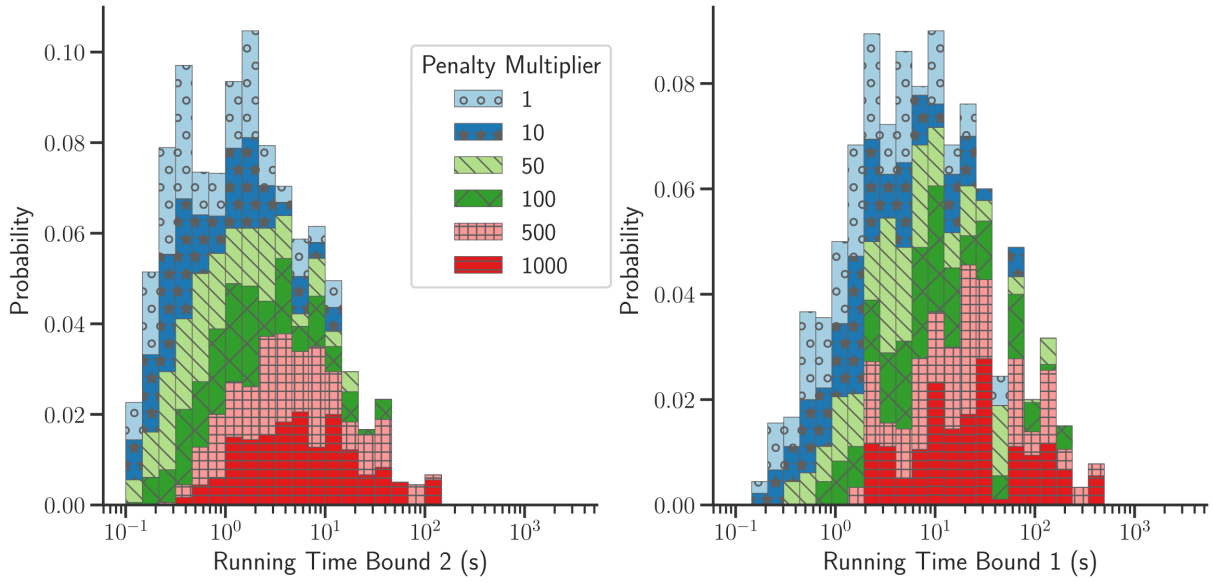


Figure 19. Histogram of the running time of simulations performed with bounds 1 and 2 separated by the penalty multiplier used.

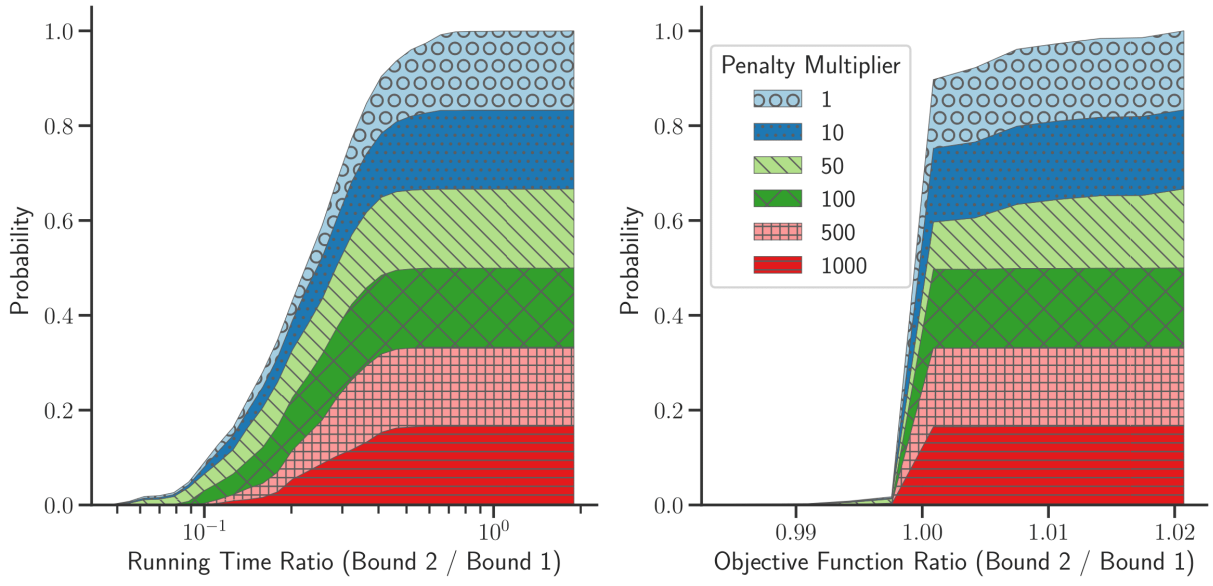


Figure 20. Cumulative distribution of the ratio between results (left: running time, right: objective function) obtained with bounds 2 and 1.

Time-dependent Networks

Detailed Results

Table 8. Time-dependent Scenarios: Avg. CO_2 Emissions (Kg)

Penalty		1	10	50	100	500	1000
Network	**Time-Window**						
0	**Wide**	2796.7	2796.7	2763.5	2754.4	2554.4	2559.7
0	**Medium**	2802.3	2798	2745.7	2745.6	2617.1	2573
0	**Narrow**	2798.1	2798.1	2733.5	2729.9	2639.3	2612.1
1	**Wide**	3927.8	3915.1	3842.2	3806.4	3780.5	3741.7
1	**Medium**	3916.3	3903.9	3833.4	3846.1	3792.1	3776.3
1	**Narrow**	3995.1	3995.1	3892	3881.1	3873.6	3817
2	**Wide**	2680.6	2682.6	2639.5	2632.8	2504	2498.8
2	**Medium**	2688.2	2683.3	2647.3	2637.8	2531.1	2504.4
2	**Narrow**	2745.2	2745.2	2698.9	2682	2647.2	2572.6
3	**Wide**	3992	3992	3833.1	3433.3	3418.4	3418.4
3	**Medium**	3958.5	3938.3	3835.4	3753.1	3440.8	3447.6
3	**Narrow**	3868.2	3858.7	3802.3	3794.7	3519.9	3521.8
4	**Wide**	2700.1	2699.9	2694.4	2639	2492.4	2476.4
4	**Medium**	2712.4	2710.9	2693	2702.4	2498.5	2525.5
4	**Narrow**	2711	2711	2688.4	2668.4	2564.9	2573.2
5	**Wide**	2360.2	2355.5	2264.2	2164.9	2156	2156
5	**Medium**	2316.4	2316.4	2255.6	2170.6	2162.1	2162.1
5	**Narrow**	2310.9	2307.4	2259.9	2193.9	2205.6	2199.5
6	**Wide**	3600.7	3600.7	3547.3	3412.5	3423.9	3421.2
6	**Medium**	3595.8	3595.8	3531.9	3416.8	3445.7	3425.2
6	**Narrow**	3631.2	3631.4	3551.7	3530.5	3523.8	3487.7
7	**Wide**	2856.8	2825.9	2822.9	2822.9	2666.7	2665.3
7	**Medium**	2851.4	2839.1	2832.1	2832.2	2666.5	2665.3
7	**Narrow**	2916.1	2914.3	2825.4	2797	2745.1	2724.6
8	**Wide**	3606.7	3593.5	3433.5	3433.2	3431	3337
8	**Medium**	3599	3591.5	3441.3	3443.6	3444.9	3365.9
8	**Narrow**	3624.6	3622.5	3504.8	3481.7	3489.4	3465.1
9	**Wide**	2661.9	2659	2598.3	2598.3	2604.3	2484.2
9	**Medium**	2660.9	2660.9	2603.3	2603.3	2604.1	2522.7
9	**Narrow**	2663	2662.7	2621.9	2628.9	2612.3	2588.9

Table 9. Time-dependent Scenarios: Avg. Duration (h)

Penalty		1	10	50	100	500	1000
Network	Time-Window						
0	Wide	32.8	32.8	33.3	33.4	49.2	49.1
0	Medium	33.2	33.2	33.8	33.8	44.8	50.6
0	Narrow	39.4	39.4	40.3	40.6	47.6	50.1
1	Wide	51.1	51.2	52.3	53.3	56.4	60.4
1	Medium	51.5	51.6	52.8	52.7	56.5	59.1
1	Narrow	58.1	58.1	59.1	59.7	62.1	70.3
2	Wide	31.9	31.9	32.6	32.8	50.6	52.6
2	Medium	32.1	32.2	32.7	33	48.9	52.4
2	Narrow	37.8	37.8	38.4	38.9	42.2	52.8
3	Wide	51.7	51.7	53.8	64.4	67.5	67.6
3	Medium	52.2	52.3	53.8	56.5	71.7	72
3	Narrow	58.4	58.4	59	59.5	75.8	76.5
4	Wide	32.6	32.6	33	33.9	44.5	47.7
4	Medium	32.7	32.7	32.9	32.8	50.2	49
4	Narrow	43.3	43.3	43.5	43.9	50.4	51.2
5	Wide	28.2	28.3	29.4	32.1	33.4	33.4
5	Medium	28.7	28.7	29.7	32	32.8	32.8
5	Narrow	29.1	29.1	29.9	31.9	32.1	32.5
6	Wide	49	49	49.8	53.1	54.4	55
6	Medium	49.1	49.1	50.2	53	54.9	60.6
6	Narrow	51.7	51.7	52.9	53.5	57.3	67.6
7	Wide	34.1	34.2	34	33.8	51.7	52
7	Medium	34.7	34.8	34.9	34.9	51.8	52
7	Narrow	46.3	46.3	47.9	48.7	51.3	55.2
8	Wide	48	48.1	50.4	50.4	50.5	70.8
8	Medium	48.1	48.2	50.4	50.5	51.1	65.4
8	Narrow	49.9	49.8	51.4	51.8	53.5	57.8
9	Wide	31.8	31.8	32.8	32.8	32.9	48.4
9	Medium	31.8	31.8	32.7	32.7	33.1	43.6
9	Narrow	32.6	32.6	33	33	35	40.3

Table 10. Time-dependent Scenarios: Avg. Nonpenalized Cost ($)

Penalty		1	10	50	100	500	1000
Network	Time-Window						
0	Wide	2740.9	2740.9	2754.6	2757.9	3557.5	3551.1
0	Medium	2763.5	2765	2775.4	2775.7	3336.8	3640.2
0	Narrow	3100	3100	3128.9	3144.9	3495.2	3622.9
1	Wide	4122.9	4125.1	4161.6	4201.7	4367.1	4570.7
1	Medium	4144.3	4146	4183.3	4184.9	4373.8	4511.4
1	Narrow	4531.7	4531.7	4552.2	4578.2	4707	5136.1
2	Wide	2654.6	2655.2	2676.1	2684.3	3617.7	3722.5
2	Medium	2667.8	2667.6	2686.3	2697.8	3533.7	3713.6
2	Narrow	2997.9	2997.9	3012.2	3035.5	3205.8	3761.1
3	Wide	4179.6	4179.6	4240.8	4686.3	4852.5	4854.3
3	Medium	4192.5	4193.7	4239.6	4363.3	5085.8	5108.1
3	Narrow	4501.4	4500.5	4515.6	4539.6	5338.5	5377.4
4	Wide	2696.3	2698.4	2718.8	2750	3276.1	3448.2
4	Medium	2709	2706.9	2713.4	2710.6	3593.6	3535.1
4	Narrow	3287.3	3287.3	3291.8	3307.1	3625.9	3672.6
5	Wide	2342.9	2343.6	2376	2489.8	2556.4	2556.4
5	Medium	2351.5	2351.5	2388.7	2487	2523.5	2523.5
5	Narrow	2372	2372.3	2400.7	2487.6	2502.7	2523
6	Wide	3898.9	3898.9	3926.4	4060.5	4132.3	4164.8
6	Medium	3904.6	3904.6	3942.2	4054.6	4167.5	4472.3
6	Narrow	4054.7	4056.1	4095.8	4123.2	4326.2	4880.5
7	Wide	2830.8	2829.4	2815.6	2802.3	3732.7	3749.7
7	Medium	2863.2	2865.3	2864.9	2864.9	3734.4	3749.7
7	Narrow	3519.2	3519.4	3574.4	3612.2	3738.5	3944.9
8	Wide	3843.8	3846.6	3919.2	3919.1	3924	5002.4
8	Medium	3848.1	3849.4	3919.4	3929	3961.3	4718.6
8	Narrow	3955.6	3951.5	3995.4	4012.1	4108.6	4336.8
9	Wide	2638.5	2638.5	2674.4	2674.4	2683.1	3489.2
9	Medium	2639.2	2639.2	2671.1	2671.1	2689.6	3237.5
9	Narrow	2682	2686.6	2692.9	2696.9	2800	3081.2

Table 11. Time-dependent Scenarios: Avg. Running Time (s)

Penalty Network	Time-Window	1	10	50	100	500	1000
0	**Wide**	17.1	11.1	19.5	37.8	53.8	72.3
0	**Medium**	12.3	6.7	9.7	14.3	21.9	33
0	**Narrow**	14.9	9.3	8.8	9	12.2	18.8
1	**Wide**	16	4.5	6.7	16.5	49.6	84.5
1	**Medium**	12.8	2.3	2.4	10.1	36.9	36.5
1	**Narrow**	52.8	20.2	16.5	23.8	30.8	41.2
2	**Wide**	4.1	1.6	1.7	4.6	21.1	13.4
2	**Medium**	1.2	1	1.2	2.1	9	5.6
2	**Narrow**	5.7	1.8	1.8	3.2	4.2	5.6
3	**Wide**	329.2	16.7	26.3	39.4	43.2	33
3	**Medium**	96.1	7.9	12	31.9	46.3	26.6
3	**Narrow**	40.9	11.6	13.5	15.6	20.8	22.8
4	**Wide**	41.1	6.9	7.6	14	15.7	21.2
4	**Medium**	18	3.3	4	5.9	10.6	12.1
4	**Narrow**	11	6.2	4.7	5.3	5.6	7.1
5	**Wide**	4.4	1.8	2.5	3.9	3.8	4.3
5	**Medium**	2.8	1	1.6	2.6	3	3.3
5	**Narrow**	2.3	0.7	0.7	1.2	1.6	1.6
6	**Wide**	98.8	14.2	17.9	46.2	98.3	59.9
6	**Medium**	26.6	6.5	7.8	21.4	106.7	127.3
6	**Narrow**	11.1	9.2	7.9	13.7	57.5	64.2
7	**Wide**	1.8	3.6	4	3.8	18.3	22.4
7	**Medium**	1.5	2.2	2.4	2.5	7.8	10.4
7	**Narrow**	6.9	4.3	4.8	5.3	5.4	5.5
8	**Wide**	130.9	10.5	14.8	21.7	143.1	238.9
8	**Medium**	26	4.9	6.7	15.3	50.2	67.9
8	**Narrow**	10.9	7.6	7.1	11.2	25.1	32.9
9	**Wide**	7.6	2.8	3.7	7.6	11.7	22.7
9	**Medium**	3.3	1.7	2.5	4.7	5.7	10.1
9	**Narrow**	1.9	1.7	1.7	2.9	3.7	6.3

Lower bound impact

The experiments ran using the lower bound from Bound 1 used sampling intervals of 0.5h and 5h for the decision spaces of the rollout policy and base policy, respectively. Figure 21 shows the running time obtained using each lower bound. Bound 2 showed significant performance

improvements even though the experiments used higher resolutions. The improvement is focused mostly on the higher penalty instances, which were also the slowest ones. Figure 22 shows the ratio between the running time from bounds 2 and 1 on the left, and the ratio between the objective function values on the right. While lower penalty instances showed some increase in running time, the performance of around 80% of all instances improved, especially for higher penalty values, the ratio is mostly lower than 1. The objective function ratio shows that most instances showed improvements in the solution, which is expected as the experiments using bound 2 use a higher resolution. However, some instances showed a slightly worse solution. This is caused by the differences in search space due to using different sampling intervals for the decision space.

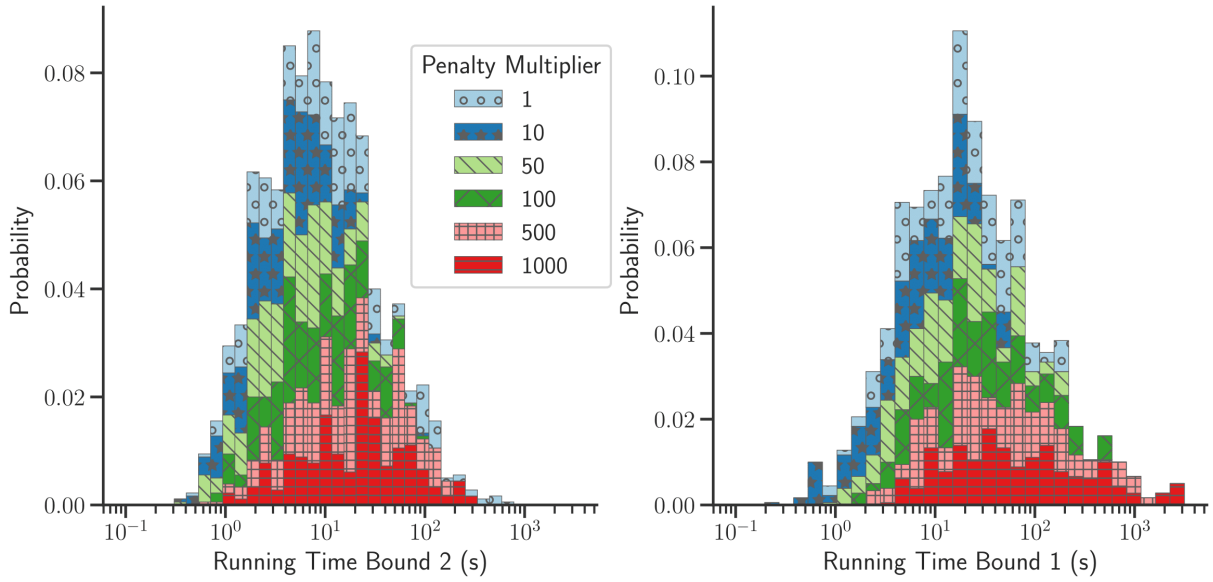


Figure 21. Histogram of the running time of simulations performed with bounds 1 and 2 separated by the penalty multiplier used.

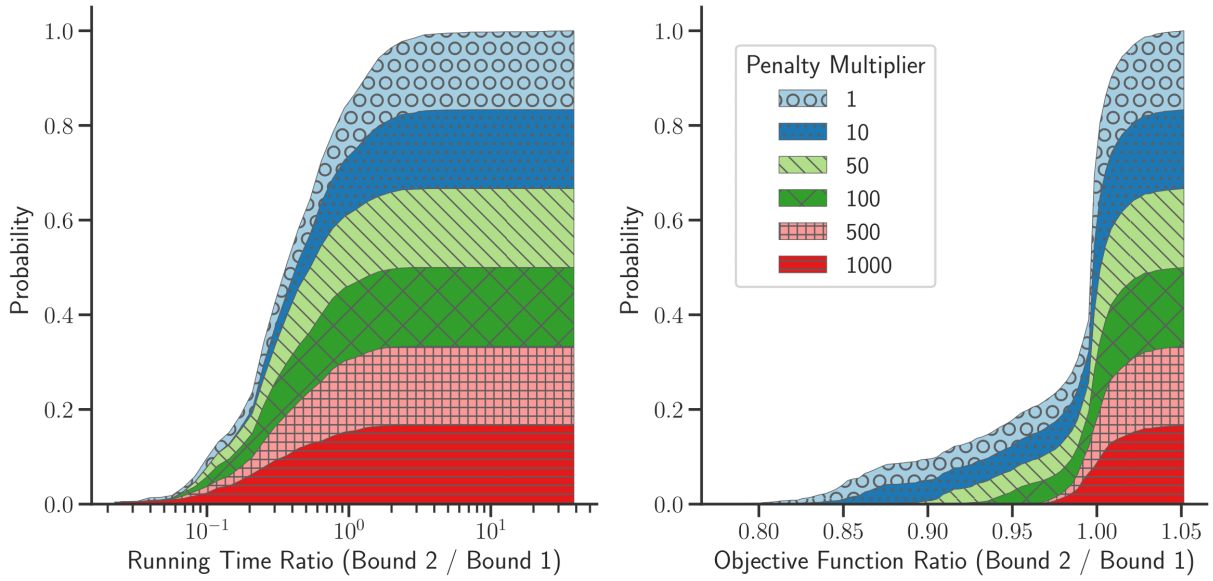


Figure 22. Cumulative distribution of the ratio between results (left: running time, right: objective function) obtained with bounds 2 and 1.

Stochastic Scenarios

Table 12. Stochastic Scenarios 1-3: Average Duration/Emissions/Cost

		Scenario	0	1	2	3
Field	**Net**	**Penalty**				
Trip Duration (h)	0	1	35.923	45.103	46.784	48.500
		50	35.923	45.103	47.022	55.796
		500	48.026	47.075	54.000	56.000
	2	1	40.831	46.000	46.000	46.500
		50	42.450	46.882	46.500	46.500
		500	47.726	47.500	53.500	54.000
	4	1	40.275	45.518	46.000	46.500
		50	42.173	45.518	46.000	46.500
		500	44.962	45.537	50.990	50.000
	5	1	36.796	47.841	63.135	46.000
		50	36.889	46.224	63.135	46.000
		500	46.166	48.485	58.215	73.025
CO_2 (kg)	0	1	2,814.387	2,791.348	2,842.667	2,935.984
		50	2,812.992	2,791.332	2,826.517	2,802.933
		500	2,705.267	2,762.947	2,706.953	2,765.926
	2	1	2,796.374	2,695.222	2,738.839	2,794.807
		50	2,674.378	2,627.899	2,688.314	2,794.807
		500	2,543.236	2,615.785	2,586.881	2,695.069
	4	1	2,759.585	2,634.713	2,674.284	2,681.677
		50	2,628.449	2,634.713	2,674.284	2,681.677
		500	2,548.114	2,634.560	2,624.490	2,642.914
	5	1	2,817.781	2,857.288	2,979.517	2,878.386
		50	2,811.225	2,681.212	3,002.406	2,878.386
		500	2,649.735	2,659.140	2,709.030	2,868.101
Non Penalized Cost	0	1	2,962.632	3,960.290	4,068.968	4,445.840
		50	2,966.606	3,960.304	4,056.589	4,518.105
		500	3,587.345	4,068.575	4,441.346	4,838.189
	2	1	3,200.070	3,918.177	3,844.269	3,770.075
		50	3,248.139	3,768.851	3,564.817	3,770.075
		500	3,524.830	3,798.615	4,020.220	4,464.673
	4	1	3,157.618	3,853.365	3,650.251	3,488.688
		50	3,227.038	3,853.365	3,650.251	3,488.688
		500	3,361.747	3,854.353	3,951.879	4,180.341
	5	1	2,994.704	4,025.289	5,208.110	3,910.644
		50	2,997.580	3,668.034	5,066.641	3,910.644
		500	3,464.060	3,768.341	4,641.981	6,263.692

Table 13 Stochastic Scenarios 1-3: Parking-related Performance

		Scenario	0	1	2	3
Field	**Net**	**Penalty**				
Illegal Parking (Events)	0	1	0.159	1.687	1.788	1.950
		50	0.159	1.687	1.743	1.305
		500	0.167	1.687	1.410	1.950
	2	1	0.100	1.180	1.180	0.885
		50	0.100	1.266	0.316	0.885
		500	0.150	1.266	0.510	1.950
	4	1	0.100	1.230	1.036	0.109
		50	0.100	1.230	1.036	0.109
		500	0.150	1.230	0.690	1.180
	5	1	0.150	1.986	3.135	1.000
		50	0.150	1.068	2.598	1.000
		500	0.150	1.050	1.410	2.850
Illegal Parking Duration (h)	0	1	1.720	16.605	17.023	14.630
		50	2.165	16.605	16.520	19.077
		500	1.379	17.592	13.370	15.810
	2	1	1.000	13.225	14.135	10.739
		50	1.025	12.261	3.961	10.739
		500	1.025	12.261	4.180	15.741
	4	1	1.000	12.666	10.577	1.297
		50	1.000	12.666	10.577	1.297
		500	1.025	12.666	5.980	16.276
	5	1	0.669	23.025	33.514	13.930
		50	0.669	10.313	28.043	13.930
		500	1.602	10.089	17.467	42.349
HOS Violation Risk (Events)	0	1	0.093	0.917	0.838	1.900
		50	0.093	0.917	0.793	1.037
		500	0.061	0.917	1.410	1.950
	2	1	0.000	1.180	0.950	0.475
		50	0.003	0.316	0.266	0.475
		500	0.150	0.316	0.510	1.000
	4	1	0.003	1.000	0.086	0.059
		50	0.050	1.000	0.086	0.059
		500	0.097	1.000	0.690	1.180
	5	1	0.048	0.048	0.835	0.950
		50	0.048	0.096	0.299	0.950
		500	0.053	0.050	1.410	2.850

		Scenario	0	1	2	3
Field	**Net**	**Penalty**				
Illegal Parking Ratio (%)	0	1	5.300	56.240	59.600	65.000
		50	5.300	56.240	58.100	43.511
		500	5.570	56.240	47.000	65.000
	2	1	5.000	59.000	59.000	44.240
		50	5.000	42.200	15.800	44.240
		500	5.000	42.200	17.000	65.000
	4	1	5.000	41.000	51.800	5.450
		50	5.000	41.000	51.800	5.450
		500	5.000	41.000	23.000	59.000
	5	1	5.000	66.200	80.380	50.000
		50	5.000	35.600	66.967	50.000
		500	5.000	35.000	47.000	95.000

Table 14. Stochastic Scenarios 4-6: Average Duration/Emissions/Cost

		Scenario	0	4	5	6
Field	**Net**	**Penalty**				
Trip Duration (h)	0	1	35.923	46.103	47.286	50.656
		50	35.923	46.103	50.674	50.702
		500	48.026	52.000	50.699	51.325
	2	1	40.831	47.000	46.000	47.000
		50	42.450	46.000	46.000	47.000
		500	47.726	54.500	53.500	49.500
	4	1	40.275	45.500	46.500	46.500
		50	42.173	45.501	46.500	46.500
		500	44.962	46.093	50.000	50.000
	5	1	36.796	45.500	46.000	46.500
		50	36.889	46.703	46.000	46.500
		500	46.166	54.500	53.898	51.000
CO_2 (kg)	0	1	2,814.387	2,775.880	2,881.018	2,776.022
		50	2,812.992	2,709.792	2,715.501	2,772.575
		500	2,705.267	2,678.361	2,713.916	2,727.823
	2	1	2,796.374	2,708.924	2,753.446	2,752.090
		50	2,674.378	2,666.435	2,751.924	2,731.147
		500	2,543.236	2,611.484	2,603.875	2,670.080
	4	1	2,759.585	2,617.445	2,659.258	2,711.199
		50	2,628.449	2,632.222	2,649.585	2,701.018
		500	2,548.114	2,619.457	2,621.383	2,625.068
	5	1	2,817.781	2,748.844	2,823.153	2,852.525
		50	2,811.225	2,656.418	2,690.156	2,851.039
		500	2,649.735	2,622.846	2,657.362	2,753.905
Non Penalized Cost	0	1	2,962.632	3,487.202	3,590.621	3,747.158
		50	2,966.606	3,489.009	3,736.394	3,749.099
		500	3,587.345	3,806.506	3,741.341	3,765.895
	2	1	3,200.070	3,509.734	3,476.040	3,523.365
		50	3,248.139	3,519.662	3,477.073	3,526.747
		500	3,524.830	3,919.881	3,953.703	3,727.539
	4	1	3,157.618	3,445.843	3,465.075	3,482.179
		50	3,227.038	3,450.396	3,471.309	3,488.680
		500	3,361.747	3,492.728	3,814.979	3,805.935
	5	1	2,994.704	3,448.064	3,492.528	3,531.138
		50	2,997.580	3,498.652	3,570.121	3,531.155
		500	3,464.060	3,921.490	4,069.101	3,763.994

Table 15. Stochastic Scenarios 4-6: Parking-related Performance

		Scenario	0	4	5	6
Field	**Net**	**Penalty**				
Illegal Parking (Events)	0	1	0.159	0.150	0.159	0.150
		50	0.159	0.150	0.159	0.150
		500	0.167	0.159	0.176	0.150
	2	1	0.100	0.100	0.100	0.100
		50	0.100	0.280	0.100	0.100
		500	0.150	0.150	0.510	0.271
	4	1	0.100	0.280	0.100	0.100
		50	0.100	0.282	0.100	0.100
		500	0.150	0.330	0.460	0.460
	5	1	0.150	0.100	0.100	0.100
		50	0.150	0.150	0.928	0.100
		500	0.150	0.150	0.510	0.100
Illegal Parking Duration (h)	0	1	1.720	1.027	1.255	2.058
		50	2.165	1.074	1.684	2.058
		500	1.379	1.416	1.911	1.864
	2	1	1.000	1.131	1.030	1.038
		50	1.025	3.145	1.148	1.038
		500	1.025	1.357	4.295	2.794
	4	1	1.000	3.145	1.035	1.038
		50	1.000	3.190	1.137	1.138
		500	1.025	3.214	5.865	5.328
	5	1	0.669	1.100	1.033	1.164
		50	0.669	1.142	4.997	1.163
		500	1.602	1.350	4.180	1.176
HOS Violation Risk (Events)	0	1	0.093	0.000	0.000	0.003
		50	0.093	0.098	0.050	0.005
		500	0.061	0.112	0.051	0.003
	2	1	0.000	0.000	0.048	0.000
		50	0.003	0.280	0.050	0.048
		500	0.150	0.100	0.269	0.271
	4	1	0.003	0.050	0.003	0.000
		50	0.050	0.052	0.050	0.050
		500	0.097	0.089	0.460	0.460
	5	1	0.048	0.050	0.000	0.000
		50	0.048	0.098	0.000	0.003
		500	0.053	0.100	0.510	0.100

		Scenario	0	4	5	6
Field	**Net**	**Penalty**				
Illegal Parking Ratio (%)	0	1	5.300	5.000	5.300	5.000
		50	5.300	5.000	5.300	5.000
		500	5.570	5.300	5.870	5.000
	2	1	5.000	5.000	5.000	5.000
		50	5.000	14.000	5.000	5.000
		500	5.000	5.000	17.000	13.550
	4	1	5.000	14.000	5.000	5.000
		50	5.000	13.423	5.000	5.000
		500	5.000	11.000	23.000	23.000
	5	1	5.000	5.000	5.000	5.000
		50	5.000	5.000	30.946	5.000
		500	5.000	5.000	17.000	5.000