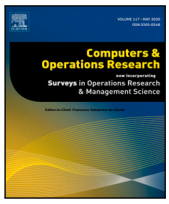
Computers & Operations Research 177 (2025) 107001

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor



Routing and driver break scheduling with working and driving regulations: A flexible approach for various general pickup and delivery problem variants

Ning Xue, Huan Jin, Tianxiang Cui *

School of Computer Science, University of Nottingham Ningbo China, 199 Taikang E Rd, Ningbo, 315100, China

ARTICLE INFO

Keywords:
General pickup and delivery problem
Driver break scheduling

ABSTRACT

This paper provides a solution to the driver break scheduling problem that can be integrated into current solvers for the generalised pick-up and delivery problem (GPDP) with a negligible increase in development costs. The driver break scheduling problem affects truck drivers who are required to adhere to working time and/or driving time regulations that oblige them to take mandatory breaks. These regulations are in force in many countries across the European Union, Great Britain, Australia, New Zealand, and the United States. We propose a flexible solution strategy that combines routing with driver break scheduling, taking into account both Regulations (EC)561/2006 and 2002/15/EC within a single day of operation. We deliberately incorporate driver breaks into the routes generated, with the aim of minimising route duration. The method proposed in this paper is particularly beneficial for the solvers that have not initially implemented driver break rules but need to introduce this functionality to accommodate customers, given the critical nature of driver break rules for effective scheduling. This is a flexible approach because the method proposed here can be applied directly to any individual route and at any stage of the scheduling process. For instance, it can be used during the constructive heuristic or the improvement heuristic phases. It is also applicable at the column generation stage of an exact method after a route has been constructed. This method is particularly beneficial when a human scheduler wishes to make manual adjustments along a single route, such as adding or removing an order from the route. Experimental analyses, grounded in two benchmark instances, affirm the efficacy of the proposed method. Results indicate that the method yields a 6.1% decrease in distance, a 1.7% reduction in duty time, and a 1.1% decrease in the number of unassigned tasks when contrasted with the standard compliant approach. We also offer guidance on how to tailor this method for driver break scheduling in accordance with the regulatory standards of Australia and New Zealand.

1. Introduction

The General Pickup and Delivery Problem (GPDP) (Savelsbergh and Sol, 1995) holds significant importance as it represents a complex class of Vehicle Routing Problems (VRP) that are integral to optimising logistics and transportation systems. This problem models real-world scenarios where a fleet of vehicles must pick up and deliver items between various locations, reflecting the operations of courier services, freight transport, and ride-sharing. Efficient solutions to GPDP can lead to substantial reductions in operational costs, improvements in delivery times, and enhanced customer satisfaction. Moreover, the complexity and NP-hard nature of GPDP drive research into advanced algorithm development, which is crucial for businesses looking to improve their logistics in the face of growing e-commerce demands and the need for sustainable operations. Solving GPDP effectively has a direct economic impact, aids in resource optimisation, and supports technological advancements in real-time routing and tracking systems.

Generally, GPDP can be viewed as a routing problem that models real-world situations where loads need to be transported from multiple origins to multiple destinations by a fleet of vehicles. Each load has a transportation volume, a pick up location and a deliver location. Each vehicle has a capacity, a start location, and an end location. The goal is to find the best routes for the vehicles to serve all the transportation requests without splitting any load. The Generalised Pickup and Delivery Problem with Time Windows (GPDPTW) extends the GPDP by incorporating additional constraints that limit the time windows within which pickups and deliveries must occur at each customer site, thereby further complicating the routing and scheduling challenges faced by logistics providers.

Drivers working time and break time should follow the driving and working regulations to ensure road safety. The integration of driver break scheduling with the GPDPTW holds immense significance in

* Corresponding author.
E-mail addresses: ning.xue@nottingham.edu.cn (N. Xue), huan.jin@nottingham.edu.cn (H. Jin), tianxiang.cui@nottingham.edu.cn (T. Cui).

https://doi.org/10.1016/j.cor.2025.107001
Received 28 August 2024; Received in revised form 20 November 2024; Accepted 28 January 2025
Available online 6 February 2025

the optimisation of logistics and transportation systems. As GPDPTW models real-world scenarios where fleets of vehicles are tasked with transporting goods between multiple origins and destinations, it inherently involves the management of driver schedules. Embedding driver break scheduling within GPDPTW algorithms ensures compliance with driving regulations, such as the EU Regulation (EC)561/2006 and Directive 2002/15/EC, which mandate working times and break periods for commercial vehicle drivers. By synchronising driver breaks with delivery schedules, the risk of driver fatigue and non-compliance with regulatory requirements is mitigated, thereby enhancing road safety and operational efficiency.

The provisions defined in both Regulations (EC)561/2006 and Directive 2002/15/EC are considered in this paper. However, due to the daily-focused nature of our customers' planning, where routes are typically planned only for the subsequent day, the paper addresses a case in which each truck driver's itinerary is limited to a single day of operation. Consequently, the focus is solely on scheduling driver **breaks** within that day. In other words, the planning does not incorporate extended **rests**, such as the 11-h daily rest or the 45-h weekly rest regulations, which extend beyond the confines of individual working days.

The Regulation (EC)561/2006, referred to as the driving regulation in this paper, sets forth standards for driving times and mandatory rest periods for commercial vehicle drivers, as described in Benus and Demirci (2020). According to this regulations, for a single day's plan, the following constraints should be met:

1. The maximum daily driving time is 9 h.
2. The rest periods could be one of the following:
    (a) For every 4.5 h of driving, drivers must take a break of at least 45 min. This break starts a new 4.5-h driving period.
    (b) For every 4.5 h driving, drivers can take either one 45-min break, or two smaller breaks, one of at least 15 min followed by another of at least 30 min.

Directive 2002/15/EC (Gibson et al., 2017), referred to as the duty regulation in this paper, is a legislative act that concerns the working time for drivers or mobile workers engaged in road transport activities. In this paper, for the sake of simplification, we use drivers to represent both drivers and mobile workers, assuming that drivers can perform not only driving, but also loading/unloading, and maintenance work. This regulation sets out the maximum working limits on working time, including driving time, other work-related activities, and on-call time. Duties of truck drivers include driving vehicles and other work-related duties such as loading and unloading, vehicle inspections and maintenance, and any other paid work.

According to this act, for a single day's plan, the following constraints should be met:

1. Drivers cannot work more than 6 h without a break. A break should be at least 15 min long.
2. Drivers need a 30-min break if they work 6 to 9 h in total.
3. Drivers need a 45-min break if they work more than 9 h in total.

As mentioned, these limits are intended to prevent fatigue and ensure that drivers have adequate time for rest and recuperation, and therefore improve road safety and ensure fair competition within the road transport sector across EU member states.

In this paper, we consider a GPDP in which the service of each customer must start within a specified time window and all tours must comply with the driver break regulations. We refer to this problem as the General Pickup and Delivery Problem with Time Windows and Driver Break Schedule (GPDPTW-DBS).

This rest of the paper is organised as follows: Section 2 reviews the related literature, discussing how our work differs from previous studies after summarising the key findings of existing research in the field. Section 3 defines the problem of integrating routing and driver break scheduling, considering both Regulations (EC)561/2006 and 2002/15/EC within a single day of operation. Section 4.1.1 presents the algorithm for generating driver breaks that comply with both regulations. Section 4.1.2 discusses some issues of inefficiency in generating compliant schedules and proposes a solution method to improve them in Section 4.1.3. Section 4.2 introduces a solution framework that integrates driver break scheduling with both local search and large neighbourhood search approaches. Section 5 reports the computational results of the proposed methods. Section 6 concludes the study.

2. Literature review

Archetti and Savelsbergh (2009) shows that it is hard to check the feasibility of a route after adding driver break regulations, let alone finding the optimal break patterns. Goel (2009) presents a labelling algorithm for (EC)561/2006 that verifies the feasibility of inserting or removing individual customers from a route. However, their approach does not account for the possibility of driver break splits (i.e., dividing the driver's break into two separate break periods).

Goel and Irnich (2017) exclusively focuses on the regulations set forth in (EC)561/2006 and proposes dominant rules for selecting the optimal routes and strategies within the column generation subproblem. Goel (2018) proposes a scheduling approach that focuses on finding a feasible schedule (break and rest activities) for a given route. Both Regulation (EC)561/2006 and Directive 2002/15/EC are considered, with the assumption that a break can only be either 45 min or 15 min followed by 30 min. Finding a schedule may need to evaluate many different sequences and durations of driver activities. In one of the authors' previous works (Goel, 2009), the driving periods are scheduled with the longest possible durations, and the off-duty periods are scheduled with the shortest possible durations required by the regulation. This study uses heuristics to generate vehicle tours that comply with some parts of the driver break and rest regulations; it only considers parts of the rules set by Regulation (EC)561/2006 and assumes that break splitting (i.e., a 45-min break split into 15 min followed by a 30-min period) is not allowed. Prescott-Gagnon et al. (2010) uses a labelling approach to find feasible partial paths with breaks and rests on the arcs. A label is a resource vector that shows the state of the path. The driving time between two customers cannot exceed 4.5 h, so no breaks are allowed between them. This study does not model driver activities in detail but only checks if a route is feasible when a customer is added using a procedure. Although this study takes into account both Regulation (EC)561/2006 and 2002/15/EC, it operates under the assumption that only a 45-min break is considered. The study does not account for the possibility of split breaks or rest periods that span across different days.

Goel and Irnich (2017) studies a more realistic version of the VRPTW that considers working-hour constraints. The paper explicitly models the driver activities and proposes a method to select the best candidate path without evaluating all possible combinations of driver activities (such as driving, waiting, breaking, etc.). The paper employs a parameter-free model for labelling to solve the pricing problem efficiently using the standard shortest path problems with resource constraints algorithm and the column generation approach. The same set of rules from Regulation (EC)561/2006 considered in Goel (2010) is applied. This approach is further improved by Tilk and Goel (2020), which utilises a bidirectional method. Sartori et al. (2022) introduces the truck driver scheduling problem with interdependent routes and resolves it by employing the label propagation algorithm. This algorithm is capable of scheduling interdependent routes in compliance with (EC)561/2006, with the assumption that break splitting is not allowed. The paper discusses potential extensions that would incorporate Directive 2002/15/EC and permit the splitting of driver breaks.

Some studies have explored the effect of delaying or advancing the vehicles' departure time, assuming that having compact duty time leads

to cost savings. These studies also consider regulation (EC)561/2006 in their models (Archetti and Savelsbergh, 2009; Kok et al., 2010).

It can be seen that most of the literature focuses on either the full set or part of the regulations of (EC)561/2006, with the assumption that driving constitutes the main activity of drivers and therefore complying with (EC)561/2006 is sufficient to satisfy 2002/15/EC.

Nonetheless, the option of break split has not been extensively explored, with the majority of researchers assuming a limited set of break patterns, such as a single 45-min break or a sequence comprising a 15-min break period followed by a 30-min break period. The later presumption is based on the provision of (EC)561/2006, which allows for a break to be split into two periods, the first being at least 15 min and the second at least 30 min.

In practice, however, the scheduling of breaks may exhibit greater flexibility, and the break patterns should be more varied to adapt to real-life situation. For instance, if a driver arrives at the customer's site 20 min ahead of the designated time window open due to more favourable traffic conditions than anticipated, this could present an opportunity to allocate a 15-min break; later on, another 15-min break period can be assigned due to a similar situation for this driver. Consider that these two 15-min breaks are allocated within a time span of 6 h; even though the two 15-min break periods are not considered a valid break under (EC)561/2006, they satisfy the rule set by 2002/15/EC, which requires a *30-min break if working between 6 and 9 h in total* and *a driver cannot work for more than 6 h without a break*.

In fact, some of those break patterns not considered in the literature mentioned above could be legitimate and are observed in real-world practices. For example, consider a route with three break periods: 30 min, 15 min, and 30 min. The first 30-min break period is required by Regulation (EC)2002/15. Obviously, the first two break periods (i.e., 30 min and 15 min) are not considered a valid break under Regulation (EC)561/2006 within the time span covering those breaks, because according to that regulation, in the case of two short breaks, one must be at least 15 min, followed by another of at least 30 min. However, if we move the time span afterward, the last two break periods (i.e., 15 min and 30 min) would be considered a valid break by Regulation (EC)561/2006. Therefore, in our study, a sequence of two short breaks consisting of 30 min and 15 min is allowed.

Furthermore, research on the interplay between driver breaks as mandated by Regulations (EC)561/2006 and No 2002/15/EC is somewhat limited. In practice, when scheduling breaks in compliance with both regulations independently, there is a risk of scheduling an excessive number of breaks. This study aims to address this issue by investigating methods to reduce the overall number and duration of driver breaks. Additionally, the paper offers strategies for promptly adjusting planned breaks in response to disruptions, such as the need for longer breaks or general delays. In response to these changes, reoptimising the entire delivery plan on a rolling basis could result in widespread adjustments to all drivers' schedules, a practice not favoured by drivers as this may lead to a sense of uncertainty and stress for the drivers. This paper introduces a solution that efficiently recalibrates only the remaining breaks for each route as necessary, ensuring compliance with both regulations (EC)561/2006 and 2002/15/EC. The primary objective of driver break scheduling is to minimise the frequency and cumulative duration of breaks while maintaining compliance with regulatory standards.

The literature review also reveals that many existing studies, such as Prescott-Gagnon et al. (2010) and Goel and Irnich (2017), employ forward labelling techniques to evaluate routes. In these studies, labels are used to represent the various states of a truck driver following a series of activities, including breaks, rest periods, driving times, and other work-related activities. These labels are updated after each activity through the use of a resource extension function (REF). As the algorithm progresses and nodes are updated (i.e., by extending labels along a route), the number of alternative labels/variables generated by the scheduling approach can increase rapidly, requiring additional procedures to delete all dominated labels. In contrast, our approach maintains a constant number of variables as node visits proceed. Similar to the strategy used in the research by Sartori et al. (2022), our paper proposes an alternative method that does not require the storage of activity data at each customer node. Instead, we dynamically update variables related to driver breaks as the route progresses and schedule breaks when certain criteria are met.

3. Problem definition

The GPDPTW-DBS is defined in a graph $G = (N, A)$, where $N = P \cup D \cup W$, and A is the set of arcs connecting pairs of nodes. Let P be the set of pickup nodes (i.e. locations) , and D be the set of delivery locations. Let M be the set of vehicles, where each vehicle $k \in M$ has a capacity Q_k, a start location k^+ and an end location k^-. Define $M^+ = \{k^+ | k \in M\}$ as the set of start locations and $M^- = \{k^- | k \in M\}$ as the set of end locations, such that $W = M^+ \cup M^-$. Every arc $(i, j) \in A, i, j \in N$ has an associated driving time δ_{ij} to travel from location i to j.

A customer request is a pair of nodes $(p, d), p \in P, d \in D$, such that a vehicle has to transport goods from p to d. Every node $i \in N$ has an associated time window $[e_i, l_i]$, where e_i is the earliest starting time and l_i is the latest starting time. A vehicle can arrive before e_i and wait to start the service, but cannot arrive later than l_i. Let q_i be the demand at node i, so that for each request (p, d), the demand $q_p \geq 0$ and $q_d = -q_p$.

A solution to the GPDPTW-DBS is a set $R = \{r_1, r_2, \ldots, r_{|R|}\}$ of routes meeting all (or most) requests. Assume route $r \in R$ is a sequence consisting of n visits $\{0, 1, 2, 3, \ldots, n-1\}$ and a driver break schedule B, which is a set containing the start time and duration of required breaks $\{(t_{ij}^1, l_{ij}^1), (t_{ij}^2, l_{ij}^2), \ldots, (t_{ij}^{|B|}, l_{ij}^{|B|})\}$. The solution must satisfy all time windows, ensure that the maximum capacity of the vehicle is never exceeded, require that each pickup is visited before the corresponding delivery location, and ensure that both pickup and delivery belong to the same route. Additionally, the vehicle must start from its start location and return to its end location.

In this paper, the objective is to find a set of routes to fulfil as much customer demands as possible (i.e. minimise the number of unassigned requests) and minimise the cost. The cost is measured by the total duty time, therefore, the route with n visits is $a_n - a_0$, where a_i indicates the arrival time of location i.

4. Methodology framework

This section comprises two key subsections, namely, Section 4.1 Generating Break Schedules and 4.2 Generating Full Solutions, which focus on integrating break schedules into routes. Section 4.2 delves into the methodology of generating comprehensive solutions encompassing vehicle routing and driver break schedules, offering a holistic perspective on the optimisation process.

4.1. Generating break schedules

This section discusses generating the break schedules complying with the two regulations, (EC)561/2006 and 2002/15/EC. The process of generating general compliant schedules is elucidated, ensuring adherence to both regulations. However, it is noted that such compliant schedules may lead to inefficient driver break assignments, causing longer route durations. To address this issue, we propose an advanced approach to enhance the efficiency of driver breaks and shorten route durations. Through the presentation of real-world examples, the inefficiencies inherent in compliant driver break schedules are highlighted, followed by the introduction of two strategies aimed at generating more advanced schedules.

Table 1
Notations used in the problem formulation.

Notation	Description
Variables	
t^{start}	duty start time.
a_i	arrival time at location i.
w_i	wait time at location i.
d_i	departure time from location i.
s_i	service time at location i.
δ_{ij}	driving time between locations i and j.
δ_{ij}^{latest}	latest driving time left to trigger a break between locations i and j.
δ_{ij}^{trip}	remaining driving time between locations i and j, i.e., the driving time before reaching location j.
L_{ij}	total break time within the leg from i to j.
t_{ij}^{m}	starting time of the m^{th} break; for clarity, we use 'drive' or 'duty' to distinguish the break generated by driving regulation ($t_{ij}^{m\|drive}$) or duty regulation ($t_{ij}^{m\|duty}$).
l_{ij}^{m}	length of the m^{th} break, where $m \in \mathbb{Z}^{+}$; similarly, for clarity, we use 'drive' or 'duty' to distinguish the break generated by driving regulation ($l_{ij}^{m\|drive}$) or duty regulation ($l_{ij}^{m\|duty}$).
B	the set of generated breaks of a route, i.e., $B = \{(t_{ij}^{0}, l_{ij}^{0}), (t_{ij}^{1}, l_{ij}^{1}), \dots\}$.
L^{duty}	accumulated working duration.
$L_i^{driving}$	accumulated driving duration at location i since last driving reset.
L_i^{break}	accumulated break duration at location i.
L_i^{tbreak}	the accumulated break since the last driving reset at location i; if $L_i^{tbreak} \geq 15$ min, the next driving break can be reduced to 30 min. Following this, a new driving period of up to 4.5 h can commence, which will reset the value of $L_i^{driving}$ to 0.
$L^{break\|work1st}$	the total break duration until the first threshold for assigning a duty break (i.e., $l^{break\|work1st}$).
$L^{break\|work2nd}$	the total break duration until the second threshold for assigning a duty break (i.e., $l^{break\|work2nd}$).
Data Constants	
l^{break}	the minimum duration of a break (set by default to 45 min).
$l^{break\|1st}$	the minimum duration of the first part of a break taken in two parts (set by default to 15 min).
$l^{break\|2nd}$	the minimum duration of the second part of a break taken in two parts (set by default to 30 min).
$dutyBreakRequirement1$	to encapsulate the regulation requiring a driver to take a 30-min break when working between 6 and 9 h in total (set by default to 30 min).
$dutyBreakRequirement2$	to account for the provision that mandates a 45-min break if working more than 9 h in total (set by default to 45 min).
$l^{break\|drive}$	the maximum driving time without a break (set by default to 270 min).
$l^{break\|work1st}$	the first threshold of assigning a duty break (set by default to 360 min).
$l^{break\|work2nd}$	the second threshold of assigning a duty break (set by default to 540 min).

4.1.1. Generating general compliant schedules

The route time scheduling algorithm handles the driver break requirements, which updates time-related information such as arrival time, waiting time, delay, and departure time for each visit in the route. This section discusses the formulas used in our scheduling algorithm as well as the logic of assigning driver breaks. All the notations can be found in Table 1.

Let {0,1,2,3, …, $n-1$} be a route consists of n visits and (i, j) be a leg in the route. We define the notations of the following variables:

We employ two constants to represent the driver break requirements as stipulated by the 2002/15/EC directive. To encapsulate the regulation that a driver needs a 30-min break if working between 6 and 9 h in total, we utilise the variable $dutyBreakRequirement1$ to denote this 30-min minimum break. Additionally, we introduce the constant $dutyBreakRequirement2$ to account for the provision that mandates a 45-min break if working more than 9 h in total.

Given the notations defined, the parameters for driving regulations (i.e., (EC) No 561/2006) and duty regulations (i.e., 2002/15/EC) are summarised in Table 1.

The arrival time at visit j depends only on the arrival time of its predecessor (a_i in above example) and waiting time, service time, driving time and break period occurred between the time range of a_i and a_j. Therefore, we can define recursive equations to compute a_i of any visit i:

$$d_i = a_i + w_i + s_i \tag{1}$$

$$a_j = d_i + \delta_{ij} + L_{ij} \tag{2}$$

The flow chart in Fig. 1 summarises the process of generating required breaks B along a leg (i, j), detailed algorithms are available in Algorithm 1 for assigning duty breaks, Algorithm 3 for assigning driving breaks, and Algorithm 4 for assigning both driving and duty breaks. With the determined arrival time a_i at visit i, we proceed to update the following variables: the departure time d_i, the arrival time at the subsequent visit a_j, the time to trigger the next drive break $t_{ij}^{m|driver}$ and the time to trigger the next duty break $t_{ij}^{m|duty}$. If the time required for a duty break or drive break exceeds the time of arrival at the next visit a_j, then no break is required along this leg. However, if a break is needed, either a duty break or a driving break is required along this leg. Arriving break is initiated immediately if the maximum driving time is reached, while the assignment of a duty break (Please refer to Algorithm 1 for detailed information) depends on the accumulated break duration from previous legs up to visit j.

More specifically, for a leg of (i, j) in the route, as long as $\delta_{ij}^{trip} > 0$, we need to generate breaks complying with both regulations and update time-related relevant variables. The logic of assigning breaks with the driving regulation (see Algorithm 3) is that: within the leg from i to j, if there is still driving time remaining, the driving time to trigger driving break (δ_{ij}^{latest}) for initiating a break is adjusted. This δ_{ij}^{latest} is calculated as the difference between the maximum allowable driving time without a break ($l^{break|drive}$) and the total duration of driving since the last break ($L_i^{driving}$). Subsequently, we assess whether the remaining driving time within the current leg is less than the δ_{ij}^{latest}. If this condition is met, the duration of driving since the last break is incremented by the remaining driving time in the current leg (i.e. δ_{ij}^{trip}) and the process of generating driving break is terminated here. Otherwise, a new driving break is generated. The duration of the new break is determined by whether the accumulated break time since the last driving reset (L_i^{tbreak}), exceeds or equals $l^{break|1st}$. If so, the duration of the new break is set to $l^{break|2nd}$; otherwise, it is set to l^{break}. With the creation of a new break, it becomes necessary to update the aforementioned time-related variables accordingly (denoted as **Update variable values** in Fig. 1), as they will be utilised to generate driver breaks during the subsequent legs.

The detailed logic for assigning duty breaks under 2002/15/EC is described in Algorithm 1 and 2, with variables updated according to Table 3. The pseudo code for assigning driving breaks under

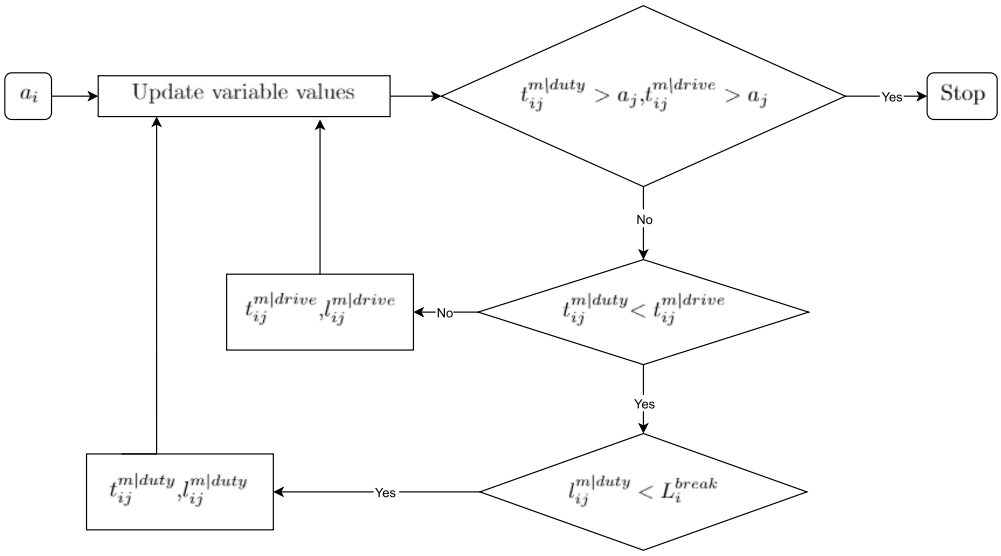


Fig. 1. Flow chart of scheduling breaks.

(EC)561/2006, i.e., for driving breaks, is available in Algorithm 3, with variables updated according to Table 2. Please consult Algorithm 4 below for the assignment of both duty and driving breaks.

Algorithm 1 Pseudo-code for *Assigning a Break for 2002/15/EC, i.e., the Duty Break*

if CheckDutyLimitAndAssignBreak($endOfPeriod$, $dutyBreakRequirement1$, $l^{break|2nd}$) **then** ▷ Algorithm 2
  $endOfPeriod$ += duration of the newly generated driver break
end if
CheckDutyLimitAndAssignBreak($endOfPeriod$, $dutyBreakRequirement2$, l^{break}) ▷ Algorithm 2

Algorithm 2 Pseudo-Code of CheckDutyLimitAndAssignBreak ($endOfPeriod$,$dutyBreakThreshold$,$breakRequired$)

if $L_i^{break} < breakRequired$ **and** $t^{start} + dutyBreakThreshold \leq endOfPeriod$ **then**
  $t_{ij}^{m|duty} = t^{start} + dutyBreakThreshold$
  $l_{ij}^{m|duty} = breakRequired - L_i^{break}$
  $B = B \cup \{(t_{ij}^{m|duty}, l_{ij}^{m|duty})\}$
  $L_{ij} = L_{ij} + l_{ij}^{m|duty}$
  $L_i^{break} = L_i^{break} + l_{ij}^{m|duty}$ ▷ New duty break
  $L_i'^{break} = L_i'^{break} + l_{ij}^{m|duty}$
  if $t_{ij}^{m|duty} \leq d_i$ **then**
    $d_i = d_i + l_{ij}^{m|duty}$
  else
    $\delta_{ij}^{latest} = t_{ij}^{m|duty} - d_i$
    $L_i^{driving} = L_i^{driving} + \delta_{ij}^{latest}$
    $\delta_{ij}^{trip} = \delta_{ij}^{trip} - \delta_{ij}^{latest}$
    $d_i = t_{ij}^{m|duty} + l_{ij}^{m|duty}$
  end if
  if $L_i^{break} \geq l^{break|2nd}$ **and** $L_i'^{break} \geq l^{break}$ **then**
    $L_i^{driving} = 0$
    $L_i'^{break} = 0$
  end if
  return true
else
  return false
end if

Algorithm 3 Pseudo-code for *Assigning a Break for (EC)561/2006*, i.e., the Driving Break

while $\delta_{ij}^{trip} > 0$ **do**
  $\delta_{ij}^{latest} = l^{break|drive} - L_i^{driving}$
  if $\delta_{ij}^{r} < \delta_{ij}^{latest}$ **then**
    $L_i^{driving} += \delta_{ij}^{m}$
    break ▷ Exit the loop
  end if
  if $L_i'^{break} \geq l^{break|1st}$ **then**
    $l_{ij}^{m|drive} = l^{break|2nd}$
  else
    $l_{ij}^{m|drive} = l^{break}$
  end if
  $t_{ij}^{m|drive} = d_i + \delta_{ij}^{latest}$
  $B = B \cup \{(t_{ij}^{m|drive}, l_{ij}^{m|drive})\}$
  $L_{ij} += l_{ij}^{m|drive}$
  $L_i^{break} += l_{ij}^{m|drive}$ ▷ New driver break
  $L_i'^{break} = 0$
  $L_i^{driving} = 0$
  $\delta_{ij}^{trip} -= \delta_{ij}^{latest}$
  $d_i += (\delta_{ij}^{latest} + l_{ij}^{m|drive})$
end while

Algorithm 4 Pseudo-Code of considering both *(EC)561/2006 and 2002/15/EC*

if $L_i^{break} < breakRequired$ **and** $t^{start} + dutyBreakThreshold \leq endOfPeriod$ **then**
  $endOfPeriod = \min(d_i + (l^{break|drive} - L_i^{driving}), d_i + \delta_{ij}^{trip})$
  Execute Algorithm 1
end if
Execute Algorithm 3

The flexibility of this method is evident in the pseudo code for assigning duty or driving breaks (refer to Algorithms 1 and 3). Provided that the necessary parameters, such as the duty break threshold and the break requirement, are specified, all relevant values can be updated iteratively at any stage of the solution process, without making any prior assumptions about the problem.

Table 2
Assign breaks according to the driving regulation (EC)561/2006.

Variable	Condition	After break
δ_{ij}^{latest}	–	$l^{break\|drive} - L_i^{driving}$
$L_i^{driving}$	$\delta_{ij}^{trip} < \delta_{ij}^{latest}$	$L_i^{driving} + \delta_{ij}^{m}$
$t_{ij}^{m\|drive}$	$\delta_{ij}^{trip} >= \delta_{ij}^{latest}$	if $L_i'^{break} >= l^{break\|1st}$ $t_{ij}^{m\|drive} = l^{break\|2nd}$ else $t_{ij}^{m\|drive} = l^{break}$
$t_{ij}^{m\|drive}$	$\delta_{ij}^{trip} >= \delta_{ij}^{latest}$	$d_i + \delta_{ij}^{latest}$
L_{ij}	$\delta_{ij}^{trip} >= \delta_{ij}^{latest}$	$L_{ij} += t_{ij}^{m\|drive}$
L_i^{break}	$\delta_{ij}^{trip} >= \delta_{ij}^{latest}$	$L_i^{break} + t_{ij}^{m\|drive}$
$L_i^{driving}$	$\delta_{ij}^{trip} >= \delta_{ij}^{latest}$	0
$L_i'^{break}$	$\delta_{ij}^{trip} >= \delta_{ij}^{latest}$	0
δ_{ij}^{trip}	$\delta_{ij}^{trip} >= \delta_{ij}^{latest}$	$\delta_{ij}^{trip} - \delta_{ij}^{latest}$
d_i	$\delta_{ij}^{trip} >= \delta_{ij}^{latest}$	$d_i + \delta_{ij}^{latest} + t_{ij}^{m\|drive}$

Table 3
Assign breaks according to the duty regulation 2002/15/EC.

Variable	Condition	After break
$t_{ij}^{m\|duty}$	–	$t^{start} + l^{break\|work1st} or l^{break\|work2nd}$
$t_{ij}^{m\|duty}$	–	$l^{break\|2nd} or l^{break} - L_i^{break}$
L_i^{break}	–	$L_i^{break} + t_{ij}^{m\|duty}$
$L_i'^{break}$	–	$L_i'^{break} + t_{ij}^{m\|duty}$
d_i	$t_{ij}^{m\|duty} <= d_i$	$d_i + t_{ij}^{m\|duty}$
δ_{ij}^{latest}	$t_{ij}^{m\|duty} > d_i$	$t_{ij}^{m\|duty} - d_i$
$L_i^{driving}$	$t_{ij}^{m\|duty} > d_i$	$L_i^{driving} + \delta_{ij}^{latest}$
δ_{ij}^{trip}	$t_{ij}^{m\|duty} > d_i$	$\delta_{ij}^{trip} - \delta_{ij}^{latest}$
d_i	$t_{ij}^{m\|duty} > d_i$	$t_{ij}^{m\|duty} + t_{ij}^{m\|duty}$
$L_i^{driving}$	$L_i^{break} >= l^{break\|2nd}$ And $L_i'^{break} >= l^{break}$	0
$L_i'^{break}$	$L_i^{break} >= l^{break\|2nd}$ And $L_i'^{break} >= l^{break}$	0

The algorithms outlined earlier are designed for scheduling driver breaks in line with the (EC) 561/2006 regulation (refer to Algorithm 3), the 2002/15/EC regulation (refer to Algorithm 1), or both (refer to Algorithm 4). When implementing both regulations concurrently, the resulting driver break schedules are termed **compliant schedules**. These schedules adhere strictly to the stipulations of both regulations but may not optimise efficiency and could result in additional, unnecessary breaks. Section 4.1.2 will present several examples to clarify this issue. To address the issues and enhance the efficiency of break scheduling—aiming to reduce both the frequency and cumulative duration of breaks—a more sophisticated method is introduced. This advanced strategy will be examined in detail in Section 4.1.3.

4.1.2. Issues with compliant schedules

To provide a comprehensive understanding of how the overlapping regulations impact driver break scheduling, the 13 scenarios depicted in Fig. 2 are examined to clarify the interaction between the driver break regulations of (EC)561/2006 and 2002/15/EC. These scenarios show that resetting driving hours can potentially improve the effectiveness of driver breaks and decrease the total route duration.

- Scenario 1 (a) The driver takes a break of 30 min after working for 6 h. Then, the driver resumes driving for another 30 min, followed by another break of 30 min, because of the 4.5 h of driving time accumulated.
- Scenario 1 (b) However, this driver break assignment can be improved by allocating a 45-min break when the duty time has been accumulated to 6 h rather than allocating an additional 30-min break when reaching 4.5 h of driving because the driving hours have been reset by the 45-min break.
- Scenario 2 (a) This scenario is similar to the previous one, but the driver in the previous scenario completes his journey after 2.5 h of driving after the first break. In this case, however, the driver continues to drive for 4 h and 40 min after the first break.
- Scenario 2 (b) Allocating 45 min to reset the driving break is no longer a good idea in this scenario because another break will be triggered due to reaching 4.5 h driving time. As a result, the journey will take 12 h and 10 min.
- Scenario 3 (a) After working for 6 h, a break of 30 min is given. Then, driving is resumed for 2.5 h more and another break of 15 min is taken, as the duty time has reached 9 h. After one more hour of driving, another break of 30 min is required, because the total driving time is 4.5 h.
- Scenario 3 (b) This driver breaks assignment can be improved by allocating a 45-min break when the duty time reaches 6 h. This journey requires no additional breaks because the 45-min break reset the driving hours.
- Scenario 4 (a) This scenario is similar to the previous one, except that in the previous scenario, the driver completes the journey after 2.5 h of driving after the first break. However, after the first break, the driver continues to drive for 4 h and 40 min.
- Scenario 4 (b) Allocating 45 min to reset the driving break is no longer a good idea in this scenario because another break will be triggered due to reaching 4.5 h driving time. As a result, the journey will take 12 h and 10 min.
- Scenario 4 (c) However, if we increase the second break in Scenario 4 (a) from 15 to 30 min, no further driver breaks are required. As a result, the journey is reduced to 11 h and 40 min.
- Scenario 5 (a) The arrival time at the customer site was 2:30, but a wait of 15 min was required due to the time window constraint. A break was assigned during the wait. A duty break of 15 min was given after working for 6 h. After that, driving was resumed for 45 min, when a driving break was needed because the total driving time was 4.5 h.
- Scenario 5 (b) This driver breaks assignment can be improved by allocating a 30-min break after 6 h of duty. Because the accumulated 45 min break reset the driving hours, this journey requires no additional breaks.
- Scenario 6 (a) This scenario is similar to the previous one. But in this case, the driver continues to drive for 4 h and 40 min after the second break.
- Scenario 6 (b) Allocating 30 min break instead of 15 min to reset the driving break is no longer a good idea in this scenario because another break will be triggered due to reaching 4.5 h driving time. As a result, the journey will take 11 h and 55 min.

4.1.3. Generating advanced schedules

As previously discussed, conforming to both regulations can result in suboptimal scheduling of drivers' breaks. Our goal is to minimise the frequency and total duration of breaks, a strategy informed by customer feedback, which suggests that regulatory-compliant break schedules can disrupt drivers' workflows and may result in unnecessary additional break periods. According to the study by Sartori et al. (2022), the researcher showed that a break schedule consisting of no more than two breaks, each at least 45 min long, is sufficient to meet the regulatory standards for both maximum continuous driving time and total daily driving time for truck drivers on single-day trips. Because breaks can be split into two parts: an initial 15-min break followed by a subsequent 30-min break, if splitting is permitted, scheduling may require up to four breaks. This would increase the number of potential schedules per route to $O(n^4)$, with n representing the number of visits on the route. The complexity of scheduling is further complicated by the inclusion of the working time rules from Directive 2002/15/EC, as the planned breaks must adhere to either the constraints of Regulation (EC)561/2006 or those specified in Directive 2002/15/EC. The timing and duration of breaks have a significant and reciprocal effect on the compliance requirements of these regulatory frameworks.

To reduce the time complexity of the algorithm, we eliminated the need for exhaustive enumeration of break schedules on each visit. Instead, we introduce a method with linear complexity; this method uses rules to check a few critical points along the route and assigns a break only when one is necessary to be generated in compliance with either (EC)561/2006 or 2002/15/EC.

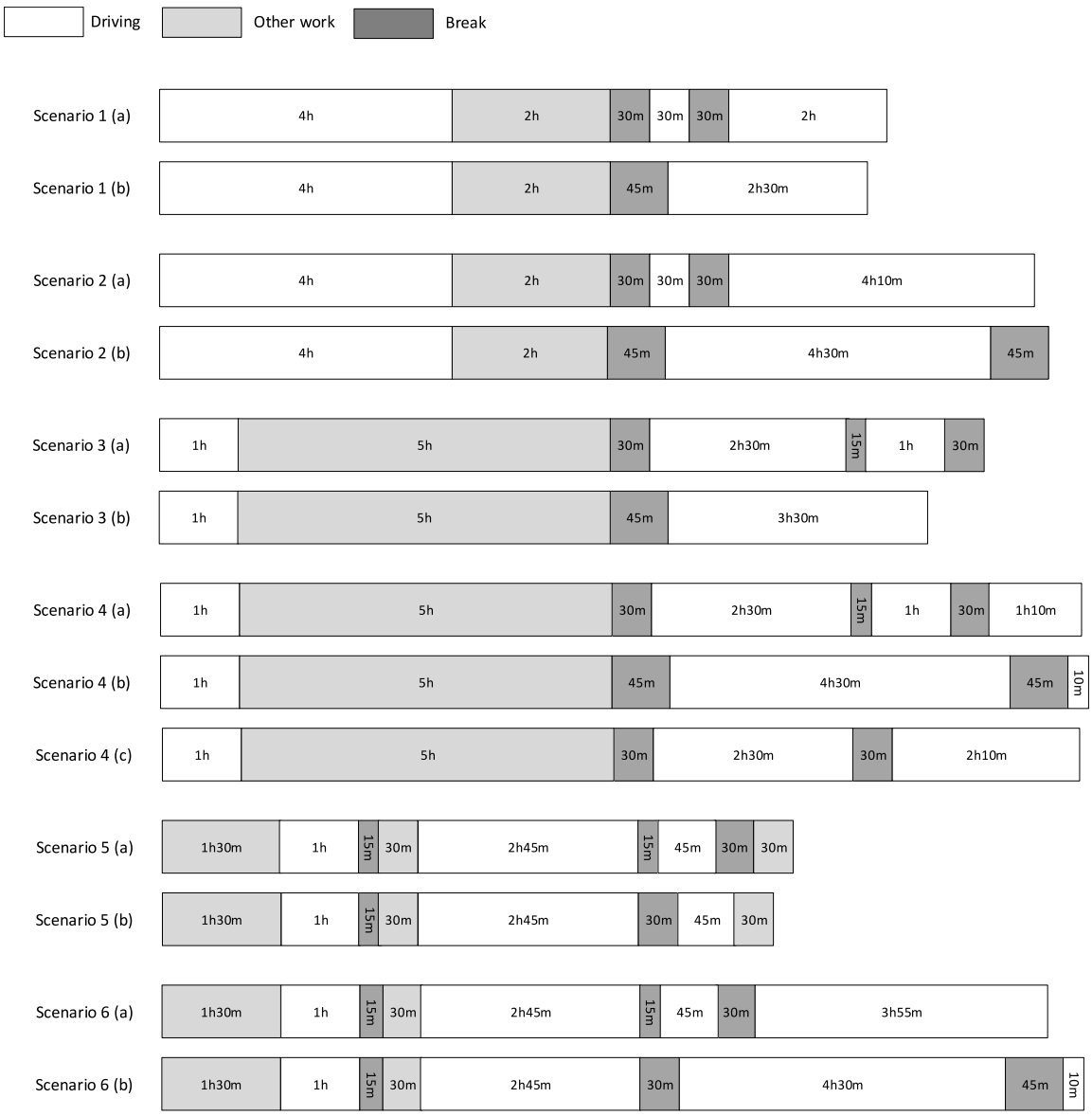


Fig. 2. Example scenarios of compliant schedules.

Upon analysing the scenarios described above, it becomes apparent that the timing of the reset for driving hours is the primary factor affecting the suboptimal scheduling of driver breaks. In this context, 'resetting' refers to a driver's capacity to extend their driving period by an additional 4.5 h, in accordance with (EC)561/2006, after a 45-min break has been taken. Advancing the timing of the driving hour reset could potentially improve break scheduling by reducing both the cumulative duration of breaks and the frequency with which they are required. This adjustment can be accomplished by prolonging the break duration when the cumulative break time up to the 6-h work period of the journey is less than the necessary 45 min required for a driving break reset (i.e. $L^{break|work1st} < l^{break}$). However, advancing the reset timing may trigger an additional 45-min break as stipulated by (EC)561/2006, should the driving requirement extend beyond 4.5 h. In such instances, it may be necessary to lengthen the break scheduled after 6 h of duty time to reset the remaining driving hours.

Consequently, in light of the aforementioned, two strategies are proposed:

- Strategy 1: When the cumulative break duration reaches the threshold of 30 min (i.e. $l^{break|work1st} = 30$) for assigning a duty break, if the number of driver breaks taken is more than zero and the total break duration of $L^{break|work1st}$ is less than 45 min (which could be the result of either a single 30-min break or two 15-min breaks by this point), the **dutyBreakRequirement1** should be adjusted from the standard 30 min to 45 min.
- Strategy 2: By the time the total break duration reaches the second threshold (i.e. $l^{break|work2nd}$) and the number of breaks exceeds one, the **dutyBreakRequirement2** should be increased from 45 min to 60 min.

The rationale for proposing the strategies is explained below: to optimise break scheduling, it is imperative to consider two pivotal thresholds for resetting driving hours. The first of these thresholds is triggered immediately after $l^{break|work1st}$, in accordance with Directive 2002/25/EC. This directive prohibits drivers from working for more than 6 h without a break and necessitates a mandatory 30-min break when the total working time extends from 6 to 9 h.

Upon reaching $l^{break|work1st}$, drivers encounter two scenarios that present opportunities to enhance compliant driver breaks scheduled by regulations (either 2002/15/EC or (EC)561/2006). This can be achieved by resetting the available driving hours before scheduling a

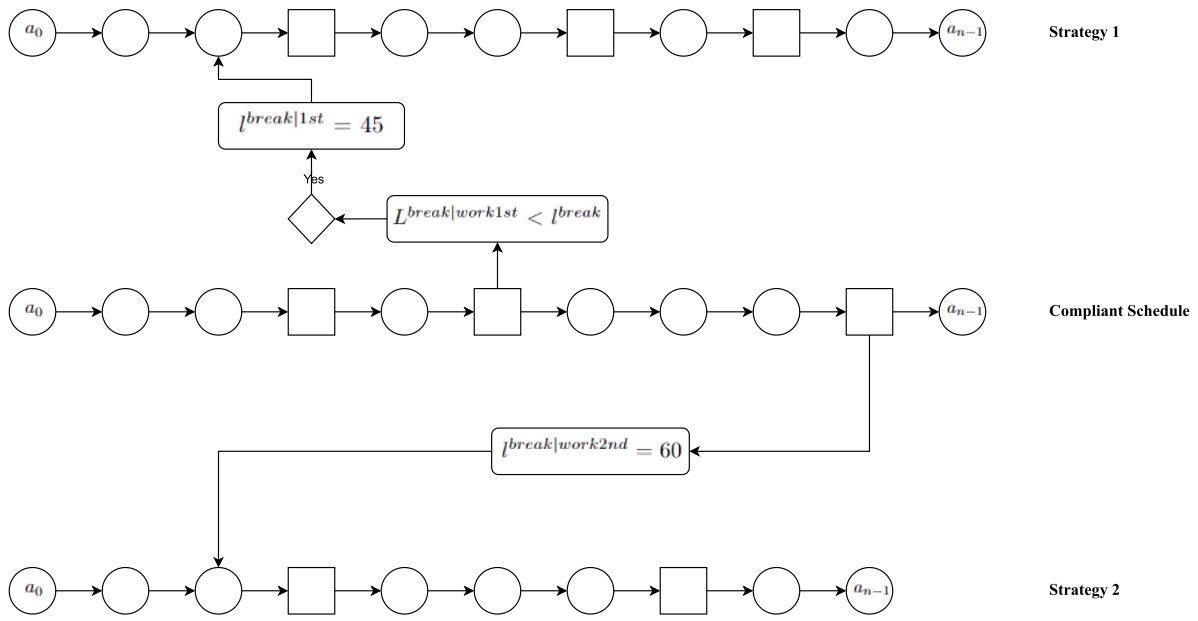


Fig. 3. Compliant Schedule and other optimisation strategies.

break, which may consist of either a single 30-min break or two 15-min breaks. To align with regulatory requirements and optimise break scheduling, we need a strategy that modifies the break durations. This involves extending the single 30-min break to 45 min (refer to Scenario 1 and 3 in Fig. 2), or, in the case of two 15-min breaks, increasing the length of the second break from 15 to 30 min (refer to Scenario 5 in Fig. 2). This can be achieved by extending **dutyBreakRequirement1** from 30 min to 45 min, which will change the break lengths from (30-min) and (15-min, 15-min) to (45-min) and (15-min, 30-min), respectively. It is crucial to note that to reset the driving hours, a 45-min break or a 30-min break following a 15-min break is required (as opposed to the reverse sequence, as stipulated by (EC)561/2006).

Similarly, there is another critical threshold at $l^{break|work2nd}$, where a 45-min break is mandatory if the total working period exceeds 9 h. At this point, drivers encounter three distinct situations that present opportunities to refine the scheduling of regulatory-compliant breaks, in accordance with either Directive 2002/15/EC or Regulation (EC)561/2006. Compliance can be enhanced by resetting driving hours prior to planning breaks, which may involve a 30-min break followed by a 15-min break, a 15-min break followed by a 30-min break, or three 15-min breaks. To align with the regulations and optimise break scheduling, we propose a strategy that involves extending the final break by 15 min (an example of this is Scenario 4 in Fig. 2). This can be achieved by increasing **dutyBreakRequirement2** from 45 min to 60 min, which would change the break schedules from (30-min, 15-min), (15-min, 15-min, 15-min), and (15-min, 30-min) to (30-min, 30-min), (15-min, 15-min, 30-min), and (15-min, 45-min), respectively.

Fig. 3 illustrates the methodology for generating driver break schedules using strategies 1 and 2. Circles represent visits, and squares represent generated driver breaks. The process initiates with the creation of a compliant schedule logic. If the number of breaks generated along the route exceeds one, a new branch is established, and the first break is modified accordingly, leading to adjustments in the time-related variables from the preceding node. Likewise, if the compliant schedule produces more than two breaks, strategy 2 is implemented.

Assume we generated additional driver breaks B from the compliant schedule using strategy 1 and 2 from the node in the route where time-related schedules must be revised on wards. B' *dominates* B If a'_j is less than a_j (meaning $route'$ takes less time than $route$) and $route'$ and $route$ are both feasible in terms of the time window of visits along the route.

4.2. Generating full solutions: embed break schedules with routes

Because the basic GPDP is known to be NP-hard, and because adding driver break scheduling increases complexity, we present a heuristic solution approach based on local search and large neighbourhood search in this section. Heuristic methods exhibit strong scalability and robustness in handling larger instances, yet they often underperform when applied to smaller instances. It can be argued that certain heuristic methods are more accessible for development, maintenance, and adaptation to new problem requirements such as the driver break feature discussed in the paper. These reasons may explain why a significant majority of commercial vehicle routing software packages prefer heuristic-based methods (Hall and Partyka, 2016).

As previously indicated, the method presented in this paper offers significant advantages for existing solvers designed for vehicle routing problems, facilitating the integration of driver break rules with minimal extra development. This technique serves as a flexible and adaptable solution that can be utilised across various route generation methods, including heuristics, column generation, and manually constructed routes. Consequently, to assess the efficacy of this method, we have conducted tests based on a heuristic strategy. However, it is important to note that this method is equally applicable to routes produced by column generation or those constructed manually.

We implemented the previously mentioned driver break scheduling algorithm on the solution framework derived from Curtois et al. (2018), Ropke and Pisinger (2006), which is capable of addressing standard PDP problems. Several publications have explored the application of meta-heuristics to specific variant variants of PDP, including instances like dynamic PDP (Gendreau et al., 2006; Ghiani et al., 2022), PDP with loading plans (Cherkesly et al., 2015) and PDP with transfers (i.e. transfer of request from one vehicle to another) (Masson et al., 2013). For more in-depth exploration of this subject, additional insights are recommended in these survey papers (Battarra et al., 2014; Berbeglia et al., 2007; Parragh et al., 2008).

The driver break scheduling method, as outlined earlier, can be employed based on specific needs. It may be triggered during route modifications or construction to ensure that all operational routes consistently include driver breaks. Alternatively, it can be applied at the end of each key phase, such as after a constructive or improvement heuristic, which can reduce computational time by decreasing the frequency of break generation and assignment. A third option is to

activate the method just before the final solution is presented, effectively serving as a repair mechanism that inserts breaks into existing routes. However, this latter approach does not guarantee feasibility, as adding breaks might exceed route duration constraints. For the purposes of this research, we have adopted the first strategy, ensuring that the method is activated following every neighbourhood function to maintain consistency in route planning.

The overall structure of the solution framework is given below. Firstly, a greedy insertion heuristic is used to iteratively build a solution by inserting unassigned customer requests into existing routes. Followed by that, local search was used to improve the initial solution. The local search contains 4 neighbourhood functions: (1) **Insert**: This operation involves selecting an unassigned customer request and inserting it into an existing route. (2) **Move**: This operation involves selecting a customer request that is already assigned to a route and removing it from that route. The removed request is then evaluated for insertion into a different route. (3) **Swap**: This operation involves selecting two customer requests, each belonging to a different route. The requests are then swapped, meaning that each request is moved to the other route. (4) **Two-Move**: This operation involves selecting three routes and three customer requests, one from each route. The requests are then moved in a pairwise fashion: the request from route 1 is moved to route 2, and the request from route 2 is moved to route 3. The goal of this operation is to explore more complex route configurations by simultaneously considering three routes and two requests reassignments.

To further enhance the solution quality, we employed a modified large neighbourhood component introduced in Shaw (1997) (Curtois et al., 2018). This approach conducts a large neighbourhood search (LNS) (Pisinger and Ropke, 2019) that integrates mechanisms for both intensification and diversification. The LNS component used for solving GPDPTW-DBS draws inspiration from Ropke and Pisinger (2006) and Curtois et al. (2018).

The algorithm for the LNS implemented in the paper is described below. At the start, a number of requests are removed from the current solution using either the *Shaw* (Shaw, 1998) or *Random* (Ropke and Pisinger, 2006; Shaw, 1998) removal heuristic. The number of requests to be removed is determined by the size of the instance and is randomly selected within a range of 5% to 50% of the total requests. The *Shaw* heuristic removes requests that are related to each other based on their spatial proximity, while the *Random* heuristic selects requests completely at random. The probability of using the Shaw heuristic is set to 0.6, and the probability of using the *Random* heuristic is set to 0.4. Next, the removed requests are reinserted back into the remaining partial solution. The unassigned requests are inserted using the regret heuristic. The goal of the regret insertion is to prioritise the insertion of requests that have fewer possible insertion locations, preventing a situation where there are too many requests and not enough available positions later on. In this study, the $k-regret$ heuristic is implemented, where k is a parameter randomly selected from 2, 3, 4, 5, or $|R|$, where $|R|$ is the number of routes in the current solution. Finally, the new solution is accepted using a *Late Acceptance Hill Climbing strategy* proposed by Burke and Bykov (2017), with a list size set to 2000. This process is repeated for a fixed number of iterations 1000, without any improvement.

We followed the Taguchi design (Roy, 2001) with the parameter values suggested by a previous study Burke and Bykov (2017) and Curtois et al. (2018) for our experiments. Other optimisation techniques, such as sensitivity analysis and dynamic adaptation, could also enhance the performance. However, since our study is not focused on (meta-)heuristic algorithms, we did not consider them in our scope.

5. Computational experiments and results analysis

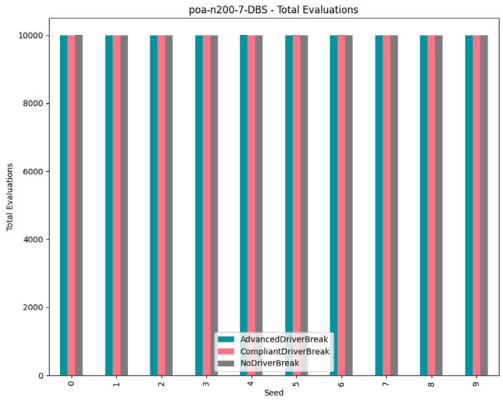
To thoroughly evaluate the performance and solution quality of our proposed methods, we modified well-established benchmark instances for the PDPTW, as introduced in the works of Li and Lim (2001) and Sartori and Buriol (2020). The (Li and Lim, 2001) benchmark instances stand as highly valuable artificial datasets that have spurred innovative research within a rigorous and competitive environment. While these artificial instances have laid the foundation for many commercial vehicle routing problem solvers, one could argue that more realistic benchmark instances could lead to even more effective methods for real-world applications. In contrast, the (Sartori and Buriol, 2020) instances were generated based on realistic locations, providing a closer representation of actual scenarios encountered in real-world problem-solving.

Customer location coordinates, customer demand, and depot coordinates are directly extracted from the problem instances. Travel times between locations, service times at customer sites, and vehicle capacities have been modified to enable vehicle journeys long enough to trigger the diverse breaks required by Regulation (EC)561/2006 and Directive 2002/15/EC. Additionally, vehicle start and end locations have been modified to include locations other than the depot.

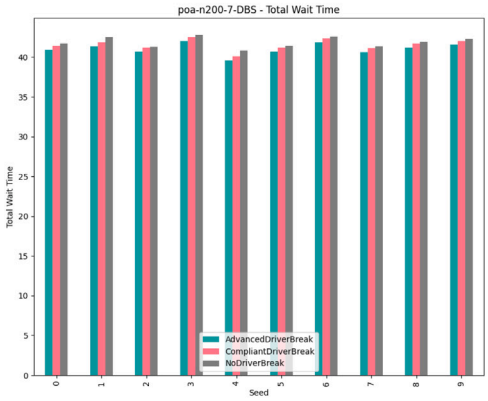
The modification of the original dataset results in a number of unassigned requests due to the increased travel and duty times introduced in the instance. To minimise the number of unassigned requests, we assign a high cost to unassigned requests. The total cost of a route is calculated based on its duty time. In order to evaluate the performance and impact of integrating driver break scheduling with the routing problem, we conducted a series of experiments. Our analysis focused on the runtime efficiency and the quality of the solutions obtained using two distinct algorithms: the standard/compliant driver break scheduling algorithm (denoted as 'Compliant' in the tables and figures) and the advanced driver break scheduling algorithm (denoted as 'Advanced' in the tables and figures). In the figures and tables presented in this section, 'NoDriverBreak' refers to the absence of a driver break scheduling setting, representing the pure routing problem, and it serves as a baseline for comparing the impact of driver break scheduling on the routing problem.

To ensure a fair comparison, we stopped the solver for all the experimental settings when they had converged and reached the same number of evaluations, which is 10,000. the term ''evaluation'' here refers to the evaluation of the objective function's value after a successful execution of a move by the local search neighbourhood function. Below, we show a few cases how the successful execution can affect results, we firstly show the result of same moves for all 10000 evaluations: Fig. 4 presents a subset of the results for the instance poan200-7-DBS. Specifically, Fig. 4(a) illustrates the evaluation metrics applied to all three settings over 10 random seeds. Fig. 4(b) depicts the total accumulated waiting time, calculated as the difference between the opening of the time window at the customer site and the arrival time of the vehicle at the customer site, for this instance. It is observed that both the standard and advanced driver scheduling algorithms result in reduced waiting times compared to scenarios without any driver break scheduling. This reduction is attributable to the strategy previously discussed, where if a driver arrives at a customer site before the time window opens, the waiting time can be utilised as a break, thus not contributing to the overall waiting time. The figure also indicates that the advanced driver break scheduling algorithm is more effective in minimising waiting times. Figs. 4(c) and 4(d) further demonstrate that the advanced algorithm is responsible for fewer and shorter driver breaks, respectively.

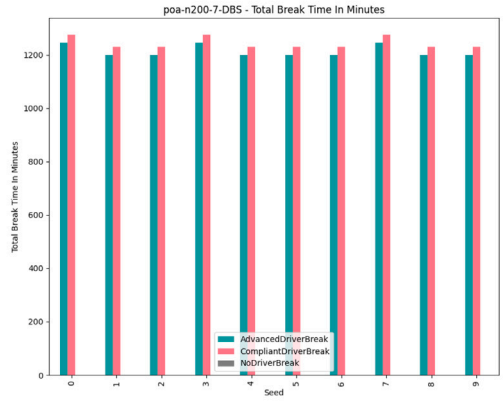
However, a local search move under the same random seed that is effective in one algorithmic setting might not be accepted by another. For example, a move that is permissible in the advanced driver break scheduling algorithm might be infeasible in the compliant driver break scheduling algorithm due to violation of time window constraints or exceedance of route duration limits. These differences cause the divergence of the search spaces between the two algorithms, leading to distinct outcomes after a series of evaluations. This case is illustrated in Fig. 5, where the same random seed was used but led to divergent results. Figs. 5(a) and 5(b) show that the advanced algorithm is superior,



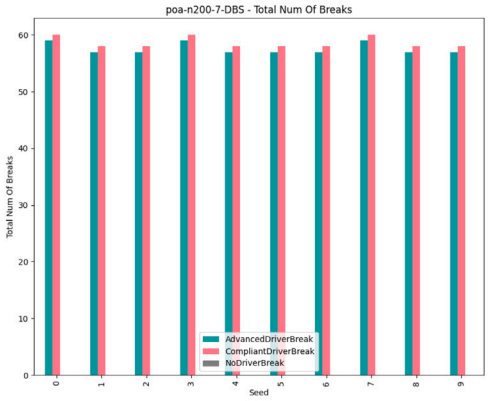
(a) Evaluations



(b) Wait Time



(c) Break time in minutes



(d) Num of breaks

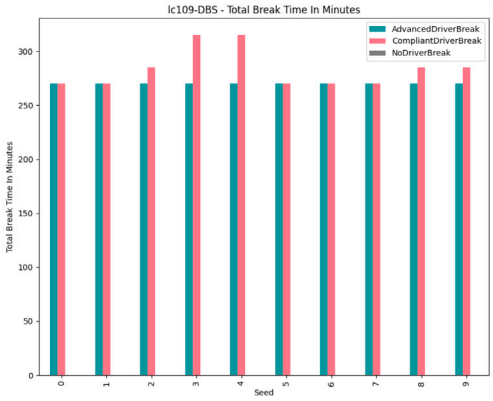
Fig. 4. Results of instance poa-n200-7-DBS.

requiring fewer and shorter driver breaks. Despite this, the compliant algorithm requires a higher number of vehicles, as seen in Fig. 5(d), even though both algorithms result in the same number of unassigned requests, as shown in Fig. 5(c). Figs. 5(e) and 5(f) indicate that, in general, the advanced algorithm reduces both the total routing distance and the duty time when compared to the compliant algorithm.

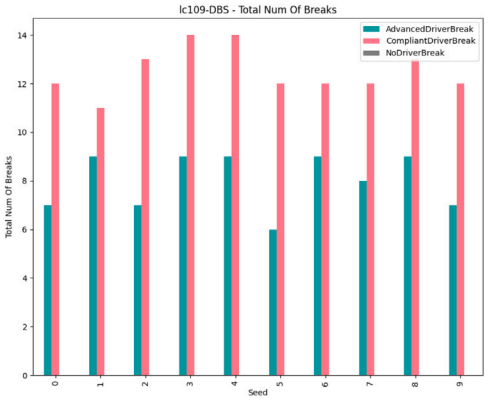
To mitigate the impact of using the same random seed but obtaining divergent solutions, as illustrated above, we now turn our attention to comparing the outcomes of integrating the compliant and advanced driver break scheduling algorithms with the routing problem across all 10 random seeds and instances. The analysis includes the standard performance metric for vehicle routing problem research—distance in kilometres—coupled with a summary of duty time in hours to assess the impact of various driver break scheduling algorithms. Given the constraint of a limited number of vehicles in the designed instances to meet all pickup and delivery requests, a summary of unassigned jobs is also provided. The summarised results can be found in Table 4. The data in these figures is aggregated for each instance; for instance, if a solution for an instance includes 10 routes, each with 2 KM, the aggregate number of distance would be 20. Each instance was executed 10 times with different random seeds, and the values depicted in the figures represent the average outcomes of these 10 runs. To save space, we only included instances where there was a difference between the compliant and advanced driver break scheduling algorithms thereafter. The sum and average of all the results are reported on the last two lines of the table.

The results show that the advanced scheduling algorithm outperforms the compliant approach. The compliant approach tends to have more distance, duty time, and unassigned count for each instance, as well as for the sum and average of all instances. On average, the compliant driver break scheduling algorithm generates 6.1% more distance, 1.7% more duty time, and 1.1% more unassigned count than the advanced counterpart. The advanced driver break algorithm produces better results than the basic one because it creates fewer and shorter breaks for the drivers. The advanced driver break algorithm performs better or equal to the compliant one in all the metrics. However, there is a trade-off between the quality of the results and the computation time. The advanced driver break algorithm takes about 35% more time to run than the basic one because it requires more complex calculations. It should also be noted that the quality of the solution is independent of scalability. When comparing the benchmark instances with 100 and 200 requests, the instances with 100 requests show an average distance reduction of 8.03%, a duty time reduction of 1.96%, and a 0.44% reduction in the unassigned count. Conversely, for the instances with 200 requests, the average distance reduction is 3.97%, the duty time reduction is 1.43%, and the reduction in the unassigned count is 1.18%.

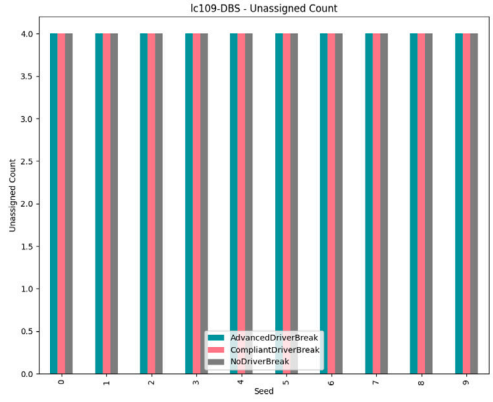
Figs. 6(a) and 6(b) provide a summary of the comparative analysis between the compliant and advanced driver break durations, as well as the count of breaks produced. Again, the results are summarised for each instance and are based on the average results of 10 different runs. The figures indicate that the advanced driver scheduling algorithm generally yields routing plans with reduced break durations for nearly all



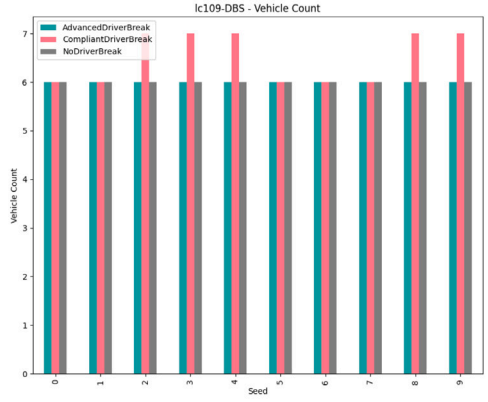
(a) Driver break duration in minutes



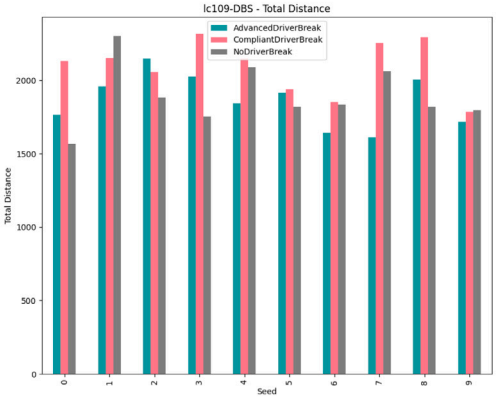
(b) Number of breaks



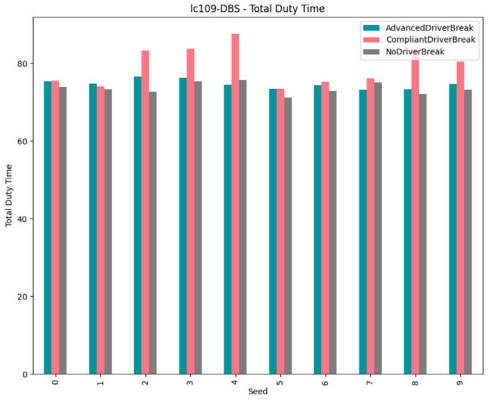
(c) Unassigned count



(d) Vehicle count



(e) Total distance

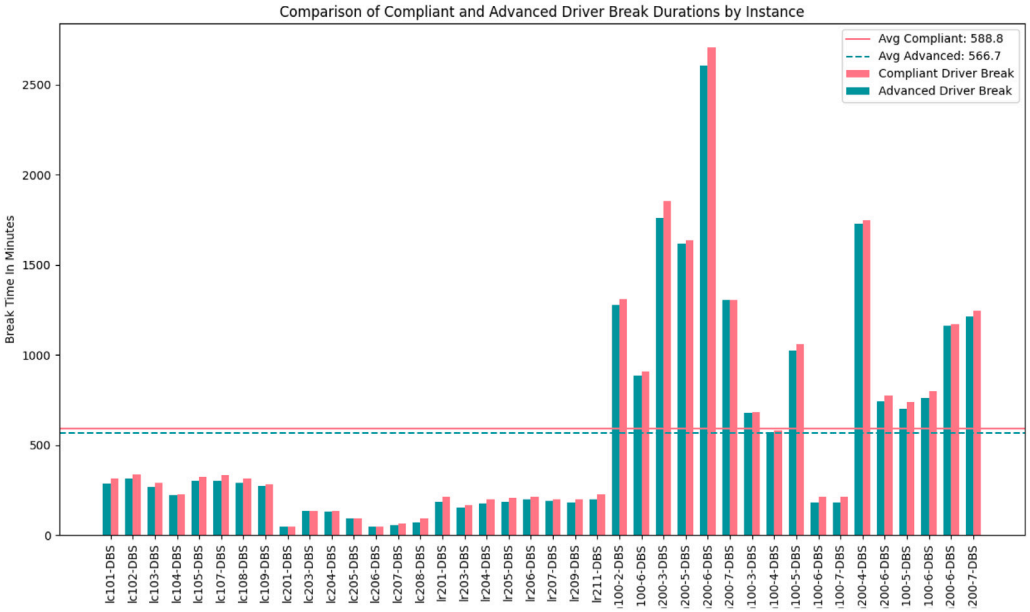


(f) Duty Time

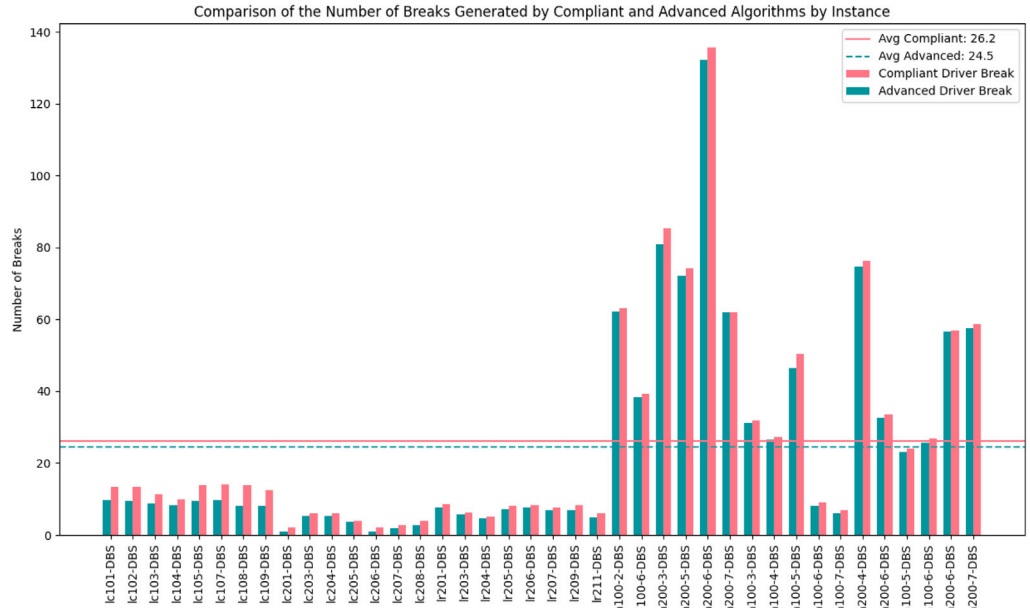
Fig. 5. Results of instance lc109-DBS.

instances, with the exceptions of lc203-DBS and lc206-DBS. However, in terms of the total number of breaks, these two instances actually show a decrease in the number of breaks generated.

To illustrate the overall trends and provide a summary of the comparison between driver break durations and the number of driver breaks, we have utilised histograms using the same data as in Figs. 6(a) and 6(b) to depict the distributional shapes. These histograms are presented in Figs. 7(a) and 7(b). We have set a wide range for the bins to enhance the clarity of the distributions. On the histograms, the x-axis indicates the measured variable, which is either the duration of breaks in minutes or the number of breaks. The y-axis, on the other hand, represents the number of occurrences of the data points within



(a) Comparison of Compliant and Advanced Driver Break Duration in Minutes by Instance



(b) Comparison of the Number of Breaks Generated by Compliant and Advanced Algorithms by Instance

Fig. 6. Comparison of distribution of driver breaks.

each bin. The histograms illustrate the results of the compliant break scheduling algorithm, showing an average break duration of 588.8 min. This distribution is slightly skewed to the right when compared to that of the advanced break scheduling algorithm, which has an average of 566.7 min. This suggests that the compliant algorithm typically schedules longer breaks than the advanced one. The figure also reveals that the advanced algorithm tends to decrease drivers' break durations, leading to a more concentrated distribution around the mean. This indicates that the advanced algorithm provides more consistent break times for drivers. Furthermore, the figure indicates that the mean break duration for the advanced algorithm is lower than that for the compliant algorithm, signifying a reduction in break time. While the compliant algorithm exhibits a broader range of break times, including some that are quite extended, the figure suggests that the advanced algorithm is more efficient and effective at scheduling drivers' breaks.

6. Conclusion

This paper studies a general pickup and delivery problem with time windows and driver break requirements, following both (EC)561/2006 and Directive 2002/15/EC. We propose a flexible approach that integrates routing and driver break scheduling. We generate routes with

Table 4
Comparison of compliant and advanced driver break scheduling.

	Distance		Duty		Unassigned	
Instance	Compliant	Advanced	Compliant	Advanced	Compliant	Advanced
lc101-DBS	1469.7	1313.7	90.2	85.9	11.4	11.4
lc102-DBS	1694.1	1474.0	95.7	90.3	6.0	6.0
lc103-DBS	1612.4	1466.9	84.5	76.9	5.1	5.0
lc104-DBS	1642.5	1539.9	65.9	64.8	1.6	1.0
lc105-DBS	1734.8	1555.9	89.9	83.6	10.0	10.0
lc107-DBS	1966.7	1652.8	95.1	89.3	8.0	8.0
lc108-DBS	2027.6	1869.3	87.8	82.0	8.0	8.0
lc109-DBS	2101.9	1853.8	79.4	74.6	4.0	4.0
lc201-DBS	219.7	202.5	11.9	11.7	46.0	46.0
lc203-DBS	880.2	839.8	38.5	38.5	28.0	28.0
lc204-DBS	1148.0	1107.3	42.3	41.6	16.0	16.0
lc205-DBS	289.0	269.0	23.2	23.0	42.0	42.0
lc206-DBS	193.5	187.0	12.8	12.8	45.0	45.0
lc207-DBS	446.3	371.3	20.8	18.7	41.0	41.0
lc208-DBS	441.0	397.0	23.2	22.2	39.0	39.0
lr201-DBS	1716.7	1623.5	58.2	53.1	7.0	7.0
lr203-DBS	1500.8	1448.8	39.9	38.3	9.0	8.8
lr204-DBS	1520.9	1452.9	35.7	33.9	5.1	4.0
lr205-DBS	1745.7	1600.3	65.8	60.4	4.0	1.7
lr206-DBS	1669.8	1594.5	59.6	57.2	3.3	2.3
lr207-DBS	1576.4	1529.8	55.7	52.3	2.6	2.4
lr209-DBS	1973.9	1875.0	52.1	48.9	0.4	0.0
lr211-DBS	1789.6	1653.6	39.8	36.4	0.0	0.0
bar-n100-2-DBS	593.3	593.3	410.1	409.6	4.0	4.0
bar-n100-6-DBS	890.2	890.2	245.4	245.0	10.2	10.2
bar-n200-3-DBS	1279.7	1266.7	526.5	521.1	38.6	38.3
bar-n200-5-DBS	1375.5	1371.5	433.9	432.6	0.0	0.0
bar-n200-6-DBS	1232.1	1214.7	895.7	885.6	4.3	3.5
bar-n200-7-DBS	674.6	670.1	378.5	374.7	63.0	62.7
ber-n100-3-DBS	541.0	538.7	194.5	194.2	2.0	2.0
ber-n100-4-DBS	461.3	458.6	175.0	174.5	1.0	1.0
ber-n100-5-DBS	729.7	719.8	313.5	311.0	10.1	10.0
ber-n100-6-DBS	131.0	131.0	50.1	49.6	44.0	44.0
ber-n100-7-DBS	170.0	170.0	57.8	57.3	43.0	43.0
ber-n200-4-DBS	1407.3	1399.7	445.7	443.5	0.0	0.0
ber-n200-6-DBS	725.0	725.0	210.6	210.1	74.4	74.4
nyc-n100-5-DBS	744.5	738.7	180.4	178.7	7.2	7.2
nyc-n100-6-DBS	751.0	748.1	183.5	182.6	6.5	6.5
poa-n200-6-DBS	926.4	914.6	338.1	333.0	62.9	62.5
poa-n200-7-DBS	926.0	926.0	344.5	344.0	61.7	61.7
Sum	44 919.8	42 355.3	6651.6	6543.5	775.4	767.6
Average	1123.0	1058.9	166.3	163.6	19.4	19.2

minimal duration and explicit driver breaks, which can be applied to many variants of the problem. It provides a solution that quickly generates viable options in response to disruptions affecting the current solution, leading to the immediate rescheduling of driver breaks. Additionally, alongside the compliant approach, we introduce and evaluate a more advanced method that aims to optimise route duration through strategic adjustments in the timing for resetting driving hours. We test and compare our approach using modified real-life and artificial benchmark instances. The results show that the advanced scheduling approach outperforms the compliant approach in terms of distance, duty time, and unassigned count.

The algorithm we proposed for generating driver break schedules is adaptable and can accommodate the driver break regulations of other countries, including Australia and New Zealand, within a single day of operation. While our research and experimentation primarily address regulations of directive 2002/15/EC and regulation EC 561/2006, we provide guidance on parameter adjustments for Australia and New Zealand in Appendix.

We identify directions for future research. Although the current solution quality satisfies the needs of the GPDP solver with integrated driver break functionality, it would be advantageous to investigate the use of this method in conjunction with exact algorithms, such as column generation, rather than relying solely on heuristic approaches. Moreover, it would be interesting to examine the effect on solution quality when the method is activated exclusively at the end of each major phase in the solving process, as opposed to its current application during route modifications or construction. This could provide valuable insights into the method's efficiency across different computational strategies.

CRediT authorship contribution statement

Ning Xue: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Huan Jin:** Writing – review & editing, Validation, Formal analysis. **Tianxiang Cui:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

Acknowledgments

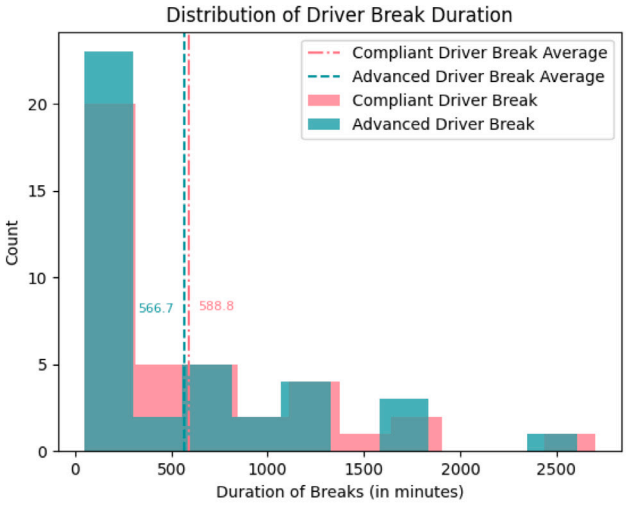
This work was supported by the Ningbo Science and Technology Bureau (Project ID: 2024S057), Ningbo Natural Science Foundation (Project ID: 2023J194), University of Nottingham Ningbo China Education Foundation (Project ID: LDS202303), Basic and Commonweal Programme of Zhejiang Natural Science Foundation (Project ID: LY24F020006), and Ningbo Government (Project ID: 2021B-008-C).

Table 5
Standard Hours (SH) rules.

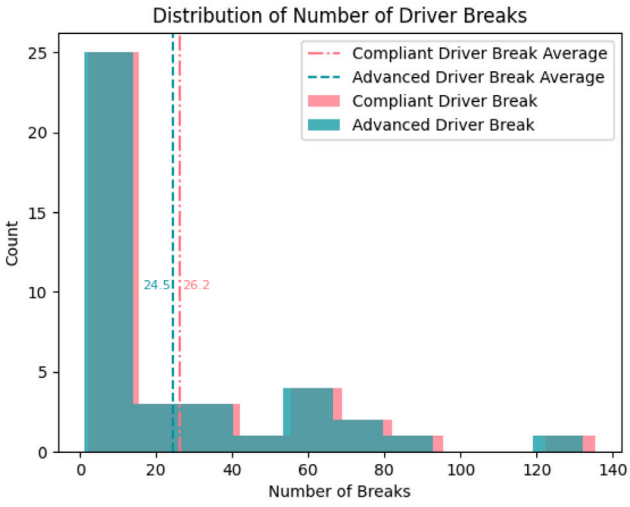
In any period of...	Maximum allowed working time	Minimum break required
5 1/2 h	5 1/4 h work time	blocks of 15 min
8 h	7 1/2 h work time	30 min rest time in blocks of 15 min
11 h	10 h work time	60 min rest time in blocks of 15 min

Table 6
Basic Fatigue Management (BFM) rules.

In any period of...	Maximum allowed working time	Minimum break required
6 1/4 h	6 h work time	blocks of 15 min
9 h	8 1/2 h work time	30 min rest time in blocks of 15 min
12 h	11 h work time	60 min rest time in blocks of 15 min



(a) Histogram-Distribution of Driver Break Duration in Minutes



(b) Histogram-Distribution of Number of Driver breaks

Fig. 7. Comparison of distribution of driver breaks.

Table 7
Parameters of minutes of service regulations in Australia.

Notation	SH Minutes	BFM Minutes
$l^{break\|1st}$	30	30
$l^{break\|2nd}$	60	60
$l^{break\|work1st}$	315	360
$l^{break\|work2nd}$	615	690

Appendix

Australia driver break rules

The Heavy Vehicle National Law (HVNL) and five sets of regulations are administered by the Australia National Heavy Vehicle Regulator (NHVR) for heavy vehicles over 4.5 tonnes gross vehicle mass. The HVNL provides three work and rest options: standard hours, Basic Fatigue Management (BFM), and Advanced Fatigue Management (AFM). The relevant rules and parameters of minutes of service regulations are summarised in Tables 5–7.

Standard hours (SH): These are the maximum work hours and the minimum rest hours allowed by the HVNL. They are designed to ensure safe driving without additional safety measures. For example, if you start work at 12:00, you must take at least 15 min of rest before 5:30; another 15 min before 8:00; and another 30 min before 11:00.

Basic Fatigue Management (BFM): This option requires NHVAS accreditation and allows for more flexible work and rest hours, such as working up to 14 h in a 24-h period. For example, if you start work at 12:00, you must take at least 15 min of rest before 6:15; another 15 min before 9:00; and another 30 min before 12:00.

Advanced Fatigue Management (AFM): This option also requires NHVAS accreditation and allows for customised work and rest hours, based on a risk management approach. However, this option is out of the scope of this study.

New Zealand driver break rules

The following are the summary of the rest rules for New Zealand heavy vehicle drivers: A driver must take a minimum 30 min break after working for 5 1/2 h continuously. That is, the maximum work time before a rest break is 5 1/2 h. A driver must not work for more than 13 h in any cumulative work day. A cumulative work day is a period of 24 h that starts when the driver starts work after a continuous rest time of at least 10 h. The New Zealand break regulation requires two breaks within 13 h of work time, which can be handled by the

Table 8
Parameters of minutes of service regulations in New Zealand.

Notation	Minutes
$l^{break\|1st}$	30
$l^{break\|2nd}$	60
$l^{break\|work1st}$	330
$l^{break\|work2nd}$	690

algorithm proposed in this paper. The relevant parameters of minutes of service regulations are summarised in Table 8.

Note that the algorithm does not support more than two duty breaks. For the Australian regulations, the Standard hour regulation and BFM regulation suggest three breaks (within 11 h and 12 h respectively), but it is possible to assign only two breaks (instead of three) and still produce valid solutions using the current implementation of the algorithm.

Data availability

Data will be made available on request.

References

Archetti, C., Savelsbergh, M., 2009. The trip scheduling problem. Transp. Sci. 43 (4), 417–431.

Battarra, M., Cordeau, J.-F., Iori, M., 2014. Chapter 6: pickup-and-delivery problems for goods transportation. In: Vehicle Routing: Problems, Methods, and Applications, Second Edition. SIAM, pp. 161–191.

Benus, J., Demirci, E., 2020. Regulation (EC) no 561/2006-review of the adopted changes on 15 july 2020. Arch. Mot. 90 (4), 59–73.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. Top 15, 1–31.

Burke, E.K., Bykov, Y., 2017. The late acceptance hill-climbing heuristic. European J. Oper. Res. 258 (1), 70–78.

Cherkesly, M., Desaulniers, G., Laporte, G., 2015. A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. Comput. Oper. Res. 62, 23–35.

Curtois, T., Landa-Silva, D., Qu, Y., Laesanklang, W., 2018. Large neighbourhood search with adaptive guided ejection search for the pickup and delivery problem with time windows. EURO J. Transp. Logist. 7 (2), 151–192.

Gendreau, M., Guertin, F., Potvin, J.-Y., Séguin, R., 2006. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transp. Res. Part C: Emerg. Technol. 14 (3), 157–174.

Ghiani, G., Manni, A., Manni, E., 2022. A scalable anticipatory policy for the dynamic pickup and delivery problem. Comput. Oper. Res. 147, 105943.

Gibson, G., Tsamis, A., Löhr, E., Weil, T.B., Levin, S., Hughes, D., 2017. Study to support the impact assessment for the revision of regulation (EC) no 1071/2009 and regulation (EC) no 1072/2009. Rep. Eur. Comm..

Goel, A., 2009. Vehicle scheduling and routing with drivers' working hours. Transp. Sci. 43 (1), 17–26.

Goel, A., 2010. Truck driver scheduling in the European union. Transp. Sci. 44 (4), 429–441.

Goel, A., 2018. Legal aspects in road transport optimization in europe. Transp. Res. Part E: Logist. Transp. Rev. 114, 144–162.

Goel, A., Irnich, S., 2017. An exact method for vehicle routing and truck driver scheduling problems. Transp. Sci. 51 (2), 737–754.

Hall, R., Partyka, J., 2016. Vehicle routing software survey: Higher expectations drive transformation. ORMS- Today 43 (1), 40–44.

Kok, A., Hans, E.W., Schutten, J.M., Zijm, W.H., 2010. A dynamic programming heuristic for vehicle routing with time-dependent travel times and required breaks. Flex. Serv. Manuf. J. 22, 83–108.

Li, H., Lim, A., 2001. A metaheuristic for the pickup and delivery problem with time windows. In: Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001. IEEE, pp. 160–167.

Masson, R., Lehuédé, F., Péton, O., 2013. An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transp. Sci. 47 (3), 344–355.

Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008. A survey on pickup and delivery problems: Part I: Transportation between customers and depot. J. Betriebswirtschaft 58, 21–51.

Pisinger, D., Ropke, S., 2019. Large neighborhood search. Handb. Metaheuristics 99–127.

Prescott-Gagnon, E., Desaulniers, G., Drexl, M., Rousseau, L.-M., 2010. European driver rules in vehicle routing with time windows. Transp. Sci. 44 (4), 455–473.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.

Roy, R.K., 2001. Design of Experiments Using the Taguchi Approach: 16 Steps to Product and Process Improvement. John Wiley & Sons.

Sartori, C.S., Buriol, L.S., 2020. A study on the pickup and delivery problem with time windows: Matheuristics and new instances. Comput. Oper. Res. 124, 105065.

Sartori, C.S., Smet, P., Berghe, G.V., 2022. Scheduling truck drivers with interdependent routes under European union regulations. European J. Oper. Res. 298 (1), 76–88.

Savelsbergh, M.W., Sol, M., 1995. The general pickup and delivery problem. Transp. Sci. 29 (1), 17–29.

Shaw, P., 1997. A new local search algorithm providing high quality solutions to vehicle routing problems. APES Group, Dept Comput. Sci. Univ. Strat. Glas. Scotl. UK 46.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: International Conference on Principles and Practice of Constraint Programming. Springer, pp. 417–431.

Tilk, C., Goel, A., 2020. Bidirectional labeling for solving vehicle routing and truck driver scheduling problems. European J. Oper. Res. 283 (1), 108–124.