

Article

A Case Study of a Transportation Company Modeled as a Scheduling Problem

Cristina Tobar-Fernández ^{1,*}, Ana Dolores López-Sánchez ² and Jesús Sánchez-Oro ¹

¹ Department of Computer Science, University Rey Juan Carlos, C/Tulipán s/n, Móstoles, 28933 Madrid, Spain; jesus.sanchezoro@urjc.es

² Department of Economic, Quantitative Method and Economic History, Universidad Pablo de Olavide, Ctra. de Utrera km 1, 41013 Sevilla, Spain; adlopsan@upo.es

* Correspondence: c.tobar.2024@alumnos.urjc.es

Abstract

This case study tackles a real-world problem of a transportation company that is modeled as a scheduling optimization problem. The main goal of the considered problem is to schedule the maximum number of jobs that must be performed by vehicles over a specific planning horizon in order to minimize the total operational costs. Here, each customer request corresponds to a job composed of multiple operations, such as loading, unloading, and mandatory jobs, each associated with a specific location and time window. Once a job is allocated to a vehicle, all its operations must be executed by that same vehicle within their designated time constraints. Due to the imposed limitations, not every job can feasibly be scheduled. To address this challenge, two distinct methodologies are proposed. The first, a Holistic approach, solves the entire problem formulation using a black-box optimizer, serving as a comprehensive benchmark. The second, a Divide-and-Conquer approach, combines a heuristic greedy algorithm with a binary linear programming, decomposing the problem into sequential subproblems. Both approaches are implemented using the solver Hexaly. A comparative analysis is conducted under different scenarios and problem settings to highlight the advantages and drawbacks of each approach. The results show that the Divide-and-Conquer approach significantly improves computational efficiency, reducing time by up to 99% and vehicle usage by around 15–20% compared to the Holistic method. On the other hand, the Holistic method better ensures that mandatory jobs are completed, although at the cost of more resources.



Academic Editors: Alessandro Mazzoccoli, Raffaele Mattera and J.E. Trinidad-Segovia

Received: 26 September 2025

Revised: 31 October 2025

Accepted: 2 November 2025

Published: 5 November 2025

Citation: Tobar-Fernández, C.; López-Sánchez, A.D.; Sánchez-Oro, J. A Case Study of a Transportation Company Modeled as a Scheduling Problem. *Mathematics* **2025**, *13*, 3547. <https://doi.org/10.3390/math13213547>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: scheduling problem; commercial constraints; time windows; divide-and-conquer algorithm; Hexaly

MSC: 90B06; 90B90

1. Introduction

In a globalized economy, the efficiency and effectiveness of transportation networks are crucial for sustaining competitive advantage across industries [1]. Companies face increasing demands, fluctuating costs, and environmental constraints, compelling them to operate with heightened efficiency and precision. To address these challenges, optimization techniques have become indispensable tools, enabling improved transportation system performance, reduced operational costs, and minimized environmental impact [2]. Among these techniques, assignment, scheduling, and routing problems stand out as three of the

most extensively studied combinatorial optimization problems, each playing a critical role in enhancing transportation network efficiency [3–5].

In the following, we briefly describe each of these problems, emphasizing their practical relevance and broad applicability in real-world scenarios, which make them fundamental for addressing operational challenges.

1. The assignment problem is an optimization problem in which the objective is to assign a set of resources (e.g., machines, vehicles, or drivers) to a set of jobs or tasks (e.g., delivery routes) in the most efficient way. The goal is typically to minimize costs or maximize efficiency while respecting constraints, such as capacity, skill levels, or regional restrictions.
2. The scheduling problem involves determining the optimal sequence and timing for a set of tasks or jobs assigned to machines to maximize efficiency, minimize costs, or achieve specific objectives within given constraints. In the context of transportation, the scheduling problem is critical for coordinating resources, such as vehicles, drivers, and loading and unloading operations, to ensure timely delivery and efficient utilization of assets.
3. The routing problem plans the optimal set of routes for a fleet of vehicles to deliver goods or provide services to a set of locations (e.g., customers or delivery points) in such a way that operational costs are minimized while satisfying all delivery constraints.

Therefore, the main differences among the previous problems are summarized in Table 1:

Table 1. Differences among assignment, scheduling, and routing problems.

	Assignment	Scheduling	Routing
Objective	Allocate resources (machines) to tasks (jobs).	Determine the optimal timing and sequence of tasks (jobs) to the assigned resources (machines).	Find the optimal routes for a fleet of vehicles (resources) to serve a set of customers (tasks).
Question	Who or which resource should perform every task?	When should each task be performed and in what order?	Which routes should each vehicle take?
Example	Assigning drivers to vehicles to perform the deliveries.	Scheduling the sequence of deliveries for a fleet of vehicles to avoid idle time, or planning driver shifts to meet demand while respecting labor regulations.	Determining the most efficient routes for a fleet of vehicles to perform the deliveries

It is important to emphasize that a strictly disjoint classification among these three problems is not possible as most real-world problems integrate key characteristics of all three families. In particular, the problem of interest in this paper focuses on optimizing the internal operations of a transportation company operating in Spain. Although this type of problem is usually modeled as a routing problem [6,7], in this research, the problem is modeled as a scheduling problem given that this formulation better captures its temporal constraints and sequential decision-making, even though it shares characteristics with assignment and routing problems. In general, a scheduling problem seeks to determine the optimal allocation of resources (machines) to a sequence of jobs, also referred to as

tasks, over a planning horizon, aiming to satisfy a set of constraints and optimize specific performance criteria. Different variants of the problem arise depending on the considered constraints and performance criteria. This problem is fundamental across fields such as manufacturing, transportation, computing, and digital system design. Broadly speaking (see, for example, [8]), the main characteristics of scheduling problems are as follows:

- Jobs must be assigned to a limited number of machines. In the present study, compatible customer job requests (hereinafter referred to as jobs) are assigned to machines (hereinafter referred to as vehicles).
- Jobs must be completed in a specific order, respecting dependencies and the necessary sequence between them. In this work, each customer pays for a job that is performed by different operations: loading, unloading, and/or mandatory jobs.
- Precedence constraints may be imposed. These require the operations of each job to be executed in a specific order, which must be respected.
- Time constraints may be considered, such as predefined time windows for jobs. In our case study, not only must time windows be satisfied but labor regulations must also be complied with, including maximum daily working hours and mandatory rest periods.
- Resource constraints may be imposed as the number of resources (e.g., machines, functional units, or vehicles) is typically limited. In the problem addressed here, the number of vehicles is limited, and the compatibility between jobs and vehicles must be taken into account.
- The main objective is to optimize certain criteria, such as minimizing total completion time, maximizing resource utilization, or reducing resource consumption, among others. In our case study, the objective is to minimize the total cost, particularly travel costs and penalty costs incurred when a job remains unplanned.

Therefore, as previously explained, and given that we are dealing with a transportation-focused problem, the resources or machines consist of a fleet of vehicles, which must be allocated efficiently to fulfill as many jobs as possible while keeping the overall costs low.

From a theoretical point of view, the number of papers dealing with scheduling problems (and all the variants) is large, but this number rises considerably if we focus on a practical point of view [9]. It is worth mentioning that a conclusion derived from that work is the relevance of a good adaptation of the search algorithm to the characteristics of the problem in order to solve real-life problems with that approach. In similar studies, the routing component is often omitted, as in this work, since movements between locations are treated as discrete tasks that do not require explicit route optimization [10].

To mention just a few of the most recent and similar papers to the considered problem, we refer the reader to [11]. The authors consider a problem with multiple facilities and one or more vehicles located in the facilities. The objective is to reduce the cost of producing and distributing customer orders while determining the allocation of customers to each vehicle, the grouping of customer orders into batches, the scheduling of production for each batch, the number of vehicles assigned to each facility, and the sequence of customer visits during each tour. Moreover, supplementary constraints are established: each customer must be visited only once; the total area, weight, volume, and other specifications of the products loaded into the vehicles must not surpass the vehicle capacity; each vehicle is required to commence and conclude its route at the same facility; subsequent to the completion of a batch production, all the products within that batch must be delivered by a specified deadline; the demands of customers allocated to a facility must not exceed that facility's capacity; and all production and distribution activities must be concluded within the designated timeframe.

The recent paper by [12] addresses a single-depot electric vehicle scheduling problem. The authors offer a scheduling approach utilizing mixed-integer linear programming to cre-

ate bus blocks that integrate electric vehicles, ensuring the effective execution of each block while accounting for recharging needs between blocks and during non-operational hours. Blocks can be seamlessly repeated on consecutive days due to the incorporation of next-day operability requirements. They address a case study in Chicago. They solve the problem by generating blocks in the first stage and then focus on optimizing block combinations to facilitate recharging between consecutive blocks while considering operational constraints to plan the vehicle schedules. They use four solution approaches, namely mixed-integer linear programming, a Divide-and-Conquer algorithm, a greedy heuristic, and a simulated annealing metaheuristic.

Another interesting problem very close to the one tackled in this research is published by [13]. The scheduling challenge for airport ferry vehicles is modeled as an unrelated parallel machine scheduling problem by the authors. Finding methods to assign a finite number of ferry vehicles to remote stand flights is the challenge. The service times and time windows are deterministic and known beforehand, so the only decision involves the allocation of ferry vehicles. To solve the problem, they propose a variable neighborhood descent algorithm.

Building on this line of research, off-the-shelf solvers like Hexaly have recently been proven to be effective and efficient for dealing with routing problems. For instance, Ref. [14] presents a novel formulation for solving the Pickup and Delivery Problem with Time Windows considering driver breaks, solving it with three solvers: Gurobi, Google OR-Tools, and Hexaly. In particular, Hexaly shows better performance for the classic formulation without considering driver breaks, while neither Gurobi nor Google OR-Tools are able to provide solutions for the problem with the new constraints derived from driver breaks. This research guides us in considering Hexaly over other solvers, such as Gurobi or OR-Tools.

The effectiveness of black-box solvers has also been demonstrated in other routing contexts. For example, in Drone Routing and Scheduling with Flexible Multiple Visits (DRSFMV) [15], a case study using real traffic census data from three southern California counties was used to evaluate different models and network configurations. While the mathematical formulation solves small and medium instances efficiently, it struggles with large-scale networks. In contrast, Hexaly achieves high-quality solutions even for large and complex instances, further highlighting its efficiency and robustness across different types of routing problems.

The Divide-and-Conquer strategy has been used in several works related to routing and scheduling problems. In [16], the VRPDiv approach is introduced, a Divide-and-Conquer framework designed to scale any existing VRP solver to handle instances with up to ten thousand targets (10k) by dividing them into smaller manageable clusters. The framework intelligently selects from a pool of clustering algorithms based on instance properties and assigns agents while considering cluster demand, time windows, and capacity limitations. VRPDiv is successfully integrated into the Bing Maps Multi-Itinerary Optimization (MIO) online service, enabling it to solve 10k target instances in under 10 min, demonstrating significant performance gains, including a 9.8% improvement in itinerary durations and a 40% improvement in target allocation compared to relevant baselines. The research presented in [17] also proposes a Divide-and-Conquer algorithm but with a focus on parallel computation. This method partitions the original complex problem into smaller independently solvable subproblems that can be executed in parallel. By utilizing a novel flow graph construction and graph min-cut concepts, the resulting algorithm efficiently identifies viable AP and BP pairs, significantly outperforming existing approaches by achieving higher routing performance at a much faster speed.

In the context of scheduling problems, Ref. [18] proposes an optimization algorithm based on the Divide-and-Conquer strategy combined with Particle Swarm Optimization

(PSO) to effectively solve large-scale Job Shop Scheduling Problems (JSSPs), with the objective of minimizing total weighted tardiness. The algorithm adopts a non-iterative framework: it first uses a simulated annealing (SA) procedure to find an optimal decomposition policy for the vast set of operations. Based on this policy, the original problem is broken down into smaller subproblems, which are then sequentially solved by a PSO algorithm to directly produce a feasible solution for the entire scheduling challenge. This approach significantly improves search guidance and convergence speed, resulting in better-quality solutions and a substantial reduction in computational time compared to previous iterative decomposition methods. Analogously, Ref. [19] presents the Divide-and-Conquer Workflow Scheduling (DQWS) algorithm, a novel approach for scheduling complex computational workflows in the cloud with the primary objective of minimizing execution cost while strictly respecting a user-defined deadline. The algorithm employs a Divide-and-Conquer strategy inspired by the workflow's critical path. DQWS iteratively identifies and schedules the critical path—the longest path in the workflow—then removes it, effectively dividing the remaining tasks into smaller, less complex mini-workflows. This process continues until only simple chain-structured workflows, known as linear graphs, are left for the final scheduling phase. Through experimental validation using various scientific workflows, DQWS demonstrates superior performance over competing methods, successfully meeting deadlines while simultaneously achieving lower monetary costs for workflow execution.

The main contributions of this paper are outlined below.

- We address a challenging case study from a transportation company, modeled through a mathematical formulation with a linear objective function and highly combinatorial constraints, such as time windows, break times, and maximum driving or working times.
- The case study is first solved using the complete mathematical formulation as a benchmark, hereinafter referred to as the *Holistic approach*.
- We propose a hybrid algorithm, hereinafter *Divide-and-Conquer*, that combines a greedy construction phase with mathematical optimization to efficiently solve the problem.
- Computational experiments on 64 instances validate the effectiveness of the Divide-and-Conquer algorithm, showing superior performance compared to the Holistic approach.
- The proposed framework exhibits high scalability and practical applicability to domains such as network design, transportation, and social network analysis.

The remainder of this paper is structured in the following way. Section 2 details the considered problem. Section 3 explains the two methodologies that we have implemented to tackle the problem. Section 4 shows and discusses the different scenarios for reviewing the pros and cons of every methodology. Finally, Section 5 concludes the paper and includes future lines of research.

2. Problem Description

In this section, we will formally describe the problem considered in this paper. As previously introduced, we are modeling a problem of interest in a transportation company as a scheduling problem. This transportation company conducts its business activity throughout the Iberian Peninsula and plans its activity over a predefined planning horizon, in this case five working days within a week. The main purpose of the problem is to schedule jobs on machines represented by vehicles in this company while ensuring the following requirements.

The company owns a heterogeneous fleet of vehicles, so each vehicle has specific characteristics in terms of capacity. Furthermore, each vehicle is uniquely assigned to a

single driver and is associated with a different depot. Therefore, the scheduling of each vehicle during the planning horizon must begin and end at its designated depot location.

Each customer requests the transportation company to perform a job consisting of a number of operations. The job may be accepted or rejected. An operation involves the loading and/or unloading of items in different locations, which must be completed in a defined order. Therefore, each operation of an accepted job has an associated precise location, and it must be within one of several time windows. It is assumed that not all jobs are accepted or completed; therefore, rejected jobs incur a penalty, in this case 10,000 monetary units, since it has been modeled as money lost. This figure is derived from historical data provided by the transportation company and reflects typical payments per job in the operational context. Using this value, the model is calibrated to reproduce realistic cost structures and economic performance. Furthermore, each operation requires a fixed service time that is known beforehand. A vehicle cannot handle two jobs simultaneously, meaning that each job must be completed sequentially; furthermore, the vehicle must be compatible with the assigned jobs.

The travel cost (distance and time) between every pair of operation locations is deterministic and known beforehand. In our specific problem, distance and time are correlated measures since we assume that vehicles travel at 75 km/h. A constant speed is chosen based on the recommendation of company experts. This recommendation is considered reasonable as it aligns with the estimated average truck speed across different road types in Spain, which is approximately 75 km/h based on legal speed limits and typical operational conditions rather than on direct measurement. It should be noted that this simplification does not compromise the validity of the proposed methodology. There is a cost per kilometer traveled, in this case 1 monetary unit. Therefore, we can translate this travel cost into monetary units, length units, or time units as appropriate.

For scheduling purposes, once a vehicle has completed a job, it is possible to schedule another one if the location of the first operation of the possible next job is within a given operating range. The operating range refers to the maximum distance a vehicle can travel empty between the location where it completed a job and the location of the next job. If no jobs exist within this distance, the vehicle is locked at the current location and is not available until the next planning horizon.

Summarizing, our goal is to assign jobs to vehicles, ensuring that each operation of a job is performed within a specified time window. This involves determining which vehicle will handle each job, taking into account factors such as vehicle capacity, proximity between the last operation of the former job and the first operation of the latter job, and when the operations for that job are scheduled.

The objective function is to minimize total cost, including both the cost per kilometer traveled and penalties for jobs that are not completed. By integrating these elements, the goal is to complete the maximum number of jobs while taking into account transportation costs.

Once the baseline information has been provided, we will incorporate a pair of business constraints into the described problem. The first one is mandatory scheduled stops, specific for each vehicle. These will be treated as special jobs with one single operation within a short imposed time window. These mandatory jobs could represent driver needs, such as medical appointments, or vehicle maintenance, such as workshop servicing. The second one will take into account rest periods for drivers, called breaks. In particular, as we are considering a planning horizon of five working days within a week, the labor rest regulations must be respected. Every driver must rest at least 11 h per day. Furthermore, it is necessary to ensure that the driving time between consecutive breaks does not exceed 9 h per day, while the total working time is limited to a maximum of 15 h per day.

Figure 1 depicts a simple schema to better understand the problem of interest. In our particular case, machines are represented by vehicles, with the only imposition of starting and ending at their depot location, and they must also respect the labor regulations. Furthermore, customers request jobs to be scheduled by paying a cost. Every job consists of operations that have an associated location, service time, and time window. These operations could be loading, unloading, or mandatory jobs.

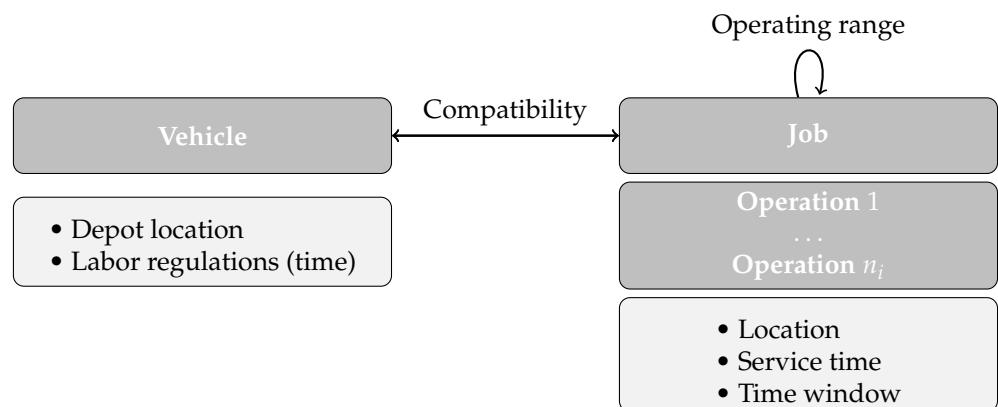


Figure 1. Problem schema.

To close this section, Figure 2 includes an illustrative example with two vehicles (denoted by V_1 and V_2) and seven job requests from customers (denoted by J_1, \dots, J_7) that must be scheduled in a planning horizon of five working days within a week. In this particular case, jobs 1, 3, 5, 6, and 7 have two operations, denoted by O_{i1} and O_{i2} for $i = 1, 3, 5, 6, 7$, which may represent loading and unloading operations, each one in a different location; job 2 has one single operation, O_{21} , which likely represents a mandatory job in a given location and has two associated time windows, while job 4 has three operations, which may first represent a loading operation in a given location, O_{41} , and then two unloading operations in two different locations, O_{42} and O_{43} . The columns of Figure 2a include the processing times for each operation and represent the start time of each operation. Note that, for simplicity in this illustrative example, it is assumed that all operations have the same duration and that the distance and travel time between operations are equal. In addition, Figure 2b shows the time/distance matrix between operations, and the last column, denoted by TW, specifies their associated time windows.

The Gantt chart is depicted in Figure 2c. Jobs 1, 3, and 6 are scheduled on the first vehicle, while the second vehicle is assigned jobs 2, 4, and 5. As can be observed, the first vehicle will start operations 1 and 2 of job 1 at times 0 and 3, respectively; then, operations 1 and 2 of job 3 are scheduled at times 7 and 10, respectively. Finally, the first vehicle ends the planning horizon, performing operations 1 and 2 of job 6 at times 17 and 22, respectively. Meanwhile, the second vehicle starts its activity at time 2 with the only operation of job 2; then, the three operations of job 4 are scheduled at times 6, 9, and 13, respectively. To finish the schedule of the second vehicle, the two operations of job 5 are planned at times 17 and 23. It is important to note that the gaps between operations in the Gantt chart represent the travel time between consecutive operations. Suppose that the cost per unit of distance is 1 μm and that the penalty for not scheduling a job is 10 μm . The total distance traveled by the vehicles is therefore 23 units, resulting in a travel cost of $23 \times 1 = 23 \mu\text{m}$. Since job 7 was not scheduled, a penalty of 10 μm is added. Consequently, in this case, the objective function value would be $23 + 10 = 33 \mu\text{m}$.

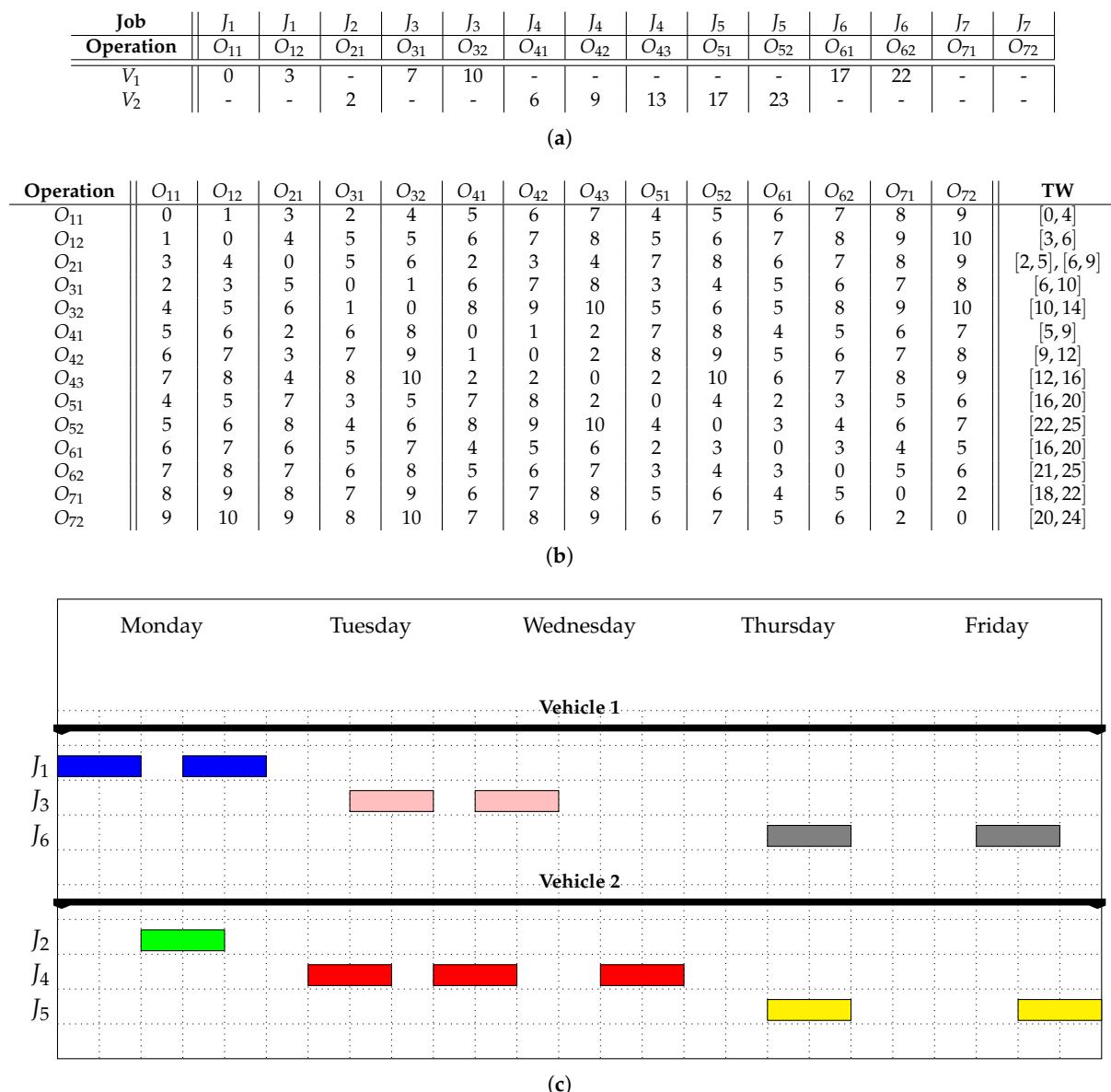


Figure 2. Illustrative example with two vehicles and six jobs. (a) Processing times for each job and operation. (b) Time/distance matrix between operations and time windows. (c) Gantt chart of the vehicle schedules where each color corresponds to a different job.

Before presenting the detailed linear mathematical formulation of the problem, the reader is referred to Notations, which summarizes the notation for all sets, parameters, and decision variables necessary to understand the model. This table provides a comprehensive overview that facilitates comprehension and ensures clarity when interpreting the subsequent formulation.

Thus, the objective function of the problem is formulated to minimize the total cost over the planning horizon, which includes both travel costs and penalties for unscheduled jobs. The objective function explicitly accounts for these components, ensuring that the scheduling of services is optimized while unscheduled jobs are penalized accordingly.

$$\begin{aligned}
 \min \quad & c \cdot \sum_{v \in V} \sum_{p,q \in O} d_{pq} \cdot y_{pqv} + \sum_{j \in J} c_j - \sum_{v \in V} \sum_{j \in J} c_j \sum_{p \in O_j} x_{pv} \\
 & = \sum_{j \in J} c_j + \sum_{v \in V} (c \cdot \sum_{p,q \in O} d_{pq} \cdot y_{pqv} - \sum_{j \in J} c_j \sum_{p \in O_j} x_{pv})
 \end{aligned} \tag{1}$$

Furthermore, the following constraints must be satisfied:

- Each job is assigned to exactly one vehicle.

$$\sum_{p \in O} x_{pv} \leq 1 \quad \forall v \in V \quad (2)$$

$$x_{pv} = x_{qv} \quad \forall p, q \in O_j \quad \forall j \in J \quad \forall v \in V \quad (3)$$

- A vehicle cannot perform a job if the two are incompatible. This ensures that all assignments respect compatibility constraints between jobs and vehicles.

$$x_{jv} \leq C_{jv} \quad \forall j \in J \quad \forall v \in V \quad (4)$$

- All operations must respect their respective time windows:

$$\sum_{k=1}^{n_p} z_{pk} = \sum_{v \in V} x_{pv} \quad \forall p \in O \quad (5)$$

$$s_p \geq st_p^k \cdot z_{pk} \quad \forall k \in \{1, \dots, n_p\} \quad \forall p \in O \quad (6)$$

$$- + s_p + D_p \leq et_p^k \cdot z_{pk} + M \cdot (1 - z_{pk}) \quad \forall k \in \{1, \dots, n_p\} \quad \forall p \in O \quad (7)$$

where M is a sufficiently large constant (big-M), which renders the constraint inactive whenever $z_{pk} = 0$.

- Each vehicle must execute its assigned operations sequentially, ensuring that no two operations overlap in time or with scheduled breaks.

$$(s_p + D_p + t_{pq} + BT \cdot |\{b \in B^v \mid s_p + D_p \leq b \leq s_q\}|) \cdot y_{pqv} \leq s_q \cdot y_{pqv} \quad \forall p, q \in O \quad \forall v \in V \quad (8)$$

Below, an equivalent linear form of the nonlinear constraints presented in (8) is given:

$$s_{pqv} \leq s_p \quad \forall p, q \in O \quad \forall v \in V \quad (9)$$

$$s_{pqv} \leq My_{pqv} \quad \forall p, q \in O \quad \forall v \in V \quad (10)$$

$$s_{pqv} \geq s_p - M(1 - y_{pqv}) \quad \forall p, q \in O \quad \forall v \in V \quad (11)$$

$$s_{pqv} \geq 0 \quad \forall p, q \in O \quad \forall v \in V \quad (12)$$

$$S'_{pqv} \leq s_q \quad \forall p, q \in O \quad \forall v \in V \quad (13)$$

$$S'_{pqv} \leq My_{pqv} \quad \forall p, q \in O \quad \forall v \in V \quad (14)$$

$$S'_{pqv} \geq s_q - M(1 - y_{pqv}) \quad \forall p, q \in O \quad \forall v \in V \quad (15)$$

$$S'_{pqv} \geq 0 \quad \forall p, q \in O \quad \forall v \in V \quad (16)$$

$$v_{pbqv} \leq y_{pqv} \quad \forall p, q \in O \quad \forall v \in V \quad \forall b \in B^v \quad (17)$$

$$v_{pbqv} \leq w_{pbq} \quad \forall p, q \in O \quad \forall v \in V \quad \forall b \in B^v \quad (18)$$

$$v_{pbqv} \geq y_{pqv} + w_{pbq} - 1 \quad \forall p, q \in O \quad \forall v \in V \quad \forall b \in B^v \quad (19)$$

$$v_{pbqv} \geq 0 \quad \forall p, q \in O \quad \forall v \in V \quad \forall b \in B^v \quad (20)$$

$$s_{pqv} + (D_p + t_{pq}) y_{pqv} + BT \sum_{b \in B^v} v_{pbqv} \leq S'_{pqv} \quad \forall p, q \in O \quad \forall v \in V \quad (21)$$

$$s_p + D_p \leq b + M(1 - w_{pbq}) \quad \forall p, q \in O \quad \forall v \in V \quad \forall b \in B^v \quad (22)$$

$$b \leq s_q + M(1 - w_{pbq}) \quad \forall p, q \in O \quad \forall v \in V \quad \forall b \in B^v \quad (23)$$

$$, m.bn{b} \quad (24)$$

where M is a sufficiently large constant (big-M), which renders some constraints inactive whenever $w_{pbq} = 0$ and others whenever $y_{pqv} = 0$.

- The operating range between operations of a vehicle must be satisfied.

$$d_{pq} \cdot y_{pqv} \leq OR_p \quad \forall p, q \in O \mid p \in O_i, q \in O_j, i \neq j \quad \forall v \in V \quad \forall i, -j \in J \quad (25)$$

- Vehicles must comply with labor regulations:
 - Vehicles must not take breaks during the execution of an operation:

$$\begin{cases} s_p - BT - b \geq \epsilon - M \cdot (1 - x_{pv}), & \text{if } b > s_p, \\ b - (s_p + D_p) \geq \epsilon - M \cdot (1 - x_{pv}), & \text{if } b \leq s_p, \end{cases} \quad \forall p \in O \quad \forall b \in B^v \quad \forall v \in V \quad (26)$$

where M is a sufficiently large constant (big-M), and ϵ is a small positive number to avoid exact equality.

- Vehicles must not exceed the maximum working time.

$$b - (b' + BT) \leq WT \quad \forall \{b, b'\} \in B^v \quad \forall v \in V \quad (27)$$

- Vehicles must not exceed the maximum driving time.

CDT_b^v : Cumulative driving time of vehicle v until the break b

$$CDT_b^v = \sum_{p,q \in O | s_q \leq b} t_{pq} y_{pqv} \quad \forall b \in B^v \quad \forall v \in V \quad (28)$$

DTB_b^v : Driving time of vehicle v between the break b and the previous break

$$DTB_b^v = CDT_b^v - \max_{b' \in B^v | b' \leq b} (CDT_{b'}^v) \leq WT \quad \forall b \in B^v \quad \forall v \in V \quad (29)$$

The nonlinear constraint is reformulated in an equivalent linear form:

$MCDT_b^v$: Maximum cumulative driving time of vehicle v up to break b

$$MCDT_b^v \geq CDT_{b'}^v \quad \forall b' \in B^v \quad b' < b, \forall v \in V \quad (30)$$

DTB_b^v : Driving time of vehicle v between break b and the previous break

$$DTB_b^v = CDT_b^v - MCDT_b^v \leq DT \quad \forall b \in B^v \quad \forall v \in V \quad (31)$$

It is worth noting that, although the model has been fully linearized, the constraints derived from the real-world case study result in a large and complex formulation, highlighting the practical challenges of representing operational conditions within an optimization framework.

3. Proposed Approaches

This section presents two different methodologies that have been implemented to tackle the real-world challenge addressed in this paper. Both approaches handle the problem from a different perspective.

In the first approach, presented in Section 3.1, we address the comprehensive problem by means of a scheduling problem formulation, including all the problem peculiarities. For example, even though in most of the scheduling problems the objective function is to minimize the makespan, i.e., the maximum completion time, we focus on minimizing the total cost by means of the maximization of the number of completed jobs since the company does not need to schedule all jobs. In this methodology, we solve the whole problem without considering any simplification to speed it up. It is important to emphasize that this first approach is proposed mainly as a formulation of the problem and is intended to serve as a benchmark. While it provides a comprehensive view of the problem and robust solutions, its computational cost can be high.

The second approach, described in Section 3.2, is designed to solve the problem by means of an enumeration procedure that generates initial feasible solutions, followed by a straightforward binary linear programming model that selects the best one. This methodology is implemented by using a partition of the whole planning horizon into periods, and it assigns jobs to vehicles in each period recursively. Then, once the jobs are allocated to vehicles in every period, an exact algorithm determines the best option to schedule the vehicles that will be used in the whole planning horizon.

Both approaches have been implemented using Hexaly (Hexaly is a global optimization solver that offers nonlinear and set-oriented modeling APIs). This solver integrates

a wide range of exact and heuristic methods to tackle complex optimization problems. From classical techniques like Simplex and Interior-Point to advanced approaches such as surrogate modeling, large neighborhood search, and statistical learning for autotuning, its flexibility allows it to adapt to various problem types. We refer the reader to [20] or [21] to deeply understand these procedures. It is worth mentioning that, although we use Hexaly's Constraint Programming APIs and thus have control over the modeling of constraints, the internal solver operates as a black-box optimizer.

3.1. Holistic Algorithm

We name this proposal the *Holistic algorithm* since the entire problem is modeled by incorporating in the scheduling problem all particular constraints considered by the transportation company to obtain high-quality solutions. As previously introduced, the solution obtained using this algorithm will be used as a benchmark solution for comparison purposes. Therefore, with this comprehensive perspective, we are able to address the entire problem.

We have opted by Hexaly to implement the Holistic algorithm because of its advanced capabilities to model features representing realistic situations. The implementation leverages the Hexaly solver, utilizing its decision variable structures, such as list-type variables used to define the operations assigned to each vehicle, and interval-type variables employed to represent specific service times; see [22]. The use of these specialized structures significantly reduces the number of decision variables, thereby accelerating the optimization process by allowing more iterations within the same computational time. Furthermore, when modeling the problem using Hexaly, these variables can be employed as parameters in the definition of constraints, simplifying both the development and interpretation of the model.

Next, the mathematical formulation presented in Section 2 is implemented as a computational model using Hexaly. Again, we refer the reader to Notations, which summarizes the notation for all sets, parameters, and decision variables necessary to implement the Holistic approach.

As in the formulation proposed in Section 2, the objective function (1) of the considered problem minimizes the total cost incurred when scheduling the service over the considered planning horizon. This function is implemented in Hexaly as follows:

$$\min \quad c \cdot \sum_{v \in V} \sum_{\{p,q\} \subseteq R^v} d_{pq} + \sum_{j \in J} c_j - \sum_{v \in V} \sum_{j \in R^v} c_j \quad (32)$$

$$= \sum_{j \in J} c_j + \sum_{v \in V} (c \cdot \sum_{\{p,q\} \subseteq R^v} d_{pq} - \sum_{j \in R^v} c_j) \quad (33)$$

To solve this problem using Hexaly, it is necessary to implement the following constraints:

- Each job is assigned to exactly one vehicle, so the sets $\{R^v\}_{v \in V}$ must be mutually exclusive. See Constraints (2) and (3) presented in Section 2.
- A vehicle cannot perform a job if the two are incompatible, in accordance with Constraint (4) in Section 2.

$$C_{jv} = 0 \implies j \notin R^v \quad \forall j \in J \quad \forall v \in V \quad (34)$$

- All operations must respect their respective time windows, as stated in Constraints (5)–(7), Section 2.

$$\bigvee_{k \in \{1, \dots, n_p\}} (s_p \geq st_p^k) \wedge (e_p \leq et_p^k) \quad \forall p \in O \quad (35)$$

- Each vehicle must execute its assigned operations sequentially, ensuring that no two operations overlap in time, nor overlap with scheduled breaks, as defined in Constraint (8), Section 2.

$$e_p + t_{pq} + BT \cdot |\{b \in B^v \mid e_p \leq b \leq s_q\}| \leq s_q \quad \forall \{p, q\} \subseteq \bigcup_{j \in R^v} O_j \quad \forall v \in V \quad (36)$$

- The duration of each operation is computed as follows, subtracting the end time and the start time. These are Hexaly-specific constraints, which define the length of the time interval for each operation.

$$e_p - s_p = D_p \quad \forall p \in O$$

- The operating range between operations of a vehicle must be fulfilled, as defined in Constraint (25), Section 2.

$$d_{pq} \leq OR_p \quad \forall \{p, q\} \subseteq \bigcup_{j \in R^v} O_j \mid p \in O_i, q \in O_j, i \neq j \quad \forall v \in V \quad (37)$$

- Vehicles must comply with labor regulations:

- Vehicles must not take breaks during the execution of an operation; see Constraint (26):

$$(s_p - BT > b) \vee (b > e_p) \quad \forall b \in B^v \quad \forall p \in \bigcup_{j \in R^v} O_j \quad \forall v \in V \quad (38)$$

- Vehicles must not exceed the maximum working time, as specified in Constraint (27):

$$b - (b' + BT) \leq WT \quad \forall \{b, b'\} \in B^v \quad \forall v \in V \quad (39)$$

- Vehicles must not exceed the maximum driving time, as defined in Constraint (29):

$$CDT_b^v = \sum_{\{p, q\} \in \bigcup_{j \in R^v} O_j \mid s_q \leq b} t_{pq} \quad \forall b \in B^v \quad \forall v \in V \quad (40)$$

$$DTB_b^v = CDT_b^v - \max_{b' \in B^v \mid b' < b} (CDT_{b'}^v) \leq DT \quad \forall b \in B^v \quad \forall v \in V \quad (41)$$

3.2. Divide-and-Conquer Algorithm

The proposed methodology is an efficient two-stage hybrid approach named Divide-and-Conquer (D&C) algorithm, designed to tackle the complex scheduling problem and yield a near-optimal solution. This approach strategically decomposes the problem into sequential stages: greedy heuristic for initial sequence generation and binary linear programming (BLP) for optimized selection. This integration allows for effective exploration of the solution space, rigorously enforcing all constraints while systematically enhancing solution quality.

3.2.1. Greedy Heuristic

The initial stage employs a greedy heuristic procedure to create feasible sequences of jobs to vehicles. This process is instrumental in establishing the foundation for subsequent

optimization. The heuristic is organized into three main phases: initialization, expansion, and finalization.

The algorithm begins by identifying all candidate jobs that can feasibly serve as the first job for each vehicle departing from its depot. A job is considered feasible if it satisfies three essential conditions:

1. Compatibility: the job is compatible with the vehicle.
2. Operation range: the location of the first operation of the job is within the maximum operating range of the vehicle (the maximum distance a vehicle can travel empty from its current location).
3. Time window: the vehicle can reach the location of the job within a specified time window.

Once the initial set of feasible candidate jobs is enumerated, the expansion stage begins the iterative process of sequence construction. For each initial candidate job, a preliminary sequence is created and added to the expansion list, which dynamically tracks all sequences currently being developed. The algorithm then iteratively extracts a sequence from the expansion list, identifying the last job in the sequence. Then, a new list of unplanned candidate jobs is generated based on spatial and temporal feasibility relative to the last job in the sequence. For each valid candidate, the current sequence is extended and the new and longer sequence is reinserted into the expansion list.

During this sequence generation, each time a new job is appended, the sequence can transition into one of three states:

1. Feasible and expanding: the sequence remains on the expansion list as it can be further extended.
2. Final sequence: the sequence must terminate with a return to the depot location (for instance, if the remaining time is insufficient for a new job), and it is moved to the final set of solutions.
3. Infeasible: the sequence violates any constraint and it is immediately discarded.

The finalization stage collects all sequences that can no longer be feasibly expanded. To manage computational complexity, a parameter called N_{seq} is defined, which sets a predefined limit on the total number of constructed sequences, halting generation once this threshold is exceeded.

To ensure operational realism and efficiency, the heuristic incorporates several critical mechanisms. Any candidate job requiring the vehicle to arrive more than 24 h before the time window opens is immediately excluded, focusing sequence growth on more efficient solutions. The total driving time and total working time since the last break are continuously tracked through counters. If assigning a new candidate job causes either counter to exceed the permitted maximum, a break is inserted, resetting both counters. Sequences that omit explicitly required mandatory jobs are automatically discarded during generation, ensuring all constructed sequences meet core operational requirements. The candidate lists are generated dynamically. From the depot, jobs are filtered by compatibility, operating range, and time window and then sorted by the earliest possible arrival time. Following a job, the algorithm evaluates the empty travel distance and transition time between the last operation of the current job and the first operation of the next job, ensuring time window compatibility.

3.2.2. Optimized Selection via Binary Linear Programming

The second stage is responsible for the optimal selection of which generated sequences are ultimately scheduled for each vehicle. The greedy heuristic provides the set of all feasible sequences, but the scheduling process itself is optimized using an exact algorithm, a binary linear programming (BLP) formulation, which aims to minimize the total cost.

Notice that the optimal selection is performed over the generated sequences and therefore does not guarantee global optimality of the solution since the sequences have been generated heuristically.

The objective is to find a schedule that minimizes the total operating cost, which includes the cost of executed services and the penalty cost for unplanned jobs. The reader is referred to Notations for the complete notation.

$$\min \quad c \cdot \sum_{v \in V} \sum_{i \in S^v} \gamma_i^v \cdot d_i^v + \sum_{j \in J} c_j \cdot (1 - \sum_{v \in V} \sum_{i \in S_j^v} \gamma_i^v) \quad (42)$$

$$\text{s.t.} \quad \sum_{i \in S^v} \gamma_i^v \leq 1 \quad \forall v \in V \quad (43)$$

$$\sum_{v \in V} \sum_{i \in S_j^v} \gamma_i^v \leq 1 \quad \forall j \in J \quad (44)$$

$$\gamma_i^v \in \{0, 1\} \quad \forall i \in S^v \quad v \in V \quad (45)$$

Equation (42) represents the total incurred cost (objective function), encompassing the cost of transportation services (first term) plus the penalty cost for any job that is not scheduled (second term). Equation (43) ensures that each vehicle $v \in V$ is assigned to at most one sequence $i \in S^v$. Equation (44) enforces that each job $j \in J$ is scheduled by at most one vehicle across all selected sequences. Finally, Equation (45) defines the decision variables γ_i^v as binary, where $\gamma_i^v = 1$ if sequence i is assigned to vehicle v and 0 otherwise.

3.2.3. Incremental D&C Strategy

Given that the exhaustive enumeration of all possible sequences grows exponentially with the problem size, the D&C algorithm implements an incremental planning strategy controlled by parameter T , which defines how the planning horizon is partitioned into periods. In our case, in order to avoid significant suboptimal solutions, we implemented the following T values:

- Full-horizon planning ($T = 1$): the algorithm solves the problem over the entire planning horizon, generating sequences that span the full horizon.
- Two-period incremental strategy ($T = 2$): the planning horizon is divided into two periods. The algorithm first solves the problem for the initial two days and then extends the resulting partial plan to cover the remainder of the planning horizon.
- Three-period incremental approach ($T = 3$): the most aggressive partitioning starts by generating sequences containing a single initial job, subsequently expanding them to cover the first three days, and finally extending them to the full planning horizon.

The two stages (greedy algorithm and exact BLP algorithm) are repeated iteratively until all partitioned periods within the planning horizon have been optimized. Crucially, the optimal scheduling decisions made in previous periods condition the starting location and time for the subsequent sequence generation of the period.

The iterative nature of the D&C approach is graphically illustrated in Figure 3. We have a planning horizon divided into two periods ($T = 2$), the first period comprises Monday (M), Tuesday (T), and Wednesday (W), while the second period includes the remaining days, Thursday (T) and Friday (F). The process is sequential: first, the greedy heuristic enumerates feasible sequences for the first period (each one represented using lines of different colors and patterns), and then the BLP selects the optimal sequence. The end point of this optimal sequence (depicted as a smooth grey line) becomes the new starting point for repeating both stages in the second period, yielding a complete optimized schedule across the entire horizon.

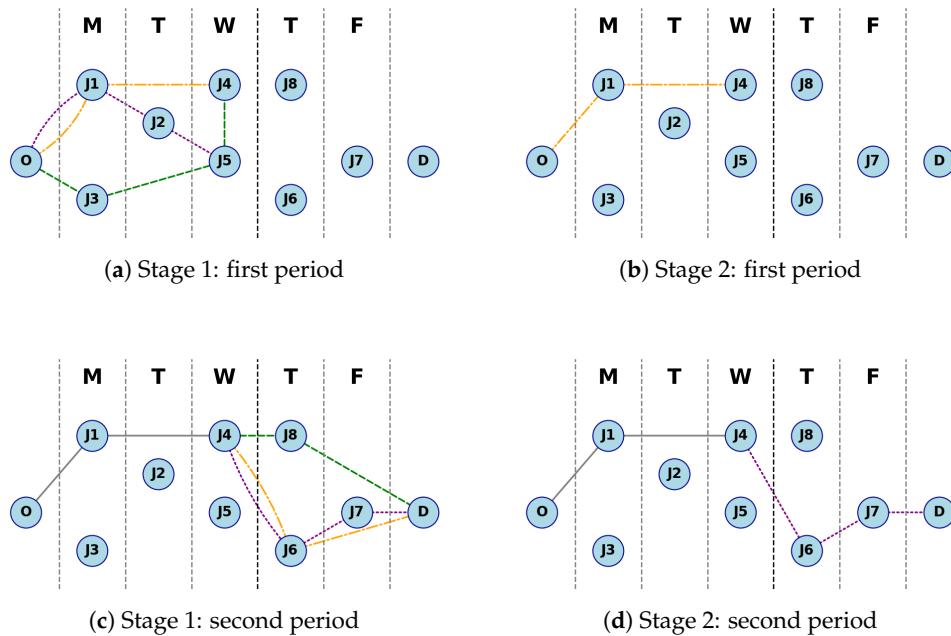


Figure 3. D&C schema.

4. Computational Results

In this section, we present and discuss the outcomes of the two approaches described in Section 3. They have been executed via a computer equipped with an AMD Ryzen 9 5950x 16-core processor (32 threads, AMD Inc., Santa Clara, CA, USA) and 128 GB of RAM. Both approaches have been implemented using Hexaly 13.0.2.

This section is structured as follows. First, the real-world instances considered in the study are described. Then, several scenarios are defined to enable the comparison between the two proposed approaches. Finally, a detailed case study is presented and discussed.

We would like to remark that, to solve every instance using the Holistic algorithm, we establish a time limit for the solver of 30 min (1800 s). In contrast, when applying the D&C algorithm, the total time is allocated across the different periods of the planning horizon and, in turn, between the two defined stages (greedy algorithm and exact algorithm). Here, the total time limit of 1800 s is evenly distributed across the periods. When the greedy algorithm requires a significant portion of the computational time, 60 s are reserved to guarantee the execution of the exact algorithm.

4.1. Description of the Instances

We will solve 16 real-world instances provided by the transportation company. In all 16 instances, jobs required by customers are located all around Spain, specifically along the Iberian Peninsula. An example of the distribution of operations on the peninsula can be seen in Figure 4, where the operations of instance 1 are depicted, each circle represents a single operation.

These original 16 real-world instances have different characteristics. Operations have to be scheduled to satisfy a hard time window in a specific day. We say that such operations have a *fixed appointment*. However, other operations are allowed to be scheduled considering soft time windows specifically, at any time within the opening hours during the appointment day, that is, with an associated wider time window. Furthermore, some operations have a *flexibility level* regarding the day to be scheduled. We have codified flexibility 0, named *null flexibility*, if the operation may be scheduled in a specific day and flexibility 1, named *a day flexibility*, if the operation may be scheduled one day before or one

day after the original appointment, all eligible days being equally valid for assignment. It is important to note that these flexibilities apply exclusively to operations that are marked with this flexibility level and do not affect other operations or the depot.



Figure 4. Operations' distribution.

Table 2 includes the main characteristics of the considered instances, that is, number of jobs paid by customers (column 2), number of operations (column 3), number of operations that have fixed appointments (column 4), and, finally, the percentage of operations with fixed appointments (column 5). Note that, in the original instances of Table 2, all operations have null flexibility.

Table 2. Instance characteristics.

Instance	#Jobs	#Operations	#Fixed Appointment	% Fixed Appointment
1	79	158	87	55.06%
2	76	155	93	60.00%
3	80	160	122	76.25%
4	70	146	120	82.19%
5	69	143	101	70.63%
6	87	175	135	77.14%
7	115	235	173	73.62%
8	124	257	162	63.04%
9	133	270	180	66.67%
10	147	306	189	61.76%
11	143	300	165	55.00%
12	130	277	102	36.82%
13	16	32	24	75.00%
14	132	280	80	28.57%
15	139	280	90	32.14%
16	157	334	124	37.13%

To test a richer set of instances, we have modified the characteristics of the original set of 16 instances, resulting in a total of 64 instances. Then, the first subset is the original set of instances. The second set assumes that all operations have dynamic flexibility, that is, they could be scheduled one day before and after the original appointment. The third set considers that all operations have no fixed appointment while preserving the original flexibility level, that is, regarding the day to be scheduled. Finally, the last set removes the fixed appointment and the flexibility level, so here we are allowing the maximum flexibility to schedule the operations.

Table 3 includes descriptive statistics for the four subsets of 16 instances regarding the jobs requested by customers, providing an overview of the instance sizes. The first column of the table shows the metrics: the mean number of jobs (*Mean*); the standard deviation of the number of jobs (*Std*); the minimum number of jobs (*Min*); the first quartile (*Q1*), representing the number of jobs that 25% of the customers requested at most; the median

(Me), representing the number of jobs that 50% of the customers requested at most; the third quartile ($Q3$), representing the number of jobs that 75% of the customers requested at most; and the maximum number of jobs (Max). Columns 2, 3, 4, and 5 detail these metrics for the four subsets, named *original*, *a day flexibility*, *no fixed appointment*, and *maximum flexibility*.

Table 3. Descriptive statistics of number of jobs for the set of instances.

	Original	A Day Flexibility	No Fixed Appointment	Maximum Flexibility
Mean	106.06	107.25	108.44	110.69
Std	38.78	39.22	38.83	40.82
Min	16.00	16.00	16.00	16.00
Q1	78.25	79.00	80.25	81.75
Me	119.50	120.00	122.50	123.00
Q3	134.50	135.00	138.25	138.25
Max	157.00	158.00	155.00	163.00

As can be observed in Table 3, the four subsets exhibit similar descriptive statistics to avoid introducing bias when comparing them. The observed difference in the number of jobs is attributable to a validation process, which removes inaccessible jobs, i.e., those that cannot be reached due to distance or time constraints. Consequently, instances with more flexible time windows result in a greater number of valid jobs.

4.2. Parameter Selection and Tuning

The Holistic algorithm does not require parameter tuning for its application, whereas the D&C algorithm involves several parameters that must be tuned. Before proceeding with a comparative evaluation of the two algorithms, a dedicated subsection is provided to that end for the D&C approach.

The first experiment focuses on establishing the appropriate number of periods in the planning horizon, T ; see Section 3.2.3. To that end, and in order to avoid overfitting, this preliminary experimentation is performed over a subset of 11 representative instances from a total of 64 instances. As our case study considers a planning horizon of five working days within a week, we have executed the algorithm for one, two, and three periods, that is, $T = 1$, $T = 2$, and $T = 3$, respectively; see Figure 5.

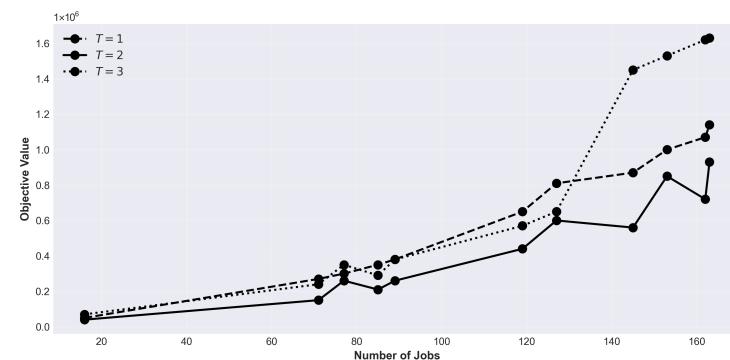


Figure 5. Objective function values across different numbers of periods.

The results presented in Figure 5 indicate that $T = 2$ consistently yields the best solutions. This outcome can be explained as follows: for $T = 1$, the problem is less granular but more complex to solve, resulting in worse solutions within the same computation time; in contrast, for $T = 3$, solutions are too granular and obtained faster but are suboptimal.

The second experiment, depicted in Figure 6, investigates the practical limit for the number of sequences generated per period, N_{seq} . A preliminary test was conducted, varying N_{seq} between 5000 and 500,000, in order to evaluate the impact on computation time and system stability.

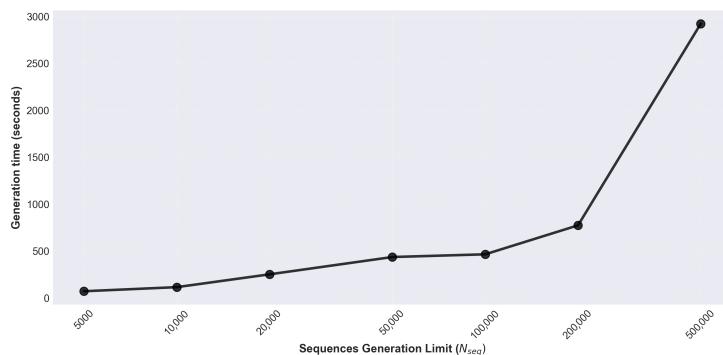


Figure 6. Trade-off between generation time and sequence generation limit.

Figure 6 shows the mean trade-off between generation time and sequence generation limit for our subset of instances. As shown in Figure 6, setting the limit to 100,000 sequences per period provides a reasonable balance between solution quality and computation time. It is worth noting that, during experimentation, memory usage was observed to exceed 90% when generating 200,000 sequences or more. This fact highlights the potential risk of instability for higher sequence limits.

Based on these results, for the subsequent experiments, we will set the planning horizon to two periods, $T = 2$, and the maximum number of sequences generated per period is limited to one hundred thousand, $N_{seq} = 100,000$.

4.3. Comparison Among Scenarios

Once the instances have been described, we will move on to solve them with the aim to compare the two approaches proposed in this paper. However, we have considered different scenarios to cover all the possibilities that can be encountered in a real situation.

- The first scenario considers that there are no mandatory jobs and labor regulations are not respected. Results are shown in Figure 7a and Table 4.
- The second scenario assumes that there are mandatory jobs and we do not respect the labor regulations. Results are included in Figure 7b and Table 5.
- The third scenario considers that there are no mandatory jobs but we respect the labor regulations. Results are enclosed in Figure 7c and Table 6.
- The last scenario supposes that there are mandatory jobs and labor regulations are respected. Results are included in Figure 7d and Table 7.

Tables 4–7 include the same metrics by rows as Table 3: Mean, Std, Min, Q1, Me, Q3, and Max related to the number of jobs that have been planned (#Jobs), the number of vehicles needed to plan such jobs (#Vehicles), the total cost measured in thousands of EUR, i.e., the objective function value, and, finally, the time required to obtain the best objective function value, i.e., the time to best (in seconds) denoted by TTB. These metrics are computed in each of the four considered scenarios and by each algorithm (by column). In all tables, we can observe that greater flexibility in the instances requires longer execution times for their resolution using any of the two methodologies than using the original instances. Additionally, we note that, on average, the D&C algorithm achieves better results than the Holistic algorithm in significantly shorter execution times in all the sets of instances. Furthermore, the D&C algorithm uses fewer vehicles to plan the jobs to be performed, which can be translated into greater customer satisfaction and resource savings for the transport company. It is important to note that the cost of vehicles is not included in the objective function as the company owns its vehicles, but, if a vehicle is not in use, the company employee can perform other work activities.

Next, we will briefly discuss each of the tables separately.

If we focus on Table 4, the D&C algorithm outperforms the Holistic algorithm in all metrics: number of jobs that have been planned, number of vehicles needed to plan such jobs, total cost, and time required to obtain the best objective function value. Then, in the first scenario, we observe greater stability in all metrics. In this scenario, the D&C algorithm outperforms the Holistic algorithm in 45 out of the 64 instances, approximately 70% of the instances.

If we look at Table 5, while the D&C algorithm schedules a greater number of customer jobs, utilizing fewer vehicles and achieving faster computation times, it fails to adequately prioritize mandatory jobs. This shortcoming results in significant penalties in the objective function, thereby explaining its inferior overall performance compared to the Holistic algorithm in this scenario. However, even though the objective function value is worse, we cannot necessarily conclude that the D&C algorithm underperforms the Holistic algorithm since more jobs are planned and fewer vehicles are used, which translates into customer satisfaction and better resource utilization since those vehicles can be used for other types of duties.

As shown in Table 6, despite having more flexibility, the constraints regarding the breaks means that we have a more restrictive and rigid operation in both methodologies. Even in this situation, the objective function value of the D&C algorithm is better than that of the Holistic one, outperforming it in 58 out of the 64 instances, approximately 90% of the instances. Note that the execution time is less sensitive to flexibility, especially in the Holistic methodology.

Finally, when we examine in more detail in Table 7, we observe that, in both methodologies, greater flexibility in the set of instances contributes to an improvement in the objective function value. Furthermore, we find that, despite the improvement in the Holistic algorithm in this fourth scenario, the D&C algorithm still plans a higher number of jobs with fewer vehicles and in less time, as in the second scenario. However, since it does not sufficiently prioritize the mandatory jobs during the construction phase, it is heavily penalized by the objective function definition.

Figure 7 displays bar charts and line charts. In the bar charts, gray bars and black bars represent the runtime (in seconds) of the Holistic algorithm and the D&C algorithm, respectively. In the line charts, the solid line and the dashed line represent the objective function value (total cost in thousands of EUR) obtained by the Holistic algorithm and the D&C algorithm, respectively.

A convergence study was conducted to analyze the evolution of the objective value over the execution time of both methodologies, as illustrated in Figure 8. The x-axis shows the time, and the y-axis depicts the objective function value at every time. The resulting plots reveal distinct behaviors in their search dynamics. The Holistic focuses on continuous incremental improvements, achieving substantial gains early, while later enhancements become increasingly marginal. This persistent search for minor improvements explains its higher TTB. Conversely, the D&C explores a discrete and finite search space, which leads to faster stabilization of the objective value and consequently lower TTB. Additionally, the D&C's greedy heuristic stage delays the appearance of initial objective evaluations, so the initial feasible solutions begin to appear later along the computational horizon (see the x-axis).

Table 4. Descriptive statistics of Scenario 1.

	Original		A Day Flexibility		No Fixed Appointment		Maximum Flexibility	
	Holistic	D&C	Holistic	D&C	Holistic	D&C	Holistic	D&C
#Jobs	Mean	55.69	61.06	56.06	62.44	57.62	65.00	58.19
	Std	13.45	18.95	13.78	20.23	13.41	19.12	14.47
	Min	14.00	13.00	14.00	12.00	15.00	14.00	15.00
	Q1	51.75	52.00	51.75	54.00	53.75	57.00	52.00
	Me	58.50	62.50	59.00	64.00	59.00	65.00	60.00
	Q3	62.50	69.25	64.25	70.25	66.00	74.50	67.25
	Max	73.00	95.00	74.00	102.00	75.00	98.00	76.00
#Vehicles	Mean	13.50	12.81	13.06	12.69	13.00	11.19	12.38
	Std	2.80	2.51	2.74	2.75	2.58	2.54	2.70
	Min	4.00	5.00	4.00	4.00	4.00	4.00	4.00
	Q1	13.75	12.00	12.75	12.00	12.75	10.75	11.75
	Me	14.50	13.00	13.50	13.50	14.00	11.00	13.00
	Q3	15.00	15.00	15.00	14.25	14.00	13.00	14.00
	Max	15.00	15.00	15.00	15.00	15.00	15.00	15.00
Total Cost	Mean	522.41	473.01	530.00	471.73	526.21	457.37	542.40
	Std	284.47	229.05	294.28	231.95	288.62	234.39	301.94
	Min	24.31	35.61	24.31	44.31	14.85	24.25	14.85
	Q1	280.04	271.37	271.71	283.31	287.63	262.29	288.50
	Me	598.74	543.65	582.05	545.00	607.51	504.57	621.28
	Q3	752.05	662.25	766.85	651.31	779.18	645.90	796.71
	Max	952.27	824.64	1011.54	830.13	902.85	842.28	931.60
TTB	Mean	480.00	36.94	766.56	252.50	800.44	189.75	1237.50
	Std	518.23	61.25	693.35	275.40	517.05	162.94	558.27
	Min	1.00	1.00	1.00	0.00	7.00	1.00	121.00
	Q1	58.75	2.50	53.00	6.75	441.75	70.25	989.25
	Me	252.00	7.00	713.50	181.00	764.50	166.50	1404.00
	Q3	922.00	37.25	1439.50	403.75	1255.25	303.25	1692.75
	Max	1713.00	223.00	1688.00	702.00	1688.00	495.00	1788.00

Table 5. Descriptive statistics of Scenario 2.

	Original		A Day Flexibility		No Fixed Appointment		Maximum Flexibility	
	Holistic	D&C	Holistic	D&C	Holistic	D&C	Holistic	D&C
#Jobs	Mean	55.94	61.88	55.88	59.81	57.56	64.00	57.62
	Std	13.26	19.88	12.85	19.33	13.54	21.05	13.84
	Min	15.00	8.00	15.00	8.00	15.00	10.00	14.00
	Q1	52.50	53.25	53.50	54.50	52.00	53.75	52.75
	Me	59.50	64.50	59.50	62.00	62.50	65.50	60.50
	Q3	63.25	73.00	63.00	71.75	65.25	73.00	64.00
	Max	71.00	94.00	70.00	88.00	72.00	103.00	74.00
#Vehicles	Mean	15.00	13.06	15.00	11.81	15.00	11.19	15.00
	Std	0.00	3.04	0.00	3.02	0.00	2.76	0.00
	Min	15.00	3.00	15.00	3.00	15.00	3.00	15.00
	Q1	15.00	12.75	15.00	11.00	15.00	10.50	15.00
	Me	15.00	14.00	15.00	12.00	15.00	12.00	15.00
	Q3	15.00	15.00	15.00	13.25	15.00	13.00	15.00
	Max	15.00	15.00	15.00	15.00	15.00	14.00	15.00
Total Cost	Mean	1925.64	2662.44	1937.21	2880.33	1932.70	2707.73	1953.96
	Std	287.54	289.43	302.60	225.53	289.51	288.13	316.42
	Min	1423.79	2206.08	1423.79	2434.90	1424.45	2199.12	1434.20
	Q1	1688.09	2362.59	1687.63	2763.73	1695.01	2482.05	1685.57
	Me	1982.31	2723.14	1987.15	2868.24	1993.42	2706.44	2005.97
	Q3	2155.44	2873.66	2187.99	3044.90	2165.30	2899.00	2223.54
	Max	2358.11	3122.06	2413.53	3212.82	2311.22	3346.24	2416.13
TTB	Mean	685.62	41.62	1020.50	254.62	891.38	209.50	1329.56
	Std	544.19	73.88	679.07	276.98	624.70	172.23	485.55
	Min	1.00	1.00	1.00	1.00	1.00	1.00	2.00
	Q1	362.50	4.00	276.75	7.75	331.75	84.00	1112.75
	Me	581.50	9.50	1151.50	182.50	936.50	196.50	1477.50
	Q3	1138.25	39.25	1667.25	405.75	1459.50	330.00	1703.25
	Max	1713.00	284.00	1743.00	712.00	1781.00	560.00	1768.00

Table 6. Descriptive statistics of Scenario 3.

	Original		A Day Flexibility		No Fixed Appointment		Maximum Flexibility	
	Holistic	D&C	Holistic	D&C	Holistic	D&C	Holistic	D&C
#Jobs	Mean	44.06	55.44	47.62	58.62	50.81	60.75	59.56
	Std	10.18	15.82	13.08	16.96	11.70	16.22	14.06
	Min	11.00	11.00	11.00	12.00	12.00	10.00	14.00
	Q1	42.75	48.50	42.50	50.75	48.75	55.00	54.50
	Me	45.50	55.50	45.50	59.50	53.50	64.00	63.00
	Q3	50.00	68.25	54.75	72.25	56.00	70.25	68.00
	Max	53.00	73.00	66.00	83.00	64.00	80.00	74.00
#Vehicles	Mean	13.88	13.38	13.94	13.81	13.88	12.31	14.38
	Std	2.73	2.73	2.79	2.54	2.83	2.57	2.50
	Min	4.00	4.00	4.00	5.00	4.00	4.00	4.00
	Q1	14.00	13.00	14.75	13.75	14.00	11.75	15.00
	Me	15.00	14.00	15.00	15.00	15.00	13.00	15.00
	Q3	15.00	15.00	15.00	15.00	15.00	14.00	15.00
	Max	15.00	15.00	15.00	15.00	15.00	15.00	15.00
Total Cost	Mean	637.22	527.24	615.18	508.47	595.49	498.29	534.57
	Std	319.03	260.97	297.08	247.06	310.90	258.24	302.72
	Min	52.94	52.83	52.94	43.97	44.18	63.04	24.48
	Q1	347.47	302.68	355.20	298.96	317.86	262.34	279.69
	Me	698.17	588.68	691.65	587.41	711.32	576.48	580.17
	Q3	901.60	702.43	814.75	708.77	892.84	697.94	788.04
	Max	1062.53	938.09	1101.41	909.56	932.78	861.34	1008.43
TTB	Mean	1107.88	5.94	988.06	187.50	1228.38	63.56	1226.00
	Std	546.75	9.96	528.47	273.59	515.86	91.39	527.87
	Min	3.00	1.00	2.00	1.00	235.00	1.00	108.00
	Q1	807.50	0.00	793.75	1.00	983.50	17.00	1066.50
	Me	1132.50	0.50	1055.00	23.50	1385.50	26.00	1472.00
	Q3	1578.50	5.00	1417.25	348.75	1629.75	62.25	1569.25
	Max	1784.00	37.00	1619.00	710.00	1786.00	327.00	1711.00

Table 7. Descriptive statistics of Scenario 4.

	Original		A Day Flexibility		No Fixed Appointment		Maximum Flexibility	
	Holistic	D&C	Holistic	D&C	Holistic	D&C	Holistic	D&C
#Jobs	Mean	40.31	54.75	43.00	57.06	47.62	61.56	55.81
	Std	11.40	16.04	13.76	18.46	10.32	17.56	12.20
	Min	12.00	8.00	5.00	7.00	13.00	10.00	15.00
	Q1	37.25	49.50	36.00	48.25	44.00	57.00	51.75
	Me	42.00	56.50	45.50	60.00	50.50	61.00	59.50
	Q3	49.25	64.25	53.00	70.00	53.00	71.50	62.00
	Max	54.00	76.00	62.00	80.00	59.00	86.00	68.00
#Vehicles	Mean	14.94	12.75	15.00	13.19	15.00	12.62	15.00
	Std	0.25	3.26	0.00	3.31	0.00	3.18	0.00
	Min	14.00	2.00	15.00	2.00	15.00	2.00	15.00
	Q1	15.00	12.00	15.00	12.00	15.00	12.00	15.00
	Me	15.00	13.00	15.00	15.00	15.00	13.00	15.00
	Q3	15.00	15.00	15.00	15.00	15.00	15.00	15.00
	Max	15.00	15.00	15.00	15.00	15.00	15.00	15.00
Total Cost	Mean	2111.82	2804.04	2085.53	2994.46	2077.38	2805.77	1983.24
	Std	336.80	296.08	307.29	334.49	348.87	258.09	324.08
	Min	1551.41	2241.27	1521.34	2478.10	1443.90	2272.65	1423.91
	Q1	1789.69	2563.69	1797.46	2701.30	1773.41	2670.96	1694.94
	Me	2169.89	2784.48	2144.43	3031.20	2166.75	2806.23	2030.40
	Q3	2401.51	3061.61	2301.25	3263.67	2315.94	2981.78	2283.56
	Max	2634.60	3241.57	2615.08	3613.53	2586.89	3227.35	2438.38
TTB	Mean	1025.06	6.62	936.12	191.38	1362.25	65.00	1421.19
	Std	635.39	10.26	469.39	278.41	488.65	92.61	295.10
	Min	2.00	1.00	121.00	1.00	576.00	1.00	761.00
	Q1	544.25	0.00	758.00	1.75	851.25	19.00	1289.75
	Me	1291.50	0.50	997.00	22.00	1646.50	31.00	1440.50
	Q3	1528.75	9.00	1141.00	357.75	1759.25	62.25	1655.00
	Max	1793.00	38.00	1664.00	730.00	1795.00	335.00	1771.00

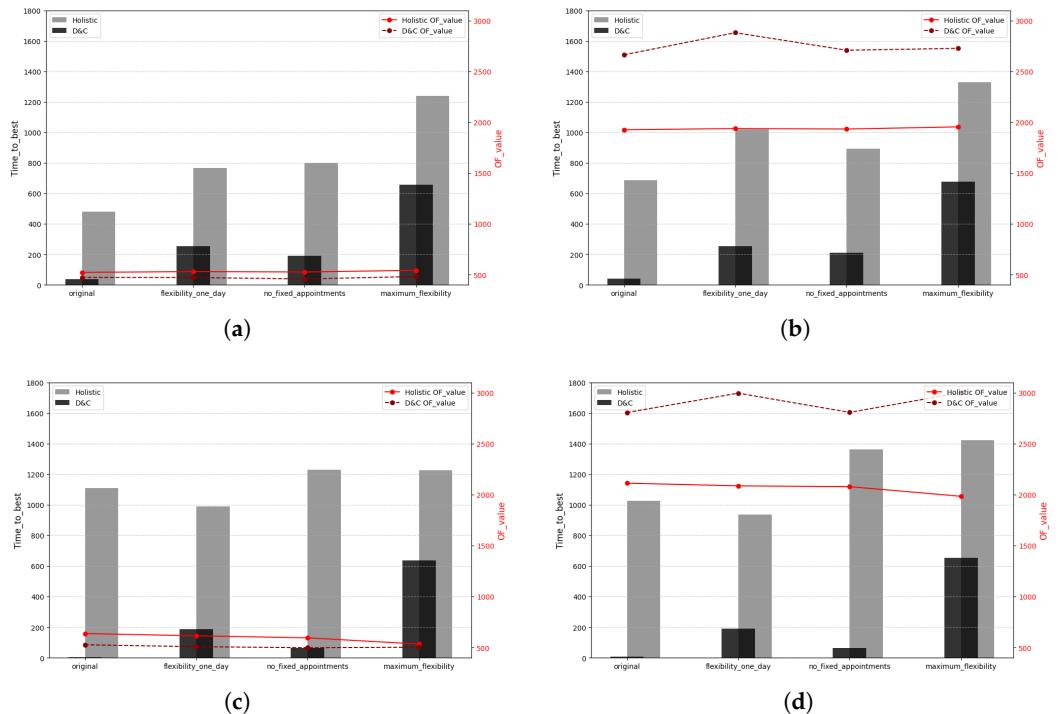


Figure 7. Comparison plots for Scenarios 1 to 4. (a) Scenario 1. (b) Scenario 2. (c) Scenario 3. (d) Scenario 4.

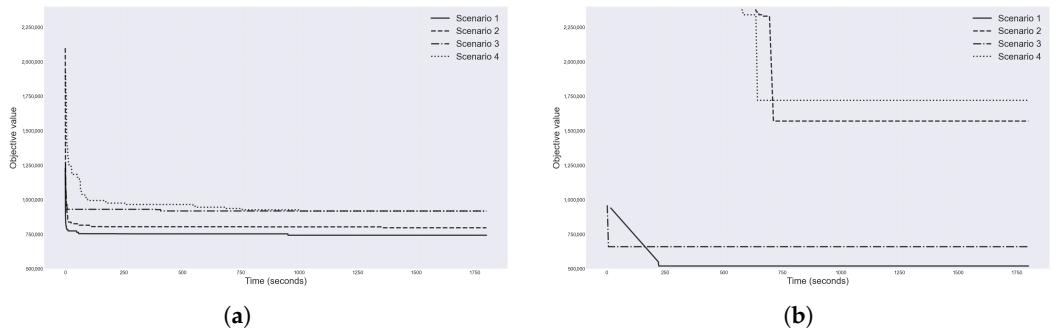


Figure 8. Convergence of the objective value for both algorithms. (a) Convergence of the Holistic algorithm. (b) Convergence of the D&C algorithm.

4.4. Case Study

In this section, we will study the behavior under four different scenarios in a single real instance; specifically, we have selected the week of 20 May 2024.

In this instance with one-hundred-fifteen customers that request a job, five of them are composed of three operations and the remaining two operations. To solve scenarios 2 and 4, a set of 29 mandatory jobs have been considered.

Table 8 shows the results of this case study under the different scenarios and with both methodologies. The first column specifies the scenario, the second column indicates the approach with which it has been resolved, columns 3 to 5 show the number of vehicles that have been scheduled during the planning horizon, the number of jobs scheduled in those vehicles, and the number of mandatory jobs scheduled, and columns 6 and 7 compute the total distance traveled (in thousands of kilometers) by all the vehicles and the total cost incurred (in thousands of EUR), which is our objective function. Finally, the last two columns detail the time to obtain the best solution, TTB, and the total time that the algorithm is being executed, TT, both in seconds. It is important to note that the total

cost and the distance are not correlated since the penalties for not completing certain jobs are high.

Table 8. Summary of the results obtained from the case study.

Scenario	Approach	#Vehicles	#Jobs	#Mandatory	Distance	Total Cost	TTB	TT
1	Holistic	14	61	0	20.03	560.03	29	1798
	D&C	12	64	0	24.14	534.15	6	6
2	Holistic	15	66	15	23.83	1913.83	603	1799
	D&C	12	67	7	26.91	2706.91	8	8
3	Holistic	15	50	0	19.12	669.12	1651	1799
	D&C	14	58	0	24.11	594.11	1	1
4	Holistic	15	45	15	23.34	2123.34	1633	1799
	D&C	11	51	5	17.07	3057.07	1	1

As we previously introduced, the first scenario considers no mandatory jobs and no breaks. Here, the D&C algorithm clearly outperforms the Holistic approach across every metric. It uses fewer vehicles (12 vs. 14), achieves a 5% lower objective function value in terms of costs, and it reaches the solution in just 6 s; see Scenario 1 in Table 8.

If we move on to the second scenario where mandatory jobs are now considered but still without scheduling breaks, we can state that the Holistic method performs more compulsory jobs (fifteen vs. seven), resulting in a better objective function, but the algorithm still demonstrates better overall performance since it uses two fewer vehicles and schedules one more job (sixty-seven vs. sixty-six) by solving the problem in only 8 s, which is a significant advantage over the 603 s required by the Holistic approach; see Scenario 2 in Table 8.

In the third scenario, where we do not consider mandatory jobs but breaks are scheduled, the method shows a clear advantage in terms of efficiency. It uses fewer vehicles (14 vs. 15) and achieves a better objective function value. Furthermore, the computational time difference is remarkable. The Holistic methodology takes 1651 s to attach the solution, while the D&C approach achieves results in less than one second, which is nearly instantaneous; see Scenario 3 in Table 8.

Finally, for the fourth scenario where everything is scheduled, mandatory jobs and breaks, the trend persists; that is, the algorithm uses fewer vehicles (11 vs. 15), covers a shorter distance, schedules more jobs (51 vs. 45), and finds the solution in less time. However, the Holistic approach completes more mandatory jobs, which is an important consideration for the objective function; see Scenario 4 in Table 8.

Summarizing, in view of the results in Table 8, the D&C approach consistently demonstrates superior computational efficiency. In all the tested scenarios, it drastically reduces the time to build solutions (TTB) compared to the Holistic method, with reductions ranging from approximately 79% to nearly 99% (see column 8). This efficiency is accompanied by a notable reduction in the number of vehicles required, with a decrease of around 6–26% on average across the scenarios (see column 3). On the other hand, the Holistic method excels in handling mandatory jobs. In scenarios with non-zero mandatory jobs, Holistic consistently completes more of these jobs than D&C (e.g., in Scenario 2, fifteen mandatory jobs vs. seven completed by D&C; in Scenario 4, fifteen vs. five; see column 5). However, this comes at the cost of increased computational time and, in several cases, higher resource usage, including a larger fleet of vehicles (see columns 8, 9, and 3).

To conclude this section, we refer the reader to Tables A1–A4, where detailed information regarding the number of jobs planned and the total distance traveled is included. Specifically, the first column includes the vehicle identification (Vehicle ID); the second and third columns include the number of jobs planned in each vehicle by the Holistic and D&C algorithms, respectively; and, finally, the fifth and sixth columns show the dis-

tance traveled (in thousands of kilometers) by each vehicle using the Holistic and D&C algorithms, respectively. As stated in Section 2, the travel cost is set to 1 monetary unit per kilometer, while the penalty cost is 10,000 monetary units since, in the considered instances and across all the scenarios, each vehicle travels between approximately 10,000 and 30,000 km, corresponding to a total travel cost ranging from 10,000 to 30,000 monetary units (see columns 4 and 5 of Tables A1–A4). Therefore, the penalty cost is of the same order of magnitude as the total travel cost of a vehicle, ensuring that the model sets high priority to assigning all jobs while still minimizing overall travel distance. Note that, in this way, travel and penalty costs are balanced in the objective function so that neither dominates the results unfairly.

5. Conclusions and Future Research

In this paper, we have solved a scheduling problem arising in a transportation company. Here, our objective is to schedule jobs into vehicles in order to minimize the total cost incurred even though, in most scheduling problems, the objective function is to minimize the makespan. Furthermore, our scheduling problem includes all the constraints imposed by the company and other legal regulations.

We have considered two approaches to solve the problem. On the one hand, a comprehensive or Holistic algorithm is implemented where constraints are considered. On the other hand, a Divide-and-Conquer algorithm is proposed in which a greedy heuristic and a binary linear programming are combined.

The Divide-and-Conquer algorithm demonstrates greater overall efficiency, excelling in both resource utilization and computational time. It consistently finds high-quality solutions faster than the Holistic approach, making it a strong contender in scenarios where rapid problem-solving and minimized resource usage are crucial. Considering the time to obtain the best solution across all the scenarios, Divide-and-Conquer has an average of 257.78 s (95% CI: [219.24, 296.32]) compared to 1031.65 s (95% CI: [959.09, 1104.21]) for the Holistic approach. Additionally, the average number of vehicles used is lower for the Divide-and-Conquer approach compared to the Holistic approach (12.55 vs. 14.25). When planning mandatory jobs, the Holistic approach outperforms the Divide-and-Conquer methodology in terms of objective value. In these cases, the objective value is slightly higher for the Divide-and-Conquer approach (2822.54 on average and 95% CI: [2770.99, 2874.11] thousands of EUR) compared to the Holistic approach (2000.93 on average and 95% CI: [1945.91, 2055.95] thousands of EUR). However, even though, in terms of mandatory jobs and total cost, the Holistic approach outperforms the Divide-and-Conquer algorithm, the latter always schedules a higher number of jobs and requires fewer vehicles. This results in increased customer satisfaction and enhanced resource utilization for the transport company. Since the company owns its fleet, vehicle-related costs are not considered in the objective function, and, consequently, when a vehicle is not in use, the corresponding employee can be assigned to other productive activities. It is important to emphasize that, in scenarios where mandatory jobs are not considered, the Divide-and-Conquer algorithm achieves better results than the Holistic algorithm across every considered metric, the objective value being slightly higher for Holistic (562.94 and 95% CI: [511.37, 614.50] thousands of EUR) compared to Divide-and-Conquer (490.19 95% CI: [446.92, 533.46] thousands of EUR). For the case study, this translates into an improvement not only in cost but also in customer satisfaction and the efficient use of human resources in the company.

Future research could explore incorporating intelligent assumptions to reduce the number of generated sequences, potentially eliminating the need for this arbitrary cap. In addition, to improve the performance of the Divide-and-Conquer approach when dealing

with mandatory jobs, it would be desirable to incorporate this objective intelligently into the sequence construction stage.

Author Contributions: Conceptualization, C.T.-F., A.D.L.-S., and J.S.-O.; methodology, C.T.-F. and A.D.L.-S.; software, C.T.-F. and J.S.-O.; validation, C.T.-F., A.D.L.-S., and J.S.-O.; formal analysis, C.T.-F.; investigation, A.D.L.-S.; resources, J.S.-O.; data curation, C.T.-F.; writing—original draft preparation, C.T.-F.; writing—review and editing, A.D.L.-S.; visualization, C.T.-F.; supervision, A.D.L.-S. and J.S.-O.; project administration, A.D.L.-S. and J.S.-O.; funding acquisition, J.S.-O. All authors have read and agreed to the published version of the manuscript.

Funding: C. Tobar-Fernández acknowledges funding from the Spanish Ministry of Science, 'Agencia Estatal de Investigación', through the 'Ayudas para contratos para la formación de doctores y doctoras en empresas y otras entidades (Doctorados Industriales) 2024' (ref. DIN2024-013339) through QOSIT CONSULTING SL. A.D. López Sánchez was supported and funded by the Spanish Ministry of "Ministerio de Ciencia e Innovación" through Project PID2022-139543OB-C41. J. Sánchez-Oro acknowledges the support of the Comunidad Autónoma de Madrid (grant ref. TEC-2024/COM-404), Ministerio de Economía y Competitividad (grant ref. PID2021-125709OA-C22), and Ministerio para la Transformación Digital y de la Función Pública (Cátedra ENIA AI4DDS, grant ref. TSI-100930-2023-3).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

Notations

Sets	Definitions
V	Set of vehicles
J_M	Set of mandatory jobs
J_C	Set of job requests from customers
$J = J_M \cup J_C$	Set of jobs
$O = \bigcup_{j \in J} O_j$	Set of operations where O_j is the set of operations of job j , $\forall j \in J$
S^v	Set of feasible job sequences that may be assigned to vehicle v , $\forall v \in V$
S_j^v	Set of feasible job sequences that may be assigned to vehicle v that contain job j , $\forall v \in V, \forall j \in J$
$TW_p = \{[st_p^i, et_p^i]\}_{i=1}^{n_p}$	Set of time windows of operation p , $\forall p \in O$, each represented by a closed interval, where n_p is the number of time windows of operation p
Parameters	Definitions
C_{jv}	Binary parameter that takes value 1 if job j and vehicle v are compatible, $\forall j \in J, \forall v \in V$
t_{pq}	Time from operation p to operation q , $\forall p, q \in O$
d_{pq}	Distance from operation p to operation q , $\forall p, q \in O$
c	Cost paid per unit of distance traveled
c_j	Cost paid if job j is not scheduled, $\forall j \in J$
OR_p	Operating range of p , $\forall p \in O$
D_p	Duration of operation p , $\forall p \in O$
BT	Break time to satisfy the labor regulations
DT	Maximum driving time to satisfy the labor regulations
WT	Maximum working time to satisfy the labor regulations
d_i^v	Total distance traveled for the sequence i and vehicle v , $\forall i \in S^v, \forall v \in V$

Decision variables	Definitions for the Mathematical model
x_{pv}	Binary variable that takes value 1 if operation p is assigned to vehicle v , $\forall p \in O, \forall v \in V$
y_{pqv}	Binary variable that takes value 1 if operation p precedes operation q for vehicle v , $\forall p, q \in O, \forall v \in V$
z_{pk}	Binary variable that takes value 1 if window k is selected for operation p , $\forall p \in O, \forall k \in 1 \dots n_p$
w_{pbq}	Binary variable that takes value 1 if break b is scheduled between operation p and q , $\forall p, q \in O \quad \forall b \in B^v \quad \forall v \in V$
$MCDT_b^v$	Continuous variable representing the maximum cumulative driving time
b_v^k	Start time of the k -th break of vehicle v , $\forall v \in V$
s_p	Start time of operation p , $\forall p \in O$
S_{pqv}	Continuous variable representing $s_p \cdot y_{pqv}$, $\forall p \in O, \forall q \in O, \forall v \in V, \forall b \in B^v$
S'_{pqv}	Continuous variable representing $s_q \cdot y_{pqv}$, $\forall p \in O, \forall q \in O, \forall v \in V, \forall b \in B^v$
v_{pbqv}	Binary variable representing $y_{pqv} \cdot w_{pbq}$, $\forall p \in O, \forall q \in O, \forall v \in V, \forall b \in B^v$
Decision variables	Definitions for the Holistic algorithm
R^v	List of jobs for vehicle v , $\forall v \in V$
B^v	List of start times of breaks for vehicle v , $\forall v \in V$
(s_p, e_p)	Interval start time and end time of operation p , $\forall p \in O$
Decision variables	Definitions for the Divide-and-Conquer algorithm
γ_i^v	Binary variable that takes value 1 if operation i is assigned to vehicle v , $\forall i \in S^v, \forall v \in V$

Appendix A. Scenario 1: Detailed Results by Vehicle

Table A1. Comparison between Holistic and D&C approaches in Scenario 1: detailed results by vehicle.

Vehicle ID	Holistic #Jobs	D&C #Jobs	Holistic Distance	D&C Distance
45	5	0	1.49	0.00
47	8	6	1.17	1.38
52	4	10	1.74	3.00
56	6	4	1.10	2.09
57	2	9	0.84	3.83
59	6	3	1.72	2.55
60	1	8	0.78	2.24
63	3	1	1.78	0.98
64	4	7	1.10	2.16
74	2	0	0.87	0.00
78	7	4	1.01	2.09
80	6	3	2.98	2.13
92	3	2	2.40	0.62
102	4	7	1.05	1.09
Total	61	64	20.03	24.15

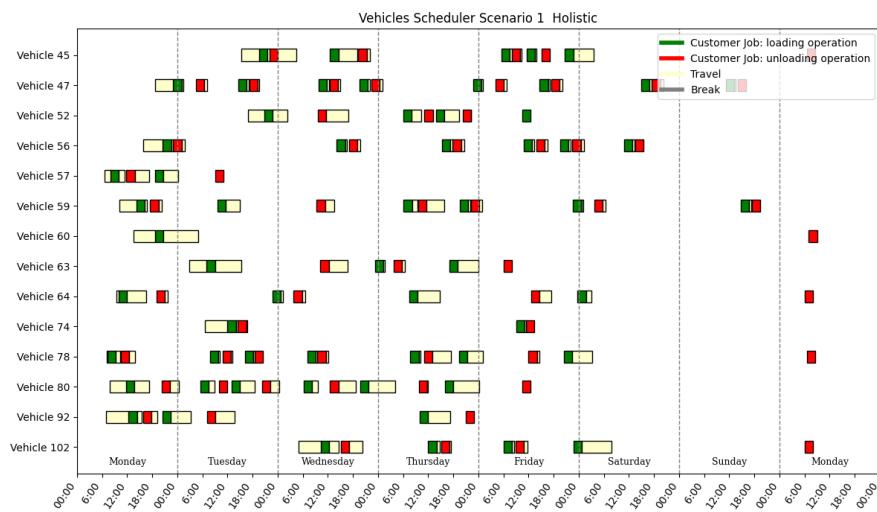


Figure A1. Gantt diagram for Holistic approach in Scenario 1.

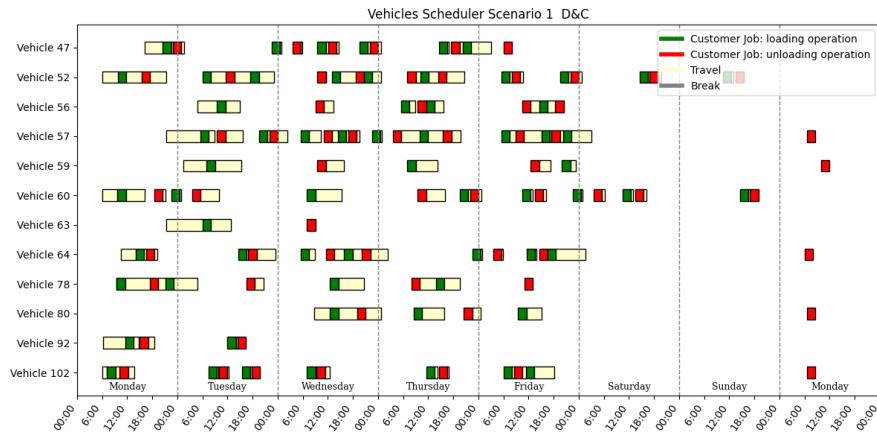


Figure A2. Gantt diagram for the Divide-and-Conquer approach in Scenario 1.

Appendix B. Scenario 2: Detailed Results by Vehicle

Table A2. Comparison between Holistic and D&C approaches in Scenario 2: detailed results by vehicle.

Vehicle ID	Holistic #Jobs	D&C #Jobs	Holistic Distance	D&C Distance
45	6	9	1.04	2.78
47	4	0	1.77	0.00
50	8	0	1.65	0.00
52	6	8	1.29	2.02
56	0	6	0.00	2.11
57	8	3	2.67	2.51
59	4	6	2.81	3.04
60	6	8	2.54	3.37
63	3	3	1.81	1.85
64	4	7	1.35	1.45
74	3	4	1.39	3.15
78	6	4	1.11	2.09
80	1	1	2.23	1.33
92	2	0	1.05	0.00
102	5	8	1.12	1.19
Total	66	67	23.83	26.91

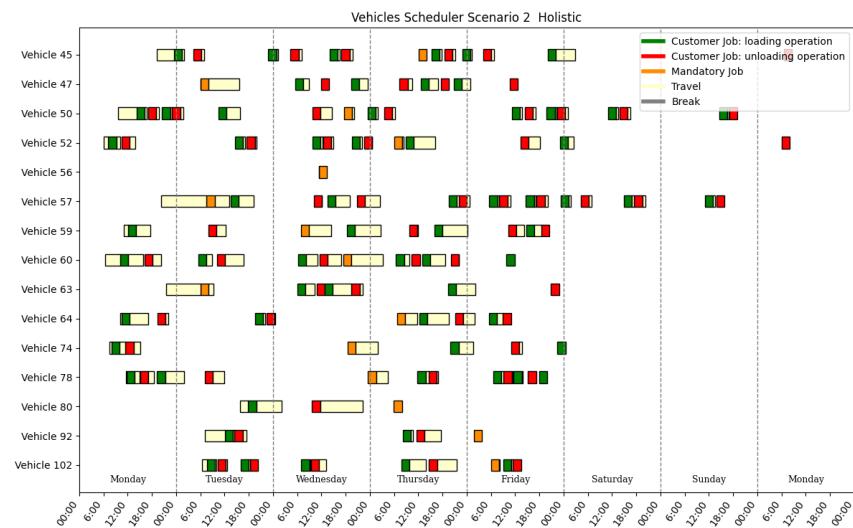


Figure A3. Gantt diagram for Holistic approach in Scenario 2.

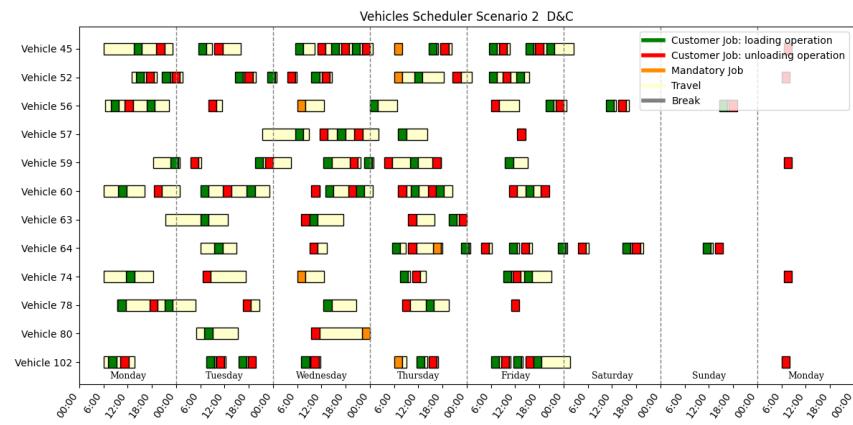


Figure A4. Gantt diagram for the Divide-and-Conquer algorithm in Scenario 2.

Appendix C. Scenario 3: Detailed Results by Vehicle

Table A3. Comparison between Holistic and D&C approaches in Scenario 3: detailed results by vehicle.

Vehicle ID	Holistic #Jobs	D&C #Jobs	Holistic Distance	D&C Distance
45	5	4	0.93	2.25
47	5	6	1.97	1.73
50	2	7	1.39	2.51
52	3	2	1.24	0.62
56	1	0	0.52	0.00
57	3	7	0.86	1.83
59	2	3	0.70	2.07
60	4	6	2.64	2.52
63	3	4	1.62	2.28
64	4	4	1.05	2.23
74	3	2	1.69	1.21
78	4	4	1.11	1.25
80	4	3	1.73	1.88
92	2	5	1.02	1.30
102	5	1	0.66	0.42
Total	50	58	19.12	14.96

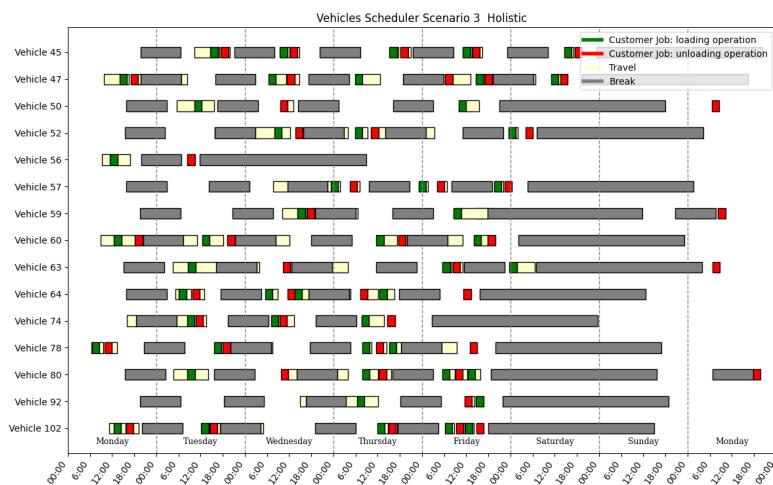


Figure A5. Gantt diagram for Holistic approach in Scenario 3.

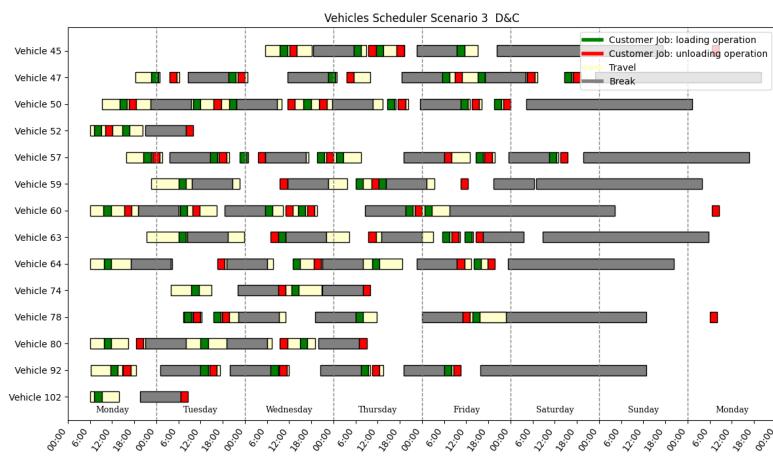


Figure A6. Gantt diagram for the Divide-and-Conquer algorithm in Scenario 3.

Appendix D. Scenario 4: Detailed Results by Vehicle

Table A4. Vehicle comparison between Holistic and D&C approaches in Scenario 4: detailed results by vehicle.

Vehicle ID	Holistic #Jobs	D&C #Jobs	Holistic Distance	D&C Distance
45	3	0	0.78	0.00
47	2	3	1.57	2.34
50	4	6	1.14	1.26
52	4	5	0.78	1.62
56	2	7	2.15	2.88
57	3	0	2.95	0.00
59	3	0	0.83	0.00
60	2	5	1.61	1.34
63	3	2	2.40	1.43
64	3	7	1.52	1.53
74	2	1	1.39	0.14
78	4	4	1.07	2.09
80	2	5	2.21	1.72
92	2	0	2.16	0.00
102	6	6	0.79	0.72
Total	45	51	23.34	17.07

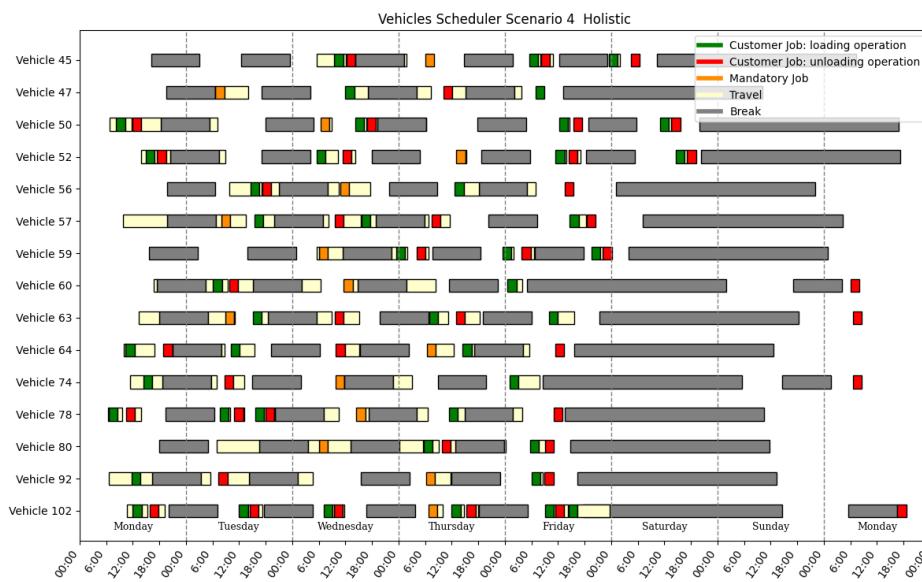


Figure A7. Gantt diagram for Holistic approach in Scenario 4.

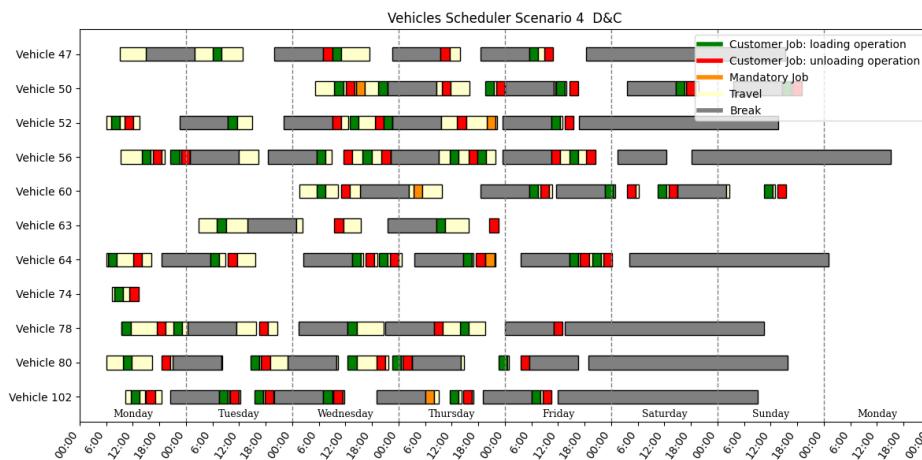


Figure A8. Gantt diagram for the Divide-and-Conquer algorithm in Scenario 4.

References

1. Morash, E.A.; Ozment, J. The strategic use of transportation time and reliability for competitive advantage. *Transp. J.* **1996**, *36*, 35–46.
2. Bharadiya, J.P. Artificial intelligence in transportation systems a critical review. *Am. J. Comput. Eng.* **2023**, *6*, 35–45. [\[CrossRef\]](#)
3. Seda, M. The assignment problem and its relation to logistics problems. *Algorithms* **2022**, *15*, 377. [\[CrossRef\]](#)
4. Kayhan, B.M.; Yıldız, G. Reinforcement learning applications to machine scheduling problems: A comprehensive literature review. *J. Intell. Manuf.* **2023**, *34*, 905–929. [\[CrossRef\]](#)
5. Mor, A.; Speranza, M.G. Vehicle routing problems over time: A survey. *Ann. Oper. Res.* **2022**, *314*, 255–275. [\[CrossRef\]](#)
6. Toth, P.; Vigo, D. *The Vehicle Routing Problem*; SIAM: Philadelphia, PA, USA, 2002.
7. Toth, P.; Vigo, D. *Vehicle Routing: Problems, Methods, and Applications*; SIAM: Philadelphia, PA, USA, 2014.
8. Walker, R.A.; Chaudhuri, S. Introduction to the scheduling problem. *IEEE Des. Test Comput.* **1995**, *12*, 60–69. [\[CrossRef\]](#)
9. Zhao, Z.; Zhou, M.; Liu, S. Iterated greedy algorithms for flow-shop scheduling problems: A tutorial. *IEEE Trans. Autom. Sci. Eng.* **2021**, *19*, 1941–1959. [\[CrossRef\]](#)
10. Durasević, M.; Jakobović, D. Heuristic and metaheuristic methods for the parallel unrelated machines scheduling problem: A survey. *Artif. Intell. Rev.* **2023**, *56*, 3181–3289. [\[CrossRef\]](#)
11. Can Atasagun, G.; Karaoglan, İ. Integrated production and outbound distribution scheduling problem with multiple facilities/vehicles and perishable items. *Appl. Soft Comput.* **2024**, *166*, 112144. [\[CrossRef\]](#)
12. Davatgari, A.; Cokyasar, T.; Verbas, O.; Mohammadian, A.K. Heuristic solutions to the single depot electric vehicle scheduling problem with next day operability constraints. *Transp. Res. Part C Emerg. Technol.* **2024**, *163*, 104656. [\[CrossRef\]](#)

13. Lv, L.; Deng, Z.; Shao, C.; Shen, W. A variable neighborhood search algorithm for airport ferry vehicle scheduling problem. *Transp. Res. Part C Emerg. Technol.* **2023**, *154*, 104262. [[CrossRef](#)]
14. Khanna, A.; Liu, F.; Gupta, S.; Pavia, S.; Mukhopadhyay, A.; Dubey, A. PDPTW-DB: MILP-Based Offline Route Planning for PDPTW with Driver Breaks. In Proceedings of the 26th International Conference on Distributed Computing and Networking, Hyderabad, India, 4–7 January 2025; pp. 73–83.
15. Mohabbati-Kalejahi, N.; Alavi, S.; Toragay, O. A Mixed-Integer Programming Framework for Drone Routing and Scheduling with Flexible Multiple Visits in Highway Traffic Monitoring. *Mathematics* **2025**, *13*, 2427. [[CrossRef](#)]
16. Mariescu-Istodor, R.; Cristian, A.; Negrea, M.; Cao, P. VRPDiv: A divide and conquer framework for large vehicle routing problems. *ACM Trans. Spat. Algorithms Syst. (TSAS)* **2021**, *7*, 23. [[CrossRef](#)]
17. Xie, K.; Tao, H.; Wang, X.; Xie, G.; Wen, J.; Cao, J.; Qin, Z. Divide and conquer for fast SRLG disjoint routing. In Proceedings of the 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Luxembourg, 25–28 June 2018; pp. 622–633.
18. Zhang, R.; Wu, C. A divide-and-conquer strategy with particle swarm optimization for the job shop scheduling problem. *Eng. Optim.* **2010**, *42*, 641–670. [[CrossRef](#)]
19. Khojasteh Toussi, G.; Naghibzadeh, M. A divide and conquer approach to deadline constrained cost-optimization workflow scheduling for the cloud. *Clust. Comput.* **2021**, *24*, 1711–1733. [[CrossRef](#)]
20. Nesterov, Y. *Lectures on Convex Optimization*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 137.
21. Michalewicz, Z. *How to Solve It: Modern Heuristics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
22. Hexaly Inc. HxOperator—Hexaly 13.0 Documentation. 2024. Available online: <https://www.hexaly.com/docs/last/changelog/hexaly130.html> (accessed on 10 December 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.