# A Constraint Satisfaction Problems Based Scalable Framework to Address Large-Scale Realistic Scheduling and Routing Problems

Liwen Zhang[a], Sara Maqrot[b], Florent Mouysset[c] and Christophe Bortolaso[d]
*Research and Technological Innovation Department, Berger-Levrault, 64 Rue Jean Rostand, Labège, France*
*{firstname.lastname}@berger-levrault.com*

Keywords: Constraint Satisfaction Problem, Scheduling and Routing Problem, Scalable Framework, OptaPlanner, Meta-Heuristic.

Abstract: Scheduling and routing solutions for organizational staff , along with decision-making support for timetabling, have become increasingly complex. This paper addresses the challenges associated with realistic large-scale generic routing and scheduling problems with a multi-day horizon. We introduce a 2-level scalable framework featuring a scalable use-case adapter and a scalable optimizer. In optimizer, a Scheduling and Routing Problem (SRP) model and a configurable constraint system are implemented using OptaPlanner. In the experimentation section, we present two real-life use cases in Spain and France, involving up to 481 activities to be performed by 18 staff members over 4 weeks. These scenarios are submitted to our model under different meta-heuristic configurations. The results demonstrate the achievement of high-quality optimized solutions within a short computing time of just 8 minutes. Additionally, a detailed investigation is conducted to interpret the scores of optimized solutions in an understandable manner.

## 1 INTRODUCTION

In the industrial field, there is a wide diversity of business areas that require the optimization of scheduling and routing solutions, considering numerous business-oriented objectives and strategic rules. Consequently, the generation of scheduling and routing solutions for staff in a given organization, even providing the decision-making support for timetabling decision makers in these organizations, has become extremely complex. This complexity is evident when it involves a large number of constraints from various use cases, such as the well-known Vehicle Routing Problem and its variants (Caceres-Cruz et al., 2014).

When the large-scale problem has to be figured out (e.g. 500 activities to schedule within 5 weeks), we observe that there is a real need for better modeling and solving the scheduling and routing problems. Since demands vary and the response capacities provided by each service-delivery organization are no less varied, there is a strong pressure on these organizations to regularly manage their service flows.

[a] https://orcid.org/0009-0003-4692-1956
[b] https://orcid.org/0000-0003-2261-4081
[c] https://orcid.org/0009-0009-1220-3933
[d] https://orcid.org/0000-0002-6635-9345

A scalable scheduling and routing framework along with the Human Machine Interface (HMI) of an operational product, supports decision-makers in capturing all the necessary knowledge to address operational scheduling and routing challenges. This framework helps describe the problem by identifying key requirements and gathering necessary information from all stakeholders. As the volume of information can become significant, integrating this framework into the organization's information system may make sense.

In this paper, we address the complexity of Scheduling and Routing Problems (SRP) in operational organizations by presenting a scalable two-level framework designed for real-world SRP. Our contribution is threefold: 1) we investigate the key concepts of SRP over various time horizons (from one day to multi-month schedules). We start by drawing from relevant content in the literature to facilitate the modeling phase, the second level of our framework. 2) Using data processed by *Adapter* (the first level of our framework), we implement an OptaPlanner-based SRP model, featuring a configurable constraint system based on the Constraint Satisfaction Problem (CSP) paradigm. This allows us to efficiently solve SRP through various meta-heuristic algorithms embedded in OptaPlanner. As far as we know, no stud-



Zhang, L., Maqrot, S., Mouysset, F. and Bortolaso, C.
A Constraint Satisfaction Problems Based Scalable Framework to Address Large-Scale Realistic Scheduling and Routing Problems.
DOI: 10.5220/0013109600003893

In *Proceedings of the 14th International Conference on Operations Research and Enterprise Systems (ICORES 2025)*, pages 45-56
ISBN: 978-989-758-732-0; ISSN: 2184-4372

ies have used an OptaPlanner-oriented optimizer to address real-life SRP that simultaneously optimize the density of activities assigned at the beginning of the scheduling horizon and balance the weekly workloads of staff members. This is further combined with traditional constraints such as respecting time windows and staff unavailability, all within a multi-day scheduling horizon. 3) For experimental validation, we apply the framework to two real-word use cases of the Preventive Maintenance Scheduling and Routing Problem (PMSRP), one in Spain and the other in France. To the best of our knowledge, an OptaPlanner-based CSP model for tackling PMSRP has not been addressed in the literature. We evaluate and compare the solution quality generated by different meta-heuristic algorithms embedded in OptaPlanner.

The remainder of this paper is structured as follows. Section 2 presents a review of the literature related to our study. In Section 3, we describe our proposed scalable framework designed to address the various SRP. Section 4 provides illustrative computational experiments and discussion. Finally, some future researches are suggested in Section 5.

## 2 LITERATURE REVIEW

Originally, the Traveling Salesman Problem (TSP) is a commonly known class in all routing and scheduling problems. According to (Gutin and Punnen, 2006), there are many variants of TSP. These authors describe a set of at least ten variants of this basic formulation, and the main differences arise from the way objective functions and constraints are approached within different contexts. For example, minimizing the total visit cost following a Hamiltonian path in the Messenger problem (Jünger et al., 1995). (Dantzig and Ramser, 1959) argue that the family of TSP problems can be considered as a subclass of the VRP, and then VRP is identified by (Bektas, 2006) as a "multi-TSP". Some variants of the VRP have been extensively studied in recent years. With aging populations and an increasing life expectancy, Home Health Care (HHC) has become a common healthcare delivery manner in Europe. Therefore, HHCRSP is a new VRP application that considers the variability of business constraints, along with the diversity of the HHC organization (Di Mascolo et al., 2021). Furthermore, with the fourth industrial revolution underway, driven by Internet of Things (IoT) technology, there is a growing interest in another widely recognized VRP application, the PMSRP, particularly in complex scenarios involving distributed systems where maintenance assets are geographically dispersed (Rashidnejad et al., 2018).

Vehicle Routing Problem (VRP) is a well-known class within Scheduling and Routing Problems (SRP). By definition, the primary objective of a VRP is to optimize the delivery routes of a fleet of vehicles from a depot to various customer sites, all while accommodating a variety of constraints that add complexity to the planning process. These constraints, often specific to different business fields, include temporal restrictions, which are especially critical in scheduling and routing. The offer-demand paradigm highlights the importance of aligning both customer and staff satisfaction with respect to time management. This involves synchronizing individual schedules to balance the needs and preferences of both parties. The decision-maker must ensure that the services are carried out without compromising quality and within an acceptable operating duration. Frequently, a time window (referring to a temporal interval specifying possible starting times for a service) characterizes either the customer's availability or their preferred service time (Ibaraki et al., 2005). For normal services, a temporal tolerance is possible. Hence, two types of time windows are retained in the literature (Dekhici et al., 2019): the hard (or fixed) time window or the soft (or flexible) time window. In (Rest and Hirsch, 2016), the authors focused on the working hours of the staff. Over a weekly horizon, for instance, if a staff signs a 25-hour-per-week contract, a maximum daily working time may also influence the definition of their schedule. A time interval can be applied to indicate the earliest start time and the latest finish time of the staff's daily working rounds (Di Gaspero and Urli, 2014). To reduce fatigue and increase work efficiency, a daily lunch break is considered in (Coelho et al., 2016).

Decision-makers must define directions within their decisions, specifying the objectives to be achieved or the risk to be avoided. A frequently used minimization criterion in this context is travel time (Dekhici et al., 2019), where total transportation costs are minimised in (Nguyen et al., 2022). When staff work capacities are exceeded, overtime costs become a significant concern. (Carello and Lanzarone, 2014) aim to reduce the overtime costs of daily workers. To ensure staff satisfaction, workloads must be balanced. (Yalçındağ et al., 2016) apply utilization rates to achieve personal workload balance. These rates compare actual workload to theoretical work capacity. In (Quintanilla et al., 2020), the authors focus on balancing the number of visits performed by each staff.

The choice of resolution methods varies depend-

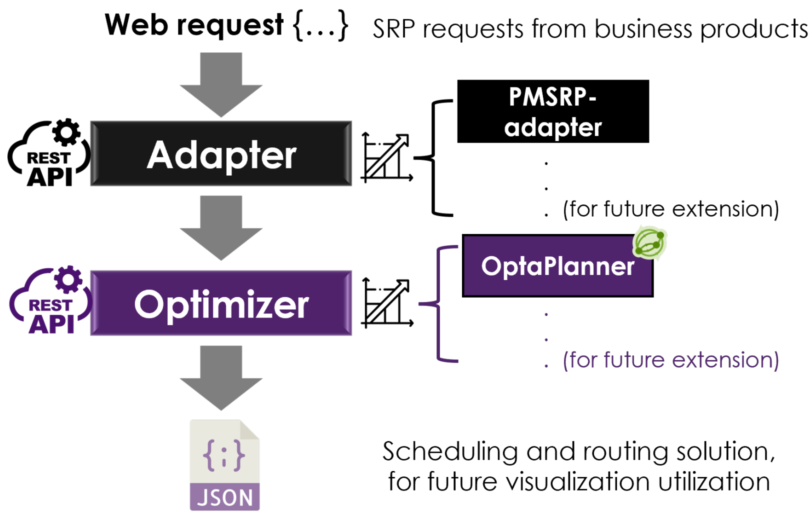



Figure 1: Overview of SRP framework architecture.

ing on the mathematical nature of the target problem. Selected resolution methods are highly tailored to specific types of routing and scheduling problems. In Operations Research (OR) related research, two main approaches are explored: exact methods and approximate methods. Exact methods face challenges when addressing real-life use cases due to the sensitivity of their Integer Linear Programming (ILP) based mathematical models to problem size. Several mathematical programming solvers, such as CPLEX and Gurobi, employ exact methods for solution search. To illustrate the size of the problem that can be handled with exact methods, in (Zhang et al., 2021), CPLEX is used to solve the daily Home Health Care Scheduling and Routing Problem (HHCSRP) based on ILP, with a size of up to 61 activities per day. However, when the size of the problem exceeds 61 activities, with the constraints considered in the model, CPLEX cannot find the optimal solution within the predefined maximum computation time of 10 hours. In such cases, approximate methods are valuable for solving large-size problems with numerous constraints in various business contexts. These methods, based on meta-heuristic algorithms, provide reliable alternatives to exact methods thanks to their capacity to efficiently explore vast search spaces within reasonable computation times. These approximate algorithms incorporate mathematical models for problem-solving, either directly or through Constraint Programming modeling. They are then coupled with meta-heuristic algorithms such as Simulated Annealing (Kirkpatrick et al., 1983) or Tabu Search (Glover and Laguna, 1998) to generate near-optimal solutions. Although few articles mention the use of approximate solvers for large-scale scheduling and routing problems, examples such as (Smirnov and Shilov, 2010) highlight the application of approximated methods embedded in the Choco solver to address real-life dynamic logistics problems. Furthermore, in (Zhang et al., 2023), OptaPlanner is used to tackle the daily HHCSRP with a size of up to 130 activities per day.

In conclusion, addressing scheduling and routing challenges involves managing the trade-off between problem complexity (diverse constraints and decision variables) and computational time for solution generation. Despite the approximation methods existing in previous research, the resolution of real-world industrial problems remains a persistent challenge due to their significant scale, diversity, and dynamic constraints in various application scenarios.

## 3 SCALABLE FRAMEWORK TO ADDRESS REAL-WORLD SCHEDULING AND ROUTING PROBLEMS

Drawing on insights from the literature review, we present a scalable framework to address the challenges related to computational time and scheduling requirements in real-world SRP.

### 3.1 Architecture Overview

Our optimization framework consists of two modules, the (*Adapter*) and the (*Optimizer*), which together form a consistent architecture. This framework provides a generative solution to the SRP. Figure 1 presents an overview of the linear process that tra-

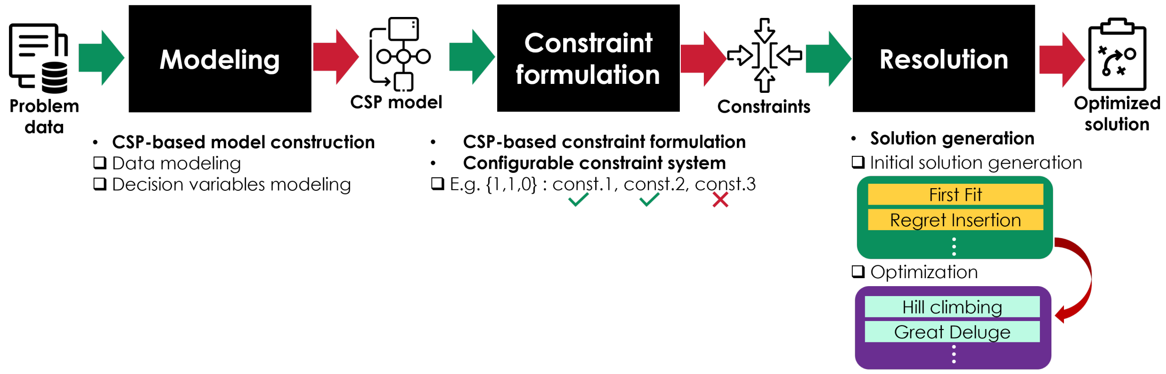



Figure 2: Illustration of three-level process in Optimizer of the scalable framework.

verses the two modules. At the end of this process, the scheduling and routing solution is obtained in a well-formatted JSON file, which can be used to display the optimized solution via a calendar or map view of a web application.

The first module (*Adapter*) is responsible for analyzing web requests from various business products related to different scheduling problems. It performs data structure adaptation and conversion to the generic API of the second module (*Optimizer*). The API defined in this module allows users to activate or deactivate embedded constraints as needed. This module can include different adapters for various use cases from related products, such as the PM-SRP adapter for the Preventive Maintenance Scheduling and Routing Problems. Additionally, this module is scalable to meet the scheduling and routing requirements of new products.

In the second module (*Optimizer*), the previously translated data from the corresponding adapter is used to optimize solutions with either an open-source or a commercial solver. Currently, only one open-source AI solver, OptaPlanner, is developed to perform SRP optimization, following a three-level process, as illustrated in Figure 2:

1. **Modeling.** Based on the CSP paradigm, as shown in Figure 3, an SRP model is constructed. Each concept is defined with its necessary parameters, and the relationships between these concepts are specified. Importantly, the decision variables (annotated by @ShadowVariable or @PlanningListVariable) are indicated in the concepts annotated by @PlanningEntity. This part is detailed in Section 3.2.1. Note that the attribute names shown in the figure after "_" correspond to the annotation used in Section 3.2.1.
2. **Constraint Formulation.** Based on the SRP model, a series of constraints is pre-formulated. These constraints can be activated or deactivated based on feedback from the adapters. The formulation of these constraints is detailed in Section 3.2.2.
3. **Resolution.** The SRP is resolved using the approximate methods embedded in OptaPlanner, as detailed in Section 3.2.3.

It is important to note that this module is also scalable, allowing for the integration of new solvers to benchmark against OptaPlanner's performance. This three-level process is illustrated in Figure 2.

OptaPlanner is selected as the first solver developed in this module due to its open-source nature and its intuitive annotation system for constructing CSP-based models. The OptaPlanner annotation system is detailed in our previous research work (Zhang et al., 2023). Additionally, the numerous embedded construction heuristics and local search-based metaheuristics enable us to select the appropriate combination of algorithms to tackle various problems. Finally, OptaPlanner is fully written in Java, which facilitates its integration into our architecture, driven by Spring Boot on the backend.

## 3.2 Optimizer Description

### 3.2.1 Modeling

#### (A) Data Model

*Definition 1:* Office is the central entity in our system, denoted by $\mathbb{O}$ and defined by the following parameters:

- $g$: Planning granularity (in seconds), serving to convert all time-related parameters into integer slot format. E.g. if $dur_a = 1$ and $\omega = 60$, then the operation time of activity $a$ is 60 seconds.
- $h$: Number of days in scheduling horizon.
- $p$: Number of activities to be scheduled within $h$.
- $q$: Number of available staff members within $h$.

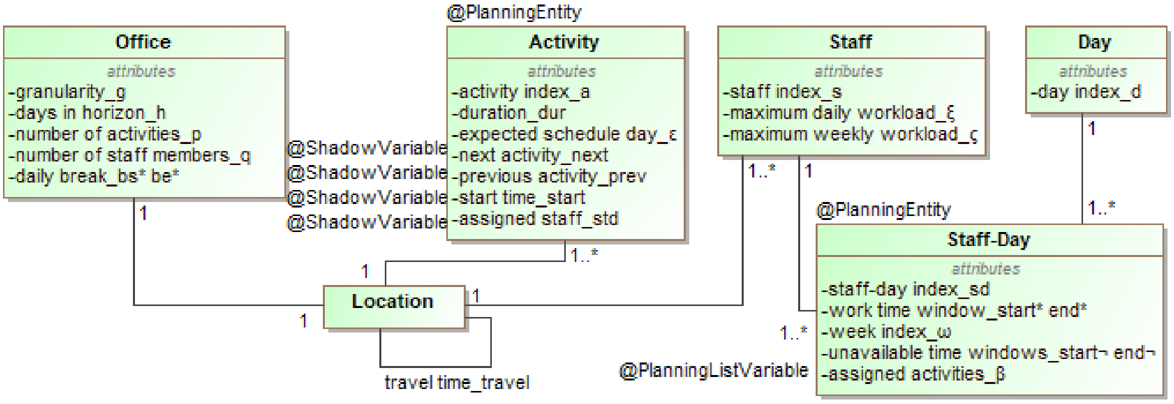



Figure 3: Overview of OptaPlanner-based SRP model.

- $[bs^*, be^*]$: Time window for a daily break.

*Definition 2:* Activity is the primary planning entity in our model, representing client-requested activities, denoted by $\mathbb{A} = \{a_1, .., a_p]$ and defined by :

- $dur_a$: Operation time required for activity $a$, including the time to park, where. $dur_a \in [1, \infty]$
- $\varepsilon_a$: Expected schedule day requested for activity $a$ (for horizon $h > 1$ ).

*Definition 3:* Staff-Day is a planning entity associated with Staff and Day, representing the set of available staff members for deployment on a given day within the scheduling horizon. Each Staff-Day element corresponds to a single workday of a staff member.

- Staff is denoted as $\mathbb{S} = \{s \mid s \in \mathbb{S}, |\mathbb{S}| = q\}$, each element $s \in \mathbb{S}$ is characterized by :
  - $\xi_s$: Maximum daily workload.
  - $\varsigma_s$: Maximum weekly workload.
- Day is denoted as $\mathbb{D} = \{d \mid d \in \mathbb{D}, |\mathbb{D}| = h\}$.
- Staff-Day is denoted as $\mathbb{SD} = \{sd \mid sd \in \mathbb{SD}, |\mathbb{SD}| = \alpha^*\}$ , where $\alpha^* = q \cdot h$ represents the total number of available staff members within the daily horizon. Each $sd \in \mathbb{SD}$ is characterized by the following parameters:
  - $[start^*_{sd}, end^*_{sd}]$: Work time window for staff member $s$ on day $d$.
  - $\omega_{sd}$: Week index within the horizon.
  - $[start^{\neg}_{sd}, end^{\neg}_{sd}]$: Unavailable time windows for staff member $s$ on day $d$.

*Definition 4:* Location is an entity associated with Office, Activity and Staff-Day, as depicted in Figure 3,. It determines their respective locations, which are essential for calculating travel time. Location is denoted by $\mathbb{L}$ and characterized by:

- $travel_{ij}$ : Travel time from $i$ to $j$, with $i, j \in [1, p] \cup [1, q] \cup \{0\}$, where $\{0\}$ represents the office. $travel_{ij}$ is computed using the Origin-Destination cost matrix solver from ESRI[1] considering real-life traffic conditions.

**(B) Decision Variables in Planning Entity**

As illustrated in Figure 3, Activity $\mathbb{A}$ and Staff-Day $\mathbb{SD}$ are annotated with @PlanningEntity, allowing for changes during the solution search process. Decision variables are the attributes within these entities that can change values. For a comprehensive exploration of the optimization model-oriented annotation system in OptaPlanner, refer to (Zhang et al., 2023).

For each $sd \in \mathbb{SD}$, there exists a planning list variable denoted by $\beta_{sd}$, which serves as the primary variable containing multiple chained activities $a \in \mathbb{A} = \{a_1, .., a_p\}$. This variable allows each activity $a \in \beta_{sd}$ to link directly to another, establishing an order among the planned entities. The chain begins at the start location of $sd \in \mathbb{SD}$ and ends with the last element of the list $\beta_{sd}$.

The travel time is calculated between each pair of elements in $\beta_{sd}$. However, the last travel time in the list $\beta_{sd}$ is calculated from the final element in $\beta_{sd}$ to the end point of the daily round of $s$, which can be the office location or the domicile location of $s$.

The following requirements guide the construction of a chain in this model :

- Each chain is open-ended and does not form a loop. Figure 4 illustrates an example of a correct chain (Chain 1) and an incorrect looping chain (Chain 3).
- A chain is linear, not a tree. Therefore, each $sd \in \mathbb{SD}$ has only one $a \in \beta_{sd}$ at the end of the chain. In Figure 4, we illustrate an incorrect tree structure-based chain (Chain 4).
- $sd \in \mathbb{SD}$ with $\beta_{sd} = \varnothing$ is also considered a 0-chain.

[1]ESRI: https://www.esri.com/en-us/home

- Only one round is allowed for $sd \in \mathbb{SD}$, corresponding to $\beta_{sd} = 1$.
- When $[bs^*, be^*]$ is defined within an Office setting, a chain can be viewed as a two-part sequence. The first part begins at $sd \in \mathbb{SD}$ and ends at $a_1 \in \mathbb{A}$ with a termination time $\geq bs^*$. The second part starts at $a_2 \in \mathbb{A}$, where a start time equal to $be^* + travel_{a_1,a_2}$, and ends at another point, $a_3 \in \mathbb{A}$. This two-part chain is illustrated in Figure 4 as Chain 2.

For each $a \in \mathbb{A}$, four decision variables are defined. These secondary variables, referred to shadow variables in the OptaPlanner ecosystem, depend on the primary variable $\beta_{sd}$, $sd \in \mathbb{SD}$. A shadow variable is a planning variable whose value can be deduced from the state of $\beta_{sd}$. The definitions of the four decision variables are as follow :

- $next_a$: The next activity following activity $a$ in a staff's daily round, with the domain $next_a = \{a_1, a_2, ..., a_p\}$
- $prev_a$: The previous activity preceding activity $a$ in a staff's daily round, with the domain $prev_a = \{a_1, a_2, ..., a_p\}$
- $start_a$: The start time of activity $a$ within a day, with the domain $start_a = [0, 86400/g]$
- $std_{asd}$: The assigned staff $s$ for staff-Day $sd$ for activity $a$, with the domain of $std_{asd} = \{s_1, s_2, ..., s_q\}$

#### 3.2.2 Constraint Formulation

All constraints stated in this section are enabled or disabled based on the different context of the target SRP. In our model, there is no objective function to characterize solution quality. Instead, each constraint is linked to a hard or soft score based on its violation. Consequently, a solution is defined by a global score, where the hard part is denoted by $\Lambda = \sum\{hard\ scores\ for\ each\ constraint\}$, and the soft part by $\Omega = \sum\{soft\ scores\ for\ each\ constraint\}$. The presence of a hard score indicates an infeasible solution, while the goal is to maximize the soft score. The higher the score, the higher the overall score, indicating a better solution.

Furthermore, a list of binary values is required in the format $\{1,1,1,1,0,0,0,0,0\}$. This list indicates that constraints (1)-(4) are activated, while constraints (5)-(9) are deactivated. It represents the necessary constraints for a target SRP, generated by the corresponding adapter based on the request from the related product.

Constraint (1) ensures that all care services are assigned to a staff member during the day, thereby prohibiting any missing activities in the target use case.

$$\begin{aligned} &\textbf{If } std_{asd} = \varnothing \vee start_a = \varnothing \textbf{ Then } \Lambda - 1 \\ &\forall a \in \mathbb{A}, \forall sd \in \mathbb{SD} \end{aligned} \tag{1}$$

Constraint (2) ensures that each activity is scheduled within the working time window of staff member $s$ on day $d$.

$$\begin{aligned} &\textbf{If } (start_a \geq end^*_{sd}) \vee (start_a + dur_a > end^*_{sd}) \\ &\vee (start_a < start^*_{sd}) \quad \textbf{Then } \Lambda - 1 \\ &\forall a \in \mathbb{A}, \forall sd \in \mathbb{SD} \end{aligned} \tag{2}$$

Constraint (3) prohibits scheduling activities during the designated unavailability of the staff member $s$ on day $d$.

$$\begin{aligned} &\textbf{If } ((start_a \leq start^{\neg}_{sd}) \wedge (start_a + dur_a \geq end^{\neg}_{sd})) \\ &\vee ((start_a > start^{\neg}_{sd}) \wedge (start_a + dur_a < end^{\neg}_{sd})) \\ &\vee ((start_a + dur_a \geq start^{\neg}_{sd}) \\ &\wedge (start_a + dur_a \leq end^{\neg}_{sd})) \\ &\vee ((start_a \geq start^{\neg}_{sd}) \wedge (start_a \leq end^{\neg}_{sd}) \\ &\textbf{Then } \Lambda - 1 \quad \forall a \in \mathbb{A}, \forall sd \in \mathbb{SD} \end{aligned} \tag{3}$$

Constraint (4) ensures that the total workload of the staff member $s$ does not exceed the maximum daily limit $\xi_s$.

$$\begin{aligned} &\textbf{If } \sum_{i \in \beta_{sd}} \sum_{\substack{j \in \beta_{sd} \cup std_{asd}: \\ i \neq j, j = i+1}} (travel_{ij} + dur_i) > \xi_s \\ &\textbf{Then } \Lambda - \left| \xi_s - \sum_{i \in \beta_{sd}} \sum_{\substack{j \in \beta_{sd} \cup std_{asd}: \\ i \neq j, j = i+1}} (travel_{ij} + dur_i) \right| \\ &\forall a \in \mathbb{A}, \forall s \in \mathbb{S}, \forall sd \in \mathbb{SD} \end{aligned} \tag{4}$$

Constraint (5) aims to align the overall workload of the staff member $s$ as closely as possible with the weekly maximum limit $\varsigma_s$.

$$\begin{aligned} &\textbf{If } \sum_{i \in \beta_{sd}} \sum_{\substack{j \in \beta_{sd^*} \cup std_{asd} \\ \cup \omega_{sd} = \omega_{sd^*}: \\ i \neq j, j = i+1}} (travel_{ij} + dur_i) > \varsigma_s \\ &\textbf{Then } \Omega - \left| \varsigma_s - \sum_{i \in \beta_{sd}} \sum_{\substack{j \in \beta_{sd^*} \cup std_{asd} \\ \cup \omega_{sd} = \omega_{sd^*}: \\ i \neq j, j = i+1}} (travel_{ij} + dur_i) \right| \\ &\forall a \in \mathbb{A}, \forall s \in \mathbb{S}, \forall sd, sd^* \in \mathbb{SD} \end{aligned} \tag{5}$$

Constraint (6) ensures that activity $a$ with $\varepsilon_a \neq \varnothing$, is scheduled as closely as possible to the expected schedule day.

$$\begin{aligned} &\textbf{If } \varepsilon_a \neq \varnothing \textbf{ Then } \Omega - | \varepsilon_a - \{d \mid d \in std_{asd}\} | \\ &\forall a \in \mathbb{A}, \forall sd \in \mathbb{SD} \end{aligned} \tag{6}$$

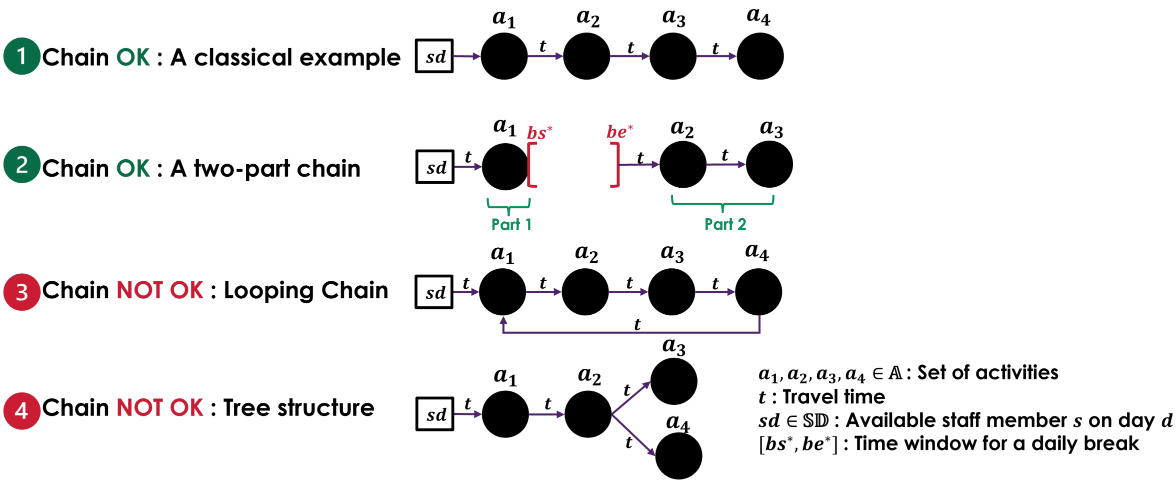



Figure 4: Illustration of the chain-based planning list variable $\beta_{sd}$, for $sd \in \mathbb{SD}$.

Constraint (7) minimizes the total travel times of all staff members, operating without any conditional requirements.

$$\Omega - \sum_{i \in \beta_{sd}} \sum_{\substack{j \in \beta_{sd} \cup std_{asd}: \\ i \neq j, j=i+1}} travel_{ij} \quad \forall a \in \mathbb{A}, \forall sd \in \mathbb{SD} \quad (7)$$

Constraint (8) optimizes the density of activities assigned at the beginning of the scheduling horizon. The main goal is to improve resource efficiency by reducing inactivity periods among staff, thus creating additional time slots for the insertion of urgently added activities. There is no conditional part for this constraint.

$$\Omega - \{d \mid d \in std_{asd}\} \quad a \in \mathbb{A}, \forall sd \in \mathbb{SD} \quad (8)$$

Constraint (9) promotes balance in the weekly workloads of the staff members, considering the travel times between assigned activities and staff unavailability as needed.

$$\textbf{If } \omega_{sd} = \omega_{s^*d^*}$$

$$\textbf{Then } \Omega - \Bigg( \sum_{i \in \beta_{sd}} \sum_{\substack{j \in \beta_{sd} \cup std_{asd}: \\ i \neq j, j=i+1}} (travel_{ij} + dur_i + \mid start^{\neg}_{sd} - end^{\neg}_{sd} \mid) - \sum_{i^* \in \beta_{s^*d^*}} \sum_{\substack{j^* \in \beta_{s^*d^*} \cup \\ std_{a^*s^*d^*}: \\ i^* \neq j^*, j^*=i^*+1}} (travel_{i^*j^*} + dur_{i^*} + \mid start^{\neg}_{s^*d^*} - end^{\neg}_{s^*d^*} \mid) \Bigg) \quad (9)$$

$$a, a^* \in \mathbb{A}, \forall sd, s^*d^* \in \mathbb{SD}$$

#### 3.2.3 Resolution

The resolution of the problem in OptaPlanner begins with the generation of an initial solution using construction heuristics within a limited time frame while aiming for the best possible quality. OptaPlanner supports five construction heuristics, including First Fit and its variants, Weakest Fit and Strongest Fit (Bays, 1977), Allocate Entity from Queue (Semeria, 2001), Regret Insertion (Diana and Dessouky, 2004), Allocation from pool (Lattner and Adve, 2002) and Scaling construction heuristics (Katayama et al., 2009).

The second phase involves iteratively searching for a solution that outperforms the current one using meta-heuristic algorithms. OptaPlanner's configuration allows for the selection of optimization algorithms and the definition of their intrinsic parameters. The optimization process ends when predefined stopping criteria are met, typically defined as reaching either a maximum computational time limit or a specific number of iterations. In addition to construction heuristics, OptaPlanner offers various types of local search meta-heuristics:

1. Simple local search: Hill Climbing (Goldfeld et al., 1966) and Late acceptance (Burke and Bykov, 2017).
2. Meta-heuristics: Tabu Search (Glover and Laguna, 1998), Simulated Annealing (Van Laarhoven and Aarts, 1987), Great Deluge (Dueck, 1993), and Variable Neighborhood Descent (Gao et al., 2008).

## 4 TESTS

In experimental section, we aim to evaluate our target problems using our framework in various configurations. The configurations of our OptaPlanner-based optimizer draw inspiration from our previous research work (Zhang et al., 2023), using the First Fit construction algorithm in conjunction with all the meta-

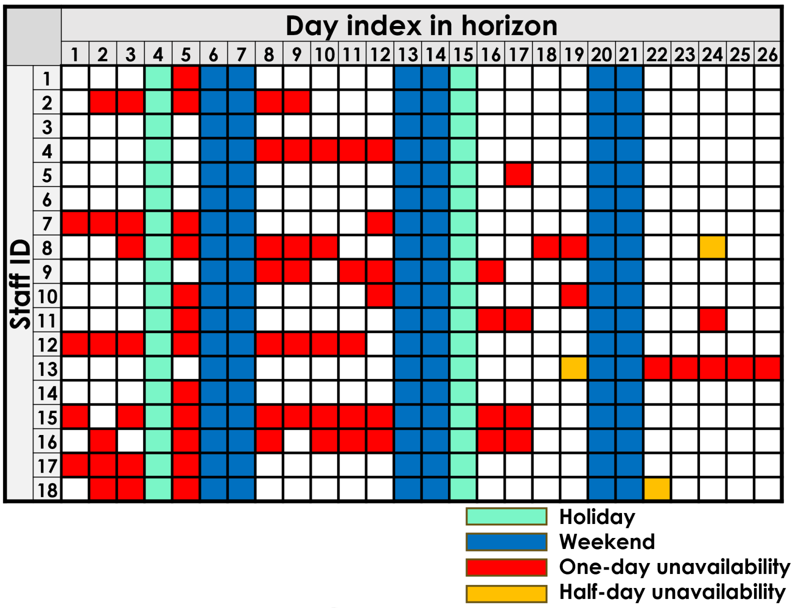



Figure 5: Staff unavailabilities aligned with the day index within the planning horizon for the French use case.

heuristics embedded in OptaPlanner except for Tabu Search. The choice of First Fit as the construction heuristic for generating the initial solution is based on the study conducted by (Macik, 2016), which provides an in-depth discussion of the configuration of the solver. This author, who examined the task assignment problem, demonstrated that First Fit is the most effective construction heuristic. Similarly, in the work of (Rios de Souza and Martins, 2020), First Fit is also used to construct the initial solution, further validating its effectiveness. Tabu Search is omitted from the testing due to the complexity of our identification rule for the ID of each planning entity, which limits the solver's ability to execute the optimization process effectively. Additionally, in line with (Zhang et al., 2023), we utilize two termination criteria: either when the global score in the form of HardSoftScore reaches zero in both its hard and soft components, or when the computation time reaches the predefined limit of 8 minutes. The choice of an 8-minute time limit is based on client feedback.

## 4.1 Realistic Dataset Statement

The dataset was extracted from the database of an operational product used by two Berger-Levrault clients: one in Spain and the other in France. The Spanish client (case ES) assigns staff to maintenance tasks on water network infrastructures. Each staff member begins and ends their daily work at the office, with repair points located within a city and a relatively short planning horizon. In contrast, the French client (case FR)

Table 1: Main data for Spanish Case and French Case.

| Parameter | Values of Case ES | Values of Case FR |
|---|---|---|
| ($\mathbb{O}$) $g$ | 60 | 60 |
| ($\mathbb{O}$) $h$ | 5 | 26 |
| ($\mathbb{O}$) $p$ | 90 | 481 |
| ($\mathbb{O}$) $q$ | 4 | 18 |
| ($\mathbb{O}$) $[bs^*, be^*]$ | $[50400, 55800]$ | - |
| ($\mathbb{A}$) $dur_a$ | see Figure 6 | see Figure 6 |
| ($\mathbb{A}$) $\varepsilon_a$ | $\varepsilon_{a_1} = 1, \varepsilon_{a_2} = 2, \varepsilon_{a_3} = 2, \varepsilon_{a_4} = 3, \varepsilon_{a_5} = 5, \varepsilon_{a_6} = 4$ | - |
| ($\mathbb{S}$) $\xi_s$ | - | $\xi_s = 32400$ |
| ($\mathbb{S}$) $\varsigma_s$ | - | $\varsigma_s = 144000$ |
| ($\mathbb{SD}$) $[start^*_{sd}, end^*_{sd}]$ | [32400,63000] | [28800,70200] |
| ($\mathbb{SD}$) $\omega_{sd}$ | $\{0\}$ | $\{1,2,3,4\}$ |
| ($\mathbb{SD}$) $[start^{\neg}_{sd}, end^{\neg}_{sd}]$ | [32400,63000], $s = 1, d = 2$ | see Figure 5 |

assigns staff to perform maintenance on machines at various client locations throughout France. Here, each staff member begins and ends their daily work at home, with a relatively long planning horizon. The parameter settings for our model for the case of ES are listed in column 2 of Table 1, while those for the case of FR are described in column 3.

Due to the extensive planning horizon and the frequent unavailability of active staff members, Figure 5 presents detailed information regarding the parameter $[start^{\neg}_{sd}, end^{\neg}_{sd}]$ for the case FR. It is important to note that the red and blue zones (representing weekends or holidays) indicate the unavailability of specific staff members throughout the day ($[start^{\neg}_{sd}, end^{\neg}_{sd}] = [28800, 70200]$). The orange zone denotes the half-day inaccessibility for staff members

Table 2: Constraints activated for FR and ES cases.

| Constraint# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Case ES | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| Case FR | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

Table 3: Comparison of solution scores: case ES.

| | SA | HC | LA | GD | VND |
|---|---|---|---|---|---|
| C. 1 | 0 | 0 | 0 | 0 | 0 |
| C. 2 | 0 | 0 | 0 | -6 | 0 |
| C. 3 | 0 | 0 | 0 | 0 | 0 |
| Λ | 0 | 0 | 0 | -6 | 0 |
| C. 6 | 0 | 0 | 0 | -1 | 0 |
| C. 7 | -341 | -562 | -1239 | -1264 | -532 |
| C. 8 | -179 | -177 | -174 | -181 | -164 |
| C. 9 | -244800 | -257600 | -195200 | -3343600 | -270000 |
| Ω | -245320 | -258339 | **-196613** | -3345045 | -270696 |

$([start_{sd}^{\neg}, end_{sd}^{\neg}] = [28800, 43200] \vee [43200, 70200])$.

Finally, Figure 6 illustrates the duration distributions for all activities in two cases.

## 4.2 Numerical Result

After conducting several interviews to understand the distinct scheduling requirements of the two clients, Table 2 illustrates the activated constraints for each case, with a value of 1 indicating that the corresponding constraint is activated and 0 otherwise.

The test is carried out on a computer with an Intel(R) Core(TM) i7-7500U processor at 2.90 GHz and 16.0 GB of RAM under the Windows 10 operating system. In both use cases, our initial evaluation involves evaluating the hard component (Λ) and the soft component (Ω) of the overall score for each solution generated under different meta-heuristic configurations in OptaPlanner : Simulated Annealing (SA), Hill Climbing (HC), Lated Acceptance (LA), Great Deluge (GD), and Variable Neighborhood Descent (VND), while utilizing the same constructive algorithm : First Fit. As demonstrated in Table 3 and Table 4, we further detail the score generated by each activated constraint.

Table 4: Comparison of solution scores: case FR.

| | SA | HC | LA | GD | VND |
|---|---|---|---|---|---|
| C. 1 | 0 | 0 | 0 | 0 | 0 |
| C. 2 | 0 | 0 | 0 | 0 | 0 |
| C. 3 | 0 | 0 | 0 | 0 | 0 |
| C. 4 | 0 | 0 | 0 | 0 | 0 |
| Λ | 0 | 0 | 0 | 0 | 0 |
| C. 5 | -4850 | -3552 | -11448 | -12458 | -8055 |
| C. 7 | -95207 | -89320 | -112087 | -113751 | -87368 |
| C. 9 | -711408 | -883310 | -2180250 | -3153722 | -3130840 |
| Ω | **-811465** | -976182 | -2303785 | -3279931 | -3226263 |

Examining Table 3 and Table 4, our initial conclusions are:

- Except for the situation where case ES uses GD for solution generation and fails to find a feasible solution due to its hard score $\Lambda(caseES, GD) = -6$, all examined meta-heuristics successfully find feasible solutions for both cases within an 8-minute computing time, even when the size of case FR is sufficiently large.
- In terms of solution quality for the two tested cases, as indicated by the soft score Ω, the best solution for case ES is achieved by LA with $\Omega(caseES, LA) = -196613$, while for case FR, SA produces the optimal solution with $\Omega(caseFR, SA) = -811465$. This demonstrates that the choice of meta-heuristic for solution generation depends on the specific use case.
- When analyzing the detailed scores for each activated constraint, specifically C.9 for achieving weekly workload balance, these values significantly influence the overall score of each solution. However, the interpretation of meaning behind this score is unclear. Consequently, we will introduce an indicator for this constraint later to better demonstrate its benefit.

For C.9, we assess the balance of workload for each working week among all staff members who have completed their assigned tasks. This is done through the introduction of a new indicator, calculated using formula (10), which calculates the average Mean Deviation (MD) of each staff member's total workload per day. The formula is represented as:

$$\underset{s \in \mathbb{S}}{M_{week}} = \frac{\sum_{i=1}^{h^*} MD(workload_s \text{ per working day})}{h^*} \quad (10)$$

Here, $h^*$ refers to the number of working days in the scheduling horizon, excluding weekends and holidays. *week* refers to the week number within the planning horizon for each use case.

For the case ES, the scheduling horizon comprises only 5 days, equivalent to 1 week. Table 5 illustrates the value $M_{week}$ for the solutions outlined in Table 3. These findings confirm that the use of LA in solution generation achieves the optimal balance in weekly workload for all staff members. This is reflected in $M(caseES, LA)_1 = 4.18$, with a gap of $= 4.18\%$ compared to the theoretical best value of 0. We observe that the weekly workload balance for GD seems to be quite far from other values when compared to other methods (e.g. : $M(caseES, GD)_1 = 58.32$ vs.

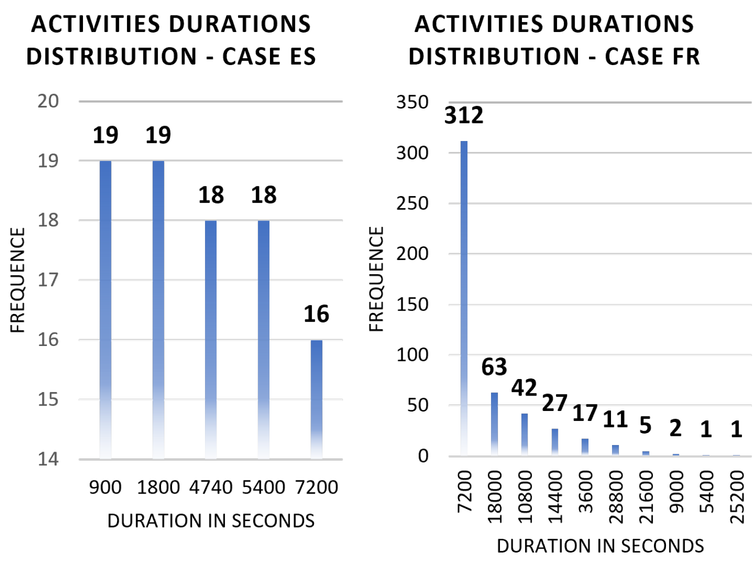



Figure 6: Activities' duration distributions for two cases.

Table 5: Weekly workload balance analysis: case ES.

| | SA | HC | LA | GD | VND |
|---|---|---|---|---|---|
| $M_1$ | 5.15 | 5.55 | **4.18** | 58.32 | 5.46 |

$M(caseES, LA)_1 = 4.18$ ). This difference arises because the final solution score generated by GD is reached after just 2 minutes of computation, with no further improvement observed over the remaining 6 minutes. This indicates that GD, when used for solution generation in our model with the given constraints of this use-case, results in being entrapped in local optima.

For the case FR, the scheduling horizon covers 26 days, equivalent to 4 weeks. As shown in Figure 7, the results confirm that the use of SA in solution generation achieves the best balance of weekly workload for all staff members. SA consistently produces the lowest $M_{week}$ over 4 weeks, with respective gaps compared to the theoretical best value of 0:

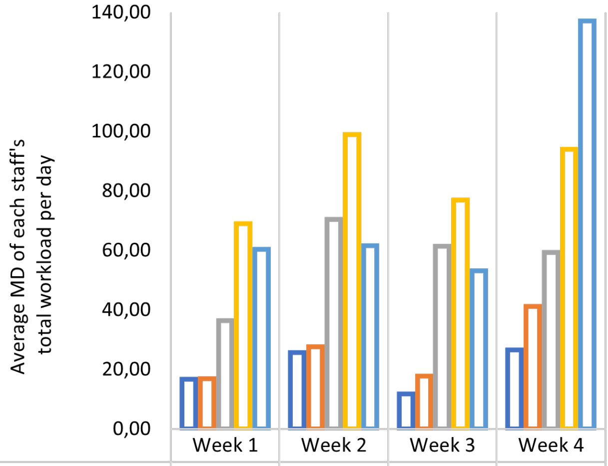
- $gap_{week1} = 16.73\%$
- $gap_{week2} = 25.73\%$
- $gap_{week3} = 11.83\%$
- $gap_{week4} = 26.64\%$

In conclusion, given the numerous staff unavailabilities within their workloads, especially for weeks 2 and 4, we observe a significant decentralization of staff unavailabilities, as shown in Figure 5. Consequently, the optimized solution is less effective in achieving an optimal weekly workload balance.

# 5 CONCLUSIONS, DISCUSSIONS AND FUTURE WORKS

This paper introduces a two-level scalable optimization framework designed to address large-scale, real-world problems with a multi-day horizon. The *Optimizer* is currently conceptualized and implemented using OptaPlanner, an open-source AI solver based on Constraint Satisfaction Problems (CSP). To address diverse scheduling requirements, we propose a scalable *Adapter* to parameterize the configurable constraint system within our Optimizer. In the experimental section, we present two real-life use cases subjected to our OptaPlanner-based Scheduling and Routing Problem (SRP) model under different configurations. The results demonstrate the achievement of high-quality optimized solutions within a short computing time. In addition, a detailed investigation is conducted to interpret the scores of optimized solutions in an understandable manner.

For future work, firstly, in terms of experimentation, we plan to expand the experimental results presented in this paper. This would provide more comprehensive evidence of the framework's scalability and adaptability across various scenarios. Incorporating additional case studies or presenting more detailed performance metrics under varying conditions (e.g., staff size, scheduling complexity) would enhance the findings and enable clients to evaluate the framework's robustness across a broader spectrum of real-world situations. A deeper analysis of the experimental outcomes is necessary, focusing on why certain configurations performed better or worse




| | Week 1 | Week 2 | Week 3 | Week 4 |
|---|---|---|---|---|
| Simulated Annealing (SA) | 16,73 | 25,73 | 11,83 | 26,64 |
| Hill Climbing (HC) | 16,82 | 27,57 | 17,80 | 41,17 |
| Lated Acceptance (LA) | 36,40 | 70,32 | 61,41 | 59,29 |
| Great Deluge (GD) | 68,97 | 98,93 | 76,84 | 94,04 |
| Variable Neighborhood Descent (VND) | 60,43 | 61,56 | 53,20 | 137,17 |

Figure 7: Weekly workload balance analysis : case FR.

and linking these findings to practical implications. Moreover, a critical reflection on the limitations of the study (e.g., computational time constraints, specific constraints that may limit applicability) will lend transparency and allow for a more balanced interpretation of the results.

Next, we will conduct additional interviews with other clients to explore their scheduling requirements. Additionally, assigning weights to each constraint to reflect their importance for different clients will improve our configurable constraint system, allowing it to effectively meet additional SRP requirements.

As this work is ongoing, we also plan to integrate additional solvers into our *Optimizer*, enabling a comparative evaluation of the solutions generated by different solvers. This analysis will help establish the framework's competitive advantage and highlight its unique benefits. Such a comparison will provide users with a clearer understanding of the framework's effectiveness relative to existing solutions.

Furthermore, our current resolution approach presented in this article focuses on combinatorial optimization theories and does not account for uncertainties. Although our approach can efficiently generate near-optimal solutions, addressing operational uncertainties using stochastic formulations of SRP is challenging due to the complexity of the problem. To overcome this limitation, we are considering integrating our proposed approach with algorithms based on a multi-agent system. This new direction will enable posterior sensitivity analysis by utilizing a multi-agent system in conjunction with solutions generated by the optimization process. During this analytical phase, we can introduce various uncertain variables, supplementing those already identified in our field observations, such as uncertainties related to travel duration or staff availability.

# REFERENCES

Bays, C. (1977). A comparison of next-fit, first-fit, and best-fit. *Communications of the ACM*, 20(3):191–192.

Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219.

Burke, E. K. and Bykov, Y. (2017). The late acceptance hill-climbing heuristic. *European Journal of Operational Research*, 258(1):70–78.

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., and Juan, A. A. (2014). Rich vehicle routing problem: Survey. *ACM Comput. Surv.*, 47(2).

Carello, G. and Lanzarone, E. (2014). A cardinality-constrained robust model for the assignment problem in home care services. *European Journal of Operational Research*, 236(2):748–762.

Coelho, L. C., Gagliardi, J.-P., Renaud, J., and Ruiz, A. (2016). Solving the vehicle routing problem with lunch break arising in the furniture delivery industry. *Journal of the Operational Research Society*, 67:743–751.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.

Dekhici, L., Redjem, R., Belkadi, K., and El Mhamedi, A. (2019). Discretization of the firefly algorithm for home care. *Canadian Journal of Electrical and Computer Engineering*, 42(1):20–26.

Di Gaspero, L. and Urli, T. (2014). A cp/lns approach for multi-day homecare scheduling problems. In *International workshop on hybrid metaheuristics*, pages 1–15. Springer.

Di Mascolo, M., Martinez, C., and Espinouse, M.-L. (2021). Routing and scheduling in home health care: A literature survey and bibliometric analysis. *Computers & Industrial Engineering*, 158:107255.

Diana, M. and Dessouky, M. M. (2004). A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological*, 38(6):539–557.

Dueck, G. (1993). New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational physics*, 104(1):86–92.

Gao, J., Sun, L., and Gen, M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Computers & Operations Research*, 35(9):2892–2907. Part Special Issue: Bio-inspired Methods in Combinatorial Optimization.

Glover, F. and Laguna, M. (1998). *Tabu Search*, pages 2093–2229. Springer US, Boston, MA.

Goldfeld, S. M., Quandt, R. E., and Trotter, H. F. (1966). Maximization by quadratic hill-climbing. *Econometrica*, 34(3):541–551.

Gutin, G. and Punnen, A. P. (2006). *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media.

Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., and Yagiura, M. (2005). Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation science*, 39(2):206–232.

Jünger, M., Reinelt, G., and Rinaldi, G. (1995). Chapter 4 the traveling salesman problem. In *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 225–330. Elsevier.

Katayama, N., Chen, M., and Kubo, M. (2009). A capacity scaling heuristic for the multicommodity capacitated network design problem. *Journal of computational and applied mathematics*, 232(1):90–101.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Lattner, C. and Adve, V. (2002). Automatic pool allocation for disjoint data structures. *SIGPLAN Not.*, 38(2 supplement):13–24.

Macik, B. M. (2016). Case management task assignment using optaplanner. *Unpublished doctoral dissertation. Master's thesis], Masaryk University Faculty of Informatics*.

Nguyen, M. A., Dang, G. T.-H., Hà, M. H., and Pham, M.-T. (2022). The min-cost parallel drone scheduling vehicle routing problem. *European Journal of Operational Research*, 299(3):910–930.

Quintanilla, S., Ballestín, F., and Pérez, Á. (2020). Mathematical models to improve the current practice in a home healthcare unit. *Or Spectrum*, 42:43–74.

Rashidnejad, M., Ebrahimnejad, S., and Safari, J. (2018). A bi-objective model of preventive maintenance planning in distributed systems considering vehicle routing problem. *Computers & Industrial Engineering*, 120:360–381.

Rest, K.-D. and Hirsch, P. (2016). Daily scheduling of home health care services using time-dependent public transport. *Flexible services and manufacturing journal*, 28:495–525.

Rios de Souza, V. and Martins, C. B. (2020). Uma análise do framework optaplanner aplicado ao problema de empacotamento unidimensional. *Revista de Sistemas e Computação-RSC*, 9(2).

Semeria, C. (2001). Supporting differentiated service classes: queue scheduling disciplines. *Juniper networks*, 27:11–14.

Smirnov, A. and Shilov, N. (2010). Ai-based approaches to solving a dynamic logistics problem. *KI-Künstliche Intelligenz*, 24:143–147.

Van Laarhoven, P. J. M. and Aarts, E. H. L. (1987). *Simulated annealing*, pages 7–15. Springer Netherlands, Dordrecht.

Yalçındağ, S., Matta, A., Şahin, E., and Shanthikumar, J. G. (2016). The patient assignment problem in home health care: using a data-driven method to estimate the travel times of care givers. *Flexible Services and Manufacturing Journal*, 28:304–335.

Zhang, L., Fontanili, F., Lamine, E., Bortolaso, C., Derras, M., and Pingaud, H. (2021). Stakeholders' tolerance-based linear model for home health care coordination. *IFAC-PapersOnLine*, 54(1):269–275. 17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021.

Zhang, L., Pingaud, H., Fontanili, F., Lamine, E., Martinez, C., Bortolaso, C., and Derras, M. (2023). Balancing the satisfaction of stakeholders in home health care coordination: a novel optaplanner csp model. *Health Systems*, 12(4):408–428.