



HAL
open science

Dynamic, fair, and efficient routing for cooperative autonomous vehicle fleets

Aitor López Sánchez, Marin Lujak, Frédéric Semet, Holger Billhardt

► **To cite this version:**

Aitor López Sánchez, Marin Lujak, Frédéric Semet, Holger Billhardt. Dynamic, fair, and efficient routing for cooperative autonomous vehicle fleets. *Expert Systems with Applications*, 2024, 251, pp.123964. 10.1016/j.eswa.2024.123964 . hal-04610658

HAL Id: hal-04610658

<https://hal.science/hal-04610658>

Submitted on 13 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamic Vehicle Routing Problem with Fair Profits and Time Windows (DVRP-FPTW): Formulation and Distributed Multi-Agent Approach

Aitor López Sánchez^{a,b}, Marin Lujak^{a,*}, Frederic Semet^b, Holger Billhardt^a

^a*CETINIA, University Rey Juan Carlos, Madrid, Spain*

^b*Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRISTAL, F-59000 Lille, France*

Abstract

This paper models and solves the Dynamic Vehicle Routing Problem with Fair Profits and Time Windows (DVRP-FPTW). The objective is to dynamically optimize routes for a fleet of vehicles serving customers within assigned time windows, aiming at fair and efficient solutions. The proposed DVRP-FPTW model builds upon VRP-FPTW by López et al. [1], ensuring adherence to customer demand, vehicle capacities, and autonomies, extending it to accommodate unforeseen events during fleet operation, such as customers' cancellations, or vehicle breakdowns. The model incorporates mandatory (previously assigned to operational vehicles) and optional (not assigned) customers, ensuring that optional ones are integrated into operational vehicle routes while not worsening their profits. The proposed distributed Multi-Agent System (MAS) solution approach utilizing asynchronous column generation heuristics dynamically adapts to unforeseen events. Systematic Egalitarian social welfare optimization is employed to iteratively maximize the profit of the least profitable vehicle, defined as the difference between the revenue earned from the visited customers and the travel cost. By prioritizing less profitable vehicles, we ensure fair treatment across the fleet considering unforeseen events, a significant improvement over existing dynamic and multi-period VRP models that rely on advance knowledge of demand changes. Our approach allows vehicle agents to maintain privacy while sharing minimal local data with a fleet coordinator agent. We propose publicly available benchmark instances for both static and dynamic VRP-FPTW. Simulation results demonstrate the ability of our DVRP-FPTW model and MAS solution approach to coordinate large, dynamically evolving cooperative fleets fairly, efficiently, and effectively in close to real-time.

Keywords: Multi-agent coordination, distributed optimization, column generation, VRP with time windows, fairness, egalitarian welfare, cooperatives

*Corresponding author

Email addresses: aitor.lopez@urjc.es (Aitor López Sánchez), marin.lujak@urjc.es (Marin Lujak), frederic.semet@centralelille.fr (Frederic Semet), holger.billhardt@urjc.es (Holger Billhardt)

1. Introduction

In agriculture cooperatives, where individual rationality meets mutual competition among farmers managing fragmented and dispersed lands with precision farming, there arises an ever-growing need for fair and efficient coordination of shared agriculture vehicles. This demand becomes pronounced as farmers aim to enhance their competitiveness across the industry and is particularly evident in sharing expensive autonomous tractor and agricultural robot (hereafter referred to as vehicle) fleets.

Agricultural cooperatives, on the other hand, as member-owned and controlled organizations, pursue common goals for both individual and mutual economic or social benefits. These organizations operate on principles that promote democratic control, active member participation, and equitable distribution of benefits for all members and, thus, inherently balance collective efforts and individual profit motives. In agricultural cooperatives, while the primary goal is collaborative work, an inherent tension exists between the collective efforts of the group and each farmer’s pursuit of maximizing individual profits within the cooperative’s regulations and value system. This inherent discord is addressed through the foundational values statement, which encapsulates the cooperative’s core principles and code of ethics. Given the natural conflict between collective objectives and individual profit motives, farmers, driven by individual rationality, choose to cooperate within the cooperative framework when the benefits outweigh those of competition. Furthermore, they are inclined to retain autonomy and control of their operational and financial decisions rather than handing over such authority to external entities.

In this context, we study the capacitated Vehicle Routing Problem with Profits (VRPP) [2, 3] where a vehicle fleet services (e.g., delivers goods to) a set of geographically dispersed customers (tasks). Each customer is associated with a revenue collected by the vehicle that serves them. Vehicle profit is defined as the difference between the revenue earned from the visited customers and its travel cost. The pivotal question we address in this paper is how to dynamically optimize a cooperative fleet’s profits. The challenge lies in balancing efficiency and fairness, considering both the cooperative’s commitment to equitable distribution and operational efficiency and the occurrence of unexpected events during fleet operation. Such events (encompassing, e.g., task alterations, vehicle additions, breakdowns or in-operation repairs) may necessitate prompt and flexible route adjustments.

Therefore, the study at hand models and solves the Dynamic Vehicle Routing Problem with Fair Profits and Time Windows (DVRP-FPTW) in this complex context, addressing the challenge of dynamically optimizing fleet profits while accommodating unforeseen events. Additionally, it adheres to vehicle capacity and autonomy constraints, preventing any vehicle from exceeding the capacity and maximum allowed distance between two depot visits, while also meeting customer demands within specified time windows.

The proposed DVRP-FPTW model incorporates mandatory and optional customers. Mandatory customers are those that, before the disruption event, were assigned to vehicles that remain operational after the event, while optional customers are those left unassigned following the event. Contrary to current models in the literature that do not oblige to visit each customer (e.g., [3]), our proposed model ensures the servicing of all mandatory customers. It incorporates optional customers into operational vehicle routes only when their inclusion does not negatively impact the profit of the vehicles.

The DVRP-FPTW is a computationally complex and intrinsically distributed problem; thus, we decompose it and apply a column generation method that initiates with a small set of variables corresponding to an initial set of routes. It iteratively generates additional variables (columns) when necessary to enhance the current solution, ultimately converging towards the optimal solution. This approach is seamlessly integrated into the Distributed Multi-Agent System (MAS) Architecture proposed in this paper, that we name DIMASA, comprising a fleet coordinator and vehicle agents, to produce a scalable and computationally efficient solution suitable for real-world application.

To ensure both fairness and efficiency in dynamic environments, we propose a systematic egalitarian social welfare optimisation approach that aims to iteratively maximize the profit of vehicles' routes in a non-decreasing order of vehicle profitability across the fleet. In addition, the column generation method maintains the solution quality and guarantees to find an optimal solution if we integrate it into a Branch and Price method. It allows for a distributed, parallel, and asynchronous computation, particularly useful for routing of large-scale real-world cooperative vehicle fleets in close-to real-time and avoids the need to share private vehicle data.

The paper is structured as follows. In Section 2, we give an overview of the state of the art. Section 3 presents the DVRP-FPTW problem formulation where we decompose the problem using a column generation (also called pricing) approach to incorporate it into the proposed DIMASA architecture. This new architecture together with our proposed distributed and scalable solution approach for the DVRP-FPTW problem is presented in Section 4. In Section 5, we give a use-case example and assess the performance of our approach on a set of publicly available benchmark instances that we propose in this paper for the static and dynamic versions of the VRP-FPTW. Here, we present the setup and results of simulation experiments with related discussion. We close the paper in Section 6 with conclusions and future work.

2. Related work

In the domain of collaborative vehicle routing problems, the sole pursuit of maximum overall benefit for the fleet is complemented by considerations for service quality, equity, and fairness (e.g., [4, 5, 6]).

In our previous work [7], we introduced the Agriculture Fleet Vehicle Routing Problem (AF-VRP), a novel problem that, to the best of our knowledge, deviates from any previously studied versions of the Vehicle Routing Problem. Uniquely

characterized by its intrinsic dynamism and decentralization, and with a specific focus on agriculture cooperatives, our approach necessitates the consideration of fundamental concepts such as fairness and equity. Our investigation delves into the exploration of how vehicles can collaborate to optimize their profits while ensuring an equitable distribution of profits.

Soriano et al. [8] investigate how to solve the Multi-Depot Vehicle Routing Problem (MDVRP) with added profit fairness constraints. They consider each depot as an independent partner with its fleet and propose a bi-objective optimization problem that minimizes the total cost and maximizes the fairness of profit distribution. A multi-objective optimization approach focusing on maximin fairness was proposed in [9] aiming to maximize the level of satisfaction for the least satisfied owner. Furthermore, [10] introduces a novel vehicle routing problem with vector profits, where each customer's revenue is linked to multiple stakeholders, aiming at maximizing the total profit while prioritizing the least satisfied stakeholder. The authors adopted a column-generation approach to solve the problem.

Fairness between customers is another important consideration in addition to the fairness between vehicles. Stavropoulou et al. [11] study a Vehicle Routing Problem (VRP) with profits and consistency constraints that uses a mixed set of mandatory and optional customers to visit. They determine profitable vehicle routes that maximize the net profit, while satisfying vehicle capacity, route duration and consistency constraints. Mancini et al. [12] and Rodríguez-Martín et al. [13] considered the distribution of customers' requirements over multiple days with added driver consistency constraints where customers must be served by the same vehicle over time.

The Dynamic Vehicle Routing Problem (DVRP) is a widely studied problem, which adapts to demand and traffic conditions that vary over time (e.g., [14, 15]). The solution methods to DVRP can be classified into heuristics, exact methods and hybrid methods. Most exact methods are based on Mixed Integer Linear Programming (MILP) models, which find an optimal solution for the current state. However, they may not remain optimal or even feasible as conditions evolve. Therefore, exact methods generally solve small instances or are used in conjunction with heuristics to obtain good-quality solutions for more complex problems (e.g., [16, 17, 18, 19, 20, 21]).

Distributed solution approaches to the DVRP generally utilize multi-agent systems (MAS) where autonomous agents interact to achieve a common goal [21, 22]. These systems enable decentralized dynamic vehicle routing, where autonomous agents coordinate decisions in response to the evolving conditions of the problem. While MAS approaches to the DVRP can manage large instances and dynamic scenarios, solution quality is not always guaranteed (e.g., [23, 22, 21]).

Barbucha [24, 25] proposes a multi-agent approach to the DVRP where agents address continuous customer requests by reallocating vehicles using a Variable Neighborhood Search method. Lujak et al. [26] confront dynamic task allocation in large, open, and collaborative vehicle fleets, using MAS to assign agent-represented vehicles to dynamically appearing tasks. In addition to these

methods, auction-based methods address larger-scale problems [27, 28]. For example, customers’ demand can be bundled and auctioned off to carriers who bid for the right to serve them. These approaches, although without quality of solution guarantees, offer flexible resource allocation and can manage large-scale scenarios within limited computation times.

Robust Vehicle Routing Problems address uncertainties in customer demands, travel times, and service times (e.g., [29]). Agra et al. [30] apply mathematical programming to ensure route feasibility within time windows across all travel times. Duan et al. [31] employ robust multi-objective particle swarm optimization to handle disturbances. The periodic VRP with service time uncertainty is addressed using a Variable Neighbourhood Search algorithm based on the worst case in [32]. Guo et al. [33] propose a two-phase method including optimal robust routes for all customers in the first phase and a multi-objective particle swarm optimization for dynamically appearing customers in the second phase.

The previous approaches do not account for the possibility of vehicle breakdowns, which, in real-world scenarios, can significantly disrupt the routing plans and result in delays, missed appointments, and additional costs. Seyyedhasani et al. [34] propose a re-optimization algorithm for the DVRP to address unexpected changes in field conditions or machinery management, transitioning from a single-depot to a multi-depot VRP to accommodate in-progress vehicle locations and capabilities. They propose a Dynamic VRP and change the initial VRP with one depot to a Multi-Depot VRP where each new depot corresponds to the current position of the vehicles when something goes wrong. A similar idea is the one we use in this article, although with a distributed approach in which the coordinator does not know where the vehicles are. This information is handled only by the vehicles themselves and they calculate their routes based on their current characteristics.

Contribution of the paper. Led by the open challenges identified in agriculture fleet vehicle routing, as presented in [7], in this paper, we propose the dynamic VRP-FPTW model that extends the static model from [1] by integrating mechanisms to adjust pre-determined routes in response to unforeseen events. The proposed problem and related distributed multi-agent based solution approach are a continuation of our previous works ([1, 35, 36]), where we proposed deterministic offline models for balancing fairness and efficiency of vehicle routes for the (static) multiple Travelling Salesman Problem and the static VRP-FPTW.

3. Dynamic VRP-FPTW

In this Section, we first provide a short description of the static VRP-FPTW, followed by the description of its dynamic counterpart. The objective is to develop a decomposed approach for the Dynamic VRP-FPTW that can be integrated into the DIMASA architecture composed of a fleet coordinator and vehicle agents, which we propose in this paper.

3.1. Static VRP-FPTW

Consider a complete arc-weighted digraph $G = (V, A)$, where $V = \{0, \dots, |V| - 1\}$ is the vertex set of size $|V|$, and A is the set of arcs $(i, j) \in V \times V$ with $i \neq j$ representing the shortest paths between any two distinct vertices i and j in a transportation network embodied by graph G . Let d_{ij} denote the distance of arc $(i, j) \in A$, and let vertex $0 \in V$ be designated as the depot. The customer (task) vertices, represented by set $N = \{1, \dots, |V| - 1\}$ are the nodes to be visited or serviced.

Each customer $i \in N$ has an associated demand q_i to be satisfied and a nonnegative revenue r_i , with the depot assigned a revenue of $r_0 = 0$. Customers are to be serviced within a specific time window $[l_i, u_i]$, where l_i and u_i denote the earliest and latest allowable visit times, respectively. A fleet K consisting of $|K|$ potentially heterogeneous vehicles is initially stationed at the depot vertex 0 whose time window $[l_0, u_0]$ determines the start and the end time of the mission, respectively. Each vehicle $k \in K$ has distinct characteristics: travel speed sp_k , autonomy (maximum travel distance) D_k , resource carrying capacity Q_k , and travel cost per unit of distance travelled o_k . The travel time t_{ijk} from vertex i to j for vehicle k is computed as $t_{ijk} = d_{ij}/sp_k$ and the corresponding travel cost is $c_{ijk} = d_{ij} \times o_k$.

The objective of the (static) VRP-FPTW, as detailed in [1], is to find a set of routes, one for each vehicle, that maximize the profit of the worst-off vehicle. This profit is calculated as the difference between the cumulative revenue from the visited customers and the total cost of the route. The fleet's mission is to visit each customer within their specified time window exactly once by exactly one vehicle, while fully meeting their demand. The fleet must start from and return to the depot after completing the service. The constraints include adherence to the vehicles' autonomy and carrying capacity. Specifically, the total travel time and resource demands of the customers to be visited, represented by the length of the route (dicycle) and the load carried, must not exceed the autonomy and load capacity of any vehicle.

For the self-completeness of this work, in the following, we give the Static Restricted Master Problem (SRMP) of the Static VRP-FPTW formulation (11)-(15) in [1] whose aim is to find the values of binary decision variables λ_k^p if route $p \in \Omega_k$ is assigned to vehicle k , where Ω_k is the set of feasible routes of vehicle k .

$$\max \quad y \tag{1}$$

$$s.t. \quad \sum_{k \in K} \sum_{p \in \Omega_k} a_{jk}^p \lambda_k^p = 1, \quad \forall j \in N, \tag{2}$$

$$\sum_{p \in \Omega_k} w_k^p \lambda_k^p \geq y, \quad \forall k \in K, \tag{3}$$

$$\sum_{p \in \Omega_k} \lambda_k^p = 1, \quad \forall k \in K, \tag{4}$$

$$y \in \mathbb{R}, \lambda_k^p \in \{0, 1\}, \quad \forall k \in K, \forall p \in \Omega_k. \tag{5}$$

where parameter $a_{jk}^p = 1$ if the customer $j \in N$ is visited on route p by vehicle k , and 0 otherwise, while parameter w_k^p is the profit obtained on route p by vehicle k . Constraints (2) are the one-on-one vehicle-customer assignment constraints. Fairness constraints (3) fix the minimum profit for each vehicle to y ; vehicle constraints (4) guarantee the use of all vehicles, while constraints (5) define variable domains. The objective function (1) maximizes profit y of the worst-off vehicle. More details can be found in [1].

The static VRP-FPTW model is designed to accommodate both heterogeneous and homogeneous vehicle fleets. It imposes no restrictions on the structure of the transportation graph, nor on the number of vehicles or customers. However, the model operates under the assumption that there is a sufficient number of vehicles to meet all customers. Thus, there may be instances where no feasible solution exists due to too restrictive constraints of the problem. To address this, we recommend implementing data pre-processing techniques that are essential for identifying such scenarios where the constraints of vehicle capacity, customer demand, and time windows might render the problem infeasible.

3.2. Dynamic VRP-FPTW formulation

The Dynamic VRP-FPTW (DVRP-FPTW) addresses the limitations of static routing in environments where conditions can change unpredictably. We define a disruptive event e as an unforeseen occurrence or change that impacts the routing solution established beforehand. The disruptions include fluctuations in customer demand, unforeseen vehicle breakdowns and unexpected delays, rendering visits to customers within their designated time windows infeasible as well as new vehicle additions to the fleet, among others.

Let $z^e \in [l_0, u_0]$ be the time of such an event. We assume that the time is synchronized throughout the fleet's vehicles. Let K' represent the set of post-disruption operational vehicles, including any possible new vehicle(s) and excluding the ones that are not operational. We define mutually exclusive subsets of customers N at time z : N_1 (already visited customers), N_2 (current locations of operational vehicles after disruption), N_3 (mandatory customers, i.e., the customers to be visited and previously assigned to set K' of operational vehicles after disruption), and N_4 (optional customers, i.e., both new customers and the customers that had been assigned to the vehicles that were impacted by a disruptive event and have not been operational afterwards). For simplicity, we assume that if a vehicle that is operational after disruption is en route to customer $i \in N$ at time z of the event, it is considered at i 's location after disruption.

Encompassing these changes, the revised routing problem is modelled on a new complete arc-weighted digraph (V', A') where $V' = \{0\} \cup N_2 \cup N_3 \cup N_4$ and A' is the set of arcs $(i, j) \in V' \times V'$. The notation is summarized in Table 1.

3.2.1. Centralized formulation for the DVRP-FPTW

The DVRP-FPTW is formulated as a mixed-integer linear program. The model includes binary variables x_{ijk} indicating whether the route of vehicle k

Table 1: Parameters and decision variables of the DVRP-FPTW

Parameters	Symbol	Description
<i>Shared fleet information \mathcal{G}</i>		
Vertices	V'	Set of vertices, $V' = \{0\} \cup N_2 \cup N_3 \cup N_4$
Arcs	A'	Set of arcs $(i, j) \in V' \times V'$
Depot	0	Depot vertex, $0 \in V'$
Visited customers	N_1	Set of visited customers
Current locations	N_2	Set of vertices where the vehicles are currently located
Mandatory customers	N_3	Set of pending mandatory customer vertices
Optional customers	N_4	Set of pending optional customer vertices
Distances	d_{ij}	Travel distance of arc $(i, j) \in A'$
Revenue	r_i	Nonnegative revenue obtained by visiting customer $i \in N_3 \cup N_4$
Demand	q_i	Nonnegative demand associated with each customer $i \in N_3 \cup N_4$
Time window	$[l_i, u_i]$	Time window for visiting customer $i \in N_3 \cup N_4$
Vehicles	K'	Updated set of operational vehicles $k \in K'$
<i>Private information \mathcal{I} for each vehicle $k \in K'$</i>		
Travel cost	o_k	travel cost of vehicle k per unit of distance travelled
Travel time	t_{ijk}	Travel time of vehicle k on arc $(i, j) \in A'$
Autonomy	D'_k	Remaining autonomy of vehicle k in terms of maximum traveling distance
Capacity	Q'_k	Remaining capacity of vehicle k
Post-disruption locations	s_k	Post-disruption vertex location of vehicle k
Disruption time	z_k	Disruption time of vehicle k at vertex s_k
Profit	y_k	Accumulated profit of vehicle k
<i>Decision variables</i>		
Route-arc assignment	x_{ijk}	$x_{ijk} = 1$ if vehicle $k \in K$ travels arc $(i, j) \in A'$ and 0 otherwise
Customer visit time	v_i	Time of the visit of customer i

passes through arc $(i, j) \in A'$, continuous variables v_i representing the arrival time of a vehicle at vertex $i \in V'$ and a continuous variable y associated with the profit of the worst-off vehicle's route. We propose next the (centralized) formulation of the DVRP-FPTW.

$$\max \quad y \quad (6)$$

$$s.t. \quad \sum_{k \in K'} \sum_{i \in V'} x_{ijk} = 1 \quad \forall j \in N_3 \quad (7)$$

$$\sum_{k \in K'} \sum_{i \in V'} x_{ijk} \leq 1 \quad \forall j \in N_4 \quad (8)$$

$$\sum_{i \in V'} x_{ijk} - \sum_{h \in V'} x_{jhk} = 0 \quad \forall j \in (N_3 \cup N_4), \forall k \in K' \quad (9)$$

$$\sum_{j \in V'} x_{sjk} = 1 \quad \forall k \in K', s = s_k \quad (10)$$

$$\sum_{j \in V'} x_{j0k} = 1 \quad \forall k \in K' \quad (11)$$

$$\sum_{i \in V'} \sum_{j \in V'} d_{ij} x_{ijk} \leq D'_k \quad \forall k \in K' \quad (12)$$

$$\sum_{i \in V'} \sum_{j \in V'} q_j x_{ijk} \leq Q'_k \quad \forall k \in K' \quad (13)$$

$$v_i + t_{ijk} \leq v_j + M(1 - x_{ijk}) \quad \forall (i, j) \in A', j \neq 0, \quad \forall k \in K' \quad (14)$$

$$v_s = z_k \quad \forall k \in K', s = s_k \quad (15)$$

$$l_i \leq v_i \leq u_i \quad \forall i \in V' \quad (16)$$

$$y \leq y_k + \sum_{i \in V'} \sum_{j \in V'} (r_i - c_{ijk}) x_{ijk} \quad \forall k \in K' \quad (17)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A', \forall k \in K'. \quad (18)$$

The proposed DVRP-FPTW model (6)–(18) aims to incorporate set N_4 of optional customers into the routes of the operational vehicles that must service their mandatory customers in N_3 while striving to maintain or improve each vehicle's profit by including into their routes optional customers.

Objective function (6) maximizes the profit of the worst-off vehicle. Mandatory customer constraints (7) ensure that each mandatory customer in N_3 is visited once by exactly one vehicle, while optional customer constraints (8) ensure that every optional customer in N_4 is visited at most once. Flow conservation constraints (9) state that if a vehicle visits a customer vertex, it should also leave it. Constraints (10) and (11) ensure that each vehicle leaves its actual vertex and returns to the depot, respectively. Vehicle k 's adjusted parameters are remaining autonomy D'_k , remaining capacity Q'_k and profit y_k accumulated at the time of the disruptive event. Constraints (12) and (13) ensure that the

total travel time and resource demands of the customers to be visited by each vehicle $k \in K'$ do not exceed its remaining autonomy D'_k nor capacity Q'_k , respectively. Constraints (14) set the arrival time at each vertex and eliminate subtours.

Each vehicle $k \in K'$ starts its revised route at time z from its current location $s_k \in N_2$ where $N_2 = \{s_k : k \in K'\}$. We use a potentially different starting time for each vehicle k since a vehicle may have been on the way to the current customer.

THE DESCRIPTION OF CONSTRAINTS (15) IS MISSING.

Constraints (16) ensure that customer time windows are satisfied. We take in consideration the vehicles' profits accumulated before the disruptive event and do not reassign already serviced customers in set N_1 . Fairness constraints (17) keep track of the worst-off vehicle's profit y . Each vehicle k 's profit y_k accumulated from the customers visited before the disruptive event is added to the profit of its customers to be visited in the newly found route. In this way we balance the efficiency and fairness of the routes considering overall profit received by each vehicle in time horizon $[l_0, u_0]$.

3.2.2. Decomposed formulation for DVRP-FPTW

Next, we present a decomposed mathematical program designed for a multi-agent system, consisting of a coordinator agent and a fleet of vehicles with computation and communication capabilities.

Applying Dantzig-Wolfe decomposition [37, 38], the dynamic centralized model (6)-(18) is restructured into a Dynamic Restricted Master Problem (DRMP) formulation of the DVRP-FPTW. Each vehicle $k \in K'$ is considered with a set of feasible routes Θ_k , starting at the current location s_k and finishing in the depot 0.

We let a_{jk}^p be a binary parameter equal to 1 if route $p \in \Theta_k$ of vehicle k visits vertex j , and let w_k^p be the profit obtained in route $p \in \Theta_k$ by vehicle $k \in K'$. Moreover, we let δ_k^p be the decision variable that equals 1 if route $p \in \Theta_k$ is selected for vehicle k . The dynamic restricted master problem (DRMP) of the DVRP-FPTW is then:

$$\max \quad y \tag{19}$$

$$s.t. \quad \sum_{k \in K'} \sum_{p \in \Theta_k} a_{jk}^p \delta_k^p = 1, \forall j \in N_3, \tag{20}$$

$$\sum_{k \in K'} \sum_{p \in \Theta_k} a_{jk}^p \delta_k^p \leq 1, \forall j \in N_4, \tag{21}$$

$$\sum_{p \in \Theta_k} w_k^p \delta_k^p + y_k \geq y, \forall k \in K', \tag{22}$$

$$\sum_{p \in \Theta_k} \delta_k^p = 1, \quad \forall k \in K', \tag{23}$$

$$y \in \mathbb{R}, \delta_k^p \in \{0, 1\}, \quad \forall k \in K', \forall p \in \Theta_k, \tag{24}$$

Objective function (19) maximizes the profit y of the worst-off vehicle. Constraints (20) assure visiting each mandatory customer $j \in N_3$. Constraints (21) assure that all optional customers $j \in N_4$ are visited at most once. Fairness constraints (22) keep track of the minimum profit of the entire fleet K' after disruption. Constraints (23) guarantee that all post-disruption operational vehicles (K') are assigned a route.

The mathematical model (19)–(24) is used by the coordinator agent for the dynamic vehicle-route assignment and the computation of shadow prices (dual variables). The shadow prices are obtained by solving the linear relaxation of the DRMP. They may be interpreted in economic terms as the opportunity cost of resources and can be positive or negative. Specifically, assignment constraints (20) and (21) generate $\pi_j \in \mathbb{R}$ for each customer $j \in N_3 \cup N_4$, indicating the marginal cost of including customer j in a route. Likewise, fairness constraints (22) and fleet utilization constraints (23) produce shadow prices $\mu_k \leq 0$ and $\alpha_k \in \mathbb{R}$, respectively, for each vehicle $k \in K'$, representing the marginal cost associated with each vehicle’s operation. The column generation subproblem associated with the DRMP is the Elementary Shortest Path Problem with Resource Constraints (ESPPRC) [39].

4. DIMASA: Distributed Multi-Agent System Architecture for the DVRP-FPTW

We propose next a new decision-making architecture for the DVRP-FPTW, leveraging a distributed MAS that we name DIMASA. Illustrated in Figure 1, this architecture includes a Coordinator Agent and multiple Vehicle Agents that mutually collaborate to find best vehicle routes. The optimization process involves solving both the restricted master problem and dedicated column generation subproblems through a distributed and asynchronous implementation of the column generation approach, aligning with methodologies in ([40, 41]).

4.1. Coordinator agent

The coordinator agent ensures an equitable distribution of profit and efficiency across the fleet through *systematic optimization of egalitarian social welfare for VRP-FPTW* that we propose in this paper and that extends the model for the multiple Travelling Salesman Problem (mTSP) from [35, 36]. This approach prioritizes vehicle performance optimization in non-decreasing order of vehicle routes’ profits.

The coordinator’s decision-making process involves two iterative phases: the assignment of customer routes to the fleet’s vehicles, before the start of the fleet’s operation (phase 1), and the adaptation of these routes when unexpected events occur during operation (phase 2), Figure 1. Phase 1 assigns a set of customers N to the fleet set K based on the static VRP-FPTW model (1)–(5) following the Systematic Coordinator Agent Algorithm – SYSCA (Algorithm 1), while phase 2 considers sets of customers in N_3 and N_4 for the assignment among available vehicles in set K' by solving the DRMP model (19)–(24), following the Dynamic Systematic Coordinator Agent Algorithm – DSYSCA (Algorithm 3).

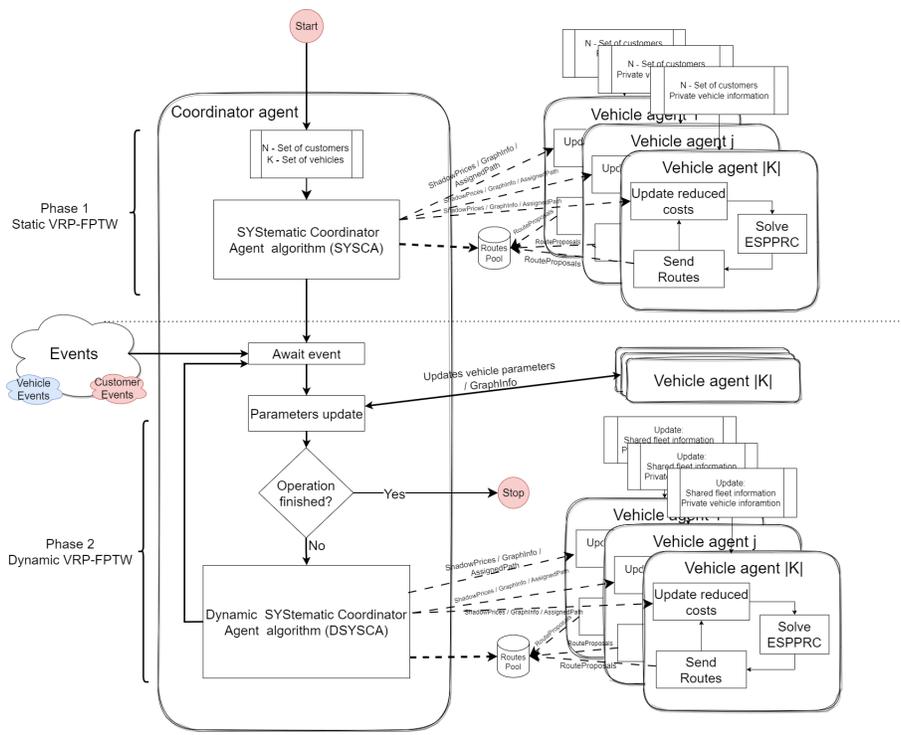


Figure 1: DIMASA architecture for the Static/Dynamic VRP-FPTW.

Algorithm 1: SYSCA Algorithm

Input: N - Set of customers, K - Set of vehicles.

Output: For all vehicles $k \in K$: Send routes p , where $(k, p) \in S$.

```
1  $K'' = K$  ;
2  $N'' = N$  ;
3  $S = \emptyset$  ;
4 while  $K'' \neq \emptyset$  do
5    $P = VRP - FTPW(N'', K'')$ ; // (Algorithm 2)
6    $(k_{min}, p_{min}) = \arg \min_{k \in K'', p \in P} \{w_k\}$  ;
7   Send AssignedRoute( $k_{min}, p_{min}$ ) to vehicle  $k_{min}$  ;
8    $K'' = K'' \setminus \{k_{min}\}$  ;
9    $N'' = N'' \setminus Customers(p_{min})$  ;
10   $S = S \cup \{(k_{min}, p_{min})\}$  ;
11 end
```

DIMASA architecture supports six types of messages between the coordinator and vehicle agents to manage information exchange: ‘*GraphInfo*’, ‘*VehicleEvent*’, ‘*AssignedRoute*’, ‘*ShadowPrices*’, ‘*RouteProposals*’, ‘*Acknowledged(k,msg)*’, and ‘*StopRouteFinding*’. Updated shared fleet information from Table 1 is found inside the ‘*GraphInfo*’ messages, required by the vehicles to update their graph information. ‘*VehicleEvent*’ message informs of a breakdown or addition of a vehicle into the fleet. An ‘*AssignedRoute*’ message sent by the coordinator indicates the route that a vehicle should follow. Also, private shadow prices μ_k and α_k are sent by the coordinator only to the corresponding vehicle $k \in K$ together with (shared) marginal costs π_j for each customer $j \in N_3 \cup N_4$ in a ‘*ShadowPrices*’ message. A ‘*RouteProposals*’ message sent from a vehicle agent to the coordinator agent contains a set of pairs $\{(k, p, w_k^p)\}$ with feasible routes with negative reduced costs p and their related profit w_k^p , if any. Ultimately, while the asynchronous nature characterizes the proposed column generation approach, it is essential to note the presence of a synchronization point within the devised solution strategy. This point of convergence is strategically facilitated through the utilization of a specific message, namely, ‘*StopRouteFinding*’.

4.1.1. Phase 1

In Phase 1, the coordinator agent uses the SYStematic Coordinator Agent (SYSCA) Algorithm (Algorithm 1) to iteratively assign a route to each vehicle across the fleet prioritizing the vehicles with the least profit.

SYSCA Algorithm (Algorithm 1). Initially, the coordinator agent receives the information about the set of customers N and vehicles K (Input). SYSCA runs over $|K|$ optimization iterations where the auxiliary sets N'' and K'' represent the pending unassigned customers and vehicles in each iteration, respectively.

In each iteration, the coordinator agent solves the static VRP-FPTW model by employing Algorithm 2 (CA Algorithm from [1]). This is done in collabora-

Algorithm 2: VRP-FPTW Algorithm

Input: N - set of customers, K - set of vehicles
Output: $P = \{(k, p, w_k), \forall k \in K\}$ - set of assigned routes

- 1 Send *GraphInfo()* to all vehicles $k \in K$;
- 2 Initialize routes;
- 3 Calculate shadow prices ($\forall j \in N : \pi_j, \forall k \in K : \mu_k$ and α_k) in SRMP;
- 4 Send *ShadowPrices*($\pi_j, \forall j \in N; \mu_k, \alpha_k$) to all vehicles $k \in K$;
- 5 **repeat**
- 6 **if** new ‘*RouteProposals*’($\{(k, p, w_k^p)\}$) received from any $k \in K$ **then**
- 7 $\Omega_k = \Omega_k \cup \{(k, p, w_k^p)\}$;
- 8 Calculate shadow prices ($\forall j \in N : \pi_j, \forall k \in K : \mu_k$ and α_k) in SRMP;
- 9 Send *ShadowPrices*($\pi_j, \forall j \in N; \mu_k, \alpha_k$) to all vehicles $k \in K$;
- 10 **end**
- 11 **until** *Termination criteria*;
- 12 Send *StopRouteFinding()* to each vehicle $k \in K$;
- 13 $\{(k, p), \forall k \in K\} = \text{SRMP}(\{\Omega_k, \forall k \in K\}, N)$;
- 14 $P = \{(k, p, w_k), \forall k \in K\}$;
- 15 Return P

tion with operational vehicle agents in K , where each one runs in parallel its own copy of the DRGV Algorithm (Algorithm 5)).

Based on the promising routes received from the vehicle agents, the coordinator agent selects a route for each one of the vehicles aiming at the maximization of the profit of the worst-off vehicle. Then, vehicle k_{min} with the minimum profit of its route p_{min} is selected and the coordinator sends an *AssignedRoute*(k_{min}, p_{min}) message to that vehicle indicating its assigned route. Vehicle k_{min} and customers (*Customers*(p_{min})) that are part of its route p_{min} , are removed from sets K'' and N'' , respectively. This process is repeated in subsequent iterations, to select the routes for the next worst-off vehicle (one by one) until all $|K|$ vehicles in the fleet are assigned a route.

CA Algorithm (Algorithm 2). For self-sufficiency of this work, we explain the VRP-FPTW solution approach – Algorithm 2 (CA Algorithm from [1]). This algorithm finds a route for each vehicle $k \in K$ by only focusing on the maximization of the profit of the worst-off vehicle.

The input to the algorithm is composed of a set of customers N and a set of vehicles K while the output is a set of vehicle routes $P = \{(k, p, w_k) \mid k \in K\}$ where the profit of the worst-off vehicle is maximized.

First, the coordinator sends a *GraphInfo()* message containing customers in N to all vehicles $k \in K$ (line 1). The routes are initialized on line 2 by creating an artificial solution for the SRMP model (1)-(5), where a single artificial vehicle visits all the customers obtaining a large (infinitely) negative profit. Thanks to this artificial solution, the linear relaxation of the model can be carried out.

Next, the coordinator calculates shadow prices by solving the linear relaxation of the SRMP (line 3), and sends a ‘*ShadowPrices*’ message to each vehicle agent with its dedicated private information, thus maintaining the privacy in the system (line 4).

Whenever the coordinator receives any vehicle’s new routes with negative reduced costs, it adds them into the set $\Omega = \{\Omega_k, \forall k \in K\}$ of columns for the SRMP (line 7) and updates and sends shadow prices to all vehicles $k \in K$ (lines 8 and 9). This asynchronous and iterative process is repeated until the Termination criteria are met (line 11).

Choosing appropriate termination criteria is a crucial algorithmic decision in column generation, as there exists no formula that works well for all instances [42, 41]. Termination criteria based on the number of iterations is not reliable in this context, since the messages exchanged among the vehicles and the coordinator agent are asynchronous, they can arrive at different iterations, so termination criteria based on the computation time or the number of columns are more suitable.

The criteria we implement in our solution approach include the receipt of an empty route proposal by all vehicles $k \in K$ (implying that vehicles cannot find any new routes with negative reduced costs), or when a given time limit is reached.

Based on the features and the analysis of our problem [1], we have fixed the time limit for the coordinator agent to 10 min for the static case and 1 min in the dynamic case, and for vehicle agent in 1 min and 10 seconds, respectively. However, in all the performed experiments, our algorithm finished before the time limit was reached, which shows its effectiveness.

However, there may exist vehicles that have not finished the computation process for finding their routes with negative reduced cost within this time limit. Thus, the coordinator agent sends to all vehicles a ‘*StopRouteFinding*’ message (line 12) to ensure that they are informed of the termination of Algorithm 2. Finally, the coordinator agent solves the SRMP (line 13) and the algorithm returns a set of routes $P = \{(k, p, w_k)\}$ (line 15). At termination, in the best case, every vehicle $k \in K$ is assigned route p , terminating the phase 1, after which, phase 2 starts. However, if the original artificial solution with the negative profit is found in the final assignment, the algorithm has not found a solution.

Thanks to the systematic optimization of egalitarian social welfare, the values of the worst-off profit y in one iteration may surpass the profit from the previous iteration(s). By adopting this iterative method, we ensure both equity and efficiency for all vehicles. The advantage of computing the systematic egalitarian social welfare in this way is that the solution of the SRMP takes advantage of the previously computed routes to produce new shadow prices and reiterate the process.

4.1.2. Phase 2

In phase 2, the coordinator agent assumes an idle state awaiting a disruptive event denoted as e within the operational timeframe $z \in [l_0, u_0]$.

Upon a disruptive event, the coordinator triggers the Dynamic Systematic Coordinator Agent Algorithm (DSYSCA) as delineated in Algorithm 3.

Algorithm 3: DSYSCA Algorithm

Input: K' - set of vehicles, N_3 - set of pending mandatory customers, N_4 - set of pending optional customers, e - disruptive event
Output: For all vehicles $k \in K'$: Send new routes p , where $(k, p) \in S'$.

```

1 if  $e$ : New_customer  $\{n\}$  then
2    $N_3 \leftarrow N_3 \cup \{n\}$  ;
3 end
4 if  $e$ : New_vehicle  $\{k\}$  then
5    $K' \leftarrow K' \cup \{k\}$  ;
6 end
7 if  $e$ : Vehicle_breakdown  $\{k\}$  then
8    $K' \leftarrow K' \setminus \{k\}$  ;
9    $N_4 \leftarrow N_4 \cup N^k$  ;
10 end
11  $N'_3 = N_3$ ,  $N'_4 = N_4$  ;
12  $S' = \emptyset$  ;
13  $K'' = K'$  ;
14 while  $K'' \neq \emptyset$  do
15    $P' = DVRP - FPTW(N'_3, N'_4, K'')$  ; // (Algorithm 4)
16    $(k_{min}, p_{min}) = \arg \min_{k \in K'', p \in P'} w_k$  ;
17   Send AssignedRoute $(k_{min}, p_{min})$  to vehicle  $k_{min}$  ;
18    $K'' = K'' \setminus \{k_{min}\}$  ;
19    $N'_3 = N'_3 \setminus Customers(p_{min})$  ;
20    $N'_4 = N'_4 \setminus Customers(p_{min})$  ;
21    $S' = S' \cup \{(k_{min}, p_{min})\}$ 
22 end

```

DSYSCA Algorithm (Algorithm 3). After classifying event e , Dynamic Systematic Coordinator Agent (DSYSCA) Algorithm updates the set of available vehicles K' , mandatory customers N_3 , and optional customers N_4 as follows: a new customer n and a new vehicle k are added to subset N_3 and K' , respectively (lines 2 and 4); a broken down vehicle k is removed from the fleet K' (line 8) and set $N^k \subset p_k$ of its unvisited customers are added to optional customers in N_4 (line 9).

Next, similar to the SYSCA algorithm (Algorithm 1), the coordinator agent instantiates auxiliary sets N'_3 , N'_4 , S' , and K'' representing the pending mandatory and optional unassigned customers, the solution set, and operational vehicles in each iteration, respectively (lines 11–13). The algorithm finds the routes for the vehicles following the systematic egalitarian social welfare optimization (lines 14–22). It performs $|K'|$ iterations where in each one, a route maximizing the profit of the worst-off vehicle k_{min} is found considering customers from N_3

and N_4 . In each iteration, the coordinator solves the DVRP-FPTW (Algorithm 4), line 15.

DVRP-FPTW Algorithm (Algorithm 4). This algorithm follows the idea of the VRP-FPTW Algorithm but solves the dynamic restricted master problem (DRMP) (19)-(24) to find shadow prices (lines 3 and 8). The coordinator keeps in its memory set Θ_k of feasible routes with negative reduced cost received from each post-disruption operational vehicle $k \in K'$.

Algorithm 4: DVRP-FPTW Algorithm

Input: N'_3 - Set of mandatory customers, N'_4 - Set of optional customers, K' - Set of operational vehicles
Output: $P' = \{(k, p, w_k), \forall k \in K'\}$ - set of assigned routes

- 1 Send *GraphInfo*(N'_3, N'_4) to all vehicles $k \in K'$;
- 2 Initialize routes;
- 3 Calculate shadow prices ($\forall j \in N'_3 \cup N'_4 : \pi_j ; \forall k \in K' : \mu_k$ and α_k) ;
- 4 Send *ShadowPrices*(π_j, μ_k, α_k) to all vehicles $k \in K'$;
- 5 **repeat**
- 6 **if** new ‘*RouteProposals*’($\{(k, p, w_k^p)\}$) received from any $k \in K'$ **then**
- 7 $\Theta_k = \Theta_k \cup \{(k, p, w_k^p)\}$;
- 8 Calculate shadow prices ($\forall j \in N'_3 \cup N'_4 : \pi_j ; \forall k \in K' : \mu_k, \alpha_k$) ;
- 9 Send *ShadowPrices*($\pi_j, \forall j \in N'_3 \cup N'_4; \mu_k, \alpha_k$) to all vehicles $k \in K'$;
- 10 **end**
- 11 **until** *Termination criteria*;
- 12 Send *StopRouteFinding*() to each vehicle $k \in K'$;
- 13 $\{(k, p), \forall k \in K'\} = \text{DRMP}(\{\Theta_k, \forall k \in K'\}, N)$;
- 14 $P' = \{(k, p, w_k), \forall k \in K'\}$;
- 15 Return P'

Whenever the coordinator receives any vehicle’s new routes with negative reduced costs, it adds them into the set $\Theta = \{\Theta_k, \forall k \in K'\}$ of columns for the DRMP (line 7) and calculates and sends shadow prices to all vehicles $k \in K'$ (lines 8 and 9). This asynchronous and iterative process is repeated until the Termination criteria are met (line 11).

‘*StopRouteFinding*’ message (line 12) is sent to all vehicles $k \in K'$ to inform them of the termination of the algorithm.

Solution $P' = \{(k, p, w_k), \forall k \in K'\}$ found by solving the DRMP contains an assigned route p for each vehicle $k \in K'$ and its associated profit w_k^p .

After an event is processed and new routes are obtained, the coordinator returns to the idle state awaiting new events. Once all the pending customers have been visited, the condition *Operation finished?* in Figure 1 returns “Yes” to indicate that the operation is over.

Algorithm 5: DRGV Algorithm followed by each vehicle $k \in K'$

Input: \mathcal{G} - shared fleet information, \mathcal{I}_k - private vehicle information
Output: $RouteProposals(\{(k, p, w_k^p)\})$ sent to coordinator agent

```
1 InitializeESPRC( $\mathcal{G}, \mathcal{I}$ );
2 terminate  $\leftarrow$  False ;
3 while !terminate do
4   msg  $\leftarrow$  WaitMessage();
5   if msg == GraphInfo() then
6     Send Acknowledged( $k, msg$ );
7     UpdateGraphInfo();
8   if msg == StopRouteFinding() then
9     Send Acknowledged( $k, msg$ );
10    terminate  $\leftarrow$  True;
11  if msg == AssignedRoute( $k, p$ ) then
12    Send Acknowledged( $k, msg$ );
13    do in parallel
14      RunRoute( $p$ )
15      if RouteFinished( $p$ ) then
16        Send RouteFinished( $k, p$ ) to coordinator agent
17        terminate  $\leftarrow$  True
18    goto line 4
19  if msg == ShadowPrices( $\pi_j, \forall j \in N; \mu_k, \alpha_k$ ) then
20    Send Acknowledged( $k, msg$ );
21     $C_G \leftarrow$  UpdateReducedCosts( $\pi_j, \forall j \in N; \mu_k, \alpha_k$ );
22     $\{p\} \leftarrow$  SolveESPPRC( $\mathcal{G}, \mathcal{I}, C_G$ );
23    Send RouteProposals( $\{(k, p, w_k^p)\}$ );
```

4.2. Vehicle agent

To generate new feasible routes with negative reduced cost (new columns in the SRMP in phase 1 and the DRMP in phase 2) in response to updated shadow prices received from the coordinator, each vehicle agent $k \in K$ maintains a message pool with the coordinator, wherein messages are retained until the vehicle agent reads and responds to them. The decision making of each vehicle is determined by the *Distributed Route Generation by Vehicle* (DRGV) Algorithm (Algorithm 5) that sends messages to the coordinator.

DRGV Algorithm (Algorithm 5). The algorithm initializes by receiving from the coordinator agent the shared fleet information \mathcal{G} (Table 1). Each vehicle k keeps in its memory its own private information \mathcal{I} .

The algorithm runs iteratively. In each iteration, a vehicle reads and processes the messages from the coordinator (lines 4-23). If the message is ‘*GraphInfo*’, it updates its own graph (line 7). The algorithm terminates when it receives a ‘*StopRouteFinding*’ message (line 8), which signals the termination of the

algorithm. Upon receipt of the ‘*AssignedRoute(k,p)*’ message (line 11), by *RunRoute(p)*, the vehicle starts visiting the tasks in its route and, in parallel, awaits messages for any route changes (line 18); if no messages are received before the route is finished, the vehicle sends *RouteFinished(k,p)* message to the coordinator (line 16) and the algorithm terminates.

Each time a vehicle receives a ‘*ShadowPrices*’ message with updated values μ_k, α_k, π_i for all $i \in N$ (line 19), it updates its reduced costs in $C_G = \{\hat{c}_{ijk} \mid (i,j) \in A\}$ (line 21). These updates are made as follows:

- $\hat{c}_{ijk} = (\pi_j + (r_i - c_{ijk}) \cdot \mu_k)$, for all arcs $(i,j) \in V \times N$, representing the revised reduced cost for arcs $(i,j) \in V \times N$ considering revenue r_i and cost c_{ijk} for vehicle k .
- $\hat{c}_{i0k} = (r_i - c_{i0k}) \cdot \mu_k + \alpha_k$, indicating the new reduced cost for all arcs $(i,0)$, where $i \in V$ returning to the depot 0.

In the *SolveESPPRC(G,I,C_G)* function (line 22), each vehicle agent resolves its own column generation subproblem, the Elementary Shortest Path Problem with Resource Constraints (ESPPRC), which is an NP-Hard problem [39].

The ESPPRC aims at finding the best routes from the current location s_k of vehicle k to the depot 0, with negative reduced cost and without violating any constraints. Because the ESPPRC is computationally expensive, we use two different algorithms to find good routes with negative reduced costs: 1) a heuristic algorithm based on the SPPRC (Shortest Path Problem with resource constraints [43]), which avoids cycles and compares routes with simple dominance based on the total reduced cost; 2) an exact algorithm that follows the ESPPRC rules as seen in [44], which makes sure that we do not miss any promising routes. Given the complexity of the ESPPRC, we have imposed a time limit within which the algorithm is expected to return a set of best solutions or an empty route if none is found. Subsequently, in the ‘*RouteProposals*’($\{(k,p,w_k^p)\}$) message (line 23), the vehicle agent transmits to the coordinator a set of its best feasible routes $\{p\}$ along with their associated profits w_k^p , thus preserving vehicle privacy.

The messages ‘*GraphInfo*’, ‘*AssignedRoute*’, ‘*StopRouteFinding*’, and ‘*Acknowledged*’ are crucial for the synchronization between the coordinator and the vehicle agents. These messages serve as synchronization points between the asynchronous algorithms. The communication system must ensure that these messages have been received timely and orderly before proceeding. If the coordinator does not receive an ‘*Acknowledged(k,msg)*’ from a vehicle, it is considered broken. The coordinator agent generates a ‘*Vehicle.breakdown{k}*’ event, adding its pending tasks to the set of optional customers N_4 .

5. Use case and functional experiments

In this Section, we first demonstrate the functioning of the proposed distributed MAS approach for DVRP-FPTW through a use case example and then discuss its computational complexity and demonstrate its scalability through extensive functional experiments.

5.1. Use case example

Consider the scenario illustrated in Figure 2. Suppose we have a set of 10 customers (tasks) that must be visited. The fleet is composed of 3 homogeneous vehicles initially positioned at the depot $\{0\}$, as shown in Figure 2a. In phase 1 of the DIMASA architecture, an initial routing plan is found. This plan assigns each vehicle to a specific set of customers, resulting in the following:

- Vehicle 1: Route $[0, 4, 6, 10, 8, 0]$ with a profit of 97.31.
- Vehicle 2: Route $[0, 5, 9, 3, 1, 0]$ with a profit of 165.78.
- Vehicle 3: Route $[0, 2, 7, 0]$ with a profit of 106.35.

Vehicle 1 starts its route, but at time $z^e = 20$, while it is visiting customer 6, it breaks down and does not send the *Acknowledged*(1, *msg*) message after receiving the *GraphInfo*() message from the coordinator agent, Figure 2b. The coordinator agent generates a *Vehicle.breakdown*{1} event. This event triggers adding the vehicle 1's pending customers $[8, 10]$ into the set of optional customers N_4 . The coordinator removes vehicle 1 from the set of functioning vehicles K' .

In this scenario, the customer sets are $N_1 = [4, 6, 5, 9, 2, 7]$, $N_2 = [9, 7]$, $N_3 = [3, 1]$ and $N_4 = [8, 10]$, and the private information of the vehicles is:

- Vehicle 1: $s_1 = 6, v_6 = 21.12, y_1 = 50.00, D'_1 = 75.32, Q'_1 = 37.0$
- Vehicle 2: $s_2 = 7, v_7 = 22.04, y_2 = 121.22, D'_2 = 77.95, Q'_2 = 32.0$
- Vehicle 3: $s_3 = 9, v_9 = 24.04, y_3 = 118.97, D'_3 = 90.87, Q'_3 = 41.0$

The coordinator agent solves the DRMP (19)-(24) considering the previous routes of the remaining operational vehicles $K' = \{2, 3\}$, Figure 2c, aiming at finding appropriate route modifications. The phase 2 of our distributed MAS-based solution approach obtains the solution illustrated in Figure 2d:

- Vehicle 2 was added tasks 10 and 8 into its new route $[7, 10, 8, 0]$ with a profit of 167.33.
- Vehicle 3 had no tasks added, maintaining its pending route $[9, 3, 1, 0]$ with an initially planned profit of 165.78.

As may be seen from this example, the proposed DIMASA architecture enables us to effectively manage unexpected events, such as a vehicle breakdown.

The proposed solution approach is designed to solve a class of problems that are known as NP-hard, which means that there is no efficient way to find the optimal solution in general, nor is there a termination criterion that ensures finding a solution. However, in practice, the DIMASA architecture works very well and can find the solutions in a relatively short amount of time, which is shown next in the functional experiments.

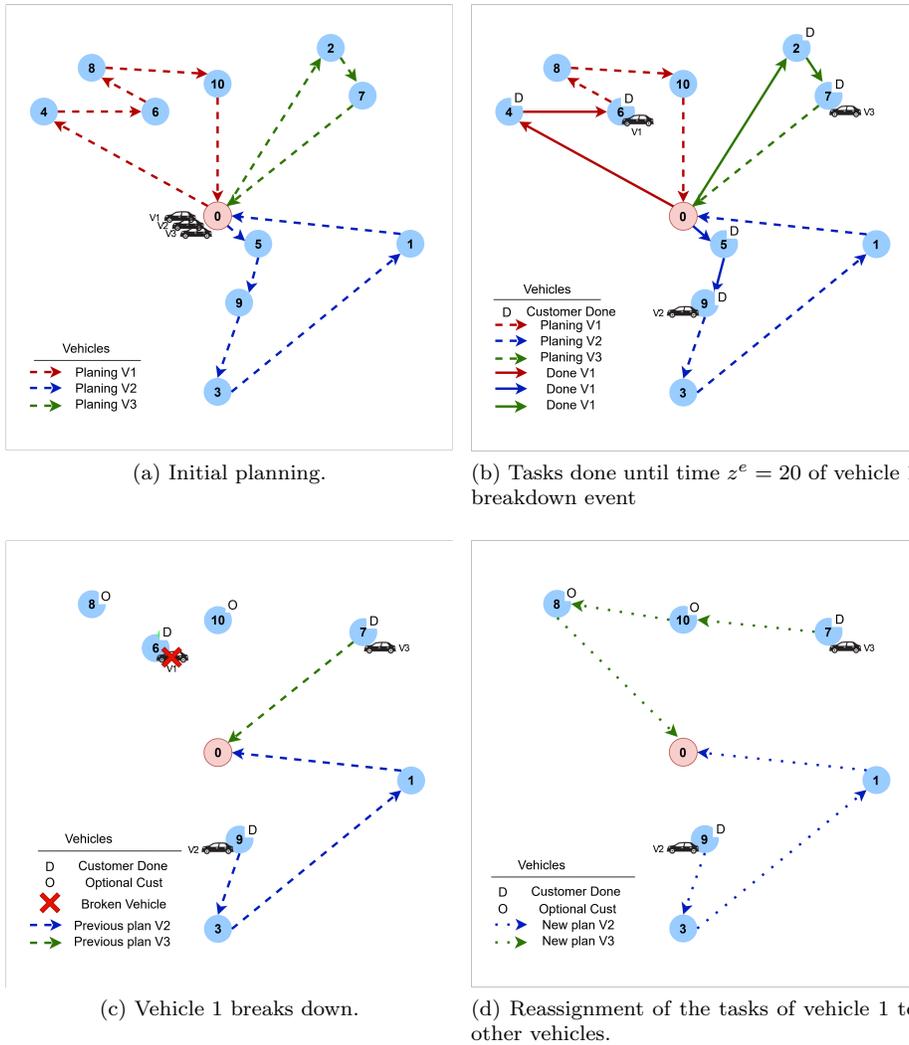


Figure 2: Example of the execution of the dynamic approach when vehicle 1 breaks down in the middle of its route.

5.2. Functional experiments

In this Section, we evaluate the performance of the proposed DIMASA architecture (in phase 1, SYSCA and VRP-FPTW Algorithm for the coordinator that interacts with the DRGV algorithm implemented in each of the vehicle agents, and in phase 2, the coordinator’s DSYSCA and the DVRP-FPTW Algorithm in interaction with the DRGV algorithm in each of the vehicle agents. We compare the performance of the two phases with the solution of the centralized model (Centr.) of the proposed DVRP-PFPTW problem.

The results of the centralized model are found by Gurobi 10.0.2 solver with gurobipy Python library with the time limit of one hour. We run the simulation experiments on an Intel Xeon Gold 6226R¹ virtualised cluster with 16 CPUs, 32 GB of RAM and clock frequency 2.9 GHz.

The proposed DVRP-PFPTW is a new problem that was not studied previously. Therefore, there are no relevant state-of-the-art benchmarks. Consequently, in this Section, we provide the strategy for creating them as well as a set of publicly accessible² benchmark instances together with a set of relevant Key Performance Indicators for the DVRP-FPTW problem.

Inspired by the generation of benchmark instances in [8, 45, 46], we consider a 2-dimensional Euclidean space of dimensions $[0, 100] \times [0, 100]$, with a single depot located in the centre of this space, i.e., at $(50, 50)$ and multiple customer vertices (customers) distributed uniformly randomly over the same. The distance matrix $\mathbf{d} = \{d_{ij} | (i, j) \in A\}$ is created based on the Euclidean distance between each pair of vertices in the given space. For simplicity, we assume that the cost and travel times matrix for each vehicle is equal to the distance matrix. The parameters related to customers include their coordinates, demand, revenue and time windows. Customer demand is uniformly randomly distributed between 1 and 10 and customer revenue between 10 and 100. The vehicle parameters include the number of vehicles, the cost, velocity, autonomy and capacity.

5.2.1. Benchmark instances for the static VRP-FPTW

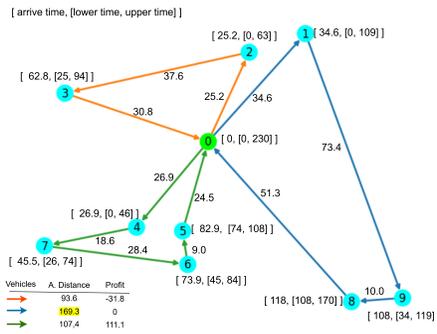
Feasibility and binding constraints. In the creation of benchmark instances, we ensure that the autonomy, capacity, and timing constraints are binding by specifically adjusting the values of time windows $[l_i, u_i]$ for each customer i as well as autonomy D_k and capacity Q_k for each vehicle k . Defining these values is still an open challenge for the capacitated VRP [46]. This is essential since if the problem is too constrained, it might not be feasible, while if it is too loose, the constraints that distinguish this problem from the general constrained VRP will not be considered. Thus, the parameters of the proposed instances are determined in such a manner that they strike a balance between being sufficiently restrictive to enforce the binding nature of associated constraints and

¹<https://www.intel.es/content/www/es/es/products/sku/199347/intel-xeon-gold-6226r-processor-22m-cache-2-90-ghz/specifications.html>

²<https://github.com/aitorls/DVRP-PFPTW-instances.git>



(a) A random graph in 2D space. Blue-customers, green-depot. Coordinates (x,y). (b) Partition of the graph depends on the vehicles.



(c) Optimal TSP solution in the partitioned space.

Figure 3: Example of generation of the parameters for the VRP-FPTW with 9 consumer vertices and 3 vehicles.

yet guaranteeing the presence of a feasible solution.

The procedure we follow is the following. First, the region of interest is partitioned into $|K|$ sections, where $|K|$ is the number of vehicles. The partition is made similarly to a cake-cutting approach, starting from the depot in the centre and adding $|K|$ equidistant rays where the angle between the rays is $360/|K|$ and the depot belongs to every created partition, as seen in Figure 3. Then, we compute the optimal route for visiting all the customer vertices from the depot 0 by a single vehicle in each of these sections by applying the travelling salesman problem (TSP). Note that these routes only take into account the distances, but not the vehicle autonomy and capacity nor the customer time windows. The information related to each of the $|K|$ found TSP routes leaving from and returning to the depot 0 includes a sequence of customer vertices, the time needed to visit each of them, the total distance travelled, and the route's accumulated customer demand. The route with the maximum accumulated

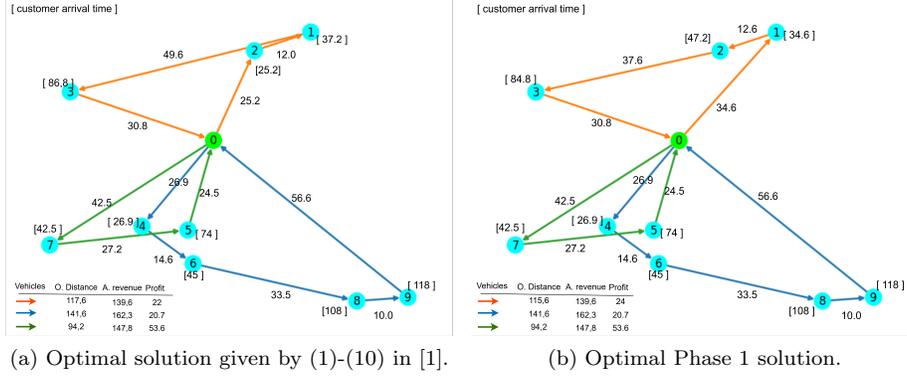


Figure 4: Examples of optimal solutions for a given instance with the Phase 1 algorithms.

demand among all routes defines the capacity and similarly, the route with the maximum distance among all routes defines autonomy for each vehicle. To set the time windows for each customer, we first determine the centre of their time window as the arrival time in the TSP route found previously. Next, we define the time window limits for each customer vertex in the same order as the one along each found TSP route. Specifically, the first visit time l_i for each customer $i \in \{1, \dots, |V| - 1\}$ is set as the centre time of the previous visited vertex and u_i is set as the centre time of the next vertex. In the case of the first customer vertex leaving from the depot, l_i is set to the departure time from the depot, and in the case of the last customer vertex i in the route before the depot, u_i is set to the arrival time at the depot. We set the first visit time l_0 of the depot to 0 while its last visit time u_0 is the maximum of the last visit times of any customer plus the time to return from this customer to the depot $u_0 = \max_{i \in N} \{u_i + t_{i0k}\}$, where vehicle k is the one that visits customer i .

In this way, we make sure that the problem has at least one feasible solution and that the autonomy, capacity, and timing constraints are binding. We give an example in Figure 3, with a randomly generated graph in 2-dimensional space in Figure 3a, the resulting partitioning of the space in Figure 3b, and the optimal TSP solution for such partitioned space in Figure 3c. Moreover, for the same instance in Figure 3, we give the optimal solution of (1)-(10) in [42], found by Gurobi solver in Figure 4a. Additionally, in Figure 4b, we provide the solution for that instance found by the proposed phase 1 algorithms.

The presented benchmark specification is used for the creation of the instances that are named SFPTW_N_K_X, where S stands for static, FPTW is the proposed kind of the VRP problem, N is the number of customers vertices 25, 50, 75 and 100, K is the number of vehicles 5, 10, 15 and 20 and X ranges from 1 to 10 representing an instance number. The number of vehicles is chosen to maintain the N/K ratio, that is, the vertices of average customers served with

a vehicle. The proposed instances are publicly accessible on GitHub ³.

5.2.2. Benchmark instances for the DVRP-FPTW

In the benchmark creation for the dynamic VRP-FPTW problem, we use the same structure as in the previous instances but extend it with dynamic events that happen throughout the route. The two possible events that we consider in the experiments are the appearance of new customers and vehicle breakdowns that can occur at any time $t \leq u_0$. When a vehicle breaks, we assume that it does not get repaired in the considered time horizon. New customers are defined by their coordinates (x, y) , demand q , revenue r and time windows $[l, u]$:

$$New_customer : < name, t, (x, y), q, r, [l, u] >,$$

while the vehicles that break down with :

$$Vehicle_breakdown : < k, t > .$$

Usually, the addition of new customers occurs in batches, as in real-world scenarios where customers often accumulate before being introduced into the system. In this paper, we carry out experiments only on the breakdown of a functional vehicle, because this case involves the removing of the associated customers' vertices from the mandatory customers set and adding them to the optional set. The benchmark instances for the DVRP-FPTW are named DFPTW_N_K_X, where N, K, and X represent the number of customers, vehicles and problem instances, respectively. These instances are similar to the static case, but they also include a list of predetermined events that take place during the execution of the simulation.

As presented in Section 3.2, the proposed DVRP-FPTW model considers mandatory customers that are part of the taken route, and optional customers that might be considered for inclusion in the route if and only if they bring benefit to a vehicle. Thus, if a new customer appears close to the maximum return time to the depot u_0 , the remaining vehicles may not have enough time to accommodate them. Similarly, if the capacity and/or autonomy of a vehicle is too limited, they may not be able to service this customer. To show the effectiveness of our Dynamic VRP-FPTW algorithm, we allocate the events of vehicle breakdown at the beginning of the execution plan. This allows for flexibility in the reassignment of customers to vehicles.

Specifically, we differentiate between scenarios where the best-off (highest profit) vehicle, the worst-off vehicle and a random vehicle suffers a breakdown. By examining these three different cases, we evaluate the performance and adaptability of our algorithm in different profit-based scenarios.

To validate that the proposed approach is able to reassign the optional customers while not worsening the vehicles' profits, within the time and autonomy

³<https://github.com/aitorls/DVRP-PFPTW-instances>

constraints, we compare our real-time dynamic algorithm with the exact solution, considering a one-hour computation time limit. The objective of this comparison is to evaluate the impact of having more time to reassign vehicles, where an hour of waiting to recalculate the route is not applicable in dynamic environments. This allows us to evaluate the effectiveness of our algorithm’s approach to react to unpredicted events, thus, achieving a dynamically adaptable real-time solution. We also compare our approach by increasing the revenue of the optional customers (extra revenue). With this experiment, we motivate vehicles to visit optional customers if autonomy and time window constraints allow it.

5.2.3. Experimental results

Next, we present the comparison of the phase 1 and phase 2 algorithms with the benchmark centralized models.

Comparison of the MAS-Phase 1 algorithms with the centralized model. The results obtained for the benchmark instances of the static VRP-FPTW are presented in Tables 2 and 3. In these tables, we show the outcomes achieved by the two models within a one-hour time limit. For each instance and method, we provide the instance name (Name), the worst-off vehicle profit among all the vehicles (Worst-off), the computation time in seconds (Time) or a start (*) if the one hour limit is reached, the best upper bound for the worst vehicle’s profit (UB) which is the value of the linear relaxation of the models, and the overall profit collected for all the vehicles (Overall). Optimal solutions, where the worst profit and the upper bound value are equal, are highlighted in bold.

In Table 2, we observe that for instances with 25 customers and 5 vehicles, the exact centralized model (Centr.) achieves the optimal solution for all 10 instances with an average computation time of 3.3 seconds, while our proposed Phase 1 algorithms achieve the optimal solution in 9 out of 10 instances with an average computation time of 0.8 seconds. In the instances with 50 customers and 10 vehicles, the centralized approach only obtains the optimal solution in 3 instances (SFPTW_50_10_[0,7,8]) after an average computation time of 1814 seconds. On the other hand, the DIMASA Phase 1 obtains the optimal solution in these same 3 instances, but in an average time of 8 seconds. For the rest of the instances with 50 customers, the centralized approach yields an average of 14.2% higher profit for the worst-off vehicle compared to our DIMASA Phase 1 solution. However, DIMASA finds the solution with an average computation time of 15.748 seconds, while the average computation time of the centralized approach is 3064 seconds. In 3 out of 10 instances with 75 customers and 15 vehicles, the centralized approach does not get any solution in one hour of computation time, Table 3. This is the case for instances SFPTW_75_15_1_[0,2,6]. Finally, when we increase the number of customers to 100, the centralized approach does not find any solution for these instances. In these cases, the centralized approach can only provide us the best bound obtained through linear relaxation.

The DIMASA Phase 1, on the other hand, always finds a solution in less than 2 minutes for instances with 75 customers with the average computation

Table 2: Static FPTW benchmark results

Name	Algorithm	Worst-off	UB	Overall	Time
SFPTW_25_5.0	Centr.	78.38	78.38	525.16	6.54
	Phase 1	78.38	—	534.43	0.42
SFPTW_25_5.1	Centr.	120.69	120.69	662.09	15.38
	Phase 1	118.42	—	718.72	0.84
SFPTW_25_5.2	Centr.	51.61	51.61	672.36	0.83
	Phase 1	51.61	—	677.58	0.60
SFPTW_25_5.3	Centr.	74.68	74.68	649.80	6.61
	Phase 1	74.68	—	633.42	1.18
SFPTW_25_5.4	Centr.	97.80	97.80	718.65	0.58
	Phase 1	97.80	—	713.80	0.49
SFPTW_25_5.5	Centr.	17.21	17.21	533.99	0.40
	Phase 1	17.21	—	550.16	0.86
SFPTW_25_5.6	Centr.	97.05	97.05	651.54	0.89
	Phase 1	97.05	—	651.57	1.68
SFPTW_25_5.7	Centr.	58.45	58.45	487.85	0.51
	Phase 1	58.45	—	466.95	0.65
SFPTW_25_5.8	Centr.	91.02	91.02	829.23	0.36
	Phase 1	91.02	—	829.23	0.76
SFPTW_25_5.9	Centr.	103.52	103.52	838.70	0.93
	Phase 1	103.52	—	840.28	0.70
SFPTW_50_10.0	Centr.	67.60	67.60	1273.24	2006.28
	Phase 1	67.60	—	1287.87	8.48
SFPTW_50_10.1	Centr.	151.26	161.54	1574.17	*
	Phase 1	137.50	—	1574.64	8.57
SFPTW_50_10.2	Centr.	141.98	147.38	1488.08	*
	Phase 1	127.05	—	1564.56	9.59
SFPTW_50_10.3	Centr.	94.24	129.82	1232.44	*
	Phase 1	55.60	—	1260.26	12.90
SFPTW_50_10.4	Centr.	142.29	175.51	1678.82	*
	Phase 1	133.54	—	1661.76	53.82
SFPTW_50_10.5	Centr.	129.63	147.05	1386.70	*
	Phase 1	111.65	—	1308.03	13.28
SFPTW_50_10.6	Centr.	112.04	166.62	1500.76	*
	Phase 1	83.08	—	1577.97	22.09
SFPTW_50_10.7	Centr.	24.12	24.12	1406.72	64.25
	Phase 1	24.12	—	1415.14	7.62
SFPTW_50_10.8	Centr.	53.02	53.02	1384.81	3372.55
	Phase 1	53.02	—	1319.68	8.24
SFPTW_50_10.9	Centr.	141.55	149.51	1577.79	*
	Phase 1	111.39	—	1595.87	12.89

Table 3: Static FPTW benchmark results

Name	Algorithm	Worst-off	UB	Overall	Time
SFPTW_75_15.0	Centr.	—	—	—	*
	Phase 1	102.98	169.89	2313.25	56.77
SFPTW_75_15.1	Centr.	174.17	—	2760.73	*
	Phase 1	162.77	203.91	2759.00	64.21
SFPTW_75_15.2	Centr.	—	—	—	*
	Phase 1	124.38	178.36	2398.53	47.34
SFPTW_75_15.3	Centr.	116.58	—	2083.51	*
	Phase 1	96.85	161.16	2089.09	72.39
SFPTW_75_15.4	Centr.	139.05	175.43	2339.88	*
	Phase 1	123.82	—	2432.46	31.08
SFPTW_75_15.5	Centr.	139.74	—	2434.89	*
	Phase 1	125.71	178.10	2375.72	51.61
SFPTW_75_15.6	Centr.	—	—	—	*
	Phase 1	106.91	183.88	2398.75	73.47
SFPTW_75_15.7	Centr.	66.24	—	2164.58	*
	Phase 1	62.00	170.01	2276.58	37.68
SFPTW_75_15.8	Centr.	137.24	—	2277.05	*
	Phase 1	124.06	169.61	2304.81	59.13
SFPTW_75_15.9	Centr.	145.35	199.14	2597.43	*
	Phase 1	145.90	—	2740.77	54.23
SFPTW_100_20.0	Phase 1	154.53	191.65	3456.50	233.09
SFPTW_100_20.1	Phase 1	158.34	218.35	3859.57	449.89
SFPTW_100_20.2	Phase 1	150.07	200.28	3503.05	609.41
SFPTW_100_20.3	Phase 1	125.13	182.87	3188.60	811.43
SFPTW_100_20.4	Phase 1	120.19	163.05	2961.78	204.06
SFPTW_100_20.5	Phase 1	80.48	206.47	3606.50	485.43
SFPTW_100_20.6	Phase 1	123.20	179.18	3327.81	101.37
SFPTW_100_20.7	Phase 1	132.61	200.05	3443.45	209.58
SFPTW_100_20.8	Phase 1	138.43	195.29	3376.28	152.66
SFPTW_100_20.9	Phase 1	148.77	193.61	3419.45	248.20

time of 55 seconds, and less than 15 minutes for instances with 100 customers, with the average computation time of 5.85 minutes. For the instances where the solution of the centralized algorithm is unknown, we use the best-bound value and determine that the theoretical gap between the obtained DIMASA Phase 1 solution and the value of the linear relaxation is within 30%. This is a theoretical bound that in practice might be significantly lower since, in general, the optimum value is not known until the problem is NP-Hard. To further improve this solution, the DIMASA Phase 1 may be introduced into a branch-and-price algorithm to find the optimal solution. However, the average gap between the DIMASA Phase 1 and the centralized method for the instances for which the centralized solution did find a solution is less than 15%.

In summary, the computation time of the proposed DIMASA Phase 1 is significantly better than the one of the centralized model, especially as the number of agents and/or tasks increases. Even more, the fleet as a whole on average, receives a larger profit from the DIMASA Phase 1 than from the centralized and exact VRP-FPTW model in Gurobi solver. Obtaining the exact solution with the centralized model is intractable and already for 75 customers and 5 vehicles, it does not find any solution in 1 hour in 3 out of 10 tested instances, while for 100 customers and 20 vehicles, it does not find a solution for any instance at all. The proposed DIMASA Phase 1 approach, on the other hand, solves all tested instances within less than 15 minutes efficiently with an average gap in relation to the exact centralized model (where it found a solution) of less than 15%.

DIMASA Phase 2 results. Tables 4, 5 and 6 show the results of the Dynamic VRP-FPTW problem with 100 customers and 20, 30 and 40 vehicles, respectively. For each experiment, we compare the DSYSCA solution, the centralized dynamic solution, and the solution with more revenues in the optional customers. We also report the information of the dynamic instance, such as the static instance name SFPTW_N_K_X where N, K and X are the number of customers, vehicles, and instance number, the broken vehicle (B. vehicle), the broken time (B. time), the finish expected time of the vehicle (exp. finish), the last time to return to the depot for all the vehicles (u_0), the current value of the worst-off vehicle (Worst-off), the number of customers of the broken vehicle that it planned to visit (Plan) and, lastly, the number of customers left without visiting due to vehicle breakdown (Remain). The DIMASA Phase 2 solution columns show the objective function value, that is profit of the worst-off vehicle (Worst-off), the number of customers that can be reassigned to another vehicle (Resign) and the time to obtain the solution, expressed in seconds (Time). The DIMASA Phase 2 (extra revenue) columns show that this approach increases the revenues of the optional customers. Finally, the last set of columns is the information about the centralized dynamic solution solving the constraints (6)-(18) on Gurobi, it shows the same three columns as before as the columns of DIMASA Phase 2.

The results show that the Phase 2 approach is very effective in handling dynamic vehicle breakdown scenarios when the broken vehicle is the worst-off vehicle in terms of profit. The Phase 2 approach can find a solution in less than

Table 4: Comparison of the results between the DIMASA Phase 2 approach and the exact centralized algorithm over the static instances SFPTW_100_20_X with 100 customers and 20 vehicles where X is the number of the instance.

X	B. vehicle	Dynamic Instance		Phase 2			Phase 2 (extra revenues)			Centralized dynamic solution						
		B. time	exp. time	% ₀	Worst-off	Plan	Remain	Worst-off	Reassign	Time	Worst-off	Reassign	Time			
0	vehicle_2	49.00	117.01	220	154.52	4	3	159.17	2	3.96	159.17	1	3.22	159.66	2	80.44
0	vehicle_12	28.18	117.80	220	154.52	5	4	156.51	1	18.46	154.53	1	12.92	163.27	3	*
0	vehicle_15	39.85	130.46	220	154.52	7	6	159.17	3	8.25	159.17	3	7.31	160.43	2	*
1	vehicle_8	19.92	118.14	210	158.34	5	4	167.05	4	15.84	167.05	4	14.15	192.04	4	*
1	vehicle_9	43.48	115.64	210	158.34	7	5	162.17	0	3.83	162.17	0	4.39	163.67	0	79.82
1	vehicle_5	34.41	132.73	210	158.34	6	5	158.34	2	8.85	158.34	1	7.08	192.04	4	*
2	vehicle_17	35.18	118.93	220	150.07	4	2	154.82	1	5.94	154.82	1	6.83	167.55	1	109.07
2	vehicle_5	24.08	97.46	220	150.07	7	6	150.07	1	12.41	150.07	5	13.17	175.47	5	*
2	vehicle_15	14.04	108.16	220	150.07	7	6	150.07	0	20.85	150.07	1	18.93	180.21	2	*
3	vehicle_9	49.07	127.87	210	125.13	4	2	137.35	1	2.06	137.35	0	2.17	146.50	1	41.18
3	vehicle_8	32.28	136.54	210	125.13	7	5	125.13	3	6.27	135.75	4	7.04	148.24	3	1178.10
3	vehicle_11	27.46	111.65	210	125.13	5	4	135.75	2	8.36	135.75	3	9.39	148.24	3	*
4	vehicle_19	31.76	127.81	210	120.19	5	4	121.74	1	13.11	121.74	2	12.05	127.30	1	*
4	vehicle_12	46.33	103.82	210	120.19	7	5	120.19	2	6.85	120.19	2	7.45	121.74	1	38.76
4	vehicle_13	55.00	136.36	210	120.19	5	4	120.19	0	2.54	120.19	2	2.91	121.74	1	57.66
5	vehicle_8	59.00	124.44	210	80.48	3	2	116.67	1	1.35	116.67	1	1.26	116.67	1	4.43
5	vehicle_18	27.76	113.93	210	80.48	8	6	80.48	0	11.42	80.48	0	10.93	80.48	1	81.48
5	vehicle_17	60.34	115.25	210	80.48	4	2	80.48	1	1.23	80.48	2	1.17	80.48	1	4.07
6	vehicle_3	60.11	123.80	200	123.20	5	4	123.99	2	1.54	127.50	2	1.37	127.50	2	7.97
6	vehicle_13	24.18	129.06	200	123.20	6	4	123.20	0	17.40	123.20	0	17.15	142.45	1	960.04
6	vehicle_9	42.06	133.34	200	123.20	7	6	123.20	2	6.38	123.20	3	7.41	123.20	2	109.51
7	vehicle_1	3.61	109.39	210	132.61	3	2	143.30	0	27.14	143.30	0	24.91	166.94	1	*
7	vehicle_12	34.80	138.33	210	132.61	6	4	132.61	0	10.18	132.61	0	9.54	156.51	0	*
7	vehicle_8	61.39	129.99	210	132.61	6	3	143.30	0	1	143.42	0	1.08	143.42	0	4.98
8	vehicle_7	46.59	127.08	220	140.92	6	3	142.25	2	4.02	142.25	2	2.29	146.32	2	41.46
8	vehicle_15	41.83	145.40	220	140.92	8	5	140.92	0	4.94	140.92	0	4.98	143.55	0	66.41
8	vehicle_16	38.28	105.75	220	140.92	5	4	142.04	1	7.19	142.04	2	6.60	148.79	1	656.04
9	vehicle_4	45.89	104.28	220	148.77	4	3	152.25	2	3.26	158.27	2	2.53	154.20	1	490.71
9	vehicle_12	16.12	98.04	220	148.77	5	4	148.77	0	12.88	148.77	0	14.98	152.39	0	*
9	vehicle_17	64.00	111.75	220	148.77	3	2	148.77	1	0.88	155.59	1	0.92	159	1	3.74
Mean		38.5	120.34	213	133.42	5.46	3.96	137.33	1.16	8.27	138.33	1.5	7.87	147	1.56	1460

Table 5: Comparison results between the Phase 2 approach and the exact centralized algorithm over the static instances SFPTW_100.30_X with 100 customers and 30 vehicles, where X is the number of the instance.

X	B. vehicle	B. time	Dynamic Instance			Phase 2			Phase 2 (extra revenues)			Centralized dynamic solution					
			Exp.	Finish	% ₀	Worst-off	Plan	Remain	Worst-off	Reassign	Time	Worst-off	Reassign	Time			
0	vehicle.25	39.00	77.28	220	2	71.44	2	1	73.65	1	10.74	75.11	1	8.38	78.06	1	*
0	vehicle.21	17.46	136.32	220	7	71.44	7	6	75.11	2	17.14	73.65	6	18.25	83.17	3	*
0	vehicle.26	12.21	32.17	220	3	71.44	3	2	71.44	0	23.94	71.44	0	23.15	71.44	0	*
1	vehicle.23	40.20	120.06	210	4	80.94	4	3	83.91	1	4.52	84.73	1	5.35	89.70	1	1554.43
1	vehicle.17	45.61	98.97	210	4	80.94	4	2	80.94	2	2.39	84.73	2	3.23	89.70	2	704.76
1	vehicle.29	61.72	133.81	210	3	80.94	3	2	80.94	1	1.15	80.94	0	0.91	80.94	1	2.34
2	vehicle.25	16.03	44.43	220	2	79.56	2	1	80.72	1	46.03	80.72	1	39.45	86.89	1	*
2	vehicle.17	48.47	148.80	220	5	79.56	5	3	80.85	2	2.64	80.85	2	2.48	81.65	1	305.75
2	vehicle.7	52.25	106.00	220	4	79.56	4	2	80.85	2	1.71	80.85	2	1.69	85.94	2	50.82
3	vehicle.15	26.57	101.54	200	3	70.43	3	2	74.21	1	12.13	74.21	2	12.60	75.63	2	*
3	vehicle.30	37.06	123.41	200	8	70.43	8	5	72.56	1	4.63	70.43	2	3.78	74.94	2	210.32
3	vehicle.27	60.81	129.94	200	3	70.43	3	2	70.43	1	0.77	70.43	0	0.92	74.94	1	2.08
4	vehicle.10	50.00	119.14	200	3	59.57	3	2	60.16	1	2.80	61.20	2	2.76	67.85	2	736.85
4	vehicle.26	52.54	108.03	200	4	59.57	4	2	59.57	2	2.08	59.57	2	1.99	60.16	0	36.22
4	vehicle.4	31.14	76.21	200	4	59.57	4	3	59.57	2	15.53	59.57	2	14.21	59.57	0	*
5	vehicle.2	52.40	137.43	210	3	65.57	3	2	87.47	0	1.91	87.47	0	2.00	87.54	0	33.97
5	vehicle.21	55.00	114.70	210	4	65.57	4	2	65.57	2	1.26	65.57	2	1.36	65.57	2	4.99
5	vehicle.3	51.24	112.99	200	3	65.57	3	2	65.57	2	13.14	65.57	2	11.87	65.57	2	767.18
6	vehicle.17	52.55	128.67	200	5	62.01	5	4	66.46	0	1.60	66.46	2	1.49	66.46	0	7.98
6	vehicle.13	22.47	122.23	200	3	62.01	3	2	62.01	1	1.43	62.01	3	1.48	66.46	1	6.30
6	vehicle.9	60.99	127.08	210	5	79.92	5	3	82.41	3	1.02	83.86	1	1.59	74.51	2	*
7	vehicle.23	45.18	98.99	210	4	79.92	4	2	79.92	2	3.87	79.92	1	1.08	82.41	2	3.01
7	vehicle.15	17.20	40.85	210	2	79.92	2	1	79.92	2	17.77	79.92	1	18.67	79.92	0	*
8	vehicle.6	33.24	97.44	220	2	64.56	2	1	64.82	0	16.35	64.82	0	10.73	75.56	1	*
8	vehicle.21	47.43	102.98	220	2	64.56	2	1	64.56	0	6.91	64.82	2	9.12	67.59	1	595.65
8	vehicle.23	41.05	102.29	220	5	64.56	5	4	64.82	2	17.90	64.56	0	3.11	72.58	2	*
9	vehicle.4	57.14	132.74	210	4	70.26	4	3	71.30	1	1.28	71.30	1	1.31	71.30	1	4.40
9	vehicle.7	20.62	124.38	210	3	70.26	3	2	70.26	2	19.27	70.26	2	15.06	74.77	2	*
9	vehicle.2	37.07	87.63	210	4	70.26	4	2	70.26	1	7.78	70.26	2	8.21	70.26	1	311.25
Mean		40.48	105.38	210	2.4	70.42	3.76	2.4	72.07	1.2	9.12	72.24	1.5	8.12	75.45	1.23	1625.3

Table 6: Comparison results between the Phase 2 approach and the exact centralized algorithm over the static instances instances SFPTW_100_40_X with 100 customers and 40 vehicles, where X is the number of the instance.

X	B. vehicle	B. time	Dynamic Instance			u ₀	Phase 2			Phase 2 (extra revenues)			Centralized dynamic solution			
			Exp.	Finish	t ₀		Worst-off	Plan	Remain	Worst-off	Reassign	Time	Worst-off	Reassign	Time	Worst-off
0	vehicle_16	42.76	109.52	220	39.48	2	1	42.29	0	9.11	42.29	0	6.02	42.29	0	*
0	vehicle_30	39.85	111.92	220	39.48	5	4	39.48	1	8.87	39.48	1	6.14	37.65	0	*
0	vehicle_24	3.00	37.72	220	39.48	2	1	39.48	0	34.24	39.48	0	23.55	41.72	0	*
1	vehicle_27	85.00	99.42	210	46.16	1	0	53.35	0	0.22	53.35	0	0.21	53.35	0	0.17
1	vehicle_1	47.16	93.59	210	46.16	4	2	46.16	2	6.60	46.16	2	4.55	49.89	2	501.70
1	vehicle_37	61.00	89.10	210	46.16	2	1	46.16	1	2.43	46.16	1	1.50	47.92	1	2.55
2	vehicle_35	21.02	42.05	220	40.95	1	0	42.17	0	41.63	42.17	0	35.98	40.80	0	*
2	vehicle_10	8.25	51.40	220	40.95	3	2	42.17	2	48.78	42.17	2	34.50	40.95	0	*
2	vehicle_5	65.00	145.56	220	40.95	3	2	42.17	0	0.69	42.17	1	0.69	42.17	0	1.43
3	vehicle_1	31.78	91.69	200	37.04	2	1	37.80	1	11.61	38.29	1	10.44	37.8	0	*
3	vehicle_7	46.44	125.68	200	37.04	5	3	38.29	1	6.32	38.29	1	3.31	38.71	1	448.38
3	vehicle_9	32.70	102.01	200	37.04	3	2	37.04	1	11.87	37.04	1	6.76	38.71	1	*
4	vehicle_11	17.72	36.11	200	22.88	2	1	24.75	0	22.72	27.21	1	16.52	24.75	0	*
4	vehicle_5	55.00	59.47	200	22.88	1	0	23.99	0	1.29	23.99	0	1.33	23.99	0	5.00
4	vehicle_2	52.00	115.53	200	22.88	3	2	24.75	2	1.99	24.75	2	1.89	24.75	2	52.67
5	vehicle_40	33.62	86.26	210	46.13	2	1	46.70	0	14.61	49.65	1	8.63	49.74	1	*
5	vehicle_5	39.20	129.69	210	46.13	3	2	46.13	0	7.46	46.13	0	4.47	48.83	2	1946.09
5	vehicle_25	17.46	105.70	210	46.13	3	2	31.07	2	27.94	46.70	2	23.42	51.00	2	2758.92
6	vehicle_36	38.47	106.35	190	30.64	3	2	31.07	0	5.32	31.07	0	10.06	40.03	1	*
6	vehicle_5	61.31	125.50	190	30.64	5	3	31.08	0	0.65	31.08	1	0.94	31.08	1	1.07
6	vehicle_35	26.17	88.02	190	30.64	2	1	31.07	1	20.60	31.07	1	15.05	40.03	1	*
7	vehicle_9	57.01	136.91	210	36.09	3	2	38.59	0	2.13	38.59	0	1.78	42.47	0	3.04
7	vehicle_29	55.07	120.69	210	36.09	3	2	36.09	1	1.63	36.09	1	2.03	36.09	1	3.91
7	vehicle_36	43.38	99.69	210	36.09	2	1	36.09	1	5.92	36.09	1	3.55	36.09	1	1319.40
8	vehicle_13	22.83	45.65	220	35.35	1	0	36.85	0	31.01	36.85	0	16.83	36.85	0	*
8	vehicle_5	37.58	101.00	220	35.35	4	3	37.34	1	11.17	37.34	1	10.78	45.07	2	357.91
8	vehicle_28	10.44	66.24	220	35.35	2	1	35.35	0	20.23	35.35	0	31.16	35.35	0	*
9	vehicle_37	29.21	109.70	200	39.48	2	1	41.02	1	12.16	41.02	1	14.88	44.96	1	*
9	vehicle_29	27.02	85.91	200	39.48	3	2	41.02	1	18.63	41.02	1	18.95	43.61	2	*
9	vehicle_20	27.00	40.00	200	39.48	1	0	41.02	0	13.71	41.02	0	17.41	43.61	0	*
Mean		37.81	91.93	208	37.42	2.6	1.5	38.52	0.6	13.384	38.73	0.76	11.11	40.34	0.73	2177.71

14 seconds on average, and the profit of the worst-off vehicle is only 5.9 % lower than the centralized solution in one hour. Moreover, the Phase 2 approach can reassign 38 % of the optional customers that could be reassigned, compared with 44 % of the centralized solution.

We present the results for different numbers of vehicles in Tables 4, 5 and 6. For each number of vehicles, we report the error of the profit of the worst-off vehicle, the percentage of optional customers that are reassigned, and the time to obtain the solution. We can see that the error of the profit decreases as the number of vehicles increases, from 7 % with 20 vehicles to 4.6 % with 30 vehicles and 4.7 % with 40 vehicles. The percentage of optional customers that are reassigned also decreases as the number of vehicles increases, from 30 % with 20 vehicles to 50 % with 30 vehicles and 40 % with 40 vehicles. The time to obtain the solution increases as the number of vehicles increases, from 8.27 seconds with 20 vehicles to 9.12 seconds with 30 vehicles and 13.384 seconds with 40 vehicles.

The results obtained by artificially increasing the rewards of the optional customers are very illustrative, if we compare the value of the worst-off vehicle, we can see that it is very similar, only 0.47 % higher. This means that increasing the revenue of the optional customers does not significantly improve the revenue of the worst-off vehicle. This is because, although the worst-off vehicle of the previous iteration can obtain an optional customer and more revenue, there are not enough optional customers for every vehicle, and the profits of the next worst-off vehicles are very similar. On the other hand, when we look at the average value of reassigned customers, it is more significant. With more revenues in the optional customers, the algorithm can reassign 48 % of them. This also implies that the other 52% of optional customers are not reachable due to the constraints of autonomy, time or capacity. Thus, compared to the original algorithm without extra revenue, the Phase 2 approach can reassign 38% of the optional customers, which means that it can reassign 80% of the possible unassigned optional customers ($38/48 = 0.79$).

The main advantage of the Phase 2 approach is that it can quickly adapt to the dynamic vehicle breakdown situation and generate a feasible solution in real-time. This is a significant advantage over the centralized approaches that have to start the planning from scratch. The Phase 2 approach can make timely decisions and reduce the impact of unforeseen events on the overall performance of the fleet.

6. Conclusions and future work

We proposed a novel mathematical model for the DVRP-FPTW that maximizes the profit of each vehicle in the fleet in a non-decreasing order of individual profits along the routes. We also proposed a multi-agent architecture for the static and dynamic VRP-FPTW, composed of vehicle agents and a coordinator agent. The coordinator agent uses the SYSCA and DSYSCA algorithms to optimize the egalitarian social welfare of the whole fleet. Our dynamic so-

lution approach extends the static VRP-FPTW solution to handle real-time contingencies, such as changes in customer requests or vehicle breakdowns.

We implemented our model and architecture in a distributed way, using a multi-agent system that decompose the computation among the agents. This makes our approach more private, flexible, robust, fast, and fair than a centralized one. This approach is ideal for cooperative vehicle-sharing scenarios, accommodating intrinsically decentralized private information while avoiding reliance on a unique decision maker.

We test our approach on a set of benchmark instances that we create for the static and dynamic VRP-FPTW problem, and we show that it can achieve high-quality solutions. As a future work, we plan to extend our model and architecture to deal with open fleets, where vehicles can join or leave the system dynamically. This would require some modifications in our algorithms and communication protocols, which we will explore in our future research.

The communication between the coordinator and vehicle agents in the proposed distributed multi-agent architecture was performed by a combination of synchronous and asynchronous communication, thus speeding up the generation of routes between vehicles. This can be implemented through message queues and has the additional advantage that the system will become more robust with regard to possible breakdowns in the communication channel or other contingencies in the exchange of asynchronous messages. However, the solution approach is heavily dependent on the quality of communication for the synchronous messages. The requirements for communication and tackling technical issues in the real world are not the subject of this paper and will be addressed in future work.

In the presented solution approach, we focused on mutually exclusive vehicle owners with limited funds and each one owning a single vehicle. The proposed decomposed VRP-FPTW model and its related distributed multi-agent solution approach is of concern for real-world dynamically changing agriculture cooperative fleet scenarios focusing on both fairness and efficiency among vehicle owners and farmers, where each vehicle disposes of private information that it should not share with the other vehicles and where the decision-making time for the generation of routes is relatively short. We showed through simulation that the proposed MAS-based distributed solution approach is computationally efficient, it is scalable and suited for large cooperative fleets, responding in close-to real time to unpredicted contingencies on the terrain. In future work we will consider also the cases where an owner owns multiple vehicles or the ownerships mutually overlap among different owners. Also, we will focus on the receding horizon control, so that the most disadvantaged vehicles in one period are prioritised in the future ones and thus ensure both fairness and efficiency of the fleet at long run.

Also, we will extend the proposed DVRP-FPTW model with vehicle tools for the agriculture robotics setting where a robot or tractor requires a detachable tool for task performance.

Acknowledgments

This work was supported by the VAE grant: TED2021-131295B-C33 funded by MCIN/AEI/10.13039/501100011033 and by the European Union's "Next Generation EU/PRTR", by grant COSASS: PID2021-123673OB-C32 funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe", by the AGROBOTS Project funded by the Community of Madrid, Spain and by the AGROBOTIX Project funded by the University Rey Juan Carlos.

References

- [1] López Sánchez A, Lujak M, Billhardt H, Semet F. Vehicle Routing Problem with Fair Profits and Time Windows (VRP-FPTW). In: Proceedings of the 2023 IEEE Conference on Systems, Man, and Cybernetics (IEEE SMC 2023). vol. In Press. IEEE; 2023. p. 1-7.
- [2] Aksen D, Aras N. Customer selection and profit maximization in vehicle routing problems. In: Operations research proceedings 2005. Springer; 2006. p. 37-42.
- [3] Archetti C, Speranza MG, Vigo D. Chapter 10: Vehicle routing problems with profits. In: Vehicle routing: Problems, methods, and applications, second edition. SIAM; 2014. p. 273-97.
- [4] Guajardo M, Rönnqvist M. A review on cost allocation methods in collaborative transportation. *International transactions in operational research*. 2016;23(3):371-92.
- [5] Gansterer M, Hartl RF. Collaborative vehicle routing: a survey. *European Journal of Operational Research*. 2018;268(1):1-12.
- [6] Gansterer M, Hartl RF. Shared resources in collaborative vehicle routing. *Top*. 2020;28:1-20.
- [7] Lujak M, Sklar E, Semet F. Agriculture fleet vehicle routing: A decentralised and dynamic problem. *AI Communications*. 2021;34(1):55-71.
- [8] Soriano A, Gansterer M, Hartl RF. The multi-depot vehicle routing problem with profit fairness. *International Journal of Production Economics*. 2023;255:108669.
- [9] Liu S, Papageorgiou LG. Fair profit distribution in multi-echelon supply chains via transfer prices. *Omega*. 2018;80:77-94.
- [10] Lee D, Ahn J. Vehicle Routing Problem with Vector Profits (VRPVP) with Max-Min Criterion. *arXiv preprint arXiv:171010550*. 2017.
- [11] Stavropoulou F, Repoussis PP, Tarantilis CD. The vehicle routing problem with profits and consistency constraints. *European Journal of Operational Research*. 2019;274(1):340-56.

- [12] Mancini S, Gansterer M, Hartl RF. The collaborative consistent vehicle routing problem with workload balance. *European journal of operational research*. 2021;293(3):955-65.
- [13] Rodríguez-Martín I, Salazar-González JJ, Yaman H. The periodic vehicle routing problem with driver consistency. *European Journal of Operational Research*. 2019;273(2):575-84.
- [14] Pillac V, Gendreau M, Guéret C, Medaglia AL. A review of dynamic vehicle routing problems. *European Journal of Operational Research*. 2013;225(1):1-11.
- [15] Psaraftis HN, Wen M, Kontovas CA. Dynamic vehicle routing problems: Three decades and counting. *Networks*. 2016;67(1):3-31.
- [16] Monroy-Licht M, Amaya CA, Langevin A, Rousseau LM. The rescheduling arc routing problem. *International Transactions in Operational Research*. 2017;24(6):1325-46.
- [17] Ulmer MW, Heilig L, Voß S. On the value and challenge of real-time information in dynamic dispatching of service vehicles. *Business & Information Systems Engineering*. 2017;59:161-71.
- [18] Ulmer MW, Streng S. Same-day delivery with pickup stations and autonomous vehicles. *Computers & Operations Research*. 2019;108:1-19.
- [19] Steever Z, Karwan M, Murray C. Dynamic courier routing for a food delivery service. *Computers & Operations Research*. 2019;107:173-88.
- [20] Ulmer MW, Goodson JC, Mattfeld DC, Hennig M. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*. 2019;53(1):185-202.
- [21] Fonseca-Galindo JC, de Castro Surita G, Neto JM, de Castro CL, Lemos AP. A multi-agent system for solving the dynamic capacitated vehicle routing problem with stochastic customers using trajectory data mining. *Expert Systems with Applications*. 2022;195:116602.
- [22] Alipour MM, Emami H, Abdolhosseinzadeh M. A MAS approach for vehicle routing problem. *Neural Computing and Applications*. 2022:1-25.
- [23] Bono G, Dibangoye JS, Simonin O, Matignon L, Pereyron F. Solving multi-agent routing problems using deep attention mechanisms. *IEEE Transactions on Intelligent Transportation Systems*. 2020;22(12):7804-13.
- [24] Barbucha D. An improved agent-based approach to the dynamic vehicle routing problem. In: *Intelligent Decision Technologies 2016: Proceedings of the 8th KES International Conference on Intelligent Decision Technologies (KES-IDT 2016)–Part I*. Springer; 2016. p. 361-70.

- [25] Barbucha D. VNS-based Multi-agent approach to the dynamic vehicle routing problem. In: Computational Collective Intelligence: 11th International Conference, ICCCI 2019, Hendaye, France, September 4–6, 2019, Proceedings, Part I 11. Springer; 2019. p. 556-65.
- [26] Lujak M, Giordani S, Omicini A, Ossowski S. Decentralizing coordination in open vehicle fleets for scalable and dynamic task allocation. *Complexity*. 2020;2020:1-21.
- [27] Los J, Schulte F, Gansterer M, Hartl RF, Spaan MT, Negenborn RR. Large-scale collaborative vehicle routing. *Annals of Operations Research*. 2022:1-33.
- [28] Los J, Schulte F, Spaan MT, Negenborn RR. An Auction-Based Multi-Agent System for the Pickup and Delivery Problem with Autonomous Vehicles and Alternative Locations. In: Dynamics in Logistics: Proceedings of the 8th International Conference LDIC 2022, Bremen, Germany. Springer; 2022. p. 244-60.
- [29] Bertsimas D, Brown DB, Caramanis C. Theory and applications of robust optimization. *SIAM review*. 2011;53(3):464-501.
- [30] Agra A, Christiansen M, Figueiredo R, Hvattum LM, Poss M, Requejo C. The robust vehicle routing problem with time windows. *Computers & operations research*. 2013;40(3):856-66.
- [31] Duan J, He Z, Yen GG. Robust multiobjective optimization for vehicle routing problem with time windows. *IEEE Transactions on Cybernetics*. 2021;52(8):8300-14.
- [32] Qi M, Xiong W, Zhou Q, Hua S. Robust periodic vehicle routing problem with service time uncertainty. In: 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE; 2018. p. 1431-5.
- [33] Guo Y, Cheng J, Ji J. Robust Dynamic Vehicle Routing Optimization with Time Windows. In: Advances in Swarm Intelligence: 7th International Conference, ICSI 2016, Bali, Indonesia, June 25-30, 2016, Proceedings, Part II 7. Springer; 2016. p. 28-36.
- [34] Seyyedhasani H, Dvorak JS. Dynamic rerouting of a fleet of vehicles in agricultural operations through a Dynamic Multiple Depot Vehicle Routing Problem representation. *biosystems engineering*. 2018;171:63-77.
- [35] López Sánchez A, Lujak M, Semet F, Billhardt H. On balancing fairness and efficiency in routing of cooperative vehicle fleets. In: CEUR Proceedings, 12th Int. Workshop on ATT 2022 co-located with IJCAI-ECAI 2022. vol. 3173; 2022. .

- [36] López Sánchez A, Lujak M, Semet F, Billhardt H. How to achieve fair and efficient cooperative vehicle routing? *AI Communications*. 2023;Pre-press(Pre-press):1-23.
- [37] Lübbecke ME, Desrosiers J. Selected topics in column generation. *Operations research*. 2005;53(6):1007-23.
- [38] Pinto T, Alves C, Mansi R, Valério de Carvalho J. Solving the multiscenario max-min knapsack problem exactly with column generation and branch-and-bound. *Mathematical Problems in Engineering*. 2015;2015.
- [39] Dror M. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*. 1994;42(5):977-8.
- [40] Basso S, Ceselli A. Asynchronous column generation. In: *2017 Proceedings of the Nineteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM; 2017. p. 197-206.
- [41] Basso S, Ceselli A. Distributed asynchronous column generation. *Computers & Operations Research*. 2022;146:105894.
- [42] Bennett KP, Demiriz A, Shawe-Taylor J. A column generation algorithm for boosting. In: *Icml*. Citeseer; 2000. p. 65-72.
- [43] Irnich S, Villeneuve D. The shortest-path problem with resource constraints and k-cycle elimination for $k \geq 3$. *INFORMS Journal on Computing*. 2006;18(3):391-406.
- [44] Feillet D, Dejax P, Gendreau M, Gueguen C. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal*. 2004;44(3):216-29.
- [45] Solomon MM. *Vehicle routing and scheduling with time window constraints: Models and algorithms*; 1984.
- [46] Uchoa E, Pecin D, Pessoa A, Poggi M, Vidal T, Subramanian A. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*. 2017;257(3):845-58.