



OPEN

# Solving a real-world package delivery routing problem using quantum annealers

Eneko Osaba<sup>1</sup>✉, Esther Villar-Rodríguez<sup>1</sup> & Antón Asla<sup>2</sup>

Research focused on the conjunction between quantum computing and routing problems has been very prolific in recent years. Most of the works revolve around classical problems such as the Traveling Salesman Problem or the Vehicle Routing Problem. The real-world applicability of these problems is dependent on the objectives and constraints considered. Anyway, it is undeniable that it is often difficult to translate complex requirements into these classical formulations. The main objective of this research is to present a solving scheme for dealing with realistic instances while maintaining all the characteristics and restrictions of the original real-world problem. Thus, a quantum-classical strategy has been developed, coined Q4RPD, that considers a set of real constraints such as a heterogeneous fleet of vehicles, priority deliveries, and capacities characterized by two values: weight and dimensions of the packages. Q4RPD resorts to the Leap Constrained Quadratic Model Hybrid Solver of D-Wave. To demonstrate the application of Q4RPD, an experimentation composed of six different instances has been conducted, aiming to serve as illustrative examples.

**Keywords** Routing problem, Package delivery, Logistics, Optimization, Quantum computing, Quantum annealer, D-Wave

Routing problems are extensively studied in the fields of transportation and optimization<sup>1–3</sup>. The interest in this particular topic stems from two main factors: their high computational complexity (often categorized as NP-Hard<sup>4</sup>), which makes them challenging to address<sup>5</sup>; and their broad applicability scenarios<sup>6</sup>, entailing that groundbreaking improvements in the creation of effective routing algorithms have positive effects on society and industry.

For the computational complexity, it is crucial to emphasize that, for even relatively modest instances of routing problems, current computers struggle to run brute force methods. As a result, throughout the past few decades, numerous time-efficient solving schemes have been proposed, such as exact methods<sup>7</sup>, heuristics<sup>8</sup>, and metaheuristics<sup>9</sup>, with the last ones being the most commonly used. Furthermore, other interesting and advanced paradigms, such as Reinforcement Learning<sup>10</sup> or Deep Learning<sup>11</sup>, have also been explored to cope with challenging routing problems.

Most solvers developed so far have been designed to be run on conventional computing resources. Quantum computing (QC<sup>12</sup>) has nevertheless become a promising alternative to these traditional devices for handling optimization and routing problems. Even though it is not a mature technology, QC has drawn much attention from the scientific community as it offers researchers and practitioners a revolutionary paradigm to solve different types of practical optimization problems<sup>13,14</sup>. Specifically, a large range of problems coming from domains such as healthcare<sup>15</sup>, economics<sup>16</sup>, industry<sup>17</sup>, energy<sup>18</sup>, or logistics<sup>19</sup> have benefited from QC lately.

The research on routing problems solved by QC approaches has been very prolific in recent years, with the Traveling Salesman Problem (TSP<sup>20</sup>) and Vehicle Routing Problem (VRP<sup>21</sup>) being the most studied cases. Specially interesting for this analysis is the extensive review published by Osaba et. al.<sup>19</sup>, stating that this new paradigm had motivated 53 research publications up to 2022. As concluded by the authors of that study, “*it is noticeable that the TSP engages most of the researchers (60,37% - 32 out of 53 papers), while the VRP amounts to the 25,52% of the contributions (13 out of 53). The rest of the papers deal with other routing problems, such as the Shortest Path Problem or the Hamiltonian Cycle*”. As of 2022, TSP<sup>22–25</sup> and VRP<sup>26–29</sup> keep maintaining a similar trend, accounting for the vast majority of scientific production.

The state-of-the-art might be classified according to their technical goals into two main branches. On the one hand, most of the works in the field pursue objectives such as unveiling the potential of quantum technologies or checking the efficiency of a particular method. This kind of work uses problems such as the TSP or VRP for

<sup>1</sup>TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain. <sup>2</sup>Serikat-Consultoría y Servicios Tecnológicos, 48009 Bilbao, Spain. ✉email: eneko.osaba@tecnalia.com

benchmarking purposes. On the other hand, other studies present advances in the application of QC to real-world-oriented routing problems, analyzing aspects such as traffic congestion<sup>30</sup>. This second category of research focuses on getting the most out of the current NISQ-era devices by implementing efficient and advanced hybrid resolution methods<sup>31</sup>. As for the latter, the study of Weinberg et al. in 2023<sup>32</sup> is probably the closest to solving a real routing problem using QC. The authors elaborate on the multi-truck vehicle routing for supply chain logistics. Being unable to solve the problem using a fully embedded approach, the authors propose an algorithm that iteratively assigns routes to trucks. This methodology allows the authors of that work to consider restrictions such as pick-up and drop-off demands and restricted driving windows.

In this context and aligned with this second category, this research proposes a novel solving scheme capable of providing a good, if not the best, solution to a real-world problem by making intelligent use of the capabilities offered by current quantum devices. To do so, the scheme, named *Quantum for Real Package Delivery* (Q4RPD):

- Focuses on solving a real-world routing problem defined by a Spanish company specializing in transport and logistics, named Ertransit. To the best of our knowledge, such a use case with delivery priorities, heterogeneous fleets, and two-dimensional descriptions of items has never been addressed in the literature from a quantum perspective. We, the authors, are confident that modeling realistic routing problems will push the limits of current research in QC and the advancement of the scientific community.
- Combines both quantum and classical computing in a hybrid method where:
  - Classical computing controls the general workflow of Q4RPD in charge of splitting the problem into affordable sub-problems, providing the resources required to set up and do the computation, ensuring correct restrictions and preferences handling, and composing the final solution. All these steps are deeply described in Sect. [Solving scheme and fundamentals](#).
  - QC is applied to the calculation of each route, i.e., the trajectory associated with a truck, or a section of a route, via the Leap Constrained Quadratic Model (CQM) Hybrid Solver (LeapCQMHybrid<sup>33</sup>) of D-Wave. This hybrid method solves problems formulated as CQM, which refers to a mathematical model defined by integer, real, and binary variables; linear, quadratic, inequality, and equality constraints; and a quadratic objective function<sup>33</sup>. All these characteristics contribute to a more user-friendly and comprehensive framework in comparison to the native Quadratic Unconstrained Binary Optimization (QUBO) formulation of most of the Quantum Processing Units (QPUs).
  - Demonstrates its application through an experimentation with six different instances aiming to serve as illustrative examples. Paraphrasing the words of Quetschlich et al.<sup>34</sup>, “*the development of quantum computing applications at the moment considers mostly toy-size problem instances*”. This same situation is highlighted by Weinberg et al.<sup>32</sup> who claim that “*there is a growing body of literature that tests quantum algorithms on miniaturized versions of problems that arise in an operations research setting*”. In this regard, the iterative nature of Q4RPD has allowed the addressing of instances close to those handled by Ertransit in day-to-day operations, both in terms of complexity and size. The rest of the article is organized as follows: in the following section, the [Problem definition](#) is introduced. After that, in [Solving scheme and fundamentals](#), the characteristics of the method implemented are described. After that, in [Single routing problem: mathematical formulation](#), the formulation of the problem addressed by the quantum device is presented. In [Experimental results](#), the applicability of the implemented system is demonstrated employing a set of instances as input. [Conclusions and future work](#) closes this paper by highlighting the main conclusions drawn from the results obtained in the experimentation and the planned future work.

## Problem definition

The routing problem addressed in this research can be defined as follows: given a set of last-mile deliveries to complete within the day (whose fulfillment in the daily time-span is feasible), a depot, and a fleet of heterogeneous and available owned and rental trucks, the **main objective** is to calculate a set of routes that satisfy all of the predetermined demands while minimizing the total costs. Exceptionally, a customer may have more than one order.

For this purpose, a **route** is defined as

*a trajectory associated with a single truck (driven by a designated single driver) that completes a group of deliveries and must start and end at the depot.*

Furthermore, in this work, the total cost of a solution is the total distance of the planned routes and the prices associated with the use of the truck. In relation to the latter, if the truck is owned by Ertransit the cost is 0. Otherwise, the use of rental vehicles incurs a cost for the service provided. Additionally, the problem is subject to the following **constraints**:

- *R1*: the capacity of each truck is measured in terms of weight (*kgs*) and dimension (*cm*<sup>3</sup>) which limits the load to be carried on each route. At the same time, the deliveries to be distributed are also defined by these parameters. We can say, thus, that the representation of the capacities is *two-dimensional*.
- *R2*: some of the packages must be delivered within a certain time frame (priorities). These deliveries are labeled with a Top-Priority (TP) tag. These priorities are similar to the time windows frequently-used in other research works<sup>35</sup>, where the lower limit of the window is 0, and the upper limit corresponds to the maximum time to carry out the delivery.
- *R3*: the duration of a route must be less than the driver's working day (which is the same for all the available workers).

In this research, and for the sake of simplicity, distance and time of travel share the same value. Lastly, the following ***business preferences*** are defined, drawn from past real-world situations faced by Ertransit that became, over the years, the operating protocol. Note that the preference order in the list relates to their relevance (if two preferences clash, the one with the higher ranking must prevail):

- *P1*: a vehicle can only complete one route, meaning that the trucks cannot be reused on that day, whether or not this decision has an impact on cost.
- *P2*: the use of owned trucks must be prioritized, even if this implies a higher overall cost in terms of distance. The motivation for this preference is twofold: on the one hand, to avoid the transactions associated with renting a vehicle, and on the other hand, to minimize the possibility of having an accident with a non-company truck.
- *P3*: preferential treatment must be given to solutions that use fewer vehicles, even if this implies a higher overall cost in terms of the distance. The purpose of this preference is to have as many trucks as possible in the depot in order to handle potential incidents or unplanned extra-routine situations. In light of this, this problem has been named *2-Dimensional and Heterogeneous Package Delivery with Priorities* (2DH-PDP).

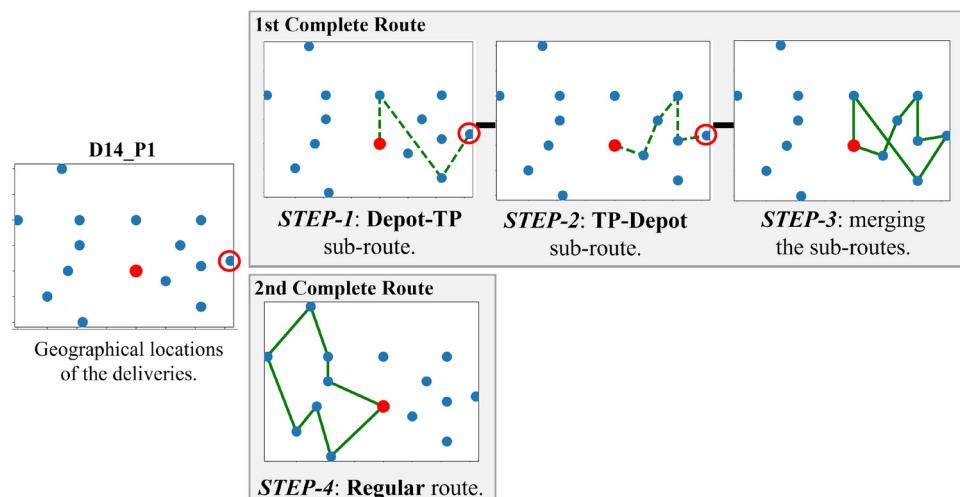
### Solving scheme and fundamentals

Given the aforementioned constraints and preferences, the resolution method implements a route search for each truck, i.e., it selects for a given truck the packages to be delivered and their delivery order. To efficiently deal with the temporal constraint *R2* of the priority items, the concept of sub-route is introduced. A sub-route is a section of a truck's path that will eventually belong to a complete regular route. Working with sub-routes allows us to adapt the maximum travel time depending on whether there is a priority constraint *R2* to resolve or not, in which case the time limit is imposed by constraint *R3*.

- Having said this, the different kinds of trajectories that may be calculated according to their origin and destination are the following: *Regular route*: this is a route completed by a single truck that starts and ends at the depot. This type of routes might be the result of a single route search process or of the chaining of several of the hereunder sub-routes. Two *regular routes* are depicted in STEP-3 and STEP-4 of Fig. 1.
- *Depot-TPsub-route*: this is a sub-route that starts at the depot and finishes at the location of a TP delivery. The main requirement for this kind of sub-route is to arrive at the destination prior to the scheduled time set by the priority item. An example of a *Depot-TP* sub-route is represented in STEP-1 of Fig. 1.
- *TP-TPsub-route*: in this type of sub-route, both the origin and destination correspond to the locations of different TP deliveries. In this case, the time at which the subroute ends must be equal to or less than the time imposed by the priority package of the destination.

*TP-Depot sub-route*: this is a sub-route which starts in a TP delivery location and ends in the depot. The STEP-2 of Fig. 1 is a instance of this *TP-Depot* sub-route. The Q4RPD solving scheme is based on an iterative process that, at each iteration, (1) picks out a truck and (2) calculates one trajectory (i.e. a regular route or a sub-route). Figure 2 represents the general workflow of Q4RPD. Right after loading the use case data but before starting the iterative process, Q4RPD begins by applying a pair of sorting heuristics:

- **Vehicle ordering heuristic**: To satisfy preferences *P2* and *P3* introduced in **Problem definition**, vehicles are ordered and hence served by the following criterion: owned trucks, sorted by their global capacity, first; and rental trucks, ordered by their capacity, second.



**Fig. 1.** A graphical step-by-step resolution of an instance composed of 14 different deliveries (D14\_P1), being one of them a TP (surrounded by a red circle).

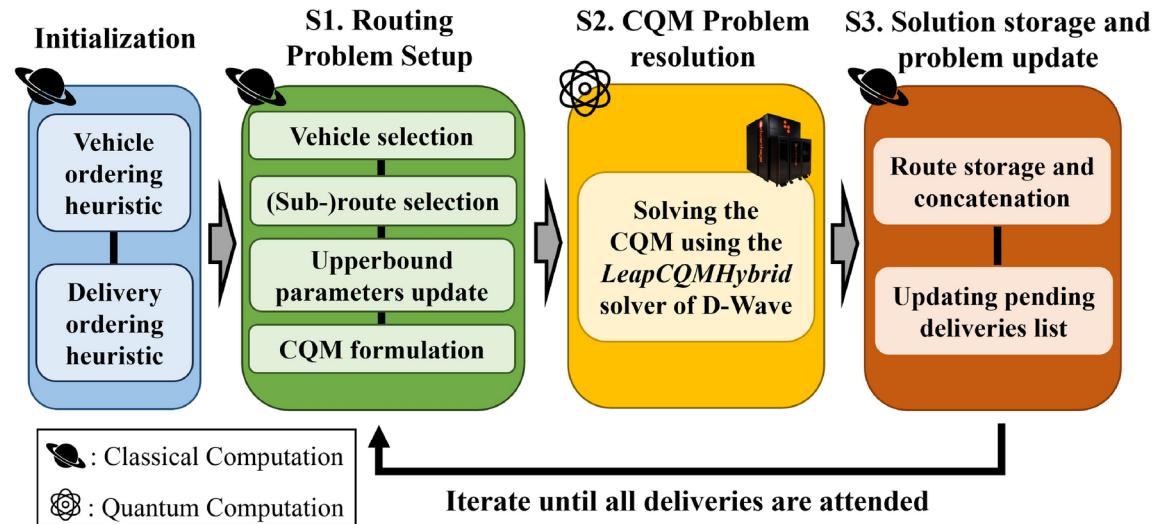


Fig. 2. General workflow of Q4RPD.

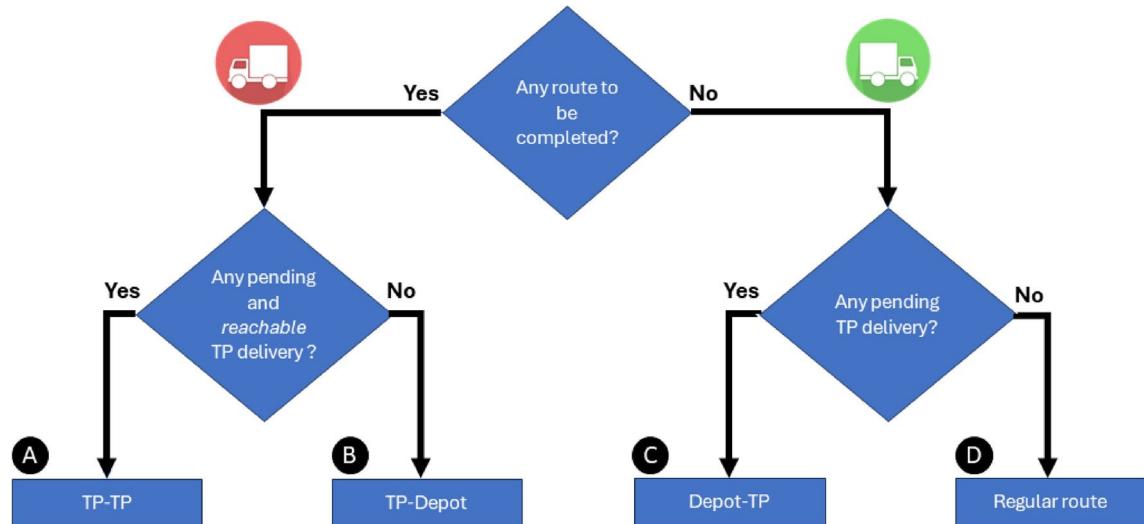


Fig. 3. (Sub-)route selection procedure with the 4 scenarios.

**Delivery ordering heuristic:** the hard constraint  $R2$  related to the deadline of the priority items, listed in the [Problem definition](#) section is addressed by forcing the treatment of the restrictive deliveries first, i.e., when truck capacities and available times are at their maximum. When a TP delivery is dequeued, this item becomes the destination of a sub-route now limited by the timing restriction of this delivery. Note, therefore, that starting the Q4RPD by handling the TP deliveries does not mean that these packages will be the first to be delivered. From this initialization onwards, an iterative solving scheme is in charge of partially solving the problem via a three-step procedure until the complete set of deliveries is satisfied:

**S1. Routing problem setup.** This first step, which is executed using classical computing, is devoted to preparing the problem instance for being subsequently solved by the quantum device. This setup consists of the following methods:

- **Vehicle selection.** Q4RPD picks the first truck in the list of available vehicles to calculate its most optimized route. It is important to clarify that the vehicle list definitively removes a truck only if it has been assigned a regular route, i.e., from depot to depot. This entails that if there is a truck whose route has not been completed yet, it will always be the first one on the list, thus prioritizing the completion of half-done routes over the start of a new one. At the same time, this procedure contributes to the rule stated by preference  $P3$ .
- **(Sub-)route selection:** At this point, four different scenarios, summarized in Fig.3, may arise:

- (A) If the selected truck had not completed its route, i.e., it was located at the TP point of a previously computed sub-route, and there are still prioritized deliveries to serve, Q4RPD checks if further TP deliveries are still *reachable* by this vehicle. A TP delivery is *reachable* if:
- The active truck's capacity is enough to serve the new demand, i.e.,  $R1$  is satisfied.
  - And the active truck can arrive on time to both this intermediate destination and the depot, and the driver's working day will not be exceeded when completing the whole route, i.e., the  $R2$  and  $R3$  constraints remain satisfiable after accepting this new sub-route.

If a TP delivery were *reachable*, a *TP-TP* sub-route would be calculated. Otherwise, a *TP-Depot* trajectory would be the objective to complete its route (B in Fig. 3).

- (B) If the selected truck had not completed its route but there were no TP items yet to be served, a *TP-Depot* sub-route to complete the route would be built.
- (C) If the truck was in the depot and there were still TP deliveries to serve, Q4RPD would take the first element out of the prioritized deliveries queue to calculate a *Depot-TP* sub-route.
- (D) If the truck was in the depot and there were no TP deliveries to serve, a *regular* route would be calculated.

It should be noted that, in the end, each complete route would be a regular and cyclic path composed of either a single *regular* route or a set of sub-routes.

- **Upper bound parameters update:** the upper bound parameters (summarized in Table 1) cannot be fixed in the problem initialization but must be set at each iteration according to the type of the trajectory.
- **Maximum duration assignment:** the  $rt$  of the new (sub-)route is calculated, embracing this criterion:
  - A For a *TP-TP* sub-route, with *reachability* already confirmed,  $rt$  is calculated as the difference between the destination's TP scheduled time and the time already spent since the beginning of the route, i.e., from the depot.
  - B For a *TP-Depot* route,  $rt$  is the difference between the driver's working day and the time already spent since the beginning of the route.
  - C For a *Depot-TP* route,  $rt$  is equal to the TP scheduled time.
  - D For a *regular* route,  $rt$  is equal to the driver's working day.
- **Usable truck capacity:** if the route of the active truck is still to be completed, capacities  $W$  and  $D$  must be updated by deducting the load already assigned to the truck in previous sub-routes from the vehicle's tare and dimension specification.
- **CQM problem formulation:** since the employed LeapCQMHybrid library solves problems modeled as CQM, the last step of this stage is to build a CQM following the formulation described in the next section ([Single routing problem: mathematical formulation](#)).

**S2. CQM problem resolution:** after the CQM is formulated, this second step consists of the resolution of this CQM by the D-Wave's LeapCQMHybrid solver, which resorts to a quantum computer to process the query. As pointed out before, this CQM could model any of the routes or sub-routes above defined.

**S3. Partial solutions storage:** this last step, executed using classical computing, is devoted to updating the whole problem. Two procedures engage in this phase:

- Route storage and concatenation. The goal of this step is to store the complete routes of every truck. If the trajectory that has just been solved corresponds to a sub-route, it is appended to the previous sub-route of the truck, if any. In these cases, through concatenation, a regular route is finally composed once the destination of the trajectory is the depot. Accordingly, the truck with a complete route will be removed from the vehicle list.
- Update the list of pending deliveries, removing all the packages already dropped in the last solved (sub-)route.

### Single routing problem: mathematical formulation

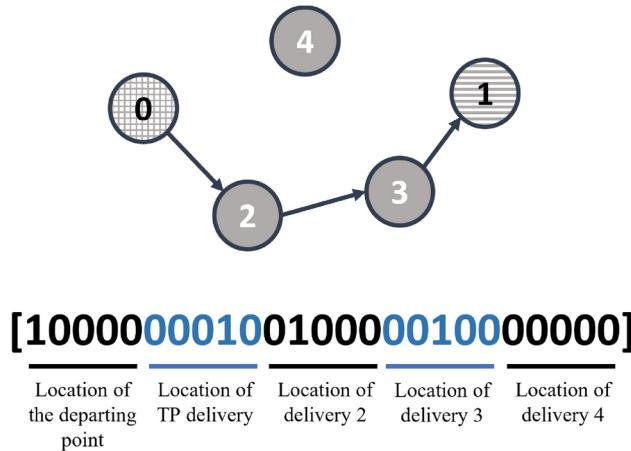
This section introduces the mathematical formulation of the problem addressed by the LeapCQMHybrid, defined after the routing problem setup where the type of route to calculate and the vehicle were determined by the *Vehicle ordering heuristic* (Step S1 in Fig. 2). As explained before and accordingly, the problem faced by the quantum device at each iteration is not the complete 2DH-PDP, but the calculation of a single (sub-)route of any

Upper bound parameters	
$rt$	The maximum possible duration of a (sub-)route.
$W, D$	The permissible maximum weight and dimension in the truck assigned to the (sub-)route.

**Table 1.** Upper bound parameters used in the formulation.

Upper bound parameters	
$rt$	The maximum possible duration of a (sub-)route.
$W, D$	The permissible maximum weight and dimension in the truck assigned to the (sub-)route.
Static parameters	
$P$	Set of deliveries still to be met.
$M$	Number of deliveries (cardinality of $P$ ).
$w_i, d_i$	Weight and dimension of delivery $i \in P$ .
$t_i$	The time limit for delivery $i$ .
$c_{i,j}, d_{i,j}$	Travel time and distance associated with going from the location of delivery $i$ to the location of delivery $j$ .
Variables	
$x_{i,p}$	Binary variable that represents if the location of delivery $i$ is visited at position $p$ of the (sub-)route, with $p \in [0, M]$ .

**Table 2.** The complete set of parameters and variables used in our formulation.



**Fig. 4.** A SRP instance composed of five delivery locations, and a tentative solution. Delivery location with  $ID=0$  represents the starting point of the route. Delivery location with  $ID=1$  represents the destination, that is, the delivery location that must be reached before a certain restricted time.

of the types listed in [Solving scheme and fundamentals](#). For this reason, this sub-problem has been coined the *Single Routing Problem* (SRP), whose variables and parameters are displayed in Table 2.

### Codification and variables

The binary codification used for the SRP is the one known as *node-based*, often used in studies of quantum computing applied to routing problems<sup>32,36,37</sup>. A solution to the SRP is represented as a set  $X = \{X_0, \dots, X_M\}$  of lists, in which each  $X_i$  is associated with a single delivery  $i$ , where  $M$  is the total number of deliveries and  $0$  represents the origin location. Furthermore,  $X_i = \{x_{i,0}, \dots, x_{i,M}\}$ , where  $x_{i,p}$  is a binary variable used for representing the position of the delivery  $i$  along the trajectory. Thus,  $x_{i,p}$  is 1 if the location of the delivery  $i$  is visited in position  $p$  of the route and 0 otherwise. A visual example comprising five delivery locations is represented in Fig. 4.

So, the challenge when solving SRP is to find the most appropriate values for variables in  $X$  in order to build the route that best optimizes the established objectives (later described in [Objectives](#)).

### Problem initialization

As part of the problem modeling, a slight initialization procedure is carried out, which is comprised of two steps:

- Firstly, the two first  $X$  components, i.e.,  $X_0$  and  $X_1$  are reserved for the preset origin and destination of the (sub-)route, respectively.
- Secondly,  $x_{0,0}$  is set to 1 to ensure that the sub-route begins at the origin. In contrast,  $X_1$ , i.e., the set of variables identifying the position of the destination in the (sub-)route, cannot be fixed in advance since, at this stage, we do not know how many deliveries will eventually be included in the (sub-)route and, therefore, what the last position of the route will be.

### Objectives

The SRP is an optimization problem that must be solved by minimizing a cost function defined by two different objectives. By multiplying each of these objectives by an appropriate weight, the user can determine the relevance

given to each of them. Hence,  $\min \sum_{i=1}^2 \omega_i o_i$  represents the cost function, being  $\omega_i$  the weight given to objective  $o_i$ . For this specific research, and as a result of an empirical study carried out in a laboratory environment,  $\omega_1=1$  and  $\omega_2=2$  have been applied. Therefore:

- $o_1$ : minimizing the total distance of the route, which can be mathematically formulated as follows:

$$o_1 = \min \sum_{i=0}^M \sum_{j=0}^M d_{i,j} x_{i,p} x_{j,p+1} \quad \forall p \in \{0, \dots, M-1\}, i \neq j. \quad (1)$$

- $o_2$ : visiting the destination location as later as possible in the route, that is, maximizing the number of deliveries carried out before reaching the final destination. This objective is materialized by maximizing the value of  $p$  in  $x_{1,p} = 1$ . This objective is formulated as follows:

$$o_2 = \min \sum_{p=0}^M \left( -x_{1,p} - \sum_{p'=p}^M x_{1,p'} \right). \quad (2)$$

- Finally, certain restrictions apply to the aforementioned objectives, which are essential to solving a real-world-oriented problem. The next subsection defines the entire pool of constraints ([Problem constraints](#)).

### Problem constraints

The above-described objectives are subjected to a set of seven different restrictions, which are:

- **Delivery-consistency:** a delivery may not be on the route or, if it is, at most once.

$$\sum_{p=0}^M x_{i,p} \leq 1 \quad \forall i \in \{0, \dots, M\}. \quad (3)$$

This constraint differs from the classical formulations of the TSP<sup>38</sup>, VRP<sup>39</sup>, and related problems in the sense that a route does not necessarily have to serve all deliveries.

- **Location-consistency:** each time slot of a truck schedule can be assigned to a single order at most.

$$\sum_{i=0}^M x_{i,p} \leq 1 \quad \forall p \in \{0, \dots, M\}. \quad (4)$$

- **Delivery-consecutiveness:** a correct calculation of objectives  $o_1$  and  $o_2$  requires that the deliveries served in the (sub-) route be consecutive, which means that there must be as many consecutive zeros from right to left in each  $X_0, X_1, \dots, X_M$  as there are surplus positions  $p$ .

$$\sum_{i=0}^M x_{i,p} - \sum_{i=0}^M x_{i,p+1} \geq 0 \quad \forall p \in \{0, \dots, M-1\}. \quad (5)$$

- **Destination-inclusion:** the destination must be in the route.

$$\sum_{p=0}^M x_{1,p} = 1. \quad (6)$$

- **Time-restriction:** the duration of the route must not be longer than  $rt$ , that is, the maximum possible duration of the route.

$$\sum_{i=0}^M \sum_{j=0}^M c_{i,j} x_{i,p} x_{j,p+1} \leq rt \quad \forall p \in \{0, \dots, M-1\}, i \neq j. \quad (7)$$

Note that, as aforementioned, in practice  $c_{i,j} = d_{i,j}$ .

- **Weight-restriction:** the maximum allowed weight  $W$  for the truck must not be surpassed.

$$\sum_{p=0}^M x_{i,p} w_i \leq W \quad \forall i \in \{0, \dots, M\}. \quad (8)$$

- **Dimension-restriction:** the maximum permitted dimension  $D$  for the route must not be exceeded.

$$\sum_{p=0}^M x_{i,p} d_i \leq D \quad \forall i \in \{0, \dots, M\}. \quad (9)$$

## Experimental results

This section is devoted to analyzing the performance of Q4RPD. First, in [Quantum solver details](#), some details on the LeapCQMHybrid are provided. After that, in [Benchmark description](#), the data employed in the experimentation is introduced. This section finishes in [Performance analysis](#) by showing the obtained results and analyzing the performance of Q4RPD.

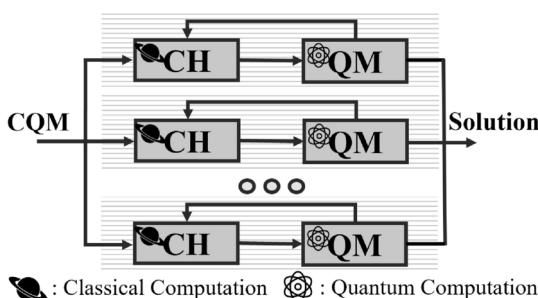
### Quantum solver details

The hybrid solver used in this paper is the CQM model of LeapCQMHybrid, implemented by D-Wave System. In a nutshell, this method is part of D-Wave's *hybrid solver service* (HSS<sup>40</sup>), which can be described as a portfolio of hybrid solvers implemented by D-Wave Systems. Methods included in HSS balance both classical and quantum computation to solve problems not fitting in QPUs. At the time this paper is being written, HSS accommodates three techniques to deal with three problem types: Binary Quadratic Models, Discrete Quadratic Models, and, lastly, CQMs.

The LeapCQMHybrid workflow, depicted in Fig. 5, applies parallel processing to the search for solutions to eventually return the best solution found among the pool of threads. Each thread consists of:

- a classical heuristic module (CH) with cutting-edge heuristic methods primarily aimed at solving the problem.
- and a quantum module (QM) in charge of guiding HM into promising areas of the search space and improving already-existing solutions by means of *quantum queries*.

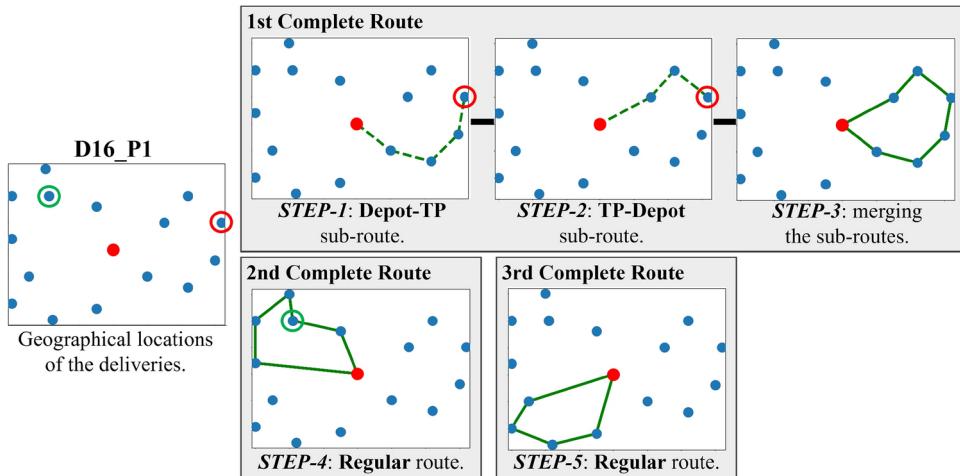
In this research, the *quantum queries* have been run on the Advantage\_system6.4 architecture, which is the most recent at the time that this work was written. This computer features 5616 qubits, arranged in a Pegasus topology.



**Figure 5.** General scheme of LeapCQMHybrid solver. CH = classical heuristic module. QM = quantum module.

Instance	Fleet description	#Variables	#Constraints
D14_P1	2o,3r	420	111
D16_P1	0o,4r	631	157
D14_P2	2o,3r	398	124
D21_P2	3o,2r	834	209
D21_P0	2o,2r	702	138
D29_P0	0o,4r	1602	232

**Table 3.** Characteristics of the six instances. The fleet is represented using Xo,Yr format, where X and Y are the number of owned and rental trucks, respectively.



**Fig. 6.** Step-by-step resolution of D16\_P1, consisting of non-priority 15 deliveries and one TP (surrounded by a red circle). Two non-priority demands belong to the same client (surrounded by a green circle), which are served by the same truck.

Regarding the parameterization of the LeapCQMHybrid solver, the only configurable parameter time\_limit was set to the default value for each instance.

The LeapCQMHybrid solver is proprietary, therefore, further information about the quantum subroutines or the number of qubits required to face an instance of SRP is not available to the general public. For additional details on this method, we refer interested readers to the D-Wave report<sup>41</sup> and to recently published works focused on real-world applications of the LeapCQMHybrid<sup>42–44</sup>.

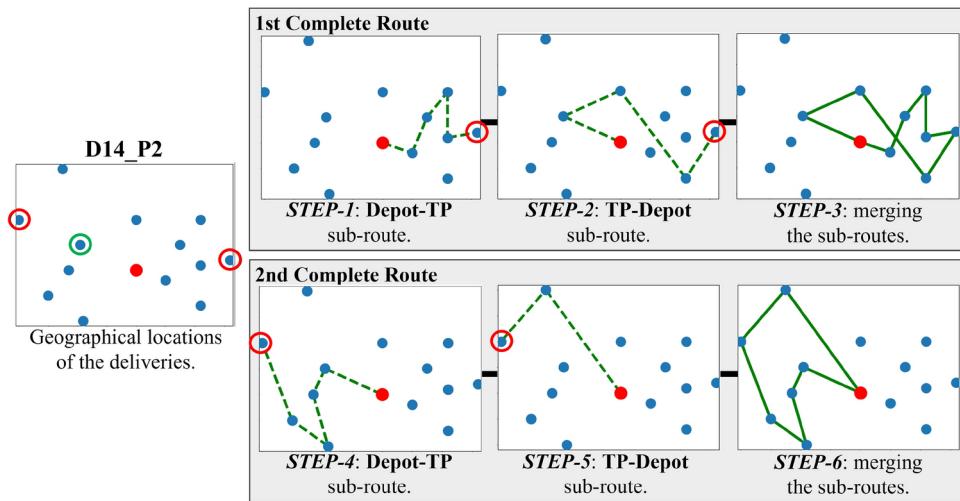
### Benchmark description

The benchmark of Q4RPD is composed of six different instances whose characteristics are summarized in Table 3. Because the problem addressed in this paper has been designed ad hoc for this research, it is not possible to find standard datasets to work with. This is why the six use cases employed have been generated specifically for this study, under the guidance of Ertransit.

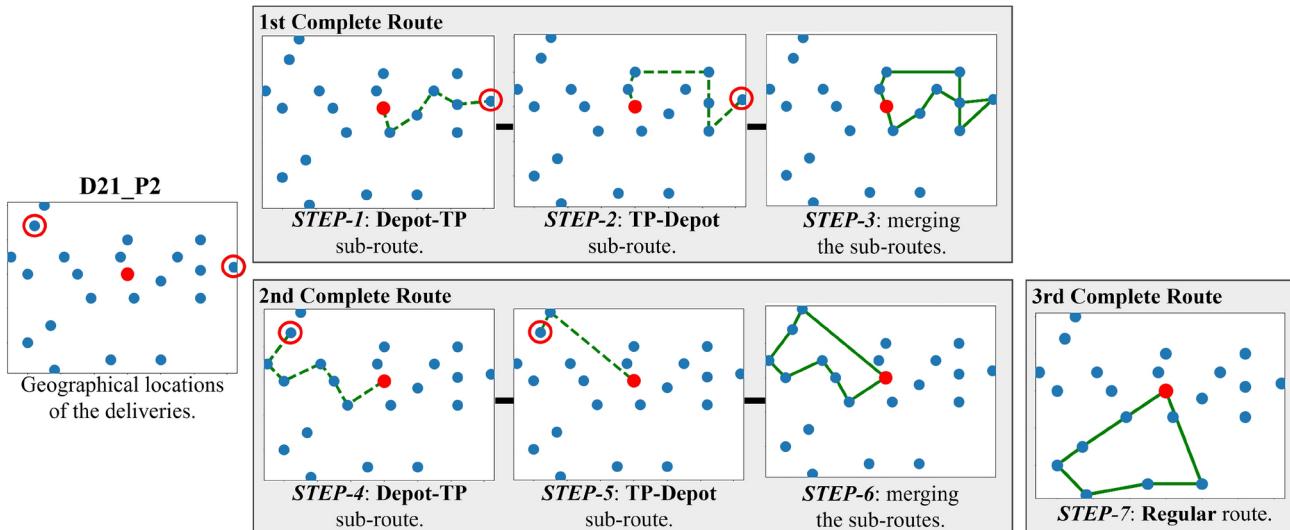
Each instance is named DX\_PY, where X represents the size of the problem in terms of the number of deliveries and Y the amount of TP deliveries. The following information per instance is also provided in Table 3: the number of owned and rental trucks of the fleet of vehicles; and the number of variables and constraints involved in the global problem formulation.

The characteristics of each instance and the main motivations for their selection for benchmarking purposes are as follows:

- D14\_P1: this is the smallest scenario which is composed of 13 non-priority and one priority delivery. The solution to D14\_P1 can be seen in Fig. 1.
- D16\_P1: the main difference between this case and the previous one is the use of a drastically reduced working day (90 minutes) to push Q4RPD into a narrower feasible solution space since in D14\_P1 the length of the routes is mostly restricted by the vehicle's capacity. The resolution process of this instance is shown in Fig. 6.
- D14\_P2 and D21\_P2: the main characteristic of these scenarios is the existence of two priority deliveries that are not *reachable* on the same route. In addition, D14\_P2 has two orders from the same customer, which must be served by two distinct trucks to satisfy the restrictions while minimizing the objective function. D21\_P2, in turn, is the most demanding instance in terms of quantum resources employed. The solutions to D14\_P2 and D21\_P2 are depicted in Figs. 7 and 8, respectively.



**Fig. 7.** Step-by-step resolution of D14\_P2, composed of 12 non-priority deliveries and two TP (surrounded by a red circle). Also, two non-priority demands correspond to the same client (surrounded by a green circle), which must be served separately.

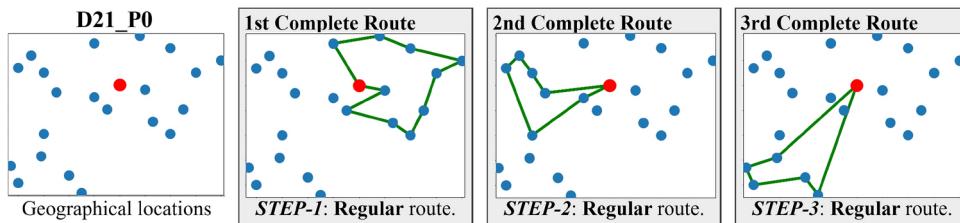


**Fig. 8.** Step-by-step resolution of D21\_P2, with 19 non-priority deliveries and two TP (surrounded by a red circle).

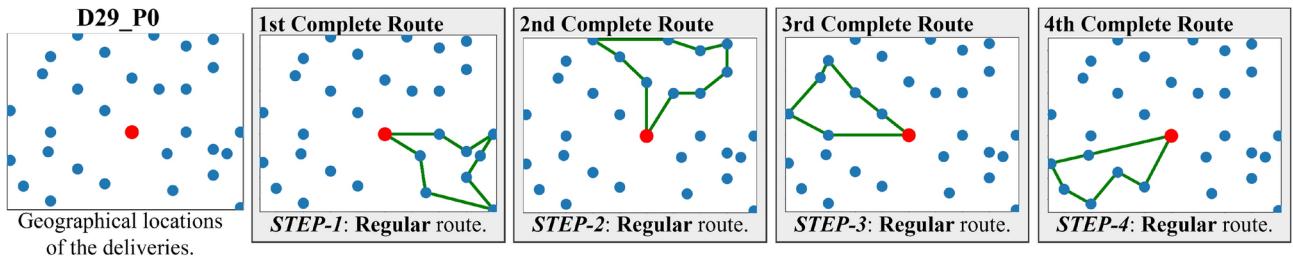
- D21\_P0 and D29\_P0: these instances are characterized by being fully composed of non-priority deliveries, and their solutions are depicted in Figs. 9 and 10. Regarding D29\_P0, this instance unveils the potential of Q4RPD to solve large and complex scenarios. In fact, this is the use case with the most paths to be calculated.

### Performance analysis

Concerning the Q4RPD performance, it is important to mention that stage **S1** (described in [Problem definition](#)) involved in the pre-selection of the (sub-)route and truck reduces the search space and therefore may affect the global performance of the solver. This means that by attending to the **business preferences** indicated by Ertransit, the heuristics may hinder or even prevent the optimal solution from being reached. Nevertheless, this stage must be computed in advance to fix the upper bound parameters that complete the mathematical formulation that eventually sets in motion the (sub-)route construction. In addition, this classical processing forces the application of the preferences *P1* (i.e., one truck for each complete route) and *P2* (i.e., owned trucks have been prioritized to minimize the cost associated with the rental of trucks). Lastly, and after conducting this experimentation, we have found that *P3* is not always covered by the currently proposed scheme. Even though this setback only occurs in rare cases, future versions of the scheme will introduce more intelligent resource allocation schemes to ensure the fulfillment of this preference.



**Fig. 9.** Step-by-step resolution of D21\_P0, composed of non-priority 21 deliveries.



**Fig. 10.** Step-by-step resolution of D29\_P0, composed of no-priority 29 deliveries.

Instance		Problem solution			Validation			Objectives of the SRP	
	#Full routes	#(Sub-)routes	R1	R2	R3	$\sum o_1$	$o_2$ optimality		
D14_P1	Fig. 1	2	[1,1,0,1]	✓	✓	✓	210.43 (+ 4.3%)	✓	
D16_P1	Fig. 6	3	[2,1,0,1]	✓	✓	✓	223.74 (+ 0.0%)	✓	
D14_P2	Fig. 7	2	[0,2,0,2]	✓	✓	✓	245.30 (+ 6.3%)	✓	
D21_P2	Fig. 8	3	[1,2,0,2]	✓	✓	✓	309.99 (+ 3.5%)	✓	
D21_P0	Fig. 9	3	[3,0,0,0]	✓	—	✓	381.46 (+ 0.0%)	✓	
D29_P0	Fig. 10	4	[4,0,0,0]	✓	—	✓	562.11 (+ 0.0%)	✓	

**Table 4.** Summary of the results obtained by Q4RPD. Number of (sub-)routes is represented as [A,B,C,D] where A=regular routes; B: Depot-TP sub-routes; C: TP-TP sub-routes and D: TP-Depot sub-routes. In Validation section: ✓ = restriction fulfilled; ✗ = restriction broken; and —: constraint not present in the instance. In Objectives of the SRP section: value in brackets represent the distance difference of the sums of  $o_1$  when using our solver Q4RPD from the optimal result found by Google OR-Tools; while ✓ in last column represents the optimality in all the  $o_2$  computed.

Having said this, the Q4RPD performance of the **objective function** and the **constraints** for the six instances are summarized in Table 4. Along with the reference to the figure in which the best-found solution of the instance is represented, the following information, divided into three main blocks, is provided in this table:

- **Problem solutions:** the number of complete routes that compose the best solution found by Q4RPD; and the number and type of (sub-)routes of that solution.
- **Validation:** in this second section, the fulfillment of constraints R1 (i.e., truck capacities), R2 (i.e., time constraints of TP deliveries), and R3 (i.e., the length of the working day) is assessed.
- **Objectives of the SRP:** concerning the efficiency of the solutions found, and for comparison purposes, a classic TSP has been selected as the baseline algorithm to assess the performance of our solution. Concretely, each route built by Q4RPD is evaluated against a TSP solution obtained from Google OR-Tools, which has been used for this study because of the following reasons: (i) it is open-source software; (ii) it has been frequently employed for benchmarking purposes<sup>45–47</sup>; and (iii) it has demonstrated a good performance for solving combinatorial optimization problems<sup>48</sup>. It is important to note that Google OR-Tools only calculates the canonical TSP. Our solver Q4RPD, in turn, is forced to meet the TPs schedule imposed by R1, which causes tweaks in the path that increase the distance but avoids violating this constraint. Specifically, the sum of all the  $o_1$  objectives calculated by Q4RPD is displayed with a percentage representing the difference with respect to the sum of the costs of the solutions obtained by Google OR-Tools. In addition to that, we also represent in Table 4 if all  $o_2$  calculated by Q4RPD have been optimized, i.e., if all the generated (sub-)routes maximize the number of deliveries attended while meeting restrictions R1 and R3.

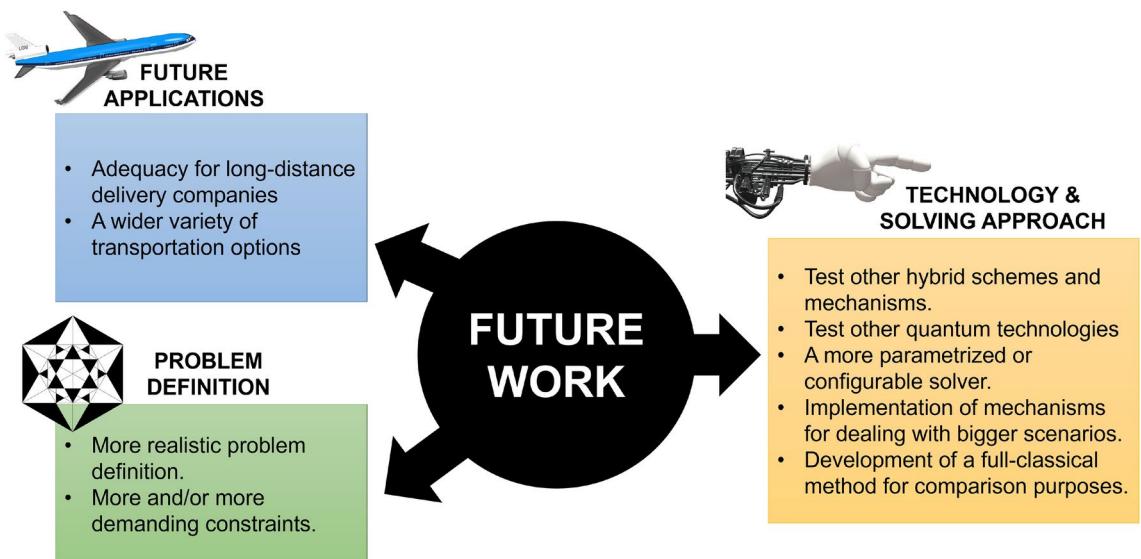
Two main conclusions can be drawn from the results. First, Q4RPD has proved to be a promising solver for dealing with 2DH-PDP. Regarding the performance of the solver in terms of constraint fulfillment and its capability to minimize the objective function:

- Q4RPD has emerged as a suitable solver to efficiently deal with the constraints imposed on the problem, always providing solutions that meet the imposed restrictions.
- In terms of distance minimization, it can be seen that Q4RPD can compete with a classical solver such as Google OR-Tools. On the one hand, in cases where there are no TP deliveries, i.e., D21\_P0 and D29\_P0, the routes calculated by Q4RPD have been demonstrated to be the optimal ones. On the other hand, in those instances with TP deliveries, Q4RPD provide solutions with a deviation of less than 6.3% in all cases. It should be noted that this deviation in the objective  $o_1$  is not attributed to the performance of the routing algorithm but to the obligation of Q4RPD to comply with the restriction  $R2$ . In fact, in those cases in which Google OR-Tools provides better results in terms of  $o_1$ , the restriction  $R2$  is not met. All this allows certifying the good performance of Q4RPD for the scenarios tackled in this research. Last but not least, the iterative nature of Q4RPD permits solving instances of a larger size than usually seen in the current NISQ-era literature. While Q4RPD can efficiently cope with problems such as D29\_P0, composed of 30 nodes, recent studies on routing problems through the QC perspective work with significantly smaller problems<sup>22,23,25,26,30,49</sup>. Finally, for the sake of replicability, all the instances employed in this experimentation as well as the detailed outcomes are openly available<sup>50</sup>.

### Conclusions and future work

In this paper, a quantum-classical system for solving a real-world logistic problem is presented, named *Quantum for Real Package Delivery* (Q4RPD). Specifically, this solving scheme deals with complex routing problems whose specifications include constraints such as a heterogeneous fleet of vehicles, priority deliveries, package description, and truck capacity according to weight and dimensions. For this reason, the problem has been coined as *2-dimensional and Heterogeneous Package Delivery with Priorities* (2DH-PDP). To demonstrate the applicability, six instances of different sizes and characteristics have been utilized. Several inspiring challenges and opportunities have been identified for future work, graphically summarized and categorized in Fig. 11:

- As for the **problem definition**:
  - An extension of the mathematical formulation of 2DH-PDP to include further real-world situations. Some examples of new features would add more realistic and/or complex cost calculations with the cost of the node-to-node trip subject to pre-calculated fuel consumption, the likelihood of road congestion,
  - The use of 3D dimensions, i.e., depth, width, and height, to categorize packages and truck capacities. This line would relate this work to other branches of research, such as the bin packing problem<sup>51</sup>.
  - The possibility of reusing the trucks to complete multiple routes.
- As for the **future applications**:
  - The extension of Q4RPD to companies with long-distance deliveries that would exceed the one-day delivery time.
  - Consider other transportation options as possible means of transport, such as cargo flights or ships.



**Fig. 11.** A graphical summary of the planned future work.

- With respect to **technology and the solving approach**:
  - The use of complexity reduction mechanisms to deal with larger scenarios efficiently by reducing the problem's size while maintaining all the constraints of the problem.
  - The design and implementation of hybrid methods incorporating another QC paradigm or technology.
  - The development of more configurable solvers that could automatically handle different company preferences or easily adapt to other companies' requirements. Another interesting avenue would be to grant the solver the ability to transform heuristics, i.e., preferences, into constraints or sub-objectives in the objective function, and the other way around.
  - The implementation of a full-classical algorithm to make a performance comparison with Q4RPD.

## Data availability

The data used, as well as the results discussed, are available at <http://dx.doi.org/10.17632/yv48pk96y1>.

Received: 27 March 2024; Accepted: 7 October 2024

Published online: 21 October 2024

## References

1. Kucukoglu, I., Dewil, R. & Cattrysse, D. The electric vehicle routing problem and its variations: A literature review. *Comput. Ind. Eng.* **161**, 107650 (2021).
2. Asghari, M. et al. Green vehicle routing problem: A state-of-the-art review. *Int. J. Prod. Econ.* **231**, 107899 (2021).
3. Konstantakopoulos, G. D., Gayialis, S. P. & Kechagias, E. P. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Oper. Res.* **22**(3), 2033–2062 (2020).
4. Hochba, D. S. Approximation algorithms for np-hard problems. *ACM Sigact News* **28**, 40–52 (1997).
5. Lenstra, J. K. & Kan, A. R. Complexity of vehicle routing and scheduling problems. *Networks* **11**, 221–227 (1981).
6. Caceres-Cruz, J., Arias, P., Guimaraes, D., Riera, D. & Juan, A. A. Rich vehicle routing problem: Survey. *ACM Comput. Surv. (CSUR)* **47**, 1–28 (2014).
7. Laporte, G. & Nobert, Y. Exact algorithms for the vehicle routing problem. In *North-Holland mathematics studies*, vol. 132, 147–184 (Elsevier, 1987).
8. Rosenkrantz, D. J., Stearns, R. E. & Lewis, P. M. II. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.* **6**, 563–581 (1977).
9. Elshaer, R. & Awad, H. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Comput. Ind. Eng.* **140**, 106242 (2020).
10. Zhang, Z., Liu, H., Zhou, M. & Wang, J. Solving dynamic traveling salesman problems with deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* **34**, 2119–2132 (2021).
11. Xin, L., Song, W., Cao, Z. & Zhang, J. Step-wise deep learning models for solving routing problems. *IEEE Trans. Ind. Inf.* **17**, 4861–4871 (2020).
12. Steane, A. Quantum computing. *Rep. Progress Phys.* **61**, 117 (1998).
13. Gill, S. S. et al. Quantum computing: A taxonomy, systematic review and future directions. *Softw. Pract. Exp.* **52**, 66–114 (2022).
14. Yang, Z., Zolanvari, M. & Jain, R. A survey of important issues in quantum computing and communications. *IEEE Commun. Surv. Tutor.* **25**(2), 1059–1094 (2023).
15. Ur Rasool, R. et al. Quantum computing for healthcare: A review. *Future Internet* **15**, 94 (2023).
16. Egger, D. J. et al. Quantum computing for finance: State-of-the-art and future prospects. *IEEE Trans. Quantum Eng.* **1**, 1–24 (2020).
17. Luckow, A., Klepsch, J. & Pichlmeier, J. Quantum computing: Towards industry reference problems. *Digit. Welt* **5**, 38–45 (2021).
18. Ajagekar, A. & You, F. Quantum computing for energy systems optimization: Challenges and opportunities. *Energy* **179**, 76–89 (2019).
19. Osaba, E., Villar-Rodriguez, E. & Oregi, I. A systematic literature review of quantum computing for routing problems. *IEEE Access* **10**, 55805–55817 (2022).
20. Gutin, G. & Punnen, A. P. *The traveling salesman problem and its variations* Vol. 12 (Springer Science & Business Media, Cham, 2006).
21. Toth, P. & Vigo, D. *The vehicle routing problem* (SIAM, 2002).
22. Delgado, I. P., Markaida, B. G., de Leceta, A. M. F. & Uriarte, J. A. O. Quantum hobbit routing: Annealer implementation of generalized travelling salesperson problem. [arXiv:2309.16522](https://arxiv.org/abs/2309.16522) (2023).
23. Qian, W. et al. Comparative study of variations in quantum approximate optimization algorithms for the traveling salesman problem. *Entropy* **25**, 1238 (2023).
24. Tsyvinski, C. & Beterov, I. A quantum algorithm for solving the travelling salesman problem by quantum phase estimation and quantum search. *J. Exp. Theor. Phys.* **137**, 210–215 (2023).
25. Le, T. V., Nguyen, M. V., Khandavilli, S., Dinh, T. N. & Nguyen, T. N. Quantum annealing approach for selective traveling salesman problem. In *ICC 2023–IEEE International Conference on Communications*, 2686–2691 (IEEE, 2023).
26. Mohanty, N., Behera, B. K. & Ferrie, C. Analysis of the vehicle routing problem solved via hybrid quantum algorithms in the presence of noisy channels. *IEEE Transactions on Quantum Engineering* (2023).
27. Poggel, B., Quetschlich, N., Burgholzer, L., Wille, R. & Lorenz, J. M. Recommending solution paths for solving optimization problems with quantum computing. In *2023 IEEE International Conference on Quantum Software (QSW)*, 60–67 (IEEE, 2023).
28. Leonidas, I. D., Dukakis, A., Tan, B. & Angelakis, D. G. Qubit efficient quantum algorithms for the vehicle routing problem on quantum computers of the nisq era. [arXiv:2306.08507](https://arxiv.org/abs/2306.08507) (2023).
29. Bentley, C. D., Marsh, S., Carvalho, A. R., Kilby, P. & Biercuk, M. J. Quantum computing for transport optimization. [arXiv:2206.07313](https://arxiv.org/abs/2206.07313) (2022).
30. Tambunan, T. D., Suksmono, A. B., Edward, I. J. M. & Mulyawan, R. Quantum annealing for vehicle routing problem with weighted segment. In *AIP Conference Proceedings*, vol. 2906 (AIP Publishing, 2023).
31. Osaba, E., Villar-Rodriguez, E., Gomez-Tejedor, A. & Oregi, I. Hybrid quantum solvers in production: how to succeed in the nisq era? [arXiv:2401.10302](https://arxiv.org/abs/2401.10302) (2024).
32. Weinberg, S. J., Sanches, F., Ide, T., Kamiya, K. & Correll, R. Supply chain logistics with quantum and classical annealing algorithms. *Sci. Rep.* **13**, 4770 (2023).
33. D-Wave Developers. Measuring Performance of the Leap Constrained Quadratic Model Solver. Tech. Rep. 14-1065A-A, D-Wave Systems Inc. (2022).
34. Quetschlich, N., Soeken, M., Murali, P. & Wille, R. Utilizing resource estimation for the development of quantum computing applications. [arXiv:2402.12434](https://arxiv.org/abs/2402.12434) (2024).
35. Kallehauge, B., Larsen, J., Madsen, O. B. & Solomon, M. M. *Vehicle routing problem with time windows* (Springer, Cham, 2005).

36. Feld, S. et al. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Front. ICT* **6**, 13 (2019).
37. Salehi, Ö., Glos, A. & Miszczak, J. A. Unconstrained binary models of the travelling salesman problem variants for quantum optimization. *Quantum Inf. Process.* **21**, 67 (2022).
38. Matai, R., Singh, S. P. & Mittal, M. L. Traveling salesman problem: An overview of applications, formulations, and solution approaches. *Travel. Salesman Problem Theory Appl.* **1**, 1–25 (2010).
39. Toth, P. & Vigo, D. *Vehicle routing: Problems, methods, and applications* (SIAM, 2014).
40. D-Wave Developers. D-Wave Hybrid Solver Service: An Overview. Tech. Rep. 14-1039A-B, D-Wave Systems Inc. (2020).
41. D-Wave Developers. Hybrid Solver for Constrained Quadratic Models. Tech. Rep. 14-1055A-A, D-Wave Systems Inc. (2021).
42. Glos, A., Kundu, A. & Salehi, Ö. Optimizing the production of test vehicles using hybrid constrained quantum annealing. *SN Comput. Sci.* **4**, 609 (2023).
43. Bożejko, W., Klempous, R., Uchroński, M. & Wodecki, M. Solving two-machine flow shop scheduling problem with total weighted number of on-time tasks maximization criterion on d-wave quantum computer. In *2023 IEEE 23rd International Symposium on Computational Intelligence and Informatics (CINTI)*, 000037–000040 (IEEE, 2023).
44. Bożejko, W., Burduk, A., Pempera, J., Uchroński, M. & Wodecki, M. Optimal solving of a binary knapsack problem on a d-wave quantum machine and its implementation in production systems. *Annals of Operations Research* 1–16 (2024).
45. Pan, X. et al. H-tsp: Hierarchically solving the large-scale traveling salesman problem. *Proc. AAAI Conf. Artif. Intell.* **37**, 9345–9353 (2023).
46. Silva, A. S. et al. Capacitated waste collection problem solution using an open-source tool. *Computers* **12**, 15 (2023).
47. Guo, Y., Ren, Z. & Wang, C. imtsp: Solving min-max multiple traveling salesman problem with imperative learning. [arXiv:2405.00285](https://arxiv.org/abs/2405.00285) (2024).
48. Da Col, G. & Teppan, E. Google vs ibm: A constraint solving challenge on the job-shop scheduling problem. [arXiv:1909.08247](https://arxiv.org/abs/1909.08247) (2019).
49. Mori, N. & Furukawa, S. Quantum annealing for the adjuster routing problem. *Front. Phys.* **11**, 1129594 (2023).
50. Osaba, E. & Villar, E. Datasets and results for the 2-dimensional and heterogeneous package delivery with priorities problem. Online at Mendeley Data. <https://doi.org/10.17632/yv48pk96y1> (2024).
51. Romero, S. V., Osaba, E., Villar-Rodriguez, E., Oregi, I. & Ban, Y. Hybrid approach for solving real-world bin packing problem instances using quantum annealers. *Sci. Rep.* **13**, 11777 (2023).

## Acknowledgements

This work was supported by the Basque Government through HAZITEK program (Q4\_Real project, ZE-2022/00033) and through Plan complementario comunicación cuántica (EXP. 2022/01341) (A/20220551). The authors thank Miguel Angel Casla, Ertransit's Director of Operations, for his assistance throughout this research.

## Author contributions

All authors conceived the research and defined the problem. All authors conceived the experiments. E.O. and E.V.R design the solver. E.O. developed the code and conducted the experimentation. E.O. and E.V.R. wrote the manuscript. All authors reviewed the manuscript.

## Declaration

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to E.O.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024