

# 2024Osaba Solving-a-real-world-package-delivery-routing-problem-using-quantum-annealers

Mikhail Lukianov

Devember 7, 2025

## Table of Contents

<b>1 Context Summary (for other chats)</b>	<b>2</b>
<b>2 Problem Definition (Lossless + Compact)</b>	<b>2</b>
2.1 Constraints	2
2.2 Preferences (Lexicographic)	3
<b>3 Solving Scheme</b>	<b>3</b>
3.1 Sub-route types	3
3.2 Each trajectory is solved as a <b>Single Routing Problem (SRP)</b> via a <b>Constrained Quadratic Model (CQM)</b> .	3
<b>4 Initialization</b>	<b>3</b>
4.1 Vehicle ordering	3
4.2 Delivery ordering	3
4.3 Loop start	3
<b>5 Iterative Construction (<math>S_1 \rightarrow S_2 \rightarrow S_3</math>)</b>	<b>4</b>
5.1 $S_1$ - Select Scenario and Parameters	4
5.1.1 Upper-bound duration $rt$	4
5.1.2 Capacity update	4
<b>6 Single Routing Problem (SRP)</b>	<b>4</b>
6.1 Parameters	4
6.2 Variables (node-based encoding)	5
<b>7 SRP Objectives</b>	<b>5</b>
7.1 Objective $o_1$ : minimize route distance	5
7.2 Objective $o_2$ : place destination as late as possible	5
7.3 Total objective	5
<b>8 SRP Constraints</b>	<b>5</b>
8.1 Delivery-consistency	5
8.2 Position-consistency	6
8.3 Destination-inclusion	6
8.4 Time-restriction	6
8.5 Weight-restriction	6
8.6 Dimension-restriction	6
<b>9 <math>S_2</math> - Solve via LeapCQMHybrid</b>	<b>6</b>
<b>10 <math>S_3</math> - Assemble Solution</b>	<b>6</b>

## 1 Context Summary (for other chats)

I'm implementing a **probabilistic optimization engine** based on **p-bits** (stochastic binary units) that operate like a classical, tunable analogue of quantum annealing. My backend already includes a generic BQM/QUBO sampler using Gibbs/Metropolis updates on p-bits, capable of solving arbitrary quadratic Hamiltonians.

Now I'm applying this backend to a **real-world delivery routing problem** inspired by the Q4RPD ("Quantum for Real Package Delivery") framework. The goal is to compute efficient delivery routes that respect operational constraints while handling realistic features: heterogeneous trucks, weight/dimension limits, deadlines for priority deliveries (TP items), and maximum driver working hours.

The full routing task is decomposed into an iterative sequence of **Single Routing Problems (SRPs)**. Each SRP computes one **route or sub-route** for a given truck using a **node-based encoding** with binary decision variables  $x_{i,p}$ , indicating whether delivery  $i$  is visited at position  $p$  in the sequence. The mathematical structure includes:

- **Objective 1:** minimize total travel distance
- **Objective 2:** schedule the destination TP point as late as possible (maximize intermediate deliveries) Combined via weighted sum.

SRP is constrained by seven real-world rules:

1. A delivery appears at most once.
2. Each position is assigned to at most one delivery.
3. Delivery sequence must be contiguous (no internal gaps).
4. Destination must appear exactly once.
5. Total route time  $\leq$  allowed duration.
6. Total weight  $\leq$  truck capacity.
7. Total volume  $\leq$  truck capacity.

These objectives and constraints are encoded as a **QUBO energy function**, which my p-bit solver minimizes. Each SRP thus becomes a probabilistic search over a structured binary space.

The outer algorithm (**Q4RPD**) orchestrates multiple SRPs, selecting trucks, determining whether to compute a full route or a sub-route (Depot→TP, TP→TP, TP→Depot, or Regular), updating remaining deliveries, and concatenating partial routes until all deliveries are assigned.

In short:

- **Backend:** custom p-bit stochastic Ising/QUBO solver.
  - **Task:** implement a full routing system by expressing each sub-routing decision as a QUBO.
  - **Goal:** reproduce the logic of Q4RPD, but powered entirely by my own probabilistic hardware-inspired solver.
- 

## 2 Problem Definition (Lossless + Compact)

We solve a **2-dimensional, heterogeneous package-delivery routing problem with priorities (2DH-PDP)**. Given deliveries  $P$ , a depot, and both **owned** and **rented** trucks, we construct depot→deliveries→depot routes minimizing:

- total travel distance,
- rental-truck fees.

A **route** is the trajectory of a single truck serving a subset of deliveries.

### 2.1 Constraints

- **R1 - 2D Capacity:**

$$\sum_{i \in \text{route}} w_i \leq W_v, \quad \sum_{i \in \text{route}} d_i \leq D_v.$$

- R2 - Priority deadlines:

$$t_{\text{arrival}}(i) \leq t_i.$$

- R3 - Daily duration:

$$\text{duration(route)} \leq T_{\text{workday}}.$$

Distance and travel time share the same numeric value.

## 2.2 Preferences (Lexicographic)

1. P1: Each truck runs at most one route per day:

$$r_v \leq 1.$$

2. P2: Prefer owned trucks over rentals.

3. P3: Prefer using fewer total trucks.
- 

## 3 Solving Scheme

The algorithm constructs **one trajectory at a time**, using **sub-routes** where necessary to satisfy TP deadlines.

### 3.1 Sub-route types

- Depot → TP
- TP → TP
- TP → Depot
- Regular route (Depot → Depot)

### 3.2 Each trajectory is solved as a Single Routing Problem (SRP) via a Constrained Quadratic Model (CQM).

## 4 Initialization

### 4.1 Vehicle ordering

Owned trucks sorted by capacity, then rented trucks sorted by capacity (enforces P2, P3).

### 4.2 Delivery ordering

TP deliveries are dequeued early (planning priority, not delivery priority).

### 4.3 Loop start

Initialize pending deliveries and enter the iterative procedure.

---

## 5 Iterative Construction ( $S1 \rightarrow S2 \rightarrow S3$ )

### 5.1 S1 - Select Scenario and Parameters

Choose the first truck in the ordered list. Its state (depot or mid-route) and the presence of remaining TP deliveries determine one scenario:

- TP→TP
- TP→Depot
- Depot→TP
- Regular route

#### 5.1.1 Upper-bound duration $rt$

- **TP→TP:**

$$rt = t_{\text{TP-destination}} - t_{\text{elapsed-from-depot}}.$$

- **TP→Depot:**

$$rt = T_{\text{workday}} - t_{\text{elapsed-from-depot}}.$$

- **Depot→TP:**

$$rt = t_{\text{TP-destination}}.$$

- **Regular route:**

$$rt = T_{\text{workday}}.$$

#### 5.1.2 Capacity update

If partial payload already assigned:

$$W \leftarrow W - \sum w_i, \quad D \leftarrow D - \sum d_i.$$

---

## 6 Single Routing Problem (SRP)

An SRP computes exactly one trajectory (a full route or sub-route).

### 6.1 Parameters

- $rt$ : allowed duration
- $W, D$ : remaining capacity
- $P$ : pending deliveries
- $M = |P|$
- $w_i, d_i$ : weight and volume
- $t_i$ : deadline
- $c_{i,j}, d_{i,j}$ : travel time and distance (numerically equal)

## 6.2 Variables (node-based encoding)

Binary variables:

$$x_{i,p} = \begin{cases} 1 & \text{if delivery } i \text{ is visited at position } p, \\ 0 & \text{otherwise,} \end{cases} \quad i \in \{0, \dots, M\}, p \in \{0, \dots, M\}.$$

Location index  $i = 0$  denotes the origin.

The route encoding is:

$$X = X_0, \dots, X_M, \quad X_i = x_{i,0}, \dots, x_{i,M}.$$

Initialization:

- $x_{0,0} = 1$
  - Destination variables are included but not position-fixed.
- 

## 7 SRP Objectives

Weights:  $\omega_1 = 1, \omega_2 = 2$ .

### 7.1 Objective $o_1$ : minimize route distance

$$o_1 = \sum_{p=0}^{M-1} \sum_{i=0}^M \sum_{j=0}^M d_{i,j} x_{i,p} x_{j,p+1}, \quad i \neq j.$$

### 7.2 Objective $o_2$ : place destination as late as possible

Destination index is 1:

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

### 7.3 Total objective

$$\min (\omega_1 o_1 + \omega_2 o_2).$$


---

## 8 SRP Constraints

### 8.1 Delivery-consistency

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

## 8.2 Position-consistency

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

## 8.3 Destination-inclusion

$$\sum_{p=0}^M x_{1,p} = 1.$$

## 8.4 Time-restriction

$$\sum_{p=0}^{M-1} \sum_{i=0}^M \sum_{j=0}^M c_{i,j}, x_{i,p}, x_{j,p+1} \leq rt, \quad i \neq j.$$

## 8.5 Weight-restriction

$$\sum_{i=0}^M w_i \sum_{p=0}^M x_{i,p} \leq W.$$

## 8.6 Dimension-restriction

$$\sum_{i=0}^M d_i \sum_{p=0}^M x_{i,p} \leq D.$$

---

## 9 S2 - Solve via LeapCQMHybrid

A CQM instance is submitted to D-Wave's hybrid solver. Quantum queries (Advantage\_system6.4) guide classical heuristics. Internal details and qubit requirements are proprietary.

## 10 S3 - Assemble Solution

- Append the new sub-route to the truck's ongoing trajectory.
- If the truck returns to depot, its route is complete and it is removed from the list.
- Remove all served deliveries from the pending set.

Repeat until no deliveries remain.