KIRO 2025 – Califrais

Guillaume Dalle, Louis Bouvier, Léo Baty, Massil Hihat

November 10, 2025

1 Industrial context

Califrais[1] is the official digital and logistics operator of the Rungis International Market, one of the largest wholesale food markets in the world, located in the suburbs of Paris. Through its online platform[2], it allows restaurants and other businesses inside Paris to order a wide variety of fresh products while mutualizing deliveries. Instead of having to place one order for fruits and vegetables, another one for meat, another one for fish, and so on, customers can purchase everything they need from the same marketplace. Then, Califrais gathers the items from suppliers, stores them in its warehouse, and handles deliveries with a fleet of trucks.

The topic of this challenge is the optimization of truck routes, a problem solved daily by Califrais using operations research methods. Customers have until midnight to complete their orders online for the next day. Then, loading must begin very quickly in the warehouse, since the first trucks depart early in the morning. Your goal is to design and implement an algorithm that schedules deliveries, taking into account all degrees of freedom and operational constraints. As a rule of thumb, your algorithm should not run for longer than 10 minutes to tackle the largest problem instance on a standard personal laptop.

2 Input data

Vehicles Califrais can rent from an unlimited fleet of vehicles. The vehicles are split into F families. Inside each vehicle family f, the vehicles are interchangeable and have common features:

- A maximum capacity w_f^{capacity}.
- A daily rental cost c_f^{rental}.
- A unit fuel cost per meter traveled c_f^{fuel}.
- A unit radius cost per squared meter c_f^{radius} (see below).
- A speed s_f
- A parking time p_f
- Temporal variation coefficients α_f, β_f, for computing travel times (see below).

Orders Califrais must serve a set of orders denoted by $\mathcal{I}$. Each order $i \in \mathcal{I}$ is defined by:

- Its ID i.
- Its spatial coordinates (φ_i, λ_i), where φ_i is the latitude and λ_i is the longitude (both in degrees).
- Its weight $w_i \in \mathbb{R}_+$.

[1] https://www.califrais.fr/
[2] https://rungismarket.com/

- Its delivery time window $[t_i^{\min}, t_i^{\max}]$.
- Its delivery duration ℓ_i.

The Califrais depot, or warehouse, is denoted by $i = 0$. All trucks depart from there at $t^{\mathrm{dep}} = 0$.

Distances Califrais operates on the Paris region road network, but taking into account every street would be too complex. Instead, we work on a complete directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ linking all orders $i \in \mathcal{I}$ to each other and to the depot 0. In other words, its vertices and edges are given by

$$\mathcal{V} = \mathcal{I} \cup \{0\} \qquad \text{and} \qquad \mathcal{E} = \{(i,j) \in \mathcal{V}^2 : i \neq j\}$$

Given two nodes i and j, we will make use of the following distances:

- Manhattan distance $\delta^{\mathrm{M}}(i,j) = |x_j - x_i| + |y_j - y_i|$
- Euclidean distance $\delta^{\mathrm{E}}(i,j) = \sqrt{|x_j - x_i|^2 + |y_j - y_i|^2}$

To perform the conversion from spherical coordinates expressed in degrees to geodesic segments with a length in meters, we use the correspondence between angles in radians and arc lengths:

$$\begin{aligned} y_j - y_i &= \rho \frac{2\pi}{360} (\varphi_j - \varphi_i) \\ x_j - x_i &= \rho \left(\cos \frac{2\pi}{360} \varphi_0 \right) \frac{2\pi}{360} (\lambda_j - \lambda_i) \end{aligned}$$

where $\rho = 6.371 \times 10^6$ m is the Earth's radius and φ_0 is the latitude of the depot (arbitrarily).

Travel times While physical distances are constant, travel times can depend on the vehicle type (big vehicles are slower) and the time of day (rush hour is bad on the *périphérique*). We express the travel time $\tau_f(i, j \mid t)$ between two nodes as a reference travel time multiplied by a time-dependent factor:

$$\tau_f(i, j \mid t) = \tau_f(i,j) \cdot \gamma_f(t)$$

The reference travel time is an affine function of the Manhattan distance:

$$\tau_f(i,j) = \frac{\delta^{\mathrm{M}}(i,j)}{s_f} + p_f$$

where s_f is the speed of the vehicle and p_f is the parking time. The influence of time is periodic with period $T = 1$ day $= 86400$ s, so we decompose it as a truncated Fourier series with angular frequency $\omega = 2\pi/T$:

$$\gamma_f(t) = \sum_{n=0}^{3} \left(\alpha_f(n) \cos(n\omega t) + \beta_f(n) \sin(n\omega t) \right)$$

You can assume that these travel times satisfy the First In, First Out (FIFO) property: if you leave later, you arrive later.

3 Solutions

Routes A solution to the delivery scheduling problem is expressed as a set $\mathcal{R}$ of routes. Each route $r \in \mathcal{R}$ is defined by the following attributes. The three first ones are enough to define the routes as decision variables, while the last two are auxiliary and can be derived from the first three and the problem data.

- The family of vehicle used $f_r \in \{1, \ldots, F\}$.

- The number of orders served n_r.
- The sequence of vertices visited, starting and ending at the depot and visiting orders in between:

$$(i_r^0, i_r^1, \dots, i_r^{n_r}, i_r^{n_r+1}) \quad \text{with} \quad i_r^0 = i_r^{n_r+1} = 0 \quad \text{and} \quad i_r^k \in \mathcal{I} \text{ for } k \in \{1, \dots, n_r\}.$$

- The sequence of arrival times at vertices $(a_r^1, \dots, a_r^{n_r}, a_r^{n_r+1})$ (not defined for the initial vertex).
- The sequence of departure times from vertices $(d_r^0, d_r^1, \dots, d_r^0, d_r^{n_r})$ (not defined for the final vertex).

We say that an order i belongs to a route r, and we write $i \in r$, if there exists $k \in \{1, \dots, n_r\}$ such that $i_r^k = v$. We say that an edge $e = (i, j)$ belongs to a route r, and we write $(i, j) \in r$, if there exists $k \in \{1, \dots, n_r\}$ such that $i_r^{k-1} = i$ and $i_r^k = j$.

Constraints We now enumerate all the constraints that must be satisfied by a solution $\mathcal{R}$:

1. Each order belongs to exactly one route (so the routes are disjoint):

$$i \in \bigsqcup_{r \in \mathcal{R}} r \qquad \forall i \in \mathcal{I},$$

   where $\bigsqcup$ denotes the disjoint union.

2. Each vehicle is loaded with the total weight of all the orders on its route. This load must respect its maximum capacity:

$$\sum_{i \in r} w_i \leq w_{f_r}^{\text{capacity}} \qquad \forall r \in \mathcal{R}$$

3. Times:

   (a) A vehicle can only depart from an order after having arrived and then performed the delivery:

$$d_r^k \geq a_r^k + \ell_{i_r^k}$$

   (b) A vehicle can only arrive at its first vertex after having departed and traveled from the depot:

$$a_r^1 \geq d_r^0 + \tau_{f_r}(0, i_r^1 \mid t^{\text{dep}}) \qquad \forall r \in \mathcal{R}$$

   (c) A vehicle can only arrive at a subsequent vertex after having departed and traveled from the previous vertex:

$$a_r^{k+1} \geq d_r^k + \tau_{f_r}(i_r^k, i_r^{k+1} \mid d_r^k) \qquad \forall r \in \mathcal{R}$$

   (d) Arrival times at each order must fall within the order's delivery window:

$$t_{i_r^k}^{\min} \leq a_r^k \leq t_{i_r^k}^{\max} \qquad \forall r \in \mathcal{R}, \forall k \in \{1, \dots, n_r\}$$

   Note that parking at an order's location and waiting until the window opens is permitted (strictly speaking, the arrival time is the time at which delivery starts).

Objective Califrais strives to minimize the total cost of all routes, composed of several components:

$$\min_{\mathcal{R}} c(\mathcal{R}) = \sum_{r \in \mathcal{R}} \left(c^{\text{rental}}(r) + c^{\text{fuel}}(r) + c^{\text{radius}}(r) \right)$$

Individual route costs are defined as follows:

1. The daily rental cost, which pushes towards cheaper vehicle types:

$$c^{\text{rental}}(r) = c^{\text{rental}}_{f_r}$$

2. The total fuel cost based on the Manhattan distance traveled, which pushes towards fewer or shorter routes:

$$c^{\text{fuel}}(r) = c^{\text{fuel}}_{f_r} \sum_{k=0}^{n_r} \delta^{\text{M}}(i_r^k, i_r^{k+1})$$

3. A penalty for the squared Euclidean radius of the route (excluding the depot), weighted by c^{radius}, which pushes towards clustered deliveries:

$$c^{\text{radius}}(r) = c^{\text{radius}}_{f_r} \left(\frac{1}{2} \max_{\substack{k \in \{1,\dots,n_r\} \\ \ell \in \{1,\dots,n_r\} \\ k \neq \ell}} \delta^{\text{E}}(i_r^k, i_r^\ell) \right)^2$$

4 Data format

Conventions and units

- Times are expressed in seconds, taking midnight as a reference for times (5:20 AM is given as 19200 for instance).
- Distances are expressed in meters.
- Weights and capacities are expressed in kilograms.
- Longitude and latitude are expressed in degrees.
- Costs are expressed in euros.

Input files We now describe the data provided by Califrais, as a set of two CSV files.

`vehicles.csv` contains one row per vehicle family (F in total), with the following columns:

`family` : f

`max_capacity` (kg): w_f^{capacity}

`rental_cost` (€): c_f^{rental}

`fuel_cost` (€/m): c_f^{fuel}

`radius_cost` (€/m^2): c_f^{radius}

`speed` (m/s): s_f

`parking_time` (s): p_f

`fourier_cos_0` (unitless): $\alpha_f(0)$

`fourier_cos_1` (unitless): $\alpha_f(1)$

fourier_cos_2 (unitless): $\alpha_f(2)$

fourier_cos_3 (unitless): $\alpha_f(3)$

fourier_sin_0 (unitless): $\beta_f(0)$

fourier_sin_1 (unitless): $\beta_f(1)$

fourier_sin_2 (unitless): $\beta_f(2)$

fourier_sin_3 (unitless): $\beta_f(3)$

instance.csv contains one row for the depot and one row per order ($|\mathcal{V}|$ in total), with the following columns:

id : i

latitude (degrees): φ_i

longitude (degrees): λ_i

order_weight (kg): w_i (empty for the depot)

window_start (s): $t_i^{\min}$ (empty for the depot)

window_end (s): $t_i^{\max}$ (empty for the depot)

delivery_duration (s): ℓ_i (empty for the depot)

For the depot, the last 4 columns are empty.

Note that the vehicles.csv is the same for all problem instances, while the instance.csv varies.

Solution file The solution you provide must consist of a single CSV file:

routes.csv contains one row per route ($|\mathcal{R}|$ in total) with the following columns:

family f_r

order_1 i_r^1

order_2 i_r^2

...

order_n_r $i_r^{n_r}$

To build this file, first find the maximum number of orders served by a route in your solution $N = \max_{r\in\mathcal{R}} n_r$. The resulting table should have $2+N$ columns, and the routes shorter than N should end with empty columns (not filled with a space or a null placeholder of any kind).

Thanks to the problem constraints, the arrival and departure times at all vertices can be reverse-engineered from the sequence of orders, so they do not need to be specified.

5 Evaluation

Each team can upload their solution files to the KIRO platform, one file per problem instance. For each solution, the platform will check feasibility and compute the total cost. If a solution uploaded is infeasible, a very large penalty cost will be assigned to it instead. Note that timing is checked with a tolerance of 10^{-5} to avoid numerical issues. Then, the score of each team will be computed as the sum of the costs of all their solutions over all problem instances. The team with the lowest total cost will be declared the winner of the challenge.

6 Guidelines

Participants are free to use any programming language, libraries, or tools they deem appropriate to solve the challenge. However, they must ensure that their solution can be executed on a standard personal laptop within a reasonable time frame (ideally under 10 minutes for the largest instances).

7 Acknowledgements

This challenge is organized in collaboration with Califrais, who provided the problem context and data. We thank their team for their support and insights throughout the development of this challenge, as well as for their sponsorship. We also thank the KI for organizing this event, and the Fondation des Ponts and the CERMICS laboratory for their support of the KIRO 2025 program.