Generation of the XTW and XLTW Instances

Eduardo Uchoa Igor Malheiros Eduardo Queiroga Rafael Martinelli
Anand Subramanian

October 31, 2026

1 Overview

We propose new VRPTW benchmark sets derived from existing CVRP instances by adding time-window data, as shown in Table 1. The VRPTW Challenge will use the XLTW set; the statistically similar XLTW-Test is only for preliminary tests and will be abandoned when competition begins. The initial plan was to produce just those two sets, akin to what was done with the XL and XL-Test sets for the CVRPLib BKS Challenge on the CVRP [Queiroga et al., 2026]. However, the limitations of the Solomon and Homberger & Gehring instances discussed in Section 2 also affect the range of 100 to 1,000 customers, where they remain the standard benchmark. *We therefore found it worthwhile to also propose the XTW set, built on the X instances.*

The official XLTW instances will remain confidential until the challenge starts on February 24, 2027. What is released now is XLTW-Test, a set for developing and tuning algorithms before the competition. XLTW-Test is generated by the same scheme as XLTW, only with a different random seed. Both are built on the same XL instances, so the depot, customer locations and demands of XLTW are already public. What remains hidden until the challenge starts is the data drawn by the generator: time windows, service times, horizon and the new vehicle capacity.

Table 1: The new benchmark sets.

Set	Base CVRP set	Customers	Role	Release
XLTW-Test	The 100 XL instances [Queiroga et al., 2026]	1,047–10,000	Testing before the challenge, to be abandoned after it starts	Nov 2, 2026
XLTW	The 100 XL instances [Queiroga et al., 2026]	1,047–10,000	Official challenge instances	Feb 24, 2027
XTW	The 100 X instances [Uchoa et al., 2017]	100–1,000	New proposed benchmark, not part of the competition	Nov 2, 2026

2 Why not follow the Solomon scheme in the new instances?

The classic Solomon [1987] instances, with 100 customers, and their extension by Homberger and Gehring [2005] (H&G), with 200 to 1,000 customers, shaped four decades of VRPTW research [1]. By today's standards, however, they are rather simplistic. Two problems are fundamental: the instances vary little, and some of them were built around a planted solution.

[1] Indeed, many benchmark sets for *several other VRP variants* were derived from those VRPTW instances.

2.1 Little variety

H&G state that their instances "have been generated according to the procedure reported by Solomon", so both sets share the same narrow design space:

- **The depot is always central.** In every Solomon and H&G instance the depot sits in the middle of the customer area (Table 2). There is no eccentric or randomly placed depot.
- **Limited sets of customers.** Solomon did not generate customer locations: coordinates and demands were taken from CVRP test problems of Christofides et al. [1979]. The 56 instances use only four sets of 100 customers, one per class: all 23 R1 and R2 instances share the same locations and demands, as do the 9 C1, the 8 C2 and the 16 RC1 and RC2 instances. The R set is problem 3 of Christofides et al. [1979], the same points as the classical instance E-n101-k8. Within a class, only the time windows change. H&G drew new customers, with apparently uniform random locations in the R classes, but repeated the pattern: for each size, the 60 instances use only four sets of customers. The R1 and R2 instances share identical customers (same locations and demands, only renumbered), as do RC1 and RC2; only C1 and C2 differ. The 300 H&G instances thus rely on just 20 sets of customers.
- **Unrealistic clustering.** The C classes consist of many compact and well-separated clusters with nearly identical cardinality. Real urban problems look very different: clusters of different sizes, not so well-separated and with substantial overlap or even merging.
- **Two horizon types.** Classes 1 have a short horizon and few customers per route; classes 2 have a long horizon and many customers per route. Route length is not varied in between.

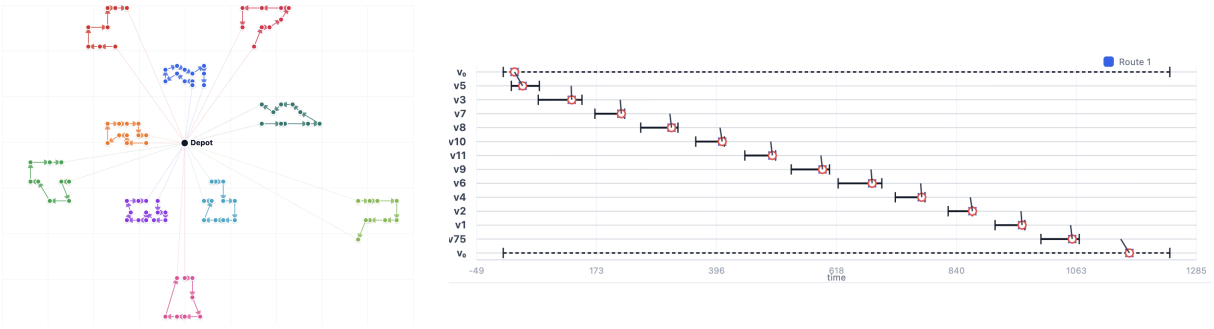
Table 2: Depot position in the Solomon and H&G instances.

Set	Customers	Customer area	Depot
Solomon R1, R2	100	about $[0, 70] \times [0, 80]$	$(35, 35)$
Solomon C1, C2, RC1, RC2	100	about $[0, 95] \times [0, 85]$	$(40, 50)$
H&G, 200 customers	200	$[0, 140]^2$	$(70, 70)$
H&G, 1,000 customers	1,000	$[0, 500]^2$	$(250, 250)$

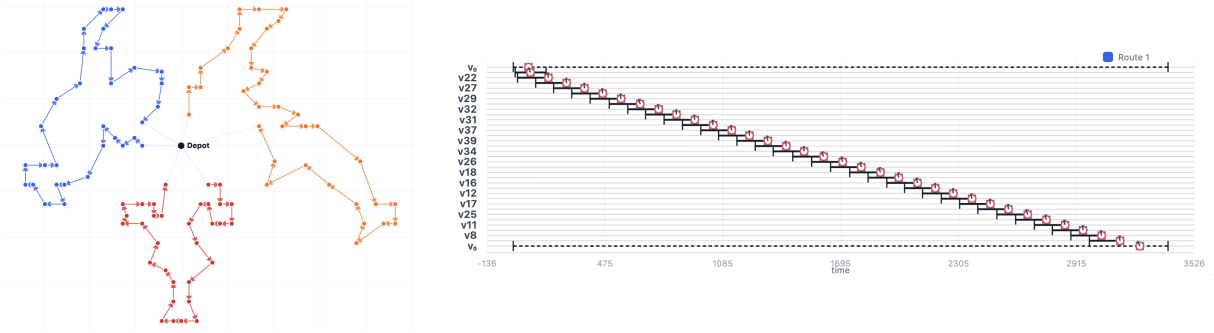
The CVRP X generator of Uchoa et al. [2017] (already classic, almost 1,000 citations) covers a much wider space: central, eccentric or random depot; random, clustered or mixed customers (different locations for each instance), with clusters of varying size grown from 3 to 8 seeds; seven demand distributions; and a wide range of route sizes.

2.2 Instances built around a planted solution

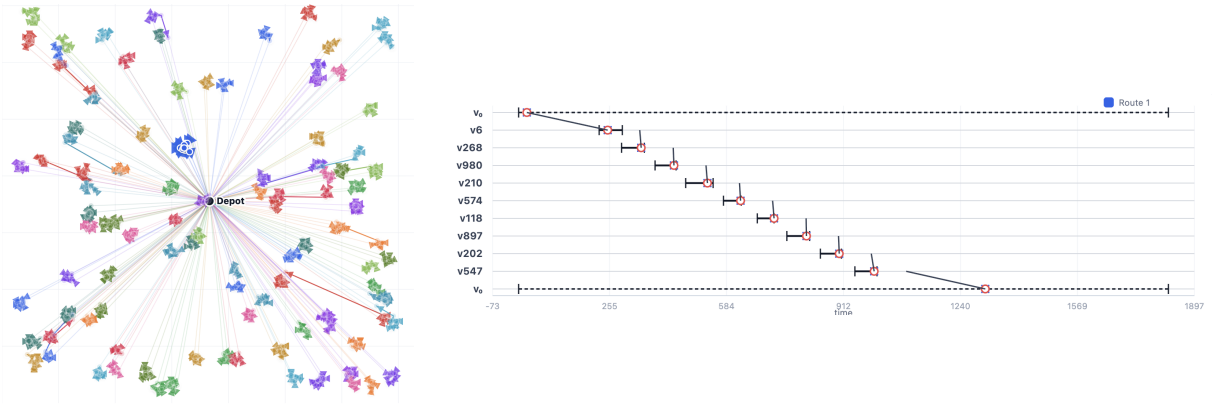
For the clustered classes C1 and C2, Solomon did not draw time windows at random. He first built TSP routes with 3-opt. Each route passes by the depot and by the customers in each cluster. He chose an orientation for each route, and then centered every customer's time window on the arrival time along that route. In his own words, this "permits the identification of a very good, possibly optimal, cluster-by-cluster solution" [Solomon, 1987]. Only the R and RC classes received time windows with random centers. H&G followed the same procedure, so their C1 and C2 classes also hide a planted solution, now with up to 1,000 customers. Figure 1 shows the outcome on C101, C201 and the H&G instance C1_10_1, with 1,000 customers. The resulting optimal routes are almost always the planted ones, their customers are visited exactly in the order induced by their position and by their time windows.



(a) C101



(b) C201



(c) C1_10_1

Figure 1: Optimal solutions (left) and the schedule of their first routes (right). In the Gantt charts, each row is a customer of the route in visiting order, the horizontal bar is its time window and the circle is the start of service. The windows form a staircase that follows the planted route.

Many clustered H&G instances are extremely easy for both heuristics and exact methods. Results over such instances can be misleading: they suggest that a method handles large VRPTW instances well when it has mostly recovered routes placed there by construction.

3 Generation procedure

XTW and XLTW instances are generated by the same procedure. From the base X or XL instance, the depot and customer coordinates and the customer demands are kept unchanged, while the original vehicle capacity is discarded. The generator then adds four kinds of data: a scheduling horizon, customer time windows, service times and a new vehicle capacity. Six parameters control this layer (Table 3).

Throughout this section, node 0 is the depot, customers are $1, \ldots, n$, d_i is the exact (unrounded) Euclidean distance from the depot to customer i, and $d_{\max} = \max_i d_i$. $U[a, b]$ denotes a continuous uniform draw and UD$[a, b]$ a discrete one. The exact distances d_i are used only internally, to compute the horizon, the windows and the service times. All of these are rounded to integers before being written to the instance. The travel distances and times of the instance itself follow the EUC_2D convention: Euclidean distances rounded to the nearest integer, as in X and XL.

Table 3: Parameters of the time-window layer. Widths and service times are fractions of the horizon H.

Parameter	Values
Horizon factor η	1.3, 1.6, 2, 3, 4
TW density ρ	60%, 80%, 100% of the customers
TW width	Narrow (5–10%), Medium (10–30%), Wide (30–50%), Mixed (one of the three, per customer)
TW center	Random, Clustered, Bimodal
Service time	Unit (1), Small (0.5–1%), Medium (1–2%), Large (2–5%), Mixed (Small, Medium or Large, per customer)
Capacity level	Tight (×0.9), Loose (×1.4), No capacity

3.1 Scheduling horizon

All routes must start and end within $[0, H]$, the working day. The horizon is tied to the geometry of each instance:

$$H = \text{round}\big(\eta\,(2\,d_{\max} + 1)\big), \qquad \eta \in \{1.3, 1.6, 2, 3, 4\}. \tag{1}$$

Coordinates and travel times are never scaled, so η sets how long the distances are relative to the day. With $\eta = 1.3$, a trip to farthest customers takes a significant part of the day and some routes are likely to be very short (in terms of number of served customers). With $\eta = 4$, the same trip is a small part of the day and some much longer routes become *possible* (yet, a combination of the other parameters may still produce an instance with shorter routes). The intermediate values produce more diversity, larger than the one available in Solomon's class 1 (short route) and class 2 (long routes)

Since window widths and service times are fractions of H, they keep the same meaning, a share of the working day, for every instance size and every value of η.

3.2 Time windows

A random subset of round(ρn) customers receives a time window $[e_i, l_i]$; the others keep the full horizon $[0, H]$. For each customer in the subset, the width w_i is drawn from $U[\alpha H, \beta H]$, where

$[\alpha, \beta]$ is the range of the chosen width level. Under Mixed, the level is first drawn uniformly among Narrow, Medium and Wide for each customer.

The center c_i of the window depends on the TW center parameter:

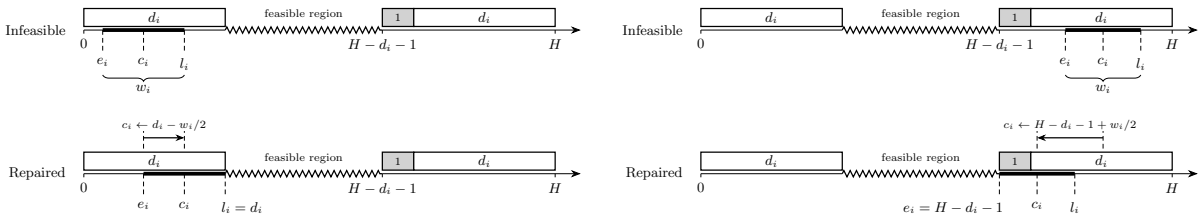
- **Random:** c_i is drawn from $U[d_i,\ H - d_i - 1]$, the interval of feasible service start times for a route that serves i alone.
- **Clustered:** a number of peak times $k \sim \mathrm{UD}[3, 5]$ is drawn, each from $U[0.15H,\ 0.85H]$. Each customer picks one peak uniformly and adds a jitter from $U[-0.1H,\ 0.1H]$. This models a few busy periods during the day.
- **Bimodal:** as Clustered, but with exactly two peaks at $0.25H$ and $0.75H$, a morning and an afternoon wave.

A window can only be feasible if it overlaps $[d_i,\ H - d_i - 1]$. When it does not, its center is shifted by the smallest amount that restores the overlap:

$$c_i \leftarrow \min\Big\{\max\Big\{c_i,\ d_i - \tfrac{w_i}{2}\Big\},\ H - d_i - 1 + \tfrac{w_i}{2}\Big\}. \tag{2}$$

The width is never reduced. Finally, the window is rounded outwards to integers, $e_i = \lfloor c_i - w_i/2 \rfloor$ and $l_i = \lceil c_i + w_i/2 \rceil$, and clipped to $[0, H]$.

Figure 2 shows the two cases in which a shift is necessary. In Figure 2a, the time window closes before the vehicle can reach customer i, so the center of the time window is shifted right to $c_i = d_i - w_i/2$ and the time window ends at $l_i = d_i$. In Figure 2b, the time window opens too late for the vehicle to serve i and return within the horizon time, so the center of the time window is shifted left to $c_i = H - d_i - 1 + w_i/2$ and the time window starts at $e_i = H - d_i - 1$. In both cases, the shift is the smallest that restores the overlap with the feasible region, and the width w_i is unchanged.



(a) Time window ending before d_i.

(b) Time window starting after $H - d_i - 1$.

Figure 2: Repair of a time window that does not overlap the feasible region.

3.3 Service times

Service times are drawn after the windows. Under Unit, $s_i = 1$ for every customer. Otherwise $s_i = \mathrm{round}(U[\alpha H, \beta H])$, where $[\alpha, \beta]$ is the range of the service level; under Mixed, the level is drawn per customer among Small, Medium and Large.

The windows above guarantee a feasible single-customer route only for a unit service time. If a longer s_i breaks that route, it is shrunk to the largest value that still fits, leaving the window unchanged:

$$\text{if } \max\{e_i, d_i\} + s_i + d_i > H: \qquad s_i \leftarrow \big\lfloor H - \max\{e_i, d_i\} - d_i \big\rfloor. \tag{3}$$

Figure 3 illustrates this repair. Since $e_i > d_i$, service starts at e_i, and the vehicle would return to the depot at $e_i + s_i + d_i > H$. The repair shrinks s_i to $\lfloor H - e_i - d_i \rfloor$, the largest integer value that keeps the route within the horizon.

Together with the window shift, this ensures that every customer can be served by a route of its own, so every instance is feasible.

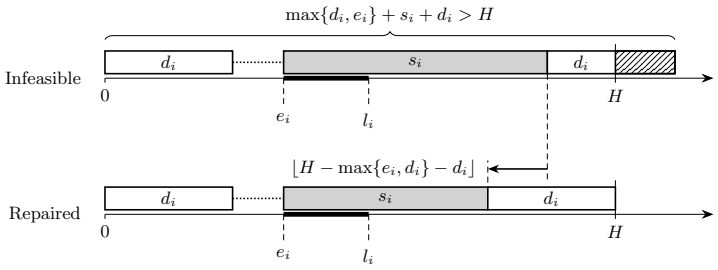


Figure 3: Repair of the service time when a single-customer route exceeds H.

3.4 Vehicle capacity

A sensible capacity depends on how many customers a route can serve, and that depends on the time windows and the horizon. The capacity is therefore set by first guessing the average number of customers per route in the uncapacitated instance, using the following formula:

$$\widehat{\kappa} = a \cdot H^{\beta_1} \cdot (\bar{s} + \bar{\nu})^{\beta_2} \cdot \bar{d}^{\beta_3} \cdot \bar{\omega}^{\beta_4} \cdot n^{\beta_5} \cdot \exp(\beta_6\, \tau) \cdot C_{\text{center}}, \tag{4}$$

where $\bar{s}$ is the mean service time, $\bar{\nu}$ the mean distance from a customer to its nearest customer, $\bar{d}$ the mean depot–customer distance, τ the fraction of customers whose window is narrower than $[0, H]$, $\bar{\omega}$ the mean width of those windows as a fraction of H, and C_{center} equals 1 for Random and is fitted for Clustered and Bimodal. All of them are measured on the instance itself, after the windows and service times are drawn. A route can visit roughly $H/(\bar{s} + \bar{\nu})$ customers, which motivates the two leading terms. Table 4 gives the coefficients of this log-linear model, obtained by a least squares fit over solutions of the uncapacitated XTW and XLTW-Test instances. A single model is used for both sets.

Table 4: Coefficients of the customers-per-route regression.

a	β_1	β_2	β_3	β_4	β_5	β_6	$C_{\text{Clustered}}$	C_{Bimodal}	R^2 (log)
0.835	1.301	−0.557	−0.704	0.400	0.082	−0.528	0.706	0.727	0.881

Let $D = \sum_i q_i$ be the total demand. The capacity of an instance with capacity factor $\phi \in \{0.9, 1.4\}$ is

$$Q = \max\Big\{ \Big\lceil \frac{D\,\widehat{\kappa}\,\phi}{n} \Big\rceil,\ \max_i q_i \Big\}, \tag{5}$$

so that Q divided by the average demand equals the predicted number of customers per route times ϕ. Tight makes the capacity binding, while Loose leaves it above what the time windows already allow. The floor $\max_i q_i$ keeps every customer servable. Under No capacity, $Q = D$ and only the time windows and the horizon constrain the routes.

Note that $\widehat{\kappa}$ is a somehow crude estimation of the average number of customers per route in the uncapacitated instance. Therefore, in some instances with $\phi = 0.9$ the capacity Q will not be much tight (because $\widehat{\kappa}$ was overestimated). Similarly, in some instances with $\phi = 1.4$, Q may be very restricting (because $\widehat{\kappa}$ was underestimated). Anyway, *on average* instances with $\phi = 0.9$ have tighter capacity than those with $\phi = 1.4$.

3.5 Assigning parameter values to instances

As in the X generation [Uchoa et al., 2017], in the XTW and XLTW sets parameter values are spread evenly over the benchmark rather than drawn independently. The base instances are sorted by number of customers. For a parameter with m possible values, every consecutive group

of m instances receives a fresh random permutation of all m values; each parameter is shuffled independently. Every subset of consecutive instances therefore contains a nearly balanced mix of each parameter, across the whole range of sizes.

3.6 Two decisions on conventions

3.6.1 Rounding the distances

In the Solomon and H&G instances, the most popular convention is to assume travel times equal Euclidean distances truncated to one decimal place. This choice is unfortunate. A value such as 8.1 has no exact representation in binary floating-point arithmetic, so the sum of such values along a route carries a small error. When a route is tight, that error can make a vehicle appear to arrive after the end of a time window, and a feasible solution is rejected.

Instance R112 illustrates the problem. Its optimal solution, of cost 948.6, contains the route 0–52–88–7–82–8–46–45–17–84–5–89–0, in which the vehicle reaches customer 5 at time 167.0, exactly the end of its window. A code that computes time in double precision obtains 167.00000000000003 and declares the route infeasible (unless a tolerance ϵ is added to every time-window check). The problem can not be solved by increasing numerical precision. A binary floating-point number represents exactly only fractions whose denominator is a power of two. For example, $8.5 = 17/2$ is 1000.1 in binary and is stored exactly, just like an integer. In contrast, $8.1 = 81/10$ has the factor 5 in its denominator, so its binary expansion $1000.0\overline{0011}$ never ends.

XTW and XLTW avoid the issue by following the TSPLIB EUC_2D convention already used in X and XL: travel distances and times are Euclidean distances rounded to the nearest integer. Time windows, service times and the horizon are also integers. Every time computation is then exact either in integer or floating-point arithmetic. Rounding to the nearest integer would have the sometimes undesirable effect of not satisfying the triangle inequality. However, in the XTW and XLTW sets, the service times are positive integers. Therefore, it is never possible to save time in a route by revisiting a customer.

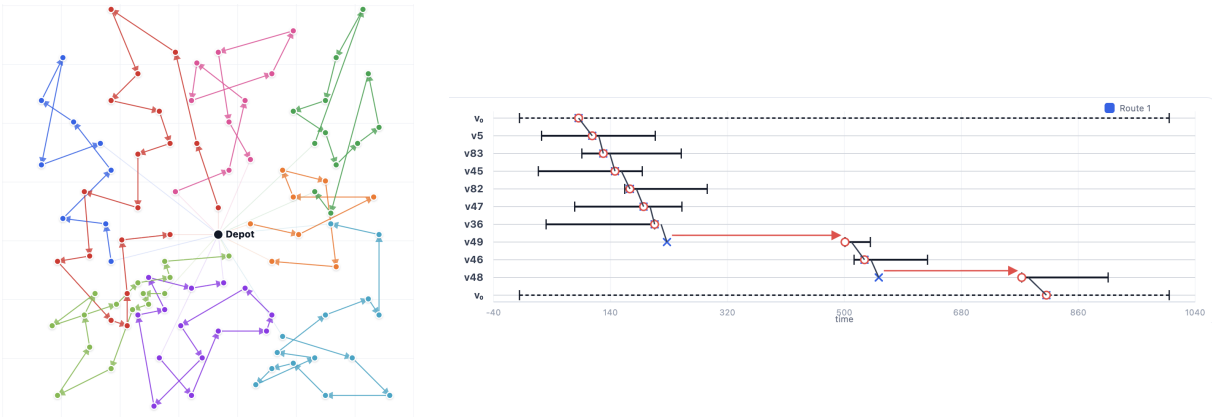
3.6.2 Objective function

Two objectives coexist in the VRPTW literature. Heuristic methods often minimize hierarchically: first the number of vehicles, then the total distance. This is the convention of the SINTEF benchmark pages[2]. Exact methods usually minimize only the total distance, which has the advantage of evaluating a solution by a single number. CVRPLib itself follows this second convention in its results for the Solomon and H&G instances.

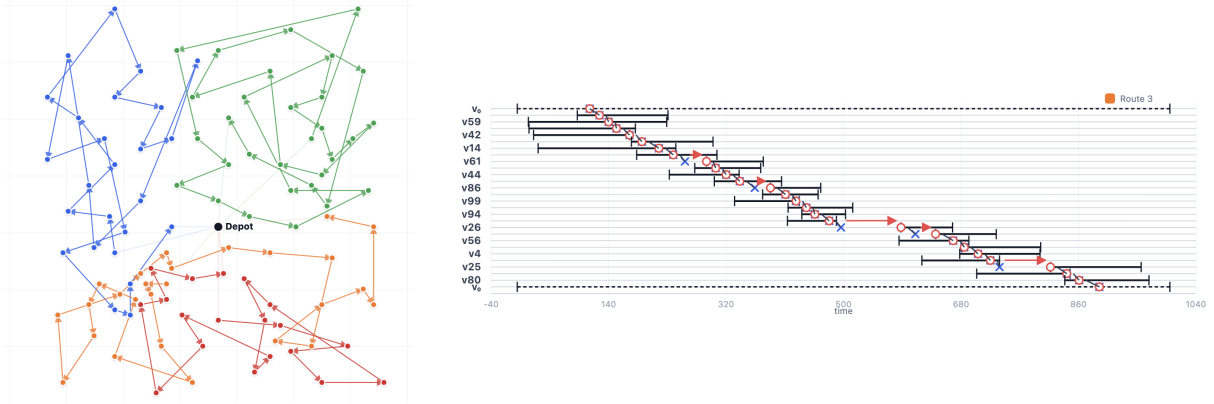
After analyzing several cases, we concluded that the hierarchical objective is the better choice. Waiting at a customer before its window opens costs nothing when only distance is minimized. As a result, distance minimization sometimes yields solutions with many routes that spend much of the day waiting. Instance R201, with EUC_2D distances, is an example (Table 5 and Figure 4). Its distance-optimal solution uses 8 routes and a total waiting time of 2,828, even with each route leaving the depot at the time that minimizes its waiting. The route shown in Figure 4(a) spends most of its duration waiting: it serves its first customers early and then waits twice for time windows that open much later. The hierarchical objective (4;1,251 $<$ 8;1,141) gives a far more reasonable solution: half the routes and a third of the waiting time, for a distance only 9.6% longer. Its routes, such as the one in Figure 4(b), serve more customers with shorter waits.

So, the newly proposed instances will adopt the hierarchical objective function.

[2] `https://www.sintef.no/projectweb/top/vrptw/`



(a) Distance only: 8 routes



(b) Number of vehicles first, then distance: 4 routes

Figure 4: Optimal solutions of R201 under the two objectives (left) and the schedule of one of their routes (right). In the Gantt charts, a cross marks the arrival at a customer, a circle the start of service, and a red arrow the waiting time until the time window opens.

Table 5: Two optimal solutions of R201 (EUC_2D distances).

Objective	Routes	Distance	Waiting time
Distance only	8	1,141	2,828
# of vehicles first, then distance	4	1,251	929

4 Parameters and initial solutions for the XTW instances

This section reports the characteristics of the 100 XTW instances and an initial solution for each, under the hierarchical objective. Table 6 describes the columns of Table 7, which form three groups: the data inherited from the base X instance, the data added by the generator of Section 3, and the initial solution. These solutions were obtained with quick and loosely controlled runs, on different machines, of the heuristics HGS [3][Vidal et al., 2013, Vidal, 2022] and PyVRP [Wouda et al., 2024] and of also the exact method VRPSolver [Pessoa et al., 2020]. Solutions proved optimal appear in bold in Table 7. Those solutions are only given as a first reference for the new set. Indeed, *the community is likely to submit many improving solutions in the next few weeks!* Scientifically rigorous comparisons of the best VRPTW methods on the Solomon, H&G, XTW and XLTW instances will be provided on February 24, 2027, as part of the effort to obtain the initial XLTW solutions for the competition.

[3]A modified implementation available at `https://github.com/Galgos-PUC-Rio/hgs_advance`.

Table 6: Columns of Table 7.

Group	Column	Description
X instance	Name	Base X instance [Uchoa et al., 2017]
	Dep	Depot position: C (Central), E (Eccentric), R (Random)
	Cust	Customer position: R (Random), C (Clustered), RC (Random-clustered), with the number of seeds in parentheses
	Dem	Demand distribution: U (Unitary), 1–10, 5–10, 1–100, 50–100 (uniform ranges), Q (depending on quadrant), SL (many small values, few large values)
	D	Total demand $\sum_i q_i$
Generator	H	Scheduling horizon
	η	Horizon factor
	ρ (%)	TW density, percentage of customers with a time window
	Width	TW width: N (Narrow), M (Medium), W (Wide), X (Mixed)
	Center	TW center: R (Random), C (Clustered, with the number of clusters in parentheses), B (Bimodal)
	Service	Service time: U (Unit), S (Small), M (Medium), L (Large), X (Mixed)
	TW (%)	Percentage of customers whose time window was shifted
	ST (%)	Percentage of customers whose service time was shrunk
	Cap	Capacity level: T (Tight), L (Loose), N (No capacity)
	ϕ	Capacity factor (– under No capacity)
	$\widehat{\kappa}$	Predicted number of customers per route in the uncapacitated instance
	Q	Vehicle capacity
Solution	Routes	Number of routes
	Distance	Total distance
	n/route	Average number of customers per route

Table 7: XTW instances and initial solutions. The columns are described in Table 6.

	X instance					Generator												Solution		
Instance	Name	Dep	Cust	Dem	D	H	η	ρ (%)	Width	Center	Service	TW (%)	ST (%)	Cap	ϕ	$\widehat{\kappa}$	Q	Routes	Distance	n/route
XTW-n101	X-n101-k25	R	RC (7)	1–100	5,147	3,498	2.0	60	X	B	X	0.0	0.0	T	0.9	13.7	634	**9**	**14,427**	11.1
XTW-n106	X-n106-k14	E	C (3)	50–100	7,864	7,912	3.0	80	W	R	L	0.0	0.0	L	1.4	22.4	2,348	**6**	**14,186**	17.5
XTW-n110	X-n110-k13	C	R	5–10	816	1,747	1.3	100	N	C (3)	S	7.3	0.0	N	–	4.5	816	**27**	**27,573**	4.0
XTW-n115	X-n115-k10	C	R	SL	1,535	2,174	1.6	60	M	R	U	0.0	0.0	L	1.4	17.6	332	**8**	**14,623**	14.2
XTW-n120	X-n120-k6	E	RC (8)	U	119	10,568	4.0	80	X	C (5)	M	0.0	0.0	N	–	27.0	119	**4**	**18,326**	29.8
XTW-n125	X-n125-k30	R	C (5)	Q	5,536	3,002	1.3	100	N	B	U	58.1	0.0	T	0.9	7.6	305	**41**	**81,129**	3.0
XTW-n129	X-n129-k18	E	RC (8)	1–10	665	4,207	1.6	60	M	C (5)	L	0.0	0.0	N	–	9.5	665	**11**	**22,347**	11.6
XTW-n134	X-n134-k13	R	C (4)	Q	8,220	5,175	3.0	80	W	B	S	0.0	0.0	L	1.4	44.8	3,880	**3**	**6,665**	44.3
XTW-n139	X-n139-k10	C	R	5–10	1,039	5,259	4.0	100	W	R	M	0.0	0.0	T	0.9	36.8	250	**5**	**14,996**	27.6
XTW-n143	X-n143-k7	E	R	1–100	7,475	5,524	2.0	100	M	B	X	0.0	0.0	N	–	12.2	7,475	**10**	**25,789**	14.2
XTW-n148	X-n148-k46	R	RC (7)	1–10	817	3,503	2.0	80	X	C (4)	U	2.0	0.0	L	1.4	22.3	174	7	19,744	21.0
XTW-n153	X-n153-k22	C	C (3)	SL	3,068	5,419	4.0	60	N	R	L	0.0	0.0	T	0.9	18.1	328	**10**	**14,751**	15.2
XTW-n157	X-n157-k13	R	C (3)	U	156	2,479	1.6	80	M	C (3)	S	8.3	0.0	T	0.9	12.3	12	**19**	**27,220**	8.2
XTW-n162	X-n162-k11	C	RC (8)	50–100	12,194	4,000	3.0	60	N	B	M	0.0	0.0	L	1.4	13.6	1,444	**10**	**21,213**	16.1
XTW-n167	X-n167-k10	E	R	5–10	1,236	3,548	1.3	100	W	R	X	0.0	0.0	N	–	14.9	1,236	11	24,293	15.1
XTW-n172	X-n172-k51	C	RC (5)	Q	8,092	1,720	1.3	100	X	R	S	0.0	0.0	L	1.4	11.9	792	**17**	**22,957**	10.1
XTW-n176	X-n176-k26	E	R	SL	3,632	4,278	1.6	60	M	B	X	2.3	1.1	N	–	12.8	3,632	**11**	**32,558**	15.9
XTW-n181	X-n181-k23	R	C (6)	U	180	2,981	2.0	80	X	C (5)	U	1.1	0.0	T	0.9	21.6	20	9	15,418	20.0
XTW-n186	X-n186-k15	R	R	50–100	13,853	7,038	3.0	80	N	B	M	0.0	0.0	N	–	15.1	13,853	**11**	**29,269**	16.8
XTW-n190	X-n190-k8	E	C (3)	1–10	1,043	9,666	4.0	60	W	C (5)	L	0.0	0.0	L	1.4	21.7	168	9	19,915	21.0
XTW-n195	X-n195-k51	C	RC (5)	1–100	9,208	2,805	2.0	100	N	R	X	0.0	0.5	T	0.9	10.4	443	21	32,957	9.2
XTW-n200	X-n200-k36	R	C (8)	Q	14,263	3,440	1.6	60	X	B	M	6.0	4.0	T	0.9	13.0	837	18	33,014	11.1
XTW-n204	X-n204-k19	C	RC (6)	50–100	15,135	5,621	4.0	100	W	R	U	0.0	0.0	N	–	88.0	15,135	4	15,699	50.8
XTW-n209	X-n209-k16	E	R	5–10	1,547	8,315	3.0	80	M	C (3)	S	0.0	0.0	L	1.4	28.8	301	6	25,128	34.7
XTW-n214	X-n214-k11	C	C (4)	1–100	10,356	1,566	1.3	60	N	B	L	12.7	8.4	L	1.4	6.0	410	**34**	**29,440**	6.3
XTW-n219	X-n219-k73	E	R	U	218	8,173	3.0	100	X	C (4)	L	0.0	0.0	T	0.9	13.8	13	17	38,217	12.8
XTW-n223	X-n223-k34	R	RC (5)	1–10	1,234	8,346	4.0	80	W	R	X	0.0	0.0	N	–	48.8	1,234	7	18,004	31.7
XTW-n228	X-n228-k23	R	C (8)	SL	3,478	2,817	2.0	60	M	C (5)	S	0.0	0.0	L	1.4	17.7	380	10	19,198	22.7
XTW-n233	X-n233-k16	C	RC (7)	Q	10,093	2,131	1.6	80	W	B	M	0.0	0.0	N	–	13.9	10,093	**14**	**20,697**	16.6
XTW-n237	X-n237-k14	E	R	U	236	3,421	1.3	100	M	R	U	0.0	0.0	T	0.9	20.5	19	13	29,447	18.2
XTW-n242	X-n242-k48	E	R	1–10	1,324	11,199	4.0	100	N	B	M	0.0	0.0	T	0.9	17.3	86	16	39,656	15.1
XTW-n247	X-n247-k50	C	C (4)	SL	6,200	1,996	1.6	80	X	R	S	0.0	0.0	N	–	20.4	6,200	12	16,431	20.5
XTW-n251	X-n251-k28	R	RC (3)	5–10	1,867	2,652	1.3	60	M	C (4)	U	7.2	0.0	L	1.4	17.1	179	**22**	**35,691**	11.4
XTW-n256	X-n256-k16	C	C (8)	50–100	19,514	2,411	2.0	60	W	C (4)	L	0.0	0.0	L	1.4	12.9	1,386	19	22,078	13.4

	X instance					Generator												Solution		
Instance	Name	Dep	Cust	Dem	D	H	η	ρ (%)	Width	Center	Service	TW (%)	ST (%)	Cap	ϕ	$\widehat{\kappa}$	Q	Routes	Distance	n/route
XTW-n261	X-n261-k13	E	R	1–100	13,381	8,269	3.0	80	N	B	X	0.0	0.4	N	–	13.7	13,381	**15**	**42,820**	17.3
XTW-n266	X-n266-k58	R	RC (6)	5–10	2,015	3,328	1.3	100	X	R	X	0.0	0.4	T	0.9	14.2	98	21	36,455	12.6
XTW-n270	X-n270-k35	C	RC (5)	50–100	20,419	2,228	1.6	60	M	R	M	0.0	0.0	T	0.9	16.9	1,155	18	24,709	14.9
XTW-n275	X-n275-k28	R	C (3)	U	274	2,177	2.0	100	W	B	U	0.0	0.0	N	–	29.1	274	7	12,657	39.1
XTW-n280	X-n280-k17	E	R	SL	3,233	7,966	3.0	80	X	C (4)	L	0.0	0.0	L	1.4	15.8	256	17	43,315	16.4
XTW-n284	X-n284-k15	R	C (8)	1–10	1,527	7,206	4.0	60	N	B	S	0.0	0.0	T	0.9	28.8	140	11	20,626	25.7
XTW-n289	X-n289-k60	E	RC (7)	Q	15,974	4,495	1.6	80	M	R	U	0.0	0.0	N	–	36.1	15,974	7	24,877	41.1
XTW-n294	X-n294-k50	C	R	1–100	14,135	2,740	2.0	100	X	C (3)	M	4.1	1.7	L	1.4	12.8	866	23	35,259	12.7
XTW-n298	X-n298-k31	R	R	1–10	1,689	5,337	3.0	80	W	C (3)	L	0.0	0.0	L	1.4	20.8	166	17	27,777	17.5
XTW-n303	X-n303-k21	C	C (8)	1–100	15,967	1,612	1.3	100	N	B	S	29.5	14.9	T	0.9	6.7	318	**66**	**70,719**	4.6
XTW-n308	X-n308-k13	E	RC (6)	SL	3,118	10,798	4.0	60	M	R	X	0.0	0.0	N	–	42.8	3,118	9	27,207	34.1
XTW-n313	X-n313-k71	R	RC (3)	Q	17,545	8,257	4.0	80	X	C (5)	U	0.0	0.0	L	1.4	74.1	5,834	5	22,388	62.4
XTW-n317	X-n317-k53	E	C (4)	U	316	3,312	1.3	100	N	R	M	0.0	0.0	N	–	10.2	316	22	45,253	14.4
XTW-n322	X-n322-k28	C	R	50–100	24,003	4,035	3.0	60	W	B	X	0.0	0.0	T	0.9	27.4	1,846	14	21,347	22.9
XTW-n327	X-n327-k20	R	RC (7)	5–10	2,452	4,642	2.0	100	M	C (5)	S	0.0	0.0	T	0.9	21.9	149	17	29,536	19.2
XTW-n331	X-n331-k15	E	R	U	330	4,345	1.6	60	X	R	L	0.0	0.0	N	–	14.8	330	24	52,495	13.8
XTW-n336	X-n336-k84	E	R	Q	16,921	4,493	1.6	80	N	B	S	8.7	3.9	L	1.4	12.2	867	30	71,687	11.2
XTW-n344	X-n344-k43	C	RC (7)	5–10	2,602	1,780	1.3	80	W	B	U	0.3	0.0	N	–	17.7	2,602	16	24,376	21.4
XTW-n351	X-n351-k40	C	C (3)	1–100	17,317	4,947	4.0	100	N	C (3)	X	0.0	0.0	L	1.4	18.6	1,287	22	26,673	15.9
XTW-n359	X-n359-k29	E	RC (7)	1–10	1,946	8,319	3.0	60	W	R	L	0.0	0.0	T	0.9	32.2	158	18	37,117	19.9
XTW-n367	X-n367-k17	R	C (4)	SL	3,647	3,495	2.0	100	X	B	M	0.0	0.0	T	0.9	15.4	139	27	36,537	13.6
XTW-n376	X-n376-k94	E	R	U	375	10,942	4.0	80	M	C (5)	U	0.0	0.0	N	–	90.1	375	5	26,569	75.0
XTW-n384	X-n384-k52	R	R	50–100	29,157	4,142	2.0	60	X	R	X	0.0	0.0	L	1.4	24.8	2,646	17	33,519	22.5
XTW-n393	X-n393-k38	C	RC (5)	5–10	2,919	1,810	1.3	60	N	R	M	0.0	0.0	L	1.4	10.2	107	33	39,962	11.9
XTW-n401	X-n401-k29	E	C (6)	Q	21,275	4,441	1.6	100	M	C (5)	L	1.8	1.2	N	–	7.7	21,275	35	79,453	11.4
XTW-n411	X-n411-k19	R	C (5)	SL	3,901	4,889	3.0	80	W	B	S	0.0	0.0	T	0.9	50.1	429	10	15,574	41.0
XTW-n420	X-n420-k130	C	RC (3)	1–10	2,325	2,210	1.6	60	N	C (5)	S	3.3	1.9	T	0.9	11.2	57	41	44,645	10.2
XTW-n429	X-n429-k61	R	R	50–100	32,261	3,583	2.0	100	M	B	U	0.5	0.0	N	–	28.4	32,261	15	34,671	28.5
XTW-n439	X-n439-k37	C	RC (8)	U	438	3,940	3.0	80	X	R	X	0.0	0.0	L	1.4	29.1	41	18	32,862	24.3
XTW-n449	X-n449-k29	E	R	1–100	22,439	3,561	1.3	80	W	B	M	0.7	0.2	N	–	15.2	22,439	24	55,839	18.7
XTW-n459	X-n459-k26	C	C (4)	Q	28,435	4,923	4.0	60	N	R	L	0.0	0.0	L	1.4	21.8	1,893	23	30,996	19.9
XTW-n469	X-n469-k138	E	R	50–100	35,171	10,926	4.0	100	X	C (3)	S	0.0	0.0	T	0.9	40.0	2,709	13	40,211	36.0
XTW-n480	X-n480-k70	R	C (8)	5–10	3,598	5,937	3.0	100	M	C (3)	U	0.0	0.0	N	–	52.4	3,598	7	28,181	68.4
XTW-n491	X-n491-k59	R	RC (6)	1–100	24,957	4,070	2.0	60	W	B	X	0.0	0.0	T	0.9	25.3	1,161	22	33,135	22.3
XTW-n502	X-n502-k39	E	C (3)	U	501	3,495	1.6	80	N	R	L	0.0	1.0	L	1.4	7.6	11	46	87,513	10.9
XTW-n513	X-n513-k21	C	RC (4)	1–10	2,905	1,755	1.3	60	W	B	M	0.2	0.2	T	0.9	16.4	84	35	38,705	14.6
XTW-n524	X-n524-k153	R	R	SL	17,067	3,141	1.6	100	M	R	M	0.0	0.0	N	–	19.3	17,067	24	44,474	21.8

	X instance					Generator												Solution		
Instance	Name	Dep	Cust	Dem	D	H	η	ρ (%)	Width	Center	Service	TW (%)	ST (%)	Cap	ϕ	$\widehat{\kappa}$	Q	Routes	Distance	n/route
XTW-n536	X-n536-k96	C	C (7)	Q	35,423	2,723	2.0	80	X	C (4)	L	4.9	7.8	L	1.4	10.6	980	58	64,534	9.2
XTW-n548	X-n548-k50	E	R	U	547	3,591	1.3	60	N	B	S	14.8	6.4	L	1.4	12.2	18	**68**	**138,797**	8.0
XTW-n561	X-n561-k42	C	RC (7)	1–10	3,055	4,208	3.0	80	W	C (5)	X	0.0	0.0	T	0.9	27.0	133	23	32,854	24.3
XTW-n573	X-n573-k30	E	C (3)	SL	6,172	8,766	4.0	100	X	R	U	0.0	0.0	N	–	139.4	6,172	3	23,954	190.7
XTW-n586	X-n586-k159	R	RC (4)	5–10	4,430	3,778	1.6	100	M	C (5)	M	1.0	0.0	L	1.4	14.6	155	30	56,383	19.5
XTW-n599	X-n599-k92	R	R	50–100	44,741	4,173	2.0	80	N	B	L	1.7	2.2	T	0.9	8.8	592	76	105,922	7.9
XTW-n613	X-n613-k62	C	R	1–100	32,002	1,832	1.3	60	W	R	S	0.0	0.0	N	–	26.1	32,002	24	33,618	25.5
XTW-n627	X-n627-k43	E	C (5)	5–10	4,713	11,205	4.0	100	X	R	X	0.0	0.2	N	–	44.1	4,713	17	41,518	36.8
XTW-n641	X-n641-k35	E	RC (8)	50–100	47,511	8,268	3.0	80	M	C (4)	U	0.0	0.0	T	0.9	74.8	4,997	10	36,786	64.0
XTW-n655	X-n655-k131	C	C (4)	U	654	2,107	1.6	60	X	B	M	1.5	1.4	L	1.4	16.1	23	31	36,780	21.1
XTW-n670	X-n670-k130	R	R	SL	16,214	7,765	4.0	60	M	R	S	0.0	0.0	T	0.9	70.3	1,535	11	37,038	60.8
XTW-n685	X-n685-k75	C	RC (6)	Q	30,299	4,009	3.0	80	N	C (4)	L	0.1	0.0	L	1.4	10.5	650	71	76,633	9.6
XTW-n701	X-n701-k44	E	RC (7)	1–10	3,797	5,492	2.0	100	W	B	X	0.0	0.0	N	–	18.7	3,797	26	60,637	26.9
XTW-n716	X-n716-k35	R	C (3)	1–100	34,266	2,570	1.3	60	N	C (3)	U	9.1	0.0	N	–	21.1	34,266	59	85,970	12.1
XTW-n733	X-n733-k159	C	R	1–10	3,969	2,228	1.6	80	W	B	S	0.0	0.0	L	1.4	22.1	168	25	39,737	29.3
XTW-n749	X-n749-k98	R	C (8)	1–100	38,414	4,103	3.0	100	M	R	M	0.0	0.0	T	0.9	32.5	1,502	26	39,298	28.8
XTW-n766	X-n766-k71	E	RC (7)	SL	11,695	3,607	1.3	100	X	C (5)	X	5.2	1.3	N	–	10.9	11,695	58	123,877	13.2
XTW-n783	X-n783-k48	R	R	Q	39,413	10,165	4.0	60	X	R	U	0.0	0.0	L	1.4	179.6	12,676	6	43,636	130.3
XTW-n801	X-n801-k40	E	R	U	800	5,571	2.0	80	W	B	L	0.0	0.0	T	0.9	16.9	16	50	93,955	16.0
XTW-n819	X-n819-k171	C	C (6)	50–100	60,914	5,542	4.0	100	M	B	L	0.0	0.0	N	–	17.4	60,914	47	62,364	17.4
XTW-n837	X-n837-k142	R	RC (7)	5–10	6,211	3,742	1.6	80	N	R	S	0.0	0.0	T	0.9	19.2	129	49	84,625	17.1
XTW-n856	X-n856-k95	C	RC (3)	U	855	2,808	2.0	60	N	C (4)	M	1.3	0.0	L	1.4	12.4	18	48	64,432	17.8
XTW-n876	X-n876-k59	E	C (5)	1–100	44,534	7,141	3.0	80	X	B	X	0.0	0.0	T	0.9	22.1	1,014	44	85,707	19.9
XTW-n895	X-n895-k37	R	R	50–100	67,057	2,875	1.3	100	M	R	U	0.0	0.0	N	–	34.1	67,057	22	50,731	40.6
XTW-n916	X-n916-k207	E	RC (6)	5–10	6,816	11,093	4.0	60	W	C (4)	M	0.0	0.0	L	1.4	48.9	511	19	48,209	48.2
XTW-n936	X-n936-k151	C	R	SL	20,743	4,073	3.0	80	W	C (5)	S	0.0	0.0	L	1.4	42.0	1,304	20	41,015	46.8
XTW-n957	X-n957-k87	R	RC (4)	U	956	2,252	1.3	60	M	R	L	0.0	0.0	N	–	14.9	956	76	90,284	12.6
XTW-n979	X-n979-k58	E	C (6)	Q	57,369	4,518	1.6	100	N	B	U	19.9	0.0	T	0.9	24.7	1,305	89	209,992	11.0
XTW-n1001	X-n1001-k43	R	R	1–10	5,557	5,074	2.0	60	X	B	X	0.3	0.4	N	–	20.6	5,557	39	80,803	25.6

Table 8 summarizes the average number of customers per route of these solutions by parameter value, and also for the 50 smallest and the 50 largest instances. The parameters affect the means and medians of route size largely as one would predict. Routes get longer as the horizon factor grows, with means from 13.9 at $\eta = 1.3$ to 47.2 at $\eta = 4$. They also get longer with wider windows, with medians of 11.9, 19.5 and 22.9 for Narrow, Medium and Wide, although the mean of Medium is slightly above that of Wide. They get shorter with longer service times, with means of 46.2, 24.0, 19.5 and 14.0 from Unit to Large. Windows concentrated around peak times give shorter routes than Random ones, and capacity shortens routes, most of all at the Tight level. The possible exception is the TW density: route size does not decrease as ρ grows, with means of 22.5, 26.1 and 26.2 and the highest median at $\rho = 80\%$. The mean route size is considerably larger in the 50 largest instances: 30.2 customers, against 19.6 in the 50 smallest (the medians are 20.5 and 16.3). This is expected. Since the customers of all instances lie in the same grid, a higher density of customers reduces the travel time from one customer to another with a compatible time window.

Table 8: Average number of customers per route of the initial solutions, by parameter value and by instance size.

Parameter	Value	Instances	Mean	Median	Min	Max
Horizon factor η	1.3	20	13.9	12.6	3.0	40.6
	1.6	20	16.6	14.6	8.2	41.1
	2.0	20	18.7	18.5	7.9	39.1
	3.0	20	28.2	21.4	9.6	68.4
	4.0	20	47.2	32.9	15.1	190.7
TW density ρ (%)	60	34	22.5	16.0	6.3	130.3
	80	33	26.1	20.0	7.9	75.0
	100	33	26.2	15.9	3.0	190.7
TW width	Narrow	25	12.5	11.9	3.0	25.7
	Medium	25	28.4	19.5	8.2	75.0
	Wide	25	26.9	22.9	13.4	50.8
	Mixed	25	31.9	20.0	9.2	190.7
TW center	Random	33	31.4	19.9	9.2	190.7
	Clustered	33	24.9	17.8	4.0	75.0
	Bimodal	34	18.7	16.7	3.0	44.3
Service time	Unit	20	46.2	33.8	3.0	190.7
	Small	20	24.0	21.6	4.0	60.8
	Medium	20	19.5	16.7	11.1	48.2
	Large	20	14.0	13.6	6.3	21.0
	Mixed	20	20.8	21.1	9.2	36.8
Capacity level	Tight	33	20.0	16.0	3.0	64.0
	Loose	33	24.2	17.5	6.3	130.3
	No capacity	34	30.4	21.0	4.0	190.7
Size n	100–330	50	19.6	16.3	3.0	62.4
	335–1,000	50	30.2	20.5	7.9	190.7

References

Nicos Christofides, Aristide Mingozzi, and Paolo Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial Optimization*, pages 315–338. Wiley, 1979.

Jörg Homberger and Hermann Gehring. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 162(1):220–238, 2005.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183(1–2):483–523, 2020.

Eduardo Queiroga, Rafael Martinelli, Anand Subramanian, Eduardo Uchoa, and Thibaut Vidal. The XL instances and the CVRPLib best known solution challenge. arXiv preprint arXiv:2601.11467, 2026.

Marius M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017.

Thibaut Vidal. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers & Operations Research*, 140:105643, 2022.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1):475–489, 2013.

Niels A. Wouda, Leon Lan, and Wouter Kool. PyVRP: A high-performance VRP solver package. *INFORMS Journal on Computing*, 36(4):943–955, 2024.