

An Open-Source Data Generator for Joint Air Traffic Flow and Capacity Management Benchmarking

Alexander Beiser ^{*,1} Thomas Gräupl ² Georg Trausmuth,² and Stefan Woltran ¹

¹TU Wien, Vienna, Austria

²Frequentis AG, Vienna, Austria

*Corresponding author: alexander.beiser@tuwien.ac.at

Abstract

Data in Air Traffic Flow and Capacity Management (ATFCM) is scarce. This hinders the development and evaluation of algorithms and simulations that improve on the state of the art or expose its deficiencies. To ease development and comparison on the computational side, we present a scalable generator of benchmark data for joint ATFCM — a recent formal model that combines delaying and rerouting aircraft with dynamic airspace configuration. For fidelity, the generator draws on three sources: the OpenSky Network COVID CC-BY dataset for flight data, OurAirports for airport locations, and X-Plane data for en-route navpoints. In this paper we present a user guide to the generator and to our two published datasets, which go up to 31,622 flights and 18,723 navpoints. Furthermore, we present the inner workings of the generator and measure the generated data from real en-route aircraft counts, over departure counts, to flight durations. At fine time granularity, departure profiles, en-route counts and origin–destination pairs lie within real day-to-day variation in four of six regions.

Keywords: Air Traffic Flow and Capacity Management; Open Data; Data Generator; Benchmark; OpenSky Network

Abbreviations: ATFCM: Air Traffic Flow and Capacity Management, ATM: Air Traffic Management

1. Introduction

Data in *Air Traffic Management (ATM)* research is frequently proprietary for fear of competitive disadvantage. This scarcity of open data is an *open challenge* in *Air Traffic Flow and Capacity Management (ATFCM)* benchmarking, manifesting in a reproducibility crisis and forcing researchers to rely on datasets ill-suited for large-scale algorithmic evaluation. While initiatives like the Open Science Alliance for ATM research [1] aim to mitigate this, accessible ATFCM datasets for benchmarking and comparing algorithms remain scarce. We partially address this gap with an open-source data generator built entirely from publicly available data. It lets researchers sample varied geographic regions and scale flight counts arbitrarily, producing instances (a day of flights on a navigation graph with capacitated sectors, Section 3.2) natively compatible with state-of-the-art exact *Mixed-Integer Programming (MIP)* and *Answer Set Programming (ASP)* solvers. Although the generator was already used in recent papers [2, 3] with two datasets¹, it was so far only sketched and not evaluated.

Contributions. In this paper we provide (Figure 1): (i) *Complete Specification*: an algorithmic and mathematical description of the generator’s stages and a practitioner’s guide to generating instances. (ii) *Instance Description*: the two published datasets, the small-scaling dataset (its first release used in [2]) and the large-scaling dataset of regional to continental instances with up to 31,622 flights and 18,723 navpoints (its first release used in [3]). (iii) *Statistical Analysis*: an analysis of the fidelity of the generated data against historical traffic. As the datasets are meant as benchmarks, we report fidelity descriptively.

¹First release: 10.5281/zenodo.20782934 (small-scaling) and 10.5281/zenodo.21869481 (large-scaling).

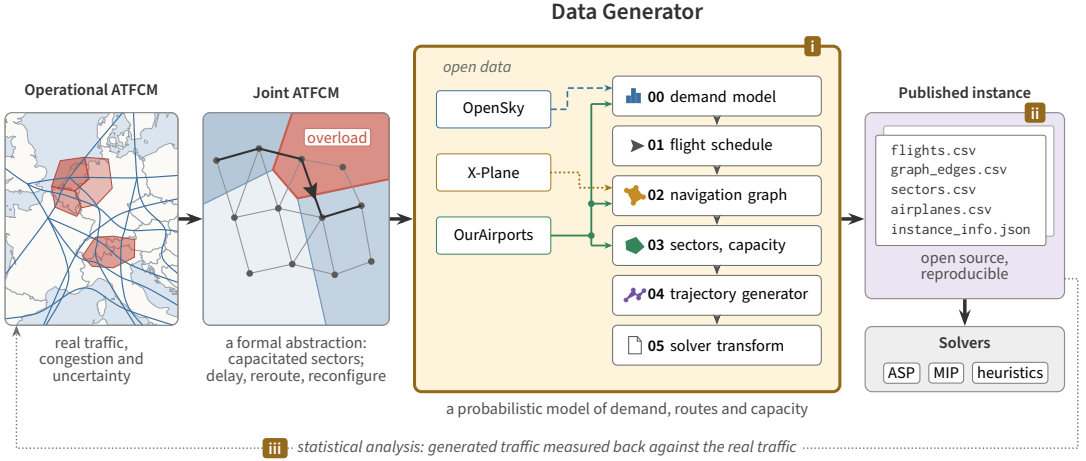


Figure 1. Our contributions (i)–(iii): the generator’s specification, the instances, and their statistical analysis. Operational ATFCM with real flows (blue), crowded sectors (red) and uncertainty; joint ATFCM, a formal model for comparing algorithms on delay, rerouting and sector reconfiguration, needing flight plans, sectors and capacities; and the six-stage generator building them from open data (grid graphs need no X-Plane data and are CC-BY usable). Dotted: the analysis (Section 5).

The large-scaling dataset holds 6,720 instances over eight regions in Europe, North America and East Asia with up to 31,622 flights and graphs of up to 19,537 vertices; the small-scaling dataset holds 200 instances of 10 to 100 flights for exact methods. At one-minute timesteps, departure profiles, en-route counts and origin–destination pairs lie within real day-to-day variation in four of six regions.

2. Related Work

2.1 Aviation Data

The historic scarcity of open data in air traffic management prompted open-data initiatives [4, 1, 5]; open surveillance data is now available through the OpenSky Network [6] and its COVID dataset [7] (we use June 2019, pre-pandemic), with open tooling [8], performance models [9] and simulators [10].

Benchmark data for ATFCM remains scarce: Pelegrín and Cerulli [11] publish a generator of tactical conflict-resolution instances with controllable congestion. Closest to ours is the ATFM-Benchmark set of Guo et al. [12, 13]: eight fixed instances from one national airspace with flight schedules, route lengths and default speeds, but without sectors or capacities. The instances BLO, mBLO, CLO and IC [14, 15, 16], used for benchmarking in the literature, are not publicly available.

Our generator differs on both counts: it generates instances, including the sectors and per-timestep capacities joint ATFCM needs, and is to our knowledge the first open generator for joint ATFCM.

2.2 Synthetic Data Generation

Synthetic data generators serve where real data is unavailable, too small for scaling studies, or insufficient to evaluate a method fully. Smith-Miles and Bowly [17] evolve instances to fill empty regions of a feature-based instance space; Bowly et al. [18] construct linear programs where feasibility holds by design; Geng et al. [19] judge generated mixed-integer programs by *solver behavior* as well as structure; and Kovaleva and Ponomarenko [20] calibrate a transport-network generator on open road graphs, with a distributional comparison and a solver-difficulty study. Synthetic air traffic is generated at the trajectory level, by deep generative models [21, 22, 23, 24] and lighter latent-space or

density-based ones [25, 26, 27], which capture rare patterns poorly [23, 24, 28] and do not guarantee physical plausibility [23]. Tabular schedule data for training delay and turnaround predictors [29, 30] lacks the graph, sectors and capacities of a joint ATFCM instance.

Our generator emits the flights together with the navigation graph, sectors and capacities they are posed against, and sets difficulty by the number of flights, the graph size and the capacity.

2.3 Synthetic Dataset Evaluation

The fidelity of synthetic data is measured by distributional distances, discriminative tests and downstream utility. Theis et al. [31] show these criteria are largely independent and must be judged against the intended application, and the choice of distance itself shapes the conclusion [32]. Lopez-Paz and Oquab [33] train a classifier to separate real from synthetic data; other work tests on real data [34] or separates whether synthetic samples lie inside the real support from whether they cover it [35].

In aviation, Olive et al. [36] propose four metric families and tabulate the realism each use case demands; for ATFCM they rate operational realism *high* and physical realism *low*. Two of their guidelines shape our evaluation: state the use case's expectations before choosing metrics, and do not score a method with a metric it was built to optimize. Our data is meant for developing and comparing algorithms, so we report fidelity descriptively, without an acceptance threshold. We compare departures, en-route counts, origin–destination pairs and more with real traffic, reporting each distance's excess over that of disjoint real samples of matched size, normalized by a null model.

3. A Practitioner's Guide to the Generator

We publish two datasets on Zenodo, the small-scaling and the large-scaling dataset², and the generator on GitHub³. Their READMEs document structure and usage; this section is an overview.

3.1 The two datasets

The large-scaling dataset holds 6,720 instances: 8 regions $\times$ 4 time granularities $\times$ 7 demand levels $\times$ 3 seeds $\times$ 10 capacity levels. Demand, the number of flights $|F|$, runs from 1,000 to 31,622, each level 1.78 times the last; Table 1 lists the regions. Instances sit in `<TIME-GRANULARITY>/<REGION>/<CAPACITY>/<FLIGHTS>_SEED<SEED>/`, where the time granularity $T_{gran} \in \{1, 4, 15, 60\}$ counts timesteps per hour and the capacity level runs from PCAP100 to PCAP010, 100 % to 10 % of nominal capacity. $T_{gran} = 1$ gives 60-minute and $T_{gran} = 60$ one-minute timesteps, which we recommend for realism.

The small-scaling dataset holds 200 instances: 5 grid regions $\times$ 10 demand levels of 10 to 100 flights $\times$ 4 seeds, all at $T_{gran} = 1$. It covers the sizes exact methods close [2], all below the 1,000 flights where the large-scaling dataset starts. Its configurations set cap-enroute 1, one flight per navpoint per timestep (airports keep 60,000; per sector under the max rule), so the instances are already tight and carry no capacity sweep; its directories are `<REGION>/<FLIGHTS>_SEED<SEED>/`. Every filed duration is at least $(hops - 1) \times 60$ min, so delays and durations in these instances are not realistic quantities.

Licensing follows the navpoint source: the four large-scaling regions built on X-Plane waypoints are GPL-2.0-or-later, the four grid regions and the small-scaling dataset CC-BY-4.0; every instance declares its license in `instance_info.json`, and `LICENSING.md` gives the reasoning.

3.2 What an instance is

Each dataset ships two data structures. The *unparsed* tree is the pipeline's output: per region the configuration (`manifest.json`), the demand model (`model/`), the navigation graph (`navgraph/`) and

²Second release: 10.5281/zenodo.23039645 (large-scaling) and 10.5281/zenodo.23039835 (small-scaling).

³GitHub: <https://github.com/alex4123/ASPaeroFlow-DataGenerator>, tag v2.0.0; Zenodo: 10.5281/zenodo.23136727.

Table 1. The eight regions of the large-scaling dataset. *Vertices* (airports plus en-route *navpoints*) is the graph size shared by all instances of a region; *hops* (route positions per flight) and *legs per aircraft* are means at $T_{gran} = 60$ and $|F| = 1,000$. *Faithful* T_{gran} is the coarsest shipped granularity with $\rho \leq 1$ (Equation (5)), at which duration still tracks distance (a dash: none); *model dep./day*, the demand model’s mean daily departures, gives the $|F|$ that matches real traffic.

Region	Graph	Vertices	Airports	Navpoints	Hops	Faithful T_{gran}	Legs/aircraft	Model dep./day
CENTRAL-EUROPE-7x7	Grid	56	7	49	5.0	15	4.3	270.6
USA-EAST-COAST-20x10	Grid	208	8	200	7.0	4	3.3	372.2
MAJOR-EUROPE-40x20	Grid	816	16	800	11.6	15	3.6	1,087.4
EAST-ASIA-40x40	Grid	1,608	8	1,600	14.9	15	3.4	420.3
DACH	Gabriel	1,508	128	1,380	26.6	60	3.5	1,051.2
CENTRAL-EUROPE	Gabriel	637	10	627	25.1	—	3.6	30.8
EUROPE	Gabriel	9,647	936	8,711	41.0	60	2.1	10,270.2
USA-MAINLAND	Gabriel	19,537	814	18,723	65.5	60	2.0	10,009.9

one directory per flight count and seed, all listed file by file in the READMEs. The solver transform turns it into the *parsed* tree of self-contained solver instances; a parsed instance holds the files a solver reads, each identifier a contiguous integer, and metadata:

```
<REGION>/PCAP050/0001000_SEED42/
  flights.csv                                flight plans, as (Flight_ID, Position, Time)
  airplanes.csv, airplane_flight_assignment.csv aircraft speeds; who flies which flight
  graph_edges.csv, airports.csv              edges, as (source, target, dist_m); airport vertices
  sectors.csv, navaid_sector_*.csv           capacity per vertex and timestep; vertex to sector
  instance_info.json, mappings/, ...         metadata, license, names behind the numbers
```

For scale, DACH at $T_{gran} = 60$, $|F| = 1,000$, PCAP050 has 26,311 flight-plan rows, one per flight and waypoint, flown by 294 aircraft over 1,508 vertices, 3,442 edges and 182 sectors; only mappings/vertex_map.csv names the airports.

Three conventions are implicit. *Capacities are per timestep*: the Capacity column of sectors.csv is the number of flights permitted in a sector during one timestep, one minute at $T_{gran} = 60$; the instances of the first release (Appendix 1) state it per hour and need converting (convert_v1_hourly_capacities.py). *sectors.csv is keyed by vertex*: the column is called Sector_ID, but the file has one row per graph vertex, and navaid_sector_assignment.csv maps the vertices onto sectors. The solver picks the rule composing a sector’s capacity from its members (the reference optimizer uses max); results under different rules are not comparable, so a paper has to name its rule. *All instances have a hard 24-hour window*: every flight departs and lands inside $[0, 24 \cdot T_{gran}]$ (Section 6); it binds every solution of the solution.

3.3 Choosing an instance

For solver benchmarking. The large-scaling dataset is a four-dimensional scaling benchmark to draw subsets from: two axes of size, the number of vertices $|V|$ and of flights $|F|$; the capacity level, which sets the congestion; and the time granularity, which sets the length of the time dimension, from the most tractable models at $T_{gran} = 1$ to the most realistic at $T_{gran} = 60$.

Three starting points: the small-scaling dataset for exact methods; DACH at $T_{gran} = 60$, $|F| = 1,000$, PCAP060 for a mid-size case that also supports fidelity claims; and EUROPE or USA-MAINLAND at $T_{gran} = 60$, $|F| = 10,000$, PCAP040 for scale. PCAP100 is overload-free by construction, since nominal capacity is the instance’s own peak occupancy; capacities are floored at 1, so no level is trivially infeasible.

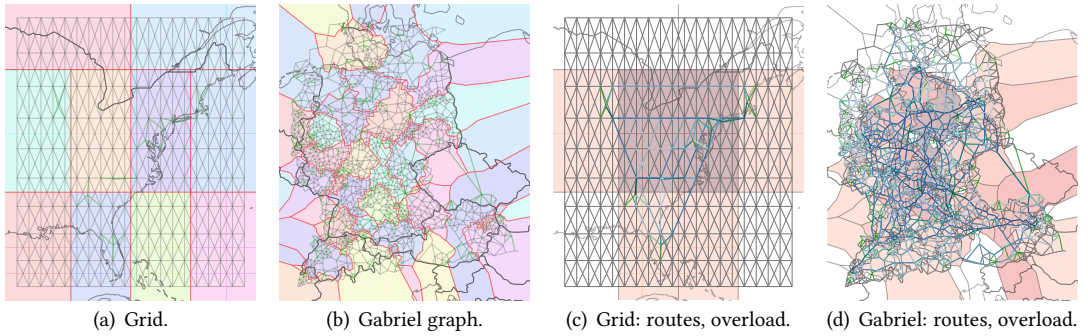


Figure 2. The two navigation graph families at $T_{gran} = 60$: the grid of USA-EAST-COAST- 20×10 and the Gabriel graph of DACH. (a, b) Initial sector partitions: the grid has few, long edges (median 314.5 km) in 18 sectors, the Gabriel graph 1,508 vertices in 182 sectors and many short edges (median 18.3 km). Edge length sets the granularity a region needs, $T_{gran} = 4$ for this grid and 60 for this Gabriel graph (*faithful* T_{gran} in Table 1). (c, d) Routes and initial overload at $|F| = 31,622$ and PCAP010: edge color counts routes, sector fill the overload, each scaled per panel. The grid concentrates traffic on a few long edges and overloads whole sectors; the Gabriel graph spreads it, so its overload follows the traffic.

For claims about real traffic. The datasets are built for benchmarking; for claims about real traffic, two rules apply.

Rule 1: take the granularity from the faithful T_{gran} column of Table 1. It depends on edge length, a property of the graph (Figure 2); below it, duration becomes $(hops - 1) \times timestep$ whatever the distance, and long routes are truncated, which shifts the origin-destination mix (Sections 4.4.4 and 5.5).

Rule 2: only a demand level matching the region's real traffic supports a fidelity claim. Read it from the *model dep./day* column: $|F| = 1,000$ matches MAJOR-EUROPE- 40×20 and DACH, $|F| = 10,000$ matches EUROPE and USA-MAINLAND, and the other four regions have no matching shipped level; any higher level is deliberate over-demand. Matches count traffic between a region's modeled airports.

3.4 Generating your own

Three commands produce a region from scratch:

```
python run_pipeline.py --config configs/04_0_dach_TG60.json
python 05_transform_for_optimizer.py --in-exp-dir <unparsed>/<REGION> --out-root <parsed>
python 06_capacity_sweep.py --exp-dir <parsed>/<REGION> --out-root <overlays>
```

The inputs are the OpenSky COVID dataset [7] and the OurAirports snapshot in every case, plus X-Plane navdata for the real-waypoint regions; all three ship in the large-scaling dataset's third_party/ directory. Setting `--grid-navpoints true` drops the X-Plane dependency, and with it the GPL. A configuration is one JSON file; the lines of 04_0_dach_TG60.json a user changes:

```
"date-start": "2019-06-01", "date-end": "2019-06-30",
"time-granularity": "60", "bin-min": "15", "timezone": "-1",
"criterion": "gabriel", "max-edge-km": "400", "min-dist-vertices-km": "10",
"sector-default-navaid-size": 50, "convex-sectors": "0", "cap-enroute": "1200",
"flight-flights": [1000, 1778, 3162, 5623, 10000, 17782, 31622],
"flight-seeds": [42, 11904657, 150699],
"considered_geographic_regions": [{"region-name": "DACH", "polygon": [...]}]
```

The polygon fixes the region, the dates its month of traffic, time-granularity is T_{gran} and bin-min the demand model's bin width; the graph keys set the topology and therefore the granularity Rule 1 demands, the sector keys the sector structure, cap-enroute the capacity before any sweep, and timezone where the day is cut.

Regenerating the large-scaling dataset from existing demand models and navigation graphs took 51.6 hours over 32 jobs, one per region and granularity, from 20 min (CENTRAL-EUROPE, $T_{gran} = 1$) to 6.4 hours (USA-MAINLAND, $T_{gran} = 60$); in every region $T_{gran} = 1$ is the cheapest. Peak memory stayed below 0.7 GB in every job, so the pipeline runs on a laptop. Downloading one archive per granularity (1.2 to 2.8 GB, 8.0 GB in all) is usually cheaper than generating a missing region or setting.

check_instances.py re-runs the structural checks (window, adjacency of consecutive positions, non-overlapping legs, rotation continuity, sector cover, flight count) over an instance or a whole tree and exits non-zero on a violation. It also reports the share of sectors ever over capacity; zero below PCAP100 means a trivially feasible instance.

Every stage from the demand model (00) to the trajectory generator (04) can be replaced with `--stage-impl <stage>=<module>:<Class>` (the solver transform, 05, takes a bare spec); `--list-stage-impls` prints each stage's contract, the implementation to copy and worked examples, so a stage can be replaced without forking the pipeline.

Tag v2.0.0 reproduces the published instances' demand models and navigation graphs exactly, and their flights except the legs the trajectory generator re-specifies: those draws depended on Python's string-hash seed, which was not pinned during generation, and capacity levels follow from the flights. Determinism across machines is unproven, as the scikit-learn version can change the navigation graph; the trajectory generator rewrites flights.csv, so re-runs start at the schedule.

4. Inner Workings of the Generator

We detail the six-stage data generator of Figure 1 in three parts: (i) the demand model and the flight schedule (airport sending strengths and origin–destination generation), (ii) the navigation graph, and (iii) the auxiliary constructions of the sector and capacity step and the trajectory generator. Their output is deliberately solver-independent, so it can be analyzed and converted to other formats; the last stage, the solver transform, turns it into a solver's input and can be replaced for another solver.

4.1 Airport sending strengths

The general idea behind the airport sending strength is to fit a Poisson model to the departures of an airport, which we derived from an abstract model of neuron sending strength in brains [37]. The OpenSky COVID dataset [7] is a set $\mathcal{F}$ of observed flights f with origin o_f , destination d_f , an aircraft identifier (aircraft_uid, one value per physical aircraft), and the UTC times t_f^f , t_f^l of first and last surveillance contact respectively. The modeled airports A are the OurAirports entries with a four-letter ICAO code, a facility type listed in airport-types and a position inside the region polygon $\mathcal{P}$, intersected with the airport-include list where one is given. The demand model is fitted on the calendar days $\mathcal{D}$ from date-start to date-end (June 2019, $|\mathcal{D}| = 30$) and keeps the observed flights $\mathcal{F}_{\langle A, \mathcal{D} \rangle} = \{f \mid f \in \mathcal{F}, \text{date}_{\text{UTC}}(t_f^f) \in \mathcal{D}, o_f, d_f \in A, o_f \neq d_f\}$ dropping round trips, which a point-to-point instance cannot represent.

Localization. Let B be the set of time bins, where Δ is the width of a bin in minutes and $1440 = 24 \cdot 60$ is the number of minutes per day, i.e., $B = \{\lceil m/\Delta \rceil - 1 \mid m \in \{1, \dots, 1440\}\}$. Before binning, both contact times t_f^f and t_f^l are shifted by a fixed offset of τ hours (timezone), and timestep 0 of an instance is midnight in that local frame. Let $\tilde{t}_f = t_f^f + \tau$ be the shifted first-seen time, μ be the minute of the day, and let $\text{day}(f)$ be the date of $\tilde{t}_f$, then let $\text{bin}(f) = \lfloor \mu(\tilde{t}_f)/\Delta \rfloor \in B$. The sending strength of airport a in bin $b \in B$ is the mean number of departures per day,

$$\lambda_{a,b} = \frac{|\{f \mid f \in \mathcal{F}_{\langle A, \mathcal{D} \rangle}, o_f = a, \text{bin}(f) = b, \text{day}(f) \in \mathcal{D}\}|}{|\mathcal{D}|} \quad (1)$$

Note that while $\mathcal{F}_{\langle A, \mathcal{D} \rangle}$ selects flights by UTC date, $\lambda_{a,b}$ attributes them by local date. This leads to cut-offs at the borders of $\mathcal{D}$. In practice, we set τ by a region's hour of least demand.

Departure sampling. For an instance with the set of flights F , the generator samples departure counts $k_{a,b} \sim \text{Pois}(\lambda_{a,b})$ over all airports and bins until at least $|F|$ departures exist, deletes a uniformly random subset of the surplus, and gives each departure f of (a, b) the departure time $t_f = (b + U)\Delta$ minutes after local midnight, $U \sim \text{Unif}[0, 1)$.

4.2 Origin–destination generation

For each sampled departure, a destination is drawn in proportion to how often that route was observed at that time of day. Let $n_{o,b,d} = |\{f \in \mathcal{F}_{\langle A, \mathcal{D} \rangle} \mid o_f = o, \text{bin}(f) = b, d_f = d\}|$ count those flights over all days, $\mathcal{A}_{o,b} = \{d \mid n_{o,b,d} > 0\}$ be the destinations served from o in bin b , and $\alpha = 0.5$ flatten the draw over it; a flight departing o in bin b then draws from

$$\Pr(d \mid o, b) = \frac{n_{o,b,d} + \alpha}{\sum_{d' \in \mathcal{A}_{o,b}} (n_{o,b,d'} + \alpha)} \quad \text{for } d \in \mathcal{A}_{o,b}, \quad \Pr(d \mid o, b) = 0 \text{ otherwise.} \quad (2)$$

The destination draw has two fallbacks: (i) $\Pr(\cdot \mid o, b')$ for the lowest bin b' with $\mathcal{A}_{o,b'} \neq \emptyset$ when $\mathcal{A}_{o,b} = \emptyset$, and (ii) the global destination marginal $q(d)$, the share of flights landing at d , when o has no departures at all. Since $\lambda_{a,b} > 0$ implies $\mathcal{A}_{a,b} \neq \emptyset$, neither fallback is reached unless the optional smoothing of Equation (1) is on, which no released configuration sets; α applies in every release. This model improves on the first release, used by Beiser et al. [2, 3], in two respects: (i) the first release's demand model discarded every flight that crossed UTC midnight, and (ii) it drew every destination from the marginal q , independent of origin and time (Appendix 1).

4.3 Navigation graph

The navigation graph $G = (V, E)$ has airports and navpoints as vertices: $V = A \cup N$ with $A \cap N = \emptyset$.

Real-waypoint mode. N holds the fixes of X-Plane's `fix.dat` and the NDB, VOR and DME navaids of its `nav.dat` that lie inside $\mathcal{P}$. A fix is labeled by its identifier and a navaid by the first word of its name, and entries sharing a label collapse into one. The collapse runs over the whole world file before $\mathcal{P}$ is applied. A navpoint's identifier is its label with the flight level appended; airports keep their ICAO code. Airports lie at altitude 0 and navpoints at the flight level `altitude` (FL300 in seven of the eight large-scaling regions). For vertices x, y with altitudes h_x, h_y and central angle $\gamma(x, y)$, and Earth radius $R_\oplus = 6,371,008.8$ m, we approximate the distance $\delta(x, y)$ by the hypotenuse of the arc at their mean altitude and their altitude difference, and the horizontal distance by the arc alone:

$$\delta(x, y) = \sqrt{(\gamma(x, y)(R_\oplus + \frac{h_x + h_y}{2}))^2 + (h_x - h_y)^2}, \quad \delta_{\text{hor}}(x, y) = \gamma(x, y)R_\oplus. \quad (3)$$

A separation filter keeps navpoints more than r_{min} (min-dist-vertices-km) apart: it visits the vertices in a fixed order, airports first, and drops navpoint y if an earlier retained x has $\delta_{\text{hor}}(x, y) \leq r_{\text{min}}$. Edges are created according to the Gabriel criterion [38], which connects two vertices if no third vertex lies inside the disc they span. Only pairs at most ℓ_{max} (max-edge-km) apart are candidates, which bounds the edge length and keeps the test cheap:

$$E = \{\{x, y\} \mid x, y \in V, \delta(x, y) \leq \ell_{\text{max}} \text{ and no } z \in V \setminus \{x, y\} \text{ has } \delta(x, z)^2 + \delta(y, z)^2 < \delta(x, y)^2\} \quad (4)$$

We ensure that (V, E) is connected by first analyzing the connected components and then iteratively joining the component of the first vertex to another one by the horizontally closest vertex pair between them; these edges may exceed ℓ_{max} .

Grid mode (grid-navpoints) reads no X-Plane data. The bounding box of $\mathcal{P}$, latitudes $[\phi_S, \phi_N]$ by longitudes $[\psi_W, \psi_E]$, is cut into n_y rows and n_x columns of tiles (the two numbers in a grid region's

name), and the navpoints are the tile centers $g_{ij} = (\phi_S + (i + \frac{1}{2})\frac{\phi_N - \phi_S}{n_y}, \psi_W + (j + \frac{1}{2})\frac{\psi_E - \psi_W}{n_x})$ for row $i < n_y$ and column $j < n_x$, adjacent when $\max(|i - i'|, |j - j'|) = 1$. Each airport joins those of its four horizontally nearest navpoints whose δ is at most $\ell_{\max}$ (350 km by default), and the horizontally nearest in any case; there is no separation filter and no connectivity step.

4.4 Auxiliary constructions

In the following, we present our implementations of sectors, capacities, and trajectories.

4.4.1 Sectors and capacities

The sector and capacity generation step makes every airport its own (atomic) sector and partitions the navpoints N into connected sectors of target size m (sector-default-navaid-size), grown by breadth-first search over the navpoint subgraph $G[N]$ from the first unassigned navpoint; $S(x)$ is the sector of x . With convex-sectors 1 (grids, CENTRAL-EUROPE) a candidate y is accepted only if every vertex on every shortest path in $G[N]$ from y to a current member is unassigned or in the sector, and those vertices join with y , so a sector can exceed m slightly.

Capacities are integers $c(x)$ per vertex in `sectors.csv`, bounding how many flights a sector may hold in one timestep; how a sector combines its vertices' values is up to the solver (Section 3.2). Airport and en-route defaults are configurable; the sweep overwrites both in the large-scaling instances.

4.4.2 Trajectory generator: filed flight plans and aircraft turnarounds

The trajectory generator turns each sampled origin-destination pair into a simplified trajectory, a path through G with integer timesteps, $((x_0, t_0), \dots, (x_K, t_K))$. Between entries j and $j + 1$ the flight counts in the sector of the nearer one, the rule of the reference optimizer: its sector $S_f(t)$ is $S(x_j)$ up to timestep $t_j + \lfloor (t_{j+1} - t_j)/2 \rfloor$ and $S(x_{j+1})$ after it.

Routing. A timestep lasts $\sigma = 60/T_{\text{gran}}$ minutes and the day window $T = \{0, \dots, W\}$ ends at timestep $W = 24 T_{\text{gran}}$. At speed v an edge e of length $\ell_e = \delta(e)$ (dist_m, unrounded) takes ℓ_e/v minutes, charged in whole timesteps and at least one: $w_v(e) = \max(1, \lceil \ell_e/(v\sigma) \rceil)$, with ℓ_e in meters and v in meters per minute, converted from the knots of `airplanes.csv`. Filed flights follow a least-cost path (Dijkstra) under w_v to its destination, with $t_j = t_{j-1} + w_v(x_{j-1}x_j)$.

Aircraft and turnaround times. A single aircraft may fly multiple flights (legs) per day. Each aircraft flies its legs in order; a leg departs from the airport where its aircraft stands. Flight turnaround times are drawn from empirical distributions of the observed flights: the pool Θ of 0 to 60 min between consecutive flights of one aircraft. Aircraft are assigned by a simple rule: a ready aircraft at airport a takes the next departure from a . An aircraft becomes ready at airport a in two ways: (i) a flight has landed at a and its turnaround time is finished, or (ii) no aircraft is ready at a , and a new one is spawned. Each leg draws a duration from its origin-destination pair and the first leg's draw fixes the aircraft's speed, $v = 430, 450$ or 480 kt. The trajectory generator recomputes durations from the route and starts a leg at the timestep of its sampled departure, $\lfloor t_f/\sigma \rfloor$, or one timestep after its aircraft's previous leg lands if that is later.

The 24-hour window. A leg whose destination is unreachable or whose route would end after W ends its aircraft's day: it and all later legs of that aircraft are dropped, and the aircraft stays parked. Each dropped leg is re-sampled, so the instance keeps $|F|$ flights: as a further leg of the parked aircraft to a reachable destination drawn by the sampled destination frequencies, or else as the first leg of a new aircraft; if neither fits, generation aborts.

Invariants. For consecutive legs f, f' of one aircraft, every entry lies inside the window, the legs are at least one timestep apart, and f' departs from the airport where f landed: (i) $0 \leq t_j \leq W$, (ii)

$t_0(f') \geq t_K(f) + 1$, and (iii) $x_0(f') = x_K(f)$. All three hold by construction and are re-tested by `check_instances.py`. The realized ground time $t_0(f') - t_K(f)$ is therefore at least one timestep and does not necessarily reproduce the drawn turnaround.

4.4.3 Nominal capacity

The nominal capacity is the busiest sector-timestep the filed plans produce, airport sectors included, and level $p \in \{0.1, 0.2, \dots, 1.0\}$ (PCAP010 to PCAP100) scales it to p at every vertex, never below one, so at $p = 1$ the plans are overload-free. With the load $L(s, t) = |\{f \in F \mid S_f(t) = s\}|$ of sector s at timestep t , they are $c^* = \max_{s,t} L(s, t)$ and $c_p(x) = \max(1, \lceil p c^* \rceil)$ for every vertex x . Only `sectors.csv` differs between levels; all rows equal, a maximum or mean composition gives each sector c_p . Since c^* depends on $|F|$, the seed and T_{gran} , a level has a fixed meaning only within one granularity.

4.4.4 Edge traversal time and time granularity

Because discrete edge traversal times are rounded up, a filed duration can exceed the flying time, by an amount that grows with the timestep duration σ . Take a flight at speed v whose route has K edges of total length ℓ_f , so flying time ℓ_f/v minutes and filed duration $t_K - t_0$ timesteps. Each edge costs at least one timestep, at least its flying time, and less than one timestep more; summing over the route gives $\max(K, \ell_f/(v\sigma)) \leq t_K - t_0 \leq K + \ell_f/(v\sigma)$.

If no edge of the route is longer than $v\sigma$, the duration is exactly K timesteps whatever ℓ_f is; hence, routing can degenerate to a minimization of the edge count. A plan has $K + 1$ entries, the hop count of Table 1; a flight occupies $t_K - t_0 + 1$ timesteps, one more than its filed duration. At graph level the regime follows from the ratio of the timestep σ to the flying time of a typical edge, where $\bar{v} = 450$ kt is the middle speed class,

$$\rho = \sigma / \text{median}_{e \in E}(\ell_e / \bar{v}). \quad (5)$$

For $\rho > 1$ at least half of all edges cost exactly one timestep at $\bar{v}$ and duration is set by the hop count; for $\rho < 1$ a typical edge takes more than one timestep, and the smaller ρ , the closer the duration to ℓ_f/v . On DACH (median edge 18.30 km, 1.318 min) $\rho = 45.53, 11.38, 3.04$ and 0.76 at $T_{gran} = 1, 4, 15, 60$; Section 3.3 turns ρ into a selection rule.

5. Statistical Analysis

This section asks how closely the generated traffic resembles the real traffic the generator was fitted to. The two properties play different roles. Conformance to the formal model of Beiser et al. [2, 3] is a requirement every released instance meets. Resemblance to real traffic is optional: it does not determine the usefulness of benchmark instances, but closer instances allow more sophisticated conclusions. What follows is the summary; the full specification, the per-region tables and the sector loads are in Appendix 2, in the supplementary material⁴.

5.1 Method: matched samples, a ceiling and a null

The target variables V1 to V15 (Table 3, Appendix 2.1) cover when and where traffic appears, which airports feed which, and how sectors load, the operational realism Olive et al. ask of ATFCM (Section 2). Following their guideline not to score a model with a metric it was built to optimize, we group the variables into four tiers by what agreement can show. *Tier 1*, fitted by no part of the generator and scored against the data: agreement is evidence of fidelity. *Tier 2*, fitted by no part of the generator either, but no measured sector loads exist, so the real side of the sector load V8, the simultaneous flow V9 and the sector visits V14 is projected onto the instance's own sectorization first (Appendix 2.6); agreement then says that synthetic traffic loads a given geometry as real traffic

⁴Submitted with this paper for review; published on Zenodo (DOI: 10.5281/zenodo.23039874).

would. *Tier 3*, fitted directly by the demand or origin–destination model: agreement is expected and only confirms that the fit is reproduced. *Tier 4*, different by design and reported without a comparison: V7, clipped by configuration (Section 4.4.2), and V12, zero because $|F|$ is an input. A single distance between a synthetic sample and a real one is not interpretable, as it has to be brought into perspective with how two real samples deviate, how two synthetic samples deviate, and what the deviation to a baseline is — we use chance as a baseline and refer to it as *null*. Each measurement therefore reports four values of one distance d for a target variable, all at the same sample size, each between two samples summarized by the variable’s statistic (Table 3):

$$\underbrace{d_{\text{real}}}_{\text{real vs. disjoint real}} \quad \underbrace{d_{\text{syn}}}_{\text{seed vs. seed}} \quad \underbrace{d_{\text{obs}}}_{\text{synthetic vs. real}} \quad \underbrace{d_{\text{null}}}_{\text{null model vs. real}} \quad (6)$$

The ceiling d_{real} is how far apart two samples of the *same* process are at this size, and the null d_{null} is what chance achieves; we report where d_{obs} falls between them, the normalized excess

$$E = \frac{d_{\text{obs}} - d_{\text{real}}}{d_{\text{null}} - d_{\text{real}}}. \quad (7)$$

$E = 0$ means the synthetic sample sits as close to real as a second real sample does, $E = 1$ means it is no better than the null, and $E > 1$ means it is further from real than the null is. Each value is divided by the ceiling and null of its own *scenario*, one region at one time granularity and flight count, so E is comparable across regions and granularities for one variable; across variables it also depends on how far each null lies from real traffic. A variable is *within real variation* in a scenario when d_{obs} falls inside the ceiling spread, the range of distances between real samples: a typical instance would not stand out among real days, which does not mean that the two processes agree; in 26 of the 35 such entries of Table 2 the interval of Figure 6 lies above $E = 0$. It is *unresolved* when the null itself falls inside the spread, so that E has no scale (Figure 3). No threshold is set, since any threshold depends on the sample size as much as on the generator; usability as a benchmark is established by the invariants of Section 4.4.2 and the solver studies of [2, 3] (Appendix 2.1).

Regions measured. Table 2 shows six regions at $T_{\text{gran}} = 60$ and, for contrast, DACH at $T_{\text{gran}} = 4$, the granularity used by Beiser et al. [3] (scenarios in Table 5, Appendix 2.2). Each of the 3 or 10 instances of a scenario is compared against every block of real days holding about $|F|$ flights (Table 5); EAST-ASIA-40x40 is measured at an unshipped $|F| = 420$, its real daily volume. USA-EAST-COAST-20x10 is reported in Appendix 2.3 for space. CENTRAL-EUROPE, at 30.8 real departures a day, would need 32 real days to match even $|F| = 1,000$, where only 29 are available, so it admits no ceiling at any shipped level; Appendix 2.3 reports it at an unshipped $|F| = 100$, too small a sample for a fidelity claim. The small-scaling dataset, built for exact solvers, is not measured.

5.2 Temporal structure and peaks

Rebuilt from the OpenSky COVID dataset, the demand model (Equation (1)) reproduces the real hourly profile to a total variation distance of at most 0.006, and every hour carrying at least 1 % of the day’s traffic is within 3.9 % of its real value in every region (Table 7, Appendix 2.4). The sampled instances keep this: the hourly departure profile V1, the en-route concurrency shape V5 and the peak structure V11 are at or inside the ceiling at $T_{\text{gran}} = 60$ in every region but CENTRAL-EUROPE-7x7, whose V5 is resolved at +0.115 (Table 2, Figure 4). Eight of these excesses are negative, because a model fitted on 30 days reproduces an average day, which is closer to the mean than any single real day; a negative excess is no sign of a better-than-real instance. On USA-MAINLAND a partial last day, cut at 16:00 local where the flight list ends, widens the real-vs-real spread past the null (V1: 0.282, 0.107 without it), which leaves all three entries unresolved (Appendix 2.1).

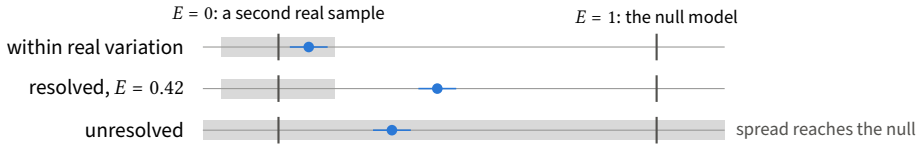


Figure 3. How to read the normalized excess E (Equation (7)). Each entry gets its own scale, from $E = 0$, the distance between two real samples, to $E = 1$, the distance of the null model. The gray band is the ceiling spread, the 5 to 95 % range of distances between real samples; the dot is the mean over all synthetic samples, the thin line its interval. Inside the band an entry is within real variation; outside it the difference is resolved and E gives its size; when the band reaches the null, the scale has no length and the entry is unresolved.

Table 2. Normalized excess (Equation (7)) per target variable and scenario, read as in Figure 3: **shaded** entries lie within real variation, **gray** entries are unresolved, the rest resolved; tiers as in Section 5.1. A region with a tight ceiling resolves differences a wide one cannot, so a larger excess alone does not mean a worse instance. CE-7x7, E-ASIA, MAJ-EU and USA are CENTRAL-EUROPE-7x7, EAST-ASIA-40x40, MAJOR-EUROPE-40x20 and USA-MAINLAND.

Tier	Variable	CE-7x7	DACH	DACH	E-ASIA	EUROPE	MAJ-EU	USA
	T_{gran}	60	4	60	60	60	60	60
1	V11 Peak structure	+0.070	+1.158	-0.021	+0.073	-0.024	+0.058	-0.004
	V13 Day boundary	-0.003	+0.231	-0.003	+0.340	+0.039	+0.009	+0.233
	V5 Concurrency shape	+0.115	+0.869	-0.058	+0.126	+0.003	+0.018	-0.021
	V4 Duration body	+0.357	+1.185	+0.160	+0.212	+0.110	+0.226	+0.268
	V10 Duration tail	+1.148	+0.639	+0.684	+2.326	+0.828	+0.483	+1.272
	V6 Legs per aircraft	+0.999	+0.305	+0.474	+1.198	+0.145	+0.796	+0.405
2	V8 Sector load	+0.072	+0.447	+0.108	+0.073	+0.077	+0.039	+0.192
	V9 Simultaneous flow	+0.083	+0.523	+0.098	+0.100	+0.082	+0.060	+0.083
	V14 Sector visits	+0.025	+0.462	+0.165	+0.086	+0.176	+0.034	+0.240
3	V1 Hourly profile	+0.023	+0.785	-0.043	+0.014	-0.035	+0.030	-0.125
	V2 $P(\text{dest.} \mid \text{orig.})$	+0.155	+0.643	+0.181	+0.115	+0.234	+0.375	+0.135
	V3 OD joint	+0.400	+0.846	+0.263	+0.161	+0.246	+0.509	+0.088

5.3 Origin–destination structure and route distributions

The first release drew every destination from the global marginal (Section 4.2), so its pairs carry no dependence on origin or time of day; the second draws from the conditional model fitted to the real traffic and reproduces it. The second release at $T_{gran} \geq 15$ is closer to the conditional model than to the marginal at each of the eight busiest origins in all eight regions: 0.0105 against 0.2628 at the busiest DACH origin at $T_{gran} = 60$. At $T_{gran} = 1$ this reverses in four regions, and on DACH also at $T_{gran} = 4$, because a window too short for the long routes makes trajectory generation re-sample legs; these instances satisfy every invariant but carry no OD-fidelity claim (Appendix 2.5). At $T_{gran} = 60$ the pair distribution V3 lies within real variation in five of six regions and V2 in four; both are fitted, so this confirms that the fit is reproduced.

Route geometry cannot be validated without tracks; the sector-visit distribution V14 is within real variation in three of six regions and within an excess of +0.25 in the other three. Sector load V8 and simultaneous flow V9, scored against projected real loads (tier 2), are reported in Appendix 2.6.

5.4 Durations, rotations, and the tails

Durations are where the generator departs from real traffic, in every region and at every granularity. This is especially visible in the tail: a typical synthetic flight lasts about as long as a real one, but

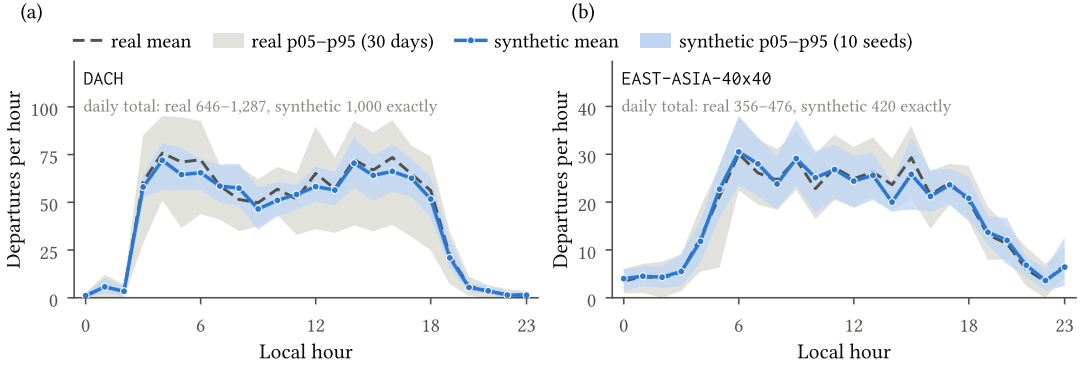


Figure 4. Hourly departure profile, synthetic against real, at $T_{gran} = 60$. The gray band is the p05–p95 spread of the 30 real days, the visual form of the ceiling; the synthetic mean stays inside it at every hour in both regions. DACH’s real band is about twice its synthetic one, the weekly cycle of Section 6.2; EAST-ASIA-40x40 has almost no weekly cycle and the two bands nearly coincide. What no seed reproduces is the daily total, fixed at $|F|$.

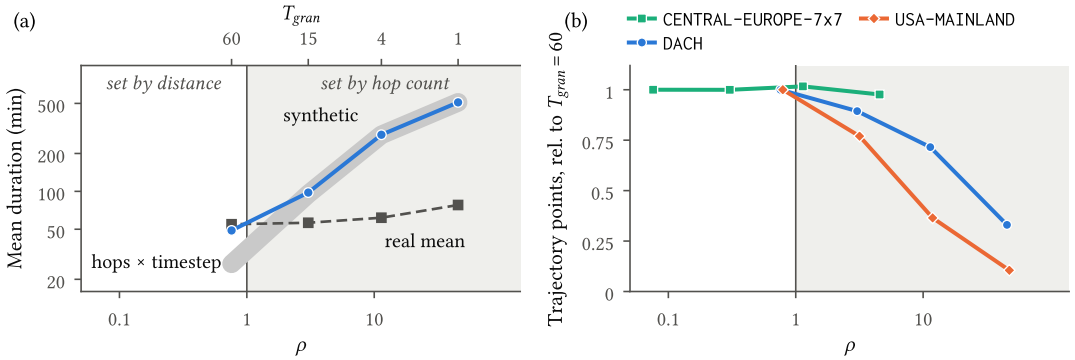


Figure 5. Time granularity sets how faithful durations are and how large an instance is, through the edge cost w_v (Section 4.4.2) and ρ (Equation (5)). (a) DACH, $|F| = 1,000$, mean time a flight occupies, $(t_K - t_0 + 1)\sigma$, over three seeds (spread within the markers): for $\rho > 1$ the synthetic mean lies on the gray band, hop count times timestep; only at $T_{gran} = 60$ ($\rho < 1$) does it approach the real mean, quantized alike. (b) Trajectory points, the size of the instance a solver processes, relative to $T_{gran} = 60$ (seed 42).

the longest synthetic flights are several times shorter than the longest real ones. Table 2 shows the split. The duration body V4, the 10th to 90th percentiles of flight duration, has an excess of +0.11 to +0.36 over the six $T_{gran} = 60$ scenarios, and the duration tail V10, from the 90th percentile to the maximum, one of +0.48 to +1.27 where it has a scale. On DACH the synthetic median is 0.93 times the real one and the 95th percentile matches it (1.00), but the longest synthetic flight lasts 89 to 103 min over the ten seeds against 682 min in the real month, a ratio of 0.15; on EAST-ASIA-40x40 it lasts 219 to 239 min against a real 1,361 (Figure 7 in Appendix 2.7).

The cause is structural, so no number of seeds and no granularity removes it. A synthetic duration is computed from the route’s length at one of three cruise speeds (Section 4.4.2): it grows almost linearly with distance, with a correlation between great-circle distance and duration of +0.993 against a real +0.741 on EAST-ASIA-40x40, takes only three values per origin–destination pair, and cannot exceed the time needed to cross the region. A real duration is a surveillance span from first to last contact and also contains taxi, holding, vectoring and diversions; at the far end the flight list also holds records whose first and last contacts are hours apart over a few hundred kilometers (EDDM–

EDDN, 137 km in 653 min), so the real maxima are not flight times. A classifier therefore separates individual real from synthetic flights almost perfectly. A feature attribution shows that duration, great-circle distance and sectors traversed explain nearly all of its “accuracy” (Appendix 2.7).

Rotations differ as well: inside the region polygon, synthetic aircraft fly more legs each (V6, +0.15 to +1.20); on EAST-ASIA-40x40, 130 aircraft fly the 420 synthetic flights of a day where 245 fly the 421 real ones. The polygon cuts real rotations short, most where a region holds fewest airports, so the excess overstates the difference most there: +0.15 on EUROPE with its 936 airports against +1.00 on CENTRAL-EUROPE-7x7 with 7.

5.5 Granularity: one mechanism to rule them all

Every divergence that moves with T_{gran} moves for one reason: once $\rho > 1$ (Section 4.4.4), the time a flight occupies becomes the hop count times the timestep and stops depending on distance (Figure 5a); concurrency, sector load and, through the postponed legs of Section 4.4.2, the departure profile follow. The same rule caps a route at 24 edges at $T_{gran} = 1$, so on DACH the mean hop count falls from 26.6 at $T_{gran} = 60$ to 8.5, legs per aircraft from 3.5 to 1.5, and the origin–destination mix shifts with them. It also makes a coarse granularity cheap, since long routes are never written: at $|F| = 1,000$, $T_{gran} = 60$ costs 9.5 times the trajectory points of $T_{gran} = 1$ on USA-MAINLAND, 3.0 times on DACH and 1.0 on CENTRAL-EUROPE-7x7, which has no long routes to lose (Figure 5b). A user choosing T_{gran} therefore chooses a point on one trade-off between fidelity and generation and solving cost; $T_{gran} = 60$ is better than $T_{gran} = 4$ on ten of the twelve variables measured at both (Appendix 2.8); the exceptions are the duration tail, where the two are effectively tied, and legs per aircraft.

6. Discussion

The generator’s primary goal is to distribute open instances for benchmarking and comparing algorithms for joint ATFCM, and no difference discussed below withdraws an instance from that use. The measurements of Section 5 do decide what an instance may be used to *claim*; this section places the generator accordingly and states its limits.

6.1 What the instances support

An instance is the six-tuple $I = (G, T_{gran}, T, sec, cap, A)$ that the joint ATFCM formulation [2, 3] takes as input: the navigation graph G , the time granularity T_{gran} , the time window T , the initial sectorization sec (the partition S), the atomic capacities cap (c_p), and the filed flight plans with their aircraft A . Every instance thus goes to a joint ATFCM solver as is, and researchers can (i) benchmark how an algorithm scales with instance size along two axes, the number of flights (10 to 31,622) and the size of the navigation graph (9 to 18,723 navpoints); and, in the large-scaling dataset, (ii) measure how it copes as nominal capacity is degraded, since the capacity sweep lowers it in steps of 10 % while demand, graph and routes stay fixed; and (iii) study how the time granularity affects its performance on the same traffic. Because the origin–destination structure follows the real one (V3, Section 5.3), the hotspots of a real-waypoint region lie roughly where real traffic creates them (V8, Table 2).

6.2 Limitations and directions

Chosen for benchmark value. The 24-hour window makes an instance self-contained but removes overnight traffic (V13: nothing on DACH, a fifth of peak concurrency on EAST-ASIA-40x40; Appendix 2.4); turnarounds are clipped at 60 min by configuration; capacities derive from the instance’s own peak and are equal at every vertex all day, so they are no operational capacities and include no airport rates [14]; and $|F|$ is fixed, so demand does not vary from day to day — its coefficient of variation is 0.000 against a real 0.074 to 0.203, which is why the instances cannot be used to study

demand uncertainty. *Data and model.* The COVID dataset records traffic as flown, already shaped by flow measures, so the instances support comparing algorithms under controlled congestion and no claim about real ATFCM workload; it misses airports without surveillance coverage, such as Beijing Capital and Shanghai Pudong. Departures are independent draws around a mean day, without banks, events or disruptions, and destinations never leave the observed origin–destination pairs. Nor is there a weekly cycle, which is most of the real spread: on DACH the weekday means differ by a factor of 1.75 and the day of the week explains 89 % of the variance in daily volume, against 1.08 and 14 % on EAST-ASIA-40x40. Routes are shortest paths in an effectively two-dimensional graph, without airways, wind, vertical profiles or the restrictions of the Route Availability Document [39]; sectors are synthetic partitions of a flat graph, with no vertical extent and no list of the configurations a center would open. *Measurement.* Section 5 measures fidelity and not hardness: the earlier versions of both datasets already proved hard for exact approaches [2, 3]. No controller or flow manager has assessed their operational realism [36]. Real durations include taxi and holding, so part of the duration gap is definitional, and the models are scored on their fitting month. Refitted on the 15 odd days of June 2019 and scored on the 15 even days, DACH changes by at most +18.7 % in d_{obs} , while on EAST-ASIA-40x40 the concurrency shape degrades by +61 % and the hourly profile by +49 %, which we attribute to its smaller fitting window.

7. Conclusion

We presented an open-source data generator that addresses an *open challenge* in ATFCM research by building benchmark instances entirely from publicly available data. Its instances are self-contained (navigation graph, filed flight plans, initial sectorization, atomic capacities), so ground delay, rerouting and sector reconfiguration can be optimized *jointly*. In this paper we supplied the three things the generator was missing: a practitioner’s guide, a detailed specification of the generator, and a statistical analysis of the fidelity of the generated datasets. The large-scaling dataset holds 6,720 instances over eight regions and four time granularities, with 1,000 to 31,622 flights and 56 to 19,537 vertices; the small-scaling dataset holds 200 instances with 10 to 100 flights.

The data generator supports *comparability* and *reproducibility* across algorithmic evaluations, lets researchers isolate a graph topology and hold it fixed while demand or capacity moves, and stress-tests algorithms at volumes well above the historical baseline of a region. Time granularity — which defines the length of a timestep — is responsible for the main divergences from real data. At one-minute timesteps and a demand level matching the region’s real traffic, departure profiles, en-route counts and origin–destination pairs mostly lie within real day-to-day variation, and such instances support limited claims beyond benchmarking; one-hour timesteps serve benchmarking only. Flight durations diverge at every granularity.

Future work. The generator’s modular architecture is open to extension by the community; we ask that work building on it cites this paper. Still, we plan updated standard versions of the navigation graph stage, which should be time-granularity adaptive and support 3D graphs beyond basic structures, with flight levels, vertical profiles and sectors with vertical limits [11]. Route restrictions such as the Route Availability Document [39] are a further step, once they can be had in machine-readable form. The demand model should gain day-to-day variation and a weekly cycle, fitted on more than one month. A solver-difficulty study [20] and an instance-space analysis [17] would characterize its hardness beyond the published solvers. Besides that, future work shall go towards stochastic environments — *convective weather* events, temporary airspace closures and aircraft course deviations — to evaluate adaptive algorithms. Generator version 3 will start every departure at timestep 1.

Open data statement

Both datasets and the benchmark results are openly available on Zenodo (large-scaling dataset: 10.5281/zenodo.23039645; small-scaling dataset: 10.5281/zenodo.23039835; benchmark results: 10.5281/zenodo.23493550); Section 3.1 describes them and their licenses. Their inputs, the OpenSky COVID dataset [7], the OurAirports snapshot and the X-Plane navdata, ship with the large-scaling dataset.

Reproducibility statement

The generator is open source at <https://github.com/alex14123/ASPaeroFlow-DataGenerator> (MIT license), tag v2.0.0 (archived on Zenodo: 10.5281/zenodo.23136727), and the configurations that produced both datasets ship with them; Section 3.4 states what the tag reproduces, and a re-run at the tag is repeatable once PYTHONHASHSEED is set. The code of the statistical analysis of Section 5 (MIT) is available at 10.5281/zenodo.23039874, with instructions in the supplementary material (Appendix 2.9).

Author contributions

Alexander Beiser: Conceptualization, Methodology, Software, Data curation, Formal analysis, Investigation, Validation, Visualization, Writing – original draft, Writing – review & editing. Thomas Gräupl, Georg Trausmuth, Stefan Woltran: Conceptualization, Writing – review & editing.

Acknowledgement

This research was supported by Frequentis and by the Austrian Science Fund (FWF) under grant 10.55776/COE12.

Use of generative AI

Over the course of the project we used large language models from several providers: OpenAI's GPT-5 in ChatGPT, Google's Gemini 3.1 Pro, and Anthropic's Claude Opus 5 and 5.5. For software, we used them as coding assistants and agentic coding assistants, for code audits, and to write the scripts for data analysis and visualization. For the paper, we used them to assist in the literature search, to formulate text from our drafts and revise it, and to help with English grammar, wording and flow. Throughout the project they also served as discussion partners that helped develop, test and criticize ideas. The authors verified the results, revised the text, and take full responsibility for the content of this paper.

References

- [1] T. Bolic, A. Cook, R. Koelle, E. Spinielli, Q. Goens, and M. Strohmeier. "Roadmap for a European Open Science Alliance for ATM Research". In: *EJTIR* 24.4 (2024), pp. 18–40.
- [2] A. Beiser, M. Hecher, N. Musliu, and S. Woltran. "Joint Air Traffic Flow and Capacity Management via Answer Set Programming". In: *LPNMR* 26. 2026.
- [3] A. Beiser, M. Hecher, N. Musliu, G. Trausmuth, and S. Woltran. "ASPaeroFlow: Decomposition Heuristics for Joint Air Traffic Flow & Capacity Management". In: *ATMOS* 26. 2026.
- [4] Marc Bourgois and Michael Sfyroeras. "Open Data for Air Transport Research: Dream or Reality?" In: *Proceedings of the International Symposium on Open Collaboration (OpenSym '14)*. ACM, 2014, pp. 1–7. DOI: 10.1145/2641580.2641602. URL: <https://doi.org/10.1145/2641580.2641602>.

- [5] Quinten Goens, Rainer Koelle, Enrico Spinielli, Tatjana Bolić, Andrew Cook, and Martin Strohmeier. “The Open Performance Data Initiative: A Foundation Supporting the European Open Science Alliance for ATM Research”. In: *Proceedings of the 14th SESAR Innovation Days (SID 2024)*. Rome, Italy, 2024. URL: https://www.sesarju.eu/sites/default/files/documents/sid/2024/papers/SIDs_2024_paper_100%20final.pdf.
- [6] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. “Bringing up OpenSky: A large-scale ADS-B sensor network for research”. In: *IPSN-14 proceedings of the 13th international symposium on information processing in sensor networks*. IEEE, 2014, pp. 83–94.
- [7] M. Strohmeier, X. Olive, J. Lübke, M. Schäfer, and V. Lenders. “Crowdsourced air traffic data from the OpenSky Network 2019–2020”. In: *Earth Syst. Sci. Data* 13.2 (2021), pp. 357–366.
- [8] Xavier Olive. “traffic, a Toolbox for Processing and Analysing Air Traffic Data”. In: *Journal of Open Source Software* 4.39 (2019), p. 1518. DOI: 10.21105/joss.01518. URL: <https://doi.org/10.21105/joss.01518>.
- [9] Junzi Sun, Jacco M. Hoekstra, and Joost Ellerbroek. “OpenAP: An Open-Source Aircraft Performance Model for Air Transportation Studies and Simulations”. In: *Aerospace* 7.8 (2020), p. 104. DOI: 10.3390/aerospace7080104. URL: <https://doi.org/10.3390/aerospace7080104>.
- [10] J. Hoekstra and J. Ellerbroek. “BlueSky ATC Simulator Project: an Open Data and Open Source Approach”. In: *ICRAT16*. 2016.
- [11] Mercedes Pelegrin and Martina Cerulli. “Aircraft Conflict Resolution: A Benchmark Generator”. In: *INFORMS Journal on Computing* 35.2 (2023), pp. 274–285. DOI: 10.1287/ijoc.2022.1265. URL: <https://doi.org/10.1287/ijoc.2022.1265>.
- [12] Tong Guo, Yi Mei, Ke Tang, and Wenbo Du. “Cooperative Co-Evolution for Large-Scale Multiobjective Air Traffic Flow Management”. In: *IEEE Transactions on Evolutionary Computation* 28.6 (2024), pp. 1644–1658. DOI: 10.1109/TEVC.2023.3328886. URL: <https://doi.org/10.1109/TEVC.2023.3328886>.
- [13] Tong Guo. *ATFM-Benchmark: Eight National-Wide Air Traffic Flow Management Instances*. GitHub repository. Companion data for [12]; a simplified subset of the original data. No licence stated. Accessed 12 September 2026. 2023. URL: <https://github.com/buaaguo tong/ATFM-Benchmark>.
- [14] Dimitris Bertsimas, Guglielmo Lulli, and Amedeo Odoni. “An Integer Optimization Approach to Large-Scale Air Traffic Flow Management”. In: *Operations Research* 59.1 (2011), pp. 211–227. DOI: 10.1287/opre.1100.0899. URL: <https://doi.org/10.1287/opre.1100.0899>.
- [15] A. Agustín, A. Alonso-Ayuso, L. F. Escudero, and C. Pizarro. “On air traffic flow management with rerouting. Part I: Deterministic case”. In: *European Journal of Operational Research* 219.1 (2012), pp. 156–166. DOI: 10.1016/j.ejor.2011.12.021. URL: <https://doi.org/10.1016/j.ejor.2011.12.021>.
- [16] David García-Heredia, Antonio Alonso-Ayuso, and Elisenda Molina. “A Combinatorial model to optimize air traffic flow management problems”. In: *Computers & Operations Research* 112 (2019), p. 104768. DOI: 10.1016/j.cor.2019.104768. URL: <https://doi.org/10.1016/j.cor.2019.104768>.
- [17] Kate Smith-Miles and Simon Bowly. “Generating New Test Instances by Evolving in Instance Space”. In: *Computers & Operations Research* 63 (2015), pp. 102–113. DOI: 10.1016/j.cor.2015.04.022. URL: <https://doi.org/10.1016/j.cor.2015.04.022>.
- [18] Simon Bowly, Kate Smith-Miles, Davaatseren Baatar, and Hans Mittelmann. “Generation Techniques for Linear Programming Instances with Controllable Properties”. In: *Mathematical Programming Computation* 12.3 (2020), pp. 389–415. DOI: 10.1007/s12532-019-00170-6. URL: <https://doi.org/10.1007/s12532-019-00170-6>.
- [19] Zijie Geng, Xijun Li, Jie Wang, Xiao Li, Yongdong Zhang, and Feng Wu. “A Deep Instance Generative Framework for MILP Solvers Under Limited Data Availability”. In: *Advances in*

- Neural Information Processing Systems 36 (NeurIPS 2023)*. 2023. DOI: 10.48550/arXiv.2310.02807. URL: <https://arxiv.org/abs/2310.02807>.
- [20] Varvara Kovaleva and Alexander Ponomarenko. *A Benchmark Generator of Realistic Logistic Transport Networks with Nonlinear Edge Costs and Transshipment Capacities*. arXiv preprint. 2026. DOI: 10.48550/arXiv.2606.20653. URL: <https://arxiv.org/abs/2606.20653>.
- [21] Gabriel Jarry, Nicolas Couellan, and Daniel Delahaye. "On the Use of Generative Adversarial Networks for Aircraft Trajectory Generation and Atypical Approach Detection". In: *Air Traffic Management and Systems IV*. Vol. 731. Lecture Notes in Electrical Engineering. Springer Singapore, 2021, pp. 227–243. DOI: 10.1007/978-981-33-4669-7_13. URL: https://doi.org/10.1007/978-981-33-4669-7_13.
- [22] Timoth  Krauth, Adrien Lafage, J r me Morio, Xavier Olive, and Manuel Waltert. "Deep Generative Modelling of Aircraft Trajectories in Terminal Maneuvering Areas". In: *Machine Learning with Applications 11* (2023), p. 100446. DOI: 10.1016/j.mlwa.2022.100446. URL: <https://doi.org/10.1016/j.mlwa.2022.100446>.
- [23] Abdulmajid Murad and Massimiliano Ruocco. "Synthetic Aircraft Trajectory Generation Using Time-Based VQ-VAE". In: *2025 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. Brussels, Belgium, 2025. DOI: 10.48550/arXiv.2504.09101. URL: <https://arxiv.org/abs/2504.09101>.
- [24] Olav Finne Praesteng Larsen, Massimiliano Ruocco, Michail Spitieris, Abdulmajid Murad, and Martina Ragosta. *Learning to Land Anywhere: Transferable Generative Models for Aircraft Trajectories*. arXiv preprint; also presented at the 15th SESAR Innovation Days (SID 2025). 2025. DOI: 10.48550/arXiv.2511.04155. URL: <https://arxiv.org/abs/2511.04155>.
- [25] Seokbin Yoon and Keumjin Lee. "Aircraft Trajectory Dataset Augmentation in Latent Space". In: *International Journal of Aeronautical and Space Sciences 27.2* (2025), pp. 1924–1935. DOI: 10.1007/s42405-025-01054-x. URL: <https://doi.org/10.1007/s42405-025-01054-x>.
- [26] Mayara Cond  Rocha Mur a and McWilliam de Oliveira. "A Data-Driven Probabilistic Trajectory Model for Predicting and Simulating Terminal Airspace Operations". In: *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*. IEEE, 2020, pp. 1–7. DOI: 10.1109/DASC50938.2020.9256644. URL: <https://doi.org/10.1109/DASC50938.2020.9256644>.
- [27] Timoth  Krauth, J r me Morio, Xavier Olive, Benoit Figue , and Raphael Monstein. "Synthetic Aircraft Trajectories Generated with Multivariate Density Models". In: *Proceedings of the 9th OpenSky Symposium*. Vol. 13. Engineering Proceedings 1. 2021, p. 7. DOI: 10.3390/engproc2021013007. URL: <https://doi.org/10.3390/engproc2021013007>.
- [28] Sebastiaan Wijnands, Alexei Sharpanskykh, and Karim Aly. "Generation of Synthetic Aircraft Landing Trajectories Using Generative Adversarial Networks". In: *Proceedings of the 14th SESAR Innovation Days (SID 2024)*. Rome, Italy, 2024. URL: https://www.sesarju.eu/sites/default/files/documents/sid/2024/papers/SIDs_2024_paper_054%20final.pdf.
- [29] Karim Aly and Alexei Sharpanskykh. "Synthetic Flight Data Generation Using Generative Models". In: *2025 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2025, pp. 1–10. DOI: 10.1109/ICNS65417.2025.10976960. URL: <https://doi.org/10.1109/ICNS65417.2025.10976960>.
- [30] Abdulmajid Murad and Massimiliano Ruocco. "Pre-Tactical Flight-Delay and Turnaround Forecasting with Synthetic Aviation Data". In: *CEAS Aeronautical Journal* (2026). DOI: 10.1007/s13272-026-00941-7. URL: <https://doi.org/10.1007/s13272-026-00941-7>.
- [31] Lucas Theis, A ron van den Oord, and Matthias Bethge. "A Note on the Evaluation of Generative Models". In: *4th International Conference on Learning Representations (ICLR 2016)*. 2016. DOI: 10.48550/arXiv.1511.01844. URL: <https://arxiv.org/abs/1511.01844>.
- [32] Sebastian Bischoff et al. "A Practical Guide to Sample-based Statistical Distances for Evaluating Generative Models in Science". In: *Transactions on Machine Learning Research* (2024). DOI: 10.48550/arXiv.2403.12636. URL: <https://arxiv.org/abs/2403.12636>.

- [33] David Lopez-Paz and Maxime Oquab. “Revisiting Classifier Two-Sample Tests”. In: *5th International Conference on Learning Representations (ICLR 2017)*. 2017. DOI: 10.48550/arXiv.1610.06545. URL: <https://arxiv.org/abs/1610.06545>. 609 610 611
- [34] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. *Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs*. arXiv preprint. 2017. DOI: 10.48550/arXiv.1706.02633. URL: <https://arxiv.org/abs/1706.02633>. 612 613 614
- [35] Ahmed Alaa, Boris van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. “How Faithful is your Synthetic Data? Sample-level Metrics for Evaluating and Auditing Generative Models”. In: *Proceedings of the 39th International Conference on Machine Learning (ICML)*. Vol. 162. Proceedings of Machine Learning Research. 2022, pp. 290–306. URL: <https://proceedings.mlr.press/v162/aaa22a.html>. 615 616 617 618 619
- [36] Xavier Olive, Junzi Sun, Mayara Condé Rocha Murça, and Timothé Krauth. “A Framework to Evaluate Aircraft Trajectory Generation Methods”. In: *Proceedings of the Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)*. 2021. URL: https://pure.tudelft.nl/ws/files/105023386/ATM_Seminar_2021_paper_25.pdf. 620 621 622 623
- [37] Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, 2014. 624 625
- [38] K. Ruben Gabriel and Robert R. Sokal. “A new statistical approach to geographic variation analysis”. In: *Systematic Zoology* 18.3 (1969), pp. 259–278. 626 627
- [39] EUROCONTROL. *Route Availability Document (RAD)*. EUROCONTROL Network Manager, updated every AIRAC cycle. Accessed 18 September 2026. URL: <https://www.nm.eurocontrol.int/RAD/>. 628 629 630

Appendix 1. Differences from the First Release

The instances of the first release, used in Beiser et al. [2, 3], were produced by an earlier version of the generator; we implemented three improvements in the second release, published with this paper.

Flights crossing UTC midnight. The first release's demand model discarded every observed flight whose first and last contact fell on different UTC dates. Where the UTC day boundary falls in the traffic trough the loss is negligible: DACH loses 0.1 % of its daily departures. Where it falls in the evening peak it is not: USA-MAINLAND loses 9.3 %, with local hour 17 (UTC-6) down 85 % (681 to 105 departures per day), and the first release's EAST-ASIA region loses 9.9 %, with no departures left in local hour 7.

Destinations. Destinations were not drawn from an origin- and time-conditional model, but from a marginal q , so first-release destinations are independent of origin and time of day. Over the eight busiest origins of DACH and of USA-MAINLAND, every origin's synthetic destinations lie closer to q than to the conditional model of Equation (2): the median total variation distance to q against that to the model is 0.031 against 0.355 for DACH and 0.080 against 0.397 for USA-MAINLAND, at $T_{gran} = 4$. In the second release the relation reverses, at 0.275 against 0.013 and 0.318 against 0.068 at $T_{gran} = 60$. At $T_{gran} = 4$ the trajectory generator changes the endpoints of 51 % and 71 % of the flights: it resamples the legs dropped at the 24-hour window (Section 4.4.2) to reachable destinations drawn by the sampled destination frequencies, an estimate of q , so DACH is again closer to q (0.233 against 0.347; Appendix 2.5) and USA-MAINLAND far from both (0.632 against 0.484).

Hop timing in USA-MAINLAND. Some first-release filed plans in USA-MAINLAND time one edge at a speed other than the aircraft's in `airplanes.csv`: 2,786 of the region's 212,900 flights at $T_{gran} = 4$. The error is one timestep, where 1,235 flights are one timestep too short and 1,551 are one timestep too long. The second-release instances satisfy the edge cost w_v (Section 4.4.2) on every one of their 108.9 million edges.

Appendix 2. Supplement to the Statistical Analysis

We show the details of Section 5 that the main text summarizes: the specification of the measurement, the per-region tables, the sector loads and the duration figure.

Appendix 2.1 Target variables and the measurement in detail

According to the concept of Olive et al. [36], ATFCM experiments require *high* operational and *low* physical realism; meaning for example that aircraft performance matters little, while peak loads do matter. Agreement on tier 3 variables means that the generator reproduces its own fit, since the models are scored on the days they were fitted to; the hold-out of Section 6.2 scores them on days they have not seen. Tier 1 variables — the peak structure V11, the day boundary V13, the concurrency shape V5, the duration body V4 and tail V10 and the legs per aircraft V6 — are consequences of the model measured against the data, so a divergence on them is informative. The tier 2 variables are consequences of the model as well, but their reference has to be built before it can be compared. For a tier 4 variable a difference is guaranteed by a configuration choice and carries no information about the generator. Table 3 lists them with their statistics, distances and tiers. Every distance used here sharpens with sample size — a total variation distance over origin-destination pairs, a correlation over sectors and a tail quantile all do — so a bare value confounds the generator with the number of flights on either side, which is why each is reported against the ceiling of Equation (6).

The null is chosen per variable. “No better than chance” means something different for a correlation and for a duration distribution, so a single global null would make Equation (7) mean-

Table 3. The target variables and their metrics; the tiers are those of Section 5.1. V1–V6, V8–V11, V13 and V14 are reported as the normalized excess (Equation (7)); V7, V12 and V15 have none and are reported on their own scales. TVD is the total variation distance, AUC the area under the ROC curve.

	Target variable	Statistic	Distance	Tier
V1	Hourly departure profile	24-bin share of departures	TVD	3
V2	$P(\text{dest.} \mid \text{origin})$	stacked per-origin destination shares	mean per-origin TVD	3
V3	Origin–destination joint	share per ordered pair	TVD	3
V4	Duration body	quantiles $p_{10} \dots p_{90}$	mean $ \log \text{ratio} $	1
V5	En-route concurrency shape	24-bin share of airborne count	TVD	1
V6	Legs per aircraft	share with 1 $\dots$ 6+ legs	TVD	1
V7	Turnaround	quantiles $p_{25} \dots p_{90}$	mean $ \log \text{ratio} $	4
V8	Per-sector load share	share of sector-timestep occupancy	TVD	2
V9	Simultaneous flow	off-diagonal sector load correlations	$1 - r$	2
V10	Duration tail	quantiles $p_{90}, p_{95}, p_{99}, \text{max}$	mean $ \log \text{ratio} $	1
V11	Peak structure	peak shares of departures and concurrency	mean $ \log \text{ratio} $	1
V12	Demand variability	coefficient of variation of daily volume	reported directly	4
V13	Day-boundary occupancy	concurrency at $t = 0$ and $t = W$, over peak	mean $ \text{difference} $	1
V14	Sector-visit distribution	share of sector entries	TVD	2
V15	Per-flight separability	classifier AUC, real against synthetic	AUC over its own ceiling	–

ingless. Each null keeps what the variable does not test and destroys what it does (Table 4). For $P(\text{destination} \mid \text{origin})$ the null is the global destination marginal.

Matching the real side. With n_{real} the region’s real departures per day, a synthetic instance of $|F|$ flights is compared against blocks of $\lfloor |F|/n_{\text{real}} \rfloor$ real days; a comparison whose two sides differ by more than 25 % in size is not run. A real flight counts only if both endpoints lie inside the region polygon and both airports are in the demand model, self-loops excluded (on DACH: 110 airports, 31,460 of 52,839 flights). Days are labeled in the local time of the statistics (offset τ , Section 4.1), a partial day with less than half the median traffic is dropped (on USA–MAINLAND, $\tau = -8$, the last local day ends at 16:00, where the flight list ends; it carries 64 % of the median traffic and is kept, and it alone puts the 95th percentile of real-against-real V1 distances at 0.282, 0.107 without it, above the null), and the days are cut into disjoint blocks under repeated random permutations, so that no block is tied to a weekday. Blocks are paired only within one permutation, and the bootstrap never pairs a block with itself. Each of the four distances is the mean over all pairs at the same sample size: d_{obs} over every instance and every real block (90 to 450 pairs: seeds times real blocks of Table 5), d_{syn} over the pairs of instances, d_{real} over the pairs of blocks within one permutation, and d_{null} over ten randomizations of every instance against every block. The interval of Figure 6 resamples the real blocks with the instances held fixed.

What the measurement can see. There are three possible outcomes, according to our classification: (i) *within real variation*, (ii) *unresolved* (E has no scale; see Section 5.1), and furthermore (iii) *resolved* when d_{obs} falls outside the ceiling spread. A region with a wide ceiling therefore produces fewer resolved differences, which reflects the region’s own day-to-day variability. Figure 6 carries the interval behind every value of Table 2 and Table 6, so a reader can see which differences the month of real data actually separates.

Table 4. The null model of each target variable. A null is applied ten times to every synthetic instance; it keeps what the variable does not test and destroys what it does, and d_{null} (Equation (6)) is the mean distance of these null instances to the real samples.

Variables	Null construction	Keeps	Destroys
V1, V5, V11, V13	every flight's departure timestep redrawn uniformly over the day	each flight's route and duration	time-of-day structure and peaks
V2	every destination redrawn from the global destination marginal	origins, times, overall destination shares	dependence of destination on origin
V3	destinations shuffled among flights	origin and destination shares	the pairing of origin and destination
V4, V10	durations drawn from an exponential distribution with the same mean	mean duration	shape and tail
V6	every leg flown by its own aircraft	the flights	rotations
V8, V14	sector labels permuted	how load (visits) spread over sectors	which sector carries it
V9	each sector's load series shifted by its own random offset	each sector's load profile and total	co-timing between sectors
V7, V12, V15	none: V7 and V12 are not compared; V15 is read against the real-vs-real AUC	—	—

Out-of-sample. The demand and origin–destination models are fitted on June 2019. To enable sensible measurements, we use hold-outs: the models are refitted on the 15 odd days of the month, instances are generated from them, and they are scored against the 15 even days. The refitted models are published with their instances alongside the analysis code (Appendix 2.9).

On acceptance criteria. A fidelity study is usually specified by target variables, metrics and acceptance criteria. We do not have acceptance criteria for how close measurements have to be, for one reason: an instance is usable when it is well specified and hard to solve, which is what the invariants of Section 4.4.2 and the solver studies of Beiser et al. [2, 3] establish. Statistical validation tells a user what a given instance additionally supports, which is why every number is reported next to the ceiling that calibrates it.

Appendix 2.2 The scenarios and the sampling floor

Every score is read against the ceiling d_{real} of Equation (6), the distance between two disjoint real samples of the same size, and two conditions decide whether it exists. *The region must supply two disjoint real blocks of the matched size.* CENTRAL–EUROPE (Section 5.1) has 29 usable days; a fidelity claim on it would need $|F|$ near 450 or below, so it is a solver benchmark only (Appendix 2.3 measures it at an unshipped $|F| = 100$).

The instance must hold enough flights per origin–destination pair. If the region's real traffic uses P distinct origin–destination pairs, an instance of $|F|$ flights draws each pair about $|F|/P$ times. As $|F|/P$ approaches one, most pairs are drawn once or twice, two seeds of the same generator fill largely disjoint sets of pairs, and the ceiling grows as large as the quantity it calibrates, so no score carries information. We therefore require $|F|$ to be about ten times P . This limits resolution, not validity: on EUROPE, with the three seeds of each $|F|$ pooled and as many real flights drawn at random from the month, the ceiling of the pair distribution tightens from 0.591 at $|F| = 1,000$ to 0.168 at 31,622, and at the shipped level of 10,000 the observed 0.260 sits below its ceiling of 0.267. V3 in Table 2 compares

Table 5. The scenarios. Each synthetic instance is compared against blocks of real days holding a comparable number of flights; *Sectors* counts the sectors loaded on both sides, over which V8 is compared (DACH has 182 in all); *real blocks* counts the disjoint blocks formed over repeated random permutations of the month. The two scenarios below the rule are reported in Appendix 2.3; *CENTRAL-EUROPE admits no ceiling at any shipped setting (Section 5.1) and is measured at an unshipped $|F| = 100$ on ten locally generated seeds. [†]EAST-ASIA-40x40 is measured at an unshipped $|F| = 420$, the level of its real traffic, on ten locally generated seeds; the other ten-seed scenarios add seven locally generated seeds to the three released ones.

Region	Graph	T_{gran}	$ F $	Sectors	Real days per block	Real blocks	Seeds
CENTRAL-EUROPE-7x7	grid	60	1,000	12	4	35	3
MAJOR-EUROPE-40x20	grid	60	1,000	26	1	30	10
EAST-ASIA-40x40	grid	60	420 [†]	22	1	30	10
DACH	Gabriel	60	1,000	163	1	30	10
DACH	Gabriel	4	1,000	160	1	30	10
EUROPE	Gabriel	60	10,000	1,066	1	30	3
USA-MAINLAND	Gabriel	60	10,000	1,014	1	30	3
USA-EAST-COAST-20x10	grid	60	1,000	14	3	50	3
CENTRAL-EUROPE	Gabriel	60	100*	37	3	45	10

one instance with one real day; real days repeat much the same schedule and lie closer ($d_{\text{real}} = 0.326$) than two seeds ($d_{\text{syn}} = 0.409$), so $d_{\text{obs}} = 0.381$ lies above that ceiling, inside its spread.

Table 5 lists the scenarios that satisfy both conditions: both graph families of Figure 2, three continents, 12 to 1,066 sectors and two orders of magnitude of demand. Most sit at the shipped demand level matching one real day; otherwise the instance is compared against blocks of several real days. The main granularity is $T_{gran} = 60$; DACH at $T_{gran} = 4$ is the granularity of the first-release instances used in [3].⁵

Appendix 2.3 Two further regions

Due to space limitations in the main part, we omitted two region evaluations from Table 2. (i) USA-EAST-COAST-20x10 is measured exactly as the scenarios of Table 2 and is reported here for space. Similar to the other grid regions, the temporal variables and the destination draw are within real variation, while durations, rotations and the spatial variables show resolved differences. Its legs-per-aircraft excess, +2.908, is the largest of any scenario and the extreme of the pattern of Section 5.4: with eight modeled airports, almost every real aircraft's day continues outside the polygon and is lost to the reference. (ii) CENTRAL-EUROPE admits no ceiling at its shipped demand levels (Section 5.1); for completeness it is measured at an unshipped $|F| = 100$ on ten locally generated seeds, against real blocks of three days. At that size its 49 origin-destination pairs receive about two draws each, far below the pair-count condition of Appendix 2.2, and its ceilings are about twice as wide as those of the other scenarios, so where it lies within real variation, that reflects the sample size and carries no fidelity claim.

Appendix 2.4 The demand model and the day boundary

Table 7 compares the shipped demand model (Equation (1)) with the model rebuilt from the COVID dataset over the same airports, for all eight large-scaling regions, before any sampling or routing. The residual in the daily total has two sources: the local-time shift of Section 4.1 leaves one day

⁵The ten seeds of that scenario are the three released instances together with seven generated locally at the same generator commit. The released instances predate the pinning of the interpreter hash seed, and at $T_{gran} = 4$ the trajectory generator draws replacement legs under it, so a released seed is not byte-identical to a local regeneration of it; the two are equivalent draws of the same sampler.

Table 6. Normalized excess (Equation (7)) for the two regions outside Table 2, in its layout and with its tiers: shaded entries lie within real variation, gray entries unresolved. The CENTRAL-EUROPE column is too small a sample to carry a fidelity claim.

Tier	Variable	USA-EAST-COAST-20x10	CENTRAL-EUROPE
	$T_{gran}, F $	60, 1,000	60, 100
1	V11 Peak structure	-0.058	+0.079
	V13 Day boundary	+0.021	-0.024
	V5 Concurrency shape	-0.044	+0.070
	V4 Duration body	+0.063	+0.062
	V10 Duration tail	+1.112	+0.026
	V6 Legs per aircraft	+2.908	+0.982
2	V8 Sector load	+0.182	+0.269
	V9 Simultaneous flow	+0.341	+0.191
	V14 Sector visits	+0.310	+0.262
3	V1 Hourly profile	-0.066	+0.070
	V2 $P(\text{dest.} \mid \text{orig.})$	-0.067	-0.157
	V3 OD joint	-0.295	+0.194

Table 7. The demand model against the COVID dataset, per region, before any flight is sampled. TVD is the total variation distance between the real and modeled shares over the 24 local hours, *max dev.* the largest relative deviation among the hours carrying at least 1 % of the day's traffic, and *cut carries* the share of the day's departures and arrivals in the hour that opens the window. The model is invariant in T_{gran} , so one audit per region covers all four granularities.

Region	Real dep./day	Model dep./day	TVD	Max dev.	Hours	Cut carries
CENTRAL-EUROPE-7x7	270.6	270.6	0.0000	0.00 %	17	0.01 %
USA-EAST-COAST-20x10	374.0	372.2	0.0040	3.89 %	20	0.20 %
MAJOR-EUROPE-40x20	1,087.4	1,087.4	0.0001	0.04 %	17	0.08 %
EAST-ASIA-40x40	420.9	420.3	0.0012	3.05 %	21	0.72 %
DACH	1,048.6	1,051.2	0.0007	0.55 %	17	0.17 %
CENTRAL-EUROPE	30.8	30.8	0.0000	0.00 %	18	0.00 %
EUROPE	10,241.8	10,270.2	0.0003	0.34 %	18	0.32 %
USA-MAINLAND	10,082.4	10,009.9	0.0060	3.62 %	21	0.54 %

missing in the $|\tau|$ hours at the window edge (-0.72% on USA-MAINLAND), and the audit's real side keeps only flights with both contact positions inside the polygon, which the model does not require ($+0.25\%$ on DACH).

Where the day is cut. The released configurations open the 24-hour window in the region's quietest hour (Section 4.1). That hour holds 0.17% of the day's departures and arrivals on DACH, 0.32% on EUROPE and 0.72% on EAST-ASIA-40x40, whose first hour would hold 5.26% with the window opening at UTC midnight. In hub-dominated regions the quietest hour is not local midnight (USA-MAINLAND: 08 UTC), so the offset is taken from the data and is not a civil time zone.

The day boundary in the instances. V13, the concurrency at $t = 0$ and $t = W$ relative to the instance's peak, shows what the window costs: nothing on DACH (-0.003), whose short-haul traffic empties at local midnight, but $+0.340$ on EAST-ASIA-40x40, which carries about a fifth of its peak concurrency across midnight while the instance carries none. The window repair plays no part, as

no flight departs at timestep 0 in any of the ten seeds; the gap is a property of the region, and V13 tells a user whether the window suits theirs.

Profile, concurrency and peaks. The eight negative excesses of V1, V5 and V11 mean that the synthetic sample is closer to the month's average than two real days are to each other; where the weekly cycle dominates the real spread (Section 6.2), synthetic instances also resemble each other more than real days do.

Appendix 2.5 Origin–destination audit

The audit of Section 5.3 pools each region's 7 demand levels and 3 seeds per granularity (CENTRAL-EUROPE-7x7 has only seven busy origins). At $T_{gran} = 1$ the result reverses: all eight DACH origins are closer to the global marginal than to the conditional model (0.6363 against 0.7739 at the busiest), as are six of eight in EUROPE and CENTRAL-EUROPE and five in MAJOR-EUROPE-40x20; on DACH this persists at $T_{gran} = 4$ (0.1964 against 0.3060). The cause is the window: at $T_{gran} = 1$ a route of more than 24 edges does not fit, so the trajectory generator replaces the dropped legs by legs to airports reachable in the remaining window (Section 4.4.2), outside the model of Equation (2). These instances are solver benchmarks without an origin–destination fidelity claim (Section 3.3).

Appendix 2.6 Sector loads and simultaneous flow

V8 compares each sector's share of the total sector-timestep occupancy, over the sectors present on both sides, and V9 the off-diagonal sector-by-sector load correlations, over the sectors whose load varies in every sample, so that ceiling and observation use the same sector pairs. Both are tier 2: nothing in the generator fits them, but their real side is projected from the COVID dataset along straight lines in latitude and longitude, since no measured sector loads exist; real flights follow airways, so their true co-variation would be higher than the projection shows.

V9 sits between +0.060 and +0.100 in all six regions at $T_{gran} = 60$, across both graph families and scenarios of 12 to 1,066 sectors, but its null is weak: shifting each sector's load in time separately also destroys the daily cycle that all sectors share, so the small excess mostly reflects that cycle and the sector volumes, which V1 and V8 measure already; V9 therefore says little about which sectors are busy together. V8 uses load shares, so an instance generated at several times a region's real volume does not read as a divergence.

The capacity structure of an instance has no real counterpart and none is claimed: real overload results from regulations absent from the COVID dataset, and the capacity levels of Section 4.4.3 are a control variable. As such they behave well: on DACH at $T_{gran} = 60$ and $|F| = 1,000$ the share of over-capacity sector-timestep cells falls monotonically from 24.44 % at PCAP010 to 0.00 % at PCAP100, and the largest single overload from 12 to 0. On CENTRAL-EUROPE-7x7 the share barely depends on $|F|$ (44.0 % to 37.6 % at PCAP010 from $|F| = 1,000$ to 31,622), so capacity level and $|F|$ act as nearly independent parameters (Section 3.3).

Appendix 2.7 Durations, rotations and the per-flight view

V4 compares the duration quantiles p_{10} to p_{90} and V10 the upper tail (p_{90} , p_{95} , p_{99} and the maximum). An excess above one means the durations are further from real than a memoryless null, which is what a hard cap against a long real tail produces. On DACH at $T_{gran} = 60$, pooling ten seeds against thirty real days, the synthetic p_{99} is 0.81 times the real one, between the close body and the distant maximum (Section 5.4). The flight schedule uses the drawn duration only for the speed class and the ready time; the trajectory generator recomputes the duration from the route (Section 4.4.2), so for a given origin–destination pair a synthetic duration takes only three values before rounding (Figure 7).

Table 8. Per-flight separability (V15). A gradient-boosted classifier separates individual real from individual synthetic flights on departure hour, duration, great-circle distance, sectors traversed, legs flown by the aircraft, turnaround and origin and destination frequency rank, grouped so that train and test share no day and no seed. *Ceiling* is the same procedure run between two halves of the real days, and *separability* is the difference. *Leading features* give the AUC lost when that feature alone is scrambled. Duration is taken on the instance's own timestep grid for both sides, so that the classifier cannot separate a synthetic flight by the fractional minutes a surveillance span has and a filed plan does not. At $T_{gran} = 4$ distance still stops mattering, because there a synthetic duration is set by the hop count and the timestep and no longer tracks distance at all (Section 5.5); at $T_{gran} = 60$ the classifier needs both, because the relationship between them identifies it.

Region	T_{gran}	$ F $	AUC	Ceiling	Separability	Leading features
DACH	60	1,000	0.9826	0.5159	+0.4667	dur. +0.278, dist. +0.265
DACH	4	1,000	0.9999	0.5202	+0.4797	dur. +0.201, dist. +0.017
MAJOR-EUROPE-40x20	60	1,000	0.9920	0.5423	+0.4497	dur. +0.362, dist. +0.317
EAST-ASIA-40x40	60	420	0.9985	0.6217	+0.3768	dur. +0.293, dist. +0.207

On EAST-ASIA-40x40 the median coefficient of variation of duration within a pair is 0.031 synthetic against 0.211 real.

A per-flight view. V15 asks a classifier to separate individual real from individual synthetic flights on eight features available on both sides, with train and test sharing no day and no seed. Since real days differ from each other and a classifier exploits that, its reference is the real-against-real AUC between two halves of the real days. Table 8 shows individual flights close to perfectly separable, almost entirely through duration, great-circle distance and the number of sectors traversed; departure hour, origin and destination frequency rank, legs per aircraft and turnaround add at most +0.088 AUC together. The DACH ceiling of 0.516 confirms that two halves of real days are nearly inseparable; the higher 0.622 of EAST-ASIA-40x40 reflects its small sample of 421 flights a day, whose day-to-day idiosyncrasies a classifier can learn. The measure is descriptive: it says how far a single synthetic flight is from a real one, not how good the instances are as optimization problems.

Rotations and turnaround. Inside the polygon, synthetic aircraft fly more legs, mostly three or four against one or two for real ones on EAST-ASIA-40x40, but the direction is not the generator's: counted over the same aircraft-days worldwide, real aircraft fly more legs than synthetic ones (4.64 against 4.33 on CENTRAL-EUROPE-7x7, 4.16 against 3.45 on DACH, 3.41 against 3.26 on EUROPE). The truncated and the complete real rotations thus bracket the synthetic ones, and a difference of shape remains. V7 differs by design: the pool Θ is clipped at 60 minutes by configuration and the realized ground time is at least one timestep (Section 4.4.2).

Appendix 2.8 Granularity in detail

On DACH at $|F| = 1,000$ the window binds at coarse granularities: 24.5 % of the flights end on the boundary of $[0, W]$ at $T_{gran} = 1$ and 7.7 % at $T_{gran} = 4$, none at 15 and 60. Raw distances are not comparable across granularities: a correlation is estimated over 1,440 time points at $T_{gran} = 60$ and over 96 at $T_{gran} = 4$, so the ceiling itself moves (0.194 against 0.388 for V9); the normalized excess is the comparable quantity.

Appendix 2.9 Reproducing the measurements

The analysis code (Zenodo, 10.5281/zenodo.23039874) computes Table 2, Table 5, Table 6 and Figure 6 with one driver, validation/run_validation.py, called once per scenario; Table 7, Table 8, the sampling-floor sweep, the origin-destination audit, the over-capacity shares and the input files of Figure 7 come from further scripts of the same archive. Its README maps each result to its script and

names the few that are not reproduced. `validation/reproduce_paper.sh` repeats the driver runs and those of Table 8 with the paper’s options and compares the output with the shipped results. From the code folder, with the published files extracted into `../inputs`, DACH at $T_{gran} = 60$ is measured by

```
python validation/run_validation.py --null-draws 10 --scale 1000 \  
  --cell ../inputs/configs/04_0_dach_TG60.json:../inputs/validation_instances_xplane_regions/  
04-0-DACH-2019-06-01--2019-06-30-CAP-ENROUTE-1200-CLUSTERSIZE-50-GABRIEL-GRAPH-V2-TG60-VAL10SEED \  
  --real-data-path ../inputs/third_party/opensky_flightlist/flightlist_20190601_20190630.csv.gz
```

A scenario is passed as `CONFIG::INSTANCE`, a region configuration and a folder laid out as one region of the unparsed tree (Section 3.2). The configurations, the flight list of the COVID dataset and the three-seed scenarios ship with the large-scaling dataset, the other scenarios with the analysis code. `--null-draws` sets how often each null is applied (default 1), `--real-day-parity` even scores the hold-out, and `--tag` labels the rows; the table scripts expect the tags of `reproduce_paper.sh`. On a laptop (AMD Ryzen 7 PRO 7840U, 29 GB) the run above takes under a minute and `reproduce_paper.sh` without EUROPE and USA-MAINLAND 7 minutes; EUROPE took 64 minutes with other runs in parallel, and V9 to V14 of USA-MAINLAND alone 19 minutes. No measured run exceeded 1.11 GB.



Figure 6. The normalized excess E of Equation (7) with its bootstrap interval, one panel per target variable, ordered by variable number from top left to bottom right, each labeled with its tier. Each panel carries the nine scenarios, the seven of Table 2 above the gap and the two of Table 6 below it. The interval is the 5th to 95th percentile of the bootstrap over the real blocks and calibrates the point estimate: where it spans $E = 0$, the mean synthetic sample sits as close to real as a second real sample does. The gray bar is the spread of single real-against-real distances (5th to 95th percentile) on the same scale: a point inside it lies within real variation, and a bar reaching $E = 1$ leaves the entry unresolved. $E = 0$ marks a real day pair and $E = 1$ the null model. Color, marker fill and line style all encode the resolution label, so the panels survive grayscale printing; values outside the axis keep a triangle at its edge with the number printed.

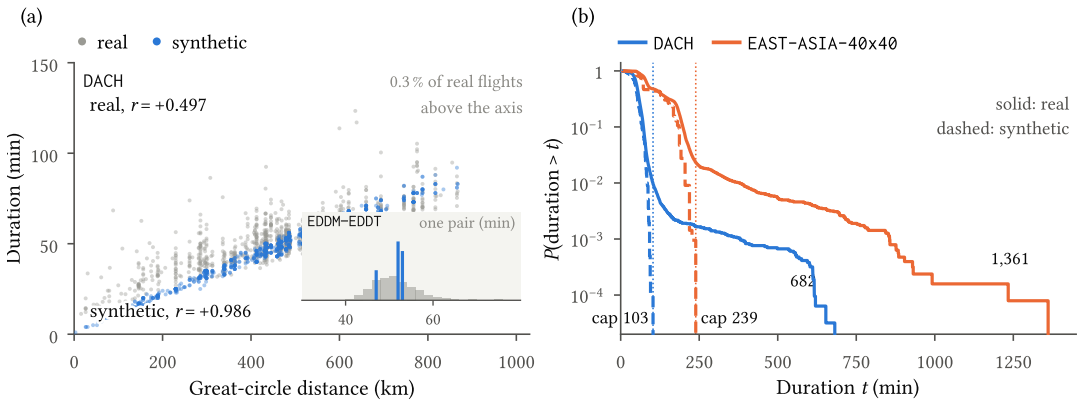


Figure 7. Why the durations diverge. (a) DACH at $T_{gran} = 60$, duration against great-circle distance for a matched random subsample of 1,000 flights per side: the synthetic duration is a near-deterministic function of distance, and the inset gives the mechanism — on one busy origin–destination pair it takes only the three values of the three speed classes of Section 4.4.2, where the real durations spread over tens of minutes. (b) Duration survival functions, real solid against synthetic dashed, with each region's route-geometry cap marked: the synthetic distribution ends at the cap and the real one continues past it, so the right tail of real durations lies outside the model class.