**Step 11 — Chance-Constrained “SLA” Certification (ch95)**

**Why we did this**

Up to Step 10 we optimized routes for **expected** performance (deterministic, quantile-buffer, SAA, Γ-robust). In many operations you must also guarantee a **service level**—e.g., “each plan should be on time with ≥95% probability under traffic variability.” Step 11 adds that *guarantee*, turning our study from “good on average” into “good with high confidence.”

Concretely, we enforce a **chance-constraint surrogate**:

a plan is “certified” only if its out-of-sample on-time rate at the **95th percentile** (p95) is **≥ 95%** (“ch95”).

This is easy to explain to stakeholders and aligns with SLA-style requirements.

**What we implemented**

**1) Post-hoc (simulation-based) certification**

We did **not** change the solvers again. Instead, we **re-screened** the plans produced in Steps 8–10 with a tougher statistical test:

* Generator: the same stochastic scenario model used earlier (log-normal multiplicative noise)
  + global CV = 0.20, link CV = 0.10 (correlated variability)
* Evaluation set: **K = 200** scenarios, common random numbers for all methods (fair).
* For each plan/instance we compute:
  + On-time% across scenarios and its empirical **p50** and **p95**.
* **Certification rule (ch95):** keep the plan only if **on-time p95 ≥ 95%**.

This is a *chance-constraint surrogate*: it doesn’t solve a hard chance-constrained VRPTW (which is expensive), but it **certifies** that the delivered plan meets the SLA under our traffic model.

**2) Champion selection under ch95**

We re-ran the champion picker with the ch95 filter turned on. Among all feasible candidate plans per instance (DET, Q-buffer, SAA, Γ), the script:

1. discards any plan failing **p95 ≥ 95%**;
2. among those that pass, selects the **lowest-cost** (distance + vehicle penalty) plan as the *champion* for that instance;
3. summarizes champions by method and plots **cost vs on-time**.

**What we ran (commands)**

# Evaluate all candidate folders on 200 scenarios

python scripts/evaluate\_plans.py `

--dirs "data\solutions\_ortools" `

"data\solutions\_quantile\m1.2\_a0" `

"data\solutions\_saa\k16\_b0p3" `

"data\solutions\_saa\k32\_b0p5" `

"data\solutions\_saa\k64\_b0p7" `

"data\solutions\_gamma\g1\_q1p645\_hybrid" `

"data\solutions\_gamma\g2\_q1p645\_hybrid" `

--labels DET Q120 SAA16-b0p3 SAA32-b0p5 SAA64-b0p7 G1 G2 `

--K 200 --seed 42 --cv\_global 0.20 --cv\_link 0.10

# Pick champions with the ch95 rule enabled

python scripts/pick\_champions.py `

--dirs "data\solutions\_ortools" `

"data\solutions\_quantile\m1.2\_a0" `

"data\solutions\_saa\k16\_b0p3" `

"data\solutions\_saa\k32\_b0p5" `

"data\solutions\_saa\k64\_b0p7" `

--labels DET Q120 SAA16-b0p3 SAA32-b0p5 SAA64-b0p7 `

--target 99 # keep aiming high for mean on-time

--ch95 # enforce p95 >= 95% on-time (the SLA)

**What we created**

* **data/reports/**
  + step8\_eval.csv, step8\_eval\_by\_method.csv (reused by Step 11)
  + champions.csv (updated: only plans that pass **ch95**)
  + champions\_stats\_by\_method.csv (stats of champions by method)
* **data/figures/**
  + champions\_cost\_vs\_ontime.png (your new scatter plot with the two surviving methods)
* **data/champions/**
  + JSON route files for the **certified** champions (one per instance)

(Your plot shows two labeled methods on the efficient frontier: **Q120** and **SAA16-b0p3**.)

**Results (high level)**

* **Deterministic (DET)** is eliminated by the ch95 filter (low reliability).
* **Γ-robust (G1/G2)** generally passes reliability but is **dominated on cost** (much higher distance/vehicles), so it doesn’t win many instances once Q120/SAA are available.
* The **winners under ch95** are:
  + **Q120 (Quantile buffer, mult=1.2)**: very high reliability and moderate cost.
  + **SAA16-b0p3**: slightly cheaper than Q120 on average, with on-time very close to 99% (still above the ch95 threshold).
* Your **champion scatter** (cost vs on-time) shows:
  + **Q120** ~ **99.4%** mean on-time with mean distance around **1 218**
  + **SAA16-b0p3** ~ **98.7%** mean on-time with mean distance around **1 166**
  + (Numbers read from your figure; the exact values per instance and overall means are in champions\_stats\_by\_method.csv.)

**Interpretation**

* **Why these two?** Q120 inflates travel times uniformly (a robust buffer), which raises reliability with a small cost bump; SAA16-b0p3 explicitly optimizes expected performance over random scenarios and finds slightly cheaper routes while still passing the 95% SLA.
* **Why not Γ-robust here?** With Γ=1–2 it tends to add extra protection on critical legs, often requiring more vehicles or detours. It’s great when missing the SLA is catastrophic; otherwise it’s costlier than you need.

**Practical guidance (how to use the outcomes)**

* If you want **one default method for all 56 instances**:
  + **Pick Q120** when you want a *simple*, very robust plan with minimal tuning.
  + **Pick SAA16-b0p3** when you can afford a few extra minutes of run-time and want the *lowest cost* plan that still satisfies the SLA.
* If you want **per-family choices**:
  + On **C** and **RC** families, Q120 and SAA16-b0p3 are both strong; Q120 may win on the tightest windows.
  + On **R** families, SAA16-b0p3 often captures cost savings while still clearing ch95.

**What to put in your report**

**One-paragraph abstract (Step 11)**

We added an SLA-style chance-constraint certification (ch95) on top of our robust VRPTW solvers. Each candidate plan (deterministic, quantile-buffer, SAA, Γ-robust) was stress-tested on 200 stochastic traffic scenarios (global CV 0.20, link CV 0.10). Plans were *certified* only if their **95th percentile on-time rate** was at least **95%**. Under this filter, **Quantile 1.2 (Q120)** and **SAA16-β0.3** dominated: both achieved ≈99% mean on-time with moderate cost, while deterministic plans failed reliability and Γ-robust plans were costlier. The final **champions** (one per instance) and their aggregated statistics are archived and plotted, providing a transparent, SLA-compliant deliverable.

**Figures to include**

* data/figures/champions\_cost\_vs\_ontime.png (your scatter plot)
  + Caption: “Certified champion plans (ch95). Average cost vs average on-time by method; label shows method and mean vehicles.”

**Tables to include**

* data/reports/champions\_stats\_by\_method.csv
  + Summarize mean distance, mean vehicles, mean on-time (p50/p95, if present) for Q120 and SAA16-b0p3 over the champion set.
* Optionally excerpt a few lines from data/reports/champions.csv
  + Show 4–6 representative instances (one per family) with method chosen, distance, vehicles, on-time p50/p95.

**Methods & settings (short text)**

* Scenario model: log-normal multiplicative noise; CV\_global=0.20, CV\_link=0.10; K=200 i.i.d. scenarios; common random numbers across methods.
* Certification rule: **on-time p95 ≥ 95%**.
* Candidate methods screened: DET, Q120, SAA16-b0p3, SAA32-b0p5, SAA64-b0p7, Γ-robust (Γ=1,2).

**Deliverables frozen for version control**

* data/champions/ (JSON routes for certified champions)
* data/reports/champions.csv, champions\_stats\_by\_method.csv
* data/figures/champions\_cost\_vs\_ontime.png
* (Keep step8\_eval\*.csv as the audit trail of the evaluation.)

**Conclusion of Step 11**

* You now have **SLA-certified** route plans for the 56 instances, plus a clear trade-off between **Q120** (simpler, ultra-robust) and **SAA16-b0p3** (cheaper, still ≥95% reliable at p95).
* These artifacts are the final inputs for Steps **12–17** (final comparison table, discussion, packaging, and annexes).