**6. Results**

**6.1 Setup recap (fixed & fair evaluation)**

All methods are evaluated on the **same 200 stochastic test scenarios** per instance (common random numbers; log-normal variability with cv\_global=0.20, cv\_link=0.10). KPIs:

* **Cost** = total distance (lower is better).
* **On-time service**: percent of customers served within due dates, reported as **p50** and **p95** over the 200 scenarios (higher is better).
* **Vehicles** used and **runtime** (s) from each method’s summary.csv.

We select a *champion* plan **per instance** (Step 11) that hits **≥99% on-time p95**, then compare methods on the **champion set** (what we would actually deploy).

**Methods compared.**

* **DET**: deterministic OR-Tools (baseline).
* **Q120**: quantile buffer (travel times × 1.20).
* **SAA16-b0p3**: stochastic SAA with 16 scenarios per move, β=0.3 (moderate conservatism).  
  (You also ran SAA32/64 and Γ-robust in Step 10; the champion picker retained Q120 and SAA16-b0p3 as the best two under the ≥99% p95 rule.)

**6.2 Champion-level outcomes**

**Who won?**

From data/reports/champions.csv and the *Champion share by family* chart:

* **SAA16-b0p3 wins 54/56 instances** across all families (C, R, RC).
* **Q120 wins 2/56** (one in R, one in RC).
* DET never qualifies at the 99% p95 service level.

**Trade-off: cost vs reliability (champions only)**

See the **Method trade-off (champion set)** scatter.

* **Mean on-time p95**: both **SAA16-b0p3** and **Q120** are essentially **~100%**.
* **Mean distance** (from your scatter):
  + **SAA16-b0p3 ≈ 1 165**
  + **Q120 ≈ 1 218**  
    → SAA lowers cost by **~4–5%** at the same reliability on the final plans.

(Exact per-method means are also listed in data/reports/final\_overall\_table.csv.)

**Cost vs deterministic baseline (for the actual champions)**

Your **Histogram of champion %-distance improvement vs DET** shows a distribution centered around **−7 % to −8 %** with a left tail down to roughly **−15 %** on harder instances. That is: even after enforcing ≥99% on-time (p95), the robust champions typically **cut cost by ~7–8%** compared to the deterministic routes.

**Qualitative sanity check**

The **champion route plots** (e.g., *C102*, *R208*, *RC208*) show tighter, more locally cohesive tours than DET’s spoke-like patterns. This is consistent with SAA’s scoring: it rewards moves that preserve slack on arcs/times where traffic variability bites.

**6.3 Tables (per-instance and overall)**

* **Per-instance final table** (Step 12):  
  data/reports/final\_per\_instance\_table.csv  
  Columns: *instance, family, method, vehicles, distance, ontime\_p50, ontime\_p95, ontime\_mean, runtime\_s*.  
  Use this in the appendix; it is the authoritative “what we would ship per file” table.
* **Overall by method** (champions only):  
  data/reports/final\_overall\_table.csv  
  Columns: *mean\_distance, mean\_vehicles, mean\_ontime\_p50, mean\_ontime\_p95, mean\_runtime\_s, n\_instances*.  
  These are the exact values behind the scatter/claims above.

**6.4 What to deploy & when**

* **Default**: **SAA16-b0p3**. It dominates on the champion set (54/56 wins) with **≈100%** on-time p95 and **~4–5%** lower cost than Q-buffer at the same reliability.
* **When to use Q-buffer (Q120)**: if you need a one-knob method with the simplest runtime control (no scenario batches) or very tight compute budgets. It still achieves **≈100%** p95 on-time and captures a couple of edge cases.
* **Do not deploy DET** under variability: under the same scenario testing in Step 9, DET averages **~40%** on-time — well below any reasonable SLA — and is therefore not robust.

**6.5 Takeaways**

1. **Stochastic beats deterministic**: incorporating variability in the objective/feasibility (SAA) delivers **near-perfect on-time** and **lower cost**, not higher.
2. **Q-buffer is a strong baseline**: a single inflation factor already closes most of the gap to SAA with almost zero tuning.
3. **SAA wins breadth**: it generalizes across C/R/RC families (54/56 wins), while Q-buffer picks up the remaining outliers.
4. **Reproducibility**: every claim is backed by CSVs saved in data/reports/ and figures in data/figures/, generated from the 56 original CSVs without editing.

**6.6 Figures & tables to include**

* **Figure 1** – *Method trade-off (champion set)*: mean distance vs on-time p95 (from Step 13).
* **Figure 2** – *Champion share by family* (stacked bars).
* **Figure 3** – *Histogram: champion %-distance improvement vs DET*.
* **Figure 4–6** – *Champion route plots*: C102, R208, RC208 (representative).
* **Table A.1** – *Per-instance final results*: final\_per\_instance\_table.csv.
* **Table A.2** – *Overall by method*: final\_overall\_table.csv.

If you want the exact numbers reproduced inline in the text, copy them directly from final\_overall\_table.csv (means) and, when needed, from final\_per\_instance\_table.csv (for any specific instance you mention).

**Paste-ready “Results & Visuals” paragraph**

**Results & Visuals.** On the 56 original Solomon instances, the **champion set** (≥99% on-time p95) is dominated by **SAA16-b0p3 (54 wins)**, with **Q120 (2 wins)**. The *Method trade-off (champion set)* scatter shows both methods at **≈100%** on-time p95, with **SAA** at **lower mean distance** (≈**1165**) than **Q120** (≈**1218**). The *Champion share by family* bars confirm SAA’s breadth across **C**, **R**, and **RC**. The *Histogram of champion %-distance improvement vs DET* centers around **−7–8%**, indicating robust plans are **cheaper** than deterministic baselines while meeting the stringent service-level target. Route plots for *C102*, *R208*, and *RC208* illustrate the qualitative effect: robust routes are more compact and less “spiky,” aligning with reliability gains.

Here’s a tight, paste-ready **management abstract** you can drop near the front of your report:

**Management abstract.** Using the 56 Solomon VRPTW instances exactly as provided, we benchmarked a deterministic OR-Tools baseline against two robust approaches: a simple **Quantile Buffer** (Q120) and a **Stochastic SAA** method (SAA16-b0p3). Plans were stress-tested on the *same* 200 simulated traffic scenarios per instance. The deterministic baseline was unreliable (≈**40%** on-time on average), while both robust methods achieved ~**99–100%** on-time at the 95th percentile. On the final **champion set** (per-instance plans meeting the ≥99% p95 on-time target), **SAA16-b0p3** won **54/56** instances and delivered the **lowest average cost** (≈**1165** distance units) versus **Q120** (≈**1218**), i.e., ~**4–5%** cheaper at the same reliability. Overall, after enforcing the service-level target, champions also cut cost by ~**7–8%** versus the deterministic plans on the same instances. All artifacts are reproducible and shipped: champion JSON plans (data/champions/), per-instance results (data/reports/final\_per\_instance\_table.csv), and method-level aggregates (data/reports/final\_overall\_table.csv) alongside figures showing the cost-vs-reliability trade-off.