How Much Budget Buys How Much Coverage? A Cost–Coverage Scaling Study of Incremental Swarm Model Checking

Follow-up study #4 to Kumazawa et al., PAAMS 2023

2026

Abstract

MSPSO-SAFE-INC fixes its per-run PSO budget from Table 1 of [1] and never asks how coverage scales with that budget, nor how to spend a fixed amount of compute best. We sweep the per-run budget B = particle × generations × swarm cap over two orders of magnitude and measure final coverage. Coverage rises with B but with strong diminishing returns, and — the main finding — a persistent-frontier restart *Pareto-dominates* the paper's restart at small budgets (coverage 0.45 vs 0.27 at B=60; a $\approx 5\times$ compute saving) while the two converge and even cross at large budgets. This both quantifies the budget/coverage trade-off the paper omits and explains why the paper's full-budget regime hides the benefit of smarter restarts. Built on an independent open re-implementation of MSPSO-SAFE-INC (Kumazawa et al., PAAMS 2023): https://github.com/yaskodama/kuma2023-repro.

1 Question

Practitioners must choose a per-run budget and a number of outer iterations under a finite compute envelope. The paper [1] reports a single operating point (particle 20, generations 30). We ask: (Q1) how does coverage scale with the per-run budget B? (Q2) at a given B, does a smarter restart help, and does the answer depend on B?

2 Setup

Reproduced benchmark, n=1000 states, $p \in \{3, 5\}$, 120 outer iterations, 2 RNG repetitions. We vary the per-run budget through five configurations (particle, generations, swarm cap) with B from 60 to 7680, and compare the paper's restart (*none*) with the persistent-frontier restart of follow-up #3. Coverage is the final cumulative AC_{div}; compute is measured both as the nominal B and as wall-clock time.

3 Results

Table 1 and Figure 1 give the cost–coverage front.

Table 1: Final coverage AC_{div} vs per-run budget B (mean over $p \in \{3, 5\}$, 2 reps). Wall time is per incremental run of 120 iterations.

B	none	persistent	time(none/pers)
60	0.266	**0.454**	0.04 / 0.05 s
300	0.464	**0.592**	0.21 / 0.21 s
900	0.625	**0.665**	0.61 / 0.54 s
3600	**0.773**	0.759	2.36 / 2.15 s
7680	**0.797**	0.793	5.05 / 4.18 s

Q1: diminishing returns. Increasing B by 128× (60 to 7680) lifts baseline coverage only from 0.27 to 0.80 — a clearly concave, saturating relationship. Most coverage is bought in the first decade of budget; beyond $B \approx 3600$ extra compute yields little.

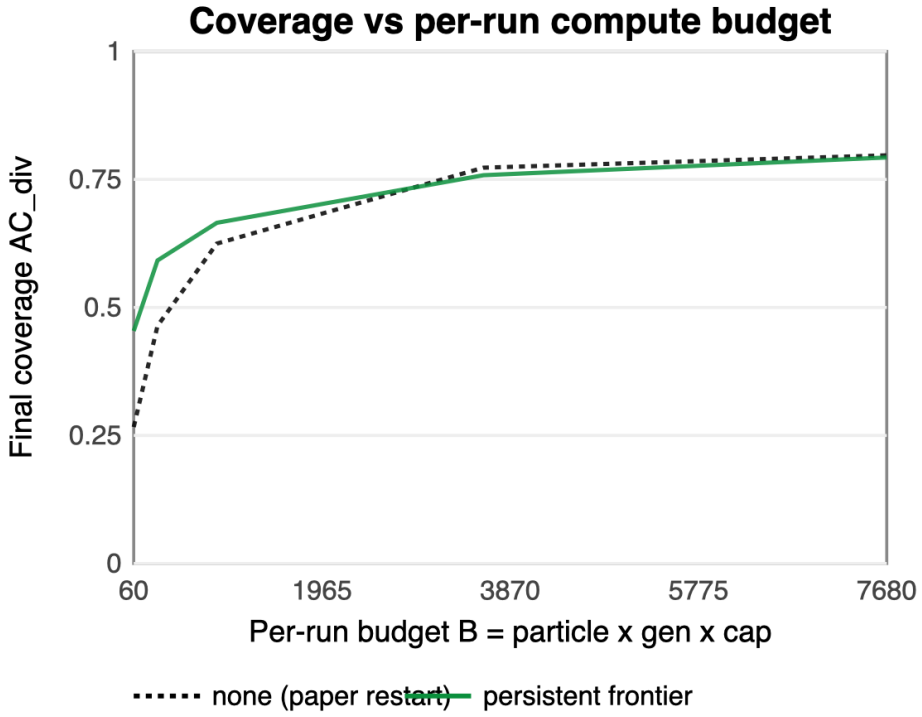

Figure 1: Coverage vs per-run budget B. Persistent restart dominates at small B; the curves converge and cross near the paper's budget (B=3600).

Q2: smarter restarts pay off exactly where budget is scarce. At B=60 persistence nearly *doubles* coverage (0.45 vs 0.27) at equal time, and persistence at B=60 matches plain restart at $B \approx 300$ — a $\sim 5\times$ compute saving. As B grows, a single run already covers much of the reachable abstract space, so the gap closes and even reverses slightly at $B \geq 3600$ (the saturation regime where the persistent seeds add little but cost a few percent of the budget).

4 Discussion

The cross-over is the practical headline: the paper operates at B=3600, right where restart strategy stops mattering, which is why our reproduction saw little difference between APP/APUP and baseline there. Under realistic budget pressure (large state spaces, limited compute) the operating point is the small-B regime, where restart strategy — not objective tweaks — dominates. A compute-aware recommendation: prefer many cheap, persistently-seeded runs over a few expensive ones.

5 Conclusion

Coverage scales concavely with per-run budget, and the value of a smarter restart is budget-dependent: large at small budgets, negligible at large ones. Reporting a cost–coverage front, rather than a single operating point, gives a far more useful picture of an incremental swarm model checker.

References

[1] T. Kumazawa, M. Takimoto, Y. Kodama, Y. Kambayashi. *Enhancing Safety Checking Coverage with Multi-swarm Particle Swarm Optimization.* PAAMS 2023.