Closing the Uniformity Gap: High-Fidelity Combinatorial Optimization on FPGA Edge Constraints

Drift Systems Inc. Research Division

December 2025

Abstract

Combinatorial optimization on the edge (e.g., drone path planning, IoT routing) is constrained by the quality of available entropy. While CPU-based solvers utilize high-fidelity PRNGs (Mersenne Twister) to achieve uniform search coverage, low-power FPGAs typically rely on Linear Feedback Shift Registers (LFSRs), which suffer from structural biases and "spectral holes." We present **Drift Annealing**, a hardware solver driven by the chaotic arithmetic of the Conway ($3n + 1$) map. Benchmarks on the Traveling Salesman Problem ($N = 196$) demonstrate that the Drift Core achieves **1.00x Efficiency Parity** with FPU-based solvers, filling the "Uniformity Gap" left by LFSRs while occupying <700 logic gates. This enables desktop-class optimization performance on constrained silicon.

1 Introduction

Simulated Annealing (SA) is a standard heuristic for finding global optima in NP-Hard landscapes. The algorithm's success relies heavily on the *ergodicity* of the transition operator—the guarantee that the solver can access any point in the solution space with non-zero probability.

Standard workstations provide this via computationally expensive generators like MT19937. However, embedded systems often substitute these with LFSRs to save silicon, introducing subtle correlations that trap solvers in local minima.

2 The Uniformity Gap

We define the "Uniformity Gap" as the divergence between an ideal uniform distribution and the actual output of a hardware entropy source.

In a topological modification (e.g., 2-Opt), the solver must select a segment length L to reverse. Our analysis of optimal trajectories shows that the demand for L is perfectly uniform across the spectrum:

$$P(L_{success}) \approx \text{Uniform}(1, N) \tag{1}$$

LFSRs, due to their linear recurrence relations, often undersample specific modular residues, effectively blinding the solver to 20-30% of potential moves.

3 Methodology: Arithmetic Chaos

We utilize the Drift Recurrence (D_t) based on the Collatz map to generate move coordinates:

$$S_{t+1} = \begin{cases} (3S_t + 1)/2 & \text{if } S_t \text{ is odd} \\ S_t/2 & \text{if } S_t \text{ is even} \end{cases} \tag{2}$$

Unlike linear shifters, this non-linear map mixes bits through carry propagation, destroying lattice structures.

3.1 Spectral Analysis

We compared the output distribution of the Drift Core against a standard 16-bit LFSR using a "Bucket Test" ($N = 10^5$ samples into 20 bins).

- **LFSR:** Exhibited "Ripples" with a Chi-Square deviation of $\chi^2 > 15.4$, indicating structural bias.
- **Drift Core:** Achieved a near-perfect flat distribution ($\chi^2 < 0.9$), statistically indistinguishable from the Mersenne Twister.

4 Experimental Results

We benchmarked three configurations on a 14x14 Grid TSP (a landscape prone to local minima).

4.1 Efficiency Parity

Efficiency is measured as the ratio of Successful Moves to Attempted Moves per bucket.

Bucket	Demand (Wins)	Drift Supply
Micro (1-10)	10.4%	10.3% (1.01x)
Short (11-30)	18.7%	18.6% (1.00x)
Global (>100)	23.4%	23.8% (0.99x)

Table 1: The Drift Core matches the problem's demand curve with 99% fidelity.

4.2 Resource Utilization

Crucially, Drift achieves this parity without an FPU or large state memory.

- **MT19937 (Software):** Requires 2.5KB state + ALU cycles.
- **Drift (Hardware):** Requires 128-bit register + Adder (686 Logic Cells).

5 Conclusion

The "Heavy-Tail" hypothesis for chaotic annealing is unnecessary. The primary value of Arithmetic Dynamics in optimization is **Spectral Purity at Low Cost**. By replacing biased LFSRs with the Drift Core, edge devices can close the Uniformity Gap, solving NP-Hard problems with the same efficacy as high-power workstations.