Autonomous Drone-Swarm Fire-Perimeter Tracing — Term Report

Authors: Zad Castañeda & Rhys Buckeye (UW Bothell)

Abstract

This quarter we extended a MASS-based multi-agent drone-swarm simulation from a basic exploration demo into a system that traces the perimeter of a synthetic wildfire using a hybrid **Particle Swarm Optimization (PSO) + Grasshopper Optimization Algorithm (GOA)** controller, with the swarm partitioned into dynamically-assigned **sub-swarms** to resist premature convergence. We built a ground-truth metrics framework, made the fire field and every algorithm coefficient runtime-configurable, and ran a 17-configuration parameter sweep across four fire shapes and the PSO/GOA/sub-swarm/size parameters. The sweep quantifies how each parameter affects perimeter coverage and surfaces several actionable findings — including that drone momentum (inertia) is essential for the approach phase, that the current multi-swarm machinery does not yet improve multi-fire coverage, and that run-to-run variance is large enough that single runs are insufficient for tuning decisions.

---

1. Objectives

The goals for the quarter were:

1. **Perimeter tracing** — move from "drones explore and find hot cells" to "drones settle on and trace the fire perimeter" (the band of cells at the fire's edge), the operationally useful behavior for wildfire monitoring.
2. **Diversity / anti-convergence** — stop the swarm from collapsing onto the first hot cell it finds, so it can cover an extended perimeter (and multiple disjoint fires) rather than a single point.
3. **Measurement** — replace eyeball assessment of the terminal animation with quantitative, ground-truth-scored efficiency metrics.
4. **Experimentation** — make the simulation parameterizable without recompiling, and use that to systematically characterize the algorithm.

This report documents the system built for (1)–(3) and presents the experimental study enabled by (4).

---

2. System Overview

2.1 Environment

The world is a **200×200 grid** of `GridCell` MASS *Places*. At initialization each cell computes a synthetic "true temperature" from a configurable field of 2-D Gaussian heat sources (Section 4.1). The perimeter the drones aim to trace is the **band of cells whose true temperature lies in `[SETTLE_TEMP, TOO_HOT)` = [70, 75)**: hot enough to be at the fire's edge, not so hot as to be inside the (impassable) core.

2.2 Drones and the control loop

40 `Drone` MASS *Agents* spawn clustered in one corner at `(10, 10)` — deliberately far from the centered fire, so reaching the perimeter is a real search problem rather than a trivial local settle. Each time step runs a fully decentralized cycle:

1. **EXPLORE** — read the current cell, update personal best, seed sensing data.
2. **Exchange #1** — each cell publishes its occupant + personal-best to its neighbors via MASS `exchangeAll`.
3. **BROADCAST** — the host pushes per-sub-swarm centroids and bests.
4. **PROPOSE** — each drone computes a desired next cell from PSO + GOA forces.
5. **Exchange #2** — cells publish movement *intents*.
6. **RESOLVE** — each drone reads neighbors' intents, breaks conflicts locally, and migrates one cell (Chebyshev step ≤ 1).

2.3 The PSO + GOA hybrid

A moving drone's velocity is a sum of three **PSO** attraction terms and a **GOA** repulsion term:

```
v ← INERTIA·v
    + COGNITIVE·r1·(personal_best − pos)     # PSO: pull to own best
    + SOCIAL·r2·(neighborhood_best − pos)     # PSO: pull to neighbors' best
    − Σ GOA_WEIGHT · (inverse-square push from each nearby drone)   # GOA:
spacing
```

- **PSO** drives the swarm toward hot regions. The social term uses the best *personal-best seen among neighbors* (neighborhood-best), not a single global best, which is what lets different parts of the swarm pursue different targets.
- **GOA** applies inverse-square repulsion between nearby drones so they spread out along the perimeter instead of stacking on one cell. Drones in a *different* sub-swarm repel each other extra hard (`INTER_SWARM_WEIGHT`).

Once a drone's cell temperature enters the perimeter band it **settles** and switches from PSO to a dedicated **perimeter-tracing** controller (follow the fire edge, repel from other settled drones to spread coverage). Settled cells are marked `has_settled` — this is the "traced" signal the metrics score.

2.4 Sub-swarms (multi-swarm PSO)

To fight PSO's tendency to converge prematurely onto one basin, the swarm is split into `NUM_SUBSWARMS` (default **4**) sub-swarms. Each step the host recomputes every sub-swarm's centroid from the drones' reports; each drone re-joins the nearest centroid (with hysteresis, `REASSIGN_MARGIN`, so labels don't thrash) — a decentralized k-means-style membership that emerges from spatial proximity. Inter-swarm repulsion then pushes the groups apart so they explore different arcs of the perimeter (or, in principle, different fires).

> Periodic *regrouping* (reshuffling/re-seeding stagnant sub-swarms) is designed but not yet implemented; it is the natural next step (Section 7).

---

3. Metrics Framework

3.1 Ground-truth perimeter P

Every run is scored against a **ground-truth perimeter set P** computed directly from the true-temperature grid: `P = { cells : |true_temp − 70| ≤ 3 }`. P is the band of cells at the fire's edge — hot enough to be perimeter, not so hot as to be inside the unreachable core. Because P is derived from the simulator's actual grid rather than the Gaussian formula, the same scoring logic works unchanged on any fire shape, including uploaded real fire maps. For the baseline Gaussian fire, |P| = 576 cells. P is the denominator in every recall metric, making it the standard against which all coverage is measured.

3.2 Two coverage notions: sensed vs. traced

Two distinct coverage types are tracked separately at every time step, reflecting the difference between *finding* part of the perimeter and *committing to* it:

Notion	Definition	Meaning
sensed	a cell any drone has visited (`discovered_temp ≥ 0`)	situational awareness
traced	a cell a *settled* drone has marked (`has_settled = 1`)	committed perimeter coverage

Traced is always a subset of sensed (traced ⊆ sensed): a drone cannot settle on a cell it has not already visited. Sensed recall is therefore always an upper bound on traced recall, and any improvement to traced performance must first come through improving sensing.

The two notions separate two distinct failure modes. A swarm with high sensed recall but low traced recall is *finding* the perimeter but failing to commit to it — drones fly over the edge without settling. A swarm with low traced precision is settling frequently but off-target — wasting settling events on core cells or cells outside the perimeter band.

3.3 Per-metric definitions and intuitions

The following metrics are computed per step and as end-of-run summaries:

Metric	Definition
sensed recall	`|sensed ∩ P| / |P|` — fraction of the true perimeter seen
traced recall	`|traced ∩ P| / |P|` — fraction actually traced
traced precision	`|traced ∩ P| / |traced|` — how much settling was on-perimeter
traced F1	harmonic mean of traced precision & recall
AUC (recall)	mean recall across all steps — rewards early coverage, not just end-state
t50 / t90 / t100	first step at which recall crosses 50 / 90 / 100% (−1 = never)
revisit ratio	`drone_steps / unique_sensed` — 1.0 = no wasted motion; higher = redundant

Sensed recall is the *ceiling metric*: if a cell has not been sensed, it cannot be traced. In the sweep, sensed recall ranged from 0.0 (inertia collapse) to 0.71 (K = 2). No configuration reached 90% sensed recall within 200 steps, primarily because the ~100-step approach phase from the corner spawn consumes roughly half the time budget before the first perimeter cell is reached.

Traced recall is the *primary mission metric* — not just finding the edge but marking it. Traced recall is always ≤ sensed recall. Across all successful configurations, traced recall ranged from 0.21 to 0.44, well below sensed recall, indicating the swarm consistently covers more ground than it commits to.

Traced precision penalizes off-target settling: a drone that parks inside the hot core or outside the edge band still counts as a settling event but contributes nothing to recall while pulling precision down. Precision of 1.0 means every settling event landed exactly on-perimeter. Across all successful runs, precision ranged from 0.58 to 0.72 — when drones settle they tend to land in approximately the right zone, though meaningful off-perimeter waste remains.

Traced F1 is the harmonic mean of precision and recall and serves as the headline single-number mission metric. The harmonic mean specifically penalizes configurations where *either* number is low: a configuration with 90% traced recall but only 10% precision scores F1 ≈ 0.18, not 0.50. F1 and AUC are the two primary comparison metrics in Section 5.

AUC (recall) is the time-averaged recall, computed separately for both sensed and traced. AUC rewards covering the perimeter *early*, not merely *eventually*: two runs that both reach 80% sensed recall by step 200 receive different AUC scores if one arrived at 80% by step 50 and the other only crossed 80% at step 195. AUC-sensed across the sweep ranged from 0.0 to 0.36; AUC-traced from 0.0 to 0.23. The low absolute values reflect the approach-phase problem: recall is near zero for the first ~100 steps while drones are still crossing open space, which drags the time-average well below the final-step value.

t50 / t90 / t100 are milestone timing metrics; −1 means the threshold was never reached within the run. In the current sweep, **all 17 configurations returned t90-sensed = −1 and t50-traced = −1**: no configuration achieved 90% sensing or 50% tracing within 200 steps. This is the sharpest quantitative statement of the approach-phase bottleneck — the 200-step budget is insufficient for the given spawn location and grid size regardless of algorithm tuning.

Revisit ratio measures flight efficiency: how many drone-steps were required on average to discover each new cell. A ratio of 1.0 would mean every step uncovered a previously unseen cell. The two-fire configuration reached 5.90, substantially higher than single-fire configurations (3.31–4.76), reflecting drones shuttling between basins without consistently committing to either. The inertia-collapse case (`pso_low_inertia`) reached 10.57 — drones with insufficient momentum churn near the spawn and cover almost nothing new.

3.4 Metric relationships

The metrics form a natural hierarchy. Sensed recall is the ceiling, measuring the outer boundary of what the swarm has found. Traced recall measures how much of that found boundary has been committed to; it is always ≤ sensed recall. Traced precision measures the quality of those commitments. F1 combines precision and recall into a single score that penalizes both missing perimeter cells and wasted settles. AUC adds the time dimension, rewarding swarms that achieve coverage quickly rather than just by the final step. Revisit ratio is orthogonal — it measures flight efficiency independently of coverage quality.

`Metrics` writes a per-step CSV and an end-of-run summary block per run. AUC and F1 are the headline numbers: AUC rewards fast coverage, F1 balances tracing the right cells against tracing enough of them.

---

4. Experimental Design

4.1 Runtime configurability (engineering enabler)

So a sweep could vary the algorithm without a recompile per run, we made the fire field and **every** PSO/GOA/structural parameter overridable by environment variable (load-time on the drone `.so`, defaults reproduce prior behavior exactly):

Knob	Env var(s)	Default
Fire shape	`FIRE_SHAPE` (`gaussian`/`ellipse`/`multi`), `FIRE_SIGMA_X/Y`, `FIRE_ANGLE`, `FIRE_SOURCES`, `FIRE_COMBINE`	centered Gaussian
PSO inertia	`PSO_INERTIA`	1.0
PSO cognitive	`PSO_COGNITIVE`	1.5
PSO social	`PSO_SOCIAL`	3.0
GOA repulsion	`GOA_WEIGHT`	2.0
Inter-swarm	`INTER_SWARM_WEIGHT`	4.0

Knob	Env var(s)	Default
repulsion		
Sub-swarm count	`NUM_SUBSWARMS`	4
Swarm size	`NUM_DRONES`	40
Run length	`TIME_STEPS`	200

4.2 Method

The study uses a **one-factor-at-a-time (OFAT)** design around a fixed baseline (the code defaults: INERTIA 1.0 / COGNITIVE 1.5 / SOCIAL 3.0, GOA_WEIGHT 2.0, INTER_SWARM_WEIGHT 4.0, K = 4, 40 drones). Group 1 sweeps the fire shape with the baseline algorithm; Groups 2–5 hold the shape at the baseline Gaussian and vary one knob, so each run's delta is attributable to a single factor. A final cross-cut tests multi-swarm vs. plain PSO on the two-fire map. All runs use 200 time steps on the 200×200 grid.

The whole matrix is driven by `src/run_experiments.sh` and aggregated by `tools/aggregate_results.py` (Appendix A).

4.3 Fire shapes tested

Shape	Configuration	Models	|P| (cells)
Gaussian	one centered blob (default)	a single roughly-circular fire	576
Ellipse	`FIRE_SIGMA_X=45 FIRE_SIGMA_Y=14 FIRE_ANGLE=30`	a wind-driven directional front	338
Two fires	`multi`, sources at (70,70) and (130,130)	two disjoint fires	424
Peanut	`multi`, `combine=sum`, two overlapping sources	a concave perimeter	392

---

5. Results

All numbers are from a **single run per configuration** at 200 steps. See Section 6.4 on variance — this caveat matters for interpretation.

5.1 Master table

Config	|P|	AUC sns	AUC trc	final sns R	final trc R	trc F1	revisit
shape_gaussian *(baseline)*	576	0.267	0.176	0.441	0.304	0.417	3.39
shape_ellipse	338	0.359	0.234	0.681	0.441	0.523	4.27
shape_twofire	424	0.276	0.139	0.455	0.208	0.305	5.90

Config	|P|	AUC sns	AUC trc	final sns R	final trc R	trc F1	revisit
shape_peanut	392	0.316	0.227	0.538	0.406	0.505	4.23
pso_social_heavy	576	0.316	0.189	0.517	0.319	0.426	4.03
pso_explore_heavy	576	0.321	0.181	0.589	0.340	0.440	4.71
pso_low_inertia	576	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	10.57
goa_weak	576	0.308	0.190	0.505	0.319	0.425	4.76
goa_strong	576	0.287	0.181	0.483	0.309	0.417	3.87
inter_off	576	0.328	0.195	0.557	0.337	0.439	4.12
inter_strong	576	0.278	0.174	0.469	0.306	0.419	4.22
k1_single	576	0.364	0.205	0.639	0.372	0.465	4.53
k2	576	0.354	0.205	0.710	0.424	0.496	3.94
k8	576	0.313	0.190	0.623	0.405	0.484	3.46
n20	576	0.161	0.141	0.273	0.245	0.365	3.31
n60	576	0.358	0.207	0.679	0.385	0.467	4.11
twofire_k1	424	0.272	0.135	0.441	0.210	0.308	5.98

(sns = sensed, trc = traced, R = recall. t50/t90/t100 omitted: no run reached 90% sensed or 50% traced recall — see 6.3.)

5.2 Fire shape (Group 1)

The **ellipse** is the easiest to cover (final sensed recall 0.68, F1 0.52) — it has the smallest perimeter (338 cells) and the swarm reaches 50% sensing fastest (t50 = step 127). The **two-fire** map is decisively the **hardest** (traced recall 0.21, F1 0.31, revisit 5.9): the swarm spawns in one corner and largely covers the nearer fire, leaving the far fire under-traced, and the wasted motion (revisit 5.9 vs. 3.4–4.3 for the others) reflects drones shuttling between basins. The **peanut** (concave) front sits in between (F1 0.51), tracing well despite its concavity.

5.3 PSO coefficients (Group 2)

- **Inertia is essential.** `pso_low_inertia` (INERTIA 0.4) is a **complete failure**: 0.0 recall on every metric and a revisit ratio of 10.6. With low momentum the velocity damps out before drones cross the ~100-cell gap from the corner spawn to the fire, so they mill near the start and never settle. This is the single most important sensitivity in the study.
- **More exploration helps coverage.** Both `pso_explore_heavy` (high inertia + cognitive) and `pso_social_heavy` beat the baseline on sensed recall (0.589 and 0.517 vs. 0.441), because spreading out exposes more of the perimeter. The baseline's social weight (3.0) is comparatively convergence-prone.

5.4 GOA repulsion (Group 3)

Repulsion has a **modest, mostly negative** effect on a single fire. `inter_off` (no inter-swarm repulsion) is the best of the group (sensed 0.557, F1 0.439), and `goa_weak` beats `goa_strong`. This is expected: with only one perimeter to cover, pushing sub-swarms apart and over-spacing drones scatters them off the single target. Inter-swarm repulsion is a multi-basin mechanism being charged a cost on a single-basin map.

5.5 Sub-swarm count K (Group 4) — and a variance warning

On the single Gaussian fire, **fewer or simpler sub-swarm structures did better than the K = 4 baseline**: k2 (sensed 0.710, F1 0.496) and k1/plain-PSO (0.639 / 0.465) and even k8 (0.623 / 0.484) all *outscored the K = 4 baseline* (0.441 / 0.417). But the ordering is non uniform (k2 > k1 > k8 > k4), which is not physically explainable by a "more/less fragmentation" story. The most likely explanation is run-to-run stochastic variance (PSO draws fresh random `r1`, `r2` every step): the K = 4 baseline run appears to be an unlucky draw. This is strong evidence that single runs are not enough to rank close configurations (Section 6.4).

5.6 Swarm size (Group 5)

The cleanest, most monotonic result: sensed recall rises **0.273 → 0.441 → 0.679** for **20 → 40 → 60** drones, with F1 and AUC rising in step. Coverage scales with swarm size, as expected, with no sign of congestion collapse at 60 drones on this grid. `n20`'s low revisit (3.31) reflects that too few drones simply leave the perimeter under-covered rather than redundantly covered.

5.7 Does multi-swarm help on two fires? (Cross-cut)

The headline test of the sub-swarm work. On the two-fire map, K = 4 (`shape_twofire`) and plain PSO (`twofire_k1`) are **statistically indistinguishable**: F1 0.305 vs. 0.308, traced recall 0.208 vs. 0.210, sensed 0.455 vs. 0.441. The multi-swarm machinery delivers no measurable benefit on the very scenario it was designed for — at least within a single 200-step run from a corner spawn. The likely cause: all sub-swarms spawn co-located in one corner and the nearer fire captures them before inter-swarm repulsion can split a group off toward the far fire. This is the central motivation for periodic regrouping (Section 7).

---

6. Discussion

6.1 What works

- **The perimeter-tracing pipeline functions end to end**: drones traverse a long approach, settle in the [70, 75) band, and mark traced cells with good precision (0.58–0.72 across every successful run — when drones settle, they settle on the *right* cells).
- **The metrics framework cleanly discriminates configurations** and exposes failure modes (e.g., the inertia collapse) that the animation alone would hide.

- **Coverage scales sensibly with swarm size**, the expected sanity check.

6.2 What doesn't (yet)

- **Multi-swarm provides no multi-fire benefit** in its current form (5.7).
- **Inter-swarm repulsion is a net cost on single fires** (5.4).
- **No run exceeds ~44% traced recall.** From a corner spawn, 200 steps is only enough to reach and partially trace the perimeter; the far side of every fire stays uncovered (all t90-sensed / t50-traced = never reached).

6.3 Coverage ceiling

The approach distance dominates the time budget. Drones spawn at (10,10); the Gaussian perimeter is a ring of radius ≈ 28 around (100,100), so the nearest perimeter cell is ~100 Chebyshev steps away. At one cell/step that is ~100 of the 200 steps spent merely *arriving*, leaving ~100 steps to spread around a 576-cell band. Higher traced recall will require either more steps, a less adversarial spawn, or a faster approach phase — not just better coefficients.

6.4 Stochastic variance (most important caveat)

PSO is stochastic, and the non-uniform K results (5.5) prove the per-config variance is large relative to the differences between configs. Several rankings in this report (e.g., which PSO/GOA variant is "best", whether K = 2 truly beats K = 8) are not trustworthy from single runs. The robust findings are the *large-effect* ones: the inertia collapse, the swarm-size monotonicity, the two-fire difficulty, and the multi-swarm null result. The close calls need replication before they drive design decisions.

---

7. Limitations & Future Work

1. **Replicate every configuration** (≥ 5 seeded runs) and report mean ± std. This is the single highest-value next step and directly addresses 6.4. (PSO currently seeds RNG nondeterministically; add a `SEED` env var.)
2. **Periodic regrouping** — reshuffle/re-seed stagnant sub-swarms every *K* steps so a group can break off toward an unattended fire. This is the designed-but-unbuilt mechanism that the two-fire null result (5.7) most directly motivates.
3. **Decouple approach from tracing** — a faster or staged approach (or a less adversarial spawn distribution) to lift the ~44% traced-recall ceiling (6.3).
4. **Re-tune the baseline.** Given that K = 2 and `inter_off` beat the K = 4 / inter-on defaults here, the defaults are likely mistuned for single fires; confirm with replicates, then consider shape-adaptive parameters.
5. **Validate on real fire maps** — the ground-truth metric already supports arbitrary maps; the synthetic shapes are a stand-in.

---

8. Conclusion

This quarter we turned an exploration demo into a measured fire-perimeter-tracing swarm: a decentralized PSO + GOA controller with dynamic sub-swarms, a ground-truth metrics framework, and a fully parameterized, no-recompile experiment harness. A 17-configuration sweep across four fire shapes and the PSO/GOA/structure/size knobs produced clear, actionable results: momentum is essential, coverage scales with swarm size, two disjoint fires are the hard case, and — importantly — the current multi-swarm design does not yet deliver its intended multi-fire advantage. The sweep also exposed that single-run comparisons are too noisy for fine tuning, setting up replication and periodic regrouping as the priorities for next quarter.

Appendix A — Reproduction

```
# build (env-tunable params, no recompile needed to vary configs)
cd ~/autonomous-swarm-core/src && source setup.sh && make clean && make build

# full sweep (17 runs, ~200 steps each) → experiments/*.log, *.csv,
summary.csv
nohup ./run_experiments.sh <netid> '<password>' > experiments/sweep.out 2>&1
&

# re-aggregate existing logs without re-running
python3 ../tools/aggregate_results.py experiments experiments/summary.csv

# single config
ONLY=shape_twofire ./run_experiments.sh <netid> '<password>'
```

Appendix B — Configuration catalog

Name	Overrides (baseline otherwise)
shape_gaussian	`FIRE_SHAPE=gaussian`
shape_ellipse	`FIRE_SHAPE=ellipse FIRE_SIGMA_X=45 FIRE_SIGMA_Y=14 FIRE_ANGLE=30`
shape_twofire	`FIRE_SHAPE=multi FIRE_SOURCES=70,70,20;130,130,20`
shape_peanut	`FIRE_SHAPE=multi FIRE_COMBINE=sum FIRE_SOURCES=85,100,24;115,100,24`
pso_social_heavy	`PSO_SOCIAL=5.0 PSO_COGNITIVE=0.5`
pso_explore_heavy	`PSO_INERTIA=2.0 PSO_COGNITIVE=3.0 PSO_SOCIAL=1.0`
pso_low_inertia	`PSO_INERTIA=0.4`
goa_weak	`GOA_WEIGHT=0.5`
goa_strong	`GOA_WEIGHT=5.0`
inter_off	`INTER_SWARM_WEIGHT=0.0`
inter_strong	`INTER_SWARM_WEIGHT=8.0`

Name	Overrides (baseline otherwise)
k1_single / k2 / k8	`NUM_SUBSWARMS=1 / 2 / 8`
n20 / n60	`NUM_DRONES=20 / 60`
twofire_k1	two-fire map with `NUM_SUBSWARMS=1`

Baseline: INERTIA 1.0, COGNITIVE 1.5, SOCIAL 3.0, GOA_WEIGHT 2.0, INTER_SWARM_WEIGHT 4.0, NUM_SUBSWARMS 4, NUM_DRONES 40, TIME_STEPS 200, 200×200 grid, spawn (10,10), perimeter band [70, 75).