# Population-Based Optimizers

Variational Quantum Algorithms · Module 6 — Optimization Dynamics

Node 6.7

## Position in Knowledge Graph

**System:** Variational Quantum Algorithms **Structural Domain:** Optimization Dynamics **Node:** 6.7

A point-based optimizer (everything in 6.1-6.5) maintains a single parameter vector and updates it based on local information. A **population-based optimizer** maintains a *set* of parameter vectors — a population — and updates them collectively.

This is a different optimization paradigm. Instead of "follow the gradient," it's "explore via diversity." Instead of "find a critical point," it's "concentrate the population on good regions of parameter space."

Population methods are the natural fit for VQA in two specific situations:

1. **Multi-modal landscapes** where a single optimizer would get stuck in a local minimum.
2. **Plateau regions** where individual optimizers can't make progress, but a population can collectively discover non-plateau regions.

This node introduces the population-based framework, walks through the major algorithm families (genetic algorithms, evolution strategies, particle swarm, CMA-ES), and sets up node 6.8 (HOPSO, the thesis's preferred hybrid).

---

## Why Populations?

The basic argument for populations:

**1. Implicit parallelism.** A population of $N$ particles samples $N$ different parameter regions simultaneously. Even if any single particle gets stuck, the population as a whole has more information about the landscape.

**2. Diversity preservation.** Population methods can be designed to maintain diversity (spread among particles), which prevents premature convergence to a single minimum. This is the population analogue of entropy regularization.

**3. No gradient required.** Most population methods are zero-order — they only need cost evaluations, no gradients. This makes them robust to non-differentiable cost functions, discrete parameters, and simulation-based settings.

**4. Robust to noise.** Aggregating over a population reduces the influence of noise on any individual evaluation. The population's mean (or median) is a more robust statistic than any single particle's cost.

**5. Embarrassingly parallel.** The $N$ cost evaluations per iteration are independent and can be run in parallel on $N$ quantum devices (or sequentially with no synchronization).

The cost: each iteration evaluates $N$ costs instead of 1, so per-iteration shot budget is $N$ times higher. The benefit needs to outweigh this cost.

---

## The Algorithm Families

Population methods come in several distinct families:

### 1. Genetic Algorithms (GA)

- Maintain a population of "individuals" (parameter vectors).
- Each generation: evaluate fitness, select the best, produce offspring via mutation and crossover.
- Inspired by biological evolution.

**Strengths:** very general, work on discrete and combinatorial problems. **Weaknesses:** slow, lots of hyperparameters, hard to tune.

### 2. Evolution Strategies (ES)

- Similar to GA but use Gaussian mutations (no crossover).
- The current "best" is updated by averaging successful mutations.
- More principled than GA, with cleaner convergence properties.

**Strengths:** simpler than GA, scales better, has theoretical foundations. **Weaknesses:** still slow compared to gradient methods on smooth problems.

### 3. CMA-ES (Covariance Matrix Adaptation Evolution Strategy)

- Hansen and Ostermeier (2001).
- The current "gold standard" of evolution strategies.
- Maintains a *covariance matrix* over parameters, adapted online to capture the local landscape geometry.
- Effectively learns a natural-gradient-like preconditioner from the population statistics.

**Strengths:** very robust, handles ill-conditioning, often competitive with gradient methods on noisy problems. **Weaknesses:** $O(P^2)$ covariance matrix, expensive for large $P$.

### 4. Particle Swarm Optimization (PSO)

- Eberhart and Kennedy (1995).
- Particles move through parameter space with velocities updated based on:
  - Inertia (continue current direction).
  - Personal best (return to best point seen).
  - Global best (move toward best point in population).
- Inspired by flocking behavior of birds.

**Strengths:** simple, intuitive, easy to implement. **Weaknesses:** can prematurely converge, requires careful inertia tuning.

### 5. Differential Evolution (DE)

- Storn and Price (1997).
- Update each particle by adding the difference between two random other particles.
- Simple and effective on rugged landscapes.

**Strengths:** simple, robust, few hyperparameters. **Weaknesses:** slow on smooth problems, no built-in noise handling.

### 6. Bayesian Optimization (BO)

- Maintain a Gaussian process posterior over the cost function.
- At each iteration, query the point that maximizes an acquisition function (e.g., expected improvement).
- Population is implicit (the GP uses all past observations).

**Strengths:** very data-efficient, principled handling of uncertainty. **Weaknesses:** $O(k^3)$ cost in number of past observations, doesn't scale to high $P$.

For VQA, **CMA-ES** and **PSO** (and the HOPSO variant in 6.8) are the most commonly used population methods. **BO** is used when shots are extremely expensive (real hardware experiments with limited time).

---

## CMA-ES In Detail

CMA-ES is the canonical sophisticated evolution strategy. Here's the core update.

State: a mean vector $\mathbf{m}_k \in \mathbb{R}^P$, a covariance matrix $C_k \in \mathbb{R}^{P\times P}$, and a step size $\sigma_k > 0$.

At each generation:

1. **Sample.** Draw $\lambda$ candidate solutions $\boldsymbol{\theta}_k^{(i)} \sim \mathcal{N}(\mathbf{m}_k, \sigma_k^2 C_k)$ for $i = 1, \ldots, \lambda$.
2. **Evaluate.** Compute $C(\boldsymbol{\theta}_k^{(i)})$ for each candidate.
3. **Select.** Sort by cost and keep the top $\mu < \lambda$ (the "elite").
4. **Update mean.** $\mathbf{m}_{k+1} = \sum_{i=1}^{\mu} w_i \boldsymbol{\theta}_k^{(i)}$, a weighted average of the elite.
5. **Update covariance.** Adapt $C_k$ using the elite's distribution: increase variance in directions where the elite spread out, decrease in directions where it concentrated.
6. **Update step size.** Adapt $\sigma_k$ based on the recent successful directions.

The covariance adaptation is the magic: over many generations, $C_k$ converges to (a scalar multiple of) the local Hessian inverse, so the sampling distribution becomes aligned with the landscape geometry. CMA-ES is, in this sense, a population-based approximation to natural gradient — and it doesn't need any gradient computations.

For a detailed treatment, see Hansen's tutorial (2016).

---

## CMA-ES For VQA

CMA-ES has been used for VQA in several papers, with mixed results:

**When it works:** - $P < 50$ (so the covariance matrix is manageable). - The cost landscape is smooth and the local structure is approximately quadratic. - Shot noise is moderate (CMA-ES handles it via population averaging).

**When it doesn't:** - Large $P$ (covariance matrix becomes prohibitive). - Strong barren plateaus (CMA-ES gets confused when there's no signal). - Very rugged landscapes (the covariance adaptation lags behind the local structure).

For typical VQE problems with 10-30 parameters, CMA-ES is a competitive alternative to Adam, sometimes finding lower minima at the cost of more total cost evaluations. For larger problems, it's typically infeasible.

---

## Particle Swarm For VQA

PSO is simpler and scales better but has its own issues:

**Update rule for particle $i$:**

$$\mathbf{v}_i^{k+1} = w\mathbf{v}_i^k + c_1 r_1(\mathbf{p}_i - \boldsymbol{\theta}_i^k) + c_2 r_2(\mathbf{g} - \boldsymbol{\theta}_i^k),$$
$$\boldsymbol{\theta}_i^{k+1} = \boldsymbol{\theta}_i^k + \mathbf{v}_i^{k+1},$$

where $w$ is the inertia, $c_1, c_2$ are acceleration coefficients, $r_1, r_2$ are uniform random numbers, $\mathbf{p}_i$ is particle $i$'s personal best, and $\mathbf{g}$ is the global best.

**Strengths:** - Very simple, no covariance matrix. - Scales linearly in $P$. - Easy to interpret and modify.

**Weaknesses:** - Can prematurely converge to the global best, losing diversity. - Sensitive to inertia/acceleration tuning. - Doesn't adapt to landscape geometry like CMA-ES does.

For VQA with many parameters, PSO is one of the few population methods that remains tractable. It is the foundation for HOPSO (Section 6.8), which is the thesis's specific PSO variant.

---

## The Diversity Problem

A central tension in population methods: **how to maintain diversity without losing convergence speed**.

Too much diversity: the population explores uniformly but never converges. Too little diversity: the population collapses to a single point (effectively becoming a point optimizer) and loses the benefits of being a population.

Mechanisms to maintain diversity:

- **Niching:** explicitly assign different particles to different "niches" of parameter space.
- **Crowding:** penalize particles that get too close to each other.
- **Mutation rate:** keep mutation high enough to spread the population.
- **Restart:** periodically reset some particles to random points.
- **Multi-modal selection:** in CMA-ES, use multi-modal variants that maintain multiple covariance matrices.

For VQA, the right diversity strategy depends on whether the landscape is multi-modal (use diversity-preserving variants) or unimodal but plateau-prone (use restart-based variants).

---

## Population Methods And Stochastic Optimization Theory

Population methods don't fit the classical stochastic optimization framework cleanly. The convergence theory for ES, CMA-ES, and PSO is its own subfield, with weaker guarantees than SGD theory.

That said, several recent results have shown that **CMA-ES is approximately equivalent to stochastic natural gradient on a Gaussian distribution** (Akimoto et al., Ollivier et al.), and that **PSO is approximately equivalent to a discrete-time approximation of the Wasserstein gradient flow** (Section 6.6).

This gives a theoretical foundation that connects population methods to the rest of the optimization literature, even though they look quite different at the algorithm level.

---

## Structural Role

This node is the **distributional / population** entry point of Module 6. It introduces the population-based paradigm, surveys the major algorithms, and sets up node 6.8 (HOPSO) as the specific hybrid that the thesis advocates.

It also forwards directly to Module 5 (regularization), where many regularization techniques have natural population-based interpretations.

---

## Relations to Other Concepts

- **Hansen 2016 (CMA-ES tutorial)** — the canonical reference for CMA-ES.
- **Eberhart-Kennedy 1995 (PSO)** — the original particle swarm paper.
- **Ollivier et al. 2017** — connecting CMA-ES to natural gradient.
- **Wasserstein gradient flow (6.6)** — the continuous limit of population dynamics.
- **Module 5 regularization** — population methods naturally implement many regularization ideas.

---

## Worked Example — PSO On A 2D Cost

Consider $C(\theta_1, \theta_2) = (\theta_1 - 1)^2 + (\theta_2 - 2)^2$, with the minimum at $(1, 2)$.

**Setup:** 5 particles, initialized uniformly at random in $[-3, 3]^2$. Inertia $w = 0.7$, acceleration $c_1 = c_2 = 1.5$, velocities initialized to zero.

**Iteration 1:** - Particle positions: $(1.5, -2), (-2, 1), (0, 0), (3, 3), (-1, -3)$. - Costs: $16.25, 10, 5, 5, 29$. - Global best so far: $(0, 0)$ or $(3, 3)$, tie at cost 5. Pick $(0, 0)$. - Each particle's personal best is its current position.

**Iteration 2:** - Each particle accelerates toward $(0, 0)$ (global best) and toward its personal best (which is its current position, so no contribution). - New velocities: $w \cdot 0 + c_1 r_1 \cdot 0 + c_2 r_2(\mathbf{g} - \boldsymbol{\theta}_i)$ — random pull toward the global best. - Particles move toward $(0, 0)$, but with noise.

**Iteration 3-10:** - The particles cluster around the current global best, but explore the area. - One eventually finds $(1, 2)$ (the true minimum) and becomes the new global best. - The rest of the population follows.

**Convergence:** typically $\sim 30$ iterations for 2D problems, $\sim 200$ for 10D problems, $\sim 1000+$ for higher dimensions. The cost per iteration is 5 evaluations (for 5 particles), so the total cost is 5x the number of iterations.

For VQA, this scales linearly in the number of particles (which gives the population effect) and roughly as a polynomial in $P$ (PSO doesn't suffer from the curse of dimensionality as badly as some methods).

---

## Visualization

*Pending. A 2D contour plot of a multi-modal landscape with 5 PSO particle trajectories overlaid. Initial random scatter, then gradual convergence on the global minimum, with some particles briefly visiting local minima. Caption: "Particles explore. Eventually, they vote with their feet."*

---

## Exercises

1. For PSO with 3 particles on $C(\theta) = \theta^2$, starting at $\theta = -2, 0, 1$ with zero velocities, run 5 iterations by hand and verify convergence toward $\theta = 0$.
2. CMA-ES adapts a covariance matrix. For a quadratic cost $C = \boldsymbol{\theta}^T A \boldsymbol{\theta}$, what does the converged covariance matrix look like in terms of $A$?
3. (VQA) For a 100-parameter VQA, estimate the per-iteration shot budget for: (a) a vanilla parameter-shift gradient; (b) CMA-ES with 50 particles; (c) PSO with 50 particles. Which is most expensive?

---

## Research Extensions

- **Quantum-aware population methods** — variants that exploit VQA structure (periodicity, parameter-shift exact gradients).

- **Hybrid gradient + population methods** — using parameter-shift gradients to inform PSO or ES updates.
- **Diversity-aware CMA-ES for plateau escape** — variants that maintain multi-modal populations on barren plateau landscapes.
- **Bayesian optimization with quantum priors** — incorporating quantum-specific structure into the GP prior.

---

## Cross-links to Other Courses

- **Evolutionary Computation** — the broader field of bio-inspired optimization.
- **Bayesian Optimization** — Gaussian process optimization, expected improvement.
- **Module 6 Section 6.6 (Optimal Transport)** — the continuous-limit framework.
- **Module 6 Section 6.8 (HOPSO)** — the specific PSO variant.

---

*Variational Quantum Algorithms · Module 6 — Optimization Dynamics · Node 6.7*
**Concepts:** stochastic-optimization, optimization-landscape, convergence
**Prerequisites:** 6.1, 6.4, 6.6
**Enables:** 6.8

*hbar.university — own.your.cognition*