Meta-Optimization of Particle Swarm Optimization

by

Magnus Erik Hvass Pedersen
What is Optimization?

• An optimization problem is a function $f$ mapping candidate solutions to a fitness measure:

$$f: \mathbb{R}^n \rightarrow \mathbb{R}$$

• The optimal solution $\hat{z} \in \mathbb{R}^n$ has the best fitness and therefore satisfies the equation (minimization):

$$\forall \hat{y} \in \mathbb{R}^n: f(\hat{z}) \leq f(\hat{y})$$

• Searching for the optimal solution can be done by iteratively following the gradient $\nabla f$

• If $f$ is not differentiable or the gradient is unknown then $f$ is a ‘black-box’ and we can use heuristic optimization.
Particle Swarm Optimization (PSO)

Pseudo-code:

• Initialize each particle $\mathbf{x}$ to a random position in the search-space and give it a random velocity $\mathbf{v}$.
• Until a termination criterion is met, update the velocity and position for each particle using the formulas:

\[
\mathbf{v} \leftarrow \omega \mathbf{v} + \phi_p r_p (\mathbf{p} - \mathbf{x}) + \phi_g r_g (\mathbf{g} - \mathbf{x})
\]

\[
\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}
\]

• Where $r_p$ and $r_g$ are random numbers, $\mathbf{p}$ is the particle’s best known position and $\mathbf{g}$ is the swarm’s best position.
Behavioural Parameters

- PSO parameters: $S$ (number of particles), $\omega$, $\phi_p$ and $\phi_g$
- Greatly influence optimization performance.
- May be determined by trial-and-error (hand-tuning), adaptive and self-adaptive schemes.
- As the number of parameters increases, the number of possible combinations increases exponentially, this is the Curse of Dimensionality for the parameter space.
- We need an efficient way of searching for the best parameters for an optimizer, that is, we need an additional layer of optimization.
Search-Space of PSO Parameters
Meta-Optimization

Meta-Optimizer
Discover behavioural parameters of Optimizer

Optimizer (PSO)
Discover solutions to Actual Problem(s).

Actual Problem(s)
Meta-Optimization

Pseudo-code:

- Initialize the meta-optimizer with a random choice of behavioural parameters for the PSO optimizer.
- Repeat the following a number of times:
  - Conduct optimization runs with PSO using the given choice of behavioural parameters.
  - Sum the results of the PSO runs to form a meta-fitness measure of its performance.
  - Modify the behavioural parameters using the optimization methodology of the meta-optimizer.
Meta-Optimization

Challenges and solutions:

- Very time-consuming because each meta-level iteration consists of evaluating the performance of the optimizer.
- The LUS method is used as meta-optimizer because it is simple yet often discovers optima within a few iterations; especially for smoother problems such as the search-spaces of behavioural parameters.
- Pre-emptive Fitness Evaluation aborts a meta-fitness evaluation once it becomes known that it does not lead to an improvement.
- Together these limit time-usage to minutes or hours depending on problem settings.
Meta-Optimized Parameters

• Good parameters depend on problem settings used in meta-optimization.

• For example, meta-optimization using 12 benchmark problems in 20 dimensions and allowing 40,000 fitness evaluations finds these PSO parameters to be good:

\[ S = 69 \quad \omega = -0.44 \quad \phi_p = -0.27 \quad \phi_g = 3.40 \]

• Compare to hand-tuned parameters:

\[ S = 50 \quad \omega = 0.729 \quad \phi_p = \phi_g = 1.4944 \]

• Note that the tuned parameters \( \omega \) and \( \phi_p \) are negative, which contradicts conventional wisdom about how PSO works.
Performance Comparison

Rosenbrock

Schwefel 1-2

Schwefel 2-21

Schwefel 2-22
References

Good Parameters for Particle Swarm Optimization

Tuning & Simplifying Heuristical Optimization

SwarmOps source-code C, C#, Java

www.hvass-labs.org