CLOP: Confident Local Optimization for Noisy Black-Box Parameter Tuning

Rémi Coulom

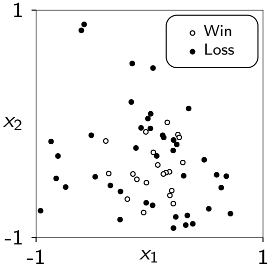
November, 2011

Advances in Computer Games 13

Noisy Black-Box Optimization

Problem: Optimizing a game-playing Program

- Heuristic parameters: evaluation, search, . . .
- Observation: game outcomes (win or loss (or draw))
- Objective: maximize probability of winning



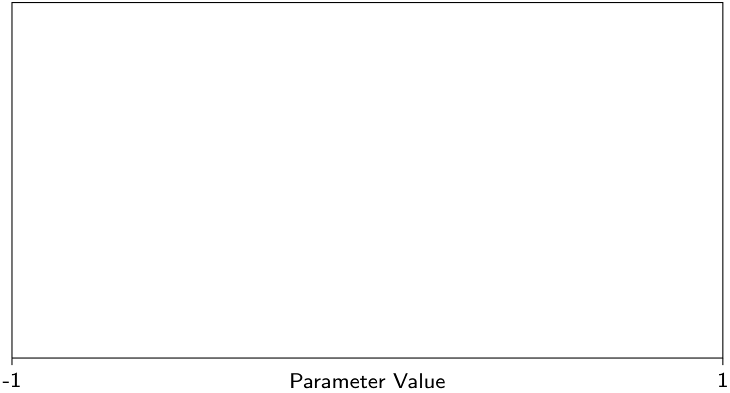
Two sub-problems

1. Estimate optimal $\vec{x}$
2. Choose next $\vec{x}$

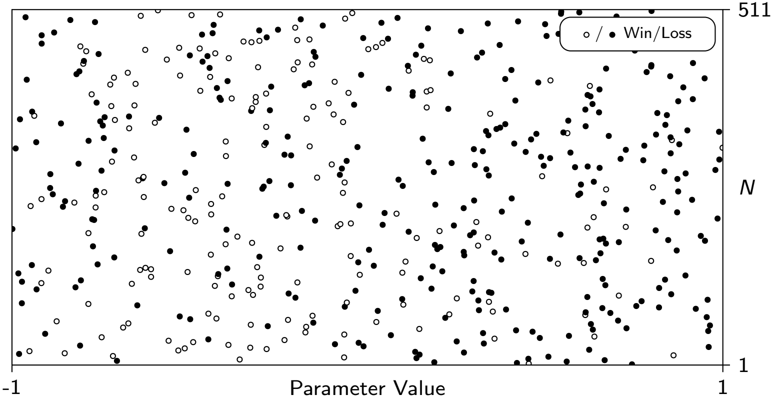
Presentation Outline

- CLOP algorithm
- Experiments on artificial data

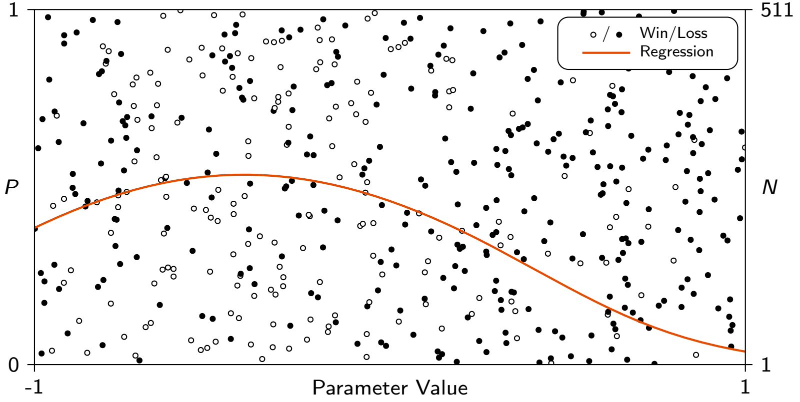
CLOP: Method for Optimum Estimation



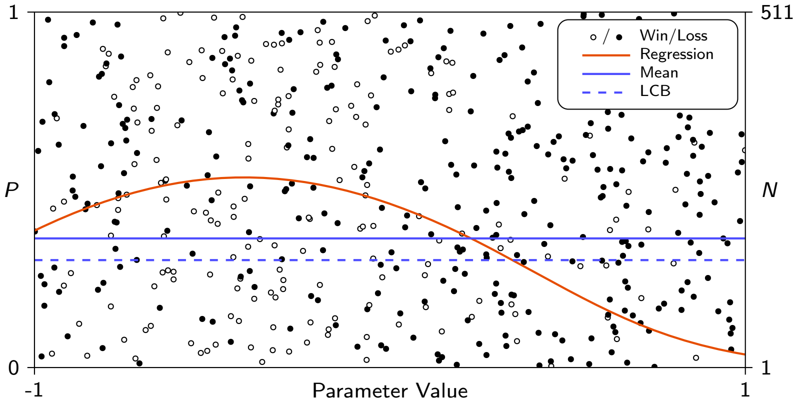
Step 1: Data



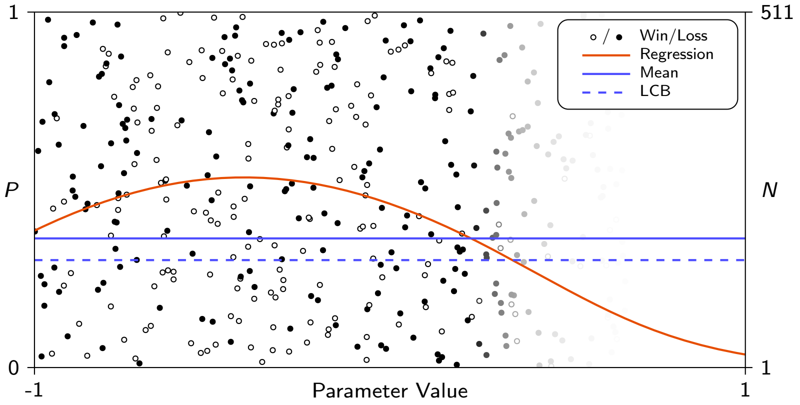
Step 2: Quadratic Logistic Regression



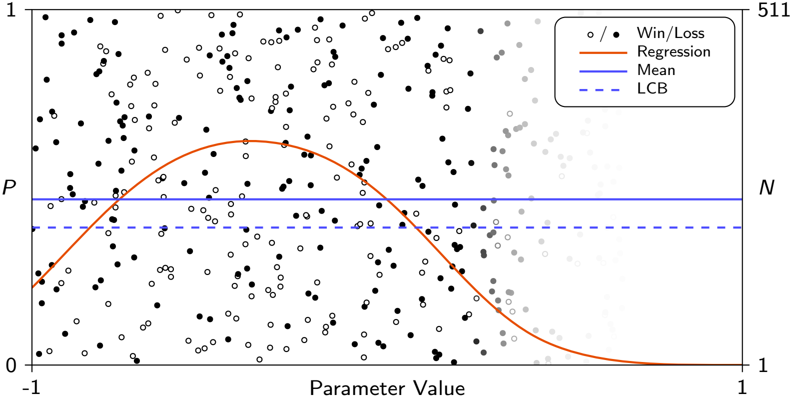
Step 3: Lower Confidence Bound



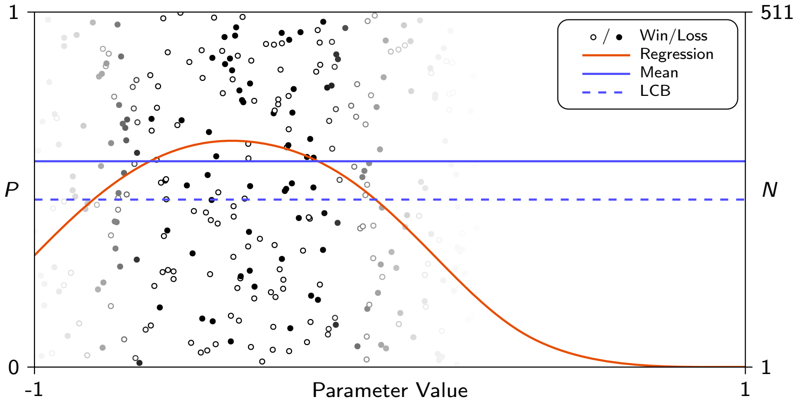
Step 4: Discard Samples Below LCB $= \mu - H\sigma$



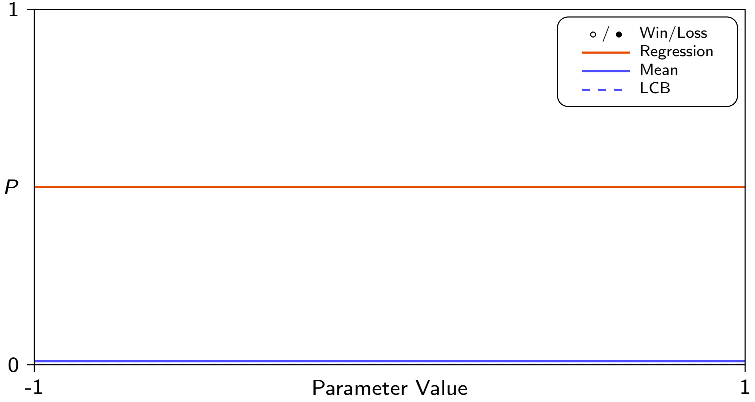
Step 5: Re-compute Regression with Remaining Samples



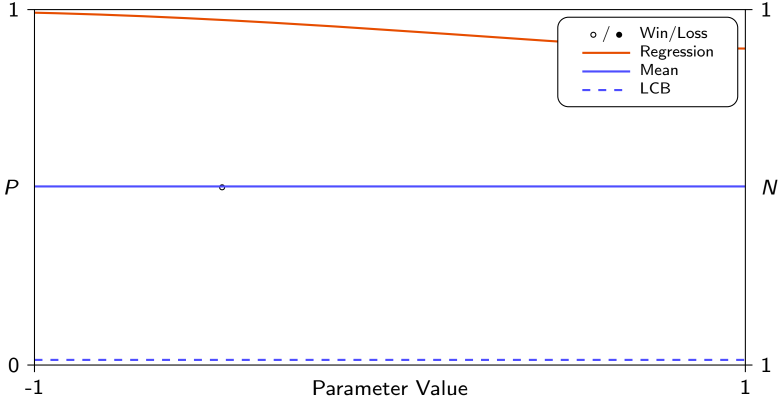
Step 6: Iterate Until Convergence



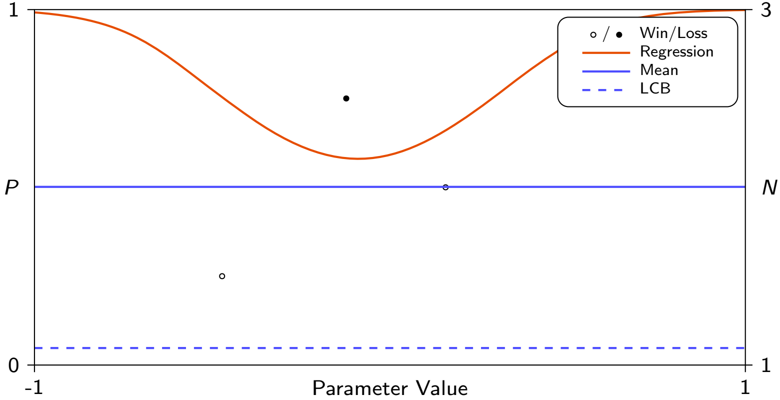
Sampling Policy: Density = Weight



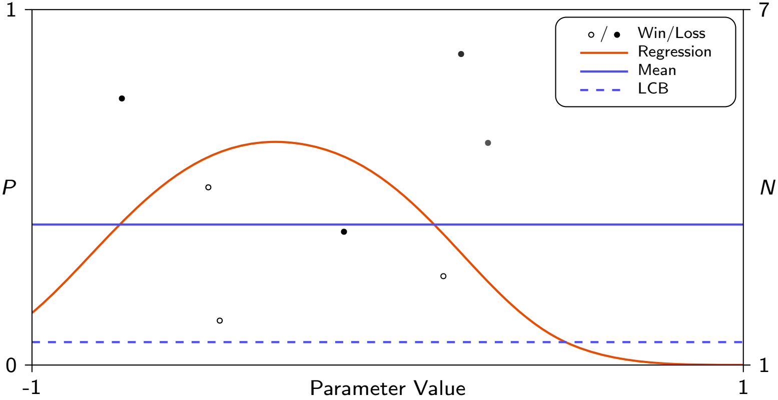
Sampling Policy: Density = Weight



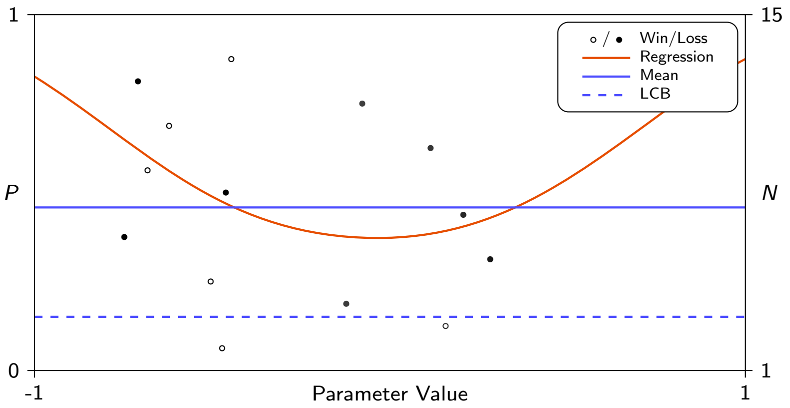
Sampling Policy: Density = Weight



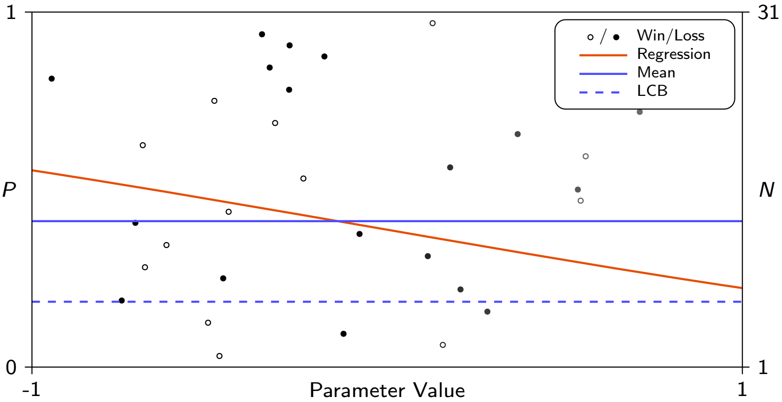
Sampling Policy: Density = Weight



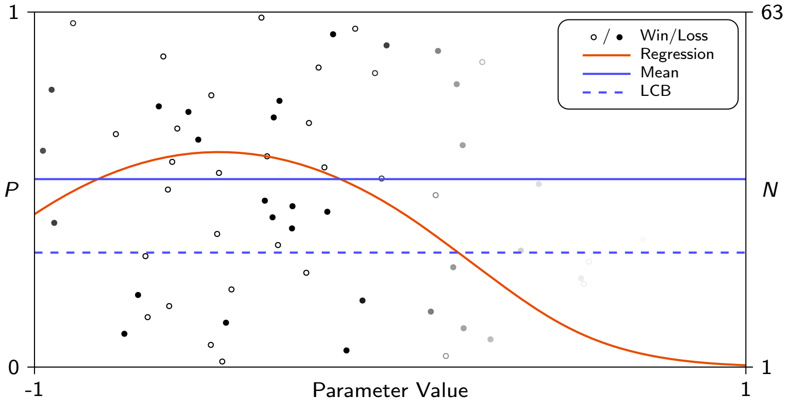
Sampling Policy: Density = Weight



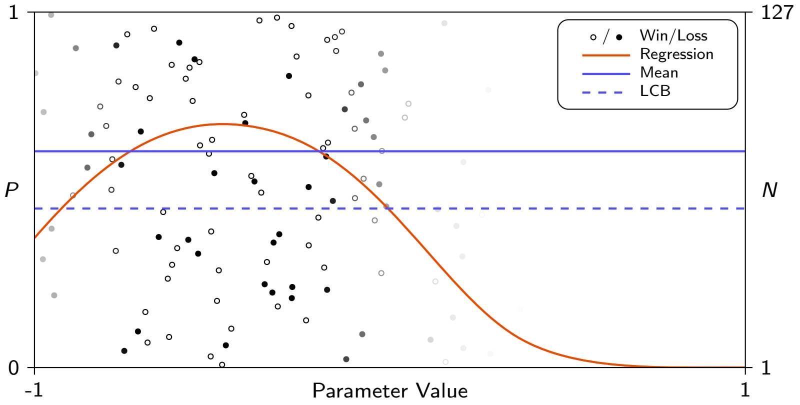
Sampling Policy: Density = Weight



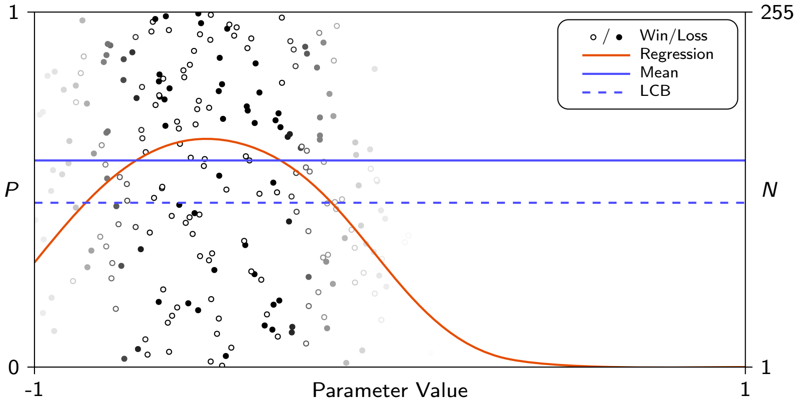
Sampling Policy: Density = Weight



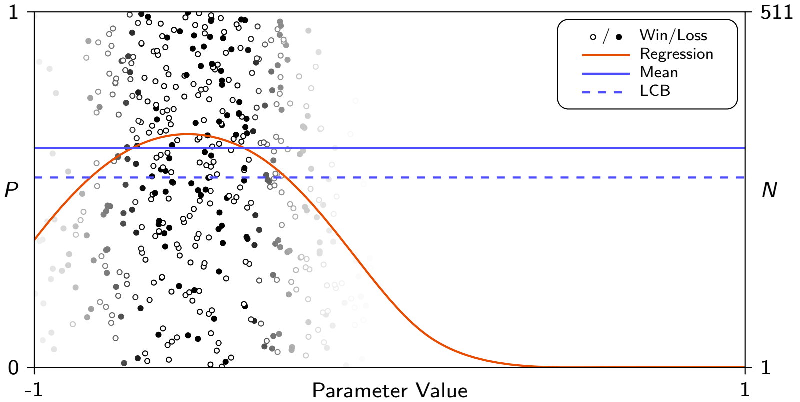
Sampling Policy: Density = Weight



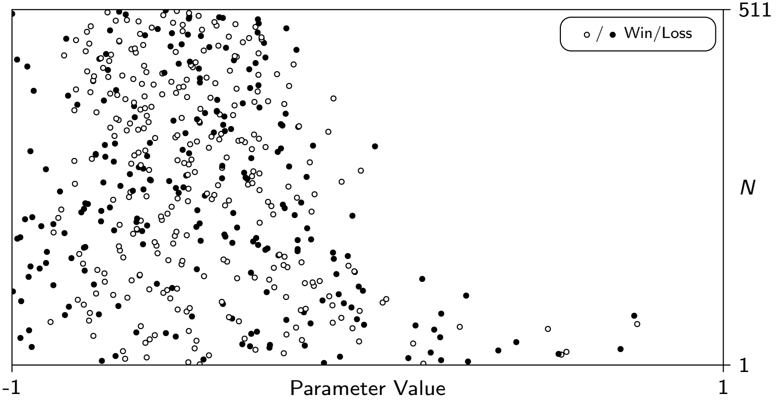
Sampling Policy: Density = Weight



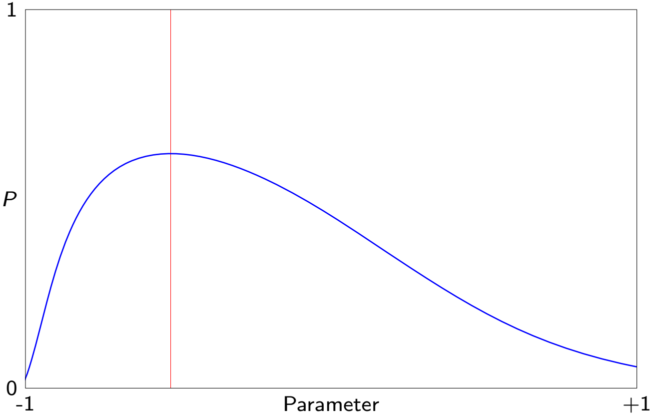
Sampling Policy: Density = Weight



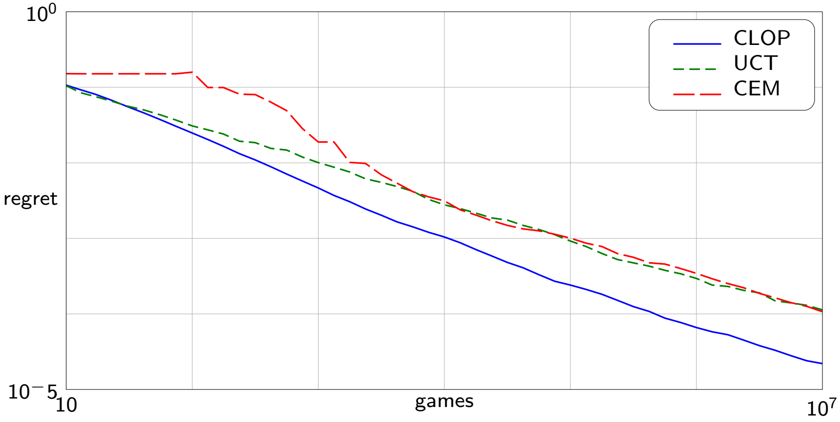
Sampling Policy: Density = Weight



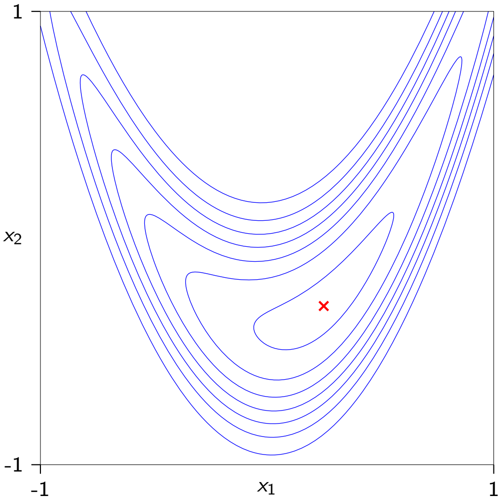
Smooth 1D function



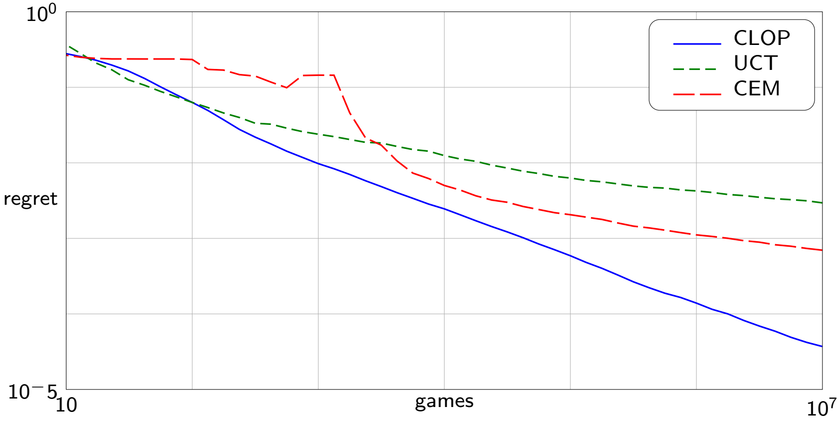
Smooth 1D function



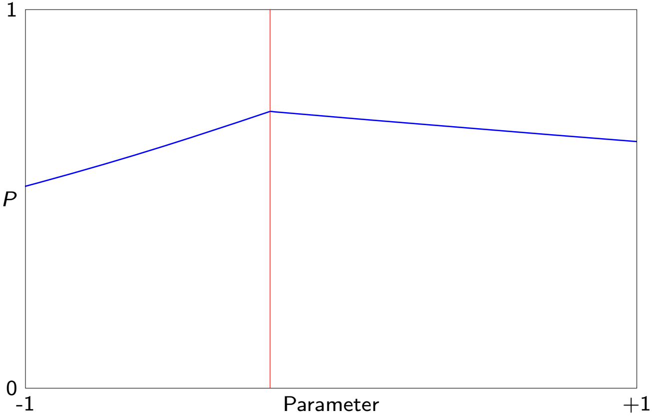
Rosenbrock



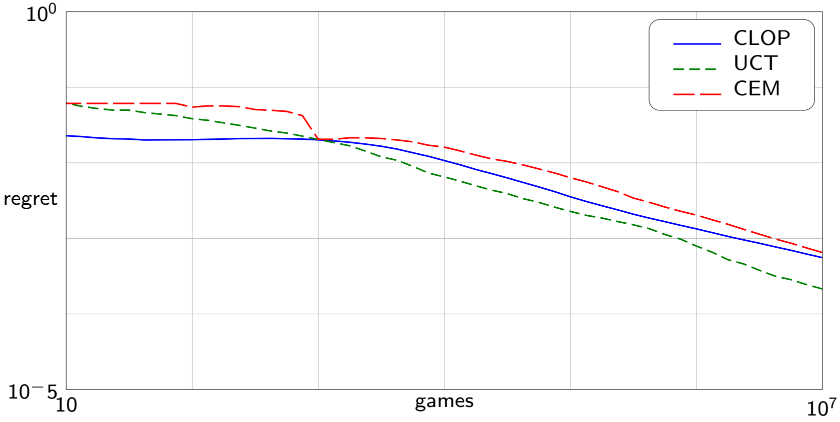
Rosenbrock



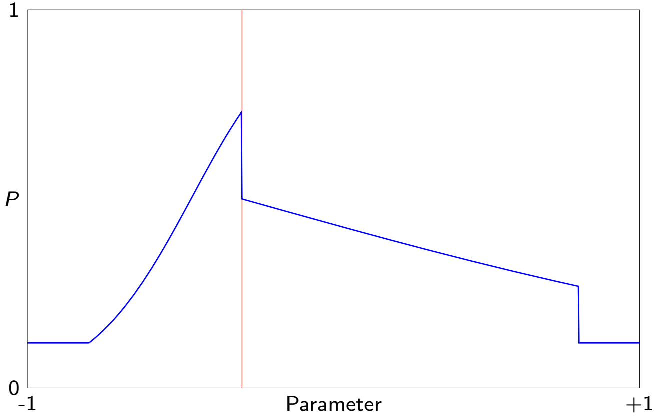
Angle



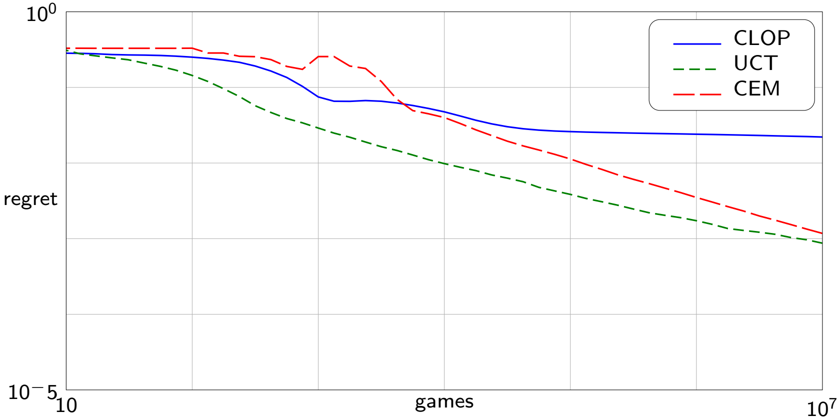
Angle



Discontinuous Function



Discontinuous Function



Conclusion

Summary of CLOP

- Much faster black-box optimizer than state of the art in games
- Foolproof: no tricky meta-parameters
- Popular freeware: `http://remi.coulom.free.fr/CLOP/`

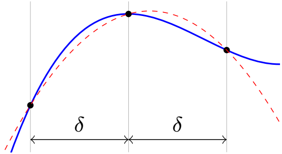
Future Work

- High-dimensional problems: more regularization, sparsity
- Apply to less noisy or noiseless problems (BBOB)
- Apply CLOP principle to other forms of regression
- Optimization from self-play
- Prove convergence

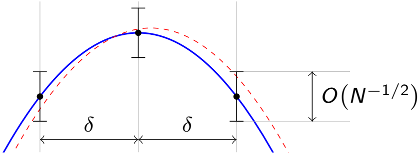
Extra Slide: Code

```
procedure QuadraticCLOP(H, x⃗_1, y_1, ..., x⃗_N, y_N)
    w_0 ← λx⃗.1                                         ▷ a function of x⃗ that returns 1
    W_0 ← N
    k ← 0

    repeat
        w ← λx⃗. min_{i=0}^{k} w_i(x⃗)                   ▷ weight function
        k ← k + 1
        q_k ← QuadraticLogisticRegression(w, x⃗_1, y_1, ..., x⃗_N, y_N)
        μ_k ← LogisticMean(w, x⃗_1, y_1, ..., x⃗_N, y_N)
        σ_k ← ConfidenceDeviation(w, x⃗_1, y_1, ..., x⃗_N, y_N)
        w_k ← λx⃗.e^{(q_k(x⃗) − μ_k)/(Hσ_k)}
        W_k ← Σ_{i=1}^{N} min(w(x⃗_i), w_k(x⃗_i))
    until W_k > 0.99 × W_{k−1}

    x⃗_{N+1} ← Random(w)                                ▷ next sample, distributed like w
    x̃⃗ ← Σ_{i=1}^{N+1} w(x⃗_i)x⃗_i / Σ_{i=1}^{N+1} w(x⃗_i)  ▷ estimated optimal
end procedure
```

Asymptotic Rate of Convergence (Intuitively)



Bias only. Regret $= O(\delta^4)$.

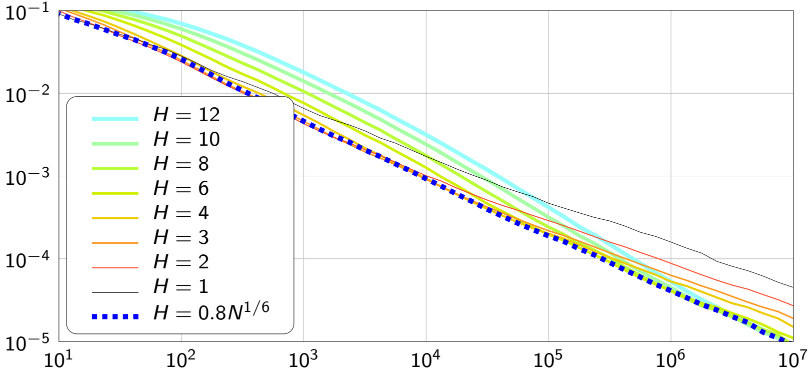


Variance only. Regret $= O(N^{-1}\delta^{-2})$.

Optimal asymptotic bias-variance tradeoff: regret $= O(N^{-2/3})$.

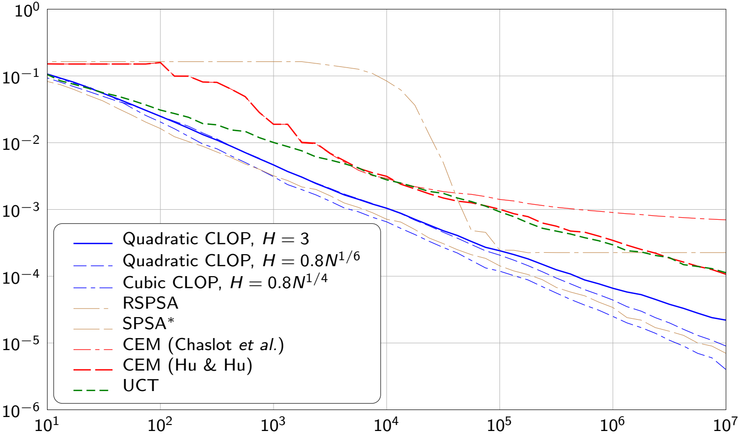
Effect of Meta-Parameter H

TODO: name of axes.

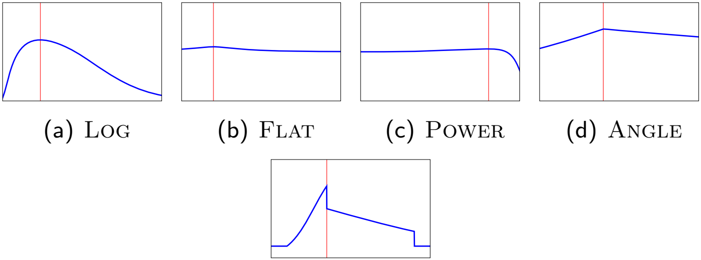


Conclusion (of many other experiments): $H = 3$ works well in practice.

Extra Slide: Many algorithms

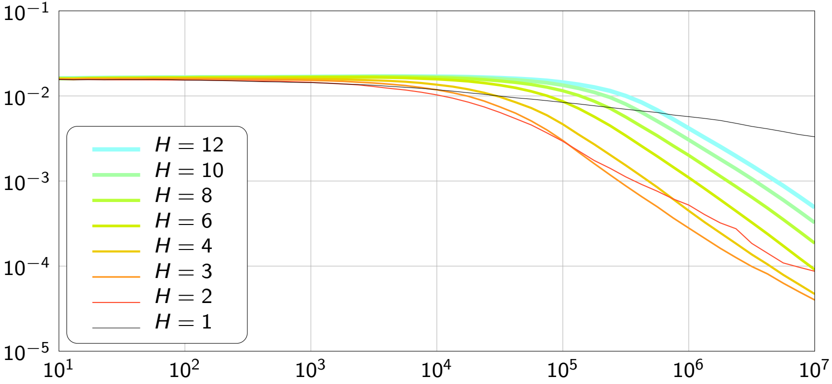


Extra Slide: 1D Problems

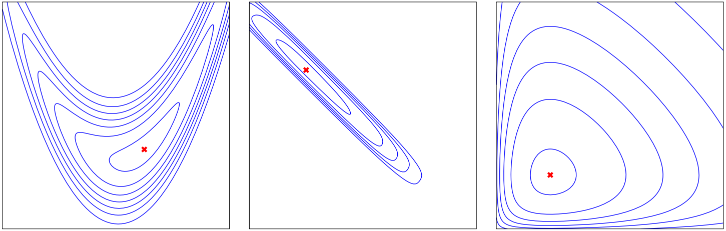


(a) LOG (b) FLAT (c) POWER (d) ANGLE (e) STEP

Effect of Meta-Parameter H on POWER

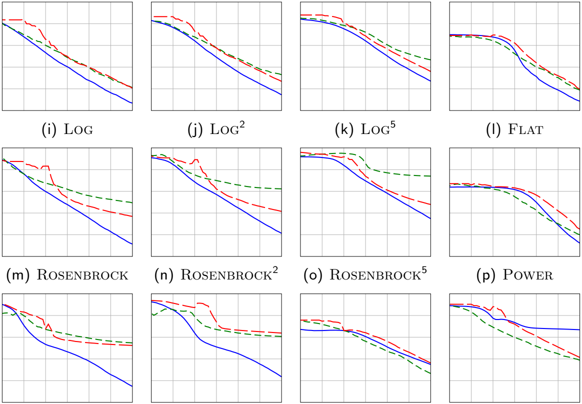


Extra Slide: 2D Problems



(f) ROSENBROCK (g) CORRELATED (h) LOG^2

Extra Slide: Performance on many problems



(i) Log (j) Log^2 (k) Log^5 (l) Flat

(m) Rosenbrock (n) Rosenbrock^2 (o) Rosenbrock^5 (p) Power

(q) Correlated (r) Correlated^2 (s) Angle (t) Step

Quadratic CLOP ($H = 3$), UCT, CEM (Hu & Hu).