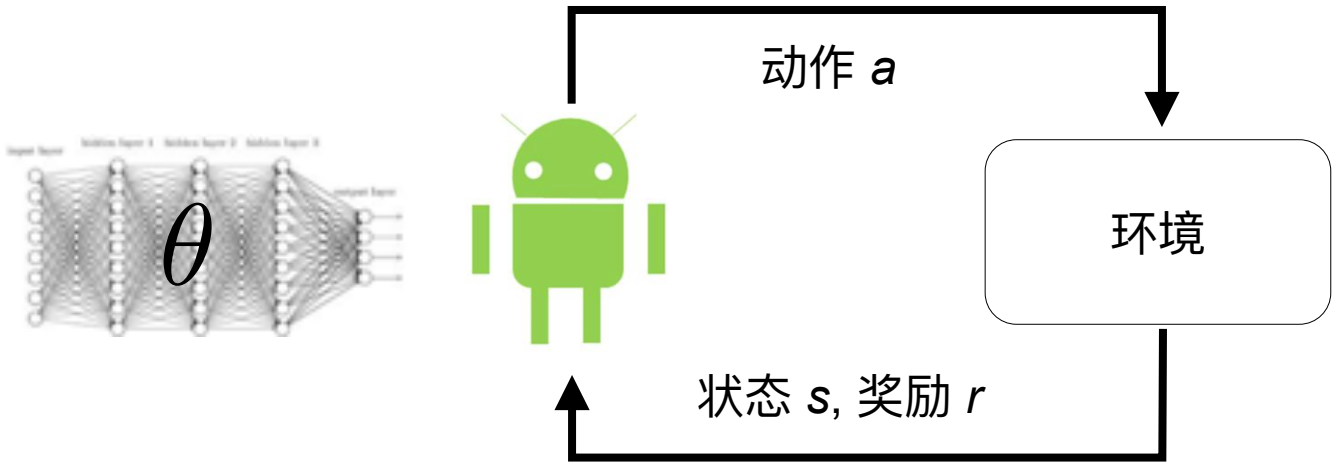
NJUAI 南京大学 人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

Lecture 3
RL by Black-box Optimization

Reinforcement learning



Agent's goal: learn a policy to maximize the return

T-step: $\sum_{t=0}^{T} r_t$ average: $\frac{1}{T}\sum_{t=0}^{T} r_t$ discounted: $\sum_{t=0}^{\infty} \gamma^t r_t$

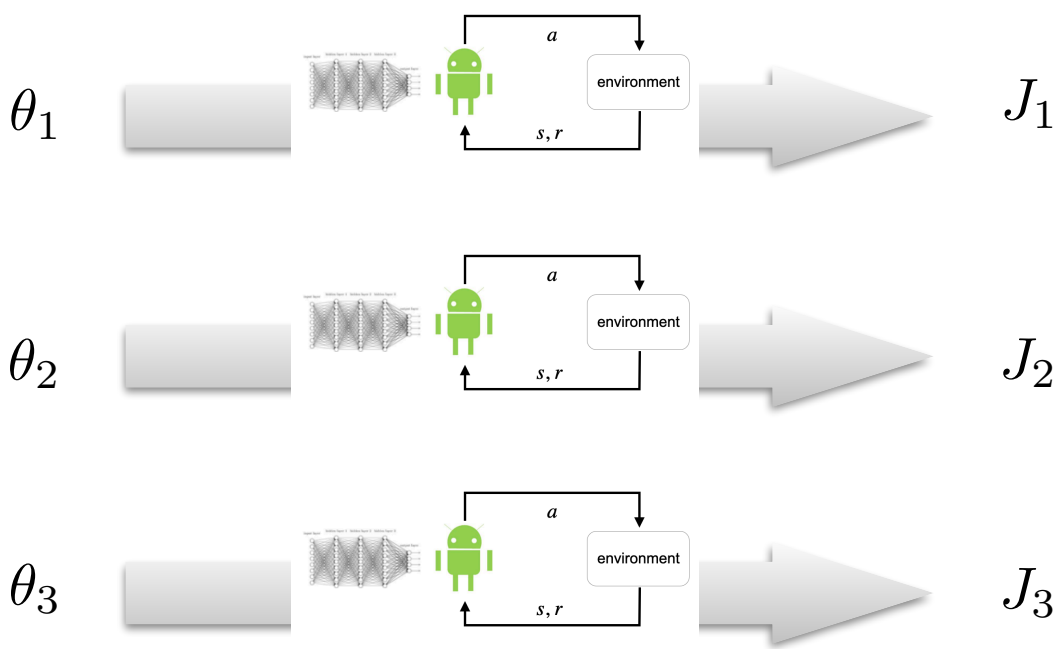
How to find the θ that maximizes the return?

$$\theta^* = \arg\max_{\theta} J(\pi_\theta)$$

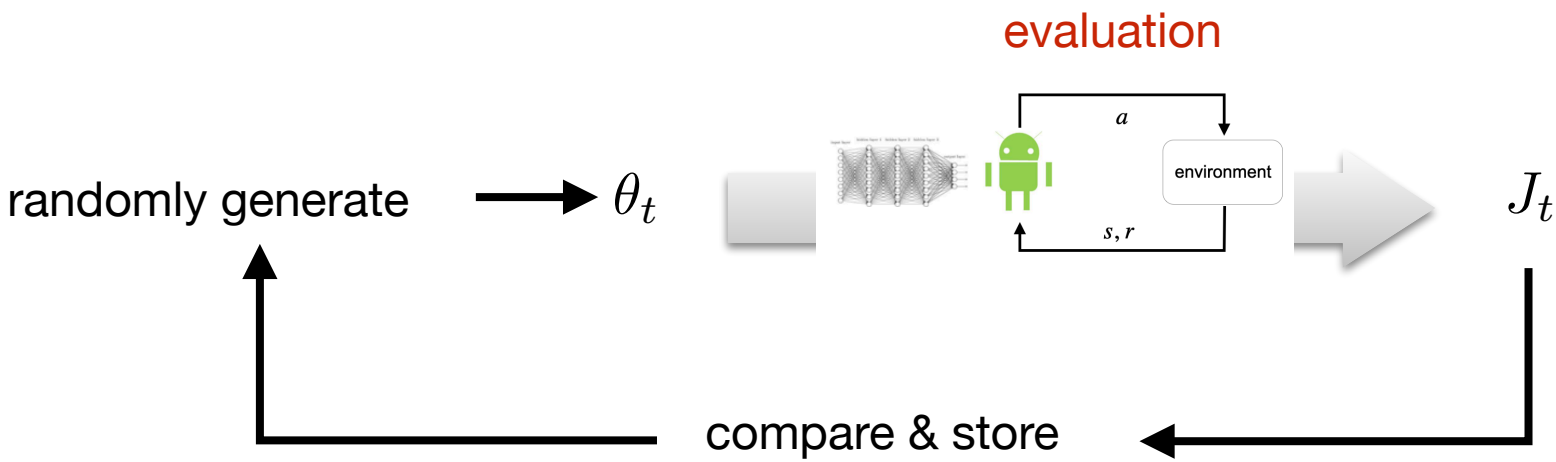
Solve theta

$$\theta^* = \arg\max_{\theta} J(\pi_\theta)$$

Can we solve θ without any information about J?



Random search

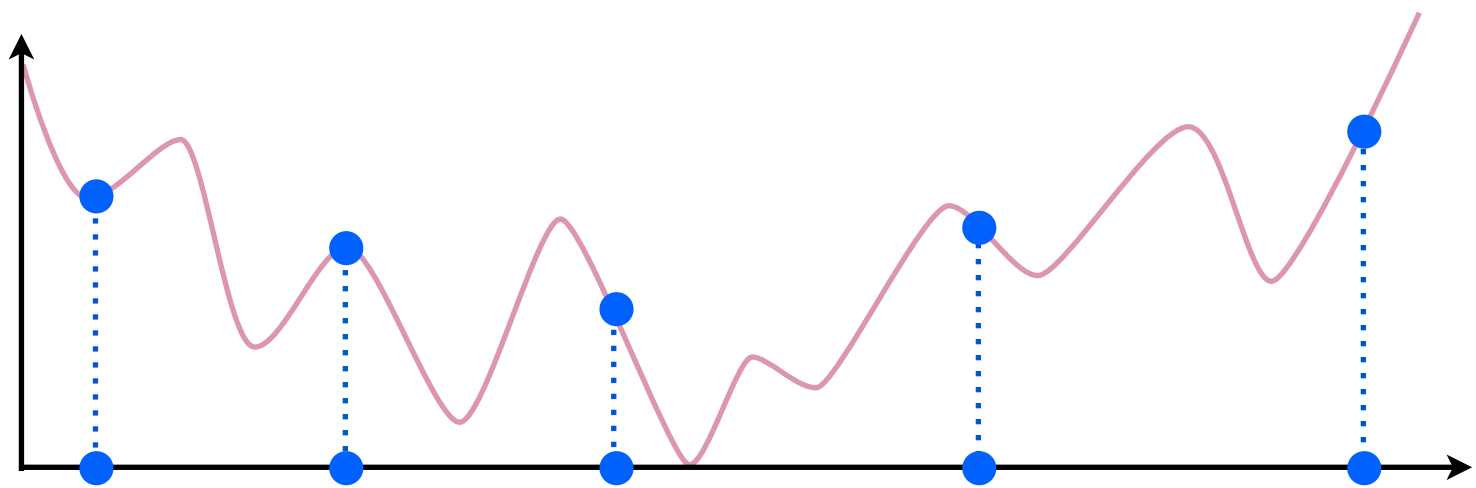


- can work !
- usually inefficient, particularly for high-dim parameters

Can be better?

general principle:

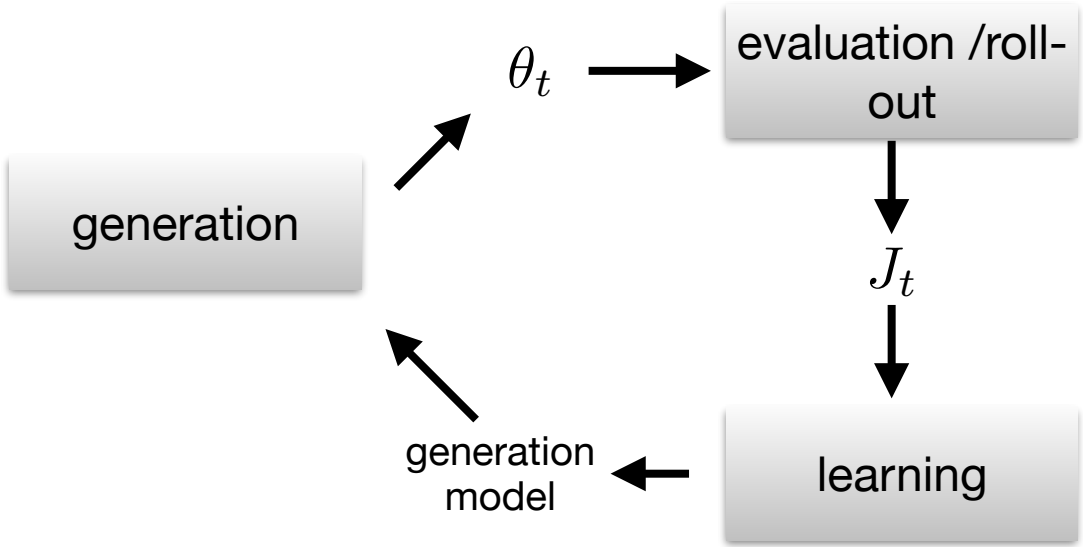
understand the problem from the samples, and generate better samples



known as

- black-box optimization
- derivative-free optimization
- zeroth order optimization

Black-box optimization



- what is the generation model?
- how to learn?

Evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

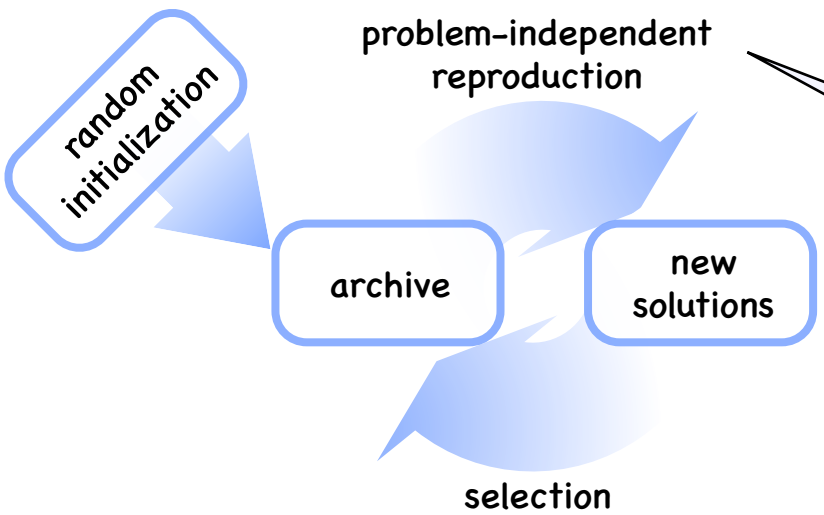
Evolutionary Strategies

[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...



for binary vector:

mutation: [1,0,0,1,0] → [1,1,0,1,0]

crossover: [1,0,0,1,0] + [0,1,1,1,0]

→ [0,1,0,1,0] + [1,0,1,1,0]

for real vector:

mutation: $x = x + \delta, \delta \sim \mathcal{N}(0, 1)$

Example: evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

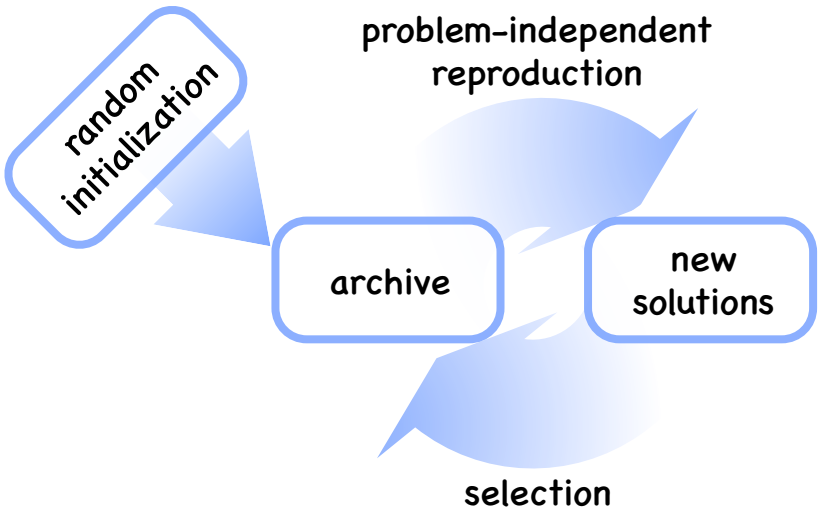
Evolutionary Strategies

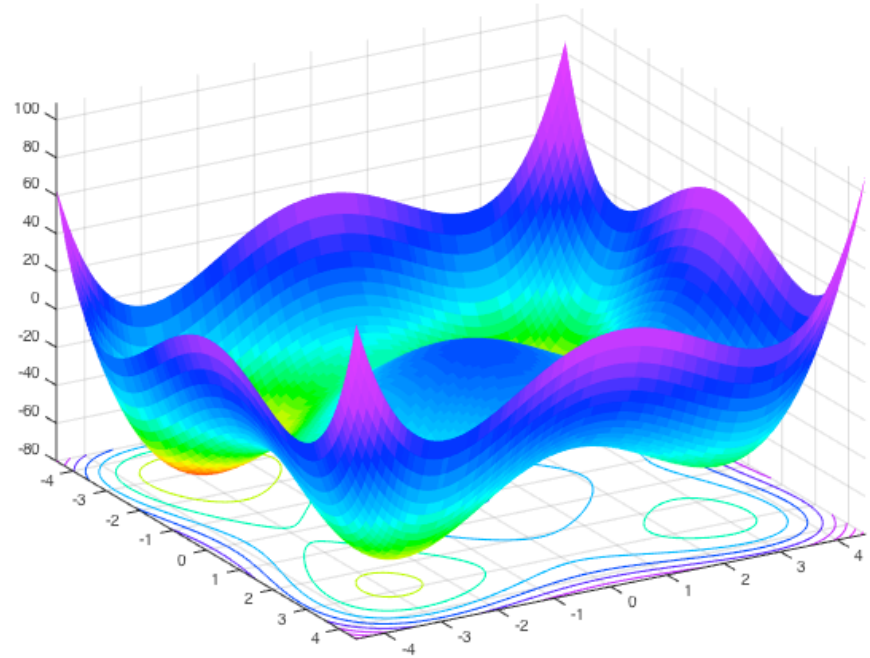
[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





Example: evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

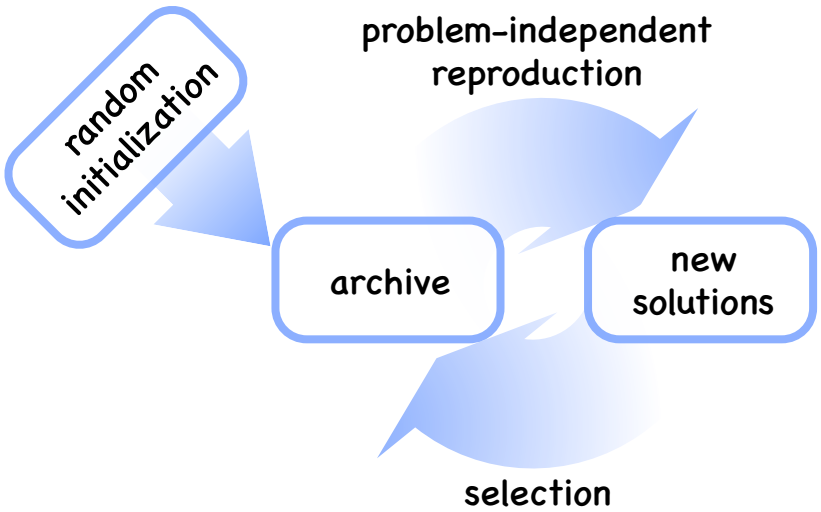
Evolutionary Strategies

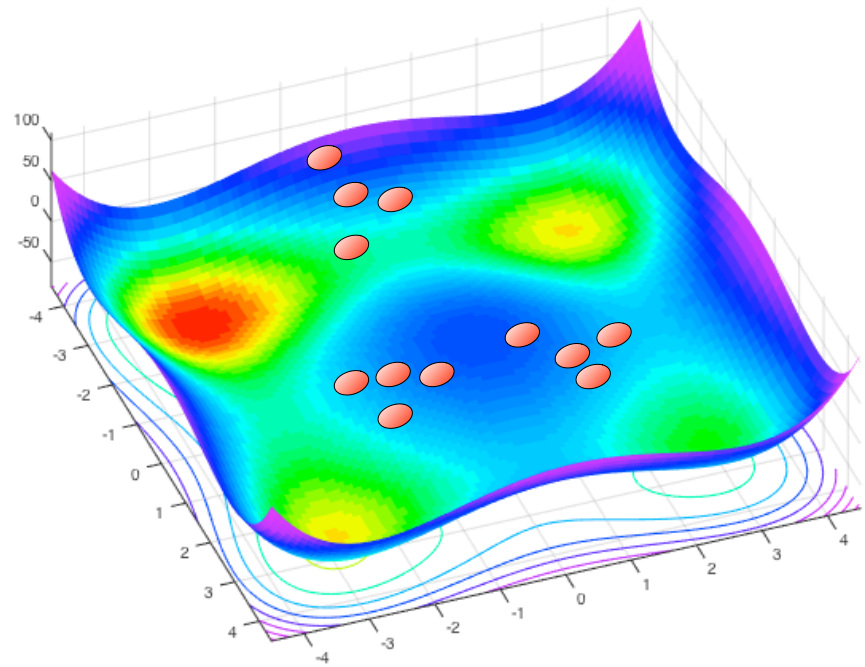
[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





Example: evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

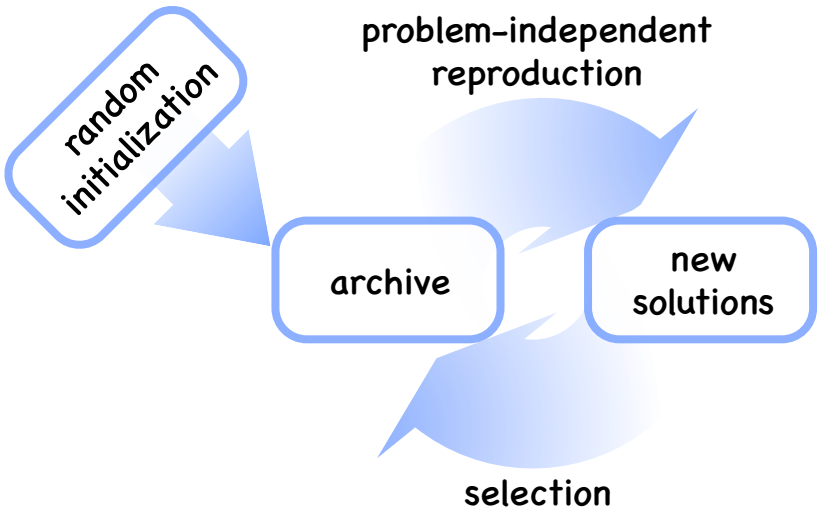
Evolutionary Strategies

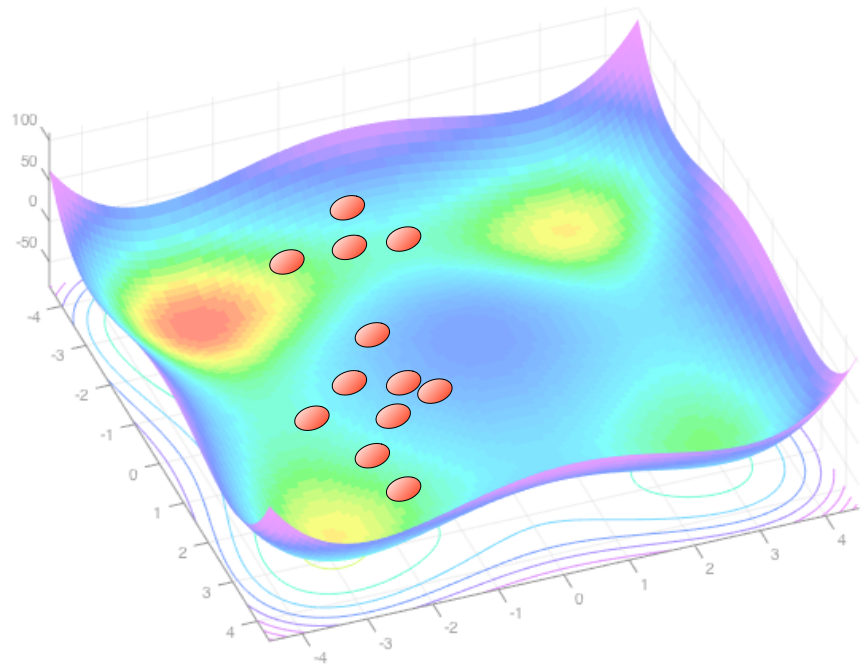
[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





Example: evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

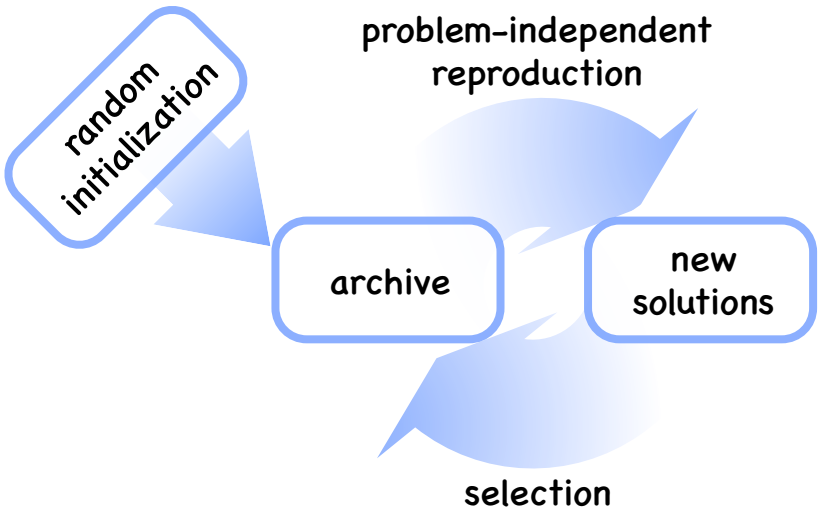
Evolutionary Strategies

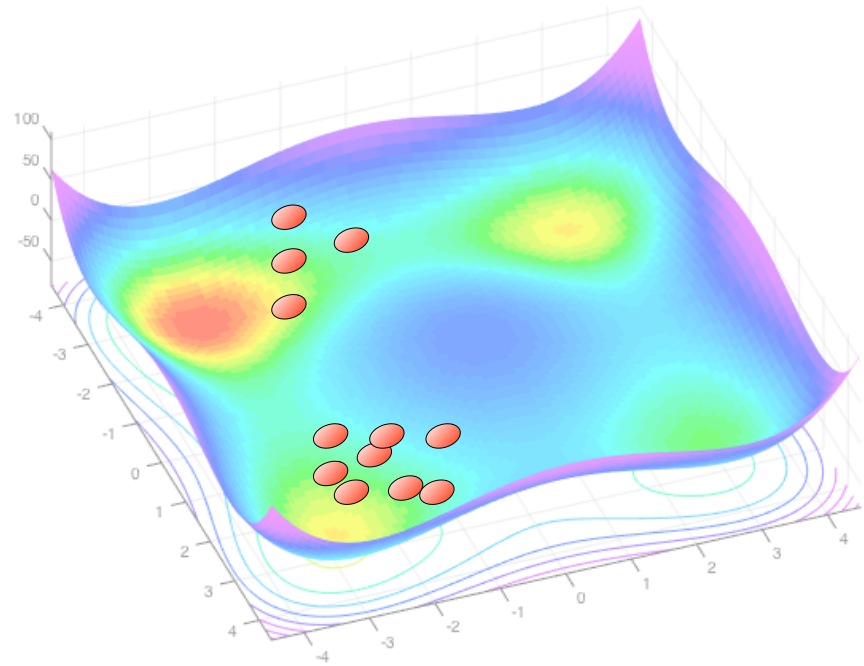
[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





Example: evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

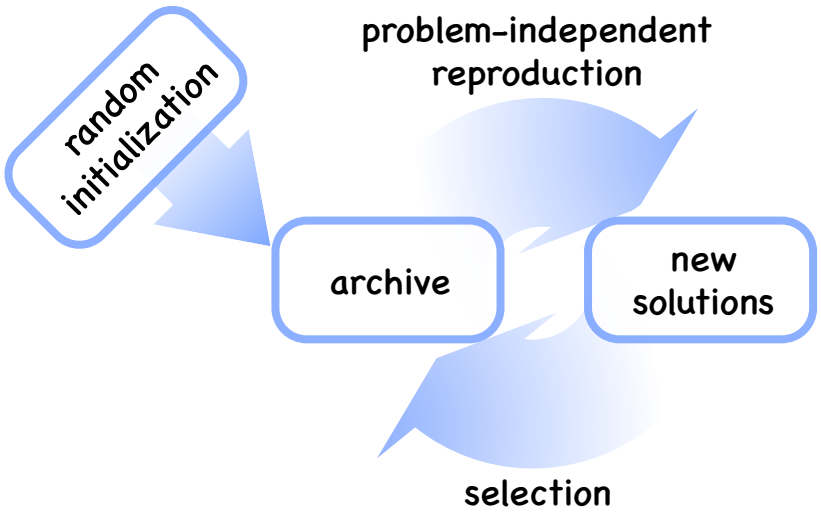
Evolutionary Strategies

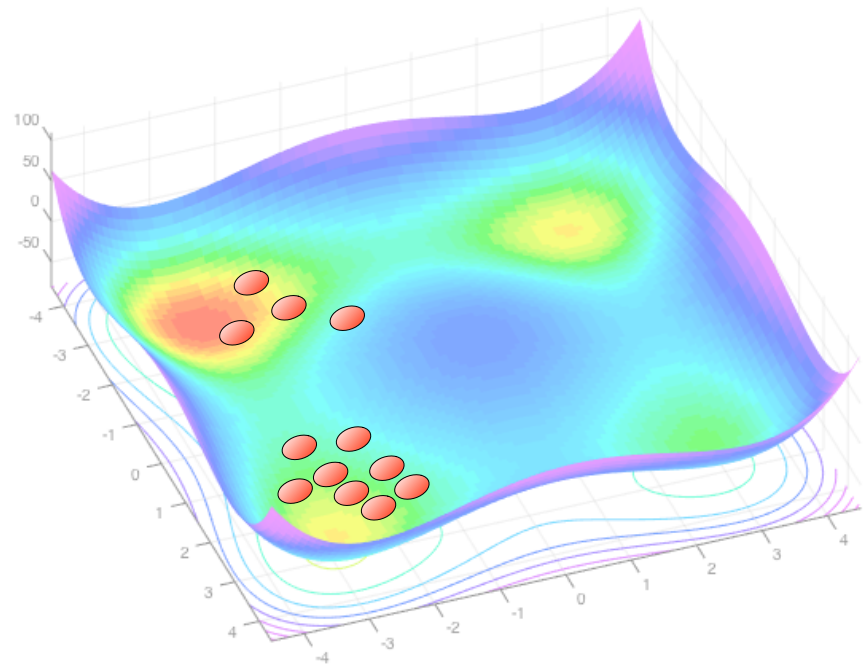
[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





Example: evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

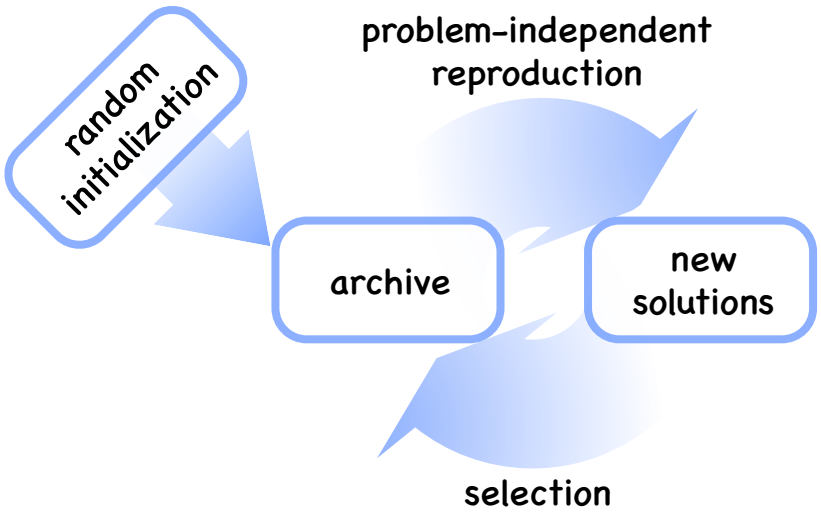
Evolutionary Strategies

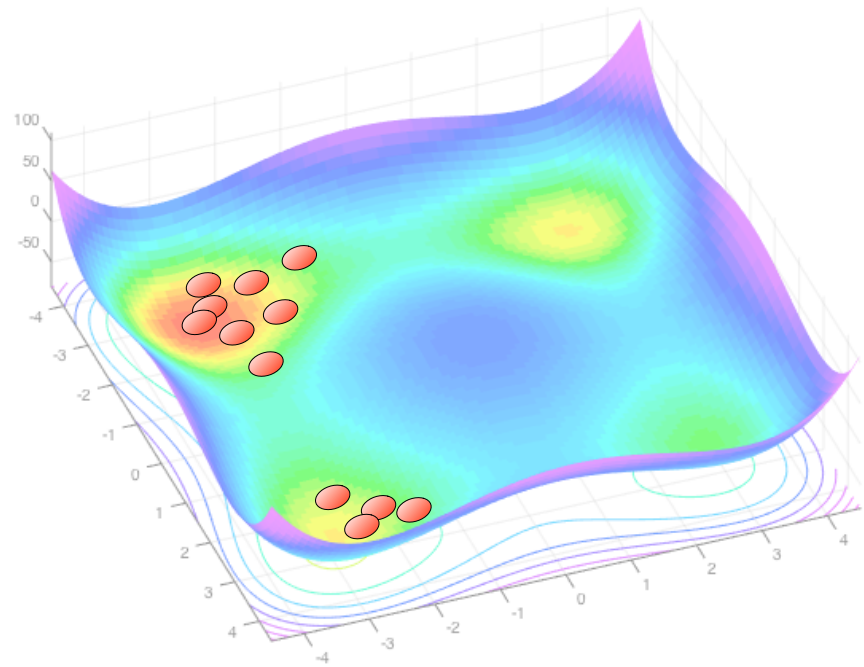
[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





Example: evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

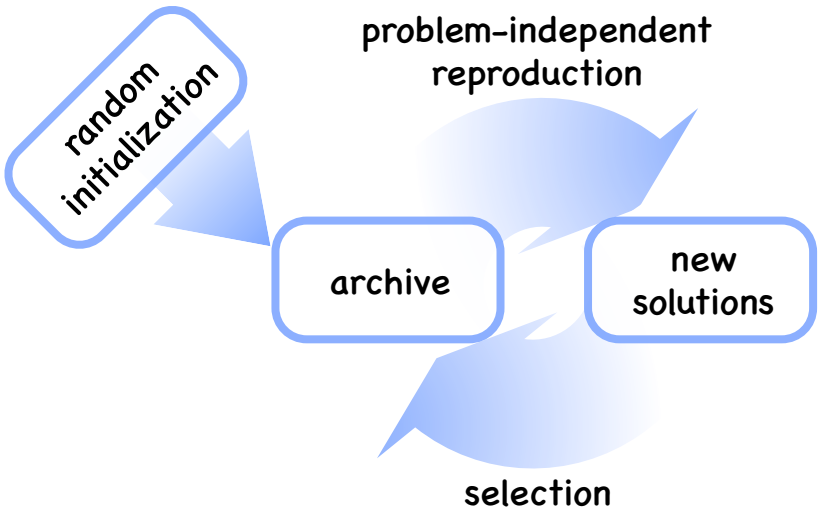
Evolutionary Strategies

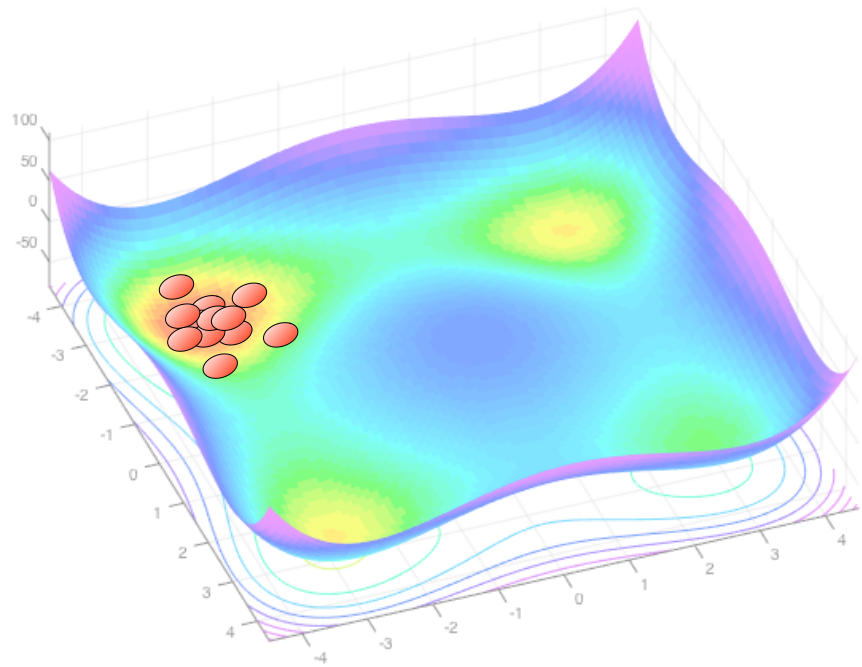
[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





Example: evolutionary algorithms

Genetic Algorithms

[J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

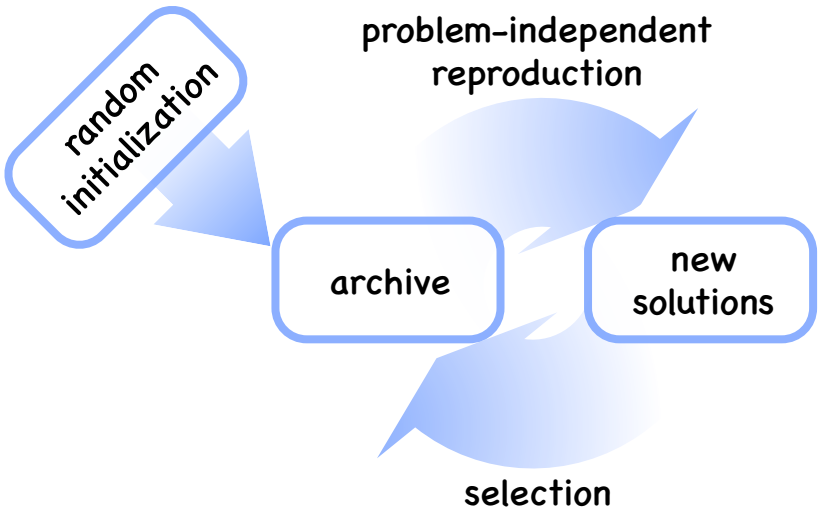
Evolutionary Strategies

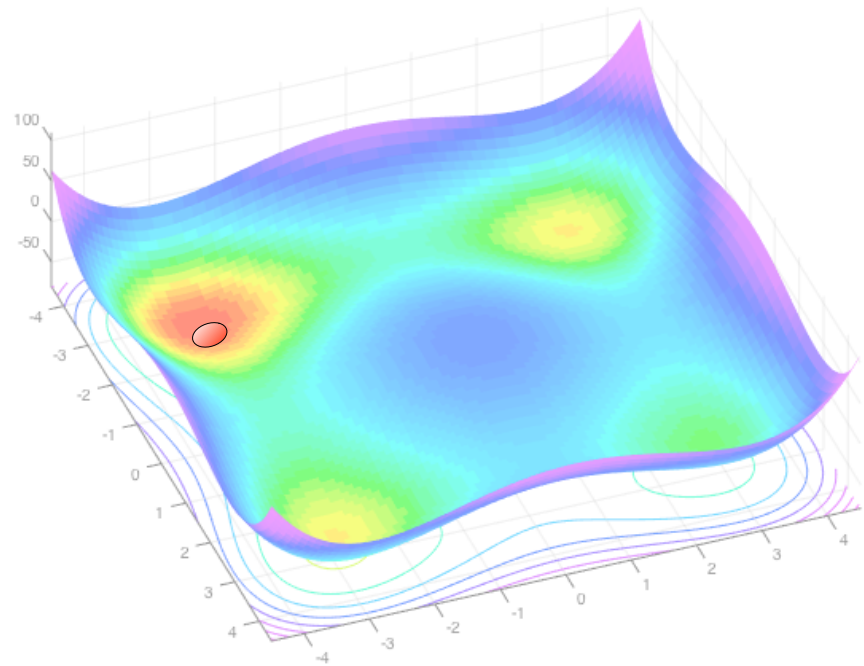
[I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming

[L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...



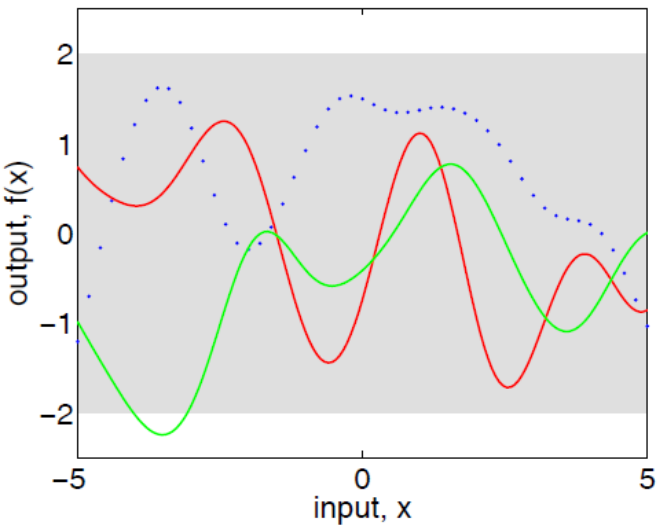


only need to evaluate solutions ⇒ calculate f(x) !

Bayesian optimization

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

A GP is a distribution over functions, completely specified by its mean function and covariance function



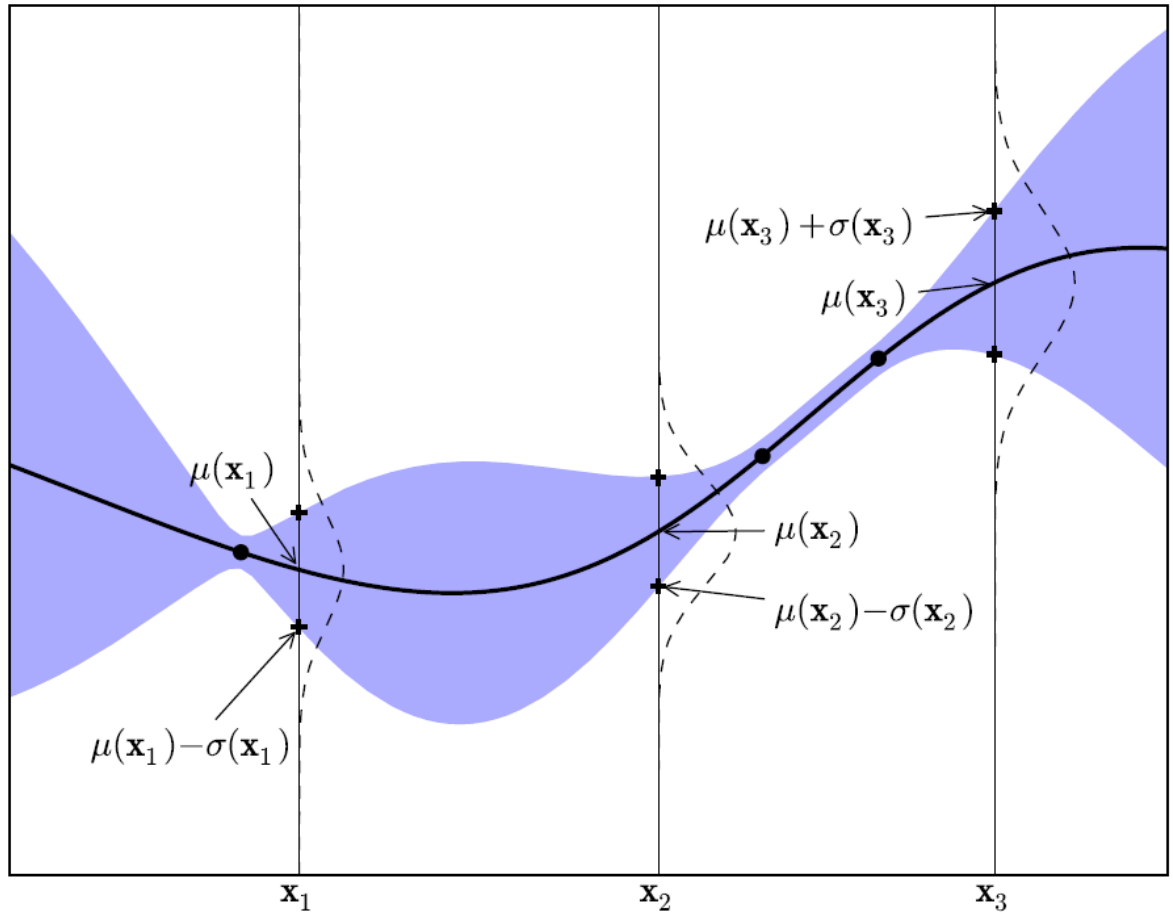
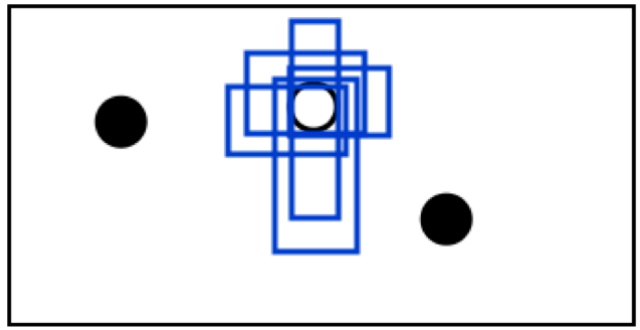


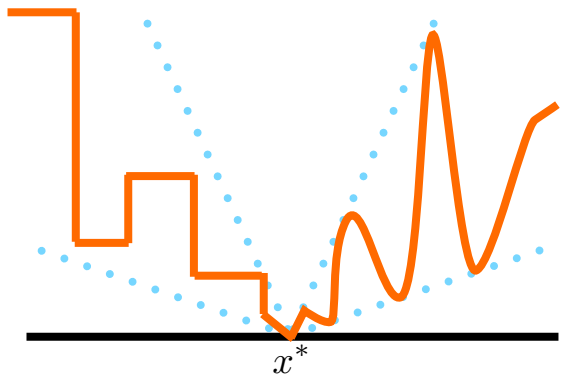
Figure 2: *Simple 1D Gaussian process with three observations. The solid black line is the GP surrogate mean prediction of the objective function given the data, and the shaded area shows the mean plus and minus the variance. The superimposed Gaussians correspond to the GP mean and standard deviation ($\mu(\cdot)$ and $\sigma(\cdot)$) of prediction at the points, $\mathbf{x}_{1:3}$.*

Classification-based optimization

classification-based optimization samples from randomized box classifiers, which classifies good from bad solutions



it is polynomial for approximating local-Holder functions



https://github.com/polixir/ZOOpt

Examples

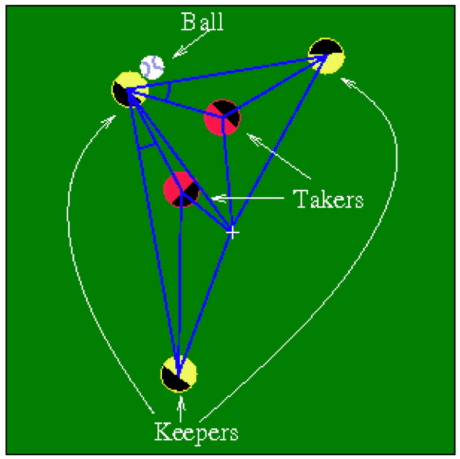
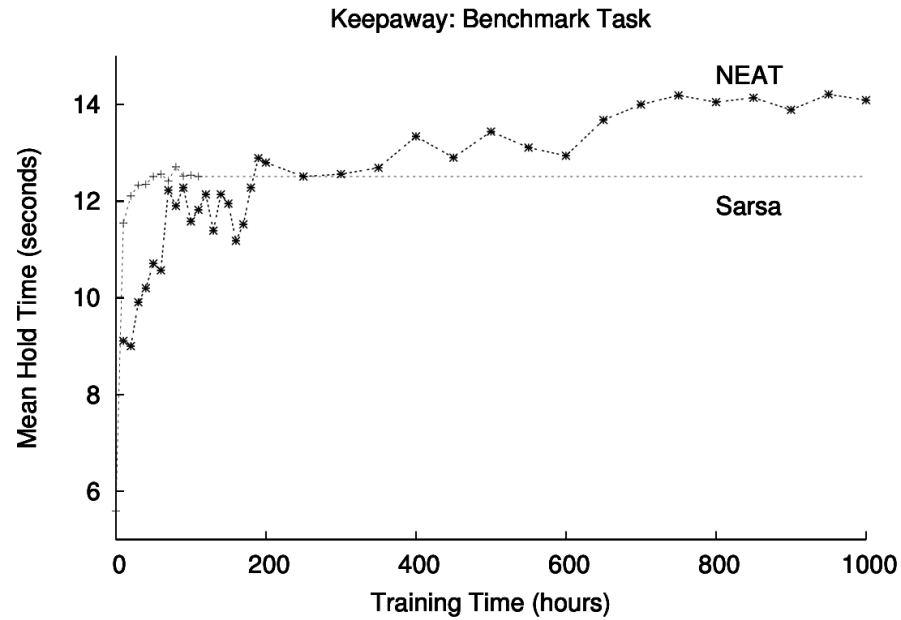
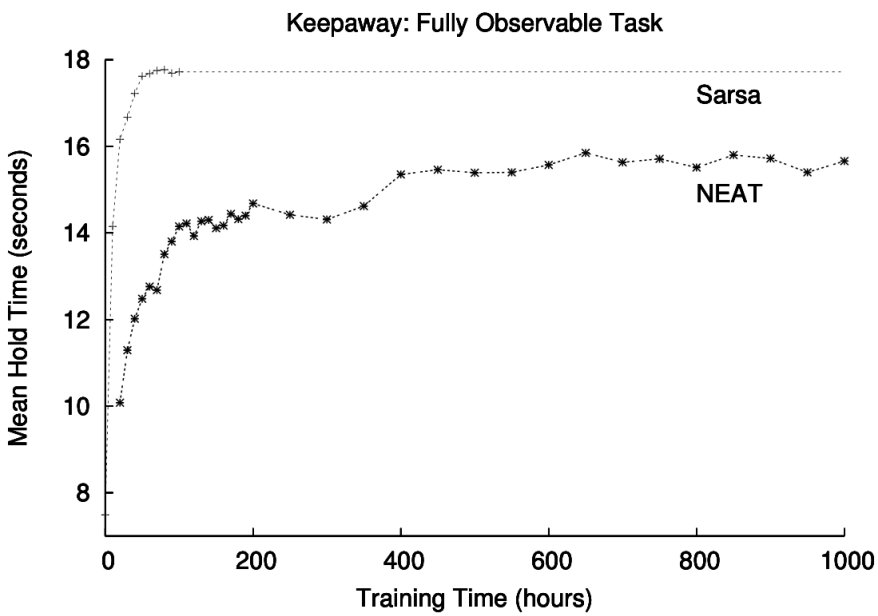
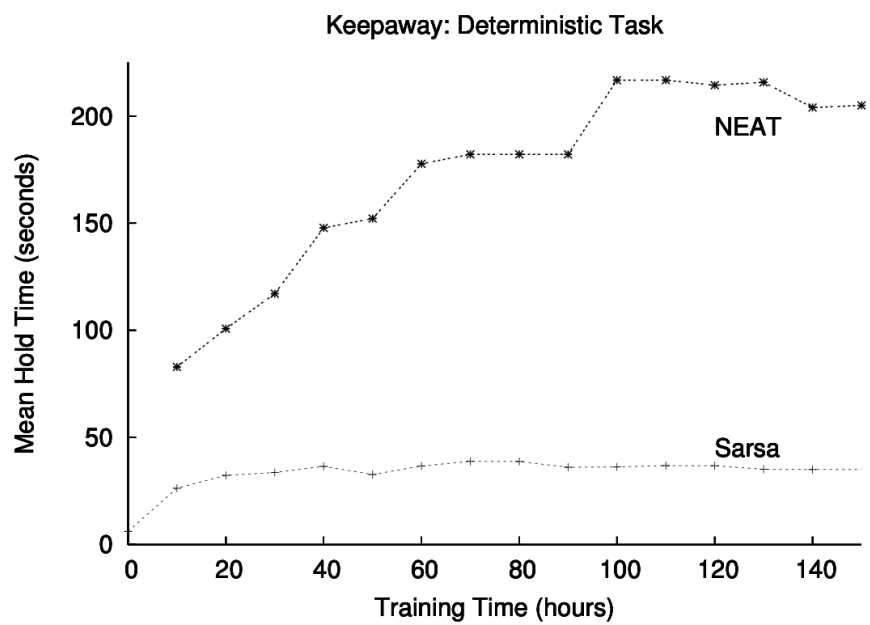


Figure 2: 13 state variables are used for learning with 3 keepers and 2 takers. The state is ego-centric and rotationally invariant for the keeper with the ball; there are 11 distances, indicated with blue lines, between players and the center of the field as well as 2 angles along passing lanes.

[Matthew E. Taylor, Shimon Whiteson, Peter Stone. Comparing evolutionary and temporal difference methods in a reinforcement learning domain. In: GECCO 2006]







For Parameter Updating

- value function based (to name a few)

  Learning Tetris using the noisy cross-entropy method (Neural Computation 2006)

- policy search (to name a few)

  Using trajectory data to improve Bayesian optimization for reinforcement learning (JMLR'14)

  Sequential classification-based optimization for direct policy search (AAAI'17)

  Back to basics: Benchmarking canonical evolution strategies for playing atari (IJCAI'18)

  Policy optimization by genetic distillation (ICLR'18)

...

For Model Selection

Designing application-specific neural networks using the structured genetic algorithm (ICCGANN'92)

Evolving neural networks through augmenting topologies (ECJ'02)

Evolutionary Function Approximation for Reinforcement Learning (JMLR'06)

Gradient-free policy architecture search and adaptation (CoRL'17)

…

Using gradients anyway: Finite difference

Gradient is often used for optimization, but no gradient!

what is gradient: $$\frac{\partial f}{\partial x} = \lim_{\delta \to 0} \frac{f(x+\delta) - f(x)}{\delta}$$

No gradient => approximate

$$\frac{\partial f}{\partial x} \approx \frac{f(x+\delta) - f(x)}{\delta}$$ use a small delta δ

more stable: $$\frac{\partial f}{\partial x} \approx \frac{1}{N} \sum_i \frac{f(x+\delta_i) - f(x)}{\delta_i}$$

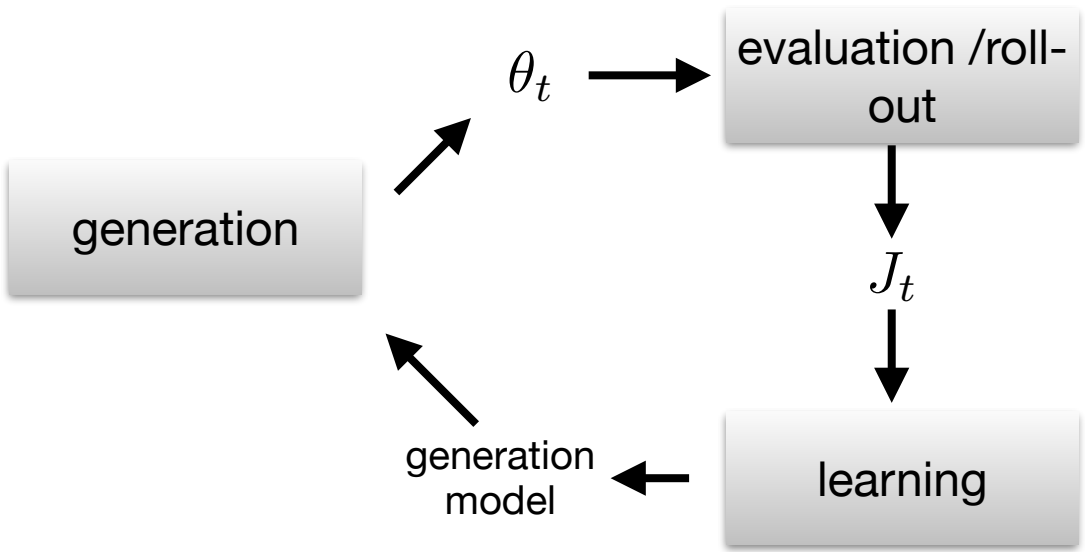
1. sample small random numbers
2. calculate average differences

Using gradients anyway: Differentiable modeling

$$\theta^* = \arg\max_{\theta} J(\pi_\theta)$$

We still have no knowledge about J.

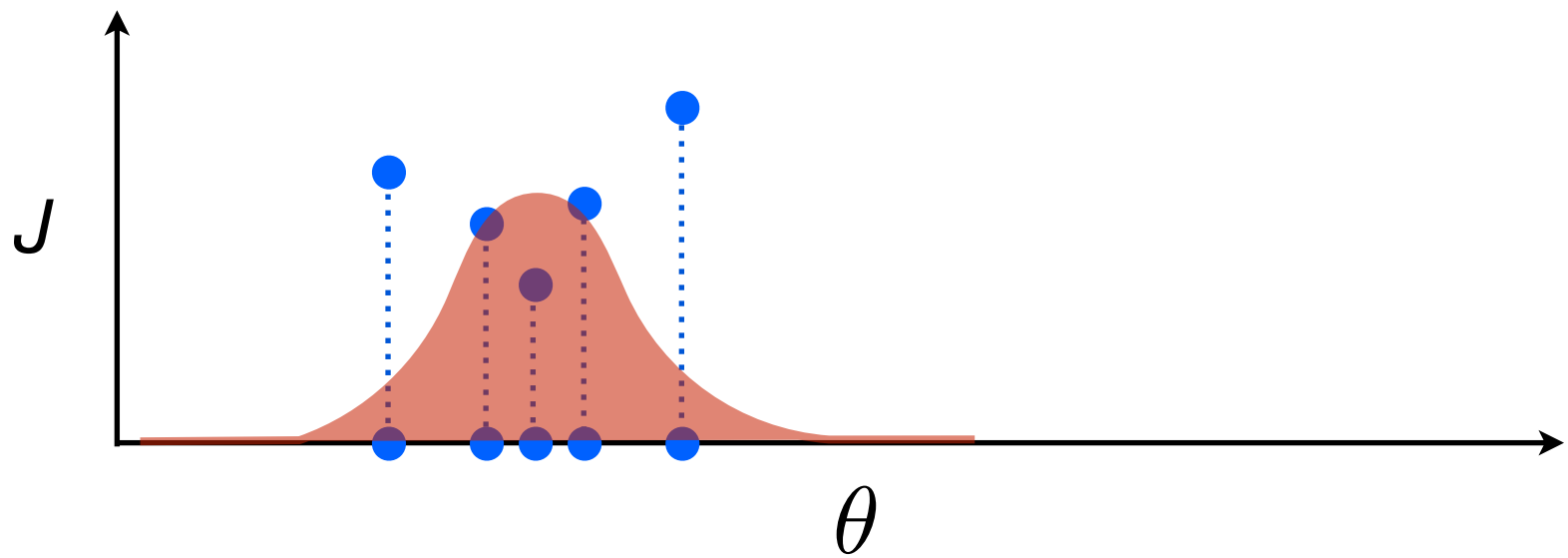
Use a differentiable **generation/sampling model !**



e.g. Gaussian model

$$p(\theta; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\theta-\mu)^2}{2\sigma^2}\right)$$

$$\partial_\mu p(\theta; \mu, \sigma^2) = \frac{(\theta-\mu)}{\sigma^2} p(\theta; \mu, \sigma^2) \qquad \text{noted} \quad \partial f = f \partial \log f$$



Q: which Gaussian model has the best samples?

Shifted objective

find a theta that maximizes *J* value

$$\theta^* = \arg\max_{\theta} J(\pi_\theta)$$

=>

find a Gaussian distribution with maximized expected *J* value

$$\mu, \sigma = \arg\max_{\mu,\sigma} E_{\theta\sim\mathcal{N}(\mu,\sigma)} J(\pi_\theta) = \arg\max_{\mu,\sigma} \int p(\theta; \mu, \sigma^2) J(\pi_\theta)\ \mathrm{d}\theta$$

Shifted objective

$$\mu, \sigma = \arg\max_{\mu,\sigma} E_{\theta\sim\mathcal{N}(\mu,\sigma)} J(\pi_\theta) = \arg\max_{\mu,\sigma} \int p(\theta; \mu, \sigma^2) J(\pi_\theta)\ \mathrm{d}\theta$$

optimization by gradient:

$$\mu = \mu + \int p(\theta; \mu, \sigma^2) \frac{\theta - \mu}{\sigma^2} J(\pi_\theta)\ \mathrm{d}\theta$$

sampling

$$\mu+ = \frac{1}{n} \sum_i \frac{\theta_i - \mu}{\sigma^2} J(\pi_{\theta_i})$$

Shifted objective

reparameterization

optimization by gradient:

$$\mu = \mu + \int p(\theta; 0, 1) \frac{\theta}{\sigma} J(\pi_{\theta * \sigma + \mu}) \, \mathrm{d}\theta$$

sampling theta from standard normal distribution

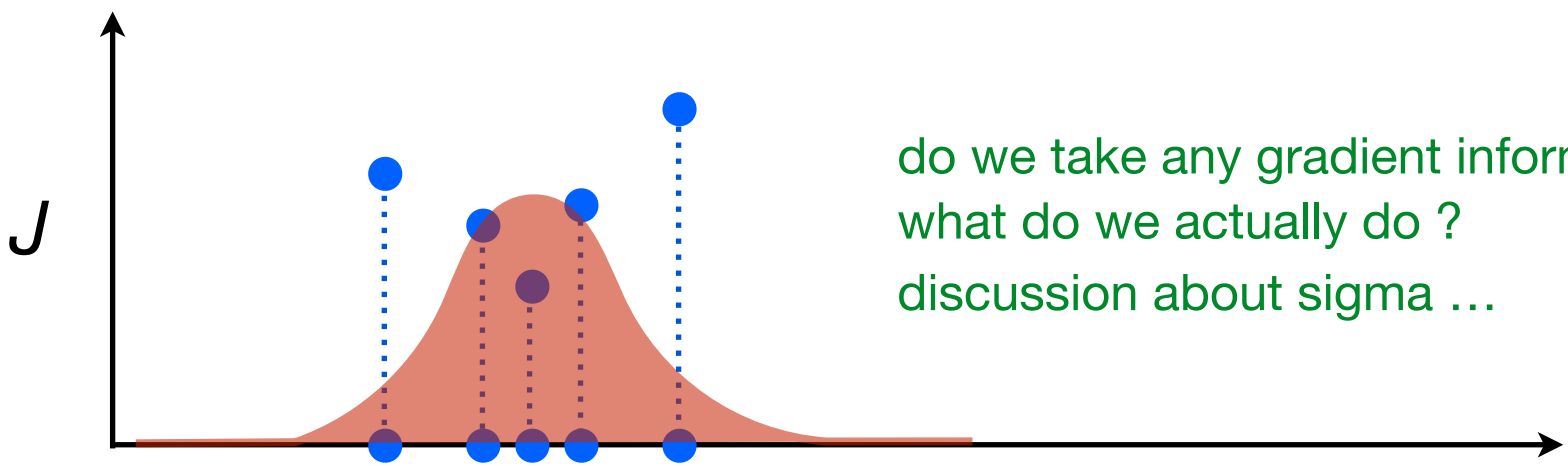
$$\mu += \frac{1}{n} \sum_i \frac{\theta_i}{\sigma} J(\pi_{\theta_i * \sigma + \mu})$$

Using gradients anyway

$$\theta^* = \arg\max_{\theta} J(\pi_\theta)$$

$$\mu, \sigma = \arg\max_{\mu,\sigma} E_{\theta\sim\mathcal{N}(\mu,\sigma)} J(\pi_\theta)$$

$$\mu+ = \frac{1}{n}\sum_i \frac{\theta_i - \mu}{\sigma^2} J(\pi_{\theta_i})$$



do we take any gradient information from *J* ?
what do we actually do ?
discussion about sigma …