Introduction to Black-Box Optimization in Continuous Search Spaces

Definitions, Examples, Difficulties

I am happy to answer questions at any time!

Problem Statement

Continuous Domain Search/Optimization

- Task: minimize an objective function (*fitness* function, *loss* function) in continuous domain

$$f : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}, \qquad \boldsymbol{x} \mapsto f(\boldsymbol{x})$$

- Black Box scenario (direct search scenario)

  - gradients are not available or not useful
  - problem domain specific knowledge is used only within the black box, e.g. within an appropriate encoding
- Search costs: number of function evaluations

Typical Applications

- model/system calibration
  - biological/chemical/physical $\Longrightarrow$ universal constants
  - production process
- optimization of control parameters
  - movements of a robot (e.g. for the RoboCup)
  - trajectory of a rocket
  - stability of a gas flame
- shape optimization
  - curve fitting
  - aero- or fluid dynamics design (airfoil, airship)

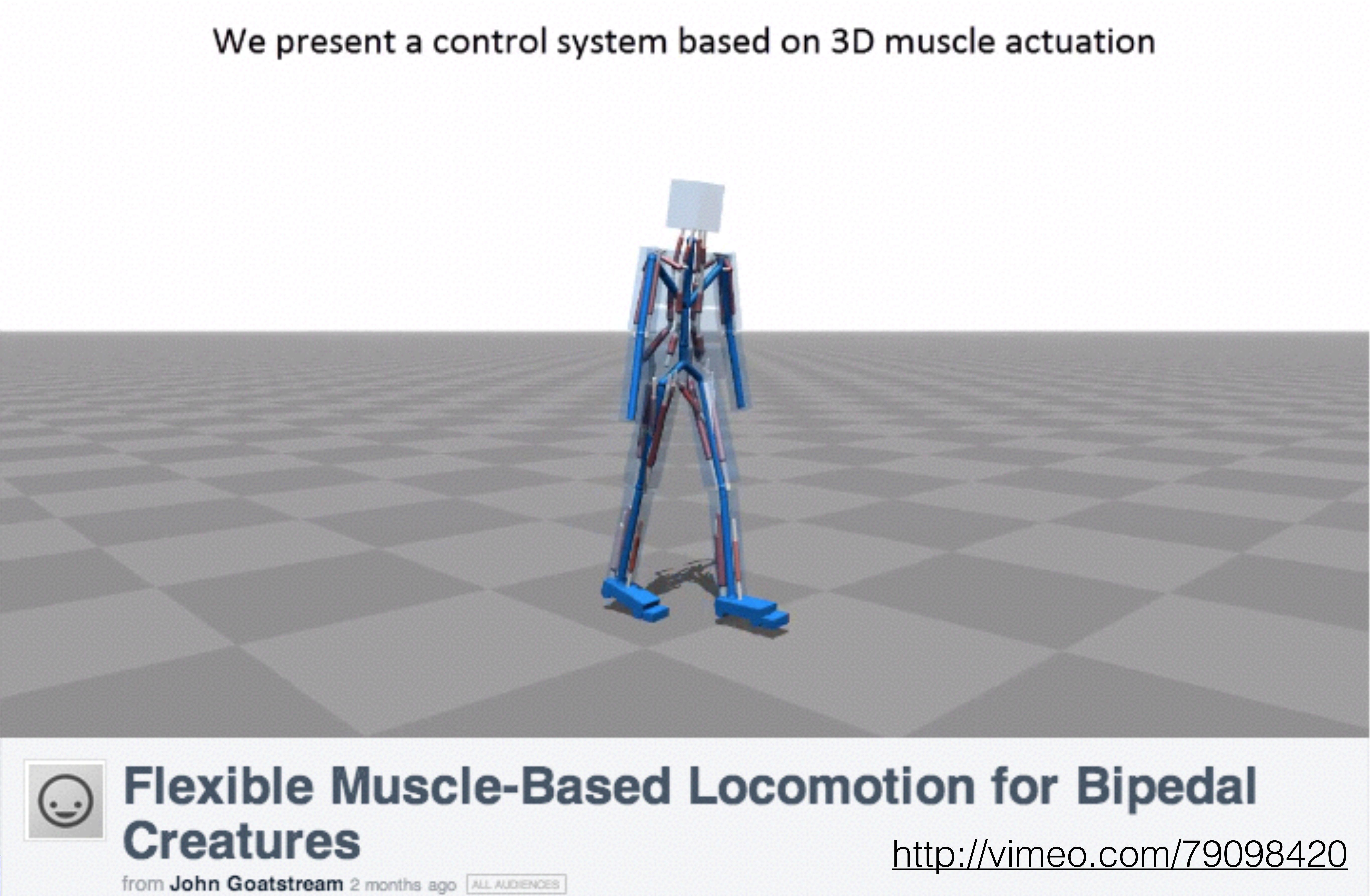
Optimization of walking gaits

http://www.icos.ethz.ch/cse/research/highlights/research_highlights_august_2004
[Dürr & Pfister 2004]

CMA-ES, Covariance Matrix Adaptation Evolution Strategy [Hansen et al 2003]
IDEA, Iterated Density Estimation Evolutionary Algorithm [Bosman 2003]
Fminsearch, downhill simplex method [Nelder & Mead 1965]

http://www.icos.ethz.ch/cse/research/highlights/research_highlights_august_2004
[Dürr & Pfister 2004]

CMA-ES, Covariance Matrix Adaptation Evolution Strategy [Hansen et al 2003]
IDEA, Iterated Density Estimation Evolutionary Algorithm [Bosman 2003]
Fminsearch, downhill simplex method [Nelder & Mead 1965]



http://vimeo.com/79098420

Problem Statement

Continuous Domain Search/Optimization

- Goal
  - fast convergence to the global optimum

    ...or to a robust solution x
  - solution x with small function value $f(x)$ with least search cost

    there are two conflicting objectives
- Typical Examples
  - shape optimization (e.g. using CFD) — curve fitting, airfoils
  - model calibration — biological, physical
  - parameter calibration — controller, plants, images
- Problems
  - exhaustive search is infeasible
  - naive random search takes too long
  - deterministic search is not successful / takes too long

Problem Statement

Continuous Domain Search/Optimization

- Goal
  - fast convergence to the global optimum

    . . . or to a robust solution $\boldsymbol{x}$
  - solution $\boldsymbol{x}$ with small function value $f(\boldsymbol{x})$ with least search cost

    there are two conflicting objectives

- Typical Examples
  - shape optimization (e.g. using CFD) — curve fitting, airfoils
  - model calibration — biological, physical
  - parameter calibration — controller, plants, images

- Problems
  - exhaustive search is infeasible
  - naive random search takes too long
  - deterministic search is not successful / takes too long

Objective Function Properties

The objective function $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ has typically moderate dimensionality, say $n \not\ll 10$, and can be

- non-linear
- non-separable
- non-convex
- multimodal

  there are possibly many local optima
- non-smooth

  derivatives do not exist
- discontinuous, plateaus
- ill-conditioned
- noisy
- . . .

Goal : cope with any of these function properties

they are related to real-world problems

Objective Function Properties

The objective function $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ has typically moderate dimensionality, say $n \not\ll 10$, and can be

- non-linear
- non-separable
- non-convex
- multimodal

  there are possibly many local optima
- non-smooth

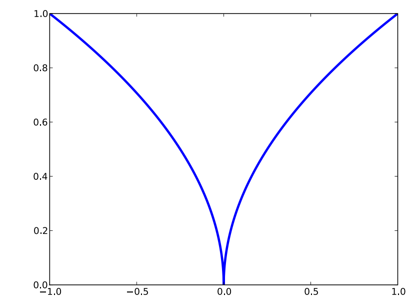
  derivatives do not exist
- discontinuous, plateaus
- ill-conditioned
- noisy
- ...

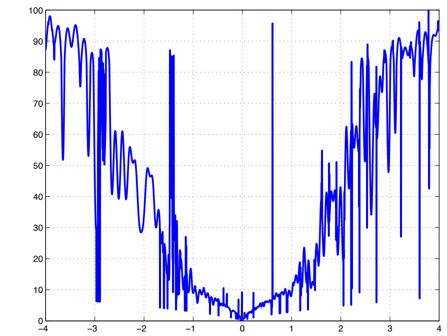
Goal : cope with any of these function properties
they are related to real-world problems

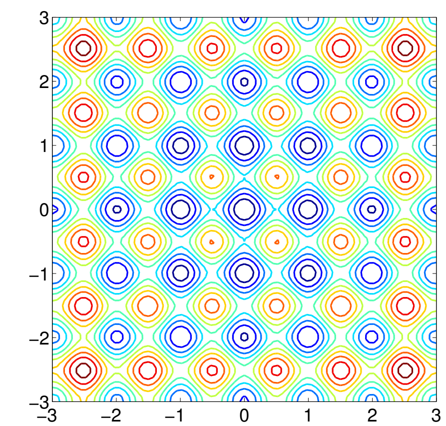
What Makes a Function Difficult to Solve?

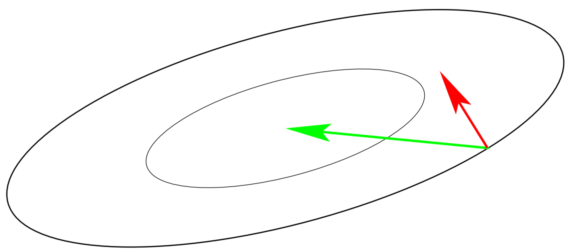
Why stochastic search?

- non-linear, non-quadratic, non-convex
  on linear and quadratic functions much better search policies are available
- ruggedness
  non-smooth, discontinuous, multimodal, and/or noisy function
- dimensionality (size of search space)
  (considerably) larger than three
- non-separability
  dependencies between the objective variables
- ill-conditioning





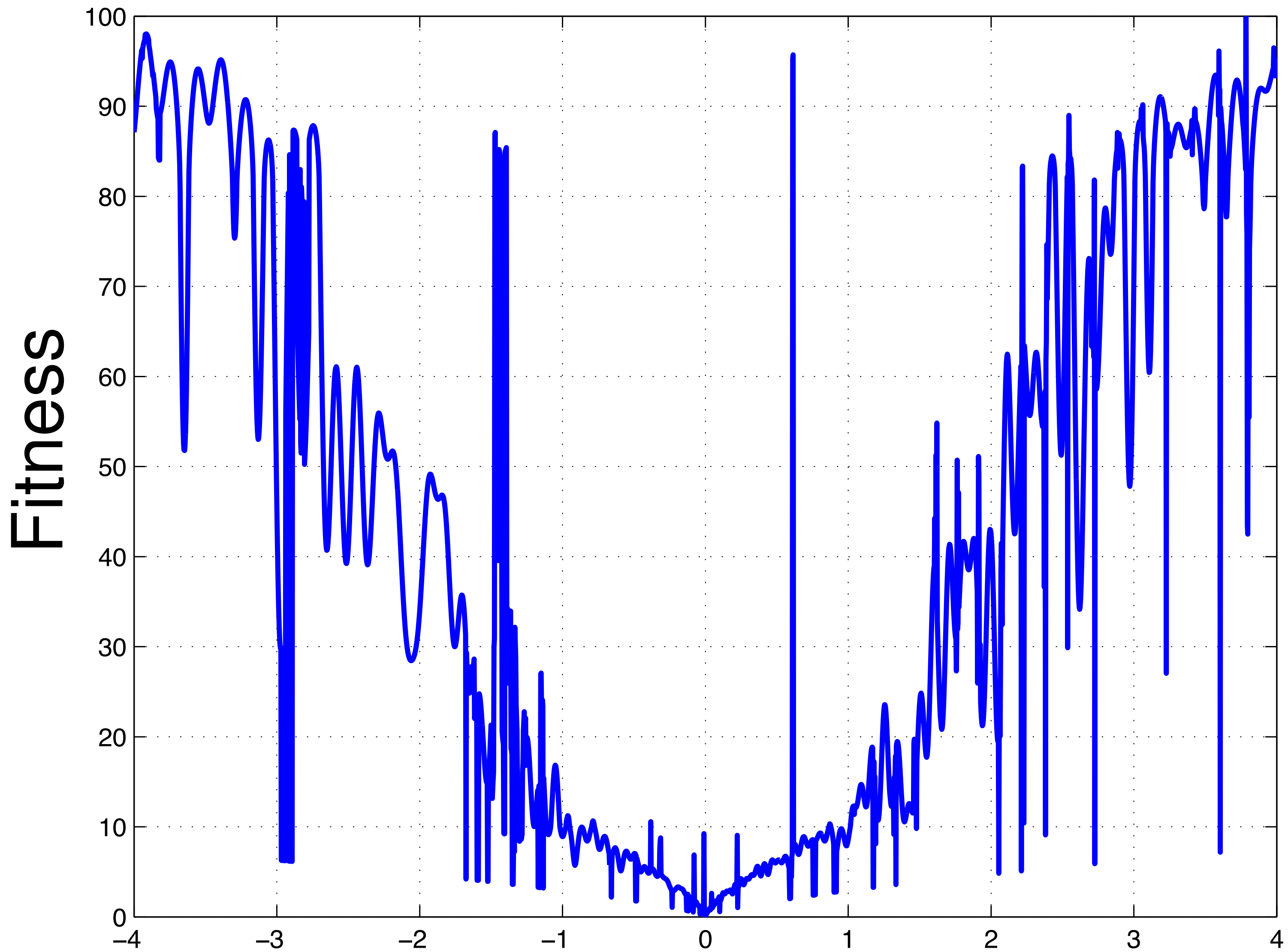




gradient direction Newton direction

Ruggedness

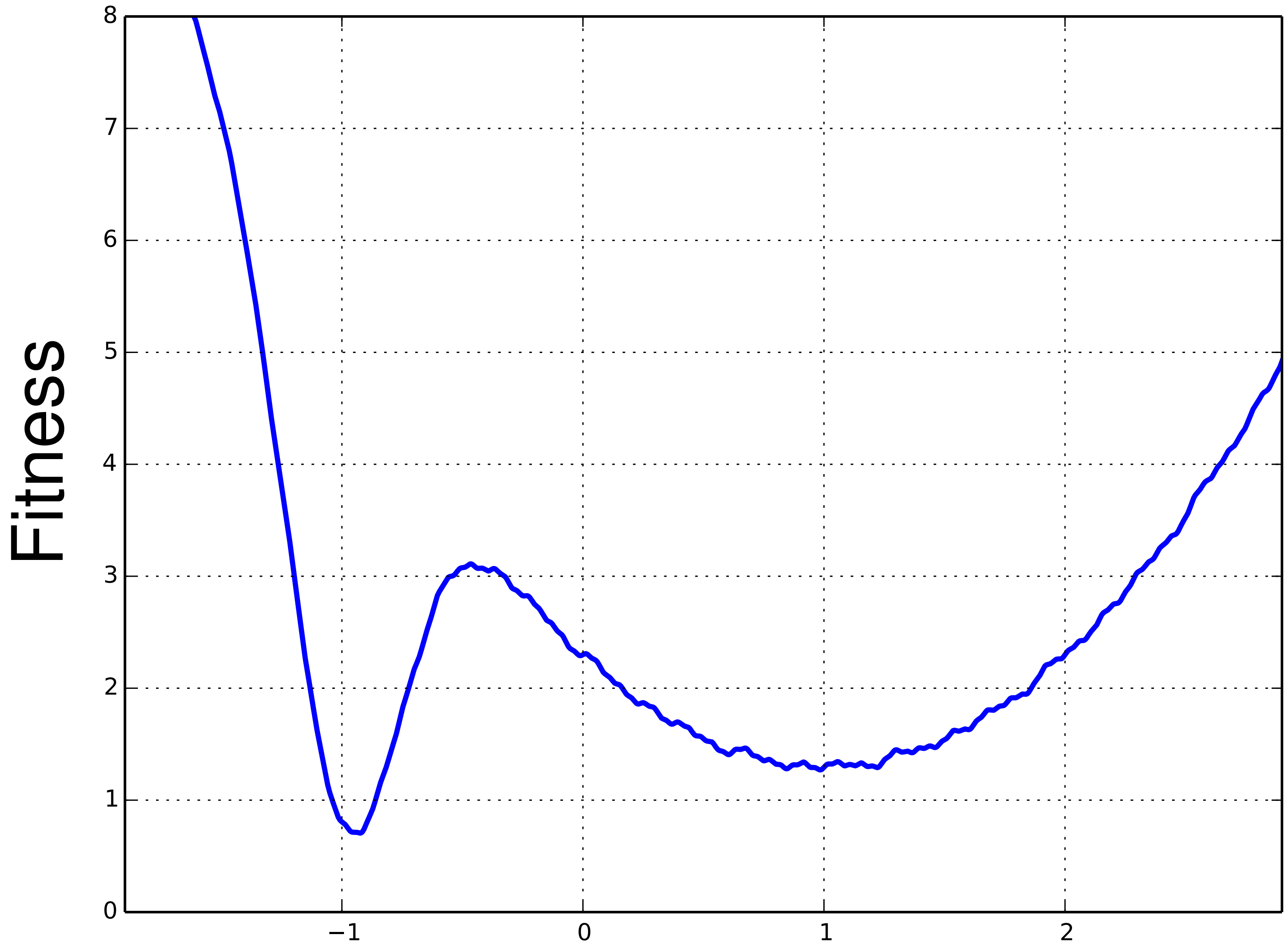
non-smooth, discontinuous, multimodal, and/or noisy



cut from a 5-D example, (easily) solvable with evolution strategies

Ruggedness

non-smooth, discontinuous, multimodal, and/or noisy



multi-funnel example

Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 20 points equally spaced onto the interval $[0, 1]$. Now consider the 10-dimensional space $[0, 1]^{10}$. To get similar coverage in terms of distance between adjacent points requires $20^{10} \approx 10^{13}$ points. 20 points appear now as isolated points in a vast empty space.

Remark: distance measures break down in higher dimensionalities (the central limit theorem kicks in)

Consequence: a search policy that is valuable in small dimensions might be useless in moderate or large dimensional search spaces. Example: exhaustive search.

Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 20 points equally spaced onto the interval $[0, 1]$. Now consider the 10-dimensional space $[0, 1]^{10}$. To get similar coverage in terms of distance between adjacent points requires $20^{10} \approx 10^{13}$ points. 20 points appear now as isolated points in a vast empty space.

Remark: distance measures break down in higher dimensionalities (the central limit theorem kicks in)

Consequence: a search policy that is valuable in small dimensions might be useless in moderate or large dimensional search spaces. Example: exhaustive search.

Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 20 points equally spaced onto the interval $[0, 1]$. Now consider the 10-dimensional space $[0, 1]^{10}$. To get similar coverage in terms of distance between adjacent points requires $20^{10} \approx 10^{13}$ points. 20 points appear now as isolated points in a vast empty space.

Remark: distance measures break down in higher dimensionalities (the central limit theorem kicks in)

Consequence: a search policy that is valuable in small dimensions might be useless in moderate or large dimensional search spaces. Example: exhaustive search.

Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 20 points equally spaced onto the interval $[0, 1]$. Now consider the 10-dimensional space $[0, 1]^{10}$. To get similar coverage in terms of distance between adjacent points requires $20^{10} \approx 10^{13}$ points. 20 points appear now as isolated points in a vast empty space.

Remark: distance measures break down in higher dimensionalities (the central limit theorem kicks in)

Consequence: a search policy that is valuable in small dimensions might be useless in moderate or large dimensional search spaces. Example: exhaustive search.

Separable Problems

Definition (Separable Problem)

A function f is separable if

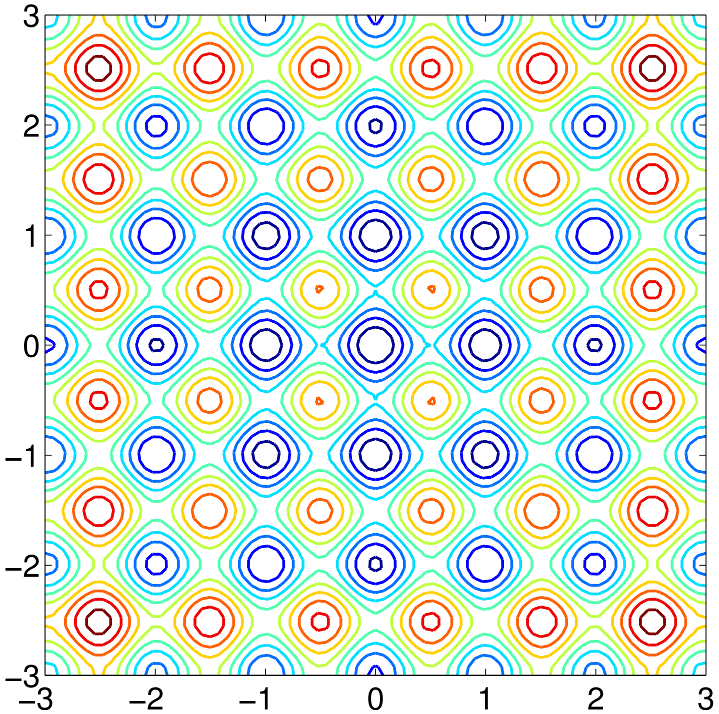
$$\arg\min_{(x_1,\ldots,x_n)} f(x_1,\ldots,x_n) = \left(\arg\min_{x_1} f(x_1,\ldots),\ldots,\arg\min_{x_n} f(\ldots,x_n)\right)$$

$\Rightarrow$ it follows that f can be optimized in a sequence of n independent 1-D optimization processes

Example: Additively decomposable functions

$$f(x_1,\ldots,x_n) = \sum_{i=1}^{n} f_i(x_i)$$

example: Rastrigin function, where $f_i = f_j \; \forall i,j$



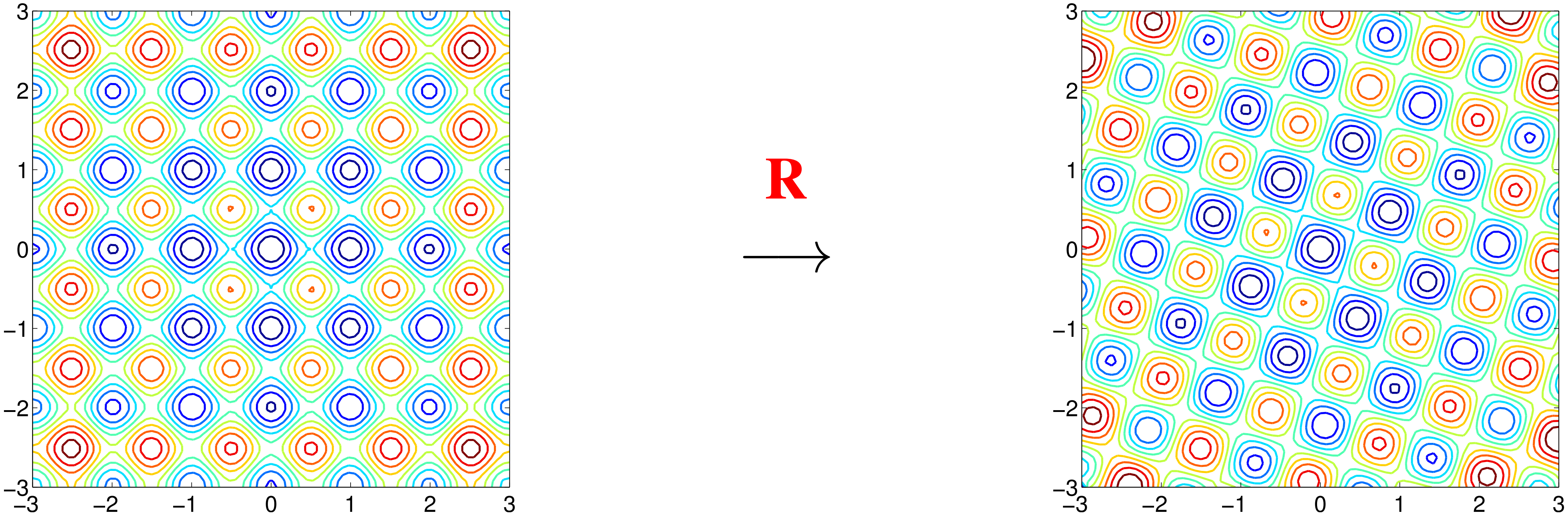
Non-Separable Problems

Building a non-separable problem from a separable one [1,2]

Rotating the coordinate system

- $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$ separable
- $f : \boldsymbol{x} \mapsto f(\mathbf{R}\boldsymbol{x})$ non-separable

R rotation matrix



[1] Hansen, Ostermeier, Gawelczyk (1995). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. Sixth ICGA, pp. 57-64, Morgan Kaufmann

[2] Salomon (1996). "Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions; A survey of some theoretical and practical aspects of genetic algorithms." BioSystems, 39(3):263-278

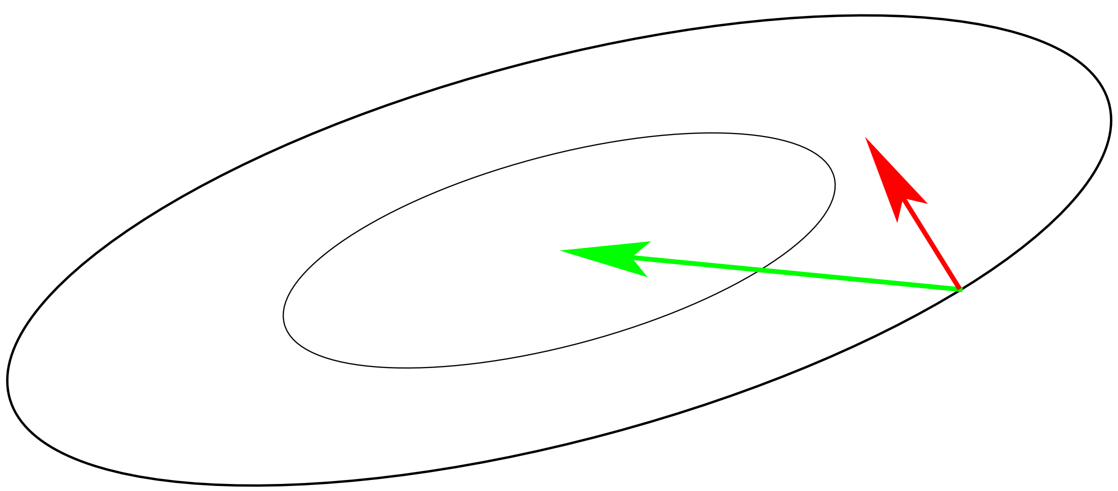
Ill-Conditioned Problems

Curvature of level sets

Consider the convex-quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^T\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}^*) = \frac{1}{2}\sum_i h_{i,i}\,(x_i - x_i^*)^2 + \frac{1}{2}\sum_{i\neq j} h_{i,j}\,(x_i - x_i^*)(x_j - x_j^*)$$

$\boldsymbol{H}$ is Hessian matrix of f and symmetric positive definite



gradient direction $-f'(\boldsymbol{x})^{\mathrm{T}}$

Newton direction $-\boldsymbol{H}^{-1}f'(\boldsymbol{x})^{\mathrm{T}}$

Ill-conditioning means squeezed level sets (high curvature). Condition number equals nine here. Condition numbers up to 10^{10} are not unusual in real world problems.

If $\boldsymbol{H} \approx \mathbf{I}$ (small condition number of $\boldsymbol{H}$) first order information (e.g. the gradient) is sufficient. Otherwise second order information (estimation of $\boldsymbol{H}^{-1}$) is necessary.

Landscape of Continuous Search Methods

Gradient-based (Taylor, local)

- Conjugate gradient methods [Fletcher & Reeves 1964]
- Quasi-Newton methods (BFGS) [Broyden et al 1970]

Derivative-free optimization (DFO)

- Trust-region methods (NEWUOA, BOBYQA) [Powell 2006, 2009]
- Simplex downhill [Nelder & Mead 1965]
- Pattern search [Hooke & Jeeves 1961, Audet & Dennis 2006]

Stochastic (randomized) search methods

- Evolutionary algorithms (broader sense, continuous domain)
  - Differential Evolution [Storn & Price 1997]
  - Particle Swarm Optimization [Kennedy & Eberhart 1995]
  - **Evolution Strategies** [Rechenberg 1965, Hansen & Ostermeier 2001]
- Simulated annealing [Kirkpatrick et al 1983]
- Simultaneous perturbation stochastic approximation (SPSA) [Spall 2000]

What Makes a Function Difficult to Solve?

. . . and what can be done

The Problem	Possible Approaches
Dimensionality	exploiting the problem structure separability, locality/neighborhood, encoding
Ill-conditioning	second order approach changes the neighborhood metric
Ruggedness	non-local policy, large sampling width (step-size) as large as possible while preserving a reasonable convergence speed
	population-based method, stochastic, non-elitistic
	recombination operator serves as repair mechanism
	restarts

. . . metaphors

What Makes a Function Difficult to Solve?

. . . and what can be done

The Problem	Possible Approaches
Dimensionality	exploiting the problem structure separability, locality/neighborhood, encoding
Ill-conditioning	second order approach changes the neighborhood metric
Ruggedness	non-local policy, large sampling width (step-size) as large as possible while preserving a reasonable convergence speed
	population-based method, stochastic, non-elitistic
	recombination operator serves as repair mechanism
	restarts

. . . metaphors

What Makes a Function Difficult to Solve?

. . . and what can be done

The Problem	Possible Approaches
Dimensionality	exploiting the problem structure separability, locality/neighborhood, encoding
Ill-conditioning	second order approach changes the neighborhood metric
Ruggedness	non-local policy, large sampling width (step-size) as large as possible while preserving a reasonable convergence speed population-based method, stochastic, non-elitistic recombination operator serves as repair mechanism restarts

. . . metaphors

Questions?