Differential Evolution

Based in part on the book “Differential Evolution: A Practical Approach to Global Optimization” by Price, Storn and Lampinen (2006)

Universiteit Leiden
The Netherlands

Differential Evolution

- Based on Darwinian evolution
- Improved version of Genetic Annealing (Price 1994)
- Storn and Price developed differential evolution in the following years
- Population based numerical optimization
- Exclusively floating-point encoding

Genetic Annealing

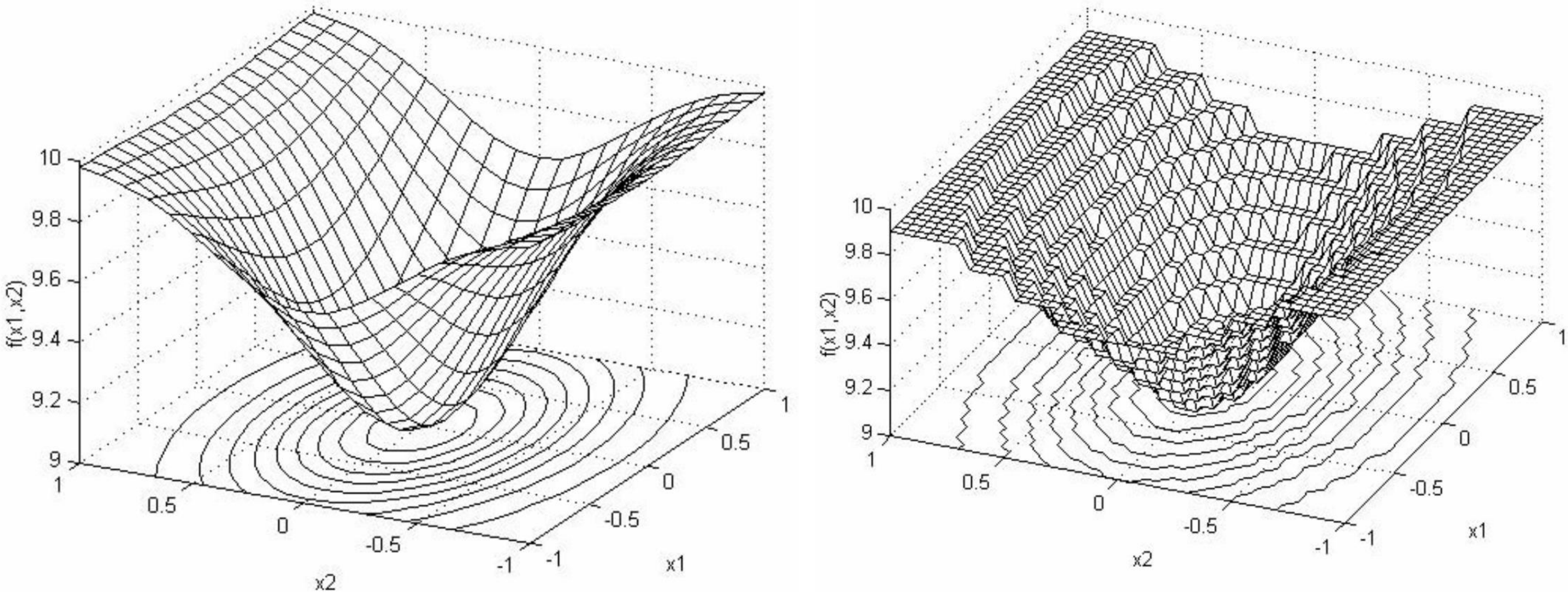
- Combination of the Genetic Algorithm and Simulated Annealing
- Slow convergence on more difficult problems

  (e.g. Chebyshev polynomial fitting)
- Difficult to tune

	Genetic Annealing	Differential Evolution
Encoding	Bit-string	Floating-point
Operations	Logical	Arithmetic
Solver type	Combinatorial	Numerical

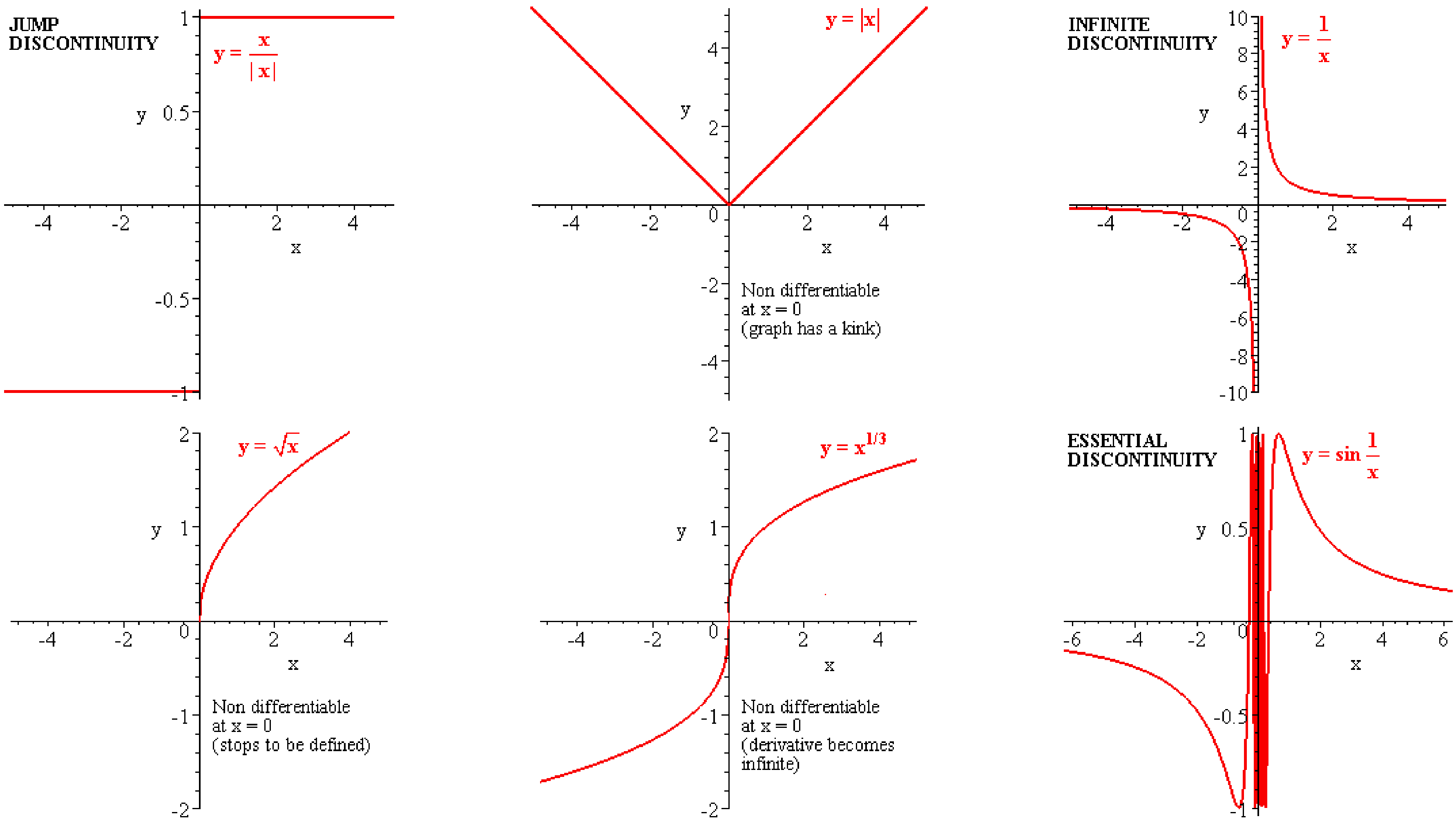
Differentiability

- Traditionally:
  - Differentiable: Gradient based optimization
  - Non-differentiable: Stochastic optimization



Source: Price, Storn, Lampinen (2006), Fig 1.5 (left) and Fig 1.7 (right)

Non-Differentiable Examples



Source: http://www-math.mit.edu/~djk/calculus_beginners/chapter09/section02.html

Differential Evolution Algorithm

1. Generate initial population of size $Np \geq 4$
2. Generate Np mutant vectors
3. Crossover parent population with mutant vectors
4. Select the best individual from each pair of parent and offspring individuals
5. If (not terminate) go to 2

Initialization

- Select parameter values for:
  - $Np \geq 4$: Population size
  - $F \in [0,1]$: Scaling factor, usually not greater than 1.0
  - $Cr \in [0,1]$: Crossover rate

$$x_{j,i,0} = rand_j(0,1) \cdot \left(b_{j,U} - b_{j,L}\right) + b_{j,L}$$

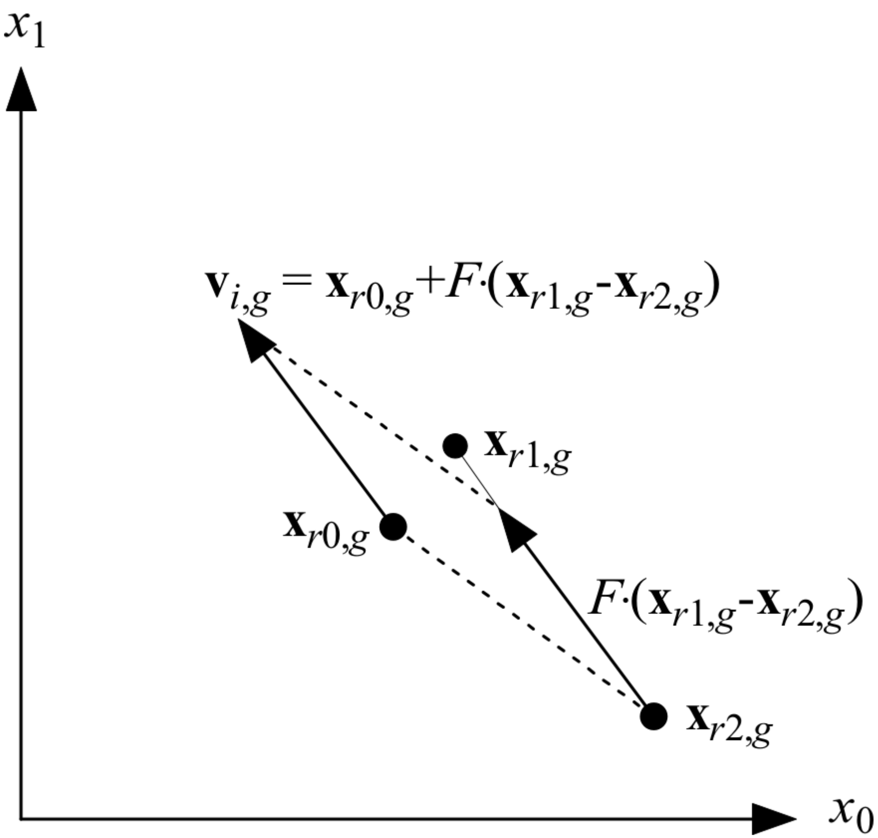
$rand(0,1)$ is uniform in $[0,1]$
i is the individual index
j is the parameter index

b_L is the lower bound
b_U is the upper bound
g is the generation (here 0)

Source: Price, Storn, Lampinen (2006), Eq. 2.4

Mutation

- $F \in [0,1]$: Scaling factor, usually not greater than 1.0



$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g})$$

v_i mutant vector to compare with parent x_i

x_{r0}, x_{r1}, x_{r2} randomly selected vectors distinct from each other and x_i

Source: Price, Storn, Lampinen (2006), Fig. 2.1, Eq. 2.5

Mutation example

$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g})$$

$$F = 0.5$$

Ind	Vec
x_1	$(4,8)$
x_2	$(6,5)$
x_3	$(9,2)$
x_4	$(1,7)$

Mut	x_{r0}	x_{r1}	x_{r2}	$(x_{r1} - x_{r2})$	$\times F$	$+x_{r0}$
v_1	x_2	x_4	x_3	$(-8,5)$	$(-4,2.5)$	$(2,7.5)$
v_2	x_3	x_4	x_1	$(-3,-1)$	$(-1.5,-0,5)$	$(7.5,1.5)$
v_3	x_1	x_2	x_4	$(5,-2)$	$(2.5,-1)$	$(6.5,7)$
v_4	x_3	x_2	x_1	$(2,-3)$	$(1,-1.5)$	$(10,0.5)$

How is the step size controlled?

→ By the difference between parameter values!

Mutation example

$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g})$$

$F = 0.5$

Ind	Vec
x_1	$(4,8)$
x_2	$(6,5)$
x_3	$(9,2)$
x_4	$(1,7)$

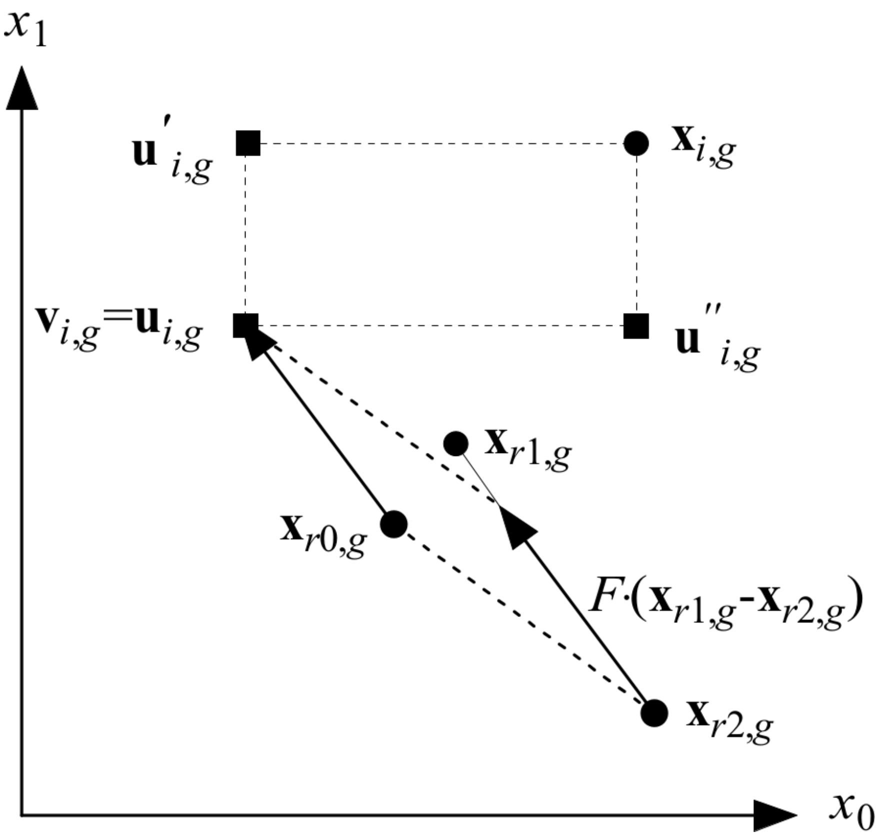
Mut	x_{r0}	x_{r1}	x_{r2}	$(x_{r1} - x_{r2})$	$\times F$	$+x_{r0}$
v_1	x_2	x_4	x_3	$(-8,5)$	$(-4,2.5)$	$(2,7.5)$
v_2	x_3	x_4	x_1	$(-3,-1)$	$(-1.5,-0,5)$	$(7.5,1.5)$
v_3	x_1	x_2	x_4	$(5,-2)$	$(2.5,-1)$	$(6.5,7)$
v_4	x_3	x_2	x_1	$(2,-3)$	$(1,-1.5)$	$(10,0.5)$

$x_{r1} = x_{r2}$

$F = 1.0?$

v	x_2	x_4	x_4	$(0,0)$	$(0,0)$	$(6,5)$
v_1	x_2	x_4	x_3	$(-8,5)$	$(-8,5)$	$(-2,10)$
v_1	x_4	x_2	x_3	$(-3,3)$	$(-3,3)$	$(-2,10)$

Crossover



$Cr \in [0,1]$: Crossover rate

u_i trial vector to compare with parent x_i

For every j:
Take parameter j from the mutant v with a probability, or from parent x otherwise

Take at least $j = j_{rand}$ from the mutant to ensure u_i does not duplicate x_i

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} \; if \left(rand_j(0,1) \leq Cr \; or \; j = j_{rand}\right) \\ x_{j,i,g} \; otherwise \end{cases}$$

Source: Price, Storn, Lampinen (2006), Fig. 2.2, Eq. 2.6

Crossover example

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} \; if \left(rand_j(0,1) \leq Cr \; or \; j = j_{rand} \right) \\ x_{j,i,g} \; otherwise \end{cases}$$

Ind	$j = 0$	$j = 1$
x_1	4	8
v_1	2	7.5
u_1	2	8

$j_{rand} = 0$

$$Cr = 1? \rightarrow u_1 = v_1$$

Selection

$$x_{i,g+1} = \begin{cases} u_{i,g} \; if\left(f(u_{i,g}) \leq f(x_{i,g})\right) \\ x_{i,g} \; otherwise \end{cases}$$

For every individual i:

Take trial vector u_i if it improves the objective value compared to parent x_i

Otherwise keep parent x_i

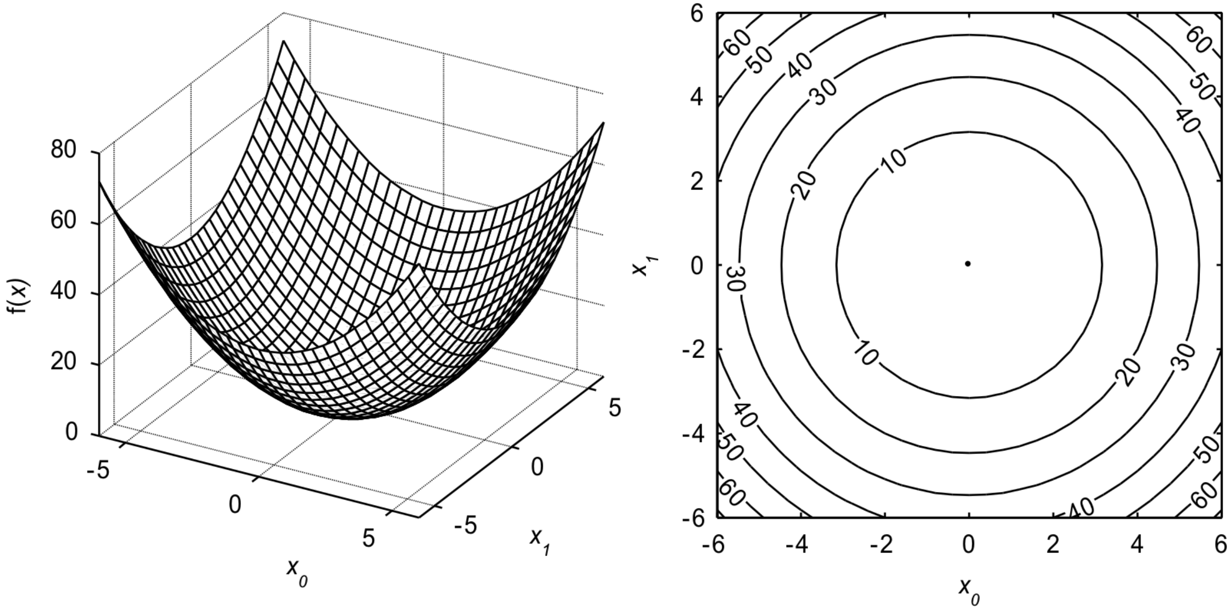
Source: Price, Storn, Lampinen (2006), Eq. 2.7

Selection example

- Objective (sphere function):

$$f(x) = \sum_{j=0}^{D-1} x_j^2 \rightarrow min$$

Ind	$j = 0$	$j = 1$	f
x_1	4	8	80
u_1	2	8	68
x_1^{new}	2	8	68



Source: Price, Storn, Lampinen (2006), Fig. A.1

Combined

$$u_{j,i,g} = \begin{cases} x_{j,r0,g} + F \cdot \left(x_{j,r1,g} - x_{j,r2,g}\right) \ if \left(rand_j(0,1) \leq Cr \ or \ j = j_{rand}\right) \\ x_{j,i,g} \ otherwise \end{cases}$$

$$j = 0,1, \dots, D-1; j_{rand} \in \{0,1, \dots, D-1\}$$
$$i = 0,1, \dots, Np - 1$$
$$g = 0,1, \dots, g_{max}$$
$$r0, r1, r2 \in \{0,1, \dots, Np - 1\}, r0 \neq r1 \neq r2 \neq i$$

$$x_{i,g+1} = \begin{cases} u_{i,g} \ if \left(f(u_{i,g}) \leq f(x_{i,g})\right) \\ x_{i,g} \ otherwise \end{cases}$$

Source: Price, Storn, Lampinen (2006), Eq. 2.8

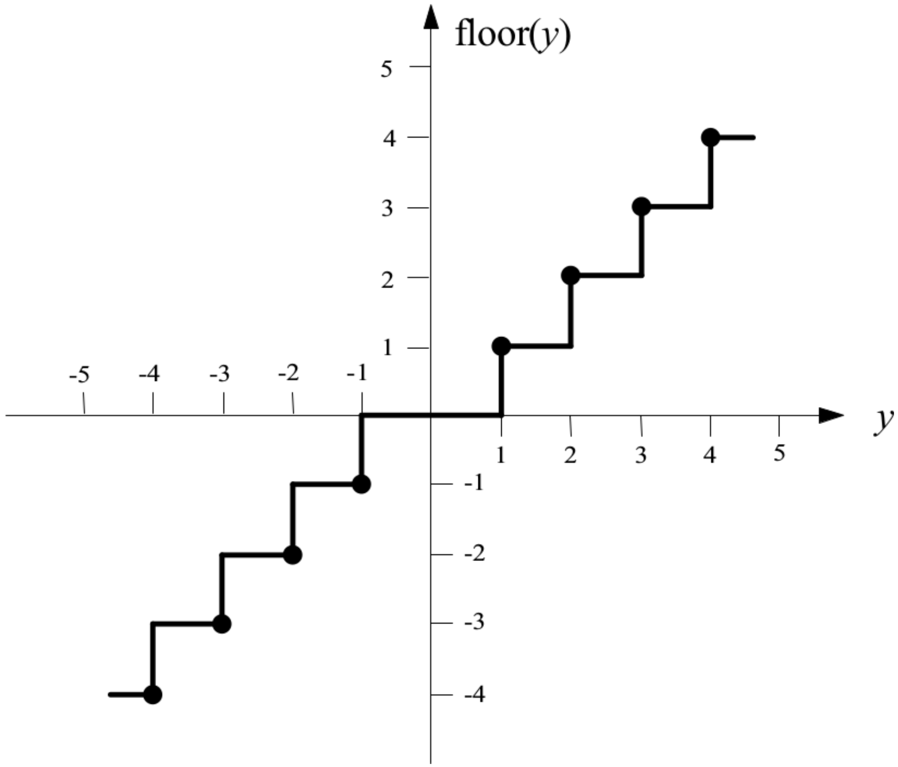
Settings

- $F > 1.0$?
  - Empirically shown to be more time consuming and less reliable
- $F = 1.0$?
  - Number of mutations halved since swapping $r0$ and $r1$ would result in the same outcome from mutation
- $r1 = r2$?
  - No mutation since their difference will be zero
- $Cr = 1.0$?
  - No crossover since the parent is never selected

Discrete parameters

- Map to and from real values

$$Q(y) = \frac{floor(k \cdot y)}{k}$$



Source: Price, Storn, Lampinen (2006), Fig. 4.2, Eq. 4.1

Algorithm outline

1. Create an initial population $\{x_{i=0}, \dots, x_{Np-1}\}$ of Np random real-valued vectors;
2. Decode each vector into a solution;
3. Evaluate fitness of each solution;
4. **Repeat**
   5. **For** *each vector* $x^j \in \{x^1, \dots, x^n\}$ **do**
      6. Select three other vectors randomly from the population;
      7. Apply difference vector to base vector to create variant vector;
      8. Combine vector x_i with variant vector to produce new trial vector;
      9. Evaluate fitness of the new trial vector;
      10. **If** *trial vector has higher fitness than* x_i **then**
          11. Replace x_i with the trial vector;
      12. **End**
   13. **End**
14. **End**

Source: Brabazon, O'Neill, McGarraghy (2015), Alg. 6.1

DE Advantages and Disadvantages

- Advantages
  - Few parameters to tune
  - Search automatically scales from global to local
- Disadvantages
  - Dependence on initial points
  - Local optima? No automatic scaling back from local to global
  - Requires decoding functions for discrete values

Differential Evolution and Evolution Strategies

	Differential Evolution	Evolution Strategy
Mutation	Vector differences	Stochastic
Recombination	Mutant with parent	Parent with parent
Selection	Individual parent and offspring comparison	Population based
Step size adaptation	Implicit through vector differences	Based on normal distribution

Recommended Reading

- Storn’s website
  - http://www1.icsi.berkeley.edu/~storn/code.html
  - Algorithm history
  - Code in various languages
  - Parameter recommendations
  - More literature

Literature

- Price, Kenneth V. "Genetic annealing." DR DOBBS JOURNAL 19.11 (1994): 127.
- Storn, R., and K. Price. "DE-a Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Space." International Computer Science Institute, Technical report TR-95-012 (1995): 1-12.
- Storn, Rainer, and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." Journal of global optimization 11.4 (1997): 341-359.
- Price, Kenneth, Rainer M. Storn, and Jouni A. Lampinen. Differential evolution: a practical approach to global optimization. Springer Science & Business Media, 2006.
- Brabazon, Anthony, Michael O’Neill, and Seán McGarraghy. Natural Computing Algorithms. Springer-Verslag Berlin Heidelberg, 2015.

Questions?

Universiteit Leiden
The Netherlands