Trends in Real-Parameter Optimization

Petr Pošík

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics
Intelligent Data Analysis Group

25. 4. 2008

Introduction
What is this talk about?
Black-Box Optimization
Local Search
Local Search Demo
Box Method
Rosenbrock's Optimization Algorithm
Rosenbrock's Algorithm Demo
Nelder-Mead Simplex Search
Nelder-Mead Simplex Demo
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Introduction

Introduction
What is this talk about?
Black-Box Optimization
Local Search
Local Search Demo
Box Method
Rosenbrock's Optimization Algorithm
Rosenbrock's Algorithm Demo
Nelder-Mead Simplex Search
Nelder-Mead Simplex Demo
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

What is this talk about?

About my personal (subjective) view on

- ✔ the important "historical" algorithms,
- ✔ their important features, and on
- ✔ the promising recent algorithms.

Outline:

1. Motivation
   - ✔ several classical optimization techniques
2. Optimization algorithm framework
   - ✔ lessons learned
   - ✔ various opt. techniques
3. State-of-the-art techniques
4. Comparison

Black-Box Optimization

Introduction
What is this talk about?
Black-Box Optimization
Local Search
Local Search Demo
Box Method
Rosenbrock's Optimization Algorithm
Rosenbrock's Algorithm Demo
Nelder-Mead Simplex Search
Nelder-Mead Simplex Demo
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Optimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$$

- ✔ $\mathbf{x}$ is the object representation,
- ✔ $\mathcal{S}$, $x \in \mathcal{S}$, is the search space of all candidate objects,
- ✔ f is the objective function to be minimized.

Introduction
What is this talk about?
Black-Box Optimization
Local Search
Local Search Demo
Box Method
Rosenbrock's Optimization Algorithm
Rosenbrock's Algorithm Demo
Nelder-Mead Simplex Search
Nelder-Mead Simplex Demo
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Optimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$$

- ✔ $\mathbf{x}$ is the object representation,
- ✔ $\mathcal{S}$, $x \in \mathcal{S}$, is the search space of all candidate objects,
- ✔ f is the objective function to be minimized.

Black-box optimization problem: Nothing is known about f, it can be

- ✔ non-convex, non-differentiable
- ✔ multimodal,
- ✔ time-dependent, noisy,
- ✔ ...

Optimizers are based on direct sampling of the search space. They require only

1. *representation* of candidate solution, and
2. *objective function* (fitness) to evaluate the solution domain.

Black-Box Optimization

Introduction
What is this talk about?
Black-Box Optimization
Local Search
Local Search Demo
Box Method
Rosenbrock's Optimization Algorithm
Rosenbrock's Algorithm Demo
Nelder-Mead Simplex Search
Nelder-Mead Simplex Demo
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Optimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$$

- ✔ $\mathbf{x}$ is the object representation,
- ✔ $\mathcal{S}$, $x \in \mathcal{S}$, is the search space of all candidate objects,
- ✔ f is the objective function to be minimized.

Black-box optimization problem: Nothing is known about f, it can be

- ✔ non-convex, non-differentiable
- ✔ multimodal,
- ✔ time-dependent, noisy,
- ✔ ...

Optimizers are based on direct sampling of the search space. They require only

1. *representation* of candidate solution, and
2. *objective function* (fitness) to evaluate the solution domain.

Real-parameter black-box optimization problem

- ✔ $\mathcal{S} = \mathcal{R}^D$ where D is the dimensionality of the search space

Local Search

Local search with *first improving strategy*:

Algorithm 1: Hill-Climbing

```
begin
    x ← Initialize()
    while not TerminationCondition() do
        y ← Perturb(x)
        if BetterThan(y, x) then
            x ← y
end
```

Local search with *first improving strategy*:

Algorithm 1: Hill-Climbing

```
begin
    x ← Initialize()
    while not TerminationCondition() do
        y ← Perturb(x)
        if BetterThan(y, x) then
            x ← y
end
```

Neighborhood used in `Perturb()`:

- ✔ deterministic (pattern) vs. stochastic
- ✔ static vs. time-dependent vs. adaptive

Local Search

Local search with *first improving strategy*:

Algorithm 1: Hill-Climbing

```
begin
    x ← Initialize()
    while not TerminationCondition() do
        y ← Perturb(x)
        if BetterThan(y, x) then
            x ← y
end
```

Neighborhood used in `Perturb()`:

- ✔ deterministic (pattern) vs. stochastic
- ✔ static vs. time-dependent vs. adaptive

DEMO:

- ✔ `Perturb` uses isotropic Gaussian distribution with static parameters

Local Search

Local search with *first improving strategy*:

Algorithm 1: Hill-Climbing

```
begin
    x ← Initialize()
    while not TerminationCondition() do
        y ← Perturb(x)
        if BetterThan(y, x) then
            x ← y
end
```

Neighborhood used in `Perturb()`:

- ✔ deterministic (pattern) vs. stochastic
- ✔ static vs. time-dependent vs. adaptive

DEMO:

- ✔ `Perturb` uses isotropic Gaussian distribution with static parameters

Local search with *best improving strategy*:

- ✔ complete search of the neighborhood in discrete spaces (impossible in $\mathcal{R}^{\mathcal{D}}$)
- ✔ knowledge of derivatives in the neighborhood would help (gradient algorithm), but we face black-box
- ✔ could be approximated by creating more candidates from the neighborhood and selecting the best one

Local search with *first improving strategy*:

Algorithm 1: Hill-Climbing

```
begin
    x ← Initialize()
    while not TerminationCondition() do
        y ← Perturb(x)
        if BetterThan(y, x) then
            x ← y
end
```

Neighborhood used in `Perturb()`:

- ✔ deterministic (pattern) vs. stochastic
- ✔ static vs. time-dependent vs. adaptive

DEMO:

- ✔ `Perturb` uses isotropic Gaussian distribution with static parameters

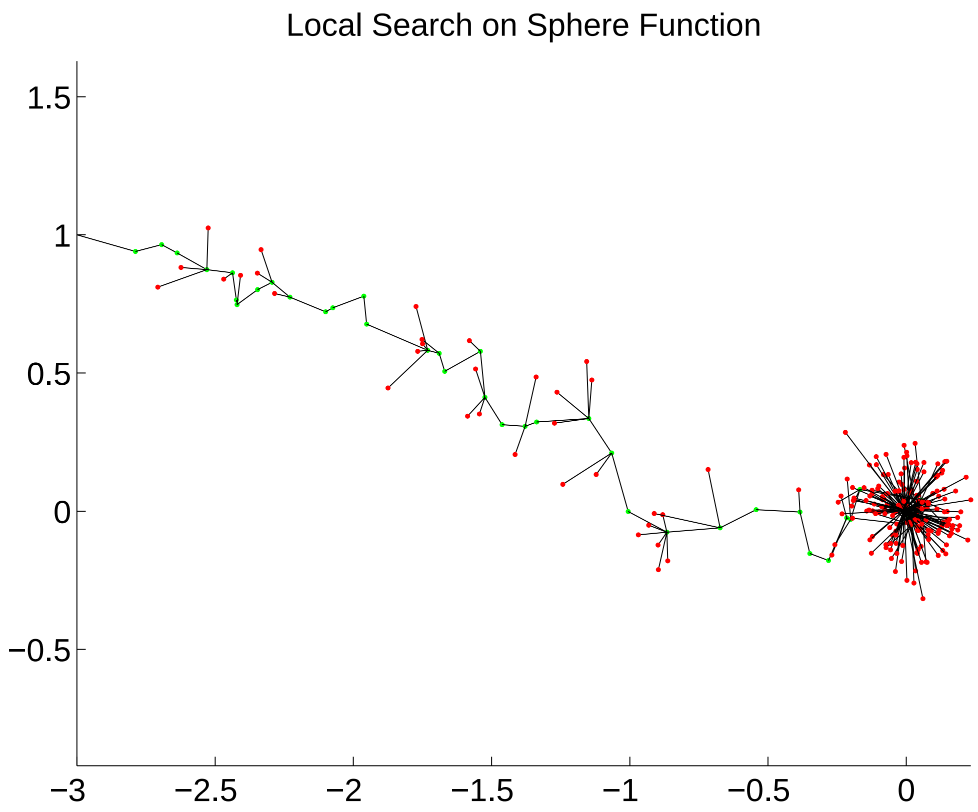
Local search with *best improving strategy*:

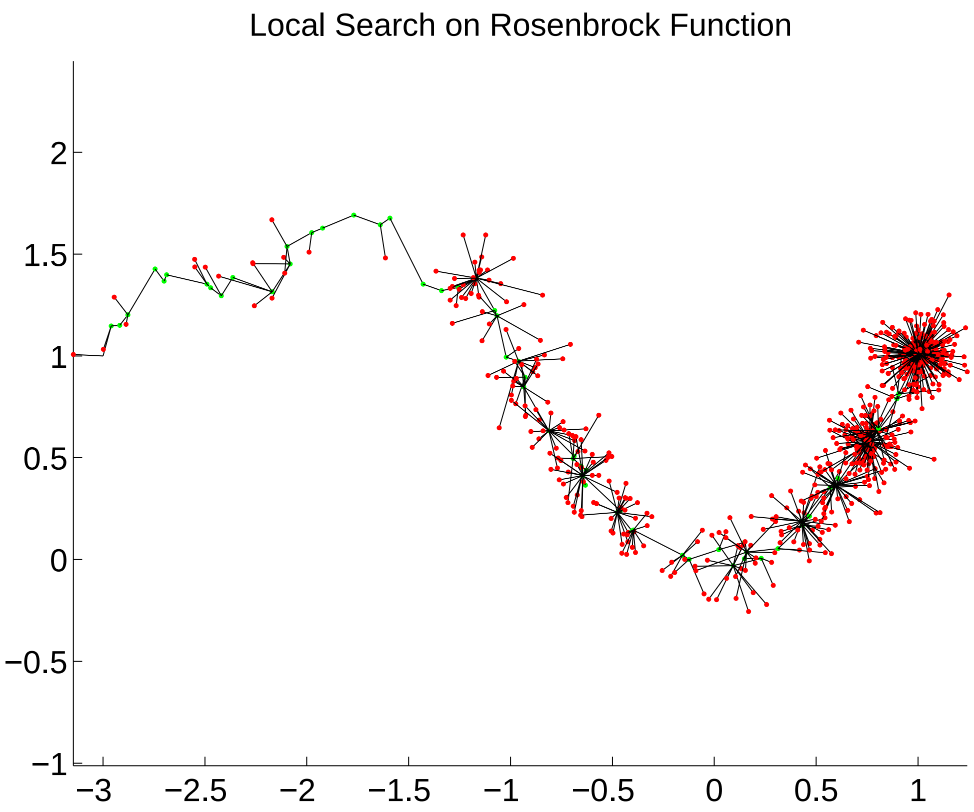
- ✔ complete search of the neighborhood in discrete spaces (impossible in $\mathcal{R}^{\mathcal{D}}$)
- ✔ knowledge of derivatives in the neighborhood would help (gradient algorithm), but we face black-box
- ✔ could be approximated by creating more candidates from the neighborhood and selecting the best one

Parallel local search:

- ✔ equivalent to restarted local search
- ✔ uses a set of points $\mathbf{x}$

Local Search Demo





Introduction
What is this talk about?
Black-Box Optimization
Local Search
Local Search Demo
Box Method
Rosenbrock's Optimization Algorithm
Rosenbrock's Algorithm Demo
Nelder-Mead Simplex Search
Nelder-Mead Simplex Demo
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Box Method

Evolutionary operation [Box57]:

Algorithm 2: Box's Evolutionary Operation

```
begin
    x ← Initialize()
    while not TerminationCondition() do
        𝒴 ← CornersOfBoxAround(x)
        x ← BestOf(𝒴)
end
```

Features:

- ✔ iterative application of 2-level full-factorial design of experiment
- ✔ 2^D candidates generated each iteration
- ✔ neighborhood in the form of a pattern
- ✔ static neighborhood parameters

Possible improvements:

- ✔ adaptive box size, e.g.
  - ✘ increase box size, if one of the corners improves solution, decrease box size, if no improvement found.

[Box57] G.E.P. Box. Evolutionary operation: Method for increasing industrial productivity. *Appl Stat*, 6(2):81–101, 1957.

Rosenbrock's Optimization Algorithm

Described in [Ros60]:

Algorithm 3: Rosenbrock's Algorithm

Input: $\alpha > 1$, $\beta \in (0,1)$

begin
    $\mathbf{x} \leftarrow$ `Initialize()`; $\mathbf{x}_o \leftarrow \mathbf{x}$
    $\{\mathbf{e}_1, \dots, \mathbf{e}_D\} \leftarrow$ `InitOrtBasis()`
    $\{d_1, \dots, d_D\} \leftarrow$ `InitMultipliers()`
    while not `TerminationCondition()` **do**
        for i=$1 \dots D$ **do**
            $\mathbf{y} \leftarrow \mathbf{x} + d_i \mathbf{e}_i$
            if `BetterThan(`y,x`)` **then**
                $\mathbf{x} \leftarrow \mathbf{y}$
                $d_i \leftarrow \alpha \cdot d_i$
            else
                $d_i \leftarrow -\beta \cdot d_i$
        if `AtLeastOneSuccInAllDirs()` **and** `AtLeastOneFailInAllDirs()` **then**
            $\{\mathbf{e}_1, \dots, \mathbf{e}_D\} \leftarrow$ `UpdOrtBasis(`$\mathbf{x} - \mathbf{x}_o$`)`
            $\mathbf{x}_o \leftarrow \mathbf{x}$
end

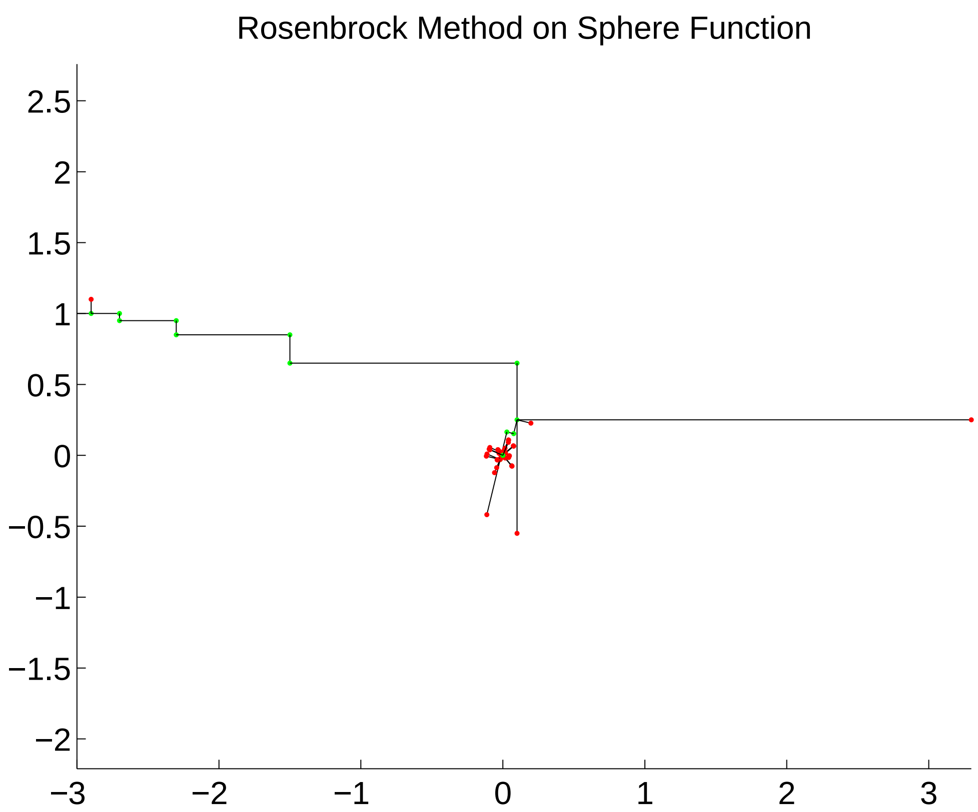
Features:

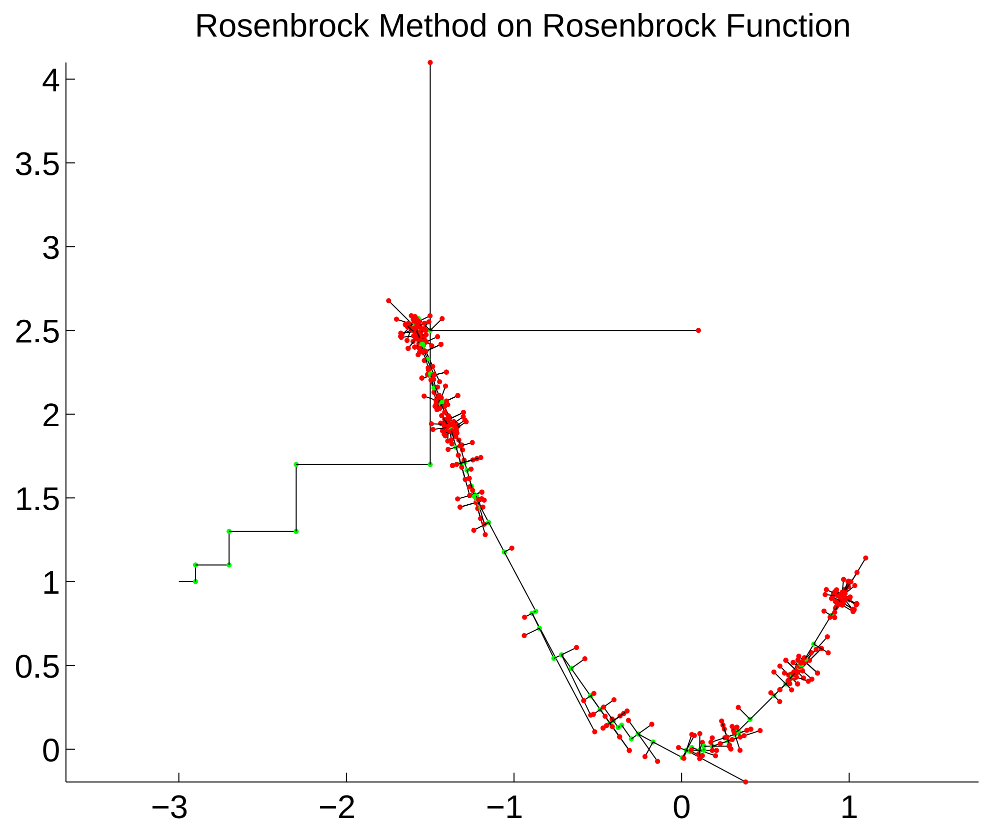
- ✔ D candidates generated each iteration
- ✔ neighborhood in the form of a pattern
- ✔ adaptive neighborhood parameters
  - ✘ distances
  - ✘ directions

DEMO

[Ros60] H.H. Rosenbrock. An automatic method for findong the greatest or least value of a function. *The Computer Journal*, 3:175–184, 1960.

Rosenbrock's Algorithm Demo



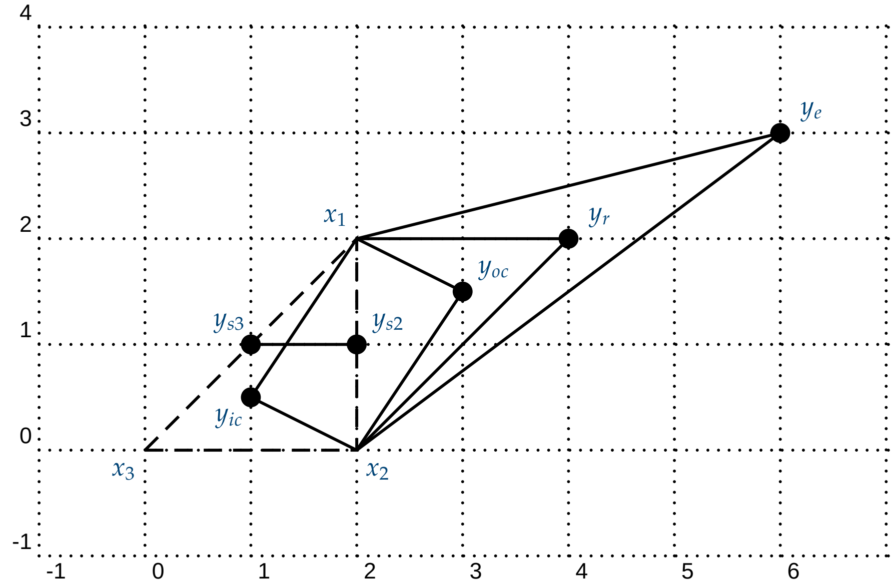


Nelder-Mead Simplex Search

Simplex downhill search (amoeba) [NM65]:

Algorithm 4: Nelder-Mead Simplex Algorithm

begin
    $(\mathbf{x}_1, \ldots, \mathbf{x}_{D+1}) \leftarrow$ `InitSimplex()`
    `so that` $f(x_1) \leq f(x_2) \leq \ldots \leq f(x_{D+1})$
    while not `TerminationCondition()` **do**
        $\bar{\mathbf{x}} \leftarrow \frac{1}{D} \sum_{d=1}^{D} \mathbf{x}_d$
        $\mathbf{y}_r \leftarrow \bar{\mathbf{x}} + \rho(\bar{\mathbf{x}} - \mathbf{x}_{D+1})$
        if `BetterThan`$(\mathbf{y}_r, \mathbf{x}_D)$ **then** $\mathbf{x}_{D+1} \leftarrow \mathbf{y}_r$
        if `BetterThan`$(\mathbf{y}_r, \mathbf{x}_1)$ **then**
            $\mathbf{y}_e \leftarrow \bar{\mathbf{x}} + \chi(\mathbf{x}_r - \bar{\mathbf{x}})$
            if `BetterThan`$(\mathbf{y}_e, \mathbf{y}_r)$ **then** $\mathbf{x}_{D+1} \leftarrow \mathbf{y}_e$;
            `Continue`
        if not `BetterThan`$(\mathbf{y}_r, \mathbf{x}_D)$ **then**
            if `BetterThan`$(\mathbf{y}_r, \mathbf{x}_{D+1})$ **then**
                $\mathbf{y}_{oc} \leftarrow \bar{\mathbf{x}} + \gamma(\mathbf{x}_r - \bar{\mathbf{x}})$
                if `BetterThan`$(\mathbf{y}_{oc}, \mathbf{y}_r)$ **then**
                $\mathbf{x}_{D+1} \leftarrow \mathbf{y}_{oc}$; `Continue`
            else
                $\mathbf{y}_{ic} \leftarrow \bar{\mathbf{x}} - \gamma(\bar{\mathbf{x}} - \mathbf{x}_{D+1})$
                if `BetterThan`$(\mathbf{y}_{ic}, \mathbf{x}_{D+1})$ **then**
                $\mathbf{x}_{D+1} \leftarrow \mathbf{y}_{ic}$; `Continue`
        $\mathbf{y}_{si} \leftarrow \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1), \quad i \in 2, \ldots, D+1$
        `MakeSimplex`$(\mathbf{x}_1, \mathbf{y}_{s2}, \ldots, \mathbf{y}_{s(D+1)})$
end



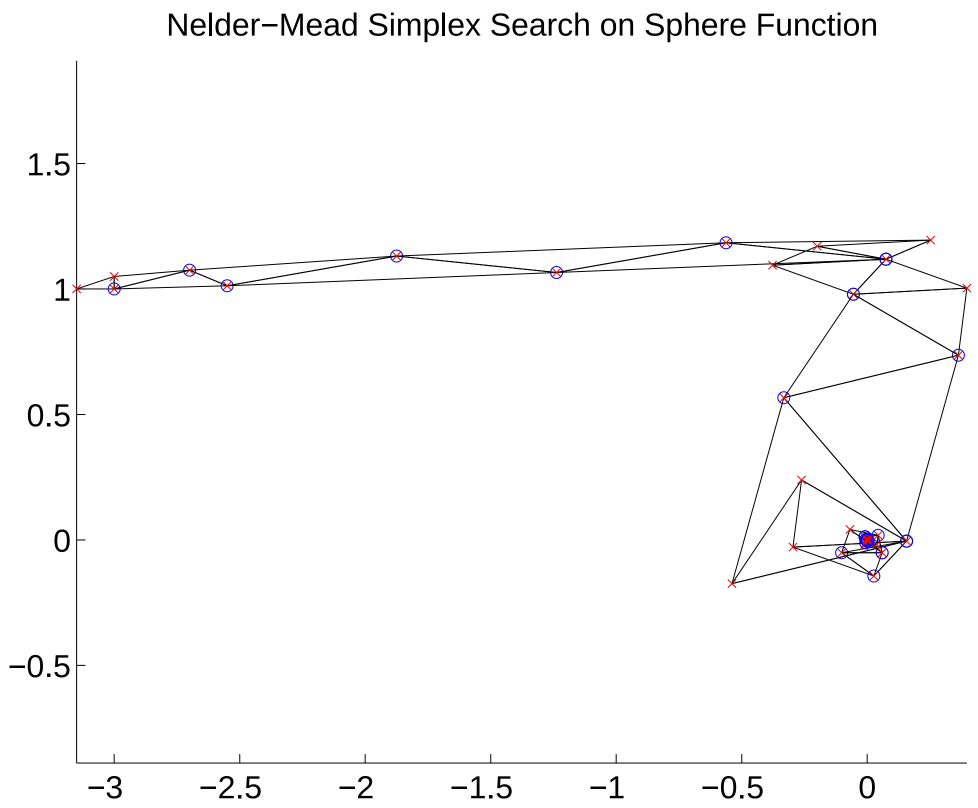
Features:

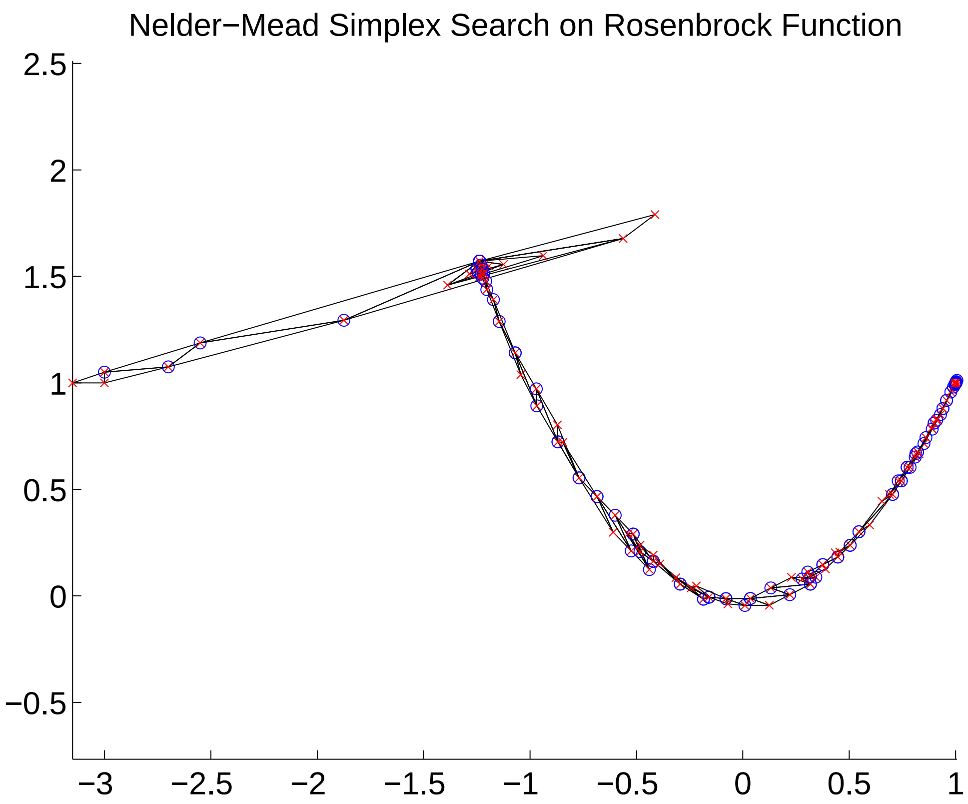
- ✔ $1 \ldots D+1$ candidates generated each iteration
- ✔ neighborhood in the form of a pattern
- ✔ static neighborhood parameters!
- ✔ adaptivity caused by changing relationships among solution vectors!

DEMO

[NM65] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

Nelder-Mead Simplex Demo





Introduction
Optimizer Description Framework
Lessons Learned Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Optimizer Description Framework

Lessons Learned

Introduction

Optimizer Description Framework

Lessons Learned

Optimizer Description Framework

Nature Inspired Optimization Techniques

Algorithm Comparison

Summary and Conclusions

- ✔ To *search for the optimum*, the algorithm must *maintain at least one base solution* (fullfiled by all algorithms).

Introduction
Optimizer Description Framework
Lessons Learned
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Lessons Learned

- ✔ To *search for the optimum*, the algorithm must *maintain at least one base solution* (fullfiled by all algorithms).
- ✔ To *adapt to the environmental changes* during the search (primarily not to changes of the fitness function itself, but to the changing position of the local neighborhood), the algorithm must either
  - ✘ *adapt the neighborhood (model)* structure or parameters (as done in Rosenbrock method), or
  - ✘ *adapt more than 1 base solutions* (as done in Nelder-Mead method), or
  - ✘ both of them.

Lessons Learned

Introduction
Optimizer Description Framework
Lessons Learned
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

- ✔ To *search for the optimum*, the algorithm must *maintain at least one base solution* (fullfiled by all algorithms).
- ✔ To *adapt to the environmental changes* during the search (primarily not to changes of the fitness function itself, but to the changing position of the local neighborhood), the algorithm must either
  - ✘ *adapt the neighborhood (model)* structure or parameters (as done in Rosenbrock method), or
  - ✘ *adapt more than 1 base solutions* (as done in Nelder-Mead method), or
  - ✘ both of them.
- ✔ The neighborhood
  - ✘ can be *finite*, have a form of a *pattern*, or
  - ✘ can be *infinite*, have a form of a *probabilistic distribution*.

Introduction
Optimizer Description Framework
Lessons Learned
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Lessons Learned

- ✔ To *search for the optimum*, the algorithm must *maintain at least one base solution* (fullfiled by all algorithms).
- ✔ To *adapt to the environmental changes* during the search (primarily not to changes of the fitness function itself, but to the changing position of the local neighborhood), the algorithm must either
  - ✘ *adapt the neighborhood (model)* structure or parameters (as done in Rosenbrock method), or
  - ✘ *adapt more than 1 base solutions* (as done in Nelder-Mead method), or
  - ✘ both of them.
- ✔ The neighborhood
  - ✘ can be *finite*, have a form of a *pattern*, or
  - ✘ can be *infinite*, have a form of a *probabilistic distribution*.
- ✔ Candidate solutions can be generated from the neighborhood of
  - ✘ one base vector (LS, Box, Rosenbrock), or
  - ✘ all base vectors (Nelder-Mead), or
  - ✘ some of the base vectors (requires *selection* operator).

Introduction
Optimizer Description Framework
Lessons Learned
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Optimizer Description Framework

1. Initialize model, $\mathcal{M}$.
2. Initialize base solution set, $\mathcal{B}$.
3. Select parent solution set, $\mathcal{P}$, from $\mathcal{B}$.
4. Update $\mathcal{M}$ based on $\mathcal{B}$ and $\mathcal{P}$.
5. Generate candidate solution set, $\mathcal{C}$, using $\mathcal{M}$, $\mathcal{B}$, and $\mathcal{P}$.
6. Update $\mathcal{B}$ based on $\mathcal{B}$, $\mathcal{P}$ and $\mathcal{C}$.

Introduction
Optimizer Description Framework
Lessons Learned
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions

Optimizer Description Framework

1. Initialize model, $\mathcal{M}$.
2. Initialize base solution set, $\mathcal{B}$.
3. Select parent solution set, $\mathcal{P}$, from $\mathcal{B}$.
4. Update $\mathcal{M}$ based on $\mathcal{B}$ and $\mathcal{P}$.
5. Generate candidate solution set, $\mathcal{C}$, using $\mathcal{M}$, $\mathcal{B}$, and $\mathcal{P}$.
6. Update $\mathcal{B}$ based on $\mathcal{B}$, $\mathcal{P}$ and $\mathcal{C}$.

Presented algorithms described in the framework:

Algorithm	$\|\mathcal{B}\|$	$\|\mathcal{P}\|$	$\|\mathcal{C}\|$	Neighb. type	Neighb. adaptivity
LS, first impr.	1	1	1	PDF	none
LS, “best impr.”	1	1	>1	PDF	none
LS, parallel	>1	$\|\mathcal{B}\|$	$\|\mathcal{B}\|$	PDF	none
Box	1	1	2^D	pattern	none
Rosenbrock	1	1	1	pattern	model
Nelder-Mead	D+1	D+1	1 or D	pattern	population

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

Nature Inspired Optimization Techniques

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

Evolution Strategies

Representation: $(\mathbf{x}, \sigma)$

- ✔ neigborhood parameters are part of the solution representation
- ✔ neighborhood is different for all parents

Select $\mathcal{P}$:

- ✔ select the whole $\mathcal{B}$ as $\mathcal{P}$

Model update:

$$\sigma \leftarrow \sigma \cdot e^{\tau z}, \quad z \sim \mathcal{N}(0,1)$$

Generate $\mathcal{C}$:

$$\mathbf{y} \leftarrow \mathbf{x} + \sigma \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

Population update:

- ✔ (μ, λ)-ES: select $\mathcal{B}_{new}$ from $\mathcal{C}$
- ✔ $(\mu + \lambda)$-ES: select $\mathcal{B}_{new}$ from $\mathcal{C}$ and $\mathcal{P}$

Summary in ODF:

- ✔ $|\mathcal{B}| = \mu$, $|\mathcal{P}| = \mu$, $|\mathcal{C}| = \lambda$
- ✔ Neighb. type: PDF (normal kernels)
- ✔ Neighb. adaptativity: population, model self-adaptation

[Rec65] Ingo Rechenberg. Cybernetic solution path of an experimental problem. Translation 1122, Royal Aircraft Establishment Library, 1965.

[Sch65] Hans-Paul Schwefel. Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik. Dipl.-ing. thesis, Technical University of Berlin, 1965.

[Sch95] Hans-Paul Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.

Select $\mathcal{P}$:

- ✔ select 4 vectors from $\mathcal{B}$ uniformly as $\mathcal{P}$
- ✔ DE/rand: $\mathbf{x}_1$ is a random member of $\mathcal{B}$
- ✔ DE/best: $\mathbf{x}_1$ is the best member of $\mathcal{B}$

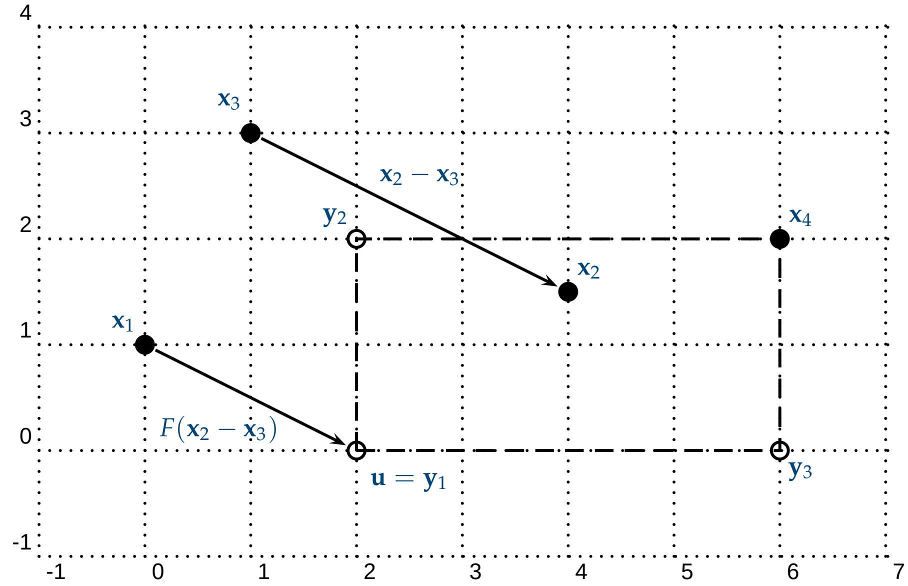
Model update: none

Generate $\mathcal{C}$:

$$\mathbf{u} \leftarrow \mathbf{x}_1 + F(\mathbf{x}_2 - \mathbf{x}_3), \quad F \in (0, 2)$$

$$y_d \leftarrow \begin{cases} u_d & \text{iff } \text{rand}_d \leq CR \text{ or } d = I_{\text{rand}} \\ x_{4,d} & \text{iff } \text{rand}_d > CR \text{ and } d \neq I_{\text{rand}} \end{cases}$$

- ✔ $\text{rand}_d \sim \mathcal{U}(0, 1)$, different for each dimension
- ✔ I_{rand} is a random index of the dimension that is always copied from $\mathbf{u}$
- ✔ $2^D - 1$ possible candidate points $\mathbf{y}$



Population update:

- ✔ if $\mathbf{y}$ is better than $\mathbf{x}_4$, replace $\mathbf{x}_4$ with $\mathbf{y}$ in $\mathcal{B}$

Summary in ODF:

- ✔ usually $|\mathcal{B}| > 1$, $|\mathcal{P}| = 4$, $|\mathcal{C}| = 1$
- ✔ Neighb. type: pattern! (PDF by randomization of F)
- ✔ Neighb. adaptativity: population

[SP95] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI, March 1995.

Particle Swarm Optimization

Initialize model:

- ✔ initialize particle velocities $\mathbf{v}_i$
- ✔ initialize personal best locations $\mathbf{x}_i^b$
- ✔ initialize global best location $\mathbf{x}^g$

Select $\mathcal{P}$:

- ✔ select the whole $\mathcal{B}$ as $\mathcal{P}$

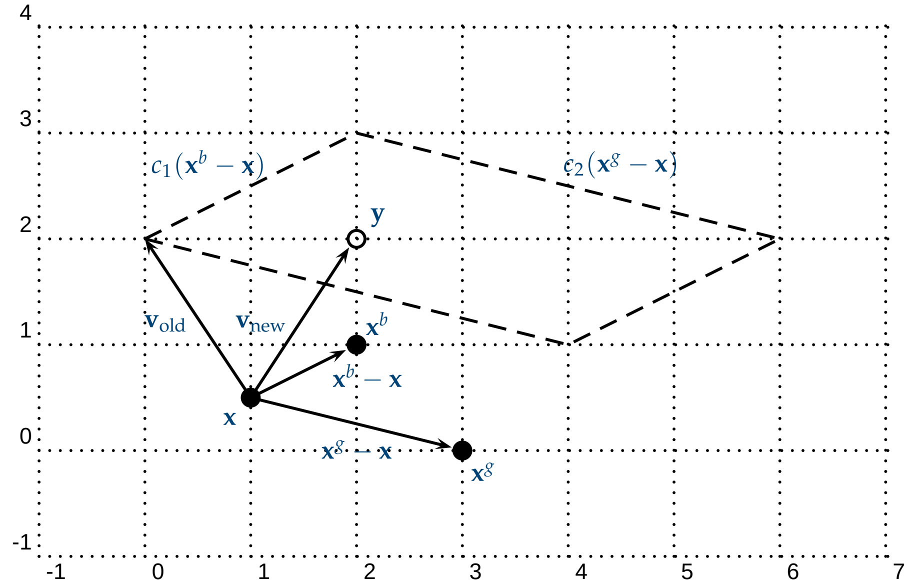
Model update:

$$\mathbf{v} \leftarrow \mathbf{v} + c_1 r_1 (\mathbf{x}^b - \mathbf{x}) + c_2 r_2 (\mathbf{x}^g - \mathbf{x})$$

- ✔ $r_1, r_2 \sim \mathcal{U}(0,1)$
- ✔ $c_1, c_2 \in (0,4)$, usually $c_1 = c_2 = 2$
- ✔ velocity vector $\mathbf{v}$ is updated to point to personal and global best solutions

Generate $\mathcal{C}$: $\mathbf{y} \leftarrow \mathbf{x} + \mathbf{v}$

- ✔ each particle i makes a step in the direction of $\mathbf{v}_i$



Population update: generational

- ✔ $\mathcal{B}_{new} = \mathcal{C}$

Summary in ODF:

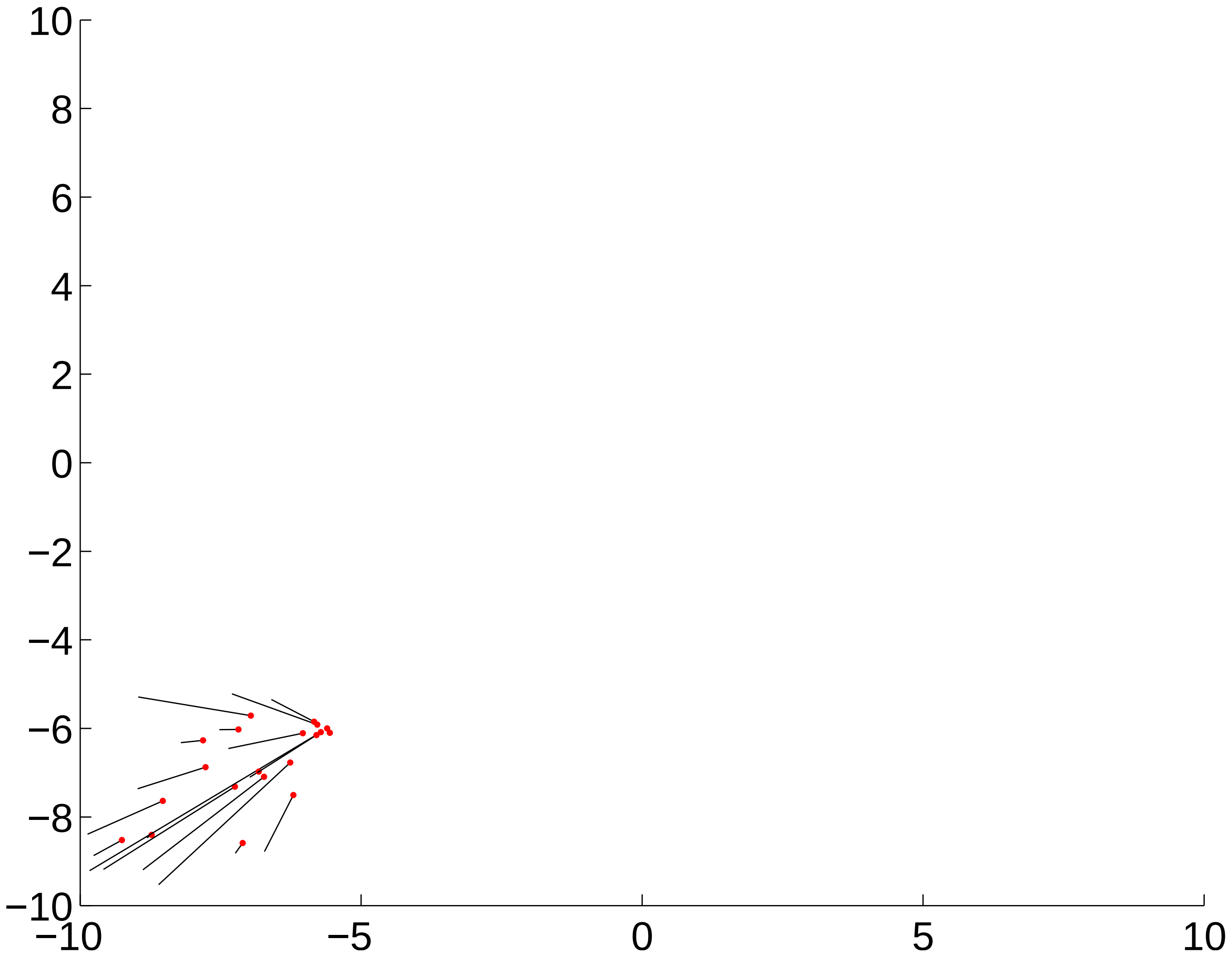
- ✔ $|\mathcal{B}| > 1$, $|\mathcal{P}| = |\mathcal{B}|$, $|\mathcal{C}| = |\mathcal{B}|$
- ✔ Neighb. type: PDF
- ✔ Neighb. adaptativity: population, model

[EK95] Russel C. Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, pages 39–43, Nagoya, Japan, 1995.

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

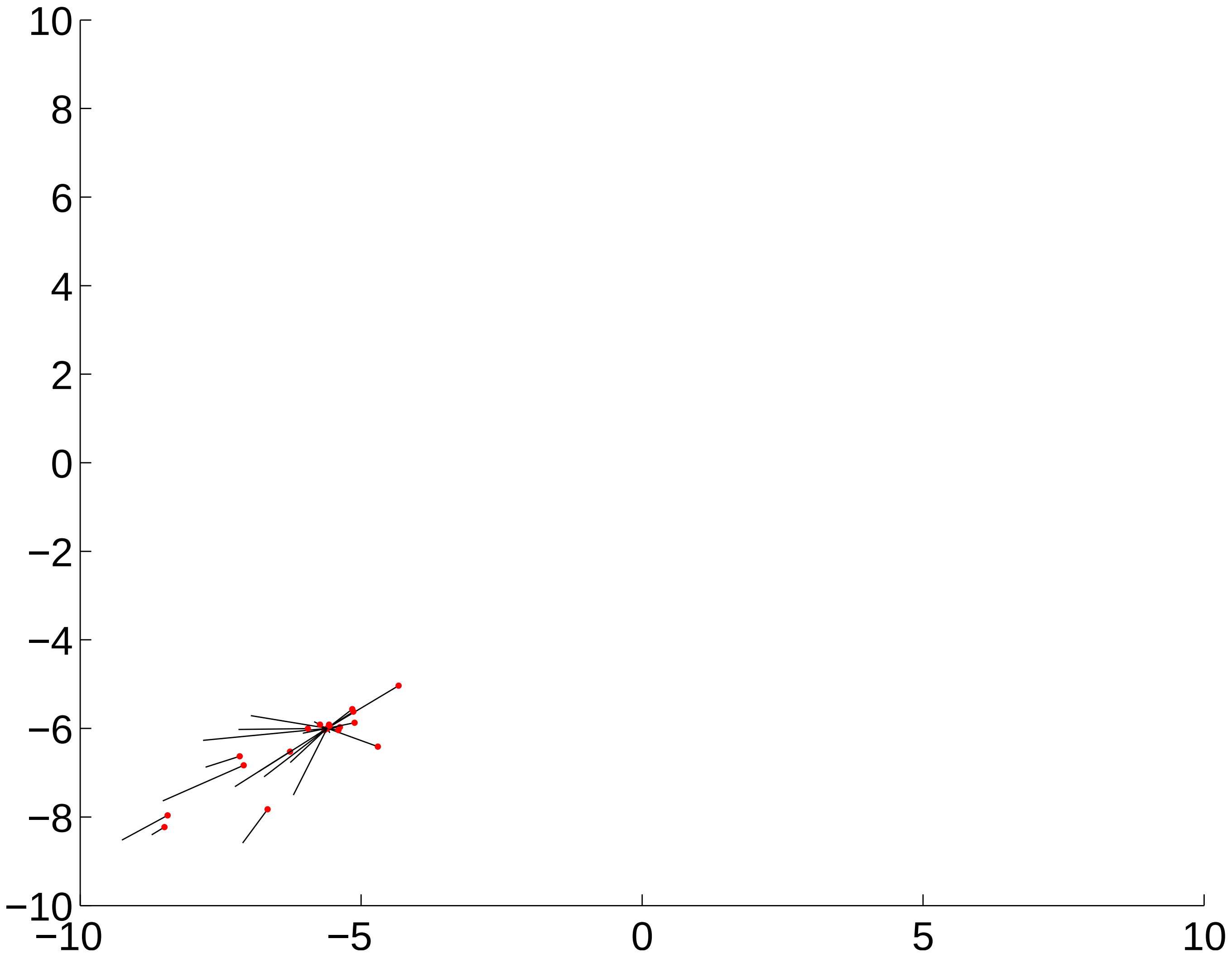
Particle Swarm Optimization Demo

PSO on the sphere function:



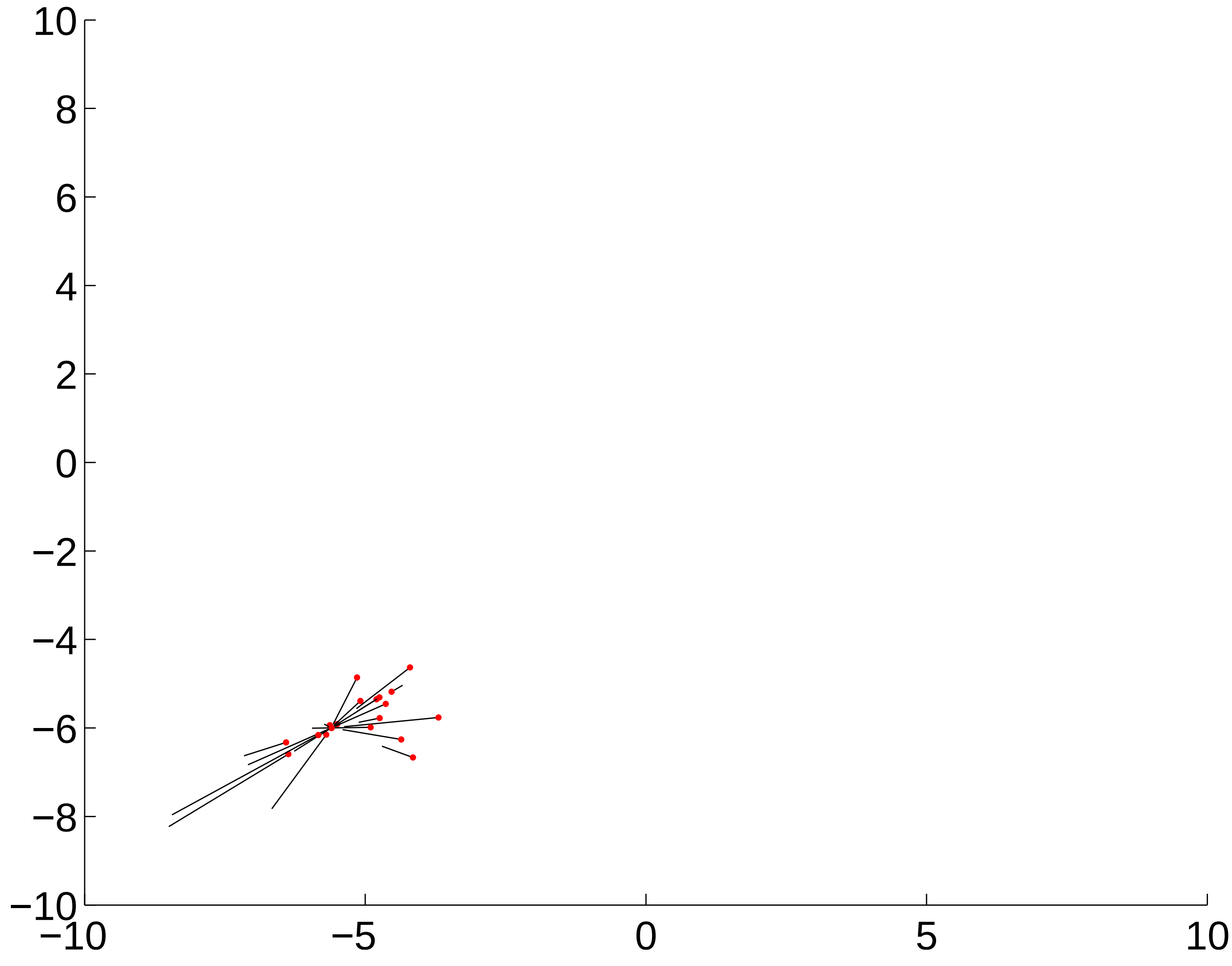
Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

PSO on the sphere function:



Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

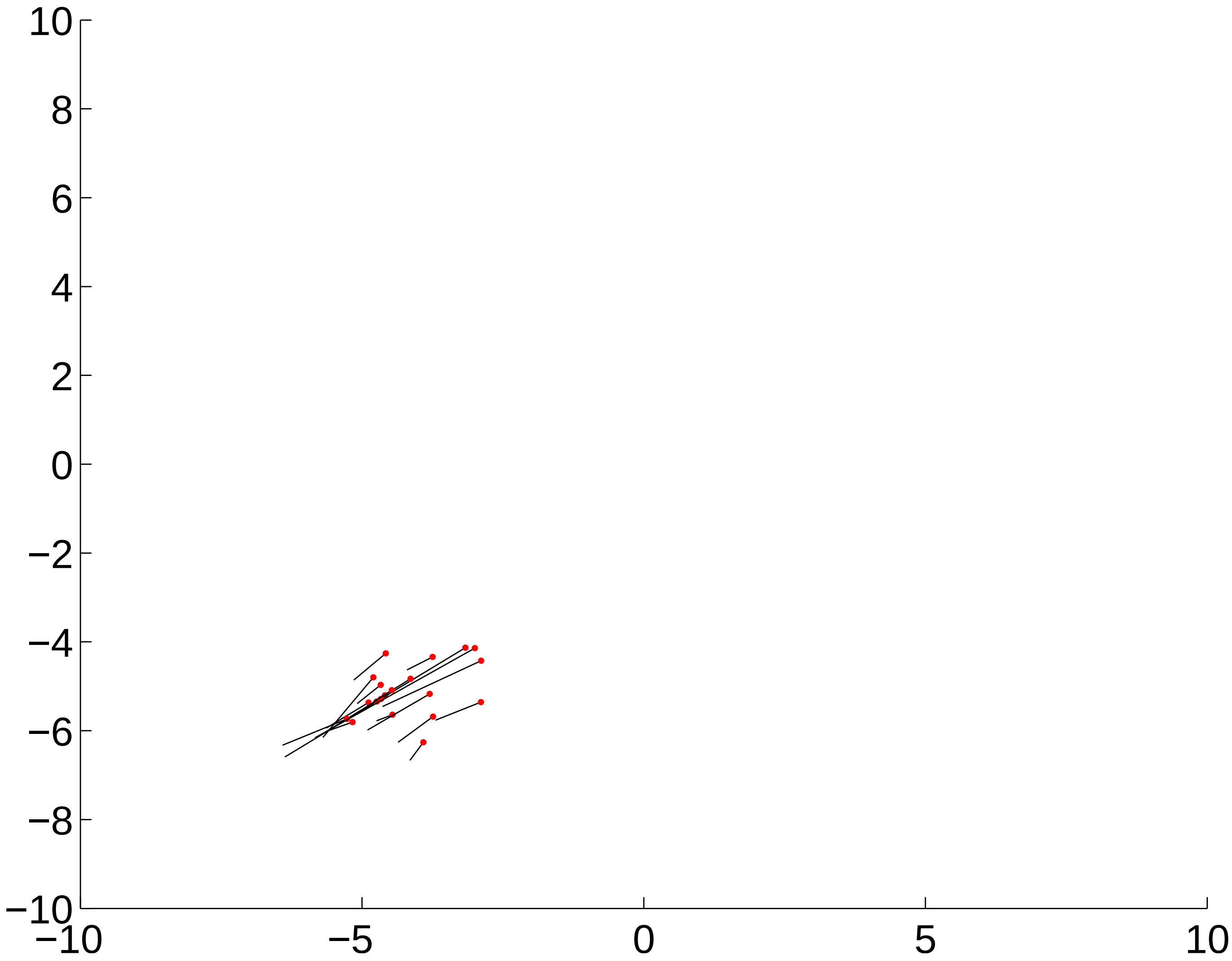
PSO on the sphere function:



Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

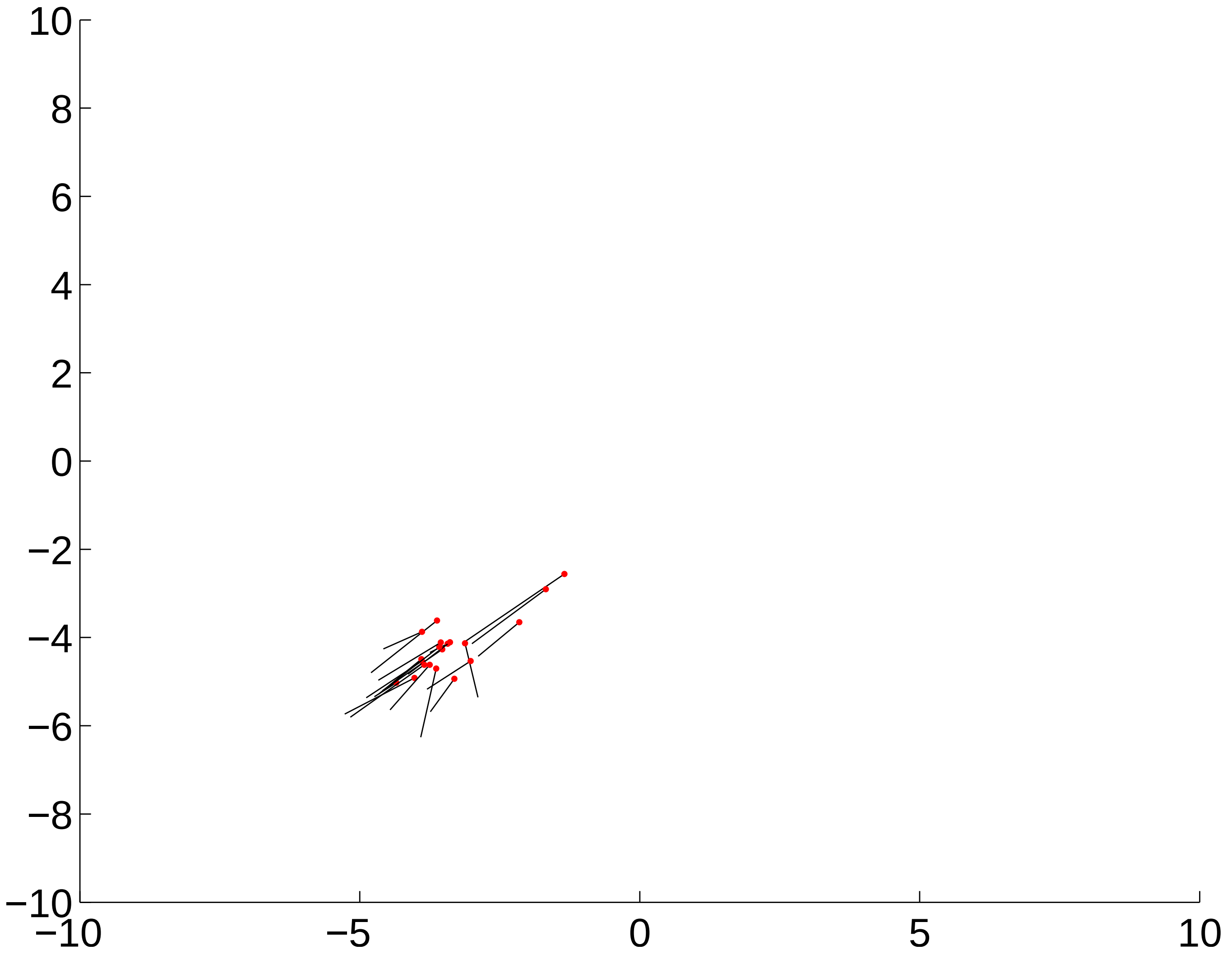
Particle Swarm Optimization Demo

PSO on the sphere function:



Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

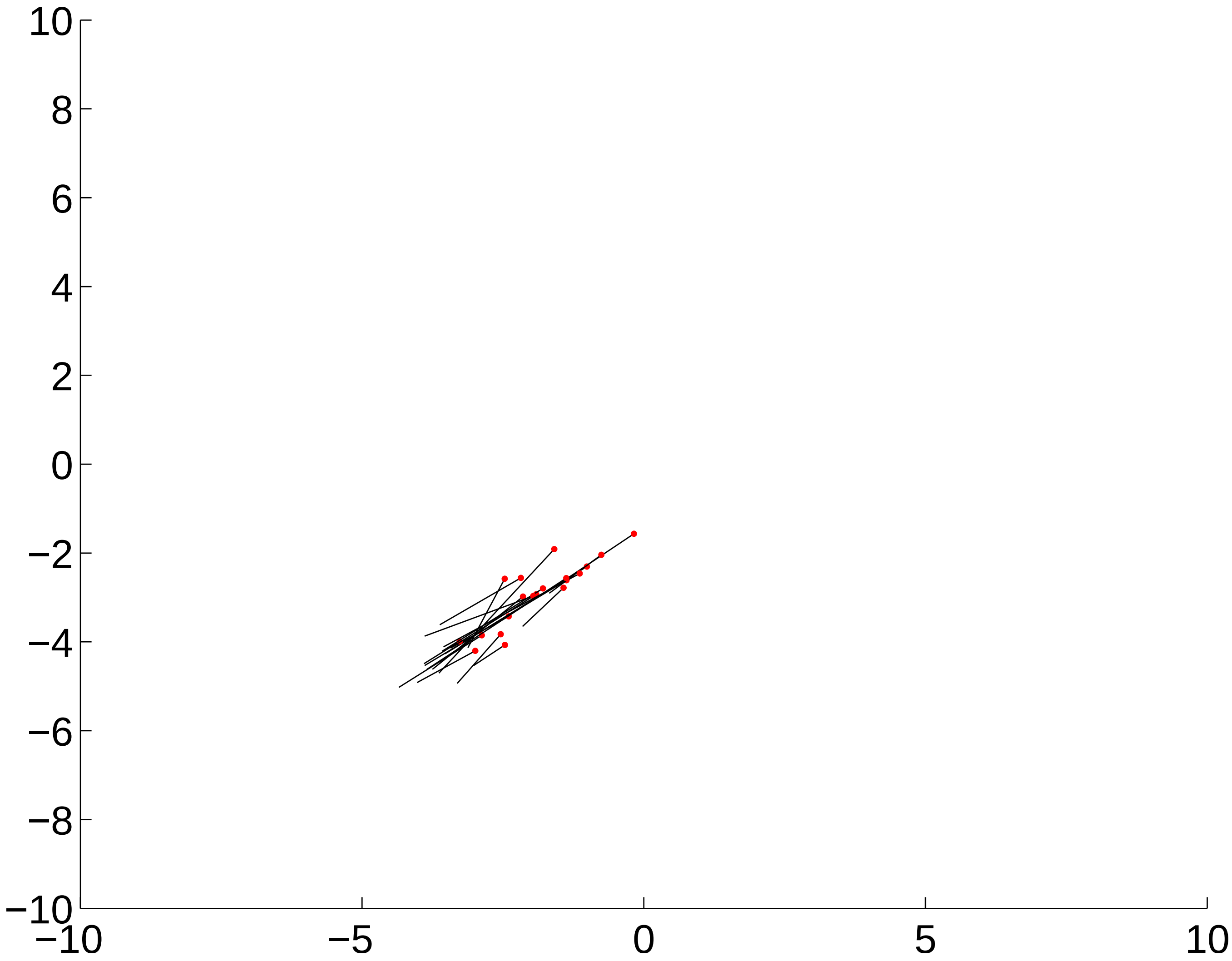
PSO on the sphere function:



Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

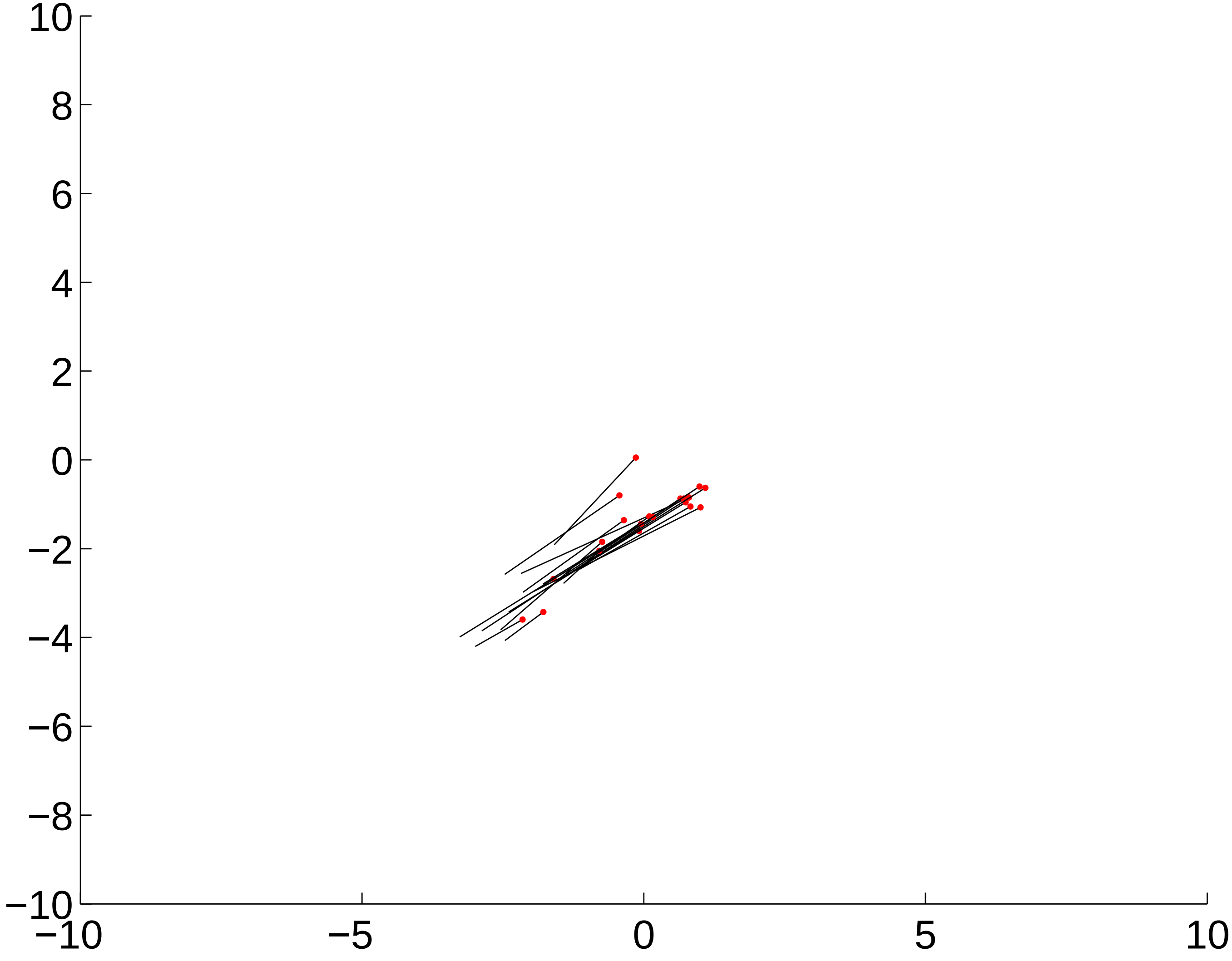
Particle Swarm Optimization Demo

PSO on the sphere function:



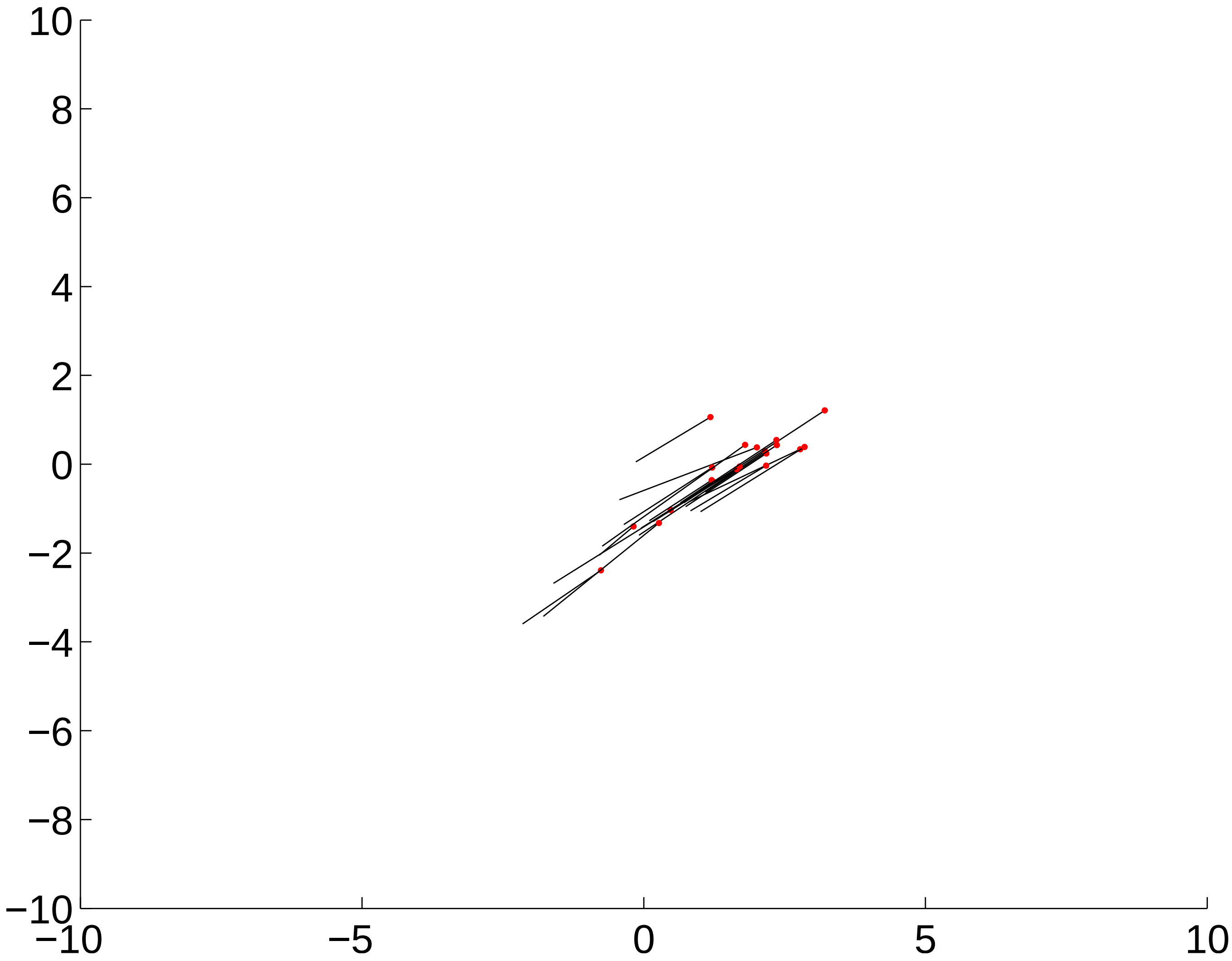
Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

PSO on the sphere function:



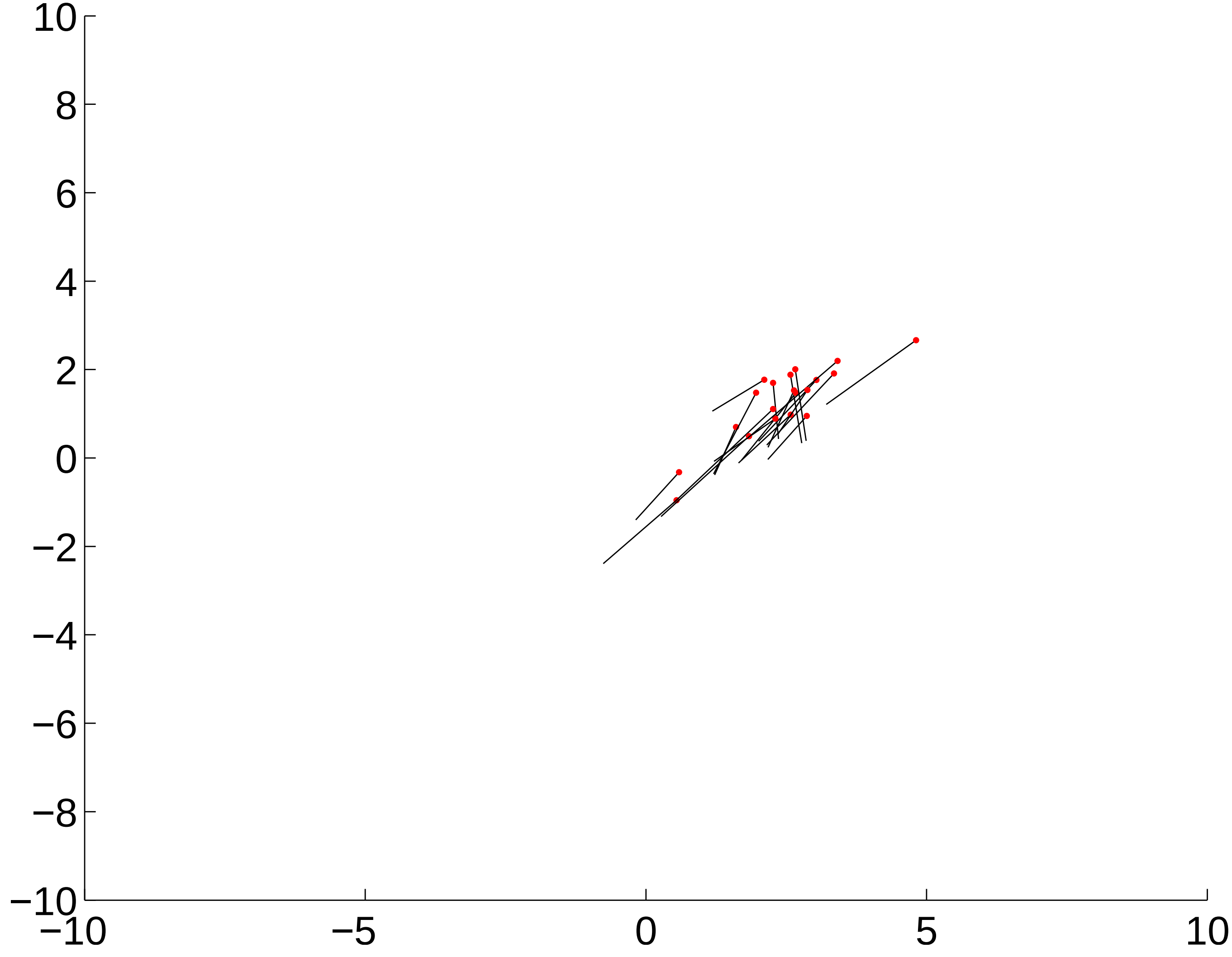
Particle Swarm Optimization Demo

PSO on the sphere function:



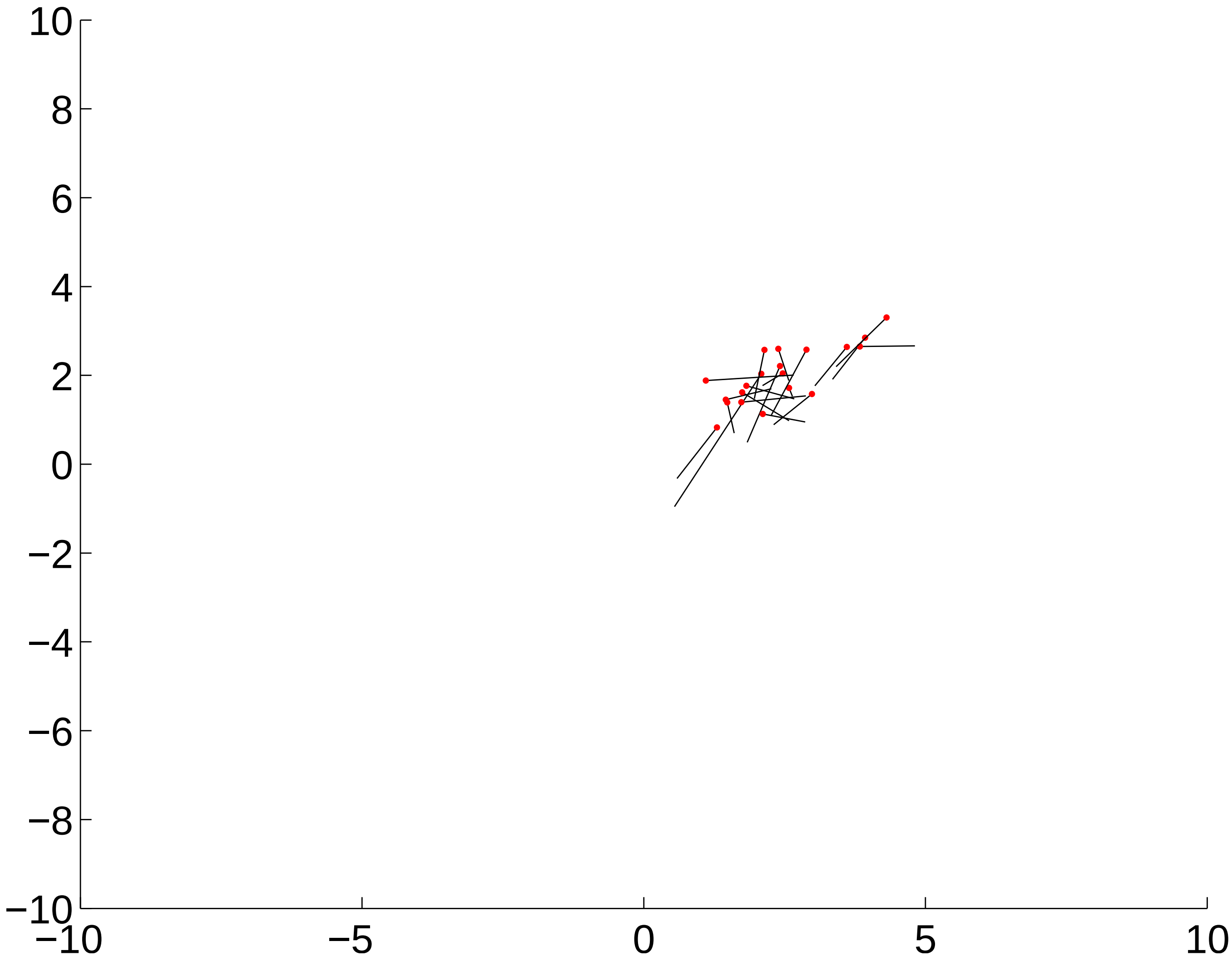
Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

PSO on the sphere function:



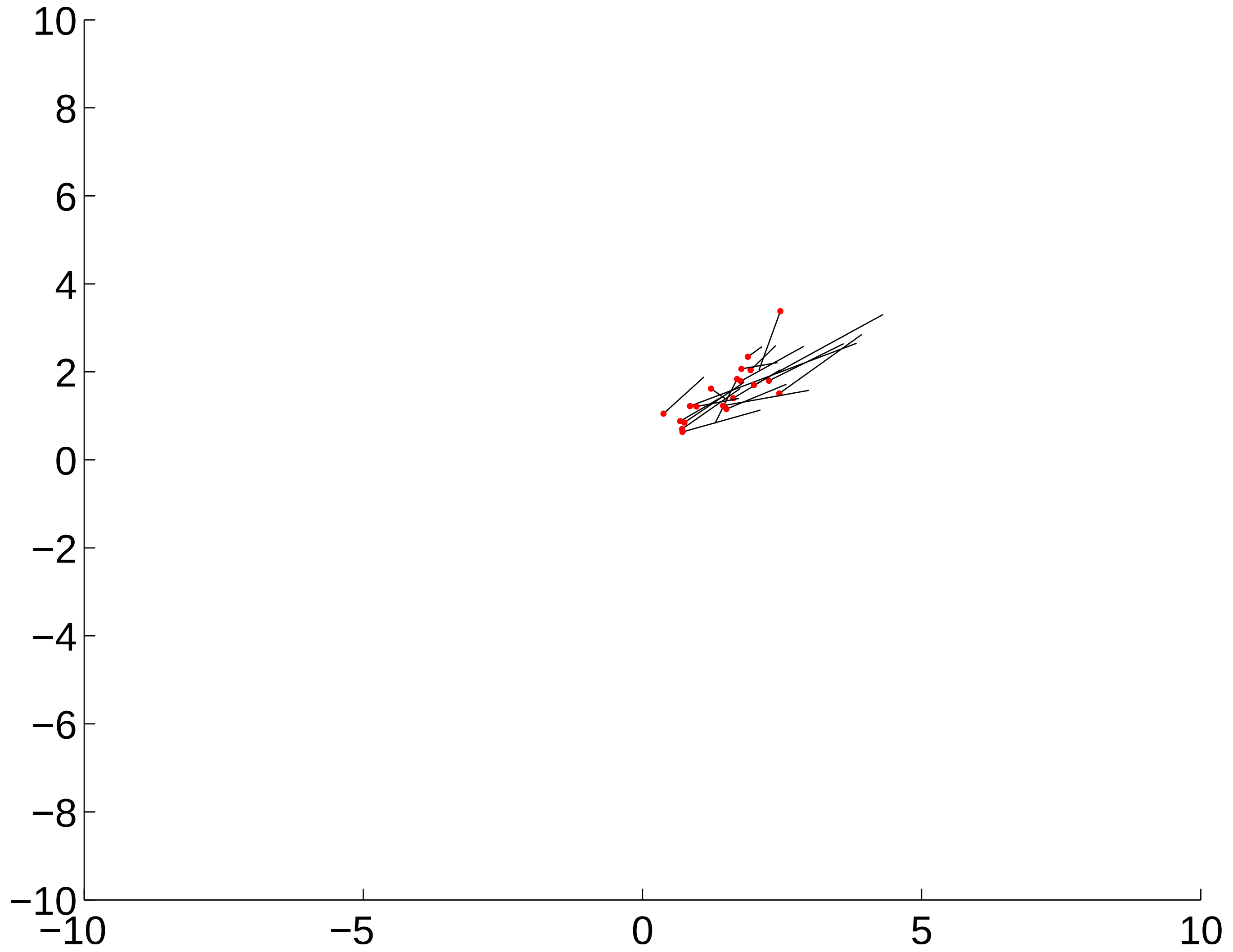
Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

PSO on the sphere function:



Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

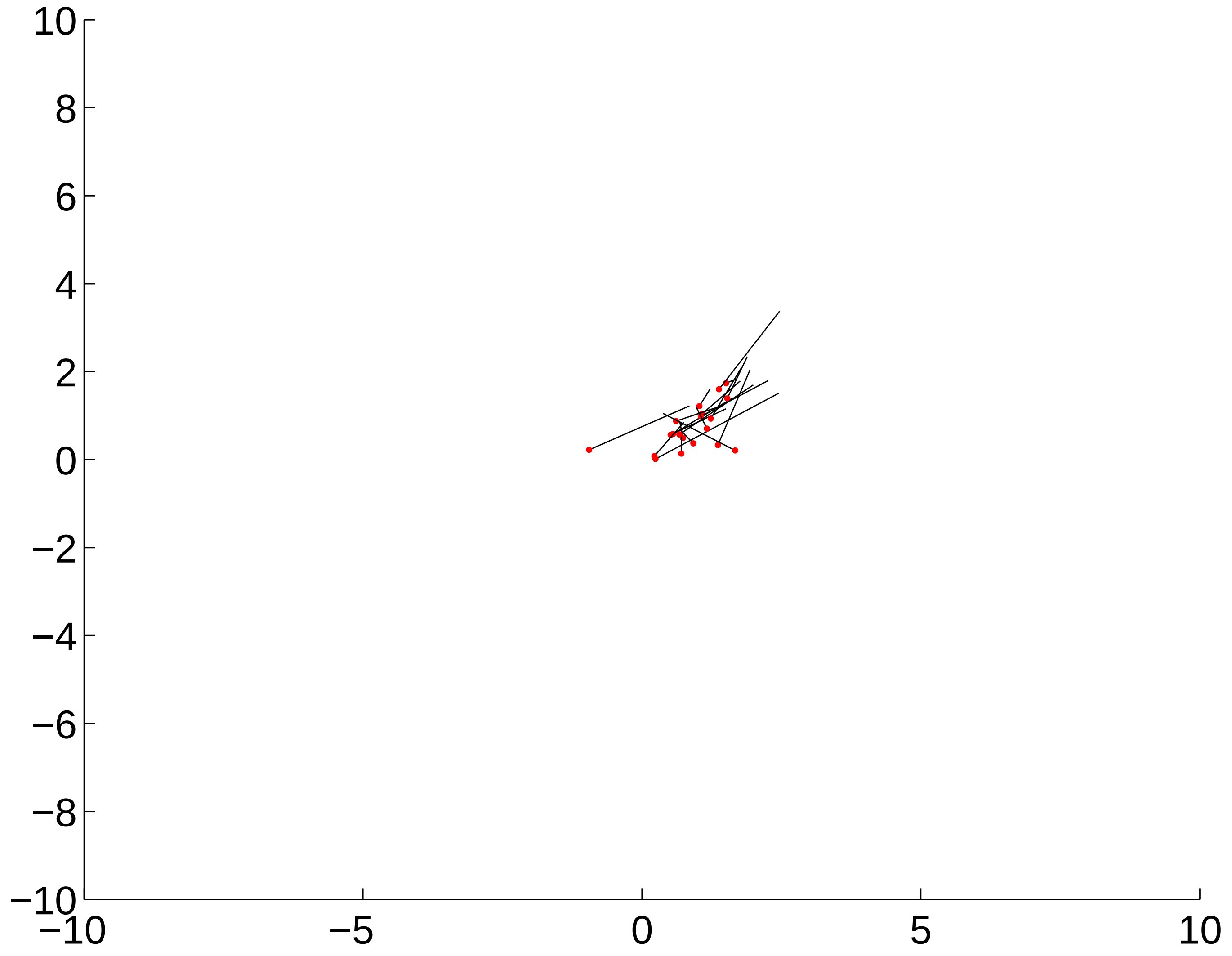
PSO on the sphere function:



Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

Particle Swarm Optimization Demo

PSO on the sphere function:



Evolution Strategy with Covariance Matrix Adaptation

Derandomized (m, λ)-ES [HO01]:

Initialize model:

- ✔ center μ, covariance matrix C and the global step size σ of the Gaussian

Select $\mathcal{P}$:

- ✔ select m best members of $\mathcal{B}$

Model update:

- ✔ Adapt μ and C using maximum likelihood estimation

$$\mu^{t+1} \leftarrow \bar{\mathbf{x}}^t_{\text{sel}}, \quad \mathbf{x}_{\text{sel}} \in \mathcal{P}$$

$$C^{t+1} \leftarrow \text{Cov}\left(\frac{\mathbf{x}^t_{\text{sel}} - \mu^t}{\sigma^t}\right)$$

- ✔ Adapt the global step size σ so that two consecutive steps, $\mu^t \rightarrow \mu^{t+1}$ and $\mu^{t+1} \rightarrow \mu^{t+2}$, are conjugated, i.e. conceptually

$$\left(\mu^{t+2} - \mu^{t+1}\right) \times C^{-1} \times \left(\frac{\mu^{t+1} - \mu^t}{(\sigma^{t+1})^2}\right) = 0$$

Generate $\mathcal{C}$ (λ candidates):

$$\mathbf{y}_i = \mu + \mathbf{z}_i, \quad \mathbf{z}_i \sim \mathcal{N}(0, \sigma \cdot C)$$

Population update:

- ✔ set $\mathcal{B}_{new}$ to $\mathcal{C}$

Summary in ODF:

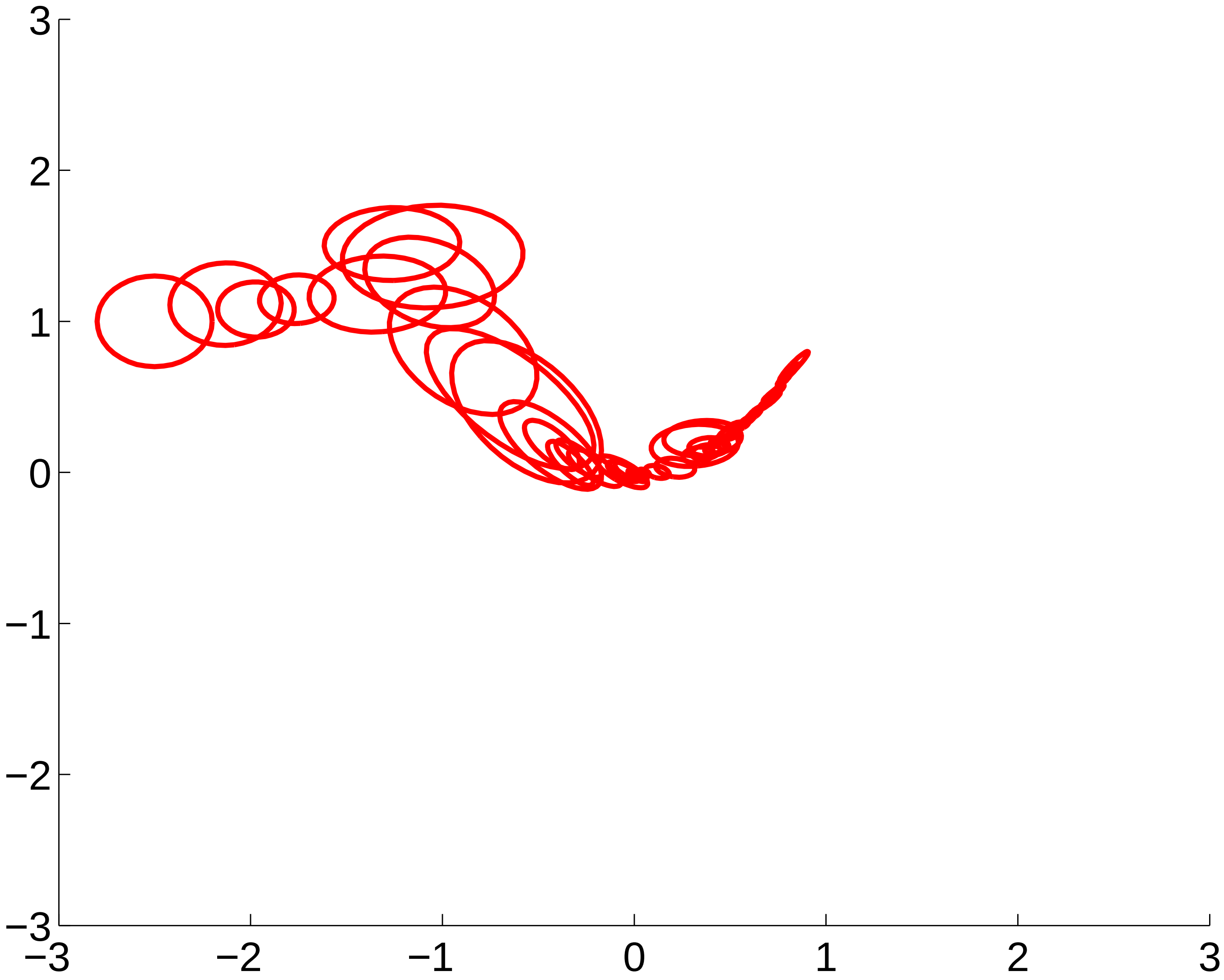
- ✔ $|\mathcal{B}| = \lambda > 1$, $|\mathcal{P}| = m$, $|\mathcal{C}| = |\mathcal{B}|$
- ✔ Neighb. type: PDF
- ✔ Neighb. adaptativity: population, model

DEMO

[HO01] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Evolution Strategies
Differential Evolution
Particle Swarm Optimization
Particle Swarm Optimization Demo
Evolution Strategy with Covariance Matrix Adaptation
CMA-ES Demo
G3+mPCX
mPCX Operator Demo
Algorithm Comparison
Summary and Conclusions

CMA-ES on the Rosenbrock function:



G3+mPCX

Generalized Generation Gap [DAJ02]:

Select $\mathcal{P}$:

- ✔ select the best member of $\mathcal{B}$ and $\mu - 1$ members randomly

Model update: none

Generate $\mathcal{C}$:

- ✔ Create λ candidates by mPCX operator

Population update:

- ✔ Choose r members of $\mathcal{B}$, create $\mathcal{R}$.
- ✔ Replace $\mathcal{R}$ with the r best sol. of $\mathcal{R} \cup \mathcal{C}$

Summary in ODF:

- ✔ $|\mathcal{B}| \approx 100$, $|\mathcal{P}| = \mu \approx 3$, $|\mathcal{C}| = \lambda \approx 2-6$
- ✔ Neighb. type: PDF
- ✔ Neighb. adaptativity: population

[DAJ02] Kalyanmoy Deb, Ashish Anand, and Dhiraj Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10(4):371–395, 2002.

Modified Parent Centric Xover (mPCX) [Deb05]:

Algorithm 5: Modified Parent Centric Crossover

Input: μ D-dimensional parents
Output: λ D-dimensional offspring

begin
- Mean of the parents: $\mathbf{g} \leftarrow \frac{1}{\mu}\sum_{i=1}^{\mu} \mathbf{x}_i$
- Direction vectors: $\mathbf{d}_i \leftarrow \mathbf{x}_i - \mathbf{g}$
- Base vectors: $\mathbf{e}_i \leftarrow \mathbf{d}_i / |\mathbf{d_i}|$
- **foreach** *offspring* $\mathbf{y}$ **do**
  - Choose 1 parent $\mathbf{x}_p$ randomly
  - Perpendicular distances D_i of the other $\mu - 1$ parents, $i \neq p$, to the line $\mathbf{g} -> \mathbf{x}_p$
  - Average distance: $\bar{D} = \frac{1}{\mu-1}\sum_{i=1, i\neq p}^{\mu} D_i$
  - Compute offspring as $\mathbf{y} \leftarrow \mathbf{x}_p + w_1 \mathbf{d}_p + \sum_{i=1, i\neq p}^{\mu} w_2 \bar{D} \mathbf{e}_i$ where $w_1 \sim \mathcal{N}(0, \sigma_1)$ and $w_1 \sim \mathcal{N}(0, \sigma_2)$

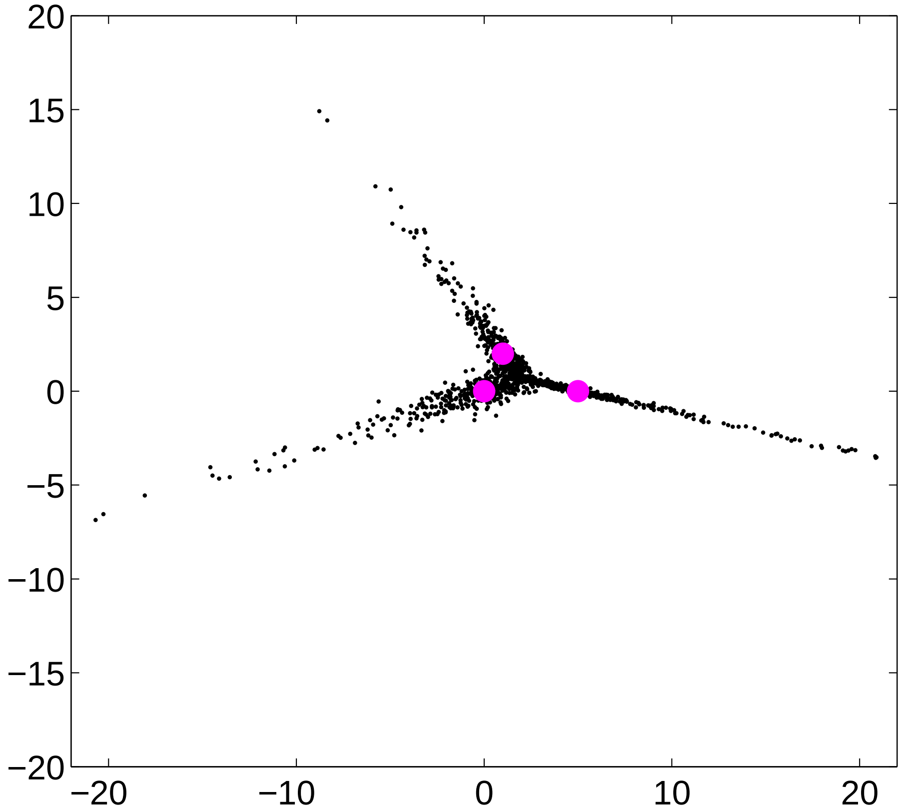
end

DEMO

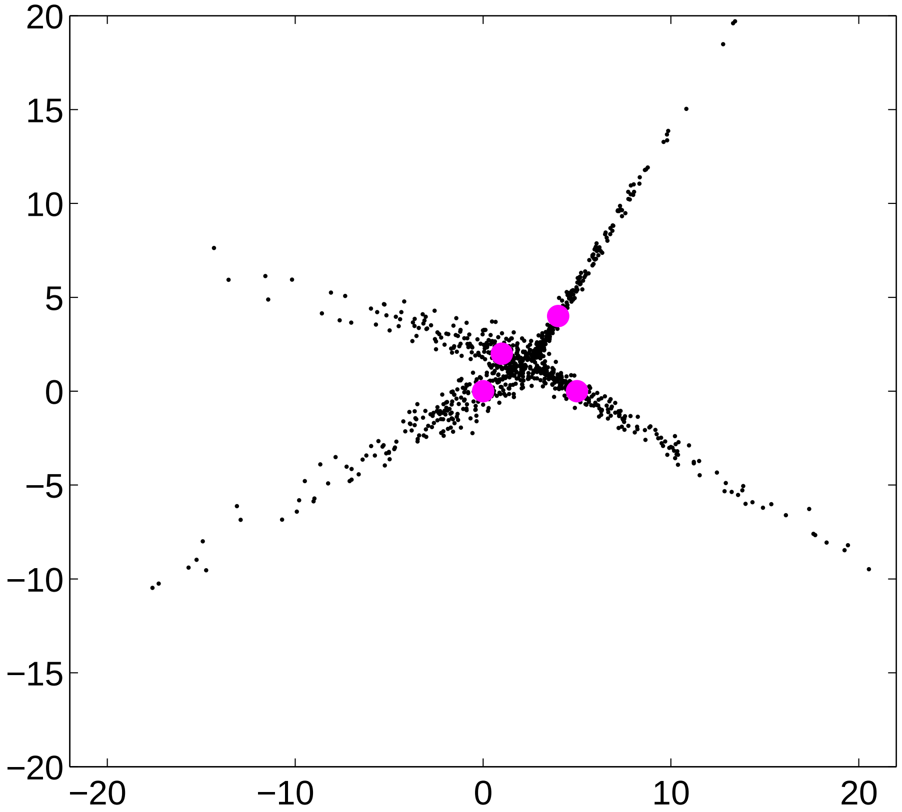
[Deb05] Kalyanmoy Deb. A population-based algorithm-generator for real-parameter optimization. *Soft Computing*, 9:236–253, 2005.

mPCX Operator Demo

Distribution of offspring when 3 2D points are used with mPCX over and over:



Distribution of offspring when 4 2D points are used with mPCX over and over:



Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
IEEE Congress on Evolutionary Computation 2005
So, what is the best algorithm?
Summary and Conclusions

Algorithm Comparison

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
IEEE Congress on Evolutionary Computation 2005
So, what is the best algorithm?
Summary and Conclusions

Special Session on Unconstrained Real-parameter Optimization:

Comparison of 11 particular representants of

- ✔ CMA-ES
- ✔ DE
- ✔ PSO
- ✔ EDA
- ✔ Co-evolution
- ✔ BLX-α operator

on 25 reference 10- and 30-dimensional optimization tasks [SHL+05].

Results comparison: thanks to Nikolaus Hansen! [Han05]

[Han05] Nikolaus Hansen. Compilation of results on the 2005 cec benchmark function set, 2005. http://www.bionik.tu-berlin.de/user/niko/cec2005compareresults.pdf.

[SHL+05] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical report, Nanyang Technological University, Singapore, May 2005. http://www.ntu.edu.sg/home/epnsugan.

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
IEEE Congress on Evolutionary Computation 2005
So, what is the best algorithm?
Summary and Conclusions

So, what is the best algorithm?

"Hmmm, it depends. . . "

Important note: Initialization

- around the true optimum (favors mean-centric algos), or
- far away form the true optimum.

Another view of the candidate creation phase:

- parent-centric
  - (LS, Box, Rosenbrock)
  - Nelder-Mead, ES, DE, PSO, G3+mPCX
- mean-centric
  - CMA-ES
  - Not discussed here: EDA, e.g. SDR-AVS-IDEA [BGR07]

Observation: those promising techniques (in red color) *combine information from more than 2 parents* when creating offspring.

[BGR07] Peter A. N. Bosman, Jörn Grahl, and Franz Rothlauf. SDR: A better trigger for adaptive variance scaling in normal EDAs. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and Evolutionary Computation*, pages 492–499, New York, NY, USA, 2007. ACM Press.

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions
Summary and Conclusions
Thank you for your attention

Summary and Conclusions

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions
Summary and Conclusions
Thank you for your attention

Summary and Conclusions

Summary:

- ✔ Survey of several "historical" optimization strategies and their important features
- ✔ One of possible taxonomies of the optimization algorithms
- ✔ Survey of several "new" nature-inspired techniques
- ✔ Recent comparison results

Conclusions:

- ✔ There is NO BEST ALGORITHM FOR ALL PROBLEMS...
  - ✘ ... but a variant of CMA-ES should be the first try.
- ✔ In real spaces it seems better to select algorithm which combines more than 2 parents to create offspring.

Not covered here:

- ✔ constraints
- ✔ multiobjective optimization

Thank you for your attention

Introduction
Optimizer Description Framework
Nature Inspired Optimization Techniques
Algorithm Comparison
Summary and Conclusions
Summary and Conclusions
Thank you for your attention

Any questions?