Differential Evolution and Particle Swarm Optimization

Luong Ngoc Hoang

University of Information Technology (UIT)
Vietnam National University - Ho Chi Minh City

3 April 2020

Overview

1. Motivation
2. Differential Evolution (DE)
3. Particle Swarm Optimization (PSO)
4. Research Questions

Simple Genetic Algorithm (sGA)

Can the sGA solve the following problem?

$$\text{Minimize} \quad f(x_0, x_1) = (x_0 - 0.5)^2 + (x_1 - 0.5)^2$$
$$x_0, x_1 \in [0, 1] \subset \mathbb{R}$$

Simple Genetic Algorithm (sGA)

Minimize $\quad f(x_0, x_1) = (x_0 - 0.5)^2 + (x_1 - 0.5)^2$

x_0	x_1
0.25	0.80
0.40	0.97

Simple Genetic Algorithm (sGA)

Minimize $\quad f(x_0, x_1) = (x_0 - 0.5)^2 + (x_1 - 0.5)^2$

x_0	x_1
0.25	0.80
0.40	0.97

$$f(0.40, 0.80) = 0.1$$

Simple Genetic Algorithm (sGA)

Minimize $\quad f(x_0, x_1) = (x_0 - 0.5)^2 + (x_1 - 0.5)^2$

x_0	x_1
0.25	0.80
0.40	0.97
0.12	0.34
0.67	0.72

Simple Genetic Algorithm (sGA)

Minimize $\quad f(x_0, x_1) = (x_0 - 0.5)^2 + (x_1 - 0.5)^2$

x_0	x_1
0.25	0.80
0.40	0.97
0.12	0.34
0.67	0.72

$f(0.40, 0.34) = 0.0356$

Simple Genetic Algorithm (sGA)

Minimize $\quad f(x_0, x_1) = (x_0 - 0.5)^2 + (x_1 - 0.5)^2$

x_0	x_1
0.25	0.80
0.40	0.97
0.12	0.34
0.67	0.72
0.55	0.81
0.52	0.04
0.85	0.23
0.76	0.65

$$f(0.52, 0.65) = 0.0229$$

Simple Genetic Algorithm (sGA)

Can sGA be used for real-valued optimization?

Simple Genetic Algorithm (sGA) - Simple Binary Encoding

A binary string (a_i) of q bits

$$(a_i), \quad a_i \in \{0, 1\}, \quad i = 0, 1, \ldots, q-1$$

can encode a real-valued variable $x \in [b_L, b_U]$, and x can be decoded by:

$$x = b_L + \frac{b_U - b_L}{2^q - 1} \cdot \sum_{i=0}^{q-1} a_i 2^i$$

Simple Genetic Algorithm (sGA) - Simple Binary Encoding

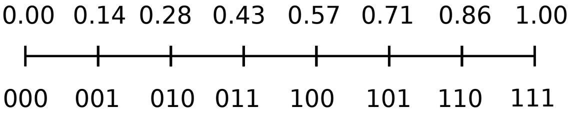
A binary string (a_i) of q bits

$$(a_i), \quad a_i \in \{0,1\}, \quad i = 0, 1, \ldots, q-1$$

can encode a real-valued variable $x \in [b_L, b_U]$, and x can be decoded by:

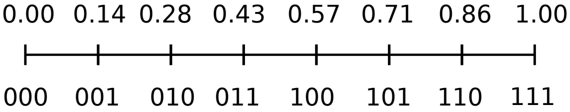
$$x = b_L + \frac{b_U - b_L}{2^q - 1} \cdot \sum_{i=0}^{q-1} a_i 2^i$$

Example: Using $q = 3$ bits to encode a real-valued variable $x \in [0, 1]$



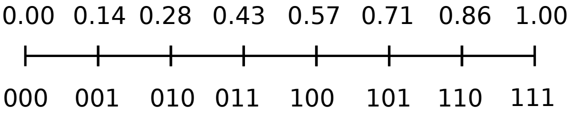
Simple Genetic Algorithm (sGA) - Simple Binary Encoding

Problems with this binary encoding?



Simple Genetic Algorithm (sGA) - Simple Binary Encoding

Problems with this binary encoding?



- The algorithm must handle q times more variables.
- Values close to each other can have different representation, e.g., 0.43 (011) and 0.57 (100).
  $\rightarrow$ The function mapping the bit string into real-valued number is multi-modal.
- ...

DIFFERENTIAL EVOLUTION

Differential Evolution - Introduction

- Developed by Storn and Price in 1995.
- Core concept: Variation by adding the scaled difference between two randomly selected individuals to a third randomly selected individual.

DE - Notations

Population $P_{x,g}$ at generation g consists of N vectors $\mathbf{x}_{i,g}$

$$\begin{aligned} P_{x,g} &= (\mathbf{x}_{i,g}), \quad i = 0, 1, \ldots, N-1, \quad g = 0, 1, \ldots, g_{max} \\ \mathbf{x}_{i,g} &= (x_{j,i,g}), \quad j = 0, 1, \ldots, D-1 \\ x_{j,i,g} &\in [b_{j,L}, b_{j,U}] \subset \mathbb{R} \end{aligned}$$

N: the population size
D: the number of real-valued parameters
g_{max}: the maximum number of generations
$[b_{j,L}, b_{j,U}]$: the range of the j-th parameter

DE - Initialization

$$
\begin{aligned}
g &= 0 \\
P_{x,0} &= (\mathbf{x}_{i,0}) \\
x_{j,i,0} &= rand_j(0,1) \times (b_{j,U} - b_{j,L}) + b_{j,L} \\
0 \leq\ & rand_j(0,1) \leq 1
\end{aligned}
$$

DE - Variation - Creating Mutant Vector

Mutant population $P_{v,g}$ contains N mutant vectors $\mathbf{v}_{i,g}$

$$
\begin{aligned}
P_{v,g} &= (\mathbf{v}_{i,g}), \quad i = 0, 1, \ldots, N-1, \quad g = 0, 1, \ldots, g_{max} \\
\mathbf{v}_{i,g} &= (v_{j,i,g}), \quad j = 0, 1, \ldots, D-1
\end{aligned}
$$

F: the scale factor, $F \in \mathbb{R}_{>0}$, and mostly $F \in (0, 1)$

For each $\mathbf{v}_{i,g}$:
$r0 \neq r1 \neq r2$: randomly selected from $\{0, 1, \ldots, N-1\} \setminus \{i\}$
$r0$: base vector index
$r1, r2$: difference vector indices
$\mathbf{x}_{r0,g}, \mathbf{x}_{r1,g}, \mathbf{x}_{r2,g} \in P_{x,g}$

$$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \times (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$$

DE - Variation - Creating Mutant Vector

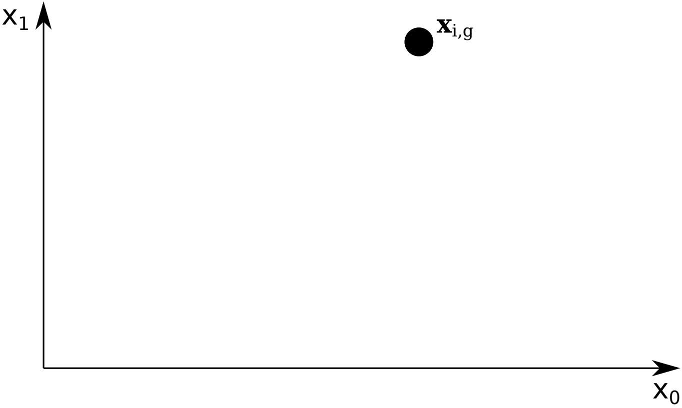


Figure: Consider each current vector $\mathbf{x}_{i,g}$.

DE - Variation - Creating Mutant Vector

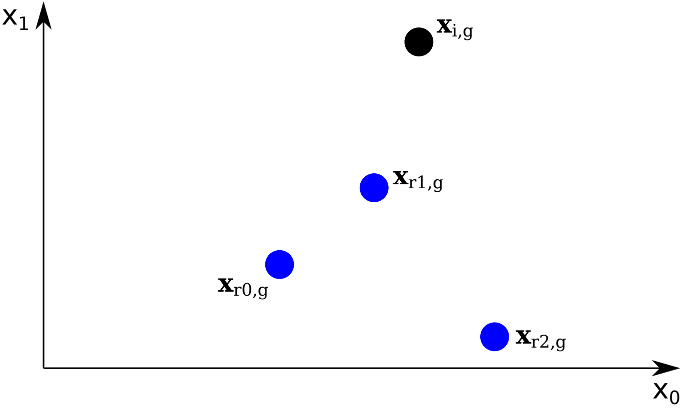


Figure: Randomly select 3 different vectors $\mathbf{x}_{r0,g}, \mathbf{x}_{r1,g}, \mathbf{x}_{r2,g}$.

DE - Variation - Creating Mutant Vector

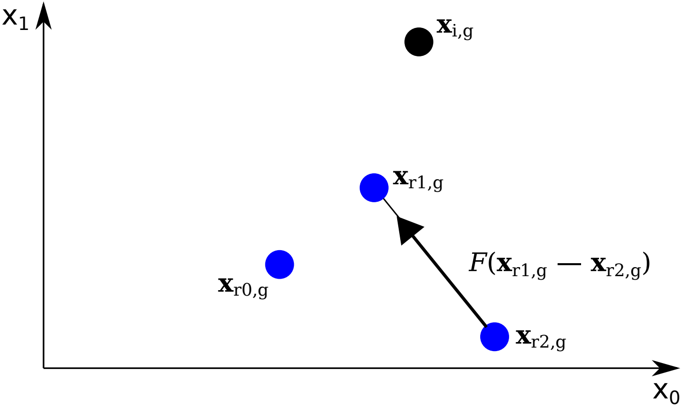


Figure: Compute the scaled difference vector $F(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$.

DE - Variation - Creating Mutant Vector

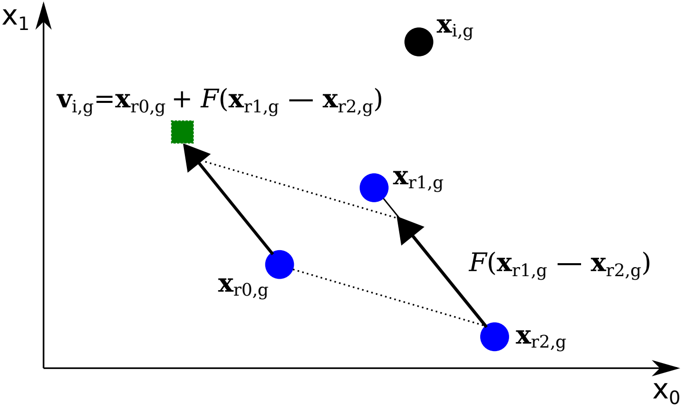


Figure: Create $\mathbf{v}_{i,g}$ by adding the scaled difference vector to the base vector $\mathbf{x}_{r0,g}$.

DE - Variation - Crossover

Trial population $P_{u,g}$ contains N trial vectors $\mathbf{u}_{i,g}$

$$P_{u,g} = (\mathbf{u}_{i,g}), \quad i = 0, 1, \ldots, N-1, \quad g = 0, 1, \ldots, g_{max}$$
$$\mathbf{u}_{i,g} = (u_{j,i,g}), \quad j = 0, 1, \ldots, D-1$$

Cr: the crossover probability, $Cr \in [0, 1]$

For each trial vector $\mathbf{u}_{i,g}$:
j_{rand}: an index randomly selected from $\{0, 1, \ldots, D-1\}$
$\mathbf{x}_{i,g}$: the target vector, $\mathbf{x}_{i,g} \in P_{x,g}$

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

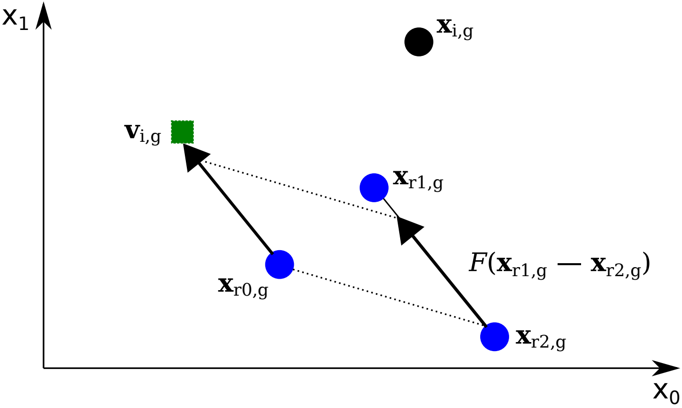


Figure: Perform crossover the current vector $\mathbf{x}_{i,g}$ with the mutant vector $\mathbf{v}_{i,g}$.

DE - Variation - Crossover

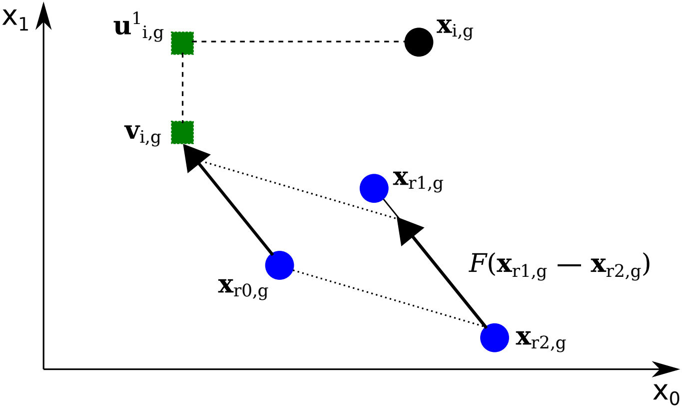


Figure: $\mathbf{u}^1_{i,g}$: x_0 is from the mutant vector and x_1 is from the current vector.

DE - Variation - Crossover

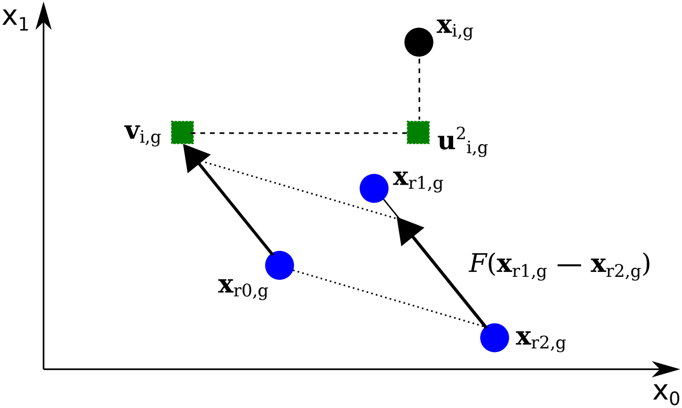


Figure: $\mathbf{u}^2_{i,g}$: x_0 is from the current vector and x_1 is from the mutant vector.

DE - Variation - Crossover

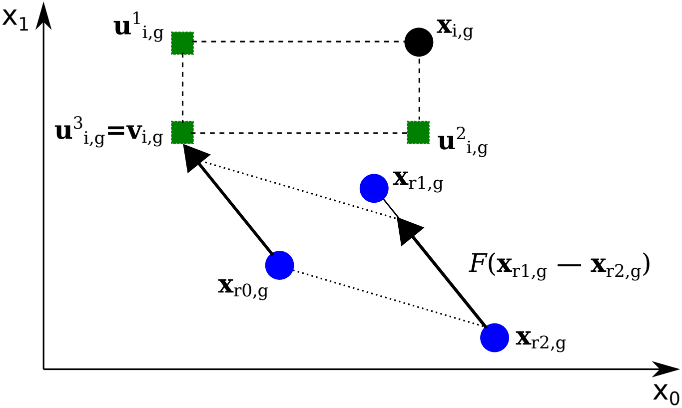


Figure: $\mathbf{u}_{i,g}^3$: both x_0 and x_1 are from the mutant vector, $\mathbf{u}_{i,g}^3 = \mathbf{v}_{i,g}$.

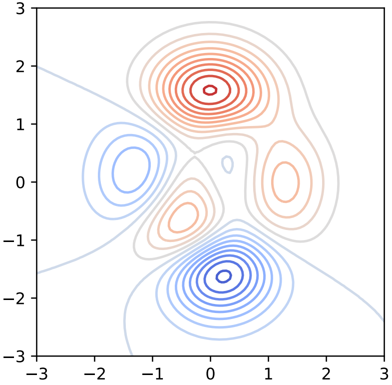
DE - Selection

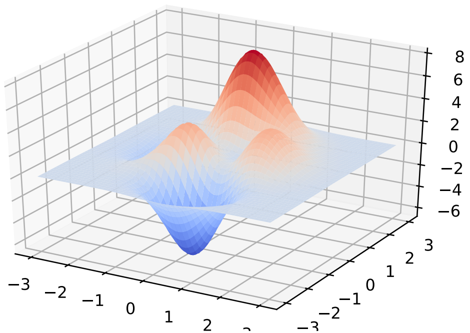
The next population $P_{x,g+1}$ consists of N vectors $\mathbf{x}_{i,g+1}$

$$P_{x,g+1} = (\mathbf{x}_{i,g+1}), \quad i = 0, 1, \ldots, N-1, \quad g = 0, 1, \ldots, g_{max}$$

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & \text{if } f(\mathbf{u}_{i,g}) \text{ is better than } f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases}$$

DE - Example





$$f(x,y) = 3(1-x)^2 \exp(-x^2-(y+1)^2)$$
$$-10\left(\frac{x}{5}-x^3-y^5\right)\exp(-x^2-y^2)-\frac{1}{3}\exp(-(x+1)^2-y^2)$$

DE - Visualization

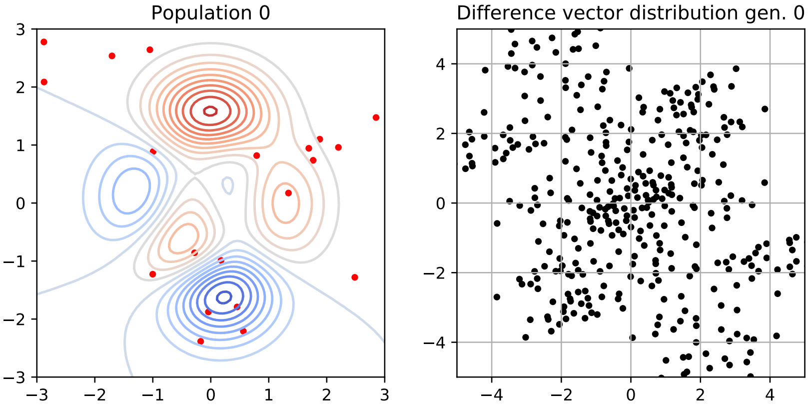


Figure: Generation 0

DE - Visualization

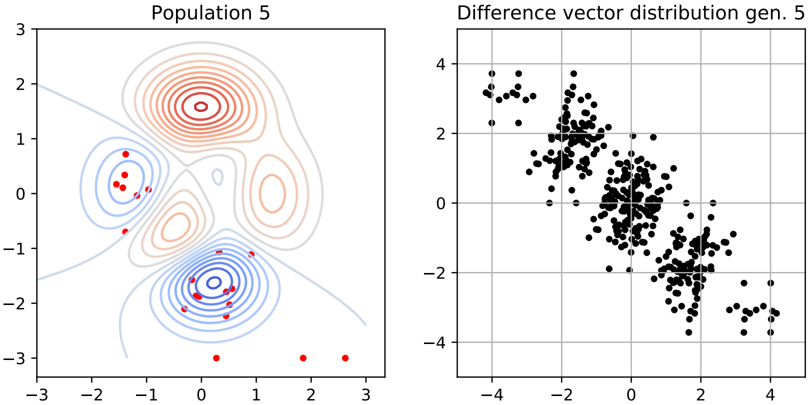


Figure: Generation 5

DE - Visualization

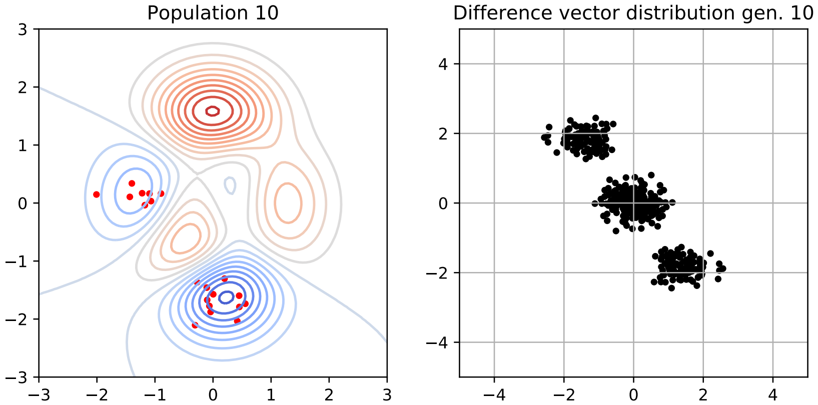


Figure: Generation 10

DE - Visualization

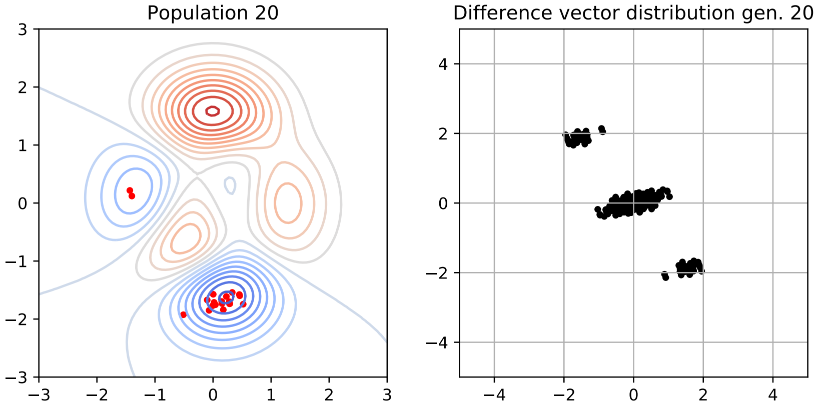


Figure: Generation 20

DE - Visualization

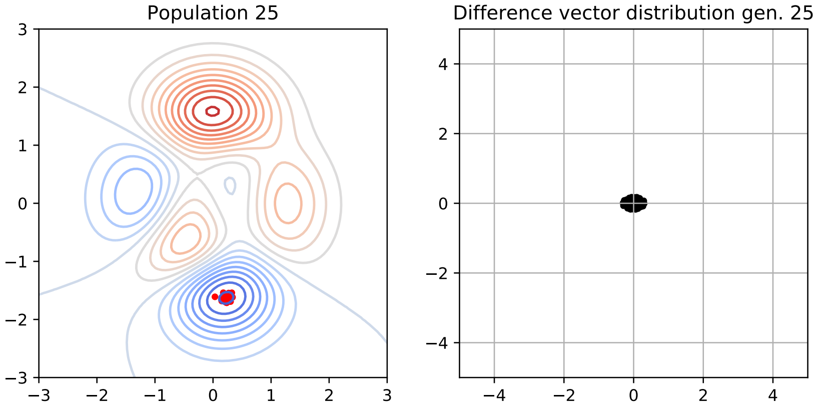


Figure: Generation 25

DE - Visualization

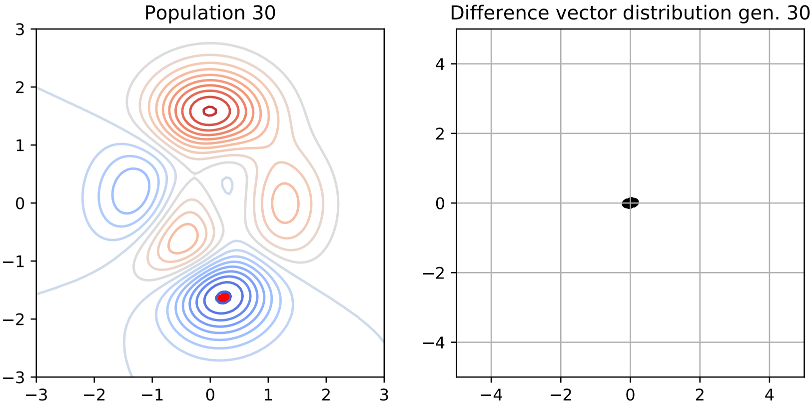


Figure: Generation 30

DE - Control Parameters

The performance of DE is influenced by its control parameters.

- Population size.
- How base vectors $\mathbf{x}_{r0,g}$ and difference vectors $\mathbf{x}_{r1,g}, \mathbf{x}_{r2,g}$ are chosen.
- The scale factor F.
- How crossover is performed.
- ...

PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization - Introduction

- First simulations done by Kennedy and Eberhart in 1995.
- Core concept: PSO maintains a swarm (i.e., a population) of particles that move through the search space. Each particle determines its movement based on:
    - Its current position.
    - Its best previous position.
    - Its neighbors' best previous position.

PSO - Notations

Swarm $P_{x,g}$ at generation g consists of N particles $\mathbf{x}_{i,g}$

$$
\begin{aligned}
P_{x,g} &= (\mathbf{x}_{i,g}), \quad i = 0, 1, \ldots, N-1, \quad g = 0, 1, \ldots, g_{max} \\
\mathbf{x}_{i,g} &= (x_{j,i,g}), \quad j = 0, 1, \ldots, D-1 \\
x_{j,i,g} &\in [b_{j,L}, b_{j,U}] \subset \mathbb{R}
\end{aligned}
$$

N: the population size
D: the number of real-valued parameters
g_{max}: the maximum number of generations

Each particle i has:

- $\mathbf{x}_{i,g}$: the current position of particle i
- $\mathbf{v}_{i,g}$: the current velocity vector of particle i
- $\mathbf{y}_{i,g}$: the best position found so far by particle i
- $\mathbf{z}_{i,g}$: the best position found so far in the neighborhood $\mathcal{N}_i$

PSO - Neighborhood Topology - Ring

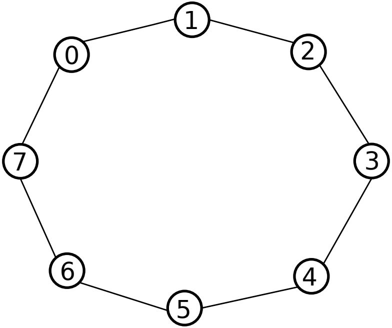


Figure: Ring topology. For example, $\mathcal{N}_3 = \{2, 3, 4\}, \mathcal{N}_0 = \{7, 0, 1\}, \ldots$

$\mathbf{z}_{i,g}$ is the best position found so far by the particles in its neighborhood (i.e., particle i and its adjacent particles).

PSO - Neighborhood Topology - Star

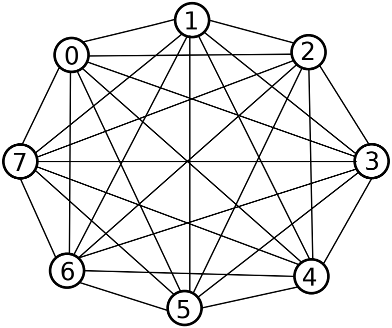


Figure: Star topology. For all i, $\mathcal{N}_i = \{0, 1, \ldots, 7\}$.

$\mathbf{z}_{i,g}$ is the best position found so far by the whole swarm.

PSO - Position Update

In each generation, the velocity vector of particle i is updated as:

$$\mathbf{v}_{i,g+1} = w\mathbf{v}_{i,g} + c_1\mathbf{r}_1 \otimes (\mathbf{y}_{i,g} - \mathbf{x}_{i,g}) + c_2\mathbf{r}_2 \otimes (\mathbf{z}_{i,g} - \mathbf{x}_{i,g})$$

w: inertia weight
c_1, c_2: acceleration constants
$\mathbf{r1}, \mathbf{r2}$: vectors of uniformly random values $\in (0, 1)$

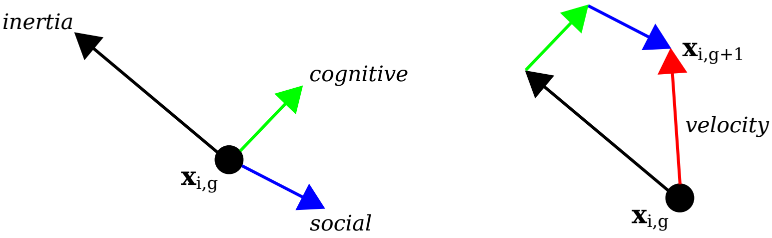
Velocity components

$w\mathbf{v}_{i,g}$: the inertia component.
$c_1\mathbf{r}_1 \otimes (\mathbf{y}_{i,g} - \mathbf{x}_{i,g})$: the cognitive component.
$c_2\mathbf{r}_2 \otimes (\mathbf{z}_{i,g} - \mathbf{x}_{i,g})$: the social component.

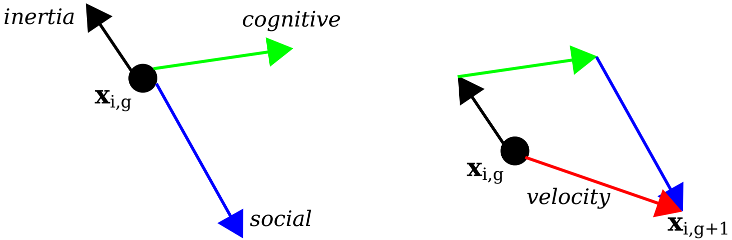
The next position of particle i is computed as:

$$\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g} + \mathbf{v}_{i,g+1}$$

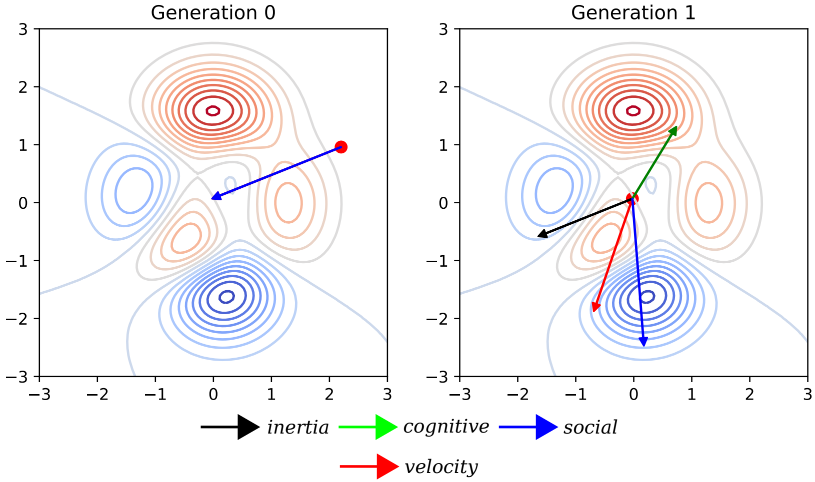
PSO - Position Update



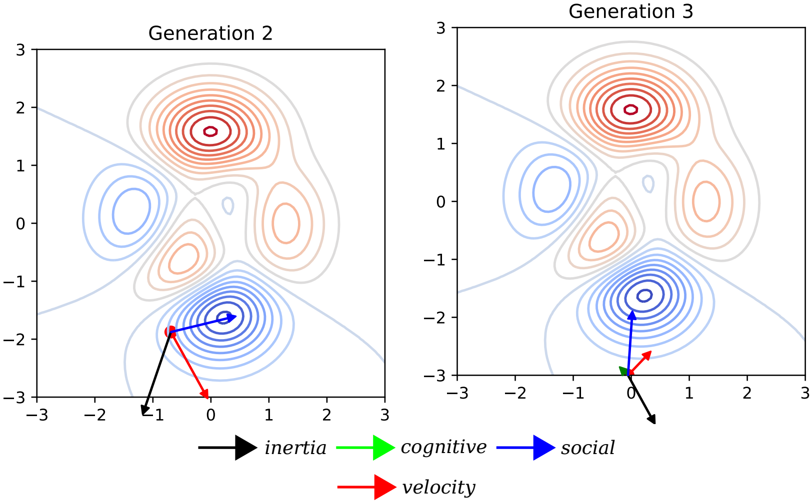
PSO - Position Update



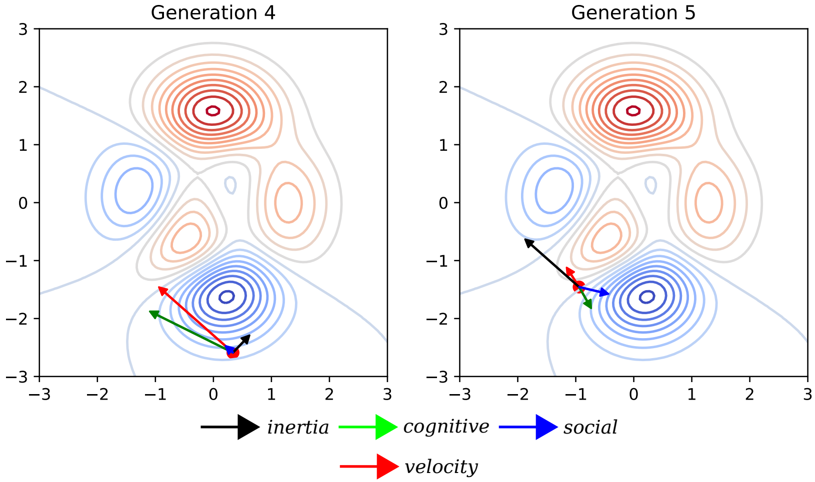
PSO - Visualization - Single Particle



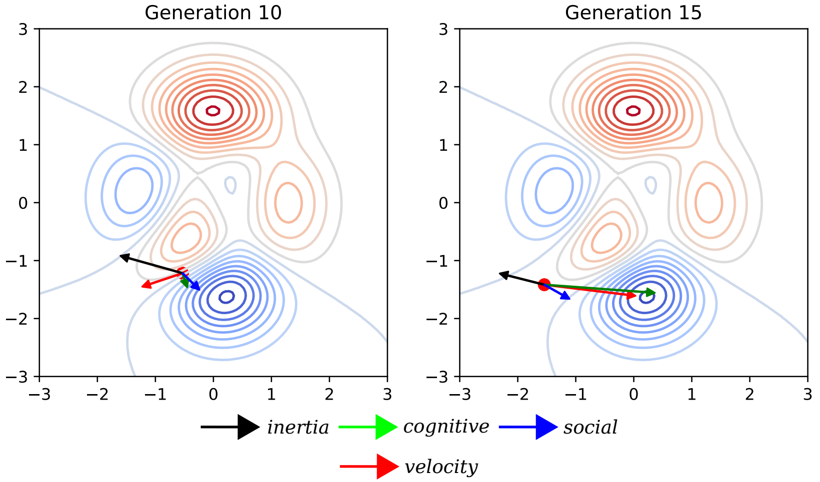
PSO - Visualization - Single Particle



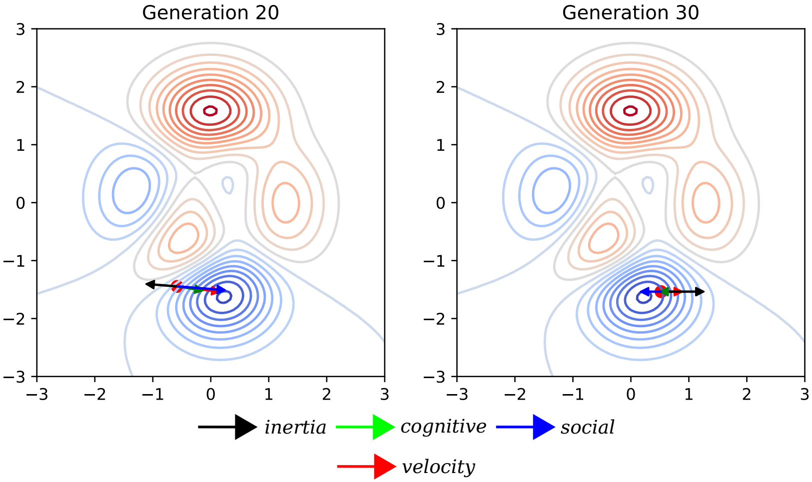
PSO - Visualization - Single Particle



PSO - Visualization - Single Particle



PSO - Visualization - Single Particle



PSO - Visualization - Swarm

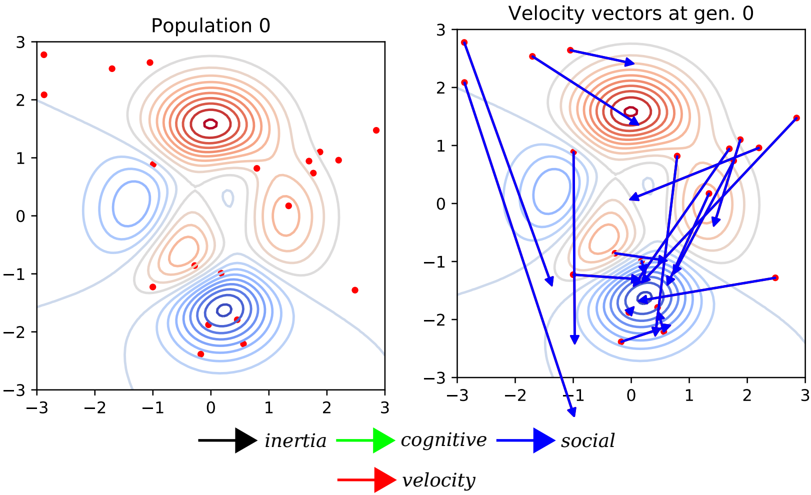


Figure: Generation 0

PSO - Visualization - Swarm

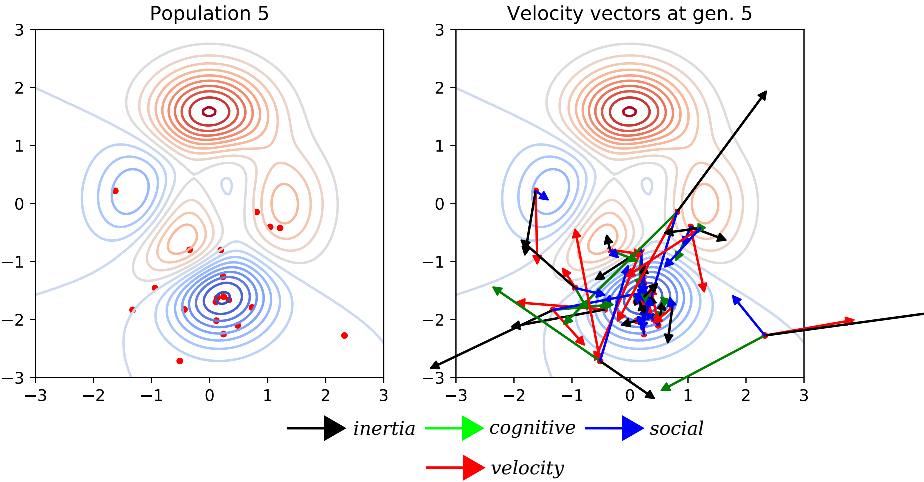


Figure: Generation 5

PSO - Visualization - Swarm

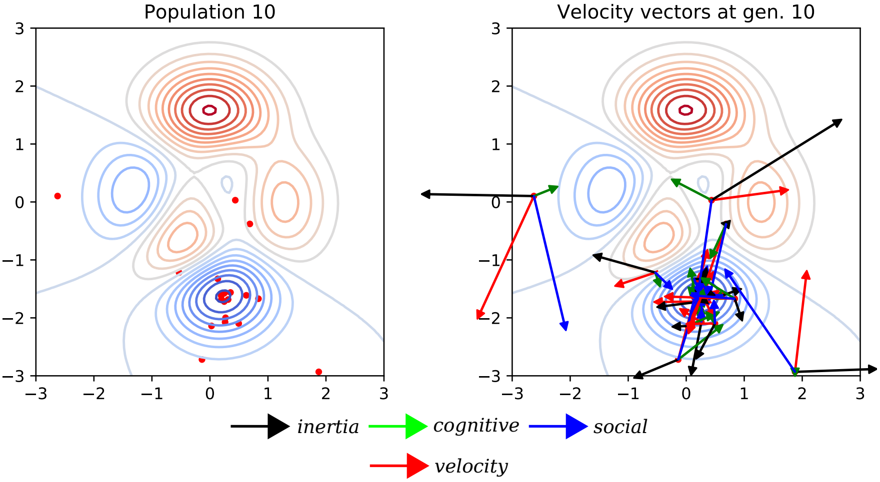


Figure: Generation 10

PSO - Visualization - Swarm

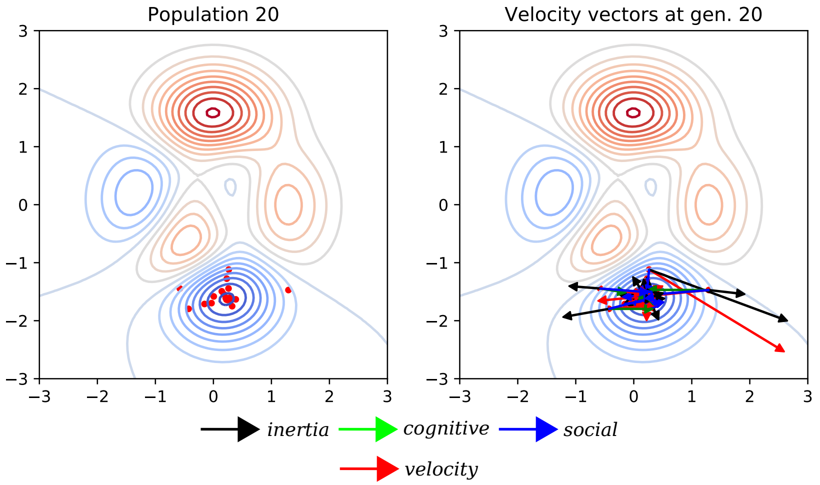


Figure: Generation 20

PSO - Visualization - Swarm

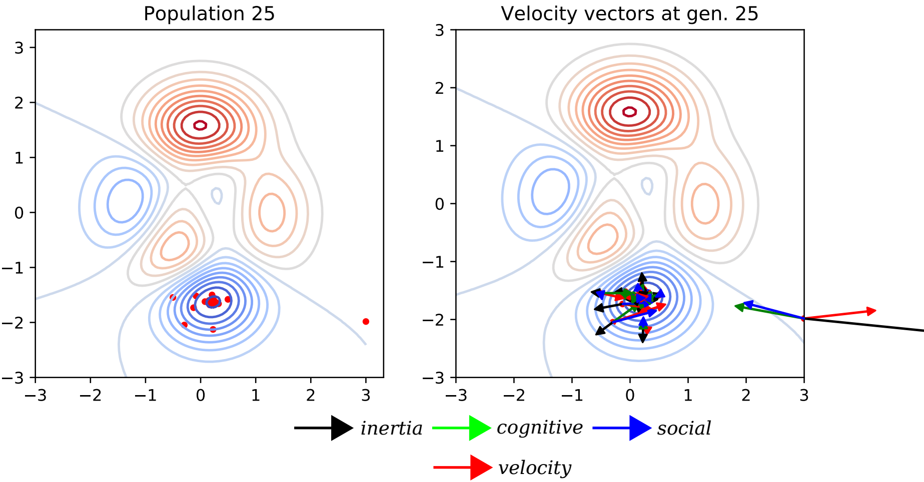


Figure: Generation 25

PSO - Visualization - Swarm

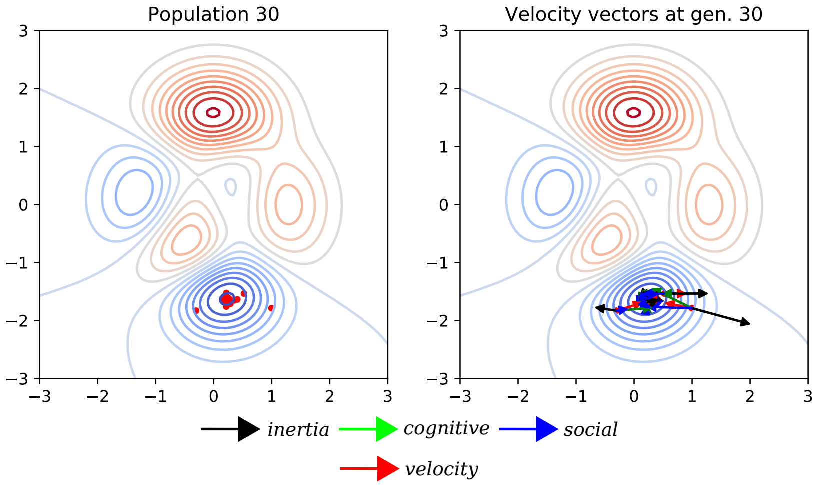


Figure: Generation 30

PSO - Control Parameters

The performance of PSO is influenced by its control parameters.

- Swarm size.
- Neighborhood topology.
- Coefficients, i.e., inertia weight w and acceleration constants c_1, c_2.
- ...

Research Questions

- Comparing DE and PSO? And comparing different evolutionary algorithms in general?
- How to set/tune/adapt control parameters?
- Considering your optimization problem, which algorithm should be used, how to customize, or how to design our own algorithm?
- ...

References

- Price K.V., Storn R.M., Lampinen J.A. (2005). The Differential Evolution Algorithm. Differential Evolution: A Practical Approach to Global Optimization, 37-134
- Riccardo P., Kennedy J., Blackwell T. (2007). Particle swarm optimization: An overview. Swarm intelligence 1 (1), 33-57