The Mechanics of Differential Evolution

Alexander G. Ororbia II

Biologically-Inspired Intelligent Systems

CSCI-633

2/20/2024

Metaheuristic: Differential Evolution (DE)

- Vector-based (population-based) algorithm; Storn & Price (1996/1997)
  - Viewed as self-organizing system
  - Individuals ‘evolve’ by recombination w/ other individuals & differentials between other individuals
- Devised for continuous search spaces, derivative-free
- No encoding/decoding required – real numbers are now solutions/chromosomes
- *DE/rand/1/bin*

DE General Mechanics

- Builds on the idea of genetic algorithms
- Three primary steps:
  - Mutation, crossover, selection

- Name convention: DE/x/y/z – x is mutation scheme, e.g., random (Rand) or best (Best), y is number of difference vectors, z is crossover scheme, e.g., binomial (Bin) or exponential (Exp) or either/agnostic (*)
  - **We will start with**: DE/Rand/1/*

Differential Evolution (Naming Convention)

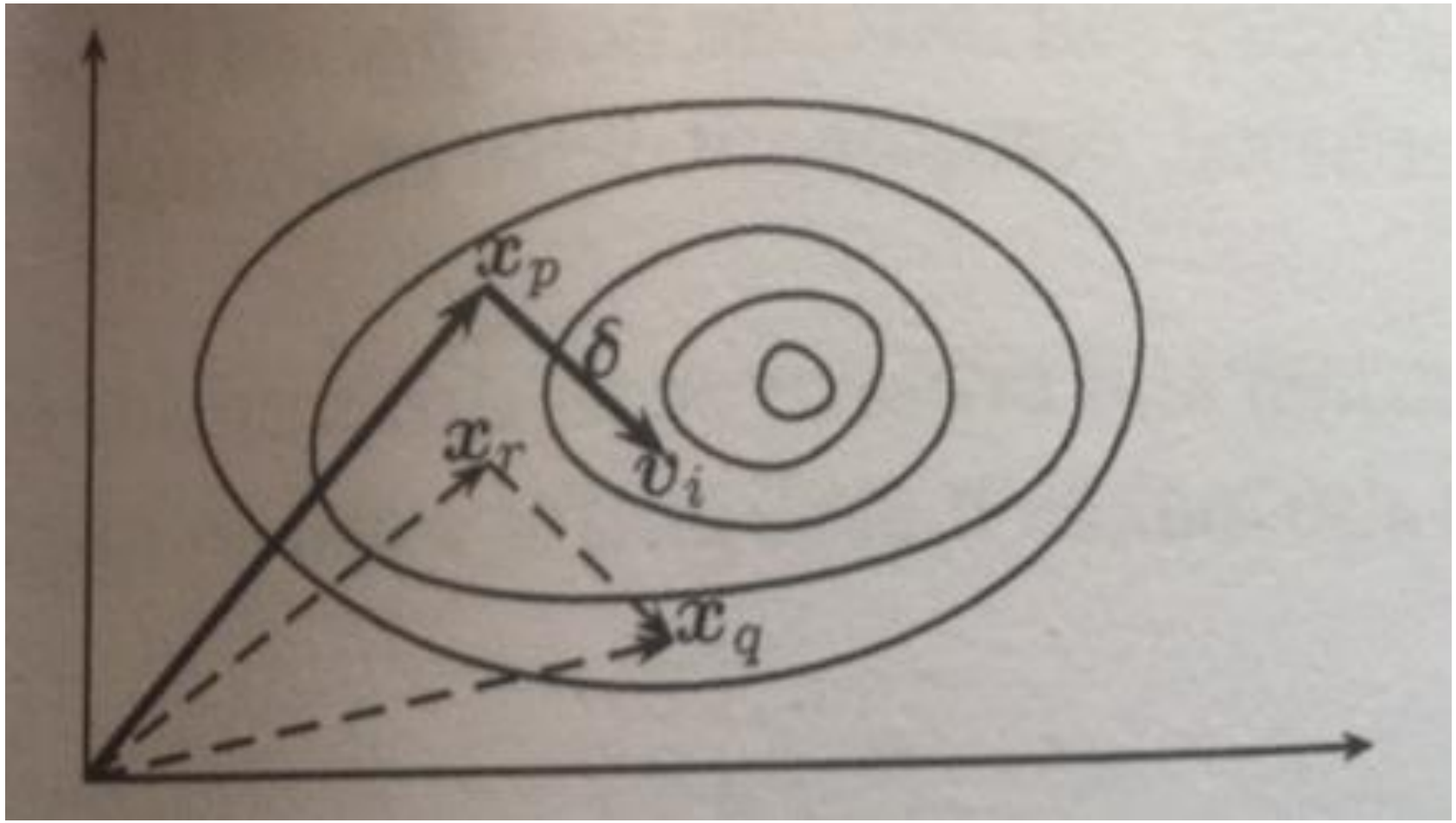
Many Variations:

- best/n/bin
- best/n/exp
- random/n/bin
- random/n/exp
- current/n/bin
- current/n/exp

Parent Selection / Number of Pairs / Recombination

- In general:
  - Perform binary or exponential recombination between current individual and another individual modified by a scaled difference between *n* pairs of other individuals


WHITE BOARD TIME!

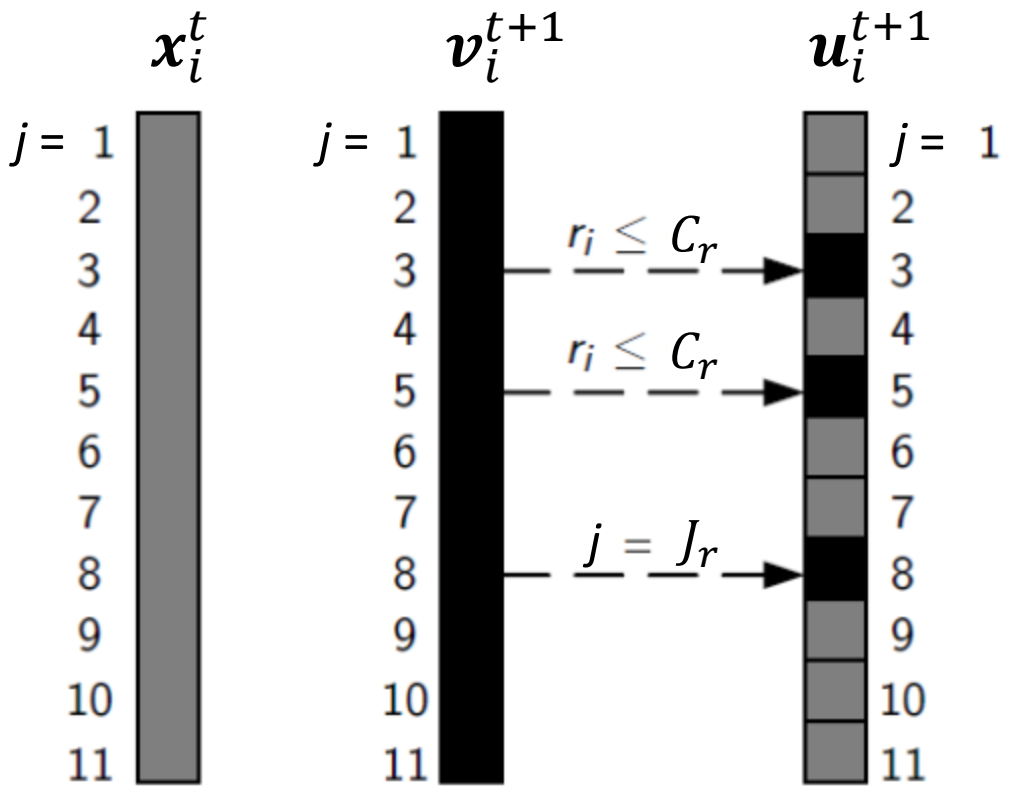


Schematic representation the application of a perturbation/mutation in DE, according to perturbation $\boldsymbol{\delta} = \alpha(\boldsymbol{x}_q^t - \boldsymbol{x}_r^t)$ (movement along function space).


WHITE BOARD TIME!

Binomial Crossover

- *Vector crossover visualization*




WHITE BOARD TIME!

Pseudocode

Differential Evolution

Initialize the population $\boldsymbol{x}$ with randomly generated solutions
Set the weight $F \in [0, 2]$ and crossover probability $C_r \in [0, 1]$
while (stopping criterion)
  for $i = 1$ to n,
  For each $\boldsymbol{x}_i$, randomly choose 3 distinct vectors $\boldsymbol{x}_p$, $\boldsymbol{x}_r$ and $\boldsymbol{x}_r$
  Generate a new vector $\boldsymbol{v}$ by DE scheme (6.2)
  Generate a random index $J_r \in \{1, 2, ..., d\}$ by permutation
  Generate a randomly distributed number $r_i \in [0, 1]$
  for $j = 1$ to d,
  For each parameter $\boldsymbol{v}_{j,i}$ (jth component of $\boldsymbol{v}_i$), update

$$\boldsymbol{u}_{j,i}^{t+1} = \begin{cases} \boldsymbol{v}_{j,i}^{t+1} & \text{if } r_i \le C_r \text{ or } j = J_r \\ \boldsymbol{x}_{j,i}^{t} & \text{if } r_i > C_r \text{ and } j \ne J_r, \end{cases}$$

  end
  Select and update the solution by (6.5)
  end
end
Post-process and output the best solution found

Figure 6.2 Pseudo code of differential evolution.

Pseudocode

Differential Evolution

Initialize the population $\boldsymbol{x}$ with randomly generated solutions
Set the weight $F \in [0, 2]$ and crossover probability $C_r \in [0, 1]$
while (stopping criterion)
  for $i = 1$ to n,
  For each $\boldsymbol{x}_i$, randomly choose 3 distinct vectors $\boldsymbol{x}_p$, $\boldsymbol{x}_r$ and $\boldsymbol{x}_r$
  Generate a new vector $\boldsymbol{v}$ by DE scheme (6.2)
  Generate a random index $J_r \in \{1, 2, ..., d\}$ by permutation
  Generate a randomly distributed number $r_i \in [0, 1]$
  for $j = 1$ to d,
  For each parameter $\boldsymbol{v}_{j,i}$ (jth component of $\boldsymbol{v}_i$), update

$$\boldsymbol{u}_{j,i}^{t+1} = \begin{cases} \boldsymbol{v}_{j,i}^{t+1} & \text{if } r_i \le C_r \text{ or } j = J_r \\ \boldsymbol{x}_{j,i}^{t} & \text{if } r_i > C_r \text{ and } j \ne J_r, \end{cases}$$

  end
  Select and update the solution by (6.5)
  end
end
Post-process and output the best solution found

Figure 6.2 Pseudo code of differential evolution.

Improves search efficiency by ensuring at least one dimension of perturbed solution is different from the original


WHITE BOARD TIME!

Questions?