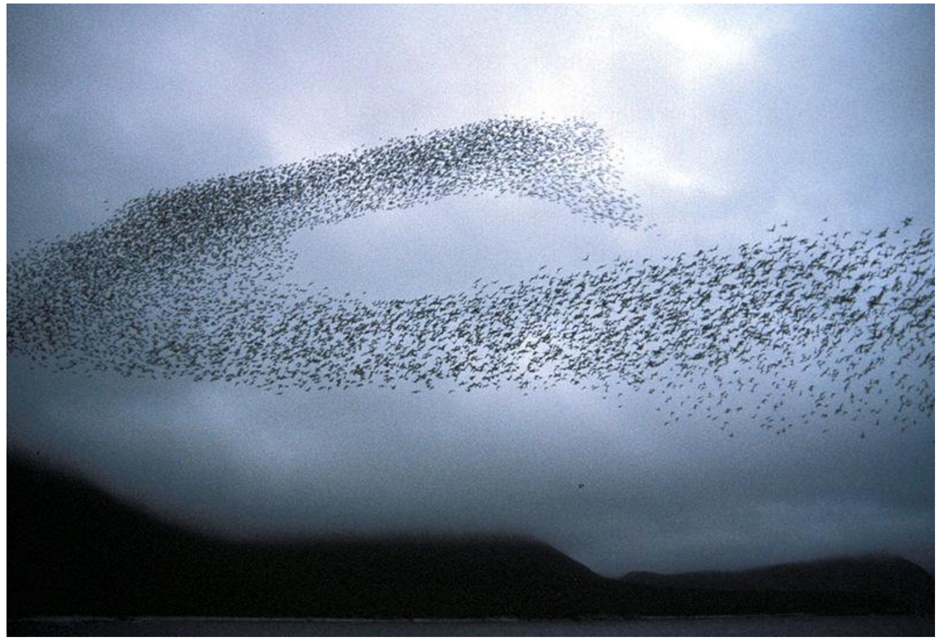
Differential Evolution

Winter 2024

When GA meets PSO



Basics of Differential Evolution

- Introduced by Storn and Price in 1996
- Optimize real parameters, mD real-valued functions, which are not necessarily continuous or differentiable
- Stochastic, population-based search
- May conceived as a mix of GA and PSO, although it appeared almost the same time as PSO

Dr. Rainer Storn

Kenneth Price

Differential Evolution

- References
  - Storn, R. and Price, K. (1997), “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces.”, Journal of Global Optimization, 11, pp. 341–359.
  - Storn, R., Price, K., and Lampinen, J. (2005). Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, Berlin.

Concepts

- Maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to simple formulae of vector-crossover and vector-mutation, and then keeping whichever candidate solution has the best score or fitness on the optimization problem

Notation

- Number of decision variables is *D*
- Size of population *N* (greater than 4)
- A parameter vector (solution) is

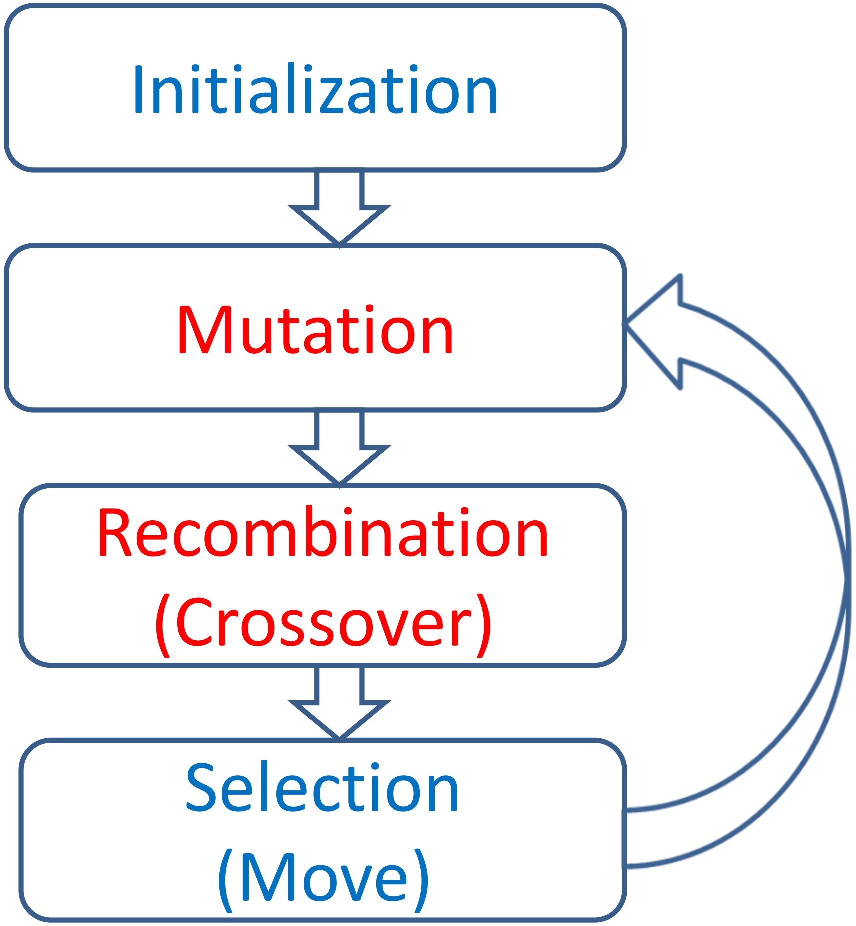
  $x_{i,t}=[x_{1,i,t}, x_{2,i,t}, \ldots x_{D,i,t}]$ $i=1,2,\ldots N$

  or

  $x_{i,t}=[x_{j,i,t}]$ $i=1,2,\ldots N;\ j=1,2,\ldots D$

  Where *t* is the generation (iteration)

Flowchart


Initialization
Mutation
Recombination
(Crossover)
Selection
(Move)

Initialization

- The initial population is chosen randomly if nothing is known about the system.
- Assume a uniform probability distribution for all random decisions unless otherwise stated.
- It is usually required to have lower and upper bounds for every decision variables

Mutation

- To perturb the population
- For a given decision vector $x_{i,t}$ randomly select three vectors $x_{r1=l,t}$, $x_{r2=j,t}$ and $x_{r3=k,t}$ such that the indices ***i***, $r1$, $r2$ and $r3$ are all different (so, we need at least 4 individuals)
- For each $x_{i,t}$, generate a new vector $v_{i,t+1}$

$$v_{i,t+1}= x_{r1=l,t} + F(x_{r2=j,t} - x_{r3=k,t})$$

F is a tuning constant (mutation scale constant)

Recombination

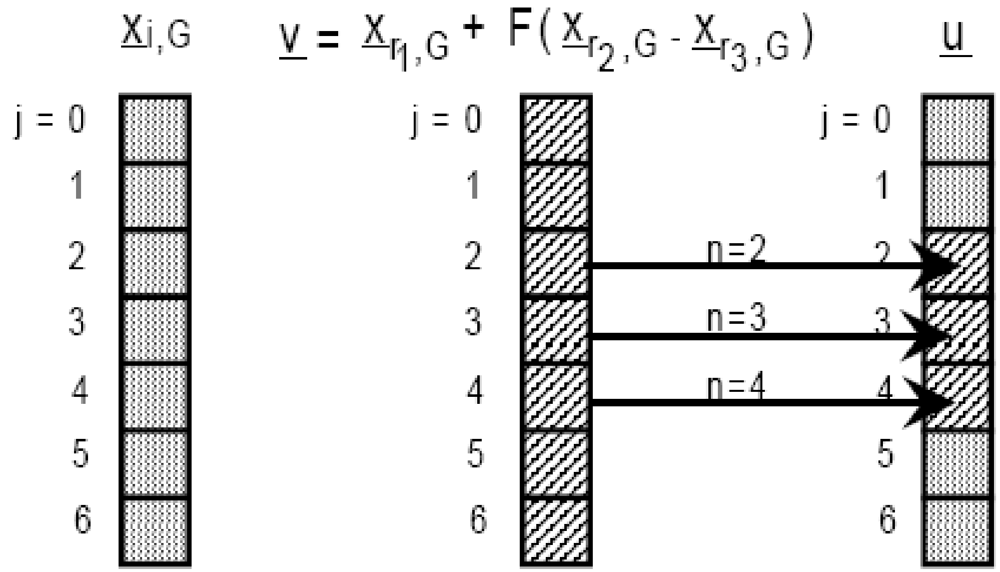
- To increase the diversity
- At each iteration, mix vectors x and v (similar to crossover)

  $u_j = v_j$ for $j = \mathrm{mod}(n,D)$, $\mathrm{mod}(n+1,D), \ldots \mathrm{mod}(n+L-1,D)$

  $u_j = x_j$ for all other j between 1 and D

  n, L are specified constants
- The mix is done when a random number is less than a pre-specified crossover probability, CR

Recombination Illustration



n=2, L=3, D=7

j=mod(2,7), mod(2+1,D), mod(2+2,D)= 2,3,4

Another way: uniform crossover

$u_{j,i,t+1} = v_{j,i,t+1}$ if rand≤CR or j=I_{rand}

$u_{j,i,t+1} = x_{j,i,t}$ if rand>CR or j≠I_{rand}

i (solution index)=*1,2,…N*

j (variable index)=*1,2,…D*

rand=U[0,1]

I_{rand} is a random integer from [1, 2,…*D*]

Selection: Move

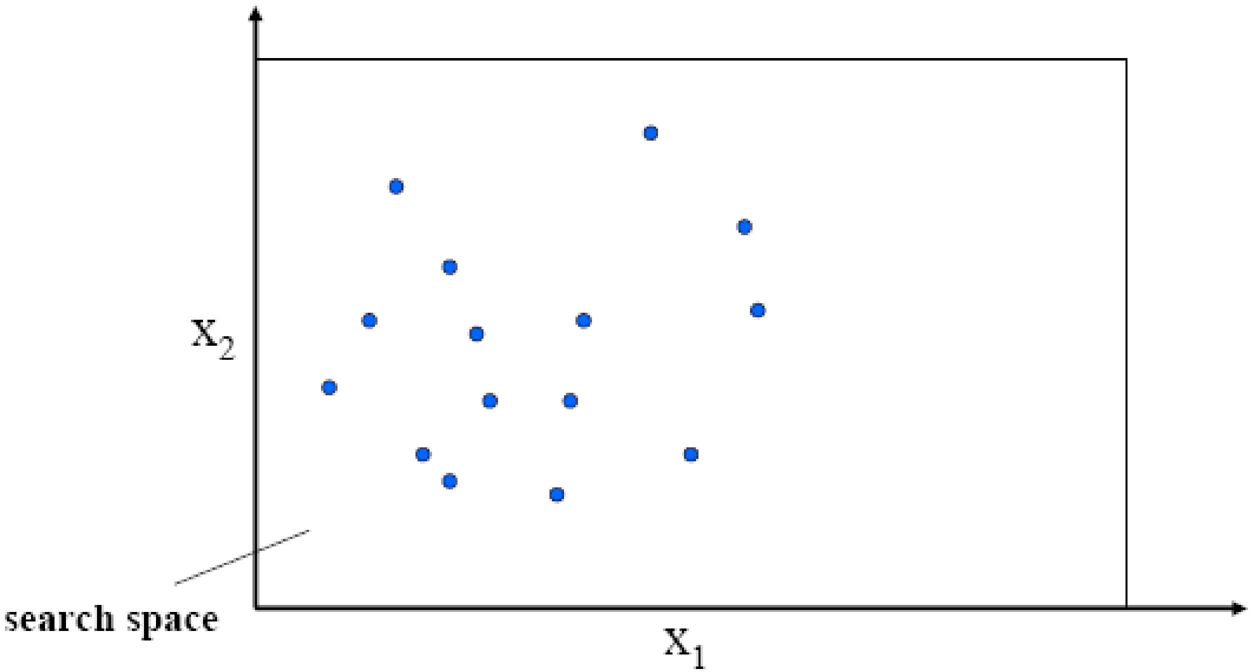
- The target vector $x_{i,t+1}$ is compared with the trial vector $u_{i,t+1}$
- The one with the better function value is admitted to the next generation

$x_{i,t+1} = u_{i,t+1}$ if $f(u_{i,t+1})$ is better than $f(x_{i,t})$

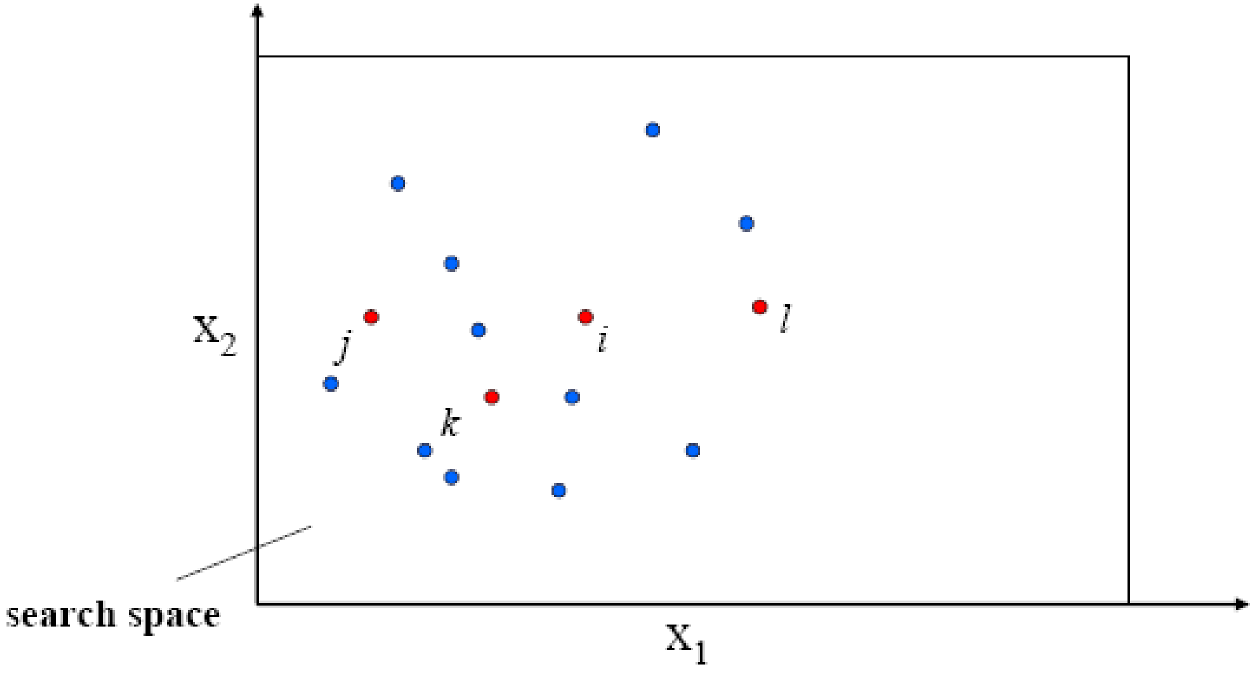
$x_{i,t+1} = x_{i,t}$ otherwise

- Greedy rule: does not allow worse move

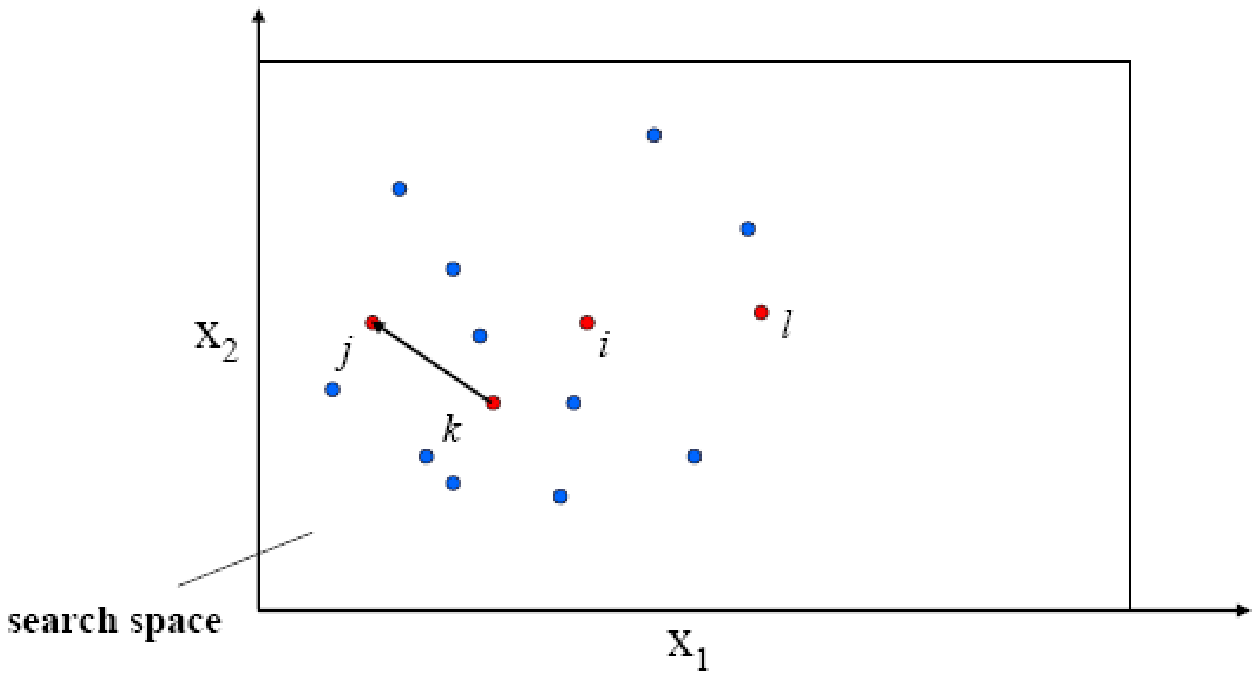
Initialization



Mutation + Recombination (1)

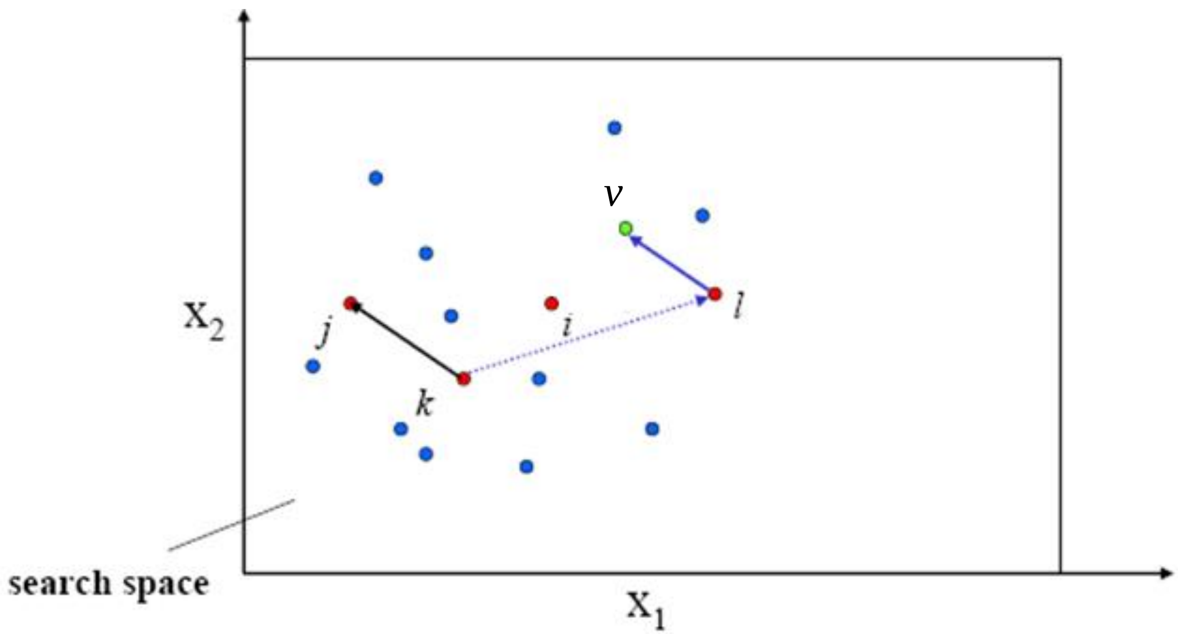


Mutation + Recombination (2)



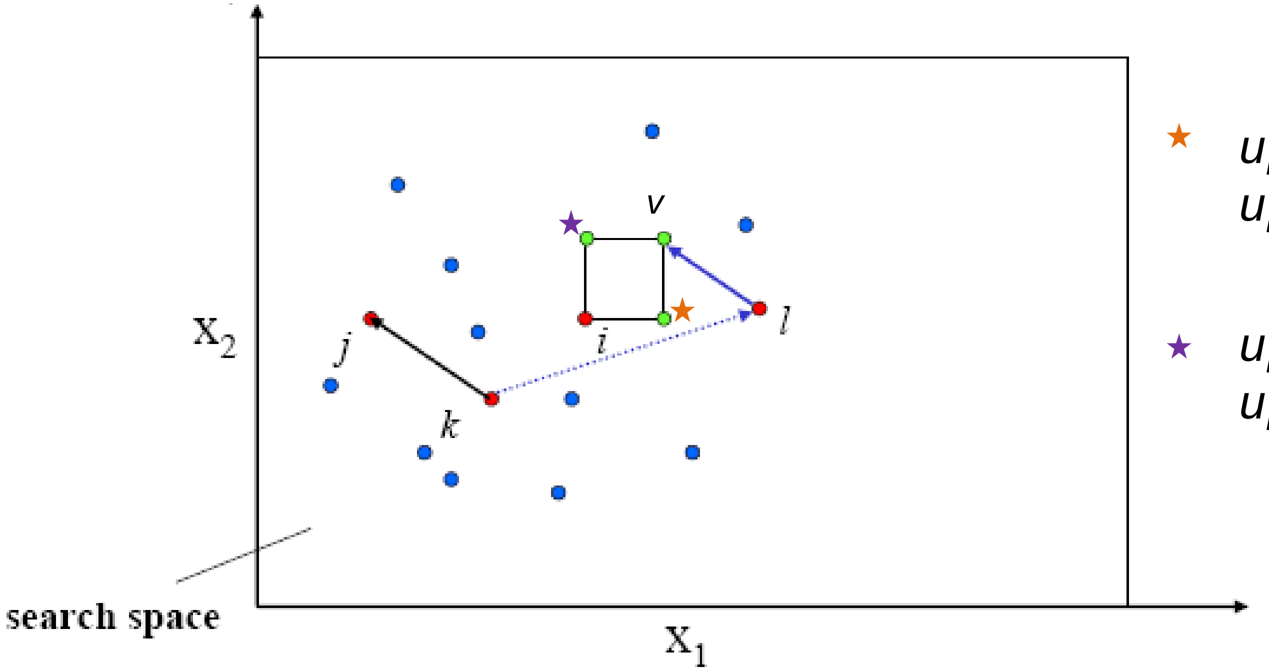
Mutation + Recombination (3)

$$v_{i,t+1} = x_{r1=l,t} + F(x_{r2=j,t} - x_{r3=k,t})$$



Mutation + Recombination (4)

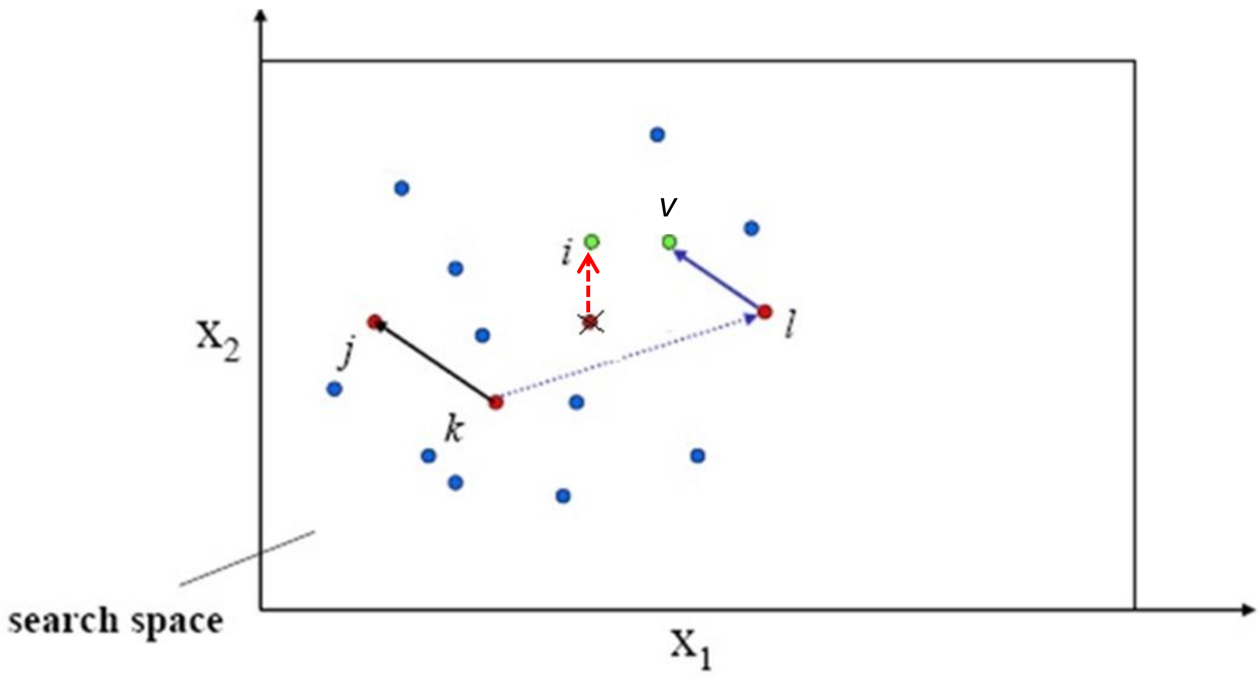
$v_{i,t+1} = x_{r1=l,t} + F(x_{r2=j,t} - x_{r3=k,t})$



★ $u_{i,1,t+1} = \mathbf{v}_{i,1,t+1}$
$u_{i,2,t+1} = x_{i,2,t}$

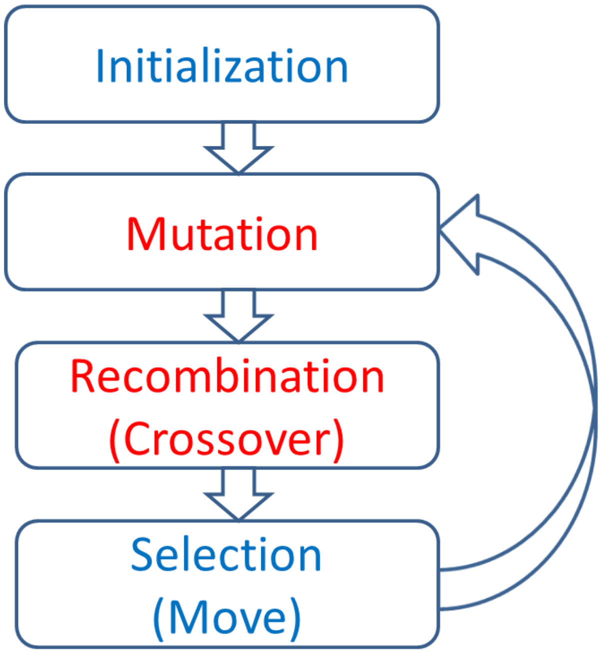
★ $u_{i,1,t+1} = \mathbf{x}_{i,1,t}$
$u_{i,2,t+1} = v_{i,2,t+1}$

Comparison & Move



Loop

- Mutation, recombination and selection continue until a stopping criterion is reached

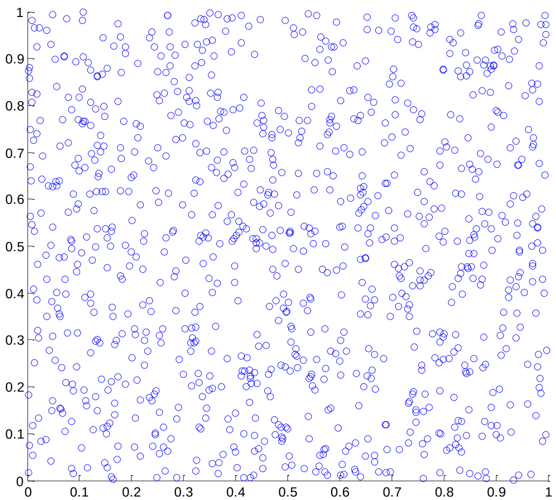


Initialization

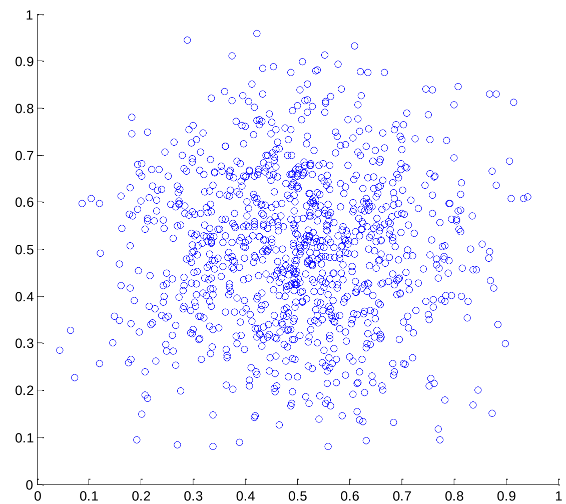
- Initialize the location of solutions
  - Uniform distribution
  - Gaussian distribution
  - Other distributions with focus on specific areas
- Such initialization may be used in other population-based meta-heuristics

Distribution of initial solutions

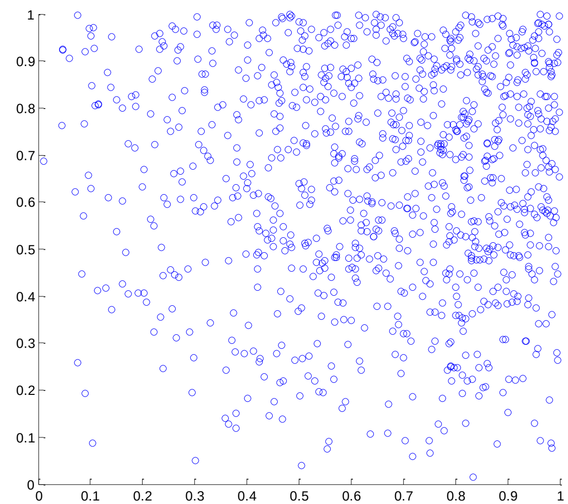
Uniform



Gaussian



Triangular



Recombination

- Crossover mechanisms
  - One point
  - Multiple points
  - Uniform
- Which one is used in conventional DE?

F and *NP*

$$v_{i,t+1} = x_{r1=l,t} + F(x_{r2=j,t} - x_{r3=k,t})$$

- Mutation Scale Factor *F*
  - Fixed at a value between 0.5 and 1.0
  - Linearly decreasing from 1.0 to 0.5
  - May be a random variable
    - Choose *F* from the interval [0.5, 1.0] randomly for each generation or for each difference vector
- Size of population *NP*
  - Roughly 5D ~ 10D

Crossover Rate

- CR can be thought of as mutation rate as a probability that a parameter inherited from a mutant
- Setting CR to a low value, e.g. CR=0.1~0.2 fosters the search along the coordinate axes, suitable for separable function.
- For high dependence between decision variables, the choice of CR=0.9~1.0 may be more appropriate.

Discrete DE

- DE may be used to handle discrete problems
  - Binary
  - Integer
  - Permutation
- The extension to discrete problems is similar to PSO
  - Rounding
  - Probability function of specific listed-item
  - Sorting order of continuous values

Variant 1: Mutation Strategies

$$\text{“DE/rand/1”}: \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G})$$

$$\text{“DE/best/1”}: \vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G})$$

$$\text{“DE/target-to-best/1”}: \vec{V}_{i,G} = \vec{X}_{i,G} + F \cdot (\vec{X}_{best,G} - \vec{X}_{i,G}) + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G})$$

$$\text{“DE/best/2”}: \vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) + F \cdot (\vec{X}_{r_3^i,G} - \vec{X}_{r_4^i,G})$$

$$\text{“DE/rand/2”}: \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) + F \cdot (\vec{X}_{r_4^i,G} - \vec{X}_{r_5^i,G}).$$

DE/rand/1/either-or

$$\vec{U}_{i,G} = \begin{cases} \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}), & \text{if } rand_i(0,1) < p_F \\ \vec{X}_{r_1^i,G} + K \cdot (\vec{X}_{r_2^i,G} + \vec{X}_{r_3^i,G} - 2.\vec{X}_{r_1^i,G}), & \text{otherwise} \end{cases}$$

Price, Storn, and Lampinen (2005). Differential Evolution: A Practical Approach to Global Optimization

Variant 2: Neighborhood Search DE

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + \begin{cases} \vec{d}_{i,G}.N(0.5, 0.5), & \text{if } rand_i(0,1) < 0.5 \\ \vec{d}_{i,G}.\delta, & \text{otherwise} \end{cases}$$

- Vector *d* represents the difference vector ($x_{r2,G} - x_{r3,G}$)
- N(0.5,0.5): Gaussian random number with mean 0.5 and standard deviation 0.5
- δ denotes a Cauchy random variable with scale parameter $\gamma = 1$

Another Scheme

- Enhance the greediness

  $v_{i,t+1} = x_{r1,t} + \lambda(x_{best,t} - x_{r1,t}) + F(x_{r2,t} - x_{r3,t})$

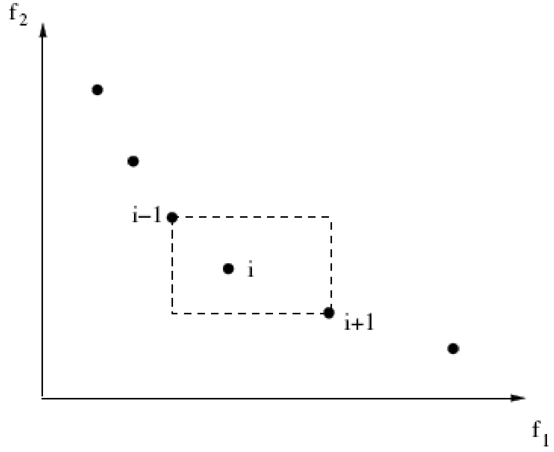
- This is a different approach, but aims at the same purpose of "gBest" in PSO

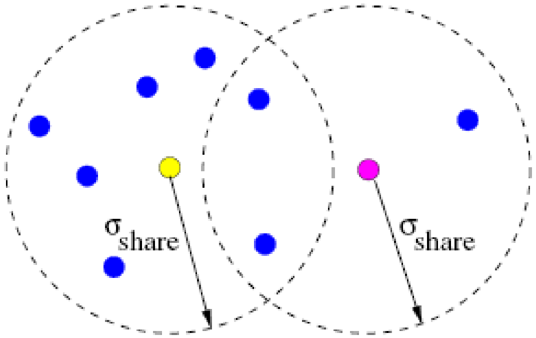
Variant 3: Neighborhood

- The definition of “the best”
- Topology of neighborhood
- Local and global neighborhood
  - Best in the neighborhood
  - Best in the entire population

Multiobjective DE: Diversity

- Crowding distance: giving preference to solutions with greater crowding distance
- Niche: giving preference to solutions located in less crowded areas

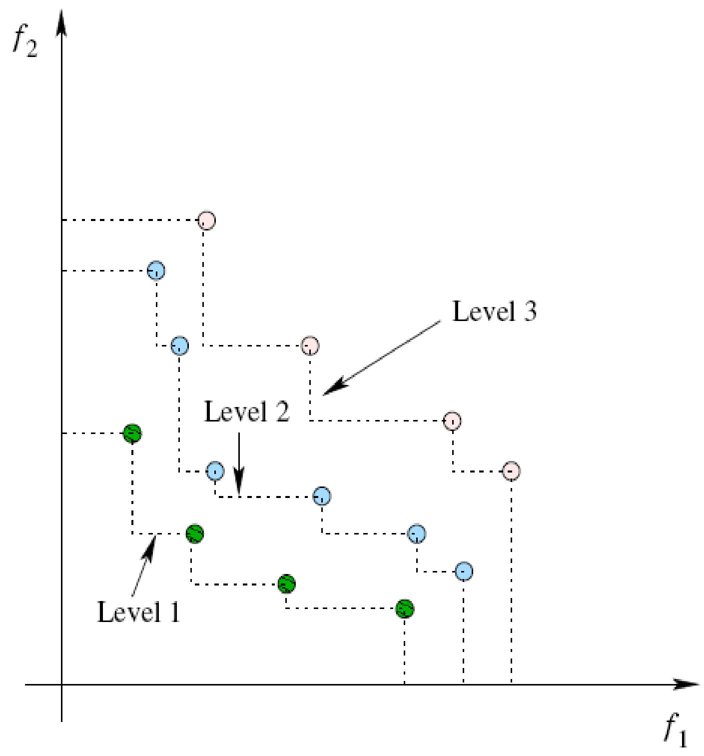




Das and Suganthan (2011). “Multi-objective Optimization Using Differential Evolution: A Survey of the State-of-the-Art.”

Multiobjective DE: Elitism

- Elite archive: store the non-dominated solutions along the search
- Apply non-dominated sorting to determine which solutions are used to produce next generation



Conclusions for DE

- DE is simple and easy to program
- Suitable for continuous domains
- Only a few parameters to adjust
- Converge fast; maybe too greedy