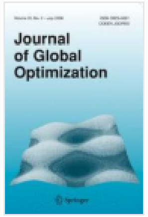
Differential Evolution

Prakash Kotecha
Debasis Maharana & Remya Kommadath
Department of Chemical Engineering
Indian Institute of Technology Guwahati

Outline

- Differential Evolution (DE)
- Difference-vector based mutation
- Recombination operators
- Variants of DE
- Detailed working of DE with an example
- Preliminary comparison of algorithms

Differential Evolution

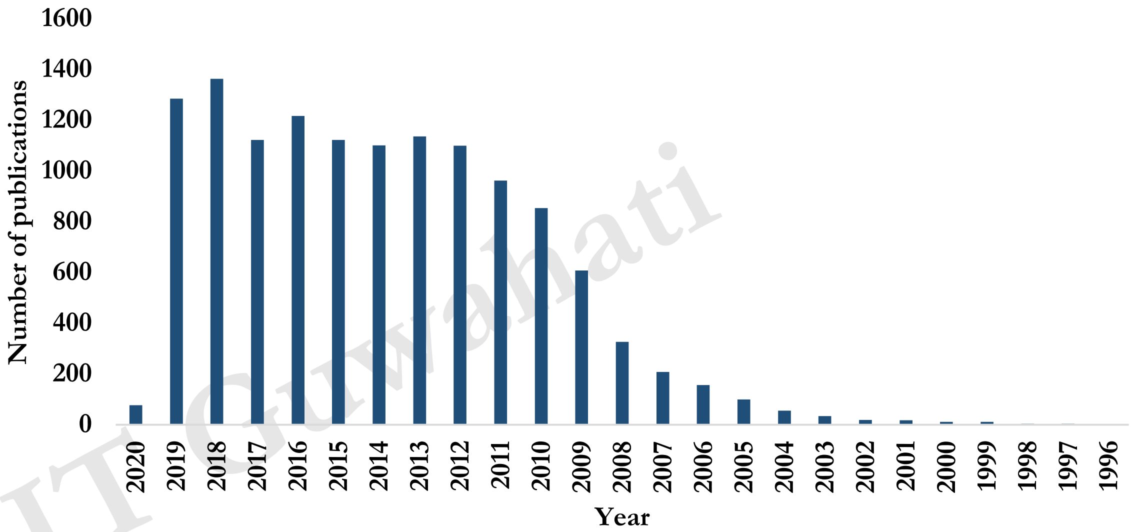


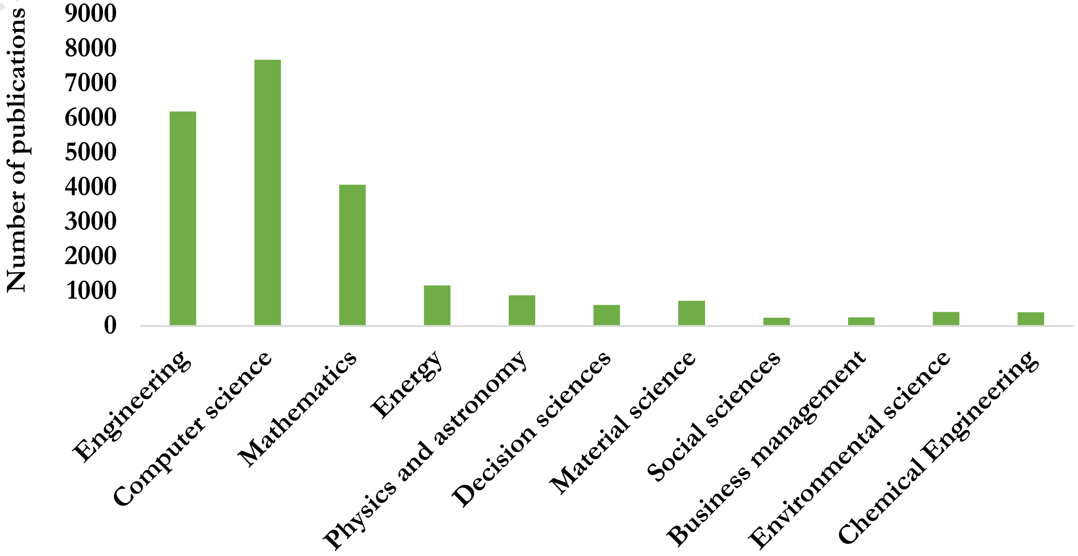
Journal of Global Optimization
December 1997, Volume 11, Issue 4, pp 341–359 | Cite as

Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces

Authors | Authors and affiliations

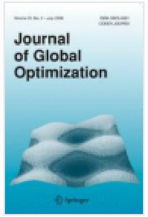
Rainer Storn, Kenneth Price





Around 13,000 publications (Scopus, Dec 2019)

Differential Evolution



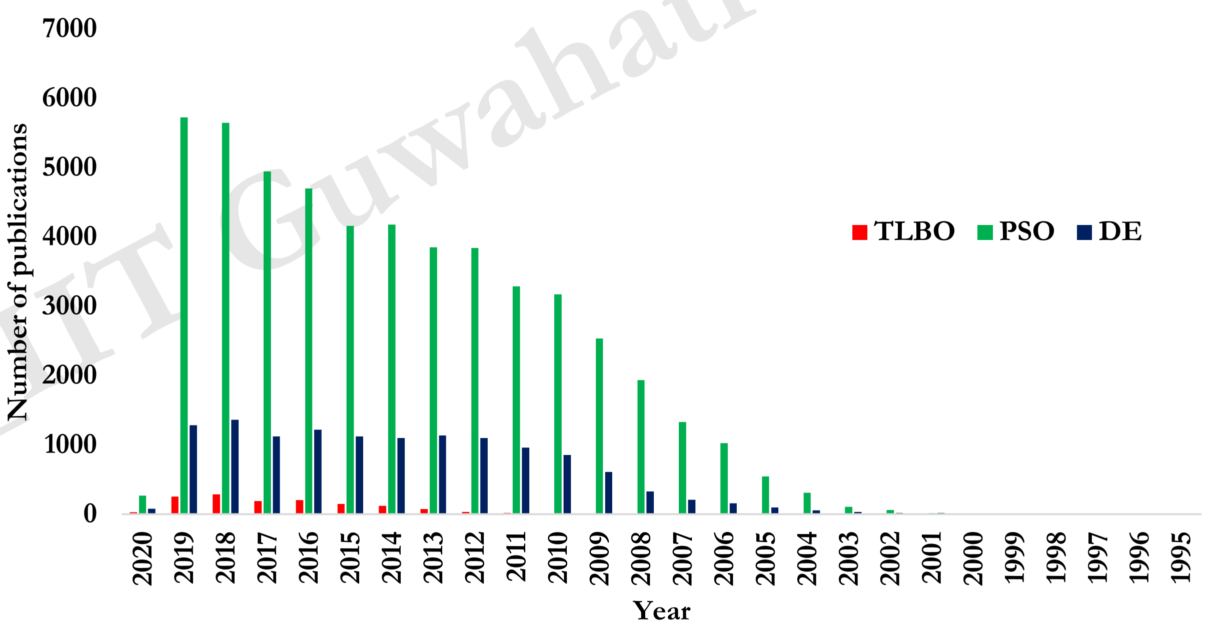
Journal of Global Optimization

December 1997, Volume 11, Issue 4, pp 341–359 | Cite as

Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces

Authors | Authors and affiliations

Rainer Storn, Kenneth Price



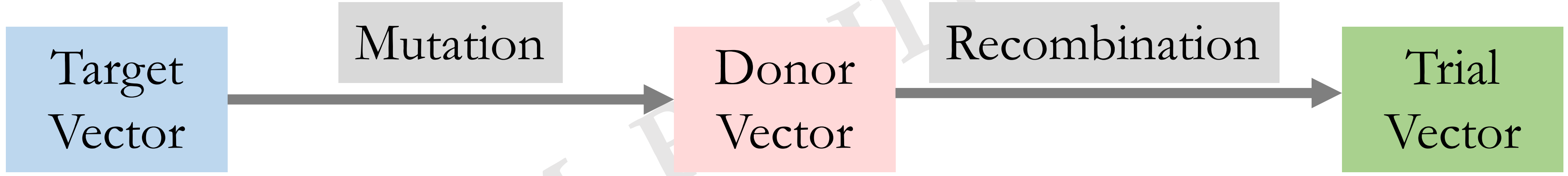
Around 13,000 publications (Scopus, Dec 2019)

Differential Evolution

- Stochastic population based technique.

- Each solution is known as genome/chromosome.

- Each chromosome undergoes mutation followed by recombination.



- Selection of better solutions is performed only after the generation of all trial vectors.

- Greedy selection is performed between target and trial vectors.

Difference-vector based mutation

- Donor vector (V) of a chromosome (X_i) is created as

$$V = X_{r_1} + F\left(X_{r_2} - X_{r_3}\right)$$

F	Scaling factor, a constant between 0 and 2
r_1, r_2, r_3	Random solutions $r_1, r_2, r_3 \in \{1,2,3,\ldots,N_p\}$ *and* $r_1 \neq r_2 \neq r_3 \neq i$

- Target vector is not involved in mutation.

- Total 4 vectors are involved in the mutation of a target vector and hence $N_p \geq 4$.

Recombination: Binomial (uniform) crossover

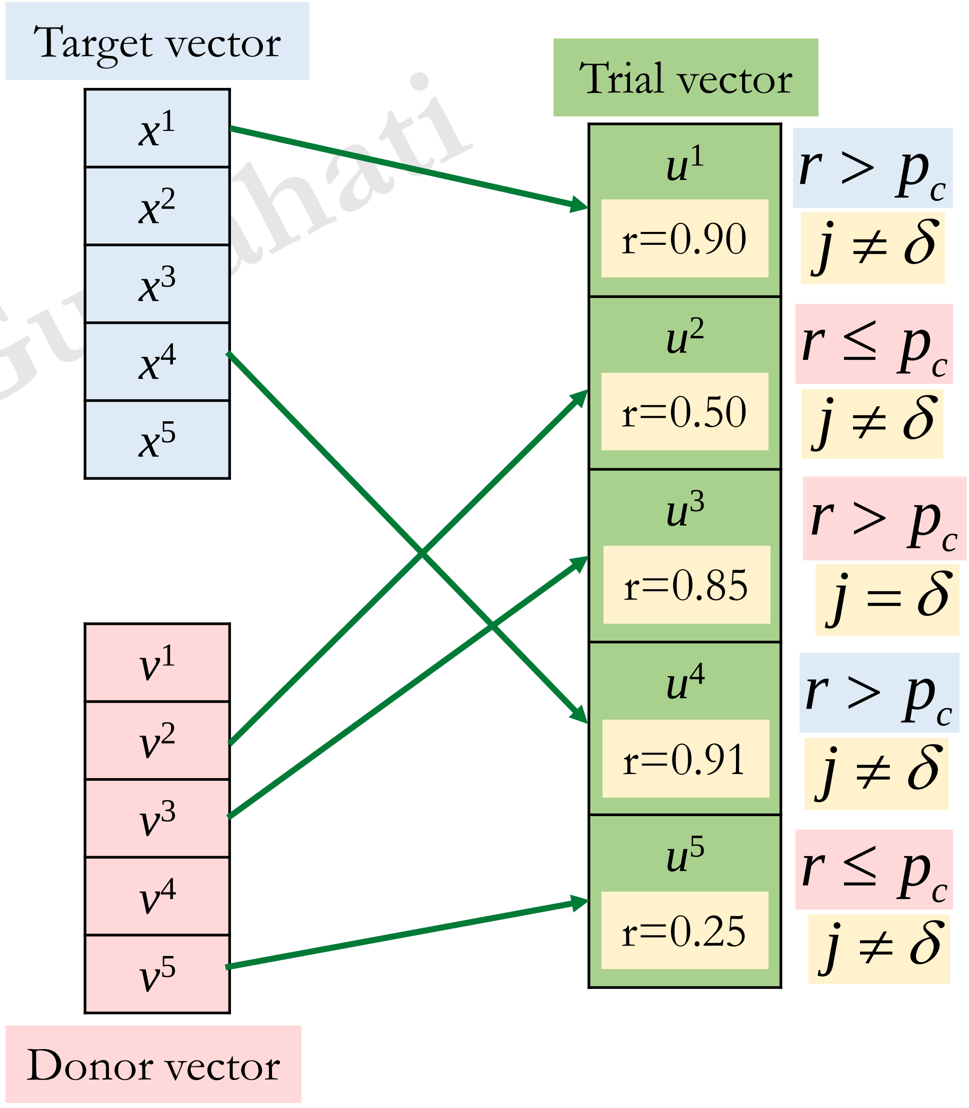
- Performed to increase the diversity
- Creation of trial vector can be

$$u^j = \begin{cases} v^j & \text{if } r \le p_c \text{ OR } j = \delta \\ x^j & \text{if } r > p_c \text{ AND } j \ne \delta \end{cases}$$

p_c	crossover probability
δ	randomly selected variable location $\delta \in \{1,2,3,\ldots,D\}$
r	random number between 0 and 1
u^j	j^{th} variable of trial vector
v^j	j^{th} variable of donor vector
x^j	j^{th} variable of target vector

- δ ensures that at least one variable is obtained from the donor vector
- Probability for crossover (p_c) is generally high
- High p_c results in more variables from donor

Let D = 5, δ = 3, p_c=0.8



Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, Journal of Global Optimization, 11, 341–359, 1997

Recombination: Binomial (uniform) crossover

- Performed to increase the diversity
- Creation of trial vector can be

$$u^j = \begin{cases} v^j & \text{if } r \le p_c \text{ OR } j = \delta \\ x^j & \text{if } r > p_c \text{ AND } j \ne \delta \end{cases}$$

p_c	crossover probability
δ	randomly selected variable location $\delta \in \{1,2,3,\ldots,D\}$
r	random number between 0 and 1
u^j	j^{th} variable of trial vector
v^j	j^{th} variable of donor vector
x^j	j^{th} variable of target vector

- δ ensures that at least one variable is obtained from the donor vector
- Probability for crossover (p_c) is generally high
- High p_c results in more variables from donor

Let D = 5, δ = 3, p_c=0.8

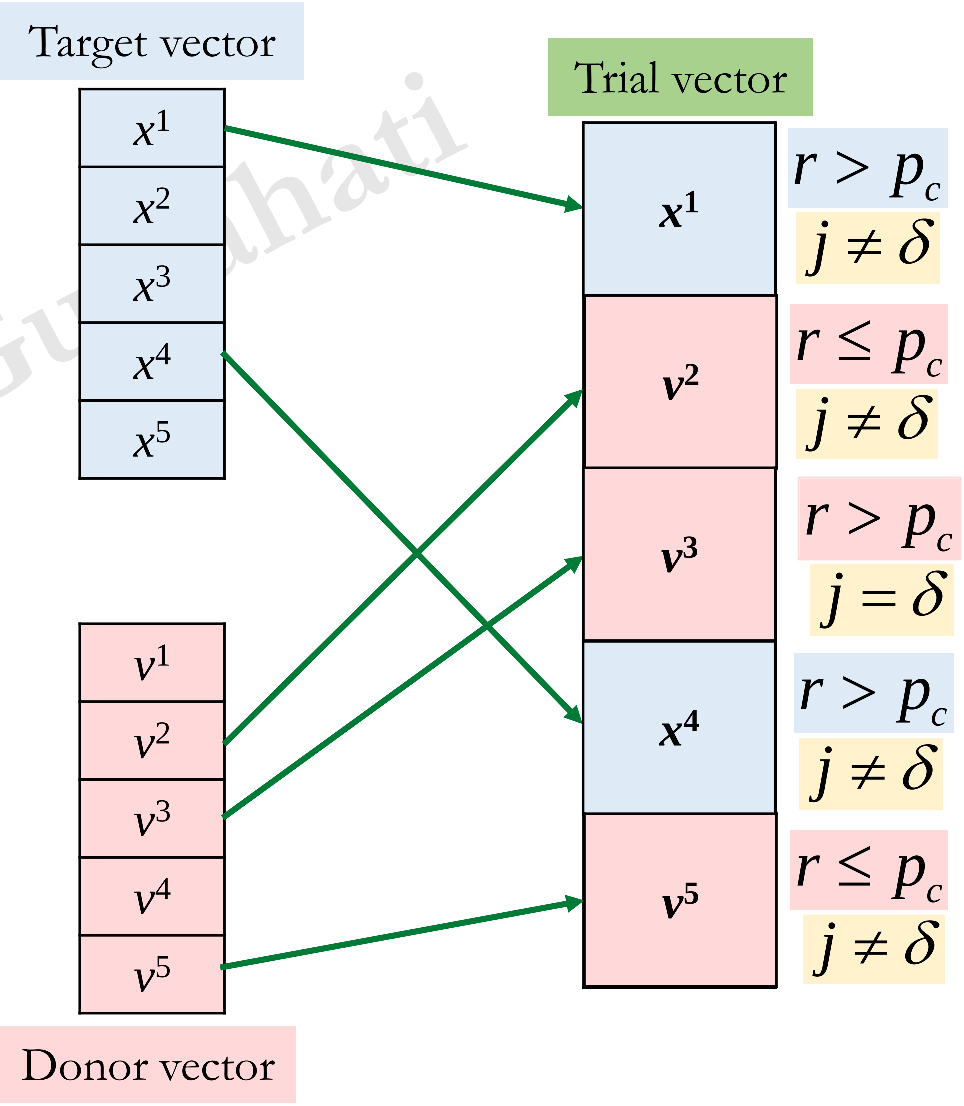
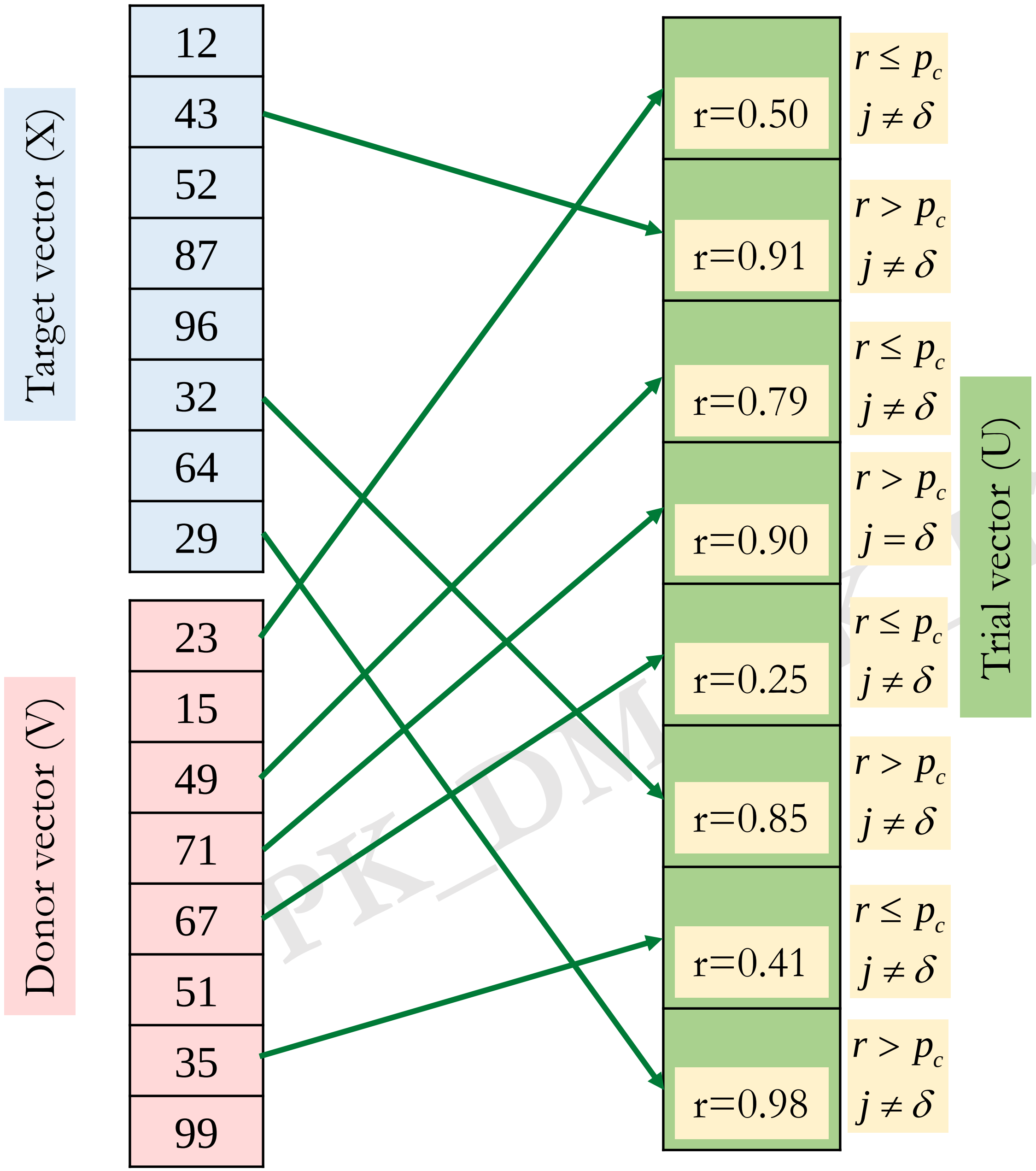


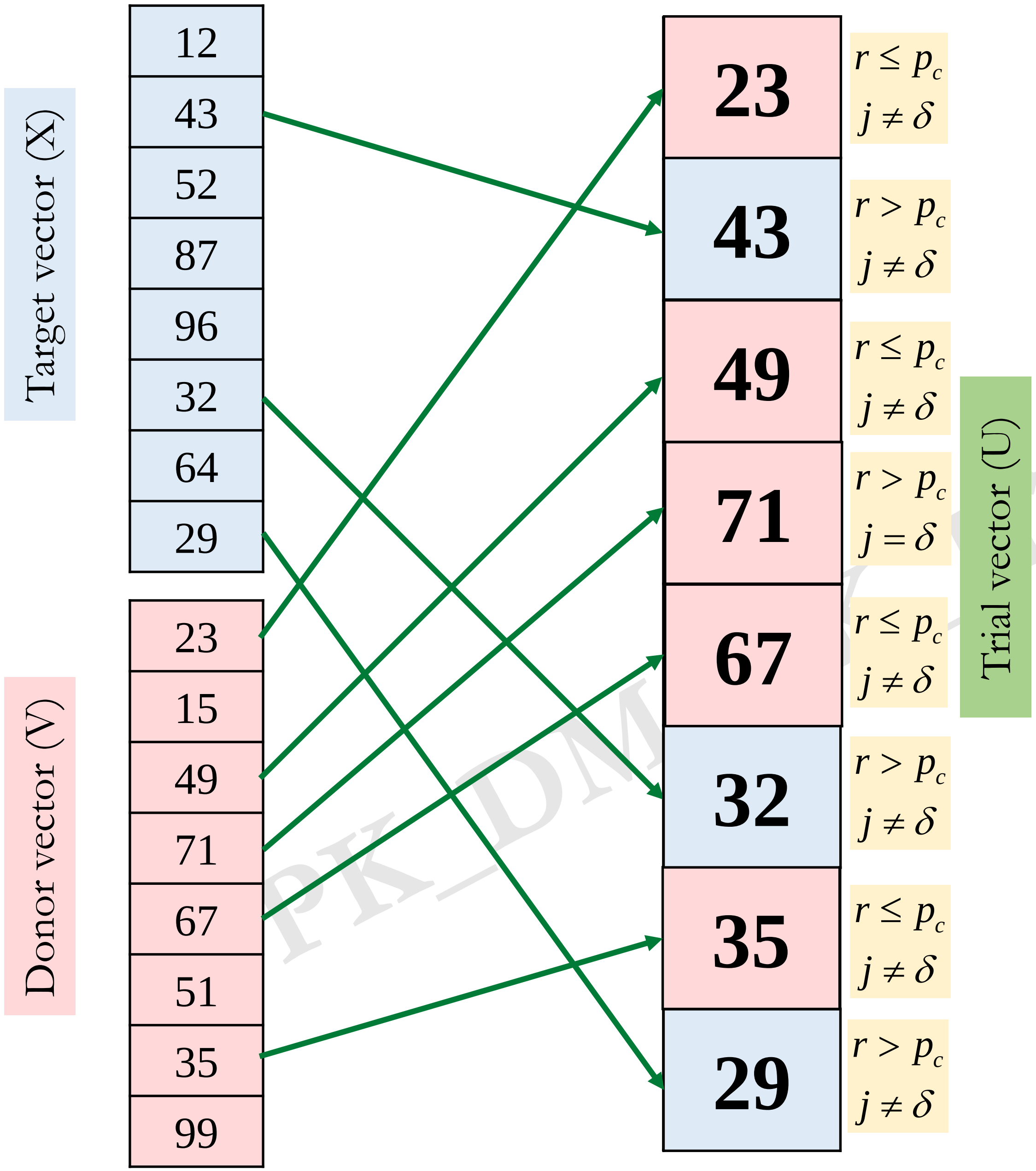
Illustration of binomial crossover



δ = 4 and p_c=0.8

$$u^j = \begin{cases} v^j & \text{if } r \le p_c \text{ OR } j = \delta \\ x^j & \text{if } r > p_c \text{ AND } j \ne \delta \end{cases}$$

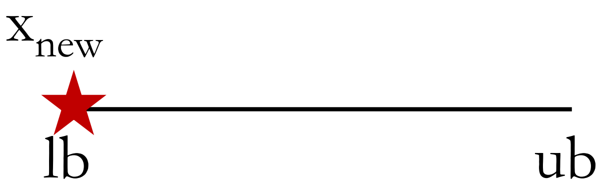
Illustration of binomial crossover



$\delta = 4$ and $p_c = 0.8$

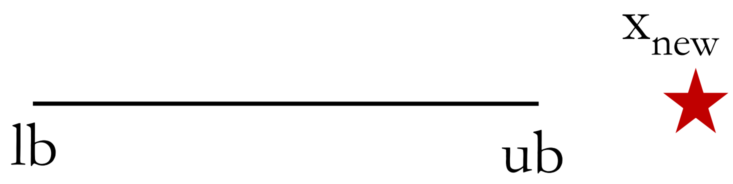
$$u^j = \begin{cases} v^j & \text{if } r \le p_c \text{ OR } j = \delta \\ x^j & \text{if } r > p_c \text{ AND } j \ne \delta \end{cases}$$

Bounding of offspring



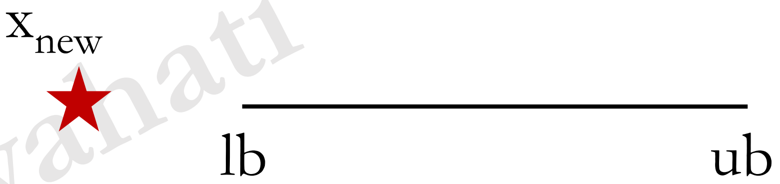
x_{new} is within bounds

No bounding required

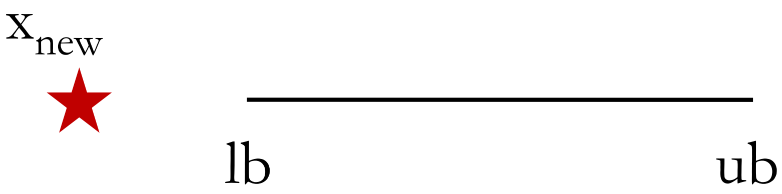


x_{new} violates the upper bound

Shift x_{new} to upper bound



x_{new} violates the lower bound



Shift x_{new} to lower bound

Selection

- Evaluate the fitness function of all offspring (f_U).

- Population is updated using greedy selection.

$$\left.\begin{aligned} X_i &= U_i \\ f_i &= f_{U_i} \end{aligned}\right\} \; if \;\; f_{U_i} < f_i$$

X and f remains the same if $f_{U_i} > f_i$

- Greedy selection is performed only after the generation of offspring by all solutions.

Pseudocode of DE

Inputs: Fitness function, lb, ub, N_p, T, F, p_c

1. Initialize a random population (P)
2. Evaluate fitness (f) of P ⬅ FE = N_p

```
for t = 1 to T
    for i = 1 to Np
        Generate the donor vector (Vi) using mutation
        Perform crossover to generate offspring (Ui)
    end
    for i = 1 to Np
        Bound Ui
        Evaluate the fitness (fUi) of Ui   <- FE = 1
        Perform greedy selection using fUi and fi to update P
    end
end
```

Generation

Selection

In one iteration, $\#FE = N_p$

For T iterations, $\#FE = N_p + N_p T$

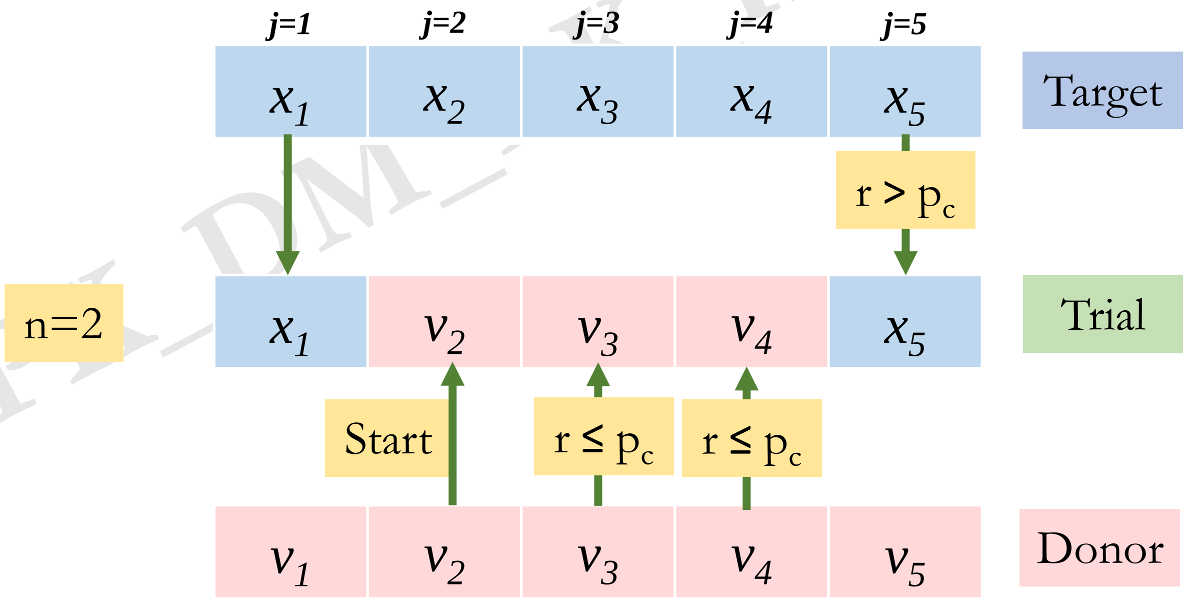
$$V = X_{r_1} + F\left(X_{r_2} - X_{r_3}\right)$$

$$u^j = \begin{cases} v^j & \text{if } r \le p_c \text{ or } j = \delta \\ x^j & \text{if } r > p_c \text{ and } j \ne \delta \end{cases}$$

$$\left.\begin{aligned} X_i &= U_i \\ f_i &= f_{U_i} \end{aligned}\right\} \text{if } \; f_{U_i} < f_i$$

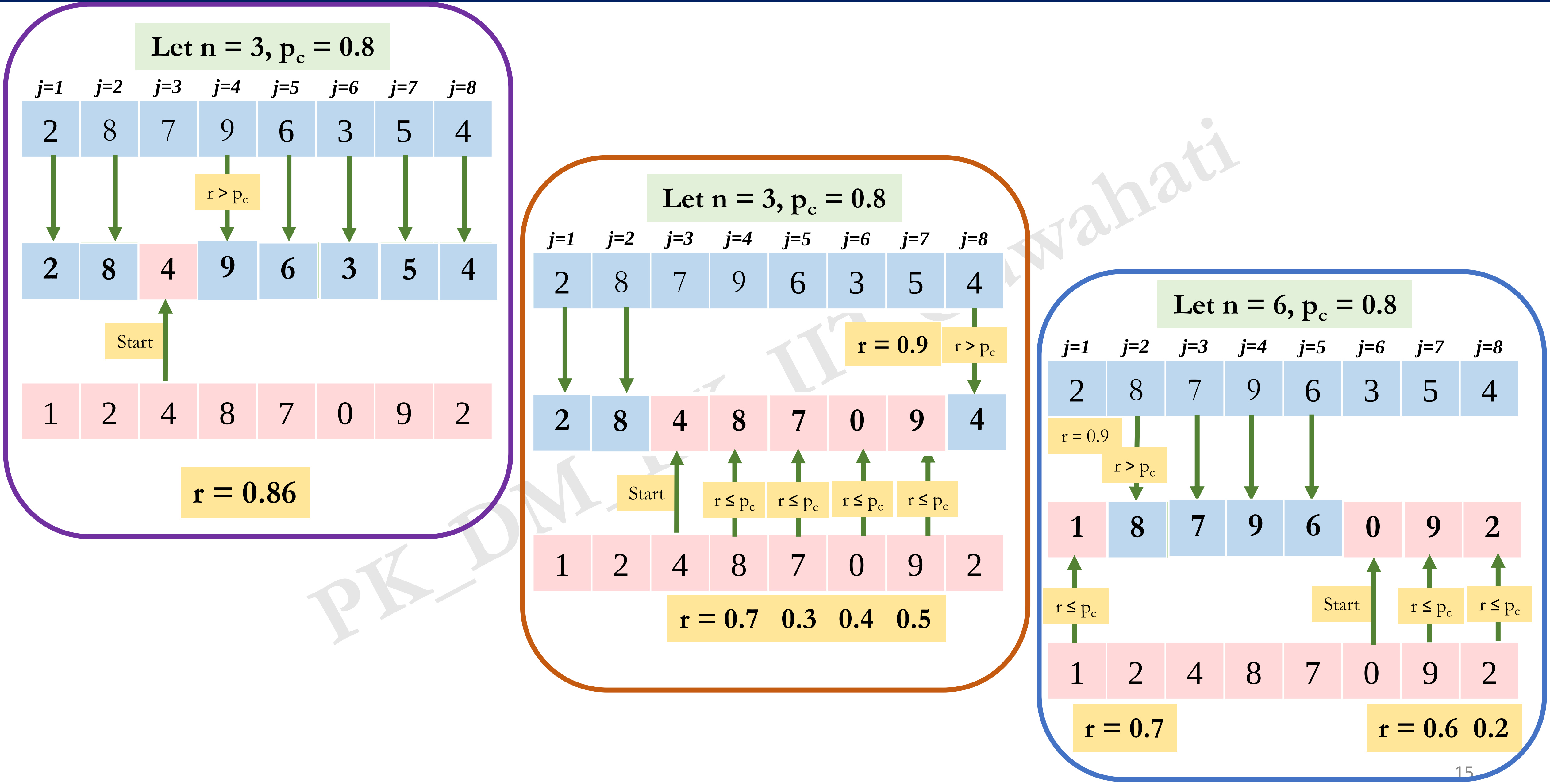
Exponential crossover

- Randomly choose an integer (n) between 1 and *D*.
- Copy the n^{th} variable from donor as n^{th} variable of trial vector.
- For the subsequent variables, generate a random number between 0 and 1, till $r > p_c$.
- If $r \leq p_c$, copy the variable from donor to trial vector.
- If r > pc, copy the remaining variables from target to trial vector.



Differential Evolution, A Practical Approach to Global Optimization, Natural Computing Series, Springer Berlin Heidelberg New York, 1998

Illustration of exponential crossover



Mutation strategies (DE/x/y/z)

- DE: Differential Evolution
- x: Vector to be mutated
- y: number of difference vectors (random solutions) required for mutation
- z: type of crossover scheme to be used (can be either exponential or binomial crossover)

Strategy	Expression for donor vector	Minimum N_p
DE/rand/1	$V = X_{r_1} + F\left(X_{r_2} - X_{r_3}\right)$	4
DE/best/1	$V = X_{best} + F\left(X_{r_1} - X_{r_2}\right)$	3
DE/rand/2	$V = X_{r_1} + F\left(X_{r_2} - X_{r_3}\right) + F\left(X_{r_4} - X_{r_5}\right)$	6
DE/best/2	$V = X_{best} + F\left(X_{r_1} - X_{r_2}\right) + F\left(X_{r_3} - X_{r_4}\right)$	5
DE/target-to-best/1	$V = X_i + F\left(X_{best} - X_i\right) + F\left(X_{r_1} - X_{r_2}\right)$	3

Working of DE: Sphere function

Consider min $f(x)=\sum_{i=1}^{4} x_i^2;\quad 0 \le x_i \le 10,\quad i=1,2,3,4$

$f(x)=x_1^2+x_2^2+x_3^2+x_4^2$

Decision variables: x_1, x_2, x_3 and x_4 and dimension D = 4

- **Step 1:** Fix the population size, number of generation, crossover probability, scaling factor

$$N_p = 5,\ T = 10,\ p_c = 0.8,\ F = 0.85$$

- **Step 2:** Generate random solutions within the domain of the decision variables

$$P=\begin{bmatrix} 4 & 0 & 1 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 1 & 2 & 8 & 3 \end{bmatrix} \qquad f=\begin{bmatrix} 81 \\ 140 \\ 35 \\ 102 \\ 78 \end{bmatrix}$$

Working of DE: first solution

$p_c = 0.8$, $F = 0.85$

$$P = \begin{bmatrix} 4 & 0 & 1 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 1 & 2 & 8 & 3 \end{bmatrix} \quad f = \begin{bmatrix} 81 \\ 140 \\ 35 \\ 102 \\ 78 \end{bmatrix}$$

$$V = X_{r_1} + F\left(X_{r_2} - X_{r_3}\right)$$

- Step 3: Generate 4 random integers between 1 and N_p

  Let $r_1 = 4$ $r_2 = 2$ $r_3 = 3$ $\delta = 1$

- Step 4: Determine donor vector (Mutation)

$$V_1 = X_4 + F\left(X_2 - X_3\right)$$
$$= [2 \;\; 1 \;\; 4 \;\; 9] + 0.85 \times \left([3 \;\; 1 \;\; 9 \;\; 7] - [0 \;\; 3 \;\; 1 \;\; 5]\right)$$
$$= [2 \;\; 1 \;\; 4 \;\; 9] + [2.55 \;\; -1.7 \;\; 6.8 \;\; 1.7]$$
$$= [4.55 \;\; -0.7 \;\; 10.8 \;\; 10.7]$$

- Step 5: Generate *D* random numbers

  Let r = [0.3 0.9 0.2 0.6]

DE: first solution

$p_c = 0.8, F = 0.85, \delta = 1$

- Step 6: Determine trial vector

j	Target Vector	Donor Vector	r	$(r \le p_c)$	$\delta \ne j$	$\delta = j$	Trial Vector
1	4	4.55	0.3	✓	–	–	4.55
2	0	-0.7	0.9	✖	✓	✖	0
3	1	10.8	0.2	✓	–	–	10.8
4	8	10.7	0.6	✓	–	–	10.7

$$P = \begin{bmatrix} 4 & 0 & 1 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 1 & 2 & 8 & 3 \end{bmatrix} \quad f = \begin{bmatrix} 81 \\ 140 \\ 35 \\ 102 \\ 78 \end{bmatrix}$$

$$u^j = \begin{cases} v^j & \text{if } r \le p_c \text{ OR } j = \delta \\ x^j & \text{if } r > p_c \text{ AND } j \ne \delta \end{cases}$$

- Step 7: Check bounds, bound if violates

$$U_1 = [\,4.55 \quad 0 \quad 10.8 \quad 10.7\,] \Rightarrow U_1 = [\,4.55 \quad 0 \quad 10 \quad 10\,]$$

$$0 \le x_i \le 10$$

$$x = lb \quad \text{if } x < lb$$
$$x = ub \quad \text{if } x > ub$$

First iteration

Value of r_1, r_2, r_3, r, δ for first iteration

$$V = X_{r_1} + F\left(X_{r_2} - X_{r_3}\right)$$

$p_c = 0.8$, $F = 0.85$

i	Target (P)	r_1	r_2	r_3	Donor (V)	r	δ	Trial (U)
1	[4 0 1 8]	4	2	3	[4.55 -0.7 10.8 10.7]	[0.3 0.9 0.2 0.6]	1	[4.55 0 10 10]
2	[3 1 9 7]	5	1	3	[4.4 -0.55 8 5.55]	[0.3 0.2 0.6 0.4]	4	[4.4 0 8 5.55]
3	[0 3 1 5]	4	2	1	[1.15 1.85 10.8 8.15]	[0.2 0.5 0.4 0.3]	4	[1.15 1.85 10 8.15]
4	[2 1 4 9]	5	3	2	[-1.55 3.7 1.2 1.3]	[0.8 0.3 0.6 0.2]	1	[0 3.7 1.2 1.3]
5	[1 2 8 3]	2	4	1	[1.3 1.85 11.55 7.85]	[0.7 0.5 0.9 0.2]	3	[1.3 1.85 8 7.85]

- Step 8: Evaluate the fitness of bounded trial vectors

$$u^j = \begin{cases} v^j & if\ r \le p_c\ OR\ j = \delta \\ x^j & if\ r > p_c\ AND\ j \ne \delta \end{cases}$$

$$U = \begin{bmatrix} 4.55 & 0 & 10 & 10 \\ 4.4 & 0 & 8 & 5.55 \\ 1.15 & 1.85 & 10 & 8.15 \\ 0 & 3.7 & 1.2 & 1.3 \\ 1.3 & 1.85 & 8 & 7.85 \end{bmatrix} \qquad f_U = \begin{bmatrix} 220.70 \\ 114.16 \\ 171.17 \\ 16.82 \\ 130.73 \end{bmatrix}$$

$$f(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2$$

DE: Greedy selection of first iteration

- Step 9: After each iteration, perform greedy solution and update population

$$P = \begin{bmatrix} 4 & 0 & 1 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 1 & 2 & 8 & 3 \end{bmatrix} \quad f = \begin{bmatrix} 81 \\ 140 \\ 35 \\ 102 \\ 78 \end{bmatrix} \quad U = \begin{bmatrix} 4.55 & 0 & 10 & 10 \\ 4.4 & 0 & 8 & 5.55 \\ 1.15 & 1.85 & 10 & 8.15 \\ 0 & 3.7 & 1.2 & 1.3 \\ 1.3 & 1.85 & 8 & 7.85 \end{bmatrix} \quad f_U = \begin{bmatrix} 220.70 \\ 114.16 \\ 171.17 \\ 16.82 \\ 130.73 \end{bmatrix}$$

Population for next iteration

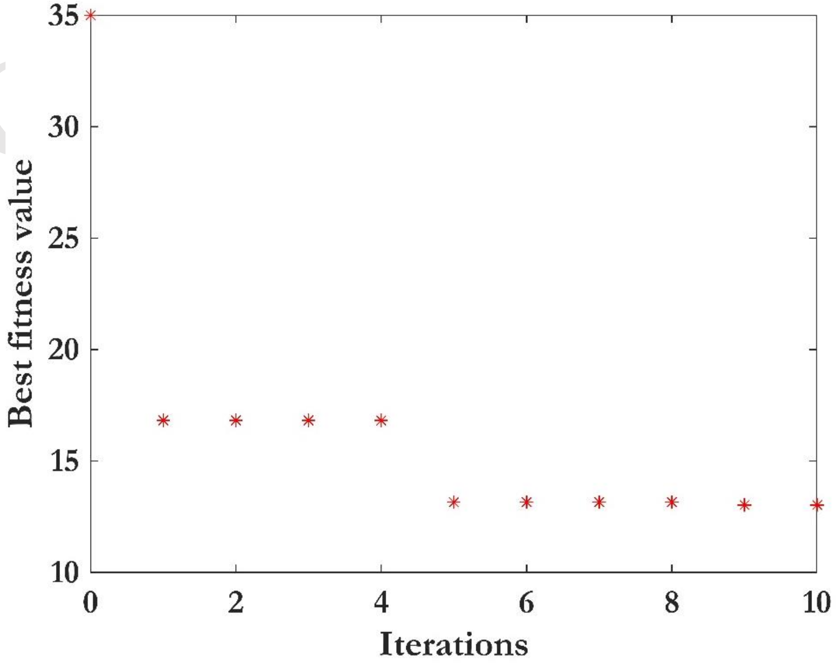
$$P = \begin{bmatrix} 4 & 0 & 1 & 8 \\ 4.4 & 0 & 8 & 5.55 \\ 0 & 3 & 1 & 5 \\ 0 & 3.7 & 1.2 & 1.3 \\ 1 & 2 & 8 & 3 \end{bmatrix} \quad f = \begin{bmatrix} 81 \\ 114.16 \\ 35 \\ 16.82 \\ 78 \end{bmatrix}$$

DE: Satisfaction of termination criterion

$$\min \quad f(x) = \sum_{i=1}^{4} x_i^2; \quad 0 \le x_i \le 10, \quad i = 1,2,3,4$$

After the completion of 10 iterations

$$P = \begin{bmatrix} 0 & 4.18 & 1.2 & 0 \\ 0 & 3.51 & 1.2 & 0 \\ 0 & 3 & 1.2 & 1.65 \\ 0 & 3.10 & 1.2 & 1.40 \\ 0 & 3.68 & 1.2 & 0 \end{bmatrix} \qquad f = \begin{bmatrix} 18.89 \\ 13.73 \\ 13.15 \\ 13.02 \\ 15.01 \end{bmatrix}$$



The minimum value of the function is **0**

Comparison between TLBO, PSO and DE

	TLBO	PSO	DE
Phases	Teacher, Learner	No phases (Position and velocity update)	No phases (Mutation and crossover)
Convergence	Monotonic	Monotonic (with g_{best} & p_{best})	Monotonic
Parameters	Population size, termination criteria	Population size, termination criteria, w, c_1 and c_2	Population size, termination criteria, P_c, F
Generation of new solution	Using other solutions, mean and best solution (part of population)	Using velocity vector, p_{best} and g_{best} (need not be the part of population)	Using other solutions (best solution is part of population)
Solution update in an iteration	Twice	Once	Once
Selection	Greedy	Always accept new solution into the population (μ, λ)	Greedy
#FE	$N_p + 2N_pT$	$N_p + N_pT$	$N_p + N_pT$

Further reading

- Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, **Journal of Global Optimization**, 11, 341, 1997
- JADE: Adaptive Differential Evolution With Optional External Archive, **IEEE Transactions on Evolutionary Computation**, 13(5), 945-958, 2009
- Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization, **IEEE Transactions on Evolutionary Computation**, 13(2), 398-417, 2009
- Recent advances in differential evolution – An updated survey, **Swarm and Evolutionary Computation**, 27, 1-30, 2016
- Differential evolution with multi-population based ensemble of mutation strategies, **Information Sciences**, 329, 329-345, 2016
- Review of Differential Evolution population size, **Swarm and Evolutionary Computation**, 32, 1-24, 2017

Thank You !!!