New Topic: “Differential Evolution”

“Differential Evolution”

For Global Optimization of Continuous Variables

(winner in an international competition among evolutionary algorithms in 1996)

Journal Paper Reference

- Storn, R. and K. Price, "Differential Evolution —A Simple and Efficient Heuristic for Global Optimization over Continouous Spaces", *Jn. of Global Optimization*, 11: 341-359, 1997.
- Since that time Storn et al. have written a book on this method and other books have also been written on the method.
- You are not required for the course to read any materials on Differential Evolution beyond what is in the lecture notes.

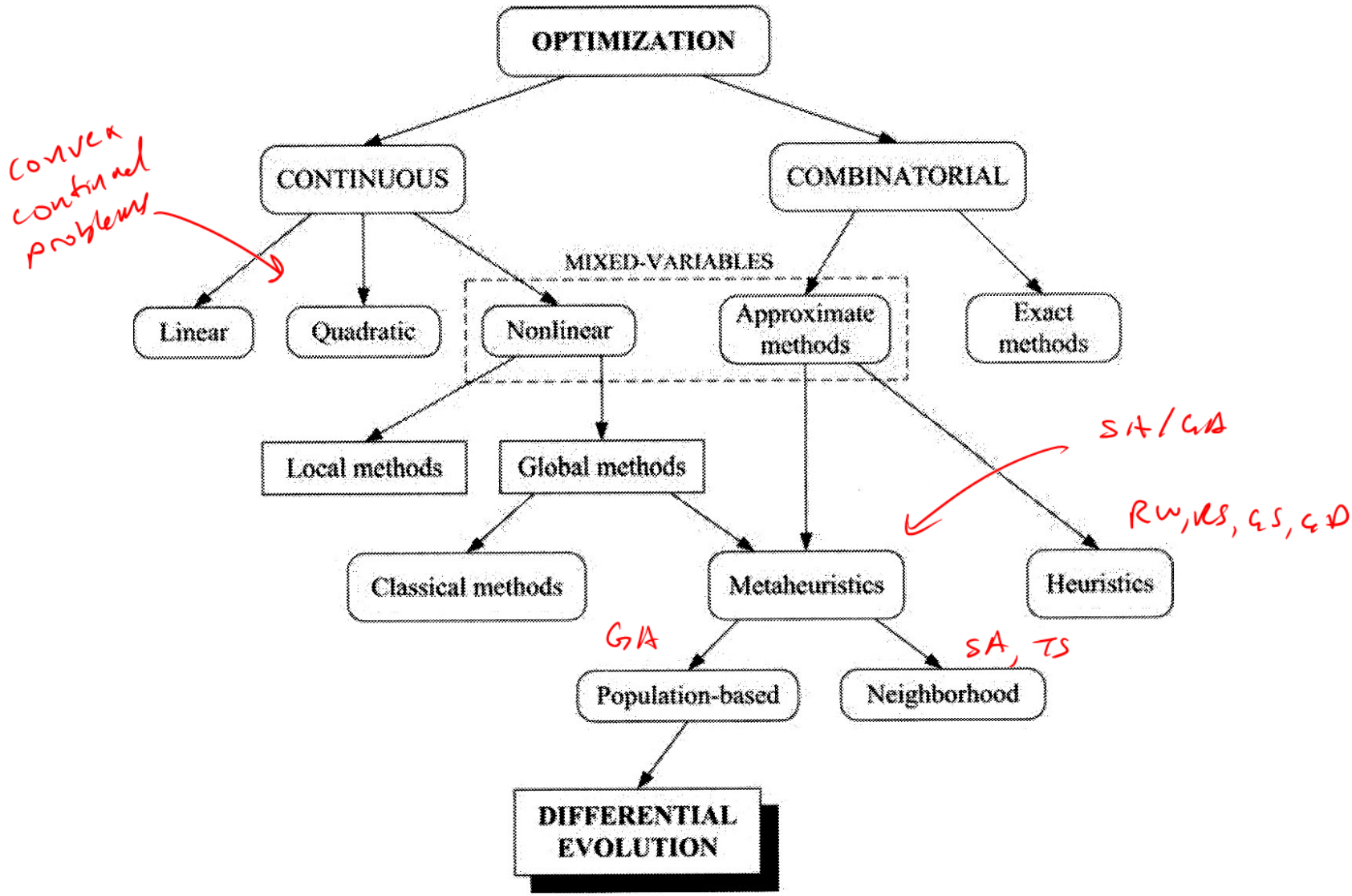


Fig. 1.2. A simple classification scheme of optimization methods: position of differential evolution.

Problem

- Assume you want to find the value of the **D** dimensional real vector x* such that

- f(x*) = Maximum f(x)
- Where L $\leq$x$\leq$H (vector constraint on each component)

Characteristics of Differential Evolution

same as GA

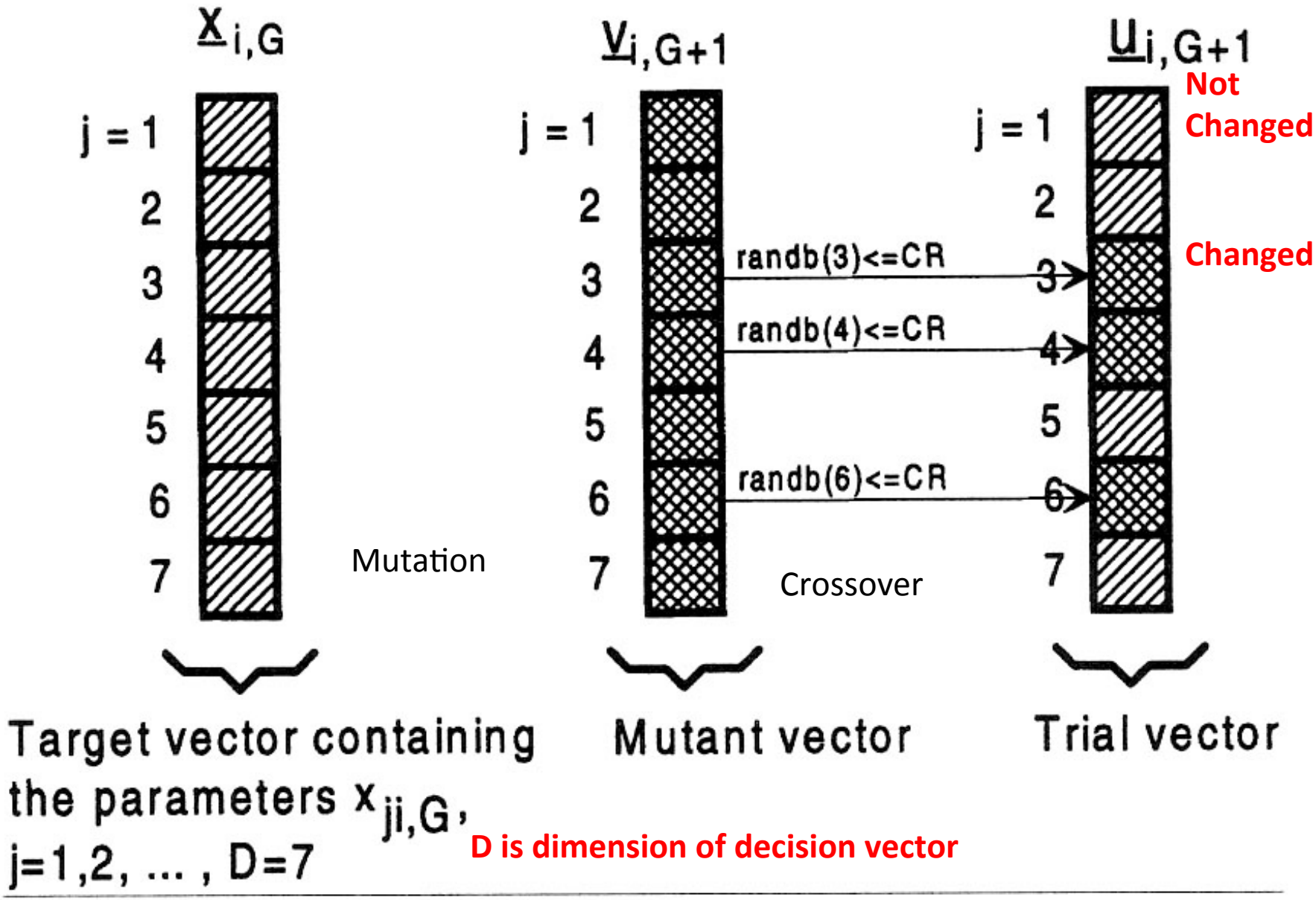
- Population based, real valued decision vector
- In generating a child from a parent, some of the components of the parent vector remain unchanged in the child.
- For the components that are changed, the new values depend on a weighted linear function of the same components in other individuals in the parent population. → this is the new thing
- Advantage that it is a relatively simple code. (C Code is given in Storn & Price 1997 paper.)

→ R.I.S.S.

DE Algorithm Parameter
these values are selected by the user

- F– a constant used in “mutation” process. ($0<F\underline{<2}$) *Typical value is .5*
- NP- size of population. *Typical value is 20*
- CR- a constant used in “crossover” *Typical value is .1*

Mutation Example for 7 dimensional decision vector



D is dimension of decision vector

Differential Evolution- Definitions

G = current generation i = population member j = vector cell in population member

- Subscript *i* is for the ith individual decision vector and *j* is for the element number in the vector. ***G*** is for the generation number
- $\boldsymbol{x}_{jiG}$ = original parent (D dimensional vector)
- $\boldsymbol{v}_{jiG}$ = "mutant" (based on weighted average of several other elements of the population)
- $\boldsymbol{u}_{jiG}$ =individual after mutation and crossover. Some of the elements are same as *x* and some of the elements are the same as *v*.
- In the paper description, dropping the j subscript is used to denote the D-dimensional vector. So $\boldsymbol{x}_{jG}$ is the decision vector with **D** elements.

these are actually vector positions in each vector

Mutation

- Pick three random (integer) numbers (uniformly distributed in [1,…,NP], where NP is the population size. Call them *r1, r2, and r3*.
- Then we generate a mutant from a combination of $x_{r1,G}$, $x_{r2,G}$, and $x_{r3,G}$
- $v_{i,G+1} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G})$ *(vector addition)*
- The next figure shows an illustration for D=2 of the 2 dimensional parent vector $x_{1,G}$ and the other parent vectors

  $x_{r1,G}$, $x_{r2,G}$, and $x_{r3,G}$

Mutation in 2 Dimensions

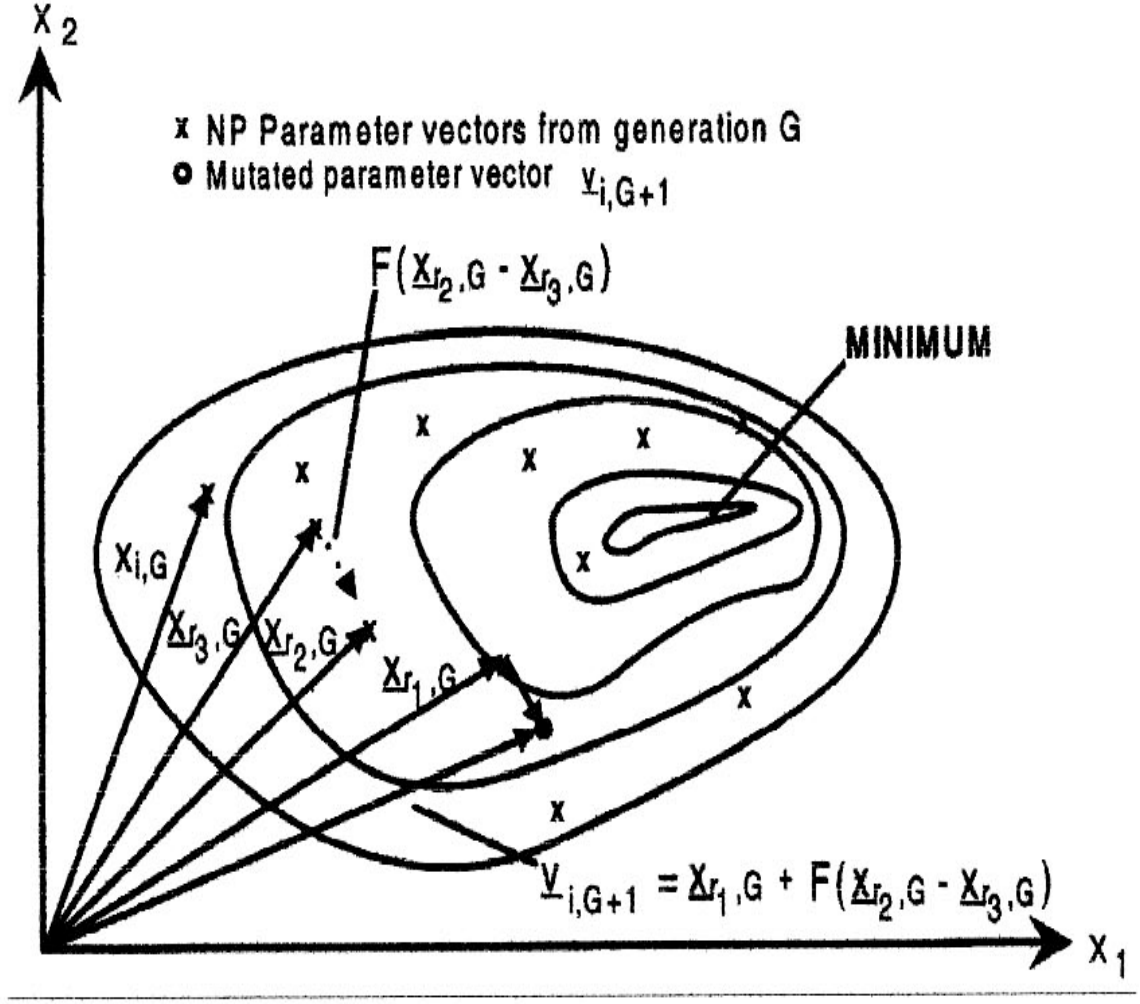


Figure 1. An example of a two-dimensional cost function showing its contour lines and the process for generating $v_{i,G+1}$.

Crossover

- The purpose of Crossover is to increase the diversity of the set of decision vector in the Child population (since the mutation alone is only a linear combination of parent population)
- $\boldsymbol{u_{i,G}}$ is the D dimensional "trial" vector after both mutation and crossover.
- $u_{i,G}$ is determined by D random number called **randb(j), j=1,…,D and an additional random number called rnbr(i).** The randb(j) are uniform randomly distributed among real numbers in [0,1] and are independent of random numbers selected for the other $u_{k,G}$ *, k not equal to i.*
- rnbr(j) is uniformly distributed among the integers 1,…,D and it indicates one of the elements of the mutant that is definitely carried over to the u vector. This is to insure that u does not equal x in all elements.

Crossover

- Then for the jth element $u_{ji,G}$ in the vector $u_{i,G}$

$u_{ji,G} = v_{ji,G}$ if randb(j) $\leq$ CR or if j=rnbr(i) (4a)

$u_{ji,G} = x_{ji,G}$ otherwise (4b)

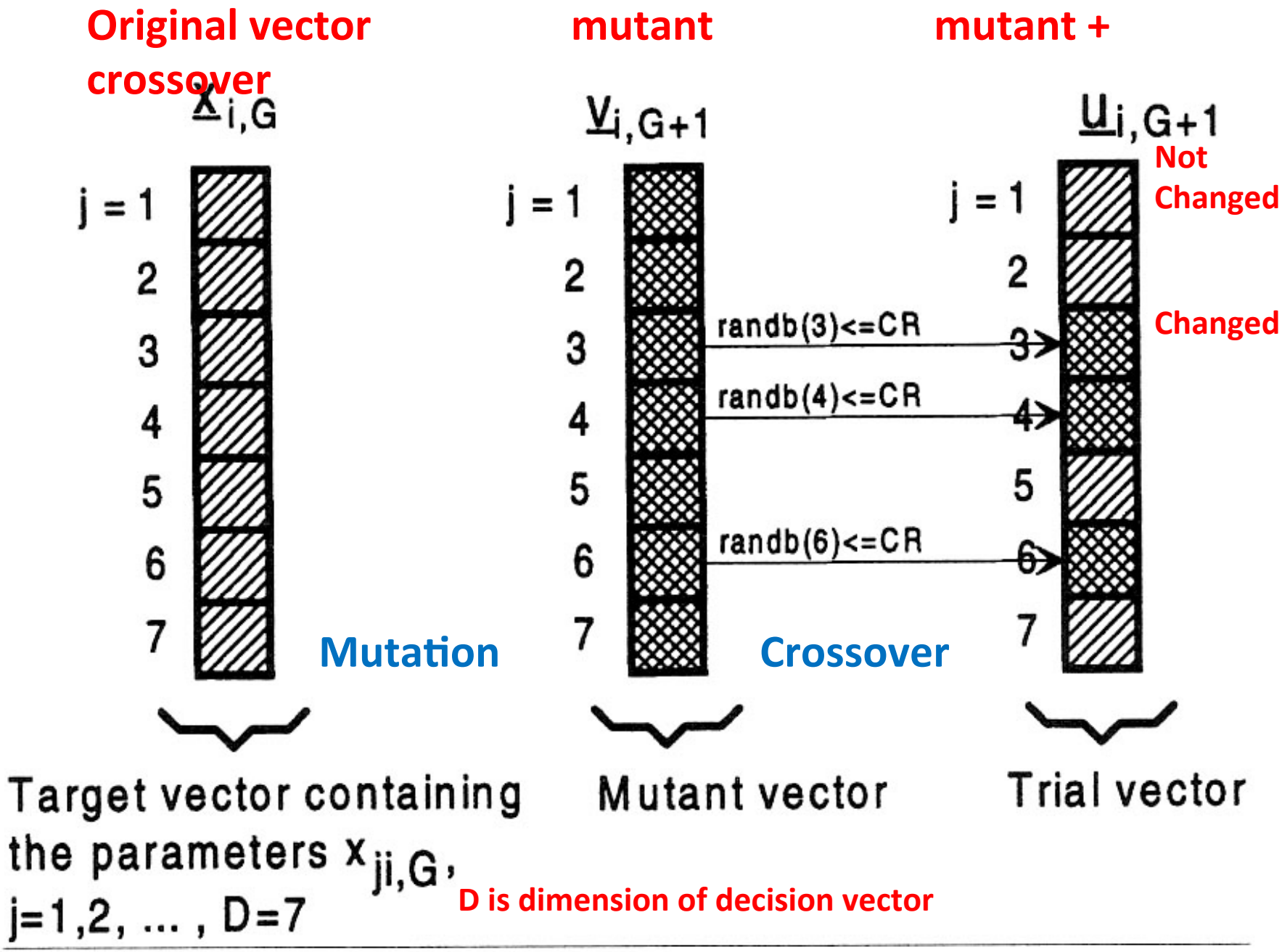
element index

generation

again $u_{j,i,G}$

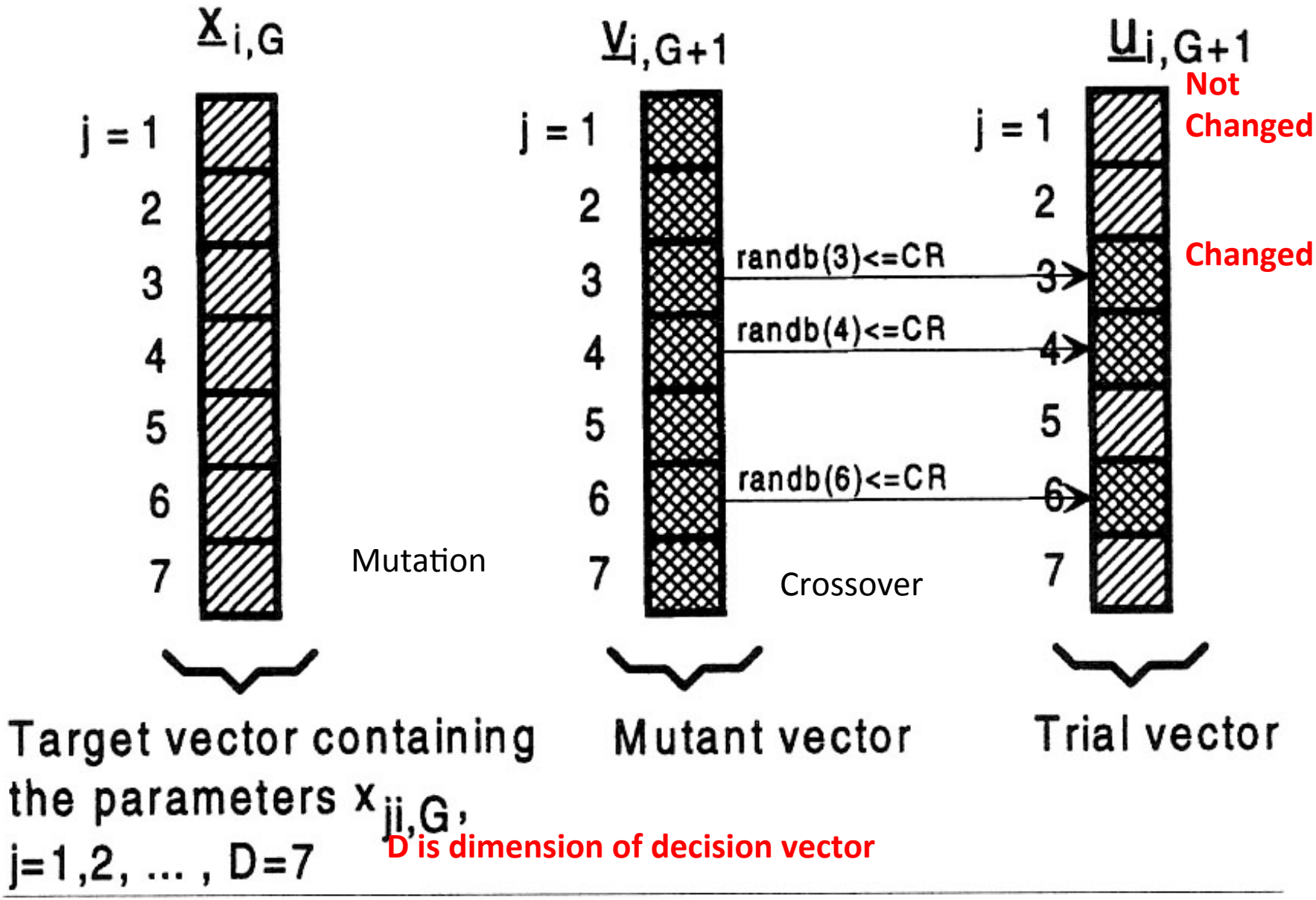
population index

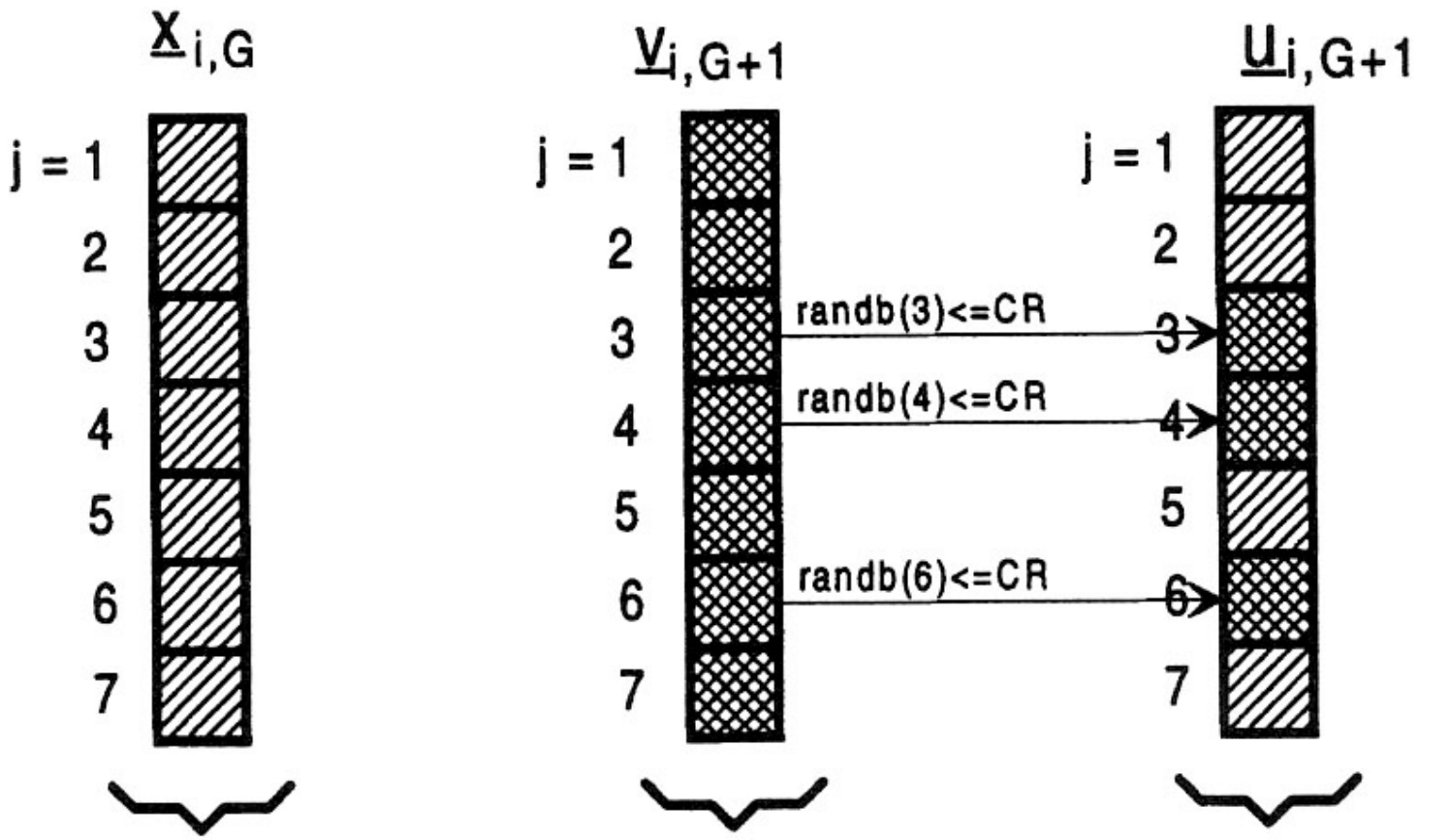
Hence as is shown on (repeated) next figure slide. the components of the decision vector $u_{i,G}$ are a mixture of components from $v_{i,G}$ (mutated vector) and $x_{i,G}$ (original parent vector) and that mixture is determined by the random variables.


Original vector
crossover
mutant
mutant +
$\underline{x}_{i,G}$
$\underline{v}_{i,G+1}$
$\underline{u}_{i,G+1}$
Not
Changed
Changed
j = 1
2
3
4
5
6
7
randb(3)<=CR
randb(4)<=CR
randb(6)<=CR
Mutation
Crossover
Target vector containing
the parameters $x_{ji,G}$,
j=1,2, ... , D=7
Mutant vector
Trial vector
D is dimension of decision vector

- Note that there is a (independent identically distributed) random variable randb(j) for each component j.
- Note also that that there is a random variable rnbr(i) to guarantee that at least one component)= rnbr(i) is perturbed. in spite of CR
- For example if CR=0, there will be at one component that is perturbed.

Example for 7 dimensional decision vector





Target vector containing the parameters $x_{ji,G}$, $j=1,2, \ldots , D=7$

Mutant vector

Trial vector

SELECTION
Reproduction Success based on Fitness

- So far the fitness of a vector has not been incorporated.
- Now we do a tournament selection to determine if the original ith parent will become a child or if the ith trial vector becomes a child.
- The fitness of f(parent)=f($x_{i,G}$) versus f(trial)=f($u_{i,G}$) is compared and the most fit version is put in the child population. So since this is a maximization problem
- If f($u_{i,G}$) > f($x_{i,G}$), then the trail vector $u_{i,G}$ is added to the child population. Otherwise the original parent $x_{i,G}$ is added to the child population.
- Then the child population becomes the parents for generation G+1.

- Hence the process is that we are going to compare one vector $\boldsymbol{x}_{iG}$ in the population against a combination of three other vectors ($x_{r1,G}$,$x_{r2,G}$, *and* $x_{r3,G}$) in the population.
- The paper compares some solutions as given in the following figures.

Comparison of DE to BGA (binary GA) and EASY algorithm on 15 problems
BGA=binary GA, EASY= H.-M. Voigt (1992 book by Springer)

Table 2. Averaged number of function evaluations (nfe) required to find the global minimum. NA stands for "not available".

$f_i(x)$	D	nfe			DE-Settings		
i		BGA	EASY	DE/rand/1/bin	NP	F	CR
11	30	NA	27,111	16,907	20	0.5	0.1
11	100	NA	104,520	56,145	20	0.5	0.1
12	10	NA	9,626	4,269	15	0.5	0.1
12	30	NA	39,333	12,859	15	0.5	0.1
13	20	3,608	6,098	12,971	25	0.5	0
13	100	25,040	45,118	73,620	25	0.5	0
14	20	66,000	26,700	8,691	20	0.5	0.1
14	100	361,722	77,250	31,796	20	0.5	0.1
15	30	19,420	13,997	12,481	20	0.5	0.1
15	100	53,860	57,628	36,801	20	0.5	0.1

Comparion of DE to SDE (Stochastic Differential Equations for Global Optimization)

Table 3. Averaged number of function evaluations (nfe) required to find the global minimum.

$f_i(x)$	nfe		DE-Settings		
i	SDE	DE/rand/1/bin	NP	F	CR
16	3,184	503	20	0.5	0
17	26,893	499	20	0.5	0
18	241,215	3,137	20	0.5	0
19, β=0.5	8,755	4,854	40	1.0	0
19, β=1.0	97,761	4,428	40	1.0	0
20	5,393	927	20	0.5	0
21, D=2	84,782	722	20	0.5	0
21, D=3	19,041	1,073	20	0.5	0
21, D=4	18,942	1,424	20	0.5	0
22, D=5	18,433	2,084	20	0.5	0
22, D=8	136,061	3,347	20	0.5	0
22, D=10	49,701	4,165	20	0.5	0

23, D=2	9,492	715	20	0.5	0
23, D=3	19,114	1,093	20	0.5	0
23, D=4	35,139	1,499	20	0.5	0
24, D=5	53,398	1,882	20	0.5	0
24, D=6	15,534	2,295	20	0.5	0
24, D=7	16,542	2,701	20	0.5	0
25	6,751	273	20	0.5	0
26	3,402	650	20	0.5	0
27	10,286	621	20	0.5	0
28, n= −m=1	4,791	907	20	0.5	0
28, n= −m=2	3,037	812	20	0.5	0
28, n= −m=3	5,028	778	20	0.5	0
28, n= −m=4	14,710	754	20	0.5	0
28, n= −m=5	51,285	751	20	0.5	0
28, n= −m=6	17,610	761	20	0.5	0
29	15,102	7,053	20	0.5	0
30	48,802	1,266	30	0.5	1.0

Conclusions

- The differential evolution algorithm described here has the advantage of being relatively simple and easily coded.
- The results presented in the 1997 and the 1996 competition are favorable to DE. (Do you have some comments on those results?)
- More variants of DE are being developed, but it is still not as widely used as GA.

Algorithm 1 Famous Differential Evolution

Require: D – problem dimension (optional)
NP, F, Cr – control parameters
GEN – stopping condition
L , H – boundary constraints

Initialize population $Pop_{ij} \leftarrow rand_{ij}[L, H]$ and Evaluate fitness $Fit_j \leftarrow f(Pop_j)$
for $g = 1$ to GEN **do**
  for $j = 1$ to NP **do**
    Choose randomly $r_{1,2,3} \in [1, \ldots, NP]$, $r_1 \neq r_2 \neq r_3 \neq j$
    Create trial individual $X \leftarrow \mathcal{S}(r, F, Cr, Pop)$
    Verify boundary constraints **if** $(x_i \notin [L, H])$ $\quad x_i \leftarrow rand_i[L, H]$
    Select better solution (X or Pop_j), and update $iBest$ if required
  end for
end for