Differential Evolution

Winter 2011

Basics of Differential Evolution

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

DE

- 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. Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, Berlin, 2005.
  - http://www.icsi.berkeley.edu/~storn/code.html

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

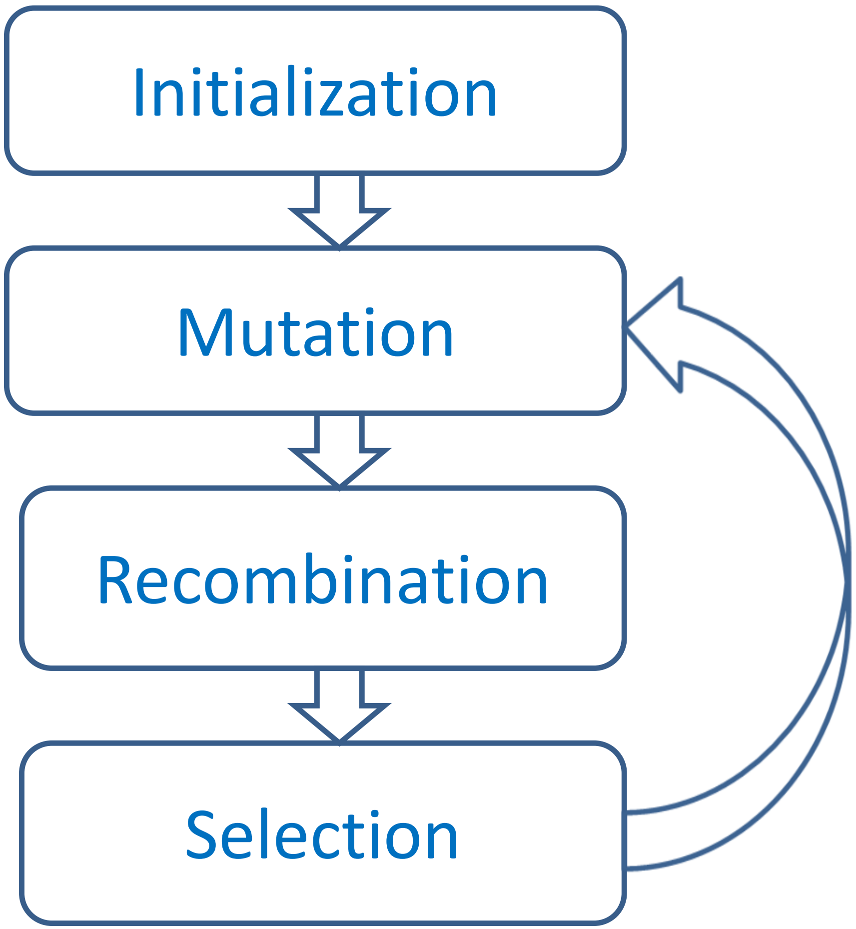
  $x_{i,t}=[x_{1,i,t}, x_{2,i,t}, \ldots x_{D,i,t}]$ $i=1,2,\ldots N$

  or

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

  Where *t* is the generation (iteration)

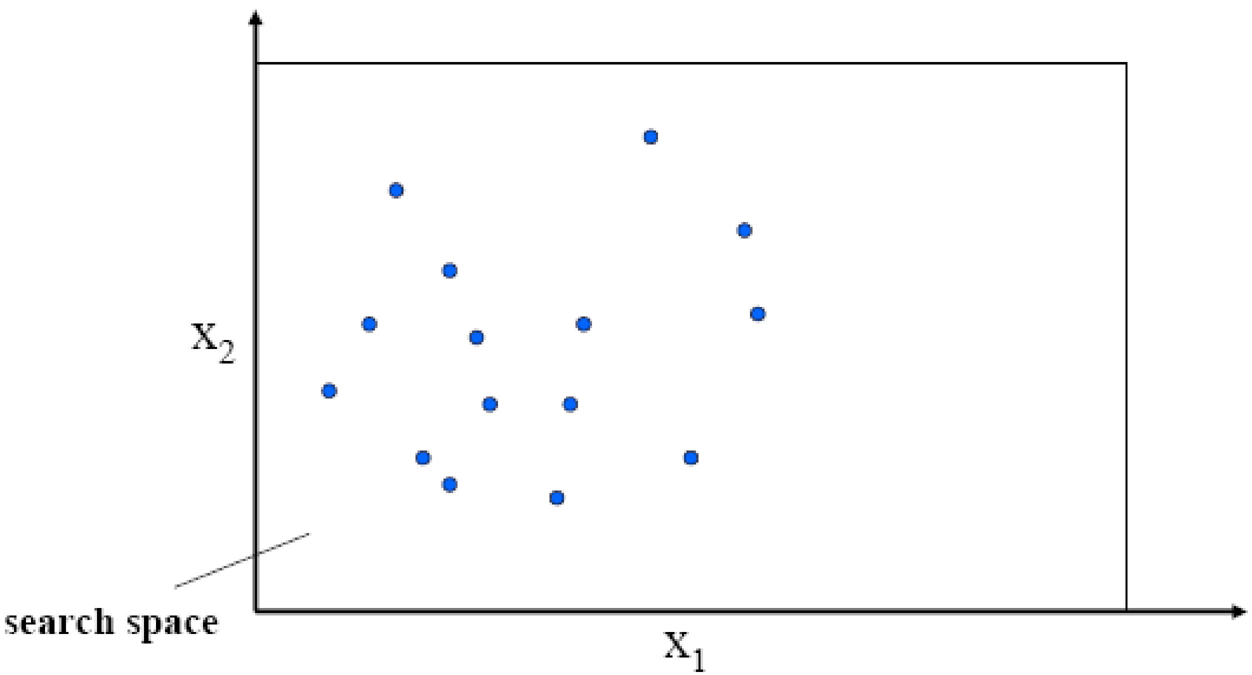
Flowchart



Initialization

- 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

Initialization



Mutation

- To expand the search space
- 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

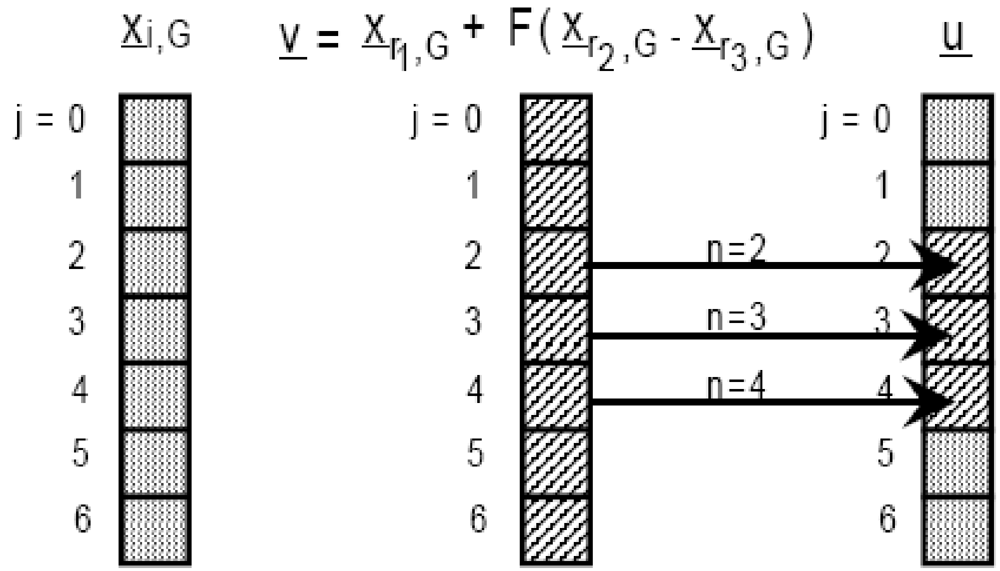
- To increase the diversity
- At each iteration, mix vectors x and v (similar to crossover)

  $u_j = v_j$ for j=mod(n,D), mod(n+1,D),…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

Another expression of 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}

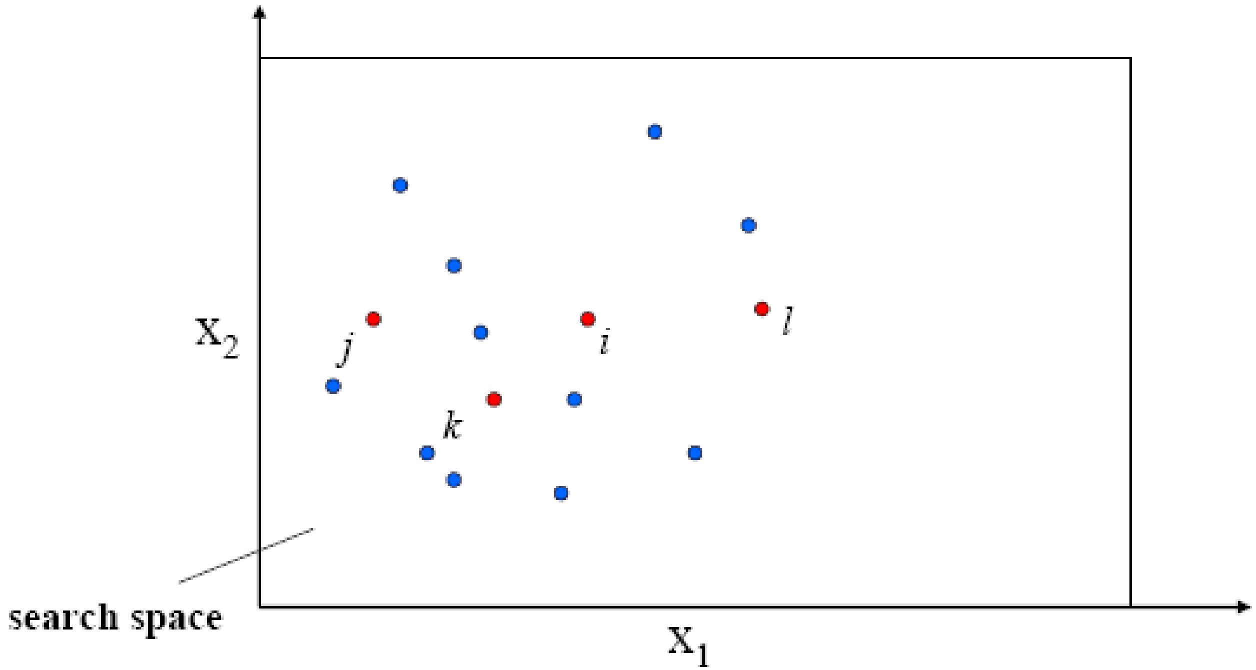
i (solution index)=*1,2,…N;*

j (variable index)=*1,2,…D*

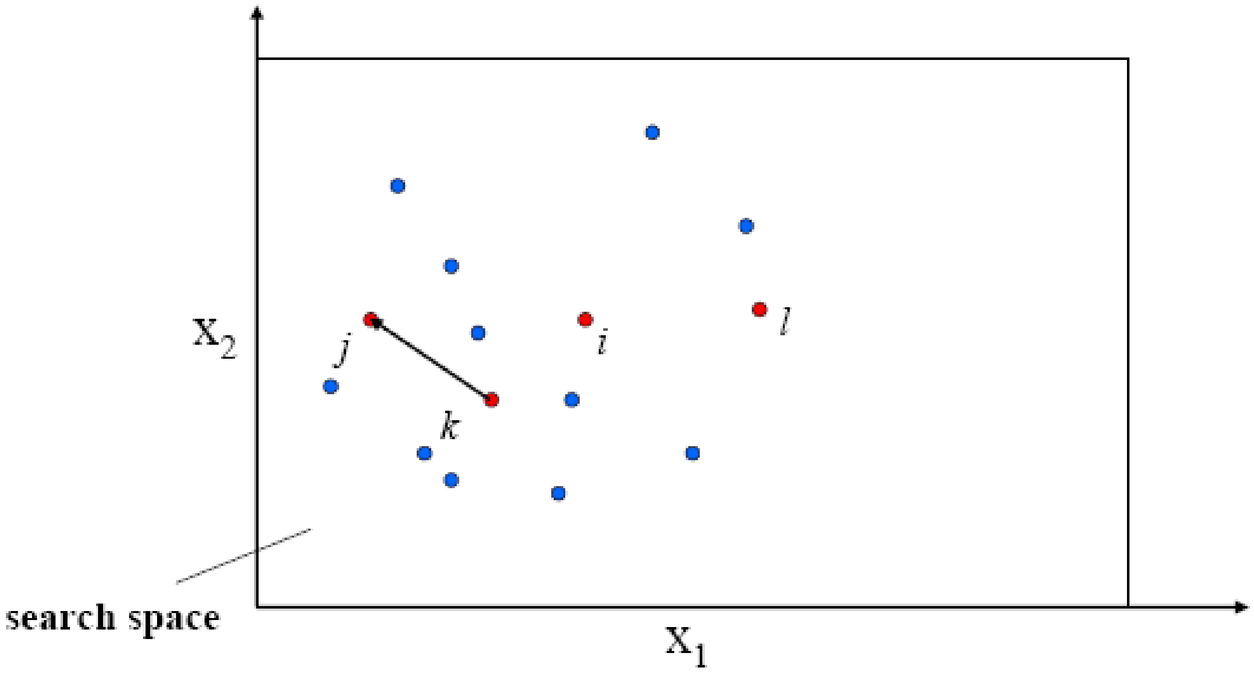
rand=U[0,1]

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

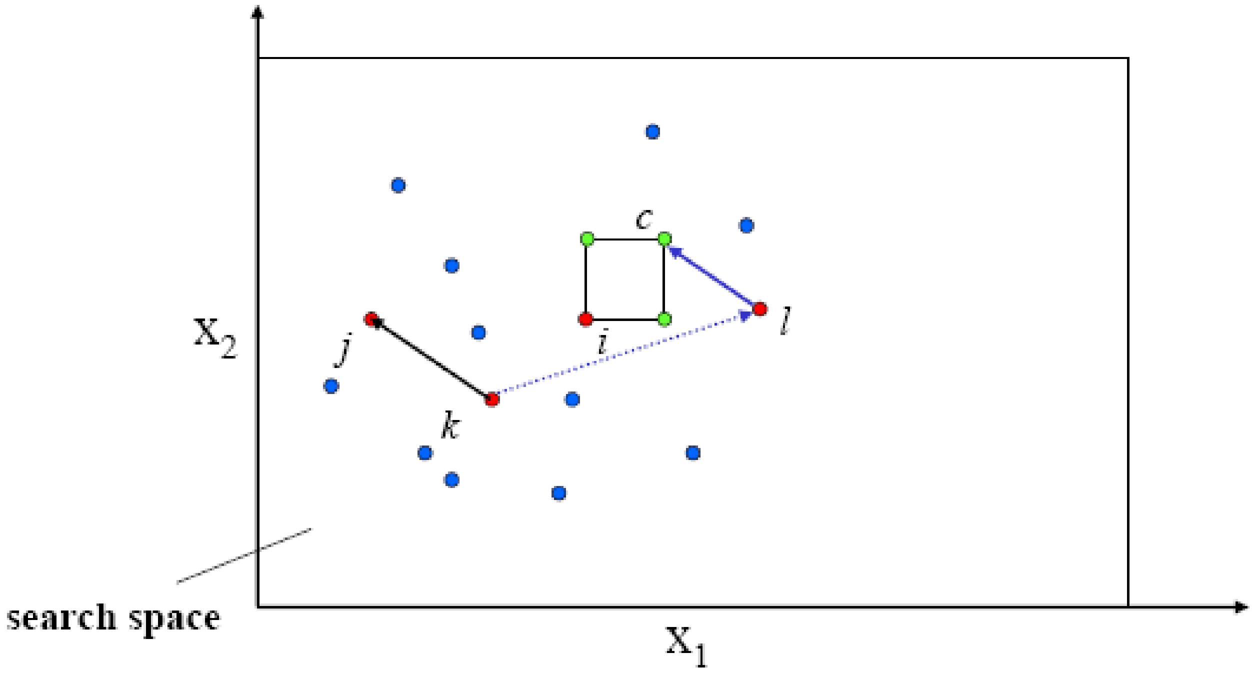
Mutation+Recombination (1)



Mutation+Recombination (2)



Mutation+Recombination (3)



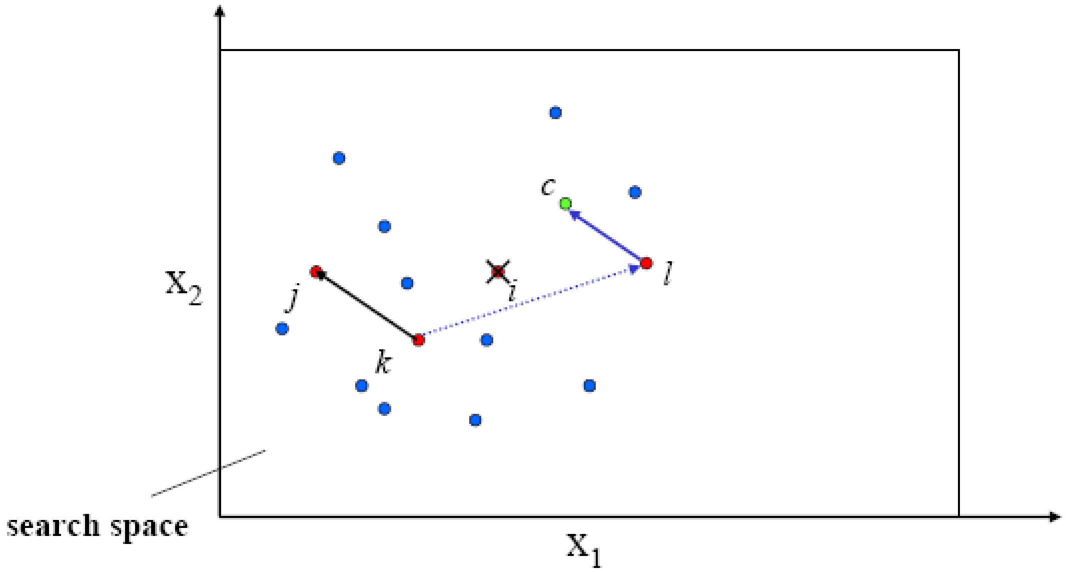
Selection

- 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

Selection



Loop

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

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

Initialization

- The population size *N* may be set to be 10 times the number of parameters
- Initialize the location of solutions
  – Uniform distribution
  – Gaussian distribution
  – Halton sequence

Halton Sequence

```
halton(i,j)
    prime[10]=[2 3 5 7 11 13 17 19 23 29]
    p1=prime[j]
    p2=p1
    sum=0
    do
    {       x = mod(i,p1)
            sum = sum + x / p2
            i = floor(i/p1)
            p2 = p2 * p1        } while i > 0

    return (sum)
```

Distribution from Halton Sequence

(i,j)	Sum	Normalized	Cumulative
(10,1)	0.3125	0.0653	0.0653
(10,2)	0.3704	0.0774	0.1427
(10,3)	0.08	0.0167	0.1595
(10,4)	0.449	0.0938	0.2533
(10,5)	0.9091	0.1900	0.4433
(10,6)	0.7692	0.1608	0.6041
(10,7)	0.5882	0.1229	0.7270
(10,8)	0.5263	0.1100	0.8371
(10,9)	0.4348	0.0909	0.9279
(10,10)	0.3448	0.0721	1.0000

Mutation Scale Factor

- Upper limit
- Lower limit

  Must be at least $F = [(1\text{-}CR/2) / N]^{0.5}$
- Usually 0.3~0.9
- May be a random variable
  - Choose *F* from the interval [0.5, 1.0] randomly for each generation or for each difference vector

Recombination

- Crossover mechanisms
  - One point
  - Multiple points
  - Uniform

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=0.2 helps optimizing separable functions since it fosters the search along the coordinate axes.
- For parameter dependence, the choice of CR=0.9 is more appropriate.

Conclusions for DE

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