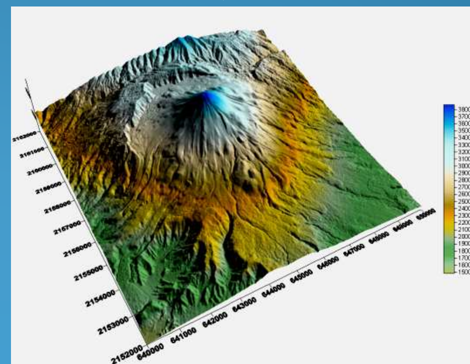
Differential Evolution (DE) algorithms for optimal solutions search: micro-macro scale applications in Earth Sciences

Lorenzo Borselli *

***National Research Council**
Research Institute for Geo-Hydrological Protection (CNR-IRPI)
Via Madonna del Piano 10,
50019, Sesto Fiorentino (Florence), ITALY
borselli@irpi.fi.cnr.it
http://www.irpi.fi.cnr.it/borselli.html



Consiglio Nazionale delle Ricerche

This seminary have as subject the optimization techniques based on a class of genetic algorithm named DIFFERENTIAL EVOLUTION and their application in the context of earth sciences research.

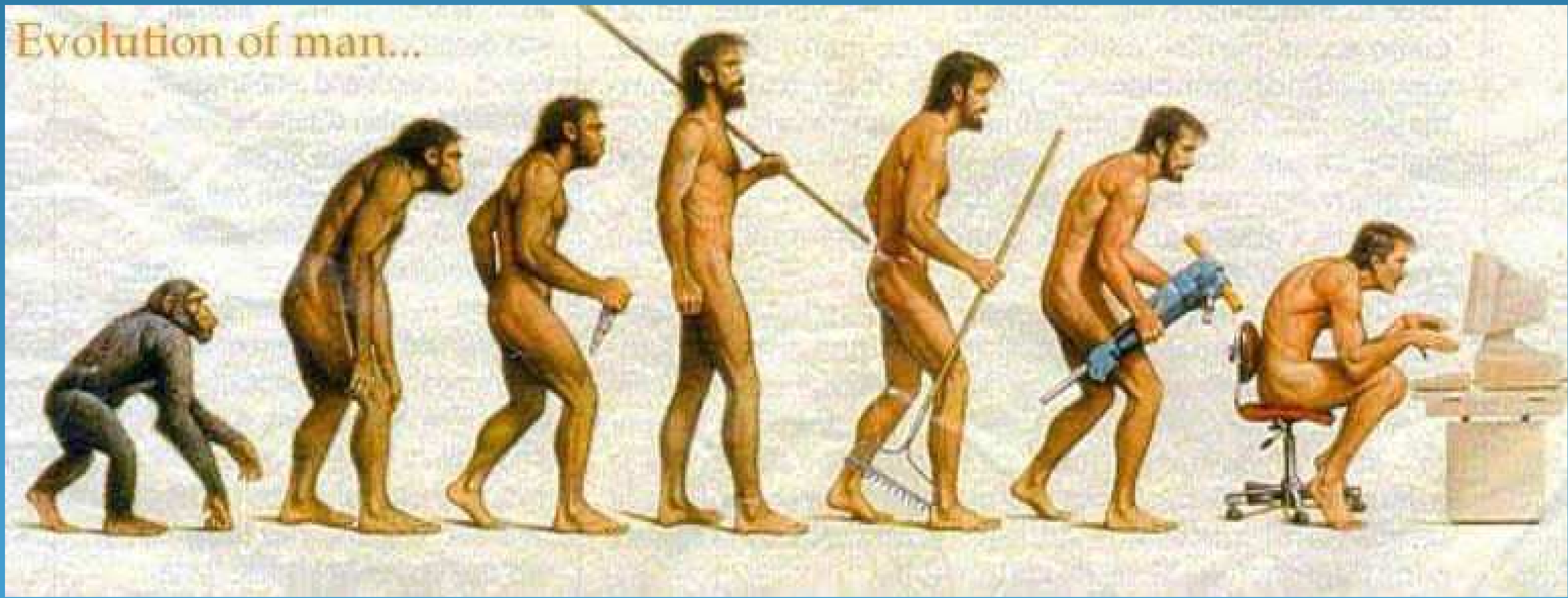
synopsis

1) Optimization and general concepts in real world and earth sciences.
2) Differential Evolution algorithms and for application in earth sciences research fields.
3) Presentation of examples of application in various fields (micro and macro scale):

- Particle shape analysis.
- Sedimentology,
- Hydrology,
- Stability of natural slopes,
- Volcano's DTM analysis

Optimization and evolution .....
...Or how to obtain the BEST RESULT
(e.g. maximum food production) with a MINIMUM
COST in terms of resources, time, energy....etc.

The human being aspires to the best possible performance. Both individuals and enterprises are looking for optimal - in other words, the best possible - solutions for situations or problems they face.



Optimization is fundamental in technological processes and in the progress of science which is the product of a long way of trials and errors in experiments , theories and models.

Typical examples of current optimization algorithms applications in technology include:

- **Protein structure prediction (minimize the energy/free energy function)**
- **Traveling salesman problem and optimal circuit design or network (minimize the path length)**
- **Chemical engineering (e.g., analyzing the Gibbs free energy)**
- **Safety verification, safety engineering (e.g., of mechanical structures, buildings)**
- **Model calibration (many engineering fields..)**
- **Many .. many others...**

Most of these problems can be expressed in mathematical terms, and so the methods of optimization undoubtedly render a significant aid in applications in science (earth science):

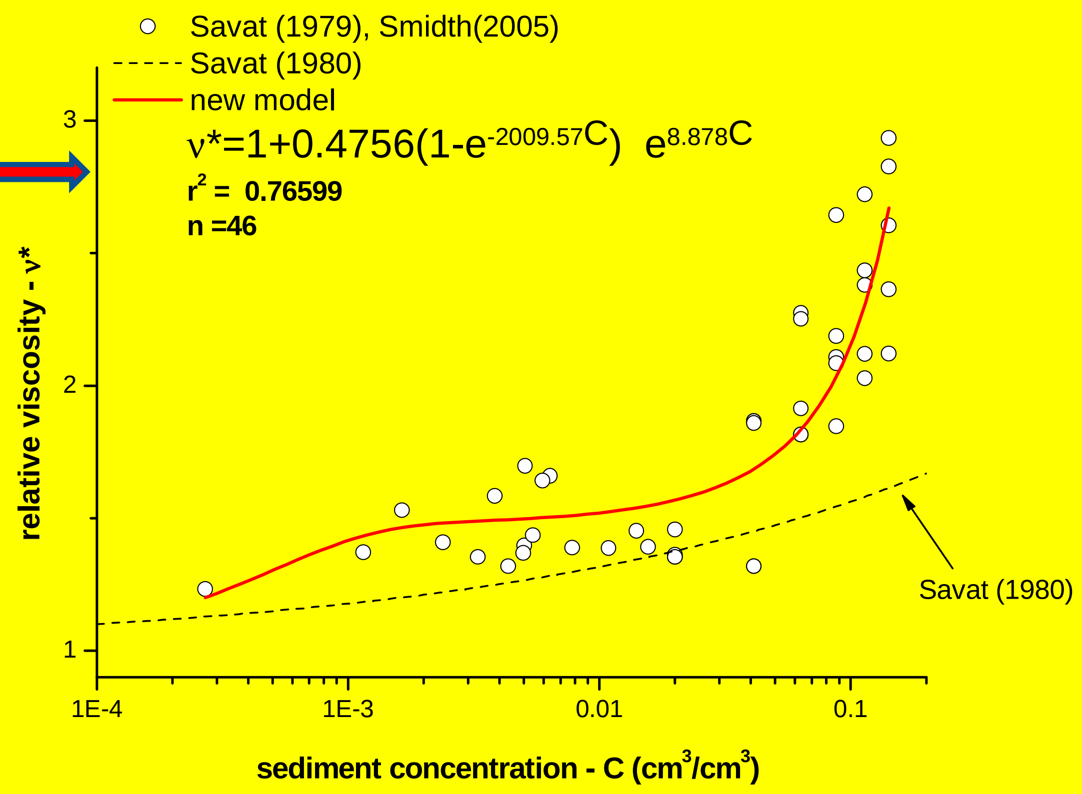
- **Fitting of nonlinear models to data**
- **Data mining**
- **Data clustering**
- **Inversion procedures**
- **Numerical Approximation of roots of functions**
- **New model paradigm validation and testing**
- **Image processing …**

Optimization in practice -1

Subject of optimization procedure:

Example non linear fitting

To fit nonlinear model to data
In the example we need to find
3 parameters



When the non linear function is
Simple the fitting can be done
With common statistical
Softwares: Statistica , origin..
Matlab , excel solver .....etc.

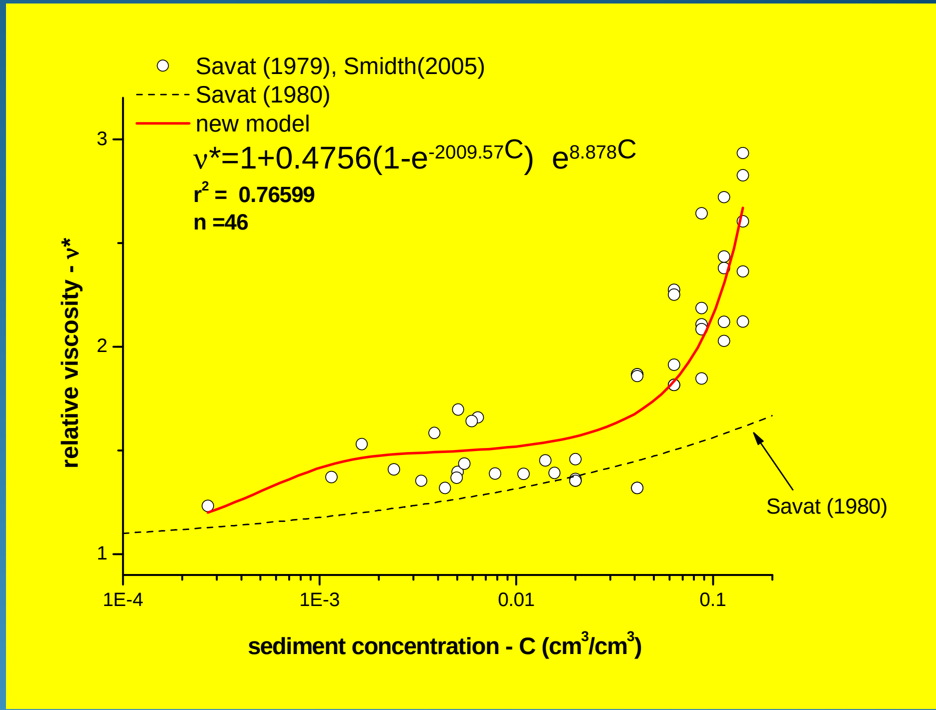
When More complex cases arise
we need of special coding ... and
Program

Optimization in practice -2

Goal

Minimize the sum of all the differences between the data and the assumed nonlinear model

The best performance coincide with the minimum possible of this sum of differences.



So our GOAL is to find the optimal 3 parameters in the nonlinear model that ensure the minimum possible difference with respect the data (residuals)

Optimization in practice -3

Cost or objective functions

$$obj = \sum_{i=1}^{n} \left(obs_i - pred_i \right)^2$$

Least squares

$$obj = \sum_{i=1}^{n} \left| obs_i - pred_i \right|$$

Sum of absolute deviate

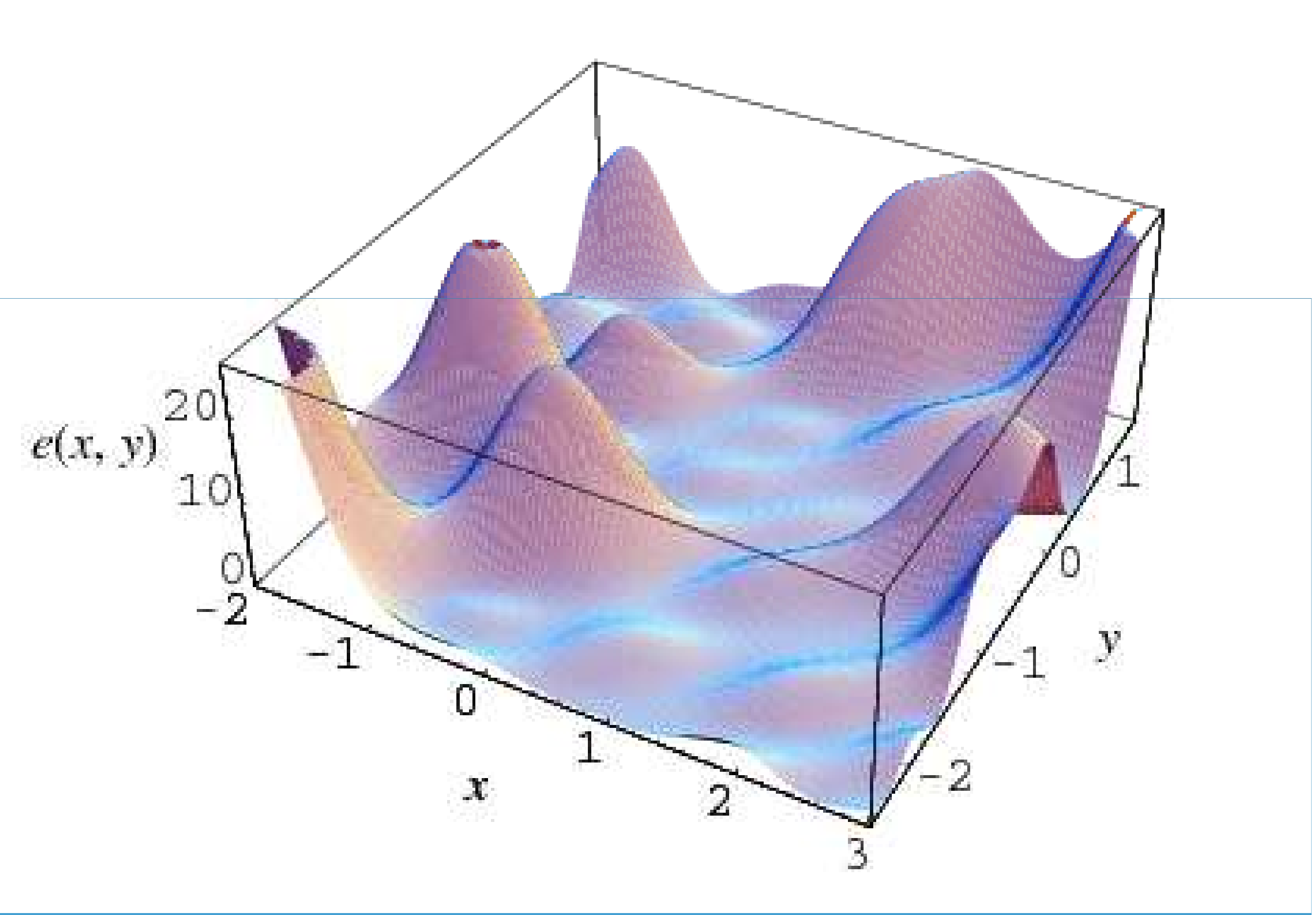
$$obj = \sum_{i=1}^{n} \frac{\left(obs_i - pred_i \right)^2}{\sigma^2_{obs}}$$

Reduced Chi square

Dimension of the problem

In this example the OBJECTIVE function is defined by TWO PARAMETERS

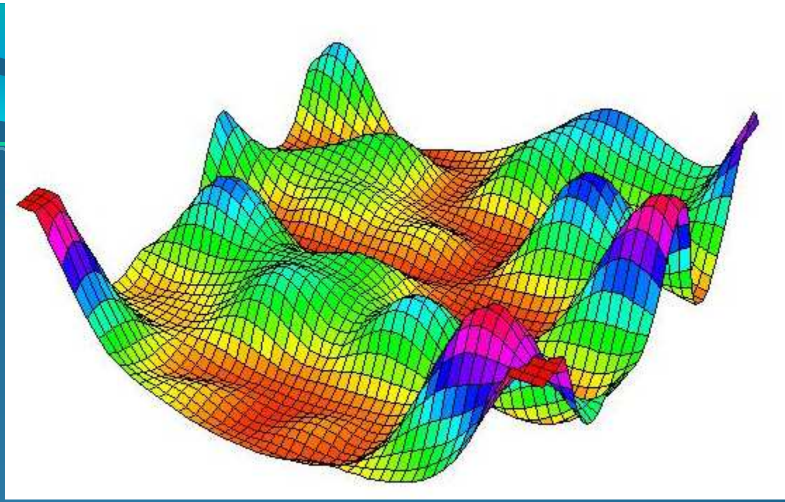
The dimension of the Problems is given by the number of parameters
That in n-dimensional Space define the Objective function



Optimization in practice -5

Searching techniques

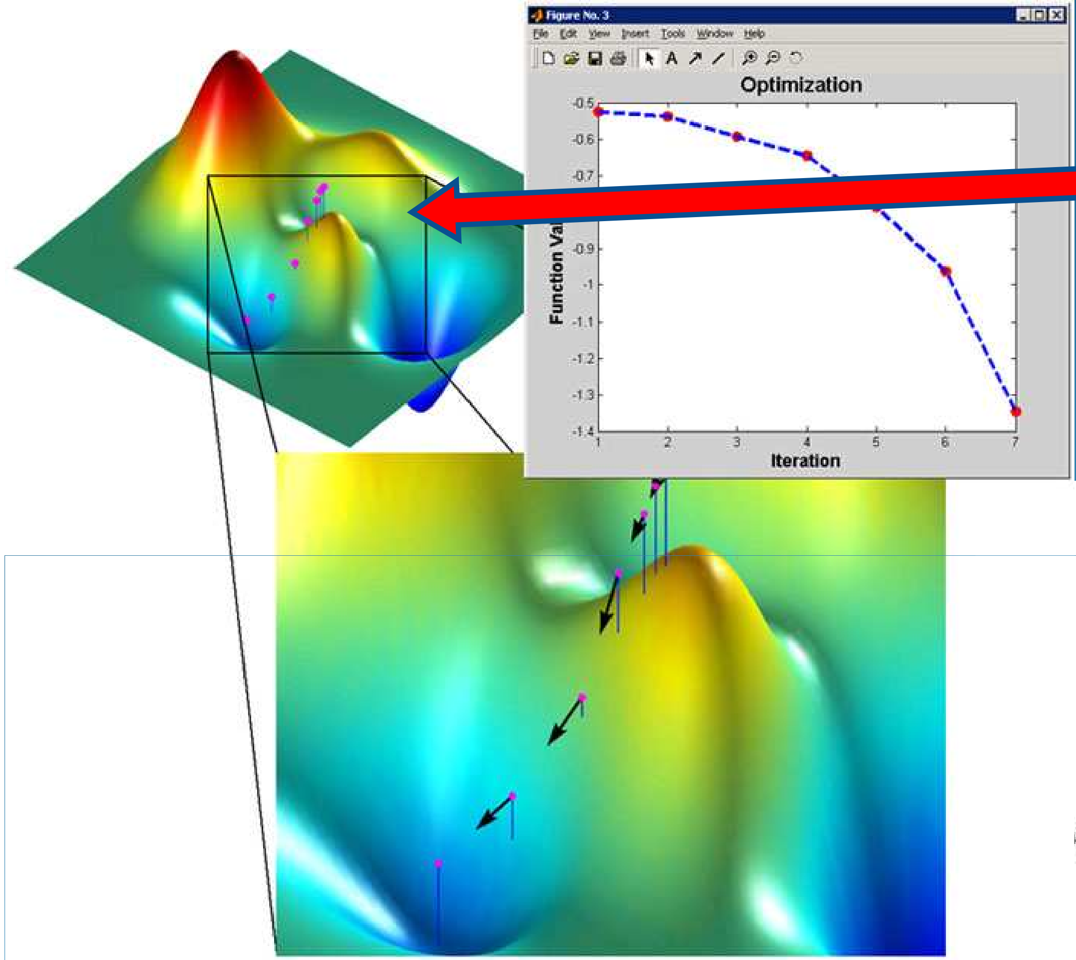
Algorithms designed to explore the n-dimensional surface of the OBJECTIVE function in order to find the place where the GOAL (minimum, maximum, or close to zero) is satisfied …



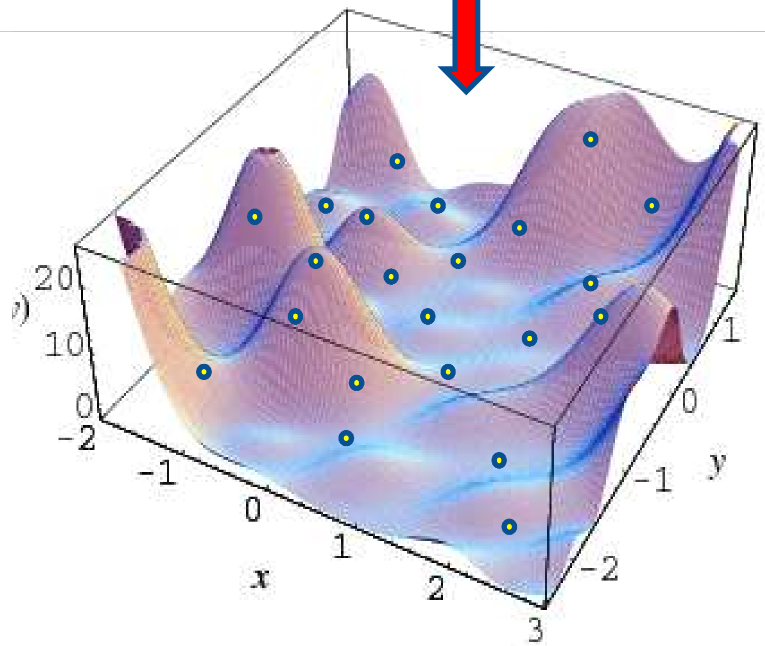
Actual methods (best known):

- Gradient descent aka steepest descent or steepest ascent
- Nelder-Mead method aka the Amoeba method
- Simplex method
- Quasi-Newton methods
- Interior point methods
- Conjugate gradient method

Local optimization versus global optimization



Single starting point
Vs
Multiple starting points
(+ interactions between points)



Optimization in practice -7

Others fundamental components in the optimization algorithms are:

Parameters space (search space for each parameter) (e.g. [-∞,+∞] ; [0, 100] …[-20.0, +5.0]).

Constraints (e.g. A≥0 ; B < 20; C < f(x,y..z)).

Penalty functions (to increase the OBJ function of a constant when a constraint is violated). Usually very high or very low constant; E.g. Penalty = 1E+10 ; ObJ=Penalty if an assigned constraint is violated

Termination criteria e.g. stop the search iteration when the difference in OBJ values between two successive iteration is Lower than a given tolerance value : e.g. $\varepsilon = 1E\text{-}12$

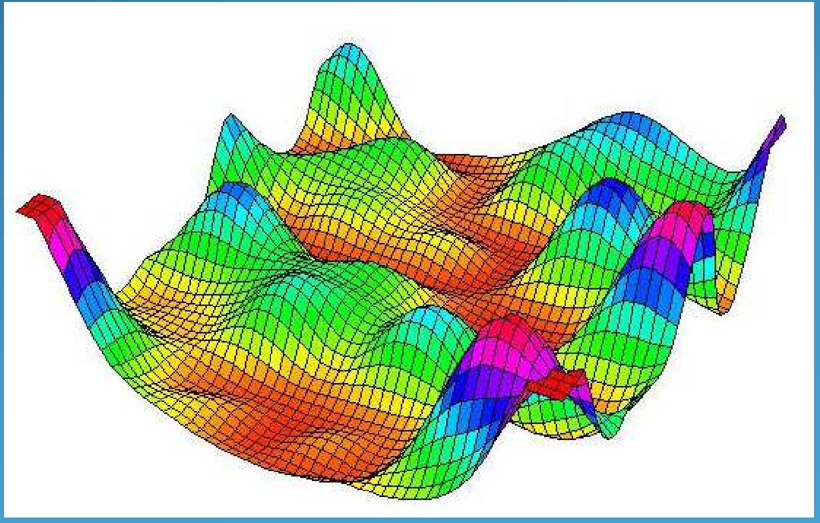
Stop if $|OBJ_{n-1} - OBJ_{n}| < \varepsilon$

Optimization in practice -8

there are a many other popular methods for mathematical optimization:

- Simulated annealing
- Tabu search
- Genetic algorithms
- Ant colony optimization
- Evolution strategy
- DIFFERENTIAL EVOLUTION
- Particle swarm optimization

……



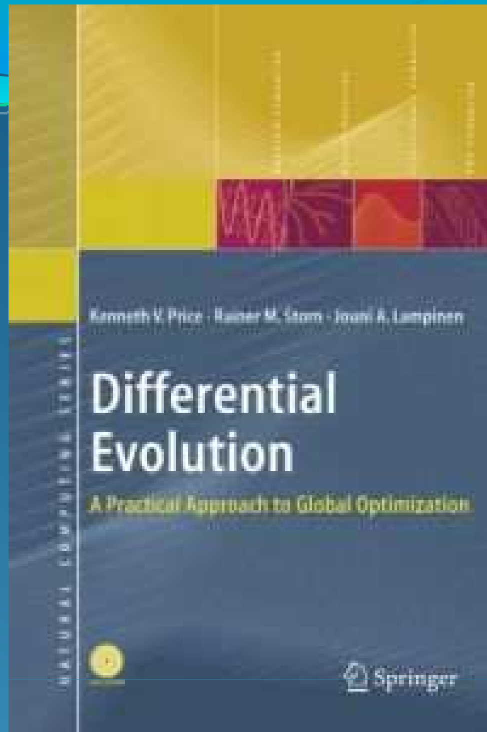
Valid criteria to choose one of them should be based on:

- **Available software (commercial , freeware, open source)**
- **Available code for programming in special Complex optimization problems**
- **Easy implementation of new problems**
- **Good performance (considering comparative benchmarks studies)**
- **Global optimization**
- **Speed of the computation**

Differential Evolution algorithms -1

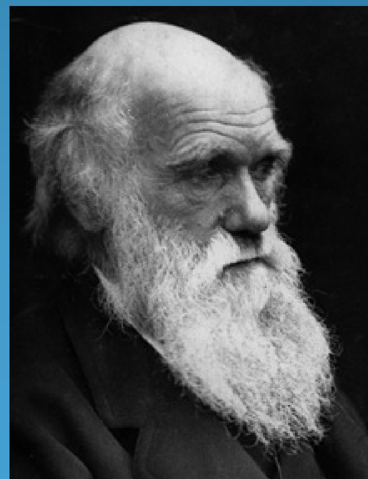
"Differential evolution (DE) is a stochastic parallel direct search evolution strategy optimization method that is fairly fast and reasonably robust. Differential evolution is capable of handling nondifferentiable, nonlinear and multimodal objective functions."

http://mathworld.wolfram.com/DifferentialEvolution.html



DE algorithm Born in the middle of 90's from some Ideas from K.Price and R. Storn. (Storn and Price 1997a,b)

Charles Darwin



This algorithm uses some paradigms from the theory of evolution but use also use many stochastic components to implement these concepts.

Since 1996-1997 appeared more than 3000 papers and technical Reports with studies and implementation of DE.

Many scientists consider the DE algorithms as one the best Optimization algorithm and useful in many applications

Differential Evolution algorithms -2

General field of application:

In all cases where there are many local optima; intricate constraints; mixed-type variables; or noisy, time-dependent or otherwise ill-defined functions, the usual methods don't give satisfactory results the DE may be a solution

DE implementations exist for well known software and programming languages

Mathematical –numerical analysis software as interpreted scripts or compiled toolboox

Mathematica (wolgram resarach)
Matlab (mathworks)
Scilab
R

And many compiled or specialized Software..

Programming languages high level (procedural or OOP)

C++
Fortran 90
Java
Pyton
Pascal
Object Pascal (by L.B)

See : http://www.icsi.berkeley.edu/~storn/code.html

Basic steps in DE algoritms

Preliminary steps

Initialization
Evaluation

iterative loop

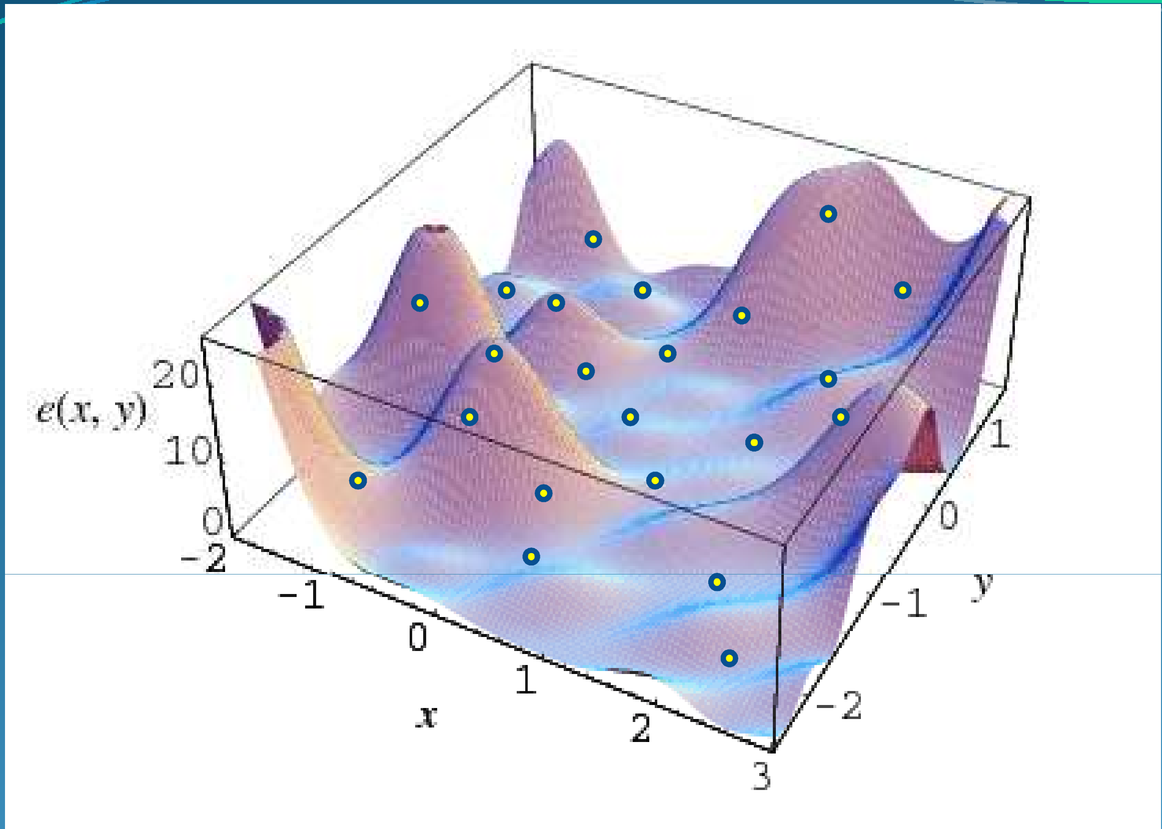
Repeat
    Mutation
    Recombination
    Evaluation
    Selection
Until *(termination criteria are met)*

Initialization – random filling the population of vectors that contains The parameters (a,b,c,d,e,f) to calculate the given OBJ function

obj = *f*(a,b,c,d,e,f)… (**6 parameters vectors - Population of *n* vectors**)

	V1	V2	V3	V4		Vn
a	200	153	99	34		170
b	20.2	32.4	9.5	4.3	……………	18.2
c	0.01	2.3	4.3	1.3		3.4
d	45.3	32.1	46.3	34.8		39.6
e	435	82	293	91	……………	103
f	349	294	673	830		327
	obj 1	obj 2	obj 3	obj 4		obj *n*

Fittness Calculation for each individuals (EVALUATION)



Vector initialization

In a population of potential solutions within an n-dimensional search space, a fixed number of vectors are randomly initialized, then evolved over time to explore the search space and to locate the minima of the objective function.

The Kernel of DE algorithm is a sequence of operations (or use of some operators) at each iteration:

- Mutation (casual mutation of genes)
- Recombination (or Crossover – exchange genes)
- Selection (the best survive – the worst die)
- Age evaluation (and individual can't survive more than a given number of generations (age) – this is recent addition to solve stagnation problems)

Differential Evolution algorithms -7

mutation –At each iteration, called a generation, new vectors are generated by the combination of vectors randomly chosen from the current population

	Vr1	Vr2	Vr3	Vi *mutant*
a	200	153	99	254
b	20.2	32.4	9.5	43.1
c	2.6	2.3	4.3	0.6
d	45.3	32.1	46.3	31.1
e	435	82	293	224
f	399	294	673	20

mutation

With Assumed F=1.0

For each *i in (1, …n) population of vectors , form a 'mutant vector' using simple*

Vi (mutant) = Vr1+F(Vr2-Vr3) or Vi (mutant) = Vr1+rand F(Vr2-Vr3)

Where *r1, r2, and r3 are three mutually distinct randomly drawn indices from (1, …n), and also distinct from i, and 0.0<F<=2.*

Recombination (or Crossover) –The new generated vectors are then mixed with the parent vector (Vparent) of the old population. This operation is called **recombination** and produces the final **trial vector**

Crossover

With Assumed CR=0.5

	Vparent	CR	Vi *mutant*	Crossover	Vtrial
a	200	0.34	254	→ OK	254
b	20.2	0.89	43.1		20.2
c	2.6	0.72	0.6		2.6
d	45.3	0.14	31.1	→ OK	31.1
e	435	0.53	224		435
f	399	0.23	20	→ OK	20

For each component of vector, draw a random number in U[0,1]. Call this **$rand_j$**.
Let 0<=CR<1 be a cutoff.
If **$rand_j$<=CR**, *then* ***$Vtrial_i = Vmutant_i$***, *else* ***$Vtrial_i = Vparent_i$***..

SELECTION – Finally, **the trial vector is accepted for the next generation** if and only if it yields a reduction in the value of the objective function. This last operator is referred to as a **selection**

Vtrial Accepted only if
OBJ(Vtrial) < OBJ(V4)
Otherwise the Vparent Is retained and it will survive with the next generation

	V1	V2	V3	V4	…	Vn	Vtrial
a	200	153	99	34		170	254
b	20.2	32.4	9.5	4.3	……………	18.2	20.2
c	0.01	2.3	4.3	1.3		3.4	2.6
d	45.3	32.1	46.3	34.8		39.6	31.1
e	435	82	293	91	……………	103	435
f	349	294	673	830		327	20
	obj 1	obj 2	obj 3	obj 4		obj *n*	obj *TRIAL*

Ageing – vectors that stay unchanged for a long time after several generations will have a too OLD age (e.g use a Cutoff age AG=30). In this case the vector Is substituted with a new one with a random in initialization

	V1	V2	V3	V4		Vn
a	200	153	99	34		170
b	20.2	32.4	9.5	4.3	....	18.2
c	0.01	2.3	4.3	1.3		3.4
d	45.3	32.1	46.3	34.8		39.6
e	435	82	293	91	....	103
f	349	294	673	830		327
AGES →	1	14	2	30		4

Age counter for each vector increase 1 at each new generation…

Differential Evolution algorithms -11

Termination criteria – the iterations **Terminate when , for example , the variance of all the OBJ function values is lower than a predefined tolerance (e.g 1E-10). The vector with BEST performance contains the OPTIMUM set of the 6 parameters .**

	V1	V2	V3	V4	…	Vn
a	1.12	1.1	1.1	1.1		1.1
b	20.2	20.2	20.1	20.2	……………	20.2
c	4.3	4.3	4.3	4.3		4.3
d	45.3	45.3	45.3	45.3		45.3
e	82	82	82	82	……………	82
f	349	394	394	394		394
OBJ Function value	1.12+3	1.12+3	1.12+3	1.12+3 BEST!!		1.12+3

•Exist several variants for DE algorithms , mainly in the MUTATION operator.

•The F parameter (weight for mutation) usually is assumed in the range F=[0.5, 1.5].

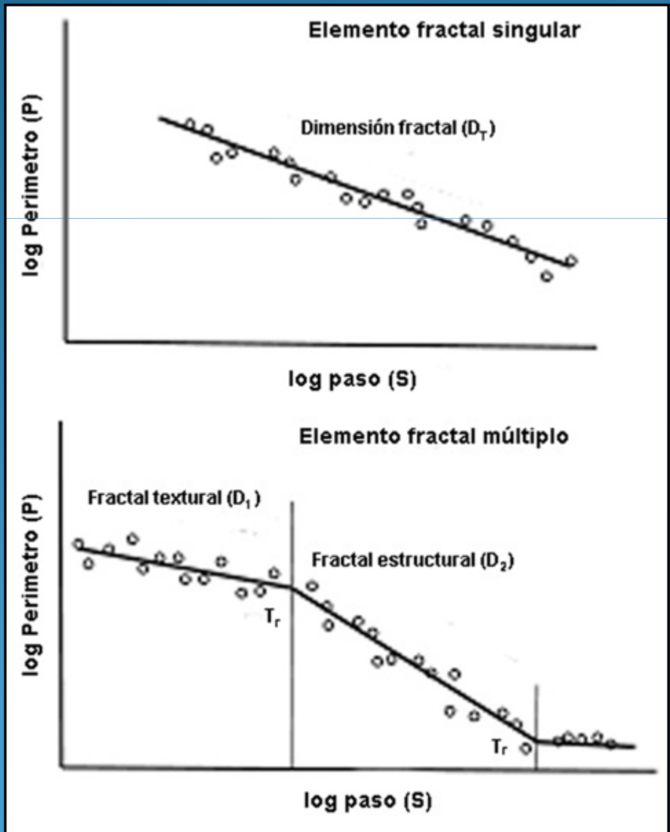
•The CR cutoff value for CROSSOVER is assumed in the range [0.5,1.0] (usually 0.8-0.9)

•The number of vectors in a population usually should be at least 10 times the number of parameters to optimize (e.g. a 6-parameters problem should have a population of 60 vectors).. But in some case these value may be higher (200-300) to guarantee GLOBAL OPTIMIZATION.

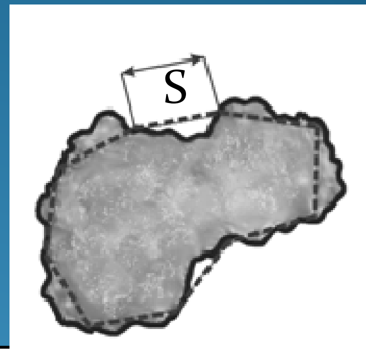
•Ageing cutoff value is usually low (AG =20-30) but this should be evaluated for each specific problem

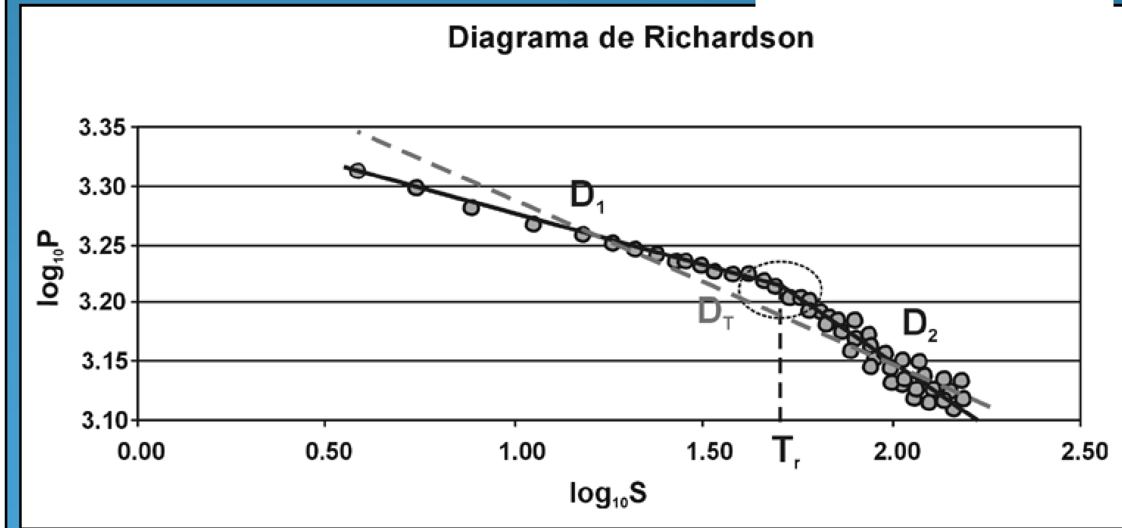
DE example of application -1 - **PARTFRACT software (Borselli & Sarocchi 2005,2006)**

PARTFRACT calculate, after a DE Global optimization of with 5 parameters, The fractal dimension textural (D1) , structural (D2) of particle's perimeter. The perimeter value are obtained with several approximantion using segments of different length S. So that the perimeter is approximated with a irregular poligon . Reducing the length S of the segment we have a longer perimeter P.

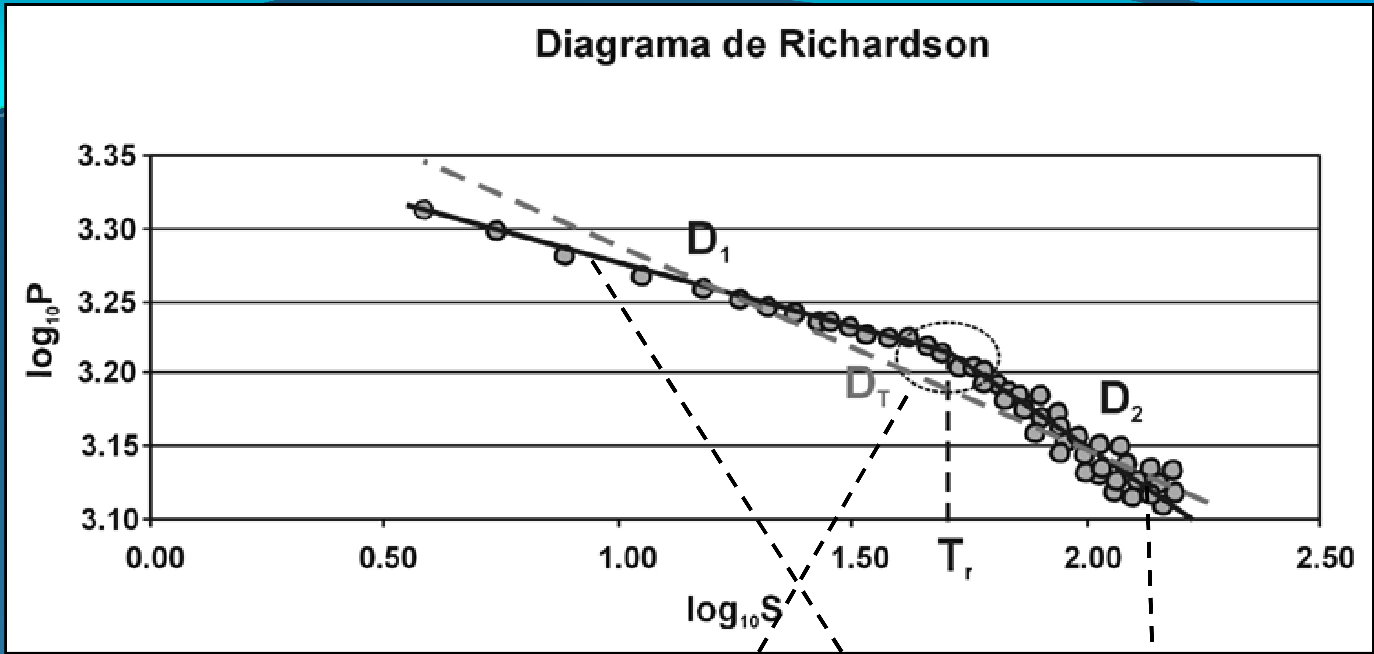


(from Sarocchi ,2006)





DE example of application – 1 - **PARTFRACT software (2)**



(from Sarocchi ,2006)

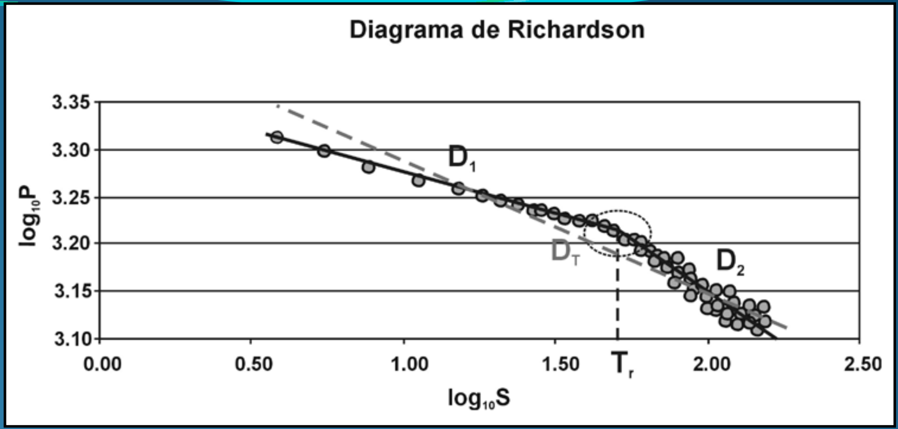
%T %textural component
%S %structural component

Output table after the Processing of a set of particles

n.	Op	Fd	D	size	Xint	Yint	S	D_1	D_2	m1	m2	q1	q2	%T	%S	dm1	dm2
1	893	238.452	1.061	50	0.8034	2.8283	6.3585	1.0349	1.0635	-0.0349	-0.0635	2.8563	2.8793	6	94	0.0058	0.0062
2	912	219.64	1.0576	50	0.7677	2.8319	5.8569	1.0401	1.0589	-0.0401	-0.0589	2.8627	2.8771	6	94	0.0043	0.0052
3	865	248.51	1.0529	50	1.2193	2.8048	16.5673	1.0251	1.0713	-0.0251	-0.0713	2.8354	2.8917	18	82	0.0029	0.0099
4	793	212.194	1.0351	50	0.9288	2.7881	8.4877	1.0199	1.0377	-0.0199	-0.0377	2.8066	2.8231	10	90	0.0039	0.006
5	878	242.904	1.067	50	1.3233	2.8074	21.0517	1.0302	1.1064	-0.0302	-0.1064	2.8474	2.9481	24	76	0.0026	0.0131
6	839	238.687	1.0535	50	1.0468	2.8048	11.1387	1.024	1.0649	-0.024	-0.0649	2.83	2.8727	12	88	0.0029	0.0074
7	866	239.094	1.0609	50	1.1056	2.8045	12.7517	1.0342	1.0724	-0.0342	-0.0724	2.8422	2.8845	14	86	0.0027	0.008

DE example of application – 1 - **PARTFRACT software (3)**

DE optimization parameters
D=5 (number of parameters)
N=150 (No. Of vectors in the population)
F=0.9 (weight factor in the mutation)
CR=0.9 (crossover probability)
AG=50 (age cutoff)
Tolerance 0.001



Constraints :

- The slopes of the textural and structural components must be included in the range -1E-19 and -1.0
- The Xm, Ym must be in the range of the observed data

OBJECTIVE FUNCTION

$$obj = \sum_{i=1}^{tr-1} (obs_i - pred_i)^2_{text} + \sum_{i=tr}^{n} (obs_i - pred_i)^2_{strut}$$

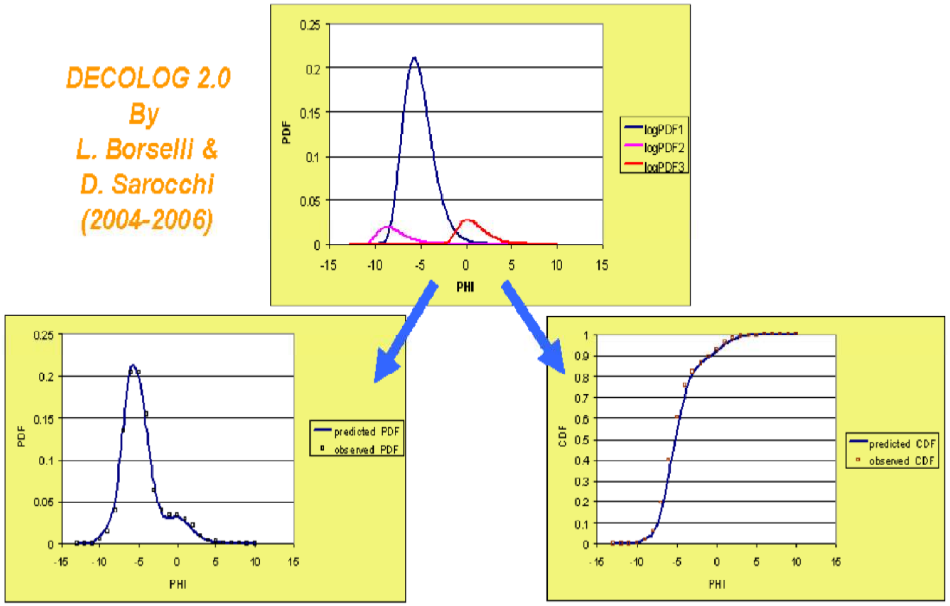
WWW.DECOLOG.ORG

Aim of DECOLOG software is develop a solution to decode the information present in the natural mixture of particles/sediments using, as paradigm, the 3-paramters log-normal distribution and particularly a defined mixture of these distributions.

DE algorithm is the Kernel of the Optimization procedure.

The optimization process in this case is MULTI-OBJECTIVE because the optimization proceeds in parallel with the best fitting of PDF(mixture) and CDF(mixture) to Experimental data….

DECONVOLUTION OF MIXTURES OF LOGNORMAL COMPONENTS FROM PARTICLE SIZE DISTRIBUTIONS

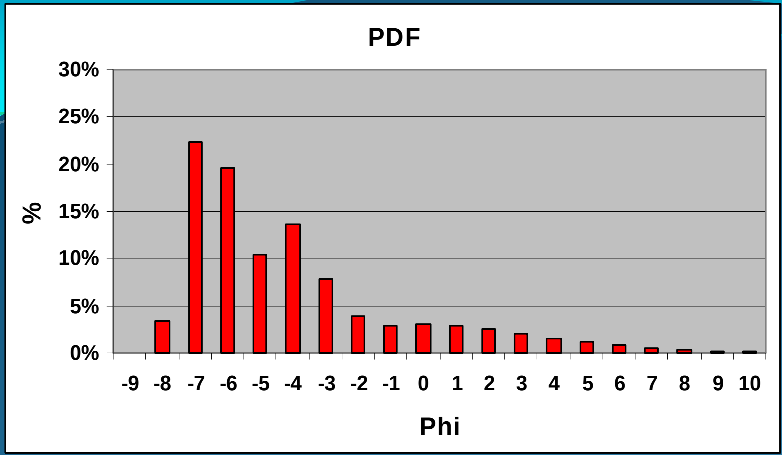


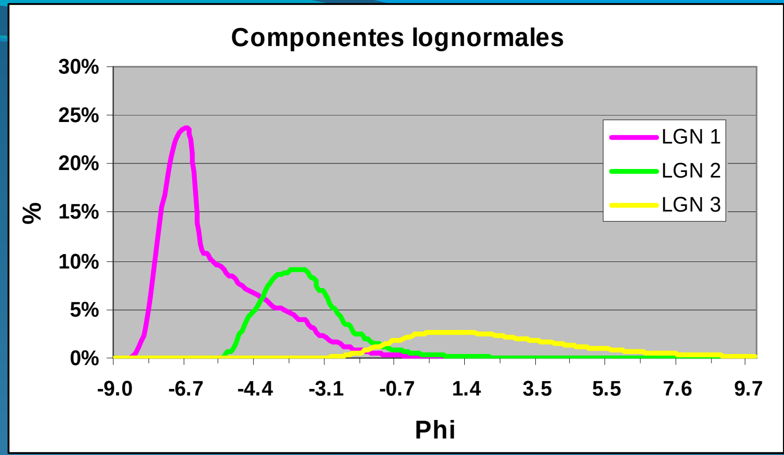
DECOLOG (rel 2.0 - 2004-2006) :

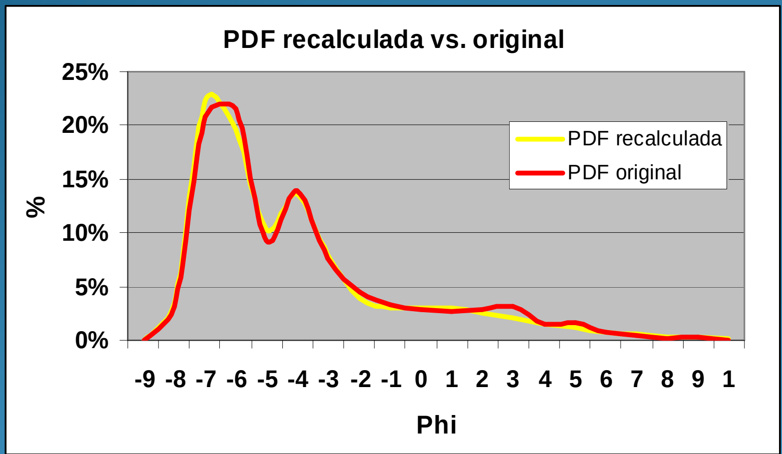
LORENZO BORSELLI
Research Institute for Hydrogeological Protection (CNR-IRPI)
Piazzale delle Cascine 15, 50144 Florence (ITALY), borselli@irpi.cnr.it

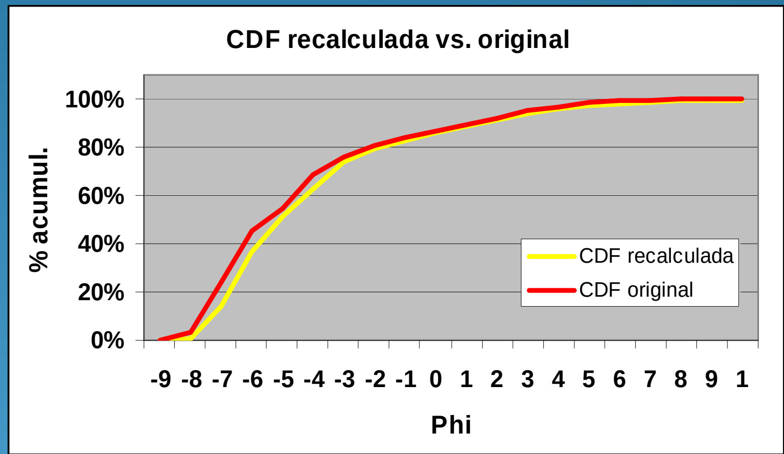
DAMIANO SAROCCHI
Instituto de Geofisica - UNAM (Mexico), d.sarocchi@mclink.it

DE example of application – 2 - **DECOLOG software (2)**









Natural MIXTURE of distributions

$$f(x)_{mix} = w_1 f_1(x) + w_2 f_2(x) + ... + w_n f_n(x)$$

PDF

$$F(x)_{mix} = w_1 F_1(x) + w_2 F_2(x) + ... + w_n F_n(x)$$

CDF

$$\sum_{i=1}^{n} w_i = 1$$

Weights of the mixture (fraction of each component)

DE example of application – 2 - DECOLOG software (3)

Global fitting statistics for CDF --------------------

Model efficiency coefficient EF :	0.9968002
Coefficient of Determination R^2 :	0.9987997
Kolmogorov-Smirnoff difference Ks :	0.0684153

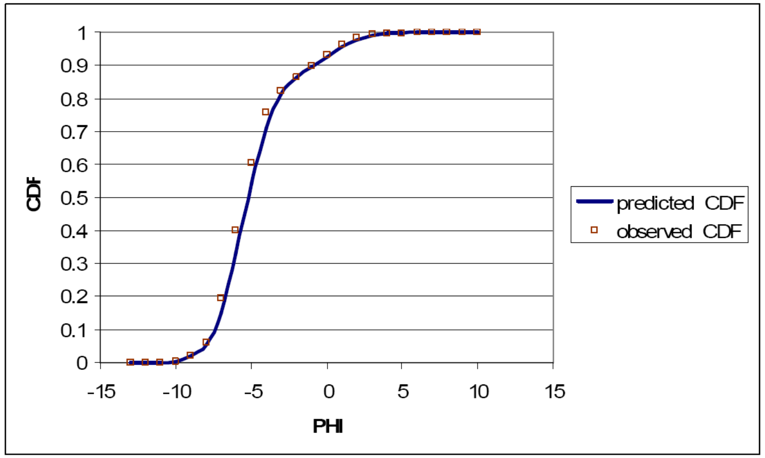


Fig 3.: global fitting performance on the observed CDF

Global fitting statistics for PDF -------------------

Model efficiency coefficient EF :	0.9895836
Coefficient of Determination R^2 :	0.9947991

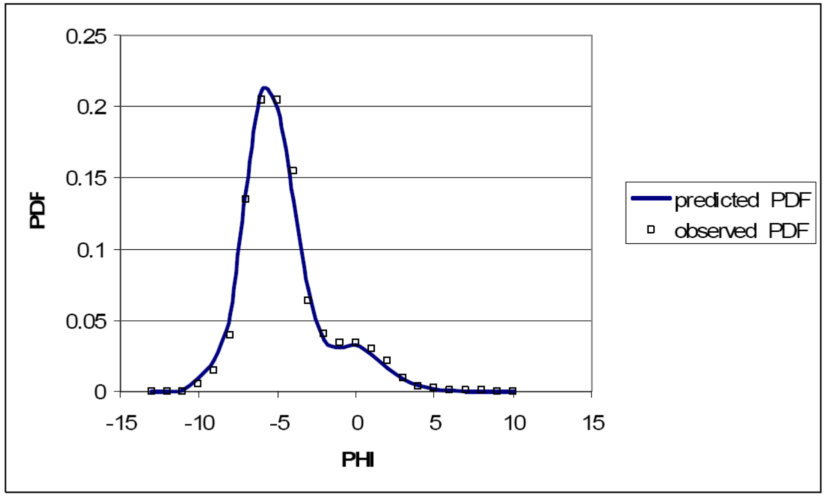


Fig 4.: global fitting performance on the observed PDF

------------------------- OPTIMUM FITTING PARAMETERS ---------

Shift1 (lambda):	**-12.8913**
Shift2 (lambda):	**-12.3366**
Shift3 (lambda):	**-4.293**
Scale1 (alpha):	**2.0256**
Scale2 (alpha):	**1.4338**
Scale3 (alpha):	**1.5945**
Shape1 (beta):	**0.2099**
Shape2 (beta):	**0.3722**
Shape3 (beta):	**0.3144**
Fraction 1 w1 :	**0.8243**
Fraction 2 w2 :	**0.0727**
Fraction 3 w3 :	**0.103**

Total minimised Objective function value: 0.0145516

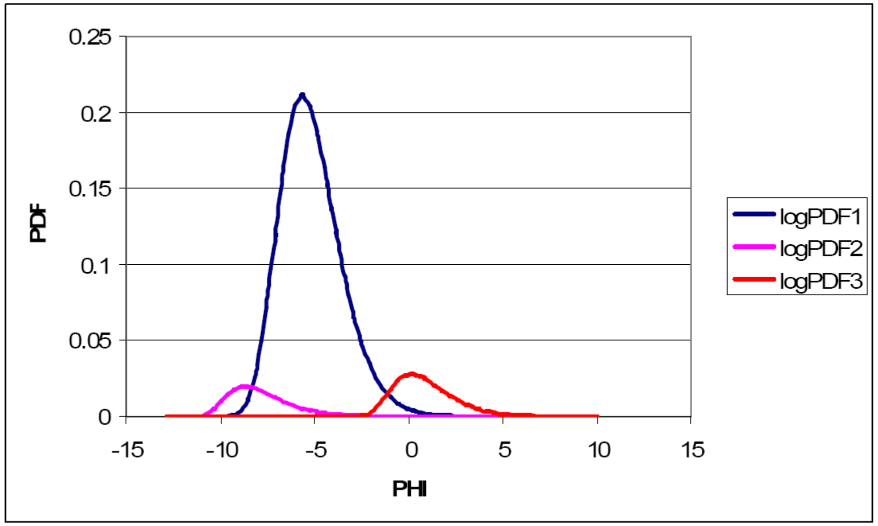


Fig.1: 3 components of decoded distribution using DECOLOG.

DE example of application – 2 - **DECOLOG software (4)**

Decoding with non linear multiobjective global optimization

A non linear multiobjective global optimization procedure has been developed to complete in efficient and robust way the decoding process. The optimization process allow to obtain the parameters $\alpha_i, \beta_i, \lambda_i, w_i$ for each distribution.

We established a concurrent fitting of the observed PDF and CDF by way of a multiobjective optimization minimizing at the same time the errors in the PDF and CDF. Because the tho objective are concurrent each optimum may be partially in conflict with the other establishing dominance.
So to obtain a result we transform the multiobective process for a computation purpose in a single objective optimization (Andersson, 2000).

multiOBJECTIVE FUNCTION :

$$obj = W_{cdf} K + W_{pdf} E_{ff}$$

$$where:$$

$$W_{pdf} = 1 - W_{cdf}$$

$$W_{cdf} = f(E_{ff})$$

Where K is the Kolmogorov-Smirnov maxmum difference between observed and computed CDF; E_{ff} is the model efficiency parameter developed by Nash and Sutcliffe (1970) that has a well recognized performances for non linear fitting, In this case we apple it to PDF part

DE example of application – 2 - **DECOLOG software (5)**

DE optimization parameters

D=11 (number of parameters for 3 component mixture)

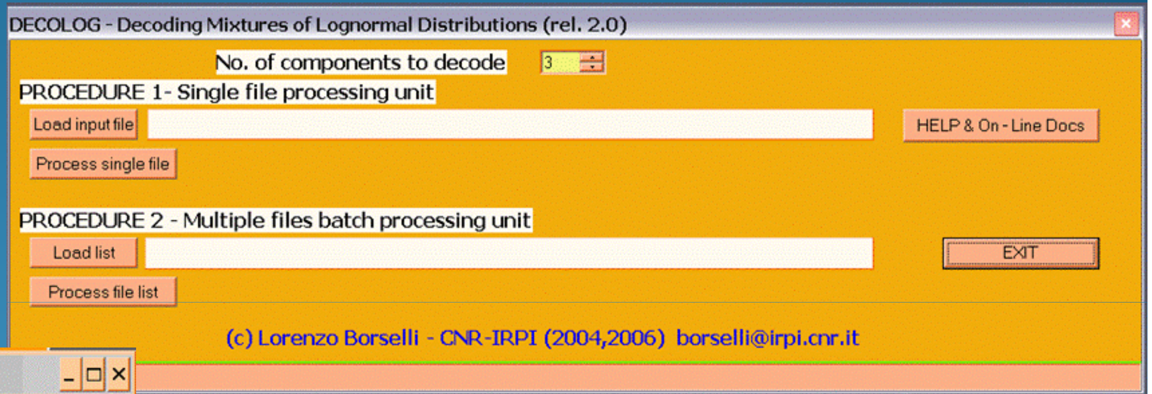
N=330 (No. Of vectors in the population)

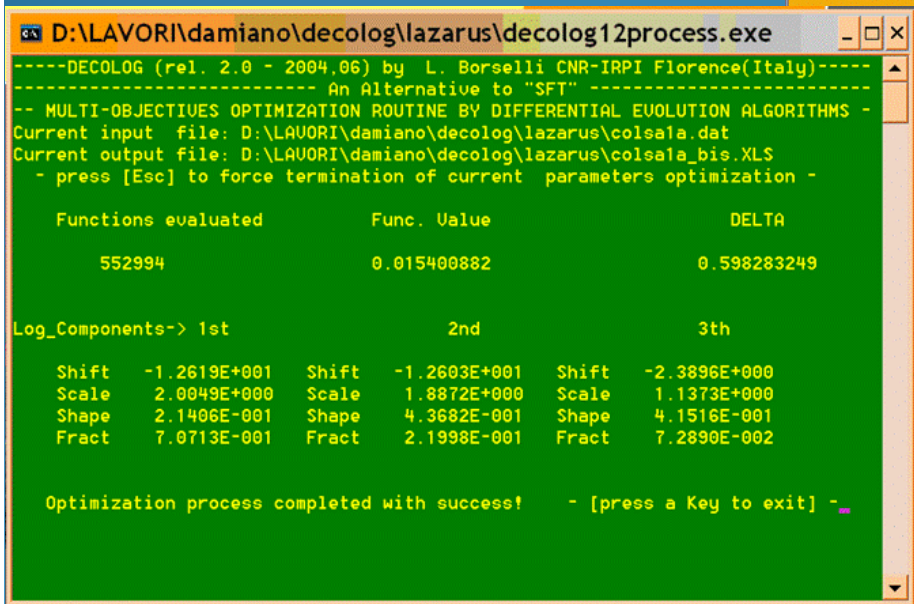
F=0.9 (weight factor in the mutation)

CR=0.9 (crossover probability)

AG=30 (age cutoff)

Tolerance 0.0001





The decolog software is a freeware tool

It can be downloaded at

www.decolog.org

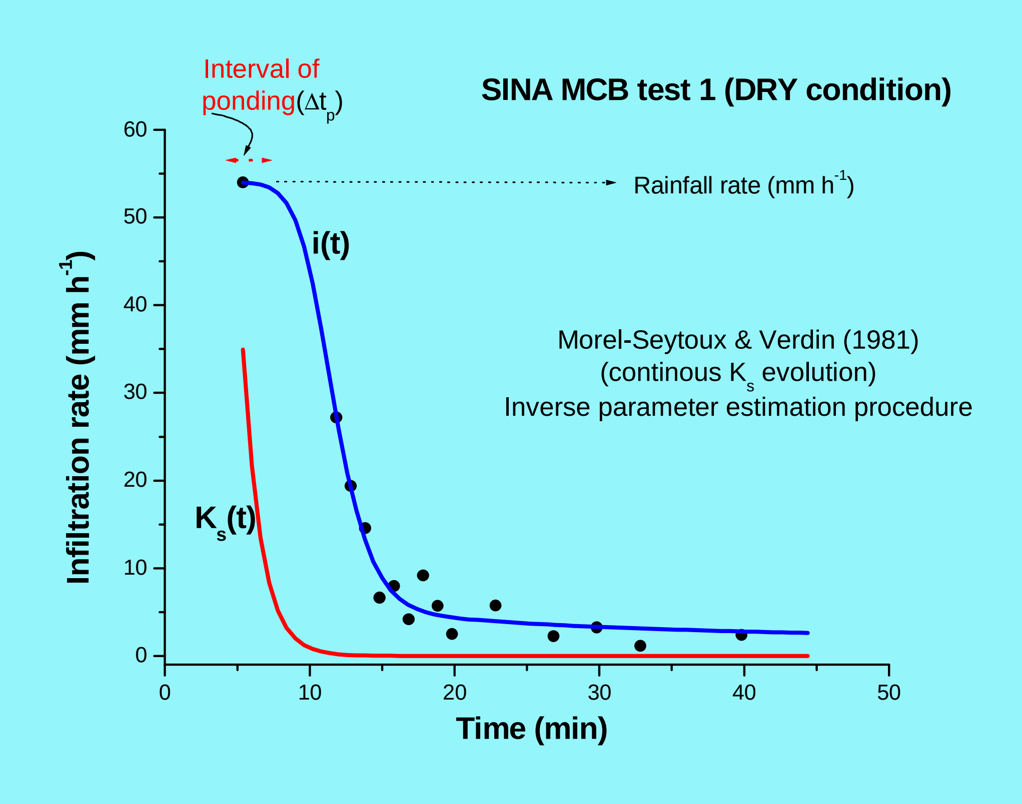
The next update at 3.0 will be in 2009

DE example of application – 3 - **INFIT software (Borselli 1998,2002)**

INFIT 1.2

Software for inversion procedures and determination of soil hydraulic parameters in condition of strong dynamic of the surfeces as for high rainfall , runoff intensity, severe soil erosion
soil surface degradation due to reduction of porosity, sealing and a consequent reduction of saturated conductivity.

Parameters to estimate G, Ks_0, Ks_f, a



Dynamic infiltration model (non linear):
Morel-Seytoux & Verdin (1981)
Borselli (1998):

$$i(t) = f(r, t_p, K_s, G, \Delta\theta)$$

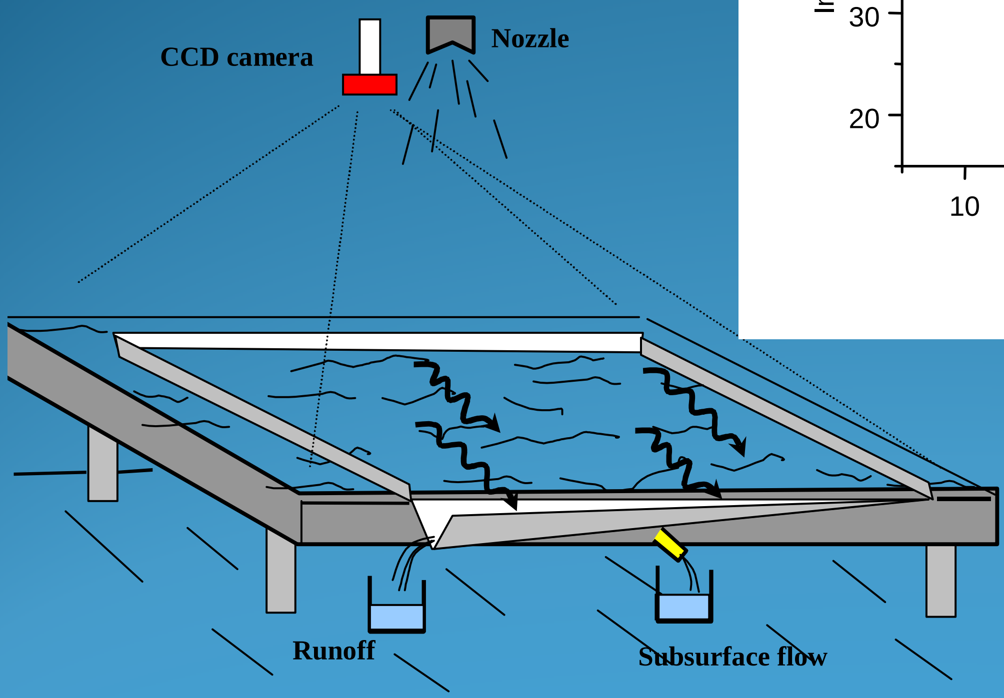
If t>tp

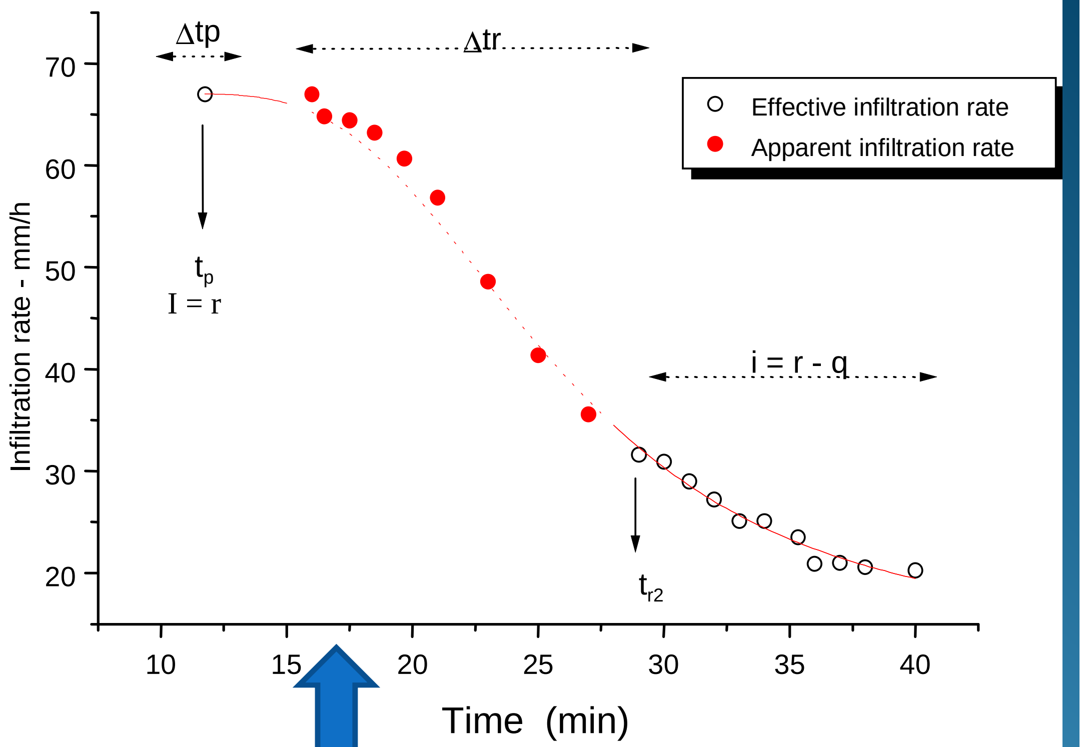
$$K_s(t - t_p) = (K_{s0} - K_{sf})e^{-a(t-t_p)} + K_{sf}$$

DE example of application – 3 - INFIT software (2)

Rainfall simulator/infiltrometer

- full cone nozzles [int. 25 to 120 mm/h]
- height of fall 4.5 m.
- plot with buffer areas,
- runoff & subsurface flow collection area 05X0.8 m
- CCD camera
- laser microprofilometer



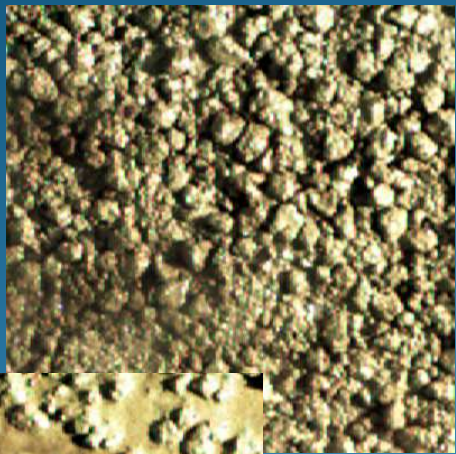


INFILTRATION RATE is computed by :

Infiltration rate= Rain. Rate - Runoff rate

DE example of application – 3 - **INFIT software (3)**

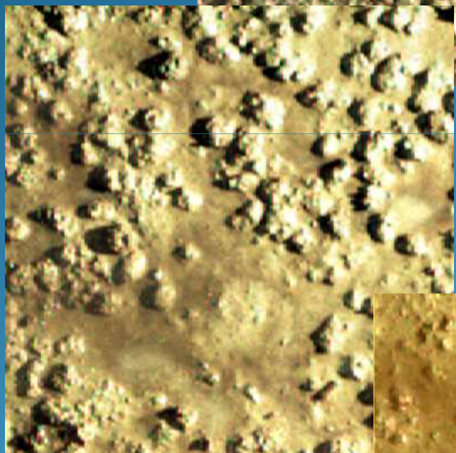
Initial surface

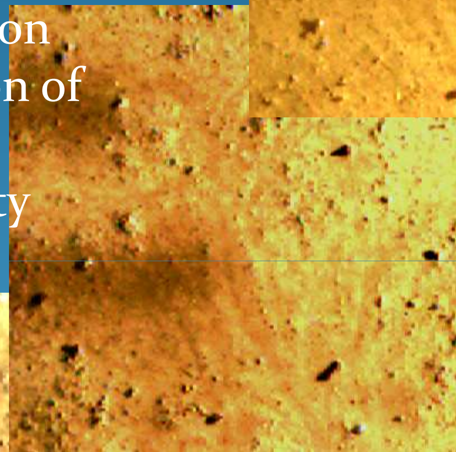


Dynamic of soil surface subject to intense rainfall and runoff and erosion

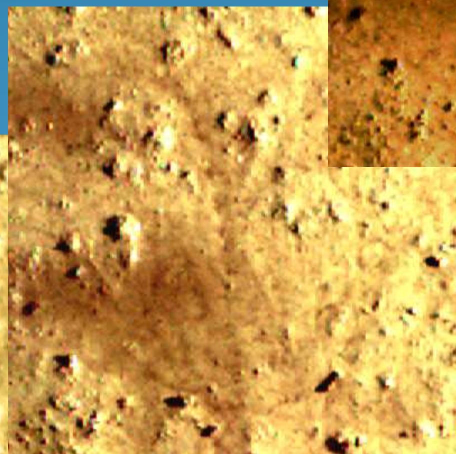
Decay or roughness
Erosion /re-deposition
sealing and reduction of water conductivity
reduction of porosity

1°





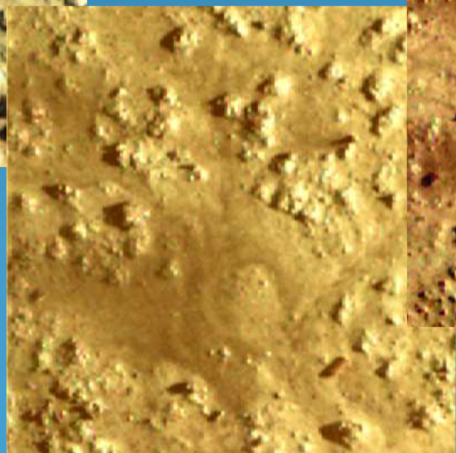
4°



3°

Soil (sina mcb)

2°



DE example of application – 3 - **INFIT software (4)**

DE optimization parameters

D=4 (number of parameters)

N=80 (No. Of vectors in the population)

F=0.9 (weight factor in the mutation)

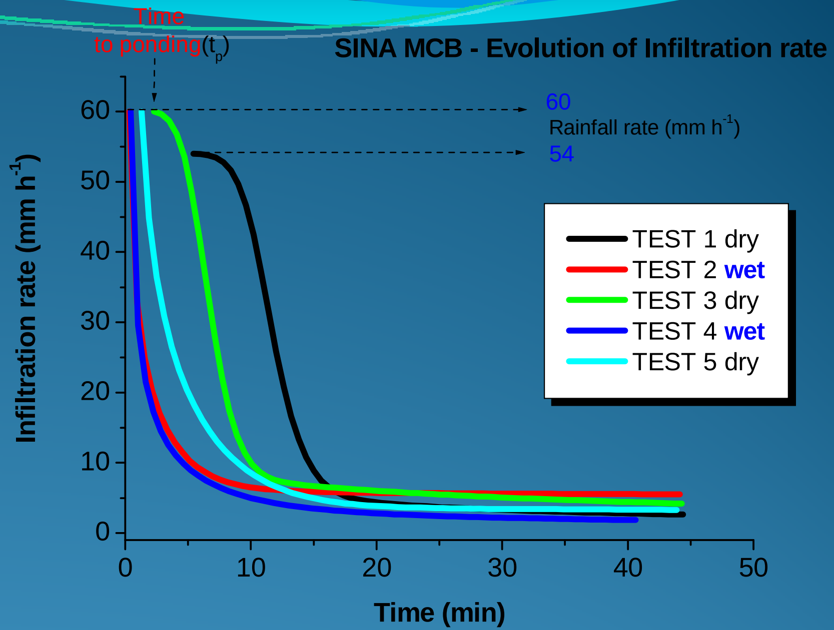
CR=0.7 (crossover probability)

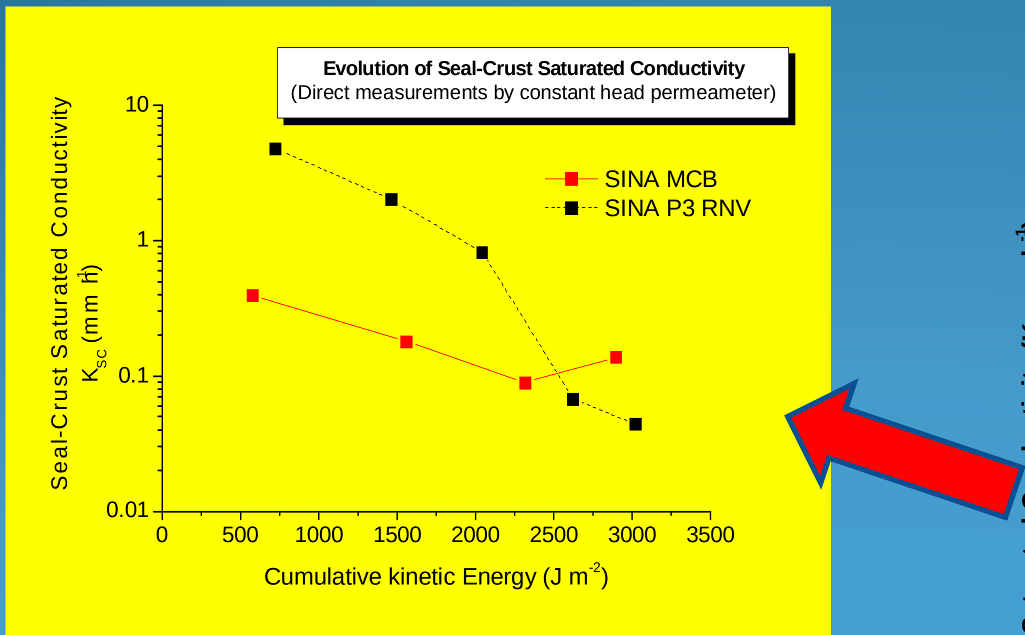
AG=30 (age cutoff)

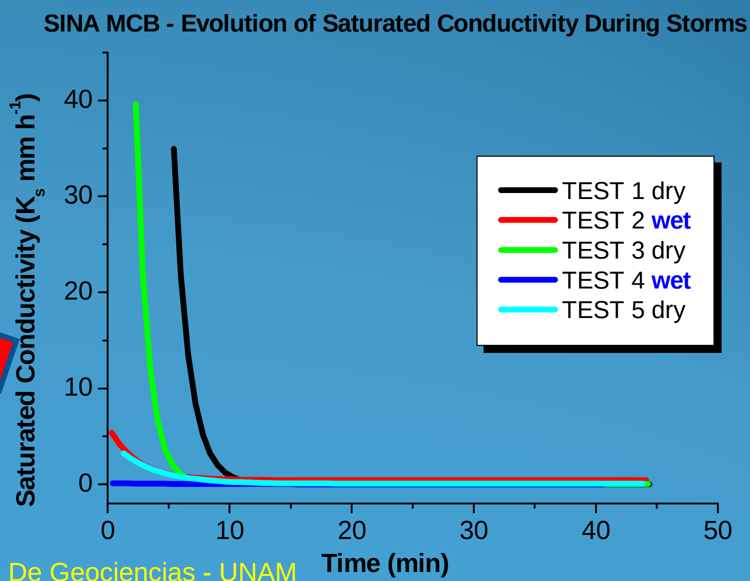
Tolerance 0.0000001

OBJECTIVE FUNCTION

$$obj = \sum_{i=1}^{n} \left(obs_i - pred_i \right)^2$$





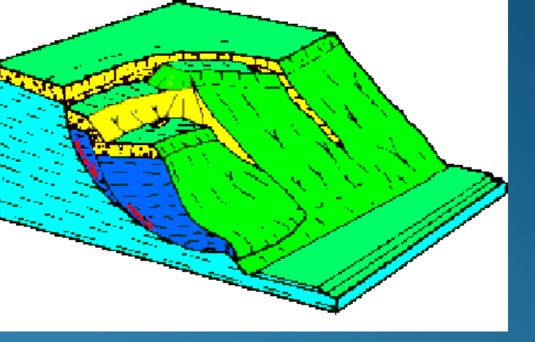


SSAP (SLOPE STABILITY ANALISYS SOFTWARE)

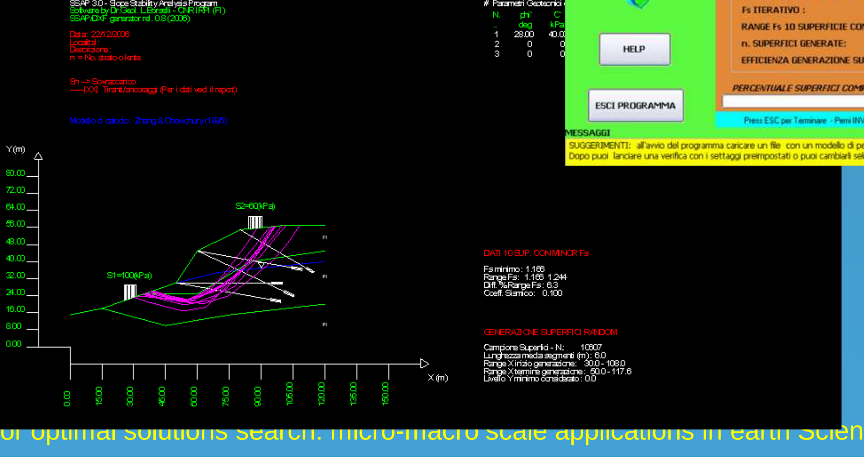
SOFTWARE for stability analysis (by limit equilibrium method) of natural and artificial slopes (soil and rock mass) with many components able to analyze hydraulic conditions and artificial Reinforcement effects, seismic action etc.

Users: geomorphologists, geologists , civil engineers, public institutions...

http://www.ssap2005.it

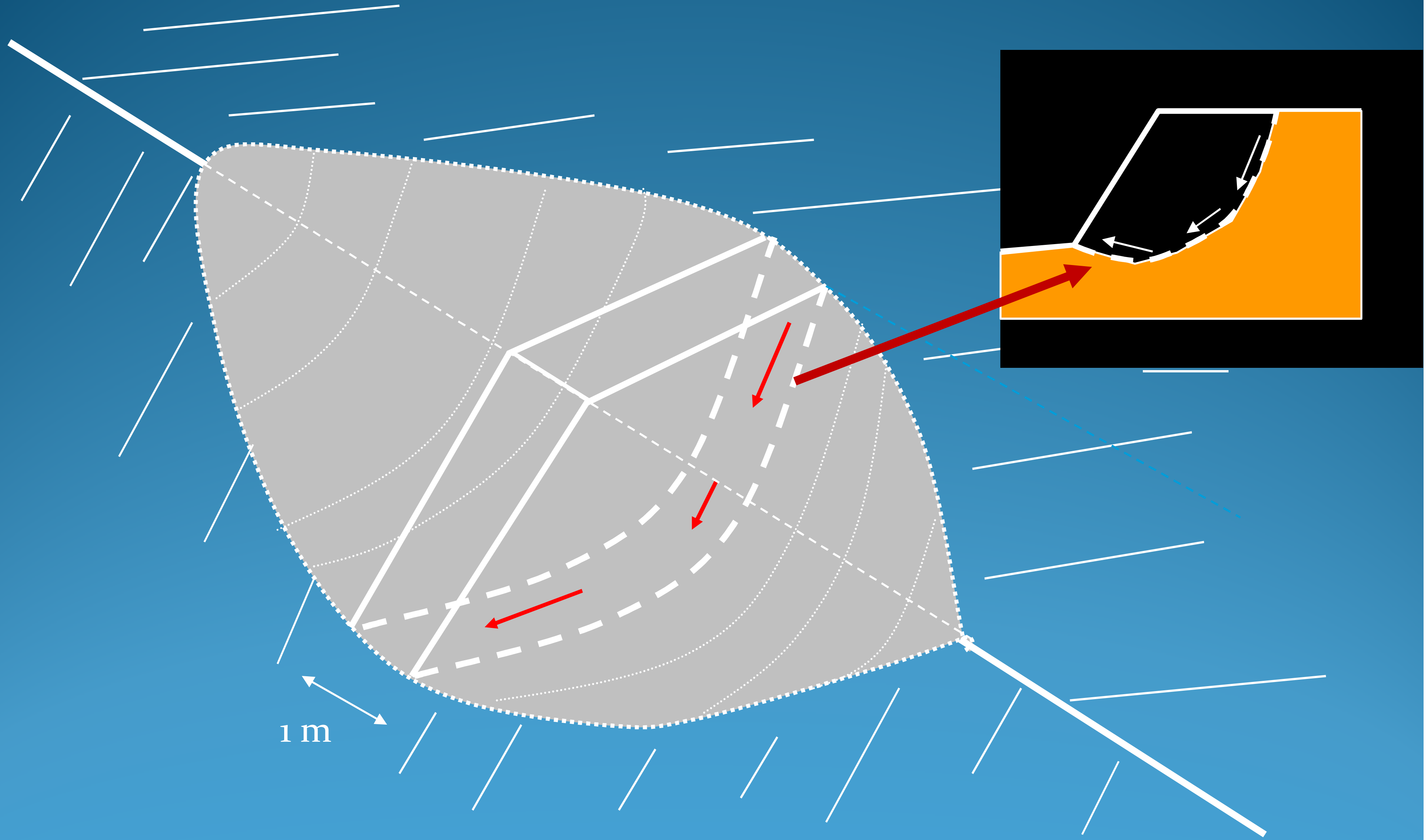


SSAP version: 3.0.2 2008



DE example of application – 4 - **SSAP software (2)**

2D slope stability analysis
SLOPE model with
1m unitary wide slice



DE example of application – 4 - **SSAP software (3)**

ITERATIVE procedure to FIND Stability factor Fs
INSTABLE if Fs <1.0

Equilibrium Horizontal forces components

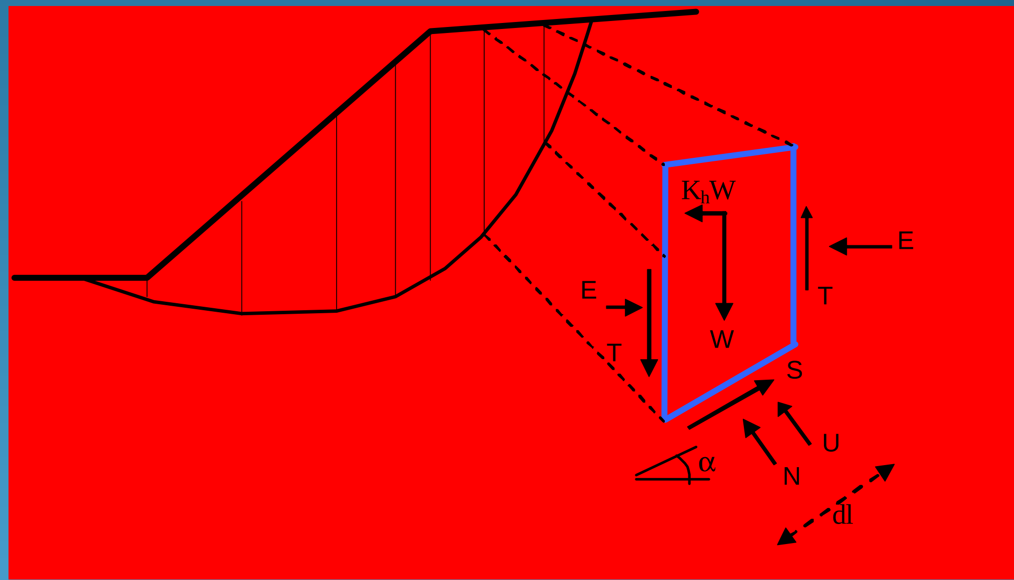
Generalized form Espinoza (1994)

$$F_S = \frac{\sum_i \{[W_i \cos\alpha_i - U_i l_i]\tan\phi'_i + c'_i l_i\} m_\alpha}{\sum_i W_i \sin\alpha_i m_\alpha + E_a - E_d + \sum_i \Delta Q_i + I_{ff}}$$

where:

$$m_\alpha = \frac{\sec\alpha_i}{1 + \frac{\tan\alpha_i \tan\phi'_i}{Fs}}$$

$$I_{ff} = \sum_i [\Delta T_i \tan(\phi'_m - \alpha_i)]$$



$$Q_i = k_h W_i - Q_{strutture}$$

Seismic coefficient
To account of sesmic effect on stability
$Kh, Kv = f(a_g max)$

DE example of application – 4 - **SSAP software (4)**

DE is used in SSAP 3.0.2 only to find
The value of Kh (seismic coefficient) that put
Stable slopes (FS>1) in critical condition of instability
FS=1.0 due to the seismic action.

Obj =| Fs-1.0|

$$F_S = \frac{\sum_i \{[W_i \cos\alpha_i - U_i l_i]\tan\phi'_i + c'_i l_i\} m_\alpha}{\sum_i W_i \sin\alpha_i m_\alpha + E_a - E_d + \sum_i \Delta Q_i + I_{ff}}$$

We have to find , by numerical method , the root of a equation where one term of it (FS) also must be obtained by a separate iterative numerical method

The DE algorithm applied to minimize the Given OBJ function allows to find the optimal *Kh*. In this case the minimum coincides with e Zero of the OBJ function

Special attention and management for the condition Where the FS computation don't Converge…

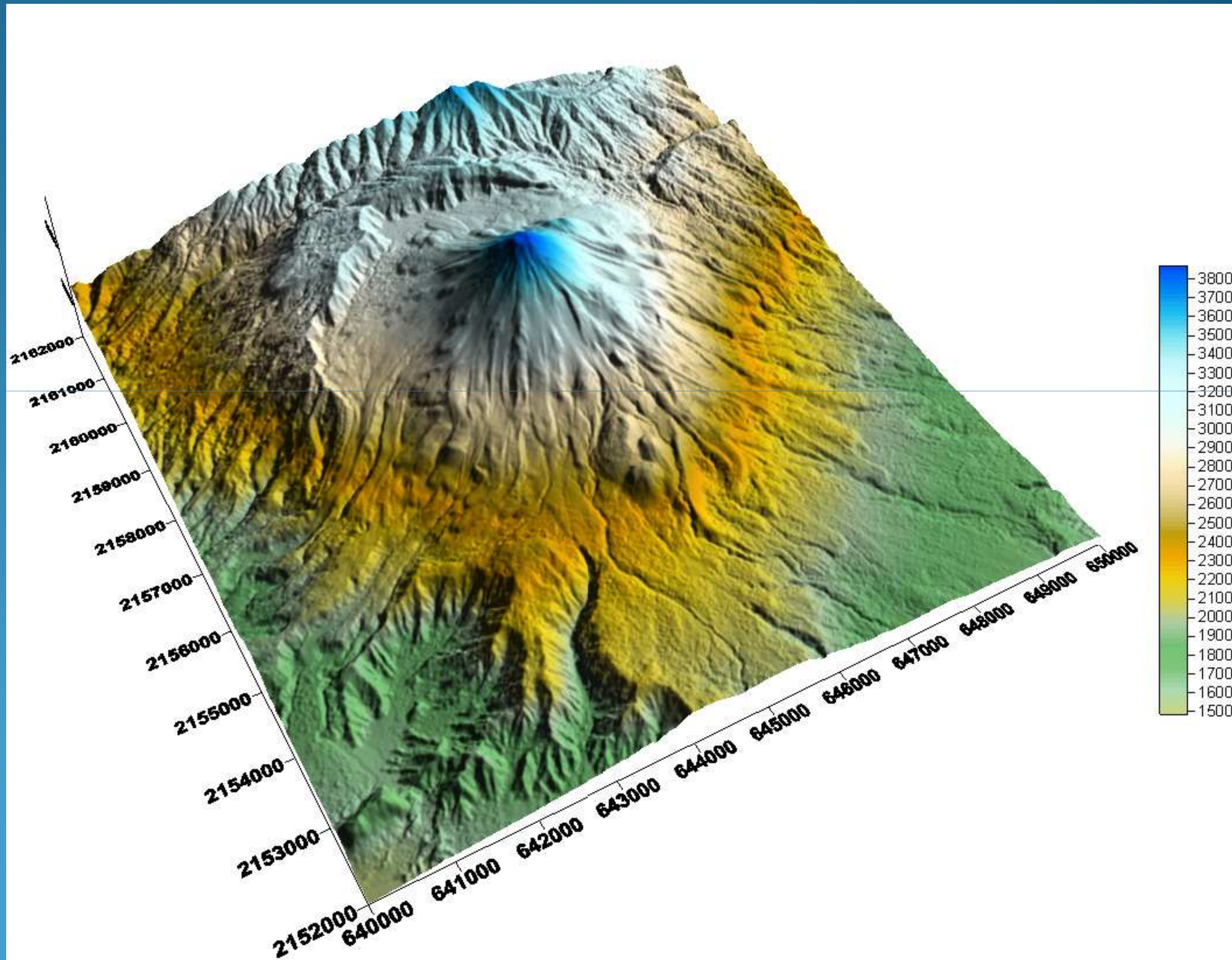
DE optimization parameters
D=1 (number of parameters)
N=50 (No. Of vectors in the population)
F=0.9 (weight factor in the mutation)
CR=0.9 (crossover probability)
AG=80 (age cutoff)
Tolerance 0.0001

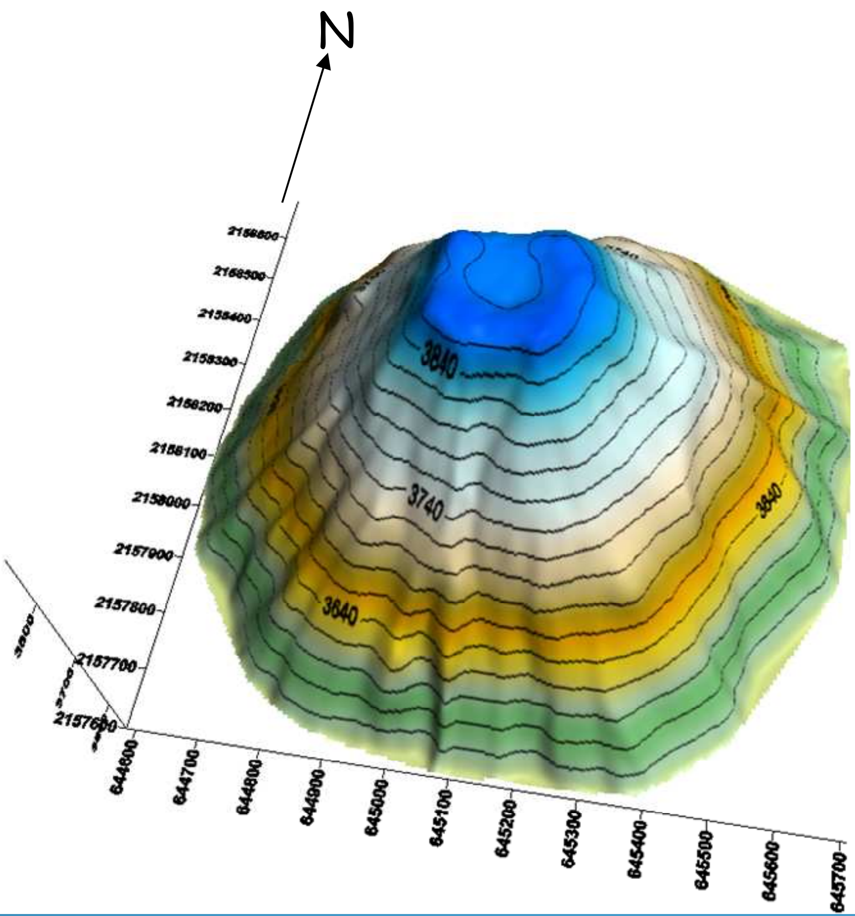
DE example of application – 5 - **VOLCANOFIT software (Borselli; 2005,2006)**

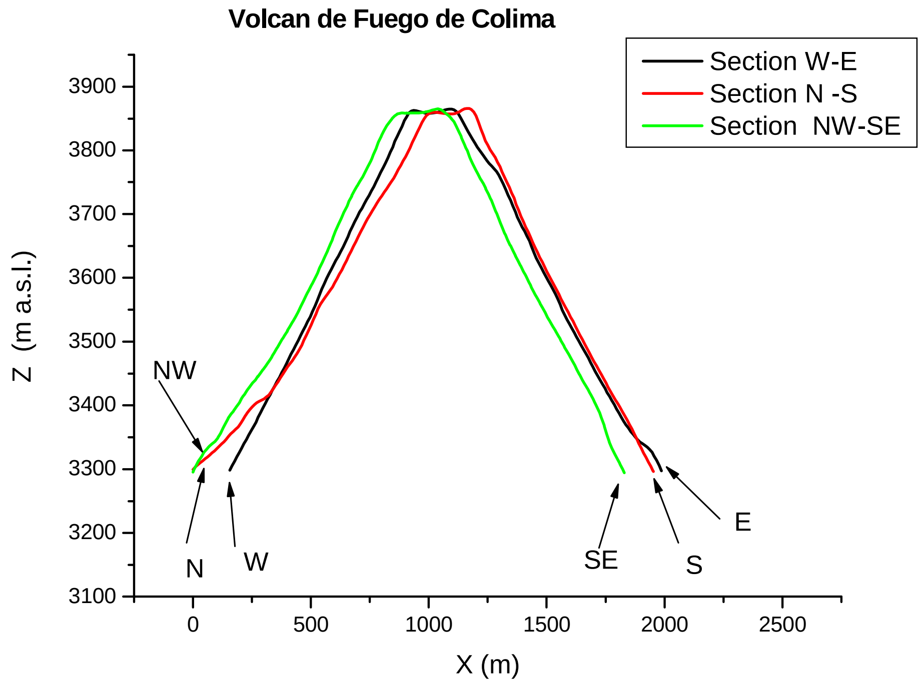
VOLCANOFIT - is a special surface fitting software made in occasion of a study , with D. Sarocchi , ans C. De la Crux, on the stability of the volcan de fuego de Colima Using also SSAP…

- Fitting of original DEM with a Truncated conical surface

- The analysis of local Mass excess, or deficit, in this portion of volcanic edifice is finally done by the computation of the residual surface after the fitting







Profiles at top of Colima volcano

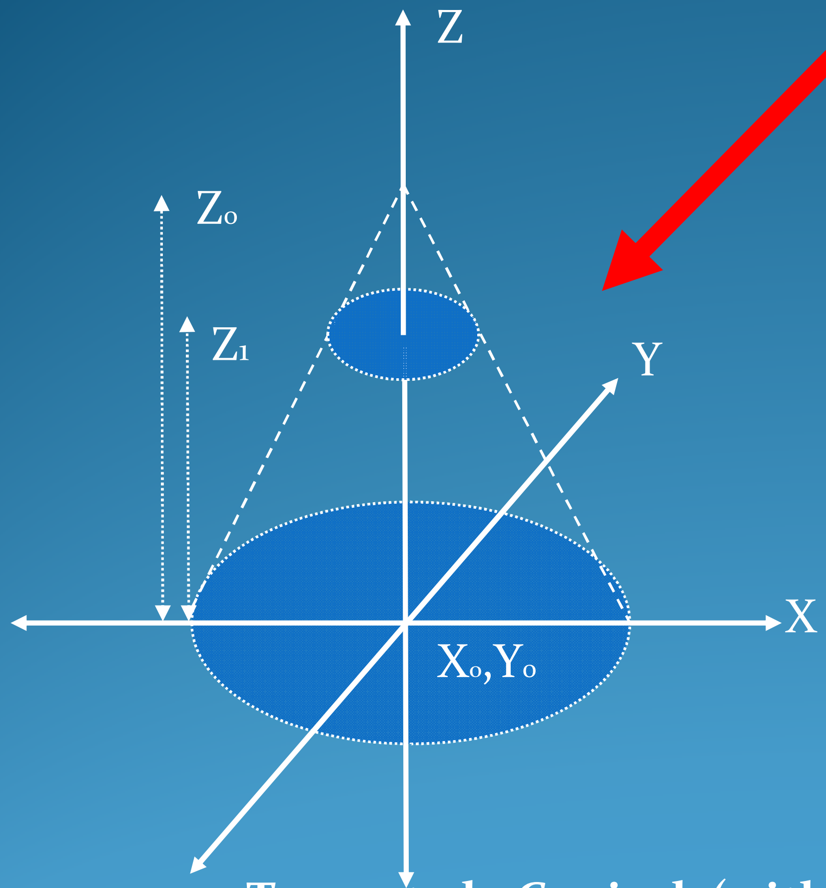
Top of Colima volcano
Above 3540 m s.l.m.

DE example of application – 5 - **VOLCANOFIT software (3)**

VOLCANOFIT fitting function
(case of Colima volcan de fuego)

$$Z = z_0 - c\sqrt{\frac{(X - x_0)^2}{a} + \frac{(Y - y_0)^2}{b}}$$

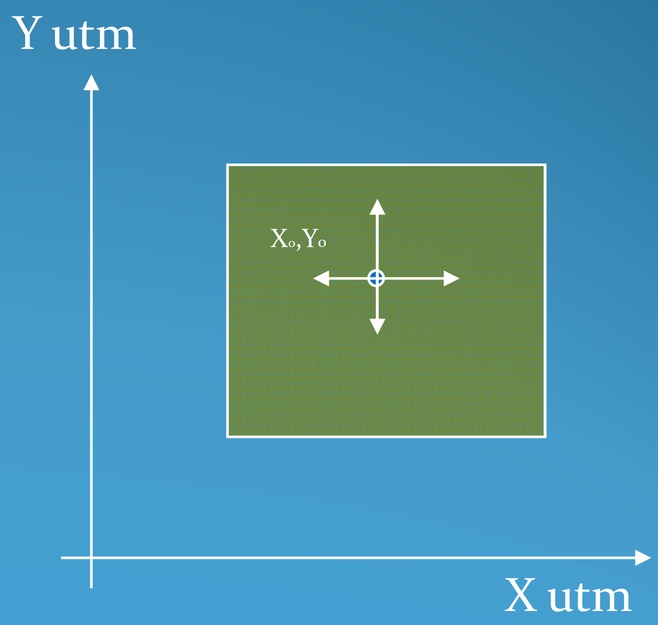
$$\text{if} \quad Z > z_1 \rightarrow Z = z_1$$



OBJECTIVE FUNCTION

$$obj = \sum_{i=1}^{n} \left| obs_i - pred_i \right|$$

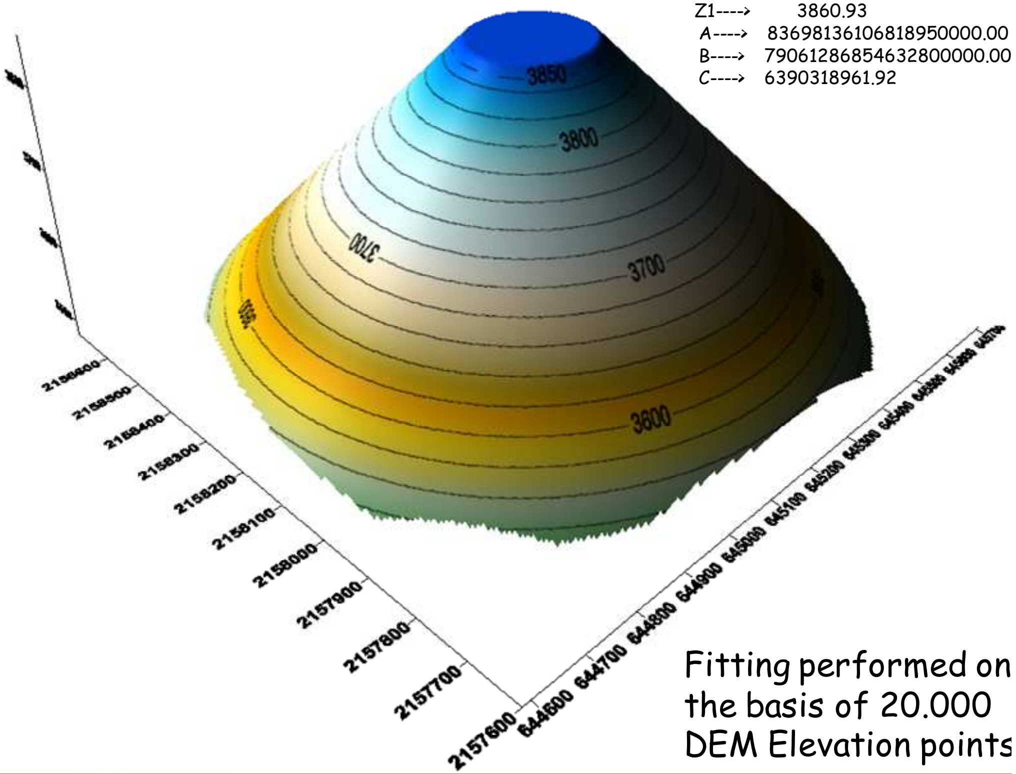
Centered on coordinate X_o, Y_o



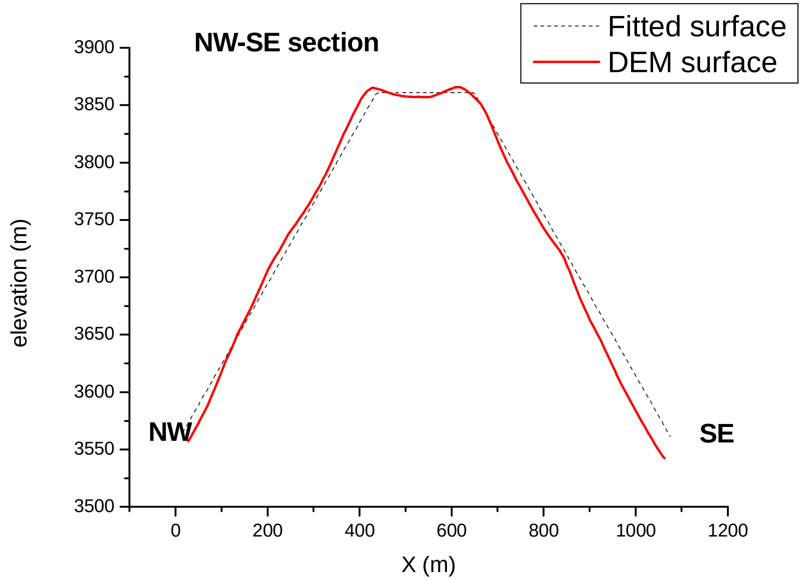
Truncated - Conical (with ellliptic base) fitting surface
Seven parameters :
X_o,Y_o,Z_o,a,b,c and Z_1

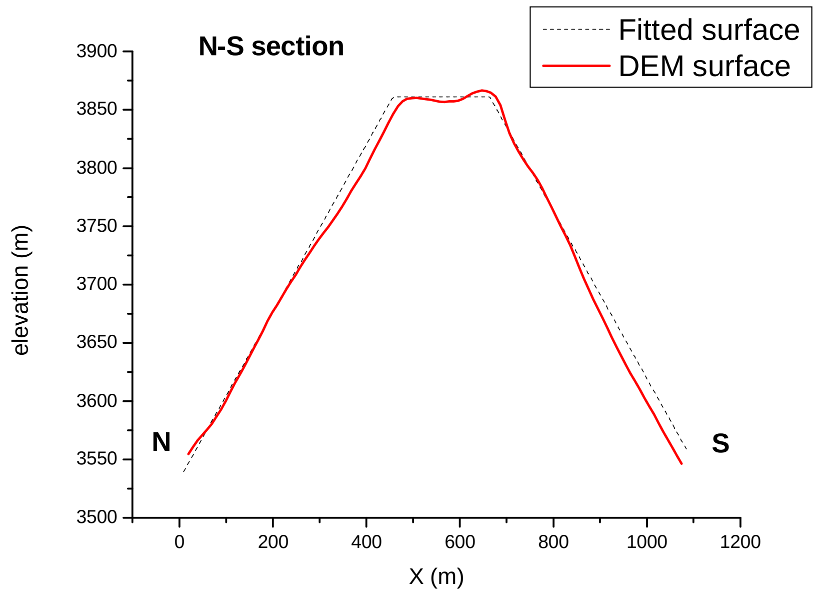
DE example of application – 5 - VOLCANOFIT software (4)

VOLOCANOFIT 0.1
-- list of Optimized parameters ---
X0----> 645135.65
Y0----> 2158134.14
Z0----> 3935.48
Z1----> 3860.93
A----> 836981361068189500000.00
B----> 790612868546328000000.00
C----> 6390318961.92



Fitting performed on the basis of 20.000 DEM Elevation points

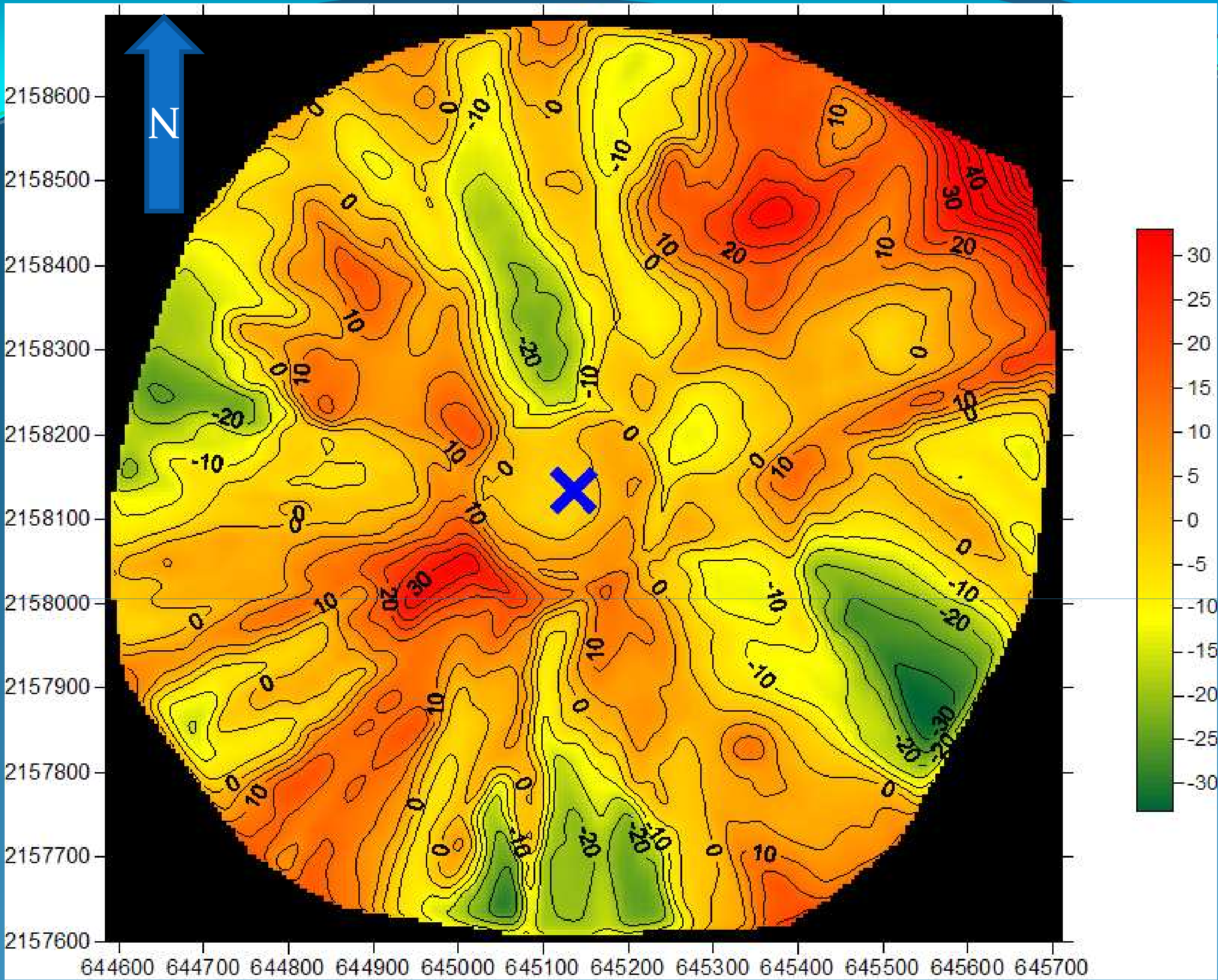




DE optimization parameters

D=7 (number of parameters)
N=250 (No. Of vectors in the population)
F=0.9 (weight factor in the mutation)
CR=0.9 (crossover probability)
AG=30 (age cutoff)
Tolerance 0.0005

DE example of application – 5 - VOLCANOFIT software (5)



In RED the major local excess of mass.

In GREEN the local deficit of mass

The **Blue Cross** is the centre of the reference fitted conical surface. It is assumed as the centre of simmetry of the upper part of the volcano

Residual surface = DEM surface NOV. 2004 – Fitted truncated cone surface)

Basic REFERENCES for Differential evolution algorithm and global optimization

•*Lampinen, J. and Zelinka, I. (2000). On Stagnation of the Differential Evolution Algorithm. In: Ošmera, P. (ed.) 2000. Proceedings of MENDEL 2000, 6th Int. Conf. On Soft Computing, June 7.–9. 2000, Brno University of Technology, Brno, Czech Republic, pp. 76–83. ISBN 80-214-1609-2. Available via Internet: ttp://www.lut.fi/~jlampine/MEND2000.ps*

• *Lampinen, J. 2001. A Bibliography of Differential Evolution Algorithm. Technical Report. Lappeenranta University of Technology, Department of Information Technology,Laboratory of Information Processing. Available via Internet:http://www.lut.fi/~jlampine/debiblio.htm .*

•*Storn, R. and Price, K., 1997a. Differential Evolution: Numerical Optimization Made Easy, Dr. Dobb's Journal, April 97, 18 - 24.*

•*Corne D., Dorigo M., Glover F. 1999. New Ideas in Optimization. McGrow-Hill. Pages 450. ISBN: 0077095065*

•*Storn R.and Price K., 1997b. Differential Evolution-a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, 11: 341-359.*

•*Storn R., 1999. System design by Constraint Adaptation and Differential Evolution, IEEE Transactions on Evolutionary Computation, 3(1): 22-34.*

•*Sen, M. and Stoffa, P.L. 1995. Global Optimization Methods in Geophysics, Elsevier.*

•*Schwefel, H. P. 1995. Evolution and Optimum Seeking, John Wiley*

•*Törn, A. and Žilinkas, A. 1989. Global Optimization, Springer-Verlag*