

GENETIC ALGORITHM FOR GOSIA: present status and future



Paweł J. Napiorkowski and Wojciech Piątek
HIL UW



Warsaw, 9-11 April, 2018

WHY GENETIC ALGORITHM FOR GOSIA

Local minima problem

- Random multi-start gradient search (OP,RAND)
- Simulated Annealing implemented into GOSIA by R.W.Ibbotson (OP,ANNL)
- Genetic Algorithm

14 1. GAs: What Are They?

The idea behind genetic algorithms is to do what nature does. Let us take rabbits as an example: at any given time there is a population of rabbits. Some of them are faster and smarter than other rabbits. These faster, smarter rabbits are less likely to be eaten by foxes, and therefore more of them survive to do what rabbits do best: make more rabbits. Of course, some of the slower, dumber rabbits will survive just because they are lucky. This surviving population of rabbits starts breeding. The breeding results in a good mixture of rabbit genetic material: some slow rabbits breed with fast rabbits, some fast with fast, some smart rabbits with dumb rabbits, and so on. And on the top of that, nature throws in a 'wild hare' every once in a while by mutating some of the rabbit genetic material. The resulting baby rabbits will (on average) be faster and smarter than these in the original population because more faster, smarter parents survived the foxes. (It is a good thing that the foxes are undergoing similar process — otherwise the rabbits might become too fast and smart for the foxes to catch any of them).

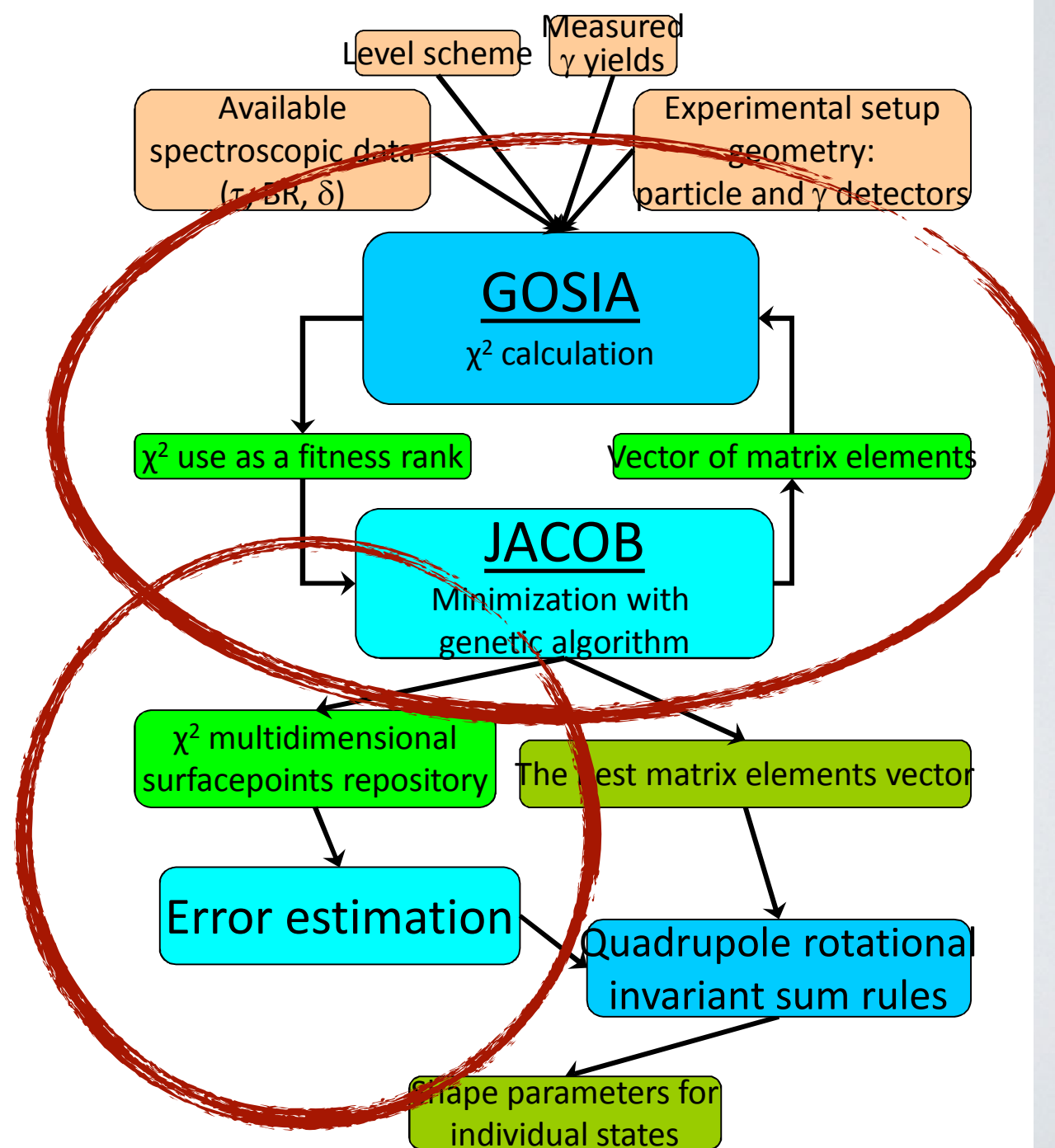
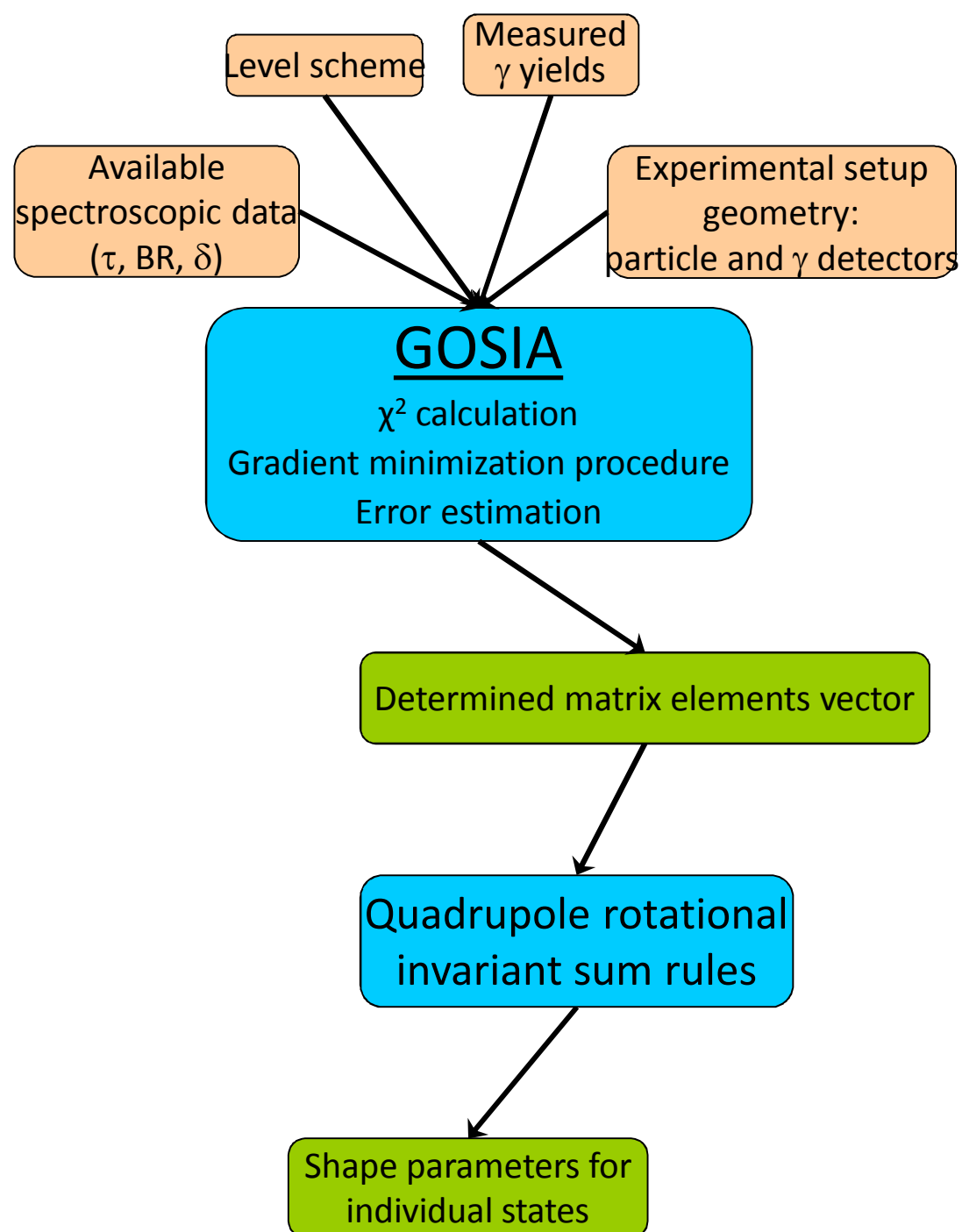
A genetic algorithm follows a step-by-step procedure that closely matches the story of the rabbits. Before we take a closer look at the structure of a genetic algorithm, let us have a quick look at the history of genetics (from [380]):

"The fundamental principle of natural selection as the main evolutionary principle has been formulated by C. Darwin long before the discovery of genetic mechanisms. Ignorant of the basic heredity principles, Darwin hypothesized fusion or blending inheritance, supposing that parental qualities mix together like fluids in the offspring

Zbigniew Michalewicz

Genetic Algorithms + Data Structures = Evolution Programs
Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2013

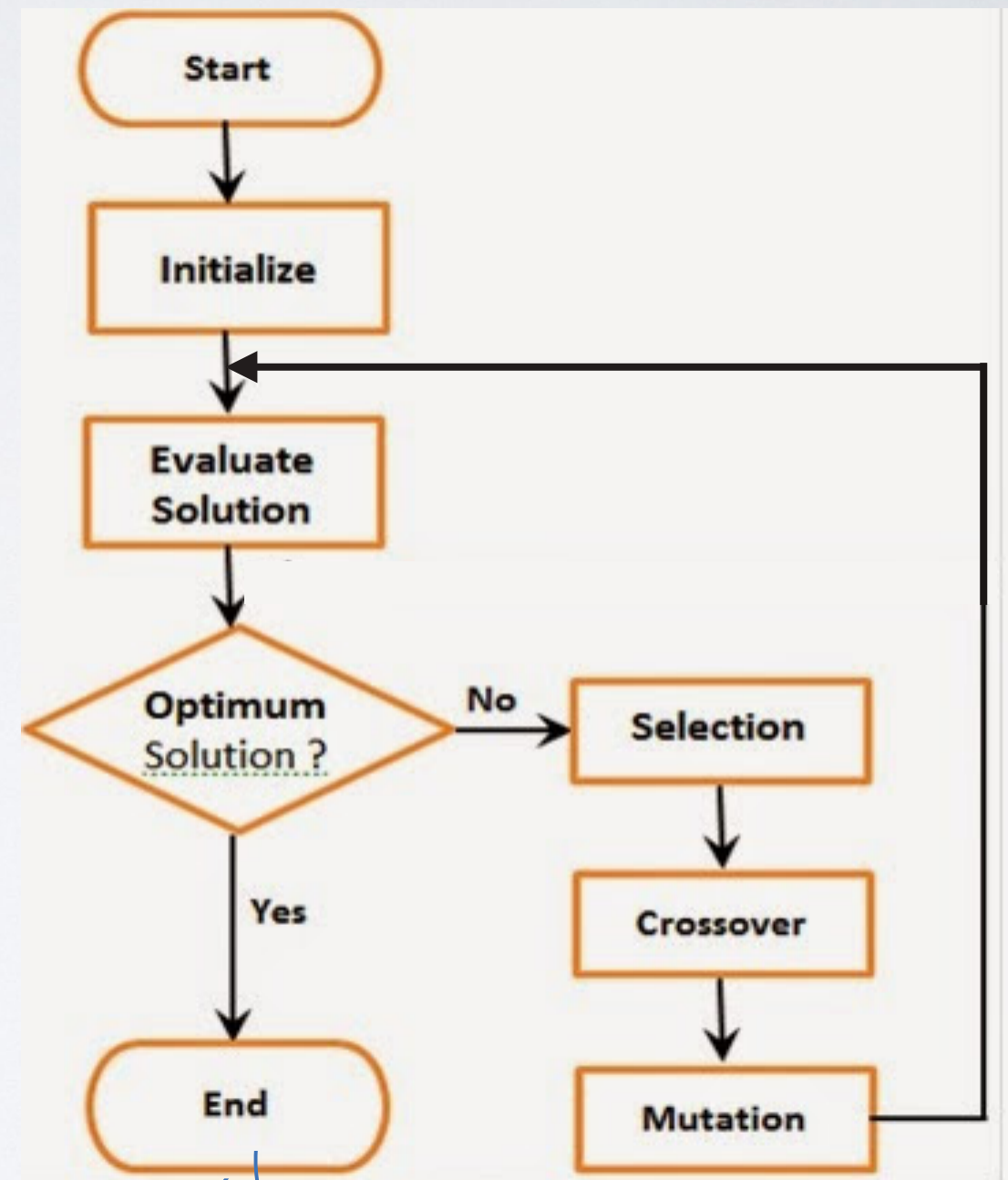
JACOB: GENETIC WITH GOSIA



GENETIC ALGORITHM

how to do

- Generation of population (sets of ME)
 1. Ranking with fitness function
(calculation of χ^2 for all sets of ME)
 2. Selection
(deleting of sets of ME with bad χ^2)
 3. Creating new sets of ME crossing
 4. Mutation: random change of ME
 5. GoTo 1.
- Stop: user satisfaction

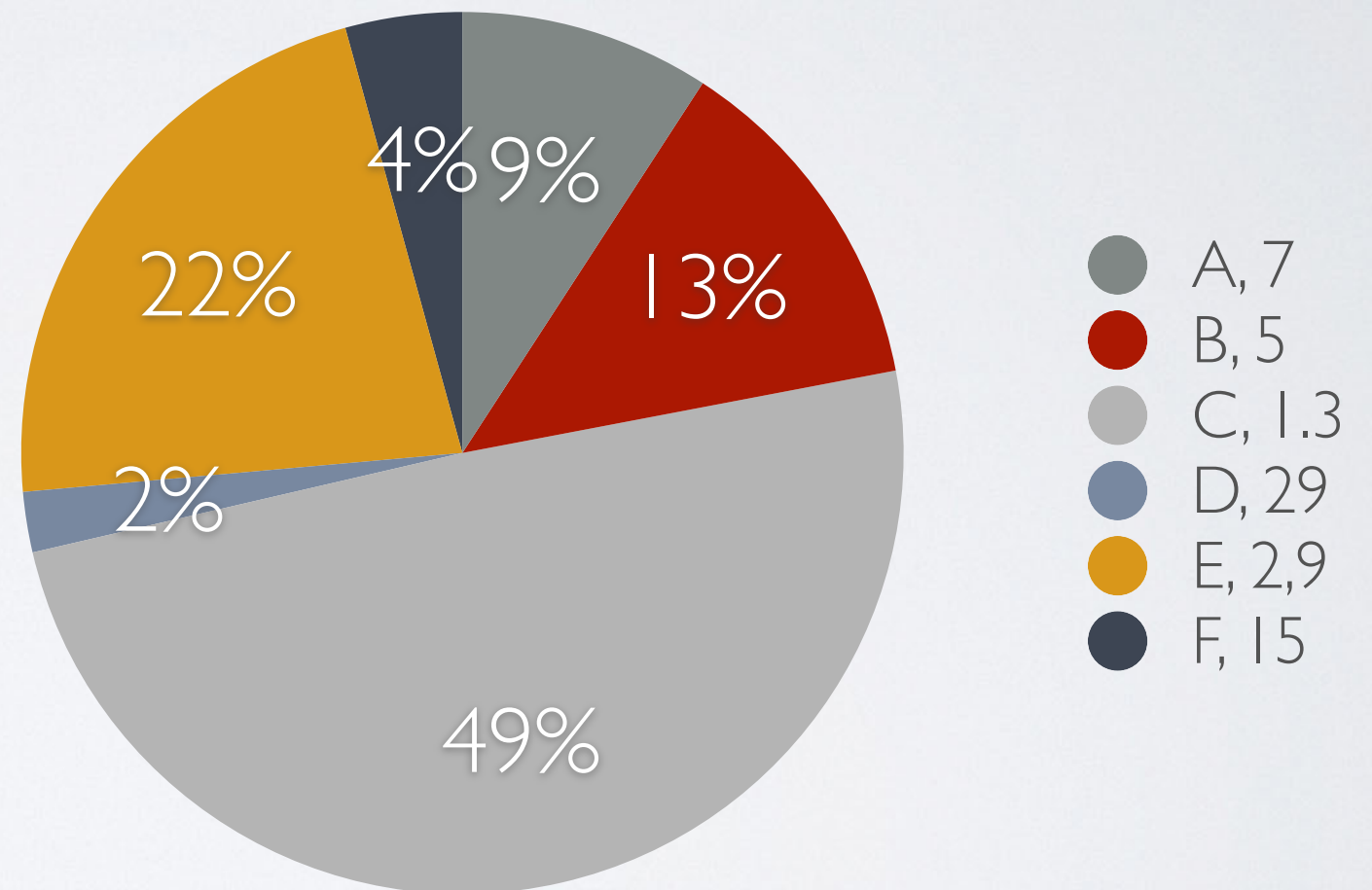


SELECTION

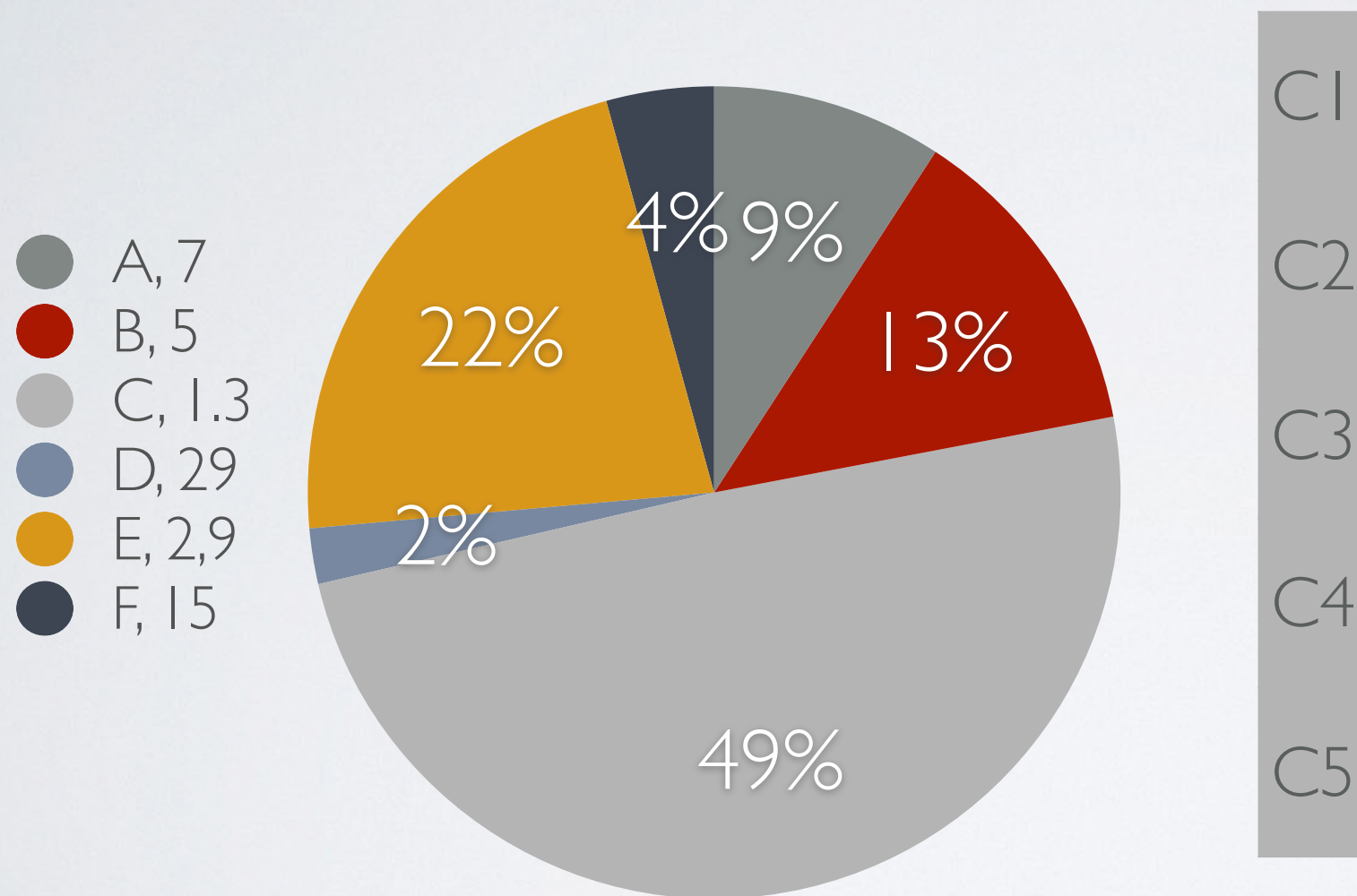
- Truncation
- Roulette
- Tournament

SELECTION

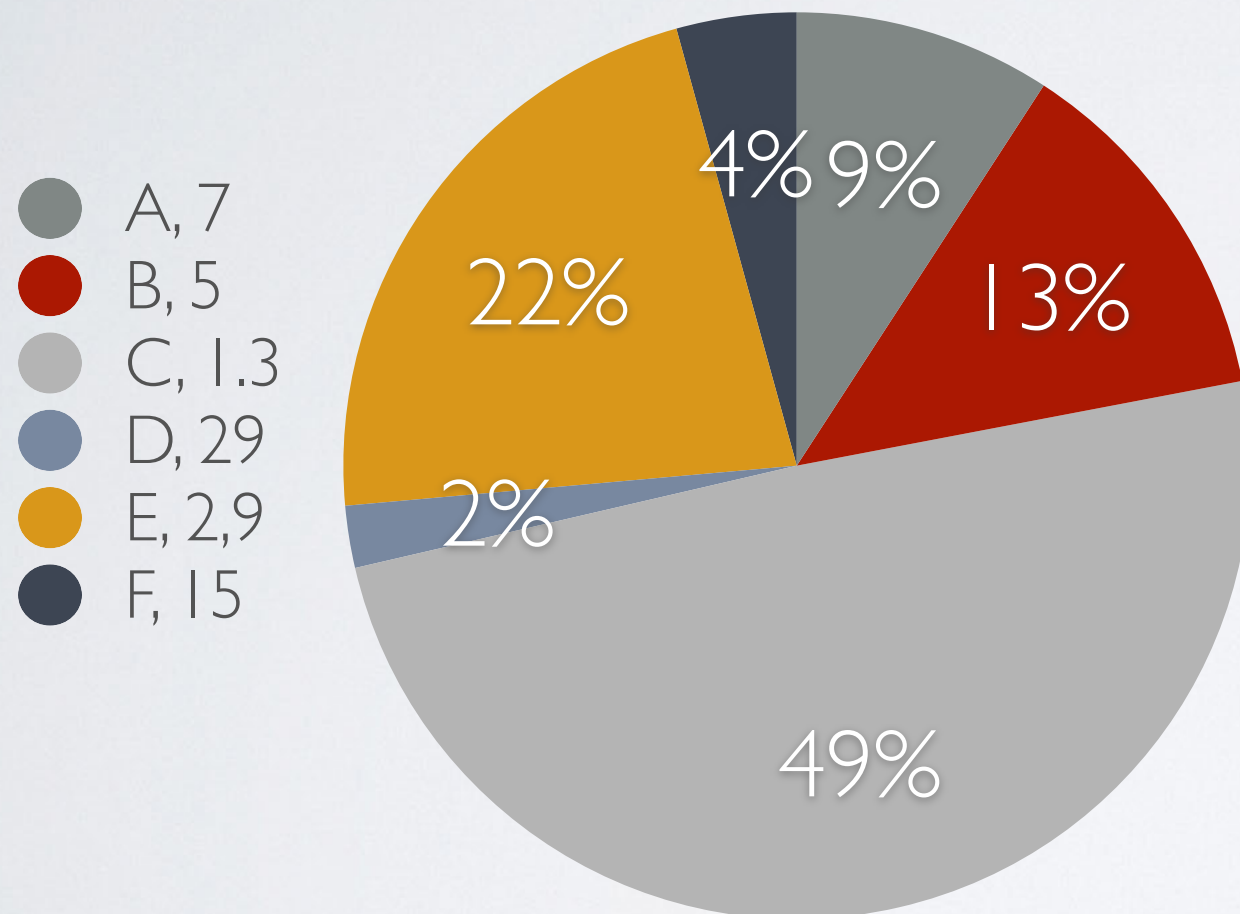
- Truncation
- **Roulette**
- Tournament



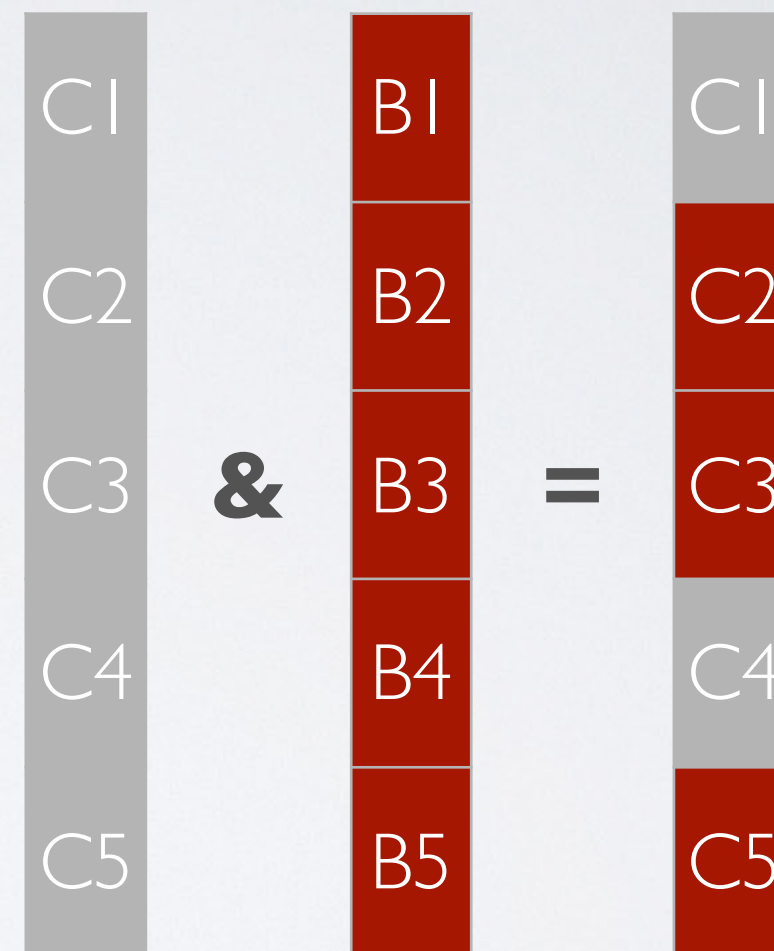
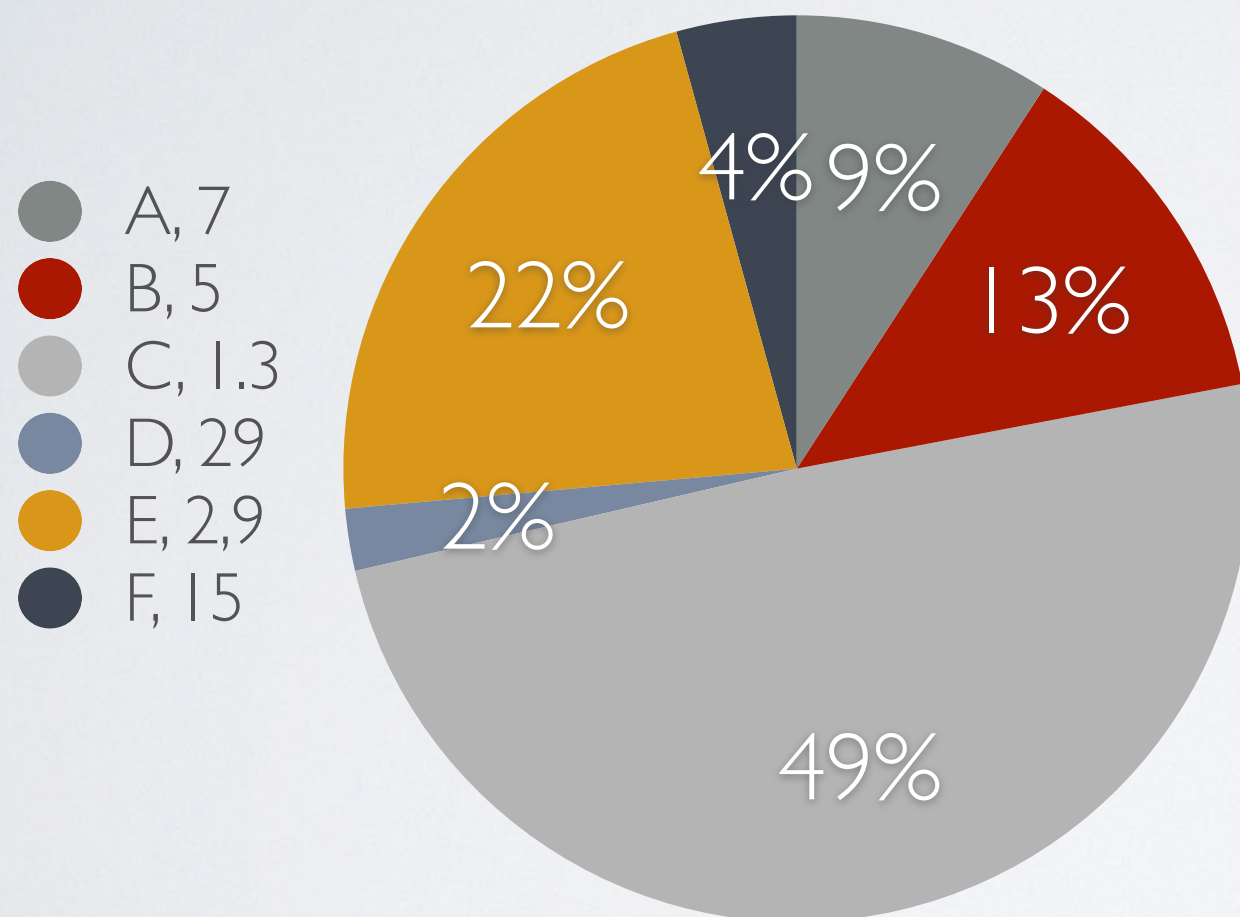
CROSSING



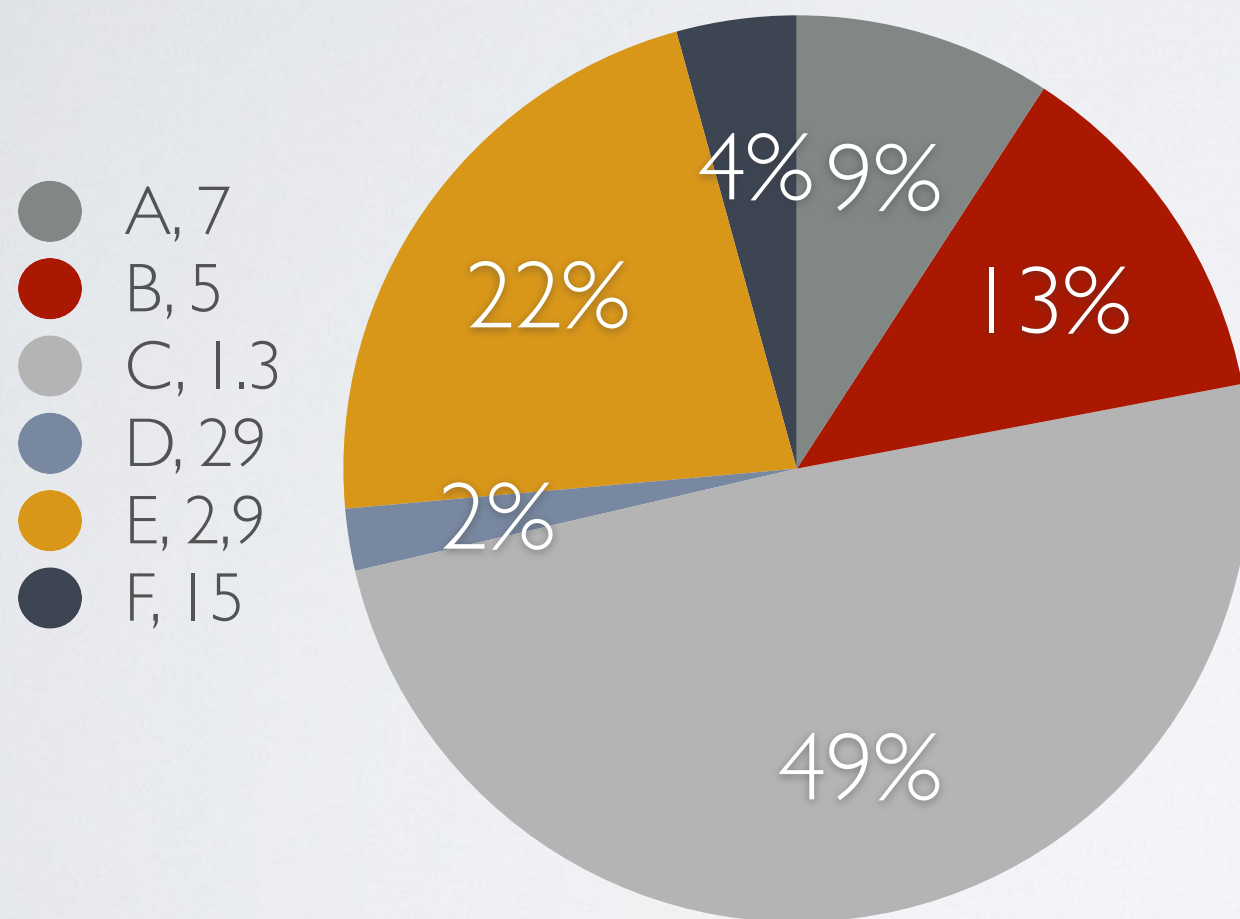
CROSSING



CROSSING



MUTATION



TO FIND MINIMUM/MINIMA

- Tested with 6D test function:
F7 - The Schaffer Function.

$$F7(x_{1,\dots,6}) = \sum_{i=1}^6 a_i x_i \sin(\sqrt{|a_i x_i|})$$

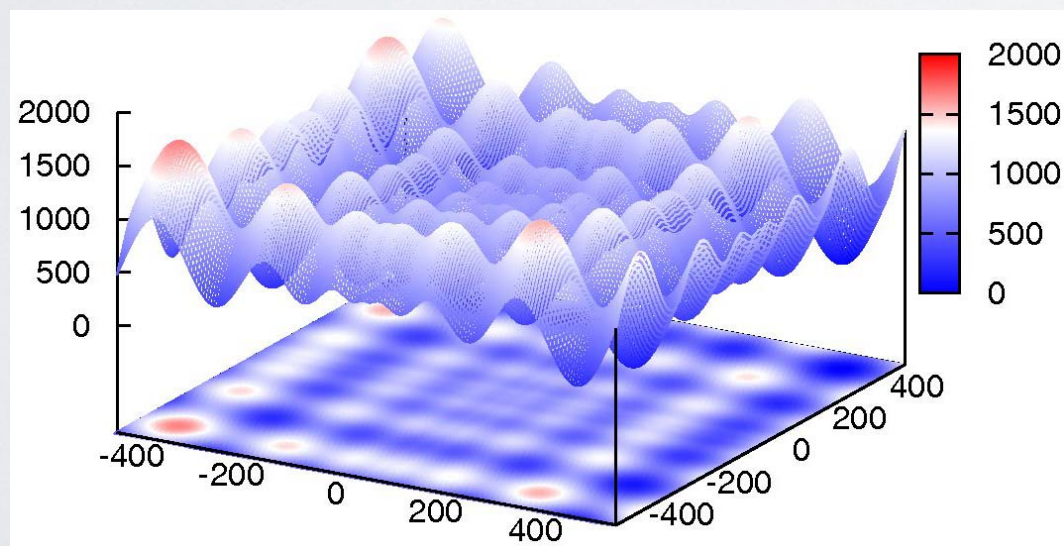
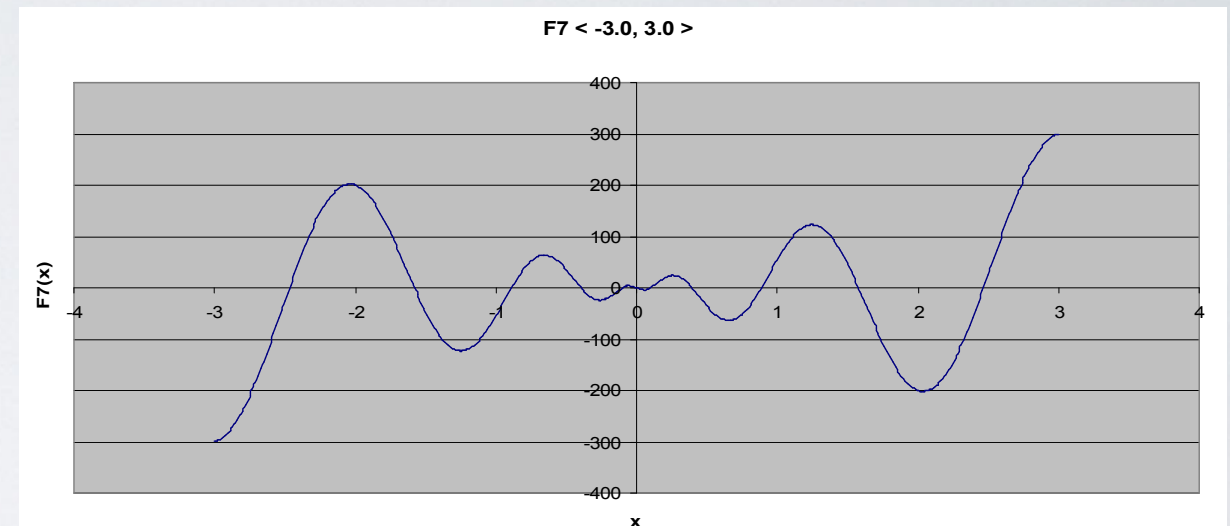
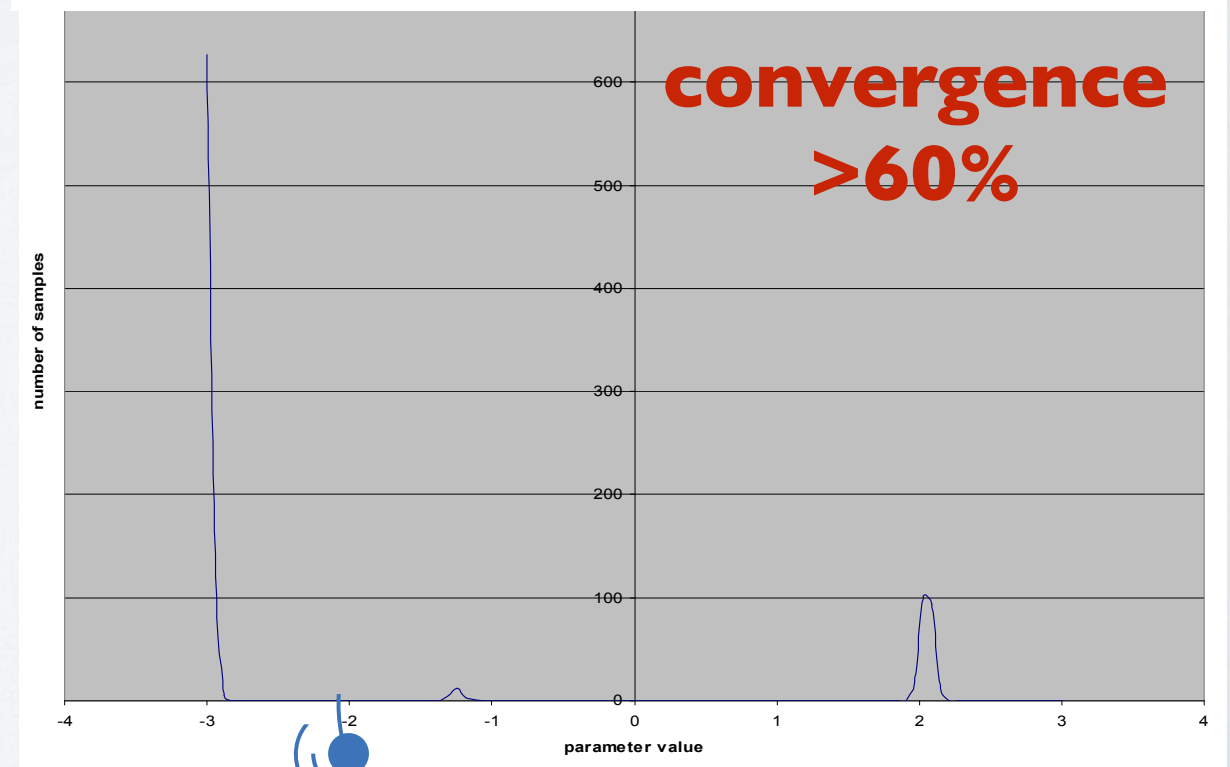


Figure 7. 2D plot of Schwefel's function.



Values found in minimisation



JACOB - GENETIC ALGORITHM IMPLEMENTATION

- the same data input scheme as in GOSIA
- resistance to the local minimum problem:
automatic scan for alternative solutions and expected better convergence to the global minimum
- error estimation based on the scan of the χ^2 surface
- parallel processing on several clustered computers connected into a network.
- the first version: for MS Windows with .NET technology
- recently: for Linux with QT library
- with graphic interface/GUI and console mode

Jacob v2.1

Simulation

Population size: 150

Best chiSq: 9,811

Average chiSq: 1828306.

Best ME vector: 0,206

Algorithm settings

Selection: truncation

Casualties [%]: 60

One generation

Number of genera

Cancel

Time measurement

Time of generation

Time of Gosia call:

Number of Gosia call

Time of selection:

Time of crossing over

Time of mutation:

Time of others:

Generation number: 1225

Generation: 193

Best ChiSq: 0.711367

ME[0]: 1.01525

ME[1]: 0.316926

ME[2]: 1.65967

ME[3]: 1.05255

Average ChiSq: 43.5546

SYMULATION MENU:

4 - Stop symulation

5 - Show time measurement

Time Of Generation: 0.06 s

Time Of Gosia Call: 0.00142857 s

Number Of Gosia Calls: 35

Time Of Selection: 0 s

Time Of Crossing Over: 0 s

Time Of Mutation: 0 s

Number Of Generations: 193

JACOB

how to run

But:
integration in a given point

```
OP, FILE
22,3,1
jacob.o
12,3,1
jacob.me
9,3,1
osiris.f9
8,3,1
osiris.f8
3,3,1
120te-data-v6.f3
4,3,1
120te-data-v6.f4
15,3,1
120te-error.f15
0 0 0
[...]
OP, MINI
2100,1,1.0E+34,.00001,1.1,1,0,1,1,0.001
OP, EXIT
```

Time of calculation: a few hours

JACOB

how to run

```
OP,FILE
22,3,1
jacob.o
12,3,1
jacob.me
9,3,1
osiris.f9
8,3,1
osiris.f8
3,3,1
120te-data-v6.f3
4,3,1
120te-data-v6.f4
15,3,1
120te-error.f15
0 0 0
[...]
OP,MINI
2100,1,1.0E+34,.00001,1.1,1,0,1,1,0.001
OP,EXIT
```

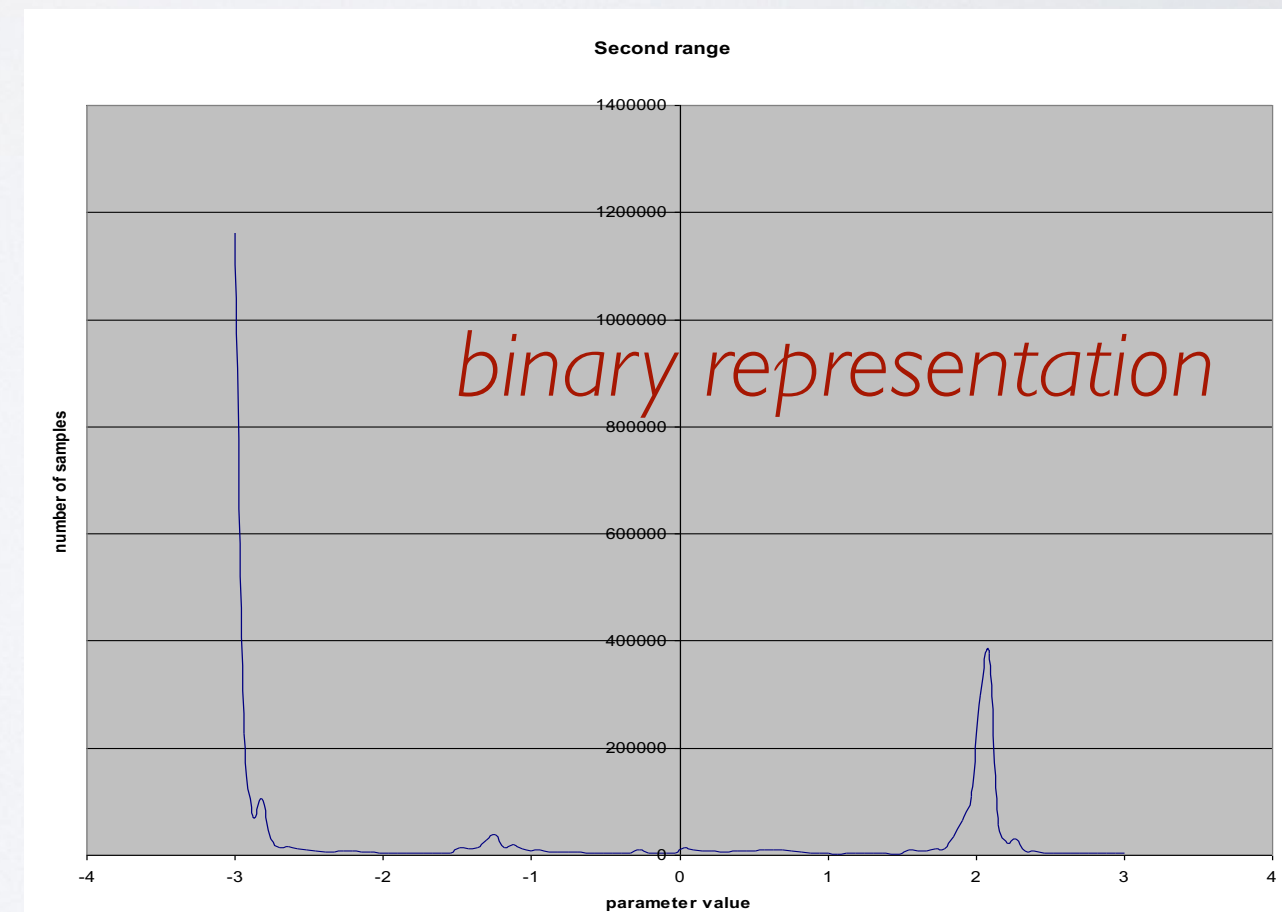
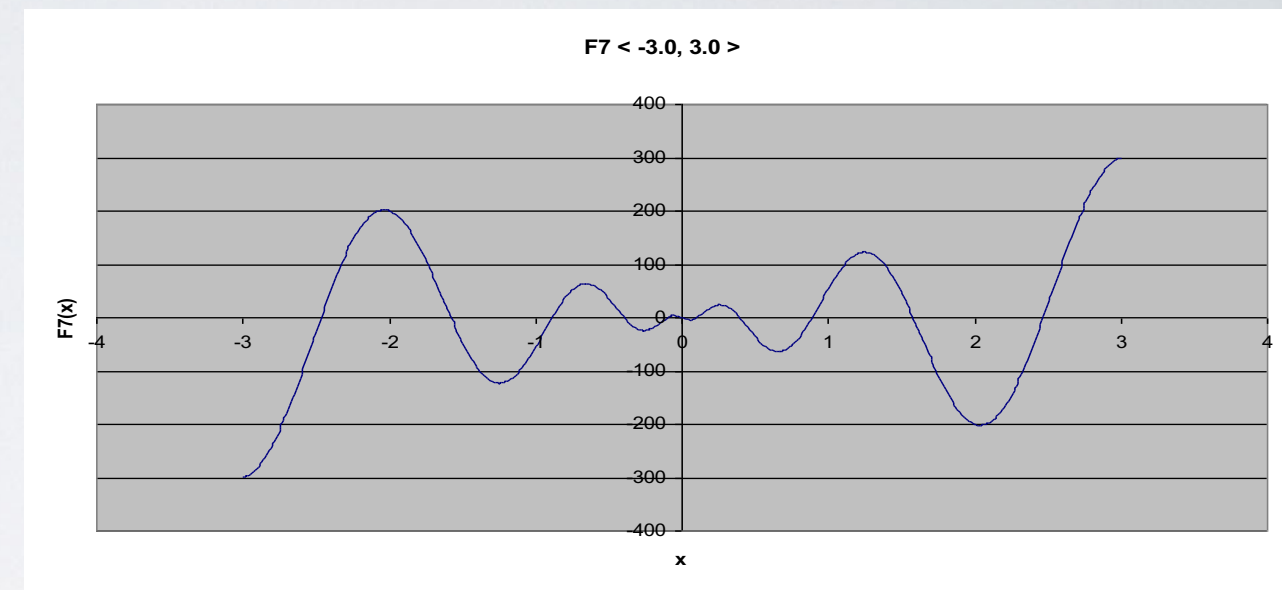
Time of calculation: a few days

Integration can be performed
in every point

```
OP,FILE
22,3,1
jacob.o
12,3,1
jacob.me
9,3,1
[...]
OP,YIEL
[...]
3 !NTAP - tape number
[...]
OP,INTG
[...]
OP,YIEL
[...]
4 !NTAP - tape number
[...]
OP,MINI
2100,1,1.0E+34,.00001,1.1,1,0,1,1,0.001
OP,EXIT
```

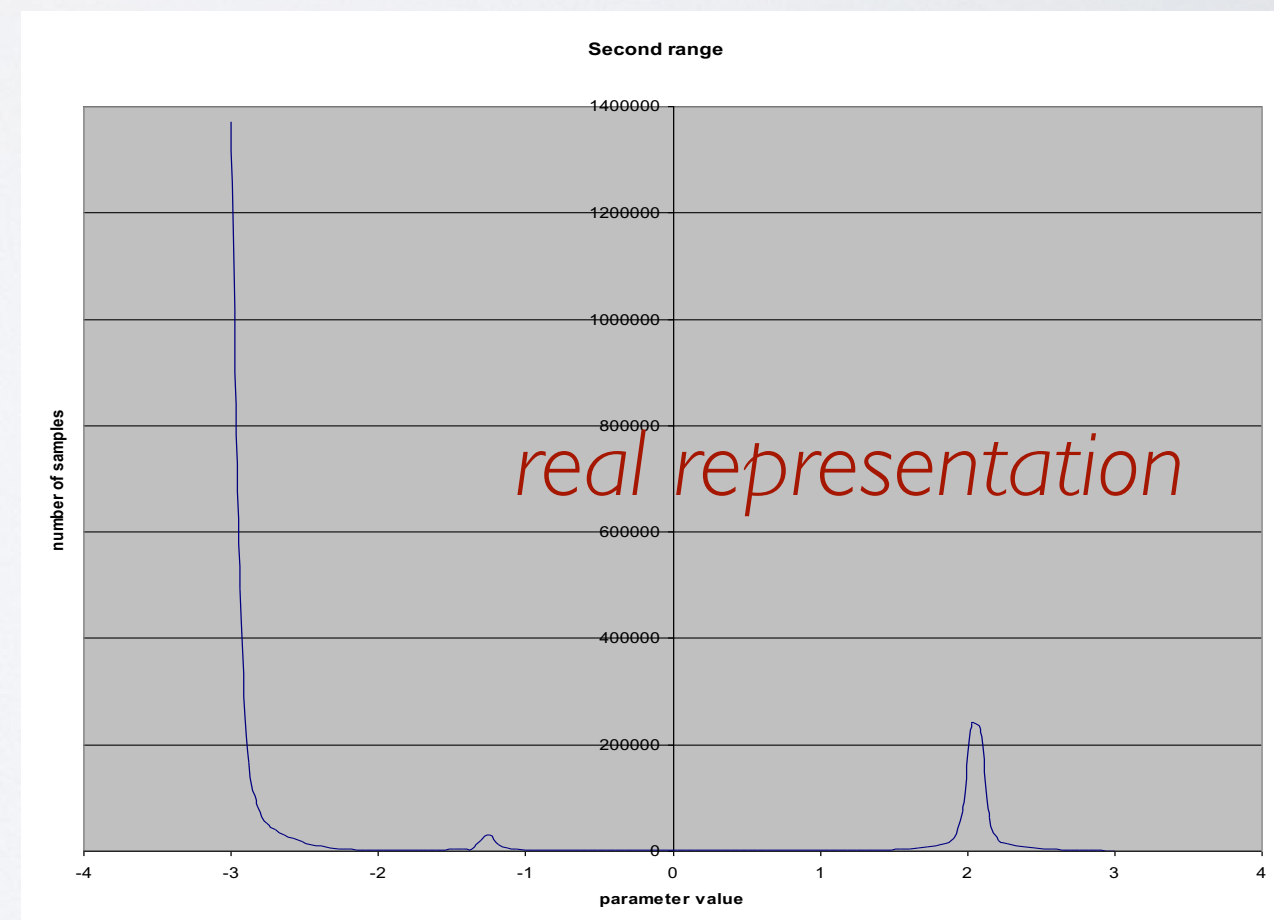
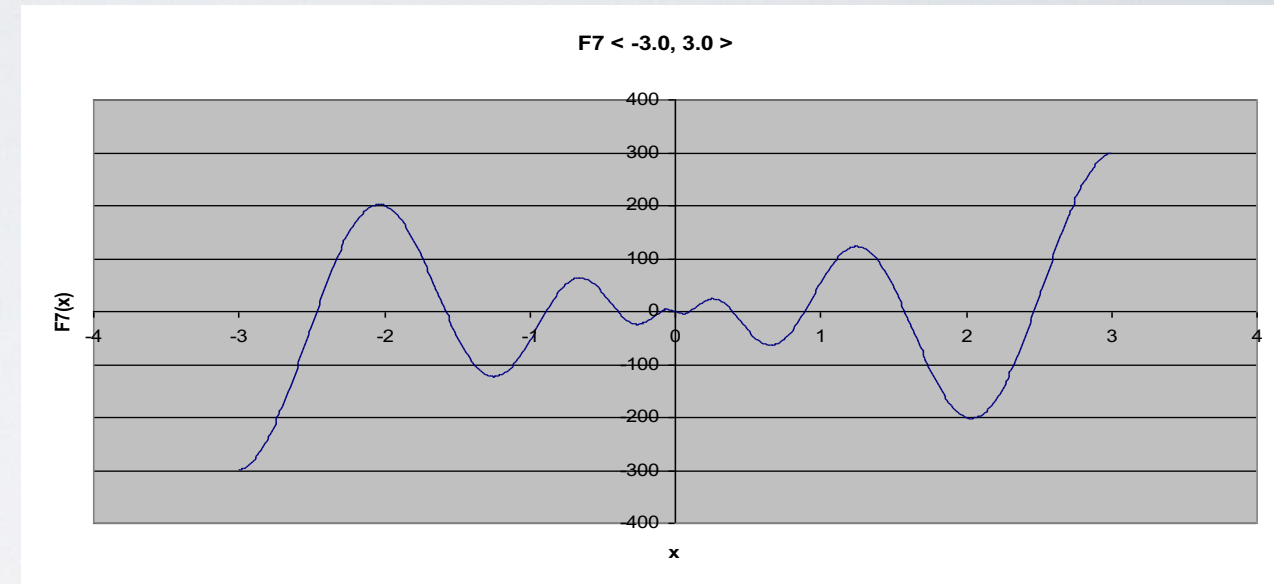
TO CHECK ALTERNATIVE SOLUTIONS

- Repository of the all tested χ^2 values
- QRepScan tool
 - find the best point to continue minimisation with GOSIA
 - scan the χ^2 surface for alternative solutions



TO CHECK ALTERNATIVE SOLUTIONS

- Repository of the all tested χ^2 values
- QRepScan tool
 - find the best point to continue minimisation with GOSIA
 - scan the χ^2 surface for alternative solutions



TO CHECK A SOLU

- Repository of the all tested χ^2 values

- QRepScan** tool

- find the best point to continue minimisation with GOSIA
- scan the χ^2 surface for alternative solutions

```
xterm
0.2501 267 -3.75 337 0.625 510 0.625 750
0.3501 324 -3.25 408 0.875 605 0.875 872
0.4501 369 -2.75 448 1.125 701 1.125 7950
0.5501 461 -2.25 493 1.375 820 1.375 817
0.6501 524 -1.75 579 1.625 7897 1.625 746
0.7501 619 -1.25 722 1.875 1006 1.875 641
0.8501 732 -0.75 1444 2.125 703 2.125 573
0.9501 6690 -0.25 6150 2.375 634 2.375 513
1.05 2316 0.25 1299 2.625 470 2.625 418
1.15 766 0.75 837 2.875 434 2.875 327
1.25 689 1.25 679 3.125 360 3.125 297
1.35 548 1.75 565 3.375 328 3.375 279
1.45 488 2.25 484 3.625 277 3.625 247
1.55 418 2.75 410 3.875 265 3.875 204
1.65 325 3.25 370 4.125 260 4.125 225
1.75 276 3.75 344 4.375 216 4.375 221
1.85 240 4.25 264 4.625 224 4.625 217
1.95 231 4.75 311 4.875 225 4.875 249

sh: pause: command not found
Number of points: 16743
Number of clusters: 0
Number of dimensions: 4

REPOSITORY SCANNER v.1.2

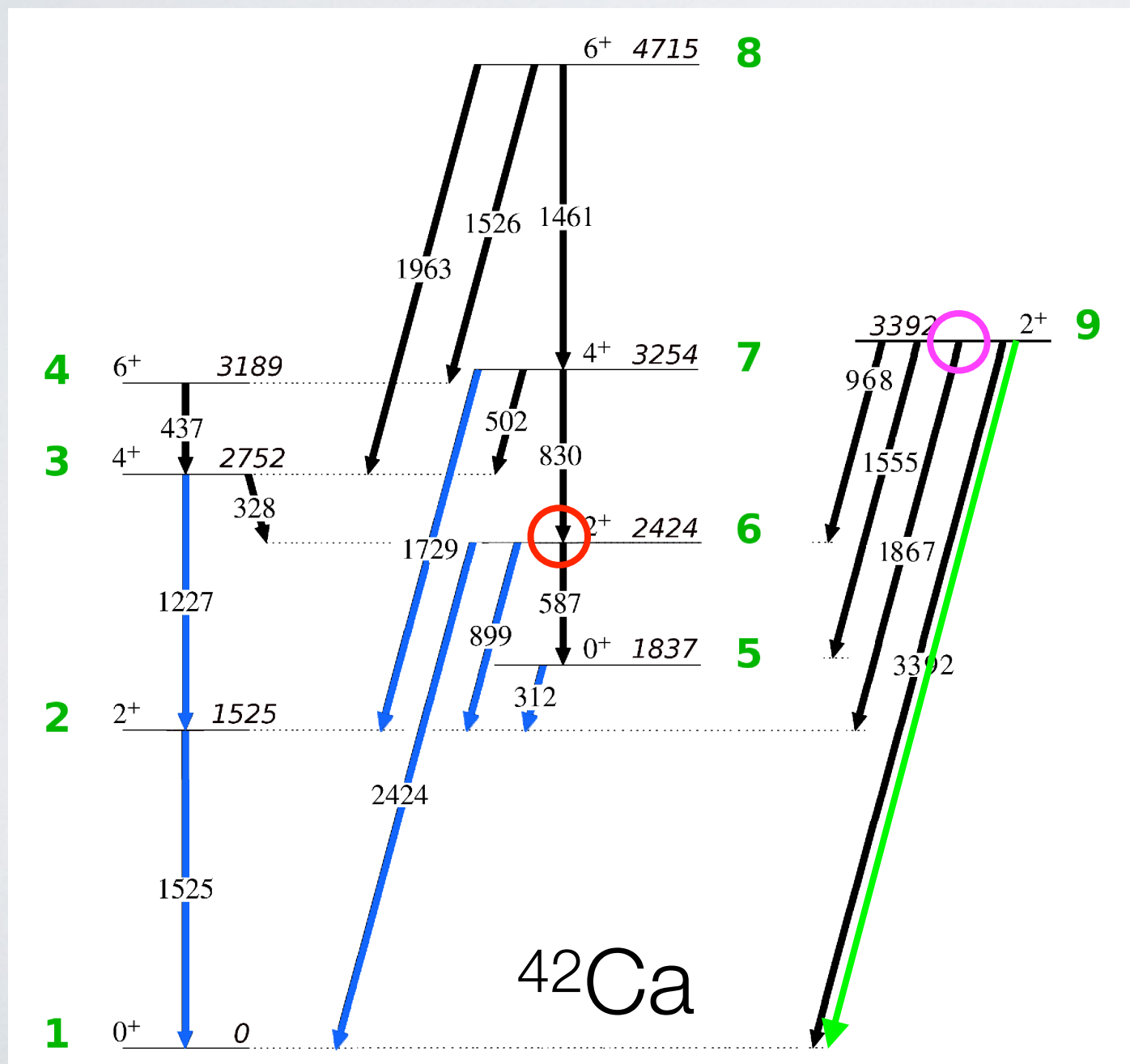
Menu:
1 - Thresholding and MDR,
2 - Cut to phenotype,
3 - Start distances calculation.

4 - Find clusters,
5 - Merge clusters,
6 - Clusters' info,
7 - Work with cluster,
8 - Sort repository by clusters,

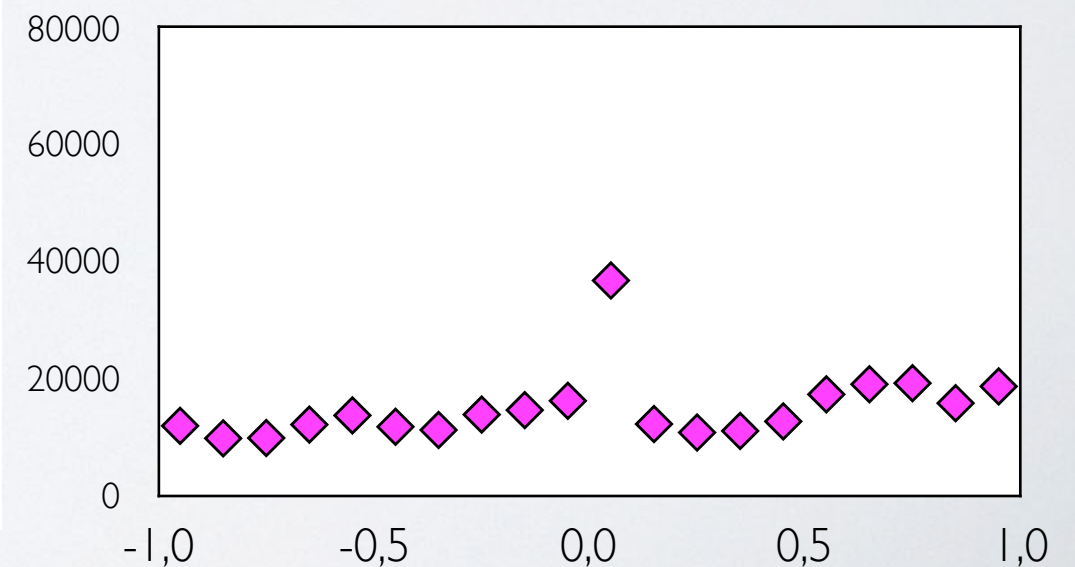
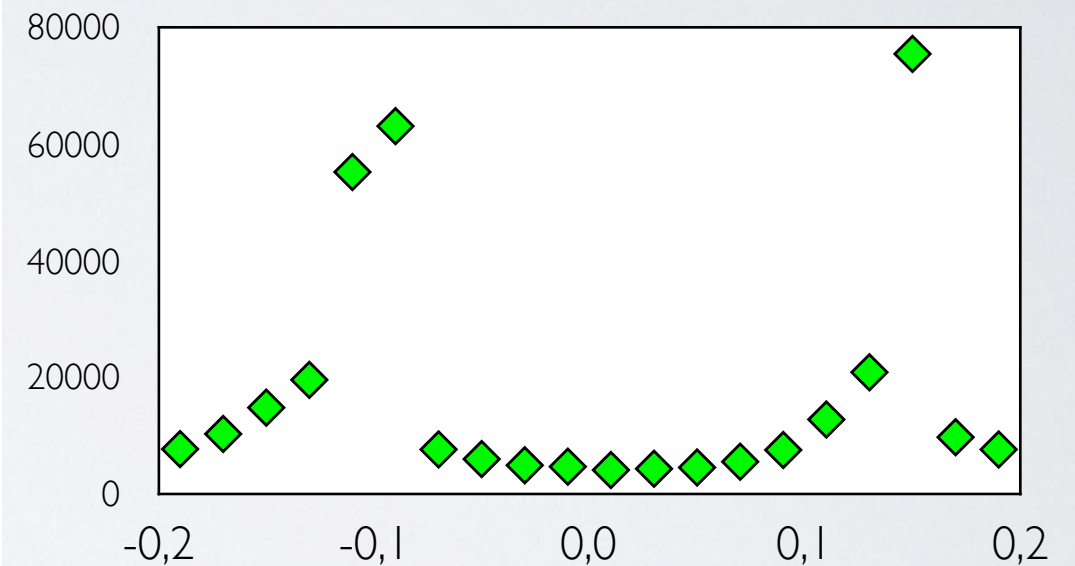
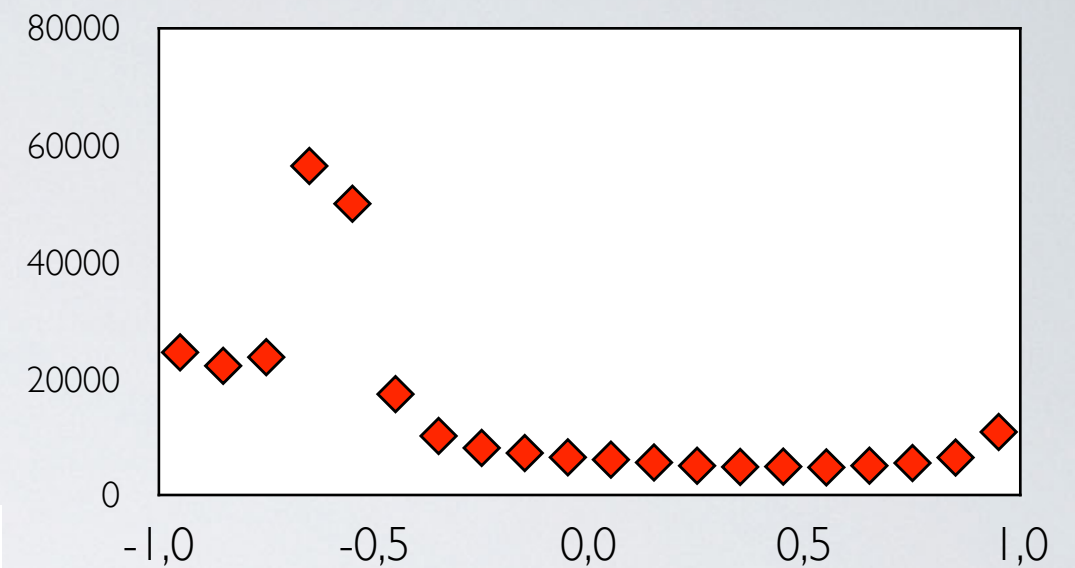
Print:
9 - the BEST point,
10 - ALL points,
11 - ALL points to GNU plot,
12 - PROJECTION,
13 - the BESTS in CLUSTERS,
14 - one CLUSTER,
15 - ranges of parameters,
16 - sampling histogram,

0 - Exit.
```

ALTERNATIVE SOLUTIONS CHECK - IN PRACTICE



K.Hadyńska-Klęk, PhD thesis



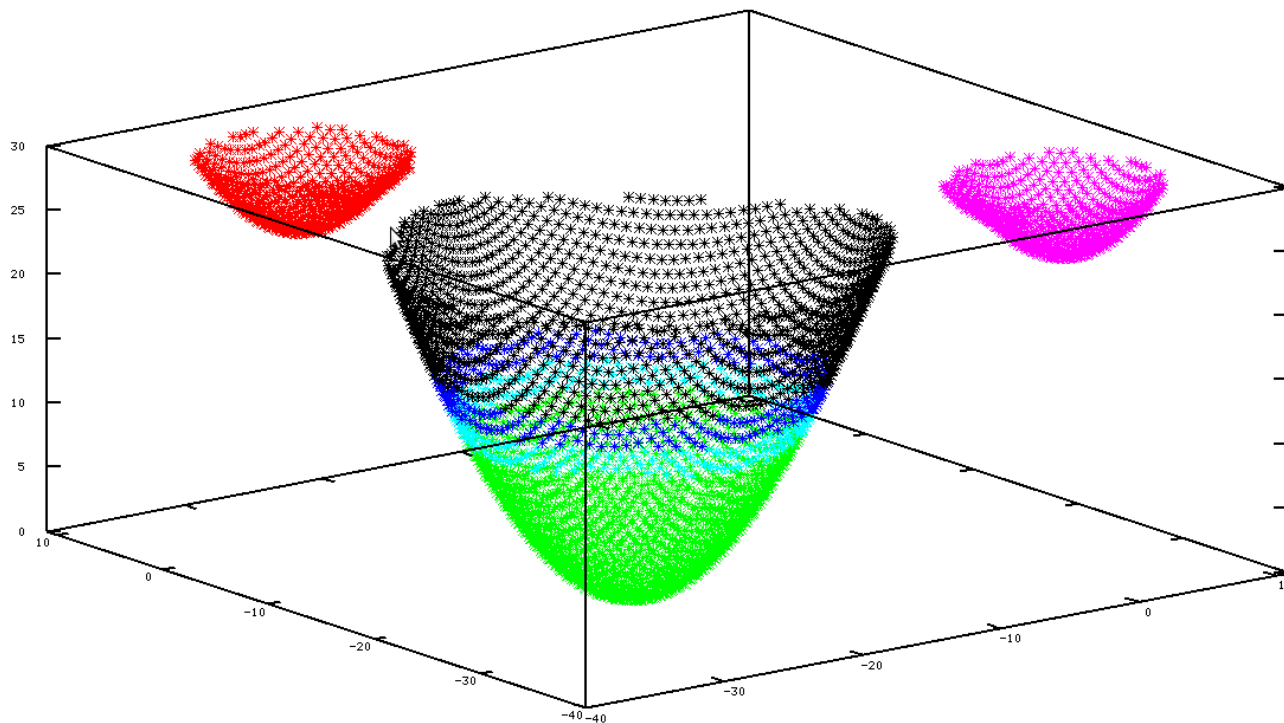
STATUS OF JACOB

- It works...
- The hybrid approach is recommended:
 1. GOSIA: test of input
 2. JACOB: find a interesting point/alternative points
 3. GOSIA: finalise the solution starting from JACOB's point(s)

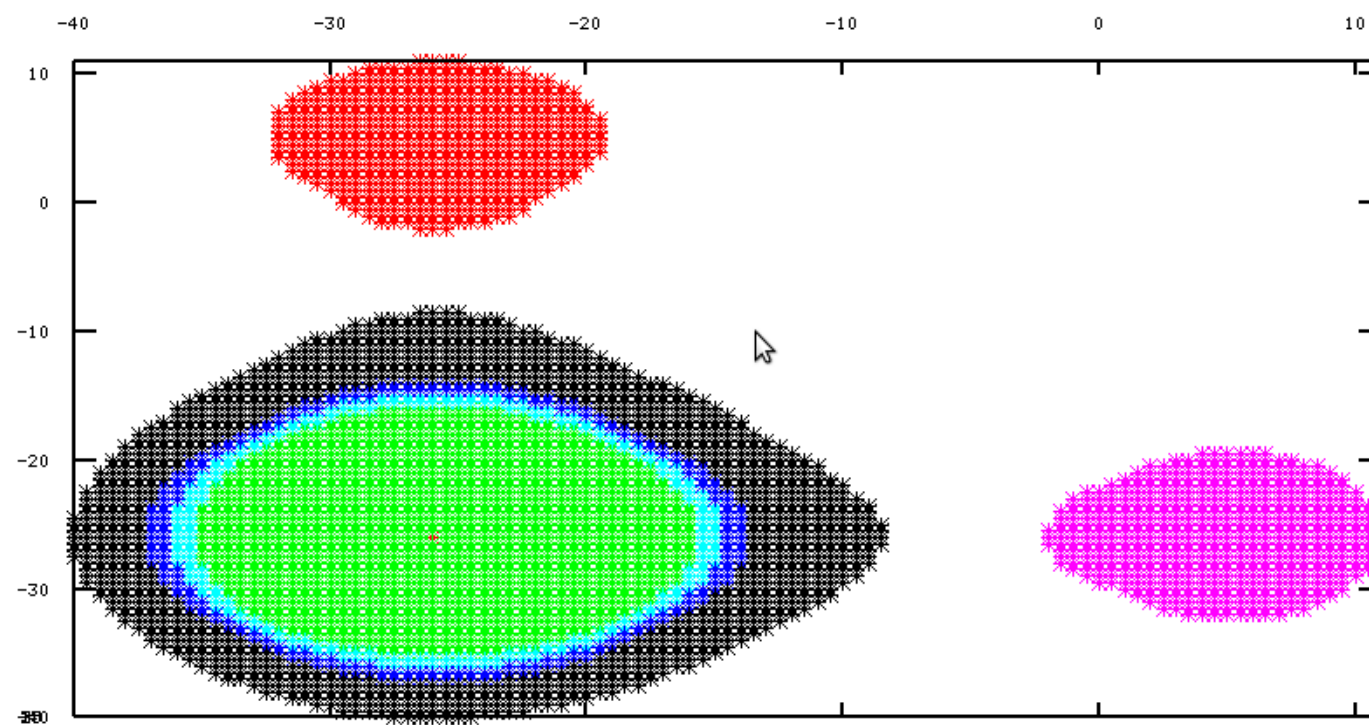
STATUS OF JACOB

- It works...
- The hybrid approach is recommended:
 1. GOSIA: test of input
 2. JACOB: find a interesting point/alternative points
 3. GOSIA: finalise the solution starting from JACOB's point(s)
- RepScan has still to be tested
- JACOB2: extension to projectile-target excitation is planned
- Parallel processing - make to be useful

FRONT LINE ALGORITHM FOR ERROR ESTIMATION



22,500 scale: 1,00000, 1,00000



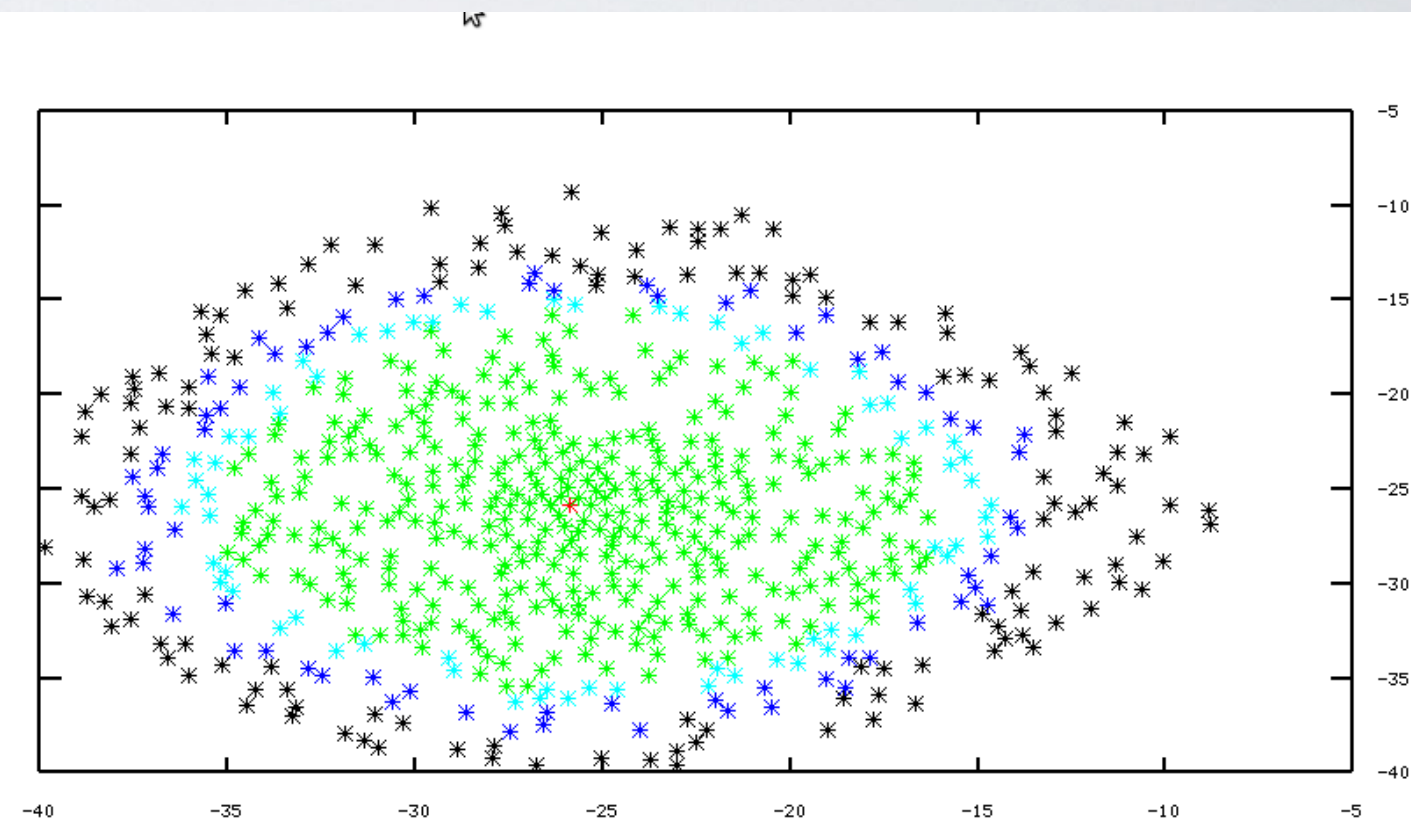
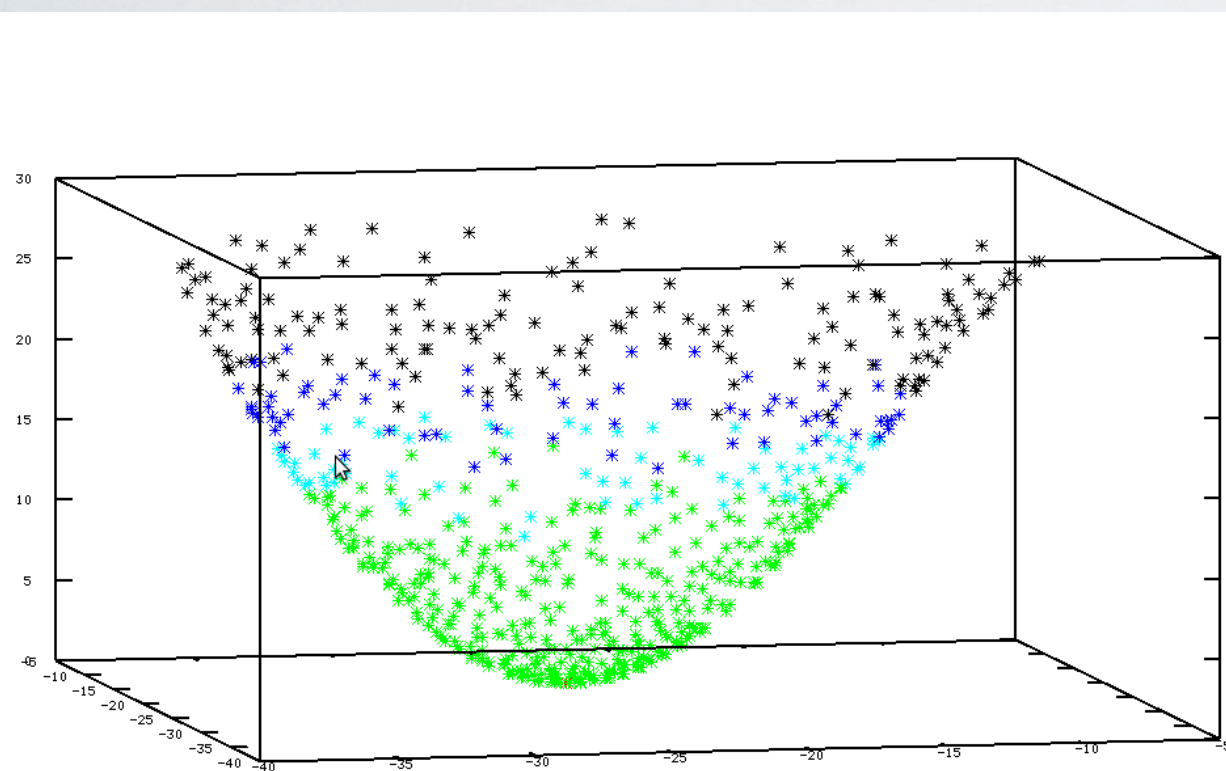
Test function - regular mesh

D.A.Piętak, J. Wojciechowski, P. J.Napiorkowski

A Front Line Algorithm For Error Estimation In Data Sets With Nonuniform Sampling Distribution

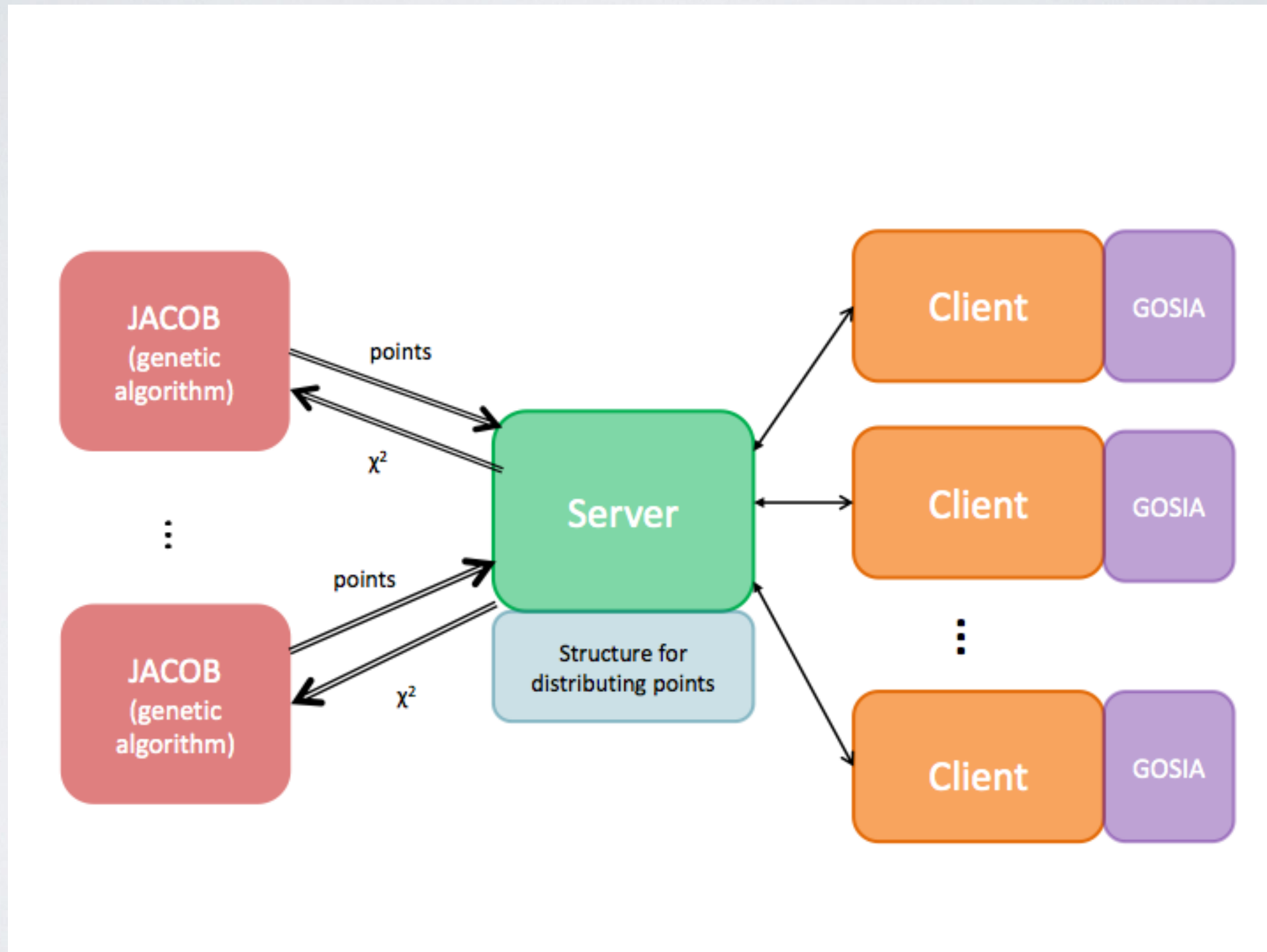
20th European Conference on Circuit Theory and Design (ECCTD), 2011

FRONT LINE ALGORITHM FOR ERROR ESTIMATION



Test function plot after a scan with JACOBI

PARALLEL PROCESSING



SUMMARY

- **JACOB* & QRepScan***

- available for Linux with QT library

- still under development but ready for testers

- www.slcj.uw.edu.pl/~pjn/JACOB

- to start with: e-mail pjn@slcj.uw.edu.pl

Index of /~pjn/Jacob

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
🔙 Parent Directory		-	
🔍 QRepScan4	30-Apr-2014 07:31	150K	
📁 Q_JACOB/	30-Apr-2014 07:35	-	
📁 Q_JACOB64/	30-Apr-2014 07:36	-	

Apache/2.2.14 (Ubuntu) Server at www.slcj.uw.edu.pl Port 80

**Demonstration:
hands-on session for advanced**

SUMMARY

- **JACOB* & QRepScan***

- available for Linux with QT library

- still under development but ready for testers

- www.slcj.uw.edu.pl/~pjn/JACOB

- to start with: e-mail pjn@slcj.uw.edu.pl

TRIBUTE TO:

- Daniel A. Piętak
 - Jacek Wojciechowski
- Warsaw University
of Technology

