

Evolutionary Computation Methods in Particle Physics

Liliana Teodorescu

Brunel
UNIVERSITY
WEST LONDON

Outline

- ❖ ***Introduction to evolutionary computation***
- ❖ ***Evolutionary algorithms***
 - ✓ *solution representation*
 - ✓ *fitness function*
 - ✓ *initial population generation*
 - ✓ *genetic and selection operators*
- ❖ ***Types of evolutionary algorithms***
 - ✓ *string and tree representations*
 - ✓ *hybrid representations*
- ❖ ***Applications in Particle Physics***
- ❖ ***Conclusions***

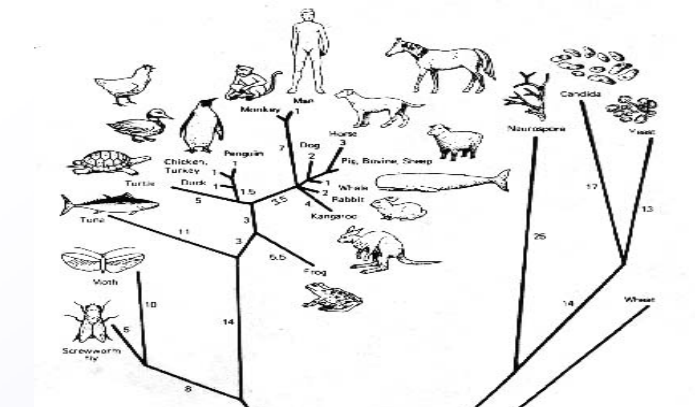
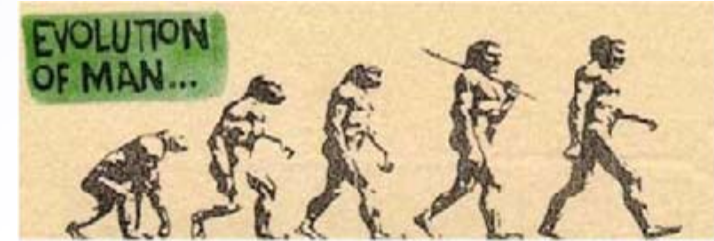
Natural evolution

3

Natural selection - organisms with favourable traits are more likely to survive and reproduce than those with unfavourable traits (Darwin & Wallace)

Population genetics - genetic drift, mutation, gene flow => explain adaptation, speciation (Mendel)

Molecular evolution - identifies DNA as the genetic material (Avery); explains encoding of genes in DNA (Watson & Crick)



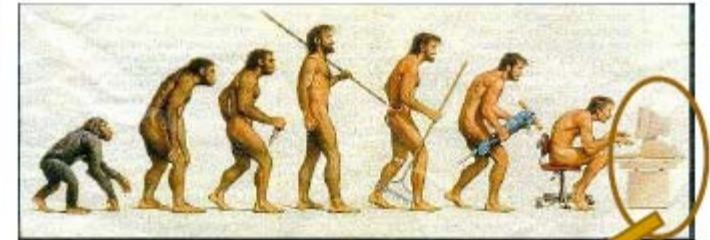
- ❖ Goal of natural evolution - to generate a population of individuals of increasing fitness (ability to survive and reproduce)

Artificial evolution

Artificial evolution - simulation of the *natural evolution* on a computer

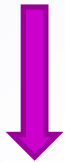


New field - **Evolutionary Computation**
(subfield of Artificial Intelligence)



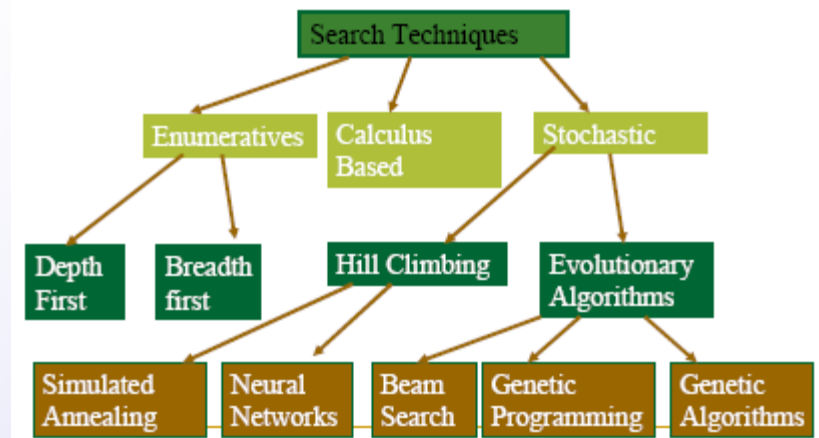
Artificial Evolution

❖ Goal of evolutionary computation - to generate a set of solutions to a problem of increasing quality



Alternative search techniques

e.g. **Evolutionary Algorithms**



Terminology

❖ Individual – candidate solution to a problem

decoding ↑ ↓ encoding

❖ Chromosome – representation of the candidate solution

❖ Gene – constituent entity of the chromosome

❖ Population – set of individuals/chromosomes

❖ Fitness function – representation of how good a candidate solution is

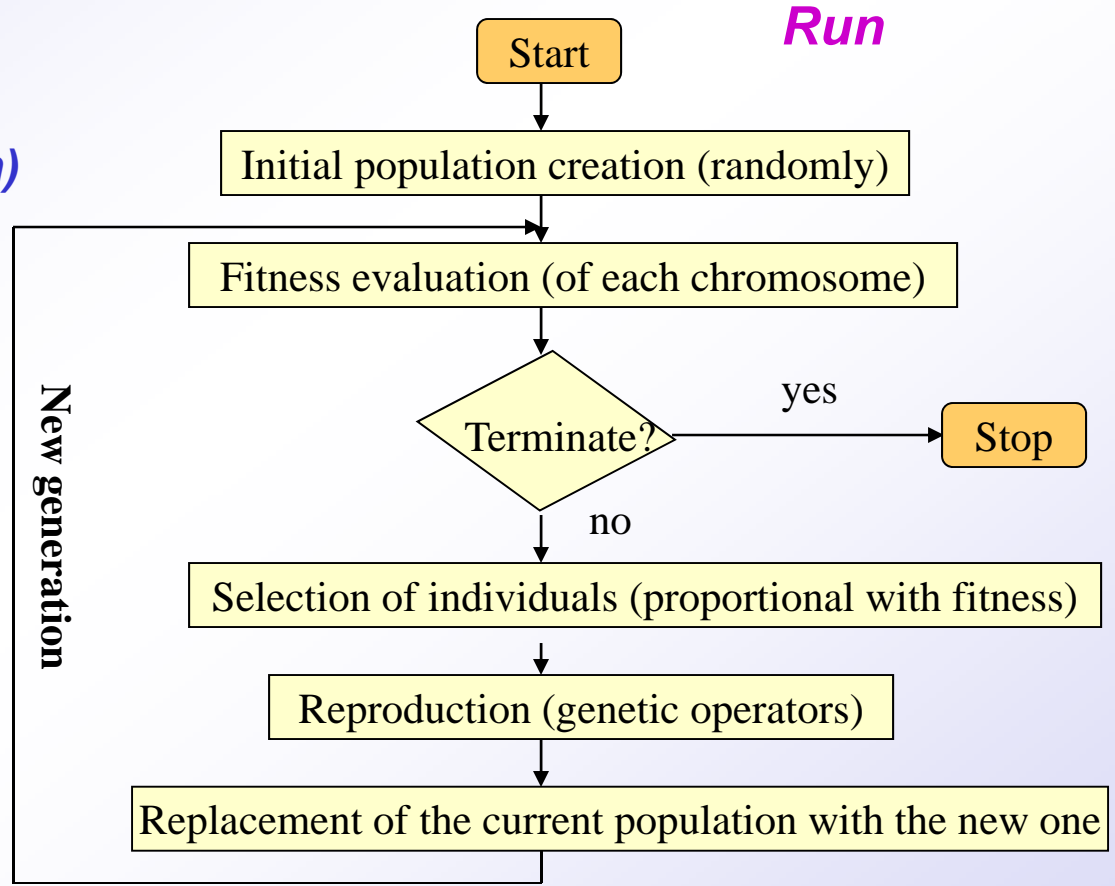
❖ Genetic operators – operators applied on chromosomes in order to create **genetic variation** (other chromosomes)

Evolutionary Algorithms

- ❖ *Problem definition*
- ❖ *Solution representation*
(encoding the candidate solution)
- ❖ *Fitness definition*
- ❖ *Run*
- ❖ *Decoding the best fitted chromosome = **solution***

Genetic operators

- ✓ *cross-over – combining genetic material from parents*
- ✓ *mutation - randomly changes the values of genes*
- ✓ *elitism/cloning – copies the best individuals in the next generation*



Solution representation

Chromosome – representation of the candidate solution

Each chromosome represents a **point in the search space**

Appropriate chromosome representation

- ✓ **very important for the success of EA**
- ✓ **influence the efficiency and complexity of the search algorithm**

Representation schemes

- ❖ **Binary strings – each bit is a boolean value, an integer or a discretized real number**
- ❖ **Real-valued variables**
- ❖ **Trees**
- ❖ **Combination of strings and trees**

Fitness function

The most important component of EA !

Fitness function - representation of **how good** (close to the optimal solution) a candidate solution is

- maps a chromosome representation into a scalar value

$$F : C^l \rightarrow \mathbb{R} \quad l - \text{chromosome dimension}$$

Fitness function needs to **model accurately** the optimisation problem

Used:

- ✓ in the selection process
- ✓ to define the probability of the genetic operators

Includes:

- ✓ all criteria to be optimised
- ✓ reflects the constraints of the problem penalising the individuals that violates the constraints

Initial population

Generation of the initial population:

- ❖ random generation of gene values from the allowed set of values (standard method)
Advantage - ensure the initial population is a uniform representation of the search space
- ❖ biased generation towards potentially good solutions if prior knowledge about the search space exists.
Disadvantage – possible premature convergence to a local optimum

Size of the initial population:

- ❖ small population – represents a small part of the search space
 - ✓ time complexity per generation is low
 - ✓ needs more generations
- ❖ large population – covers a large area of the search space
 - ✓ time complexity per generation is higher
 - ✓ needs less generations to converge

Reproduction (genetic) operators

10

Purpose

- ❖ *to produce offspring from selected individuals*
- ❖ *to replace parents with fitter offspring*

Typical operators

- ❖ cross-over – *creates new individuals combining genetic material from parents*
- ❖ mutation - *randomly changes the values of genes (introduces new genetic material)*
 - *has **low probability** in order not to distort the genetic structure of the chromosome and to generate loss of good genetic material*
- ❖ elitism/cloning – *copies the best individuals in the next generation*

*The exact **structure** of the operators – dependent on the type of EA*

Selection operators

11

Purpose - to select individuals for applying reproduction operators

❖ Random selection – individuals are selected randomly, without any reference to fitness

❖ Proportional selection – the probability to select an individual is proportional with the fitness value

$$P(C_n) = \frac{F(C_n)}{\sum_{n=1}^N F(C_n)}$$

$P(C_n)$ – selection probability of the chromosome C_n

$F(C_n)$ – fitness value of the chromosome C_n

✓ Normalised distribution by dividing to the maximum fitness - accentuate small differences in fitness values (**roulette wheel method**)

❖ Rank-based selection – uses the rank order of the fitness value to determine the selection probability (not the fitness value itself)
e.g. non-deterministic linear sampling – individual sorted in decreasing order of the fitness value are randomly selected

❖ Elitism – k best individuals are selected for the next generation, without any modification
 k – called generation gap

EA vs classical optimisation

	EA	CO
Transition from one point to another in the search space	✓ Probabilistic rules ✓ Parallel search	✓ Deterministic rules ✓ Sequential search
Starting the search process	Set of points	One point
Search surface information that guides to the optimal solution	No derivative information (only fitness value)	Derivative information (first or second order)

Types of Evolutionary Algorithms

13

Hundreds of versions !

String based

- ❖ **Genetic Algorithms (GA)** (*J. H. Holland, 1975*)
- ❖ **Evolutionary Strategies (ES)** (*I. Rechenberg, H-P. Schwefel, 1975*)

Tree based

- ❖ **Genetic Programming (GP)** (*J. R. Koza, 1992*)

Hybrid representations

- ❖ **Developmental Genetic Programming (DGP)** (*W. Benzhaf, 1994*)
- ❖ **Gene Expression Programming (GEP)** (*C. Ferreira, 2001*)

Main differences

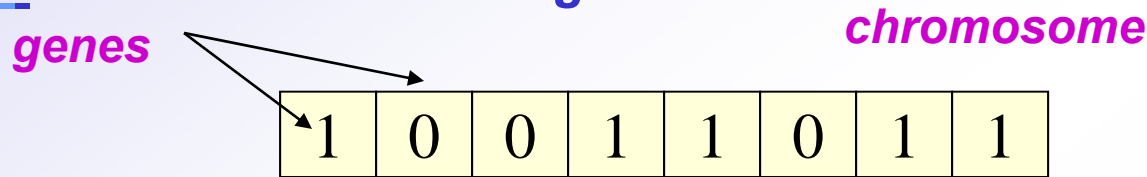
- ✓ **Encoding method (solution representation)**
- ✓ **Reproduction method**

Genetic Algorithms

Solution representation

Chromosome - fixed-length binary string (common technique)

Gene - each bit of the string

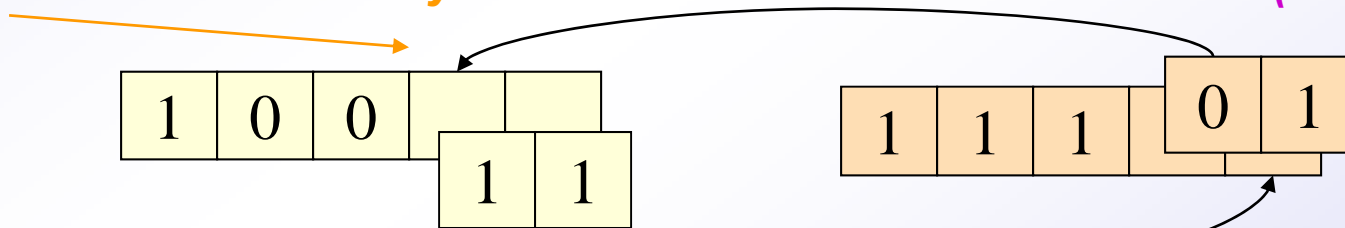


Reproduction

Cross-over (recombination) – exchanges parts of two chromosomes

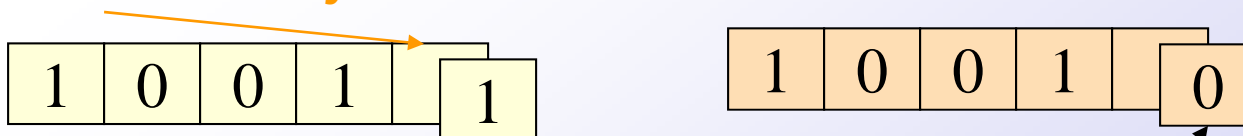
Point chosen randomly

(usual rate 0.7)



Mutation – changes the gene value (usual rate 0.001-0.0001)

Point chosen randomly



GA in PP

Mainly for large-scale optimisation and fitting problems

Experimental PP

- ❖ **event selection optimisation** (A. Drozdetskiy et. al. Talk at ACAT2007)
- ❖ **trigger optimisation** (L1 and L2 CMS SUSY trigger – NIM A502 (2003) 693)
- ❖ **neural-network optimisation for Higgs search**
(F. Haki et.al., talk at STAT2002)

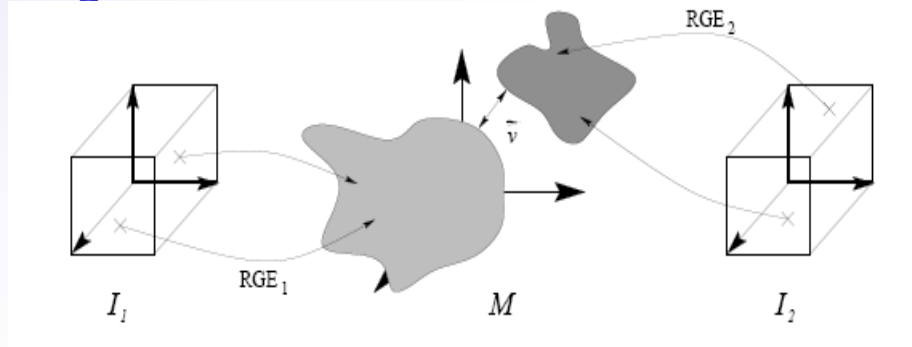
Theoretical/phenomenological PP

- ❖ **fitting isobar models to data for $p(\gamma, K^+) \Lambda$** (NP A 740 (2004)147)
- ❖ **discrimination of SUSY models** (JHEP 0407:069,2004; hep-ph/0406277)
- ❖ **lattice calculations** (NP B 73 (1999) 847; 83-84 (2000)837)

GA in PP - example

Discrimination of SUSY models (B.C. Allanach et.al, JHEP 0407:069,2004)

GA used to estimate a rough accuracy required for sparticle mass measurements and predictions to distinguish SUSY models



I_k – input space of free parameters of model k

M – space of physical measurements (sparticle masses)

Each point in I_k is (potentially) mapped into M with a set of renormalisation group equations (RGE) \Rightarrow model footprint

$$\text{Distance measure } \Delta = \frac{|\vec{M}_A - \vec{M}_B|}{|\vec{M}_A + \vec{M}_B|} \quad A, B - \text{points in two footprints}$$

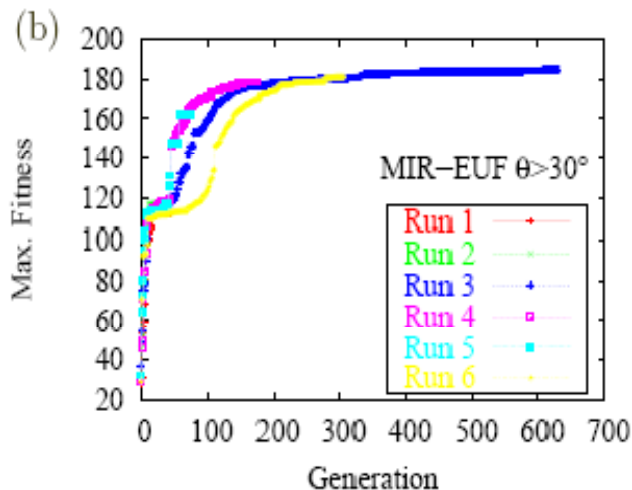
Minimum Δ (over points in input space) – estimate of accuracy of mass measurements needed to distinguish the models

GA in PP - example (cont.)

GA used to minimise Δ

Chromosome – real numbers: values of the free parameters of the two models to be compared

$$Fitness \equiv \frac{1}{\Delta} = \frac{|\vec{M}_A + \vec{M}_B|}{|\vec{M}_A - \vec{M}_B|} = \sqrt{\frac{(M_{\tilde{\chi}_1^0, A} + M_{\tilde{\chi}_1^0, B})^2 + \dots + (M_{\tilde{\tau}_2, A} + M_{\tilde{\tau}_2, B})^2}{(M_{\tilde{\chi}_1^0, A} - M_{\tilde{\chi}_1^0, B})^2 + \dots + (M_{\tilde{\tau}_2, A} - M_{\tilde{\tau}_2, B})^2}}$$



	MIR	EUR
θ	42.2	77.2
ϕ	33.2	36.3
$\tan \beta$	3.4	4.2
$m_{3/2}$	1194	991
Δ	0.0054	
Fitness	184.6	

MIR – mirage scenario
EUR – early unification

$$\Delta = 0.5\%$$

Genetic Programming

18

GP search for the **computer program** to solve the problem, not for the solution to the problem.

Computer program - any computing language (in principle)
- LISP (List Processor) (in practice)

LISP - highly symbol-oriented

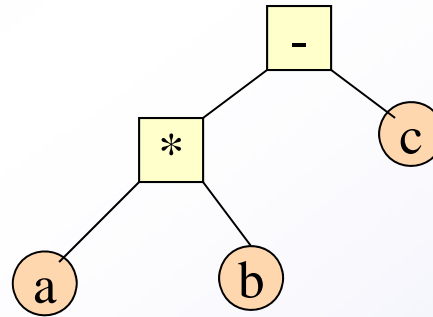
Mathematical expression

$a*b-c$

S-expression

$(-(*ab)c)$

Graphical representation of S-expression



functions (+,*)
and
terminals (a,b,c)
(variables or constants)

Solution representation

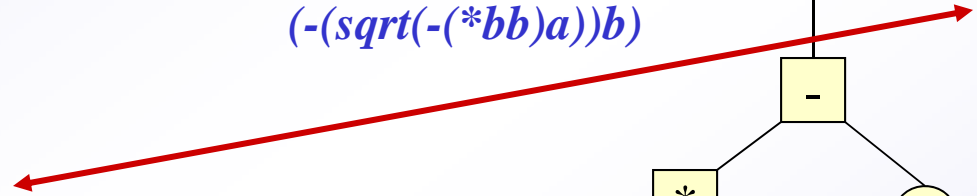
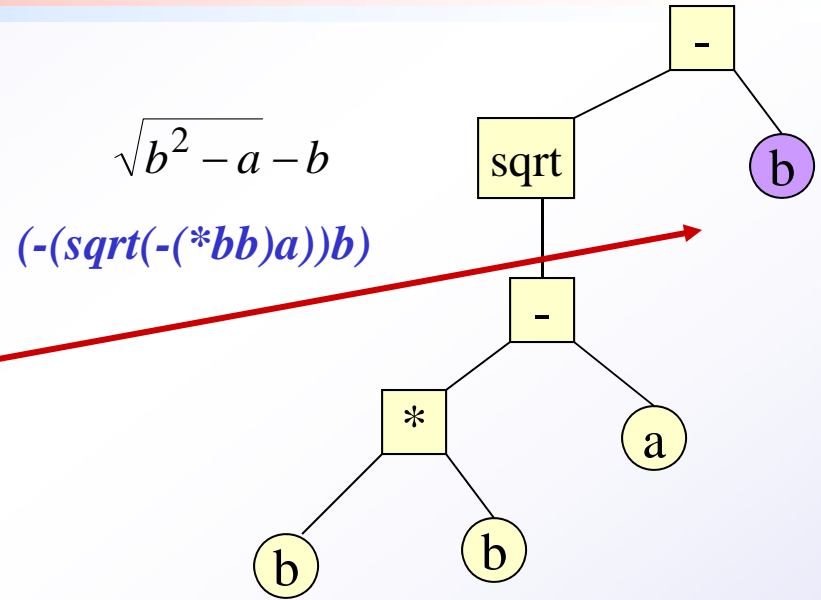
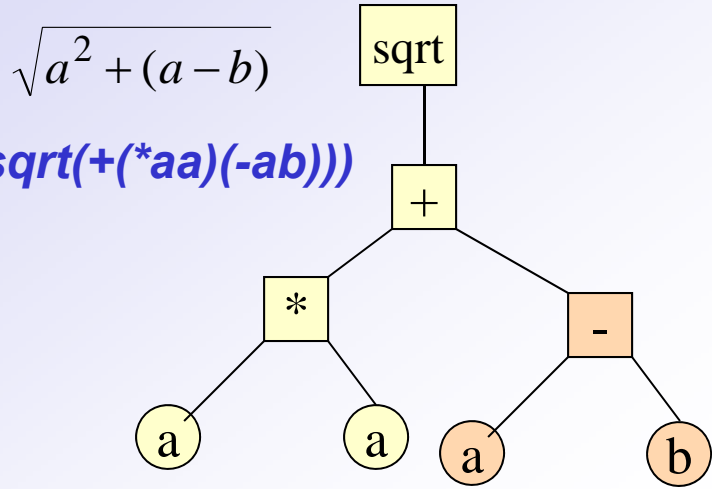
Chromosome: S-expression - variable length => more flexibility
- syntax constraints => invalid expressions

Reproduction

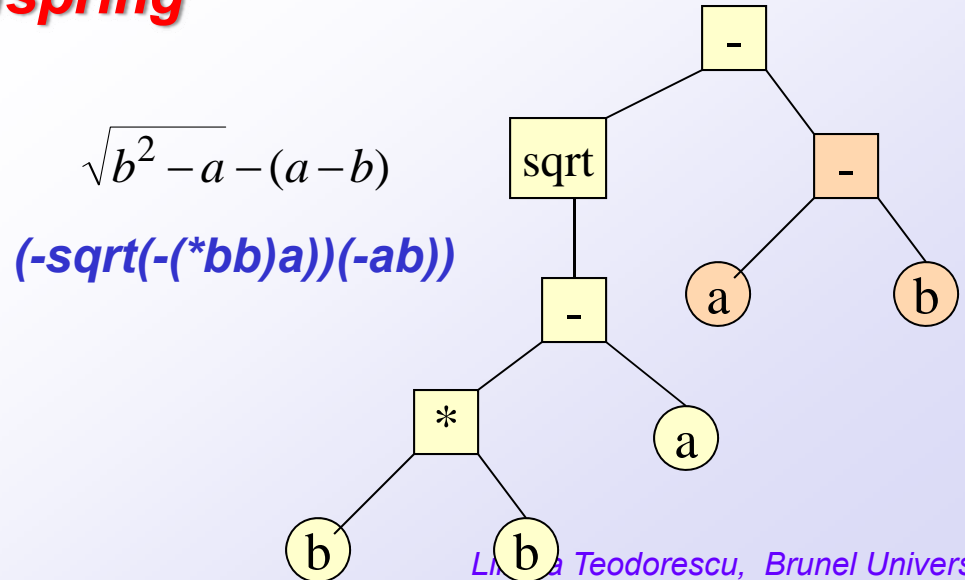
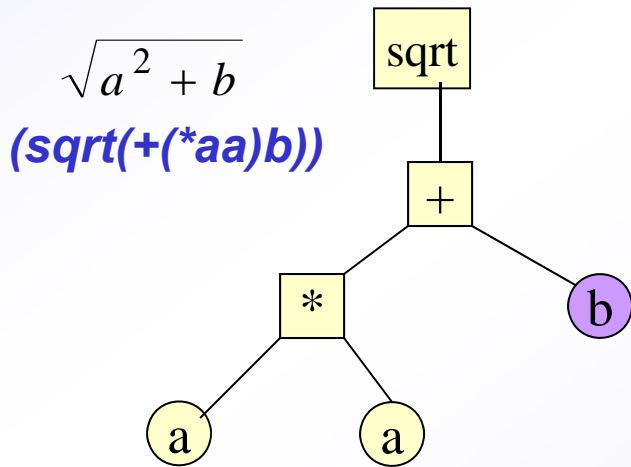
Cross-over (recombination) and Mutation (usually)

Cross-over operator

Parents



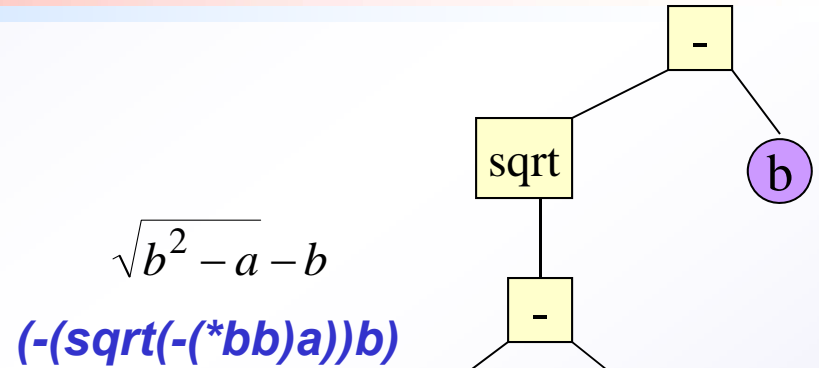
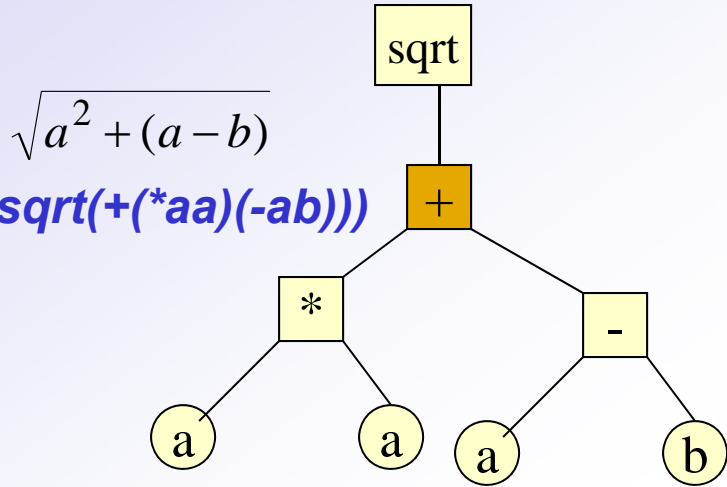
Offspring



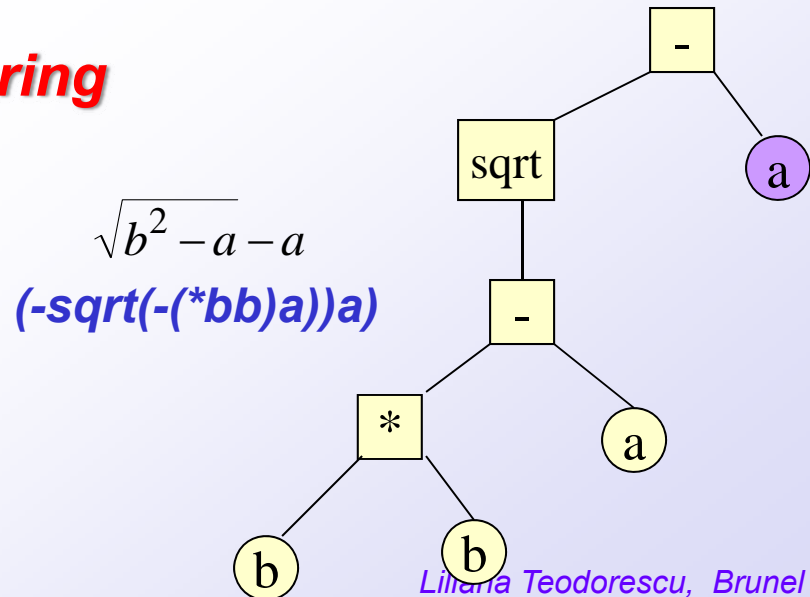
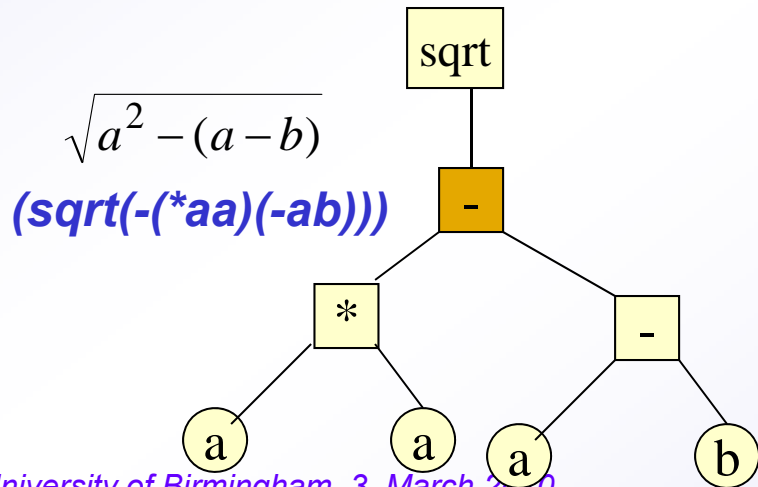
Mutation operator

- ❖ *function replaced by another function*
- ❖ *terminal replaced by another terminal*

Parents



Offspring



GP in PP

Experimental PP - event selection

- ❖ Higgs search in ATLAS *K. Cranmer et.al., Comp. Phys. Com 167, 165 (2005).*
- ❖ D , D_s and Λ_c decays in FOCUS (*J.M. Link et. al., NIM A 551, 504 (2005); PL B624, 166 (2005)*)

e.g. Search for $D^+ \rightarrow K^+ \pi^+ \pi^-$ (FOCUS)

Chromosome: candidate cuts/selection rules - tree of:

- ❖ **functions: mathematical functions and operators, boolean operators**
- ❖ **variables: vertexing variables, kinematical variables, PID variables**

Fitness function (will be minimised)

$$\frac{S + B}{S^2} \times 10000(1 + 0.005 \times n)$$

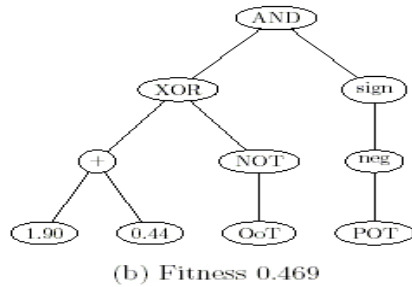
n - number of tree nodes

penalty based on the size of the tree

(big trees must make significant contribution to bkg reduction or signal increase)

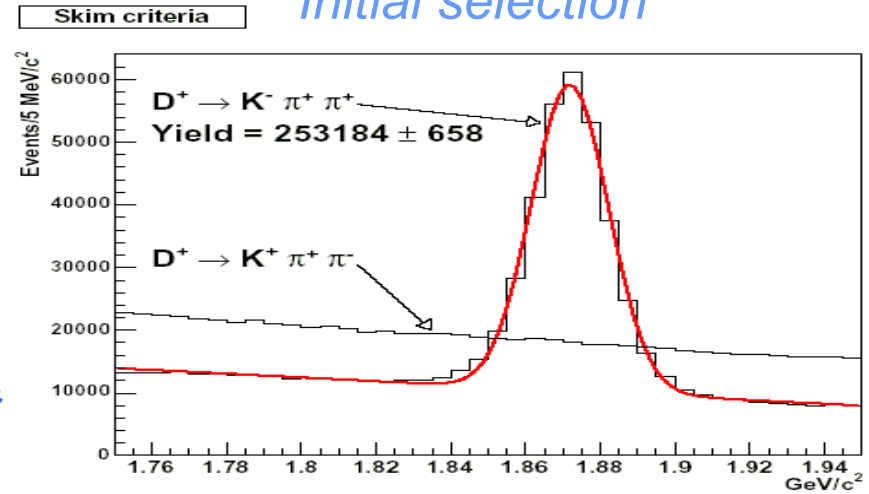
GP in PP

Best fitted chromosomes from generation 0

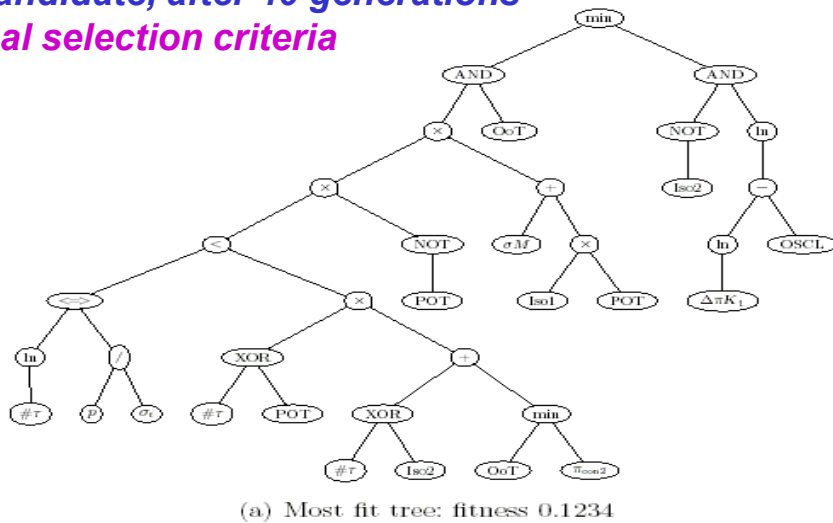


Inter point in target Decay vertex out of target

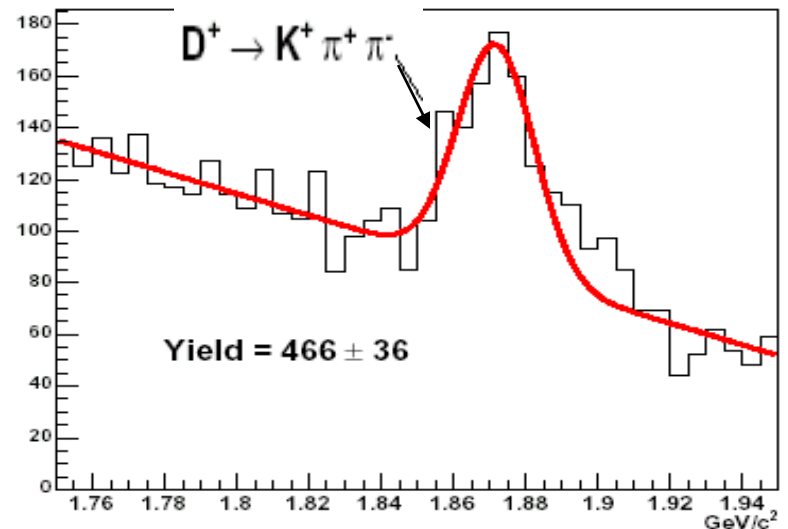
Initial selection



Best candidate, after 40 generations = final selection criteria

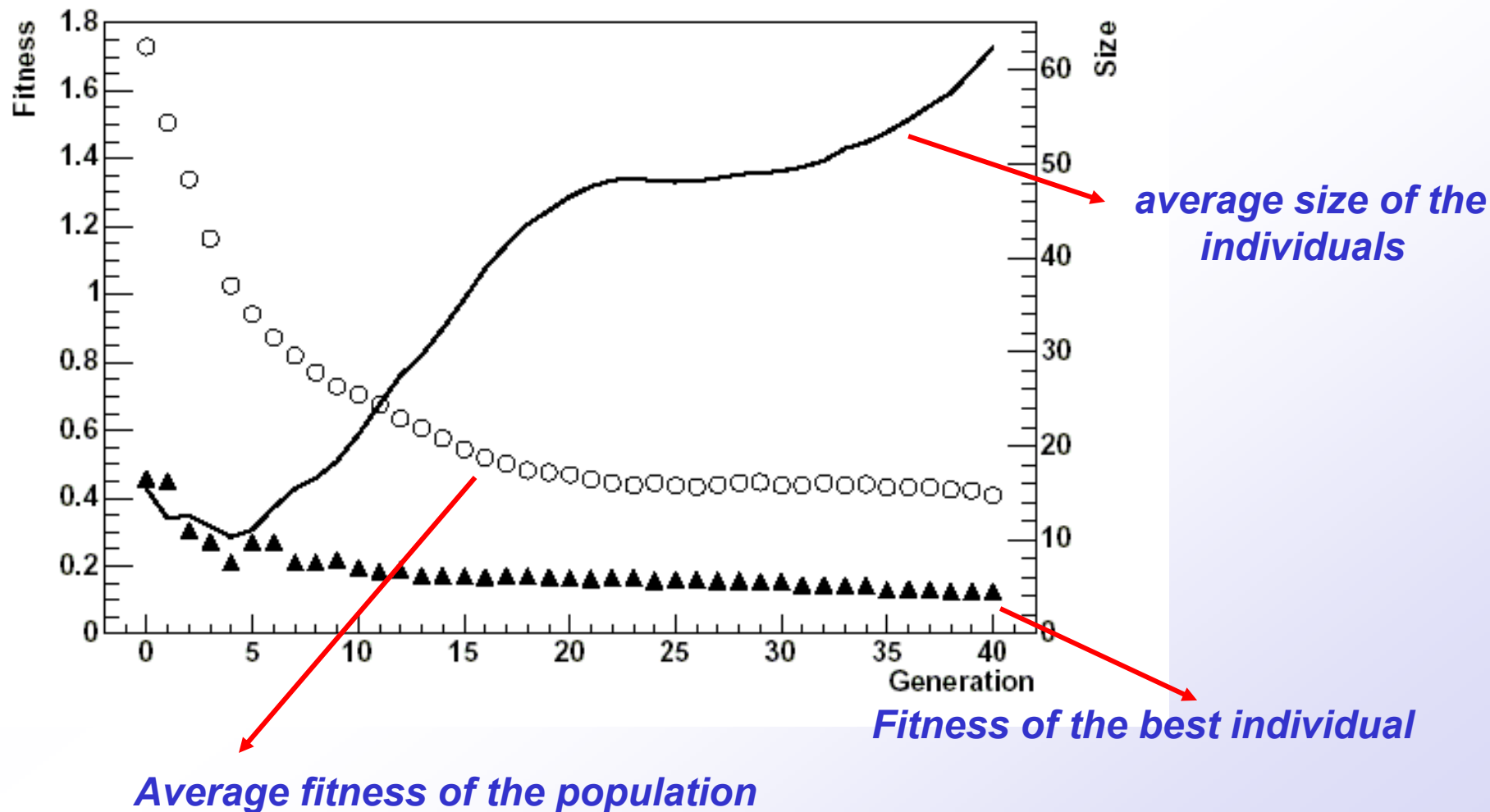


Final selection



GP in HEP (cont.)

Evolution graph



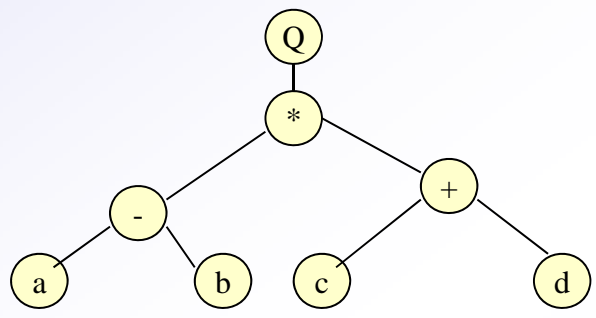
Gene Expression Programming (GEP)

Chromosome - sequence of symbols (functions and terminals)

Head (h) Tail (t) $t=h(n-1)+1$
 Q^*+abcd $aaabbb$ $n - \text{highest arity}$

↓ mapping

Expression tree (ET)



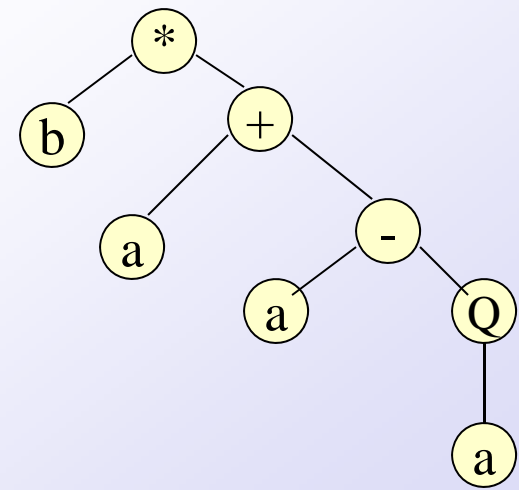
↓ Translation (as in GP)

Mathematical expression

$$\sqrt{(a - b) \cdot (c + d)}$$

ET ends before the end of the gene!

**b+a-aQab+//+b+babbabbbababbaaa*



GEP (cont.)

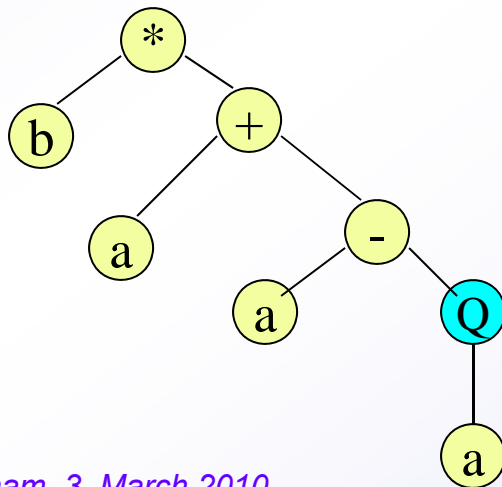
Reproduction

Genetic operators *applied on chromosomes* not on ET =>
always produce syntactically correct structures!

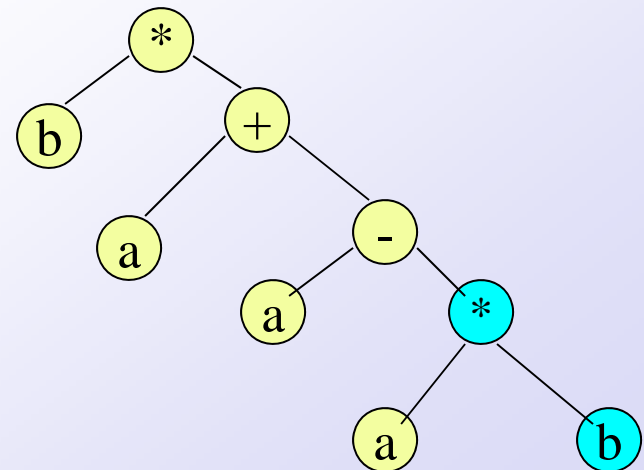
- ❖ *Cross-over* – exchanges parts of two chromosomes
- ❖ *Mutation* – changes the value of a node
- ❖ *Transposition* – moves a part of a chromosome to another location in the same chromosome

e.g. Mutation: Q replaced with *

*b+a-aQab+//+b+babbabbababbaaa



*b+a-a*ab+//+b+babbabbababbaaa



GEP for event selection

L. Teodorescu, *IEEE Trans. Nucl. Phys.*, vol. 53, no.4, p. 2221 (2006)

L. Teodorescu, D. Sherwood, *Comp Phys. Comm.* 178, p 409 (2008)

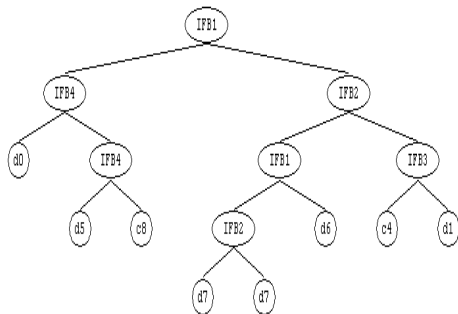
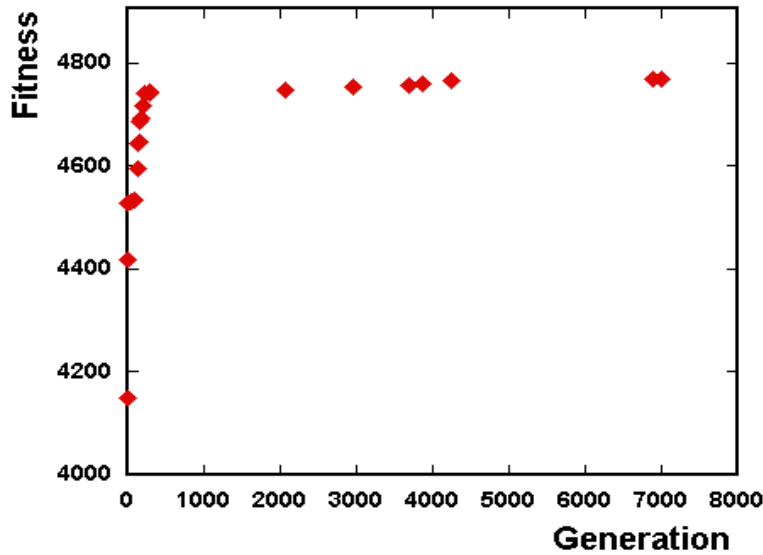
also talks at. CHEP06, ACAT2007 (PoS(ACAT)051 and ACAT2008 (PoS(ACAT)066)

CERN Yellow Report CERN-2008-02

- ❖ **cuts/selection criteria finding** for signal/background classification
- ❖ **fitness function** - number of **events correctly classified** as signal or background (maximise classification accuracy) – limitation imposed by the software available at the time
- ❖ **input functions** - logical functions => cut type rules
- common mathematical functions
- ❖ **input data** - Monte-Carlo simulation from BaBar experiment for Ks production in e^+e^- (~ 10 GeV), $K_S \rightarrow \pi^+ \pi^-$

Model evolution

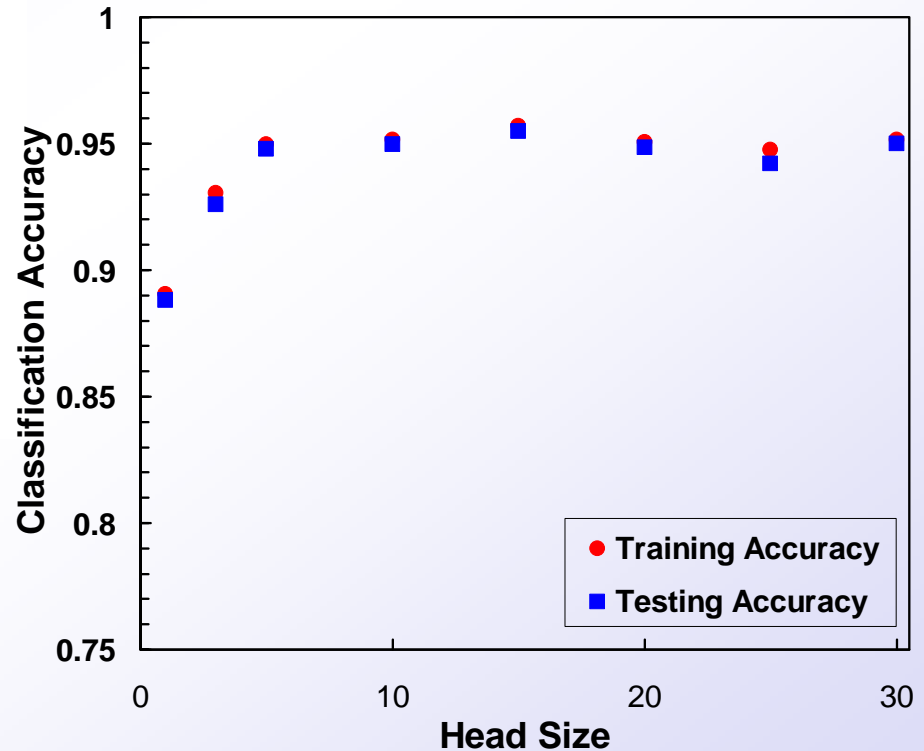
No. of genes = 1, Head length = 10



$F_{sig} \geq 5.26,$
 $R_{xy} < 0.19,$
 $d_{oca} < 1,$
 $P_{chi} > 0$

Classification Accuracy = 95%

Model complexity



Classification rules

GEP analysis – optimises classification accuracy

Head	Selection criteria
1	$F_{sig} \geq 9.93$
2	$F_{sig} \geq 8.80, doca < 1$
3	$F_{sig} > 3.67, R_{xy} \leq P_{chi}$
4	$F_{sig} > 3.67, R_{xy} \leq P_{chi}$
5	$F_{sig} \geq 3.63, R_z \leq 2.65, R_{xy} < P_{chi}$
7	$F_{sig} \geq 3.64, R_{xy} < P_{chi}, P_{chi} > 0$
10	$F_{sig} \geq 5.26, R_{xy} < 0.19, doca < 1, P_{chi} > 0$
20	$F_{sig} > 4.1, R_{xy} \leq 0.2, SFL > 0.2, P_{chi} > 0, doca > 0, R_{xy} \leq Mass$

Cut-based (standard) analysis – optimises signal significance

$F_{sig} \geq 4.0$

$R_{xy} \leq 0.2cm$

$SFL \geq 0cm$

$P_{chi} > 0.001$

Reduction

S: 15%

B: 98%

$doca \leq 0.4cm$

$|R_z| \leq 2.8cm$

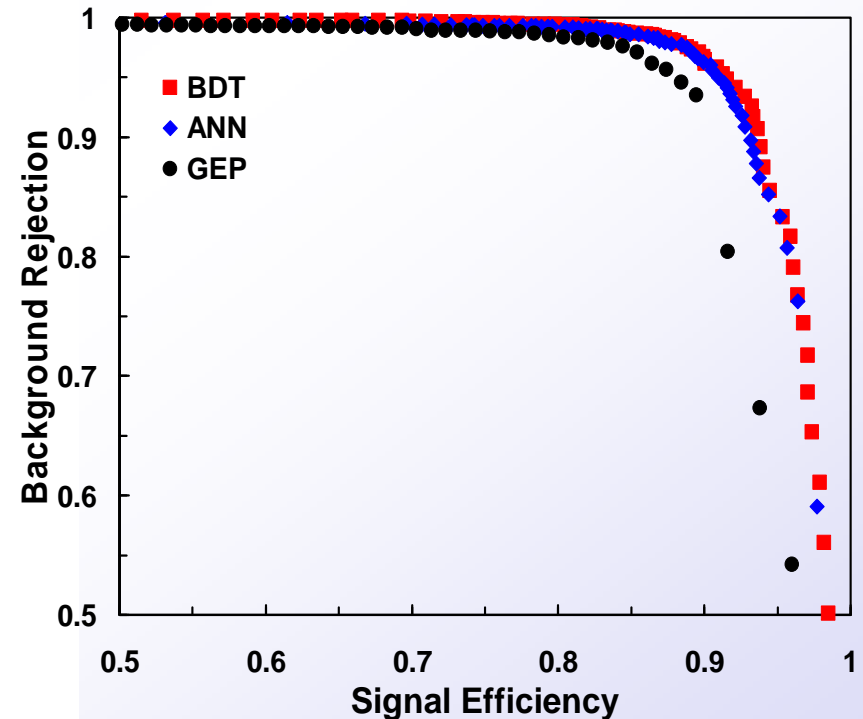
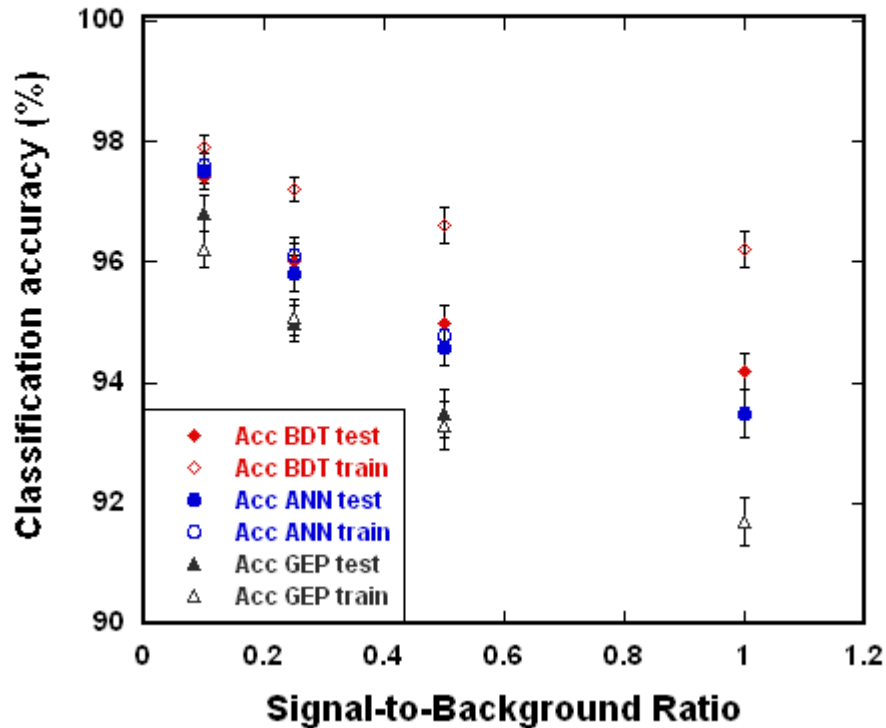
Reduction

S: 16%

B: 98.3%

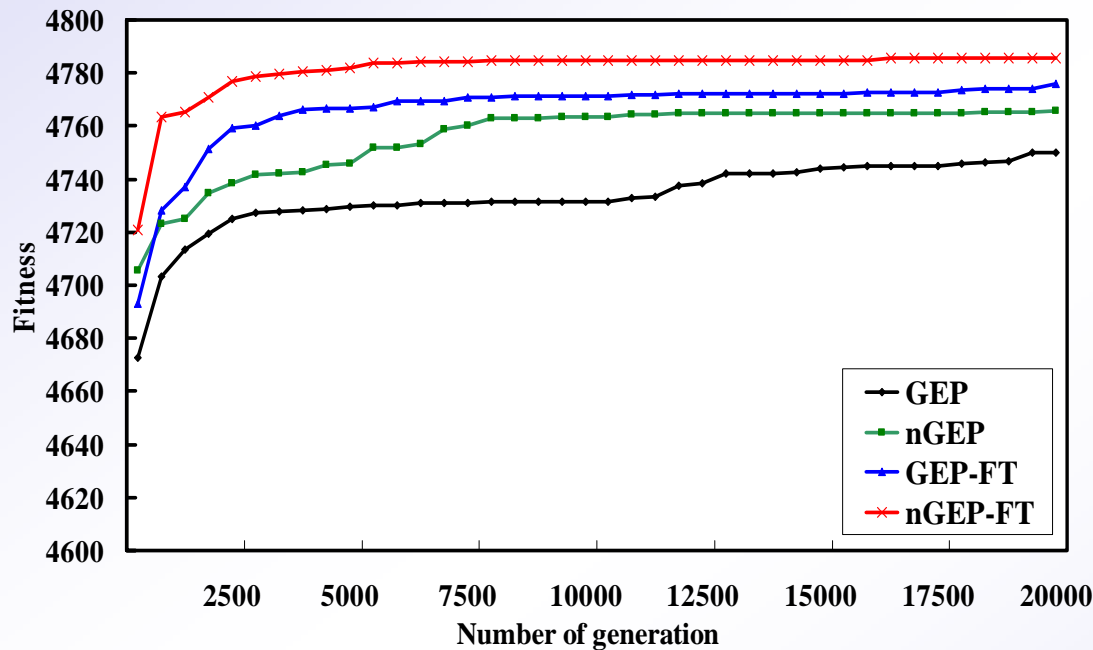
GEP vs ANN and BDT

5000 events, 8 variables, GEP - 38 functions



GEP - improvements

30



nGEP – new methods for creating constants

GEP-FT - evolution controlled by an online threshold on fitness

*FT = average fitness per generation * scaling factor*

Scaling factor optimised (typical values between 0.5 to 1.5)

GEP - Further developments

31

3-year project funded by EPSRC

Detailed studies and further developments of GEP

- characterise and improve the solution evolvability*
- hybrid algorithms (GEP + statistical methods)*
- classification and clustering algorithms*

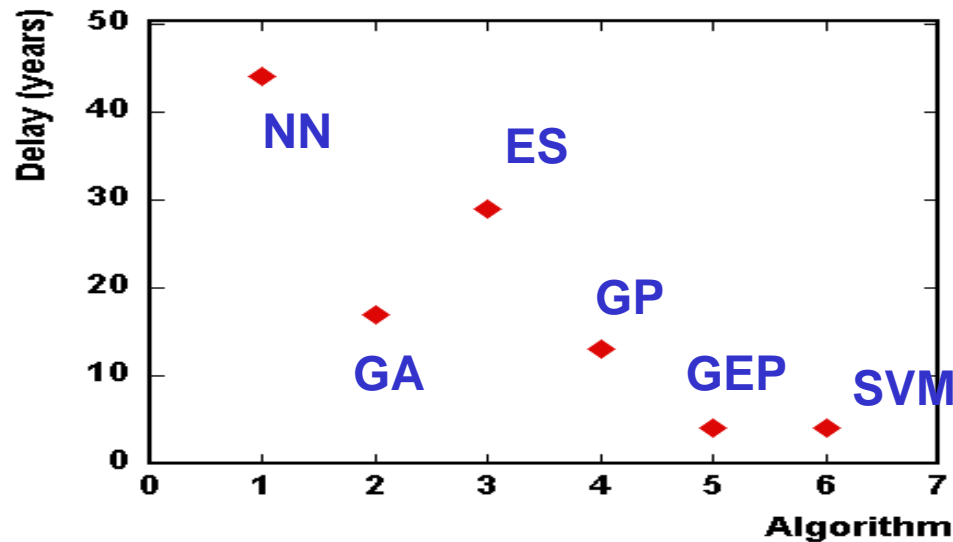
LHC data – test-bed for outcomes of the project => HEP analysis

Small team: myself, one RA, two Ph.D. students

PP usage of AI algorithms

32

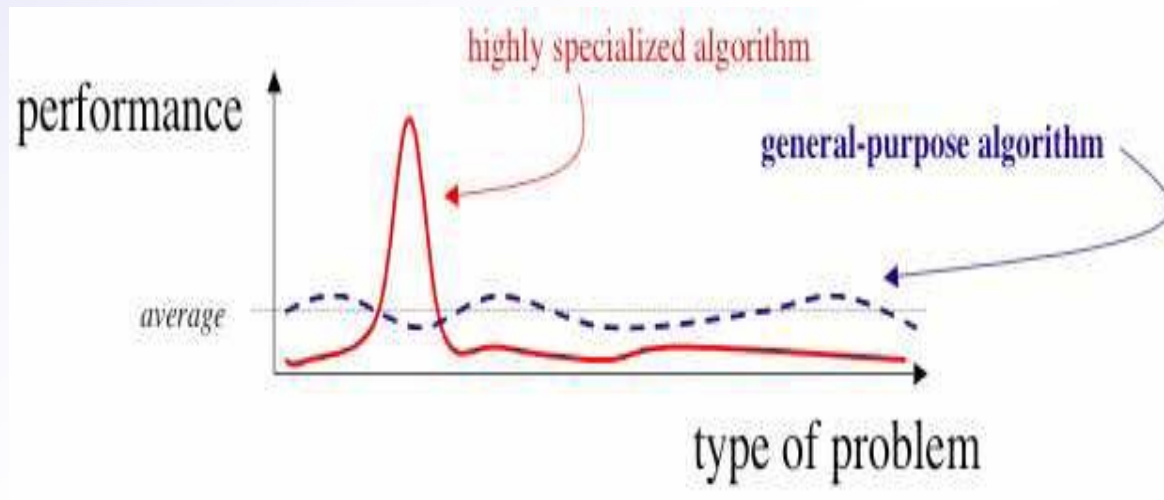
Particle physics – more and more open to new algorithms



Particle physics – in more need of powerful algorithms

General remarks

Wolpert D.H., Macready W.G. (1997), **No Free Lunch Theorem** for Optimization, *IEEE Transactions on Evolutionary Computation* 1, 67.



In PP

- ***used only general purpose algorithms so far***
- ***need more specialised versions?***

Conclusions

Evolutionary algorithms in PP

- ❖ ***used but not extensively (at present)***
- ❖ ***proved to work correctly***
- ❖ ***good performance – optimal solutions, not trapped in local minima***
- ❖ ***need more specialised versions for reaching much better performance***
- ❖ ***disadvantage – high computational time***
 - ***prospects for change – new, faster algorithms, more computing power***



Organized in collaboration with Brunel University

Director

Francois Flückiger, CERN

Local Organising Committee Chair

Liliana Teodorescu

Technical Manager

Andreas Hirstius, CERN
Giuseppe Lo Presti, CERN

School Administrator

Fabienne Baud-Lavigne, CERN
E-mail: computing.school@cern.ch

Data Technologies
Base Technologies
Physics Computing

Lecturers

Bertrand Bellenot, CERN, Geneva, Switzerland
Predrag Buncic, CERN, Geneva, Switzerland
François Flückiger, CERN, Geneva, Switzerland
Rudi Frühwirth, HEPHY, Vienna, Austria
Aatos Heikkinen, HIP, Helsinki, Finland
Robert G. Jacobsen, University of California at Berkeley, USA
Sverre Jarp, CERN, Geneva, Switzerland
Sebastian Lopienski, CERN, Geneva, Switzerland
Pere Mato, CERN, Geneva, Switzerland
Axel Naumann, CERN, Geneva, Switzerland
Andrzej Nowak, CERN, Geneva, Switzerland
Alberto Pace, CERN, Geneva, Switzerland
Bernd Panzer -Steindel, CERN, Geneva, Switzerland
Ivica Puljak, University of Split, Croatia

Deadline for Application
9th May 2010

www.cern.ch/CSC