Evolutionary Computation

in

High Energy Physics

Liliana Teodorescu

Brunel UNIVERSITY WEST LONDON

Outline

- Introduction to evolutionary computation
- Basics of evolutionary algorithms
- Applications of evolutionary algorithms to HEP
- Conclusion

Evolutionary Computation

- *Evolutionary computation* simulates the natural evolution on a computer
- *Goal of natural evolution* – to generate a population of individuals with increasing fitness (increasing ability to survive and reproduce in a specific environment)
- *Goal of evolutionary computation* - to generate a set of solutions (to a problem) of increasing quality

Terminology

- *Individual* – candidate solution to a problem

decoding ↑↓ encoding

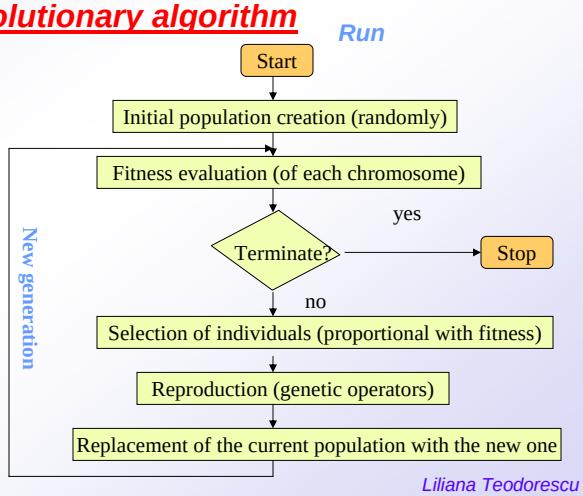
- *Chromosome* – representation of the candidate solution
- *Gene* – constituent entity of the chromosome
- *Population* – set of individuals/chromosomes
- *Fitness function* – representation of how good a candidate solution is
- *Genetic operators* – operators applied on chromosomes in order to create genetic variation (other chromosomes)

Evolutionary Algorithms

Natural evolution simulation - core of the evolutionary algorithms:
optimisation algorithms (iteratively improve the quality of the solutions until an optimal/feasible solution is found)

Basic evolutionary algorithm

- Problem definition
- Encoding of the candidate solution
- Fitness definition
- Run
- Decoding the best fitted chromosome = solution



Classes of Evolutionary Algorithms

- *Genetic Algorithms (GA)* *(J. H. Holland, 1975)*
- *Evolutionary Strategies (ES)* *(I. Rechenberg, H-P. Schwefel, 1985)*
- *Genetic Programming (GP)* *(J. R. Koza, 1992)*
- *Gene Expression Programming (GEP)* *(C. Ferreira, 2001)*

Main differences

- *Encoding method*
- *Reproduction method*

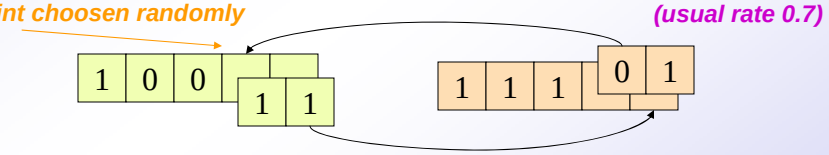
Genetic Algorithms

- ***Encoding***

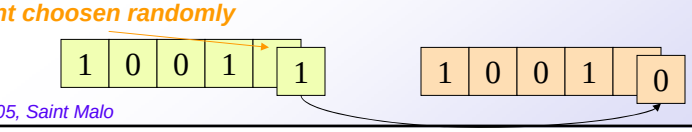
Chromosome - fixed-length binary string (common technique)
Gene - each bit of the string

- ***Reproduction***

Recombination (crossover) – exchanges parts of two chromosomes



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



Evolutionary Strategies

7

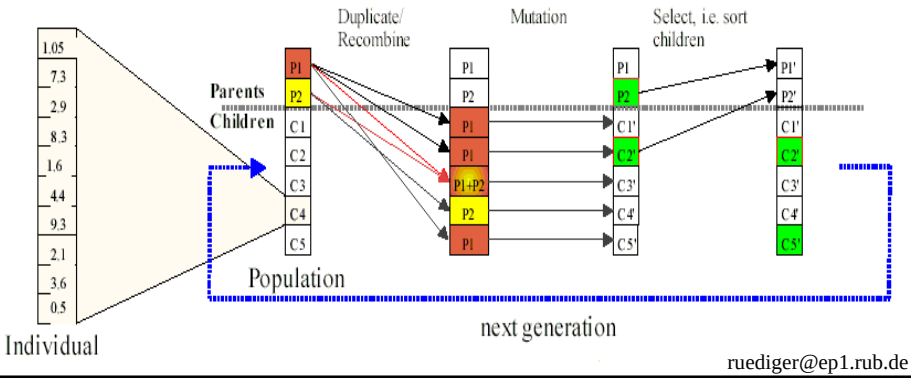
❖ **Encoding**

Chromosome – **set of floating point numbers**

❖ **Reproduction**

Duplication (cloning) **or** *Recombination (addition)* **of parents**

Mutation **of children: adding a gaussian distributed variable to each child**



ruediger@ep1.rub.de

GA and ES in HEP

8

Mainly for large-scale optimisation and fitting problems

Experimental HEP

- **event selection optimisation** *(MIN A534 (2004) 147)*
- **trigger optimisation** *(L1 and L2 CMS SUSY trigger – NIM A502 (2003) 693)*
- **neural-netwok optimisation** *for Higgs search (F. Hakl et.al., talk at STAT2002)*

Theoretical/phenomenological HEP

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

e.g. Cuts optimisation with ES

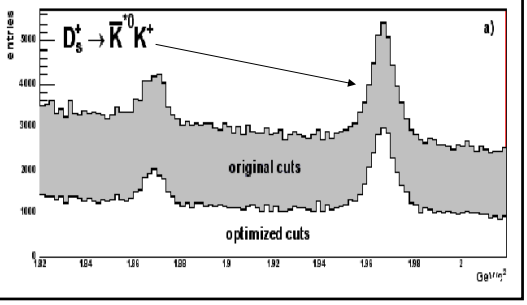
(MIN A534 (2004) 147)

Chromosome: **cut values**

$\cos(\theta_H)$, p_{Ds} , mass constraint, vertex fit probability

Fitness function: **$sig^2=S^2/(S+2B)$**

45.4% improvement in sig^2



Genetic Programming

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

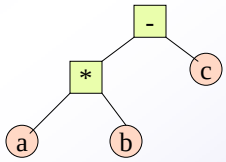
Graphical representation of S-expression

Mathematical expression

a*b-c

S-expression

(-(*ab)c)



functions (+,) and terminals (a,b,c)*

❖ *Encoding*

Chromosome: S-expression - variable length => more flexibility
- sintax constraints => invalid expressions produced in the evolution process must be eliminated => waste of CPU

❖ *Reproduction*

Recombination (crossover) and Mutation (usualy)

GP in HEP

***Experimental HEP* - event selection**

- ❖ *Higgs search in ATLAS (physics/0402030)*
- ❖ *D, D_s and Λ_c decays in FOCUS (hep-ex/0503007, hep-ex/0507103)*

e.g. Search for $D^+ \to K^+\pi^+\pi^-$ *(hep-ex/0503007)*

Chromosome: candidate cuts - tree of:

- ❖ *functions: mathematical functions and operators, boolean operators*
- ❖ *variables: vertexing variables, kinematical variables, PID variables*
- ❖ *constants: reals (-2,2), integers (-10,+10)*

In total: 55

Fitness function (will be minimised)

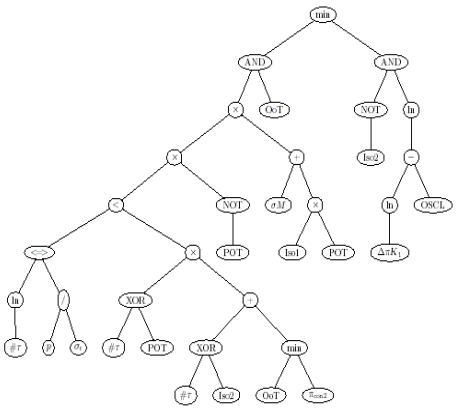
$$\frac{S+B}{S^2}\times 10000(1+0.005\times n)$$

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)

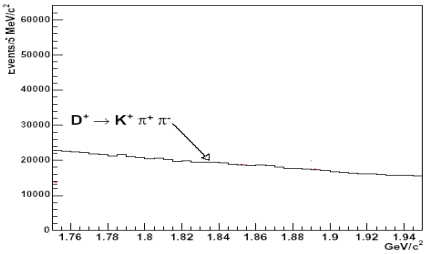
GP in HEP (cont.)

Best candidate, after 40 generations = final selection criteria

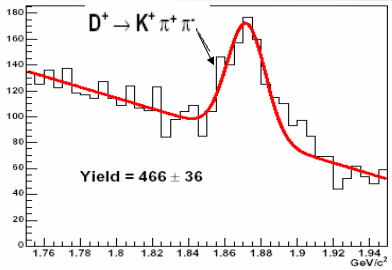


(a) Most fit tree: fitness 0.1234

Skim criteria — *Initial selection*



Final selection



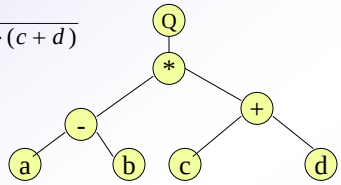
Gene Expression Programming

- ***search for the computer program that solve the problem (as GP)***
- ***works with two entities: chromosomes and expression trees***

Encoding

Candidate solution represented by an expression tree (ET)

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



ET encoded in a chromosome: read ET from left to right and from top to bottom

Q*-+abcd

Q means sqrt

Chromosome – has one or more genes of equal length
Gene – head: contains both functions and terminals (length h)
- tail: contains only terminals (length t)

t=h(n-1)+1 *n – number of arguments of the function with the highest number of arguments*

e.g. ****b+a-aQab+//+b+babbabbbababbaaa***

Gene Expression Programming (cont.)

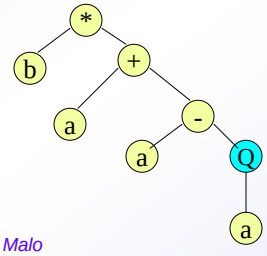
Reproduction

Genetic operators applied on chromosoms not on ET =>
always produce sintactically correct structures!

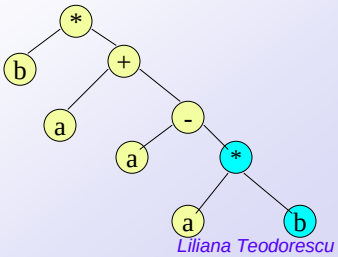
- ***Recombination***
- ***Mutation***
- ***Transposition – a part of the chromosome moved to another part of the same chromosome***

*e.g. Mutation: Q replaced with **

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



***b+a-a*ab+//+b+babbabbbababbaaa**



GEP in HEP

Underway...

My interest ...

See my talk at IEEE on Nuclear Science Conference in October!

Conclusions

Evolutionary computation in HEP:

- *promising and ... Fun!*
- *many areas to be investigated:*
  - *applications to other problems from HEP*
  - *understand advantages and disadvantages*
  - *better comparison with other methods*

Many other methods from computer science and engineering to be exploited