G$^{a}_{S}$SA$_{T}$A, A GENETIC ALGORITHM AS AN ALTERNATIVE TOOL FOR SECURITY AUDIT TRAILS ANALYSIS

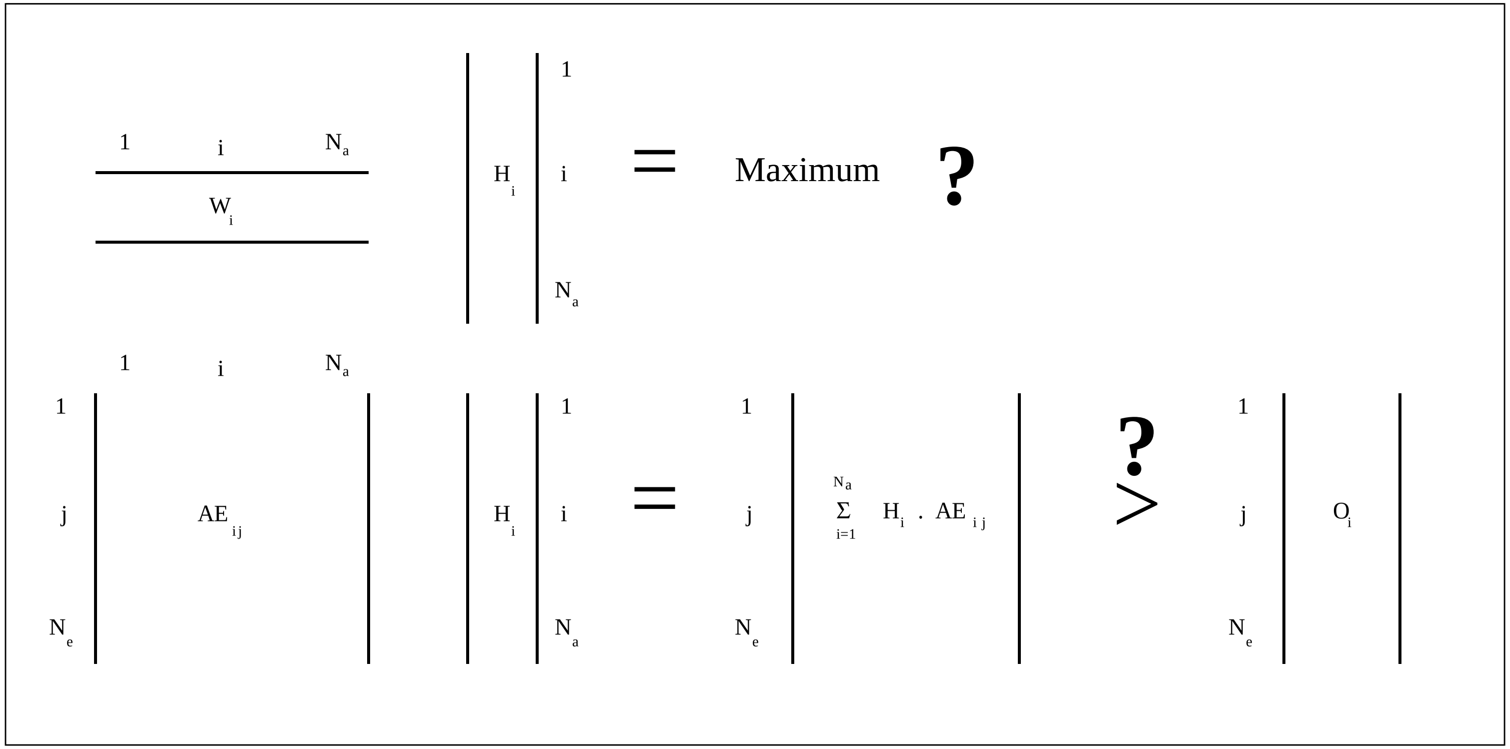
Ludovic Mé
Ludovic.Me@supelec.fr
http://www.supelec-rennes.fr/rennes/si/equipe/lme/

Supélec
BP28
F35511 Cesson-Sévigné Cedex
FRANCE
tel.: (+33) 299.84.45.00

G$^{a}_{S}$SA$_{T}$A: Main Ideas

- To investigate misuse detection
- No timing aspect in attack scenarii
- A pessimistic approach
- A heuristic mechanism (genetic algorithm)

Our View of the Security Audit Trail Analysis



⇑ .Misuse detection .No timing aspect

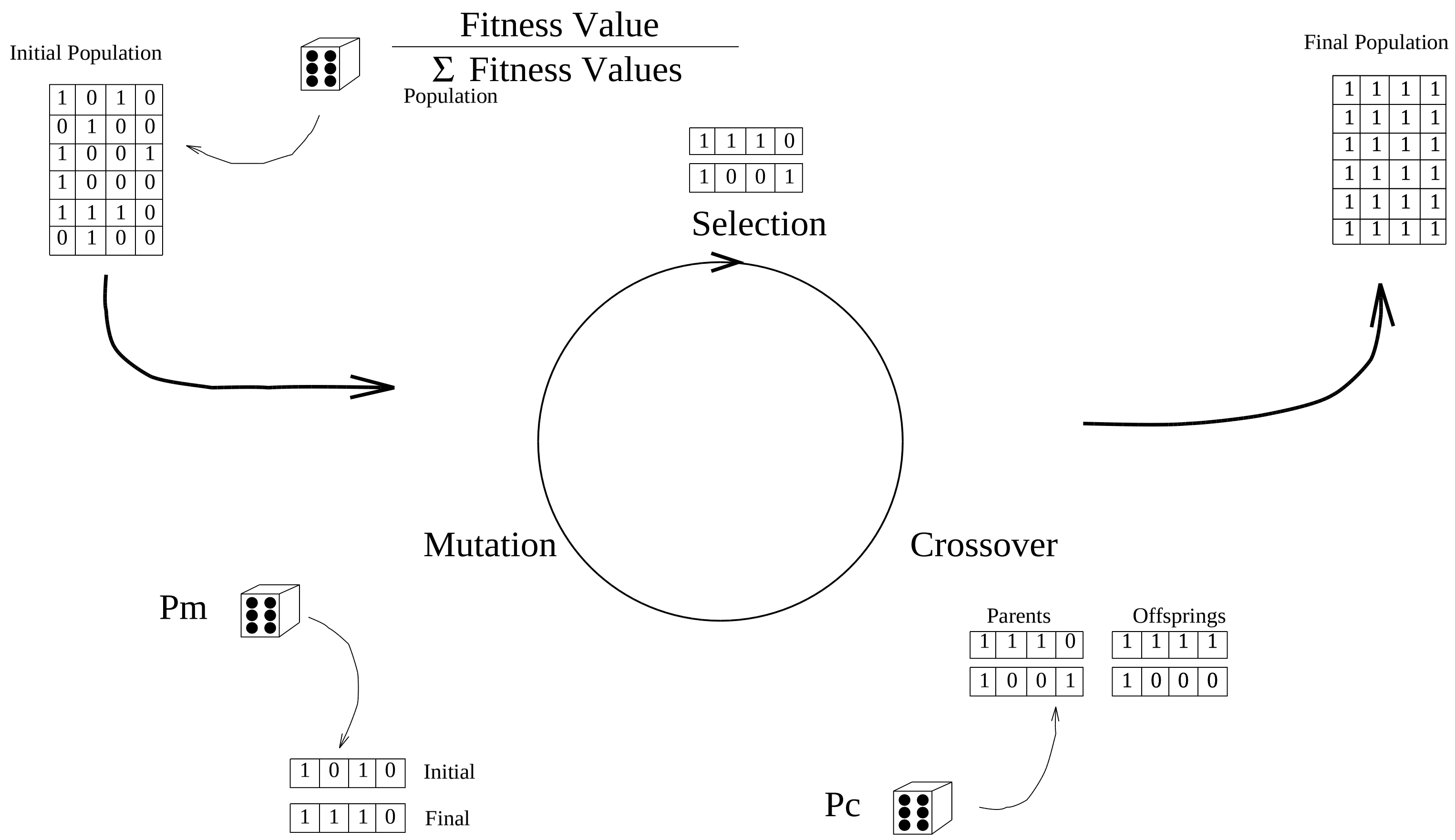
⇑ A pessimistic approach

An Heuristic Approach to Find the *H* Vector

- 2^{N_a} possible values $\Rightarrow$ systematic exploration impossible
- A heuristic approach:
  - A hypothesis is made
  - Hypothesis assessment
  - According to this evaluation, derivation of a new (and better) hypothesis

  This process is repeated until a solution is found
- A tool: a genetic algorithm

A Simple Genetic Algorithm



$$\frac{\text{Fitness Value}}{\sum_{\text{Population}} \text{Fitness Values}}$$

Initial Population

1	0	1	0
0	1	0	0
1	0	0	1
1	0	0	0
1	1	1	0
0	1	0	0

Selection

1	1	1	0
1	0	0	1

Crossover

Parents				Offsprings			
1	1	1	0	1	1	1	1
1	0	0	1	1	0	0	0

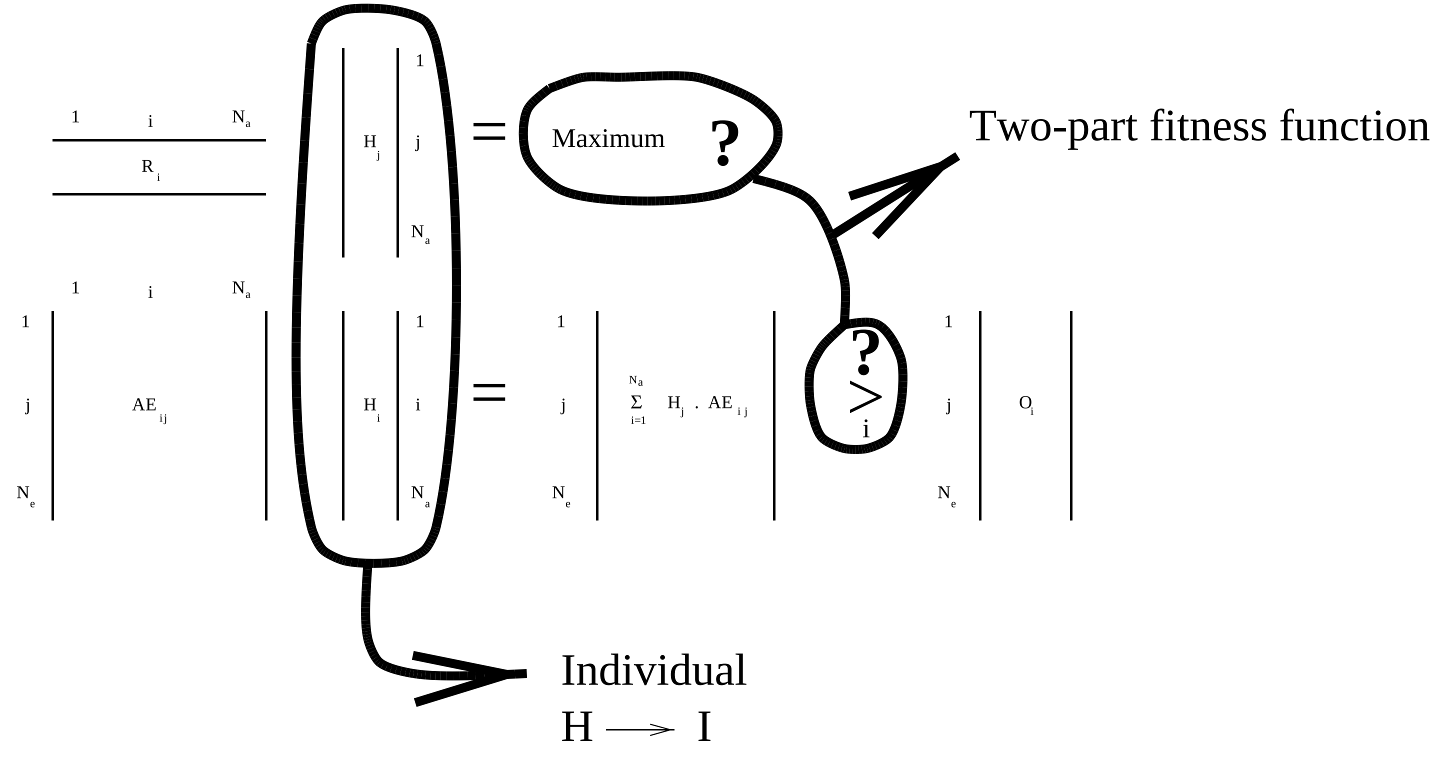
Mutation

1	0	1	0	Initial
1	1	1	0	Final

Final Population

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Individuals and Fitness Function



Experiments

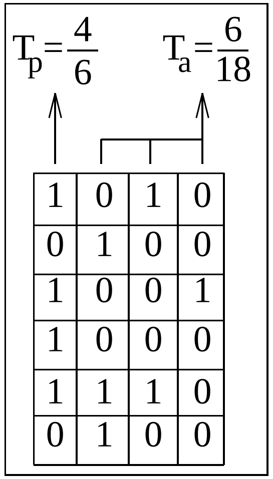
- Data generated by the AIX audit sub-system
- Users: sequences of commands over a 30 minute period (no attack)
- The attack base contains between 24 and 200 attacks
- Attacks are included in the audit vectors generated from the sequences of commands
- Questions:
  - How does the population evolve?
    What is the final population?
  - Is the running time satisfactory?
    How does it evolve in function of the number of attacks in the base?

How to Evaluate the Quality of the Results ?

Defining the Ratios T_p and T_a

T_p ⇒ number of individuals in which bits corresponding to present attacks equal 1 out of the total number of individuals (ideally $T_p = 1$)

T_a ⇒ number of individuals in which bits corresponding to absent attacks equal 1 out of the total number of individuals (ideally $T_a = 0$)



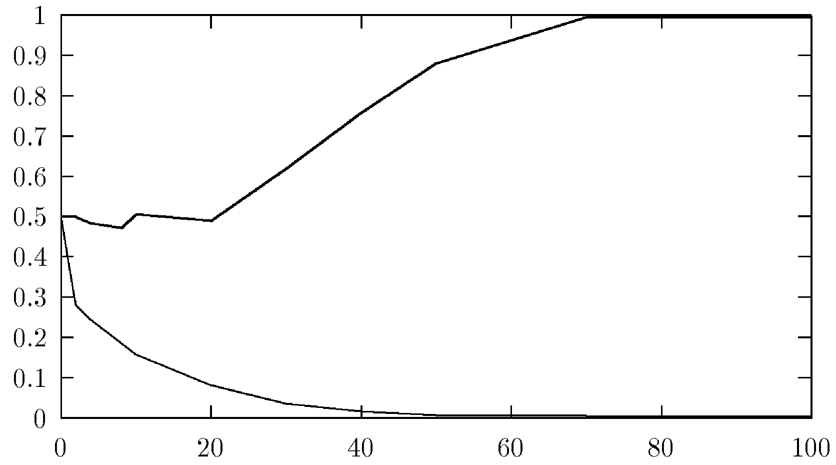
$T_p = \frac{4}{6}$ $T_a = \frac{6}{18}$

1	0	1	0
0	1	0	0
1	0	0	1
1	0	0	0
1	1	1	0
0	1	0	0

We **know** that only attack 1 was performed

Ludovic Mé
Ludovic.Me@supelec.fr

A Classical Evolution of T_p and T_a



- The population converges ⇒ A good discrimination between present and absent attacks
- The number of attacks actually present in the trail have no influence on this result

Execution Time vs Number of Attacks in the Base

Number of attacks	Execution time	Exploration rate
24	18”	3×10^{-3}
40	32”	5×10^{-8}
100	104”	5.9×10^{-26}
200	625”	6.3×10^{-56}

$P_c = 0.7$, $P_m = 0.002$, 500 individuals
λ generations for constant T_p and T_a
28 types of events in the matrix
IBM RS6000 320

- The running time does not grow exponentially
- The duration of the audit session has no influence on the running time

Conclusion

- What we do not do:
  - We cannot detect the multiple realization of a particular attack
  - We do not precisely locate attacks in the audit trail
- Future work:
  - Use GASSATA in a real environment (some code should be rewritten)
  - Improve our attack base
  - Find a comparative measurement process