Applications of Immune System Computing

Ricardo Hoar

What kind of applications?

- Computer Security
- Pattern Recognition
- Data Mining and Retrieval
- Multi-Agent Systems
- Design Optimization
- Control Applications
- Robotics
- …

A Distributed Architecture for a Self Adaptive Computer Virus Immune System

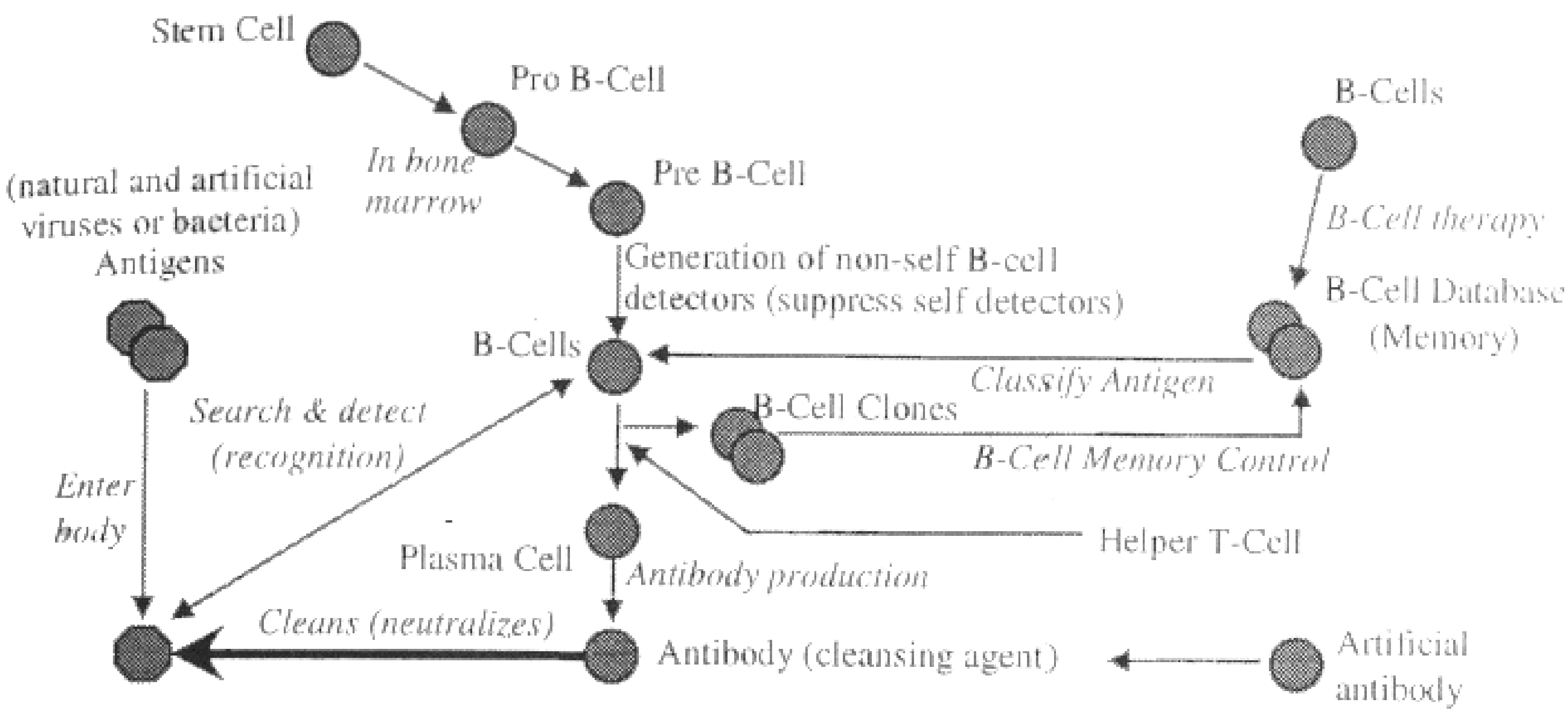
Gary B. Lamont, Robert E. Marmelstein,
and David A. Van Veldhuizen

- Simplified Biological IS Model (BIS)
- Relationships between BIS and CVIS
- CVIS model
- Discussion of some algorithms involved in CVIS

Simplified Biological IS Model

- Extracellular BIS
  - High level set of interacting components:
    - **Generator/Repressor**
      - B-cells, antibodies
    - **Detector**
      - Detect antigen , detect host/non host
    - **Classifier**
      - Once antigen detected, B-cell determines type
    - **Purger**
      - Eg. Macrophage , antigen purging or cleansing
    - **BIS memory**
      - A store of successful B-cell threat responses
    - **Adaptation process**
      - Continual updating in reaction to imperfect coverage of all pathogens

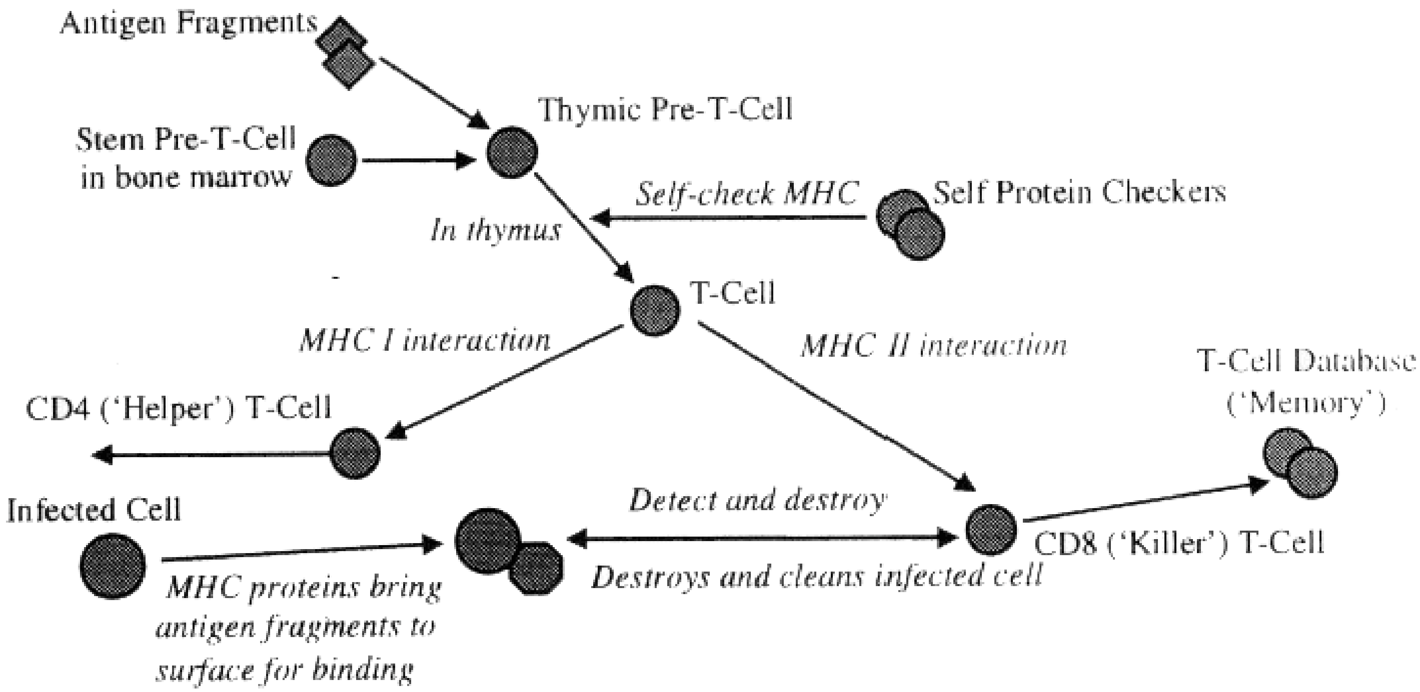
Extracellular BIS (Diagram)



Intracellular BIS

- Attempts to find antigens within living human cells.
- Generate “Helper” T Cells which can promote antibody prodection from B cells
- Antigen Presentation
- Major Histocompatability Complex (recognition by T cells)

Intracellular BIS (Diagram)



Computer Virus

- Significant Computer Threat
- High “birth rate” of new viruses
- Inability of Anti-Virus software to detect the newest Viruses.

Current Methods for Virus Scanning

- Current Virus scanning Software looks for bit patterns known to belong to a specific virus. Additionally deductive techniques use “rules of thumb” to identify programs that exhibit “virus like” behaviors.
- Although reliable , these methods rely on static knowledge bases, resulting in a the need for continual updating.

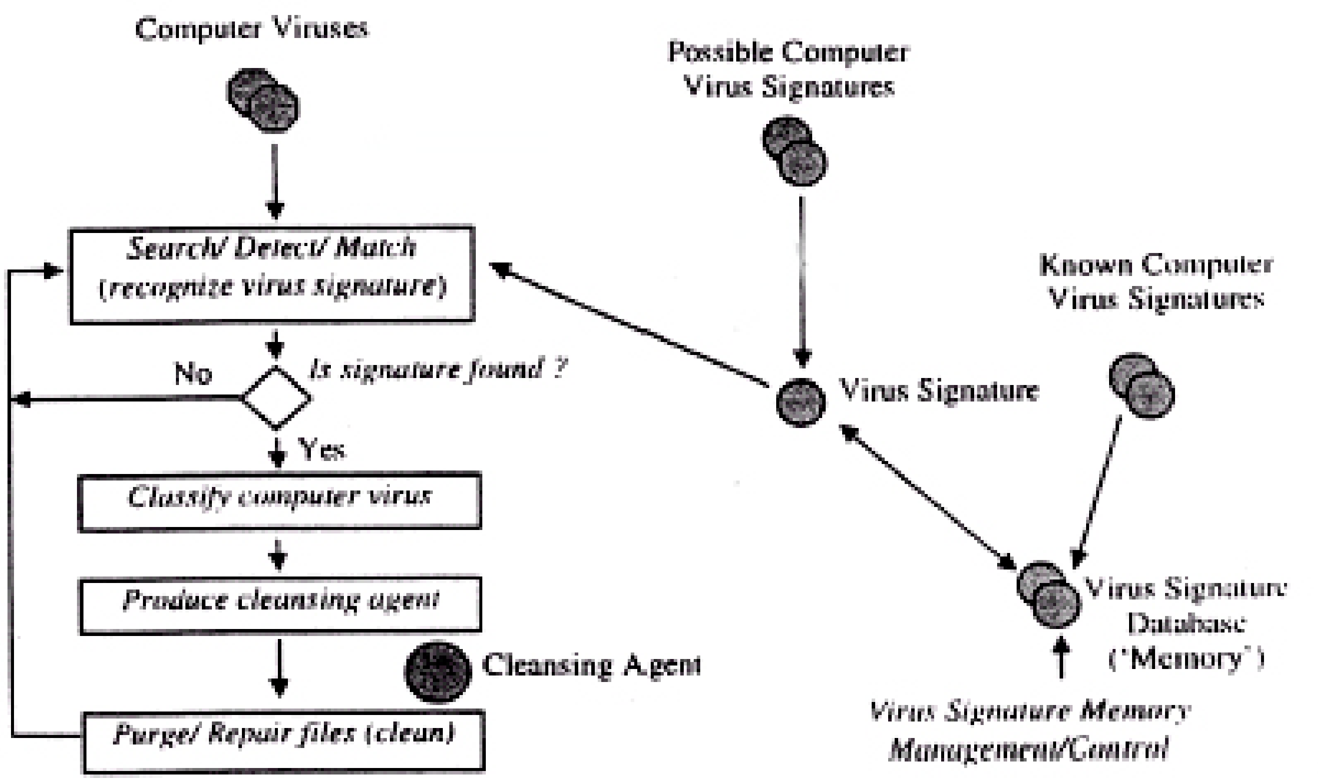
More robust method needed

- Why not apply the principals from immune computing to this obvious application of scanning for Viruses?
- Which components of BIS can be used to define a Computer Virus Immune System ?
- What are the main implementation challenges?

Computer Virus Immune System

- Components
  - Genereate/Suppress Virus
    - Generate random signatures, Compare signatures to prior sig.
  - Classify Virus
    - Isolate virus based on its characteristics, signature extraction
  - Purge Virus
    - Purge the virus and repair damaged system resources
  - Augment Virus Database
    - If new virus, add to memory
- Main Challenge
  - Replicating BIS inherent parallelism

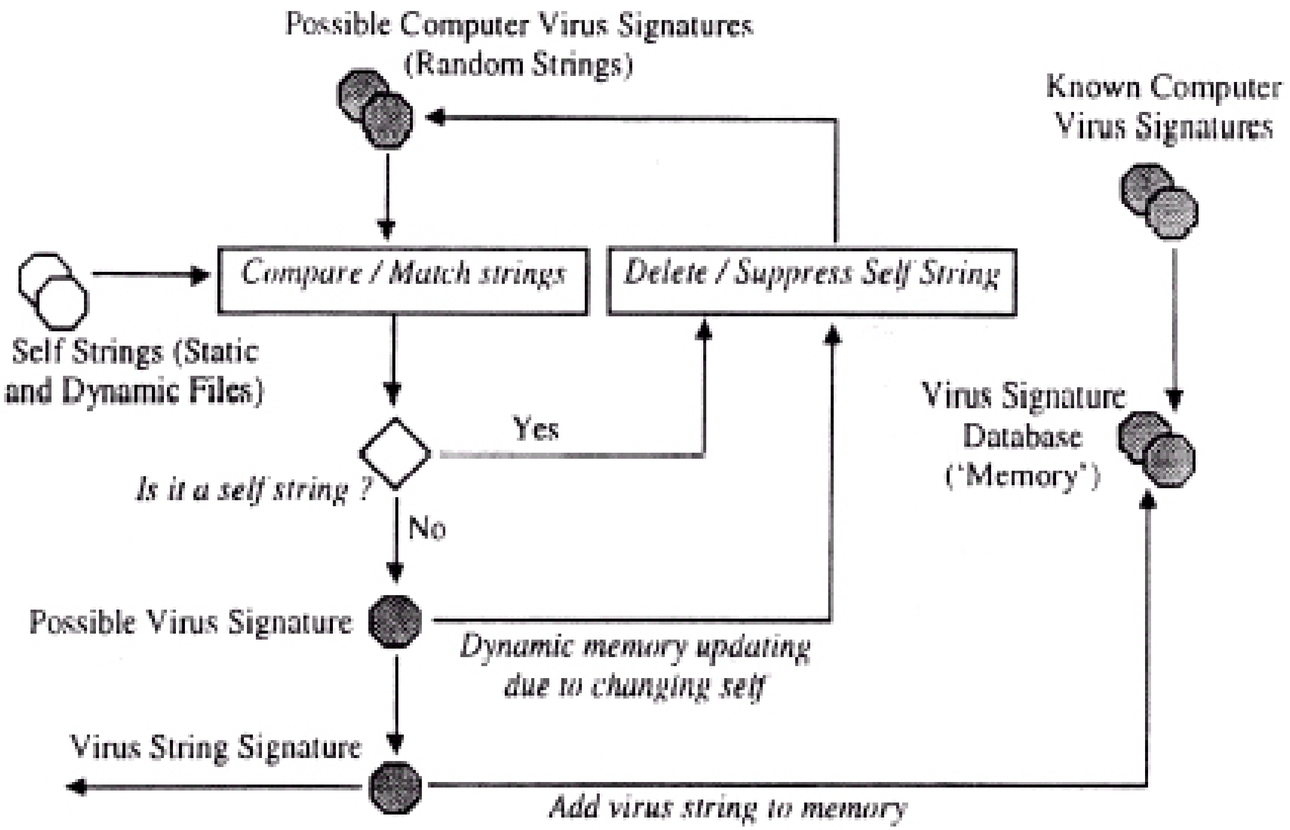
Generic CVIS Algorithm



Self/Non-Self Determination

- Distinguishing legitimate computer resources from those corrupted by a computer virus
- Accomplished via *detectors* generated at random and compared to protected data
- Requires a significant number of detectors
- Can become cumbersome if protecting changing files due to creation of new detectors

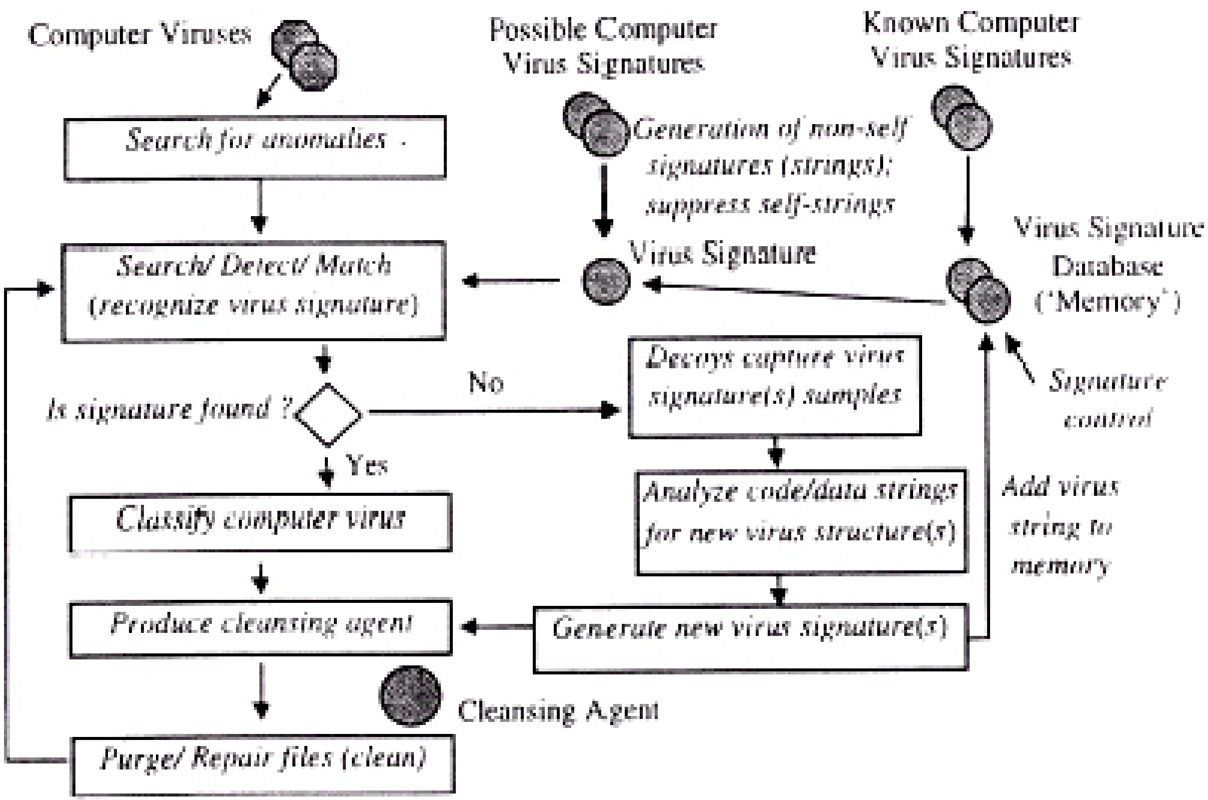
Self/Non-Self Determination Algorithm



Virus Decoy

- Uses decoy programs whose sole purpose is to become infected
- Infected decoy can automatically extract viral signature
- Does not require the regeneration with changing files
- Must be used in conjunction with another method to identify classified viruses

Virus Decoy Algorithm



Immunity by Design: An Artificial Immune System

Steven A. Hofmeyr
and Stephanie
Forrest

ARCHITECTURE

- To preserve generality, we represent both the protected system (self) and infectious agents (nonself) as dynamically changing sets of bit strings.
- In cells of the body the profile of expressed proteins (self) changes over time, and likewise, we expect our set of protected strings to vary over time.
- The body is subjected to different kinds of infections over time; we can view nonself as a dynamically changing set of strings.

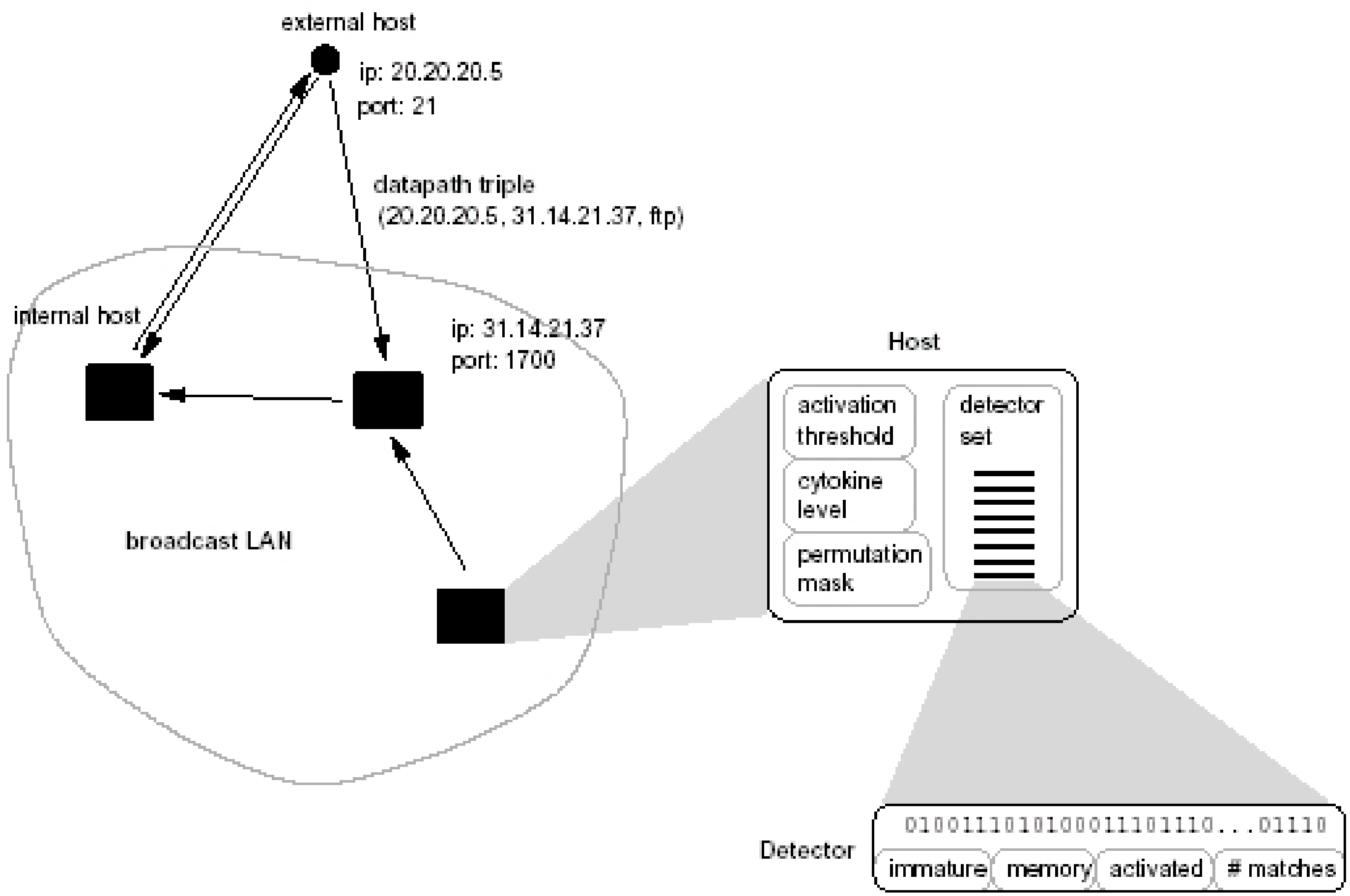
EXAMPLE: NETWORK SECURITY

- We define self to be the set of normal pair wise connections (at the TCP/IP level) between computers.
- A connection is defined in terms of its “data-path triple”—the source IP address, the destination IP address, and the service (or port) by which the computers communicate. (49 bit string)
- Self signifies recognized familiar addresses while Non-self represents “foreign” addresses

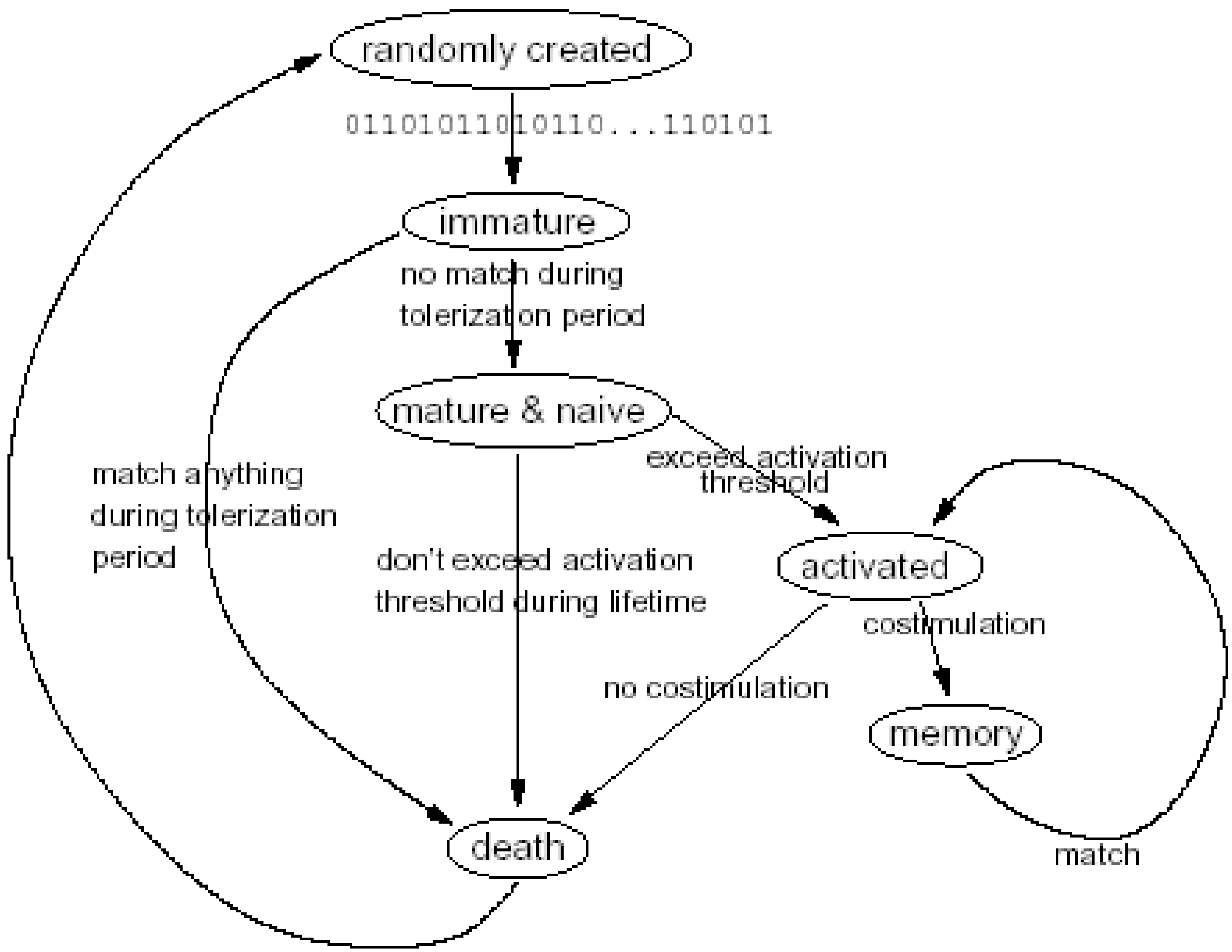
NETWORK SECURITY

- Each detector cell is represented by a 49 bit string.
- Detection = String Matching
- New detectors are randomly generated and eliminated if they are matched while still immature (removal of self)
- Mature detectors can activate an alarm if a threshold is reached or be removed if they remain unmatched.
- This balance between naïve immature and mature cells gives the system adequate adaptability to new antigens.

The Architecture of the AIS.



Lifecycle of a detector



EXPERIMENTAL RESULTS

- Two data sets were collected:
- The self set was collected over 50 days.
- Self = 1.5 million datapaths mapped to 49-bit binary strings.
- At time 0 in the simulation a synthetic attack was detected with probability $p = 0.23$.
- After letting the system respond and adapt for 3 months attack detected with probability 0.76, demonstrating the effectiveness of AIS for learning

Combinatorial Optimization (n-TSP Problem)

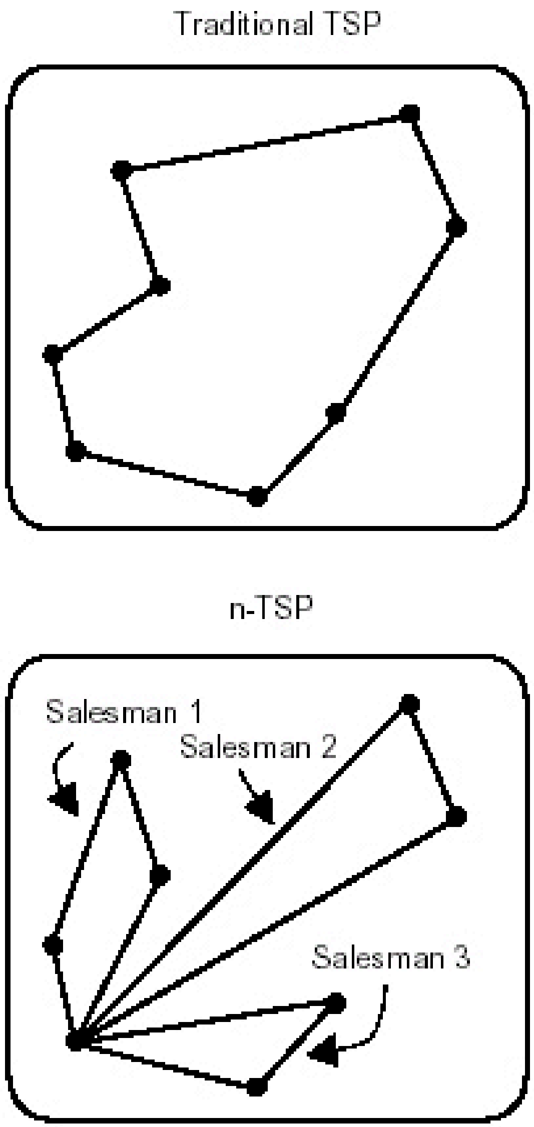
Combinatorial Optimization (n-TSP Problem)

- Endo *et al*. (1998) and Toma *et al.* (1999) proposed an adaptive optimization algorithm based on the *immune network* model and *MHC peptide presentation*. In this model, immune network principles were used to produce adaptive behaviors of agents and MHC was used to induce competitive behaviors among agents. The agents possessed a sensor, mimicking MHC peptide presentation by macrophages, the T-cells were used to control the behavior of agents and the B-cells were used to produce behaviors.

Problem Comparison

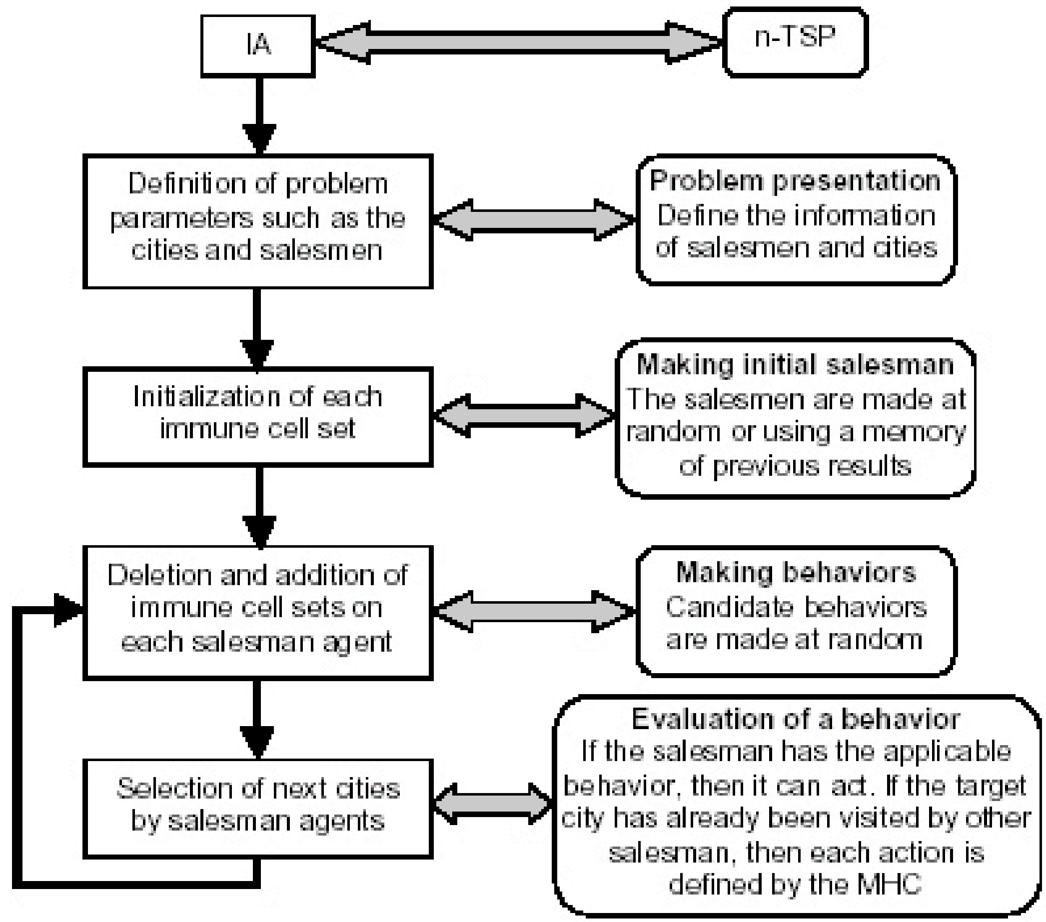
Table 4: Immune cells and molecules and their roles in the n-TSP problem solving.

Immune System	Role in the n-TSP problem
Antigen	Contains information about the cities and salesmen
Macrophage	Selects the city number that the salesman agent must visit
T-cells	Help the activation of B-cell
B-cells	Produce antibodies
Antibody	Perform the behavior of an agent



Traditional TSP
n-TSP
Salesman 1
Salesman 2
Salesman 3

(a)



IA
n-TSP
Definition of problem parameters such as the cities and salesmen
Problem presentation
Define the information of salesmen and cities
Initialization of each immune cell set
Making initial salesman
The salesmen are made at random or using a memory of previous results
Deletion and addition of immune cell sets on each salesman agent
Making behaviors
Candidate behaviors are made at random
Selection of next cities by salesman agents
Evaluation of a behavior
If the salesman has the applicable behavior, then it can act. If the target city has already been visited by other salesman, then each action is defined by the MHC

(b)

Refrences

- de Castro, L, Zuben, F. *ARTIFICIAL IMMUNE SYSTEMS: PART II – A SURVEY OF APPLICATIONS* Technical Report DCA-RT 02/00 accessed from: http://www.cs.plu.edu/pub/faculty/spillman/seniorprojarts/ids/part2.pdf
- Hofmeyr, S., Forrest S. *Immunity by Design: An Artificial Immune System*
- Lamont, G., Marmelstein R., Veldhuizen D. *A Distributed Architecture for a Self Adaptive Computer Virus Immune System*
  New Ideas in Optimization
- Dasgupta, D. *Artificial Immune systems: Theory and Applications* Tutorial WCCI 2002, Honolulu Hawaii.