IST Project N° 027568

Integrated Project

IRRIIS

Integrated Risk Reduction of Information-based Infrastructure Systems

Deliverable D 2.2.3
Overview on Bio-inspired operation strategies

Due date of deliverable: 31 July 2007
Actual submission date: 10 September 2007

Revision [1.0.0]

Technical University Dresden

Dissemination Level

PU	**Public**

Start date of project: 01 February 2006 Duration: 3 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Authors	**Limor Issacharoff (TUD)** **Lubos Buzna (TUD)**

Contributor	**Dirk Helbing (TUD)**

Work package	**WP 2.2: LCCI interdependency analysis: interdependency understanding**
Task	**Task 2.2.4 Formulation of strategies for the management of interacting networks (after month 18)**

Table of contents

Figures

Abbreviations

ABC	ant based control
ACO	ant colony optimization
ADIS	adaptive immune system
AIS	artificial immune system
AHWN	ad hoc wireless networks
CI	critical infrastructures
DNA	Deoxyribonucleic acid
GA	genetic algorithm
GP	genetic programming
IS	immune system
LAN	local area network
LCCI	large complex critical infrastructures
PSO	particle swarm optimization
RAR	reliability adaptability and robustness
SGA	simple genetic algorithm

Motivation

One of the future (after month18) tasks in the IRRIIS project is the formulation of bio-inspired operating strategies for LCCI (T.2.2.4). As in the whole project, the formulated models should relate to a system of systems, i.e. they should include the interdependency between the different critical infrastructures and strive to achieve a better performance and an efficient and fast recovery in case of failures. Bio-inspired methods proved to be efficient tools for the solution of various problems, including technological-systems related problems. It is thus natural to try to and apply bio-inspired methods also to the field of Critical Infrastructures (CI). The purpose of this review is thus twofold. Firstly, to collect information about existing bio-inspired strategies which were already implemented successfully and could be used by the IRRIIS project consortium; and secondly, to hint or suggests possible new direction to be followed. We have therefore tried to give some insight on both the biological phenomena and the models developed on their basis. Most models/algorithms presented in this review are aimed at increasing the robustness of the system either by means of better design, control and maintenance, and improved recovery strategies. In the description of application models we restricted ourselves to the IRRIIS main domains: electricity power systems and telecommunication systems (computer networks, ad hoc networks etc).

Connection to other deliverables

This review is a preliminary step in the fulfillment Deliverable 2.2.7 "Provision of bio-inspired algorithms treating LCCI interdependencies". In task T2.2.4 we are to "derive new approaches to crisis management and recovery by analysis of biological systems, which has not been thoroughly done in the past". We therefore begin by exploring the existing approaches and applications. After month 18 of the project, task 2.2.4 is to take place and the approaches will be formalized as specific algorithm/s – such that they are translatable to actual codes and can be integrated e.g. in SimCIP. More specific details can be given only once the tasks in a more advanced stage.

1. Introduction

Though the term 'Bio-inspired' has entered the scientific community lexicon only recently, the actual use of biological mechanisms as an inspiration for models and algorithms in various fields of science was already practiced centuries ago by researchers such as Leonardo Da Vinci, whose drawings of flying machines where inspired by bats, birds and other flying animals. Modern science has rediscovered this approach in the middle of the last century, when it took the name `Bionics'. The term Bionics (or, in modern jargon, Biomimetics), coined in the 1950's[1] refers to the application of methods and systems found in nature to the study and design of engineering modern technology. Also Artificial Intelligence – the science and engineering of making intelligent machines, rose in the same time period. Along with the progress in the different biological sciences such as genetics, bio-molecular and bio-cellular biology, more and more bio-related fields emerged such as biomedicine, bioinformatics, biotechnology, biomechanics etc. A schematic drawing of the different fields and the connection between them is depicted in Figure1.

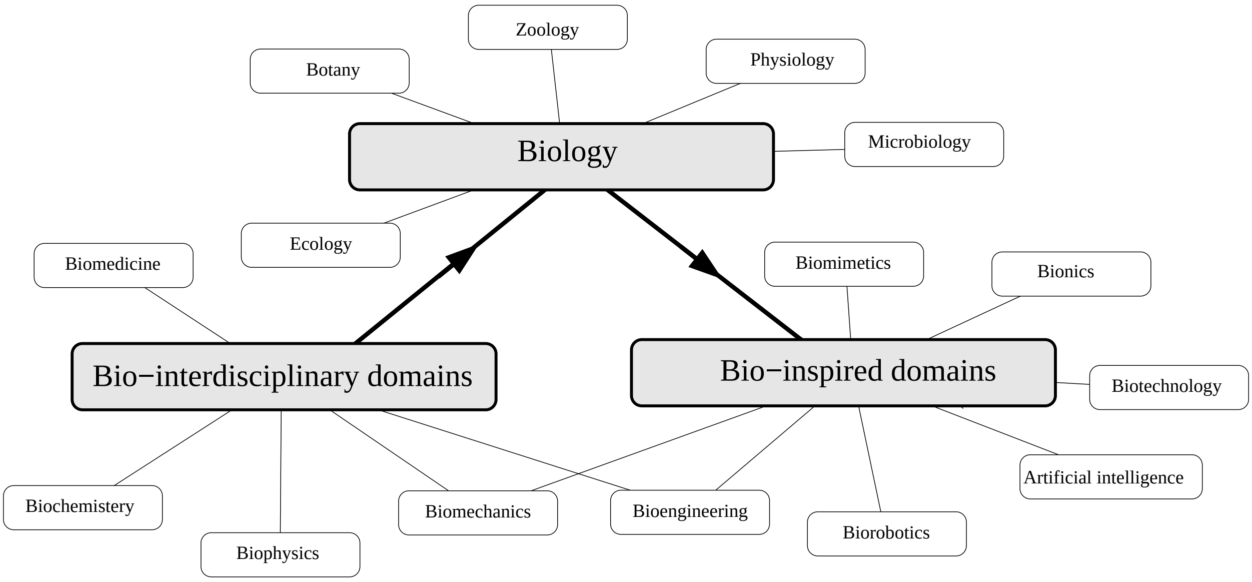


Figure1: A schematic diagram of bio-related domains. In Bio-inspired domains, biological methods are implemented to different fields while in the Bio-interdisciplinary domains knowledge from various fields is used in the study of biological system.

Some biological systems, such as ant colonies, display a remarkable degree of stability, adaptability to environmental changes and resistance to failures in spite of the fact that they are essentially self-organized and distributed. Such properties would be highly desirable in several contemporary engineering systems. Indeed in the management of large complex critical infrastructures (LCCI) networks it is crucial to satisfy the requirements of reliable supply of services with minimum costs. Moreover they are subject to internal and external conditions that change in time; demand, operation-costs and systems' functionality are dynamic properties and their behaviour is not always predictable. Most importantly, they normally operate on multiple levels so that a centralized control gets more and more difficult as the complexity of the system increases. Highly adaptive systems and adaptive operation strategies are required to maintain high quality of service under these changes, including situations where failures may cause part of

[1] The term Bionic was coined in 1958 by J. E. Steele. The term Biomimetics was coined by Otto Schmitt.

the system to malfunction. Reliability, adaptability and robustness (the **RAR** properties) are therefore important properties which one should aim to achieve by the proper design and operation of the systems. Some biological systems, which possess RAR and other valuable properties, can then provide guidelines for design and operation of complex engineering systems.

The next step is identifying the ingredients of biological reality that can be “transported” into the technological context. Here, the rise of Complexity Science has helped to reveal many common features shared by technological and biological systems, thus triggering a number of studies in which tools and knowledge from biology are implemented in technology fields. This has opened the door for the search of RAR mechanisms and other valuable properties of biological systems that could be used to improve the design of other complex networks including LCCI networks. However, the issue of interdependencies (which is crucial in engineering applications) has just started to be properly treated within Complexity Science and there are but a few well-defined interdependencies models, none of which is bio-inspired in the sense described above. However, in the study of the individual networks (i.e. neglecting interdependencies) and especially for the different technological networks, bio-inspired approaches seem to be a growing trend. Here we focus our attention on these approaches, and particularly on those that are of interest for IRRIIS key issues, namely the electricity power system and its supporting telecommunication infrastructures, the latter including fixed and ad hoc communication network and computer networks.

Bio-inspired methods for individual networks are implemented to models of many technological networks and in some cases such methods are already integrated in the real systems. The main biological approaches in these models can be classifies into three groups:

- Evolutionary based methods (genetic computing)
- Social insects based methods (swarm intelligence)
- Immune system based methods.

The first ones implement nature’s evolutionary mechanisms of natural selection and diversity in the search of better solutions in optimization problems: the formulation of the problems is derived from genetics with chromosome-like strings representing a population of possible solutions. Social insect based methods (a.k.a. Swarm Intelligence) are mostly derived from empirical observations of the behaviour of ant colonies and bee colonies, and usually translate into computational techniques. The last category of methods uses analogies with the working of the human immune system to develop responses to intrusion detection in computer networks. The above points will be treated separately in the following three sections, each composed of a short biological introduction and a part on applications.

2. Evolutionary based methods

2.1 Theory of evolution and natural selection

Modern evolution theory describes the process of change over time in the inherited characteristics of the living organism. Charles Darwin introduced the theory of **natural selection**, which is one of the main mechanisms controlling the evolutionary process. Natural selection ensures that in each generation only the fittest organisms survive, i.e. those who have the characteristics, which allow them to perform best in their surrounding (reproduce, gather food etc). The information on inherited characteristics is found in the organism's cells. The cell is the basic unit of all organisms; it contains chromosomes, which are long strands of DNA; proteins, which are transcribed by the instructions contained in DNA; and other molecules (metabolites), which are resulting from protein-protein interactions. DNA is a molecule composed of four types of bases. The different regions in the DNA are called "genes" (if they code information to produce proteins); they are separated by large portions of apparently un-useful DNA regions (called "non-coding" regions). The specific assembly of chromosomes (the "genotype") determines the inherited characteristics (traits) of an individual by defining its "phenotype". There are several mechanisms which control the change of chromosomes, external influences such as radiation and virus can change the DNA sequence as well as internal processes such as errors in DNA replication. These changes are called **mutations**. Chromosomes also change in the process of sexual **reproductio**n. Offspring of sexual organisms contain a random mixture of their parents' chromosomes thus each new generation is different to some extent from its predecessor. Natural selection, reproduction and mutation are the principle ideas adopted and developed for the formulation of Evolutionary computing models, methods and algorithms.

2.2 Evolutionary methods – Genetic Algorithms (GA)

Genetic algorithms (**GA**) is the more widely known and used evolutionary computing method. It was first introduced by Professor John Holland as a model for the evolutionary process in nature [Holland 1975]. The algorithm in the base of this model is now referred to as the simple genetic algorithm (SGA) and the main idea behind it is related to the competition between populations of evolving candidate solution of a given problem. The population is a set of possible solutions (the organisms). Each individual of the population is described by a chromosome. Each possible solution has a different chromosome structure. In SGA, a chromosome is given in the form of a binary string. As in the evolution of organisms the population of solutions now represented by strings, is evolving by performing repetitively the operations of selection, reproduction and mutation. Here selection of the nature is replaced by a fitness function which measures the fitness of the solutions; reproduction is done by a process corresponding to the sexual reproduction i.e. the offspring chromosomes are produced by mixing chromosomes of parent solutions; mutation is the random process of change in the chromosomes. The main step in the algorithm can then be summarized as follows:

- Initialization - At first, a population of candidate solutions (strings) is generated.
- Selection - to decide which solution is chosen for reproduction (creation of new solutions), a fitness function evaluates each of the individuals of the populations. Those with the best fitness values are chosen.
- Reproduction - a new generation is produced by applying the crossover operator in the chosen individuals. Crossover can be done in various ways, the simplest one is to cut two

strings in some given identical position creating four strings segments and then constructing two new strings by interchanging these segments.

- Mutation - Following the crossover some of the strings in the new generation subject to mutation. In binary string representation this would mean that one or more of the bits in a string change their value from one to zero or vice-versa.

Each of these steps: Initialization, selection, reproduction and mutation is a challenging task by itself. First, the choice of representation of the chromosome and the fitness function must be adapted to the specific problem in question. Further, the evolutionary operators can be implemented in many different ways and finally the parameters of the algorithm must be set such that the solution will converge in reasonable time and by using reasonable amount of computational resources, this includes e.g. deciding the size of population and mutation rate. In more advanced versions of GA that followed the seminal SGA, various techniques were developed with the aim of improving the performance of GA, e.g. the binary representation was replaced by the more intuitive and flexible non-binary representation (see 2.3). Further, the selection process which was traditionally performed on the base of the fitness function value assigned to the solution population was revised as experiences in several fields have pointed on the fact that best solutions are found by mating the best individuals also with individuals with a lower fitness; this has the advantage of introducing into the method a fictitious "temperature" to prevent the falling into local minima.

The theory of GA and the expansion of the method discussed here are very well documented in literature. Most papers recommend the classical textbook by [Goldberg 1989]. Other examples for GA introductory are [Tomassini 1995] who gives a basic tutorial of GA including a step-by-step example of implementation. He further discusses the progress in GA with regards to representation, selection and genetic operators. Finally the implementation of parallel GA is discussed with the example of grid and island implementation. It should be noticed here that indeed, GAs are highly parallelizable and therefore they are prone to be efficiently treated with large computational machines. This is an especially advantageous feature of GA (as apposed to for example simulated-annealing) in the treatment of very hard optimization problems, which are often encountered in the real world, especially in technological LCCI system. The basic theorem of GA **schemata,** introduced by Holland and advancement in theory are discussed in [Srinivas 1994].

2.3 GA applications to Electricity Power Systems

It is difficult to find a subtopic of power systems engineering where GA was not applied. We list some of these subtopic and the related problems to which GAs were applied in the end of this section. The operation strategy we chose to introduce as an example is that of **power system restoration** as the subject of recovery procedures is a reoccurring topic in IRRIIS[2].

<u>Supply restoration and optimal load shedding in power system distribution networks</u>

In power distribution systems islanding of distribution zones is frequently necessary for performing network maintenance tasks. Islanding can also result from unexpected faults which will cause the activation of defence mechanism which disconnects some zones. Restoration is the operation of reconnecting the islanded zones. "Service restoration is essentially accomplished by transferring the loads in the out-of-service area to the neighbouring supporting feeders via 'ON-OFF' control of different switches in the distribution system" [Kumar 2006]. In the operation of

[2] "supporting automated recovery and service continuity in critical situations" is in the definition of the projects objective [IRRIIS annex]. Specific related task is e.g. task 2.1.4

restoration the order in which switches are opened and closed cannot be arbitrary, for both operational and economical reasons. Therefore, the restoration problem is mainly to decide the best order of operation. The number of switches needed to be considered in the solution depends on the specific configuration of the problem (the size of island, location etc) but can range from very few to tens and sometimes hundred of switches. Along with the additional restrictions the power system is subjected to, this makes the restoration problem a highly complicated multi-objective multi-constraint task. We now present one possible application of GA applied for the solution of this problem which was presented by [Luan 2002]. According to [Luan 2002] the operation of restoration should result with

I. Minimal possible load shed

II. Minimal switching operation

III. Minimal energy losses

while keeping the following constraints:

1. Conservation of system's radial structure (no loops).
2. No violation of busbar voltage limits.
3. No current overloading in the lines.

Requirements I-III are mostly for economical reasons while constraints 1-3 are more physical constraints for safety operation. Solution of the problem with evolutionary approach is done according to the genetic algorithm approach; first the problem is represented accordingly:

1. solution is represented by a chromosome (string),
2. a fitness function to evaluate the goodness of the solution is defined,
3. suitable operators for the chosen stings are defined for the operations of crossover and mutations.

We remind that the solution of the problem, i.e. the string, should represent the order in which the switches of the system are to be turned on or off. In [Luan 2002] this is done with an integer permutation encoding scheme. In this method the chromosome is a set of n_{sw} integers where n_{sw} is the number of switches in the study. The position of elements in the string (chromosome) defines the order of operation: thus for a system of three switches the string $[3,1,2]$ means that operations will be preformed first on switch 3 then on switch 1 and last on switch 2. The state of the switches in the final solution is denoted by the signs '+' for an open state and '-' for closed state. The 'quality of a string will be evaluated by the fitness function which encapsulates requirements I-III and constraints 2,3:

$$f = a_1 P_{LL} + a_2 P_{PL} + a_3 I_{CO} + a_4 V_D + a_5 \sum_{i=1}^{n_{sw}} C_i$$

where P_{LL} is the relative loss of loads (load shed), P_{PL} is the relative power loss, I_{CO} is the relative current over load, V_D is the relative voltage deviation from limits and C_i is the cost operation on switch i when passing from the existing state to the state required by the evaluated configuration (string). The assignment of different costs to the switches allows to consider, both the number of operations and the operations costs. E.g. the operation of a manual switch or an automated switch can be very different especially from time point of view.

Finally suitable evolutionary operators need to be chosen. The basic operation of the crossover operator is to exchange string-segments of parent solutions, where parents are two strings randomly chosen from the solution population. For the purpose of demonstration suppose we are solving a configuration of 10 switches and that the two parent-strings chosen randomly from the solution population are 'A' and 'B' (see Figure 2)[3]. The substrings to be exchanged are randomly chosen with uniform probability of starting and ending sites, in this case site 4 and site 6 correspondingly. Notice however, that a simple exchange i.e. replacing the substring of 'A' ([5,6,7]) with the substring of 'B' ([2,3,0]) is not possible since each of the strings needs to contain all the integers 1-10 (each switch should appear once in each string). Therefore, the operation of crossover is done by the OX order operator, that switches substrings, such that the resulting strings are still valid solutions. The operation of the OX operator as given in [Goldberg 1989] is demonstrated in Figure 2. The mutation operator makes changes in the offspring (the strings resulting from the crossover). Mutation is done simply by exchanging the location of two integers in a string.

(a) A = 9 8 4 '5 6 7' 1 3 2 0
B = 8 7 1 '2 3 0' 9 5 4 6

(b) B = 8 * 1 '2 3 0' 9 * 4 *
(c) B = 2 3 0 '* * *' 9 4 8 1

(d) OX(B,A) = 2 3 0 '5 6 7' 9 4 8 1
(e) OX(A,B) = 5 6 7 '2 3 0' 1 9 8 4

Figure 2: The operation of the OX crossover operator applied to the 'A' and 'B' parent strings. (a) a substring location is randomly chosen with uniform probability of starting and ending sites and the corresponding location in the parent strings are marked. (b) The integers appearing in the substring of 'A' are removed from 'B'. (c) Starting in the substring end position, all elements of 'B' are shifted to the left (cyclic motion) such that the substring location is freed. (d) The substring of 'A' is put to the gap in the 'B'. (e) The operation applied to 'B' are then applied in the same manner to 'A' such that in the end the substring of 'B' is put to 'A'.

After the chromosome, the fitness function and the operators are defined, a population of strings is randomly generated. At first it is necessary to validate that the randomly generated strings will represent feasible configuration i.e. the configuration a string represents should comply with constraint **1** (radiality). If we think of the distribution system as a graph in which the switches are edges and the system elements between the nodes, called zones, are the nodes of the graph then the radiality constraint means that the graph must not contain cycle (loops). Figure 3 depicts a distribution system and Figure 4 is the corresponding graph. Cycle detection in graphs is a standard problem and can be solved in several ways (for the exact solution proposed by the authors please consult the paper). So in the validation process the configuration resulting from each string is checked for cycles. If a cycle is found, one of the switches in the cycle is opened; in the graph this means the removal of an edge and, in the string, it means turning + to – sign for the string element representing the edge). The resulting string is then a feasible solution.

Next the strings are evaluated by the fitness function where the rest of the constraints and requirements of the solution are evaluated. Strings with the best values of fitness function (the lower the value the best is the solution) are chosen to formulate parent's population; this step is sometimes referred to as the **selection** step as only the best of the population survive and are selected for the production of the new generation. The parents are subjected to the reproduction operator and undergo crossover creating offspring population. Finally parents and offspring are subjected to the mutation operator and the resulting population is the set of new solutions (second generation). This set now undergoes the whole procedure as the first set and so on. The

[3] In principle each of the integers should be signed with a '+' or '-' indicating the state of the switches. In the demonstration we omit the signs for reasons of clarity.

termination of the above described procedure is done once solutions present good enough fitness value. The process of convergence to the best solution depends on many parameters especially on the size of the problem (string length) and how the parameters of the algorithm, such as the population size and mutation rate, are chosen. The same problem can be represented in many ways and it is up to the implementers to find the most suitable parameters. Other GA approaches to the problem of restoration can be found e.g. in [Bretas 1998, Kumar 2006].

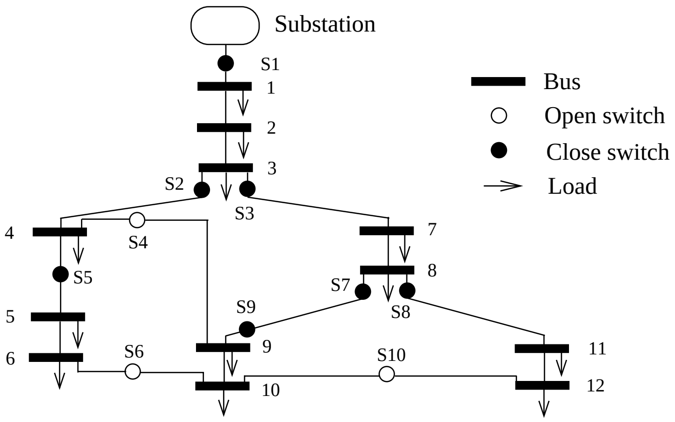


Figure 3: Network configuration of a power distribution system

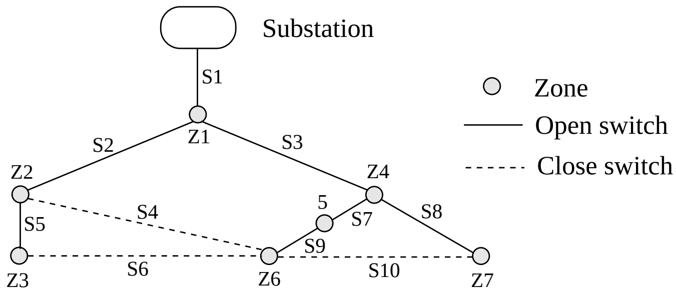


Figure 4: Graph representation of the power distribution network presented in Figure 3. The nodes of the graph represent the zones (areas between switches) and the edges represent the switches

2.4 Other applications of evolutionary methods

As previously mentioned many of the power system strategies of operation are derived by using evolutionary methods. A survey by [Nara1997] on GA application of power systems identifies the following problem areas to which GA is applied:

- **Distribution planning and operation**
- **Reactive power planning**
- **Economic Dispatch**
- **Generation and transformation planning**
- **unit commitment and generator maintenance scheduling**

Application for communication systems include:

- **routing algorithms for fixed communication network** [Wedde 2006]

- **location area management** in cellular networks [Karaoglu 2005]

and further solutions for optimization problems in wireless networks can be found e.g. in [Das 2004].

An extension of the genetic algorithm class is the genetic programming. This evolutionary method has also several interesting application. In short, **Genetic programming** (GP) is a genetic algorithm in which each individual of the population is a computer program and the objective of the optimization is the 'discovery' or creation of a new program. It was first introduced by J. R. Koza [Koza 1992] mainly with the aim of creating programs which can program. GP has some surprising application, many of then are described in a survey by [Willis 1997]. Among the most interesting examples are:

- application of system modelling in chemical processes such as the **generation of non-linear dynamic models of biotechnological batch and fed-batch fermentation** [Bettenhausen 1995].
- Application in control area such as development of **vehicle control system** [Hampo 1992].
- Applications in **optimization and scheduling such as Traffic signal timing control** [Montana 1996].
- Applications for design e.g. **design of electrical circuits** [Koza 1996].

For further details on the various evolutionary algorithms we refer the reader to the Hand book of Evolutionary Computation by [DeJong 1997].

3. Swarm Intelligence

Swarm is a large number of insects or other small organisms, especially when in motion. Bees, ants and bacteria are usually found in large groups of individuals, called swarms. This term is also commonly used to describe bird flocks and fish schooling.

Swarm intelligence is a class of algorithms inspired by swarm behaviour; it refers to a class of artificial intelligence techniques inspired by the study of collective behaviour in decentralised, self-organised, systems. In general the attractiveness of swarm intelligence driven methods comes from the relative simplicity of the insects/animals which succeed in accomplishing complex tasks in a decentralized manner, based mostly on local interactions.

3.1 Ant as social insects

Ants are insects of the family Formicidae which is believed to have evolved around 120 to 170 million years ago from the vespoid wasps. There are around 12,000 different species of ants that inhabit almost every part of land of our planet excluding the most extreme cold places such as Antarctica and Iceland. Ants are considered to be one of nature's most successful insects due to their highly organized colonies with population size ranging from several hundreds to sometimes million of individuals. Prominent characteristics of ants colonies include a decentralized structure and a distributed task allocation mechanism which allow the colony to accomplish the needed operations for its survival without 'being told' what to do. Among those tasks are:

- building and maintenance of the nest,
- caring for brood (ants in first stages of life),
- finding food and carrying it to the nest,
- and protecting the nest from invaders and other dangers.

In most cases, tasks are preformed in a decentralized manner i.e. there is no one specific ant which orders or directs the other ants. All operations are believed to be invoked by local interaction. The complexity of the above mentioned tasks is often quite demanding e.g. broods are highly delicate and need to be kept in very specific conditions and the operation of the ants ensures regulation of the nest temperature within limits of 1 degree Celsius. Ants developed quite amazing capabilities also in finding food e.g. they are able to create chains from their own bodies allowing them to cross water streams and they can form bridges in a similar way to overcome obstacles. Yet there is no one ant which is important for the accomplishments of the task, in general a single ant is unable of doing any of the operations performed by the colony and usually the ants are not trained for a specific assignment. Rather all ants can perform the task of the colony. The number of ants participating in a specific task depends on the nature of the task and the general conditions the colony is subjected to; if a big food source is located, just a few ants will be occupied in nest maintenance and more will be participating in the transport of food to the nest. This subject of task allocation in ant colonies is discussed in [Gordon 2001], and further observations on the collective behaviour of ants are reported in [Hölldobler 1994, Frank 1989]. In summary, ants can be regarded as a collective of relatively simple autonomous elements which are able to achieve complicated assignments in a decentralized manner. The ant colony is robust to perturbation which influences the colony population size by operating with an efficient task allocation method. It is robust to changes (to some extent) in the environmental conditions (changes in temperature, changes in food availability, changes in ground conditions) and therefore can be named adaptive. The mechanisms producing those properties, if well understood, might be helpful in the design of other networks composed of many autonomous relatively simple elements. One of the interesting behaviors of ants which is well understood and explained is their ability to track the shortest path on the way to their food. The ant colony optimization is a powerful algorithm that was developed with the inspiration of this behaviour. We present this algorithm in the following section.

3.2 Ant colony optimization

Ant colony optimization or ACO is a special class of ant strategies which was first introduced by M. Dorigo and colleagues as a metaheuristic algorithm for solving hard combinatorial optimization problems [Dorigo 1991]. Derivation of this algorithm was based on the observation of ants while performing one of their main tasks - **foraging**. Foraging is the task of exploring the territory around the nest in search of food sources and then transporting the food back to the nest. Certain species of ants, the Argentine ants, were found to leave a trail of **pheromone** on their way to and back from a food source [VanVorhis 1986]. Pheromone is a chemical substance used by ants as a form of communication. In the specific task of foraging, the argentine ants use this substance to mark the route leading to the food. The ants can control the type and intensity of the pheromone they produce, depending on the quantity and quality of the detected food source. The pheromones are therefore a form of signs or messages left by one ant to be read by another. This form of communication in which social insects communicate indirectly through the environment is called **Stigmergy** [Grassé 1959]. This type of indirect communication enables the ants to find the shortest path between their nest and the food source as was demonstrated in an experiment preformed by [Goss 1989].

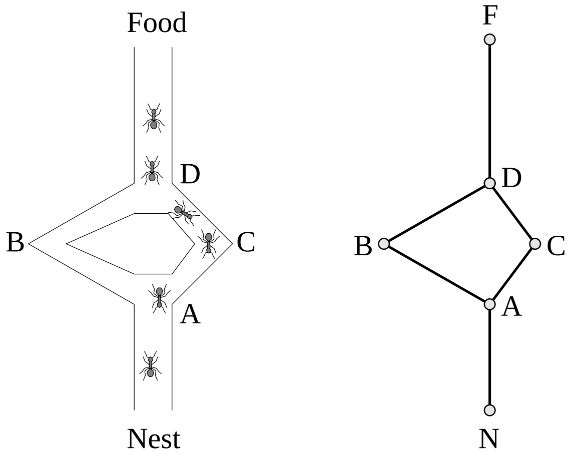


Figure 5: Setup of the shortest path experiment performed by [Goss 1989] and the corresponding graph representation

In this experiment depicted in Figure 5, ants food was deposited in some distant location from the ants net at point **F**. To reach the food ants could choose one of two ways passing through point **B** or point **C**. A few minutes after the foraging starts most ants are seen to be concentrated on one of the routes leading to the food, the shortest one. Goss et al explain this result in the following way. The ants need to make a choice once on the way to the food (at point **A)** and once on the way back to the nest (at point **D**). Assuming ants are more likely to choose the path with the higher pheromone concentration, one would expect that in the beginning of the foraging when none of the routes is yet signed, that both routes will be chosen with the same probability. Once path is chosen the ants move towards the food leaving a pheromone trail as they advance. Ants which chose the shortest path will arrive to the food first and until the moment ants start to head back the amount of pheromone on the shortest path will be higher then on the longer path. Thus, when the first ant supplied with food will arrive from point **F** to **D,** it will most probably choose the shortest path (going through point **C**) increasing the pheromone level even more thus increasing the probability of the other ants to come back in this way. Increasing the relative difference in the path length should increase relative time difference for arriving to the food thus one expects more ants will choose the shortest path. Both assumptions, impartial choice of route in the beginning of the foraging and correlation between number of ants on the shortest path and the relative difference in the paths length were validated in the experiment. Summarizing the experiment we learn that

- Ant leave pheromone trail on the way **to** and **from** food source
- Ant prefer marked paths i.e. paths which other ants took before
- Given a choice between several marked paths the ants tend to prefer the path where concentration of pheromone is the highest.

These are the basic principles which inspired Dorigo et al in the development of the ant colony optimization algorithm. The "translation" of the ant's solution to the shortest path, into an algorithm for finding solutions of a given (different) problem is now straightforward. First one needs to map the problem in the form a graph *G(N,E)*. It is important to notice that the graph *does not* represent the solution space rather it is a way of representing the problem. Turning back to the experiment performed by Goss, a reasonable graph representation of the problem would be to present points N,A,B,C,D and F as the nodes of the graph and the trail segments connecting those node as the graph's edges (Figure 5). The solution is an ordered set of nodes e.g. {N,A,C,F} or {N,A,B,F}. Ant agents are constructing solutions by moving in the problem space and in the process leaving a trail of artificial pheromone. The principle parameters of the algorithm are

1. τ_{ij} - the amount of pheromone the ant leaves on the path *ij*,
2. $\Delta\tau_{ij}$ - the amount of pheromone decreased from trail *ij* in each time step of the solution.

The first parameter is usually referred to as **pheromone trail** and the second as **evaporation.** The latter is also driven from reality as pheromone trails tend to evaporate with time. The decision of how to set the values of the parameters depends on the specific problem. In more progressive versions of this algorithm some other parameters are added. This is discussed in detail in [Maniezzo 2004].

Bacteria Chemotaxis

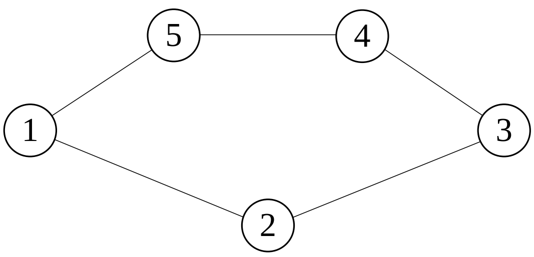
Before presenting a concrete example of an ACO application, we would like to mention one other mechanism for detecting food sources used by a different biological example of swarms, the bacteria. Similarly to ants, bacteria trace nutrient sources in a process called **Chemotaxis.** By chemotaxis, bacteria find not only food, but in general, this is the process by which motile bacteria sense changes in their chemical environment and move to more favourable conditions [Rao 2004]. In air and fluids, concentration of chemicals, e.g. the odour of some nutrient, are sporadic and so the bacteria cannot follow a continuous trail leading to the source, rather "the searcher detects odour in a sporadic sequence of distinct events arising from its encounters with patches of fluid (or air)" [Vergassola 2007]. Vergassola et al. suggest a chemotactic-search algorithm they call 'infotaxis' where information plays the role of chemical concentration and the searcher decides on its advancement based on sporadic cues and partial information. According to the authors the proposed search mechanism can be applied in the context of searching with sparse information.

3.3 Routing algorithms for telecommunication systems

Being based on shortest-path finding principles, the ACO is more intuitively implemented for routing problems[4]. Routing is an important problem in information network such as the Internet fixed and ad hoc telecommunication networks. In the following we present an operation strategy for load balancing in switch circuit networks. These are relatively simple networks and thus are more convenient for demonstrating the implementation of the ACO, further complication of the example presented below and implementation for other type of communication networks can be found in the comprehensive review of bio-inspired routing algorithms for fixed telecommunication network by [Wedde 2006].

Load balancing in fixed telecommunication network

Circuit switched networks are a part of the physical media over which a call is communicated. They are composed of switches and communication lines. Figure 6 depicts a schematic representation of such network. Each switch (circles in the figure) has a fixed number of circuits which defines the number of calls that can be initiated or pass though it at a given moment in time. Initiating a call means connecting a circuit in the call source-switch to a circuit in the call destination-switch via the network. E.g. in the network of Figure 6, a call from node 1 to node 3 can be directed either through node 2 or through nodes 4 and 5. If all the circuits of a node are used, the node is congested and any call that needs to start, end or pass in this node will be dropped (not initiated). The load balancing problem is therefore a routing problem in which the goal is to route calls such that no call will be dropped or if necessary only minimal number of calls will be dropped. An ACO based solution to this problem was first introduced by [Shoonderwoerd 1996] by the name *ant based control* model (ABC). In this model calls are generated by a standard call generator such that in each time step some calls are initiated with random source, destinations and duration. Calls are routed according to the standard deterministic routing table procedure[5].



T3	2	4
1	P_{12}	P_{14}
2	P_{22}	P_{24}
4	P_{42}	P_{44}
5	P_{52}	P_{54}

Figure 6: Diagram of a circuit-switched communication network with 5 switches (the circles). The table on the right is the routing table of switch 3. Rows represent the possible destinations and columns represent nearest neighbours

[4] Routing is maybe the more intuitive implementation but there are many other problems ACO can be applied to very efficiently. A list of implementation is given in the end of this chapter.

[5] To initiate a call from node *i* to destination *j* the following steps are made: in routing table of node *i* find the element of row *j* with the highest value, set path from *i* to the node corresponding to the column of the element (suppose it is *k*) path=$i \rightarrow k$. Now look in routing table of node *k*, choose the element of row *j* with the highest value, set path from *k* to the node corresponding to the column of the element (say it is *m*). Now path=$i \rightarrow k \rightarrow m$. Repeat this until arriving to *j*.

A routing table is located on each switch. An example of such table is shown in Figure 6. The elements of each table are continuously updated by the ants passing through them leaving a trail of pheromones. The ants are not to be confused with the calls; they are separate entities and circulate in the network regardless of the calls[6]. Ants are generated randomly on the network and the node in which they were generated in is their source node. Each ant also has a destination node it will try to reach; also the destination node is randomly generated. To reach the destination node, ants make a probabilistic choice between the possible paths where the probabilities are given in the routing table. E.g. an ant generated in node 3 with destination node 1 can choose to go through node 2 or node 4. The probability to go through node 2 is given by T_{12}^{3} which is the element in the first row (destination node 1) in the second column (neighbour node 2) of table routing of node 3. Similarly the probability to go through node 4 is given by T_{14}^{3}.

Now we can see how the ants also change the routing tables. When the ant makes a move from node *i* to a neighbouring node *k*, arriving to node *k* it will update its routing table by adding some amount of pheromone to the element T_{si}^{k} of the table, where *s* is the ant's source node and *i* is the node it just came from. E.g. suppose an ant that was launched from node 1 (Figure 6) towards node 4, has just done the move from node 2 to node 3 (ants source node and destination are randomly chosen). Arriving to node 3 the ant will update the table element T_{12}^{3} of this node by increasing the value of P_{12} in ΔP [7]. Since elements of the tables are probabilities, each time an element in row *j* is updated all elements of this row are changed according to the simple rule:

- $P_{jx} = \dfrac{P_{jx} + \Delta P}{1 + \Delta P}$ *x* is the node the ant came from and P_{jx} is the element it updated
- $P_{jl} = \dfrac{P_{jl}}{1 + \Delta P}$ *l* is any element in row *j* such that $l \neq x$

Basically the ants are leaving information regarding the route they have used (the pheromone trail). Now it is important to decide what would be ΔP. For this the authors introduce, **aging** and **delaying**. Supposing an ant has an initial value of pheromone equal to ΔP. Each time the ant moves from one node to another its age is increased in one unit and ΔP decreases. Therefore the aging of the ant makes the ant less influential. If the path an ant takes is too long the increase of elements in the table will be insignificant (this can be calibrated by choosing the way ΔP is decreased). This however gives preferences for choosing shortest path which is not necessarily the good for balancing the loads but by introducing the delay ants who reach congested node are detained in this node an amount of time proportional to the congestion level. During this time they continue to age. The combination of aging and delay achieves the favouring of short but un-congested routes. For more details regarding the adjustment of the parameters of the model the reader is referred to the original work [Shoonderwoerd 1996].

3.4 Other application of ACO

The S-ACO was the first in a serious of algorithm that followed in the proceeding years. "Next generations" of ACO proved to be powerful metaheuristics and their efficiency was proved both

[6] Ants are independent of the calls but the congestion level of the node does influence them as will be seen later in the text.

[7] Here ΔP has a similar function to the pheromone trail τ_{ij} in the S-ACO of Dorigo with the difference that in the later pheromone trail are attributed of the edges of the graph while in this model they are associated with the routing table elements.

on theoretical and experimental bases. The ACO, in its advanced forms, is applicable to both discrete and continuous problems. Proof of convergence was first given by [Gutjahr 2000] for a variant of the ACO called "graph-based" ant system. later on he developed this argument, obtaining the improvement of all previous results and proving a property of the same strength of the tightest one so far obtained in the whole metaheuristic area for simulated annealing [Hajek 1988]. Theory, models and proof of convergence of ACO algorithms are extensively discussed in [Dorigo 2005].

Applications to communication networks are mainly in the development of different **routing strategies.** A comprehensive survey of ant based routing strategies is reported in [Wedde 2006].

As opposed to evolutionary methods, the ants algorithm being somewhat a newer method, are not yet very practiced in the field of power systems. However there are some recent examples for the implementation of ant based methods for the **power system restoration problem** [Chin 2005, Ling 2005].

Industry-oriented applications are reported by [Krishnaiyer 2002] who reviewed 118 papers for identifying typical applications of ant algorithms and then listed and classified them by the algorithm name and type of problem it solves. Among others, the list includes:

- job scheduling
- flow manufacturing
- pickup and delivery
- structural design problems

Other application areas are reported in [Dorigo 2004] and [Maniezzo 2004] they include:

- sequential ordering
- vehicle routing
- quadratic assignment
- plan merging
- driver scheduling
- graph colouring
- dynamic multiple criteria balancing

3.5 Particle Swarm optimization algorithm

Particle swarm optimization (PSO) was introduced by [Kennedy and Eberhart 1995]. It is a simple stochastic optimisation technique inspired by models of social behaviour of fish schooling or bird flocking, which turned out (after some modifications) to be a powerful optimization tool. Different variants of PSO were proposed, but the most standard one assumes a swarm of particles representing the potential solutions. Particles are "flying" in the solution space and the coordinates x_i of each particle, represent one solution. The starting positions are initialised randomly. The change in the position of particle i is described by the vector v_i. The size of vectors x_i and v_i is equal to the dimension (number of variables) of the optimization problem. Each particle updates its position in discrete time using the best encountered position based on its own exploration L^i_{Best}, best swarm overall encountered position G^i_{Best} and the previous value of vector v_i:

$$v_i^{k+1} = wv_i^k + c_1 r_1 \left(L_{Best}^i - x_i^k \right) + c_2 r_2 \left(G_{Best} - x_i^k \right) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

where

c_1 and c_2 are two positive constants;

r_1 and r_2 are two random numbers drawn from range of [0,1];

w is the inertia weight, which is usually continuously decaying with the number of **time** steps.

The quality of each position in the solution space can be evaluated by a fitness function. L_{Best}^i is the position (x_i) which corresponds to the best solution ever encountered by particle *i*. G_{Best} is best encountered position of all particles.

The searching process continues until some constraint is met e.g., on maximum number of time steps, on number of time steps without improving the global optimum etc. To avoid situations in which solution advance towards an unfeasible position one can: shorten the length of vector v_i, use the projection method or by adding to the fitness function an additional term weighting the unfeasible solutions by large costs (as done often in GAs – see Sec 2.3).

In the recent years several PSO have been applied in various research and application areas and it was demonstrated that PSO gets often better results in a faster and cheaper way compared with other methods. An overview of PSO applications to power system operations is given in [Alrashidi 2006]. According to this review particle swarm optimization has been applied to

- reactive power control and power losses reduction,
- optimal power flow,
- power system controller design,
- and power transformer protection.

4. Artificial immune system (AIS)

4.1 The human immune system

The immune system (IS) is another example of a highly complex system which evolved for protecting the body. As such it is a good candidate for being one of the "bio-systems" from which we could learn on protection mechanisms. In this section we outline in general terms some of the mechanisms of the IS that are now understood and could, in principle, be applied to other non – biological systems. This section is based on the "Interpretive introduction to the immune system" by [Hofmeyr 2001].

The IS can be viewed as the aggregation of mechanisms whose purpose is to maintain homeostasis. As a part of this mission the immune system needs to identify the existence of harmful organisms within the body the **non-self,** as apposed to the non-harmful organisms within the body termed **self**. Once non-self is identified, the IS can attack it and try to destroy it. The ability of the immune system to distinguish between self and non-self is the basic principle of the IS detection mechanism and as we shall see in the following section, it is the basic concept used in the modeling of artificial immune systems. Non-self can be **pathogens, toxins** and in some cases also the body's own cells. The Pathogens are micro-organisms which can cause sickness. Bacteria, parasites, viruses, and fungi are some of the more common pathogens which our body is exposed to. Each of the aforementioned types of non-self is dealt with by different IS mechanisms. In this chapter we relate mostly to the reaction of the IS to pathogens.

<u>Pathogens penetration to the body</u>

The first 'mission' of the Pathogen in its attack on the body is to enter the body[8]. To protect itself form the invaders the body is supplied with several defense layers. The first elementary body protection is the skin which is thick enough to prevent entry of most pathogens and, in addition, it produces certain substances which are harmful to them. Therefore the skin is quite efficient in preventing penetrations. Most common way then for pathogens entry is in the occasion of rapture in the defense, e.g. via cuts, or by using 'back doors' e.g. food is a common transporter of infection sources. Once overcoming the first barrier, the pathogen needs to cope with some physiological conditions such as the pH level and the body's temperature which are in many cases lethal. Pathogens who survive are now free to 'stroll' around the body and to infect its cells i.e. until they are confronted with two other defense mechanisms the **innate immune** and the **adaptive immune** systems.

<u>The innate immune system</u>

The first response of the body to an invasion is usually in the scale of a few hours and is initiated by the innate immune system. This system is composed of cells, molecules and proteins which can find and destroy invaders. Both detection and elimination depend on chemical bonding. The immune cells have on their surface receptors which can chemically react (bind) with **antigens. Antigens** are molecules present on the surface of invaders. The operation of binding is in fact the recognition of non-self. One of the main differences between the innate immune and the adaptive immune which will be discussed later on, is related to the type of receptors. Especially, receptors of innate immune cells are 'fixed' i.e. are generated with a certain "storage" of receptors of the innate immune cells and these do not change in time. The innate system is therefore regarded as a static system as apposed to the adaptive immune system as we shall see later on. Three main systems compose the innate immune response the **complement, endocytic** and the **phagocytic** systems.

[8] Diseases can be cause also on the surface of the body e.g. skin diseases but this is less relevant for our discussion.

The complement system is a group of proteins which circulate in the plasma and they are more probable to be the first who encounter and react to the presence of invaders. These proteins have the ability to detect antigens and bind with them. By binding the complement are activated. The activation process means a sequence of reactions which progresses the process of detection and destruction. These reactions are:

- killing invaders
- inflammation
- marking of invaders
- activation of phagocytes

Complement can kill pathogens after they bind to them; in the process called **Lysis,** they disrupt the bacteria's membrane, thus destroying it. In addition they set an alarm announcing the detection of invaders in the form of **Inflammation.** Inflammation is the increase of blood flow to the infected area allowing different type of immune cells (including the adaptive immune cells) to arrive faster and identify the existence of the foreign bodies. When help arrives e.g. in the form of phagocytes, the complement who marked the invaders make it easier for the arriving forces to find and attack the invaders. The **Phagocytes** are cells specialized in finding and 'eating' harmful invaders. Three main types of phagocytes are: **granulocyte**, **dendritic** cells and **macrophages,** they differ in the speed of reaction and specialize in different kinds of infections. The granulocytes are very fast and attack in big numbers, while the macrophages are slower but bigger. Macrophages recognize pathogens but they can also recognize complements; thus a complement covering a pathogen makes it easier on the macrophage to find it.

<u>The adaptive immune system</u>

The adaptive immune system (ADIS) is probably the most fascinating part of the immune system. As its name implies, it is an adaptive system with the ability to learn and maintain a memory of past events for improving reaction in the future. The operation of the ADIS is based on a primary and secondary response. The primary response is the process of first recognition that operates when an invader enters the system for the first time. This process becomes apparent some days after the beginning of the intrusion and is completed with the extraction of the pathogen some weeks after. The secondary response relays on previous events. If the same invader enters the body a second time the ADIS recognizes it almost immediately (before it has a chance to re-infect the body) and destroys it. The principle of action of both the primary and the secondary response are similar to those presented in the innate system. These principle operations are recognition, activation and elimination. However, while the innate is a static system, the adaptive immune is a dynamic system with learning and memory capabilities.

Recognition and activation– the ADIS consists of **lymphocytes**, which are white blood cells acting as detectors of pathogens. Each lymphocyte has on the order of 10^5 receptors on its surface. These receptors can bind with **epitopes**, which are locations in the surface of a pathogen. Receptors and epitopes are electrically charged three dimensional structures and their bonding depends on the level of affinity between their corresponding structures. There are two important points in the operation of a receptor which is worth recalling. First, lymphocytes are different one from the other by the type of receptors they have, while receptors of the same lymphocyte are identical, i.e. they are able to bind only with a small subset of possible epitope with which they have high enough affinity; thus different lymphocytes are needed for the identification of different pathogens. This is an efficient method for the identification not only of an intruder but also of the specific type of the intruder thus allowing also a specific and more efficient reaction. Secondly, to activate a lymphocyte it is not enough to have one match; rather there is a certain threshold for the number of bindings that should occur. Having a threshold makes the identification more accurate since the higher the affinity between the receptors and the epitopes, the higher the number of bindings. The ability of the body to detect the intruders is therefore highly dependent on the amount of different receptors present. To ensure the variety of the receptors the body uses the recombination mechanism which is basically the crossover operation discussed in chapter 1 in the GA description.

Adaptation and extermination – One class of lymphocytes responsible for recognition is called the B-cells. The name is derived from the fact that these cells mature in the brain marrow. B-cells are able to 'learn' the right form of receptors needed for a specific case of intrusion and then to adapt their form accordingly. This process runs as follows. B-cells circulate in the body until they find an intruder to which they can bind. Once a B-cell is activated (according to the process described above) it migrates to one of the hundreds lymph nodes which are present in the body. There it mutates in a very high rate producing new B-cell offspring. The new 'born' cells, which can be slightly different then the original clone because of the mutation procedure, will try to bind to epitopes of pathogens captured in the lymph. Those which will succeed in binding can leave the lymph node and proceed to the next stage of differentiation. The unsuccessful cells will die after a short time. This process of proliferation allows the system to adapt it self to the specific type of new pathogen against which the body should be protected. A B-cell which has undertaken this process is said to be mature. Mature cells leave the lymph node and differentiate into **plasma B-cells** and **memory B-cells**. Plasma B-cells are able to create **antibodies** which bind to the pathogen's and destroy it. Memory B-cells are those cells which have developed receptors with very high affinity to that of the invaders'; these cells are the ones to initiate the secondary response i.e. in case an invader enters the body a second time they will be immediately recognized by the memory cells and elimination process will be fast and efficient.

The above description is a simplification of a much more complex and detailed set of processes; however it is an attempt to capture the essential mechanisms which are used in the modeling of the artificial immune system. One important point which deserves a further explanation concerns with the ability of the immune cells to differentiate self and non-self i.e. the mechanism by which the binding of the immune cell with self is avoided. This is discussed in the following.

Tolerance of self - Self cells are protected from the innate immune cells by having regulatory proteins on their surfaces which prevent binding; this is possible because of the 'known in advance' form of the innate cells. However, there is no guarantee that mutated B-cells will not match one of the self cells. The solution to this lies within the activity of another type of lymphocyte cells of the ADIS, the **T-cells**. These cells are needed for the activation of the B-cells and are the 'second condition' previously mentioned (apart from the thresholds of number of binding epitopes). The name of T-cells derives from the fact that they mature in the thymus. A T-cell has three possible life stages:

- immature stage – when T-cells learn to recognize non-self
- mature stage – the adult cells circulate in the body until they encounter non-self, or until they die
- memory stage – T-cells who were activated by binding to an invader become memory cells and have unlimited life time.

The process of maturation of a T-cell is the first time period of its life in the thymus. The thymus is the 'classroom' of the T-cells, being a location in the body where most self epitopes are expressed. T-cells, by their nature, try to bind to the different epitopes they encounter in the thymus. If successful in binding, they will be destroyed. This is because they have a large probability to bind with self. Those T-cells which pass the immature stage without binding become mature and leave the thymus to 'wonder around'. Since they have learnt to differentiate self and non-self they can assist the B-cells in recognizing an invader. If they encounter a B-cell which is bound they are able, by a complex signaling procedure, to inform the B-cell whether it is bound to a self or to a non-self. Only in the case they identify the non-self, the second condition for the activation of the B-cell is fulfilled and the B-cells are activated. Otherwise the B-cell, who wrongly binds to self, will die.

Summary

In conclusion of this section we would like to stress some of the more important aspects of the immune systems as was described above:

- Layer protection: The immune system is composed of several layers of protection different in their level of complexity and efficiency.
- Multiple escalating reactions – the first reaction to invasion is fast but not 100% efficient, thereafter it is followed by a more precise but slow response.
- Distributed control – immune cells operate independently of each other and of any other central entity.
- Adaptive mechanism – parts of the immune system are capable of learning and they can adapt their form and reaction depending on the type of attack the body is subjected to. Moreover they can remember what they have learnt and reply with better efficiency in future scenarios.

4.2 Artificial Immune Systems

A model for the protection of a local-area broadcast network [Hofmeyr 1999]

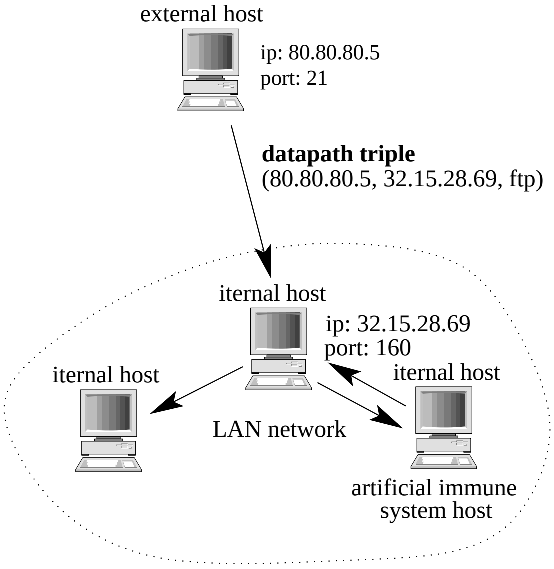
The basic feature of the artificial immune system is the self/non-self detection mechanism derived from its counterpart in the living beings immune system. In this model the authors define self as a set of normal pair-wise connections between computers at the TCP-IP level. Such connection can be internal, between two computers in the LAN or external between computers in and outside the LAN (see

Figure 7). Each connection is defined by a unique single-bits string. The string is the "data-path-triple" representation of the connection i.e. the source IP address, the destination IP address and the service or port of communication. Non-self is also a set of connections represented by a single-bits string of the same length as that of the self. The difference being that non-self consists of those connections, which are not normally observed on the LAN. The proposed model is based on two operations: detectors creation and destruction and detection of non-self as explained in the rest of this section.

Detectors creation and destruction

In this model the complex ADIS is simplified by the merging of B-cells and T-cells into one entity called detector. Detectors are continuously created and destroyed in fixed locations of the network e.g. in some pre-determined hosts. A detector (a self string itself), similarly to the T-cells, can be in one of three stages, immature, mature and memory. Not all stages are necessarily reached by all detectors. Firstly a detector is created as a random sequence of bits and enters a tolerance period in which it is defined as an immature detector. If during this period it encounters a matching string (matching is explained later on), it is destroyed and a new detector is generated instead. This is in analogy to the tolerance period the T-cells go through in the thymus. In both cases the purpose is to create a set of detectors (T-cells or strings) which will be able to distinguish between self and non self. The tolerance period needs to be long enough to allow the detector to see all, or at least most, of the self; thus in the end of this period detectors which did not match any string are most likely to be matched only with non-self. After the tolerance period the surviving detectors change status to 'mature'. This status is also time limited. If during the 'maturity' stage a detector encounters a certain amount of matching strings, set by some fixed threshold, it will change status at the end of its lifetime to 'memory' otherwise it is destroyed and replaced by a new generated detector. The memory detectors have an unlimited lifetime and, like their counterparts memory B-cells, have a lower activation threshold. This enables the secondary reaction – next time an attack or fault of the same form will be present in the LAN the memory detectors can recognize it almost immediately.

Figure 7: Schematic view of artificial immune system in LAN. Hosts in the LAN can communicate between them selves and with external hosts. The data triple is composed of the sender IP, receiver IP and the service identifier. The AIS located in one of the hosts observes the traffic in the network and searches for suspicious datapath triple.



Detection of non-self

The result of an encounter of a detector with another string can be one of two options, matched or mismatched. This is determined by comparing the bits of the two strings. If more then r bits are equal then the strings match, otherwise they do not. r is an integer parameter of the problem. The action taken after detection is made depends on the life-stage of the detector. For an immature detector encounter with a matched string results in the destruction of the detector (as a part of the tolerance stage) while an encounter with a mismatched string results in no action. For both mature and memory detectors, encounter with matched increases a counter and encounter with mismatched invokes no action. Each detector is supplied with its own counter. Activation of the detector is performed if the threshold defined for it is reached. For mature detectors the threshold is set to some integer t_{ma}>1 and for memory detectors the threshold is 1. The thresholds, similarly to the ones which exist for the activation of the B-cells, distinguish between mature and memory detectors. The mature detectors will identify new type of non-self supplying the preliminary response and memory detectors will identify repetitive attacks (or fault) – secondary response. We recall though that immune cells are usually activated only after two conditions are met, the first is the threshold and the second is the co-stimulation. E.g. the co-stimulation of the B-cells aimed to avoid binding with self. Since in this model there is only one type of detectors the auto-immune problem needs to be dealt differently. To confront this problem the authors suggested the following mechanism, once a mature counter threshold is reached a message is sent to a higher authority e.g. the system administrator. He in turn needs to confirm the detection. If no confirmation arrives the detector is assumed to have made a false recognition and is destroyed, otherwise, all detectors who received confirmation are compared. The winners (those with the highest match) become memory cells. This strategy gives the role of the T-cells to the system operator. The competition performed between the approved matching strings increases the accuracy of detection in analogy to the mutated B-cells which if not well matched – die. The subject of reaction is not dealt in this model.

4.3 Applications of AIS

The model presented in the previous section was developed by the Adaptive Computation Group at the University of New Mexico. It seems that much of the work done in the field of AIS should be credited to this group. Forrest and Hofmeyr, who belong to the aforementioned group, review some of the basic principles of AIS as well as some of their works in "an introductory to immunology as information processing" [Forrest 2001]. Problems related to computers security they have addressed include **virus detection** [D'haeseleer 1996], **host-based intrusion detection** [Forrest 1997], **automated response** [Somayaji 2000], and **network intrusion detection** [Hofmeyr 2000]. References to other works can be found in a review on AIS by [Timmis 2004] who describes several AIS based models applied to different problems from the fields of **data mining**, **robotics** and **computer security**. A review on immune system approaches to **intrusion detection** is given in [Aickelin 2004]. In this review Aickelin et al compare ten proposed immune-based algorithms. According to their survey all algorithms are based on the immune self/non-self recognition model combined with negative selection similarly to the model described above. Only one of the reviewed algorithms includes response [Kephart 1994]. Recently it has been suggested to implement methods from AIS also to **ad hoc wireless networks misbehavior detection** [Sarfijanović 2005]. In [Drozda 2006] the authors summarize some of the more common misbehaviors in ad hoc wireless networks (AHWN) and review some AIS based approaches that might be applicable in this regard. The same authors suggest a basic construction for an AHWN [Drozda 2005] specifying the basic modules they think are needed for an efficient detection.

Conclusions

The purpose of this document was to overview bio-inspired strategies and their applications in the domain of Technological Infrastructures. The introduced topics, Evolutionary Algorithms, Swarm Intelligence and Artificial Immune Systems are new approaches derived based on the observation of biological systems' behaviour and development. Though some of the suggested techniques, especially GA, were implemented in practice for the solution of problems in real systems, there is still much to be done both theoretically and in practice and the interest in Bio-inspired research seems to be increasingly growing – in parallel with the successful implementation it yields. In our future work (mainly D.2.2.7) we will search for possible applications of the ideas and principles inspired by biological systems, as presented in this document, and we shall strive to develop techniques and algorithm that can be implemented to Interdependencies related problems.

Acknowledgments

The authors would like to thank Andreas Deutsch, Uri Hershberg and Vittorio Rosato for fruitful discussions, remarks and suggestions.

References

[Aickelin 2004] U. Aickelin, J. Greensmith, J. Twycross: Immune system approaches to intrusion detection - a review. In G. Nicosia et al., ed.: Proc. of the Third International Conference on Artificial Immune Systems. Number 3239 in Lecture Notes in Computer Science, Springer, 316-329

[Alrashidi 2006] M.R. Alrashidi, M.E. El-Hawary: A survey of Particle Swarm Optimization Applications in Power System Operation, Elec. Pow. Comp. Sys., 34, 1349,1357

[Bettenhausen 1995] K.D. Bettenhausen, P. Marenbach: Self-organizing modelling of biotechnological batch and fed-batch fermentations, Proc. EUROSIM'95, 1995

[Bretas 1998] N.G. Bretas, A.C.B. Delbem, A. de Carvalho: Optimal energy restoration for general distribution systems bygenetic algorithms, POWERCON '98, China, 1, 43-47

[Chin 2005] H-C. Chin, Y-S. Su: Application of the ant-based network for power system restoration, Trans. and Dist. Conf. & exhi., Asia and Pacific, Dalian, china

[Das 2004] S. K. Das, N. Banerjee, A. Roy: Solving optimization Problems in Wireless networks using Genetic Algorithms. In: Handbook of Bio-inspired Algorithms

[D'haeseleer 1996] P. D'haeseleer, S. Forrest, P. Helman: An Immunological Approach to Change Detection: Algorithms, Analysis, and Implementations, Proc. of the 1996 IEEE Symposium on computer Security and Privacy, Los Alamitos, CA

[DeJong 1997] K. De Jong. D. David, B. Fogel, H-P. Schwefel: A history of evolutionary computation. In: T. Back, DB. Fogel, Z. Michalewicz (eds), Handbook of Evolutionary computation. Oxford University Press, New York, and Institute of Physics Publishing, Bristol. A2.3:1-12

[Dorigo 1991] M. Dorigo, V. Maniezzo, A. Colorni:Positive feedback as a search strategy,Technical report 91-016, Dipartamento di Elettronica, Politecnico di Milano

[Dorigo 2004] M. Dorigo, T. Stützle: Ant Colony Optimization, MIT Press, Cambridge, MA

[Dorigo 2005] M. Dorigo, C. Blum: Ant colony optimization theory: A survey, Theo. Comp. Sci., 344, 243-278

[Drozda 2005] M. Drozda, H. Szczerbicka, T. Bessey, M. Becker, R. Barton: Approaching Ad Hoc Wireless Networks with Autonomic Computing: A Misbehaviour Perspective, Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'05), Philadelphia, PA, 723-733

[Drozda 2006] M. Drozda, H. Szczerbicka: Artificial Immune Systems: Survey and Applications in Ad Hoc Wireless Networks, Proc. International

Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS'06, 485-492

[Forrest 1997] S. Forrest, S. Hofmeyr, A. Somayaji: Computer Immunology, Comm, ACM, 40(10), 88-96

[Forrest 2001] S. Forrest, S.A. Hofmeyr: Immunology as Information Processing. In: L.A. Segal, I. R. Cohen, Design Principles for the Immune System and Other Distributed Autonomous Systems, Oxford University Press, 361-388

[Frank 1989] N.R. Frank: Army Ants: A collective Intelligence. In American Scientist, 77

[Goldberg 1989] D. E. Goldberg: Genetic algorithms in Search, Optimization and Machine learning, Wesley

[Gordon 2001] D.M. Gordon: Task Allocation in Ant Colonies. In: L.A. Segal, I. R. Cohen, Design Principles for the Immune System and Other Distributed Autonomous Systems, Oxford University Press, 293-299

[Goss1989] S. Goss, S. Aron, J.L. Deneubourg, J.M. Pasteels. Self-organized shortcuts in the Argentine ant. Naturewissenschaften, 76:579-581, 1989

[Grasse 1959] P.P. Grasse': LA Reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. La The'orie de la stigmergie: essai d'interpre'tation du comportment des termites constructeurs. Insects Sociaux, 6, 41-88

[Gutjahr 2000] W.J. Gutjahr: A graph-based Ant System and its convergence. Future Generation Computer Systems. 16, 873-888

[Hajek 1988] B. Hajek: Cooling schedules for optimal annealing, Math. of OR, 13, 311-329

[Hampo 1992] R.J. Hampo, K.A. Marko: Application of Genetic Programming to the control of vehicle systems, Prod. for the intelligent vehicles symposium, Detroid, Mi, USA

[Hofmeyr 1999] S.A. Hofmeyr, S. Forrest: Immunity by Design: An Artificial Immune System, Proceedings of the Genetic and Evolutionary Computation Conference, 2, 1289-1296

[Hofmeyr 2000] S.A. Hofmeyr, S.Forrest: Architecture for an Artificial Immune System, Evolutionary Computation, 8(4), 443-473

[Hofmeyr 2001] S.A. Hofmeyr: An Interpretative Introduction to the Immune System. In: L.A. Segal, I. R. Cohen, Design Principles for the Immune System and Other Distributed Autonomous Systems, Oxford University Press, 3-26

[Holland 1975] J. Holland: Adaptation in Natural and Artificial Systems, the University of Michigan press, 1975

[Hölldobler 1994] B. Hölldobler, E.O. Wilson: Journey to the Ants. Bellknap Press and Harvard University Press

[IRRIIS annex] IRRIIS, Annex I – “Description of work” p.11

[Karaoglu 2005] B. Karaoglu, H. Topcuoglu, F. Gurgen: Evolutionary Algorithms for Location Area Management, in F. Rothlauf et al. (Eds.): Evoworkshops 2005, Springer-Verlag Berlin, LNCS 3449, 175-184, 2005

[Kennedy 1995] J. Kennedy, R. Eberhart: Particle Swarm Optimization, IEEE International conference on Neural Networks, 4, 1942-1948

[Kephart 1994] J.O. Kephart: A Biologically Inspired Immune System for Computers. Proc. of the Fourth Inter. Workshop on the Synthesis and Simulation of Living Systems, MIT Press, 130--139

[Koza 1992] J. Koza: Genetic Programming: on the programming of the computers by means of natural selection, MIT press, Massachussets, USA

[Koza 1996] J.R. Koza, F.H. Bennett, D. Andre, M.A. Keane: Automated WYWIWYG design of both the topology and component values of electrical circuits using genetic programming, Proc. of the First Annual Conf. on Genetic Programming - GP'96, Stanford University, USA, 121-131

[Krishnaiyer 2002] K. Krishnaiyer, S. Hossein Cheraghi: (ANT Algorithms: Review and Future Applications, IERC Conference

[Kumar 2006] Y. Kumar, B. Das, J. Sharma: Service restoration in distribution system using non-dominated sorting genetic algorithm, Ele. Pow. Sys. Res. 76, 768-777, 2006

[Lin 2005] L. Ling, L. Zhiwei, H. Shaoxian, W. Gang: A distributed model for power systems restoration based on ant colony optimization algorithm, Trans. and Dist. Conf. & exhi., Asia and Pacific, Dalian, china

[Luan 2002] W.P. Luan, M.R. Irving, J.S. Daniel: Genetic algorithm for supply restoration and optimal load shedding in power system distribution networks, IEE Porc.-Gener. Transm. Distrib., 149(2), 2002

[Maniezzo 2004] V. Maniezzo, L.M. Gambardella, F. De Luigi: Ant Colony Optimization. In: by G.C. Onwubolu, B. V. Babu: New Optimization Techniques in Engineering, Springer-Verlag Berlin Heidelberg, 101-117

[Montana 1996] D.J. Montana, S. Czerwinski: Evolving control laws for a network of traffic signals, Proc. First annual Conf. on Genetic Programming - GP'96, Stanford University, USA, 333-338

[Nara 1997] K. Nara: Genetic algorithm for power systems planning, Proc. of the 4t Inter. Conf, on Advances in Power Systems Control, Operation and Managment, APSCOM-97, Hong Kong

[Rao 2004] C. V. Rao, J.R. Kirby, A.P. Arkin: Design and diversity in Bacterial Chemotaxis: A Comparative Study in Escherichia coli and Bacillus subtilis, PLoS Biology, 2, (2), 239-252

[Sarafijanovic 2004] S. Sarafijanovic, J-Y. Le Boudec: An Artificial Immune System for Misbehaviour Detection in Mobile Ad-Hoc Networks with Virtual Thymus, Clustering, Danger Signal and Memory Detectors, Proceedings of ICARIS-2004, 3rd International Conference on Artificial Immune Systems}, 342-356,

[Somayaji 2000] A. Somayaji, S. Forrest: Automated response using system-call delay. In: Usenix Security Symposium

[Srinivas 1994] M. Srinivas, L.M. Patnaik: Genetic algorithms: a survey, IEEE Computer, 26, 17-26

[Timmis 2004] J. Timmis, T. Knight, L.N. De Castro, E. Hart: An Overview of Artificial Immune Systems. In: R Paton and H Bolouri and M Holcombe and J H Parish and R Tateson, Computation in Cells and Tissues: Perspectives and Tools for Thought, Springer , 51-86

[Tomassini 1995] M. Tomassini: A survey of genetic algorithms. In: D. Stauffer: Annual Reviews of Computational Physics, World Scientific, 3, 87--118

[VanVorhis 1986] S.E. Van Vorhis Key, T.C. Baker: Observations on the trail disposition and recruitment behaviours of the Argentine ant. Ann. Entomol. Soc. Am. 79, 283-288

[Vergassola 2007] M. Vergassola, E. Villermaux, B.I. Shraiman: 'Infotaxis' as a strategy for searching without gradients, Nature, 445, 406-409

[Wedde 2006] H.F. Wedde, M. Farooq: A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks,Journal of Systems Architecture, 52

[Willis 1997] M.-J. Willis, H.G. Hiden, P. Marenbach, B. McKay, G.A. Montague: Genetic programming: an introduction and survey of applications, IEE Conference Publications, CP446, 314-319

[Schoonderwoerd 1996] R. Schoonderwoerd, O.E. Holland, J.L. Bruten, L.J M. Rothkrantz: Ant-Based Load Balancing in Telecommunications Networks, Adaptive Behaviour, MIT Press,2,169-207