Artificial Immune Systems for Solving the Traveling Salesman Problem

A use of CLONALG in Pathfinding and Optimization

Bachelor Thesis

FH Campus Wien

Vorgelegt von:
Sebastian Ukleja

Personenkennzeichen
XXXXXXXXXXX

Abgabe am:
XX.YY.20ZZ

Abstract

(E.g. “This thesis deals with...”)

Abbrevations

BIS	Biological Immune System
AIS	Artificial Immune System
Ab	Antibody
Ag	Antigen
TSP	Traveling Salesman Problem

Keywords

Artificial Immune System
Clonal Selection Algorithm
Negative Selection
Traveling Salesman Problem

Contents

1 Introduction **1**

2 Biological Immune System **2**
2.1 Negative Selection 3
2.2 Clonal Selection 3

3 Artificial Immune System **5**
3.1 Basics of an AIS 5
3.2 Negative Selection in AIS 6
3.3 Clonal Selection in an AIS 6

4 Traveling Salesman Problem **9**
4.1 Background 9
4.2 Mathematical definition 9

5 Evaluation **11**
5.1 Setup 11
5.2 Results 12

6 Related work **13**

7 Conclusio **14**

A Appendix **15**
Figures 16
Tables 17
Literaturverzeichnis 18

Chapter 1

Introduction

Textkörper mit Bild

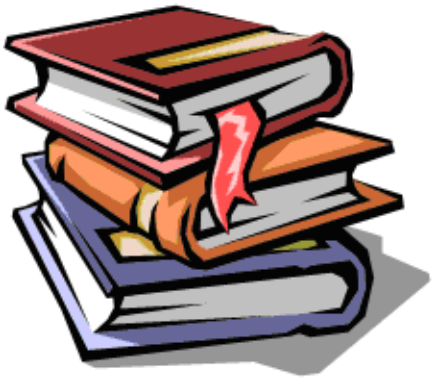

Figure 1.1: Ein Stapel Bücher
state of the art

Chapter 2

Biological Immune System

A biological immune system (BIS) has many features that are useful in machine learning. It can be described with the following terms [TAN16]:

- Distributed
- Parallel
- Multi-level
- Distinguishes between self and non-self
- Noise resistant
- Self-organized
- Associative memory

It is distributed because there is no need for a central control instance, the BIS acts where it need to and it does so immediately without supervising. It can operate on multiple parts of the body simultaneously thus it is parallel. The BIS works in different levels. At first there is the physical barrier, the skin and the body fluids. If this level fails the innate immune system responds, which is a pre-programmed immune reaction that can respond to known and non-changing threats. Finally, there is the adaptive immune system if all of the previous levels fail. This is the system most interesting for computational science, because of its ability to learn on its own. A very important ability of the BIS is the distinguishing between any self-element of the body and the non-self or potential threat. It is noise resistant because it can react to variations of known threats. The BIS does not need any supervising through the brain or any other central system to organize its work flow, therefore its self-organized. Finally, it has an associative memory which is used to react on similar and variant of threats already encountered. As mentioned above, the adaptive immune system is the most interesting part of the BIS for modelling an artificial immune system (AIS). The immune system uses many principles to be effective, in terms of computational science the principles of negative selection and clonal selection are specifically interesting.

2.1 Negative Selection

The adaptive immune system uses two kind of lymphocytes to counter a threat. The T-Lymphocyte and the B-Lymphocyte. Both have different roles but both lymphocytes must have the ability to distinguish between the self and the non-self-cells in the body. A fault in this system can not only lead to an infection, it could trigger an autoimmune reaction because a self-cell could be identified as a non-self threat. To avoid this the Lymphocytes are generated through the process of negative selection. While the B-Cells will be developed in the bone marrow and the T-Cells in the thymus the process is exactly the same. Both cell types are presented to a wide range of self-cells. If any of them react to such a self-cell, the Lymphocyte will be killed and another will be generated. This process is repeated till there are only B- and T-Cell which react to non-self cells [TAN16]. This process will be imitated in the AIS in generating detector sets.

After that the lymphocytes will be released into the tissue and the blood system. The B-cells are called antibodies (Ab) in this context, and any non-self cell is called an antigen (Ag). The B-cell has the ability to recognize and dock to a specific Ag. The ability to recognize an Ag is called affinity. If an Ab has a high affinity to an Ag it is especially good in recognizing and countering this Ag. Because the generation of Lymphocytes and therefore the affinity to different Ag's is random, additional measures are necessary to improve effectiveness.

2.2 Clonal Selection

If a B-Cell encounters an antibody it is able to proliferate (divide) into multiple terminal cells which are clones of the cell. The cell is not only cloned, the different clones will be mutated to improve the affinity to the antigen. The process of cloning and mutating will be repeated till a population and affinity threshold is reached which ensures the most efficient response to the Ag. Only the Ab's with the highest affinity score will be cloned and mutated. If the affinity is high enough, the B-Cell can proliferate into memory cells which will stay after the response and ensure that a secondary response to a similar Ag will be much faster than the initial one [DEC02]. This memory cells represent the associative memory of the adaptive immune system.

The principles of negative selection and clonal selection are important concepts in designing an artificial immune system. The clonal selection aspect of the BIS is basically the learning system. The cloning process is a form of reinforcement learning and leads to a continues improvement [DEC02]. The mutation process itself is called affinity maturation. Random changes in the genes leads to changes in affinity in every single clone. The mutation process is inverseji to the affinity level. Higher affinity level means lower mutation rate [DEC02]. Clonal selection is based on the basic evolutionary theory of Charles Darwin.

The three basic principles are [DEC02]:

- repertoire diversity (high population of Ab's)
- genetic variation (random changes to the population (blind variation))
- natural selection (high affinity Ab's will reproduce and maintained)

These are high level abstract concepts and are only used for a very brief overview of the immune system. The BIS is far more complex but the details are out of scope of this thesis.

Chapter 3

Artificial Immune System

3.1 Basics of an AIS

A definition of an AIS is given by [TAN16]: "Articial Immune System (AIS) is a computational intelligence system inspired by the working mechanism and principles of the biological immune system" An AIS can be used in machine learning and is comparable to an artificial neural net in terms of different application fields. It shares some similarities with genetic algorithms due to the cloning and mutation process in clonal selection.

Generally speaking: a set of detectors (antibodies) react to a specific anomaly (antigen). This anomaly could be a malicious code, an IP address and port combination that is not allowed in the network or a pattern that has to be classified. What represents an antigen in the algorithm completely depends on the context and the use of the algorithm. An AIS can be used in an Intrusion Detection System [PAM17], learning and pattern recognition [DEC02], for recommender systems, data mining and clustering [BUR14] and optimization [NAN08].

The basic steps of an AIS can be summarized as [TAN16]:

1. Initialize/present antigen
2. Initialize antibody population
3. Calculate affinity for each antibody to the antigen
4. Check life cycle of each antibody and update it
5. If stopping condition met go to 6 else go to 3
6. Output antibody population

Step 2 and Step 4 are the steps where negative selection and clonal selection will be relevant in most forms of AIS algorithms.

At the time of writing there are mainly 5 concepts that are used in an AIS. The aforementioned negative selection, clonal selection and more recently the immune network theory, the danger theory and the dendritic cells theory.

3.2 Negative Selection in AIS

A typical AIS has a set of detectors which react to any non-self data of the system. The most common way to generate these detectors is to create them randomly and let them undergo a negative selection in which they are presented to self data sets. If one of the detectors recognizes a self data, therefore the affinity is high enough, it will be destroyed. In this case the AIS follows the process that is found in the BIS.

It is important to decide in which way affinity will be measured and which conditions will trigger an immune reaction. If the data that is used consists of numerical values, or is easily converted into a vector of those, the Euclidean distance [4] is commonly used as a mean to measure affinity [TAN16]: (Euclidean Distance Formula)

The Euclidean distance gives us the straight-line distance between two points in an Euclidean space. In the formula above xi and yi are values from two n-dimensional vector x and y.

Another possibility is the use of different variants of the Hammington distance [3]. The Hammington distance is the number of bits that must be changed in two bit strings of the same length to make both strings identical. A common variant of this is the use of the so called r-continues bits. In this case a detector recognizes an element if r continues bit are identical with the element.

Non-self element	0010110
Detector	0011101

In the example above the first 3 continues bit match in both strings. If the threshold of the detector is r = 3, it would react to this bit string if the number of continues bits is 3 or higher. In case of recommender systems another way to measure the affinity is to use the Pearson correlation [5]. But the Pearson correlation is not suited for optimization tasks or specifically the use in the travelling salesman problem.

3.3 Clonal Selection in an AIS

Negative Selection is good for generating a population of detectors. To mimic the learning abilities of the biological immune system, the clonal selection principle can be used. The CLONALG algorithm was proposed by [DEC02] in 2002 and is the foundation of many AIS algorithms that use the clonal selection principle. The CLONALG is also present in more recent variations of the AIS like in aiNet and in algorithms the utilize the immune network theory [CITACION NEEDED]. This thesis will focus on

the CLONALG algorithm and some of its variations, as these algorithms will be used to generate the data in 5.

3.3.1 CLONALG

The CLONALG algorithm mimics the clonal selection of the BIS on an abstract level. It will be applied mainly at step 4 in the basic AIS sequence in 3. The algorithm starts with generating a initial population C. It then calculates the affinity of the population to an antigen which is called the fitness. The n best antibodies of the population C will be chosen, determined by a fixed fitness value, and form the antibody set S. Every antibody in S will be cloned and represents our clone set P. Now the clone set will be mutated, every clone get randomly changed relatively to its affinity value. After the mutation the n best clones out of P are chosen. Then a new population will be generated and the process begins anew till a stopping condition is met.
A simple pseudocode can look like this, based on [RIFF09]:

Algorithm 1 Simple CLONALG pseudo code

```
Generate initial population C of A antibodies
Calculate Fitness(C)
while stopping condition not met do
    S= Select the n best antibodies from C
    P= Generate clones of the antibodies in S
    Mutate(P)
    C= Select the n best antibodies out of P
    C= C + New population A-n
end
```

Usually the stopping condition is a given amount of evaluation where no increase in fitness is achieved or felt below a given threshold. The initial CLONALG algorithm as proposed by [DEC02] operates with static parameters and don't adjust anything besides the mutation. Different parameters are needed for different applications fields as was also stated in [DEC02]. This parameters must be changed beforehand and can't adapt dynamically during the runtime of the algorithm.

Although the algorithm is very simple and efficient in solving different tasks like multimodal problems or pattern recognition, drawbacks do exist [GARRET04]. Choosing how many member should be cloned is difficult. The lack of adaptive parameters can lead to inefficiency because of bad scalability and too many evaluations which could have been avoided [Garret04].

3.3.2 Adaptive CLONALG variants

To make the algorithm more efficient and adaptive, some variants where proposed. There are different parameter control strategies for the CLONALG algorithm. One of these variants is proposed by [RIFF09]. It is based on the idea of reinforcement learning. The antibody sets will be either rewarded for a high affinity or penalized for

a low one. This will be achieved through population control. The reward is the increase in antibody population for a given set and the penalty is the decrease in population [RIFF09]. The mutation factor is governed by the population size. This adaptive technique allows the algorithm to adjust the core parameters during runtime and makes it more efficient in problem solving and hardware usage as shown in [RIFF09].
Another method is proposed by [Garret04]. Some techniques from the evolutionary algorithms will be applied to the CLONALG in this approach.

Chapter 4

Traveling Salesman Problem

4.1 Background

One of the first publications of the traveling salesman problem was by the mathematician Karl Menger in the 1920's [Applegate 1991]. The problem itself was discussed earlier by Sir William Rowam Hamilton and Kirkman [Matai 2010]. It describes a graph with a certain amount of nodes with known distances between the nodes. The goal is to find a route where every node is visited exactly one time and the route ends at the node where it started. The route should be the shortest possible.((example picture of an tsp))

The algorithmic complexity for a symmetric graph with n nodes is: $\frac{n(n-1)}{2}$ [Applegate 1991]. The graph is symmetric if the distance between two nodes n and m is the same as the distance between m and n. Asymetric traveling salesman problems also exist with a higher complexity of $n(n-1)$
The complexity of the problem is categorized as non-deterministic polynomial hard (NP-Hard). The runtime of an algorithm can scale exponentially with the number of nodes in the graph. No algorithm is able to solve the problem in polynomial time.
Current algorithms work with heuristics to solve the TSP.

The TSP is common in many real world applications. Drilling of printed circuit boards, computer wiring, order picking in warehouses, vehicle routing and DNA sequencing are some some fields where the TSP is present [Matai 2010].

4.2 Mathematical definition

The symmetric graph is defined as $G = (V, E)$ where $V = \{1, .., n\}$ are the vertex or the nodes and $E = \{(i, j) : i, j \in V, i < j\}$ are the edges or routes. Additionally there is an arc set $A = \{i, j) : i, j \in V, i\}$ which defines all routes in the graph, no route can be used twice. A cost matrix is defined on the edge or the arc set. Usually the cost matrix is calculated using the Euclidean distance. [Matai 2010].

A typical TSP consists of a set of cities (V), distances between the cities (E) and the cost measured in the euclidean distance between the cities (C). All TSP used in this thesis will follow this convention.

Chapter 5

Evaluation

5.1 Setup

The CLONALG algorithm will be applied to a set of 17 different TSP problems from the TSP library TSPLIB95 [1]. The elitist ant colony algorithm will be applied to the same set of problems.

The stopping criteria will be no improvement after a set amount of evaluations. No improvement means that the algorithm has not found a shorter route in it's actual iteration compared to the last one, this is often also called stagnation. The criteria for the results are:

1. Score
2. Time in ms
3. Evaluations
4. Percentage of optimal score

The score is measured as the summarized Euclidean distance of the presented best tour. To compare the results the relative distance to the optimal score will be used. The algorithm will be run multiple times on one TSP therefore the arithmetic mean of each criteria will be the end result. The CLONALG algorithm will be applied to the problem set multiple times with different parameters. The adaptive algorithms will be applied only once because of the dynamic parameters. The parameters which will be altered are

1. Population size
2. Mutationfactor
3. Selection size
4. Random replacements

The first set of parameters will be the default parameters provided by the OAT. The second one are the parameters that are proposed by DeCastro [DEC02] for solving TSP problems about the size of 30 nodes.
The adaptive variant of the CLONALG algorithm will use the default vaulues at the beginning and adjust the paramaters during runtime.
To measure the results, a modified version of the optimization algorithm toolkit (OAT) [2] will be used. The changes are a slightly different set of TSP Problems used in the domain and the addition of an adaptive CLONALG hybrid algorithm. The used TSP problems are listed in the appendix. The distances between the nodes in the TSP problem are measured as Euclidean distance. The implemented CLONALG algorithm is based on the specifications in [DEC02]. The adaptive variants expand the concept based on [RIFF09] and [GARRETT04]. Both algorithms will be applied 100 times on every single TSP problem.

The hardware is a i5-3320M dual core cpu with 2,60ghz each and 8gb of ram, run on a Windows 10 operating system with only the minimal background tasks. All cpu and ram usage shown in this thesis is after other processes are taken into account.

5.2 Results

5.2.1 No improvements

The algorithms are run 100 times on every TSP. The stopping criteria are no improvements after 10000 iterations of the algorithm. This number is chosen to give the algorithm enough time without restricting it to es specific amount of seconds. The cpu usage spiked around 58% and was average around 30%.

Chapter 6

Related work

Chapter 7

Conclusio

Appendix A

Appendix

List of TSP problems taken from TSPLIB95 library:
a280
berlin52
ch130
ch150
eil101
eil51
eil76
kroA100
kroC100
kroD100
lin105
pcb442
pr2392
pr76
st70
tsp225
ulysses22

List of Figures

1.1 Ein Stapel Bücher 1

List of Tables

Bibliography