Chapter 6

Evolving Artificial Lymphocytes

This chapter proposes an approach to evolve an optimal initial set of ALCs using a genetic algorithm (GA). An optimal set of ALCs can be defined as a set of trained ALCs with the largest non-self space coverage and with minimum overlap among the ALCs in the set. Thus, the problem that needs to be optimised is multi-objective. GAIS (Genetic Artificial Immune System) uses a GA to search the problem space for solutions to these objectives. Figure 6.1 (page 50) illustrates the flow layout of the GAIS algorithm and the function of the GA within GAIS. GAIS has an initial empty set of ALCs. The set is populated by evolved ALCs which are obtained from the GA one-by-one (sequentially), until the set of ALCs has converged. The converged ALC set is used to classify any pattern as non-self or self. Therefor the GAIS algorithm has two phases:

- The evolutionary process to evolve the best ALC using a GA.
- The detection process of non-self patterns.

The first phase forms part of the training process of the ALCs. The role of the GA in GAIS is explained in section 6.1. After training, the evolved ALC set is used to classify any pattern as self or non-self. The classification process is discussed in section 6.2.

6.1 Genetic Algorithm to Evolve an ALC

The success and efficiency of GAIS is mainly influenced by the quality of the ALCs used to detect non-self patterns. An ALC is randomly initialised and trained with either the adapted negative selection method or positive selection method. The trained ALC usually forms part of a static set of trained ALCs which is used to detect non-self patterns. This section shows how a GA can be used to evolve a dynamic set of ALCs. The main objective of the GA is to evolve an

ALC to be added to the existing active ALC set such that the evolved ALC maximises its ADT (if the ALC was trained with the adapted negative selection method) or minimises its ADT (if the ALC was trained with the positive selection method) and to minimise the average overlap with existing ALCs in the active ALC set. In the case of the adapted negative selection, an ALC with the maximum hamming distance from the set of self patterns implies that the maximum non-self space is covered by the ALC without overlapping with the self set. In the case of positive selection, an ALC with the minimum hamming distance from the set of self patterns implies that the similarity between the evolved ALC's receptor and the set of self patterns needs to be maximised for maximum non-self space coverage. Additionally, the GA also maximises the distance between the new ALC and the ALCs in the active ALC set. Maximising the distance between the new ALC and the active ALC set guarantees that the evolved ALC has the lowest average overlap with the existing set of ALCs, and this forces greater coverage of non-self space. Thus, a GA is used in GAIS to optimise a multi-objective goal. The purpose of the GA is to evolve one optimal ALC to be added to the active set of ALCs.

The GA has an initial population of I ALCs. The initial population is randomly initialised (as shown in figure 6.1). The fitness of each ALC in the population is calculated to be proportional to the ADT and overlap with other ALCs (refer to section 6.1.3). The evolutionary process stops as soon as the maximum number of generations is reached or the fitness of the ALC population has converged. The fittest ALC in the population is then selected to be added to the active ALC set.

The following is the pseudo-code summary for the GA in phase 1 to evolve an ALC:

```
1. set $g=0$ ($g$ counts the number of generations)
2. randomly initialise $I$ chromosomes as population $H_g$
3. while no convergence in $H_g$
    (a) evaluate the fitness of each chromosome in $H_g$ using equation (6.1) or (6.2)
    (b) apply cross-over
        i. select two parents
        ii. produce offspring from the two selected parents and add them to the offspring
            set
    (c) apply mutation
```

i. select a candidate chromosome from the offspring set

ii. mutate the selected candidate

(d) select the new generation from the previous generation and the offspring set

(e) $g = g + 1$

The operators and other design aspects of the algorithm are explained in more detail below.

6.1.1 ALC as Chromosome

As explained in section 3.3, each chromosome represents a potential solution and consists of the set of parameters (genes) to the problem that needs to be optimised. *Alleles* are specific values from the domain of the corresponding parameter assigned to the gene.

As explained in Section 5.2, each ALC has a receptor that is a bit-string of fixed length k. These bit-strings are used to detect or match patterns as explained in section 5.2.3. Therefor all patterns that need to be classified by an ALC must first be coded to a bit-string. A technique known as *binning* is used to discretise patterns with floating-point values. *Binning* calculates the valid interval of values for each attribute c in the data set. The interval is calculated by subtracting the minimum value of the attribute from the maximum value of the attribute. The calculated interval of each attribute is then divided into b bins. Each bin represents a group of values. To determine into which group a value of the specific attribute falls, the following calculation is used

$$G(x_{c,j}) = \frac{x_{c,j} - min_{c=1,\dots C}\{x_{c,j}\}}{max_{c=1,\dots C}\{x_{c,j}\} - min_{c=1,\dots C}\{x_{c,j}\}} * b$$

where $x_{c,j}$ is the floating-point value x of attribute c in pattern j and C is the number of attributes in a pattern.

The floating-point value in the pattern is now in nominal form. To code patterns from nominal to binary, the number of groups or bins an attribute can have is used to determine the number of bits necessary to represent an attribute's values. The number of bits is calculated as $N = \frac{log(b)}{log2}$ bits. Then the nominal value in each attribute can be coded to binary using standard binary encoding. The binary encoded chromosome in effect represents the receptor of an ALC.

6.1.2 The Initial Population

The receptor of an ALC is randomly initialised. The initial population of the GA is a set of these randomly initialised ALCs. Random selection ensures that the initial population of chromosomes has a good uniform coverage of the search space. The size of the population, I, is static throughout the evolutionary process.

6.1.3 Evaluating the Chromosome's Fitness

The ALCs in the AIS can be trained with one of two selection methods (as explained in section 5.2.2). There are two objectives that need to be optimised by the GA. The first objective to be optimised is to evolve an ALC with the largest average hamming distance, $\chi(D,\mathbf{r})$, from an existing set of ALCs. This objective ensures that the GA evolves an ALC with the lowest average overlap with an existing set of ALCs. The average hamming distance from an existing set of ALCs is calculated as follows

$$\chi(D,\mathbf{r}) = \frac{\sum_{l=1}^{l\leq P}\gamma(\mathbf{d}_l,\mathbf{r})}{P}$$

where D is the active ALC set, P is the number of ALCs in the active set D, $\mathbf{d}_l$ is the receptor of the l-th ALC in the active set and $\mathbf{r}$ is the receptor of the ALC from which the average hamming distance must be calculated. γ is the hamming distance between $\mathbf{d}_l$ and $\mathbf{r}$, defined as

$$\gamma(\mathbf{d}_l,\mathbf{r}) = \sum_{i=1}^{i\leq k} XOR(d_{l_i}, r_i)$$

where XOR is the exclusive-or between the bits of $\mathbf{d}_l$ and $\mathbf{r}$, i is the bit-index and k is the size of the receptor.

The second objective that needs to be optimised by the GA is to evolve an ALC with an optimised affinity distance threshold. The second objective differs for each selection method as explained below.

6.1.3.1 Adapted Negative Selection Fitness

In the case of the adapted negative selection, the GA needs to evolve an ALC that has the maximum affinity distance threshold a_{neg}. This objective ensures that the GA evolves an ALC with

the maximum hamming distance from the set of self patterns. An ALC with maximum hamming distance from the set of self patterns implies that a maximum non-self space is covered by the ALC without overlapping with the self set.
The fitness function υ_{neg} for the adapted negative selection is then defined as

$$\upsilon_{neg} = w_1 a_{neg} + w_2 \chi(D, \mathbf{r}) \tag{6.1}$$

with $w_1 + w_2 = 1.0$ where $w_1, w_2 \in [0, 1]$. w_1 and w_2 are weights that determine the influence that a_{neg} and $\chi(D, \mathbf{r})$ have on the fitness of an ALC, respectively. Since there is no conflict between a_{neg} and $\chi(D, \mathbf{r})$, the sub-objectives are only weighted with w_1 and w_2. With $w_1 = 1.0$, $\chi(D, \mathbf{r})$ has no influence on the ALC's fitness and the fitness is determined only by a_{neg}. With $w_2 = 1.0$, a_{neg} has no influence on the ALC's fitness and the fitness is determined by $\chi(D, \mathbf{r})$. The value of w_2 is calculated as $w_2 = 1.0 - w_1$, and the value of w_1 is problem dependent, obtained through cross-validation.

6.1.3.2 Positive Selection Fitness

In the case of positive selection, the GA needs to evolve an ALC that has the minimum affinity distance threshold a_{pos}. This objective ensures that the GA evolves an ALC with the minimum hamming distance from the set of self patterns. This objective implies that the similarity between the evolved ALC's receptor and the set of self patterns needs to be maximised. The similarity between two patterns is calculated by the complement of their hamming distance, i.e. $k - a_{pos}$ where k is the length of the receptor and a_{pos} the ADT.
The fitness function υ_{pos} for positive selection is defined as

$$\upsilon_{pos} = w_1(k - a_{pos}) + w_2 \chi(D, \mathbf{r}) \tag{6.2}$$

where w_1 and w_2 have the same meaning as for the adapted negative selection.

6.1.4 Parent Selection

The chromosomes that are selected for crossover to produce a set of offspring is randomly selected from an elitist set of chromosomes (elitism is discussed in section 3.3.3.). The generation gap is calculated as

$$\varsigma = I * e$$

where e is the rate of elitism, with $e \in (0,1]$.

A small value of e results in more diversity among the individuals for the next generation. If e is too big there will be less diversity among individuals for the next generation. After the elitist set has been created, the GA applies uniform crossover (as explained in section 3.3.1.2) between the randomly selected parents with probability, p_c, to produce offspring. The produced offspring forms part of the offspring set for the current generation. The crossover probability, p_c, decreases with an increase in the number of generations, i.e.

$$p_c = 1.0 - \frac{g}{G}$$

where g is the completed number of generations and G is the maximum allowed number of generations. With initial value $g = 0$, p_c will have an initial value equal to 1.0, implying a high probability of crossover. As the population evolves and becomes fitter, the need to exchange genetic material between fit individuals decreases. Thus the probability of crossover decreases as the population evolves.

6.1.5 Mutation

The offspring created through the uniform crossover operator (as explained above), is randomly selected for mutation. The GA applies random mutation (as explained in section 3.3.2.2) on the selected individual with probability, p_m. The probability, p_m, is calculated for each selected individual using

$$p_m = \frac{k - \chi(D,\mathbf{r})}{k}$$

where k is the length of the ALC's receptor and $\chi(D,\mathbf{r})$ is the average hamming distance between the ALC and the existing set of ALCs. $\frac{k-\chi(D,\mathbf{r})}{k}$ gives the average similarity rate between the ALC's receptor and the existing set of ALCs. Since one of the objectives is to ensure that the GA evolves an ALC with the lowest average overlap with an existing set of ALCs, the similarity rate should determine p_m. A high similarity rate implies that the ALC needs to be mutated with a high probability (mutation rate) to evolve a mutated ALC with the lowest average overlap with an existing set of ALCs. A low similarity rate implies that the ALC has a low average overlap with an existing set of ALCs and thus the rate of mutation must be low.

6.1.6 Selection of the New Population

The new population is selected from the parents of the current population and the offspring. All individuals (parents) in the elitist set survive to the next generation to ensure that the maximum fitness in the population does not decrease. The remainder of the new population is filled with the fittest offspring, determined using linear ranking (as explained in section 3.3.3).

6.1.7 Convergence of the GA

In the above GA algorithm, convergence in step 4 is reached when the maximum number of generations is reached or when the difference in the 2-point moving average between the fitness of the new population and the fitness of the previous generation's population is less than μ_T. The 2-point moving average is calculated as follows

$$\mu_g = \frac{\mu(H_{g-1}) + \mu(H_g)}{2.0}$$

where $\mu(H_{g-1})$ is the average fitness of population H_{g-1}, $\mu(H_g)$ is the average fitness of population H_g, and g is the current generation number. μ_g is calculated after each generation. The population H_g has converged if $|\mu_{g-1} - \mu_g| < \mu_T$ for $g = 1, ..., WindowSize$. A small difference in moving average, i.e. less than μ_T, implies that the reproduction and selection operators on the population H_g resulted in a minor change on the average fitness of the population and thus further evolution is unnecessary.

6.2 The GAIS Algorithm

This section explains the classification process of GAIS. The GA in phase 1 is repeatedly applied until the evolved active ALC set has converged (convergence of the active ALC set is explained in section 6.2.2). The ALCs in the converged active set, D_L, are then trained with one of the selection methods (as explained in section 5.2.2). After training, the *EMR* is set to 0.0 and GAIS tries to detect a set of patterns equal to the iteration size, *IS*, as non-self patterns using D_L. When a pattern is detected as non-self, the hit counter, *HC*, of each ALC that detected the non-self pattern is updated (as explained in section 5.2.3). If a pattern is not detected as non-self, the ALCs in the annihilated set is used to find a match. If a match is found, the annihilated ALC becomes mature and is moved to D_L. After all the patterns in an iteration have been classified, the life counter, $\tau(IS)$, of each ALC in the active set D_L is calculated to determine in which state an ALC

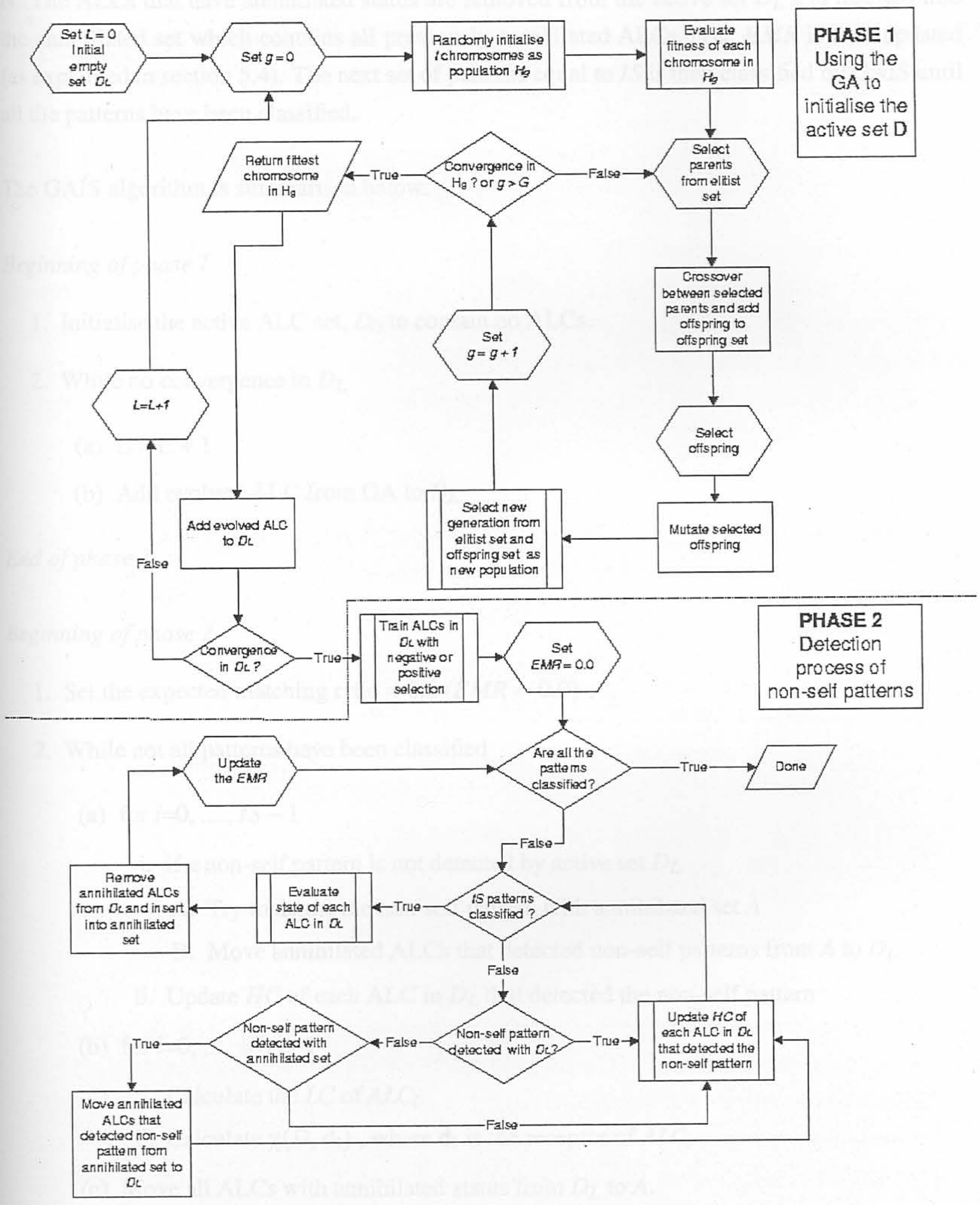
Set L = 0 Initial empty set DL
Set g = 0
Randomly initialise I chromosomes as population Hg
Evaluate fitness of each chromosome in Hg
PHASE 1 Using the GA to initialise the active set D
Select parents from elitist set
Convergence in Hg ? or g > G
False
True
Return fittest chromosome in Hg
Crossover between selected parents and add offspring to offspring set
Set g = g + 1
L=L+1
Select offspring
Select new generation from elitist set and offspring set as new population
Mutate selected offspring
Add evolved ALC to DL
False
Convergence in DL ?
True
Train ALCs in DL with negative or positive selection
Set EMR = 0.0
PHASE 2 Detection process of non-self patterns
Update the EMR
Are all the patterns classified?
True
Done
False
Remove annihilated ALCs from DL and insert into annihilated set
Evaluate state of each ALC in DL
True
IS patterns classified ?
False
Non-self pattern detected with annihilated set
False
Non-self pattern detected with DL?
True
Update HC of each ALC in DL that detected the non-self pattern
True
Move annihilated ALCs that detected non-self pattern from annihilated set to DL
False

Figure 6.1: Flow layout of the Artificial Immune System

is. The ALCs that have annihilated status are removed from the active set D_L and inserted into the annihilated set which contains all previously annihilated ALCs. The *EMR* is then updated (as explained in section 5.4). The next set of patterns equal to *IS* is then classified by GAIS until all the patterns have been classified.

The GAIS algorithm is summarised below:

```
Beginning of phase 1

1. Initialise the active ALC set, $D_0$ to contain no ALCs.

2. While no convergence in $D_L$

    (a) L = L + 1

    (b) Add evolved ALC from GA to $D_L$

End of phase 1

Beginning of phase 2

1. Set the expected matching ratio = 0.0 ($EMR = 0.0$)

2. While not all patterns have been classified

    (a) for $i$=0, ...., $IS - 1$

        i. if a non-self pattern is not detected by active set $D_L$

            A. Try to detect the non-self pattern with annihilated set $A$

            B. Move annihilated ALCs that detected non-self patterns from $A$ to $D_L$

        ii. Update $HC$ of each ALC in $D_L$ that detected the non-self pattern

    (b) for $l$=0, ...., $|D_L|$ - 1

        i. Calculate the $LC$ of $ALC_l$

        ii. Calculate $\chi(D, \mathbf{d}_l)$ , where $\mathbf{d}_l$ is the receptor of $ALC_l$

    (c) Move all ALCs with annihilated status from $D_L$ to $A$.

    (d) Update the $EMR$ (as explained in section 5.4)
```

End of phase 2

Several aspects of the above algorithm need to be explained in more detail, for example the initialisation process of D in phase 1 and the convergence of D in phase 1, step 2. These aspects are discussed in the following sections.

6.2.1 Initialisation of the ALC Set

The initial ALC set is defined as the empty set $D_0 = \{\}$. $|D|$ increases after the best ALC as determined by the GA is added to D, i.e. $D_L = D_{L-1} \cup \{ALC_l\}$ where ALC_l is the evolved ALC and $l = 0, \ldots\ldots, L$. L is the number of ALCs evolved by the GA before the active ALC set converged. For each $l > 0$, the existing active set D_L is used by the fitness function to ensure that the newly evolved ALC_l is on average the furthest away from the existing active ALC set D_L.

6.2.2 Convergence of the ALC Set

In the above GAIS algorithm, convergence in phase 1, step 2 is reached when the difference in the 2-point moving average between the fitness of D_{L-1} and D_L is less then μ_T. The 2-point moving average is calculated as follows

$$\mu_L = \frac{\mu(D_{L-1}) + \mu(D_L)}{2.0}$$

where $\mu(D_{L-1})$ is the average fitness of set D_{L-1} and $\mu(D_L)$ is the average fitness of set D_L. μ_L is calculated after each evolved ALC is added to D_{L-1}. The set D_L has converged if $|\mu_{l-1} - \mu_l| < \mu_T$ for $l = L - WindowSize, \ldots, L$. After the ALC set has converged, all the ALCs in the active set are trained with one of the selection methods (as explained in section 5.2.2).

6.3 Conclusion

This chapter showed how a GA can be used to evolve an optimal set of ALCs using the proposed operators. The initialisation of a population, the selection operator, reproduction operators and the fitness function of the GA were discussed. The GAIS algorithm was presented and the initialisation of the ALC set were discussed. There are many other operators that can be considered when implementing the GA (as discussed in chapter 3), but the chosen operators have satisfying results (presented in the next chapter). Finding the best operators for the evolutionary process falls outside the scope of this dissertation. The focus is on classifying non-self patterns from seen self patterns with a set of ALCs that consists solely of ALCs with mature and memory status. The next chapter presents and analyses the results obtained from the GAIS classifier.