ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΜΟΝΤΕΛΑ ΤΕΧΝΗΤΩΝ ΑΝΟΣΟΠΟΙΗΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΓΙΑ ΤΗΝ ΕΞΟΡΥΞΗ ΓΝΩΣΗΣ ΑΠΟ ΣΥΝΟΛΑ ΔΕΔΟΜΕΝΩΝ

Διπλωματική Εργασία

ΒΑΣΙΛΕΙΟΣ Κ. ΚΑΡΑΚΑΣΗΣ

Επιβλέπων: Α.-Γ. Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2005

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΜΟΝΤΕΛΑ ΤΕΧΝΗΤΩΝ ΑΝΟΣΟΠΟΙΗΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΓΙΑ ΤΗΝ ΕΞΟΡΥΞΗ ΓΝΩΣΗΣ ΑΠΟ ΣΥΝΟΛΑ ΔΕΔΟΜΕΝΩΝ

Διπλωματική Εργασία

ΒΑΣΙΛΕΙΟΣ Κ. ΚΑΡΑΚΑΣΗΣ

Επιβλέπων: Α.-Γ. Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Α.-Γ. Σταφυλοπάτης	Σ. Κόλλιας	Π. Τσανάκας
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2005

Βασίλειος Κ. Καρακάσης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Στην παρούσα εργασία μελετώνται τα Τεχνητά Ανοσοποιητικά Συστήματα και πώς μπορούν αυτά να συνδυαστούν με υπάρχουσες τεχνικές για την εξόρυξη γνώσης από σύνολα δεδομένων. Το Ανοσοποιητικό Σύστημα αποτελεί ένα εξαιρετικά ευφυές και αποτελεσματικό σύστημα αναγνώρισης και αντιμετώπισης ξένων μικροοργανισμών που εισέρχονται στο ανθρώπινο σώμα. Η μελέτη της συμπεριφοράς του και των μεθόδων που χρησιμοποιεί μπορεί να καταδείξει ένα σύνολο νέων τεχνικών μάθησης μηχανών.

Στην εργασία αυτή εξετάζεται η αρχή της επιλογής των κλώνων και αρχικά εφαρμόζεται σε δύο προβλήματα αναγνώρισης ψηφιακών χαρακτήρων. Η μέθοδος αυτή χρησιμοποιώντας μία τεχνική υπερβολικής μετάλλαξης και ένα μικρό αρχικό πληθυσμό αντισωμάτων, επιδεικνύει πολύ καλά αποτελέσματα όσον αφορά στην ταχύτητα σύγκλισης.

Στην συνέχεια η μέθοδος της αρχής της επιλογής των κλώνων συνδυάζεται με την τεχνική του Προγραμματισμού Γονιδιακής Έκφρασης (ΠΓΕ) για την εξόρυξη γνώσης από σύνολα δεδομένων. Ο ΠΓΕ αποτελεί την φυσική εξέλιξη των Γενετικών Αλγορίθμων και του Γενετικού Προγραμματισμού, συνδυάζοντας τα κύρια πλεονεκτήματα των δύο μεθόδων, και εξαλείφοντας παράλληλα τα κυριώτερα μειονεκτήματά τους. Για την εξόρυξη γνώσης από τα δεδομένα εισάγεται αρχικά η έννοια του Αντιγόνου Κλάσης Δεδομένων, το οποίο αναπαριστά μία κλάση δεδομένων του προβλήματος. Για την εξέλιξη των κανόνων χρησιμοποιήθηκε ο αλγόριθμος επιλογής κλώνων με μοναδική προσθήκη την φάση της διόρθωσης των υποδοχέων. Για την αναπαράσταση των αντισωμάτων υϊοθετήθηκε η αναπαράσταση που χρησιμοποιείται από τον ΠΓΕ για τα χρωμοσώματα.

Η υβριδική τεχνική που προτείνεται σε αυτή την εργασία, δοκιμάστηκε σε προβλήματα αξιολόγησης του UCI repository, συγκεκριμένα στα προβλήματα MONK και στο πρόβλημα Pima Indians Diabetes. Σε όλα τα προβλήματα τα αποτελέσματα ήταν ιδιαίτερα ικανοποιητικά, υστερώντας ελάχιστα σε ακρίβεια σε σχέση με την απλή μέθοδο του ΠΓΕ, αλλά υπερτερώντας κατά πολύ σε ταχύτητα σύγκλισης και οικονομία υπολογιστικών πόρων.

Όροι κλειδιά: Τεχνητά Ανοσοποιητικά Συστήματα, Αρχή Επιλογής Κλώνων, Προγραμματισμός Γονιδιακής Έκφρασης, Εξόρυξη Δεδομένων, Αντιγόνα Κλάσης Δεδομένων.

Abstract

The thesis in hands focuses on Artificial Immune Systems and on their application in data mining problems, in combination with existing techniques. The natural Immune System is a very intelligent and effective pattern recognition system, which can successfully recognise and destroy almost any foreign organism that has invaded into the human body. A deeper insight into the techniques and methods utilized by the immune system could provide a number of new machine learning techniques.

In this thesis the clonal selection principle is examined and subsequently applied to two character recognition problems. This method achieves very good results in terms of convergence rate by using a small initial antibody population and a hypermutation mechanism.

This method is then coupled with Gene Expression Programming (GEP), so as to perform a data mining task. GEP is the descendant of Genetic Algorithms and Genetic Programming and eliminates their main disadvantages, though it preserves their advantageous features. In order to perform the data mining task, the notion of Data Class Antigens is introduced, which is used to represent a class of data. The rules produced are evolved by the clonal selection algorithm, to which a receptor editing step has been added. The antibodies are coded as GEP chromosomes.

The proposed hybrid technique is tested on some benchmark problems of the UCI repository, and in particular on the set of MONK problems and the Pima Indians Diabetes problem. In both problems, the results in terms of prediction accuracy are very satisfactory, albeit slightly less accurate than those obtained by a conventional GEP technique. In terms of convergence rate and computational efficiency, however, the herein proposed technique markedly outperforms the conventional GEP algorithm.

Key terms: Artificial Immune Systems, Clonal Selection Principle, Gene Expression Programming, Data Mining, Data Class Antigens.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κατάλογος Προγραμμάτων

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Τα τελευταία χρόνια η πρόοδος που σημειώνεται στις επιστήμες της Βιολογίας και της Γενετικής, έχει αποκαλύψει με πολύ λεπτομέρεια τον τρόπο λειτουργίας σημαντικών βιολογικών οργάνων και φαινομένων, με αποτέλεσμα η γνώση μας για τις διάφορες εκφράσεις της ζωής να είναι πιο πλήρης από ποτέ. Η πρόοδος αυτή, ήδη από τα πρώτα χρόνια των μεγάλων βιολογικών ανακαλύψεων, δεν έχει αφήσει αδιάφορους τους επιστήμονες των διαφόρων θετικών επιστημονικών κλάδων (μαθηματικούς, φυσικούς, μηχανικούς, κ.λπ.), οι οποίοι προσπαθούν με κάθε τρόπο, να «εμφυτεύσουν» την ευφυΐα της φύσης στις μηχανές. Μάλιστα, δεν θα ήταν υπερβολή να πούμε, ότι η πρόοδος στον τομέα των υπολογιστικών συστημάτων τα τελευταία 50 περίπου χρόνια είναι εξίσου σημαντική με την πρόοδο στην επιστήμη της Βιολογίας, επιτρέποντάς μας να μιλούμε, πλέον, για πραγματικά ευφυή συστήματα στην υπηρεσία του συγχρόνου ανθρώπου και πολιτισμού.

1.1 Βιολογικά Μοντέλα Μάθησης Μηχανών

Με τον όρο *μάθηση μηχανών* εννοούμε την κατασκευή προγραμμάτων υπολογιστών, τα οποία βελτιώνονται αυτόματα και χωρίς κάποια εξωτερική παρέμβαση (Mitchell, 1996). Η έννοια της βελτίωσης ενός προγράμματος αφορά στην βελτίωση της λύσης που προτείνει για κάποιο συγκεκριμένο πρόβλημα. Τελικός στόχος είναι μέσω της διαδικασίας εκπαίδευσης (training), το πρόγραμμα να προτείνει μία βέλτιστη λύση για το πρόβλημα. Ένα μοντέλο μάθησης μηχανών αποτελεί μία συγκεκριμένη μέθοδο κατασκευής ενός αυτο-βελτιουμένου προγράμματος υπολογιστή.

Ο αριθμός των μοντέλων μάθησης μηχανών που έχουν προταθεί από την εποχή που ξεκίνησε η έρευνα στο πεδίο της υπολογιστικής νοημοσύνης (περίπου στα μέσα της δεκατίας του ΄40) μέχρι σήμερα, είναι πραγματικά μεγάλος και θα ήταν ασύμφορο να περιγραφούν όλα. Στην συνέχεια θα περιγραφούν πολύ συνοπτικά κάποια από τα σημαντικότερα μοντέλα μάθησης μηχανών, τα οποία είναι εμπνευσμένα από αντίστοιχα βιολογικά φαινόμενα.

1.1.1 Τεχνητά Νευρωνικά Δίκτυα

Η δημιουργία των *Τεχνητών Νευρωνικών Δικτύων* ή *ΤΝΔ (Artificial Neural Networks, ANN)* οφείλεται στην διαπίστωση, ότι ο τρόπος λειτουργίας του ανθρωπίνου εγ-

κεφάλου είναι τελείως διαφορετικός από τον τρόπο λειτουργίας ενός συμβατικού ψηφιακού υπολογιστή. Ο ανθρώπινος εγκέφαλος έχει την ικανότητα να οργανώνει τα δομικά του κύτταρα, γνωστά και ως *νευρώνες*, με τέτοιο τρόπο, ώστε να μπορεί να εκπληρώνει εξαιρετικά σύνθετες εργασίες (π.χ. αναγνώριση προτύπων, αντίληψη αντικειμένων, ακριβής έλεγχος κινήσεων, κ.λπ.) σε χρόνο της τάξεως των ms. Εάν οι ίδιες εργασίες ανετίθεντο στον πιο γρήγορο υπολογιστή σήμερα, κάποιες θα διεκπαιραιώνονταν σε πολλαπλάσιο χρόνο, ενώ κάποιες άλλες θα ήταν αδύνατο ακόμα και να ολοκληρωθούν. Αυτό που δημιουργεί αυτή την θεμελιώδη διαφορά μεταξύ του ανθρωπίνου εγκεφάλου και μιας οποιαδήποτε μηχανής, είναι ότι ο πρώτος είναι ένα εξαιρετικά σύνθετο, μη γραμμικό και ταυτόχρονα παράλληλο σύστημα επεξεργασίας πληροφοριών. Ακόμη, η ιδιότητα του εγκεφάλου να δημιουργεί τους δικούς του κανόνες, για να αντιλαμβάνεται το περιβάλλον του (απόκτηση εμπειρίας), αλλά και η ικανότητά του να προσαρμόζεται σε αυτό, είναι ακόμα δύο στοιχεία που τον διαφοροποιούν-ενισχύουν σε σχέση με τις κοινές υπολογιστικές μηχανές.

Ένα ΤΝΔ είναι επομένως κάποια μηχανή υπολογισμού[1], που προσπαθεί να μοντελοποιήσει τον τρόπο που χρησιμοποιεί ο εγκέφαλος για την επίλυση ενός συγκεκριμένου προβλήματος. Ένας ορισμός του ΤΝΔ θα μπορούσε να ήταν (Aleksander και Morton, 1990):

> *Ένα νευρωνικό δίκτυο είναι ένας κατανεμημένος επεξεργαστής με πολύ μεγάλη παραλληλία, αποτελούμενος από απλές υπολογιστικές μονάδες, ο οποίος έχει την φυσική τάση, να αποθηκεύει εμπειρική γνώση και να την διαθέτει προς χρήση. Ομοιάζει με τον εγκέφαλο κατά δύο έννοιες:*
>
> 1. *Το δίκτυο αποκομίζει την γνώση από το περιβάλλον του μέσω μίας διεργασίας μάθησης.*
> 2. *Οι συνδέσεις μεταξύ των νευρώνων, γνωστές ως συναπτικά βάρη, χρησιμοποιούνται για να αποθηκεύουν την αποκτηθείσα γνώση.*

Η βασική δομική μονάδα ενός νευρωνικού δικτύου είναι το *νευρώνιο*, το οποίο είναι μία πολύ απλή υπολογιστική μονάδα με m εισόδους και μία έξοδο. Πολλά νευρώνια συνδέονται σε μία πολυεπίπεδη δομή και δημιουργούν ένα νευρωνικό δίκτυο. Οι έξοδοι ενός επιπέδου κατευθύνονται μέσω συνδέσεων στους νευρώνες του επομένου επιπέδου. Στις συνδέσεις αυτές ανατίθεται μια τιμή ή συντελεστής, που ονομάζεται *συναπτικό βάρος*, και η οποία καθορίζει το ποσοστό συνεισφοράς αυτής της σύνδεσης στην έξοδο του επομένου νευρώνα. Τέλος, κάθε νευρώνας υλοποιεί μία μη γραμμική συνάρτηση, η έξοδος της οποίας αποτελεί και την έξοδο του νευρώνα. Η είσοδος της συνάρτησης αυτής είναι το σταθμισμένο από τα συναπτικά βάρη άθροισμα των εισόδων του νευρώνα. Το τυπικό μοντέλο ενός νευρώνα παρουσιάζεται στο Σχήμα 1.1 και ορίζεται από τις σχέσεις (Haykin, 1999)

$$y_k = \varphi(u_k - b_k) \tag{1.1}$$

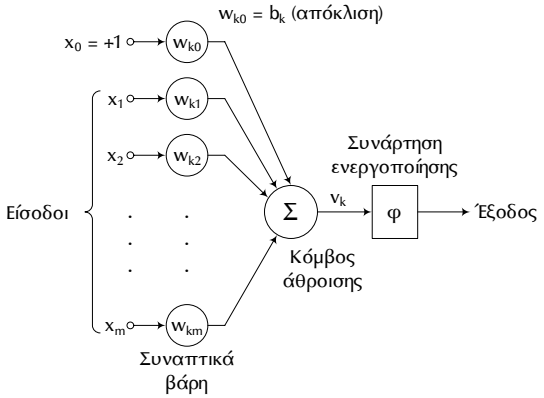
και

$$u_k = \sum_{j=1}^{m} w_{kj} x_j, \tag{1.2}$$

όπου $x_1, x_2, \ldots, x_m$ είναι τα σήματα εισόδου[2], $w_{k1}, w_{k2}, \ldots, w_{km}$ είναι τα συναπτικά βάρη του νευρώνα k, u_k είναι το σταθμισμένο άθροισμα των εισόδων, b_k είναι

[1] Εδώ δεν αναφερόμαστε σε μία σαφώς ορισμένη υλοποίηση μιας τέτοιας υπολογιστικής μηχανής, αλλά γενικά σε μία θεωρητική μηχανή, δηλ. ένα μοντέλο υπολογισμού.

[2] Μπορεί να είναι και οι έξοδοι του προηγουμένου επιπέδου, εάν αναφερόμαστε σε ένα βαθύτερο επίπεδο.



Σχήμα 1.1: Το μη γραμμικό μοντέλο ενός νευρωνίου.

μία τεχνητή απόκλιση (bias) που δίνεται στον νευρώνα k, $\varphi(\cdot)$ είναι η συνάρτηση ενεργοποίησης, και τέλος y_k είναι η έξοδος του νευρώνα k.

Η διαδικασία που χρησιμοποιείται για την εκπαίδευση του δικτύου ονομάζεται *αλγόριθμος μάθησης* και σκοπός της είναι, να μεταβάλλει τα συναπτικά βάρη του δικτύου, έτσι ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα.

Τα νευρωνικά δίκτυα έχουν μία σειρά από πλεονεκτήματα, τα οποία τα έχουν καταστήσει ιδιαίτερα δημοφιλή. Συνοπτικά αναφέρουμε την μη γραμμικότητα, την αντιστοίχιση εισόδου-εξόδου, την προσαρμοστικότητα σε νέες συνθήκες, την ανοχή στον θόρυβο (γενίκευση), κ.α. Από την άλλη, η πλήρως κατανεμημένη φύση τους θέτει και κάποιους περιορισμούς, όπως το ότι δεν μπορούν να χρησιμοποιηθούν αυτούσια σε έμπειρα συστήματα ή συστήματα εξόρυξης γνώσης, λόγω της εγγενούς αδυναμίας τους να τεκμηριώσουν την έξοδό τους, πράγμα που οφείλεται στο γεγονός, ότι η γνώση τους για το εκάστοτε πρόβλημα είναι κατανεμημένη σε ολόκληρο το δίκτυο.

Κλείνοντας την παράγραφο για τα ΤΝΔ, να αναφέρουμε ότι η μελέτη τους ξεκίνησε με την πρωτοπόρο δουλειά των McCulloch και Pitts (1943), ενώ λίγα χρόνια αργότερα σημαντική ώθηση έδωσε το Perceptron του Rosenblatt (1957). Τελευταίος σημαντικός σταθμός για την μελέτη των ΤΝΔ ήταν το 1986 με την δουλειά των Rumelhart et al., που δημιούργησαν τον *αλγόριθμο ανάστροφης μάθησης (back-propagation algorithm)*.

1.1.2 Γενετικοί Αλγόριθμοι

Οι *Γενετικοί Αλγόριθμοι* ή *ΓΑ (Genetic Algorithms, GAs)* ανήκουν σε μία γενικότερη ομάδα αλγορίθμων με το όνομα *Εξελικτικοί Αλγόριθμοι* ή *ΕΑ (Evolutionary Algorithms, EAs)*, οι οποίοι αποτελούν μία απλουστευμένη μορφή της βιολογικής εξέλιξης και της φυσικής επιλογής, όπως αυτή θεμελιώθηκε από τον Darwin (1859). Η βιολογική εξέλιξη εφαρμόστηκε για πρώτη φορά σε συστήματα υπολογιστών από τον Holland (1975), ενώ αργότερα εισηγήθηκαν νέες τεχνικές όπως ο *Γενετικός Προγραμματισμός (ΓΠ)* και ο *Προγραμματισμός Γονιδιακής Έκφρασης (ΠΓΕ)*. Οι τεχνικές αυτές περιγράφονται αναλυτικότερα στο Κεφάλαιο 4.

Οι γενετικοί αλγόριθμοι διατηρούν ένα πληθυσμό υποψηφίων λύσεων του προβλήματος, τον οποίο εξελίσσουν και αναπαράγουν. Η εξέλιξη του πληθυσμού γίνεται χρησιμοποιώντας ένα σύνολο γενετικών τελεστών, που εφαρμόζονται πάνω σε συγκεκριμένα άτομα του πληθυσμού. Τα άτομα του πληθυσμού ονομάζονται

χρωμοσώματα και είναι ακολουθίες συμβόλων από κάποιο αλφάβητο Σ (συνήθως $\Sigma = \{0, 1\}$). Κάθε χρωμόσωμα μπορεί να χωριστεί σε μικρότερες υπο-ακολουθίες συμβόλων, οι οποίες αποτελούν τα γονίδια.

Οι γενετικοί τελεστές που εφαρμόζονται στα χρωμοσώματα είναι:

Μετάλλαξη (mutation) Ο τελεστής αυτός μεταλλάσσει ένα ή περισσότερα σημεία του χρωμοσώματος, αλλάζοντας τα σύμβολα που υπάρχουν στις θέσεις αυτές.

Διασταύρωση (crossover) Ο τελεστής της διασταύρωσης ανταλλάσσει γενετικό υλικό μεταξύ δύο ή περισσοτέρων χρωμοσωμάτων (*διασταύρωση πολλαπλών γονέων–multiparent crossover*). Όταν έχουμε διασταύρωση δύο γονέων, επιλέγεται τυχαία ένα σημείο στα χρωμοσώματα και γίνεται ανταλλαγή των υπο-ακολουθιών των χρωμοσωμάτων εκατέρωθεν του σημείου. Ο τελεστής αυτός ονομάζεται και τελεστής *ανασύνθεσης (recombination)*.

Πέραν των δύο γενετικών τελεστών που αναφέραμε, οι ΓΑ αλλά και όλοι οι εξελικτικοί αλγόριθμοι βασίζονται σε μία μέθοδο επιλογής και αναπαραγωγής[3] των καλυτέρων υποψηφίων. Στην βιολογική εξέλιξη κάθε άτομο ενός πληθυσμού προσαρμόζεται με διαφορετικό τρόπο στο περιβάλλον του. Έτσι, άλλα άτομα είναι καλά προσαρμοσμένα και κάποια άλλα δεν είναι, με αποτέλεσμα η πιθανότητα επιβίωσης των τελευταίων να είναι ιδιαίτερα μικρή. Έχουμε με αυτό τον τρόπο μία φυσική επιλογή των καλύτερα προσαρμοσμένων ατόμων του πληθυσμού. Η ιδιότητα ενός ατόμου να προσαρμόζεται στο περιβάλλον του ονομάζεται *προσαρμογή (fitness)* του ατόμου. Με εντελώς παρόμοιο τρόπο, ένας ΓΑ εφαρμόζει μία *συνάρτηση προσαρμογής (fitness function)* σε όλα τα μέλη του πληθυσμού και τα κατατάσσει αναλόγως της τιμής αυτής της συνάρτησης. Στην συνέχεια, το κάθε μέλος πληθυσμού παράγει απογόνους ανάλογα την τιμή προσαρμογής του, οπότε προκύπτει ένα είδος επιλεκτικής αναπαραγωγής, όπου οι καλύτεροι υποψήφιοι αναπαράγονται, ενώ οι χειρότεροι εξαλείφονται. Η διαδικασία αυτή ονομάζεται *επιλογή (selection)*. Μόλις ολοκληρωθεί και αυτή η φάση, τότε έχει ολοκληρωθεί μία *γενιά εξέλιξης*. Γενικά, η διαδικασία της επιλογής οδηγεί ολόκληρο τον πληθυσμό σε μία τοπικά ή ολικά βέλτιστη λύση, καθ΄ ότι τείνει να εξομοιώσει όλα τα άτομα του πληθυσμού. Το φαινόμενο αυτό ονομάζεται *γενετική ολίσθηση (genetic drift)*. Αντιθέτως, οι μεταλλάξεις και οι διασταυρώσεις βοηθούν τον ΓΑ, να εξερευνήσει καλύτερα το πεδίο των υποψηφίων λύσεων και να αποφύγει τις τοπικά βέλτιστες λύσεις.

Συνοψίζοντας, ένας ΓΑ θα μπορούσε να περιγραφεί ως εξής (βλ. επίσης Σχήμα 1.2):

Αλγόριθμος ΓΑ (Γενετικός Αλγόριθμος). Αλγόριθμος μάθησης μηχανών βασισμένος στην αρχή της φυσικής επιλογής.

ΓΑ1. Αρχικοποίηση του πληθυσμού.

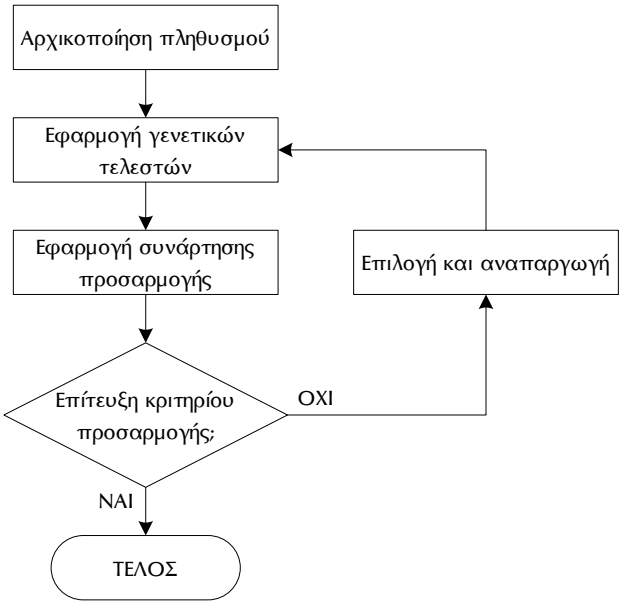
ΓΑ2. Εφαρμογή γενετικών τελεστών (διασταύρωση-μετάλλαξη).

ΓΑ3. Εφαρμογή συνάρτησης προσαρμογής στον πληθυσμό.

ΓΑ4. Εάν επετεύχθη το ζητούμενο κριτήριο προσαρμογής, τότε τέλος. Ειδ΄ άλλως συνέχισε στο Βήμα ΓΑ5.

ΓΑ5. Επιλογή των καλυτέρων υποψηφίων και αναπαραγωγή τους βάσει ενός σχήματος αναπαραγωγής. Επανάληψη από το Βήμα ΓΑ2. ∎

[3] Στην βιβλιογραφία κάποιοι συγγραφείς κατατάσσουν και την διαδικασία της επιλογής και της αναπαραγωγής στους γενετικούς τελεστές.

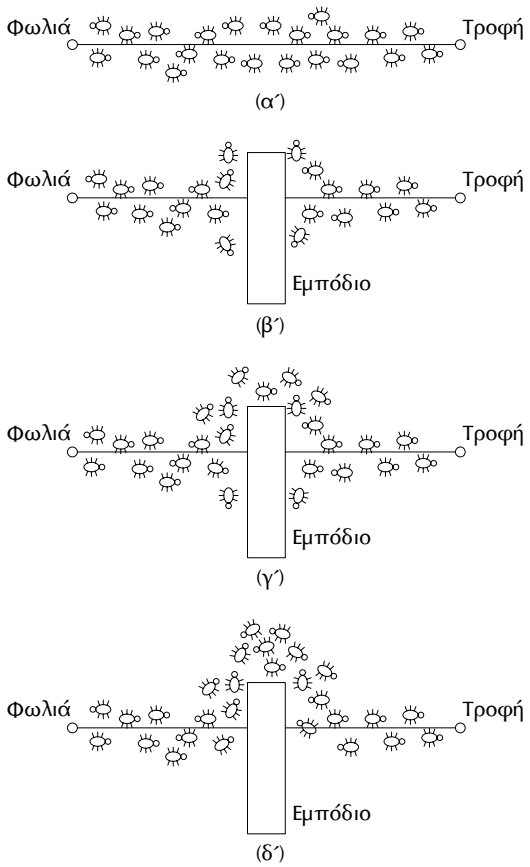


Σχήμα 1.2: Διάγραμμα ροής γενετικού αλγορίθμου.

Το μεγάλο πλεονέκτημα των γενετικών αλγορίθμων, αλλά και γενικότερα των εξελικτικών αλγορίθμων, είναι η μεγάλη ενδογενής ευελιξία τους, καθ΄ ότι δεν επιλύουν το πρόβλημα με μαθηματικό τρόπο, αλλά με βιολογικό. Οι ΕΑ έχουν την δυνατότητα να προσδιορίζουν μία βέλτιστη ή σχεδόν βέλτιστη λύση, ανεξάρτητα εάν το πρόβλημα είναι μη γραμμικό, διακριτού χρόνου, υπόκειται σε ισοτικούς ή ανισοτικούς περιορισμούς, ή ακόμα και αν είναι μη πολυωνυμικά πλήρες (NP-complete) (Τζαφέστας, 2002).

1.1.3 Αποικίες Μυρμηγκιών

Τα μυρμήγκια έχουν την ικανότητα να βρίσκουν και να ακολουθούν τον συντομότερο δρόμο από την τροφή στη φωλιά τους. Επιπλέον, μπορούν και προσαρμόζονται σε αλλαγές του περιβάλλοντός τους. Για παράδειγμα εάν ο δρόμος που ήδη ακολουθούν αποκλειστεί λόγω κάποιου εμποδίου, τότε είναι ικανά και πάλι να ανακαλύψουν το συντομότερο μονοπάτι σε σχέση με τις νέες συνθήκες. Ο τρόπος με τον οποίο τα μυρμήγκια ανακαλύπτουν και διατηρούν ένα μονοπάτι, οφείλεται σε μία ουσία που εκκρίνουν, την φερομόνη. Τα μυρμήγκια ακολουθούν πάντα το μονοπάτι με την μεγαλύτερη ποσότητα φερομόνης. Έστω λοιπόν, ότι αρχικά τα μυρμήγκια έχουν ήδη ανακαλύψει μία διαδρομή από την τροφή τους έως στην φωλιά, όπως φαίνεται στο Σχήμα 1.3α΄. Εάν κάποια χρονική στιγμή παρουσιαστεί ένα εμπόδιο, τότε τα μυρμήγκια που βρίσκονται ακριβώς μπροστά από το εμπόδιο, θα μοιραστούν εξίσου προς τις δύο κατευθύνσεις (δεξιά-αριστερά), για να αποφύγουν το εμπόδιο (Σχήμα 1.3β΄). Εκείνα όμως τα μυρμήγκια που θα ακολουθήσουν το πιο σύντομο μονοπάτι γύρω από το εμπόδιο, θα συναντήσουν γρηγορότερα το ίχνος της φερομόνης από την προηγούμενη διαδρομή και επομένως, η συνολική ποσότητα φερομόνης στην συντομότερη διαδρομή θα αυξηθεί πιο γρήγορα απ΄ ότι στην μεγαλύτερη διαδρομή. Καθώς νέα μυρμήγκια θα καταφθάνουν στο εμπόδιο, αυτά θα διαλέγουν την διαδρομή με την περισσότερη φερομόνη (Σχήμα 1.3γ΄), που εν προκειμένω είναι η συντομότερη, ώσπου τελικά, όλα τα μηρμύγκια θα ακολουθήσουν αυτή την διαδρομή (Σχήμα 1.3δ΄).



Σχήμα 1.3: Προσδιορισμός ελαχίστου μονοπατιού από τα μυρμήγκια: (α΄) αρχικό μονοπάτι, (β΄) εμφάνιση εμποδίου, (γ΄) τα μυρμήγκια προτιμούν το συντομότερο μονοπάτι, (δ΄) ολόκληρη η αποικία ακολουθεί το συντομότερο μονοπάτι.

Βάσει αυτής της λογικής οι Gambardella και Dorigo (1997) ανέπτυξαν έναν αλγόριθμο για την επίλυση του προβλήματος του πλανωδίου πωλητή (TSP). Αρχικά m συνολικά μυρμήγκια τοποθετούνται τυχαία στις πόλεις του προβλήματος. Σε κάθε βήμα τα μυρμήγκια πηγαίνουν σε μία νέα πόλη ανανεώνοντας το ίχνος φερομόνης για αυτή την πλευρά του γράφου (*τοπική ανανέωση ίχνους*). Όταν όλα τα μυρμήγκια έχουν ολοκληρώσει ένα κύκλο στο γράφο, τότε το μυρμήγκι που έκανε την μικρότερη διαδρομή, αυξάνει την ποσότητα της φερομόνης σε αυτή την διαδρομή (*καθολική ανανέωση ίχνους*). Τα μυρμήγκια επιλέγουν τον επόμενο κόμβο που θα επισκεφθούν βάσει μιας πιθανοτικής συνάρτησης, η οποία λαμβάνει υπ΄ όψιν τόσο την ποσότητα φερομόνης που υπάρχει στην ακμή μεταξύ του τρέχοντος και του μέλλοντος κόμβου, όσο και την απόσταση μεταξύ των δύο κόμβων (βάσει μιας ευρετικής συνάρτησης). Η συνάρτηση επιλογής νέου κόμβου πρέπει να δίνει στο κάθε μυρμήγκι δύο επιλογές: είτε αυτό να ακολουθήσει το μονοπάτι με την περισσότερη φερομόνη, ακολουθώντας την εμπειρία της αποικίας, είτε να ακολουθήσει ένα εντελώς καινούργιο μονοπάτι. Η καθολική ανανέωση ίχνους γίνεται αυξάνοντας την ποσότητα φερομόνης κατά ένα ποσό αντιστρόφως ανάλογο του μήκους του κύκλου. Αντιθέτως, η τοπική ανανέωση ίχνους πρέπει να γίνεται με τέτοιο τρόπο, ώστε να υπάρχει πιθανότητα να μειωθεί η ποσότητα φερομόνης μίας ακμής[4], έτσι

[4]Μπορεί να πει κανείς ότι έτσι προσομοιώνεται η σταδιακή εξάτμιση της φερομόνης.

ώστε μία αρκετά «δυνατή» ακμή, να μην προτιμάται από όλα τα μυρμήγκια, με αποτέλεσμα να εγκλωβίζεται ο αλγόριθμος σε τοπικά ακρότατα. Γενικά, μπορεί να πει κανείς ότι το σύστημα της αποικίας μυρμηγκιών είναι ένα σύστημα ενισχυτικής μάθησης, όπου οι καλύτερες ακμές συνεχώς ενδυναμώνονται–αυξάνεται η φερομόνη τους.

1.1.4 Σμήνη Σωματιδίων

Τα *σμήνη σωματιδίων (particle swarms)* αποτελούν και αυτά μία από τις τελευταίες μεθόδους στο πεδίο της μάθησης μηχανών και προτάθηκε το 1995 από τους Kennedy και Eberhart. Η μέθοδος αυτή προσπαθεί να προσομοιώσει την κοινωνική συμπεριφορά των πτηνών με σκοπό την επίλυση προβλημάτων βελτιστοποίησης (particle swarm optimization–PSO).

Με τον όρο κοινωνική συμπεριφορά των πτηνών εννοείται ο τρόπος, με τον οποίο συμπεριφέρονται και αλληλεπιδρούν τα πτηνά, όταν βρίσκονται σε ένα σμήνος. Εάν προσέξει κανείς την συμπεριφορά ενός σμήνους πτηνών, θα παρατηρήσει ότι όλες οι κινήσεις (αλλαγή κατεύθυνσης, διασπορά, ανασυγκρότηση, κ.λπ.) γίνονται ταυτόχρονα από ολόκληρο το σμήνος. Επιπλέον έχει διαπιστωθεί, ότι σε ένα σμήνος η πληροφορία-γνώση που υπάρχει για τον γύρω χώρο, την τροφή, κ.λπ. μοιράζεται από όλα τα μέλη του. Αυτό είναι μία πολύ σημαντική διαπίστωση, καθ΄ ότι δίνει μία εξελικτική «χροιά» στην κίνηση του σμήνους–πάντα θα επικρατεί η καλύτερη γνώση που έχει μέχρι στιγμής αποκομιστεί. Η έννοια αυτή αποτελεί και την βάση της βελτιστοποίησης με σμήνη σωματιδίων.

Σκοπός της μεθόδου είναι η προσαρμογή των διανυσμάτων της ταχύτητας των σωματιδίων του σμήνους, έτσι ώστε το σμήνος να κατευθυνθεί στον ζητούμενο στόχο, δηλαδή στο ολικό ελάχιστο του χώρου λύσεων. Έστω λοιπόν, ότι υπάρχει ένα σμήνος από N σωματίδια, τα οποία κινούνται στον n-διάστατο χώρο με ταχύτητες $\mathbf{v}_i$, $i = 1, \ldots, N$. Η θέση κάθε σωματιδίου δίνεται από το διάνυσμα $\mathbf{p}_i$, $i = 1, \ldots, N$, ενώ υποθέτουμε ότι κάθε σωματίδιο «θυμάται» την θέση, έστω $\mathbf{b}_i$, $i = 1, \ldots, N$, στην οποία είχε πετύχει την καλύτερη ταχύτητα. Έστω ακόμη i_b ο αριθμός του σωματιδίου με την καλύτερη ταχύτητα σε όλο το σμήνος. Ονομάζοντας τώρα τις μήτρες διαστάσεων $N \times n$ που σχηματίζονται από τα διανύσματα στήλη $\mathbf{v}_i$, $\mathbf{p}_i$ και $\mathbf{b}_i$, ως **V P**, και **B**, αντιστοίχως, η εξίσωση ανανέωσης των ταχυτήτων των μελών του σμήνους μπορεί να γραφεί ως

$$v_{ij}^{new} = v_{ij}^{old} + 2r \cdot (b_{ij} - p_{ij}) + 2r \cdot (b_{ii_b} - p_{ij}), \tag{1.3}$$

όπου r είναι τυχαία μεταβλητή που ακολουθεί την ομοιόμορφη κατανομή $U(0, 1)$, v_{ij}, p_{ij} και b_{ij} είναι στοιχεία των μητρών **V**, **P** και **B** αντιστοίχως.

Κλείνοντας, αξίζει να αναφερθεί ότι βασικό πλεονέκτημα της μεθόδου των σμηνών σωματιδίων, είναι ότι αν και χρησιμοποιεί την ιδέα των ΕΑ, είναι αρκετά πιο γρήγορη από αυτούς.

1.1.5 Τεχνητά Ανοσοποιητικά Συστήματα

Τα *Τεχνητά Ανοσοποιητικά Συστήματα* ή *ΤΑΣ (Artificial Immune Systems, AIS)*, όπως και οι αποικίες μυρμηγκιών και τα σμήνη σωματιδίων, αποτελούν και αυτά μία καινούργια μέθοδο στον τομέα της μάθησης μηχανών, που προτάθηκε πριν από περίπου 10 χρόνια (Dasgupta, 1997). Τα ΤΑΣ βασίζονται στους μηχανισμούς του ανθρωπίνου ανοσοποιητικού συστήματος, το οποίο έχει την ικανότητα, να αναγνωρίζει και να αντιμετωπίζει με επιτυχία σχεδόν κάθε παθογόνο μικροοργανισμό που εισβάλλει στον άνθρωπο. Το ανοσοποιητικό σύστημα διαθέτει επιπλέον και

μνήμη, μία ιδιαίτερα σημαντική ιδιότητα, καθ΄ ότι του επιτρέπει να αναγνωρίσει και να ανταποκριθεί πολύ πιο άμεσα σε μία εισβολή από παθογόνα, που έχουν προσβάλει και παλαιότερα τον οργανισμό.

Γενικά, ένα ΤΑΣ διατηρεί ένα πληθυσμό από *αντισώματα*, τα οποία και τροποποιεί με την πάροδο του χρόνου, έτσι ώστε να μπορεί να αναγνωρίζει επιτυχώς τα προς αναγνώριση *αντιγόνα*. Τα αντισώματα εξελίσσονται μέσω μίας διαδικασίας που λέγεται *υπερ-μετάλλαξη (hypermutation)*, ενώ ταυτόχρονα μπορούν να υπόκεινται σε ένα είδος ανασύνθεσης, όπως συμβαίνει και με τους ΓΑ. Τα αντισώματα που αναγνωρίζουν καλύτερα τα αντιγόνα διατηρούνται στην μνήμη του ΤΑΣ. Στην συνέχεια παρουσιάζεται ξανά το ίδιο αντιγόνο στον πληθυσμό, οπότε ενεργοποιούνται τα αντισώματα μνήμης μαζί με κάποια εντελώς καινούργια αντισώματα. Καθώς το αντιγόνο επανεμφανίζεται στον πληθυσμό, τα αντισώματα συνεχώς θα βελτιώνονται, ώσπου τελικά θα μπορούν να το αναγνωρίζουν πλήρως. Η διαδικασία αυτή ομοιάζει με τους ΓΑ, καθ΄ ότι και εδώ τα αντισώματα εξελίσσονται μέχρις ότου να ελαχιστοποιήσουν ή μεγιστοποιήσουν μία αντικειμενική συνάρτηση[5]. Παρ΄ όλα αυτά, η διάταξη των βημάτων της εξέλιξης είναι διαφορετική στα ΤΑΣ, γεγονός που μαζί με την υπερ-μετάλλαξη και την διαφορετική φύση της ανασύνθεσης των αντισωμάτων, τα καθιστά μία αρκετά ελκυστική εναλλακτική λύση σε σχέση με τους ΓΑ, όπως θα φανεί στα επόμενα κεφάλαια.

Ένα ΤΑΣ μπορεί να χρησιμοποιηθεί, για την ανίχνευση λαθών σε ηλεκτρονικά ψηφιακά κυκλώματα (Bradley and Tyrrell, 2002), για την ανίχνευση εισβολών σε δίκτυα υπολογιστών (Dasgupta και González, 2002; Harmer et al., 2002), κ.α. Στις περιπτώσεις αυτές, η λειτουργία του ΤΑΣ διαφέρει λίγο από την λειτουργία που μόλις περιγράφηκε. Τέτοιου είδους ΤΑΣ χρησιμοποιούν συνήθως την *αρχή της αρνητικής επιλογής (negative selection principle)* για να μπορέσουν να ξεχωρίσουν τα στοιχεία του δικού τους συστήματος από τους εισβολείς (*self-nonself discrimination*).

Περισσότερα και αναλυτικότερα στοιχεία τόσο για τα πραγματικά όσο και για τα τεχνητά ανοσοποιητικά συστήματα θα παρατεθούν στα κεφάλαια 2 και 3.

1.2 Δομή του κειμένου

Στο κεφάλαιο αυτό έγινε μία σύντομη εισαγωγή σ΄ εκείνα τα μοντέλα μάθησης μηχανών, που είναι εμπνευσμένα από βιολογικά φαινόμενα. Στην συνέχεια, στο Κεφάλαιο 2, εξετάζεται το ανθρώπινο ανοσοποιητικό σύστημα από βιολογικής πλευράς, έτσι ώστε ο αναγνώστης να εξοικειωθεί με βασικές έννοιες των ανοσοποιητικών συστημάτων. Στο Κεφάλαιο 3 εξετάζονται τα Τεχνητά Ανοσοποιητικά Συστήματα και το πώς αυτά εφαρμόζουν τις αρχές των πραγματικών ανοσοποιητικών συστημάτων, επιλύεται δε παράλληλα ένα πρόβλημα αναγνώρισης προτύπου με χρήση ενός ΤΑΣ. Συνεχίζοντας στο Κεφάλαιο 4, γίνεται μία εισαγωγή στον Προγραμματισμό Γονιδιακής Έκφρασης, ο οποίος αποτελεί την εξέλιξη του Γενετικού Προγραμματισμού. Στο Κεφάλαιο 5 παρουσιάζεται μία νέα μέθοδος που αναπτύχθηκε για εξόρυξη δεδομένων, η οποία συνδυάζει τα ΤΑΣ με τον ΠΓΕ. Τέλος, στο Παράρτημα Α παρουσιάζεται το Java Artificial Immune Framework , μία υποδομή λογισμικού, που σχεδιάστηκε και υλοποιήθηκε, με σκοπό την ανάπτυξη εφαρμογών που βασίζονται σε ΤΑΣ.

[5]Εν προκειμένω πρόκειται για την συνάρτηση διαφοράς ή ομοιότητας μεταξύ αντισωμάτων και αντιγόνων.

ΚΕΦΑΛΑΙΟ 2

Το Ανοσοποιητικό Σύστημα

Τα τελευταία χρόνια, καθώς η λειτουργία του ανθρωπίνου ανοσοποιητικού συστήματος αρχίζει να αποσαφηνίζεται, το επιστημονικό ενδιαφέρον γύρω από αυτό το πεδίο συνεχώς αυξάνεται. Το ανοσοποιητικό σύστημα αποτελείται από ένα σύνολο κυττάρων, μορίων και οργάνων, τα οποία έχουν την ικανότητα να αντιλαμβάνονται και να καταπολεμούν μολυσμένα ή δυσλειτουργούντα (καρκινικά) κύτταρα του ιδίου οργανισμού (*infectious self*), αλλά και να καταστέλουν την δράση εξωγενών μολυσματικών μικροοργανισμών (*infectious nonself*) (Von Zuben και De Castro, 1999). Χωρίς το ανοσοποιητικό σύστημα κάθε μόλυνση θα οδηγούσε αναπόφευκτα στον θάνατο. Τα κύτταρά του παρακολουθούν σε συνεχή βάση την λειτουργία του οργανισμού και είναι σε θέση να αναγνωρίσουν και να καταπολεμήσουν σχεδόν οποιονδήποτε μικροοργανισμό εισβάλλει στον οργανισμό. Παράλληλα είναι σε θέση, να αναγνωρίζουν όλα τα κύτταρα του ιδίου οργανισμού, έτσι ώστε να μην επιτίθενται σε αυτά (βλ. αυτοανοσία, Μπαρώνα-Μάμαλη et al. (1999)). Ακόμη, το ανοσοποιητικό σύστημα διαθέτει μνήμη, ενθυμούμενο όλα τα παθογόνα με τα οποία έχει έρθει σε επαφή στο παρελθόν, έτσι ώστε σε μία δεύτερη έκθεση του οργανισμού στο ίδιο παθογόνο να αντιδράσει πολύ πιο άμεσα και αποτελεσματικά. Οι βασικότερες ιδιότητες του ανοσοποιητικού συστήματος μπορούν να συνοψισθούν ως εξής (Von Zuben και De Castro, 1999):

- *μοναδικότητα*: κάθε άτομο έχει το δικό του ανοσοποιητικό σύστημα με τις δικές του δυνατότητες και ευαισθησίες.
- *αναγνώριση εισβολέων*: τα επικίνδυνα μόρια, που δεν ανήκουν στο σώμα, αναγνωρίζονται και καταστρέφονται από το ανοσοποιητικό σύστημα.
- *αναγνώριση δυσλειτουργίας*: το ανοσοποιητικό σύστημα έχει την δυνατότητα να αναγνωρίσει και να αντιδράσει σε παθογόνα, με τα οποία το σώμα δεν έχει έρθει σε επαφή προηγουμένως.
- *κατανεμημένη αναγνώριση*: τα κύτταρα του συστήματος είναι κατανεμημένα σε όλο το σώμα, και το κυριώτερο δεν υπόκεινται σε κεντρικό έλεγχο.
- *ατελής αναγνώριση (ανοχή στον θόρυβο)*: για την αντιμετώπιση ενός παθογόνου, δεν είναι απαραίτητη η απόλυτη αναγνώρισή του, επομένως το σύστημα είναι ευέλικτο.

- *ενισχυτική μάθηση και μνήμη*: το σύστημα μπορεί να «μάθει» την δομή των παθογόνων, με τα οποία έχει έρθει σε επαφή, έτσι ώστε η αντίδρασή του σε μία μελλοντική έκθεσή του στα ίδια παθογόνα, να είναι πολύ πιο γρήγορη και πολύ πιο αποδοτική.

Στην συνέχεια αυτού του κεφαλαίου αναλύεται η δομή και η λειτουργία του ανοσοποιητικού συστήματος, αρχίζοντας από μία περιγραφή της ανατομίας και των συστατικών κυττάρων του, και καταλήγωντας στους μηχανισμούς άμυνας που χρησιμοποιεί εναντίων των διαφόρων παθογόνων.

2.1 Η ανατομία του ανοσοποιητικού συστήματος

Τα όργανα του ανοσοποιητικού συστήματος είναι κατανεμημένα σε ολόκληρο το σώμα και είναι γνωστά ως *λεμφικά όργανα*, καθ΄ ότι είναι υπεύθυνα για την δημιουργία και μεταφορά των *λευκοκυττάρων* (λευκά αιμοσφαίρια) και των *λεμφοκυττάρων*, τα οποία, όπως θα φανεί στην συνέχεια, αποτελούν και τα κύρια κύτταρα που λαμβάνουν μέρος στην ανοσολογική αντίδραση του οργανισμού. Μέσα στα λεμφικά όργανα, τα λεμφοκύτταρα έρχονται σε επαφή με άλλα σημαντικά μη λεμφικά όργανα του οργανισμού, είτε κατά την διάρκεια της ωρίμανσής τους, είτε κατά την διάρκεια της ανοσολογικής αντίδρασης. Τα λεμφικά όργανα μπορούν να χωριστούν σε δύο κατηγορίες:

1. Τα *πρωτογενή* ή *κεντρικά* λεμφικά όργανα, τα οποία είναι υπεύθυνα για την παραγωγή νέων λεμφοκυττάρων, και
2. Τα *δευτερογενή* ή *περιφερειακά* λεμφικά όργανα, όπου τα λεμφοκύτταρα έρχονται σε επαφή με τον πληθυσμό των αντιγόνων.

Τα λεμφικά όργανα και οι κύριες λειτουργίες τους παρατίθενται στην συνέχεια (βλ. Σχήμα 2.1):

Αδενοειδής απόφυση και αμυγδαλές Εξειδικευμένοι λεμφικοί αδένες που περιέχουν κύτταρα για την προστασία του αναπνευστικού συστήματος.

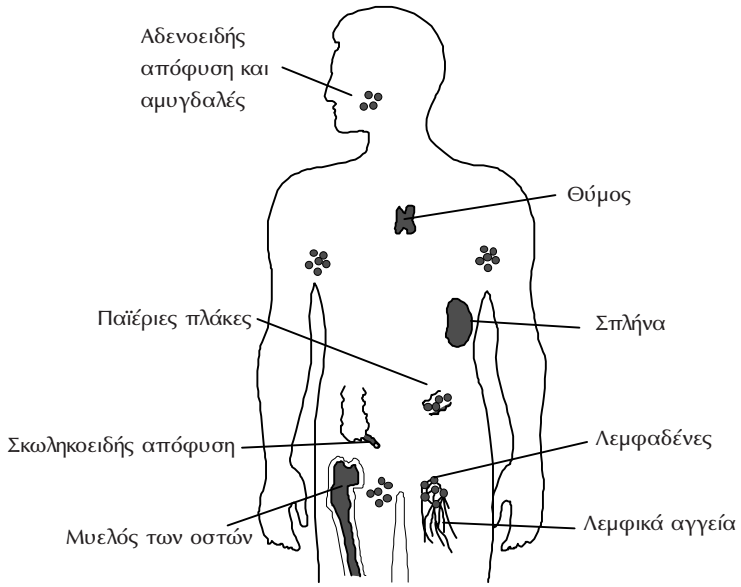
Λεμφαγγεία Αποτελούν ένα δίκτυο αγγείων, τα οποία μεταφέρουν το λεμφικό υγρό στα όργανα του ανοσοποιητικού συστήματος και στο αίμα. Το λεμφικό υγρό χρησιμοποιείται για την μεταφορά των λεμφοκυττάρων και των αντιγόνων μέσα στο σώμα.

Μυελός των οστών Ο μυελός των οστών είναι υπεύθυνος για την δημιουργία των κυττάρων του ανοσοποιητικού συστήματος (λεμφοκύτταρα και λευκοκύτταρα).

Λεμφαδένες Οι λεμφαδένες αποτελούν σημεία σύγκλισης του δικτύου των λεμφαγγείων, όπου συγκεντρώνονται τα κύτταρα του ανοσοποιητικού συστήματος.

Θύμος Κατά την πρώτη φάση της ανοσολογικής αντίδρασης (βλ. §2.4) κάποια λεμφοκύτταρα καταλήγουν στον θύμο, όπου ωριμάζουν, μεταμορφούμενα σε Τ-λεμφοκύτταρα (βλ. §2.2), τα οποία έπειτα λαμβάνουν μέρος στην ανοσολογική αντίδραση.

Σπλήνα Στην σπλήνα τα λευκοκύτταρα καταστρέφουν τους μικροοργανισμούς που έχουν εισβάλλει στο αίμα.



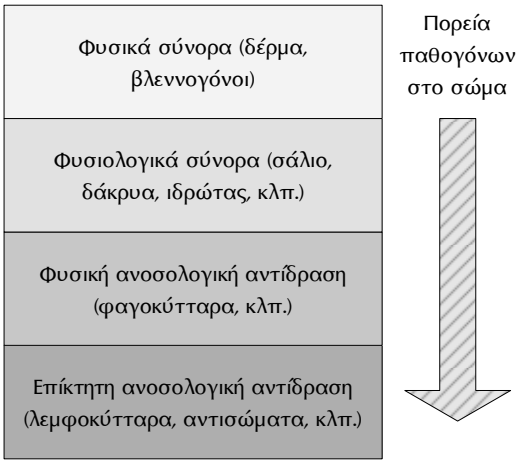
Σχήμα 2.1: Ανατομία του ανοσοποιητικού συστήματος (λεμφικά όργανα).

Σκωληκοειδής απόφυση και παϊέριες πλάκες του λεπτού εντέρου Πρόκειται για εξειδικευμένους λεμφικούς αδένες, που προορίζονται για την προστασία του πεπτικού συστήματος.

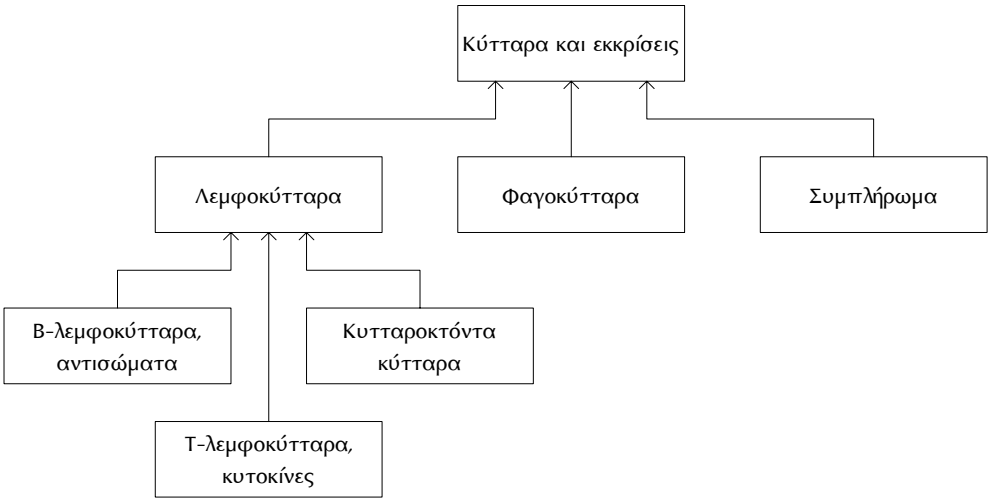
Η αρχιτεκτονική του ανοσοποιητικού συστήματος είναι πολυεπίπεδη, όπως φαίνεται στο Σχήμα 2.2. Τα όργανα που αναφέραμε, αναλαμβάνουν δράση, αφότου κάποιο παθογόνο εισέλθει στον οργανισμό, και συμβάλλουν στην φυσική και επίκτητη ανοσολογική αντίδραση ή απλά ανοσία (βλ. §2.4). Προτού όμως εισέλθει, το παθογόνο θα πρέπει να διασχίσει πρώτα τα *φυσικά* και *φυσιολογικά* σύνορα του οργανισμού (Von Zuben και De Castro, 1999).

Τα φυσικά σύνορα του οργανισμού αποτελούνται από το δέρμα, τους βλεννογόνους αδένες και το σμήγμα. Το υγιές δέρμα, λόγω της σύστασής του–πολλές και πυκνά διατεταγμένες στοιβάδες κυττάρων, αποτελεί έναν άριστο φραγμό για την είσοδο των μικροοργανισμών στο σώμα. Οι βλεννογόνοι αδένες, οι οποίοι καλύπτουν εξωτερικές κοιλότητες του σώματος, εκκρίνουν μία κολλώδη ουσία, την *βλέννα*, η οποία παγιδεύει τους περισσότερους μικροοργανισμούς που προσπαθούν να εισέλθουν στο σώμα. Ακόμη, οι βλεννογόνοι αδένες του ανωτέρου αναπνευστικού συστήματος διαθέτουν τριχίδια και βλεφαρίδες, οι οποίες κινούμενες απωθούν τους διαφόρους μικροοργανισμούς. Τέλος, το *σμήγμα*, μία λιπαρή ουσία που εκκρίνεται από τους σμηγματογόνους αδένες του δέρματος, δημιουργεί ένα προστατευτικό στρώμα στην επιφάνειά του (Μπαρώνα-Μάμαλη et al., 1999).

Τα φυσιολογικά σύνορα του οργανισμού είναι οι διάφορες ουσίες, όπως το σάλιο, ο ιδρώτας και τα δάκρυα, που εκκρίνονται από το σώμα, δημιουργώντας αντίξοες συνθήκες για την ανάπτυξη των διαφόρων μικροοργανισμών. Τέλος, όσοι μικροοργανισμοί καταφέρουν να ξεπεράσουν τους φραγμούς και να εισέλθουν στο στομάχι, θανατώνονται από το γαστρικό υγρό.



Σχήμα 2.2: Η πολυεπίπεδη δομή του ανοσοποιητικού συστήματος.



Σχήμα 2.3: Ταξινόμηση κυττάρων του ανοσοποιητικού συστήματος.

2.2 Τα κύτταρα του ανοσοποιητικού συστήματος

Το ανοσοποιητικό σύστημα αποτελείται από μία πληθώρα διαφορετικών κυττάρων, τα οποία παράγονται στο μυελό των οστών. Από εκεί μέσω του αίματος και των λεμφαγγείων κυκλοφορούν σε ολόκληρο το σώμα. Μερικά από αυτά χρησιμοποιούνται για την μη ειδική άμυνα του οργανισμού (φυσική ανοσία), ενώ κάποια άλλα εξειδικεύονται για την καταπολέμηση ενός συγκεκριμένου αντιγόνου. Με τον όρο *αντιγόνο*, εννοείται ένα οποιοδήποτε τμήμα (πρωτεΐνες, ένζυμα, μόρια από την πλασματική μεμβράνη, κλπ.) ενός μικροοργανισμού (παθογόνο) που έχει εισέλθει στον οργανισμό-ξενιστή, το οποίο χρησιμοποιείται από το ανοσοποιητικό σύστημα για την αναγνώριση και καταπολέμηση του μικροοργανισμού. Μία ταξινόμηση των κυττάρων του ανοσοποιητικού συστήματος παρουσιάζεται στο Σχήμα 2.3.

2.2.1 Λεμφοκύτταρα

Τα λεμφοκύτταρα ανήκουν στα λευκά αιμοσφαίρια (λευκοκύτταρα) και αναλαμβάνουν τον κύριο όγκο της ανοσολογικής αντίδρασης. Χωρίζονται σε δύο κατηγορίες, τα Β-λεμφοκύτταρα και τα Τ-λεμφοκύτταρα. Τα λεμφοκύτταρα παράγονται στον μυελό των οστών και διαφοροποιούνται στα διάφορα όργανα του λεμφικού συστήματος (θύμος, σπλήνα, κλπ.). Όταν ο οργανισμός δεν είναι μολυσμένος, τα λεμφοκύτταρα βρίσκονται σε μία κατάσταση ηρεμίας και απλά κυκλοφορούν στο αίμα και στους λεμφαδένες[1].

Κύριος ρόλος των Β-λεμφοκυττάρων είναι η παραγωγή και η έκκριση αντισωμάτων. Τα αντισώματα είναι πρωτεΐνες που συνδέονται σε συγκεκριμένα αντιγόνα, που βρίσκονται στην επιφάνεια του παθογόνου. Με αυτό τον τρόπο είτε εξασθενούν το παθογόνο, είτε ειδοποιούν-ενεργοποιούν άλλα κύτταρα του ανοσοποιητικού συστήματος (φαγοκύτταρα), τα οποία εν συνεχεία το καταστρέφουν. Κάθε Β-λεμφοκύτταρο παράγει μόνο ένα είδος αντισώματος, το οποίο μπορεί να αναγνωρίσει μόνο ένα συγκεκριμένο είδος αντιγόνου.

Τα Τ-λεμφοκύτταρα ρυθμίζουν την συμπεριφορά άλλων κυττάρων και παράλληλα επιτίθενται άμεσα στα μολυσμένα κύτταρα του οργανισμού. Ονομάζονται έτσι, διότι ωριμάζουν στον θύμο (thymus). Υπάρχουν διάφορα ήδη Τ-λεμφοκυττάρων, τα οποία περιγράφονται στην συνέχεια:

Βοηθητικά Τ-λεμφοκύτταρα Τα λεμφοκύτταρα αυτά φροντίζουν για την ενεργοποίηση των Β-λεμφοκυττάρων, άλλων Τ-λεμφοκυττάρων, των μακροφάγων, κ.α. Είναι επίσης γνωστά ως Τ4-λεμφοκύτταρα ή CD4.

Κυτταροτοξικά Τ-λεμφοκύτταρα Τα κυτταροτοξικά λεμφοκύτταρα επιτίθενται και καταστρέφουν, εκκρίνοντας τοξικές ουσίες, τα ξένα για τον οργανισμό κύτταρα, τα καρκινικά, αλλά και τα κύτταρα που έχουν μολυνθεί από κάποιον ιό. Είναι επίσης γνωστά και ως Τ8-λεμφοκύτταρα.

Κατασταλτικά Τ-λεμφοκύτταρα Σκοπός αυτών των λεμφοκυττάρων είναι να καταστείλουν την ανοσολογική αντίδραση, μόλις εξαλειφθεί το αίτιο που την προκάλεσε. Τα λεμφοκύτταρα αυτά είναι ζωτικής σημασίας, διότι διαφορετικά η ανοσολογική αντίδραση του οργανισμού θα ήταν ανεξέλεγκτη, με αποτέλεσμα να έχουμε αλλεργικές αντιδράσεις και αυτοάνοσα νοσήματα.

Γενικά, τα Τ-λεμφοκύτταρα ρυθμίζουν την ανοσολογική αντίδραση εκκρίνοντας ουσίες, τις κυτοκίνες ή πιο συγκεκριμένα τις λεμφοκίνες, οι οποίες ειδοποιούν και ενεργοποιούν άλλα κύτταρα, ενώ παράλληλα μπορούν να εξοντώσουν και τα κύτταρα στόχους (Von Zuben και De Castro, 1999).

Στα λεμφοκύτταρα ανήκει και άλλη μία κατηγορία θανατηφόρων κυττάρων, τα *κυτταροκτόνα κύτταρα (natural killer cells–NK cells).* Τα κύτταρα αυτά διαθέτουν κόκκους γεμάτους ισχυρά χημικά και επιτίθενται συνήθως σε καρκινικά κύτταρα. Παράλληλα, συμβάλλουν και στην ρύθμιση της ανοσολογικής αντίδρασης, εκκρίνοντας μεγάλες ποσότητες λεμφοκινών.

2.2.2 Φαγοκύτταρα

Τα φαγοκύτταρα ανήκουν στα λευκά αιμοσφαίρια και έχουν την ικανότητα να εγκλωβίζουν και να διασπούν ξένα κύτταρα ή σωματίδια. Κάποια φαγοκύτταρα

[1]Αυτά τα λεμφοκύτταρα είναι στην πραγματικότητα Β-λεμφοκύτταρα μνήμης, όπως θα φανεί στην συνέχεια.

έχουν την ικανότητα να παρουσιάζουν στα Τ-λεμφοκύτταρα αντιγόνα, ενεργοποιώντας με αυτό τον τρόπο την επίκτητη ανοσολογική αντίδραση (βλ. §2.4). Τα φαγοκύτταρα διακρίνονται στα μακροφάγα και στα κοκκιώδη.

Τα μακροφάγα μετακινούνται μέσα στο αίμα και στους λεμφαδένες. Μάλιστα, έχουν την δυνατότητα να διαπερνούν τα τοιχώματα των αιμοφόρων αγγείων, με αποτέλεσμα να καταφθάνουν γρήγορα στο σημείο της μόλυνσης. Γενικά, τα μακροφάγα είναι κύτταρα με πολλές λειτουργίες, ανάμεσα στις οποίες είναι η παρουσίαση αντιγόνων στα λεμφοκύτταρα και η καταστροφή των νεκρών λεμφοκυττάρων ύστερα από μία φλεγμονή.

Τέλος, τα κοκκιώδη είναι κύτταρα που διαθέτουν κόκκους γεμάτους με ισχυρά χημικά (όπως και τα κυτταροκτόντα κύτταρα), τα οποία απελευθερώνουν και καταστρέφουν τους ξένους μικροοργανισμούς. Τα κοκκιώδη λαμβάνουν κυρίως μέρος στην έμφυτη ανοσολογική αντίδραση.

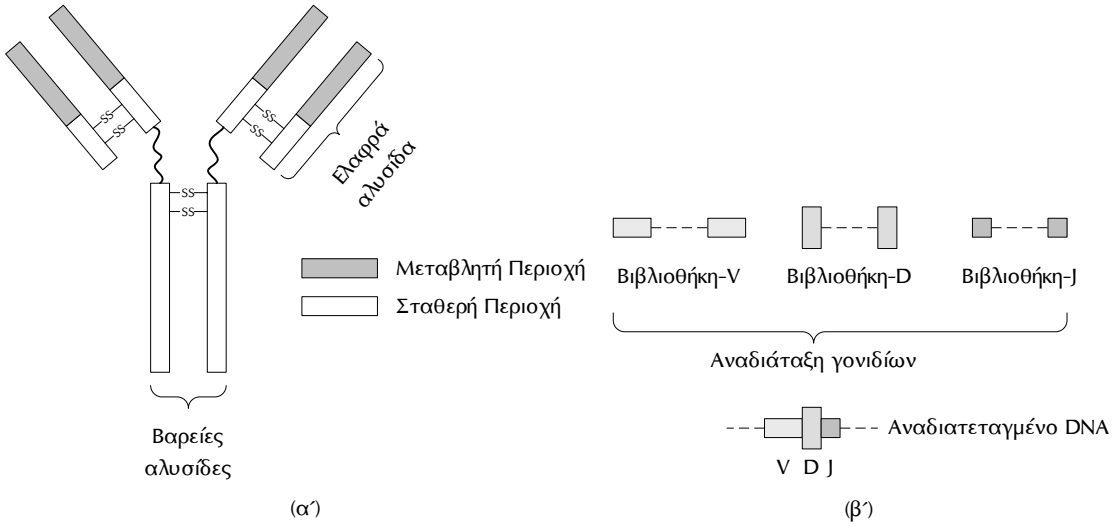
2.2.3 Το συμπλήρωμα

Το συμπλήρωμα πρόκειται για ένα σύμπλεγμα 20–25 πρωτεϊνών, οι οποίες βρίσκονται στο πλάσμα του αίματος και «συμπληρώνουν» την λειτουργία των αντισωμάτων. Όταν ο οργανισμός έχει μολυνθεί, τότε στο σώμα κυκλοφορούν αντισώματα. Τα αντισώματα αντιδρούν με το συμπλήρωμα (μέσω του σταθερού τους τμήματος) και το ενεργοποιούν. Αυτό έχει σαν αποτέλεσμα, οι πρωτεΐνες του συμπληρώματος να δημιουργήσουν μία αλυσωτή αντίδραση, το αποτέλεσμα της οποίας είναι ένα άλλο σύμπλεγμα πρωτεϊνών, το οποίο καταστρέφει τα τοιχώματα του εισβολέα, καθιστώντας τον εύκολη λεία για τα μακροφάγα.

2.3 Το Αντίσωμα

Τα αντισώματα αποτελούν τα κυριώτερα μόρια του ανοσοποιητικού συστήματος, καθ΄ ότι συμβάλλουν με καθοριστικό τρόπο στην αναγνώριση και εξουδετέρωση των αντιγόνων. Το *αντίσωμα (antibody)* ή *ανοσοσφαιρίνη (immunoglobulin)* είναι ένα μεγάλο πρωτεϊνικό μόριο, το οποίο αποτελείται από 4 πολυπεπτιδικές αλυσίδες, που συνδέονται μεταξύ τους με ισχυρούς ομοιοπολικούς δεσμούς, και σχηματίζουν μία δομή σχήματος **Y** (Σχήμα 2.4α΄). Οι δύο μεγαλύτερες αλυσίδες ονομάζονται *βαρείες αλυσίδες*, ενώ οι δύο μικρότερες ονομάζονται *ελαφρές αλυσίδες*. Κάθε αλυσίδα αποτελείται από δύο περιοχές: την σταθερή περιοχή (C-region) και την μεταβλητή περιοχή (V-region). Η μεταβλητή περιοχή είναι υπεύθυνη για την πρόσδεση του αντισώματος στο αντιγόνο, καθ΄ ότι μπορεί και σχηματίζει ένα είδος θύλακα γύρω από αυτό. Αντιθέτως, η σταθερή περιοχή δεν συμμετέχει στην αναγνώριση του αντιγόνου, αλλά μπορεί να προκαλέσει ένα σύνολο άλλων λειτουργιών του ανοσοποιητικού μηχανισμού, όπως είναι η ενεργοποίηση του συμπληρώματος (βλ. §2.2.3). Στην πραγματικότητα, η μεταβλητή περιοχή του αντισώματος δεν συνδέεται εξολοκλήρου στο αντιγόνο, αλλά μπορεί να χωριστεί περαιτέρω σε υπο-περιοχές, οι οποίες αποτελούν και τα ακριβή σημεία σύνδεσης αντισώματος-αντιγόνου (*antigen binding sites*). Οι περιοχές αυτές ονομάζονται *παράτοπα*. Αντιστοίχως, ένα αντιγόνο αποτελείται και αυτό από υπο-περιοχές (*antibody binding sites*), στις οποίες μπορούν και συνδέονται συγκεκριμένα παράτοπα. Οι περιοχές αυτές ονομάζονται *επίτοπα*. Οι έννοιες των παρατόπων και των επιτόπων χρησιμοποιούνται κατά την παρουσίαση του μοντέλου του χώρου σχήματος (§3.2).

Η γενετική πληροφορία για την δημιουργία των αντισωμάτων βρίσκεται στο DNA των φυλετικών κυττάρων του οργανισμού και μεταφέρεται από γενεά σε γενεά. Έχει διαπιστωθεί όμως, ότι η πληροφορία αυτή δεν είναι αποθηκευμένη



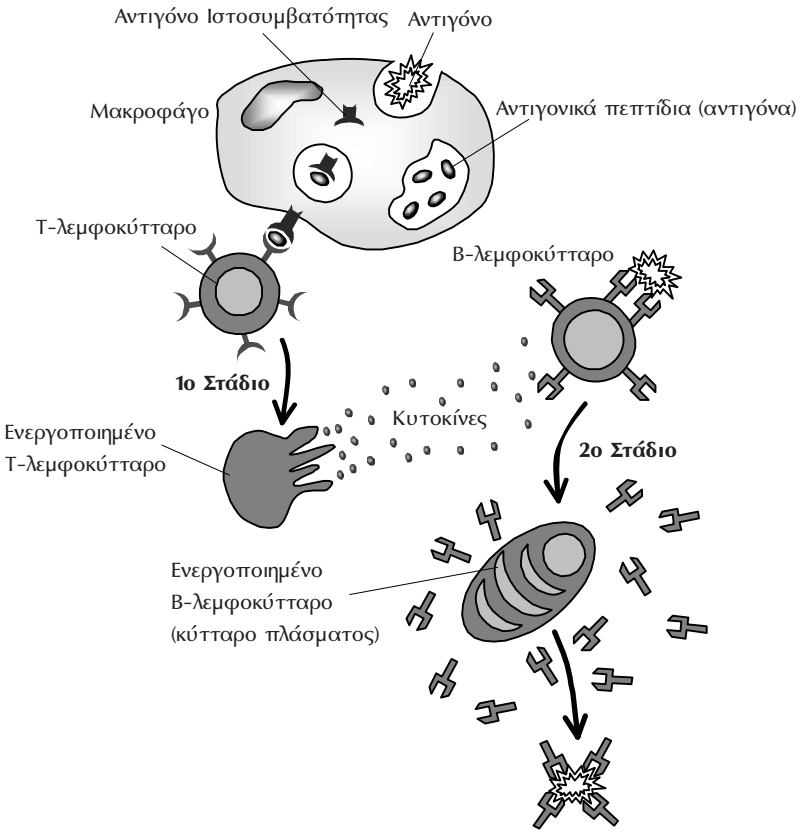
Σχήμα 2.4: (α΄) Η δομή του αντισώματος. Οι πολυπεπτιδικές αλυσίδες που το αποτελούν, συνδέονται με ισχυρούς δισουλφιδικούς δεσμούς. (β΄) Η ανασύνθεση V(D)J, που οδηγεί στην διαμόρφωση της μεταβλητής περιοχής του αντισώματος. Ακριβώς ένα γονίδιο χρησιμοποιείται από κάθε βιβλιοθήκη.

σε ένα μόνο γονίδιο, όπως συμβαίνει γενικά με τις πρωτεΐνες. Αντίθετα, για την κατασκευή ενός αντισώματος χρησιμοποιείται η πληροφορία που υπάρχει σε 3 διαφορετικές βιβλιοθήκες γονιδίων, γνωστές με την ονομασία *βιβλιοθήκες V, D και J*. Όταν δημιουργείται ένα λεμφοκύτταρο, επιλέγονται γονίδια και από τις 3 βιβλιοθήκες, σχηματίζοντας ένα μοναδικό γονίδιο στο DNA του λεμφοκυττάρου, το οποίο στην συνέχεια μεταφράζεται στο αντίστοιχο αντίσωμα. Η διαδικασία αυτή ονομάζεται *ανασύνθεση V(D)J (V(D)J recombination)*. Αφότου δημιουργηθεί το DNA του λεμφοκυττάρου, εισάγονται μεταλλάξεις με υψηλό ρυθμό (*υπερ-μετάλλαξη*), διαφοροποιώντας ακόμα περισσότερο την δομή του αντισώματος.

Οι δύο αυτοί μηχανισμοί διαφοροποίησης των αντισωμάτων δρουν συμπληρωματικα και βρίσκονται κάτω από αυστηρό έλεγχο κατά την ανάπτυξη των Β-λεμφοκυττάρων (Von Zuben και De Castro, 1999). Η ανασύνθεση των γονιδίων των αντισωμάτων επιτρέπει την δημιουργία μίας μεγάλης ποικιλίας αντισωμάτων, εκ των οποίων λίγα είναι αυτά που αναγνωρίζουν καλά κάποιο συγκεκριμένο αντιγόνο. Με την διαδικασία της υπερ-μετάλλαξης όμως, τα αντισώματα εξελίσσονται, βελτιώνοντας τον τρόπο σύνδεσής τους με τα αντίστοιχα αντιγόνα. Με άλλα λόγια, η ανασύνθεση V(D)J προσφέρει ποικιλία και διαφορετικότητα στον πληθυσμό των αντισωμάτων, ενώ η υπερ-μετάλλαξη εξειδικεύει την ανοσολογική αντίδραση. Οι δύο αυτές διαδικασίες θα συζητηθούν αναλυτικότερα στο Κεφάλαιο 3, καθ΄ ότι αποτελούν βασικές παραμέτρους ενός τεχνητού ανοσοποιητικού συστήματος.

2.4 Μηχανισμοί άμυνας του ανοσοποιητικού συστήματος

Το ανοσοποιητικό σύστημα, όπως ήδη ανεφέρθη, παρουσιάζει μία πολυεπίπεδη αρχιτεκτονική (βλ. Σχήμα 2.2). Ένας μικροοργανισμός για να εισέλθει στο σώμα, θα πρέπει πρώτα να περάσει τα φυσικά και φυσιολογικά σύνορα του οργανισμού (βλ. §2.1). Εάν το καταφέρει, τότε έρχεται αντιμέτωπος αρχικά με την μη ειδική



Σχήμα 2.5: Τα δύο πρώτα στάδια της επίκτητης ανοσολογικής αντίδρασης.

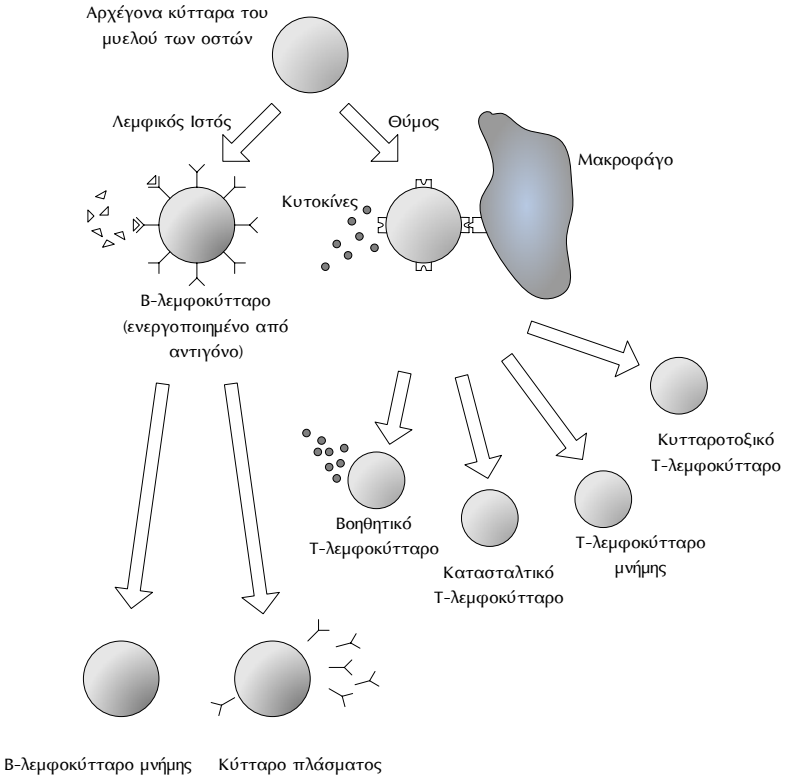
άμυνα του οργανισμού (φυσική ανοσία) και στην συνέχεια με την ειδική άμυνα του οργανισμού (επίκτητη ανοσία).

Η *φυσική ανοσία* ή *σύμφυτη ανοσία* υπάρχει στον οργανισμό από την στιγμή που θα γεννηθεί, και έχει την δυνατότητα να αναγνωρίζει και να καταστρέφει συγκεκριμένα μικρόβια. Η δράση της βασίζεται κυρίως στα φαγοκύτταρα και στα κυτταροκτόνα κύτταρα. Τα κύτταρα αυτά διαθέτουν στην επιφάνειά τους ειδικούς υποδοχείς, οι οποίοι μπορούν και αναγνωρίζουν συγκεκριμένες μοριακές δομές που υπάρχουν σε κάποια μικρόβια. Βασική ιδιότητα των κυττάρων αυτών είναι, ότι δεν συγχέουν τα κύτταρα του ιδίου οργανισμού με τα κύτταρα των μικροβίων, αποφεύγοντας έτσι πιθανούς τραυματισμούς υγιών κυττάρων. Σημαντικό ρόλο στην φυσική ανοσία παίζει επίσης το συμπλήρωμα (§2.2.3), καθώς και άλλες ουσίες που δημιουργούν αντίξοες συνθήκες για την ανάπτυξη των μικροοργανισμών. Τέλος, στην φυσική ανοσία συγκαταλέγεται και η αύξηση της θερμοκρασίας του σώματος ύστερα από μία μόλυνση (πυρετός), καθ΄ ότι επιβαρύνει την επιβίωση των μικροβίων.

Η *επίκτητη ανοσολογική* αντίδραση ή *ανοσία* έπεται της φυσικής ανοσολογικής αντίδρασης και ενεργοποιείται από αυτή. Ολοκληρώνεται σε τρία στάδια:

Στάδιο 1. [Ενεργοποίηση Τ4-λεμφοκυττάρων] Όταν ένα παθογόνο εισέλθει στον οργανισμό, ενεργοποιούνται πρώτα τα μακροφάγα[2], τα οποία λόγω των

[2]Γενικά ενεργοποιούνται και άλλα κύτταρα που έχουν την ικανότητα να παρουσιάζουν αντιγόνα



Σχήμα 2.6: Η διαφοροποίηση των λεμφοκυττάρων.

υποδοχέων που έχουν στην επιφάνειά τους, συνδέονται στο παθογόνο, το εγκλωβίζουν και το διασπούν. Στην συνέχεια τμήματα του παθογόνου (*αντιγονικά πεπτίδια*) συνδέονται με μία πρωτεΐνη του μακροφάγου, η οποία είναι χαρακτηριστική για κάθε άτομο και ονομάζεται *σύμπλεγμα* ή *αντιγόνο ιστοσυμβατότητας* (*morphohistocompatibility complex–MHC*), και εκτίθενται στην επιφάνειά του. Τα βοηθητικά Τ-λεμφοκύτταρα (Τ4) αναγνωρίζουν τον συνδυασμό του αντιγονικού πεπτιδίου και του αντιγόνου ιστοσυμβατότητας, ενεργοποιούνται και αρχίζουν να εκκρίνουν ειδικές ουσίες, τις λεμφοκίνες.

Στάδιο 2. [Ενεργοποίηση Β-λεμφοκυττάρων–*χυμική ανοσία*] Οι λεμφοκίνες που εκκρίνονται από τα Τ-λεμφοκύτταρα, ενεργοποιούν τα Β-λεμφοκύτταρα, τα οποία αρχίζουν να πολλαπλασιάζονται και να διαφοροποιούνται, δημιουργώντας *κύτταρα πλάσματος (plasma cells)*. Τα κύτταρα πλάσματος εκκρίνουν τα αντισώματα, τα οποία προσκολλώνται στα αντιγόνα, αδρανοποιούν τον παθογόνο μικροοργανισμό και παράλληλα συμβάλλουν στην αναγνώρισή του από τα μακροφάγα, τα οποία στην συνέχεια τον καταστρέφουν. Ακόμη, τα Β-λεμφοκύτταρα σε αντίθεση με τα Τ-λεμφοκύτταρα, μπορούν να ενεργοποιηθούν και από ελεύθερα αντιγόνα (μη συνδεδεμένα με αντιγόνα ιστοσυματότητας). Τέλος, δεν διαφοροποιούνται όλα τα Β-λεμφοκύτταρα σε κύτταρα πλάσματος, αλλά κάποια από αυτά μετατρέπονται σε κύτταρα μνήμης (*Β-λεμφοκύτταρα μνήμης*), τα οποία παραμένουν ανενεργά. Τα κύτταρα μνήμης ενεργοποιούνται

(*Antigen Presenting Cells–APCs*).

αμέσως σε περίπτωση που το ίδιο αντιγόνο εμφανιστεί ξανά στον οργανισμό, οδηγώντας έτσι σε μία πολύ αμεσότερη αντίδραση (*δευτερογενής ανοσολογική αντίδραση*)[3]. Γενικά, η επανειλημμένη έκθεση του οργανισμού στο ίδιο αντιγόνο οδηγεί σε βελτίωση των Β-λεμφοκυττάρων μνήμης. Το φαινόμενο αυτό ονομάζεται *ωρίμανση σύνδεσης (affinity maturation)*, και αναλύεται περισσότερο στο Κεφάλαιο 3.

Στάδιο 3. [Ενεργοποίηση Τ-λεμφοκυττάρων–*κυτταρική ανοσία*] Τα βοηθητικά Τ-λεμφοκύτταρα, ταυτόχρονα με τα Β-λεμφοκύτταρα, ενεργοποιούν και τα Τ8-λεμφοκύτταρα (κυτταροτοξικά), τα οποία καταστρέφουν τα ξένα κύτταρα, που έχουν εισβάλει στον οργανισμό (βλ. §2.2.1). Τέλος, μόλις αντιμετωπισθεί η μόλυνση, ενεργοποιούνται τα κατασταλτικά Τ-λεμφοκύτταρα, τα οποία καταστέλλουν την ανοσολογική αντίδραση. ∎

Στο Σχήμα 2.5 παρουσιάζονται τα δύο πρώτα στάδια της επίκτητης ανοσολογικής αντίδρασης, ενώ στο Σχήμα 2.6 παρουσιάζεται σχηματικά ο τρόπος διαφοροποίησης των λεμφοκυττάρων.

[3]Η αντίδραση του οργανισμού στην πρώτη εμφάνιση ενός αντιγόνου λέγεται *πρωτογενής ανοσολογική αντίδραση*.

ΚΕΦΑΛΑΙΟ 3

Τεχνητά Ανοσοποιητικά Συστήματα

Ένα *Τεχνητό Ανοσοποιητικό Σύστημα* ή *ΤΑΣ* (*Artificial Immune System, AIS*) αποτελεί ένα υπολογιστικό μοντέλο, το οποίο βασίζεται σε ιδέες εμπνευσμένες από την λειτουργία και την δομή του πραγματικού ανοσοποιητικού συστήματος. Οι Von Zuben και De Castro (1999), μάλιστα, εισάγουν τον όρο της *ανοσολογικής μηχανικής (immune engineering)*, ωστόσο δεν ορίζουν ένα σαφή διαχωρισμό μεταξύ ΤΑΣ και ανοσολογικής μηχανικής, αλλά ούτε και ταυτίζουν τις δύο έννοιες. Η ανοσολογική μηχανική είναι μία διαδικασία μετα-σύνθεσης, η οποία χρησιμοποιεί την πληροφορία που υπάρχει σε ένα δεδομένο πρόβλημα, για να ορίσει ένα εργαλείο λύσης, και στην συνέχεια να το εφαρμόσει για να επιλύσει το πρόβλημα (Von Zuben και De Castro, 1999). Με άλλα λόγια, θα μπορούσε να πει κανείς ότι η ανοσολογική μηχανική παρέχει τρόπους επίλυσης δυσκόλων προβλημάτων βασισμένους σε αρχές του πραγματικού ανοσοποιητικού συστήματος, ενώ τα ΤΑΣ αποτελούν μία εφαρμογή αυτών των τροπων.

Ο ορισμός και η ανάπτυξη ενός πλήρους ΤΑΣ περιλαμβάνει, γενικά, μία πληθώρα θεμάτων, μεταξύ των οποίων είναι:

- υβριδικές δομές και αλγόριθμοι, οι οποίοι λαμβάνουν υπ΄ όψιν τούς μηχανισμούς του ανοσοποιητικού συστήματος.
- υπολογιστικοί αλγόριθμοι βασισμένοι σε αρχές του ανοσοποιητικού συστήματος, όπως είναι η κατανεμημένη επεξεργασία, η αρχή της επιλογής των κλώνων και η θεωρία του ανοσοποιητικού δικτύου.
- βελτιστοποίηση βασισμένη στην ανοσία, μάθηση, αυτο-οργάνωση, τεχνητή ζωή (artificial life), μοντέλα γνώσης, συστήματα πολλών πρακτόρων (multiple-agent systems), σχεδίαση και χρονοπρογραμματισμός, αναγνώριση προτύπων και ανίχνευση δυσλειτουργίας (anomaly detection).
- εργαλεία ανοσολογικής μηχανικής.

Σε αυτό το κεφάλαιο θα συζητηθούν κάποιες βασικές αρχές των ανοσοποιητικών συστημάτων, οι οποίες εφαρμόζονται στα ΤΑΣ. Στην συνέχεια παρουσιάζεται ένας αλγόριθμος βασισμένος σε αυτές τις αρχές, ο οποίος εφαρμόζεται στην αναγνώριση ψηφιακών χαρακτήρων. Ο αλγόριθμος αυτός αποτελεί και την βάση του αλγορίθμου που παρουσιάζεται στο Κεφάλαιο 5, ο οποίος μπορεί να επιλύει προβλήματα εξόρυξης γνώσης από σύνολα δεδομένων.

3.1 Η αρχή της επιλογής των κλώνων

Η *αρχή* ή *θεωρία της επιλογής των κλώνων (clonal selection principle/theory)* αναφέρεται στον αλγόριθμο, τον οποίο χρησιμοποιεί το ανοσοποιητικό σύστημα, προκειμένου να αντιδράσει στην εμφάνιση ενός αντιγόνου. Προτάθηκε το 1959 από τον Burnet. Θεμελιώνει την ιδέα, ότι μόνο τα λεμφοκύτταρα που αναγνωρίζουν καλύτερα το αντιγόνο επιλέγονται για να πολλαπλασιαστούν. Η αρχή της επιλογής των κλώνων εφαρμόζεται, τόσο στα Β-λεμφοκύτταρα, όσο και στα Τ-λεμφοκύτταρα.

Όταν ένα αντιγόνο παρουσιαστεί στον οργανισμό, τότε ενεργοποιούνται τα βοηθητικά Τ-λεμφοκύτταρα, τα οποία μέσω των ουσιών που εκκρίνουν ενεργοποιούν τα Β-λεμφοκύτταρα (βλ. §2.4). Τα Β-λεμφοκύτταρα άπαξ και ενεργοποιηθούν αρχίζουν να πολλαπλασιάζονται και να παράγουν κλώνους[1]. Μόλις η διαδικασία της διαίρεσης τελειώσει, τα Β-λεμφοκύτταρα ωριμάζουν και αρχίζουν να εκκρίνουν αντισώματα σε πολύ μεγάλες ποσότητες (κύτταρα πλάσματος). Το κάθε Β-λεμφοκύτταρο εκκρίνει αντισώματα που μπορούν να συνδέονται-αναγνωρίζουν μόνο ένα συγκεκριμένο αντιγόνο. Κάποια από τα Β-λεμφοκύτταρα, που ενεργοποιούνται κατά την φάση της ανοσολογικής αντίδρασης, δεν διαφοροποιούνται σε κύτταρα πλάσματος, αλλά σε κύτταρα μνήμης, τα οποία παραμένουν ανενεργά, μέχρις ότου το ίδιο αντιγόνο εμφανιστεί ξανά στον οργανισμό. Μόλις κάτι τέτοιο συμβεί, αυτά τα κύτταρα μνήμης ενεργοποιούνται άμεσα και αρχίζουν να εκκρίνουν αντισώματα.

Η αρχή της επιλογής των κλώνων (Σχήμα 3.1) διέπει την διαδικασία παραγωγής των Β-λεμφοκυττάρων, και επομένως των αντισωμάτων, και υπαγορεύει τα εξής:

1. Τα νέα κύτταρα είναι κλώνοι των γονέων τους και υπόκεινται σε μία διαδικασία σωματικής μετάλλαξης[2] με υψηλό ρυμθό (υπερ-μετάλλαξη).

2. Τα νέα κύτταρα που στρέφονται[3] κατά του ιδίου οργανισμού καταστρέφονται.

3. Πολλαπλασιασμός και διαφοροποίηση κατά την επαφή των ωρίμων κυττάρων με τα αντιγόνα.

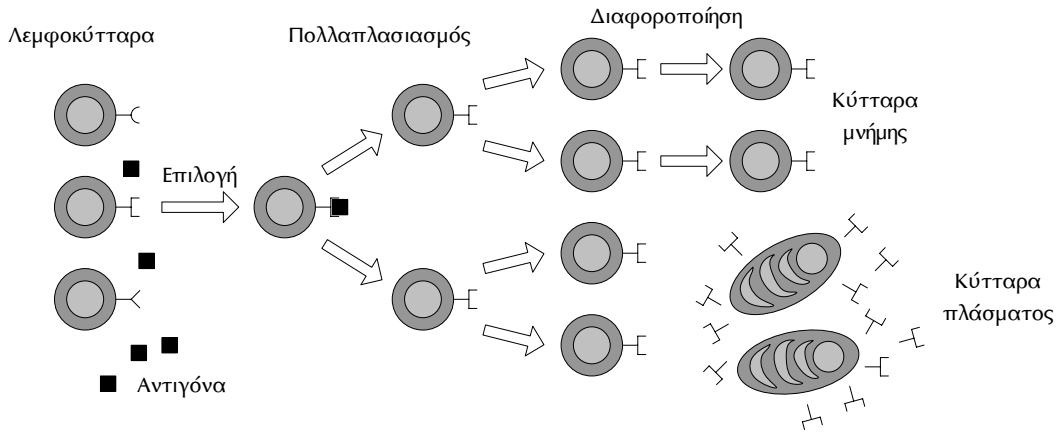
4. Η παραμονή κλώνων που στρέφονται κατά του ιδίου οργανισμού, ακόμα και έπειτα από την φάση της καταστροφής τους, είναι η αιτία των αυτοάνοσων νοσημάτων.

Ο τρόπος επιλογής των καλυτέρων Β-λεμφοκυττάρων γίνεται με τον ακόλουθο τρόπο: όταν παρουσιαστεί το αντιγόνο στον οργανισμό, κάποια από τα Β-λεμφοκύτταρα που έχουν κατάλληλους υποδοχείς προσδένονται στο αντιγόνο. Η ποιότητα της σύνδεσης λεμφοκυττάρου-αντιγόνου ποικίλει· άλλα λεμφοκύτταρα αναγνωρίζουν καλύτερα το αντιγόνο, ενώ άλλα όχι. Εν τω μεταξύ τα Τ-λεμφοκύτταρα προσδένονται και αυτά στο αντιγόνο, και αρχίζουν να εκκρίνουν τις κυτοκίνες, οι οποίες ενεργοποιούν τα Β-λεμφοκύτταρα. Όμως δεν ενεργοποιούνται όλα τα Β-λεμφοκύτταρα στον ίδιο βαθμό, αντιθέτως η ενεργοποίησή

[1] Τα κύτταρα πολλαπλασιάζονται με *μίτωση*, που στην ουσία πρόκειται για την διαδικασία διαίρεσης του κυττάρου σε δύο πανομοιότυπα με το αρχικό κύτταρα κλώνους.

[2] *Σωματική* χαρακτηρίζεται η μετάλλαξη που συμβαίνει στα σωματικά κύτταρα. Επειδή τα λεμφοκύτταρα είναι σωματικά κύτταρα, γι΄ αυτό τον λόγο οι μεταλλάξεις που συμβαίνουν σε αυτά χαρακτηρίζονται ως σωματικές. Στο εξής θα αναφέρουμε απλά τον όρο *μετάλλαξη*, καθ΄ ότι ασχολούμαστε μόνο τέτοιου είδους μεταλλάξεις.

[3] Τα κύτταρα αυτά έχουν αναπτύξει υποδοχείς ή αντισώματα που αναγνωρίζουν κύτταρα ή μόρια του ιδίου οργανισμού, με αποτέλεσμα να καταστρέφουν τα υγιή κύτταρά του.



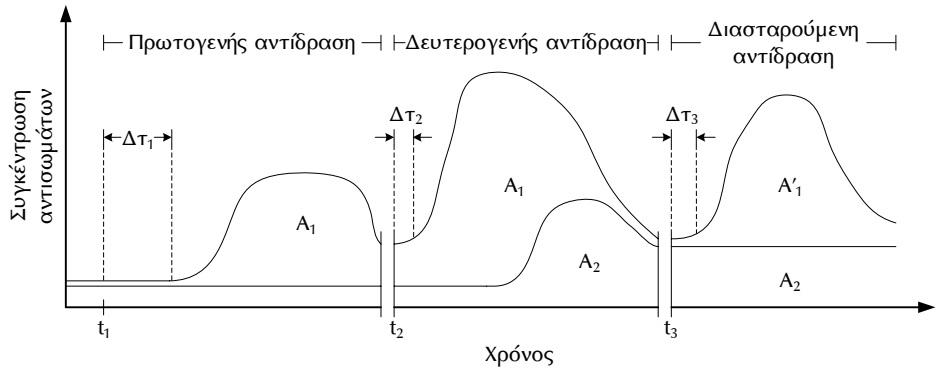
Σχήμα 3.1: Η αρχή της επιλογής των κλώνων.

τους είναι αντίστοιχη της ποιότητας της σύνδεσής τους με το αντιγόνο. Όσο πιο καλή είναι η σύνδεση, τόσο περισσοτέρους κλώνους θα παράγουν, ενώ κάποια κύτταρα μπορεί να μην διαιρεθούν καθόλου. Με αυτό τον τρόπο επομένως, επιλέγονται πάντα τα καλύτερα Β-λεμφοκύτταρα, για να παράγουν αντισώματα, ενώ τα χειρότερα είτε εκλείπουν, είτε καταστρέφονται άμεσα, εάν έχουν υποδοχείς που αναγνωρίζουν κύτταρα του ιδίου οργανισμού.

Όμως, μία τέτοια διαδικασία επιλογής, όπως και η φυσική επιλογή, οδηγεί τον πληθυσμό στην ομοιομορφία, πράγμα το οποίο είναι ανεπιθύμητο. Για τον λόγο αυτό, αφενός τα λεμφοκύτταρα καθώς πολλαπλασιάζονται μεταλλάσσονται, και αφετέρου ο πληθυσμός τους είναι δυνητικά εξαιρετικά μεγάλος, έτσι ώστε να επιτυγχάνεται η αναγνώριση οποιουδήποτε αντιγόνου. Πράγματι, στον άνθρωπο το πλήθος των διαφορετικών λεμφοκυττάρων που μπορούν να παραχθούν, είναι της τάξης του 10^{12}.

3.1.1 Ενισχυτική μάθηση και μνήμη του ανοσοποιητικού συστήματος

Κατά την διάρκεια της ζωής του ένας οργανισμός αναμένεται να έρθει σε επαφή με το ίδιο αντιγόνο πολλές φορές. Κατά την πρώτη επαφή του εγείρεται η *πρωτογενής ανοσολογική αντίδραση*, κατά την οποία ένας μικρός πληθυσμός από Β-λεμφοκύτταρα ενεργοποιείται και αρχίζει να παράγει αντισώματα με διαφορετική ποιότητα σύνδεσης (*affinity*) το καθένα. Όταν αντιμετωπιστεί η μόλυνση, κάποια από τα Β-λεμφοκύτταρα, που παρήγαγαν αντισώματα με σχετικά υψηλή ποιότητα σύνδεσης, «αποθηκεύονται» για μελλοντική χρήση. Όταν αργότερα εμφανιστεί πάλι το ίδιο αντιγόνο στον οργανισμό (*δευτερογενής ανοσολογική αντίδραση*), τότε αυτά τα κύτταρα μνήμης ενεργοποιούνται άμεσα και αρχίζουν να πολλαπλασιάζονται με μεγάλο ρυθμό, παράγοντας απευθείας καλά αντισώματα. Παράλληλα, Β-λεμφοκύτταρα χωρίς ιδιαίτερα καλούς υποδοχείς, συνεχίζουν να εξελίσσονται. Εάν στο τέλος της ανοσολογικής αντίδρασης έχουν παρουσιαστεί λεμφοκύτταρα με ακόμα καλύτερους υποδοχείς, τότε κάποια από αυτά εισέρχονται στον πληθυσμό των κυττάρων μνήμης. Καθώς, λοιπόν, ο οργανισμός θα εκτίθεται επαναλαμβανόμενα στο ίδιο αντιγόνο, τα λεμφοκύτταρα μνήμης θα βελτιώνονται συνεχώς. Επομένως, θα μπορούσε να χαρακτηρίσει κανείς το ανοσοποιητικό σύστημα, ως ένα *σύστημα ενισχυτικής μάθησης (reinforcement learning system)*, το οποίο βελτιώνει συνεχώς την απόδοσή του στην επίλυση του προβλήματος της αναγνώρισης και



Σχήμα 3.2: Συγκέντρωση των αντισωμάτων κατά την πρωτογενή, δευτερογενή και διασταυρουμένη αντίδραση του ανοσοποιητικού συστήματος.

αντιμετώπισης ενός αντιγόνου.

Εάν προσπαθούσε κανείς να αποδόσει σχηματικά την ανοσολογική αντίδραση του οργανισμού στην επανειλημμένη εμφάνιση ενός αντιγόνου, θα κατέληγε στο Σχήμα 3.2. Έστω ότι την χρονική στιγμή t_1 εμφανίζεται το αντιγόνο A_1, με το οποίο υποθέτουμε ότι ο οργανισμός δεν έχει έρθει σε επαφή στο παρελθόν. Αρχικά δεν υπάρχουν αντισώματα για το συγκεκριμένο αντιγόνο, οπότε μέχρι να δημιουργηθεί το κατάλληλο αντίσωμα περνά ένα χρονικό διάστημα $\Delta\tau_1$ (*χρονική υστέρηση–lag*). Την χρονική στιγμή $t_1 + \Delta\tau_1$ εμφανίζεται το κατάλληλο αντίσωμα, οπότε αρχίζει να παράγεται μαζικά, και επομένως αυξάνεται η συγκέντρωσή του μέχρι μία συγκεκριμένη τιμή. Στην συνέχεια η συγκέντρωση παραμένει για λίγο σταθερή και έπειτα φθίνει, καθ΄ ότι έχει αντιμετωπιστεί επιτυχώς το συγκεκριμένο αντιγόνο. Αξίζει να παρατηρήσει κανείς, ότι η συγκέντρωση των αντισωμάτων, αν και μειώνεται μετά το πέρας της ανοσολογικής αντίδρασης, εντούτοις δεν πέφτει στα αρχικά επίπεδα, γεγονός που αποδεικνύει την ύπαρξη Β-λεμφοκυττάρων μνήμης για το συγκεκριμένο αντιγόνο.

Έστω ότι λίγο αργότερα, την χρονική στιγμή t_2, εμφανίζονται στον οργανισμό δύο αντιγόνα. Το A_1, το οποίο είχε εμφανιστεί και την χρονική στιγμή t_1, και το A_2, το οποίο δεν έχει εμφανιστεί ξανά. Η αντίδραση του οργανισμού στο A_2 είναι εντελώς όμοια με την αντίδραση που είχε επιδείξει κατά την πρώτη εμφάνιση του A_1. Αυτό που χρήζει προσοχής, είναι η αντίδρασή του στην επανεμφάνιση του A_1. Λόγω της ύπαρξης των κυττάρων μνήμης, η υστέρηση που εμφανίζεται είναι πολύ μικρότερη απ΄ ότι προηγουμένως, $\Delta\tau_2 \ll \Delta\tau_1$, ενώ και η συγκέντρωση των αντισωμάτων φθάνει σε πολύ υψηλότερα επίπεδα αυτή την φορά. Επιπλέον, ο συνολικός χρόνος της ανοσολογικής αντίδρασης είναι μικρότερος.

Τέλος, έστω ότι την χρονική στιγμή t_3 εμφανίζεται ένα αντιγόνο A_1', το οποίο είναι όμοιο με το A_1. Η αντίδραση του οργανισμού σε αυτήν την περίπτωση είναι παρόμοια με την αντίδρασή του στην περίπτωση της επανεμφάνισης του A_1, την χρονική στιγμή t_2. Αυτό οφείλεται στο γεγονός ότι τα αντισώματα μπορούν και αναγνωρίζουν μοριακές δομές του αντιγόνου· αποτελούν στην ουσία ένα είδος μοριακού-χημικού συμπληρώματος του αντιγόνου. Έτσι, όταν εμφανιστεί το A_1', οι υποδοχείς των κυττάρων μνήμης μπορούν να συνδεθούν αρκετά καλά σε αυτό, καθ΄ ότι η μοριακή του δομή είναι παρόμοια με αυτή του A_1, για το οποίο υπάρχει ανοσία. Βέβαια, η ποιότητα της σύνδεσης είναι ελαφρώς υποδεέστερη, γι΄ αυτό και η συγκέντρωση των παραγομένων αντισωμάτων είναι λίγο μικρότερη, όμως αυτό δεν εμποδίζει την άμεση ($\Delta\tau_3 \approx \Delta\tau_2$) και αποτελεσματική αντιμετώπιση και αυτού

του αντιγόνου. Η απόκριση αυτή του ανοσοποιητικού συστήματος, ονομάζεται *απόκριση διασταυρουμένης αντίδρασης (cross-reactive response)*.

Η ιδιότητα της διασταυρουμένης αντίδρασης καθιστά την μνήμη του ανοσοποιητικού συστήματος *μνήμη συσχέτισης (associative memory)*, καθ΄ ότι τα δεδομένα που αποθηκεύονται (Β-λεμφοκύτταρα ή αντισώματα) μπορούν και ανακτώνται κατά την εμφάνιση των ίδιων ή σχετικών δεδομένων (αντιγόνα A_1 και A_1'). Παράλληλα, η μνήμη είναι ανθεκτική τόσο σε θόρυβο στα δεδομένα (εμφάνιση μεταλλαγμένων αντιγόνων), όσο και σε πιθανές αποτυχίες των συστατικών της (καταστροφή κάποιων λεμφοκυττάρων μνήμης).

3.1.2 Ωρίμανση σύνδεσης

Η επαναλαμβανόμενη έκθεση του οργανισμού στο ίδιο αντιγόνο οδηγεί στην βελτίωση της ανοσολογικής του αντίδρασης. Το φαινόμενο αυτό, που οφείλεται στους μηχανισμούς διαφοροποίησης των λεμφοκυττάρων, αλλά και στο γεγονός ότι στην μνήμη του ανοσοποιητικού συστήματος εισάγονται πάντα τα καλύτερα Β-λεμφοκύτταρα, ονομάζεται *ωρίμανση της ανοσολογικής αντίδρασης* ή πιο απλά *ωρίμανση σύνδεσης (affinity maturation)*.

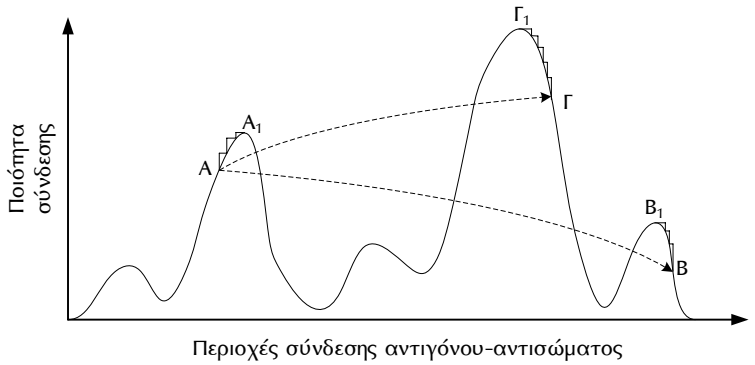
Οι μηχανισμοί διαφοροποίησης των υποδοχέων των Β-λεμφοκυττάρων είναι δύο:

1. η *υπερ-μετάλλαξη (hypermutation)*, και
2. η *διόρθωση των υποδοχέων (receptor editing)*.

Κατά την υπερ-μετάλλαξη εισάγονται με μεγάλο ρυθμό τυχαίες αλλαγές στο γενετικό υλικό των Β-λεμφοκυττάρων. Οι αλλαγές αυτές είναι γονιδιακές αλλαγές[4], που επηρεάζουν το γονίδιο των Β-λεμφοκυττάρων που είναι υπεύθυνο για την δημιουργία της μεταβλητής περιοχής των αντισωμάτων (V-region, βλ. §2.3). Με αυτό τον τρόπο παράγεται ένα μεγάλο πλήθος αντισωμάτων με διαφορετική ποιότητα σύνδεσης. Λόγω της τυχαιότητας των γενετικών αλλαγών, πολλά από τα μεταλλαγμένα Β-λεμφοκύτταρα θα έχουν υποδεεστέρους υποδοχείς σε σχέση με τα αρχικά, ενώ άλλα μπορεί να έχουν αναπτύξει υποδοχείς που στρέφονται κατά του ιδίου οργανισμού. Επομένως, θα πρέπει να υπάρχουν και ισχυροί μηχανισμοί επιλογής των καλυτέρων Β-λεμφοκυττάρων, ή αντιστρόφως καταστροφής των χειροτέρων, έτσι ώστε στην μνήμη του ανοσοποιητικού συστήματος να εισάγονται μόνο Β-λεμφοκύτταρα καλής ποιότητας.

Αν και ο Burnet κατά την διατύπωση της θεωρίας της επιλογής των κλώνων, υποστήριξε ότι τα Β-λεμφοκύτταρα διαφοροποιούνται μόνο βάσει του μηχανισμού της υπερ-μετάλλαξης, πιο πρόσφατες έρευνες (Nussenzweig, 1998) έδειξαν ότι πέρα από την διαδικασία επιλογής κλώνων, λαμβάνει χώρα και μία διαδικασία *επιλογής μορίων (molecular selection)*. Πιο συγκεκριμένα, βρέθηκε ότι Β-λεμφοκύτταρα με υποδοχείς χαμηλής ποιότητας ή με υποδοχείς εχθρικούς για τον οργανισμό, υποβλήθηκαν σε μία διαδικασία *διόρθωσης υποδοχέων (receptor editing)*. Κατά την διαδικασία αυτή, τα Β-λεμφοκύτταρα κατέστρεψαν τους υποδοχείς χαμηλής ποιότητας, και ανέπτυξαν εντελώς νέους υποδοχείς μέσω της ανασύνθεσης V(D)J (βλ. §2.3). Αν και η διόρθωση υποδοχέων (επιλογή μορίων) δεν είχε αρχικά προβλεφθεί, μπορεί εύκολα να ενταχθεί στην θεωρία του Burnet, εάν δεχθούμε ότι λαμβάνει χώρα πριν την επιλογή των κλώνων. Με άλλα λόγια, αφότου πολλαπλασιασθούν και υποστούν υπερ-μετάλλαξη, τα Β-λεμφοκύτταρα διορθώνουν τους υποδοχείς τους, και τέλος επιλέγονται.

[4] *Γονιδιακές* ονομάζονται οι γενετικές αλλαγές που επηρεάζουν κάποιο γονίδιο.



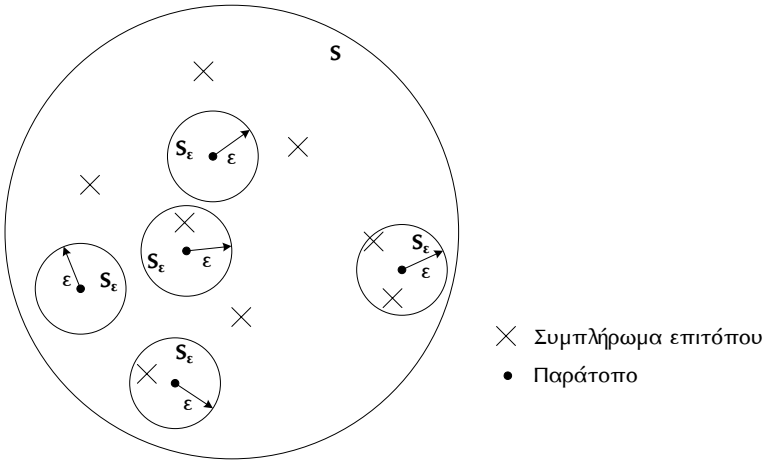
Σχήμα 3.3: Δισδιάστατη απεικόνιση του χώρου σύνδεσης αντιγόνου-αντισώματος. Η υπερ-μετάλλαξη οδηγεί στην ανακάλυψη τοπικών ακροτάτων, ενώ η διόρθωση υποδοχέων μπορεί να οδηγήσει στην εύρεση του ολικού ακροτάτου.

Η ύπαρξη αυτών των δύο μηχανισμών διαφοροποίησης των Β-λεμφοκυττάρων μπορεί να εξηγηθεί βάσει του Σχήματος 3.3. Στο σχήμα αυτό απεικονίζονται στον οριζόντιο άξονα οι περιοχές σύνδεσης του αντισώματος με το αντιγόνο, ενώ στον κατακόρυφο άξονα απεικονίζεται η ποιότητα σύνδεσης. Έστω ότι κατά την πρώτη εμφάνιση ενός αντιγόνου στον οργανισμό, παράγεται το αντίσωμα A, το οποίο έχει μία μέτρια ποιότητα σύνδεσης, όπως φαίνεται στο σχήμα. Εάν θεωρήσουμε ότι ο μόνος μηχανισμός διαφοροποιήσης των Β-λεμφοκυττάρων είναι η υπερ-μετάλλαξη, τότε ύστερα από κάποιες επανεμφανίσεις του ίδιου αντιγόνου, τα αντισώματα που θα παράγονται θα βρίσκονται στο τοπικό μέγιστο A_1. Επομένως, ο μηχανισμός της υπερ-μετάλλαξης μπορεί να βελτιώσει *μόνο* τοπικά την ποιότητα των αντισωμάτων. Αντίθετα, η διαδικασία της διόρθωσης των υποδοχέων μπορεί να προκαλέσει μεγάλα «άλματα» στο πεδίο των περιοχών σύνδεσης, οδηγώντας ενδεχομένως σε αντισώματα καλύτερης ποιότητας, σημείο Γ, απεγκλωβίζοντας έτσι τον αλγόριθμο μάθησης του ανοσοποιητικού συστήματος από τοπικά ακρότατα. Τέλος, για να διατηρείται η διαφορετικότητα του πληθυσμού των λεμφοκυττάρων, επιτρέπεται σε ένα μικρό ποσοστό χαμηλότερης ποιότητας λεμφοκυττάρων, να εισέρχεται στον πληθυσμό των κυττάρων μνήμης.

Ρύθμιση της διαδικασίας υπερ-μετάλλαξης Η υπερ-μετάλλαξη αποτελεί έναν εξαιρετικό μηχανισμό για την εκλέπτυνση και την γρήγορη απόκριση της ανοσολογικής αντίδρασης, όμως για να είναι αποδοτική θα πρέπει να υπόκειται σε αυστηρό έλεγχο. Λόγω της τυχαιότητας των γενετικών αλλαγών, οι πλειονότητά τους θα οδηγήσει σε χειρότερα Β-λεμφοκύτταρα. Εάν κάποιο Β-λεμφοκύτταρο μεταλλάχθηκε επιτυχώς, τότε εάν συνεχίσει να μεταλλάσσεται με τον ίδιο ρυθμό, η πιθανότητα να αναιρεθεί η επιτυχημένη μετάλλαξη είναι αυξημένη. Για τον λόγο αυτό, θα πρέπει να υπάρχει επιπλέον ένας μηχανισμός ελέγχου του ρυθμού των μεταλλάξεων. Λεμφοκύτταρα, τα οποία έχουν αναπτύξει καλούς υποδοχείς, θα πρέπει να μεταλλάσσονται με πολύ μικρό ρυθμό, ή ακόμα και να μην μεταλλάσσονται καθόλου.

3.2 Το μοντέλο του χώρου σχήματος

Το σύνολο των Β-λεμφοκυττάρων του οργανισμού θεωρείται ότι είναι *πλήρες*· μπορεί να αναγνωρίσει οποιοδήποτε αντιγόνο του παρουσιαστεί, είτε αυτό είναι

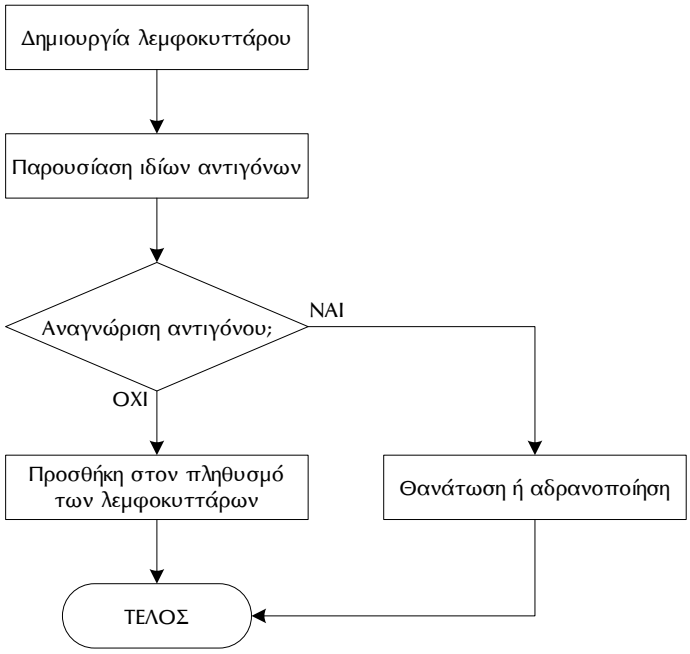


Σχήμα 3.4: Ο χώρος σχήματος. Ένα παράτοπο μπορεί να αναγνωρίσει με επιτυχία όλα τα επίτοπα που βρίσκονται στον χώρο S_ϵ.

τεχνητά κατασκευασμένο είτε όχι, και οποιαδήποτε μοριακή δομή του ιδίου οργανισμού. Ακόμα και αν κάποιος μπορούσε να αφαιρέσει από τα αντισώματα το 90% των εξειδικευμένων χαρακτηριστικών τους, όπως αυτά εκφράζονται μέσω των λεμφοκυττάρων, τα λεμφοκύτταρα με τα εναπομείναντα χαρακτηριστικά θα αποτελούσαν και πάλι ένα πλήρες σύνολο.

Για να εξηγηθεί ποσοτικά αυτό το φαινόμενο, εισήχθη η έννοια του *μοντέλου του χώρου σχήματος (shape-space model, Perelson και Oster (1979))*. Η σύνδεση ενός αντισώματος με το αντιγόνο επιτυγχάνεται μέσω μίας ποικιλίας χημικών δεσμών (ετεροπολικών δεσμών, δεσμών υδρογόνου, δεσμών van der Waals, κλπ.), το οποίο απαιτεί την στενή προσέγγιση των δύο μορίων. Απαιτείται, επομένως, το σχήμα, τα φορτία, αλλά και τα άτομα των δύο μορίων να είναι συμπληρωματικά, ή τουλάχιστον να υπάρχουν εκτεταμένες περιοχές συμπληρωματικότητας (regions of complementarity). Όλα τα χαρακτηριστικά που απαιτούνται από ένα μόριο (αντίσωμα ή αντιγόνο) για να συνδεθεί, έστω τμηματικά, με κάποιο άλλο μόριο, αποτελούν το *γενικευμένο σχήμα (generalized shape)* του μορίου αυτού. Εάν υποθέσουμε ότι μπορούμε να περιγράψουμε το γενικευμένο σχήμα μιας περιοχής σύνδεσης ενός αντισώματος (παράτοπο, βλ. §2.3) με L μεταβλητές, τότε προκύπτει ένας χώρος S διάστασης L, ο οποίος περιγράφει όλα τα δυνατά γενικευμένα σχήματα των παρατόπων. Ο χώρος αυτός ονομάζεται *χώρος σχήματος* (Σχήμα 3.4) και γενικά είναι πεπερασμένος, καθ΄ ότι τα χαρακτηριστικά της σύνδεσης παίρνουν τιμές από πεπερασμένα σύνολα τιμών.

Το κάθε παράτοπο απεικονίζεται ως ένα σημείο του χώρου S. Επειδή η σχέση παρατόπων-επιτόπων είναι σχέση συμπληρωματική (το επίτοπο είναι συμπλήρωμα του παρατόπου και αντιστρόφως), έπεται ότι και το συμπλήρωμα του επιτόπου μπορεί να απεικονιστεί ως ένα σημείο του χώρου S. Όπως ήδη ανεφέρθη, το σύνολο των παρατόπων είναι πλήρες, το οποίο σημαίνει ότι τα παράτοπα θα πρέπει να καλύπτουν όλο τον χώρο S. Όμως, ο αριθμός των παρατόπων σε έναν οργανισμό είναι πεπερασμένος. Επομένως, για να καλυφθεί όλος ο χώρος S, θα πρέπει κάθε παράτοπο P να μπορεί να αναγνωρίζει όλα τα συμπληρώματα των επιτόπων που βρίσκονται σε ένα συγκεκριμένο υποσύνολο του S, έστω S_ϵ, όπου ϵ αυθαίρετη σταθερά. Ο χώρος S_ϵ ονομάζεται *περιοχή αναγνώρισης (recognition region)* του παρατόπου P. Η ποιότητα σύνδεσης του παρατόπου με τα επίτοπα της περιοχής αναγώρισης δεν είναι σταθερή, αλλά εξαρτάται από την



Σχήμα 3.5: Ο αλγόριθμος αρνητικής επιλογής που χρησιμοποιείται από το ανοσοποιητικό σύστημα για τον διαχωρισμό ιδίου-ξένου.

απόσταση συμπληρώματος-παρατόπου στον χώρο S. Η ύπαρξη της περιοχής αναγνώρισης και η ατελής σύνδεση παρατόπου-επιτόπου εξηγεί την ανοσολογική απόκριση διασταυρούμενης αντίδρασης (βλ. §3.1.1).

3.3 Διαχωρισμός ιδίου-ξένου

Ο *διαχωρισμός ιδίου-ξένου (self-nonself discrimination)* συνιστά την ιδιότητα του ανοσοποιητικού συστήματος να αναγνωρίζει όλα τα κύτταρα του οργανισμού του. Αποτελεί μία από τις σημαντικότερες λειτουργίες του, καθ΄ ότι προφυλάσσει τον οργανισμό από αυτοάνοσες παθήσεις. Τα μόρια του ιδίου οργανισμού αντιμετωπίζονται και αυτά ως αντιγόνα (*ίδια αντιγόνα*). Η ιδιότητα του ανοσοποιητικού συστήματος να μην αντιδρά στα ίδια αντιγόνα ονομάζεται *ανοχή έναντι του ιδίου (self-tolerance)*, ή απλά *ανοχή*.

Για να επιτύχει τον διαχωρισμό ιδίου-ξένου, το ανοσοποιητικό σύστημα χρησιμοποιεί μία τεχνική, που ονομάζεται *αρνητική επιλογή (negative selection)*. Η τεχνική αυτή έγκειται στο γεγονός, ότι δεν πολλαπλασιάζονται όλα τα λεμφοκύτταρα που αναγνωρίζουν ένα αντιγόνο, αντιθέτως εκείνα που αναγνωρίζουν ίδια αντιγόνα, είτε θανατώνονται είτε αδρανοποιούνται. Η αρνητική επιλογή εφαρμόζεται τόσο στα Τ-λεμφοκύτταρα, όσο και στα Β-λεμφοκύτταρα, ενώ λαμβάνει χώρα στον θύμο και στα πρωτογενή λεμφικά όργανα (βλ. §2.1), όπου συγκεντρώνονται μεγάλες ποσότητες ιδίων αντιγόνων. Τα όργανα αυτά είναι προστατευμένα με τέτοιο τρόπο από το έμφυτο ανοσοποιητικό σύστημα, έτσι ώστε να μην επιτρέπουν την εισαγωγή ξένων αντιγόνων. Εάν κάποιο λεμφοκύτταρο συνδεθεί με επιτυχία με κάποιο από τα αντιγόνα των οργάνων αυτών, τότε θανατώνεται, καθ΄ ότι έχει συνδεθεί με ένα ίδιο αντιγόνο και επομένως στρέφεται κατά του ιδίου οργανισμού. Η αρνητική επιλογή, όμως, μπορεί να λάβει χώρα και εκτός των πρωτογενών οργάνων. Για να ενεργοποιηθεί ένα λεμφοκύτταρο και να αρχίσει να

πολλαπλασιάζεται, δεν αρκεί η επιτυχής σύνδεσή του με κάποιο αντιγόνο, αλλά είναι απαραίτητη και η ύπαρξη συγκεκριμένων ουσιών-σημάτων, που δηλώνουν την ύπαρξη ανοσολογικής αντίδρασης. Εάν τέτοια σήματα απουσιάζουν, σημαίνει ότι το λεμφοκύτταρο προσδέθηκε σε κάποιο κύτταρο του ιδίου οργανισμού, και έτσι αδρανοποιείται. Στο Σχήμα 3.5 παρουσιάζεται ο αλγόριθμος της αρνητικής επιλογής.

3.4 Μάθηση μηχανών βασισμένη στην αρχή της επιλογής των κλώνων

Έχοντας μελετήσει τους μηχανισμούς και τους αλγορίθμους που χρησιμοποιεί το ανοσοποιητικό σύστημα για την αναγνώριση και αντιμετώπιση των ξένων οργανισμών, η ανάπτυξη ενός αντιστοίχου υπολογιστικού αλγορίθμου μηχανικής μάθησης, είναι σχετικά άμεση. Στην παρούσα παράγραφο παρουσιάζουμε έναν αλγόριθμο μάθησης μηχανών, βασισμένο στον αλγοριθμο CLONALG (de Castro και Von Zuben, 2002), τον οποίο εφαρμόζουμε σε δύο προβλήματα αναγνώρισης χαρακτήρων. Οι αρχές του ανοσοποιητικού συστήματος που υιοθετεί ο αλγόριθμος είναι οι εξής:

1. διατήρηση ενός συνόλου κυττάρων μνήμης,
2. επιλογή και πολλαπλασιασμός των κυττάρων που εμφάνισαν την καλύτερη συμπεριφορά,
3. αδρανοποίηση των κυττάρων με υποδοχείς χαμηλής ποιότητας,
4. ωρίμανση σύνδεσης, και
5. επιλογή των καλυτέρων κλώνων, ανανέωση του συνόλου των κυττάρων μνήμης, και διατήρηση της διαφορετικότητας του συνολικού πληθυσμού.

Στην συνέχεια αυτής της εργασίας δεν θα γίνεται διάκριση μεταξύ λεμφοκυττάρων και αντισωμάτων, καθ΄ ότι από υπολογιστικής απόψεως είναι ισοδύναμα. Τα αντισώματα αποτελούν την έκφραση του γενετικού υλικού των λεμφοκυττάρων, και επομένως οποιαδήποτε αλλαγή συμβεί στα λεμφοκύτταρα αντικατοπτρίζεται στα αντισώματα. Άντιστοίχως, όταν γίνεται αναφορά σε πληθυσμό αντισωμάτων, θα εννοείται ο πληθυσμός των λεμφοκυττάρων που τα παράγουν. Ακολούθως παρατίθενται κάποιοι βασικοί ορισμοί.

Ορισμός 3.1. *Έστω Σ ένα πεπερασμένο σύνολο συμβόλων (αλφάβητο). Τότε ένα αντίσωμα a ορίζεται ως μία ακολουθία συμβόλων του Σ μήκους $l \in \mathbb{N}$, δηλαδή*

$$a = s, \quad \textit{όπου } s \in \Sigma^* \textit{ και } |s| = l.$$

Το σύνολο Σ^* αποτελεί το σύνολο όλων των ακολουθιών συμβόλων που μπορούν να προκύψουν από το σύνολο Σ.

Ο ορισμός αυτός του αντισώματος είναι συμβατός με το μοντέλο του χώρου σχήματος (βλ. §3.2). Πράγματι, εάν θεωρήσουμε μία απαρίθμηση του συνόλου Σ, δηλαδή ένα σύνολο $T \subset \mathbb{N}$ με $|T| = l$, και μία αμφιμονοσήμαντη συνάρτηση απεικόνισης $g : \Sigma \to T$, τότε μέσω της g κάθε αντίσωμα $a \in \Sigma^*$ απεικονίζεται στον διακριτό χώρο $S = T^l$. Ο χώρος αυτός αποτελεί και τον χώρο σχήματος του προβλήματος.

Ένα αντιγόνο ορίζεται εντελώς αντίστοιχα με το αντίσωμα:

Ορισμός 3.2. *Έστω* Σ *ένα πεπερασμένο σύνολο συμβόλων (αλφάβητο). Τότε ένα αντιγόνο* g *ορίζεται ως μία ακολουθία συμβόλων του* Σ *μήκους* $l \in \mathbb{N}$*, δηλαδή*

$$g = s, \quad \textit{όπου } s \in \Sigma^* \textit{ και } |s| = l.$$

Το αλφάβητο Σ και το μήκος l της ακολουθίας είναι κοινά για τα αντισώματα και τα αντιγόνα, και επομένως κάθε αντιγόνο μπορεί και αυτό να απεικονιστεί στον χώρο S. Το ότι απεικονίζονται τα ακριβή αντιγόνα και όχι τα συμπληρώματά τους, δεν αντιβαίνει στον ορισμό του χώρου σχήματος, καθ΄ ότι η σχέση μεταξύ του αντιγόνου και του συμπληρώματός του είναι αντιστρέψιμη. Επομένως, εάν ένα αντίσωμα αναγνωρίσει το αντιγόνο, τότε αναγνωρίζει και το συμπλήρωμά του, και αντιστρόφως.

Για να απλοποιηθεί ο ορισμός των αντισωμάτων και των αντιγόνων, μπορεί να οριστεί μία γλώσσα πάνω στο αλφάβητο Σ:

Ορισμός 3.3. *Έστω* Σ *το αλφάβητο από το οποίο δομούνται τα αντισώματα και τα αντιγόνα, τότε ως γλώσσα των αντισωμάτων ορίζεται το σύνολο*

$$\mathcal{L} = \{s \in \Sigma^* \textit{ και } |s| = l,\ l \in \mathbb{N}\}.$$

Επομένως, η γλώσσα των αντισωμάτων αποτελείται από όλες τις δυνατές ακολουθίες συμβόλων του Σ μήκους l. Βάσει αυτού του ορισμού το αντίσωμα και το αντιγόνο, μπορούν πολύ απλά να οριστούν ως στοιχεία της γλώσσας $\mathcal{L}$, δηλαδή $a, g \in \mathcal{L}$.

Για να επιτύχει την αναγνώριση των αντιγόνων, ο αλγόριθμος χρησιμοποιεί ένα σύνολο αντισωμάτων $\mathcal{P} \subseteq \mathcal{L}$. Το σύνολο αυτό αποτελεί τον πληθυσμό των αντισωμάτων. Γενικά ισχύει $|\mathcal{P}| \ll |\mathcal{L}|$. Ορίζονται επιπλέον δύο σύνολα, τα $\mathcal{M}$ και $\mathcal{R}$, τέτοια ώστε

$$\mathcal{M} \cup \mathcal{R} = \mathcal{P} \text{ και } \mathcal{M} \cap \mathcal{R} = \emptyset.$$

Το σύνολο $\mathcal{M}$ αποτελεί την μνήμη του αλγορίθμου, ενώ το σύνολο $\mathcal{R}$ αποτελείται από τα υπόλοιπα κύτταρα.

Τα προς αναγνώριση αντιγόνα απαρτίζουν το σύνολο $\mathcal{G} \subseteq \mathcal{L}$. Για να μπορέσει ο αλγόριθμος να αναγνωρίσει τα αντιγόνα θα πρέπει να ισχύει $|\mathcal{G}| \leq |\mathcal{M}| \leq |\mathcal{P}|$. Μεταξύ αντιγόνων και κυττάρων μνήμης ορίζεται μία απεικόνιση

$$K : \mathcal{G} \to \mathcal{M}.$$

Η απεικόνιση αυτή αντιστοιχίζει τα αντιγόνα με τα κύτταρα μνήμης που τα αναγνωρίζουν. Η απεικόνιση αυτή συνήθως δεν αποτελεί συνάρτηση, καθ΄ ότι για ένα συγκεκριμένο αντιγόνο μπορεί να υπάρχουν περισσότερα από ένα κύτταρα μνήμης που το αναγνωρίζουν. Επί παραδείγματι έστω ότι $\mathcal{G} = \{g_0, g_1\}$ και $\mathcal{M} = \{m_0, m_1, m_2\}$, τότε η απεικόνιση K, για την οποία ισχύουν

$$\begin{aligned} K(g_0) &= m_0, \\ K(g_0) &= m_1, \\ K(g_1) &= m_2, \end{aligned}$$

ορίζει ότι τα κύτταρα μνήμης m_0 και m_1 αναγνωρίζουν το αντιγόνο g_0, ενώ το κύτταρο μνήμης m_2 αναγνωρίζει το αντιγόνο g_1. Το γεγονός ότι η K δεν είναι συνάρτηση, δημιουργεί κάποιο πρόβλημα κατά την φάση ανανέωσης της μνήμης του πληθυσμού, καθ΄ ότι δεν είναι σαφές ποιο ακριβώς κύτταρο μνήμης θα αντικατασταθεί. Για τον λόγο αυτό ο αλγόριθμος θα πρέπει να ακολουθεί και μία συγκεκριμένη πολιτική ανανέωσης της μνήμης, π.χ. στις περιττές εμφανίσεις

του g_0 να ανανεώνει το m_0, ενώ στις άρτιες να ανανεώνει το m_1. Αυτό που χρήζει προσοχής είναι ότι τόσο το σύνολο μνήμης $\mathcal{M}$, όσο και η απεικόνιση K είναι ορισμένες εξ΄ αρχής, *προτού* αρχίσει να εκτελείται ο αλγόριθμος, και παραμένουν σταθερές–η μεν σε μέγεθος, η δε διατηρώντας την ίδια απεικόνιση–καθ΄ όλη την διάρκεια εκτέλεσης. Επομένως, κάθε αντιγόνο θα αναγνωρίζεται από συγκεκριμένο αντίσωμα μνήμης, ή μιλώντας πιο υπολογιστικά κάθε αντιγόνο θα αναγνωρίζεται από το αντίσωμα που βρίσκεται στην θέση μνήμης που υποδεικνύει η απεικόνιση K. Επιπλέον βάσει της απεικόνισης K μπορούν να ορισθούν τα σύνολα

$$\mathcal{M}_i = \{m : m = K(g),\ g \in \mathcal{G}\}, \quad 1 \leq i \leq n = |\mathcal{G}|,$$

για τα οποία ισχύουν

$$\bigcap_{i=1}^{n} \mathcal{M}_i = \emptyset, \text{ και}$$
$$\bigcup_{i=1}^{n} \mathcal{M}_i \subseteq \mathcal{M}.$$

Ορισμός 3.4. *Το σύνολο μνήμης $\mathcal{M}$ λέγεται ελάχιστο όταν*

$$\bigcup_{i=1}^{n} \mathcal{M}_i = \mathcal{M}.$$

Στην περίπτωση αυτή δεν υπάρχουν ανενεργά αντισώματα στην μνήμη, δηλαδή αντισώματα που δεν αναγνωρίζουν κανένα αντιγόνο, και η απεικόνιση K ορίζει πλέον μία διαμέριση επί του συνόλου μνήμης $\mathcal{M}$.

Πρόταση 3.1. *Κάθε σύνολο πληθυσμού με ένα μη ελάχιστο σύνολο μνήμης μπορεί αναχθεί σε ένα ισοδύναμο σύνολο πληθυσμού με ελάχιστο σύνολο μνήμης.*

Απόδειξη. Εάν θεωρήσουμε το σύνολο $M_u = M - \bigcup_{i=1}^{n} \mathcal{M}_i$, το οποίο περιέχει όλα τα κύτταρα μνήμης που δεν αναγνωρίζουν κανένα αντιγόνο, ή αλλιώς τα κύτταρα για τα οποία η K^{-1} δεν ορίζεται, και αναδιατάξουμε τα στοιχεία των συνόλων $\mathcal{R}$ και $\mathcal{M}$, έτσι ώστε

$$\mathcal{R}' = \mathcal{R} \cup \mathcal{M}_u, \text{ και}$$
$$\mathcal{M}' = \mathcal{M} - \mathcal{M}_u,$$

τότε το σύνολο $\mathcal{M}'$ είναι ελάχιστο και παράλληλα ο πληθυσμός $\mathcal{P}' = \mathcal{R}' \cup \mathcal{M}'$ περιέχει τα ίδια ακριβώς κύτταρα με τον $\mathcal{P}$, ενώ η μνήμη του παρουσιάζει την ίδια λειτουργικότητα. □

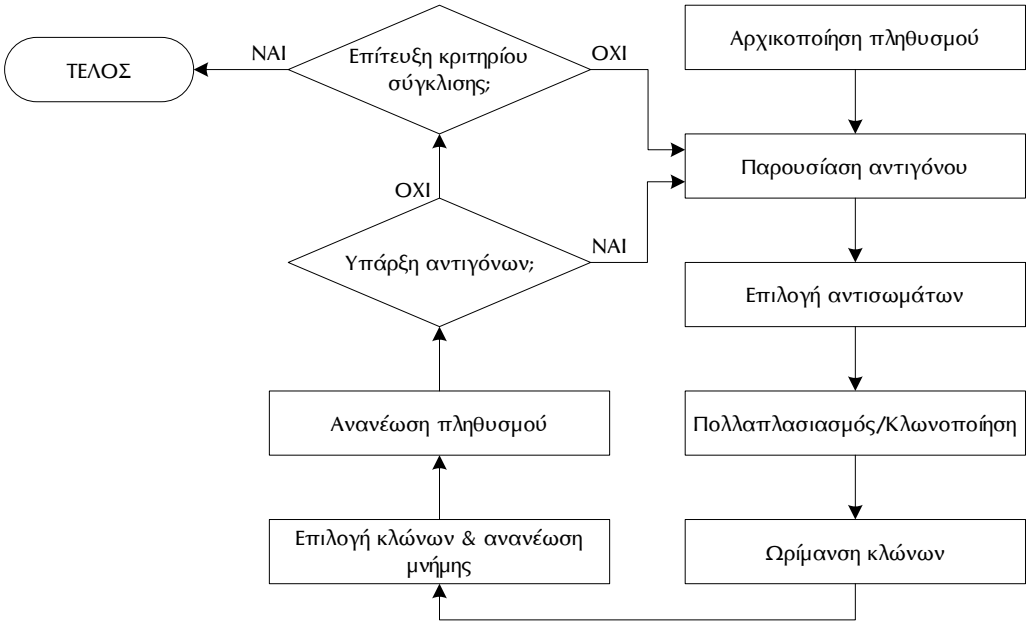
Ορίζεται, τέλος, μία συνάρτηση σύνδεσης f, η οποία χαρακτηρίζει την ποιότητα σύνδεσης αντισώματος-αντιγόνου (affinity). Η συνάρτηση αυτή θα πρέπει να παίρνει ως ορίσματα ένα αντιγόνο και ένα αντίσωμα και να επιστρέφει μία τιμή ενδεικτική της ποιότητας σύνδεσης των δύο μορίων. Θα μπορούσε επομένως να οριστεί ως

$$f : \Sigma^* \times \Sigma^* \to \mathbb{R}$$

ή λόγω της ισοδυναμίας του Σ^* με τον χώρο T^l, ως

$$f : T^l \times T^l \to \mathbb{R}.$$

Η συνάρτηση αυτή θα πρέπει να δίνει υψηλές τιμές, όταν το αντίσωμα και το αντιγόνο είναι αρκετά όμοια, και χαμηλότερες στην αντίθετη περίπτωση. Συνήθως



Σχήμα 3.6: Αλγόριθμος μάθησης μηχανών βασισμένος στην αρχή της επιλογής των κλώνων.

η f είναι κανονικοποιημένη στο διάστημα $[0,1]$, δίνοντας 0 όταν το αντίσωμα και το αντιγόνο είναι συμπληρωματικά, και 1 όταν υπάρχει πλήρης ταύτιση μεταξύ αντιγόνου και αντισώματος.

Έχοντας περιγράψει και ορίσει τις βασικές παραμέτρους του, ο αλγόριθμος μπορεί να περιγραφεί ως εξής (Σχήμα 3.6):

Αλγόριθμος ΕΚ (Αλγορίθμος επιλογής κλώνων). Πρόκειται για έναν αλγόριθμο μάθησης μηχανών για την αναγνώριση και αποθήκευση προτύπων, ο οποίος βασίζεται στην αρχή της επιλογής των κλώνων που χρησιμοποιεί το ανοσοποιητικό σύστημα για την αναγνώριση νέων αντιγόνων.

ΕΚ1. [Αρχικοποίηση του πληθυσμού] Για κάθε αντίσωμα $a_i \in \mathcal{P}, \quad 1 \leq i \leq |\mathcal{P}|$ επιλέγεται μία τυχαία ακολουθία συμβόλων $s_i \in \mathcal{L}$ και ανατίθεται σε αυτό, $a_i \leftarrow s_i$. Επιπλέον ορίζεται το σύνολο $\mathcal{G}_r \subseteq \mathcal{L} : \mathcal{G}_r = \mathcal{G}$.

ΕΚ2. [Παρουσίαση αντιγόνου στον πληθυσμό] Επιλέγεται τυχαία ένα αντιγόνο $g_i \in \mathcal{G}_r, 1 \leq i \leq |\mathcal{G}_r|$ και παρουσιάζεται στον πληθυσμό. Κατά την παρουσίαση του αντιγόνου στον πληθυσμό υπολογίζεται η συνάρτηση σύνδεσης f για κάθε αντίσωμα του πληθυσμού. Προκύπτει, τότε, το σύνολο

$$\mathcal{V} = \{v_j : v_j = f(a_j, g_i), 1 \leq j \leq |\mathcal{P}|\}.$$

Το σύνολο αυτό περιέχει την ποιότητα σύνδεσης κάθε αντισώματος του πληθυσμού με το αντιγόνο g_i. Τέλος, το αντιγόνο g_i αφαιρείται από το $\mathcal{G}_r$, οπότε

$$\mathcal{G}_r \leftarrow \mathcal{G}_r - \{g_i\}.$$

ΕΚ3. [Επιλογή των καλυτέρων αντισωμάτων] Βάσει των στοιχείων του συνόλου $\mathcal{V}$ επιλέγονται τα n_b αντισώματα που επέδειξαν την καλύτερη ποιότητα σύνδεσης. Τα αντισώματα αυτά αποτελούν πλέον το σύνολο $\mathcal{B}$, $|\mathcal{B}| = n_b$.

ΕΚ4. [Πολλαπλασιασμός των καλυτέρων αντισωμάτων] Κάθε αντίσωμα του συνόλου $\mathcal{B}$ πολλαπλασιάζεται (κλωνοποιείται) βάσει της ποιότητας σύνδεσής του με το αντιγόνο g_i. Όσο καλύτερη είναι η ποιότητα σύνδεσης ενός αντισώματος, τόσο περισσοτέρους κλώνους θα δώσει. Το προκύπτον σύνολο κλώνων αποτελεί το σύνολο $\mathcal{C}$.

ΕΚ5. [Ωρίμανση των κλώνων] Κάθε στοιχείο c_j του συνόλου $\mathcal{C}$ μεταλλάσσεται με ρυθμό α_j, ο οποίος εξαρτάται από την ποιότητα σύνδεσης του κλώνου c_j με το αντιγόνο g_i. Όσο καλύτερη είναι η ποιότητα σύνδεσης, τόσο μικρότερος είναι ο ρυθμός των μεταλλάξεων, έτσι ώστε να μην εισάγονται αναιρετικές αλλαγές στο αντίσωμα (βλ. §3.1.2). Το σύνολο των μεταλλαγμένων κλώνων συμβολίζεται με $\mathcal{C}_m$.

ΕΚ6. [Υπολογισμός ποιότητας σύνδεσης των κλώνων] Εφαρμόζεται η συνάρτηση f σε κάθε στοιχείο του συνόλου $\mathcal{C}_m$, και προκύπτει το σύνολο

$$\mathcal{V}' = \{v'_j : v'_j = f(c'_j, g_i), \quad 1 \leq j \leq |\mathcal{C}_m|\}.$$

Το σύνολο αυτό περιέχει την ποιότητα σύνδεσης κάθε μεταλλαγμένου κλώνου.

ΕΚ7. [Ανανέωση μνήμης] Βάσει του συνόλου $\mathcal{V}'$ επιλέγονται οι n_m καλύτεροι κλώνοι, οι οποίοι αποτελούν το σύνολο $\mathcal{B}'$. Εφαρμόζεται στην συνέχεια η απεικόνιση Κ στο αντιγόνο g_i, οπότε προκύπτει το σύνολο $\mathcal{M}_i$ των αντισωμάτων μνήμης που είναι υποψήφια για αντικατάσταση. Βάσει της πολιτικής ανανέωσης της μνήμης, την οποία ακολουθεί ο αλγόριθμος, προκύπτει ένα τελικό σύνολο κυττάρων $\mathcal{M}'_i$, τέτοιο ώστε $n_m = |\mathcal{M}'_i| \leq |\mathcal{M}_i|$. Τα κύτταρα μνήμης του συνόλου αυτού θα αντικατασταθούν από τα επιλεχθέντα κύτταρα, αν και μόνο αν τα τελευταία παρουσιάζουν καλύτερη ποιότητα σύνδεσης. Θα πρέπει να ισχύει επομένως

$$f(m, g_i) < f(a, g_i), \quad m \in \mathcal{M}'_u,\ a \in \mathcal{B}'.$$

ΕΚ8. [Ανανέωση πληθυσμού] Υπάρχουν δύο τρόποι ανανέωσης του πληθυσμού, έτσι ώστε να διατηρηθεί η διαφορετικότητά του. Στην πρώτη περίπτωση επιλέγονται n_r κύτταρα από το σύνολο $\mathcal{V}'$ και εισάγονται στον πληθυσμό, αντικαθιστώντας κάποια ήδη υπάρχοντα. Στην δεύτερη περίπτωση επιλέγονται τα n_d χειρότερα κύτταρα από το πληθυσμό $\mathcal{P}$ και αντικαθίστανται με εντελώς νέα. Τα νέα κύτταρα είναι τυχαίες ακολουθίες της γλώσσας $\mathcal{L}$.

ΕΚ9. [Συνθήκη τέλους] Εάν ισχύει $\mathcal{G}_r \neq \emptyset$, τότε ο αλγόριθμος επαναλαμβάνεται από το Βήμα ΕΚ2. Διδ΄ άλλως ελέγχεται κάποιο κριτήριο σύγκλισης των αντισωμάτων της μνήμης $\mathcal{M}$, με τα αντιγόνα του συνόλου $\mathcal{G}$. Εάν δεν έχει επιτευχθεί σύγκλιση, τότε $\mathcal{G}_r \leftarrow \mathcal{G}$ και ο αλγόριθμος επαναλαμβάνεται από το Βήμα ΕΚ2. Σε αντίθετη περίπτωση ο αλγόριθμος τερματίζεται. Μόλις ο αλγόριθμος φθάσει σε αυτό το σημείο ($G_r = \emptyset$), έχει ολοκληρώσει μία *γενιά εξέλιξης*. ∎

Στο Βήμα ΕΚ5 του αλγορίθμου (ωρίμανση κλώνων), μεταλλάσσονται τα αντισώματα του συνόλου $\mathcal{C}$ των κλώνων. Οι μεταλλάξεις που συμβαίνουν, εισάγονται ως τυχαίες αλλαγές στα μεμονωμένα σύμβολα, που αποτελούν την ακολουθία συμβόλων του αντισώματος c_j. Ο ρυθμός μεταλλάξεων αποτελεί την πιθανότητα, με την οποία μπορεί ένα σύμβολο του αντισώματος να αλλάξει, και είναι σταθερός για κάθε σύμβολο της ακολουθίας. Επομένως σε ένα αντίσωμα μήκους l, που

μεταλλάσσεται με ρυθμό α εισάγονται κατά μέσο όρο, $n = \alpha \cdot l$ αλλαγές. Εάν συμβεί μετάλλαξη, το νέο σύμβολο που θα εισαχθεί προκύπτει με τυχαίο τρόπο: απαριθμούνται όλα τα σύμβολα του αλφαβήτου Σ, και βάσει μίας τυχαίας μεταβλητής που ακολουθεί την ομοιόμορφη κατανομή στο διάστημα της απαρίθμησης, εκλέγεται το νέο σύμβολο.

Τέλος μία πιο προσεκτική μελέτη στην δομή και λειτουργία του αλγορίθμου, μπορεί να οδηγήσει στην επόμενη πρόταση.

Πρόταση 3.2. *Ο αλγόριθμος διατηρεί πάντα στην μνήμη του τα καλύτερα αντισώματα που έχει να επιδείξει ο πληθυσμός σε οποιοδήποτε αντιγόνο, υπό την προϋπόθεση ότι η πολιτική ανανέωσης της μνήμης μπορεί να ανανεώσει όλα τα αντισώματα μνήμης. Επομένως*

$$f(m, g) \geq f(a, g),$$
$$\forall m \in \mathcal{M} : K(g) = m,\ \forall g \in \mathcal{G},\ \forall a \in \mathcal{R} = \mathcal{P} - \mathcal{M}.$$

Απόδειξη. Η απόδειξη της πρότασης αυτής γίνεται με επαγωγή στον αριθμό των επαναλήψεων του αλγορίθμου. Κάθε φορά που ο αλγόριθμος φθάνει στο Βήμα ΕΚ7 υπάρχουν δύο ενδεχόμενα:

- $f(m, g) \geq f(a, g)$, οπότε το αντίσωμα μνήμης m παρεμένει αμετάβλητο, ή
- $f(m, g) < f(a, g)$, οπότε $m \leftarrow a$, και άρα τελικά $f(m, g) = f(a, g)$.

Σε κάθε περίπτωση επομένως ισχύει

$$f(m, g) \geq f(a, g). \tag{3.1}$$

Το Βήμα ΕΚ9 εξασφαλίζει ότι το Βήμα ΕΚ7 θα εκτελεστεί για κάθε $g \in \mathcal{G}$. Επομένως η σχέση (3.1) θα ισχύει για κάθε $g \in \mathcal{G}$. Εάν θεωρήσουμε χωρίς βλάβη της γενικότητας ότι το σύνολο $\mathcal{M}$ είναι ελάχιστο (Πρόταση 3.1), και ότι η πολιτική ανανέωσης της μνήμης επιτρέπει σε όλα τα κύτταρα να ανανεωθούν, τότε η σχέση (3.1) θα ισχύει και για κάθε $m \in \mathcal{M}$ Επομένως ο αλγορίθμος θα διατηρεί πάντα στην μνήμη του τα καλύτερα αντισώματα όλου του πληθυσμού. □

Από την ανωτέρω πρόταση και από το γεγονός ότι το κριτήριο σύγκλισης υπολογίζεται βάσει των κυττάρων μνήμης (βλ. Βήμα ΕΚ9) προκύπτει εύκολα το εξής πόρισμα:

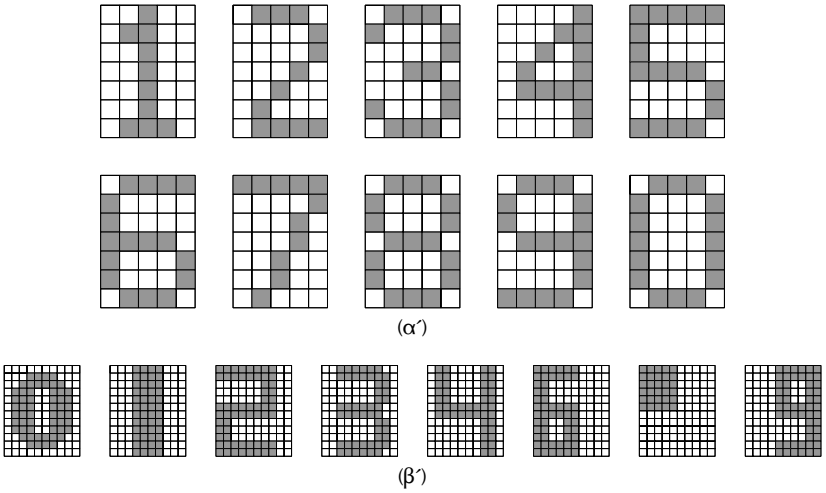
Πόρισμα 3.1. *Ο αλγόριθμος δεν μπορεί να αποκλίνει.*

Έχοντας αναλύσει την δομή και την λειτουργία του αλγορίθμου από θεωρητικής πλευράς, στις επόμενες παραγράφους αναλύεται η συμπεριφορά του σε δύο πραγματικά προβλήματα αναγνώρισης προτύπου.

3.4.1 Το πρόβλημα αναγνώρισης ψηφιακών χαρακτήρων

Το πρόβλημα της αναγνώρισης ψηφιακών χαρακτήρων μπορεί να χωριστεί σε δύο φάσεις:

- την φάση *εκπαίδευσης*, και
- την φάση *ανάκτησης*.



Σχήμα 3.7: Τα δύο σύνολα φηφιακών χαρακτήρων που χρησιμοποιήθηκαν για την εκπαίδευση του αλγορίθμου: (α΄) ένα σύνολο 10 φηφιακών χαρακτήρων ανάλυσης 5×7, (β΄) ένα σύνολο 8 ψηφιακών χαρακτήρων ανάλυσης 10×12.

Κάθε σύστημα αναγνώρισης χαρακτήρων ή γενικότερα προτύπων, περνά διαδοχικά και από τις δύο φάσεις. Κατά την πρώτη φάση παρουσιάζονται στο σύστημα τα ακριβή πρότυπα χωρίς θόρυβο, και εκείνο βάσει ενός συγκεκριμένου αλγορίθμου (*αλγόριθμος μάθησης*), προσπαθεί να αποθηκεύσει μία εικόνα αυτών των προτύπων. Στην δεύτερη φάση, η οποία ονομάζεται γενικά και *φάση δοκιμής*, το σύστημα έρχεται αντιμέτωπο με πραγματικά δεδομένα, τα οποία συνήθως έχουν κάποιο ποσοστό θορύβου. Στην περίπτωση αυτή το σύστημα θα πρέπει να αποκριθεί ανακτώντας το ακριβές πρότυπο. Ο αλγόριθμος που παρουσιάστηκε στην παράγραφο 3.4 ασχολείται με την φάση εκπαίδευσης ενός συστήματος αναγνώρισης προτύπων, αν και η φάση ανάκτησης, όπως θα φανεί στην συνέχεια, θα μπορούσε να υλοποιηθεί αρκετά απλά, λόγω της πολύ καλής συμπεριφοράς του αλγορίθμου εκπαίδευσης.

Ο αλγόριθμος που προτείνεται σε αυτή την παράγραφο, δοκιμάστηκε σε δύο σύνολα ψηφιακών χαρακτήρων, διαφορετικής πολυπλοκότητας το καθένα, και επέδειξε πολύ καλή συμπεριφορά. Σε πολύ σύντομο χρονικό διάστημα κατάφερε να αναγνωρίσει και να αποθηκεύσει επιτυχώς κάθε πρότυπο του αρχικού συνόλου. Τα σύνολα χαρακτήρων φαίνονται στο Σχήμα 3.7. Το πρώτο σύνολο (Von Zuben και De Castro, 1999) αποτελείται από 10 χαρακτήρες ανάλυσης 5×7, ενώ το δεύτερο αποτελείται από 8 χαρακτήρες ανάλυσης 10×12, και είναι αντίστοιχο με το σύνολο χαρακτήρων που προτάθηκε από τον Lippman (1988). Για την απεικόνιση των χαρακτήρων χρησιμοποιούνται μόνο δύο χρωματικές αποχρώσεις (άσπρο-μαύρο).

3.4.2 Παράμετροι του αλγορίθμου

Αλφάβητο και συνάρτηση ποιότητας σύνδεσης

Για την αναπαράσταση των χαρακτήρων, εφόσον υπάρχουν μόνο δύο χρωματικές αποχρώσεις, χρησιμοποιήθηκαν δυαδικά αντισώματα συνολικού μήκους ίσου με τον αριθμό εικονοστοιχείων (pixels) κάθε χαρακτήρα. Επομένως, το αλφάβητο των αντισωμάτων είναι το σύνολο $\Sigma = \{0, 1\}$, ενώ το μήκος τους είναι $l = 35$ και $l = 120$ για τα δύο σύνολα χαρακτήρων, αντιστοίχως. Ως συνάρτηση ποιότητας σύνδεσης

ορίσθηκε η συμπληρωματική της κανονικοποιημένης απόστασης Hamming, η οποία αποτελεί ένα μέτρο ομοιότητας δυαδικών ακολουθίων. Συγκεκριμένα, η συνάρτηση ποιότητας σύνδεσης γράφεται ως

$$f = 1 - \frac{H_d}{H_{d_{max}}}, \tag{3.2}$$

όπου H_d είναι η απόσταση Hamming μεταξύ αντισώματος-αντιγόνου, και $H_{d_{max}}$ είναι η μέγιστη δυνατή απόσταση Hamming. Η συνάρτηση f δίνει 1 στην περίπτωση που το αντίσωμα ταυτίζεται με το αντιγόνο και 0 στην περίπτωση που είναι συμπληρωματικά. Εάν θεωρήσουμε ότι ένα αντίσωμα μήκους l αποτελείται από την ακολουθία συμβόλων $a = \{a_1, a_2, \ldots, a_l\}$ και ένα αντιγόνο από την ακολουθία $g = \{g_1, g_2, \ldots, g_l\}$, τότε η απόσταση Hamming, H_d, δίνεται από την σχέση

$$H_d = \sum_{i=1}^{l} \delta_i, \text{ όπου } \delta_i = \begin{cases} 1, & \text{αν } a_i \neq g_i \\ 0, & \text{ειδ΄ άλλως} \end{cases}. \tag{3.3}$$

Από την σχέση αυτή προκύπτει επίσης ότι $H_{d_{max}} = l$, στην περίπτωση που το αντίσωμα και το αντιγόνο είναι συμπληρωματικά. Επομένως η σχέση (3.2) γίνεται

$$f = 1 - \frac{H_d}{l}, \tag{3.4}$$

όπου H_d είναι η απόσταση Hamming όπως ορίσθηκε στην εξίσωση (3.3), και l είναι το μήκος των αντισωμάτων και των αντιγόνων.

Έλεγχος κλωνοποίησης

Στο βήμα ΕΚ4 του αλγορίθμου που προτείνεται, τα αντισώματα που επιλέχθηκαν στο προηγούμενο βήμα, κλωνοποιούνται και δημιουργούν ένα σύνολο κλώνων $\mathcal{C}$. Ο αριθμός των κλώνων δεν είναι ίδιος για κάθε αντίσωμα, αλλά τα αντισώματα που συνδέονται καλύτερα με τα αντιγόνα δίνουν περισσοτέρους κλώνους. Για να υπολογισθεί ο αριθμός των κλώνων κάθε αντισώματος, τα n_b αντισώματα που επιλέχθηκαν στο βήμα ΕΚ3, ταξινομούνται κατά φθίνουσα σειρά ποιότητας σύνδεσης, έτσι ώστε για κάθε $a_j, a_{j+1} \in \mathcal{B}$ να ισχύει

$$f(a_j, g_i) \geq f(a_{j+1}, g_i),$$

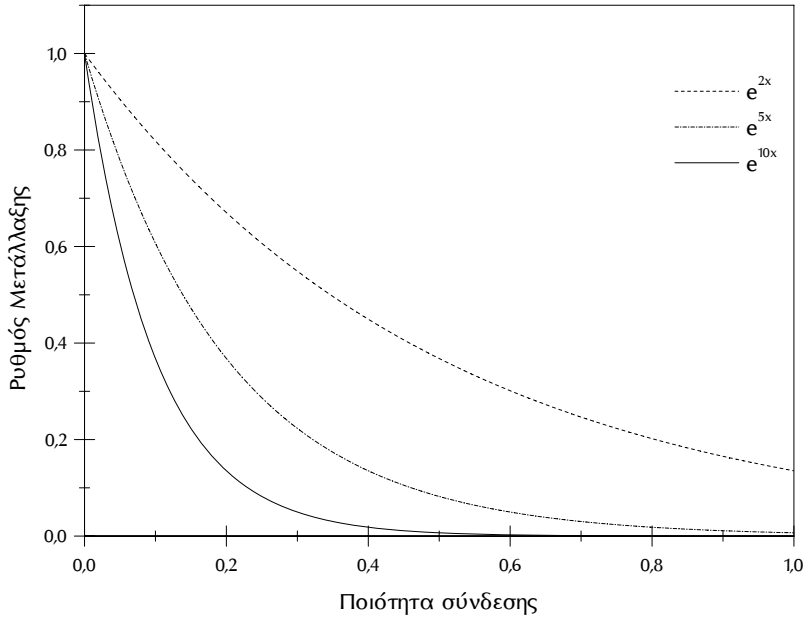
όπου g_i είναι το αντιγόνο που παρουσιάστηκε στον πληθυσμό, και f είναι συνάρτηση ποιότητας σύνδεσης. Έχοντας ταξινομήσει το σύνολο $\mathcal{B}$, ο αριθμός των κλώνων που θα δώσει το αντίσωμα a_i, δίνεται από την σχέση

$$n_i = R\left(\frac{\beta \cdot n_b}{i}\right), \quad 1 \leq i \leq n_b, \tag{3.5}$$

όπου β είναι μία σταθερά που ονομάζεται *παράγοντας κλωνοποίησης*, και R είναι η συνάρτηση στρογγυλοποίησης, η οποία μπορεί να οριστεί ως

$$R(x) = \begin{cases} \lfloor x \rfloor, & \text{αν } x - \lfloor x \rfloor < 0.5 \\ \lceil x \rceil, & \text{ειδ΄ άλλως} \end{cases}. \tag{3.6}$$

Επομένως, εάν θεωρήσουμε ότι $\beta = 10$ και $n_b = 10$, το καλύτερο αντίσωμα θα δώσει 100 κλώνους, το δεύτερο καλύτερο θα δώσει 50, κ.ο.κ. Το γεγονός ότι στην εξίσωση (3.5) ο αριθμός των κλώνων υπολογίζεται βάσει της σχετικής ποιότητας σύνδεσης των αντισωμάτων και όχι της απόλυτης είναι αρκετά ευεργετικό. Αυτό



Σχήμα 3.8: Η συνάρτηση υπολογισμού του ρυθμού μεταλλάξεων. Ο ρυθμός εξασθένησης ρ παίζει σημαντικό ρόλο στην σύγκλιση του αλγορίθμου.

έχει σαν αποτέλεσμα, ακόμα και στις πρώτες εμφανίσεις του αντιγόνου στον πληθυσμό, ο αριθμός των κλώνων που παράγονται να είναι αρκετά μεγάλος, ακόμα και αν η ποιότητά τους δεν είναι αρκετά καλή. Αυτό έρχεται σε αντίθεση με την λειτουργία του πραγματικού ανοσοποιητικού συστήματος (βλ. Σχήμα 3.2), όπου στην πρωτογενή ανοσολογική αντίδραση ο αριθμός των αντισωμάτων που παράγονται είναι αρκετά μικρότερος σε σχέση με την δευτερογενή ή τις μεταγενέστερες αντιδράσεις. Στην περίπτωση, όμως, του αλγορίθμου που περιγράφουμε, το μεγάλο πλήθος κλώνων το οποίο ούτως ή άλλως θα παραχθεί σε *κάθε* ανοσολογική αντίδραση, αυξάνει την πιθανότητα να εισαχθούν επιτυχημένες μεταλλάξεις κατά την φάση της ωρίμανσης από τις πρώτες κιόλας επαφές με το αντιγόνο, με αποτέλεσμα να έχουμε μία επιτάχυνση της σύγκλισης, με σχετικά μικρό αρχικό πληθυσμό.

Έλεγχος ρυθμού μετάλλαξης

Ο ρυθμός μετάλλαξης (βλ. Βήμα ΕΚ5) θα πρέπει να υπόκειται σε αυστηρό έλεγχο κατά την διάρκεια της ωρίμανσης των κλώνων. Οι καλύτεροι κλώνοι θα πρέπει να μεταλλάσσονται με σχετικά μικρό ρυθμό, έτσι ώστε να μην εισάγονται αναιρετικές αλλαγές. Αντιθέτως, οι κλώνοι με χαμηλή ποιότητα σύνδεσης θα πρέπει να μεταλλάσσονται με μεγαλύτερο ρυθμό, με την ελπίδα ότι κάποιες από αυτές τις αλλάγες θα είναι ευεργετικές. Για την εκτίμηση του ρυθμού μετάλλαξης α χρησιμοποιήθηκε η εκθετική συνάρτηση (Σχήμα 3.8)

$$\alpha(x) = \alpha_{max} e^{-\rho \cdot x}, \quad \alpha_{max} \leq 1, \tag{3.7}$$

όπου α_{max} είναι ο μέγιστος επιτρεπόμενος ρυθμός μετάλλαξης, ρ είναι η αντίστροφη σταθερά χρόνου της εκθετική συνάρτησης, και x είναι ποιότητα σύνδεσης αντισώματος-αντιγόνου κανονικοποιημένη στο διάστημα $[0, 1]$.

Η παράμετρος ρ αποτελεί μία από τις βασικότερες παραμέτρους του αλγορίθμου, καθ΄ ότι καθορίζει σε μεγάλο βαθμό την ταχύτητα σύγκλισης. Πολύ μικρές τιμές της, θα οδηγούν σε μεγάλους ρυθμούς μετάλλαξης ακόμα και για αρκετά καλά αντισώματα, πράγμα το οποίο γενικά θα προκαλεί χειρότερες τιμές ποιότητας σύνδεσης. Επομένως ο αλγόριθμος θα συγκλίνει αρκετά πιο αργά. Το αντίστροφο φαινόμενο συμβαίνει όταν η τιμή της ρ είναι αρκετά μεγάλη. Στην περίπτωση αυτή ο ρυθμός μεταλλάξεων μειώνεται πολύ γρήγορα, με αποτέλεσμα αντισώματα χαμηλής ποιότητας να μεταλλάσσονται με πολύ μικρούς ρυθμούς, επιβραδύνοντας έτσι την σύγκλιση του αλγορίθμου. Για το πρόβλημα της αναγνώρισης ψηφιακών χαρακτήρων που μελετάται, ιδανικές τιμές για την παράμετρο ρ είναι μεταξύ 4 και 5. Έτσι όπως έχει γραφεί η σχέση (3.7), δεν ισχύει πάντα $\alpha(1) \approx 0$, όπως θα ήταν αναμενόμενο. Κάτι τέτοιο όμως δεν έχει καμία επίδραση στην σύγκλιση του αλγορίθμου, εφόσον ο αλγόριθμος διατηρεί πάντα στην μνήμη του τα καλύτερα αντισώματα (βλ. Πρόταση 3.2). Αν κάποιος κλώνος ενός αντισώματος με μέγιστη ποιότητα σύνδεσης μεταλλαχθεί και μειωθεί η ποιότητα σύνδεσής του, δεν πρόκειται να εισέλθει στην μνήμη του αλγορίθμου. Το μόνο μειονέκτημα του γεγονότος ότι $\alpha(1) \neq 0$ είναι ότι προστίθεται άσκοπα υπολογιστικός φόρτος στο σύστημα που εκτελεί τον αλγόριθμο, καθ΄ ότι εκτελείται μία μετάλλαξη που εκ προοιμίου δεν πρόκειται να δώσει καλύτερο αποτέλεσμα. Για τον λόγο αυτό θα μπορούσε κανείς να απαιτήσει ρητά $\alpha(1) = 0$, οπότε η συνάρτηση $\alpha(x)$ θα μπορούσε να οριστεί ως

$$\alpha(x) = \begin{cases} \alpha_{max} e^{-\rho \cdot x}, & 0 \leq x < 1 \\ 0 & x = 1 \end{cases} . \tag{3.8}$$

Πολιτική ανανέωσης της μνήμης

Η πολιτική ανανέωσης της μνήμης εξαρτάται άμεσα από την σχέση του πληθαρίθμου του συνόλου μνήμης $\mathcal{M}$ και του συνόλου αντιγόνων $\mathcal{G}$, καθώς και από την απεικόνιση K μεταξύ αντιγόνων και αντισωμάτων μνήμης. Και στα δύο προβλήματα αναγνώρισης χαρακτήρων ισχύει $|\mathcal{M}| = |\mathcal{G}|$, και η πολιτική ανανέωσης είναι αρκετά απλή και άμεση. Η απεικόνιση K απεικονίζει κάθε αντιγόνο g σε ένα και μόνο αντίσωμα μνήμης m, και άρα η K είναι στην περίπτωση αυτή συνάρτηση και μάλιστα αντιστρέψιμη. Επομένως, το σύνολο $\mathcal{M}$ συσχετίζεται με το σύνολο $\mathcal{G}$ ως εξής:

$$\mathcal{M} = \{m : m = K(g),\ \forall g \in \mathcal{G}\},\ \text{ή}\ \mathcal{M} = K(\mathcal{G}).$$

Αντιστοίχως, το σύνολο $\mathcal{G}$ μπορεί να συσχετιστεί με το σύνολο μνήμης $\mathcal{M}$ βάσει της σχέσης

$$\mathcal{G} = \{g : g = K^{-1}(m),\ \forall m \in \mathcal{M}\},\ \text{ή}\ \mathcal{G} = K^{-1}(\mathcal{M}).$$

Η πολιτική ανανέωσης της μνήμης ορίζει απλά, ότι θα αντικαθίστανται πάντα εκείνο το αντίσωμα μνήμης, το οποίο υποδεικνύει η απεικόνιση K. Για πάραδειγμα έστω ότι η μνήμη αποτελείται από τα αντισώματα m_i, $i = 1, 2, 3$, και αντίστοιχα το σύνολο των αντιγόνων είναι το $\mathcal{G} = \{g_1, g_2, g_3\}$. Ορίζουμε την απεικόνιση K, έτσι ώστε

$$K(g_i) = m_i, \quad i = 1, 2, 3.$$

Έτσι, όταν στον πληθυσμό παρουσιαστεί το αντιγόνο g_1, το αντίσωμα που θα αντικατασταθεί θα είναι το m_1, όταν παρουσιαστεί το g_2, θα αντικατασταθεί το m_2, κ.ο.κ.

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΑΝΑΓΝΩΡΙΣΗΣ ΨΗΦΙΑΚΩΝ ΧΑΡΑΚΤΗΡΩΝ		
Παράμετρος	*Τιμή*	
	Πρόβλημα 1	*Πρόβλημα 2*
Μέγιστος αριθμός γενεών	500	500
Κατώφλι σύγκλισης (ϵ)	0,0	0,0
Μήκος αντισωμάτων (l)	35	120
Μέγεθος πληθυσμού ($\|\mathcal{P}\|$)	15	10
Μέγεθος μνήμης ($\|\mathcal{M}\|$)	10	8
Αντισώματα προς επιλογή (n_b)	5	5
Αντισώματα προς αντικατάσταση (n_r)	0	0
Αντισώματα προς ανανέωση (n_d)	0	0
Μέγιστος ρυμθός μετάλλαξης (α_{max})	0,7	0,7
Εξασθένηση ρυθμού μετάλλαξης (ρ)	5,0	5,0
Παράγοντας κλωνοποίησης (β)	20,0	20,0
Σύγκλιση (γενιές εξέλιξης)	$16 \pm 3{,}42$	$40 \pm 2{,}07$

Πίνακας 3.1: Οι παράμετροι του αλγορίθμου επιλογής κλώνων για την επίλυση των δύο προβλημάτων ψηφιακών χαρακτήρων του Σχήματος 3.7.

Κριτήριο σύγκλισης

Ως κριτήριο σύγκλισης του αλγορίθμου χρησιμοποιείται το κανονικοποιημένο Μέσο Τετραγωνικό Σφάλμα (ΜΤΣ ή MSE) του συνόλου μνήμης σε σχέση με το σύνολο των αντιγόνων. Το κανονικοποιημένο ΜΤΣ των δύο συνόλων ορίζεται ως

$$e = \frac{1}{n} \sum_{i=1}^{n} d_i^2, \tag{3.9}$$

όπου $n = |\mathcal{M}| = |\mathcal{G}|$, και d_i είναι η κανονικοποιημένη απόσταση Hamming του αντισώματος μνήμης m_i και του αντιγόνου g_i, για τα οποία θα πρέπει να ισχύει $m_i = K(g_i)$. Λόγω της σχέσης (3.2), ισχύει επίσης $d_i = 1 - f(a_i, g_i)$. Ο αλγόριθμος θεωρείται ότι συγκλίνει όταν

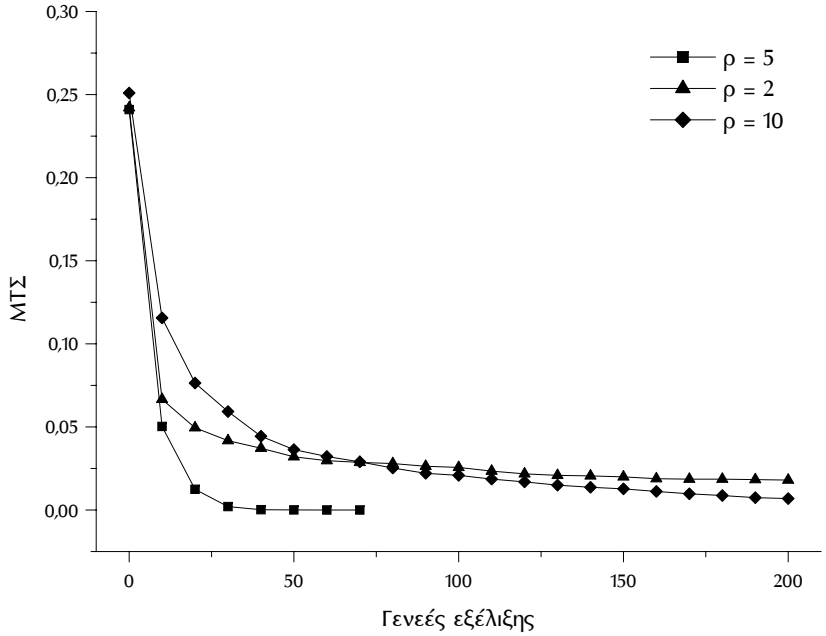
$$e = \frac{1}{n} \sum_{i=1}^{n} d_i^2 < \epsilon, \quad \epsilon > 0. \tag{3.10}$$

Το ϵ ονομάζεται *κατώφλι σύγκλισης* και αποτελεί παράμετρο του αλγορίθμου.

Στην συνέχεια σχολιάζονται τα αποτελέσματα της εκτέλεσης του αλγορίθμου και γίνεται μία σύντομη αναφορά στον τρόπο με τον οποίο οι σημαντικότερες παράμετροί του επηρεάζουν την σύγκλιση.

3.4.3 Αποτελέσματα και ανάλυση σύγκλισης

Ο αλγόριθμος που προτείνεται σε αυτό το κεφάλαιο επέδειξε πολύ καλή συμπεριφορά και στα δύο προβλήματα αναγνώρισης χαρακτήρων. Χρησιμοποιώντας τις τιμές των παραμέτρων που φαίνονται στον Πίνακα 3.1, επετεύχθη πλήρης σύγκλιση (μηδενικό κατώφλι σύγκλισης) για το μεν πρώτο πρόβλημα (βλ. Σχήμα 3.7α΄) ύστερα από κατά μέσο όρο 16 γενεές εξέλιξης με τυπική απόκλιση $3{,}42$ γενεές, ενώ για το δεύτερο πρόβλημα (βλ. Σχήμα 3.7β΄) απαιτήθηκαν κατά μέσο όρο 40 γενεές με τυπική απόκλιση $2{,}07$ γενεές. Τα αποτελέσματα αυτά είναι ιδιαίτερα ενθαρρυντικά για την ποιότητα του αλγορίθμου, ειδικά εάν αναλογισθεί κανείς ότι ο αντίστοιχος

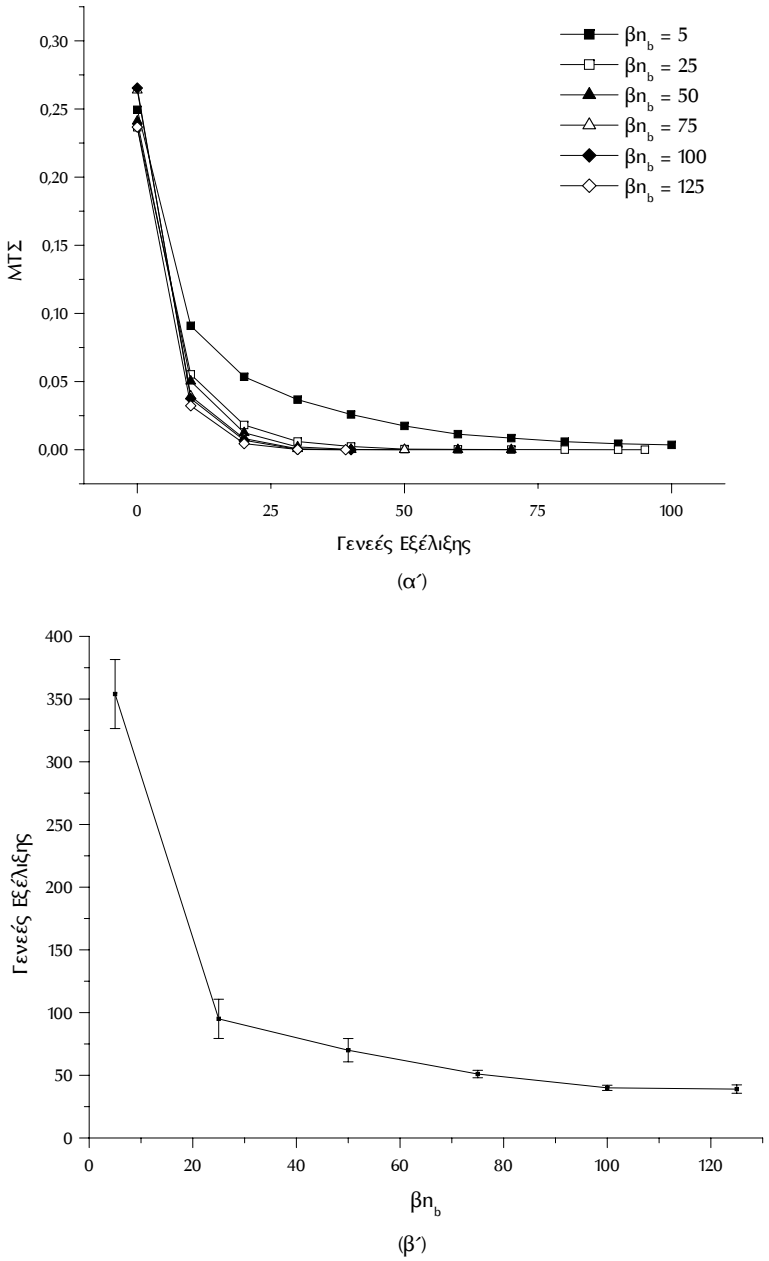


Σχήμα 3.9: Εξάρτηση της σύγκλισης του αλγορίθμου από την παράμετρο ρ. Πολύ μικρές ή πολύ μεγάλες τιμές εξασθένησης οδηγούν σε πολύ αργή σύγκλιση.

αλγόριθμος CLONALG των de Castro και Von Zuben χρειάζεται 250 γενεές για να λύσει το πρόβλημα των χαρακτήρων του Lippman. Επιπλέον, οι μικρές τιμές της τυπικής απόκλισης αποτελούν ένδειξη της σταθερότητας και της αξιοπιστίας του αλγορίθμου.

Για την επίτευξη γρήγορης σύγκλισης, όμως, είναι απαραίτητη η προσεκτική εκλογή των παραμέτρων του αλγορίθμου. Όπως ήδη ανελύθη από θεωρητικής πλευράς στην προηγούμενη παράγραφο, σημαντικός είναι ο ρόλος της παραμέτρου ρ, που δηλώνει την εξασθένηση του ρυθμού μετάλλαξης καθώς αυξάνεται η ποιότητα σύνδεσης αντισώματος-αντιγόνου. Στο Σχήμα 3.9 φαίνεται διαγραμματικά η σύγκλιση του αλγορίθμου για τρεις διαφορετικές τιμές της παραμέτρου ρ ($\rho = 2$, $\rho = 5$, και $\rho = 10$). Στο συγκεκριμένο πρόβλημα θεωρούμε ότι ο αλγόριθμος δεν συγκλίνει, εάν ξεπεράσει τις 500 γενεές εξέλιξης και δεν έχει καταφέρει να αναγνωρίσει πλήρως (μηδενικό ΜΤΣ) τα αντισώματα.

Αυτό που αξίζει να παρατηρήσει κανείς μελετώντας το διάγραμμα σύγκλισης στο Σχήμα 3.9, είναι οι καμπύλες σύγκλισης για $\rho = 2$ και $\rho = 10$, οι οποίες αν και δεν καταφέρνουν να συγκλίνουν έχουν λίγο διαφορετική μορφή. Μέχρι τις 70 γενεές περίπου υπερτερεί η καμπύλη $\rho = 2$, αλλά από εκεί και πέρα ο ρυθμός σύγκλισής της είναι πολύ μικρός, με αποτέλεσμα η καμπύλη $\rho = 10$ να παρουσιάζει καλύτερα αποτελέσματα. Κάτι τέτοιο είναι αναμενόμενο καθ΄ ότι ο μεγάλος ρυθμός μετάλλαξης που επιβάλλει η εξασθένηση $\rho = 2$ είναι ευεργετικός στις αρχικές γενεές, όπου η ποιότητα σύνδεσης αντισωμάτων-αντιγόνων είναι χαμηλή, αλλά καθίσταται ανασταλτικός παράγοντας για την περαιτέρω βελτίωση των αντισωμάτων, όταν αυτά φθασουν σε ένα ικανοποιητικό επίπεδο ποιότητας σύνδεσης. Πράγματι στις 70 γενεές το κανονικοποιημένο ΜΤΣ είναι περίπου 0,029, που αντιστοιχεί σε αντισώματα μνήμης με μέση ποιότητα σύνδεσης 83%, η οποία είναι μία ικανοποιητική τιμή. Επομένως, από εκεί και πάνω η καμπύλη $\rho = 10$ που ακολουθεί μία πιο μετριοπαθή πολιτική μεταλλάξεων, ευεργετείται περισσότερο, με αποτέλεσμα να συγκλίνει αργά αλλά σταθερά.



Σχήμα 3.10: Εξάρτηση της σύγκλισης του αλγορίθμου από το γινόμενο βn_b. Μεγάλες τιμές του οδηγούν γενικά σε καλύτερη σύγκλιση, καθ΄ ότι αυξάνεται η διαφορετικότητα του πληθυσμού των κλώνων. (α΄) Καμπύλες σύγκλισης. (β΄) Μέση σύγκλιση και τυπική απόκλιση συναρτήσει του γινομένου βn_b.

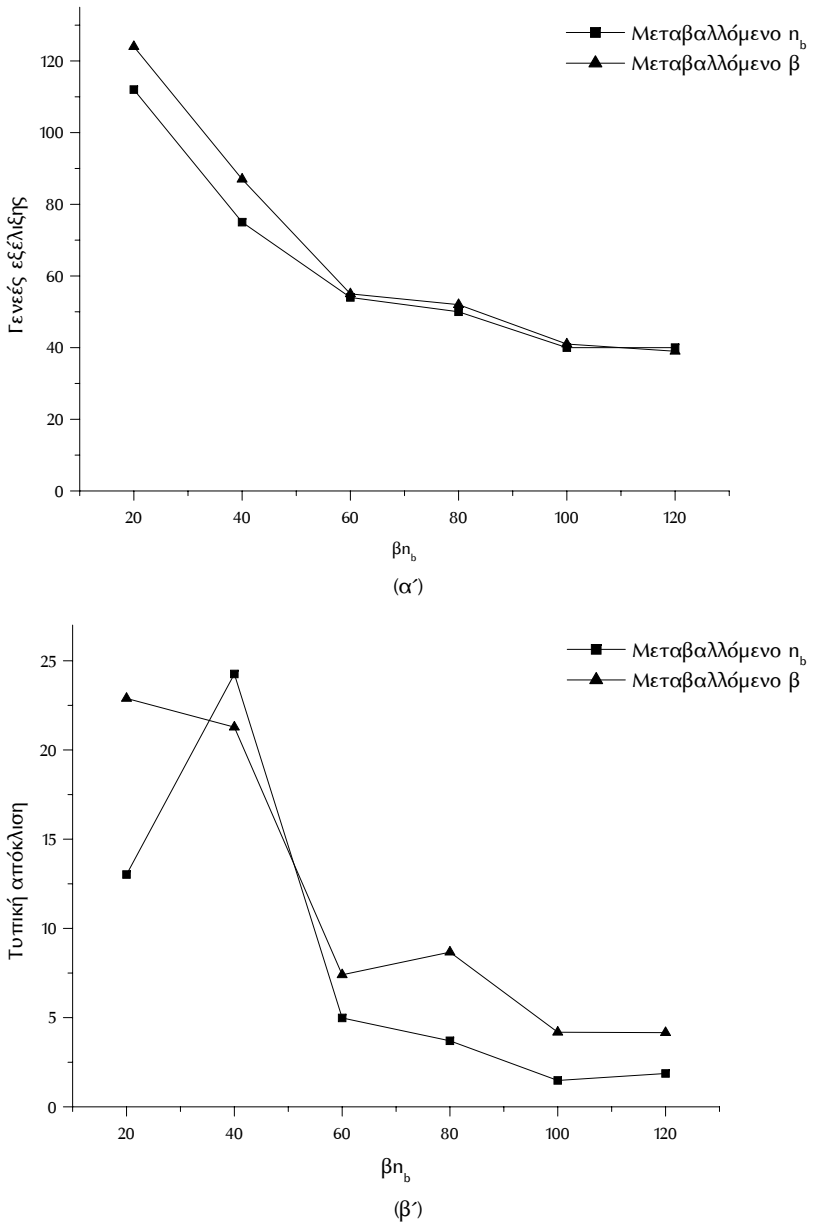
ΕΞΑΡΤΗΣΗ ΣΥΓΚΛΙΣΗΣ ΑΠΟ ΤΟ ΓΙΝΟΜΕΝΟ βn_b		
βn_b	*Σύγκλιση (γενιές εξέλιξης)*	
	$n_b = 4$	$\beta = 20$
20	124±22,89	112±13,02
40	87±21,28	75±24,26
60	55±7,40	54±4,98
80	52±8,67	50±3,70
100	41±4,18	40±1,48
120	39±4,16	40±1,87

Πίνακας 3.2: Εξάρτηση σύγκλισης από το γινόμενο βn_b.

Εξίσου σημαντικό ρόλο στην σύγκλιση του αλγορίθμου παίζει και το γινόμενο βn_b, όπου β είναι ο παράγοντας κλωνοποίησης, και n_b είναι ο αριθμός των κυττάρων που επιλέγονται για να κλωνοποιηθούν στο Βήμα ΕΚ3. Το γινόμενο αυτό καθορίζει το πλήθος των κλώνων που θα δημιουργηθούν κατά την φάση της κλωνοποίησης (βλ. Βήμα ΕΚ4), όπως υποδηλώνει και η σχέση (3.5), βάσει της οποίας υπολογίζεται ο αριθμός των κλώνων που θα δώσει κάθε επιλεχθέν κύτταρο. Όπως φαίνεται στο Σχήμα 3.10α΄ μεγαλύτερες τιμές του γινομένου αυτού οδηγούν γενικά σε γρηγορότερη σύγκλιση του αλγορίθμου. Αυτό οφείλεται στο γεγονός ότι όσο μεγαλύτερο είναι το γινόμενο βn_b, τόσο περισσότεροι θα είναι οι κλώνοι που θα δημιουργηθούν, με αποτέλεσμα οι μεταλλάξεις που θα εισαχθούν στο αμέσως επόμενο βήμα του αλγορίθμου (Βήμα ΕΚ5–ωρίμανση κλώνων) θα μπορέσουν να δημιουργήσουν ένα σύνολο μεταλλαγμένων κλώνων με μεγάλη ποικιλία και διαφορετικότητα. Αυτό έχει ως αποτέλεσμα την αύξηση της πιθανότητας εύρεσης ποιοτικών αντισωμάτων στο σύνολο αυτό, και επομένως βελτίωση της σύγκλισης.

Βέβαια, όπως μπορεί κανείς να παρατηρήσει στο Σχήμα 3.10β΄, η βελτίωση της σύγκλισης δεν μπορεί να είναι απεριόριστη. Πέρα από μία συγκεκριμένη τιμή του γινομένου βn_b, ο αλγόριθμος παύει να συγκλίνει πιο γρήγορα, ενώ παράλληλα γίνεται και πιο ασταθής (μεγαλύτερη απόκλιση). Αυτό οφείλεται στο γεγονός, ότι από ένα σημείο και πέρα ο πληθυσμός των κλώνων είναι ήδη αρκετός, για να «φιλοξενήσει» όλους σχεδόν τους διαφορετικούς και ταυτόχρονα ποιοτικούς κλώνους, που μπορούν να παραχθούν με τον δεδομένο ρυθμό μετάλλαξης. Πάντως, η αύξηση του πληθυσμού των κλώνων θα πρέπει να εκτελείται με προσοχή, καθ΄ ότι μπορεί να προσθέσει σημαντικό υπολογιστικό φόρτο στο σύστημα που εκτελεί τον αλγόριθμο, χωρίς να βελτιώσει ουσιαστικά την σύγκλιση.

Τέλος, άλλο ένα σημείο άξιο προσοχής όσον αφορά στη σχέση του γινομένου βn_b με την σύγκλιση του αλγορίθμου, είναι το πόσο μπορεί να επηρεάσει την σύγκλιση η ανεξάρτητη μεταβολή των β και n_b δεδομένης της τιμής του γινομένου. Η εξάρτηση αυτή απεικονίζεται στο Σχήμα 3.11 και στον Πίνακα 3.2, όπου φαίνεται η μέση τιμή της σύγκλισης και η τυπική απόκλιση μεταβάλλοντας μόνο μία από τις παραμέτρους β και n_b κάθε φορά. Αυτό που μπορεί να παρατηρήσει κανείς είναι, ότι αν και οι διαφορές στην μέση τιμή είναι αμελητέες, ειδικά για μεγάλες τιμές του γινομένου βn_b, οι διαφορές στην τυπική απόκλιση δεν είναι τόσο μικρές. Διατηρώντας τον παράγοντα κλωνοποίησης σταθερό και μεταβάλλοντας τον αριθμό των αντισωμάτων που επιλέγονται για κλωνοποίηση, έτσι ώστε το γινόμενο βn_b να έχει την επιθυμητή τιμή, προκύπτει μικρότερη τυπική απόκλιση (για μεγάλες τιμές του γινομένου).



Σχήμα 3.11: Ανεξάρτητη μεταβολή των β και n_b. Όταν δεν μεταβάλλονταν, οι τιμές των β και n_b ήταν: $\beta = 20$ και $n_b = 4$. (α΄) Μέση τιμή. (β΄) Τυπική απόκλιση. Μεταβολή του n_b οδηγεί σε μικρότερες τιμές.

ΚΕΦΑΛΑΙΟ 4

Εισαγωγή στον Προγραμματισμό Γονιδιακής Έκφρασης

Ο *Προγραμματισμός Γονιδιακής Έκφρασης* ή *ΠΓΕ* (*Gene Expression Programming, GEP*) είναι μία νέα εξελικτική τεχνική βελτιστοποίησης, που προτάθηκε από τον Ferreira (2001α). Αποτελεί την φυσική εξέλιξη των Γενετικών Αλγορίθμων και του Γενετικού Προγραμματισμού. Έχοντας αποβάλει τα εγγενή μειονεκτήματα των δύο αυτών τεχνικών, όπως είναι η ταύτιση γονοτύπου-φαινοτύπου, η περιορισμένη εκφραστικότητα των χρωμοσωμάτων και η περιορισμένη ευελιξία στην περίπτωση του ΓΠ, αποτελεί μία πολλά υποσχόμενη τεχνική για την επίλυση συνθέτων προβλημάτων βελτιστοποίησης. Στην συνέχεια αυτού του Κεφαλαίου επιχειρείται μία ανάλυση της τεχνικής του ΠΓΕ, όπως αυτή ορίστηκε από τον δημιουργό της (Ferreira, 2001β), έτσι ώστε ο αναγνώστης να αποκτήσει μία πλήρη εικόνα για τις δυνατότητες και τα χαρακτηριστικά αυτής της νέας μεθόδου. Στο επόμενο κεφάλαιο η μέθοδος αυτή θα συνδυασθεί με τον αλγόριθμο επιλογής των κλώνων, που παρουσιάστηκε στο Κεφάλαιο 3 (βλ. §3.4), για να πραγματοποιηθεί εξόρυξη γνώσης από σύνολα δεδομένων.

4.1 Το υπόβαθρο του ΠΓΕ

Προτού προχωρήσει κανείς στην ανάλυση της μεθόδου του ΠΓΕ, θα ήταν σκόπιμο να παρουσιαστούν σύντομα οι μέθοδοι και οι ιδέες εκείνες, που απετέλεσαν την βάση για την ανάπτυξη του ΠΓΕ. Ο ΠΓΕ βασίστηκε σε δύο προϋπάρχουσες εξελικτικές τεχνικές, συγκεκριμένα τους Γενετικούς Αλγορίθμους και τον Γενετικό Προγραμματισμό. Κοινό και βασικό στοιχείο και των τριών τεχνικών είναι, ότι αποτελούν μία απλή μοντελοποίηση της φυσικής επιλογής και εξέλιξης, που εφαρμόζεται στην φύση. Η κύρια διαφορά τους, όμως, έγκειται στην αναπαράσταση των ατομών του πληθυσμού (χρωμοσώματα) που εξελίσσουν. Στους ΓΑ τα χρωμοσώματα είναι ακολουθίες συμβόλων σταθερού μήκους, στον ΓΠ είναι μη γραμμικές οντότητες διαφορετικού μεγέθους και σχήματος (συνακτικά δένδρα–parse trees), ενώ στον ΠΓΕ πρόκειται για ακολουθίες συμβόλων σταθερού μήκους, που όμως μεταφράζονται σε μη γραμμικές οντότητες ποικίλων σχημάτων και μεγεθών (δένδρα έκφρασης–expression trees).

4.1.1 Βιολογικό υπόβαθρο

Στην παράγραφο αυτή θα θα γίνει μία σύντομη αναφορά στο βιολογικό υπόβαθρο, επί του οποίου η οικογένεια των ΓΑ στηρίζει την αναπαράσταση των ατόμων του πληθυσμού που χρησιμοποιεί. Θα παρουσιαστούν έννοιες λίγο-πολύ γνωστές, όπως είναι το χρωμόσωμα, ο γονότυπος, κ.α., αλλά και λιγότερο προφανείς όπως είναι η γονιδιακή έκφραση και τα κωδικόνια, οι οποίες υιοθετούνται από τον ΠΓΕ. Η βιολογική εξέλιξη και η φυσική επιλογή, ο άλλος θεμέλιος λίθος των ΓΑ, δεν αναλύονται στην παρούσα παράγραφο.

Το γενετικό υλικό όλων των κυττάρων και των περισσοτέρων ιών είναι το DNA, εκτός από κάποιες εξαιρέσεις ιών οι οποίοι έχουν ως γενετικό υλικό το RNA (RNA-ιοί). Οι δομικές διαφορές μεταξύ DNA και RNA είναι πολύ μικρές: τα νουκλεοτίδια[1] του RNA αντί για το μόριο δεσοξυριβόζη περιέχουν ριβόζη[2], ενώ τα μόρια RNA αντί για την βάση θυμίνη περιέχουν την βάση ουρακίλη. Οι βασικές ιδιότητες και λειτουργίες του γενετικού υλικού θα μπορούσαν να συνοψιστούν στα εξής (Αλεπόρου-Μαρίνου et al., 1999):

- *Αποθήκευση της γενετικής πληροφορίας.* Στο γενετικό υλικό περιέχονται οι πληροφορίες, που καθορίζουν όλα τα χαρακτηριστικά ενός οργανισμού. Οι πληροφορίες αυτές οργανώνονται σε λειτουργικές μονάδες, που ονομάζονται *γονίδια*.

- *Η διατήρηση και η μεταβίβαση της γενετικής πληροφορίας.* Το γενετικό υλικό των κυττάρων βάσει της ικανότητάς του να αυτοδιπλασιάζεται, μπορεί και μεταφέρεται μέσω της κυτταρικής διαίρεσης από γενεά σε γενεά κυττάρων και κατ΄ επέκταση από οργανισμό σε οργανισμό.

- *Η έκφραση των γενετικών πληροφοριών.* Το γενετικό υλικό ελέγχει την σύνθεση των πρωτεϊνών, οι οποίες και αποτελούν την έκφρασή του, καθ΄ ότι όλη η πληροφορία για την κατασκευή τους υπάρχει σ΄ αυτό.

Το γενετικό υλικό ενός κυττάρου αποτελεί το *γονιδίωμά* του. Τα σωματικά κύτταρα των ανωτέρων οργανισμών διατηρούν στον πυρήνα τους δύο αντίγραφα του γενετικού υλικού (*διπλοειδή κύτταρα*), αντίθετα στα γεννητικά κύτταρα (γαμέτες) διατηρείται μόνο ένα αντίγραφο (*απλοειδή κύτταρα*). Επιπλέον, το γενετικό υλικό των ανωτέρων οργανισμών δεν αποτελείται από ένα μόνο μόριο, αλλά από ένα σύνολο μορίων. Κάθε ένα από αυτά τα μόρια υπόκειται σε μία πολυεπίπεδη αναδίπλωση, έτσι ώστε να μπορέσει να χωρέσει στον πυρήνα του κυττάρου. Το αναδιπλωμένο αυτό μόριο αποτελεί το *χρωμόσωμα*[3].

Η πληροφορία που υπάρχει στο DNA ενός κυττάρου ή στο RNA ενός RNA-ιού, περνά στις επόμενες γενεές των κυττάρων, μέσω της ικανότητας του DNA και του RNA να αυτοδιπλασιάζονται-αντιγράφονται, ενώ μεταφέρεται στις πρωτεΐνες μέσω της διαδικασίας της *μεταγραφής* και της *μετάφρασης*. Αυτή η ροή της πληροφορίας αποτελεί και το *κεντρικό δόγμα* της Μοριακής Βιολογίας. Κατά την διαδικασία της μεταγραφής ενός γονιδίου δημιουργείται ένα μόριο RNA, βάσει της γενετικής πληροφορίας που υπάρχει σε αυτό το γονίδιο. Το μόριο αυτό είναι συμπληρωματικό του αντιστοίχου γονιδίου του DNA. Συνήθως το RNA που δημιουργήθηκε σε αυτή την φάση της μεταγραφής, δεν είναι έτοιμο να μεταφραστεί

[1] Τα *νουκλεοτίδια* είναι συμπλέγματα μορίων, τα οποία συνδέονται μεταξύ τους δημιουργώντας της αλυσίδες του γενετικού υλικού (πολυνουκλεοτιδικές αλυσίδες).

[2] Από εδώ προέρχεται και η διαφορά στην ονομασία τους: ριβονουκλεϊκό οξύ (RNA) έναντι δεσοξυριβονουκλεϊκού οξέος (DNA).

[3] Στην πραγματικότητα το DNA δεν έχει την γνωστή μορφή των χρωμοσωμάτων καθ΄ όλη την διάρκεια της ζωής του κυττάρου, αλλά μόνο κατά την φάση της διαίρεσής του. Τον υπόλοιπο χρόνο παραμένει λιγότερο πυκνό και είναι πολύ δύσκολο να διακριθεί.

σε πρωτεΐνη (*πρόδρομο RNA*), και επομένως υπόκειται σε μία διαδικασία ωρίμανσης στο εσωτερικό του πυρήνα του κυττάρου. Κατά την φάση της ωρίμανσης αποκόπτονται κάποια τμήματα του προδρόμου μορίου, οπότε προκύπτει το τελικό ώριμο μόριο (*ώριμο RNA*), το οποίο πρόκειται να μεταφραστεί σε πρωτεΐνη. Αυτή η διαδικασία αποτελεί μία από τις σημαντικότερες ανακαλύψεις της μοριακής βιολογίας, καθ΄ ότι υποδηλώνει ότι τα γονίδια των ανωτέρων οργανισμών είναι *ασυνεχή*, δηλαδή η ακολουθία βάσεων του DNA που μεταφράζεται σε πρωτεΐνη, διακόπτεται από ακολουθίες οι οποίες δεν μεταφράζονται. Τέλος, η μεταγραφή καθορίζει ποια γονίδια θα εκφραστούν σε ποια κύτταρα και σε ποιο στάδιο της ανάπτυξης του οργανισμού.

Κατά την διαδικασία της μετάφρασης το ώριμο RNA εξέρχεται από τον πυρήνα του κυττάρου και βάσει ενός κώδικα αντιστοίχισης βάσεων-αμινοξέων αποκωδικοποιείται και δημιουργείται η πρωτεΐνη. Επειδή ο αριθμος των αμινοξέων που απαρτίζουν όλες τις πρωτεΐνες είναι 20, κάθε πρωτεΐνη θα κωδικοποιείται από 3 νουκλεοτιδικές βάσεις. Οι βάσεις αυτές αποτελούν ένα *κωδικόνιο*. Επομένως, η αλληλουχία των κωδικονίων στο RNA καθορίζει την αλληλουχία των αμινοξέων στην πρωτεΐνη. Ο έλεγχος της μετάφρασης πραγματοποιείται από μία σειρά *κωδικονίων έναρξης* και από ένα *κωδικόνιο λήξης*, τα οποία όταν βρεθούν στην ακολουθία κωδικονίων του RNA, σημαίνουν την έναρξη και την λήξη της διαδικασίας της μετάφρασης, αντιστοίχως. Ο κώδικας αντιστοίχισης κωδικονίων-αμινοξέων ονομάζεται *γενετικός κώδικας* και είναι κοινός για σχεδόν κάθε οργανισμό. Ο γενετικός κώδικας έχει και άλλες σημαντικές ιδιότητες (Αλεπόρου-Μαρίνου et al., 1999), όπως ότι είναι συνεχής, μη επικαλυπτόμενος, εκφυλισμένος (ένα αμινοξύ μπορεί να κωδικοποιείται από περισσότερα του ενός κωδικόνια), κ.α.

Ολόκληρη η διαδικασία που μόλις περιγράφηκε, δηλαδή η ενεργοποίηση ενός γονιδίου, η μεταγραφή του και εν τέλει η μετάφραση του σε πρωτεΐνη, αποτελεί την *γονιδιακή έκφραση* του οργανισμού. Η γονιδιακή έκφραση δεν λαμβάνει την ίδια μορφή σε όλα τα κύτταρα, αλλά σε διαφορετικά κύτταρα εκφράζονται διαφορετικά γονίδια. Επιπλέον, σε διαφορετικούς οργανισμούς τα ίδια γονίδια μπορούν να εκφράζονται με διαφορετικούς τρόπους. Τα γονίδια αυτά, τα οποία βρίσκονται στην ίδια θέση στα χρωμοσώματα των δύο οργανισμών και ελέγχουν την ίδια ιδιότητα, ονομάζονται *αλληλόμορφα γονίδια*. Το σύνολο των αλληλομόρφων γονιδίων ενός οργανισμού αποτελεί τον *γονότυπό* του, ενώ ο τρόπος με τον οποίο αυτά τα γονίδια εκφράζονται και αλληλεπιδρούν με το περιβάλλον (εξωτερική εμφάνιση, βιοχημική σύσταση του οργανισμού, κτλ.), αποτελεί τον *φαινότυπο* του οργανισμού.

4.1.2 Γενετικοί Αλγόριθμοι

Μία σύντομη παρουσίαση των ΓΑ έγινε ήδη στο Κεφάλαιο 1 (βλ. §1.1.2), οπότε στην παράγραφο αυτή θα γίνει κυρίως αναφορά στα στοιχεία εκείνα, που μπορούν να περιορίσουν τις δυνατότητες της τεχνικής αυτής στην επίλυση συνθέτων προβλημάτων.

Ένα χρωμόσωμα ενός ΓΑ κατά την φάση της επιλογής θα επιβιώσει βασιζόμενο αποκλειστικά στην δομή και τις ιδιότητες, που το ίδιο έχει να επιδείξει. Με άλλα λόγια η συνάρτηση προσαρμογής αξιολογεί το ίδιο το χρωμόσωμα και όχι μία έκφραση του χρωμοσώματος. Επομένως, το χρωμόσωμα εκτός από τα να είναι υπεύθυνο για την μεταφορά της πληροφορίας στις επόμενες γενεές, είναι παράλληλα και το αντικείμενο της επιλογής. Με αυτόν τον τρόπο στα άτομα του πληθυσμού ενός ΓΑ ο γονότυπος ταυτίζεται με τον φαινότυπο. Αυτή η διττή συμπεριφορά των χρωμοσωμάτων είναι αρκετά περιοριστική.

Άλλος ένας περιορισμός των ΓΑ είναι το γεγονός, ότι τα χρωμοσώματα έχουν

σταθερό μήκος, πράγμα, που σε συνδυασμό με το γεγονός της ταύτισης γονοτύπου-φαινοτύπου, δεν επιτρέπει την αποκωδικοποίηση ενός μόνο μέρους του χρωμοσώματος, όπως συμβαίνει με τα πραγματικά γονίδια· η λύση του προβλήματος είναι πάντα ολόκληρο το χρωμόσωμα. Αν σε αυτό προστεθεί και το γεγονός, ότι συνήθως το αλφάβητο συμβόλων των χρωμοσωμάτων ενός ΓΑ είναι αρκετά περιορισμένο, καταλαβαίνει κανείς, ότι οι δυνατότητες αυτών των συστημάτων περιορίζονται σημαντικά. Γενικά θα μπορούσε να πει κανείς, ότι ο πληθυσμός των ΓΑ ομοιάζει πιο πολύ με ένα πληθυσμό από μόρια RNA, όπου ολόκληρη η δομή του RNA καθορίζει την λειτουργικότητά του και επομένως την προσαρμογή του στο περιβάλλον.

4.1.3 Γενετικός Προγραμματισμός

Ο Γενετικός Προγραμματισμός προτάθηκε το 1985 από τον Cramer και εξελίχθηκε λίγα χρόνια αργότερα από τον Koza (1992), με σκοπό να επιλύσει το πρόβλημα του σταθερού μεγέθους των χρωμοσωμάτων των ΓΑ. Τα χρωμοσώματα στον ΓΠ είναι μη γραμμικές οντότητες διαφορετικού μεγέθους και σχήματος. Επιπλέον, το αλφάβητο που χρησιμοποιείται από τα χρωμοσώματα αυτά, είναι σαφώς πιο πλούσιο σε σχέση με το αλφάβητο των ΓΑ, δημιουργώντας ένα πιο ευέλικτο σύστημα. Οι μη γραμμικές οντότητες που αναφέρθηκαν προηγούμενως, αποτελούν στην ουσία συντακτικά δένδρα εκφράσεων γραμμένων στο αλφάβητο των χρωμοσωμάτων. Απο βιολογικής απόψεως θα μπορούσε να πει κανείς, ότι τα χρωμοσώματα του ΓΠ ομοιάζουν περισσότερο με τις πρωτεΐνες παρά με το ίδιο το γενετικό υλικό.

Παρ΄ όλο που τα συντακτικά δένδρα επέλυσαν το πρόβλημα της περιορισμένης εκφραστικότητας των χρωμοσωμάτων των ΓΑ, η εφαρμογή των γενετικών τελεστών σε αυτά παρουσιάζει μεγάλες δυσκολίες, καθ΄ ότι είναι πολύ εύκολο να προκύψουν μη έγκυρα συντακτικά δένδρα. Επομένως, οι γενετικοί τελεστές είτε θα πρέπει να περιοριστούν είτε να επαναπροσδιορισθούν. Ο Koza ορίζει τρεις γενετικούς τελεστές για τον ΓΠ, αν και στην πράξη χρησιμοποιείται κυρίως μόνο ο ένας από αυτούς:

Ανασύνθεση (recombination) Είναι ο κυριώτερος και πιο συχνά χρησιμοποιούμενος τελεστής του ΓΠ. Ο τελεστής αυτός επιλέγει κάποια υποδένδρα των συντακτικών δένδρων των χρωμοσωμάτων των γονέων και τα ανταλλάσσει. Σκοπός αυτού του τελεστή είναι, ανταλλάσσοντας μικρά και μαθηματικώς συνεπή τμήματα, να δομείται μέσω της εξέλιξης η τελική λύση του προβλήματος.

Μετάλλαξη (mutation) Ο τελεστής της μετάλλαξης στον ΓΠ διαφέρει από τον αντίστοιχο τελεστή στους ΓΑ, καθ΄ ότι δεν προκαλεί σημειακές μεταλλάξεις στα χρωμοσώματα. Αντιθέτως, επιλέγει ένα κόμβο στο δένδρο του χρωμοσώματος και αντικαθιστά ολόκληρο το υπόδενδρο που ξεκινά από αυτόν τον κόμβο με ένα εντελώς νέο υπόδενδρο, το οποίο έχει δημιουργηθεί με τυχαίο τρόπο.

Μετάθεση (permutation) Ο τελεστής της μετάθεσης είναι καινούργιος στον ΓΠ. Επιλέγει δύο κόμβους, οι οποίοι είναι δομικά ισοδύναμοι (είτε δύο τερματικούς κόμβους είτε δύο κόμβους με συναρτησιακά σύμβολα ιδίου βαθμού) και τους ανταλλάσσει.

Το βασικό μειονέκτημα του ΓΠ έγκειται στην αδυναμία δραστικής τροποποίησης των χρωμοσωμάτων, όσο πλούσιο και αν είναι το αλφάβητό τους. Κανένας από τους τελεστές που μόλις περιγράφηκαν, δεν μπορεί να τροποποιήσει σε μεγάλο

βαθμό την δομή των χρωμοσωμάτων. Μάλιστα η μετάλλαξη και η μετάθεση δεν την τροποποιούν καθόλου. Εάν λάβει κανείς τέλος υπ΄ όψη και το γεγονός, ότι και στην περίπτωση του ΓΠ δεν υπάρχει διαχωρισμός μεταξύ γονοτύπου και φαινοτύπου, δεν θα είναι δύσκολο να διαπιστώσει ότι και ο ΓΠ προβάλλει σοβαρούς περιορισμούς.

4.2 Προγραμματισμός Γονιδιακής Έκφρασης

Ο Προγραμματισμός Γονιδιακής Έκφρασης έρχεται να καλύψει τις αδυναμίες των ΓΑ και του ΓΠ, και παράλληλα να συνδυάσει τα θετικά στοιχεία των δύο τεχνικών. Βασικότερο στοιχείο του είναι, ότι καταργεί την ταύτιση γονοτύπου-φαινοτύπου, από την οποία δεσμεύονταν οι προηγούμενες τεχνικές· υπάρχει πλέον σαφής διαχωρισμός μεταξύ των δύο εκφράσεων, όπως ακριβώς συμβαίνει και στην φύση. Τα χρωμοσώματα στον ΠΓΕ είναι σταθερού μήκους όπως στους ΓΑ, διατηρώντας έτσι το πλεονέκτημα της απλότητας και της ευκολίας χειρισμού τους, αλλά εκφράζονται σε μη γραμμικές οντότητες, αντίστοιχες με τα συντακτικά δένδρα του ΓΠ. Στην περίπτωση αυτή οι μη γραμμικές οντότητες ονομάζονται *δένδρα έκφρασης (ΔΕ)*. Για την ανάγωση και έκφραση της γενετικής πληροφορίας που είναι αποθηκευμένη στα χρωμοσώματα του ΠΓΕ, ο Ferreira ανέπτυξε μία νέα μαθηματική γλώσσα, την οποία ονόμασε Karva. Τα χρωμοσώματα του ΠΓΕ είναι επιπλέον δομημένα με τέτοιο τρόπο, ώστε να εφαρμόζονται όλοι οι γενετικοί τελεστές των ΓΑ χωρίς κανένα σχεδόν περιορισμό, αλλά τα δένδρα έκφρασης να παραμένουν έγκυρα.

Ο ΠΓΕ, όπως θα φανεί και από την ανάλυση που ακολουθεί, είναι μία απλή και αποδοτική ιδέα, η οποία μιμείται αρκετά καλά τον τρόπο, με τον οποίο η γενετική πληροφορία μεταφέρεται από το DNA στις πρωτεΐνες. Η ακολουθία συμβόλων των χρωμοσωμάτων καθορίζει επακριβώς το ΔΕ του χρωμοσώματος, όπως ακριβώς τα κωδικόνια καθορίζουν την ακολουθία των αμινοξέων. Επιπλέον, το χρωμόσωμα του ΠΓΕ μεταφράζεται σε ΔΕ σύμβολο προς σύμβολο, ενώ παράλληλα εάν δοθεί το ΔΕ, μπορεί να καθοριστεί το χρωμόσωμα που το δημιούργησε. Αυτές οι δύο υποστάσεις των χρωμοσωμάτων, η ακριβής ακολουθία και η έκφρασή της, αποτελούν δύο πολύ ισχυρά εργαλεία του ΠΓΕ, δίνοντας του σαφές πλεονέκτημα έναντι των ΓΑ και του ΓΠ.

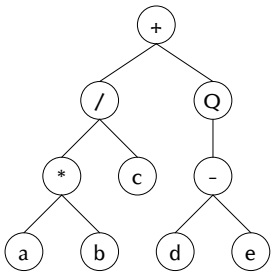
4.2.1 Το γονιδίωμα του ΠΓΕ

Το χρωμόσωμα του ΠΓΕ είναι μία ακολουθία συμβόλων σταθερού μήκους και μπορεί να περιέχει ένα ή περισσότερα γονίδια. Παρ΄ όλο που το χρωμόσωμα είναι σταθερού μήκους, μπορεί να μεταφράζεται σε ΔΕ διαφορετικού σχήματος και μεγέθους.

Γονότυπος και φαινότυπος

Η δομή των γονιδίων του ΠΓΕ[4] γίνεται καλύτερα κατανοητή βάσει των ανοικτών πλαισίων ανάγνωσης. Στην βιολογία με τον όρο *ανοικτό πλαίσιο ανάγνωσης ή ΑΠΑ (open reading frame ή ORF)* υπονοείται η διαδρομή με βήμα ενός κωδικονίου (τριπλέτας βάσεων) από το κωδικόνιο έναρξης στο κωδικόνιο λήξης του RNA κατά την φάση της μετάφρασης. Το γονίδιο όμως που παρήγαγε αυτό το μόριο RNA, μπορεί να είναι σαφώς μεγαλύτερο, καθώς το τελικό μόριο RNA είναι το προϊόν μίας διαδικασίας ωρίμανσης (βλ. §4.1.1). Επιπλέον ακόμα και το ώριμο RNA δεν μεταφράζεται ολόκληρο σε πρωτεΐνη, αλλά έχει δύο περιοχές στην αρχή και στο

[4]Στην συνέχεια τα γονίδια του ΠΓΕ θα αναφέρονται ενίοτε και ως ΠΓΕ-γονίδια.



Σχήμα 4.1: Το συντακτικό δένδρο της έκφρασης $\frac{a \cdot b}{c} + \sqrt{d - e}$.

τέλος του, οι οποίες δεν μεταφράζονται[5]. Εντελώς αντίστοιχα στον ΠΓΕ δεν αποκωδικοποιείται ολόκληρο το γονίδιο σε ΔΕ, αλλά μόνο ένα τμήμα του. Το κωδικόνιο έναρξης είναι πάντοτε το πρώτο σύμβολο του γονιδίου, αλλά η θέση του κωδικονίου λήξης δεν είναι σταθερή, δημιουργώντας με τον τρόπο αυτό ΔΕ διαφορετικού μεγέθους και σχήματος. Επομένως, κάθε γονίδιο του ΠΓΕ μπορεί να περιέχει στο τέλος του μία αμετάφραστη περιοχή. Αγνοώντας προς το παρόν την αμετάφραστη περιοχή των γονιδίων του ΠΓΕ, θα μελετηθεί το πώς η γλώσσα Karva απεικονίζει ένα γονίδιο σε ΔΕ, και αντιστρόφως.

Έστω η αλγεβρική έκφραση

$$\frac{a \cdot b}{c} + \sqrt{d - e}.$$

Η έκφραση αυτή μπορεί εύκολα να αναπαρασταθεί ως το ΔΕ που φαίνεται στο Σχήμα 4.1, όπου Q είναι η συνάρτηση της τετραγωνικής ρίζας. Ο γονότυπος που παρήγαγε αυτό το ΔΕ, μπορεί να κατασκευαστεί εύκολα, εάν διατρέξει κανείς το ΔΕ κατά πλάτος:

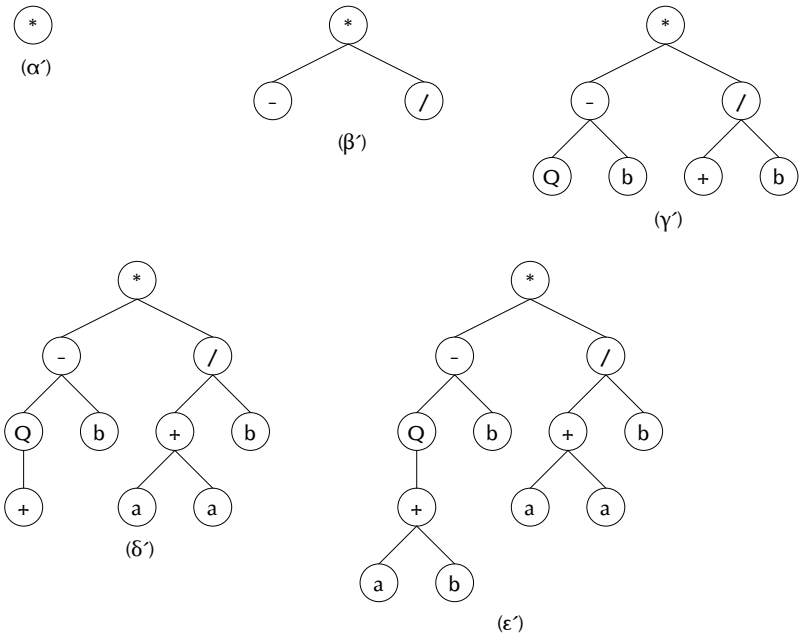
```
0123456789
+/Q*c-abde
```

Η παράσταση αυτή αποτελεί ένα ΑΠΑ μήκους 10, το οποίο αρχίζει με το σύμβολο + και τερματίζεται με το σύμβολο e. Η παράσταση αυτή ονομάζεται *παράσταση-K*, λόγω της γλώσσας Karva. Αντιστρόφως, έστω η παράσταση-K:

```
012345678901
*-/Qb+b+aaab
```

Η μετατροπή της σε ΔΕ είναι απλή και άμεση (Σχήμα 4.2). Στην θέση της ρίζας τοποθετείται το σύμβολο που βρίσκεται στην θέση 0 του γονιδίου, δηλαδή το * (Σχήμα 4.2α΄). Επειδή το σύμβολο είναι συναρτησιακό, και ο βαθμός του είναι 2, ο κόμβος * θα έχει δύο παιδιά, τα οποία θα είναι τα σύμβολα στις θέσεις 1 και 2 (Σχήμα 4.2β΄). Τα δύο νέα σύμβολα που προστέθηκαν έχουν βαθμό 2 το καθένα, οπότε στο επόμενο επίπεδο του ΔΕ θα εισαχθούν 4 κόμβοι. Τα σύμβολα των κόμβων αυτών θα είναι τα σύμβολα στις θέσεις 3–6 του ΠΓΕ-γονιδίου. Με αντίστοιχο τρόπο κατασκευάζεται και το υπόλοιπο ΔΕ όπως φαίνεται στο Σχήμα 4.2. Όλοι οι κόμβοι με τερματικά σύμβολα αποτελούν τα φύλλα του δένδρου, και δεν προστίθενται επιπλέον παιδιά σε αυτούς. Η κατασκευή του ΔΕ ολοκληρώνεται, όταν στο τελευταίο επίπεδο υπάρχουν μόνο τερματικά σύμβολα.

[5]Οι περιοχές αυτές ονομάζονται *5-3 αμετάφραστες περιοχες*.



Σχήμα 4.2: Τα βήματα της κατασκευής ενός ΔΕ από ένα ΠΓΕ-γονίδιο. Η κατασκευή γίνεται ανά επίπεδο και διατρέχοντας σύμβολο προς σύμβολο το γονίδιο. Εδώ παρουσιάζεται η κατασκευή του ΔΕ που προκύπτει από το γονίδιο: `*-/Qb+b+aaab`.

Η γλώσσα Karva παρέχει μεγάλη απλότητα στην αναπαράσταση των γονιδίων και των ΔΕ, και παράλληλα προσφέρει μεγάλη ευκολία στην μετατροπή των εκφράσεων από την μία μορφή στην άλλη. Το μεγάλο όμως πλεονέκτημα του ΠΓΕ είναι η ευελιξία, που προσφέρουν τα μεταβλητά ΑΠΑ σε συνδυασμό με το σταθερό μήκος των γονιδίων. Ένα ΑΠΑ στον ΠΓΕ μπορεί να έχει μήκος που κυμαίνεται από ένα σύμβολο (το κωδικόνιο έναρξης ταυτίζεται με το κωδικόνιο λήξης σε αυτή την περίπτωση), έως ολόκληρο το ΠΓΕ-γονίδιο, δίνοντας αντίστοιχα απλά ή σύνθετα ΔΕ. Με κατάλληλη επιλογή του μήκους των γονιδίων, όπως θα φανεί στην επόμενη παράγραφο, τα ΔΕ που θα προκύπτουν από τα ΑΠΑ θα είναι πάντα έγκυρα. Ο λόγος ύπαρξης της αμετάφραστης περιοχής στο τέλος των ΠΓΕ-γονιδίων είναι να διατηρεί το μέγεθος των γονιδίων σταθερό και ανεξάρτητο από το μήκος του ΑΠΑ. Με αυτό τον τρόπο επιτρέπεται η εφαρμογή των ίδιων γενετικών τελεστών που χρησιμοποιούνται και στους ΓΑ χωρίς περιορισμούς και ειδικές παραδοχές, όπως συμβαίνει στον ΓΠ.

Το γονίδιο του ΠΓΕ

Το γονίδιο του ΠΓΕ αποτελείται από δύο μέρη: την *κεφαλή* και την *ουρά*. Στην κεφαλή του γονιδίου μπορούν να εμφανίζονται είτε συναρτησιακά είτε τερματικά σύμβολα. Αντιθέτως, στην ουρά μπορούν να εμφανίζονται μόνο τερματικά σύμβολα. Τα μεγέθη αυτών των δύο περιοχών του ΠΓΕ-γονιδίου δεν εκλέγονται αυθαίρετα, αλλά το μέγεθος της ουράς εξαρτάται από το μέγεθος της κεφαλής και από τον μέγιστο βαθμό των συναρτησιακών συμβόλων. Τα δύο αυτά μεγέθη συνδέονται μέσω της σχέσης

$$t = h(n - 1) + 1, \tag{4.1}$$

όπου t είναι το μέγεθος της ουράς, h το μέγεθος της κεφαλής, και n είναι ο μέγιστος βαθμός των συναρτησιακών συμβόλων του αλφαβήτου του ΠΓΕ. Συνήθως, κατά την επίλυση ενός προβλήματος με τον ΠΓΕ, εκλέγεται αυθαίρετα το μήκος h της κεφαλής των γονιδίων, και το μήκος της ουράς υπολογίζεται βάσει της σχέσης (4.1). Το ολικό μήκος του γονιδίου θα είναι επομένως

$$L = h + t \tag{4.2}$$

ή αντικαθιστώντας την σχεση (4.1)

$$L = h \cdot n + 1. \tag{4.3}$$

Η τελευταία σχέση υποδηλώνει ουσιαστικά, ότι το μέγεθος του ΠΓΕ-γονιδίου είναι αρκετό, για να μπορέσει να κωδικοποιήσει μία έκφραση, στην οποία θα χρησιμοποιηθούν h συναρτησιακά σύμβολα με τον μέγιστο βαθμό (ο παράγοντας 1 στο άθροισμα υποδηλώνει την ρίζα του ΔΕ, η οποία δεν μετράται στο γινόμενο $h \cdot n$). Με αυτό τον τρόπο επομένως εξασφαλίζεται, ότι τα ΠΓΕ-γονίδια θα κωδικοποιούν πάντοτε συντακτικά ορθά ΔΕ.

Έστω για παράδειγμα ένα ΠΓΕ-γονίδιο με μήκος κεφαλής $h = 15$, σύνολο συναρτησιακών συμβόλων $F = \{Q, *, /, -, +\}$ και σύνολο τερματικών συμβόλων $T = \{a, b\}$. Ο μέγιστος βαθμός των συναρτησιακών συμβόλων είναι $n = 2$, οπότε στην περίπτωση αυτή το μήκος της ουράς, βάσει της σχέσης (4.1), θα είναι $t = 15(2-1)+1 = 16$. Επομένως το γονίδιο θα έχει μήκος $L = 15 + 16 = 31$. Ένα συγκεκριμένο γονίδιο με αυτά τα χαρακτηριστικά μπορεί να είναι το ακόλουθο, το οποίο μεταφράζεται στο ΔΕ του Σχήματος 4.3α΄ (η ουρά του γονιδίου διακρίνεται με πλάγια γραμματοσειρά):

```
012345678901234567890123456789 0
/aQ/b*ab/Qa*b*-abababbabbbbbba
```

Στην περίπτωση αυτή το ΑΠΑ που μεταφράζεται σε ΔΕ, ξεκινά από την θέση 0 και τελειώνει στην θέση 7, παρ΄ όλο που ολόκληρο το γονίδιο τελειώνει στην θέση 30. Εάν υποθέσει κανείς, ότι συμβαίνει μία μετάλλαξη στην θέση 2 του γονιδίου αλλάζοντας το σύμβολο `Q` στο `+`, τότε η δομή του ΔΕ αλλάζει ριζικά, όπως φαίνεται στο Σχήμα 4.3β΄ και το γονίδιο γίνεται:

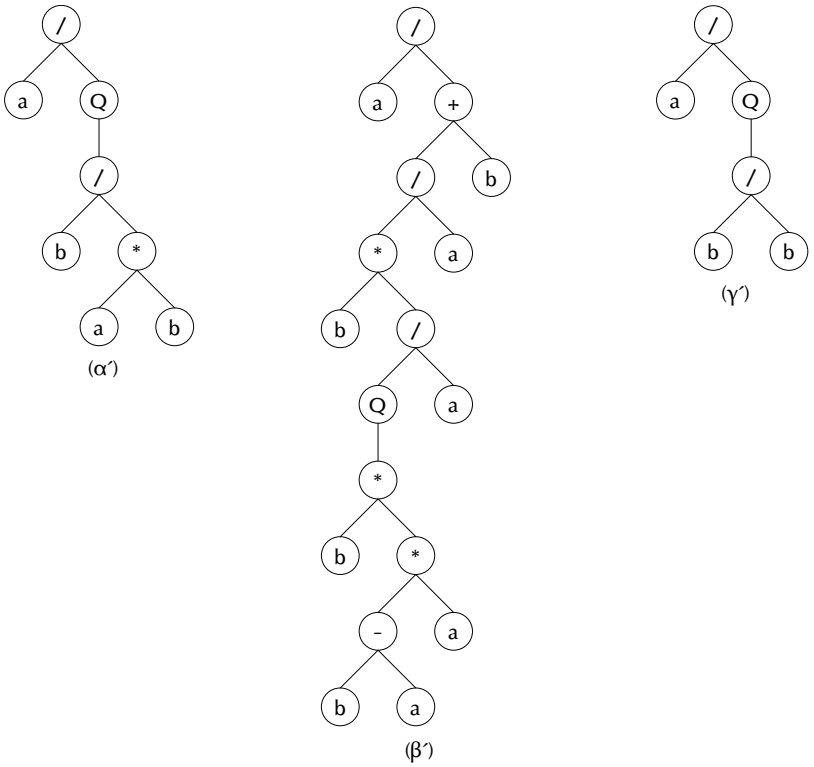
```
012345678901234567890123456789 0
/a+/b*ab/Qa*b*-abababbabbbbbba
```

Στην περίπτωση αυτή το ΑΠΑ τελειώνει στην θέση 17. Εντελώς αντίστοιχα μία μετάλλαξη θα μπορούσε να μειώσει το μήκος του ΑΠΑ. Για παράδειγμα μεταλλάσσοντας το σύμβολο στην θέση 5 του αρχικού γονιδίου από `*` σε `b`, το γονίδιο γίνεται:

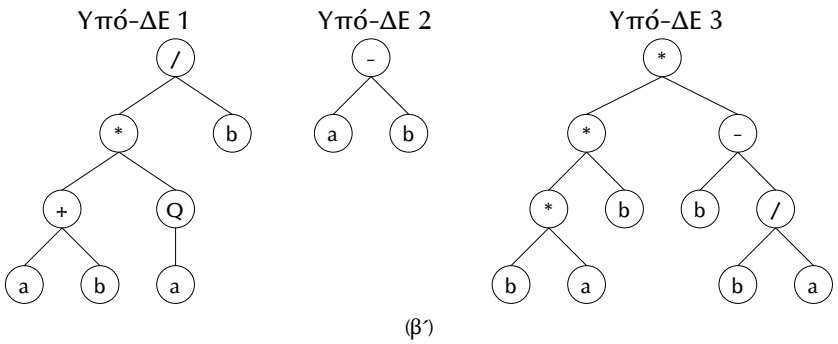
```
012345678901234567890123456789 0
/a+/bbab/Qa*b*-abababbabbbbbba
```

και το ΑΠΑ τελειώνει πλέον στην θέση 5, ενώ το ΔΕ είναι αυτό που φαίνεται στο Σχήμα 4.3γ΄.

Από αυτό το παράδειγμα γίνεται φανερή η εξαιρετική ευελιξία των γονιδίων του ΠΓΕ. Το κάθε γονίδιο μπορεί να παραγάγει ένα ΔΕ με μεταβλητό αριθμό κόμβων, ο οποίος κυμαίνεται από ένα κόμβο, όταν το πρώτο σύμβολο του γονιδίου είναι τερματικό, έως τόσους κόμβους όσο είναι το μέγεθος του γονιδίου, όταν όλες οι θέσεις της κεφαλής του γονιδίου καταλαμβάνονται από συναρτησιακά σύμβολα με τον μέγιστο βαθμό. Τέλος αυτό που αξίζει να παρατηρήσει κανείς, είναι ότι οι



Σχήμα 4.3: Μετάλλαξη ΠΓΕ-γονιδίων. Ακόμη και η μετάλλαξη ενός μόνο σημείου μπορεί να επιφέρει δραστικές αλλαγές στην δομή του ΔΕ. Παρ΄ όλα αυτά παραμένει πάντα συντακτικά ορθό. (α΄) Το ΔΕ του αρχικού γονιδίου (`/aQ/b*ab/Qa*b*-`*ababaababbabbbba*). (β΄) Το ΔΕ ύστερα από την αλλαγή του συμβόλου `Q` στήν θέση 2, στο σύμβολο `+`. (γ΄) Το ΔΕ ύστερα από την αλλαγή του συμβόλου `*t` στήν θέση 5, στο σύμβολο `b`.



Σχήμα 4.4: Έκφραση των ΠΓΕ-γονιδίων ως υπο-ΔΕ. (α΄)Ένα χρωμόσωμα με 3 γονίδια. Κάθε γονίδιο ξεκινά από την θέση 0, ενώ οι ουρές των γονιδίων σημαίνονται με πλάγια γραμματοσειρά. (β΄) Τα υπο-ΔΕ των γονιδίων του χρωμοσώματος.

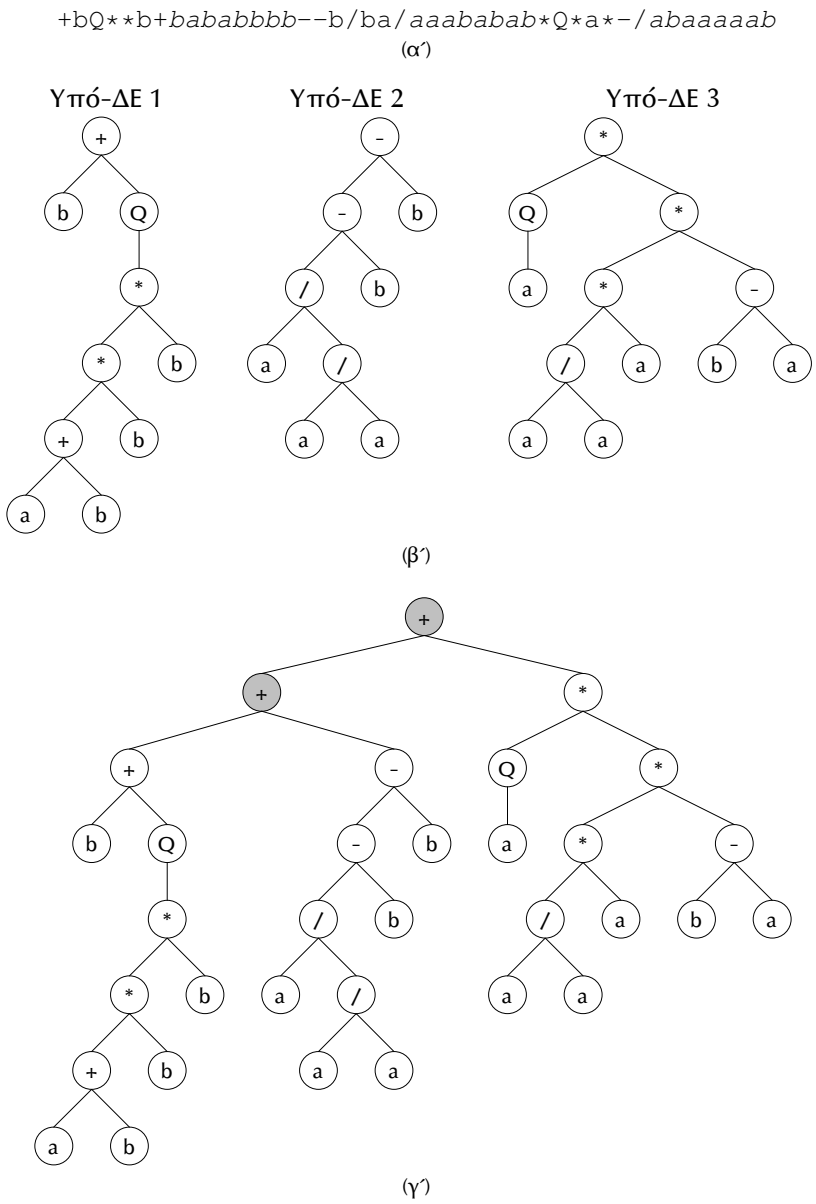
μεταλλάξεις που συνέβησαν, ήταν σημειακές μεταλλάξεις στο ΠΓΕ-γονίδιο, όπως ακριβώς συμβαίνει στους ΓΑ, χωρίς μάλιστα να υπόκεινται σε κάποιον σοβαρό περιορισμό. Ο μόνος περιορισμός είναι, ότι θα πρέπει να σέβονται την δομή του γονιδίου, δηλαδή να μην εισάγουν συναρτησιακά σύμβολα στην ουρά. Περισσότερα όμως για τους γενετικούς τελεστές του ΠΓΕ θα ειπωθούν στην σχετική παράγραφο.

Χρωμοσώματα με πολλά γονίδια

Ένα χρωμόσωμα του ΠΓΕ μπορεί να περιέχει περισσότερα του ενός γονίδια. Η διάταξη των γονιδίων μέσα στο χρωμόσωμα είναι σειριακή, ενώ το κάθε γονίδιο διατηρεί την δομή που περιγράφηκε στην προηγούμενη παράγραφο. Κάθε γονίδιο του χρωμοσώματος μπορεί να λειτουργεί εντελώς ανεξάρτητα, δημιουργώντας ένα ξεχωριστό ΔΕ (Σχήμα 4.4). Το ΔΕ αυτό ονομάζεται υπόδενδρο έκφρασης ή υπο-ΔΕ, καθ΄ ότι είναι το ΔΕ ενός μόνο γονιδίου και όχι ολοκλήρου του χρωμοσώματος. Τα υπο-ΔΕ συνήθως δεν είναι ασύνδετα, αλλά εφαρμόζεται σε αυτά μία συνάρτηση σύνδεσης, η οποία τα συνδυάζει και σχηματίζει το τελικό ΔΕ του χρωμοσώματος. Έστω για παράδειγμα το εξής χρωμόσωμα που περιεχει τρία γονίδια:

```
012345678901234012345678901234012345678901234
+bQ**b+bababbbb--b/ba/aaababab*Q*a*-/abaaaaab
```

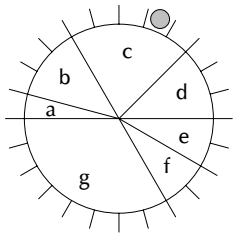
Τα υπο-ΔΕ αυτών των γονιδίων φαίνονται στο Σχήμα 4.5β΄, ενώ στο Σχήμα 4.5γ΄ φαίνεται το τελικό ΔΕ, το οποίο προκύπτει από την εφαρμογή της συνάρτησης της πρόσθεσης στα επιμέρους υπο-ΔΕ των γονιδίων του χρωμοσώματος. Η συνάρτηση σύνδεσης αποτελεί και αυτή μία παράμετρο του ΠΓΕ και καθορίζεται εξ αρχής, όπως συμβαίνει με το μήκος της κεφαλής των γονιδίων και τα σύνολα συμβόλων. Το είδος της συνάρτησης σύνδεσης που θα χρησιμοποιηθεί, εξαρτάται τόσο από το προς επίλυση πρόβλημα, όσο και από το σύνολο των συναρτησιακών συμβόλων που χρησιμοποιείται. Συνήθως, όταν το σύνολο συναρτησιακών συμβόλων αποτελείται από αλγεβρικές συναρτήσεις, όπως συμβαίνει στην περίπτωση που μελετήθηκε, τότε ως συνάρτηση σύνδεσης χρησιμοποιείται είτε η πρόσθεση είτε ο πολλαπλασιασμός. Από την άλλη εάν χρησιμοποιούνται λογικές συναρτήσεις, όπως οι IF, AND, NOT, κλπ., τότε προτιμώνται οι συναρτήσεις IF και OR. Η συνάρτηση



Σχήμα 4.5: Το ΔΕ ενός χρωμοσώματος με πολλά γονίδια. Κατασκευάζονται πρώτα τα υπό-ΔΕ των γονιδίων και ύστερα συνδέονται με την συνάρτηση σύνδεσης. (α΄) Ένα χρωμόσωμα με 3 γονίδια. (β΄) Τα υπο-ΔΕ των γονιδίων. (γ΄) Το ΔΕ ολοκλήρου του χρωμοσώματος. Η συνάρτηση σύνδεσης είναι οι σκιασμένοι κόμβοι.

Προσαρμογή πληθυσμού:

$$F(a) = 1$$
$$F(b) = 3$$
$$F(c) = 5$$
$$F(d) = 3$$
$$F(e) = 2$$
$$F(f) = 2$$
$$F(g) = 8$$



Σχήμα 4.6: Το σχήμα επιλογής του τροχού τύχης, όπου F είναι η συνάρτηση προσαρμογής και $\mathcal{P} = \{a, b, c, d, e, f, g\}$ είναι ο πληθυσμός. Στην συγκεκριμένη περιστροφή του τροχού επιλέχθηκε το μέλος c για αναπαραγωγή.

IF μπορεί να οριστεί ως μία συνάρτηση I με τρία ορίσματα

$$I(x, y, z) = \begin{cases} y, & x = 0 \\ z, & x \neq 0 \end{cases} . \tag{4.4}$$

4.2.2 Γενετικοί τελεστές

Η ευελιξία που προσφέρει η γλώσσα Karva στην αναπαράσταση των γονιδίων του ΠΓΕ και των ΑΠΑ, καθώς επίσης το σταθερό μέγεθος των γονιδίων και των χρωμοσωμάτων, επιτρέπουν την ανάπτυξη μίας πληθώρας γενετικών τελεστών για τον ΠΓΕ χωρίς σοβαρούς περιορισμούς. Ο Ferreira (2001β) ορίζει τέσσερις γενετικούς τελεστές για τον ΠΓΕ, μεταξύ των οποίων βρίσκεται και ο τελεστής της επιλογής και αναπαραγωγής των χρωμοσωμάτων. Οι υπόλοιποι είναι η *μετάλλαξη* και η *ανασύνθεση*, οι οποίοι ορίζονται παρόμοια με τους αντιστοίχους τελεστές των ΓΑ, ενώ προστίθεται και ένας νέος τελεστής, η *μετατόπιση*.

Επιλογή και αναπαραγωγή

Ο ΠΓΕ χρησιμοποιεί ως σχήμα επιλογής και αναπαραγωγής των ατόμων του πληθυσμού το σχήμα του *τροχού τύχης (roulette wheel selection)*. Το σχήμα αυτό είναι στοχαστικό και επιλέγει τους καλυτέρους υποψηφίους βάσει μίας τυχαίας διαδικασίας. Σχηματικά ο τρόπος που λειτουργεί η επιλογή βάσει του τροχού τύχης, φαίνεται στο Σχήμα 4.6 (Xavier, 1997). Ο τροχός τύχης χωρίζεται σε τομείς και σε κάθε μέλος του πληθυσμού ανατίθεται ένας αριθμός τομέων ανάλογα με την τιμή της προσαμογής του. Στην συνέχεια εκλέγεται με τυχαίο τρόπο ένας τομέας του τροχού και το μέλος του πληθυσμού, στο οποίο ανήκει ο τομέας, επιλέγεται προς αναπαραγωγή. Η διαδικασία αυτή επαναλαμβάνεται τόσες φορές, όσο είναι το μέγεθος του πληθυσμού, οπότε μετά το πέρας της έχει δημιουργηθεί μία νέα γενεά ατόμων. Η τεχνική του τροχού τύχης μιμείται με αρκετά καλό τρόπο την φυσική επιλογή, καθ΄ ότι τα πιο καλά προσαρμοσμένα άτομα του πληθυσμού θα δώσουν γενικά περισσοτέρους απογόνους, ενώ παράλληλα δίνεται η δυνατότητα αναπαραγωγής και σε λιγότερο προσαρμοσμένα άτομα. Στην πράξη η τεχνική αυτή έχει επιδείξει πολύ καλή συμπεριφορά.

Μετάλλαξη

Οι μεταλλάξεις που συμβαίνουν στον ΠΓΕ, είναι εντελώς αντίστοιχες με τις σημειακές μεταλλάξεις, που συμβαίνουν στους ΓΑ. Ο μόνος περιορισμός έγκειται στο γεγονός, ότι πρέπει πάντα να διατηρείται η δομή του ΠΓΕ-γονιδίου. Έτσι, στην κεφαλή του γονιδίου οποιοδήποτε σύμβολο (τερματικό ή μη) μπορεί να αλλάξει σε οποιοδήποτε άλλο (τερματικό ή μη), αλλά στην ουρά επιτρέπονται αλλαγές μόνο μεταξύ τερματικών συμβόλων. Αυτός ο περιορισμός εξασφαλίζει, ότι ανεξαρτήτως του πλήθους των μεταλλάξεων το ΠΓΕ-γονίδιο θα μεταφράζεται πάντα σε ένα έγκυρο ΔΕ. Έστω για παράδειγμα το εξής χρωμόσωμα, το οποίο περιέχει τρία γονίδια:

```
0123456789001234567890012345678900123456789
Q+bb*bbbaba-**--abbbaaQ*a*Qbbbaab
```

Εάν υποθέσει κανείς, ότι γίνονται τρεις μεταλλάξεις, έστω στην θέση 4 του γονιδίου 1, στην θέση 0 του γονιδίου 2 και στην θέση 2 του γονιδίου 3, και τα νέα σύμβολα που εισάγονται, είναι αντιστοίχως /, Q και +, τότε το προκύπτον χρωμόσωμα θα είναι (τα σημεία μετάλλαξης παρουσιάζονται με έντονη γραμματοσειρά):

```
0123456789001234567890012345678900123456789
Q+bb/bbbabaQ**--abbbaaQ*+*Qbbbaab
```

Αυτό που αξίζει κανείς να παρατηρήσει στις μεταλλάξεις του ΠΓΕ, είναι δύο κυρίως σημεία: (i) εάν ένα μη τερματικό σύμβολο μεταλλαχθεί σε ένα τερματικό ή αντιστρόφως, ή ακόμα ένα μη τερματικό σύμβολο μεταλλαχθεί σε μη τερματικό σύμβολο διαφορετικού βαθμού, τότε το ΔΕ του γονιδίου μπορεί να μεταβληθεί δραματικά (Σχήμα 4.3), και (ii) μία μετάλλαξη μπορεί να μην προκαλέσει καμία μεταβολή στο ΔΕ, εάν συμβεί στην αμετάφραστη περιοχή του γονιδίου.

Μετατόπιση και ακολουθίες εισαγωγής

Η μετατόπιση (transposition) πρόκειται για ένα νέο γενετικό τελεστή, που εισάγει ο ΠΓΕ. Κατά την μετατόπιση επιλέγονται τμήματα γονιδίων μεταβλητού μεγέθους και μετατοπίζονται σε διαφορετικό σημείο του χρωμοσώματος. Οι ακολουθίες αυτές μπορούν να χωριστούν σε δύο κατηγορίες: (i) στις *ακολουθίες εισαγωγής* ή *AE* (*insertion sequences–IS*), και στις (ii) *ακολουθίες εισαγωγής ρίζας* ή *AEP* (*root insertion sequences–RIS*). Οι ΑΕ είναι ακολουθίες συμβόλων, που το πρώτο σύμβολό τους είναι είτε τερματικό είτε μη τερματικό. Επιπλέον οι ακολουθίες αυτές μπορούν να μετατοπιστούν και να μεταπηδήσουν στην κεφαλή ενός γονιδίου, ποτέ όμως δεν μπορούν να μετατοπιστούν στην αρχή της κεφαλής του (ρίζα του ΔΕ). Οι ΑΕΡ, αντιθέτως, πρέπει ως πρώτο σύμβολό τους, να έχουν ένα μη τερματικό σύμβολο, ενώ μετατοπίζονται πάντα στην αρχή (ρίζα) των γονιδίων. Ορίζονται τρία είδη μετατοπίσεων, τα οποία και αναλύονται στην συνέχεια.

Μετατόπιση ΑΕ Οποιαδήποτε ακολουθία συμβόλων στο χρωμόσωμα του ΠΓΕ μπορεί να γίνει ΑΕ. Η διαδικασία της μετατόπισης ΑΕ συμβαίνει με τον ακόλουθο τρόπο: αρχικά επιλέγεται τυχαία το χρωμόσωμα, στο οποίο θα συμβεί η μετατόπιση ΑΕ, βάσει ενός *ρυθμού μετατόπισης ΑΕ* (ρ_{is}). Στην συνέχεια επιλέγονται πάλι τυχαία η αρχή και το τέλος της ακολουθίας ΑΕ μέσα στο χρωμόσωμα, καθώς και το μήκος της. Τα μήκη των ακολουθιών ΑΕ δεν είναι εντελώς αυθαίρετα, αλλά κατά την έναρξη του αλγορίθμου έχει οριστεί ένα σύνολο μηκών ΑΕ· από αυτό το σύνολο επιλέγεται τυχαία το μήκος της ακολουθίας ΑΕ. Τελικά, τοποθετείται ένα

αντίγραφο της ΑΕ στην θέση εισαγωγής. Έστω ότι επιλέχθηκε για μετατόπιση ΑΕ το εξής χρωμόσωμα:

```
012345678901234560123456789012345 6
-ab+Q-baabaabaabQ***+-/aababbaaaa
```

Έστω ότι η ακολουθία `a+Q` του γονιδίου 1 επιλέγεται να γίνει ΑΕ, και έστω ότι εισάγεται στην θέση 3 του γονιδίου 2. Τότε το προκύπτον χρωμόσωμα θα είναι:

```
012345678901234560123456789012345 6
-ab+Q-baabaabaabQ*+a+Q*+ababbaaaa
```

Η εισαγωγή της ΑΕ αναγκάζει τα σύμβολα της κεφαλής, που βρίσκονται δεξιά του σημείου εισαγωγής, να ολισθήσουν τόσες θέσεις, όσο είναι το μήκος της ακολουθίας. Τα σύμβολα της ουράς *δεν* ολισθαίνουν, οπότε η ολίσθηση της κεφαλής προκαλεί την απώλεια των δεξιοτέρων συμβόλων (στο παράδειγμα εξαφανίστηκε η ακολουθία `-/a`). Με αυτό τον τρόπο εξασφαλίζεται, ότι το τροποποιημένο γονίδιο θα εξακολουθεί να δίνει συντακτικά ορθά ΔΕ.

Μετατόπιση ΑΕΡ Η μετατόπιση ΑΕΡ είναι αντίστοιχη της μετατόπισης ΑΕ, με την διαφορά ότι εδώ μετατοπίζονται ΑΕΡ και όχι απλές ΑΕ. Επειδή μία ΑΕΡ πρέπει να αρχίζει με μη τερματικό σύμβολο, η αρχή της επιλέγεται πάντα από την κεφαλή ενός γονιδίου. Η διαδικασία της επιλογής γίνεται ως εξής: αρχικά επιλέγεται τυχαία ένα σύμβολο της κεφαλής του γονιδίου. Εάν είναι συναρτησιακό, τότε το σύμβολο αυτό αποτελεί και την αρχή της ΑΕΡ. Σε αντίθετη περίπτωση σαρώνονται τα σύμβολα αριστερά από το αρχικό σύμβολο, μέχρις ότου βρεθεί ένα συναρτησιακό σύμβολο, το οποίο και θα αποτελέσει την αρχή της ΑΕΡ. Το μήκος της ΑΕΡ επιλέγεται και στην περίπτωση αυτή τυχαία από ένα σύνολο μηκών, όπως συμβαίνει και με τις κοινές ΑΕ. Έστω για παράδειγμα το παρακάτω χρωμόσωμα δύο γονιδίων:

```
012345678901234560123456789012345 6
*-bQ/++/babbabbba//Q*baa+bbbabbbbb
```

Έστω ότι η ακολουθία `Q/+` του γονιδίου 1 επιλέχθηκε τυχαία να γίνει ΑΕΡ, που θα εισαχθεί στην ρίζα του γονιδίου 1. Τότε, το χρωμόσωμα θα γίνει:

```
012345678901234560123456789012345 6
Q/+*-bQ/babbabbba//Q*baa+bbbabbbbb
```

Και στην περίπτωση αυτή τα σύμβολα της κεφαλής ολισθαίνουν δεξιά κατά το μήκος της ΑΕΡ, με αποτέλεσμα τα δεξιότερα σύμβολα της κεφαλής να χάνονται. Και εδώ τα προκύπτοντα ΔΕ είναι συντακτικά ορθά.

Μετατόπιση γονιδίου Η μετατόπιση γονιδίου εφαρμόζεται σε χρωμοσώματα με περισσότερα του ενός γονίδια (βλ. §4.2.1). Κατά την μετατόπιση γονιδίου ένα ολόκληρο γονίδιο μετατοπίζεται στην αρχή του χρωμοσώματος. Σε αντίθεση με τις άλλες μορφές μετατόπισης, το αρχικό γονίδιο στην περίπτωση αυτή χάνεται και τα γονίδια που βρίσκονται πριν από αυτό στο χρωμόσωμα, ολισθαίνουν κατά μία θέση γονιδίου δεξιά. Έστω για παράδειγμα το εξής χρωμόσωμα με τρία γονίδια:

```
012345678901201234567890120123456789012
/+Qa*bbaaabaa*a*/Qbbbbbabb/Q-aabbaaabbb
```

Έστω ότι επιλέχθηκε το γονίδιο 3 για να μετατοπιστεί. Τότε το νέο χρωμόσωμα θα είναι:

```
012345678901201234567890120123456789012
/Q-aabbaaabbb/+Qa*bbaaabaa*a*/Qbbbbbabb
```

Η μετατόπιση γονιδίων το μόνο που επιτυγχάνει σε επίπεδο δομής χρωμοσώματος, είναι η αναδιάταξη των γονιδίων. Μάλιστα σε περιπτώσεις που η συνάρτηση σύνδεσης των γονιδίων είναι αντιμεταθετική, δεν αλλάζει ούτε η τιμή της προσαρμογής του χρωμοσώματος. Παρ΄ όλα αυτά ο τελεστής αυτός σε συνδυασμό με τους τελεστές της ανασύνθεσης, οι οποίοι συζητώνται στην συνέχεια, μπορεί να δώσει καλά αποτελέσματα. Επιπλέον, εάν η συνάρτηση σύνδεσης δεν είναι αντιμεταθετική, όπως η συνάρτηση IF, η μετατόπιση των γονιδίων μπορεί να δημιουργήσει χρωμοσώματα με εντελώς διαφορετική προσαρμογή.

Ανασύνθεση

Ο ΠΓΕ υποστηρίζει τρία είδη ανασύνθεσης: *ανασύνθεση ενός σημείου (one-point recombination), ανασύνθεση δύο σημείων (two-point recombination)* και *ανασύνθεση γονιδίων (gene recombination)*. Σε κάθε είδος ανασύνθεσης επιλέγονται τυχαία δύο χρωμοσώματα, τα οποία ανταλλάσσουν μεταξύ τους γενετικό υλικό, και παράγουν δύο νέους απογόνους. Οι απόγονοι είναι συνήθως τόσο διαφορετικοί μεταξύ τους, όσο διαφορετικοί είναι και με τους γονείς τους.

Ανασύνθεση ενός σημείου Κατά την ανασύνθεση ενός σημείου επιλέγεται τυχαία ένα κοινό σημείο στα δύο χρωμοσώματα γονείς και ανταλλάσσεται μεταξύ τους το γενετικό υλικό στα δεξιά του σημείου. Για παράδειγμα έστω ότι δύο γονείς είναι:

```
01234567890123456012345678901234567
+*-b-Qa*aabbbbaaa-Q-//b/*aabbabbab
++//b//-bbbbbbbbb-*-ab/b+bbbaabbaa
```

Έστω επίσης ότι ως σημείο διαχωρισμού των δύο χρωμοσωμάτων επιλέγεται η θέση 6. Τότε τα δύο χρωμοσώματα διαχωρίζονται στο σημείο αυτό και ανταλλάσσουν την ακολουθία συμβόλων δεξιά της θέσης 6. Επομένως, οι απόγονοί τους θα έχουν την μορφή:

```
01234567890123456012345678901234567
+*-b-Q/-bbbbbbbbb-*-ab/b+bbbaabbaa
++//b/a*aabbbbaaa-Q-//b/*aabbabbab
```

Η ανασύνθεση ενός σημείου είναι ο δεύτερος σημαντικότερος τελεστής του ΠΓΕ ύστερα από τον τελεστή της μετάλλαξης. Γενικά μπορεί να προσδώσει σημαντική διαφορετικότητα στον πληθυσμό, καθ΄ ότι συνήθως οι απόγονοι που προκύπτουν, έχουν αρκετά διαφοροποιημένα χαρακτηριστικά σε σχέση με τους γονείς τους.

Ανασύνθεση δύο σημείων Η ανασύνθεση δύο σημείων είναι εντελώς αντίστοιχη της ανασύνθεσης ενός σημείου, μόνο που σε αυτή την περίπτωση επιλέγονται τυχαία δύο σημεία, στα οποία θα διαχωριστούν τα χρωμοσώματα. Στην συνέχεια τα δύο χρωμοσώματα ανταλλάσσουν το γενετικό τους υλικό, που βρίσκεται μεταξύ αυτών των δύο σημείων. Έστω επομένως οι γονείς:

```
01234567890123456012345678901234567
*-+Q/Q*Qaaabbbbab QQab*++-aabbabaab
Q/-b-+/abaabbbaab/*-aQa*babbabbabb
```

Εάν ως σημεία διαχωρισμού επιλεγούν η θέση 4 του γονιδίου 1 και η θέση 7 του γονιδίου 2, τότε οι απόγονοι των δύο γονέων θα είναι:

```
01234567890123456012345678901234 56
*−+Q/+/abaabbbaab/*−aQa*−aabbabaab
Q/−b−Q*Qaaabbbbab QQab*++babbabbabb
```

Αυτό που αξίζει να παρατηρήσει κανείς είναι, ότι εάν τα σημεία διαχωρισμού επιλεγούν έτσι, ώστε να βρίσκονται εντός της αμετάφραστης περιοχής ενός γονιδίου, τότε οι απόγονοι της ανασύνθεσης δύο σημείων έχουν τα ίδια ΑΠΑ με τους γονείς τους.

Ο τελεστής της ανασύνθεσης δύο σημείων αποτελεί και αυτός έναν από τους σημαντικούς τελεστές του ΠΓΕ, καθ΄ ότι μπορεί να προσδώσει την απαραίτητη διαφορετικότητα στον πληθυσμό. Μάλιστα, ο συνδυασμός των τελεστών της ανασύνθεσης ενός ή δύο σημείων και της μετάλλαξης μπορεί να επιλύσει δυνητικά οποιοδήποτε πρόβλημα.

Ανασύνθεση γονιδίου Κατά την ανασύνθεση γονιδίου ανταλλάσσονται μεταξύ των χρωμοσωμάτων ολόκληρα γονίδια. Τα γονίδια που θα ανταλλαχθούν επιλέγονται τυχαία. Για παράδειγμα έστω τα χρωμοσώματα των γονέων:

```
012345678901201234567890120123456789012
/+/ab−aabbbbb−aa***aaabaaa−+−−babbbbaab
+baQaaaabaaba*−+a−aabbabbb/ab/+bbbabaaa
```

Έστω ότι επιλέγεται να ανταλλαχθεί το γονίδιο 2, τότε οι απόγονοι θα είναι:

```
012345678901201234567890120123456789012
/+/ab−aabbbbb*−+a−aabbabbb−+−−babbbbaab
+baQaaaabaaba−aa***aaabaaa/ab/+bbbabaaa
```

Η κύρια λειτουργία του τελεστή αυτού είναι να αναδιατάσσει τα γονίδια των χρωμοσωμάτων μέσα στον πληθυσμό. Αν και τα γονίδια που ανταλλάσσονται, είναι συνήθως αρκετά διαφορετικά μεταξύ τους, δημιουργώντας αντιστοίχως διαφορετικά χρωμοσώματα, εντούτοις ο τελεστής της ανασύνθεσης γονιδίων δεν μπορεί να κατασκευάσει νέα γονίδια. Για τον λόγο αυτό δεν μπορεί να προσδώσει σημαντική διαφορετικότητα στον πληθυσμό, όπως συμβαίνει με τις ανασυνθέσεις ενός ή δύο σημείων.

ΚΕΦΑΛΑΙΟ 5

Μία νέα μέθοδος εξόρυξης γνώσης από σύνολα δεδομένων βασισμένη σε ΤΑΣ και στον ΠΓΕ

Έχοντας παρουσιάσει αναλυτικά στα προηγούμενα κεφάλαια τα ΤΑΣ και τον ΠΓΕ, στο παρόν Κεφάλαιο προτείνεται μία νέα μέθοδος για την εξόρυξη γνώσης από σύνολα δεδομένων, βασιζόμενη σε ένα συνδυασμό των δύο τεχνικών. Σκοπός του συνδυασμού αυτού είναι να συνδυάσει το πλεονέκτημα της ταχείας σύγκλισης του αλγορίθμου που παρουσιάστηκε στο Κεφάλαιο 3, με την ευελιξία και την εκφραστικότητα που προσφέρει η γλώσσα Karva, η οποία χρησιμοποιείται για την αναπαράσταση των χρωμοσωμάτων του ΠΓΕ. Τα πρώτα αποτελέσματα αυτής της μεθόδου, τα οποία παρουσιάζονται στην παρούσα εργασία, είναι ιδιαίτερα ενθαρρυντικά σε σχέση με τις κλασσικές τεχνικές εξόρυξης γνώσης και καταδεικνύουν αφενός τις εξελικτικές δυνατότητες των ΤΑΣ και αφετέρου την εκφραστική δύναμη του ΠΓΕ.

Στο Κεφάλαιο αυτό γίνεται αρχικά μία σύντομη εισαγωγή στις βασικές αρχές που διέπουν την εξόρυξη γνώσης από σύνολα δεδομένων (βλ. §5.1). Στην συνέχεια παρουσιάζονται οι προσθήκες και οι αλλαγές που έγιναν στον αλγόριθμο επιλογής κλώνων και στο μοντέλο του ΠΓΕ, έτσι ώστε να μπορέσουν να συνεργασθούν αρμονικά (βλ. §5.2 και §5.3). Έπειτα, στην §5.4 παρουσιάζεται ο τρόπος, με τον οποίο, η μέθοδος που προτείνεται, επιτυγχάνει την ταξινόμηση των δεδομένων. Τέλος, στην §5.5 παρουσιάζονται τα αποτελέσματα της προτεινόμενης μεθόδου κατά την εφαρμογή της σε συγκεκριμένα προβλήματα αξιολόγησης.

5.1 Εξόρυξη γνώσης από σύνολα δεδομένων

Η *εξόρυξη γνώσης από σύνολα δεδομένων (data mining)* ή απλά *εξόρυξη από δεδομένα (ΕΔ)* αφορά στην διαδικασία της αυτόματης ανάλυσης και εξαγωγής γνώσης από τα δεδομένα μίας βάσης δεδομένων (ΒΔ), χρησιμοποιώντας ένα ή περισσοτέρους αλγορίθμους μάθησης μηχανών. Σκοπός της εξόρυξης γνώσης είναι η ανακάλυψη τάσεων και σχημάτων στα δεδομένα. Η γνώση που αποκομίζεται από μία

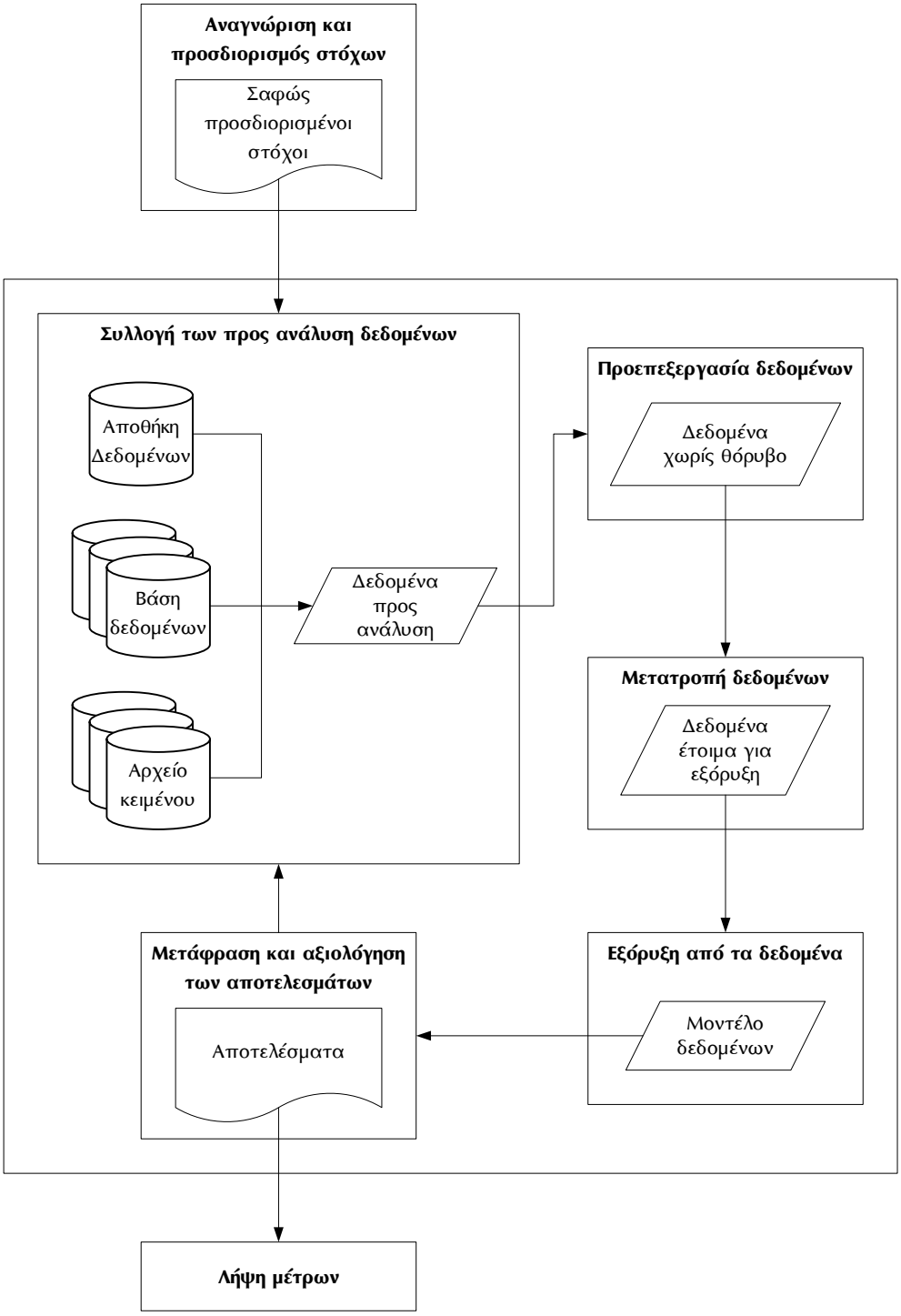
σύνοδο εξόρυξης από δεδομένα (data mining session), παρουσιάζεται συνήθως ως ένα μοντέλο ή ως μία γενίκευση των δεδομένων. Αν και οι τεχνικές μάθησης που χρησιμοποιούνται στην ΕΔ είναι ποικίλες, όλες βασίζονται στην επαγωγική μάθηση. Η *επαγωγική μάθηση (induction-based learning)* είναι η διαδικασία μάθησης, κατά την οποία σχηματίζονται γενικοί ορισμοί που περιγράφουν την προς μάθηση ιδέα, παρατηρώντας συγκεκριμένα στιγμιότυπα και εκφράσεις της ιδέας αυτής. Μία τέτοιου είδους μάθηση είναι αναμενόμενη στην ΕΔ, εφόσον το ζητούμενο είναι η περιγραφή γενικών κανόνων, που περιγράφουν, τα εξ΄ ορισμού εξειδικευμένα, δεδομένα μίας ΒΔ.

Ο τρόπος μάθησης ενός συστήματος ΕΔ μπορεί να είναι είτε επιβλεπόμενος είτε μη επιβλεπόμενος. Κατά την *επιβλεπόμενη μάθηση (supervised learning)* παρουσιάζονται στο σύστημα τα προς ανάλυση δεδομένα και εκείνο προσπαθεί με κάποιον αλγόριθμο επιβλεπομένης μάθησης, να κατασκευάσει ένα μοντέλο, το οποίο να μπορεί να ταξινομεί σωστά τα νέα δεδομένα που παρουσιάζονται στο σύστημα. Συνήθως, κατά την επιβλεπομένη μάθηση τα χαρακτηριστικά (attributes) των δεδομένων χωρίζονται σε δύο κατηγορίες: τα *χαρακτηριστικά εισόδου (input attributes)* και τα *χαρακτηριστικά εξόδου (output attributes).* Τα πρώτα αποτελούν την είσοδο του αλγορίθμου επιβλεπομένης μάθησης, ενώ τα δεύτερα αποτελούν τους στόχους που πρέπει να επιτευχθούν κατά την διαδικασία της μάθησης. Βάσει των χαρακτηριστικών εξόδου, τα προς ανάλυση δεδομένα (στιγμιότυπα) μπορούν να χωριστούν σε κατηγορίες, στις οποίες τα χαρακτηριστικά εξόδου θα είναι κοινά. Τα στιγμιότυπα που ανήκουν σε μία συγκεκριμένη κατηγορία ονομάζονται *θετικά παραδείγματα (positive examples)* της κατηγορίας αυτής. Αντιστοίχως, τα στιγμιότυπα που δεν ανήκουν στην κατηγορία αυτή ονομάζονται *αρνητικά παραδείγματα (negative examples).* Κατά την διαδικασία εκπαίδευσης στην επιβλεπόμενη μάθηση, παρουσιάζονται στο σύστημα όλα ή ένα μέρος των στιγμιοτύπων των δεδομένων, και το σύστημα θα πρέπει, ιδανικά, κατά το πέρας της εκπαίδευσης, να ταξινομεί σωστά αφενός όλα τα εκπαιδευτικά δεδομένα, και αφετέρου όλα ή σχεδόν όλα τα δεδομένα που μπορεί να του παρουσιαστούν στο μέλλον. Στην πραγματικότητα μία τέτοια απαίτηση είναι ανέφικτη, και επομένως θυσιάζεται συνήθως λίγη ακρίβεια κατά την ταξινόμηση των στιγμιοτύπων των δεδομένων εκπαίδευσης, έτσι ώστε να βελτιωθεί η ικανότητα γενίκευσης του συστήματος (αποφυγή *υπερβολικής προσαρμογής* στα δεδομένα–*overfitting*).

Κατά την *μη επιβλεπομένη μάθηση (unsupervised learning)* τα χαρακτηριστικά των προς ανάλυση δεδομένων δεν χωρίζονται σε χαρακτηριστικά εισόδου και εξόδου, αλλά αντιμετωπίζονται σαν ένα όλον. Αυτό έχει σαν αποτέλεσμα, τα δεδομένα να μην χωρίζονται εκ προοιμίου σε *ομάδες (clusters)* ή *κατηγορίες (classes).* Το σύστημα εξόρυξης δεδομένων, βάσει ενός αλγορίθμου, προσπαθεί να ανακαλύψει κοινές αρχές και ομοιότητες μεταξύ των δεδομένων, χωρίζοντάς τα εκείνο σε ομάδες με όμοια χαρακτηριστικά. Η μετάφραση του νοήματος και η ανακάλυψη της πραγματικής υπόστασης των σχηματιζομένων ομάδων έγκειται στα άτομα που θα επεξεργαστούν και θα αναλύσουν τα αποτελέσματα της συνόδου ΕΔ.

Συνήθως, ο όρος *εξόρυξη από δεδομένα* χρησιμοποιείται παράλληλα με τον όρο *Ανακάλυψη Γνώσης σε ΒΔ (Knowledge Discovery in Databases)* ή *ΑΓΒΔ (KDD).* Αν και οι δύο όροι χρησιμοποιούνται συνήθως ισοδύναμα, η διαδικασία ΑΓΒΔ είναι γενικά ευρύτερη της ΕΔ, αποτελουμένη από ένα σύνολο επιμέρους διαδικασιών (Σχήμα 5.1), μία εκ των οποίων είναι η ΕΔ. Θα μπορούσε κανείς να χαρακτηρίσει την διαδικασία ΑΓΒΔ ως μία διαδικασία επτά βημάτων (Roiger και Geatz, 2003):

Βήμα 1. [Αναγνώριση και προσδιορισμός στόχου] Το βήμα αυτό είναι ένα από τα πιο σημαντικά και παράλληλα ένα από τα πιο δύσκολα της διαδικασίας ΑΓΒΔ. Απαιτείται η κατανόηση του πεδίου επί του οποίου θα



Σχήμα 5.1: Το μοντέλο της διαδικασίας Ανακάλυψης Γνώσης σε Βάσεις Δεδομένων.

εφαρμοστεί η διαδικασία της ανακάλυψης γνώσης και ο σαφής καθορισμός των στόχων της διαδικασίας. Επιπροσθέτως, στο βήμα αυτό θα πρέπει να καθοριστούν τα κριτήρια επιτυχίας/αποτυχίας της μεδόδου, οι μέθοδοι εξόρυξης γνώσης που θα χρησιμοποιηθούν, ο προϋπολογισμός και ο προγραμματισμός του έργου, κ.α.

Βήμα 2. [Συλλογή των προς ανάλυση δεδομένων] Έχοντας προσδιορίσει τους στόχους της διαδικασίας ΑΓΒΔ, συλλέγονται τα δεδομένα, τα οποία θα επεξεργαστούν και θα αναλυθούν στην συνέχεια. Τα δεδομένα αυτά μπορεί να προέρχονται από ποικίλες πηγές, όπως είναι αποθήκες δεδομένων (data warehouses), κοινές σχεσιακές ΒΔ ή ακόμα και από απλά αρχεία κειμένου.

Βήμα 3. [Προεπεξεργασία των δεδομένων] Το στάδιο αυτό ασχολείται με τον χειρισμό του θορύβου και των πιθανών ασυνεπειών στα δεδομένα. Λόγω του μεγάλου όγκου πληροφορίας που συλλέγεται στο προηγούμενο βήμα, είναι πολύ πιθανό να υπάρχουν διπλές εγγραφές, μη έγκυρες τιμές κάποιων χαρακτηριστικών ή ακόμα και ελλιπή δεδομένα. Στο στάδιο αυτό θα πρέπει να διαχειρίζονται αυτές οι περιπτώσεις, είτε απορρίπτοντας τις λανθασμένες ή ελλιπείς εγγραφές, είτε συμπληρώνοντάς τες με έναν ευρετικό τρόπο. Τέλος, στο στάδιο αυτό λαμβάνει χώρα και η διαδικασία της *εξομάλυνσης των δεδομένων (data smoothing)*, κατά την οποία επιδιώκεται η μείωση των τιμών ενός αριθμητικού χαρακτηριστικού, ή η εξάλειψη εγγραφών, των οποίων οι τιμές κάποιων χαρακτηριστικών κείνται εκτός των τυπικών ορίων διακύμανσής τους (*outliers*). Γενικά, το στάδιο της προεπεξεργασίας των δεδομένων είναι αρκετά σημαντικό και μπορεί να χαρακτηριστεί καθοριστικό για την τελική έκβαση ολόκληρης της διαδικασίας ΑΓΒΔ.

Βήμα 4. [Μετατροπή των δεδομένων] Τα προς ανάλυση δεδομένα ακόμα και ύστερα από το στάδιο της προεπεξεργασίας δεν είναι συνήθως έτοιμα για εξόρυξη. Αυτό συμβαίνει διότι οι περισσότεροι αλγόριθμοι εξόρυξης από τα δεδομένα έχουν συγκεκριμένες απαιτήσεις από τα δεδομένα, έτσι ώστε να μπορέσουν να δουλέψουν σωστά. Τέτοιες απαιτήσεις είναι η κανονικοποίηση των δεδομένων, η μετατροπή των αριθμητικών χαρακτηριστικών σε κατηγορηματικά ή αντιστρόφως, η απομάκρυνση χαρακτηριστικών που δεν παίζουν σημαντικό ρόλο στην ταξινόμηση των δεδομένων, η δημιουργία νέων σημαντικών χαρακτηριστικών ως συνδυασμών λιγότερο σημαντικών χαρακτηριστικών και τέλος ο διαχωρισμός του συνόλου δεδομένου στο *σύνολο εκπαίδευσης (training set)* και στο *σύνολο δοκιμής (test set)*.

Βήμα 5. [Εξόρυξη από τα δεδομένα] Στο στάδιο αυτό τα προς ανάλυση δεδομένα έχουν διαμορφωθεί κατάλληλα και είναι έτοιμα να τροφοδοτήσουν τον αλγόριθμο ΕΔ. Το αποτέλεσμα αυτού του σταδίου είναι ένα γενικευμένο μοντέλο που περιγράφει τα δεδομένα.

Βήμα 6. [Μετάφραση και αξιολόγηση των αποτελεσμάτων] Κατά την φάση αυτή αξιολογούνται τα αποτελέσματα της συνόδου ΕΔ και αποφασίζεται εάν θα επαναληφθεί η ίδια διαδικασία από το Βήμα 2, με σκοπό να ληφθούν ακόμη καλύτερα αποτελέσματα. Επιπλέον εάν το μοντέλο που παρήγαγε η σύνοδος ΕΔ είναι ικανοποιητικό, μεταφράζονται οι κανόνες και τα χαρακτηριστικά του μοντέλου σε γλώσσα αντιληπτή από τους απλούς χρήστες.

Βήμα 7. [Λήψη μέτρων] Το στάδιο αυτό αποτελεί το τελικό στάδιο μίας διαδικασίας ΑΓΒΔ, κατά το οποίο η γνώση που αποκομίστηκε από την διαδικασίας της εξόρυξης εφαρμόζεται στην πράξη. ∎

Κλείνοντας αυτή την σύντομη εισαγωγή στην ΕΔ, αξίζει να αναφερθεί ότι η τεχνική της ΕΔ δεν είναι κατάλληλη για όλα τα προβλήματα εξόρυξης γνώσης. Πράγματι, εάν η γνώση που ζητείται είναι μία απλή αναφορά σχετική με τις πληροφορίες που υπάρχουν σε μία ΒΔ (*ρηχή γνώση–swallow knowledge*), τότε γλώσσες επερωτήσεων, όπως είναι η SQL, ή εργαλεία επεξεργασίας πολυδιαστάτων δεδομένων, όπως είναι τα εργαλεία OLAP (On-Line Analytical Process) είναι ικανά να δώσουν ικανοποιητικά αποτελέσματα. Ακόμη, σε περιπτώσεις που τα διαθέσιμα δεδομένα δεν είναι ποιοτικά, ένα έμπειρο σύστημα ίσως δώσει καλύτερα αποτελέσματα.

5.2 Προσθήκες στον αλγόριθμο επιλογής κλώνων

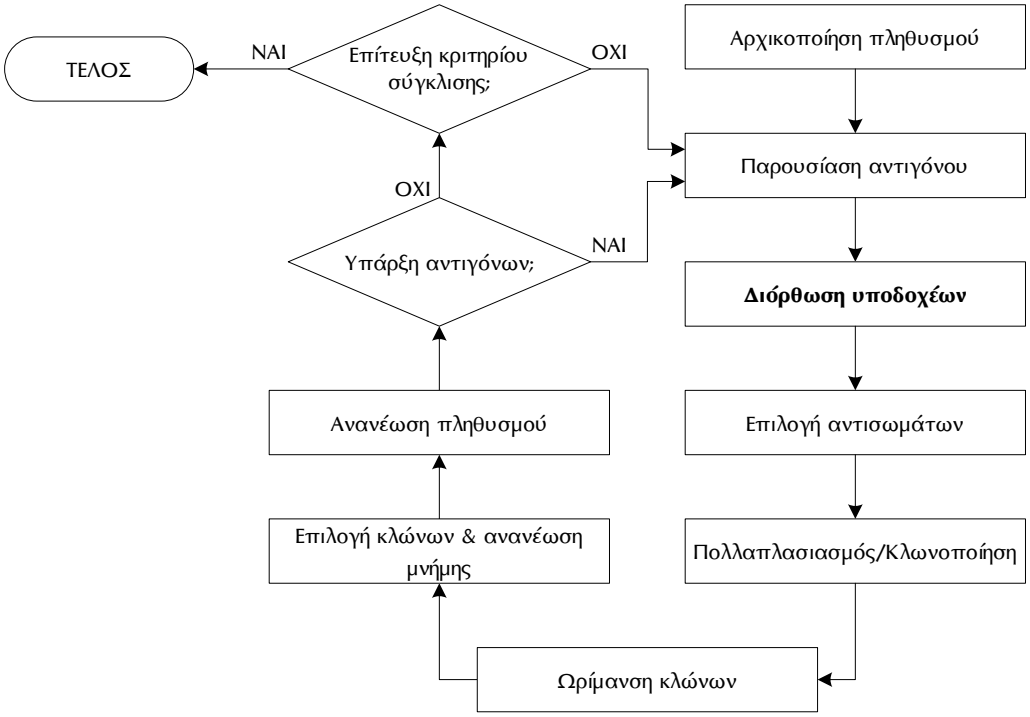
Όπως έγινε σαφές από την συζήτηση που προηγήθηκε, η διαδικασία της εξόρυξης γνώσης είναι μία επίπονη και απαιτητική διαδικασία μάθησης. Για τον λόγο αυτό κρίθηκε σκόπιμη η τροποποίηση του αλγορίθμου επιλογής των κλώνων που παρουσιάστηκε στο Κεφάλαιο 3 (§3.4), έτσι ώστε να επιτρέπει μεγαλύτερη διαφορετικότητα στον πληθυσμό των αντισωμάτων. Η μηχανισμός της υπερ-μετάλλαξης που χρησιμοποιείται, εκτελεί συνήθως τοπική αναζήτηση στο πεδίο των λύσεων, ενώ η ανανέωση του πληθυσμού που συμβαίνει στο προτελευταίο βήμα (Βήμα ΕΚ8) φροντίζει για την διατήρηση της διαφορετικότητας του πληθυσμού. Αν και σε αυτή την μορφή ο αλγόριθμος δίνει πολύ καλά αποτελέσματα στην περίπτωση της αναγνώρισης ψηφιακών χαρακτήρων, στην περίπτωση της ΕΔ είναι υποδεέστερος, λόγω της αυξημένης πολυπλοκότητας του προβλήματος. Για τον λόγο αυτό εισήχθη ακόμη ένας μηχανισμός εισαγωγής διαφορετικότητας στον πληθυσμό, έτσι ώστε να εξερευνάται καλύτερα το πεδίο λύσεων του προβλήματος. Ο μηχανισμός αυτός πρόκειται για την διόρθωση των υποδοχέων, που παρουσιάστηκε στο Κεφάλαιο 3 (§3.1.2). Κατά την διαδικασία αυτή τα κύτταρα που επέδειξαν χαμηλή ποιότητα υποδοχέων ή που ανέπτυξαν εχθρικούς για τον ίδιο οργανισμό υποδοχείς, καταστρέφουν τους υποδοχείς αυτούς και δημιουργούν εντελώς νέους μέσω της ανασύνθεσης V(D)J (βλ. §2.3).

Η διόρθωση των υποδοχέων εισάγεται ως ένα νέο βήμα στον αλγόριθμο επιλογής κλώνων μεταξύ των βημάτων ΕΚ2 και ΕΚ3 (Σχήμα 5.2), δηλαδή αφότου παρουσιαστούν τα αντιγόνα στον πληθυσμό και προτού γίνει η επιλογή των καλυτέρων αντισωμάτων. Κατά την διαδικασία της διόρθωσης των υποδοχέων επιλέγονται τα n_e χειρότερα αντισώματα, τα οποία θα αντικατασταθούν από εντελώς νέα αντισώματα που θα προκύψουν από ανασύνθεση V(D)J. Κατά την ανασύνθεση αυτή επιλέγονται τα n_p καλύτερα αντισώματα, τα οποία θα αποτελέσουν την δεξαμενή των αντισωμάτων από τα οποία θα κατασκευαστούν τα νέα αντισώματα. Η ανασύνθεση V(D)J ολοκληρώνεται σε 5 βήματα (Σχήμα 5.3). Στον παρακάτω αλγόριθμο l_c είναι το τρέχον μήκος του αντισώματος που κατασκευάζεται, l_g είναι το μήκος του τμήματος γονιδίου που επιλέγεται κάθε φορά και L είναι το μήκος των αντισωμάτων του αλγορίθμου.

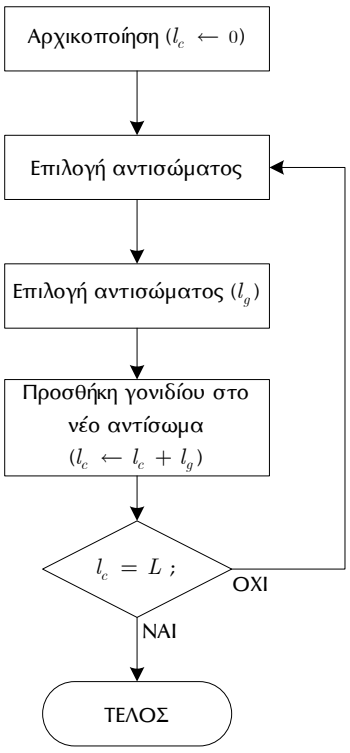
Αλγόριθμος VDJ (Ανασύνθεση V(D)J). Ο αλγόριθμος υπαγορεύει τον τρόπο δημιουργίας νέων αντισωμάτων μέσω της ανασύνθεσης V(D)J.

VDJ1. [Αρχικοποίηση] Αρχικά το μήκος του νέου αντισώματος είναι μηδεν·

$$l_c \leftarrow 0.$$



Σχήμα 5.2: Ο τροποποιημένος αλγόριθμος επιλογής κλώνων. Το νέο βήμα (διόρθωση υποδοχέων) σημειώνεται με έντονη γραμματοσειρά.



Σχήμα 5.3: Υλοποίηση της ανασύνθεσης V(D)J.

VDJ2. [Επιλογή αντισώματος] Τυχαία επιλογή ενός από τα n_p αντισώματα της δεξαμενής των αντισωμάτων.

VDJ3. [Επιλογή γονιδίου] Επιλογή ενός τμήματος τυχαίου μήκους l_g από το αντίσωμα που επιλέχθηκε στο προηγούμενο βήμα. Το μήκος του τμήματος αυτού θα πρέπει να είναι μικρότερο ή ίσο από το υπολειπόμενο μήκος του αντισώματος που κατασκευάζεται, δηλαδή

$$l_g \leq L - l_c.$$

VDJ4. [Προσθήκη γονιδίου στο νέο αντίσωμα] Το γονιδίο που επιλέχθηκε τοποθετείται στο τέλος του νέου αντισώματος, οπότε το μήκος του γίνεται

$$l_c \leftarrow l_c + l_g.$$

VDJ5. [Συνθήκη τέλους] Εάν $l_c = L$, τότε η διαδικασία της ανασύνθεσης τελειώνει, ειδ΄ άλλως επαναλαμβάνεται από το Βήμα VDJ2. ∎

Η διαδικασία που μόλις περιγράφηκε, επαναλαμβάνεται για κάθε ένα από τα n_e αντισώματα που πρόκειται να αντικατασταθούν. Στην παραπάνω περιγραφή με τον όρο γονίδιο γίνεται αναφορά σε ένα τμήμα αυθαιρέτου μήκους του αντισώματος, επομένως δεν θα πρέπει να γίνεται σύγχυση με τα γονίδια του ΠΓΕ· στην περίπτωση που τα αντισώματα έχουν την μορφή των χρωμοσωμάτων του ΠΓΕ, τότε ο όρος γονίδιο, όπως χρησιμοποιείται σε αυτή την περίπτωση, υπονοεί ένα τμήμα αυθαιρέτου μήκους ολοκλήρου του χρωμοσώματος.

5.3 Προσθήκες και αλλαγές στο μοντέλο του ΠΓΕ

Η προσθήκη της διόρθωσης των υποδοχέων και της ανασύνθεσης V(D)J απαιτεί τροποποιήσεις και στο μοντέλο του ΠΓΕ, εφόσον τα αντισώματα αναπαρασταθούν ως χρωμοσώματα τέτοιου τύπου. Για τον λόγο αυτό εισάγεται η έννοια της *ανασύνθεσης πολλών σημείων (multipoint recombination)*, η οποία είναι αντίστοιχη της ανασύνθεσης ενός ή δύο σημείων που παρουσιάστηκε στο Κεφάλαιο 4 (§4.2.2). Η ιδιαιτερότητα του τελεστή αυτού είναι, ότι το σύνολο των σημείων στα οποία χωρίζονται τα χρωμοσώματα γονείς, δεν είναι εκ προοιμίου σταθερό, αλλά μπορεί να κυμαίνεται σε κάθε εφαρμογή του τελεστή. Επιπλέον, κατά την ανασύνθεση αυτή μπορούν να συμμετέχουν περισσότερα των δύο χρωμοσωμάτων (*ανασύνθεση πολλών γονέων*). Ο τρόπος με τον οποίο εφαρμόζεται, ομοιάζει αρκετά με τον αλγόριθμο της ανασύνθεσης V(D)J (βλ. Σχήμα 5.3). Αρχικά επιλέγονται n χρωμοσώματα, τα οποία θα αποτελέσουν τους γονείς των νέων χρωμοσωμάτων. Ο αριθμός των γονέων είναι σταθερός και αποτελεί παράμετρο του αλγορίθμου. Στην συνέχεια επιλέγεται τυχαία ένα σημείο διαχωρισμού για κάθε χρωμόσωμα-γονέα. Το σημείο διαχωρισμού κάθε επομένου χρωμοσώματος θα πρέπει να βρίσκεται δεξιότερα του σημείου διαχωρισμού του προηγουμένου χρωμοσώματος. Η διαδικασία επιλογής σημείων διαχωρισμού συνεχίζεται, μέχρις ότου κάποιο σημείο διαχωρισμού συμπέσει με το τέλος του χρωμοσώματος. Εάν διαχωρισθούν όλα τα χρωμοσώματα μία φορά, και το σημείο διαχωρισμού δεν έχει ακόμα συμπέσει με το τέλος του χρωμοσώματος, τότε επαναλαμβάνεται η ίδια διαδικασία από το πρώτο χρωμόσωμα, το οποίο αποκτά και άλλο σημείο διαχωρισμού. Η διαδικασία αυτή επαναλαμβάνεται προσθέτοντας συνεχώς σημεία διαχωρισμού στα χρωμοσώματα, μέχρις ότου κάποιο σημείο διαχωρισμού συμπέσει με το τέλος του χρωμοσώματος. Το αποτέλεσμα αυτής της διαδικασίας είναι ένα χρωμόσωμα, το

οποίο αποτελείται από τα τμήματα των χρωμοσωμάτων-γονέων, που βρίσκονται μεταξύ των σημείων διαχωρισμού. Η διαδικασία αυτή μπορεί να γίνει καλύτερα κατανοητή, εάν θεωρήσει κανείς το επόμενο παράδειγμα: έστω ότι το αλφάβητο των χρωμοσωμάτων είναι $\Sigma = \{Q, *, /, -, +, a, b\}$, όπου το σύνολο $T = \{a, b\}$ είναι το σύνολο των τερματικών συμβόλων. Έστω ακόμη ότι η κεφαλή των γονιδίων έχει μήκος $h = 5$, οπότε το μήκος τους από την σχέση (4.3) θα είναι $L = 5 \cdot 2 + 1 = 11$. Έστω, τέλος, ότι το κάθε χρωμόσωμα περιέχει 3 γονίδια και για την ανασύνθεση πολλών σημείων επιλέγονται $n = 3$ γονείς:

```
01234567890012345678900123456789O
Q+bb*babbaba-**--abbbaaQ*a*Qbbbaab
/-++Qbababb Q**abbabbaaQ*ab+abaaab
-+Qbabaaabb/Q**aababbab**+*Qaaabab
```

Έστω ότι, αρχικά, ως σημεία διαχωρισμού των χρωμοσωμάτων επιλέγονται η θέση 6 του πρώτου γονιδίου στο πρώτο χρωμόσωμα, η θέση 2 του δευτέρου γονιδίου στο δεύτερο χρωμόσωμα και η θέση 9 του δευτέρου γονιδίου στο τρίτο χρωμόσωμα. Επειδή έως αυτό το σημείο κανένα σημείο διαχωρισμού δεν έχει συμπέσει με το τέλος του χρωμοσώματος, η διαδικασία αυτή συνεχίζεται κυκλικά από το πρώτο χρωμόσωμα. Έστω, λοιπόν, ότι ως επόμενη θέση διαχωρισμού επιλέγεται η θέση 3 του γονιδίου 3 του πρώτου χρωμοσώματος, ενώ η επόμενη θέση διαχωρισμού είναι το τέλος του δευτέρου χρωμοσώματος, οπότε πλέον έχουν προσδιορισθεί όλα τα σημεία διαχωρισμού. Τότε, το νέο χρωμόσωμα θα αποτελείται από τα τμήματα των χρωμοσωμάτων μεταξύ των σημείων διαχωρισμού (τμήματα με έντονη γραμματοσειρά) και θα είναι το εξής:

```
01234567890012345678900123456789O
Q+bb*bababbQ**+aababaaQ*ab+abaaab
```

Η διαδικασία της ανασύνθεσης πολλών σημείων και πολλών γονέων μπορεί να προσφέρει σημαντικά οφέλη στην διατήρηση της διαφορετικότητας του πληθυσμού, ενώ παράλληλα μιμείται αρκετά καλά την πραγματική διαδικασία της ανασύνθεσης V(D)J.

Για τις ανάγκες του αλγορίθμου επιλογής των κλώνων οι μόνοι τελεστές του ΠΓΕ που είναι χρήσιμοι, είναι ο τελεστής της μετάλλαξης και ο τελεστής της ανασύνθεσης πολλών σημείων, που μόλις περιγράφηκε. Οι υπόλοιποι τελεστές (μετατόπιση, ανασύνθεση ενός ή δύο σημείων) δεν χρησιμοποιούνται.

Όσον αφορά στα σύνολα συμβόλων που χρησιμοποιούνται στον ΠΓΕ, ορίζεται επιπλέον ένα ακόμα σύνολο $C \subseteq T$, όπου T είναι το σύνολο των τερματικών συμβόλων. Το σύνολο C αποτελεί ένα σύνολο σταθερών, δηλαδή τερματικών συμβόλων με εξ΄ αρχής προσδιορισμένες και σταθερές τιμές. Το ότι το σύνολο των σταθερών είναι υποσύνολο του συνόλου των τερματικών συμβόλων, επιτρέπει την άμεση υιοθέτησή του από το μοντέλο ΠΓΕ χωρίς καμία επιπλέον προσαρμογή στους κανόνες των γενετικών τελεστών. Η μόνη διαδικασία που επηρεάζεται από αυτή την προσθήκη, σε επίπεδο αλγοριθμικό και όχι αυστηρά μαθηματικό, είναι η διαδικασία αποτίμησης των ΔΕ, η οποία θα πρέπει να γνωρίζει τις τιμές των σταθερών τερματικών συμβόλων.

Τελευταία αλλαγή στο μοντέλο του ΠΓΕ είναι η επέκτασή του, ώστε να μπορεί να χειρίζεται χρωμοσώματα, τα οποία αντί για απλά σύμβολα, περιέχουν ακολουθίες συμβόλων (strings). Η αλλαγή αυτή είναι περισσότερο προγραμματιστικής φύσεως παρά μαθηματικής ή αλγοριθμικής, και για τον λόγο αυτό αναλύεται περισσότερο στο Παράρτημα Α, όπου αναπτύσσεται η υποδομή λογισμικού JAIF.

5.4 Ταξινόμηση μέσω του αλγορίθμου επιλογής κλώνων και του ΠΓΕ

Ο αλγόριθμος επιλογής κλώνων είναι μία τεχνική επιβλεπομένης μάθησης, κατά την οποία παρουσιάζονται κάποια πρότυπα στον αλγόριθμο, και εκείνος θα πρέπει μέσω μίας εξελικτικής διαδικασίας, όπως περιγράφηκε στο Κεφάλαιο 3, να μπορεί στο μέλλον να τα αναγνωρίζει. Επομένως, πρωταρχικό μέλημα για την ταξινόμηση των δεδομένων μέσω του αλγορίθμου επιλογής κλώνων είναι ο καθορισμός των προτύπων, δηλαδή των αντιγόνων, από τα δεδομένα του προβλήματος. Σημαντικές παράμετροι είναι επίσης η μορφή της αναπαράστασης των αντισωμάτων και η αντιμετώπιση των αριθμητικών και κατηγορηματικών χαρακτηριστικών των δεδομένων. Ακόμη η συνάρτηση πρόσδεσης αντισωμάτων-αντιγόνων αποτελεί μία από τις πιο καθοριστικές παραμέτρους για την σωστή ταξινόμηση των δεδομένων. Τέλος θα πρέπει να λαμβάνονται μέτρα για την αποφυγή της υπερβολικής προσαρμογής του αλγορίθμου στα δεδομένα εισόδου, έτσι ώστε να μην δημιουργούνται πολύ εξειδικευμένες κατηγορίες δεδομένων.

5.4.1 Αναπαράσταση προτύπων

Στην περίπτωση της εξόρυξης από δεδομένα, τα πρότυπα που πρέπει να παρουσιαστούν στον αλγόριθμο, είναι πολύ πιο σύνθετα από τις απλές ακολουθίες συμβόλων που παρουσιάστηκαν στο Κεφάλαιο 3. Εδώ τα πρότυπα είναι ολόκληρα σύνολα εγγραφών μίας ΒΔ, και επομένως θα πρέπει κανείς να αναθεωρήσει τον τρόπο αναπαράστασης των αντιγόνων. Ο Ορισμός 3.2, ο οποίος ορίζει τα αντιγόνα ως ακολουθίες συμβόλων ενός αλφαβήτου, είναι αρκετά περιοριστικός. Για τον λόγο αυτό, στο πρόβλημα της ΕΔ τα αντιγόνα ορίζονται ως μία ακολουθία εγγραφών της ΒΔ. Επειδή η αναπαράσταση αυτή των αντιγόνων δεν εντάσσεται στα πλαίσια της γραμμικής ακολουθίας συμβόλων και κατ΄ επέκταση του μοντέλου του χώρου σχήματος (βλ. §3.2), θα μπορούσε να ονομαστεί *γενικευμένο αντιγόνο (generic antigen)*.

Κάθε γενικευμένο αντιγόνο θα περιέχει όλα τα σύνολα εγγραφών που ανήκουν σε μία συγκεκριμένη ομάδα ή κλάση δεδομένων, επομένως το αντιγόνο αυτό θα είναι αντιπροσωπευτικό της κλάσης αυτής (*αντιγόνο κλάσης δεδομένων ή ΑΚΔ–data class antigen ή DCA*). Εφόσον καθορισθούν τα χαρακτηριστικά εξόδου ή τα χαρακτηριστικά στόχοι της επιβλεπομένης μάθησης, χωρίζονται τα δεδομένα σε κλάσεις και ανατίθενται στα αντίστοιχα ΑΚΔ. Επομένως, σε κάθε πρόβλημα ΕΔ δημιουργούνται τόσα ΑΚΔ όσες είναι και οι κλάσεις των δεδομένων του προβλήματος.

5.4.2 Αναπαράσταση αντισωμάτων και αναγνώριση ΑΚΔ

Τα αντισώματα αναπαρίστανται ως πλήρως λειτουργικά χρωμοσώματα του ΠΓΕ, αποκτώντας όλες τις ιδιότητές τους, όπως ο διαχωρισμός γονοτύπου-φαινοτύπου, τα μεταβλητά ανοικτά πλαίσια ανάγωσης (ΑΠΑ), τα σταθερό μήκος και την πληθώρα των γονιδίων από τα οποία μπορούν να αποτελούνται. Όπως μπορεί πολύ έυκολα να παρατηρήσει κανείς, η δομή αυτών των αντισωμάτων και η δομή των ΑΚΔ είναι εντελώς διαφορετικές. Αυτό έρχεται σε αντίθεση με την περίπτωση της αναγνώρισης ψηφιακών χαρακτήρων, όπου αντισώματα και αντιγόνα ήταν ακολουθίες συμβόλων του ιδίου αλφαβήτου. Στην περίπτωση αυτή σκοπός της μάθησης ήταν η ταύτιση των δύο ακολουθιών. Στην ΕΔ, όμως, η έννοια της αναγνώρισης ενός αντιγόνου λαμβάνει διαφορετικές διαστάσεις· ένα αντίσωμα αναγνωρίζει καλύτερα ένα ΑΚΔ, όταν μπορεί και ταξινομεί σωστά τις εγγραφές

του. Ο τρόπος με τον οποίο ένα αντίσωμα ταξινομεί μία εγγραφή ενός ΑΚΔ βασίζεται σε ένα είδος μάθησης-ταξινόμησης που ονομάζεται *μάθηση ενός-έναντι-όλων (one-against-all learning)*. Έστω ένα αντίσωμα κωδικοποιημένο στην μορφή χρωμοσώματος του ΠΓΕ και έστω ότι το ΔΕ αυτού αντιστοιχεί στην παράσταση $P(\mathbf{x})$, όπου $\mathbf{x}$ είναι το διάνυσμα, που περιέχει τις ακριβείς τιμές των τερματικών μη σταθερών συμβόλων του αλφαβήτου των αντισωμάτων. Τότε το αντίσωμα θα ταξινομήσει μία εγγραφή $\mathbf{r}$ του ΑΚΔ στην κλάση που αντιπροσωπεύει αυτό το ΑΚΔ, αν και μόνο αν $P(\mathbf{r}) > 0$. Σε αντίθετη περίπτωση δεν θα την ταξινομήσει στην κλάση δεδομένων του ΑΚΔ. Επομένως ισχύει ο ορισμός:

Ορισμός 5.1. *Μία εγγραφή $\mathbf{r}$ ενός ΑΚΔ g, που αντιπροσωπεύει την κλάση δεδομένων $\mathcal{C}_g$, ταξινομείται στην κλάση αυτή από ένα αντίσωμα τύπου-ΠΓΕ, που μεταφράζεται στην παράσταση P, αν και μόνο αν $P(\mathbf{r}) > 0$. Διάφορα άλλως δεν ταξινομείται.*

Ο ορισμός αυτός είναι ιδιαίτερα σημαντικός, διότι συσχετίζει με συστηματικό τρόπο δύο φαινομενικά ασύνδετες αναπαραστάσεις, τα αντισώματα τύπου-ΠΓΕ και τα ΑΚΔ. Επιπλέον, επιτρέπει στον αλγόριθμο επιλογής κλώνων να εφαρμοστεί σε προβλήματα ταξινόμησης, χωρίς καμία ουσιαστική μετατροπή, πέρα από τον κατάλληλο ορισμό των αντισωμάτων και των αντιγόνων και τις αντίστοιχης συνάρτησης σύνδεσης. Αυτή η ανεξαρτησία από την αναπαράσταση καθιστά τον αλγόριθμο επιλογής κλώνων αρκετά ευέλικτο και ικανό να επιλύσει μία πληθώρα διαφορετικών προβλημάτων, χωρίς καμία μετατροπή στην δομή και την λειτουργία του. Παρ΄ όλα αυτά δεν έχει οριστεί ακόμα κάποιο ποσοτικό μέτρο της ομοιότητας αντισωμάτων-αντιγόνων παρά μόνο ο τρόπος συσχέτισής τους. Στην §5.4.3, όπου θα παρουσιαστεί και η συνάρτηση σύνδεσης αντισωμάτων-αντιγόνων, θα γίνει εκτενέστερη αναφορά σε ποσοτικά μέτρα σύγκρισης.

Για ένα δεδομένο πρόβλημα ταξινόμησης ή ΕΔ θα πρέπει να οριστεί το αλφάβητο των αντισωμάτων τύπου-ΠΓΕ. Ως συναρτησιακά σύμβολα του αλφαβήτου χρησιμοποιούνται μαθηματικοί ή λογικοί τελεστές και μαθηματικές συναρτήσεις. Μία από τις βασικότερες συναρτήσεις που χρησιμοποιούνται είναι η συνάρτηση της λογικής σύγκρισης, IF. Η λογική σύγκριση IF θα μπορούσε να υλοποιηθεί ως

$$I(x, y, z) = \begin{cases} y, & x > 0 \\ z, & x \leq 0 \end{cases} . \tag{5.1}$$

Ο ορισμός αυτός διαφέρει από τον ορισμό της λογικής σύγκρισης IF, όπως τον ορίζει ο Ferreira (2001β) κατά την παρουσίαση του μοντέλου του ΠΓΕ (βλ. Σχέση (4.4) στο Κεφάλαιο 4), όπου το αποτέλεσμα της I ήταν y, όταν $x = 0$. Παρ΄ όλα αυτά ο ορισμός αυτός υπερτερεί του ορισμού του Ferreira, διότι επιτρέπει την δημιουργία τμηματικά συνεχών συναρτήσεων, κάτι το οποίο είναι εξαιρετικά χρήσιμο στα πραγματικά προβλήματα ταξινόμησης.

Τα τερματικά σύμβολα του αλφαβήτου ορίζονται βάσει των χαρακτηριστικών των δεδομένων που είναι έτοιμα για εξόρυξη. Κάθε χαρακτηριστικό αντιστοιχίζεται σε ένα τερματικό σύμβολο και αντιστρόφως. Αυτό έχει σαν αποτέλεσμα, η διαδικασία της αποτίμησης της τιμής του ΔΕ ενός αντισώματος να είναι άμεση· σε κάθε τερματικό σύμβολο ανατίθεται η τιμή του αντιστοίχου χαρακτηριστικού και αποτιμάται η παράσταση του ΔΕ.

Η φύση των ΔΕ των αντισωμάτων, αλλά και το σύνολο των συναρτησιακών συμβόλων, που αποτελείται συνήθως από μαθηματικές συναρτήσεις, απαιτεί όλα τα χαρακτηριστικά των δεδομένων να είναι αριθμητικά. Επομένως, θα πρέπει να υπάρχει ένα τρόπος μετατροπής των κατηγορηματικών ή ονομαστικών χαρακτηριστικών σε αριθμητικά. Η τεχνική που χρησιμοποιείται για τον σκοπό αυτό ονομάζεται κατ΄ ευφημισμόν *«δυαδικοποίηση» (binarization)*, διότι μετατρέπει κάθε

μη αριθμητικό χαρακτηριστικό, σε ένα σύνολο αριθμητικών χαρακτηριστικών με πεδίο τιμών το σύνολο $\{0,1\}$. Έστω για παράδειγμα ένα χαρακτηριστικό a, το οποίο μπορεί να πάρει τιμές από το σύνολο $D = \{\nu_1, \nu_2, \nu_3\}$, όπου τα ν_i, $i = 1,2,3$ είναι γενικά ακολουθίες συμβόλων[1]. Τότε η διαδικασία της «δυαδικοποίησης» θα δημιουργήσει τρία διαφορετικά χαρακτηριστικά, έστω τα a_1, a_2 και a_3, με πεδίο τιμών το σύνολο $D' = \{0,1\}$. Ο τρόπος ανάθεσης τιμών στα a_i, $i = 1,2,3$ γίνεται βάσει της σχέσης

$$a_i = \begin{cases} 1, & \text{εάν } a = \nu_i \\ 0, & \text{ειδ΄ άλλως} \end{cases}, \quad i = 1,2,3.$$

Επομένως, κάθε φορά μόνο ένα από τα χαρακτηριστικά a_i θα είναι 1 και όλα τα άλλα θα είναι 0. Το πλεονέκτημα της μεθόδου της «δυαδικοποίησης» έγκειται στο γεγονός, ότι δεν εισάγει κανενός είδους διάταξη μεταξύ των νέων χαρακτηριστικών, όπως ακριβώς συνέβαινε και με το αρχικό κατηγορηματικό χαρακτηριστικό. Αντιθέτως, εάν κανείς όριζε ένα χαρακτηριστικό b, στην θέση του κατηγορηματικού χαρακτηριστικού a, με πεδίο τιμών $D'' = \{1,2,3\}$ και ανέθετε στο b τιμές βάσει της σχέσης

$$b = i, \quad \text{εάν } a = \nu_i, \qquad i = 1,2,3$$

τότε θα καθόριζε εμμέσως στο χαρακτηριστικό b μία διάταξη, η οποία δεν υφίσταται στο χαρακτηριστικό a. Από την άλλη πλευρά, το βασικό μειονέκτημα της διαδικασίας της «δυαδικοποίησης» είναι ότι μπορεί να δημιουργήσει μεγάλο πλήθος χαρακτηριστικών, πράγμα το οποίο, γενικά, δημιουργεί δυσκολίες στους αλγορίθμους ταξινόμησης, διότι αυξάνεται κατά πολύ η τάξη μεγέθους του προβλήματος.

5.4.3 Συνάρτηση σύνδεσης και αλγόριθμος κάλυψης

Η συνάρτηση σύνδεσης αντισωμάτων-αντιγόνων στην περίπτωση της ΕΔ αποτελεί ένα μέτρο, που καταδεικνύει την ποιότητα της ταξινόμησης που επιτυγχάνει το αντίσωμα σε σχέση με την ακριβή κλάση δεδομένων του ΑΚΔ. Ένα αντίσωμα τύπου-ΠΓΕ μπορεί να θεωρηθεί ως ένας κανόνας, ο οποίος διαχωρίζει τα προς ανάλυση δεδομένα σε δύο κατηγορίες: σε αυτά που τον ικανοποιούν (θετικά παραδείγματα), και σε αυτά που δεν τον ικανοποιούν (αρνητικά παραδείγματα).

Ορισμός 5.2. *Μία εγγραφή* $\mathbf{r}$ *των προς ανάλυση δεδομένων λέγεται ότι ικανοποιεί ένα κανόνα* R *σε μορφή αντισώματος τύπου-ΠΓΕ, όταν* $P(\mathbf{r}) > 0$*, όπου* P *είναι η παράσταση του ΔΕ του κανόνα* R*.*

Βάσει αυτού του ορισμού, μπορεί να οριστεί επίσης το σύνολο των εγγραφών του συνόλου δεδομένων, οι οποίες ικανοποιούν τον κανόνα R. Το σύνολο αυτό ονομάζεται *κάλυψη (coverage)* του κανόνα R και στην περίπτωση κανόνων κωδικοποιημένων κατά ΠΓΕ ορίζεται ως εξής:

Ορισμός 5.3. *Κάλυψη ενός κανόνα* R *σε μορφή αντισώματος τύπου-ΠΓΕ ορίζεται το σύνολο*

$$\mathcal{C}_R = \{\mathbf{r} : P(\mathbf{r}) > 0\},$$

όπου P *είναι η έκφραση στην οποία αντιστοιχεί το ΔΕ του κανόνα* R*.*

Κάθε κανόνας εξελίσσεται για να ταξινομεί τα δεδομένα μίας συγκεκριμένης κλάσης δεδομένων, επομένως η ποιότητα του κανόνα θα αξιολογείται βάσει της

[1] Για να είναι το χαρακτηριστικό a κατηγορηματικό, τα ν_i δεν θα πρέπει να ανήκουν σε κανένα αριθμητικό σύνολο.

ικανότητάς του να ταξινομεί ως θετικά όλα τα θετικά παραδείγματα της κλάσης και ως αρνητικά όλα τα αρνητικά παραδείγματά της. Τα μέτρα που αξιολογούν την ικανότητα αυτή είναι η *καθολικότητα (completeness)* και η *συνέπεια (consistency)* του κανόνα και αναλύονται ευθύς αμέσως.

Καθολικότητα και συνέπεια κανόνων

Έστω μία κλάση δεδομένων $\mathcal{C}$ και έστω P, N το πλήθος των θετικών και αρνητικών παραδειγμάτων, αντιστοίχως. Γενικά ισχύει ότι $|\mathcal{C}| = P$ και $|\overline{\mathcal{C}}| = N$, όπου $\overline{\mathcal{C}}$ είναι το συμπληρωματικό σύνολο του $\mathcal{C}$, δηλαδή το σύνολο των εγγραφών που δεν ανήκουν στην κλάση $\mathcal{C}$. Έστω επίσης ένας κανόνας R, που προσπαθεί να ταξινομήσει τα δεδομένα της κλάσης $\mathcal{C}$. Από το σύνολο των δεδομένων έστω ότι p είναι το πλήθος των θετικών παραδειγμάτων της κλάσης $\mathcal{C}$ που καλύπτει ο κανόνας, και n το πλήθος των αρνητικών παραδειγμάτων της κλάσης που καλύπτονται από τον R. Τότε ως *καθολικότητα* του κανόνα R ορίζεται το πηλίκο

$$\text{compl}(R) = \frac{p}{P}. \tag{5.2}$$

Με άλλα λόγια η καθολικότητα ενός κανόνα R για την ταξινόμηση των στιγμιοτύπων μία κλάσης δεδομένων $\mathcal{C}$ είναι το ποσοστό των θετικών παραδειγμάτων της κλάσης, που καλύπτονται από τον κανόνα. Αντιστοίχως, η *συνέπεια* ενός κανόνα R ορίζεται ως το πηλίκο

$$\text{cons}(R) = \frac{p}{p + n}. \tag{5.3}$$

Επομένως, η συνέπεια ενός κανόνα είναι το ποσοστό των θετικών παραδειγμάτων που καλύπτονται από τον κανόνα επί του συνόλου των παραδειγμάτων (θετικών και αρνητικών) που καλύπτει ο κανόνας.

Ένας κανόνας R ταξινομεί με τον καλύτερο τρόπο τα δεδομένα μίας κλάσης δεδομένων $\mathcal{C}$, όταν $\text{compl}(R) = 1$ και $\text{cons}(R) = 1$. Αυτό σημαίνει, ότι ο κανόνας θα πρέπει να καλύπτει αυστηρά και μόνο τα θετικά παραδείγματα της κλάσης $\mathcal{C}$· με άλλα λόγια θα πρέπει να ταξινομεί σωστά τόσο τα θετικά όσο και τα αρνητικά παραδείγματα της κλάσης. Στην πράξη όμως, οι απαιτήσεις για μέγιστη καθολικότητα και μέγιστη συνέπεια είναι συνήθως αντικρουόμενες. Ένας ταξινομητής που καλύπτει τα περισσότερα θετικά παραδείγματα μίας κλάσης δεδομένων (μεγάλη καθολικότητα), συνήθως καλύπτει και αρκετά αρνητικά παραδείγματα, οδηγώντας σε μικρές τιμές συνεπείας. Αντιστρόφως, ένας αρκετά συνεπής κανόνας καλύπτει συνήθως λίγα από τα θετικά παραδείγματα της κλάσης (μικρή καθολικότητα). Η σημαντικότητα καθενός από τους δύο αυτούς παράγοντες καθορίζεται κάθε φορά από το εξεταζόμενο πρόβλημα και η συνάρτηση σύνδεσης θα πρέπει να δίνει περισσότερο βάρος στον σημαντικότερο από αυτούς.

Η συνεισφορά των τιμών της καθολικότητας και της συνεπείας στην τελική ποιότητα του κανόνα, δεν είναι πάντα προφανής. Έστω για παράδειγμα ένας κανόνας με καθολικότητα 15% και συνέπεια 75%. Εάν το πλήθος των θετικών παραδειγμάτων της κλάσης είναι αρκετά μικρό, τότε ο κανόνας αυτός μπορεί να θεωρηθεί αρκετά καλός, καθ΄ ότι η συνέπεια είναι πιο δύσκολο κριτήριο για να επιτευχθεί στην προκειμένη περίπτωση, όπου τα αρνητικά παραδείγματα είναι πολύ περισσότερα από τα θετικά. Στην αντίθετη περίπτωση, όπου τα θετικά παραδείγματα είναι πολύ περισσότερα από τα αρνητικά, ο κανόνας αυτός είναι μάλλον χαμηλής ποιότητας. Για να αντιμετωπιστούν τέτοιες καταστάσεις εισάγεται ένα άλλο μέτρο ποιότητας κανόνων, το *κέρδος συνεπείας (consistency gain)* (Michalski

και Kaufman, 1999). Το κέρδος συνεπείας ενός κανόνα R ορίζεται ως

$$\text{consig}(R) = \left(\frac{p}{p+n} - \frac{P}{P+N} \right) \frac{P+N}{N}. \tag{5.4}$$

όπου τα p, n, P και N είναι όπως ορίστηκαν παραπάνω. Στην παραπάνω σχέση ο λόγος $\frac{p}{p+n}$ υποδεικνύει την κατανομή των θετικών παραδειγμάτων που καλύπτει ο αλγόριθμος, ενώ ο λόγος $\frac{P}{P+N}$ υποδεικνύει την κατανομή των όλων των θετικών παραδειγμάτων στο σύνολο των δεδομένων. Τέλος, ο παράγοντας $\frac{P+N}{N}$ χρησιμοποιείται για να είναι το κέρδος κανονικοποιημένο στο διάστημα $[0, 1]$. Όπως μπορεί να παρατηρήσει κανείς, το κέρδος συνεπείας λύνει το πρόβλημα που αναφέραμε. Πράγματι στην περίπτωση που το πλήθος των θετικών παραδειγμάτων P της κλάσης είναι αρκετά μικρό, η διαφορά $\frac{p}{p+n} - \frac{P}{P+N}$ μπορεί να πάρει σημαντικές τιμές, εάν η συνέπεια του κανόνα είναι μεγάλη. Σε αντίθετη περίπτωση, όταν το P είναι αρκετά μεγάλο, μεγάλες τιμές συνέπειας δεν οδηγούν εύκολα σε μεγάλο κέρδος συνεπείας. Γενικά, το κέρδος συνεπείας αποτελεί ένα μέτρο της ποιότητας ενός κανόνα, συγκρίνοντας τον με μία εντελώς τυχαία διαδικασία ταξινόμησης. Στην περίπτωση που $\text{consig}(R) = 0$, τότε ο κανόνας R είναι ισοδύναμος με μία τυχαία διαδικασία ταξινόμησης, ενώ όταν $\text{consig}(R) < 0$, τότε είναι χειρότερος από την τυχαία ταξινόμηση των δεδομένων.

Συνάρτηση σύνδεσης

Η συνάρτηση σύνδεσης που χρησιμοποιήθηκε στην περίπτωση του ταξινομητή που προτείνεται, είναι ίδια με την συνάρτηση προσαρμογής που χρησιμοποιούν οι Zhou et al. (2003) για τον ταξινομητή ΠΓΕ που προτείνουν. Συγκεκριμένα, η συνάρτηση σύνδεσης ορίζεται ως

$$f(R) = \begin{cases} 0, & \text{consig}(R) < 0 \\ \text{consig}(R) \cdot e^{\text{compl}(R)-1}, & \text{consig}(R) \geq 0 \end{cases}. \tag{5.5}$$

όπου $\text{consig}(R)$ είναι το κέρδος συνεπείας του κανόνα R και $\text{compl}(R)$ είναι η καθολικότητά του. Η χρήση της εκθετικής συνάρτησης κάνει την συνάρτηση σύνδεσης να προτιμά του κανόνες με μεγαλύτερη συνέπεια. Πράγματι, εάν κανείς θέσει $x = \text{consig}(R)$ και $y = \text{compl}(R)$, τότε η συνάρτηση σύνδεσης για $x \geq 0$ μπορεί να γραφεί ως

$$f(x, y) = xe^{y-1}$$

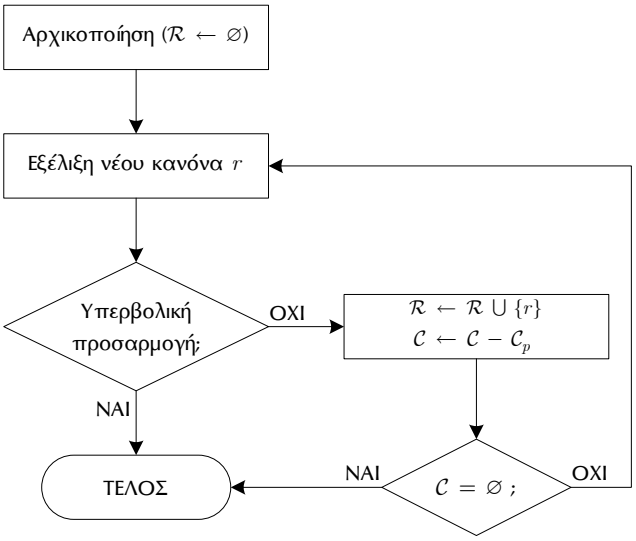
Παίρνοντας τις μερικές παραγώγους $\frac{\partial f}{\partial x}$ και $\frac{\partial f}{\partial y}$ προκύπτει

$$\frac{\partial f}{\partial x} = e^{y-1}$$
$$\frac{\partial f}{\partial y} = xe^{y-1}.$$

Επειδή όμως $x \leq 1$, εφόσον πρόκειται για το κέρδος συνεπείας, προκύπτει ότι

$$\frac{\partial f}{\partial y} \leq \frac{\partial f}{\partial x}.$$

Επομένως, η συνάρτηση σύνδεσης επηρεάζεται περισσότερο από μεταβολές του κέρδους συνεπείας, ευνοώντας κατ΄ αυτό τον τρόπο κανόνες με μεγάλο κέρδος. Τέλος, όπως έυκολα μπορεί κανείς να διαπιστώσει, η συνάρτηση σύνδεσης είναι κανονικοποιημένη στο διάστημα $[0, 1]$.



Σχήμα 5.4: Ο αλγόριθμος κάλυψης που χρησιμοποιείται για την κάλυψη των παραδειγμάτων μίας κλάσης δεδομένων.

Αλγόριθμος κάλυψης

Σε πρακτικά προβλήματα ΕΔ, οι ιδέες που περιγράφουν τις κλάσεις των δεδομένων είναι ιδιαίτερα πολύπλοκες και συνήθως δεν μπορούν να περιγραφούν-καλυφθούν από ένα μόνο κανόνα. Για τον λόγο αυτό εξελίσσονται περισσότεροι κανόνες για κάθε κλάση δεδομένων. Ο αλγόριθμος με τον οποίο δημιουργούνται οι κανόνες, με σκοπό να καλύψουν όλα τα παραδείγματα της κλάσης δεδομένων, ονομάζεται *αλγόριθμος κάλυψης (covering algorithm)*. Ο αλγόριθμος κάλυψης, που χρησιμοποιείται στον ταξινομητή που παρουσιάζεται στο παρόν Κεφάλαιο, είναι σχετικά απλός. Για κάθε κλάση δεδομένων εξελίσσεται αρχικά ένας κανόνας, με σκοπό να καλύψει όσο το δυνατόν περισσότερα θετικά παραδείγματα και όσο το δυνατόν λιγότερα αρνητικά παραδείγματα της κλάσης αυτής. Η ποιότητα του κάθε κανόνα αξιολογείται βάσει της συνάρτησης σύνδεσης που περιγράφηκε στην προηγούμενη παράγραφο. Εάν ο κανόνας αυτός δεν καλύψει όλα τα θετικά παραδείγματα της κλάσης, τότε αφαιρούνται από την κλάση αυτή τα παραδείγματα που κάλυψε ο κανόνας, και εξελίσσεται ένας νέος κανόνας με σκοπό να καλύψει τα εναπομείναντα θετικά παραδείγματα. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου καλυφθούν όλα τα θετικά παραδείγματα της κλάσης ή μέχρις ότου ενεργοποιηθεί κάποιος μηχανισμός αποφυγής υπερβολικής προσαρμογής. Πιο αυστηρά, έστω $\mathcal{C}$ η κλάση δεδομένων για την οποία θα αναπτυχθεί ένα σύνολο κανόνων $\mathcal{R}$. Τότε ο αλγόριθμος κάλυψης μπορεί να περιγραφεί από την παρακάτω ακολουθία βημάτων (Σχήμα 5.4):

Αλγόριθμος CV (Αλγόριθμος κάλυψης). Ο αλγόριθμος αυτός δημιουργεί ένα σύνολο κανόνων το οποίο καλύπτει τα δεδομένα της κλάσης δεδομένων $\mathcal{C}$.

CV1. [Αρχικοποίηση] Αρχικά το σύνολο κανόνων $\mathcal{R}$ είναι κενό·

$$\mathcal{R} \leftarrow \emptyset.$$

CV2. [Εξέλιξη νέου κανόνα] Στο στάδιο αυτό εξελίσσεται ένας νέος κανόνας r, και έστω ότι τα θετικά παραδείγματα που καλύπτει, αποτελούν το σύνολο $\mathcal{C}_p$.

CV3. [Ελεγχος κριτηρίου υπερβολικής προσαρμογής] Ο κανόνας r προστίθεται προσωρινά στο σύνολο $\mathcal{R}$ και ελέγχεται αν προκαλεί υπερβολική προσαρμογή στα δεδομένα. Εάν κάτι τέτοιο ισχύει, ο r αφαιρείται από το $\mathcal{R}$ και ο αλγόριθμος τερματίζεται. Σε αντίθετη περίπτωση ο αλγόριθμος συνεχίζει στο επόμενο βήμα.

CV4. [Απομάκρυνση παραδειγμάτων που καλύφθηκαν] Στο βήμα αυτό ο κανόνας r προστίθεται στο σύνολο κανόνων και ταυτόχρονα αφαιρούνται από την κλάση δεδομένων, τα παραδείγματα που καλύπτει. Επομένως

$$\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathrm{r}\}$$
$$\mathcal{C} \leftarrow \mathcal{C} - \mathcal{C}_{\mathrm{p}}.$$

CV5. [Συνθήκη τέλους] Εάν $\mathcal{C} \neq \emptyset$, τότε ο αλγόριθμος επαναλαμβάνεται από το Βήμα CV2. Ειδ΄ άλλως τερματίζεται. ∎

Οι κανόνες στο Βήμα CV2 του αλγορίθμου εξελίσσονται βάσει του αλγορίθμου επιλογής κλώνων, όπου οι κανόνες κωδικοποιούνται ως αντισώματα τύπου-ΠΓΕ και οι κλάσεις δεδομένων ως ΑΚΔ, όπως περιγράφηκε στις προηγούμενες παραγράφους.

5.4.4 Αποφυγή υπερβολικής προσαρμογής στα δεδομένα

Ένα σύνηθες πρόβλημα που καλούνται να αντιμετωπίσουν οι αλγόριθμοι ταξινόμησης δεδομένων, είναι η υπερβολική προσαρμογή των κανόνων που παράγουν στα δεδομένα. Αυτό γίνεται κατανοητό, εάν αναλογιστεί κανείς, ότι κύριο μέλημα των αλγορίθμων αυτών είναι η όσο το δυνατόν καλύτερη ταξινόμηση (μέγιστη καθολικότητα και συνέπεια) των προς ανάλυση δεδομένων, ή πιο απλά των δεδομένων εκπαίδευσης. Η συμπεριφορά αυτή θα ήταν απολύτως αποδεκτή, στην περίπτωση που τα δεδομένα εκπαίδευσης ήταν τελείως απηλλαγμένα από θόρυβο, κάτι το οποίο στην πράξη δεν ισχύει. Έτσι, οι αλγόριθμοι ταξινόμησης τείνουν να προσαρμόζονται καλά στα θορυβώδη δεδομένα εις βάρος της ικανότητας γενίκευσής τους. Με άλλα λόγια θα μπορούσε να πει κανείς, ότι «εξειδικεύονται» στα εκπαιδευτικά δεδομένα, μην μπορώντας να ταξινομήσουν σωστά καινούργια δεδομένα.

Το πρόβλημα της υπερβολικής προσαρμογής έχει απασχολήσει πολύ την επιστημονική κοινότητα της μάθησης μηχανών και έχουν προταθεί αρκετές τεχνικές αποφυγής της· παρ΄ όλα αυτά καμία δεν αποτελεί πανάκεια. Μία από αυτές τις τεχνικές, η οποία έχει επιδείξει καλή συμπεριφορά σε αρκετά προβλήματα ταξινόμησης και ΕΔ, είναι η *αρχή του ελαχίστου μήκους περιγραφής (Minimum Description Length principle)*, η απλά η *αρχή MDL*. Η αρχή MDL στηρίζεται σε δύο βασικές αρχές: (i) την αρχή του Occam, και (ii) το θεώρημα της δεσμευμένης πιθανότητας του Bayes.

Η *αρχή του Occam*[2] πρόκειται για μία κατά βάση εμπειρική αρχή, η οποία δεν έχει αποδειχθεί θεωρητικά, αλλά η πράξη συνήθως την επιβεβαιώνει. Η αρχή αυτή υπογορεύει το εξής:

Αρχή του Occam Από ένα σύνολο υποθέσεων που προσαρμόζονται στα δεδομένα, πρέπει να προτιμάται η απλούστερη υπόθεση.

Με τον όρο *υπόθεση* εννοείται στην ουσία ένας κανόνας, που ταξινομεί τα στιγμιότυπα μίας κλάσης δεδομένων ενός προβλήματος ταξινόμησης. Οι ενδείξεις που συνηγορούν στην αλήθεια της αρχής του Occam, αλλά και οι περιπτώσεις στις

[2]Στην ξένη βιβλιογραφία έχει επικρατήσει να ονομάζεται «ξυράφι του Occam» (Occam's razor).

οποίες μπορεί να αμφισβητηθεί, ξεφεύγουν από τους στόχους αυτής της εργασίας και γι΄ αυτό δεν θα αναλυθούν. Μία συζήτηση επί της αρχής του Occam παρατίθεται από τον Mitchell (1996).

Το θεώρημα δεσμευμένης πιθανότητας του Bayes και η αρχή MDL

Το θεώρημα δεσμευμένης πιθανότητας του Bayes παρέχει ένα τρόπο υπολογισμού της πιθανότητας ισχύος μίας υπόθεσης για μία κλάση δεδομένων, δεδομένου ότι έχει παρατηρηθεί ένα παράδειγμα αυτής της κλάσης. Έστω h μία υπόθεση για μία κλάση δεδομένων D, τότε η πιθανότητα ισχύος της h δεδομένου ότι έχει παρατηρηθεί ένα παράδειγμα της D, δίνεται από την σχέση (θεώρημα Bayes)

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}, \tag{5.6}$$

όπου με $P(x|y)$ δηλώνεται η πιθανότητα ισχύος του ενδεχομένου x, δεδομένου ότι ισχύει το ενδεχόμενο y (*δεσμευμένη πιθανότητα* του x). Επομένως, $P(D|h)$ είναι η πιθανότητα να παρατηρηθεί ένα δεδομένο της κλάσης D, δεδομένου ότι ισχύει η υπόθεση h. Η πιθανότητα $P(h)$ αποτελεί την πιθανότητα να ισχύει εκ προοιμίου η υπόθεση h. Για τον λόγο αυτό η $P(h)$ ονομάζεται *εκ προοιμίου πιθανότητα (a priori probability)* της υπόθεσης h, σε αντίθεση με την πιθανότητα $P(h|D)$ που ονομάζεται *εκ των υστέρων πιθανότητα (a posteriori probability)* της h. Τέλος, η πιθανότητα $P(D)$ δηλωνει την εκ των προτέρων πιθανότητα να παρατηρηθεί κάποιο δεδομένο της κλάσης D, χωρίς να είναι γνωστή κάποια υπόθεση. Η πιθανότητα $P(D)$ μπορεί να υπολογιστεί ως εξής: έστω H ένα σύνολο ξένων υποθέσεων[3] για την κλάση D. Τότε από το θεώρημα ολικής πιθανότητας (Κοκολάκης και Σπηλιώτης, 1991), η πιθανότητα $P(D)$ θα δίνεται από την σχέση

$$P(D) = \sum_i P(D|h_i)P(h_i), \quad \text{όπου } h_i \in H. \tag{5.7}$$

Από την παραπάνω σχέση μπορεί να παρατηρήσει κανείς, ότι η πιθανότητα $P(D)$ είναι σταθερή για ένα συγκεκριμένο πρόβλημα, καθ΄ ότι οι πιθανότητες $P(D|h_i)$ και $P(h_i)$ είναι και αυτές σταθερές και γνωστές εκ προοιμίου.

Συνήθως, σε ένα πρόβλημα μάθησης διατίθεται ένα σύνολο υποθέσεων H για μία κλάση δεδομένων D και απαιτείται να βρεθεί η πιθανότερη υπόθεση h, δεδομένου ότι έχει παρουσιαστεί κάποιο από τα παραδείγματα της D. Η υπόθεση αυτή ονομάζεται υπόθεσης *μέγιστης εκ των υστέρων πιθανότητας* ή *ΜΥΠ (maximum a posteriori probability, MAP),* και ορίζεται ως

$$\begin{aligned} h_{MAP} &\equiv \underset{h\in H}{\operatorname{argmax}} P(h|D) \\ &= \underset{h\in H}{\operatorname{argmax}} \frac{P(D|h)P(h)}{P(D)} \\ &= \underset{h\in H}{\operatorname{argmax}} P(h|D)P(h). \end{aligned} \tag{5.8}$$

Στην εξίσωση αυτή απηλείφη ο παρανομαστής $P(D)$, διότι όπως εξηγήθηκε είναι σταθερός και ανεξάρτητος της υπόθεσης h. Εάν όλες οι υπόθεσεις του H έχουν την ίδια εκ προοιμίου πιθανότητα, τότε η σχέση (5.8) μπορεί να γραφεί απλούστερα ως

$$h_{MAP} = h_{ML} = \underset{h\in H}{\operatorname{argmax}} P(D|h). \tag{5.9}$$

[3] Δύο υποθέσεις ονομάζονται *ξένες* μεταξύ τους, όταν τα σύνολα των παραδειγμάτων που καλύπτουν είναι ξένα μεταξύ τους.

Η υπόθεση h_{ML} ονομάζεται *υπόθεση μεγίστης πιθανοφάνειας (maximum likelihood hypothesis)*.

Η *αρχή του ελαχίστου μήκους περιγραφής* προκύπτει από τον ορισμό της υπόθεσης MAP, εάν αυτός θεωρηθεί από την σκοπιά της θεωρίας πληροφορίας. Πράγματι η εξίσωση (5.8) μπορεί να γραφεί ισοδύναμα ως

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} \log_2 P(D|h) + \log_2 P(h),$$

όπου η υπόθεση MAP μεγιστοποιεί το $\log_2$ του γινομένου $P(h|D)P(h)$. Αντιστοίχως, η σχέση αυτή είναι ισοδύναμη με την ελαχιστοποίηση του $-\log_2$. Επομένως, η υπόθεση MAP μπορεί να οριστεί ως

$$h_{MAP} = \underset{h \in H}{\operatorname{argmin}} -\log_2 P(D|h) - \log_2 P(h). \tag{5.10}$$

Εάν η εξίσωση (5.10) εξηγηθεί βάσει της θεωρίας πληροφορίας (Shannon και Weaver, 1949), τότε η υπόθεση MAP αποτελεί εκείνη την υπόθεση που, δεδομένης μίας συγκεκριμένης αναπαράστασης για τα δεδομένα και τις υποθέσεις, έχει το ελάχιστο μήκος. Για να γίνει αυτό κατανοητό, θα πρέπει κανείς να έχει υπ΄ όψη του το βασικό συμπέρασμα της θεωρίας πληροφορίας. Έστω, λοιπόν, το πρόβλημα σχεδίασης μίας κωδικοποίησης C, η οποία θα πρέπει να κωδικοποιεί κάποιο τυχαίο μήνυμα i, το οποίο επιλέγεται προς αποστολή με πιθανότητα p_i, έτσι ώστε να ελαχιστοποιείται ο αριθμός των bits που αποστέλλονται κατά την μετάδοσή του. Οι Shannon και Weaver (1949) απέδειξαν ότι ο βέλτιστος κώδικας C είναι εκείνος ο κώδικας, ο οποίος κωδικοποιεί κάθε μήνυμα i με $-\log_2 p_i$ bits. Το μήκος του μηνύματος i σε bits βάσει της κωδικοποίησης C ονομάζεται *μήκος περιγραφής του μηνύματος* i *βάσεις της κωδικοποίησης* C και συμβολίζεται ως $L_C(i)$.

Βάσει της λογικής αυτής οι όροι της εξίσωσης 5.10 μπορούν να εξηγηθούν ως εξής:

- Ο όρος $-\log_2 P(h)$ υποδηλώνει το μήκος περιγραφής της υπόθεσης h, υπό την προϋπόθεση ότι το σύνολο υποθέσεων H έχει κωδικοποιηθεί με την βέλτιστη κωδικοποίηση C_H. Επομένως

$$L_{C_H} = -\log_2 P(h).$$

- Ο όρος $-\log_2 P(D|h)$ υποδηλώνει το μήκος περιγραφής της κλάσης δεδομένων D, δεδομένης της υπόθεσης h, υπό την προϋπόθεση ότι το σύνολο D έχει κωδικοποιηθεί με την βέλτιστη κωδικοποίηση $C_{D|h}$. Στην περίπτωση αυτή υποθέτουμε ότι τόσο ο αποστολέας όσο και ο παραλήπτης γνωρίζουν την υπόθεση h. Τότε θα ισχύει

$$L_{C_{D|h}} = -\log_2 P(D|h).$$

Επομένως η εξίσωση (5.10) μπορεί να γραφεί πλέον ως συναρτήση των μηκών περιγραφής της υπόθεσης και των δεδομένων ως

$$h_{MAP} = \underset{h \in H}{\operatorname{argmin}} L_{C_H} + L_{C_{D|h}}. \tag{5.11}$$

Εάν επιλεγούν τυχαίες κωδικοποιήσεις για το σύνολο των υποθέσεων και τα δεδομένα, έστω C_1 και C_2, τότε η αρχή MDL μπορεί να εκφραστεί ως εξής:

Αρχή Ελαχίστου Μήκους Περιγραφής Από ένα σύνολο υποθέσεων H θα πρέπει να επιλέγεται η υπόθεση h_{MDL}, για την οποία ισχύει

$$h_{MDL} = \underset{h \in H}{\operatorname{argmin}} L_{C_1} + L_{C_2}. \tag{5.12}$$

Αυτό που πρέπει να παρατηρήσει κανείς, είναι ότι δεν ισχύει πάντα $h_{MAP} = h_{MDL}$, παρά μόνο στην περίπτωση που $C_1 = C_H$ και $C_2 = C_{D|h}$. Επομένως, για να είναι η υπόθεση MDL ίση με την υπόθεση μεγίστης εκ των υστέρων πιθανότητας, θα πρέπει το σύνολο υποθέσεων και το σύνολο δεδομένων να έχουν κωδικοποιηθεί με τον βέλτιστο τρόπο. Η εκλογή της βέλτιστης κωδικοποίησης αποτελεί και το κύριο πρόβλημα της αρχής MDL. Εάν οι κωδικοποιήσεις C_1 και C_2 είναι βέλτιστες, θα πρέπει να ισχύει

$$L_{C_1} = -\log_2 P(h)$$
$$L_{C_2} = -\log_2 P(D|h).$$

Αυτό πρακτικά σημαίνει, ότι για να ελεγχθεί εάν μία κωδικοποίηση που προτείνεται είναι βέλτιστη, θα πρέπει να είναι γνωστές οι εκ των προτέρων πιθανότητες ισχύος του συνόλου των υποθέσεων, και των δεδομένων εκπαίδευσης, δεδομένου του συνόλου υποθέσεων. Κάτι τέτοιο όμως, στα περισσότερα προβλήματα ταξινόμησης ή ΕΔ δεν υφίσταται. Επομένως, η αρχή MDL αποτελεί μόνο μία ένδειξη για το ποια μπορεί να είναι η βέλτιστη υπόθεση, χωρίς πάντα να μπορεί να την προσδιορίζει επακριβώς. Στο σημείο αυτό ανακύπτει και πάλι το θέμα της συζήτησης γύρω από την αρχή του Occam και της ισχύος της· η αρχή MDL προβάλλει ένα επιχείρημα ότι η μικρότερη ή απλούστερη υπόθεση πρέπει να προτιμάται, αλλά από την άλλη δεν εξασφαλίζει, ότι αυτό πρέπει να γίνεται σε κάθε περίπτωση.

Εφαρμογή της αρχής MDL στην αποφυγή της υπερβολικής προσαρμογής

Το κύριο μέλημα για την επιτυχή εφαρμογή της αρχής MDL είναι η κατάλληλη κωδικοποίηση του συνόλου των υποθέσεων και του συνόλου των δεδομένων. Στην περίπτωση του αλγορίθμου που εξετάζεται στο παρόν Κεφάλαιο, το σύνολο των υποθέσεων είναι ο πληθυσμός των αντισωμάτων που διατηρεί ο αλγόριθμος επιλογής κλώνων και συγκεκριμένα τα ΔΕ, στα οποία μεταφράζονται τα αντισώματα. Επομένως, θα πρέπει να βρεθεί μία κωδικοποίηση C_1, η οποία να αντικατοπτρίζει το μέγεθος των ΔΕ· όσο περισσοτέρους κόμβους έχει (μεγαλύτερη υπόθεση), τόσο μεγαλύτερο πρέπει να είναι και το μήκος της κωδικοποιημένης μορφής του. Επειδή τα αντισώματα είναι τύπου-ΠΓΕ, η ζητούμενη κωδικοποίηση μπορεί να παρασχεθεί από την γλώσσα Karva του ΠΓΕ. Πράγματι, η γλώσσα αυτή, όπως αναλύθηκε στο Κεφάλαιο 4, απεικονίζει τα χρωμοσώματα του ΠΓΕ σε ΔΕ με συνεπή και συμπαγή τρόπο. Κάθε ΔΕ μπορεί να απεικονιστεί σε ένα ανοικτό πλαίσιο ανάγνωσης (ΑΠΑ) μεταβλητού μήκους. Όσο περισσοτέρους κόμβους έχει το ΔΕ, τόσο μεγαλύτερο είναι το ΑΠΑ και αντιστρόφως. Επομένως, η κωδικοποίηση που παρέχει η γλώσσα Karva, εκπληρώνει τις προϋποθέσεις που τέθηκαν για την κωδικοποίηση C_1. Το μήκος σε bits της κωδικοποίησης κατά Karva μίας υπόθεσης h θα δίνεται από την σχέση

$$L_h = \log_2 N_c \cdot L_{ORF}, \tag{5.13}$$

όπου N_c είναι το πλήθος των διαφορετικών συμβόλων του αλφαβήτου των αντισωμάτων, δηλαδή $N_c = |F \cup T|$, όπου F και T είναι τα σύνολα των συναρτησιακών και των τερματικών συμβόλων, αντιστοίχως. Η ποσότητα L_{ORF} αντιστοιχεί στο μήκος του ΑΠΑ του ΔΕ της υπόθεσης h, και στο εξής θα ονομάζεται *ενεργό μήκος*

(effective length) του αντισώματος τύπου-ΠΓΕ, καθ΄ ότι ένα ΑΠΑ είναι το τμήμα του αντισώματος που παρουσιάζει λειτουργικότητα. Επομένως, το μήκος της κωδικοποίησης του συνόλου υποθέσεων H, θα δίνεται από την σχέση

$$\begin{aligned} L_H &= \sum_i L_{h_i} \\ &= \log_2 N_c \sum_i L_{\text{eff}_i}, \end{aligned} \tag{5.14}$$

όπου L_{eff_i} είναι το ενεργό μήκος του αντισώματος που κωδικοποιεί την υπόθεση h_i.

Έχοντας κωδικοποιήσει με συνεπή τρόπο το σύνολο υποθέσεων, μένει να βρεθεί μία κωδικοποίηση C_2 για τα δεδομένα εκπαίδευσης, δεδομένου του συνόλου των υποθέσεων. Έστω, λοιπόν, $\langle x_1, x_2, \ldots, x_m \rangle$ η ακολουθία στιγμιοτύπων των δεδομένων, και έστω $\langle f(x_1), f(x_2), \ldots, f(x_m) \rangle$ η ταξινόμησή τους. Θεωρούμε επιπλέον, χωρίς βλάβη της γενικότητας, ότι το σύνολο των δεδομένων είναι γνωστό και στον αποστολέα και στον παραλήπτη. Εάν η υπόθεση h που αποστέλλεται, ταξινομεί σωστά τα δεδομένα $\langle x_1, x_2, \ldots, x_m \rangle$, τότε δεν υπάρχει ανάγκη αποστολής καμίας επιπλέον πληροφορίας, καθ΄ ότι ο παραλήπτης θα μπορέσει να συμπεράνει την σωστή ταξινόμηση, έχοντας την υπόθεση h. Εάν όμως η υπόθεση h δεν ταξινομεί σωστά όλα τα δεδομένα, τότε θα πρέπει να αποσταλλούν οι εξαιρέσεις. Μία εξαίρεση μπορεί να κωδικοποιηθεί ως το ζεύγος $\langle i, f(x_i) \rangle$, $i = 1, \ldots, m$, όπου ο δείκτης i υποδεικνύει ποιο στιγμιότυπο ταξινομήθηκε λάθος, ενώ η ποσότητα $f(x_i)$ είναι η σωστή ταξινόμηση του στοιχείου. Για να κωδικοποιηθεί αυτό το ζεύγος τιμών απαιτούνται $\log_2 m$ bits για την κωδικοποίηση του i και $\log_2 k$ για την κωδικοποίηση του $f(x_i)$, εάν θεωρήσει κανείς ότι τα δεδομένα χωρίζονται συνολικά σε k κλάσεις.

Στην περίπτωση του αλγορίθμου που μελετάται στο Κεφάλαιο αυτό, η ανωτέρω κωδικοποίηση μπορεί να απλοποιηθεί περαιτέρω, εάν λάβει κανείς υπ΄ όψη του, ότι ένας κανόνας, ή μία υπόθεση h σύμφωνα με την συζήτηση αυτής της παραγράφου, χωρίζει τα δεδομένα σε δύο κατηγορίες: στα δεδομένα που καλύπτει και στα δεδομένα που δεν καλύπτει. Επομένως, δεν χρειάζεται να αποστέλλεται πληροφορία για την ακριβή κλάση δεδομένων στην οποία ανήκει ένα παράδειγμα, καθ΄ ότι αυτή υπονοείται από την υπόθεση h, που έχει ήδη αποσταλεί. Επιπλέον, στην προκειμένη περίπτωση, δεν είναι αναγκαίο η κωδικοποίηση C_2 να δίνει την δυνατότητα να κωδικοποιηθούν όλα τα στιγμιότυπα του προβλήματος, απαιτώντας $\log_2 m$ bits, αλλά αρκεί να επιτρέπει την κωδικοποίηση όλων των εξαιρέσεων του κανόνα. Επομένως, το πλήθος των bits που απαιτούνται για την αποστολή των εξαιρέσεων μπορεί να υπολογιστεί από την σχέση

$$L_e = \log_2 \binom{N_r}{N_{f_p}} + \log_2 \binom{N - N_r}{N_{f_n}}, \tag{5.15}$$

όπου N_r είναι το σύνολο των παραδειγμάτων που καλύπτει ο κανόνας, N_{f_p} είναι το σύνολο των *εσφαλμένων θετικών (false positive)* ταξινομήσεων, N_{f_n} είναι το σύνολο των *εσφαλμένων αρνητικών (false negative)* ταξινομήσεων, ενώ N είναι το πλήθος των δεδομένων. Με τον όρο *εσφαλμένη θετική* ταξινόμηση εννοείται ότι ο κανόνας ταξινόμησε ως θετικό, δηλαδή κάλυψε, ένα αρνητικό παράδειγμα της κλάσης δεδομένων. Αντιστρόφως, με τον όρο *εσφαλμένη αρνητική* ταξινόμηση εννοείται ότι ο κανόνας ταξινόμησε ως αρνητικό, δηλαδή δεν κάλυψε, ένα θετικό παράδειγμα της κλάσης δεδομένων. Έχοντας αυτούς τους ορισμούς κατά νου, μπορεί κάποιος εύκολα πλέον, να διαπιστώσει ότι το άθροισμα της εξίσωσης (5.15) αποτελεί το πλήθος των bits που απαιτούνται για την κωδικοποίηση των εσφαλμένων ταξινομήσεων του κανόνα που αποστέλλεται.

Η σχέση (5.15) μπορεί να εφαρμοστεί αυτούσια και στην περίπτωση που χρειάζεται να μεταδοθούν οι εξαιρέσεις ενός συνόλου κανόνων, αρκεί να οριστεί σαφώς ο τρόπος κάλυψης. Η κάλυψη ενός κανόνα δίνεται από τον Ορισμό 5.3. Παρ΄ όλα αυτά η κάλυψη ενός συνόλου κανόνων δεν μπορεί να βρεθεί τόσο απλά, και συνήθως υπολογίζεται διαδικαστικά μέσω ενός αλγορίθμου. Ο αλγόριθμος για την εύρεση της κάλυψης ενός συνόλου κανόνων $\mathcal{R}$, που χρησιμοποιείται από την προτεινόμενη μέθοδο ταξινόμησης, είναι σχετικά απλός. Σε κάθε εγγραφή των δεδομένων εκπαίδευσης εφαρμόζονται με την σειρά οι κανόνες του συνόλου, μέχρις ότου κάποιος ενεργοποιηθεί. Εάν ενεργοποιηθεί, τότε η εγγραφή αυτή προστίθεται στο σύνολο κάλυψης του συνόλου κανόνων, ειδ΄ άλλως παραμένει εκτός αυτού. Τέλος, εάν δεν ενεργοποιηθεί κανένας κανόνας, τότε θεωρείται ότι η εγγραφή δεν καλύπτεται. Έχοντας, επομένως, ορίσει σαφώς την διαδικασία εύρεσης του συνόλου κάλυψης ενός συνόλου κανόνων, το μήκος περιγραφής του, θα δίνεται από την σχέση

$$L_{\mathcal{R}} = L_e + L_t, \tag{5.16}$$

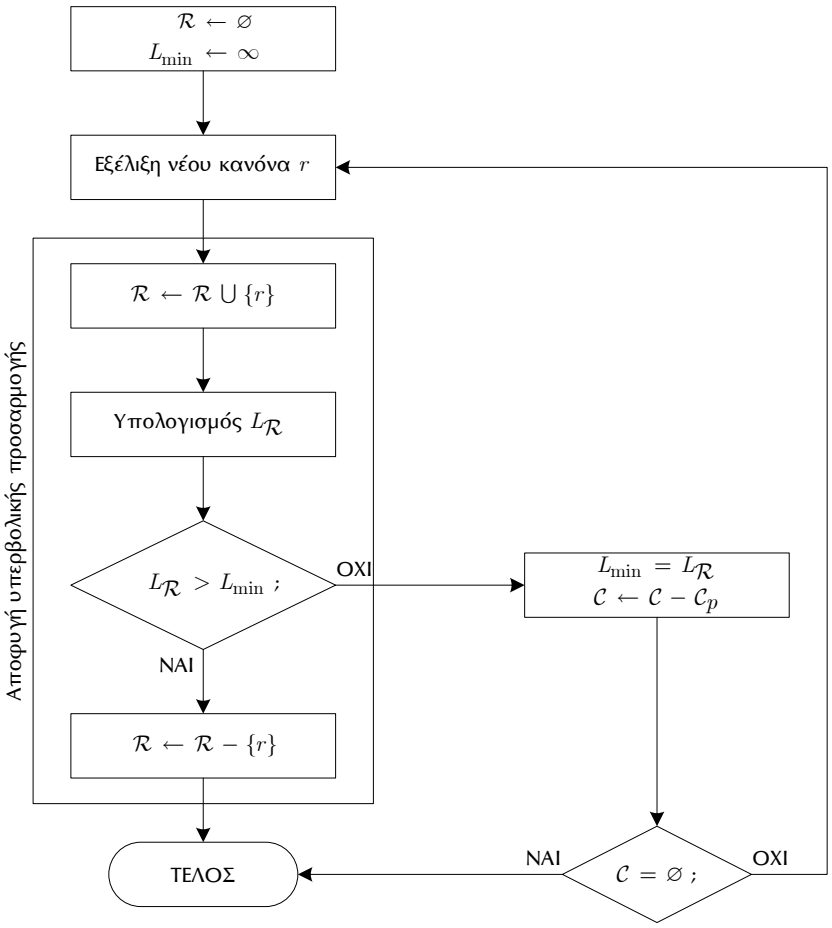
όπου $L_t = L_H$ είναι το μήκος κωδικοποίησης της «θεωρίας» που υπαγορεύει το σύνολο κανόνων $\mathcal{R}$ και L_e είναι το μήκος κωδικοποίησης των εξαιρέσεων του συνόλου των κανόνων, όπως ορίστηκε στην εξίσωση (5.15).

Η εξίσωση (5.16) ως απόρροια της αρχής MDL, πάσχει και αυτή, δυστυχώς, από το μειονέκτημα, ότι δεν μπορεί να εγγυηθεί με ασφάλεια την ορθότητα της απόφασης να απορριφθεί ένας κανόνας επειδή αυξάνει το μήκος περιγραφής του συνόλου κανόνων. Αυτό συμβαίνει, διότι στην πράξη είναι πολύ δύσκολο να αποδειχθεί ότι οι κωδικοποιήσεις που επιλέχθηκαν για τους κανόνες και για τα δεδομένα, είναι οι βέλτιστες. Για τον λόγο αυτό, εισάγεται ένας *παράγοντας χαλάρωσης* w, ο οποίος ρυθμίζει την βαρύτητα του μήκους κωδικοποίησης της θεωρίας. Έτσι, η εξίσωση (5.16) γράφεται

$$L_{\mathcal{R}} = L_e + w \cdot L_t,\ 0 \leq w \leq 1. \tag{5.17}$$

Εάν $w = 0$, τότε η θεωρία δεν παίζει κανένα ρόλο στο ολικό μήκος περιγραφής του συνόλου κανόνων. Η περίπτωση αυτή είναι ισοδύναμη με την περίπτωση που δεν εφαρμόζεται καθόλου η αρχή MDL, οπότε η υπερβολική προσαρμογή δεν είναι δυνατόν να αποφευχθεί. Πράγματι, εάν θεωρήσει κανείς ξανά τον αλγόριθμο κάλυψης που χρησιμοποιείται (Σχήμα 5.4), είναι εύκολο να παρατηρήσει, ότι με την δημιουργία κάθε νέου κανόνα το μήκος $L_{\mathcal{R}}$ θα μειώνεται, εφόσον με την προσθήκη νέων κανόνων καλύπτονται όλο και περισσότερα παραδείγματα της κλάσης δεδομένων. Ως εμπειρικό κανόνα για τον προσδιορισμό του παράγοντα w, θα μπορούσε κανείς να προτείνει, ότι όσο πιο «θορυβώδη» είναι τα προς ανάλυση δεδομένα, τόσο μεγαλύτερη πρέπει να είναι η τιμή του w, έτσι ώστε να αποφεύγεται η υπερβολική προσαρμογή. Βέβαια, για την σωστή επιλογή του w, δεν θα πρέπει να αγνοείται και το μέσο μήκος των κανόνων που παράγει ο εκάστοτε ταξινομητής. Εάν για παράδειγμα ένας ταξινομητής παράγει, γενικά, μεγάλου μήκους κανόνες, αλλά ποιοτικούς, τότε η τιμή του παράγοντα w δεν θα πρέπει να είναι πολύ μεγάλη, διότι θα οδηγεί σε πρόωρη «κλάδευση» (pruning) των κανόνων. Επομένως, η τιμή του παράγοντα w θα πρέπει να επιλέγεται με προσοχή και με σεβασμό στις ιδιαιτερότητες του κάθε ταξινομητή.

Κλείνοντας την παράγραφο για την αποφυγή της υπερβολικής προσαρμογής στα δεδομένα, θα πρέπει να εξηγηθεί σύντομα, πώς η αρχή MDL εισάγεται στον αλγόριθμο κάλυψης που περιγράφεται στην παράγραφο 5.4.3 (Σχήμα 5.4). Ο αλγόριθμος κάλυψης διατηρεί επιπλέον το ελάχιστο μήκος περιγραφής του συνόλου κανόνων, που έχει παρατηρηθεί μέχρι στιγμής. Αφότου εξελίξει ένα κανόνα, τον



Σχήμα 5.5: Ο αλγόριθμος κάλυψης με αποφυγή υπερβολικής προσαρμογής βασισμένη στην αρχή MDL.

εισάγει στο σύνολο κανόνων $\mathcal{R}$ και υπολογίζει το μήκος περιγραφής του νέου συνόλου βάσει της σχέσης (5.17). Εάν το προκύπτον μήκος είναι μικρότερο από το ελάχιστο που διατηρεί ο αλγόριθμος, τότε ο κανόνας διατηρείται στο σύνολο $\mathcal{R}$ και ανανεώνεται το ελάχιστο μήκος περιγραφής. Σε αντίθετη περίπτωση ο κανόνας προκαλεί υπερβολική προσαρμογή και εξάγεται από το σύνολο κανόνων, ενώ στην συνέχεια τερματίζεται ο αλγόριθμος. Η μορφή του αλγορίθμου κάλυψης με τον έλεγχο βάσει του κριτηρίου MDL φαίνεται στο Σχήμα 5.5.

5.4.5 Παραγωγή τελικού συνόλου κανόνων

Τελευταίο μέλημα ενός συστήματος ΕΔ είναι η παραγωγή του τελικού συνόλου κανόνων. Μέχρι στιγμής έγινε αναφορά στο πώς παράγονται μεμονωμένα σύνολα κανόνων για κάθε κλάση δεδομένων. Σε αυτό το βήμα θα πρέπει το σύστημα να συνδυάσει όλα τα σύνολα κανόνων που προέκυψαν στα προηγούμενα βήματα, έτσι ώστε να δημιουργήσει το τελικό μοντέλο των δεδομένων, που θα παρουσιαστεί στον χρήστη ως αποτέλεσμα της διαδικασίας ΕΔ. Το μοντέλο αυτό θα πρέπει να είναι σε θέση να ταξινομεί με συνέπεια οποιοδήποτε δεδομένο παρουσιαστεί στο μέλλον. Στο σημείο αυτό ανακύπτουν δύο προβλήματα:

- Δύο η περισσότεροι κανόνες ταξινομούν το ίδιο στιγμιότυπο σε διαφορετικές

κλάσεις δεδομένων (*διαφωνία ταξινόμησης–classification conflict*).

- Ένα στιγμιότυπο δεν μπορεί να ταξινομηθεί από κανένα κανόνα (*απόρριψη δεδομένων–data rejection*).

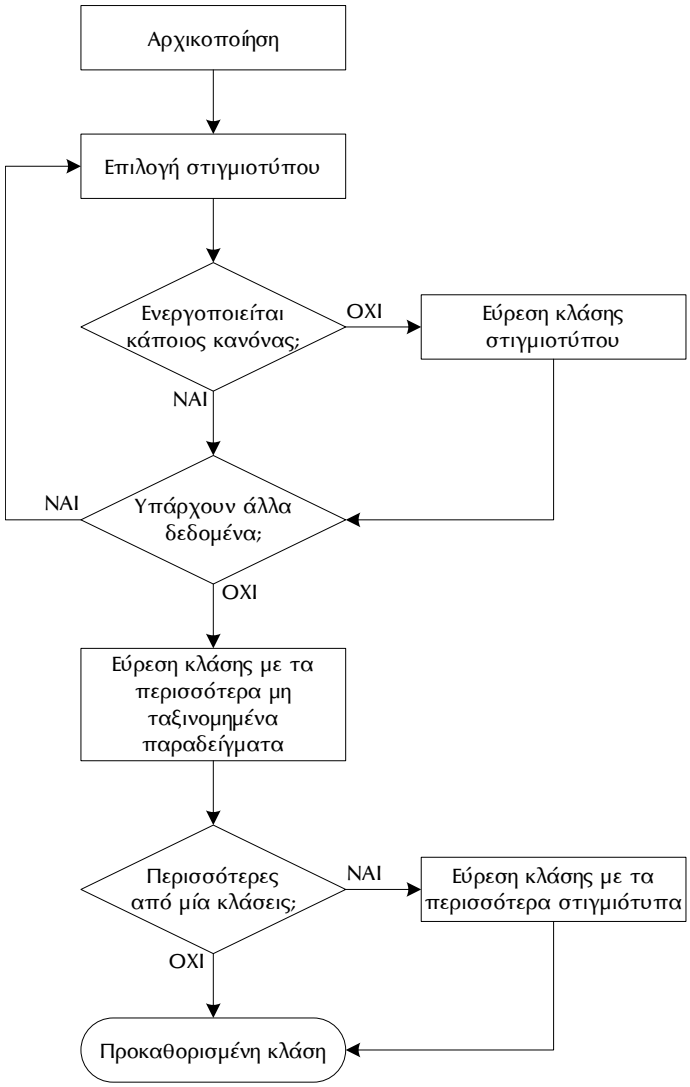
Για να επιλυθεί το πρώτο πρόβλημα, θα πρέπει η μέθοδος ταξινόμησης να ορίζει ένα τρόπο εφαρμογής των κανόνων, έτσι ώστε να αποφεύγεται το πρόβλημα της διαφωνίας ταξινόμησης. Η λύση στο δεύτερο πρόβλημα παρέχεται με την μορφή μίας *προκαθορισμένης κλάσης (default class)* δεδομένων. Η κλάση αυτή πρέπει να είναι μία από τις κλάσεις δεδομένων του προβλήματος. Όποιο στιγμιότυπο απορριφθεί από το σύνολο κανόνων, ταξινομείται στην προκαθορισμένη κλάση.

Στην μέθοδο που προτείνεται στο Κεφάλαιο αυτό, μόλις τελειώσει η φάση παραγωγής των κανόνων, τα ανεξάρτητα σύνολα κανόνων που έχουν παραχθεί, συνενώνονται και δημιουργούν ένα σύνολο κανόνων που περιέχει όλους τους κανόνες που παρήγαγε το σύστημα. Στη συνέχεια, οι κανόνες ταξινομούνται βάσει της ποιότητάς τους (βλ. συνάρτηση σύνδεσης, §5.4.3). Το ταξινομημένο αυτό σύνολο κανόνων αποτελεί το τελικό σύνολο κανόνων, που παρουσιάζεται στο χρήστη.

Ο ορισμός της προκαθορισμένης κλάσης δεδομένων γίνεται αφότου οριστεί το τελικό σύνολο κανόνων. Για κάθε κλάση δεδομένων του προβλήματος διατηρείται μία μεταβλητή, που αποθηκεύει τον αριθμό των παραδειγμάτων της κλάσης που δεν ταξινομήθηκαν από κανένα κανόνα του συνόλου. Στην συνέχεια εξετάζονται διαδοχικά όλα τα δεδομένα του προβλήματος. Εάν κάποιο δεν ταξινομηθεί από κανένα κανόνα, τότε βρίσκεται η κλάση δεδομένων στην οποία ανήκει και αυξάνεται η μεταβλητή των μη ταξινομημένων στιγμιοτύπων της κλάσης αυτής. Μόλις εξεταστούν όλα τα δεδομένα του προβλήματος, επιλέγεται ως προκαθορισμένη κλάση, η κλάση με τα περισσότερα μη ταξινομημένα στιγμιότυπα. Σε περίπτωση που δύο οι περισσότερες κλάσεις έχουν τον ίδιο αριθμό μη ταξινομημένων παραδειγμάτων, επιλέγεται ως προκαθορισμένη κλάση, η κλάση με τον μεγαλύτερο αριθμό στιγμιοτύπων. Η διαδικασία εύρεσης της προκαθορισμένης κλάσης απεικονίζεται στο Σχήμα 5.6.

5.5 Εφαρμογή της μεθόδου σε προβλήματα αξιολόγησης

Έχοντας περιγράψει και αναλύσει όλες τις συνιστώσες του αλγορίθμου ταξινόμησης με συνδυασμό ΤΑΣ και ΠΓΕ, στην παρούσα παράγραφο θα παρουσιαστούν τα αποτελέσματα της εφαρμογής του σε δύο προβλήματα αξιολογήσης (benchmark problems), που μπορεί κανείς να βρει στην ηλεκτρονική βιβλιοθήκη συνόλων δεδομένων αξιολόγησης του πανεπιστημίου UCI της Καλιφόρνια (UCI repository). Το πρώτο από τα δύο προβλήματα (σειρά προβλημάτων MONK) πρόκειται για ένα τεχνητό πρόβλημα που δημιουργήθηκε από μία ομάδα επιστημόνων (Thrun et al., 1991) με σκοπό την αξιολόγηση των αλγορίθμων μάθησης μηχανών. Το δεύτερο πρόβλημα (πρόβλημα Pima Indians Diabetes) πρόκειται για ένα πραγματικό πρόβλημα, που βασίζεται στα δεδομένα μίας ιατρικής ΒΔ για μία μορφή διαβήτη μίας φυλής Ινδιάνων, που ζεί στις ΗΠΑ. Ως μέτρο αξιολόγησης του αλγορίθμου, σε κάθε πρόβλημα χρησιμοποιείται η *ακρίβεια πρόβλεψης (prediction accuracy)* του συνόλου κανόνων που παράγεται. Ως ακρίβεια πρόβλεψης ορίζεται το ποσοστό των επιτυχημένων προβλέψεων του συνόλου κανόνων επί ολοκλήρου του συνόλου δεδομένων.



Σχήμα 5.6: Ο αλγόριθμος εύρεσης της προκαθορισμένης κλάσης δεδομένων.

5.5.1 Τα προβλήματα MONK

Τα προβλήματα MONK βασίζονται σε ένα τεχνητό κόσμο ρομπότ, στον οποίο τα ρομπότ περιγράφονται από έξι διαφορετικά χαρακτηριστικά:

x_1 :	ΣΧΗΜΑ_ΚΕΦΑΛΗΣ	$\in$	ΣΤΡΟΓΓΥΛΟ, ΤΕΤΡΑΓΩΝΙΚΟ, ΟΚΤΑΓΩΝΙΚΟ
x_2 :	ΣΧΗΜΑ_ΣΩΜΑΤΟΣ	$\in$	ΣΤΡΟΓΓΥΛΟ, ΤΕΤΡΑΓΩΝΙΚΟ, ΟΚΤΑΓΩΝΙΚΟ
x_3 :	ΧΑΜΟΓΕΛΑΣΤΟ	$\in$	ΝΑΙ, ΟΧΙ
x_4 :	ΚΡΑΤΑΕΙ	$\in$	ΣΠΑΘΙ, ΜΠΑΛΟΝΙ, ΣΗΜΑΙΑ
x_5 :	ΧΡΩΜΑ_ΚΟΣΤΟΥΜΙΟΥ	$\in$	ΚΟΚΚΙΝΟ, ΚΙΤΡΙΝΟ, ΠΡΑΣΙΝΟ, ΜΠΛΕ
x_6 :	ΕΧΕΙ_ΓΡΑΒΑΤΑ	$\in$	ΝΑΙ, ΟΧΙ

Πάνω σε αυτό τον κόσμο ρομπότ ορίζονται τρία δυαδικά προβλήματα ταξινόμησης. Σε κάθε πρόβλημα δίνεται η περιγραφή μίας κλάσης ρομπότ· ένα ρομπότ είτε ανήκει είτε δεν ανήκει σε αυτή την κλάση. Συνολικά, στον κόσμο υπάρχουν 432 διαφορετικά ρομπότ, αλλά σε κάθε πρόβλημα παρέχεται ως σύνολο εκπαίδευσης ένα τυχαίο υποσύνολο του κόσμου με συγκεκριμένο μέγεθος. Σκοπός ενός αλγορίθμου ταξινόμησης είναι, αφότου εκπαιδευθεί, να μπορέσει να ταξινομήσει σωστά ολόκληρο των πληθυσμό των ρομπότ και παράλληλα να παράσχει, εάν αυτό είναι δυνατόν, μία περιγραφή της ταξινόμησης που παρήγαγε. Τα τρία προβλήματα MONK είναι:

- **Πρόβλημα** M_1. Η κλάση ρομπότ που ορίζει το πρόβλημα αυτό περιγράφεται από τον εξής κανόνα:

  $$(\text{ΣΧΗΜΑ\_ΚΕΦΑΛΗΣ} = \text{ΣΧΗΜΑ\_ΣΩΜΑΤΟΣ}) \text{ ή}$$
  $$(\text{ΧΡΩΜΑ\_ΚΟΣΤΟΥΜΙΟΥ} = \text{ΚΟΚΚΙΝΟ})$$

  Το σύνολο εκπαίδευσης αποτελείται από 132 τυχαία εκλεγμένα παραδείγματα από τον κόσμο των ρομπότ, ενώ δεν εισάγεται θόρυβος.

- **Πρόβλημα** M_2. Η κλάση ρομπότ του προβλήματος αυτού ορίζεται από την πρόταση:

  > Ακριβώς δύο από τα χαρακτηριστικά του ρομπότ έχουν την *πρώτη* δυνατή τιμή τους.

  Για παράδειγμα ένα ρομπότ που έχει στρογγυλό κεφάλι και στρογυλλό σώμα, αλλά δεν είναι χαμογελαστό, δεν κρατάει σπαθί, δεν φοράει κόκκινο κοστούμι και δεν έχει γραβάτα, ανήκει στην κλάση που ορίζει το πρόβλημα M_2. Το σύνολο εκπαίδευσης αποτελείται από 169 τυχαία εκλεγμένα παραδείγματα, ενώ και στην περίπτωση αυτή δεν υπάρχει θόρυβος.

- **Πρόβλημα** M_3. Το πρόβλημα αυτό ορίζεται από τον κανόνα:

  $$(\text{ΧΡΩΜΑ\_ΚΟΣΤΟΥΜΙΟΥ} = \text{ΠΡΑΣΙΝΟ και ΚΡΑΤΑΕΙ} = \text{ΣΠΑΘΙ}) \text{ ή}$$
  $$(\text{ΧΡΩΜΑ\_ΚΟΣΤΟΥΜΙΟΥ} \neq \text{ΜΠΛΕ και ΣΧΗΜΑ\_ΚΕΦΑΛΗΣ} \neq \text{ΟΚΤΑΓΩΝΙΚΟ})$$

  Το σύνολο εκπαίδευσης στην περίπτωση αυτή, αποτελείται από 122 τυχαία εκλεγμένα παραδείγματα, ενώ εισάγεται και 5% θόρυβος (λάθος ταξινομήσεις).

Τα προβλήματα M_1 και M_3 δίνονται σε *διαζευκτική κανονική μορφή (disjunctive normal form–DNF)* και μπορούν σχετικά εύκολα να επιλυθούν από αλγορίθμους συμβολικής μάθησης, όπως είναι τα Δένδρα Απόφασης (Desicion Trees) και ο αλγόριθμος που προτείνεται στο Κεφάλαιο αυτό. Αντιθέτως, το πρόβλημα M_2 είναι αρκετά δύσκολο να εκφραστεί σε μορφή DNF και επόμενως είναι δύσκολο να επιλυθεί από τέτοιους αλγορίθμους.

Παράμετροι αλγορίθμου και επίλυση των προβλημάτων

Επειδή ο κόσμος των προβλημάτων MONK περιγράφεται από κατηγορηματικά χαρακτηριστικά, χρησιμοποιείται η τεχνική της «δυαδικοποίησης», που παρουσιάστηκε στην §5.4.2. Έτσι το χαρακτηριστικό x_1 χωρίζεται σε τρία δυαδικά χαρακτηριστικά (x_{11}, x_{12}, x_{13}), το x_2 πάλι σε τρία, κ.ο.κ. Με αυτό τον τρόπο, τα μετασχηματισμένα προβλήματα θα αποτελούνται από 17 χαρακτηριστικά.

Τα αντισώματα χρησιμοποιούν συναρτησιακά σύμβολα από το σύνολο

$$F = \{I, +, -, \times, O, A\},$$

όπου I είναι το συναρτησιακό σύμβολο της λογικής συνάρτησης IF, O είναι το σύμβολο της λογικής συνάρτησης OR και A είναι το σύμβολο της λογικής συνάρτησης AND. Οι συναρτήσης O και A ορίζονται ως

$$O(x, y) = \begin{cases} 1, & \text{αν } x \neq 0 \text{ ή } y \neq 0 \\ 0, & \text{ειδ΄ άλλως} \end{cases}, \tag{5.18}$$

$$A(x, y) = \begin{cases} 1, & \text{αν } x \neq 0 \text{ και } y \neq 0 \\ 0, & \text{ειδ΄ άλλως} \end{cases}. \tag{5.19}$$

ενώ η συνάρτηση IF ορίζεται από την σχέση (5.1) (βλ. §5.4.2). Οι υπόλοιποι παράμετροι του αλγορίθμου για κάθε πρόβλημα παρουσιάζονται στον Πίνακα 5.1. Αυτό που αξίζει να παρατηρήσει κανείς είναι οι διαφορές στις τιμές του παράγοντα χαλάρωσης w. Στα προβλήματα M_1 και M_2, τα οποία δεν περιέχουν καθόλου θόρυβο, μπορεί κανείς με ασφάλεια να μειώσει τον παράγοντα w, καθ΄ ότι δεν πρόκειται να συμβεί υπερβολική προσαρμογή. Για τον λόγο αυτό δίνεται στο w η αρκετά μικρή τιμή $w = 0,1$, με σκοπό να παραχθούν όσο το δυνατόν πιο ακριβείς κανόνες. Αντιθέτως, στο πρόβλημα M_3, επειδή υπάρχει θόρυβος, δίνεται μία μεγαλύτερη τιμή στο w ($w = 0,3$), έτσι ώστε να αποφευχθεί η υπερβολική προσαρμογή. Επιπλέον και οι δύο τιμές του w είναι σχετικά μικρές, καθ΄ ότι όπως αποδεικνύεται στην πράξη, ο αλγόριθμος που προτείνεται παράγει σχετικά μεγάλους κανόνες. Τέλος, ως πρόσθετο κριτήριο τερματισμού του αλγορίθμου τίθεται και ένας μέγιστος αριθμός κανόνων ανά κλάση που μπορεί να παράγει.

Για την αξιολόγηση του αλγορίθμου ως σύνολα εκπαίδευσης και δοκιμής και για τα τρία προβλήματα χρησιμοποιήθηκαν τα αντίστοιχα αρχεία που παρέχονται από το UCI repository. Και στα τρία προβλήματα ο προτεινόμενος αλγόριθμος επέδειξε πολύ καλή συμπεριφορά επιτυγάνοντας μεγάλες τιμές ακριβείας πρόβλεψης (Πίνακας 5.2). Στο πρόβλημα M_1 πέτυχε ακρίβεια πρόβλεψης 100%, όπως και ο αλγόριθμος ΠΓΕ των Zhou et al. (2003). Στα άλλα δύο προβλήματα μένει ελαφρώς πίσω σε σχέση με τον κλασσικό ΠΓΕ, πετυχαίνοντας ακρίβεια 93,52% στο πρόβλημα M_2 έναντι 99,07% του ΠΓΕ, και 98,61% έναντι 100% στο M_3. Παρ΄ όλα αυτά το σημαντικότερο πλεονέκτημα του αλγορίθμου που προτείνεται, είναι η ταχύτητα σύγκλισης, αλλά και οι μικρές απαιτήσεις του σε υπολογιστικούς πόρους, λόγω του πολύ μικρού αρχικού πληθυσμού που διατηρεί (Πίνακας 5.3). Πράγματι, για την επίτευξη αυτών των τιμών ακριβείας για τα προβλήματα M_1 και M_3 απαιτήθηκαν 100 γενεές εξέλιξης, ενώ για το πρόβλημα M_2 απαιτήθηκαν 1000, την στιγμή που ο αλγόριθμος ΠΓΕ εκπαιδεύεται για 5000 γενεές. Επιπλέον ο αλγόριθμος που προτείνεται, διατηρεί ένα σταθερό πληθυσμό 10 ατόμων για τα προβλήματα M_1 και M_3, που εν δυνάμει πλησιάζουν τα 228 μόνο κατά την διάρκεια της κλωνοποίησης (βλ. σχέση (3.5) στην §3.4.2). Αντιθέτως, ο κλασσικός ΠΓΕ χρησιμοποιεί και διαχειρίζεται ένα σταθερό πληθυσμό 1000 ατόμων. Στο πρόβλημα M_2 λόγω της δυσκολίας αναπαραγωγής του σε μορφή DNF, χρησιμοποιείται

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ MONK (ΠΑΡΑΜΕΤΡΟΙ ΑΛΓΟΡΙΘΜΟΥ)			
Παράμετρος	*Τιμή*		
	Πρόβλημα M_1	*Πρόβλημα* M_2	*Πρόβλημα* M_3
Μέγιστος αριθμός γενεών	100	1000	100
Μέγιστος αριθμός κανόνων/κλάση	3	4	3
Μήκος κεφαλής γονιδίων (h)	33	33	33
Γονίδια/αντίσωμα	1	1	1
Μήκος αντισωμάτων (L)	100	100	100
Μέγεθος πληθυσμού ($\|\mathcal{P}\|$)	10	100	10
Μέγεθος μνήμης ($\|\mathcal{M}\|$)	1	1	1
Αντισώματα προς επιλογή (n_b)	5	50	5
Αντισώματα προς αντικατάσταση (n_r)	0	0	0
Αντισώματα προς ανανέωση (n_d)	0	0	0
Αντισώματα προς επεξεργασία (n_e)	2	10	2
Μέγεθος δεξαμενής αντισωμάτων (n_p)	2	10	2
Μέγιστος ρυμθός μετάλλαξης (α_{max})	1,0	1,0	1,0
Εξασθένιση ρυθμού μετάλλαξης (ρ)	5,0	5,0	5,0
Παράγοντας κλωνοποίησης (β)	20,0	20,0	20,0
Παράγοντας χαλάρωσης (w)	0,1	0,3	0,1

Πίνακας 5.1: Οι παράμετροι του αλγορίθμου για την επίλυση των προβλημάτων MONK.

ΣΥΓΚΡΙΣΗ ΑΚΡΙΒΕΙΑΣ ΚΑΝΟΝΩΝ (ΠΡΟΒΛΗΜΑΤΑ MONK)			
Αλγόριθμος	*Πρόβλημα*		
	M_1	M_2	M_3
C4.5	75,70%	65%	97,20%
C4.5Rules	100%	66,20%	96,30%
GEP	100%	99,07%	100%
AIS+GEP	**100%**	**93,52%**	**98,61%**

Πίνακας 5.2: Σύγκριση ακριβείας κανόνων διαφόρων αλγορίθμων ΕΔ. Οι διαφορές μεταξύ του κλασσικού GEP και του αλγορίθμου που προτείνεται είναι πολύ μικρές.

ΣΥΓΚΡΙΣΗ ΣΥΓΚΛΙΣΗΣ ΚΑΙ ΠΛΗΘΥΣΜΩΝ				
Αλγόριθμος	*Γενεές Εξέλιξης*		*Πληθυσμός*	
	M_1, M_3	M_2	M_1, M_3	M_2
GEP	5000	5000	1000	1000
AIS+GEP	100	1000	10(228)	100(4498)

Πίνακας 5.3: Σύγκριση σύγκλισης και πληθυσμών του αλγορίθμου GEP και του AIS+GEP που προτείνεται. Οι διαφορές στην ταχύτητα σύγκλισης και στους υπολογιστικούς πόρους που απαιτούνται είναι εμφανείς. Σε παρένθεση δηλώνονται οι μέγιστες τιμές του πληθυσμού του AIS+GEP κατά την φάση της κλωνοποίησης.

ένας αρχικός πληθυσμός 100 ατόμων, που εν δυνάμει φθάνει τα 4498 άτομα κατά την διάρκεια της κλωνοποίησης.

Τα σύνολα κανόνων που παρήχθησαν φαίνονται στην συνέχεια. Αυτό που αξίζει να παρατηρήσει κανείς είναι η πολυπλοκότητα των κανόνων για το πρόβλημα M_2, κάτι το οποίο θα πρέπει να είναι αναμενόμενο, δεδομένης της φύσης του συγκεκριμένου προβλήματος, καθ΄ ότι δεν μπορεί να αναπαρασταθεί σε μορφή DNF.

MONK 1

```
IF ( AND(AND(OR(-(IF(a5_1,a1_3,OR(+(a1_1,a4_3),3)),a2_2),
    a5_4),2),*(a2_3,a1_2)) > 0 ) THEN
    Class0
ELSEIF ( AND(+(*(a6_1,+(a5_2,-(2,a1_1))),a6_2),
    AND(IF(a2_1,a1_2,*(-(a5_1,AND(2,+(OR(1,a4_3),
    *(-(a4_2,a2_2),a2_3)))),*(AND(2,a1_1),1))),
    OR(1,a4_2))) > 0 ) THEN
    Class0
ELSEIF ( *(AND(a1_3,OR(a2_2,a2_1)),
    +(a5_2,a5_4)) > 0 ) THEN
    Class0
ELSEIF ( a5_1 > 0 ) THEN
    Class1
ELSEIF ( AND(IF(AND(a5_2,*(a5_1,IF(a1_3,-(IF(a6_2,a2_3,
    IF(AND(a5_4,1),a2_2,a5_3)),+(AND(a4_3,a5_2),a2_2)),
    -(OR(OR(a4_1,a2_2),a6_1),a1_1)))),a2_2,*(a2_3,
    -(a1_3,AND(a5_1,1)))),a1_3) > 0 ) THEN
    Class1
ELSEIF ( IF(a1_1,a2_1,a2_2) > 0 ) THEN
    Class1
ELSE
    Class0
ENDIF
```

MONK 2

```
IF ( *(AND(IF(AND(a6_1,a5_1),a4_3,a2_1),a3_2),IF(1,
    IF(+(a1_2,a3_2),-(a1_1,+(a5_2,*(+(a2_1,a6_2),
    -(a6_2,a4_3)))),-(*(IF(a4_3,a1_2,*(a1_2,a3_1)),
    +(2,a1_2)),a5_3)),a4_2)) > 0 ) THEN
    Class0
ELSEIF ( +(AND(-(+(AND(*(+(a3_1,a6_1),a4_1),IF(IF(a6_2,
    a3_1,a3_2),2,a2_1)),a4_3),a4_3),a2_3),-(-(1,*(+(a5_1,
    OR(a4_1,a1_1)),IF(AND(3,a6_1),3,OR(a2_2,a2_3)))),
    OR(AND(a2_1,2),+(-(-(a4_1,a1_1),IF(a5_4,a3_1,a3_1)),
    a6_2)))) > 0 ) THEN
    Class0
ELSEIF ( -(*(IF(a1_1,a2_2,IF(a5_1,OR(a3_2,a1_1),a5_2)),
    a6_1),OR(a3_1,IF(AND(-(AND(a5_3,a1_1),a1_1),a6_1),
    IF(a4_3,-(*(a4_3,1),IF(a1_1,a5_4,a5_2)),+(a2_3,+(a4_1,
    a5_2))),-(a5_2,IF(OR(a6_2,a3_2),OR(a2_1,a4_1),
```

```
    a6_1))))) > 0 ) THEN
    Class1
ELSEIF ( AND(IF(+(a2_1,a5_1),-(OR(a5_1,IF(OR(a1_2,IF(a4_3,
    +(2,a2_1),a5_4)),OR(-(a3_1,a4_1),a1_1),OR(a4_3,
    OR(-(OR(a2_2,2),a1_1),*(a5_2,a5_4))))),a6_2),a3_2),
    a6_2) > 0 ) THEN
    Class0
ELSEIF ( AND(*(+(a6_1,a5_1),OR(IF(IF(a3_2,a2_1,a4_1),a4_1,
    a1_1),OR(a2_1,AND(IF(+(*(-(a3_2,a1_2),a4_3),+(a6_2,
    *(a3_1,a4_1))),-(a3_1,a4_3),a5_1),a6_1)))),
    OR(a4_1,a3_1)) > 0 ) THEN
    Class0
ELSEIF ( *(IF(OR(a2_1,IF(-(a4_2,a3_1),a4_3,+(IF(a1_1,a1_1,
    a2_1),AND(a3_1,a5_2)))),OR(IF(a1_2,OR(-(a5_3,a4_2),
    a6_2),a1_3),IF(+(IF(a5_4,a1_2,a4_2),AND(a1_1,a2_1)),
    a3_1,2)),a5_1),AND(-(AND(+(a4_1,a5_1),2),a3_1),
    a6_2)) > 0 ) THEN
    Class1
ELSEIF ( -(AND(a3_1,*(-(a2_2,AND(a1_2,a2_3)),
    IF(*(OR(IF(a6_2,+(a3_1,a4_3),IF(a1_1,a1_1,a5_1)),a3_2),
    OR(-(+(a5_1,a6_1),OR(a5_1,2)),a1_1)),*(+(a4_3,a2_2),
    a2_3),*(*(IF(a3_2,a2_3,2),OR(2,a2_3)),2)))),
    a4_1) > 0) THEN
    Class1
ELSE
    Class1
ENDIF
```

MONK 3

```
IF ( +(*(1,+(+(a2_1,a2_2),-(-(+(-(2,AND(a5_4,1)),a5_2),2),
    IF(IF(3,2,a5_3),a5_2,*(*(a4_3,a1_3),OR(3,a5_1))))))),
    *(OR(a4_3,*(1,a3_1)),AND(a5_3,a4_1))) > 0 ) THEN
    Class1
ELSEIF ( a2_3 > 0 ) THEN
    Class0
ELSE
    Class0
ENDIF
```

5.5.2 Το πρόβλημα Pima Indians Diabetes

Τα δεδομένα του προβλήματος Pima Indians Diabetes αποτελούνται από 8 αριθμητικά χαρακτηριστικά και ανήκουν σε 2 κλάσεις. Η πρώτη κλάση (*Κλάση 0*) αντιπροσωπεύει τα άτομα, στα οποία το τεστ διαβήτη είχε αρνητικό αποτέλεσμα, ενώ η δεύτερη κλάση (*Κλάση 1*) αντιπροσωπεύει τα διαβητικά άτομα. Συνολικά υπάρχουν 768 εγγραφές, εκ των οποίων οι 500 ανήκουν στην Κλάση 0, ενώ οι υπόλοιπες 268 ανήκουν στην Κλάση 1. Η δυσκολία του προβλήματος Pima Indians

ΕΠΙΛΥΣΗ Pima Indians Diabetes	
Παράμετρος	*Τιμή*
Μέγιστος αριθμός γενεών	500
Μέγιστος αριθμός κανόνων/κλάση	3
Μήκος κεφαλής αντισωμάτων (h)	33
Μήκος αντισωμάτων (L)	100
Γονίδια/αντίσωμα	1
Μέγεθος πληθυσμού ($\|\mathcal{P}\|$)	20
Μέγεθος μνήμης ($\|\mathcal{M}\|$)	1
Αντισώματα προς επιλογή (n_b)	10
Αντισώματα προς αντικατάσταση (n_r)	0
Αντισώματα προς ανανέωση (n_d)	0
Αντισώματα προς επεξεργασία (n_e)	5
Μέγεθος δεξαμενής αντισωμάτων (n_p)	5
Μέγιστος ρυμθός μετάλλαξης (α_{max})	1,0
Εξασθένιση ρυθμού μετάλλαξης (ρ)	5,0
Παράγοντας κλωνοποίησης (β)	20,0
Παράγοντας χαλάρωσης (w)	0,3

Πίνακας 5.4: Οι παράμετροι του αλγορίθμου για την επίλυση του προβλήματος Pima Indians Diabetes.

Diabetes έγκειται στο γεγονός, ότι τα δεδομένα του δεν μπορούν να χωριστούν εύκολα σε κατηγορίες, καθ΄ ότι υπάρχουν πολλές εγγραφές που δεν μπορούν να ταξινομηθούν (outliers). Για τον λόγο αυτό, κατά τον καθορισμό των παραμέτρων του αλγορίθμου, θα πρέπει να δοθεί ιδιαίτερη προσοχή στον τομέα της υπερβολικής προσαρμογής. Οι παράμετροι που χρησιμοποιήθηκαν φαίνονται στον Πίνακα 5.4, όπου μπορεί κανείς να παρατηρήσει την αυξημένη τιμή του παράγοντα χαλάρωσης ($w = 0,3$), που σκοπό έχει την αποφυγή της υπερβολικής προσαρμογής.

Όσον αφορά στο σύνολο συναρτησιακών συμβόλων του αλφαβήτου, χρησιμοποιήθηκαν κυρίως αλγεβρικές συναρτήσεις, καθ΄ ότι τα χαρακτηριστικά του προβλήματος είναι αριθμητικά. Έτσι, στην περίπτωση αυτή το σύνολο συναρτησιακών συμβόλων είναι $F = \{+, -, \times, \div, Q\}$, όπου Q είναι η συνάρτηση της τετραγωνικής ρίζας.

Και σε αυτή την περίπτωση ο αλγόριθμος που προτείνεται, επέδειξε συμπεριφορά αντίστοιχη του απλού ΠΓΕ, όντας περίπου 2,5% υποδεέστερος (Πίνακας 5.5). Παρ΄ όλα αυτά συνέκλινε πολύ γρήγορα επιτυγχάνοντας την επίδοση αυτή σε 500 γενεές, έναντι 5000 του ΠΓΕ. Η διαφορά στον αρχικό πληθυσμό ήταν και σε αυτή την περίπτωση σημαντική, με τον προτεινόμενο αλγόριθμο να διατηρεί ένα πληθυσμό 20 ατόμων, με μέγιστη τιμή κατά την κλωνοποίηση τα 586 άτομα, ενώ ο ΠΓΕ διατηρεί συνεχώς πληθυσμό 1000 ατόμων. Επιπλέον, το σύνολο κανόνων που δημιουργήθηκε από τον προτεινόμενο αλγόριθμο ήταν αρκετά πιο συμπαγές αποτελούμενο από 2 κανόνες έναντι 5 του ΠΓΕ. Κλείνοντας την παράγραφο αυτή, να σημειωθεί ότι η τεχνική αξιολόγησης, που χρησιμοποιήθηκε για το πρόβλημα Pima Indians Diabetes ήταν η τεχνική της *διασταυρουμένης αξιολόγησης 5 τμημάτων (5-cross-validation)*. Κατά την τεχνική αυτή χωρίζονται τα δεδομένα σε 5 τμήματα και ο αλγόριθμος εκτελείται 5 φορές, χρησιμοποιώντας ως σύνολο εκπαίδευσης ένα τμήμα κάθε φορά, και τα υπόλοιπα τμήματα ως σύνολο δοκιμής. Ως τελική τιμή ακριβείας του αλγορίθμου, χρησιμοποιείται ο μέσος όρος και των 5 δοκιμών.

ΑΚΡΙΒΕΙΑ ΚΑΝΟΝΩΝ ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ Pima Indians Diabetes			
Αλγόριθμος	*Μέση Ακρίβεια*	*Μέγιστη ακρίβεια*	*Κανόνες*
GEP	69,70%	74,90%	5
AIS+GEP	67,19%	76,47%	2

Πίνακας 5.5: Σύγκριση ακριβείας κανόνων για το πρόβλημα Pima Indians Diabetes.

Οι κανόνες που παρήγαγε ο προτεινόμενος αλγόριθμος φαίνονται στην συνέχεια.

Pima Indians Diabetes

```
IF ( SQRT(-(7,IF(a7,+(/(2,*(+(IF(SQRT(a1),+(+(7,1),
    IF(a5,2,3)),a1),a4),/(IF(SQRT(-(a4,a7)),7,1),
    IF(a6,*(+(1,a4),a8),IF(a8,a7,2))))),a1),
    SQRT(a6)))) > 0 ) THEN
    Class0
ELSEIF ( -(-(-(*(a1,a6),IF(+(-(a7,a7),7),IF(a4,
    /(7,+(+(a6,a7),-(3,a4))),+(5,7)),a1)),+(a2,
    /(1,/(/(a7,3),a8)))),5)] > 0) THEN
    Class1
ELSE
    Class0
ENDIF
```

5.6 Συμπεράσματα και μελλοντικές επεκτάσεις

Ολοκληρώνοντας στο σημείο αυτό την ανάλυση τόσο του συνδυασμού του αλγορίθμου επιλογής των κλώνων με τον ΠΓΕ, όσο και γενικότερα την συζήτηση γύρω από τα Τεχνητά Ανοσοποιητικά Συστήματα, έχουν γίνει πλέον φανερές οι δυνατότητες που μπορεί να παρέχουν οι ιδέες και οι αρχές του Ανοσοποιητικού Συστήματος στην μάθηση των μηχανών. Οι απαιτήσεις που υπάρχουν από το ανοσοποιητικό σύστημα είναι πραγματικά πολύ μεγάλες, πράγμα που έχει οδηγήσει στην ανάπτυξη ιδιαίτερα πολύπλοκων και συγχρόνως άκρως αποδοτικών τεχνικών για την αντιμετώπιση πάσης φύσεως ασθενειών. Η ταχύτητα απόκρισης, η ακρίβεια, η γενίκευση, αποτελούν λίγες μόνο από τις απαιτήσεις για την επιτυχή αντιμετώπιση ενός ξένου οργανισμού· και σε όλες αυτές το ανοσοποιητικό σύστημα ανταποκρίνεται με εξαιρετική επιτυχία.

Με την ανάλυση που προηγήθηκε, έγινε μία προσπάθεια επίδειξης των δυνατοτήτων που μπορεί να προσφέρει η μοντελοποίηση του ανοσοποιητικού συστήματος στην μάθηση των μηχανών. Πράγματι, τόσο σε προβλήματα σχετικά απλά, όπως η αναγνώριση ψηφιακών χαρακτήρων που παρουσιάστηκε στο Κεφάλαιο 3, όσο και σε αρκετά πιο σύνθετα όπως είναι η εξόρυξη γνώσης από δεδομένα, που συζητήθηκε στο Κεφάλαιο αυτό, τα ΤΑΣ επέδειξαν μία αξιόλογη συμπεριφορά. Μάλιστα δεν θα ήταν υπερβολή να πει κανείς, ότι οι διαφορές στην ταχύτητα σύγκλισης, χωρίς κάποια σημαντική απώλεια στην ακρίβεια, θέτουν νέα δεδομένα στην ΕΔ με εξελικτικούς αλγορίθμους. Επιπλέον, ο μηχανισμός της κλωνοποίησης δείχνει να συμπληρώνει ιδανικά την υπερ-μετάλλαξη, που είναι κατά κύριο λόγο

υπεύθυνη για την μεγάλη ταχύτητα σύγκλισης, επιτρέποντας στον αλγόριθμο να διατηρεί μόνο ένα πολύ μικρό σύνολο πληθυσμού. Ο πληθυσμός των αντισωμάτων αυξάνεται μόνο κατά την εμφάνιση ενός αντιγόνου, οπότε και δημιουργείται η απαραίτητη διαφορετικότητα, ενώ καθ΄ όλη την υπόλοιπη διάρκεια εκτέλεσης παραμένει σε πολύ χαμηλά επίπεδα, διευκολύνοντας με αυτό τον τρόπο τον χειρισμό του. Το γεγονός αυτό έχει άμεσο αντίκτυπο στους υπολογιστικούς πόρους που τελικά χρησιμοποιεί ο προτεινόμενος αλγόριθμος, αλλά και στην ταχύτητά του. Θα μπορούσε να πει κανείς, ότι παρέχει ένα πολύ καλό συμβιβασμό μεταξύ ακριβείας και ταχύτητας, διατηρώντας όλα τα καλά χαρακτηριστικά του ΠΓΕ, όπως είναι η αυξημένη εκφραστικότητα και ευελιξία των χρωμοσωμάτων, η διαφορά γονοτύπου-φαινοτύπου, η καλή κάλυψη του χώρου λύσεων, προσθέτοντας σε αυτά το χαρακτηριστικό της ταχύτητας.

Παρ΄ όλα αυτά, μειονεκτήματα εξακολουθούν να υπάρχουν, τα οποία και θα κατευθύνουν την μελλοντική έρευνα γύρω από το νέο πεδίο της ΕΔ μέσω αλγορίθμων του ανοσοποιητικού συστήματος. Το πρόβλημα της υπερβολικής προσαρμογής στα δεδομένα, παρά την χρήση του κριτηρίου MDL, εξακολουθεί να υπάρχει, όπως μπορεί να παρατηρήσει κανείς από τις μέτριες, αν και εφάμιλλες του απλού ΠΓΕ, τιμές ακριβείας του συνόλου κανόνων στο πρόβλημα Pima Indians Diabetes. Επομένως, ένα πεδίο έρευνας θα είναι η ανάπτυξη νέων ή η βελτίωση υπαρχουσών μεθόδων αποφυγής της υπερβολικής προσαρμογής.

Δεύτερο πεδίο που χρήζει περαιτέρω μελέτης, είναι η ανάπτυξη μεθόδων βασισμένες στις αρχές του ανοσοποιητικού συστήματος, για την ακόμα καλύτερη κάλυψη του χώρου των λύσεων. Στην παρούσα εργασία χρησιμοποιήθηκε ένα απλό είδος ανασύνθεσης πολλών σημείων και πολλών γονέων για την υλοποίηση της ανασύνθεσης V(D)J. Τελικά, όμως, ίσως να μην είναι αρκετό για την πλήρη κάλυψη του χώρου λύσεων, όπως μπορεί κανείς να παρατηρήσει από την ελαφρώς υποδέεστερη συμπεριφορά στο πρόβλημα MONK M_2. Θα μπορούσε να χρησιμοποιηθεί ένα διαφορετικό σχήμα επιλογής των γονέων βάσει ενός μέτρου σχετικής ποιότητας σύνδεσης (Bersini, 2002).

Τέλος, σε συνδυασμό με όσα αναφέρθηκαν προηγούμενως, ερευνητικό ενδιαφέρον παρουσιάζει και η αξιολόγηση των αποτελεσμάτων στην περίπτωση που χρησιμοποιούνται αντισώματα τύπου-ΠΓΕ με περισσότερα του ενός γονίδια.

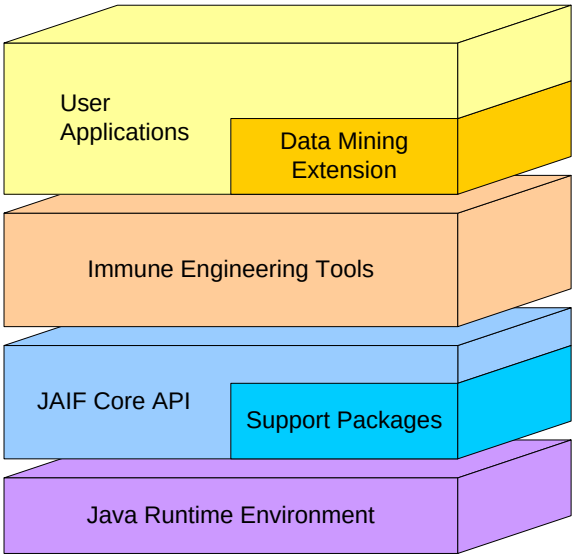
ΠΑΡΑΡΤΗΜΑ Α

Java Artificial Immune Framework

Παράλληλα με την ανάπτυξη των αλγορίθμων που περιγράφηκαν στα προηγούμενα κεφάλαια, αναπτύχθηκε και μία υποδομή λογισμικού για την υποστήριξή τους. Σκοπός της υποδομής αυτής, που ονομάστηκε Java Artificial Immune Framework (JAIF), πέρα από την υποστήριξη των αλγορίθμων, είναι η παροχή ενός συνόλου προγραμματιστικών εργαλείων, για την δημιουργία και ανάπτυξη νέων αλγορίθμων βασισμένων στο ανοσοποιητικό σύστημα. Κατά την διάρκεια της σχεδίασης, δόθηκε ιδιαίτερη προσοχή στις δυνατότητες εύκολης και γρήγορης επέκτασης της λειτουργικότητας της υποδομής, ενώ παράλληλα επιδιώχθηκε ένας σαφής διαχωρισμός των λειτουργιών κάθε τμήματος. Η δομή του JAIF, επομένως, είναι μία πολυεπίπεδη δομή, ορίζοντας σαφείς διεπιφάνειες (interfaces) μεταξύ των επιπέδων. Με αυτό τον τρόπο καθίσταται δυνατή η εύκολη συντήρηση και επέκταση της λειτουργικότητας ενός συγκεκριμένου επιπέδου, χωρίς να επηρεάζονται τα άλλα επίπεδα της υποδομής.

Στις επόμενες παραγράφους του Παραρτήματος αυτού, παρουσιάζονται εν συντομία οι βασικές λειτουργίες και δυνατότητες που προσφέρει το JAIF, αποκαλύπτοντας ταυτόχρονα την ευελιξία της δομής του. Σκοπός αυτού του Παραρτήματος δεν είναι να παράσχει μία λεπτομερή περιγραφή κάθε κλάσης ή μεθόδου του JAIF, κάτι για το οποίο είναι υπεύθυνη η τεκμηρίωση (documentation) που ακολουθεί το πρόγραμμα, αλλά να παρουσιάσει στον χρήστη τον τρόπο, με τον οποίο τα διάφορα μέρη της υποδομής συνδυάζονται, με σκοπό την δημιουργία συνθέτων δομών και αλγορίθμων ανοσολογικής μηχανικής, όπως αυτοί που παρουσιάστηκαν στα προηγούμενα κεφάλαια. Επιπλέον λειτουργεί ως κατευθυντήριος γραμμή για τον προγραμματιστή, που θα αποφασίσει να «κτίσει» κάποια εφαρμογή πάνω από το JAIF, αλλά και για εκείνον που σκοπεύει να το εμπλουτίσει με περαιτέρω λειτουργικότητα.

Τελειώνοντας το εισαγωγικό σημείωμα, θα πρέπει να αναφερθεί ότι το JAIF, όπως γίνεται προφανές από το όνομά του, αναπτύχθηκε στην γλώσσα προγραμματισμού Java, και συγκεκριμένα κάνοντας χρήση της έκδοσης 1.5 του *Συνόλου Εργαλείων Ανάπτυξης Εφαρμογών Java (Java Development Kit–JDK)*, η οποία παρέχει στον προγραμματιστή μία πληθώρα προχωρημένων χαρακτηριστικών, όπως είναι οι *γενικευμένοι τύποι δεδομένων (generic data types)*, μέθοδοι μεταβλητού αριθμού ορισμάτων (variable argument methods), κ.α. Ο αναγνώστης θα πρέπει να είναι σχετικά εξοικειωμένος με τα νέα χαρακτηριστικά της γλώσσας και τις δυνατότητες



Σχήμα A.1: Η δομή του Java Artificial Immune Framework.

που προσφέρουν, έτσι ώστε να μπορέσει να κατανοήσει απρόσκοπτα κάποια στοιχεία της διεπιφανείας προγραμματισμού (API) που παρέχει το JAIF.

A.1 Η δομή του JAIF

Η δομή του Java Artificial Immune Framework αποτελείται από 3 λειτουργικά επίπεδα (Σχήμα A.1). Το χαμηλότερο επίπεδο (Core API) αποτελεί την κύρια μονάδα του JAIF, παρέχοντας ένα σύνολο βασικών δομών και λειτουργιών, οι οποίες χρησιμοποιούνται από τα ανώτερα στρώματα, με σκοπό την δημιουργία πιο συνθέτων και πιο λειτουργικών στοιχείων. Το Core API του JAIF υλοποιείται στο πακέτο `jaif`. Οι λειτουργίες του επιπέδου αυτού μπορεί να πλαισιώνονται από πακέτα υποστήριξης (support packages), τα οποία παρέχουν επεκτάσεις στο βασικό API. Στην τρέχουσα έκδοση του JAIF υπάρχει ένα πακέτο υποστήριξης, το `jaif.gep`, το οποίο δίνει την δυνατότητα να υποστηριχθεί ο ΠΓΕ.

Στο αμέσως ανώτερο επίπεδο παρέχονται συγκεκριμένα εργαλεία ανοσολογικής μηχανικής, όπως είναι ο αλγόριθμος επιλογής κλώνων, τα οποία συνδυάζουν τα στοιχεία του Core API, προσφέροντας λειτουργίες και δυνατότητες, οι οποίες μπορούν χρησιμοποιηθούν άμεσα από μια εφαρμογή χρήστη (user application).

Στην τρέχουσα έκδοση του JAIF, το τελευταίο επίπεδο της υποδομής είναι το επίπεδο χρήστη. Στο επίπεδο αυτό ανήκουν οι εφαρμογές που χρησιμοποιούν την υποδομή του JAIF, για να εκτελέσουν μία εργασία βασισμένη σε τεχνητά ανοσοποιητικά συστήματα. Παράδειγμα τέτοιας εφαρμογής είναι και η επέκταση του JAIF για την υποστήριξη της εξόρυξης από δεδομένα (JAIF Data Mining Extension), η οποία στηρίζεται εξ΄ ολοκλήρου στα εργαλεία που παρέχουν τα κατώτερα επίπεδα.

A.2 Η βασική προγραμματιστική διεπιφάνεια

Η βασική προγραμματιστική διεπιφάνεια του JAIF (Core API) ορίζει ένα σύνολο από διεπιφάνειες (interfaces) και κλάσεις για την περιγραφή των δομικών στοιχείων μίας εφαρμογής βασισμένης σε ανοσοποιητικά συστήματα. Σκοπός του Core API είναι

να παράσχει μία όσο το δυνατόν γενική και ευέλικτη δομή, η οποία να επιτρέπει αφενός την εύκολη επέκταση της λειτουργικότητάς, και αφετέρου να δίνει στον χρήστη την δυνατότητα να ορίζει δικές του υλοποιήσεις των βασικών δομών. Για τον λόγο αυτό οι περισσότερες κλάσεις στο Core API είναι αφηρημένες (abstract classes), καθορίζοντας απλά τις λειτουργίες κάθε μονάδας. Βέβαια, για λόγους ευκολίας του τελικού χρήστη, παρέχονται κάποιες συγκεκριμένες υλοποιήσεις κοινά χρησιμοποιουμένων δομών, οι οποίες, όμως, δεν είναι δεσμευτικές, και ο καθένας μπορεί να τις παρακάμψει.

Οι πρώτες βασικές δομές που υλοποιεί το Core API είναι τα αντισώματα και τα αντιγόνα, τα οποία ορίζονται ως αφηρημένες κλάσεις. Σύμφωνα με την συζήτηση στο Κεφάλαιο 3, θα μπορούσε κανείς να ενοποιήσει τα αντισώματα και τα αντιγόνα υπό την στέγη μίας γενικότερης δομής ακολουθίας συμβόλων. Αν και αυτό από μαθηματικής απόψεως είναι εφικτό και αποδεκτό, τα αντισώματα και τα αντιγόνα παρουσιάζουν στην πράξη διαφορετική λειτουργικότητα· για παράδειγμα τα αντισώματα θα πρέπει να είναι τέτοιες δομές, ώστε να μπορούν να μεταλλαχθούν, θα πρέπει επίσης να ανατίθεται σε αυτές μία τιμή ενδεικτική της ποιότητας σύνδεσής τους με κάποιο αντιγόνο, κ.ο.κ. Αντιθέτως, τα αντιγόνα δεν είναι τίποτε άλλο από απλές γραμμικές δομές, οι οποίες θα πρέπει να αναγνωριστούν από συγκεκριμένα αντισώματα. Επιπλέον, μία θεώρηση των αντιγόνων ως κοινών ακολουθιών συμβόλων, θα έθετε εμμέσως ένα περιορισμό στην δυνατότητά τους να αναπαραστήσουν και γενικευμένα πρότυπα (βλ. αντιγόνα κλάσεων δεδομένων). Για τον λόγο αυτό κρίθηκε σκόπιμη η δημιουργία δύο διαφορετικών κλάσεων για τα αντισώματα και τα αντιγόνα.

A.2.1 Αντισώματα

Η λειτουργικότητα των αντισωμάτων του JAIF οριοθετείται από την αφηρημένη κλάση `Antibody`, η οποία μοντελοποιεί ένα αφηρημένο γραμμικό αντίσωμα. Δύο στοιχεία που είναι κοινά σε όλα τα αντισώματα είναι το μήκος τους και η ποιότητα σύνδεσής τους με κάποιο αντιγόνο. Για τον λόγο αυτό ορίζονται δύο προστατευμένα (protected) πεδία, τα `length` και `affinity`, έτσι ώστε να είναι άμεσα προσβάσιμα από κάθε συγκεκριμένη υποκλάση. Επιπλέον, ορίζονται οι μέθοδοι `getLength()`, `getAffinity()` και `setAffinity()`, οι οποίες απλά επιστρέφουν ή θέτουν την αντίστοιχη τιμή. Το μήκος του αντισώματος μπορεί να καθοριστεί μόνο μέσω του κατασκευαστή της κλάσης, ο οποίος απλά θέτει `this.length = `*`length`*.

```
import jaif.Antibody;

public Antibody(int length);

public int      getLength();
public double   getAffinity();
public void     setAffinity(double aff);
```

Αυτό που θα πρέπει να προσέξει κανείς, είναι ότι η ποιότητα σύνδεσης αποτελεί μία ιδιότητα της κλάσης `Antibody`, η οποία τίθεται ρητά από τον χρήστη. Επομένως, όταν υλοποιείται κάποιος αλγόριθμος, που κάνει χρήση της δομής των αντισωμάτων, θα πρέπει επίσης να μεριμνά, ώστε η τιμή της ποιότητας σύνδεσης που αποθηκεύεται κάθε στιγμή στο αντίσωμα να είναι η επιθυμητή.

Ένα αντίσωμα θα πρέπει να έχει την ικανότητα να μεταλλάσσεται, πράγμα για το οποίο φροντίζουν οι δύο μέθοδοι `mutate()` που παρέχονται.

```
import jaif.Antibody;

public abstract Antibody    mutate(int pos);
public abstract Antibody    mutate(double rate);
```

Η πρώτη μεταλλάσσει το σύμβολο που υπάρχει στην θέση *pos* του αντισώματος, ενώ η δεύτερη μεταλλάσσει ολόκληρο το αντίσωμα με ρυθμό *rate*. Το πώς ακριβώς θα γίνεται η μετάλλαξη, αλλά και το πώς θα αντιμετωπίζονται πιθανώς λανθασμένες τιμές για τις παραμέτρους *pos* και *rate*, αφήνεται στην υλοποίηση. Συνήθως, σε περιπτώσεις λάθους εγείρονται οι εξαιρέσεις `IndexOutOfBoundsException` και `IllegalArgumentException`, αντιστοίχως. Αυτή η τακτική ακολουθείται και από τις συγκεριμένες υλοποιήσεις αντισωμάτων που παρέχονται από το JAIF. Τέλος, και οι δύο μέθοδοι επιστρέφουν το μεταλλαγμένο αντίσωμα.

Η κλάση `Antibody` υλοποιεί, επιπλέον, τις διεπιφάνειες `Cloneable` και `Comparable<Antibody>`, έτσι ώστε αφενός να δίνεται η δυνατότητα στα αντισώματα να κλωνοποιούνται, και αφετέρου να μπορούν να συγκριθούν μεταξύ τους. Η υλοποίηση της μεθόδου `clone()` που παρέχει η κλάση `Antibody`, καλεί απλά την αντίστοιχη μέθοδο της κλάσης `Object` και επομένως, οι υποκλάσεις θα πρέπει ενδεχομένως να ορίσουν ξανά την μέθοδο αυτή, εάν επιθυμείται μία αντιγραφή σε βάθος (deep copy) του αντισώματος. Όσον αφορά στην μέθοδο `compareTo()`, η σύγκριση γίνεται βάσει της ποιότητας σύνδεσης των αντισωμάτων που συγκρίνονται. Για την επιστρεφομένη τιμή ακολουθείται η σύμβαση που ορίζεται από την τεκμηρίωση της διεπιφανείας `Comparable<T>`. Έτσι, εάν η ποιότητα σύνδεσης αυτού του αντισώματος είναι μικρότερη, τότε επιστρέφεται -1, εάν είναι μεγαλύτερη επιστρέφεται 1, και στην περίπτωση που υπάρχει ισότητα επιστρέφεται 0.

Τέλος, για την υποστήριξη της ανασύνθεσης V(D)J, όπως αυτή ορίστηκε στην §5.2, ορίζεται μία ακόμα μέθοδος, η οποία μπορεί να προσπελαύνει ένα μόνο τμήμα του αντισώματος. Σε συμφωνία με τις αρχές του ΠΓΕ, το τμήμα αυτό ονομάζεται *ακολουθία κωδικονίων (codon sequence)*.

```
import jaif.*;

public abstract CodonSequence
getCodonSequence(int start, int len);
```

Η συνάρτηση `getCodonSequence()` θα πρέπει να επιστρέφει την ακολουθία συμβόλων που ξεκινά από την θέση *start* και τελειώνει στην θέση *start+len−1*. Ο χειρισμός των περιπτώσεων λάθους αφήνεται, και σε αυτή την περίπτωση, στις συγκεκριμένες υλοποιήσεις. Η μέθοδος αυτή επιστρέφει ένα αντικείμενο που υλοποιεί την διεπιφάνεια `CodonSequence`, έτσι ώστε να υπάρχει απόλυτη ελευθερία στην εσωτερική αναπαράσταση των αντισωμάτων. Η διεπιφάνεια `CodonSequence` ορίζει τρεις μεθόδους.

```
import jaif.CodonSequence;

public String   getCodon(int pos);
public String[] getCodons(int start, int len);
public int      getLength();
```

Οι μέθοδοι `getCodon()` και `getCodons()` επιστρέφουν το σύμβολο ή τα σύμβολα που υπάρχουν σε μία συγκεκριμένη θέση, ή σε μία σειρά διαδοχικών συμβόλων της ακολουθίας κωδικονίων. Στο Java Artificial Immune Framework για να υπάρχει μεγάλη ευελιξία στην επιλογή των συμβόλων, τα σύμβολα αναπαρίστανται ως `String`s και επομένως ο όρος σύμβολο χρησιμοποιείται καταχρηστικά. Ένα μαθηματικό σύμβολο αναπαρίσταται ως μία ακολουθία συμβόλων στο JAIF, η οποία όμως χρησιμοποιείται ως μία μοναδική οντότητα. Για παράδειγμα, εάν το μαθηματικό σύμβολο Q αναπαριστά την συνάρτηση της τετραγωνικής ρίζας, τότε αυτό μπορεί να αναπαρίσταται στο JAIF από την ακολουθία συμβόλων `SQRT`, η οποία όμως θα χρησιμοποιείται πάντα σαν μία οντότητα, όπως ακριβώς συμβαίνει και με το μαθηματικό σύμβολο Q. Τέλος, η συνάρτηση `getLength()` επιστρέφει το μήκος της ακολουθίας κωδικονίων, που αντιπροσωπεύει το συγκεκριμένο αντικείμενο `CodonSequence`.

Πέρα από το γεγονός ότι η διεπιφάνεια `CodonSequence` αφήνει τελείως ελεύθερη την επιλογή της εσωτερικής αναπαράστασης των αντισωμάτων, αναλαμβάνει επίσης την αρμοδιότητα της μετατροπής της εσωτερικής αναπαράστασης των αντισωμάτων σε ακολουθίες συμβόλων του JAIF. Σε αντίθετη περίπτωση, τα αντισώματα θα επιφορτίζονταν και με αυτή την εργασία, καθιστώντας πιο σύνθετη την υλοποίησή τους.

Συγκεκριμένες υλοποιήσεις της κλάσης `Antibody`

Στο Java Artificial Immune Framework παρέχονται τρεις συγκεκριμένες υλοποιήσεις της κλάσης `Antibody`, οι οποίες μπορούν να χρησιμοποιηθούν σε διαφορετικά προβλήματα. Συγκεκριμένα παρέχονται δυαδικά (`BinaryAntibody`), διανυσματικά (`VectorAntibody`) και τύπου-ΓΓΕ αντισώματα (`GEPAntibody`).

Ένα `BinaryAntibody` υλοποιείται ως μία ακολουθία από bits, χρησιμοποιώντας την κλάση `java.util.BitSet`, και μπορεί να κατασκευαστεί με δύο τρόπους.

```
import jaif.BinaryAntibody;
import java.util.BitSet;

public BinaryAntibody(int length);
public BinaryAntibody(int length, BitSet initGene);
```

Στην πρώτη περίπτωση κατασκευάζεται ένα αντίσωμα μήκους *`length`*, το οποίο παράλληλα αρχικοποιείται ως μία τυχαία ακολουθία 0 και 1. Στην δεύτερη περίπτωση, η αρχική ακολουθία bits του αντισώματος λαμβάνεται από το `BitSet` *`initGene`*. Η ακολουθία *`initGene`* δεν είναι υποχρεωτικό να έχει μήκος *`length`*· σε περίπτωση που είναι μεγαλύτερη, θα χρησιμοποιηθούν μόνο τα πρώτα *`length`* bits της, ενώ στην αντίθετη περίπτωση, τα υπόλοιπα bits του δυαδικού αντισώματος θα συμπληρωθούν με 0.

Οι μέθοδοι μετάλλαξης της υπερκλάσης `Antibody` υλοποιούνται αντιστρέφοντας τα bits στις θέσεις που γίνεται η μετάλλαξη. Πέρα από τις δύο βασικές μεθόδους μετάλλαξης, η κλάση `BinaryAntibody` ορίζει δύο ακόμα μεθόδους μετάλλαξης.

```
import jaif.BinaryAntibody;

public BinaryAntibody mutate(int[] pos);
public BinaryAntibody mutate(int start, int end);
```

Η πρώτη μέθοδος αντιστρέφει τα bits στις θέσεις που ορίζει ο πίνακας *pos*, ενώ η δεύτερη αντιστρέφει τα bits μεταξύ των θέσεων *start* και *end*, καλώντας απλά `flip(start, end)` στο `BitSet` που υλοποιεί το αντίσωμα. Και στις δύο περιπτώσεις επιστρέφεται το μεταλλαγμένο αντίσωμα. Τέλος, η κλάση `BinaryAntibody` ορίζει ξανά την συνάρτηση `clone()`, έτσι ώστε να επιστρέφει μία εις βάθος αντιγραφή του αντισώματος.

Ένα διανυσματικό αντίσωμα, `VectorAntibody`, αποτελεί στην ουσία ένα διάνυσμα στον χώρο $\mathbb{R}^n$. Κατά την κατασκευή του, μπορεί και αυτό είτε να αρχικοποιηθεί τυχαία, είτε ρητά.

```
import jaif.VectorAntibody;

public VectorAntibody(int length);
public VectorAntibody(double[] initGene);
```

Ο πρώτος κατασκευαστής δημιουργεί ένα διάνυσμα διαστάσεων `length` και το αρχικοποιεί τυχαία με τιμές στο διάστημα $[0, 1]$. Στην δεύτερη περίπτωση το αντίσωμα αρχικοποιείται βάσει του διανύσματος *initGene*, το οποίο καθορίζει τόσο τις αρχικές τιμές του νέου αντισώματος, όσο και το μήκος του. Κατά τα άλλα, η κλάση `VectorAntibody` δεν υλοποιεί κάποια καινούργια μέθοδο σε σχέση με αυτές που ορίζονται στην υπερκλάση `Antibody`, ενώ παράλληλα παρέχει μία αντιγραφή σε βάθος μέσω του επαναπροσδιορισμού της μεθόδου `clone()`.

Κλείνοντας την παράγραφο για τα δυαδικά και διανυσματικά αντισώματα, να σημειωθεί ότι στην τρέχουσα έκδοση του JAIF, τα αντισώματα αυτά δεν υποστηρίζουν ακόμα την μέθοδο `getCodonSequence()`, με αποτέλεσμα να εγείρουν μία `UnsupportedOperationException`, σε περίπτωση που η μέθοδος αυτή κληθεί σε κάποιο από αυτά τα αντισώματα.

Δημιουργία αντισωμάτων ανεξαρτήτως του τύπου τους

Κατά την κατασκευή προγραμματιστικών εργαλείων ανοσολογικής μηχανικής προκύπτει συχνά η περίπτωση, να είναι αναγκαία η δημιουργία αντισωμάτων χωρίς να είναι γνωστός εκ των προτέρων ο τύπος τους. Επειδή, όπως είναι αναμενόμενο, ο κατασκευαστής και οι μέθοδοι της υπερκλάσης `Antibody` δεν μπορούν να καλύψουν όλες τις περιπτώσεις δημιουργίας νέων αντισωμάτων, καθώς πρέπει να είναι αρκετά γενικές, ακολουθείται το σχήμα του *εργοστασίου αντικειμένων (object factory)*, έτσι ώστε να είναι εφικτή η δημιουργία ενός αντισώματος οποιουδήποτε τύπου. Έτσι δημιουργείται η διεπιφάνεια `AntibodyFactory`, η οποία μπορεί να κατασκευάζει αντισώματα τύπου `T`. Η διεπιφάνεια αυτή ορίζει τρεις μεθόδους.

```
import jaif.*;

public interface AntibodyFactory<T extends Antibody>

T           newAntibody();
T           newAntibody(CodonSequence[] fragments);
Class<T>    getAntibodyType();
```

Για κάθε τύπο αντισώματος δημιουργείται και ένα αντίστοιχο `AntibodyFactory`, το οποίο υλοποιεί αυτές τις μεθόδους. Οι δύο πρώτες αναλαμβάνουν την κατασκευή ενός αντισώματος, με την δεύτερη να κατασκευάζει το αντίσωμα από ένα σύνολο ακολουθιών κωδικονίων. Οι παράμετροι του αντισώματος τίθενται

συνήθως με μεθόδους που παρέχει η υλοποίηση του `AntibodyFactory` για τον συγκεκριμένο τύπο αντισώματος. Επομένως, εάν ένα στοιχείο προγράμματος (component) επιθυμεί να κατασκευάζει αντισώματα ανεξαρτήτως τύπου, θα καλεί την μέθοδο `newAntibody()` ενός `AntibodyFactory`, χωρίς να ασχολείται ούτε με την ακριβή υλοποίηση της κλάσης ούτε με τον ακριβή τύπο του αντισώματος.

Αν και γενικώς μπορεί να είναι επιθυμητό από κάποιο στοιχείο προγράμματος εργαλείου, να κατασκευάζονται αντικείμενα οποιουδήποτε τύπου, χωρίς να είναι γνωστός ο ακριβής τύπος του αντικειμένου, εντούτοις μερικές φορές ο τύπος του αντικειμένου είναι απαραίτητος. Μία σχετικά συνήθης τέτοια περίπτωση προκύπτει, όταν επιθυμείται η δυναμική δημιουργία ενός πίνακα χωρίς να είναι γνωστός ο ακριβής τύπος των αντικειμένων του κατά την διαδικασία της μεταγλώττισης. Σε αυτές της περιπτώσεις ο πίνακας του επιθυμητού τύπου δημιουργείται με την βοήθεια του *μηχανισμού του αντικατοπτρισμού (reflection mechanism)* που παρέχει η γλώσσα Java, αρκεί να είναι γνωστός ο ακριβής τύπος των αντικειμένων κατά τον χρόνο εκτέλεσης. Γενικά, ο μηχανισμός του αντικατοπτρισμού σε συνδυασμό με τους γενικευμένους τύπους δεδομένων μπορεί να δημιουργήσει ιδιαίτερα ασφαλή προγράμματα, καθ΄ ότι επιτρέπει να γίνεται αυστηρότερος έλεγχος τύπων (type checking) κατά την φάσης της μεταγλώττισης, αποφεύγοντας με τον τρόπο αυτό σφάλματα τύπων κατά τον χρόνο εκτέλεσης (Bracha, 2004). Ο μηχανισμός του αντικατοπτρισμού χρησιμοποιείται στην κλάση `AntibodyPool` για την δημιουργία πινάκων με συγκεκριμένο τύπο αντισωμάτων. Για τους λόγους αυτούς, η διεπιφάνεια `AntibodyFactory` ορίζει επιπλέον την μέθοδο `getAntibodyType()`, η οποία επιστρέφει τον τύπο του αντισώματος με την μορφή ενός αντικειμένου `Class<T>`.

Ως παράδειγμα υλοποίησης της διεπιφάνειας `AntibodyFactory`, στο Πρόγραμμα Α.1 φαίνεται η υλοποίηση που παρέχεται στο JAIF για την δημιουργία δυαδικών αντισωμάτων. Αυτό που αξίζει να παρατηρήσει κανείς στο παράδειγμα αυτό, είναι ο ορισμός της μεθόδου `setGeneSize()`, η οποία θέτει το μέγεθος των αντισωμάτων που θα δημιουργούνται μέσω της `newAntibody()`. Παράλληλα ορίζεται και μία προκαθορισμένη τιμή για το μέγεθος των νέων αντισωμάτων (`DEF_ABSIZE`), σε περίπτωση που δεν οριστεί κάποια ρητά. Έτσι, κάθε νέο αντίσωμα θα έχει μήκος `geneSize`.

```
package jaif;

public class BinaryAntibodyFactory
  implements AntibodyFactory<BinaryAntibody> {

  private static final int  DEF_ABSIZE = 50;
  private int               geneSize;

  private BinaryAntibodyFactory() {

    geneSize = DEF_ABSIZE;
  }

  public static BinaryAntibodyFactory getInstance() {

    return( new BinaryAntibodyFactory() );
  }

  public BinaryAntibody newAntibody() {
```

```
    return( new BinaryAntibody(geneSize) );
  }

  public BinaryAntibody
    newAntibody(CodonSequence[] fragments) {

    // not yet implemented
    throw new UnsupportedOperationException();
  }

  public Class<BinaryAntibody> getAntibodyType() {

    return BinaryAntibody.class;
  }

  public void setGeneSize(int size) {

    geneSize = size;
  }

  public int getGeneSize() { return geneSize; }
}
```

Πρόγραμμα Α.1: Η υλοποίηση της διεπιφάνειας `AntibodyFactory` για την δημιουργία δυαδικών αντισωμάτων.

Α.2.2 Αντιγόνα

Οι λειτουργικές απαιτήσεις από ένα αντιγόνο στα πλαίσια του Java Artificial Immune Framework είναι πολύ μικρότερες σε σχέση με τις απαιτήσεις από ένα αντίσωμα. Έτσι, η υπερκλάση όλων των αντιγόνων, `Antigen`, ορίζει μόνο μία καινούργια συνάρτηση και ένα κατασκευαστή για να επιστρέφεται και να τίθεται, αντιστοίχως, το μήκος του αντιγόνου.

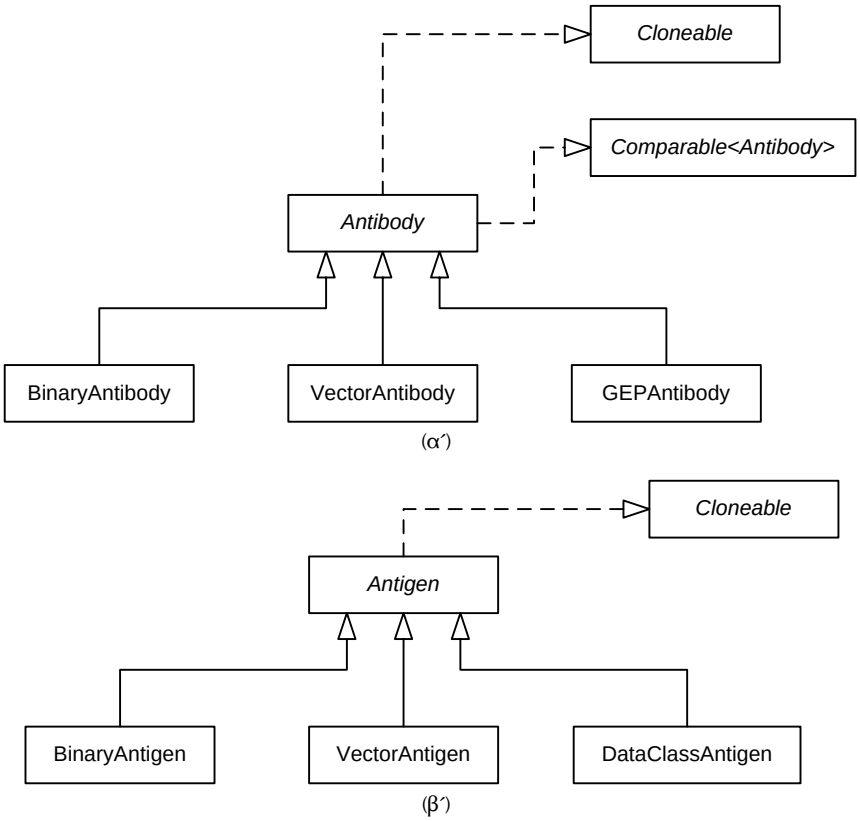
```
import jaif.Antigen;

public Antigen(int length);

public int  getLength();
```

Όπως φαίνεται από το πρωτότυπο της μεθόδου `getLength()`, καμία δεν είναι αφηρημένη· παρ΄ όλα αυτά η κλάση `Antigen` δηλώνεται ως αφηρημένη. Ο λόγος που γίνεται αυτό, είναι ότι θα πρέπει να αποφεύγεται ρητά η δημιουργία αντικειμένων τύπου `Antigen`, καθ΄ ότι δεν παρέχεται μία συγκεκριμένη υλοποίηση ενός αντιγόνου, και επομένως η έννοια του αντιγόνου όπως την αναπαριστά η κλάση `Antigen` είναι αφηρημένη.

Συγκεκριμένες υλοποίησεις της κλάσης `Antigen`, παρέχονται από τις κλάσεις `BinaryAntigen`, `VectorAntigen` και `DataClassAntigen`. Η πρώτη αντιπροσωπεύει ένα αντιγόνο που αναπαρίσταται ως μία ακολουθία δυαδικών ψηφίων, η δεύτερη ένα αντιγόνο που αναπαρίσταται ως ένα n-διάστατο διάνυσμα, ενώ η τρίτη παρέχεται από το πακέτο `jaif.gep` και αντιπροσωπεύει ένα Αντιγόνο



Σχήμα Α.2: Η ιεραχία κλάσεων των αντισωμάτων και των αντιγόνων.

Κλάσης Δεδομένων (ΑΚΔ). Γενικά, τα `BinaryAntigen` χρησιμοποιούνται παράλληλα με τα `BinaryAntibody`, ενώ ανάλογη αντιστοιχία υπάρχει και μεταξύ των `VectorAntigen` και `VectorAntibody`.

Η ιεραρχία των κλάσεων των αντισωμάτων και των αντιγόνων φαίνεται στο Σχήμα Α.2.

Α.2.3 Ποιότητα σύνδεσης και εκτιμητές παραμέτρων

Η συμπεριφορά και η πορεία εκτέλεσης ενός εξελικτικού αλγορίθμου πολύ συχνά εξαρτώνται από το αποτέλεσμα μιας σειράς συναρτήσεων ή γενικότερα υπολογιστικών διαδικασιών, που εφαρμόζονται σε ολόκληρο τον πληθυσμό ή σε τμήματά του. Παράδειγμα τέτοιων διαδικασιών στον αλγόριθμο επιλογής κλώνων είναι ο υπολογισμός της συνάρτησης ποιότητας σύνδεσης αντισωμάτων-αντιγόνων, η διαδικασία υπολογισμού του ρυθμού μετάλλαξης, κ.α. Θα μπορούσε, επομένως, να πει κανείς ότι ολόκληρες αυτές οι υπολογιστικές διαδικασίες αποτελούν παραμέτρους του αλγορίθμου, οι οποίες θα πρέπει κάθε φορά να καθορίζονται από τον χρήστη.

Για να υλοποιηθούν αυτές οι ιδέες με ένα συνεπή και παράλληλα αντικειμενοστρεφή τρόπο, εισάγεται η έννοια του *εκτιμητή παραμέτρου (parameter evaluator)*. Ο εκτιμητής παραμέτρου είναι ένα αντικείμενο, σκοπός του οποίου είναι να εκτιμά-υπολογίζει μία παράμετρο ή μία ποσότητα του αλγορίθμου, όπως είναι ο ρυθμός μετάλλαξης ή η ποιότητα σύνδεσης, η οποία μπορεί να μεταβάλλεται κατά την διάρκεια της εκτέλεσης. Στο JAIF ορίζονται 3 εκτιμητές παραμέτρων:

`AffinityEvaluator`, `MutationEvaluator` `CloneEvaluator`. Σκοπός του τελευταίου είναι να υπολογίζει τον αριθμό των κλώνων που θα δώσει κάθε αντίσωμα κατά την φάση της κλωνοποίησης. Και οι τρεις εκτιμητές ορίζονται είτε ως αφηρημένες κλάσεις είτε ως διεπιφάνειες στο πακέτο `jaif.evaluators`, έτσι ώστε να δίνεται η μέγιστη ευελιξία στον χρήστη να παράσχει τους δικές του υλοποιήσεις.

Εκτιμητής ποιότητας σύνδεσης

Ένας εκτιμητής ποιότητας σύνδεσης θα πρέπει να κληρονομεί την κλάση `AffinityEvaluator`. Η κλάση αυτή θα πρέπει να συσχετίζει ένα αντίσωμα τύπου `T` και ένα αντιγόνο τύπου `S`, ενώ παράλληλα ορίζει δύο αφηρημένες μεθόδους.

```
import jaif.*;
import jaif.evaluators.AffinityEvaluator;

public abstract class
    AffinityEvaluator<T extends Antibody,
                      S extends Antigen>

public abstract double  computeAffinity(T ab);
public abstract double  computeAffinity(T ab, S ag);
```

Η πρώτη μέθοδος, θα μπορούσε να πει κανείς, ότι χρησιμοποιεί καταχρηστικά το συνθετικό *affinity*, καθ΄ ότι στην πραγματικότητα πρόκειται περισσότερο την προσαρμογή (fitness) ενός αντισώματος. Ο λόγος ύπαρξής της είναι για να επιτρέπει την δημιουργία εκτιμητών ποιότητας σύνδεσης και σε περιπτώσεις αλγορίθμων βελτιστοποίησης, οι οποίοι, γενικά, δεν χρησιμοποιούν πρότυπα (de Castro και Von Zuben, 2002). Η δεύτερη μέθοδος συγκρίνει ένα αντίσωμα τύπου `T` με ένα αντιγόνο τύπου `S` και επιστρέφει ένα μέτρο της ποιότητας σύνδεσης τους. Γενικά δεν είναι απαραίτητο κάποιος εκτιμητής ποιότητας σύνδεσης να υποστηρίζει και τις δύο μεθόδους με συνέπεια, αρκεί αυτό να ορίζεται σαφώς στην τεκμηρίωσή του. Για παράδειγμα ο εκτιμητής `HammingEvaluator`, που παρέχεται από το JAIF και υπολογίζει την αντίστροφη απόσταση Hamming μεταξύ ενός δυαδικού αντισώματος και ενός δυαδικού αντιγόνου, δεν υποστηρίζει την μέθοδο με το μοναδικό όρισμα, διότι κάτι τέτοιο δεν θα είχε νόημα.

Παράλληλα με αυτές τις δύο μεθόδους, παρέχονται και οι αντίστοιχες μέθοδοι υπολογισμού της ποιότητας σύνδεσης μίας σειράς αντισωμάτων.

```
import jaif.*;
import jaif.evaluators.AffinityEvaluator;

public double[] computeAffinity(T[] ab);
public double[] computeAffinity(T[] ab, S ag);
```

Οι μέθοδοι αυτοί καλούν επαναληπτικά της αντίστοιχες αφηρημένες μεθόδους και επιστρέφουν ένα πίνακα με τις ποιότητες σύνδεσης.

Μία ακόμα σημαντική παράμετρος που ορίζεται στην κλάση `AffinityEvaluator` είναι η προστατευμένη `boolean` μεταβλητή `autoUpdateAffinities`. Η παράμετρος αυτή θα πρέπει να λειτουργεί ως υπόδειξη για τις υποκλάσεις, έτσι ώστε να ενημερώνουν την ποιότητα σύνδεσης των αντισωμάτων, των οποίων την ποιότητα σύνδεσης υπολογίζουν. Έτσι, εάν η παράμετρος αυτή είναι `true`, οι εκτιμητές της

ποιότητας σύνδεσης θα πρέπει να καλούν *ab*.setAffinity(aff), όπου aff είναι η ποιότητα σύνδεσης του αντισώματος *ab*. Η προκαθορισμένη τιμή της παραμέτρου αυτής είναι false, ενώ για τον χειρισμό της παρέχονται οι μέθοδοι

```
import jaif.evaluators.AffinityEvaluator;

public boolean  isAutoUpdateAffinities();
public void     setAutoUpdateAffinities(boolean val);
```

των οποίων η λειτουργικότητα είναι εμφανής.

Εκτιμητές ρυθμού μετάλλαξης και αριθμού κλώνων

Το JAIF παρέχει δύο ακόμα διεπιφάνειες για τον υπολογισμό του ρυθμού μετάλλαξης ενός αντισώματος σε σχέση με την ποιότητα σύνδεσής του σε κάποιο αντιγόνο και για τον υπολογισμό του αριθμού των κλώνων που θα πρέπει να δώσει κάθε ένα αντίσωμα ενός συνόλου αντισωμάτων. Οι διεπαφάνειες αυτές είναι οι MutationEvaluator και CloneEvaluator, αντιστοίχως. Ένας εκτιμητής του ρυθμού μετάλλαξης των αντισωμάτων θα πρέπει να υλοποιεί τις δύο μεθόδους του αφηρημένου MutationEvaluator.

```
import jaif.evaluators.MutationEvaluator;

public double   computeMutationRate(double affinity);
public double[] computeMutationRate(double[] affinity);
```

Οι μέθοδοι θα πρέπει να υπολογίζουν τον ρυθμό μετάλλαξης ενός αντισώματος, δεδομένης της ποιότητας σύνδεσής του *affinity*.

Ένας εκτιμητής του αριθμού των κλώνων θα πρέπει να υλοποιεί τρεις μεθόδους.

```
import jaif.evaluators.CloneEvaluator;

public int[]    computeClones(double[] affinity);
public boolean  isSorted();
public boolean  setSorted(boolean val);
```

Η πρώτη από αυτές τις μεθόδους υπολογίζει τον αριθμό των κλώνων για κάθε ένα αντίσωμα του οποίου η ποιότητα σύνδεσης δίνεται στον πίνακα *affinity*. Μεταξύ των στοιχείων του πίνακα αυτού και του πίνακα που επιστρέφεται υπάρχει «1-1» αντιστοιχία, δηλαδή το i στοιχείο του επιστρεφομένου πίνακα δηλώνει τον αριθμό των κλώνων, που θα πρέπει να δώσει το αντίσωμα, του οποίου η ποιότητα σύνδεσης είναι αποθηκευμένη στο στοιχείο *affinity*[i].

Αρκετές φορές για να υπολογιστεί ο αριθμός των κλώνων ενός αντισώματος, ο εκτιμητής του αριθμού των κλώνων θα πρέπει να ταξινομήσει τα αντισώματα βάσει της ποιότητας σύνδεσής τους. Εάν όμως τα αντισώματα είναι ήδη ταξινομημένα, τότε η μέθοδος computeClones() εισάγει επιπλέον υπολογιστικό φόρτο. Για τον λόγο αυτό ένας εκτιμητής κλώνων θα πρέπει να υλοποιεί και τις μεθόδους isSorted() και setSorted(), έτσι ώστε ο χρήστης του εκτιμητή να μπορεί να ρυθμίσει την λειτουργία του, με σκοπό την βελτίωση της απόδοσης. Εάν η σειρά την αντισωμάτων δεν παίζει κανένα ρόλο στον υπολογισμό του αριθμού των κλώνων, τότε αυτές οι μέθοδοι μπορεί να είναι κενές.

Οι εκτιμητές του JAIF

Το Core API του JAIF παρέχει ένα σύνολο διαφορετικών εκτιμητών. Για την ποιότητα σύνδεσης αντισωμάτων-αντιγόνων παρέχονται οι κλάσεις `HammingEvaluator` και `EuclideanEvaluator`, οι οποίες υπολογίζουν την αντίστροφη απόσταση Hamming ή την ευκλείδεια απόσταση μεταξύ δύο αντισωμάτων. Συγκεκριμένα η κλάση `HammingEvaluator` εφαρμόζεται μόνο μεταξύ δυαδικών αντιγόνων και δυαδικών αντισωμάτων ιδίου μήκους, και υπολογίζει την αντίστροφη απόσταση Hamming βάσει της σχέσης (3.2) του Κεφαλαίου 3. Στην γενική περίπτωση το αποτέλεσμα της `computeAffinity()` δεν είναι κανονικοποίημενο. Για να επιτευχθεί αυτό, πρέπει να κληθεί η μέθοδος `setNormalized(true)`. Εάν σε έναν εκτιμητή Hamming κληθεί η μέθοδος `computeAffinity()` με μοναδικό όρισμα ένα αντίσωμα, τότε υπολογίζεται η αντίστροφη απόσταση Hamming αυτού του αντισώματος από ένα δυαδικό αντιγόνο ιδίου μήκους, του οποίου όλα τα bits είναι 0. Τέλος, εάν τα μήκη αντισώματος και αντιγόνου είναι διαφορετικά, εγείρεται μία `IllegalArgumentException`.

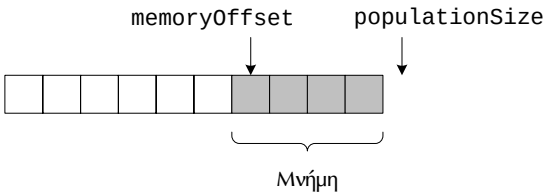
Ένας εκτιμητής της ευκλειδείας απόστασης, `EuclideanEvaluator`, εφαρμόζεται μόνο μεταξύ διανυσματικών αντισωμάτων ιδίου μήκους, ειδ΄ άλλως και σε αυτή την περίπτωση εγείρεται μία `IllegalArgumentException`. Στην περίπτωση που η μέθοδος `computeAffinity()` κληθεί με μοναδικό όρισμα ένα αντίσωμα, τότε υπολογίζεται η απόσταση του αντισώματος από την αρχή των αξόνων.

Για τον υπολογισμό του αριθμού των κλώνων ενός συνόλου αντισωμάτων παρέχεται η κλάση `InverseCEval`, η οποία υπολογίζει τον αριθμό των κλώνων των αντισωμάτων, βάσει της σχέσης (3.5). Η μέθοδος `computeClones()` υποθέτει ότι ο πίνακας *`affinity`* δεν είναι ταξινομημένος κατά φθίνουσα σειρά, και επομένως τον ταξινομεί κάθε φορά που καλείται. Για τον λόγο αυτό, σε περίπτωση που ο πίνακας με τις ποιότητες σύνδεσης είναι ήδη ταξινομημένος με την επιθυμητή σειρά, θα πρέπει να καλείται η μέθοδος `setSorted(true)`, για να αποφεύγεται η εκ νέου ταξινόμηση του πίνακα. Για τον υπολογισμό του αριθμού των κλώνων η παράμετρος n_b τίθεται ίση με το μήκος του πίνακα *`affinity`*, ενώ ο παράγοντας κλωνοποίησης τίθεται μέσω της μεθόδου `setCloneFactor()`.

Για τον υπολογισμό του ρυθμού μετάλλαξης παρέχονται δύο κλάσεις, οι `ExpMEval` και `LinearMEval`. Η πρώτη υπολογίζει τον ρυθμό μετάλλαξης βάσει μίας φθίνουσας εκθετικής συνάρτησης, όπως αυτή που ορίζεται από την σχέση (3.7), ενώ η δεύτερη τον υπολογίζει μέσω μία γραμμικής σχέσης. Οι μέθοδοι αυτών των δύο εκτιμητών για τον υπολογισμό του ρυθμού μετάλλαξης δεν παρουσιάζουν κάποια ιδιατερότητα, ενώ παράλληλα διατίθενται και οι κατάλληλες μέθοδοι για τον προσδιορισμό των παραμέτρων κάθε συνάρτησης, όπως είναι ο μέγιστος ρυθμός μετάλλαξης ή η εξασθένηση.

A.2.4 Ο πληθυσμός των αντισωμάτων

Μία από τις βασικότερες και πιο λειτουργικές μονάδες που παρέχει το JAIF είναι η κλάση `AntibodyPool`, που αντιπροσωπεύει τον πληθυσμό των αντισωμάτων. Η λειτουργικότητα της κλάσης αυτής είναι διπλή: από την μία παρέχει ένα σύνολο μεθόδων για τον άμεσο και εύκολο χειρισμό του συνόλου του πληθυσμού, δίνοντας την δυνατότητα στον χρήστη, να ορίσει εκείνος τις δικές του μεθόδους για τις διάφορες ενέργειες επί του πληθυσμού, όπως είναι η επιλογή των αντισωμάτων, η ανανέωση, κ.λπ. Από την άλλη, η κλάση αυτή παρέχει ένα σύνολο μεθόδων, οι οποίες αυτοματοποιούν πολλές από τις λειτουργίες που γίνονται επί του πληθυσμού των αντισωμάτων, διευκολύνοντας και απλοποιώντας κατά πολύ την διαδικασία ανάπτυξης ενός αλγορίθμου ανοσοποιητικών συστημάτων.



Σχήμα A.3: Η θέση της μνήμης στον πληθυσμό των αντισωμάτων.

Η κλάση `AntibodyPool` επειδή έχει την δυνατότητα να χειρίζεται ένα σύνολο διαφορετικών τύπων αντισωμάτων και αντιγόνων, ορίζεται ως ένας γενικευμένος τύπος δεδομένων, έτσι ώστε να παρέχεται η μέγιστη ασφάλεια κατά τον χρόνο εκτέλεσης.

```
import jaif.AntibodyPool;

public class AntibodyPool<T extends Antibody,
                          S extends Antigen>
```

Ένα αντικείμενο `AntibodyPool` θα πρέπει, επομένως, να συσχετίζεται κατά την δήλωσή του, χωρίς αυτό να είναι υποχρεωτικό, με ένα συγκεκριμένο τύπο αντισωμάτων `T` και ένα συγκεκριμένο τύπο αντιγόνων `S`, που αναγνωρίζονται από τα αντιγόνα τύπου `T`.

Στην τρέχουσα υλοποίηση της κλάσης `AntibodyPool` τα αντισώματα αποθηκεύονται σε ένα μοναδικό πίνακα, χωρίς να χρησιμοποιείται κάποια διαφορετική δομή για την μνήμη, η οποία αντιπροσωπεύεται *πάντα* από τα τελευταία στοιχεία του πίνακα. Με άλλα λόγια, έαν η μνήμη έχει μέγεθος `memorySize`, τα τελευταία `memorySize` στοιχεία του πίνακα του πληθυσμού, αντιστοιχούν στα αντισώματα μνήμης (Σχήμα A.3). Το σημείο του πίνακα του πληθυσμού, από το οποίο ξεκινά η μνήμη ονομάζεται *μετατόπιση μνήμης (memory offset)*. Το μέγεθος της μνήμης και η μετατόπισή της συσχετίζονται μέσω της σχέσης:

```
memorySize := populationSize - memoryOffset
```

Επομένως, το μέγεθος της μνήμης μπορεί να καθοριστεί είτε άμεσα θέτοντας την παράμετρο `memorySize` είτε έμμεσα μέσω της παραμέτρου `memoryOffset`.

```
import jaif.AntibodyPool;

public void setMemorySize(int size);
public void setMemoryOffset(int off);
```

Και οι δύο αυτές μέθοδοι εγείρουν μία `IllegalArgumentException` σε περίπτωση που οι τιμές των *`size`* και *`off`* είναι μη έγκυρες.

Μέθοδοι αμέσου χειρισμού του πληθυσμού

Η κλάση `AntibodyPool` παρέχει ένα σύνολο μεθόδων για τον χειρισμό του πληθυσμού των αντισωμάτων. Οι μέθοδοι αυτοί λειτουργούν κατά κάποιο τρόπο ως ένα είδος διεπιφανείας μεταξύ της εσωτερικής αναπαράστασης του πληθυσμού και του χρήστη. Μέσω αυτών ο χρήστης αναλαμβάνει πλήρη έλεγχο επί του

πληθυσμού, και μπορεί να προσπελάσει ή να θέσει οποιοδήποτε στοιχείο του πληθυσμού, ή ακόμα και να αντικαταστήσει ένα σύνολο στοιχείων με κάποια άλλα.

```
import jaif.*;

public T    get(int index);
public T[]  get(int off, int len);
public void replace(T[] src, int start, int len,
                    int off, boolean blind,
                    boolean replaceWorse);
public void set(int index, T ab);
```

Οι μέθοδοι `get()` επιστρέφουν ένα συγκεκριμένο στοιχείο ή ένα σύνολο διαδοχικών στοιχείων του πληθυσμού. Σε περίπτωση που ο δείκτης κάποιου στοιχείου που ζητείται είναι εκτός των ορίων του πληθυσμού, εγείρεται μία `IndexOutOfBoundsException`. Οι μέθοδοι `set()` και `replace()` επιτελούν την αντίθετη λειτουργία· θέτουν ένα συγκεκριμένο στοιχείο ή ένα σύνολο διαδοχικών στοιχείων του πληθυσμού σε μία συγκεκριμένη τιμή. Παρ΄ όλα αυτά η μέθοδος `replace()` μπορεί, μέσω των παραμέτρων *`blind`* και *`replaceWorse`*, να παράσχει περισσότερο έλεγχο επί της διαδικασίας της αντικατάστασης των αντισωμάτων του πληθυσμού. Εάν η παράμετρος *`blind`* είναι `false`, τότε προτού γίνει η αντικατάσταση των στοιχείων του πληθυσμού, ελέγχεται εάν με αυτή την ενέργεια θα αντικατασταθούν και κύτταρα μνήμης. Εάν κάτι τέτοιο πρόκειται να συμβεί, τότε εγείρεται μία `IllegalArgumentException` και η αντικατάσταση αποτυγχάνει. Όσον αφορά στην παράμετρο *`replaceWorse`*, εάν είναι `true`, τότε τα κύτταρα του πληθυσμού αντικαθίστανται από τα νέα κύτταρα, μόνο στην περίπτωση που τα τελευταία έχουν επιδείξει καλύτερη ποιότητα σύνδεσης. Εδώ θα πρέπει να αναφερθεί, ότι η `replace()` για τον έλεγχο της ποιότητας σύνδεσης καλεί απλά την `getAffinity()` της κλάσης `Antibody`. Επομένως θα πρέπει ο χρήστης να έχει φροντίσει, ώστε οι τιμές της ποιότητας σύνδεσης των αντισωμάτων του πληθυσμού και των αντισωμάτων του πίνακα *`src`* να αναφέρονται στο ίδιο αντιγόνο.

Αυτοματοποίηση κοινών εργασιών

Η κλάση `AntibodyPool` παρέχει ένα σύνολο μεθόδων για την αυτοματοποίηση ενός συνόλου εργασιών που μπορούν να εκτελεστούν επί του συνόλου του πληθυσμού. Οι εργασίες που αυτοματοποιούνται από την κλάση αυτή είναι:

- η αρχικοποίηση του πληθυσμού,
- η εμφάνιση ενός αντιγόνου στον πληθυσμό και αξιολόγησή του,
- η διόρθωση υποδοχέων και η επιλογή των καλυτέρων αντισωμάτων,
- η ανανέωση μνήμης, και
- η ανανέωση πληθυσμού.

Αρχικοποίηση Για την αρχικοποίηση του πληθυσμού των αντισωμάτων παρέχονται δύο μέθοδοι.

```
import jaif.*;

public void initialize();
public void initialize(AntibodyFactory<T> factory);
```

Η διαδικασία της αρχικοποίησης είναι ιδιαίτερα απλή, καθώς απλά καλείται η μέθοδος `newAntibody()` είτε του εργοστασίου αντισωμάτων που είναι συσχετισμένο με τον πληθυσμό είτε του εργοστασίου `factory`, που παρέχεται ρητά από τον χρήστη, τόσες φορές, όσο είναι το μέγεθος του πληθυσμού. Γενικά, κάθε αντικείμενο `AntibodyPool` συσχετίζεται με ένα `AntibodyFactory` είτε μέσω του κατασκευαστή του είτε μέσω της μεθόδου `setAntibodyFactory()`. Κάθε φορά που χρειάζεται να δημιουργηθούν καινούργια αντισώματα, η εργασία αυτή ανατίθεται σε αυτό το εργοστάσιο αντισωμάτων, εκτός και αν ο χρήστης παράσχει ρητά κάποιο άλλο, όπως συμβαίνει στην περίπτωση της `initialize(AntibodyFactory<T> factory)`. Στις περιπτώσεις που ο χρήστης παρέχει ρητά κάποιο εργοστάσιο αντισωμάτων, το εργοστάσιο αυτό χρησιμοποιείται μόνο για την συγκεκριμένη εργασία, χωρίς να τροποποιείται το εργοστάσιο που είναι συσχετισμένο με το αντικείμενο `AntibodyPool`.

Αξιολόγηση του πληθυσμού Η κλάση `AntibodyPool` παρέχει δύο ειδών αξιολογήσεις του πληθυσμού: μία απόλυτη και μία σχετική με ένα αντιγόνο. Ο πρώτος τρόπος μπορεί να χρησιμοποιηθεί σε αλγορίθμους βελτιστοποίησης, όπου δεν υπάρχουν αντιγόνα, ενώ ο δεύτερος τρόπος χρησιμοποιείται κυρίως σε αλγορίθμους αναγνώρισης προτύπων, όπου τα αντιγόνα είναι παρόντα. Μέσω του κατασκευαστή ή της μεθόδου `setAffinityEvaluator()` συσχετίζεται με το αντικείμενο `AntibodyPool` ένας εκτιμητής ποιότητας σύνδεσης. Η αξιολογήση κάθε στοιχείου του πληθυσμού ανατίθεται σε αυτό τον εκτιμήτη, εκτός και αν καθορίζεται διαφορετικά από τον χρήστη. Για την αξιολόγηση του πληθυσμού διατίθενται τρεις μέθοδοι.

```
import jaif.*;

public void evaluate();
public void present(S ag);
public void present(S ag, AffinityEvaluator<T,S> eval);
```

Η μέθοδος `evaluate()` είναι υπεύθυνη για την απόλυτη αξιολογήση του πληθυσμού και είναι ισοδύναμη με την `present(null)`. Οι μέθοδοι `present()` αξιολογούν τον πληθυσμό σε σχέση με το αντιγόνο *ag*. Για να είναι σωστή η αξιολόγηση του πληθυσμού, θα πρέπει ο εκτιμητής ποιότητας σύνδεσης που χρησιμοποιείται να υποστηρίζει το είδος της αξιολόγησης που απαιτείται. Για παράδειγμα, εάν απαιτείται η απόλυτη αξιολόγηση του πληθυσμού, θα πρέπει η κλήση `eval.computeAffinity(ab)` να επιστρέφει ένα έγκυρο μέτρο αξιολόγησης. Αντιστοίχως, στην περίπτωση της αξιολόγησης βάσει ενός αντιγόνου, η κλήση `eval.computeAffinity(ab, ag)` θα πρέπει να επιστρέφει ένα έκγυρο αποτέλεσμα. Το αντιγόνο που παρουσιάστηκε τελευταία φορά στον πληθυσμό, αποθηκεύεται και μπορεί να κανείς το λάβει καλώντας την μέθοδο `getLastPresentedAntigen()`.

Επιλογή των καλυτέρων αντισωμάτων Για την επιλογή των καλυτέρων αντισωμάτων του πληθυσμού, η κλάση `AntibodyPool` παρέχει τις μεθόδους `selectBest()`.

Η επιλογή βασίζεται στην ποιότητα σύνδεσης που επέδειξαν τα αντισώματα του πληθυσμού κατά την παρουσίαση του τελευταίου αντιγόνου (τελευταία κλήση των `present()` ή της `evaluate()`). Κατά την διαδικασία της επιλογής ο πίνακας που περιέχει τα αντισώματα του πληθυσμού δεν τροποποιείται. Αυτό πρακτικά σημαίνει, ότι οι μέθοδοι αμέσου χειρισμού του θα επιστρέψουν τα ίδια ακριβώς αντισώματα, που θα επέστρεφαν και προτού κληθεί κάποια από τις μεθόδους `selectBest()`. Επιπλέον, ούτε τα αντισώματα μνήμης επηρεάζονται. Για να επιτευχθεί αυτό, δεδομένου ότι οι μέθοδοι `sort` της κλάσης `java.util.Arrays` αναδιατάσσουν τα στοιχεία του προς ταξινόμηση πίνακα, διατηρείται παράλληλα με τον πληθυσμό των αντισωμάτων και ένας πίνακας από αντικείμενα της εσωτερικής κλάσης `AntibodyPool.AffinityEntry`. Τα αντικείμενα αυτά αποθηκεύουν την θέση ενός αντισώματος στον πίνακα του πληθυσμού και την ποιότητα σύνδεσής του. Επιπλέον η κλάση `AffinityEntry` υλοποιεί τη διεπιφάνεια `Comparable` παρέχοντας μία σύγκριση των αντισωμάτων επί της ποιότητας σύνδεσής τους. Η κλάση αυτή παρέχει τρεις μεθόδους και ένα κατασκευαστή μέσω του οποίου τίθενται η θέση και η ποιότητα σύνδεσης του αντισώματος.

```
import jaif.AntibodyPool;

public AntibodyPool.AffinityEntry(int pos, double aff);

public int  compareTo(AntibodyPool.AffinityEntry other);
public double   getAffinity();
public int      getPosition();
```

Θα μπορούσε να πει κανείς, ότι τα αντικείμενα της κλάσης `AffinityEntry` λειτουργούν ως δείκτες για τα αντισώματα του πληθυσμού (Σχήμα Α.4). Ο πίνακας που αναδιατάσσεται κατά την φάση της επιλογής είναι ο πίνακας των αντικειμένων `AffinityEntry` και επομένως ο πληθυσμός των αντισωμάτων παραμένει αμετάβλητος.

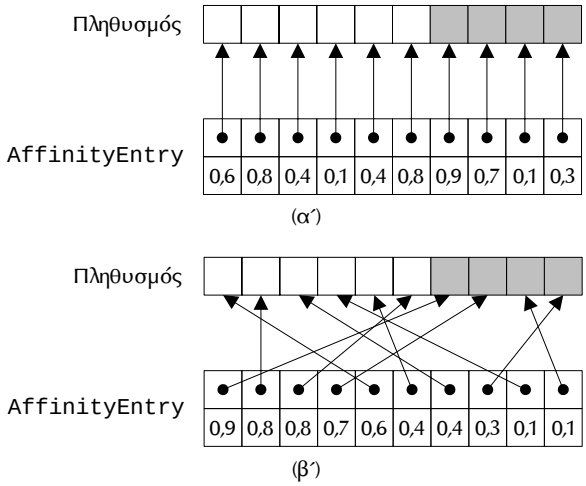
Οι μέθοδοι που παρέχει η κλάση `AntibodyPool` για την επιλογή των καλυτέρων αντισωμάτων είναι δύο.

```
import jaif.AntibodyPool;
import java.util.Comparator;

public T[]  selectBest(int n);
public T[]  selectBest(int n,
        Comparator<? super AffinityEntry> comp,
        boolean copy);
```

Και οι δύο επιλέγουν τα *n* καλύτερα αντισώματα και τα επιστρέφουν σε ένα πίνακα. Παρ΄ όλα αυτά η δεύτερη μέθοδος είναι πιο γενική, παρέχοντας επιπλέον την δυνατότητα στον χρήστη, να καθορίσει εκείνος, μέσω του αντικειμένου `Comparator`, τον τρόπο με τον οποίο θα γίνεται η σύγκριση των αντικειμένων `AffinityEntry`. Τέλος, αν η παράμετρος *copy* είναι `true` επιστρέφεται ένα αντίγραφο των επιλεγμένων αντισωμάτων, ειδ΄ άλλως επιστρέφεται μία αναφορά σε αυτά. Γενικα, το αν θα επιστρέφεται αντίγραφο ή αναφορά στα επιλεγμένα αντισώματα αποτελεί μία ιδιότητα της κλάσης `AntibodyPool`, η οποία μπορεί να τεθεί μέσω της μεθόδου `setCopyOnSelect()`.

Τέλος, μέσω της μεθόδου `enableReceptorEditing()` μπορεί κανείς να ενεργοποιήσει την διόρθωση των υποδοχέων. Εάν είναι ενεργοποιημένη, τότε η μέθοδος



Σχήμα Α.4: Ο τρόπος ταξινόμησης των αντισωμάτων από την μέθοδο `selectBest()`. (α΄) Ο πληθυσμός και ο πίνακας των `AffinityEntry` πριν την ταξινόμηση, (β΄) ο πληθυσμός και ο πίνακας των `AffinityEntry` μετά την ταξινόμηση. Ο πληθυσμός παραμένει αμετάβλητος και ταξινομείται ο πίνακας των `AffinityEntry`.

`selectBest()` προτού επιστρέψει θα εκτελέσει την διόρθωση των υποδοχέων, όπως αυτή περιγράφεται στην §5.2.

Ανανέωση μνήμης Για τον χειρισμό της μνήμης, η κλάση `AntibodyPool` παρέχει μία μόνο καινούργια μέθοδο.

```
import jaif.AntibodyPool;

public boolean  addToMemory(T ab, int pos);
```

Η μέθοδος αυτή *προσπαθεί* να θέσει στην θέση μνήμης `pos` το αντίσωμα `ab`. Οι θέσεις μνήμης μετρώνται από το `memoryOffset` και ένθεν. Επομένως, η θέση `pos` αντιστοιχεί στην θέση `memoryOffset + pos` του πίνακα του πληθυσμού. Η μέθοδος αυτή δεν είναι πάντα επιτυχής, καθ΄ ότι το αντίσωμα `ab` εισέρχεται στην μνήμη, μόνο στην περίπτωση που είναι καλύτερο από το αντίσωμα που αντικαθιστά στην θέση `pos`. Στην περίπτωση που η εισαγωγή στην μνήμη είναι επιτυχής επιστρέφεται `true`. Αυτό που χρήζει προσοχής είναι ότι η μέθοδος αυτή, δεν ασχολείται με το αν η ποιότητα σύνδεσης του `ab` και του αντισώματος μνήμης που αντικαθιστά αναφέρονται στο ίδιο αντιγόνο· αυτό θα πρέπει να εξασφαλίζεται από τον χρήστη.

Γενικά, η ανανέωση της μνήμης μπορεί να γίνει και με τις μεθόδους αμέσου χειρισμού του πληθυσμού, μόνο που σε αυτή την περίπτωση θα πρέπει οι μέθοδοι αυτές να συνδυαστούν με την μέθοδο `getMemoryOffset()`.

Ανανέωση πληθυσμού Για την ανανέωση του πληθυσμού των αντισωμάτων παρέχονται από το JAIF δύο μέθοδοι.

```
import jaif.*;
import java.util.Comparator;

public void refresh();
public void refresh(int ncells,
        AntibodyFactory<T> factory,
        Comparator<? super AffinityEntry> comp);
```

Η δεύτερη από αυτές είναι η πιο γενική και επιλέγει τα *ncells* χειρότερα αντισώματα βάσει του Comparator *comp* και τα αντικαθιστά με εντελώς νέα αντισώματα που παράγονται από το εργοστάσιο αντισωμάτων *factory*. Η απλή refresh() τις προκαθορισμένες τιμές για τις παραμέτρους αυτές, οι οποίες μπορούν να τεθούν από κατάλληλες μεθόδους. Εάν επιθυμείται μόνο η αλλαγή του Comparator που χρησιμοποιείται, τότε επειδή δεν υπάρχει κάποια μέθοδος για να τον θέτει ρητά, θα πρέπει να χρησιμοποιείται η παρακάτω κλήση:

```
pool.refresh(pool.getRefreshValue(),
             pool.getAntibodyFactory(),
             comp);
```

Η μέθοδος getRefreshValue() επιστρέφει το προκαθορισμένο πλήθος αντισωμάτων που ανανεώνονται από την μέθοδο refresh().

A.2.5 Υποστήριξη του ΠΓΕ

Το πακέτο jaif.gep παρέχει στο Java Artificial Immune Framework την δυνατότητα να χειρίζεται αντισώματα τύπου-ΠΓΕ. Οι κλάσεις του πακέτου αυτού, όπως θα φανεί και στην συνέχεια κατά την περιγραφή του, εναρμονίζονται πλήρως με το Core API του JAIF. Αυτό έχει ως αποτέλεσμα, να μπορούν να ενσωματωθούν εύκολα χωρίς σημαντικές προσθήκες ή αλλαγές σε ήδη υπάρχοντα κώδικα, που βασίζεται πάνω στην βασική προγραμματιστική διεπαφή του JAIF.

Αναπαράσταση συμβόλων και συναρτήσεις του ΠΓΕ

Για την αναπαράσταση των συμβόλων του ΠΓΕ παρέχεται η κλάση GEPSymbol. Η κλάση αυτή μπορεί να αναπαραστήσει οποιοδήποτε σύμβολο που εμφανίζεται στον ΠΓΕ, είτε αυτό είναι τερματικό, είτε συναρτησιακό είτε σταθερά. Ένα GEPSymbol έχει γενικά τρεις ιδιότητες:

1. Το ακριβές σύμβολο (raw symbol) με το οποίο το συγκεκριμένο GEPSymbol εμφανίζεται στις εκφράσεις του ΠΓΕ.
2. Εάν το σύμβολο είναι σταθερά, τότε συσχετίζεται με αυτό και μία σταθερή τιμή.
3. Εάν το σύμβολο είναι συναρτησιακό, τότε συσχετίζεται με αυτό και μία συγκεκριμένη συνάρτηση.

Η πρώτη ιδιότητα ανήκει σε όλα τα σύμβολα-ΠΓΕ, ενώ οι άλλες δύο έχουν νόημα μόνο στις περιπτώσεις που αναφέρθηκαν. Σε αντίθεση με τον ορισμό του ΠΓΕ από τον Ferreira, το ακριβές σύμβολο ενός GEPSymbol κωδικοποιείται ως String, δηλαδή ως μία ακολουθία συμβόλων. Αυτό γίνεται για να δίνεται η μέγιστη ελευθερία στην επιλογή των συμβόλων, καθ΄ ότι τα σύμβολα ενός χαρακτήρα σε

μία γλώσσα προγραμματισμού είναι πεπερασμένα, σε αντίθεση με τα μαθηματικά σύμβολα, που είναι άπειρα.

Το είδος ενός συμβόλου-ΠΓΕ μπορεί να καθοριστεί είτε κατά την κατασκευή του είτε εκ των υστέρων.

```
import jaif.gep.*;

public GEPSymbol(String symbol);
public GEPSymbol(String symbol, double val);
public GEPSymbol(String symbol, GEPFunction funct);
```

Ο πρώτος κατασκευαστής δημιουργεί ένα τερματικό σύμβολο, ο δεύτερος μία σταθερά με τιμή *val*, ενώ ο τρίτος δημιουργεί ένα συναρτησιακό σύμβολο με συνάρτηση *funct*. Γενικά, αφότου δημιουργηθεί ένα σύμβολο επιτρέπεται να αλλάξει είδος, για παράδειγμα από τερματικό να γίνει συναρτησιακό ή σταθερά.

```
import jaif.gep.*;

public void setFunction(GEPFunction funct);
public void makeConstant(double val);
```

Η μέθοδος `setFunction()` μετατρέπει αυτό το σύμβολο σε συναρτησιακό αναθέτοντας του την συνάρτηση *funct*, εκτός και αν *funct* == `null`, οπότε το σύμβολο μετατρέπεται σε τερματικό. Η μέθοδος `makeConstant()` μετατρέπει αυτό το σύμβολο σε σταθερά με τιμή *val*. Στο πακέτο `jaif.gep` ορίζονται επιπλέον τα εξής βασικά συναρτησιακά σύμβολα: `ADD`, `SUB`, `MUL`, `DIV`, `IF`, `AND`, `OR`, `NOT`, `SQRT`. Τα `ADD`, `SUB`, `MUL`, `DIV` αντιστοιχούν στις 4 βασικές αριθμητικές πράξεις, το `SQRT` αντιστοιχεί στην τετραγωνική ρίζα, τα `IF`, `AND`, `OR` ορίζονται όπως στις σχέσεις (5.1), (5.19) και (5.18) αντιστοίχως, ενώ το `NOT` ορίζεται ως εξής:

$$N(x) = \begin{cases} 0, & \text{αν } x \neq 0 \\ 1, & \text{ειδ΄ άλλως} \end{cases}$$

Τέλος, η κλάση `GEPSymbol` παρέχει μερικές ακόμα μεθόδους για την αναγνώριση του είδους του συμβόλου, οι οποίες όμως αναλύονται στην τεκμηρίωση του πακέτου `jaif.gep`.

Για τον ορισμό συναρτήσεων που πρόκειται να χρησιμοποιηθούν με συναρτησιακά σύμβολα, παρέχεται η διεπιφάνεια `GEPFunction`. Η διεπιφάνεια αυτή ορίζει δύο μεθόδους.

```
import jaif.gep.*;

public double   call(Double... args) throws GEPException;
public int      getArity();
```

Η μέθοδος `call()` αποτελεί την υλοποίηση της συνάρτησης και για τον λόγο αυτό δέχεται μεταβλητό αριθμό ορισμάτων, έτσι ώστε να μπορεί να περιγράψει οποιαδήποτε συνάρτηση. Σε περίπτωση που συμβεί κάποιο λάθος στην συνάρτηση θα πρέπει να εγείρεται μία `GEPException`. Η εξαίρεση αυτή επεκτείνει τον βασικό τύπο εξαίρεσης του JAIF, την `JAIFException`. Η μέθοδος `getArity()` επιστρέφει το πλήθος των ορισμάτων που δέχεται η συνάρτηση. Το πακέτο `jaif.gep` παρέχει τις υλοποιήσεις των συναρτήσεων των βασικών συναρτησιακών συμβόλων,

που ορίστηκαν προηγουμένως. Ως παράδειγμα υλοποίησης της διεπιφάνειας `GEPFunction`, στο Πρόγραμμα Α.2 φαίνεται η υλοποίηση της συνάρτησης `IF`, που παρέχεται από το JAIF.

```
class GepIf implements GEPFunction {

  private static final int  arity = 3;

  public int getArity() { return arity; }

  public double call(Double ... args) throws GEPException {

    if (args.length < arity)
      throw new GEPException("GepIf: too few arguments");

    if (args[0] > 0.)
      return args[1];
    else
      return args[2];
  }
}
```

Πρόγραμμα Α.2: Η υλοποίηση της διεπιφανείας `GEPFunction` για τον ορισμό της συνάρτησης `IF`.

Το αντίσωμα τύπου-ΠΓΕ

Ένα αντίσωμα τύπου-ΠΓΕ αντιπροσωπεύεται από την κλάση `GEPAntibody`. Η κλάση αυτή υλοποιεί μόνο τον γενετικό τελεστή της μετάλλαξης, και όχι όλους τους γενετικούς τελεστές που ορίζονται από τον Ferreira (βλ. §4.2.2), καθ΄ ότι είναι ο μόνος που χρησιμοποιείται από τον αλγόριθμο επιλογής κλώνων. Η ανασύνθεση πολλών σημείων και πολλών γονέων που αναφέρεται στην §5.3, ουσιαστικά υλοποιείται από τον μηχανισμό των εργοστασίων αντισωμάτων, μέσω της μεθόδου `newAntibody(CodonSequence[] fragments)`. Ακόμη, υποστηρίζονται πλήρως αντισώματα πολλών γονιδίων.

Κατά την κατασκευή ενός `GEPAntibody` προσδιορίζονται το μέγεθος της κεφαλής ενός γονιδίου, *hlen*, το πλήθος των γονιδίων, *ngenes* και το αλφάβητο των αντισωμάτων, *alphabet*.

```
import jaif.gep.*;

public GEPAntibody(int hlen, int ngenes,
                   GEPSymbol[] alphabet);
```

Ο κατασκευαστής αυτός, αφότου επιτελέσει κάποιες βασικές αρχικοποιήσεις, καλεί την μέθοδο `setAlphabet(alphabet)`, η οποία υπολογίζει το μήκος ενός γονιδίου του αντισώματος βάσει της σχέσης (4.3), και εκτελεί όλες τις ενέργειες που σχετίζονται με το αλφάβητο των αντισωμάτων. Γενικά, η `setAlphabet()` μπορεί να κληθεί από τον χρήστη και ανεξάρτητα, οποιαδήποτε χρονική στιγμή, μεταβάλλοντας τα χαρακτηριστικά του αντισώματος, που σχετίζονται με το αλφάβητο. Η κλάση `GEPAntibody` για την αναπαράσταση ολοκλήρου του αντισώματος διατηρεί εσωτερικά ένα πίνακα από `String`, όπου κάθε `String` αντιπροσωπεύει το ακριβές σύμβολο του συμβόλου-ΠΓΕ του αλφαβήτου. Η μέθοδος `getChromosome()`

επιστρέφει ένα αντίγραφο αυτού του πίνακα συμβόλων. Για να είναι δυνατή η αναγνώριση του τύπου του κάθε συμβόλου (τερματικό, σταθερά, συναρτησιακό), διατηρείται επίσης μία απεικόνιση μεταξύ του ακριβούς συμβόλου του αντισώματος και του αντιστοίχου συμβόλου-ΠΓΕ. Η απεικόνιση αυτή γίνεται προσιτή στον χρήστη μέσω της μεθόδου `getSymbolMap()`, η οποία επιστρέφει ένα αντικείμενο τύπου `Map<String, GEPSymbol>`.

Σε αντίθεση με τα άλλα αντισώματα που παρέχει το JAIF (`BinaryAntibody`, `VectorAntibody`), ο κατασκευαστής ενός αντισώματος τύπου-ΠΓΕ δεν αρχικοποιεί το αντίσωμα· κάθε σύμβολό του είναι αρχικά `null`. Επομένως, θα πρέπει να κληθεί μία από τις μεθόδους `initialize()`.

```
import jaif.gep.GEPAntibody;

public void initialize();
public void initialize(String[] chrom);
public void initialize(String rep, boolean asRule);
```

Η πρώτη μέθοδος αρχικοποιεί το αντίσωμα ως μία τυχαία ακουλουθία συμβόλων. Στην κεφαλή του κάθε γονιδίου τα τερματικά σύμβολα (απλά ή σταθερές) και τα συναρτησιακά σύμβολα εκλέγονται με την ίδια πιθανότητα (50%). Εσωτερικά, τα τερματικά και τα συναρτησιακά σύμβολα αποθηκεύονται σε διαφορετικές δομές, οπότε για την εκλογή ενός στοιχείου της κεφαλής επιλέγεται με πιθανότητα 50% μία δομή, και από αυτή την δομή εκλέγεται τυχαία ένα σύμβολο. Η δεύτερη μέθοδος αρχικοποιεί το αντίσωμα από το ΠΓΕ χρωμόσωμα *chrom*. Τέλος, η τρίτη μέθοδος `initialize()` αρχικοποιεί ρητά ένα αντίσωμα από την αναπαράστασή του *rep*. Ένα αντίσωμα τύπου-ΠΓΕ μπορεί να αναπαρασταθεί με δύο τρόπους: είτε (i) ως μία ακολουθία συμβόλων, είτε (ii) ως μία μαθηματική έκφραση (ή κανόνας) που αντιστοιχεί στο ΔΕ του αντισώματος. Το πρώτο είδος αναπαράστασης αντιστοιχεί στην περίπτωση όπου *asRule* `== false`, οπότε η αναπαράσταση *rep* αντιστοιχεί στην ακολουθία συμβόλων του αντισώματος. Επειδή τα σύμβολα του ΠΓΕ αναπαρίστανται ως `String`, θα πρέπει να ορίζεται ένα διαχωριστικό συμβόλων. Πράγματι, η κλάση `GEPAntibody` ορίζει ένα προκαθορισμένο διαχωριστικό, το `"' "`, με το οποίο θα πρέπει να διαχωρίζονται τα σύμβολα-ΠΓΕ στην αναπαράσταση συμβόλων. Το διαχωριστικό αυτό μπορεί να αλλαχθεί μέσω της μεθόδου `setSeparator()`, αλλά δεν θα πρέπει να ταυτίζεται με κάποιο σύμβολο-ΠΓΕ για να μην δημιουργείται σύγχυση. Όταν το αντίσωμα αναπαρίσταται ως κανόνας, τότε θα πρέπει η παράμετρος *asRule* να είναι `true`. Στην περίπτωση αυτή, η μαθηματική έκφραση *rep* θα πρέπει να είναι γραμμένη σε προθεματική μορφή, όπου πρώτα γράφεται το συναρτησιακό σύμβολο και στην συνέχεια μέσα σε παρενθέσεις και διαχωρισμένα με κόμματα δίνονται τα ορίσματα. Για παράδειγμα η έκφραση:

```
((a + b) OR (b - c)) * SQRT(a)
```

θα πρέπει να γραφεί στην μορφή:

```
*(OR(+(a,b),-(b,c)),SQRT(a))
```

Γενικά, η μαθηματική έκφραση ενός γονιδίου τύπου-ΠΓΕ αποτελεί στην ουσία την έκφραση ενός ανοικτού πλαισίου ανάγνωσης (ΑΠΑ) του γονιδίου, οπότε το μήκος της είναι συνήθως μικρότερο του μήκους του γονιδίου. Στην περίπτωση αυτή, η μέθοδος `initialize()` συμπληρώνει ολόκληρο το υπόλοιπο αντίσωμα με το τελευταίο σύμβολο του ΑΠΑ. Επειδή η αναπαράσταση *rep* αναπαριστά μόνο ένα

κανόνα, η `initialize()` έχει νόημα μόνο στην περίπτωση, που το αντίσωμα τύπου-ΠΓΕ αποτελείται από ένα μόνο γονίδιο.

Κλείνοντας την συζήτηση για τους τρόπους ρητής αρχικοποίησης ενός αντισώματος τύπου-ΠΓΕ, θα πρέπει να αναφερθεί, ότι καμία από τις μεθόδους αυτές δεν ελέγχει εάν η δομή του αντισώματος είναι έγκυρη. Ο μόνος έλεγχος που παρέχεται και στις δύο περιπτώσεις ρητής αρχικοποίησης, είναι εάν τα σύμβολα του αντισώματος αρχικοποίησης ανήκουν στο αλφάβητο του προς αρχικοποίηση αντισώματος, ενώ στις περιπτώσεις αρχικοποίησης από χρωμόσωμα-ΠΓΕ και από ακολουθία συμβόλων, ελέγχεται εάν τα μήκη των αντισωμάτων είναι κοινά. Εάν κάποια από τις παραπάνω συνθήκες δεν ισχύει, εγείρεται μία `IllegalArgumentException`.

Οι μέθοδοι `mutate()` για την μετάλλαξη του αντισώματος τύπου-ΠΓΕ, αλλά και το εργοστάσιο `GEPAntibodyFactory` δεν παρουσιάζουν κάποια ιδιαιτερότητα, και αναλύονται στην τεκμηρίωση του πακέτου `jaif.gep`. Το μόνο που πρέπει να σημειωθεί, είναι ότι η μέθοδος `newAntibody()` της κλάσης `GEPAntibodyFactory` αρχικοποιεί με τυχαίο τρόπο το νέο αντίσωμα. Τέλος, η κλάση `GEPAntibody` παρέχει επιπλέον μεθόδους για τον χειρισμό ενός αντισώματος τύπου-ΠΓΕ, ανάμεσα στις οποίες είναι και οι μέθοδοι `getExpression()` και η `getDescriptionLength()`, οι οποίες επιστρέφουν την αναπαράσταση αυτού του αντισώματος σε αλγεβρική μορφή, και το μήκος περιγραφής του, αντιστοίχως. Το μήκος περιγραφής ενός αντισώματος με πολλά γονίδια προκύπτει από το άθροισμα των ΑΠΑ κάθε γονιδίου χωριστά συν το πλήθος των συμβόλων της συνάρτησης σύνδεσης (link function), που χρησιμοποιούνται για την σύνδεση των γονιδίων του αντισώματος.

Υπολογισμός της έκφρασης ενός αντισώματος τύπου-ΠΓΕ

Το πακέτο υποστήριξης του ΠΓΕ παρέχει επιπλέον και έναν εκτιμητή ποιότητας σύνδεσης του αντισώματος `GEPAntibody`. Στην πραγματικότητα πρόκειται για μία κλάση που υπολογίζει την τιμή της έκφρασης του αντισώματος `GEPAntibody`.

```
import jaif.gep.*;
import java.util.Map;

public class GEPEvaluator extends
    AffinityEvaluator<GEPAntibody, Antigen>

public double   computeAffinity(GEPAntibody ab);
public double   computeAffinity(GEPAntibody ab,
                                Antigen ag);
public void     setAssignments(Map<String, Double> map);
```

Η κλάση `GEPEvaluator` αποτελεί παράδειγμα ενός εκτιμητή ποιότητας σύνδεσης που εκτιμά την απόλυτη ποιότητα σύνδεσης ενός αντισώματος, η οποία εν προκειμένω ταυτίζεται με την τιμή του ΔΕ του αντισώματος τύπου-ΠΓΕ. Επομένως, ο `GEPEvaluator` δεν συσχετίζει ένα αντίσωμα `GEPAntibody` με κάποιο συγκεκριμένο αντιγόνο, και για τον λόγο αυτό συσχετίζεται με το αφηρημένο αντιγόνο `Antigen`[1]. Οι δύο μέθοδοι `computeAffinity()` είναι εντελώς ισοδύναμοι· και οι δύο υπολογίζουν την τιμή της έκφρασης του αντισώματος *ab*. Για να γίνει ο υπολογισμός της έκφρασης του αντισώματος, θα πρέπει να δοθούν στον εκτιμητή

[1]Τυπικά θα μπορούσε να συσχετιστεί με οποιοδήποτε αντιγόνο, αλλά χρησιμοποιείται η κλάση `Antigen` για λόγους απλότητας.

`GEPEvaluator` οι τιμές των τερματικών συμβόλων (πλην των σταθερών), πράγμα το οποίο επιτυγχάνεται μέσω της μεθόδου `setAssignments()`. Η απεικόνιση *map* συσχετίζει το ακριβές σύμβολο ενός τερματικού `GEPSymbol` με την τιμή του. Ο υπολογισμός της τιμής της έκφρασης του αντισώματος γίνεται επί τόπου (in situ), χωρίς να μετατρέπεται το αντίσωμα στο αντίστοιχο ΔΕ. Αυτό έχει σαν αποτέλεσμα, η μέθοδος `computeAffinity()` να είναι αρκετά γρήγορη και αρκετά φειδωλή σε υπολογιστικούς πόρους. Σε αντίθεση με τους άλλους εκτιμητές ποιότητας σύνδεσης, ένας `GEPEvaluator` μπορεί να εγείρει μία `IllegalArgumentException` σε περίπτωση που συμβεί κάποιο λάθος κατά τον υπολογισμό της έκφρασης του αντισώματος. Τέτοια λάθη είναι για παράδειγμα να μην έχει δοθεί η τιμή ενός τερματικού συμβόλου, ή τα γονίδια του αντισώματος τύπου-ΠΓΕ να μην επαρκούν για τον υπολογισμό της συνάρτησης σύνδεσης του αντισώματος. Στο Πρόγραμμα Α.3 παρουσιάζονται οι περισσότερες από τις λειτουργίες των αντικειμένων `GEPAntibody`. Αρχικά, ορίζεται ένα αλφάβητο, που περιέχει όλων των ειδών σύμβολα-ΠΓΕ, και κατασκευάζεται μέσω ενός `GEPAntibodyFactory` ένα αντίσωμα `GEPAntibody`. Στην συνέχεια ορίζονται οι τιμές των τερματικών συμβόλων και ανατίθενται σε έναν εκτιμητή `GEPEvaluator`, ο οποίος υπολογίζει την έκφραση του αντισώματος τύπου-ΠΓΕ.

```
import jaif.*;
import jaif.gep.*;
import java.util.HashMap;

public class GEPTest {

  public static final int   HEAD_LEN = 10;
  public static final int   NGENES   = 5;

  public static void main(String[] args) {

    GEPSymbol[] alphabet =
      new GEPSymbol[] { GEPSymbol.ADD,
                        GEPSymbol.SUB,
                        GEPSymbol.MUL,
                        GEPSymbol.DIV,
                        GEPSymbol.IF,
                        new GEPSymbol("a"),
                        new GEPSymbol("b"),
                        new GEPSymbol("c"),
                        new GEPSymbol("1", 1.0),
                        new GEPSymbol("2", 2.0) };

    GEPAntibodyFactory    fact = new GEPAntibodyFactory();
    fact.setHeadLength(HEAD_LEN);
    fact.setGeneCount(NGENES);
    fact.setAlphabet(alphabet);
    fact.setGeneLinkFunction(GEPFunction.IF);

    // Initialize the antibody
    GEPAntibody   ab = fact.newAntibody();
    System.out.println(ab);

    // Evaluate the antibody
```

```
    GEPEvaluator    eval = new GEPEvaluator();
    HashMap<String, Double>  assignMap =
      new HashMap<String, Double>();
    assignMap.put("a", 3.);
    assignMap.put("b", 2.);
    assignMap.put("c", 1.);
    eval.setAssignments(assignMap);

    // Print the assignments
    System.out.printf("a = %f, b = %f, c = %f%n",
                      assignMap.get("a"),
                      assignMap.get("b"),
                      assignMap.get("c"));

    // Print the GEP expression, its value,
    // and the description length
    System.out.printf("%s = %f%n", ab.getExpression(),
                      eval.computeAffinity(ab));
    System.out.println("descr len = " +
                       ab.getDescriptionLength());
  }
}
```

Πρόγραμμα Α.3: Ένα παράδειγμα κατασκευής και υπολογισμού της έκφρασης ενός αντισώματος τύπου-ΠΓΕ.

Α.3 Εργαλεία ανοσολογικής μηχανικής

Στην τρέχουσα έκδοση του Java Artificial Immune Framework το μοναδικό εργαλείο ανοσολογικής μηχανικής που παρέχεται, είναι ο αλγόριθμος επιλογής κλώνων, όπως αυτός περιγράφηκε στα Κεφάλαια 3 και 5. Η κλάση που τον υλοποιεί είναι η `ImmuneAlgorithm` και βρίσκεται στο πακέτο `jaif.alg`, όπως και όλες οι κλάσεις που υλοποιούν κάποιον αλγόριθμο βασισμένο στο ανοσοποιητικό σύστημα. Η κλάση `ImmuneAlgorithm`, απο προγραμματιστικής απόψεως, δεν εισάγει κάποιο καινούργιο στοιχείο, απλά συνδυάζει τις λειτουργίες που παρέχονται από το Core API, έτσι ώστε να υλοποιήσει τον αλγόριθμο επιλογής κλώνων. Για τον ορισμό των διαφόρων παραμέτρων του αλγορίθμου, παρέχει ένα σύνολο μεθόδων, οι οποίες καλούν συνήθως τις αντίστοιχες μεθόδους του Core API. Στην τρέχουσα έκδοση οι κατασκευαστές του `ImmuneAlgorithm` είναι αρκετά σύνθετοι απαιτώντας ένα αρκετά μεγάλο σύνολο παραμέτρων για να κατασκευάσουν το αντικείμενο. Σε μελλοντικές εκδόσεις μπορεί αυτό να βελτιωθεί. Ο αλγόριθμος αρχίζει να εκτελείται καλώντας την μέθοδο `fit()`. Τα προς αναγνώριση πρότυπα, γνωστοποιούνται στον αλγόριθμο μέσω της μεθόδου `setTargets()`.

```
import jaif.*;
import jaif.alg.ImmuneAlgorithm;

public class ImmuneAlgorithm<T extends Antibody,
                             S extends Antigen>

public void fit();
public void setTargets(S[] targets);
```

Ένα παράδειγμα προσδιορισμού των παραμέτρων του αλγορίθμου και εκτέλεσής του φαίνεται στο Πρόγραμμα Α.4, όπου εφαρμόζεται ο αλγόριθμος επιλογής κλώνων στο δεύτερο πρόβλημα αναγνώρισης ψηφιακών χαρακτήρων, που περιγράφηκε στην §3.4.1. Το αρχείο `targets.txt` που εμφανίζεται στον κώδικα, είναι το αρχείο που περιέχει τους κωδικοποιημένους χαρακτήρες.

```
import jaif.*;
import jaif.alg.*;
import jaif.evaluators.*;
import java.io.*;
import java.util.Arrays;

public class IATest {

  private static final String   TARGET_FILE = "targets.txt";
  private static final int      NTARGETS    = 8;
  private static final int      ABSIZE      = 120;
  private static final int      NGEN        = 500;
  private static final double   EPS         = 0.0;
  private static final int      POPSIZE     = 10;
  private static final int      MEMSIZE     = NTARGETS;
  private static final int      NSELECT     = 4;
  private static final int      NREPLACE    = 0;
  private static final double   MUT_DECAY   = 5.;
  private static final double   MAX_MUTRATE = .7;
  private static final int      NREFRESH    = 0;
  private static final float    CLONE_FACT  = 20.0f;
  private static final int      NCELLS_EDIT = 0;

  public static void main(String[] args) {

    BinaryAntigen[] targets = new BinaryAntigen[NTARGETS];
    try {
      // read in the targets
      BufferedReader    in =
        new BufferedReader(new FileReader(TARGET_FILE));
      String    line;
      String    ag = "";
      int       i = 0;

      while ( (line = in.readLine()) != null) {
        if (line.equals("")) {
          targets[i++] = new BinaryAntigen(ag);
          ag = "";
        } else
          ag += line;
      }

      // Print parameters
      System.out.println("Antibody size: " + ABSIZE);
      System.out.println("Number of generations: " + NGEN);
      System.out.println("Eps: " + EPS);
      System.out.println("Population size: " + POPSIZE);
```

```
      System.out.println("Memory size: " + MEMSIZE);
      System.out.println("Members selected: " + NSELECT);
      System.out.println("Members replaced: " + NREPLACE);
      System.out.println("Mutation decay: " + MUT_DECAY);
      System.out.println("Maximum mutation rate: " +
                         MAX_MUTRATE);
      System.out.println("Refresh value: " + NREFRESH);
      System.out.println("Clone factor: " + CLONE_FACT);
      System.out.println("Cells edited: " + NCELLS_EDIT);
      System.out.println("TARGETS:");
      System.out.println(Arrays.toString(targets));
      // Set up the antibody factory
      BinaryAntibodyFactory   fact =
        BinaryAntibodyFactory.getInstance();
      fact.setGeneSize(ABSIZE);
      // Set up the evaluators
      AffinityEvaluator<BinaryAntibody, BinaryAntigen>
        affeval = new HammingEvaluator(true, true);
      MutationEvaluator   muteval = new ExpMEval(MAX_MUTRATE,
                                                 MUT_DECAY);
      InverseCEval        cloneeval = new InverseCEval();
      cloneeval.setCloneFactor(CLONE_FACT);
      // Set up the algorithm
      ImmuneAlgorithm<BinaryAntibody, BinaryAntigen>  alg =
        new ImmuneAlgorithm<BinaryAntibody, BinaryAntigen>
        (NGEN, EPS, POPSIZE, MEMSIZE,
         NSELECT, NREPLACE, NREFRESH, affeval,
         muteval, cloneeval, fact, targets);
      alg.enableReceptorEditing(false, NCELLS_EDIT,
                                NCELLS_EDIT);
      System.out.println("algorithm is running...");
      alg.fit();
    } catch (Exception e) {
      System.err.println("IATest error: " + e.getMessage());
    }
  }
}
```

Πρόγραμμα A.4: Εφαρμογή του αλγορίθμου `ImmuneAlgorithm` στο πρόβλημα αναγνώρισης ψηφιακών χαρακτήρων του Κεφαλαίου 3.

A.4 Επέκταση για Εξόρυξη από Δεδομένα

Η επέκταση του Java Artificial Immune Framework για εξόρυξη από δεδομένα μπορεί να χωριστεί σε δύο μέρη. Στο πρώτο μέρος ανήκουν κλάσεις και διεπιφάνειες, που είναι υπεύθυνες για τον χειρισμό των συνόλων δεδομένων. Στο δεύτερο μέρος ανήκουν κλάσεις, οι οποίες ασχολούνται με την διαδικασία της εξόρυξης από τα δεδομένα. Όλες αυτές οι κλάσεις ανήκουν στο πακέτο `jaif.dm`.

Ο χειρισμός των δεδομένων που παρέχεται αυτή την στιγμή από το JAIF αν και σχετικά απλός, είναι αρκετά ευέλικτος. Παρέχει την δυνατότητα να ανακτηθούν τα προς εξόρυξη δεδομένα είτε από ένα τοπικό αρχείο, είτε ένα αρχείο που βρίσκεται στο δίκτυο, είτε ακόμα και από μία βάση δεδομένων μέσω JDBC. Επιπλέον, με

την εισαγωγή των δρομέων παρέχει ένα εύκολο και ανεξάρτητο από την πηγή δεδομένων τρόπο, για τον χειρισμό των εγγραφών ενός συνόλου δεδομένων.

A.4.1 Πρόσβαση στα δεδομένα

Ο τρόπος με τον οποίο ένα πρόγραμμα που χρησιμοποιεί το JAIF, μπορεί να έχει πρόσβαση στα προς εξόρυξη δεδομένα, είναι μέσω κάποιας κλάσης που υλοποιεί την διεπιφάνεια `DataLoader`. Σκοπός των κλάσεων που υλοποιούν αυτή την διεπιφάνεια είναι αφενός να συνδεθούν με την πηγή δεδομέων και αφετέρου να μετατρέψουν τα προς εξόρυξη δεδομένα στην μορφή που αυτά γίνονται κατανοητά από το JAIF. Επομένως, οι υλοποιήσεις αυτής της διεπιφάνειας εξαρτώνται άμεσα από την μορφή των δεδομένων, και μπορεί να είναι είτε πολύ απλές είτε πολύ σύνθετες. Η διεπιφάνεια `DataLoader` ορίζει 4 μεθόδους.

```
import jaif.dm.*;
import java.net.URL;

public void connect(URL dburl)
    throws DatabaseException;
public DataSet  getDataSet() throws DatabaseException;
public void     closeConnection()
    throws DatabaseException;
public DataClassAntigen[] getDataClasses()
    throws DatabaseException;
```

Οι μέθοδοι `connect()` και `closeConnection()` χρησιμοποιούνται για την σύνδεση με την πηγή δεδομένων. Από την γενική μορφή της μεθόδου `connect()` είναι φανερό, ότι επιτρέπει την διαχείριση τοπικών ή μη αρχείων, αλλά και αρχείων ΒΔ μέσω των κατάλληλων οδηγών JDBC. Ο χειρισμός της σύνδεσης με την πηγή δεδομένων, αλλά και ο τρόπος εξαγωγής των δεδομέων γίνεται διαφανώς ως προς τον χρήστη της διεπιφάνειας. Μέσω της μεθόδου `getDataSet()` επιστρέφονται τα προς εξόρυξη δεδομένα στην μορφή ενός συνόλου δεδομένων τύπου `DataSet`, το οποίο αναγνωρίζεται από την υποδομή του JAIF. Η μέθοδος `getDataClasses()` επιστρέφει τις κλάσεις των δεδομένων του προβλήματος στην μορφή αντιγόνων κλάσεων δεδομένων (ΑΚΔ). Και οι 4 μέθοδοι μπορούν να εγείρουν μία εξαίρεση `DatabaseException` η οποία δηλώνει, ότι κάποιο λάθος συνέβη είτε κατά την διαδικασία της σύνδεσης με την πηγή των δεδομένων είτε κατά την επεξεργασία τους.

Τα σύνολα δεδομένων στο JAIF αναπαρίστανται ως μία ακολουθία εγγραφών `RowData`. Κάθε εγγραφή `RowData` αποτελείται από ένα σύνολο χαρακτηριστικών και τις αντίστοιχες τιμές τους, οι οποίες προσδιορίζονται κατά την κατασκευή της εγγραφής. Η κλάση αυτή παρέχει επίσης μεθόδους για την ανάκτηση της τιμής ενός ή όλων των χαρακτηριστικών, ενώ μπορεί να επιστρέψει και τον πίνακα απεικόνισης που χρησιμοποιεί για τις τιμές των χαρακτηριστικών.

```
import jaif.dm.RowData;
import java.util.Map;

public RowData(String[] attnam, double[] attval);

public Map<String, Double>  getAttributeMap();
public double[] getAttributeValues();
public double   getAttributeValue(String attname)
    throws AttributeNotFoundException;
```

Η μέθοδος `getAttributeValue()` επιστρέφει την τιμή του χαρακτηριστικού που ζητείται. Αν δεν υπάρχει εγείρει μία εξαίρεση `AttributeNotFoundException`. Η εξαίρεση αυτή επεκτείνει τον γενικό τύπο εξαιρέσεων του JAIF, `JAIFException`. Η μέθοδος `getAttributeValues()` επιστρέφει όλες τις τιμές των χαρακτηριστικών τις εγγραφής με την σειρά που προσδιορίστηκαν κατά την δημιουργία της εγγραφής. Τέλος, η `getAttributeMap()` επιστρέφει την απεικόνιση μεταξύ του ονόματος των χαρακτηριστικών και της τιμής τους. Η μέθοδος αυτή είναι ιδιαίτερα χρήσιμη, καθ΄ ότι η επιστρεφομένη τιμή της μπορεί να χρησιμοποιηθεί άμεσα από την μέθοδο `setAssignments()` ενός αντικειμένου `GEPEvaluator`, στην περίπτωση που τα τερματικά σύμβολα του αντισώματος τύπου-ΓΠΕ έχουν την ίδια ονομασία με τα χαρακτηριστικά του συνόλου δεδομένων.

A.4.2 Σύνολα δεδομένων και δρομείς

Ένα σύνολο δεδομένων ορίζεται από την κλάση `DataSet`. Εσωτερικά, ένα αντικείμενο τύπου `DataSet` διατηρεί συνήθως μία λίστα από αντικείμενα `RowData`, τα οποία έχουν κοινά χαρακτηριστικά. Το σύνολο των χαρακτηριστικών του συνόλου δεδομένων ορίζεται κατά την κατασκευή του, ενώ προαιρετικά, για λόγους αποδοτικότητας, μπορεί να δίνεται και μία αρχική εκτίμηση για το μέγεθος του συνόλου δεδομένων, έτσι ώστε να παραχωρείται εξ΄ αρχής μία περιοχή μνήμης για τα δεδομένα. Νέες εγγραφές προστίθενται στο τέλος του συνόλου δεδομένων μέσω της μεθόδου `addRecord()`. Αυτό που πρέπει να παρατηρήσει κανείς, είναι ότι η μέθοδος αυτή λαμβάνει ως όρισμα μόνο τις τιμές των χαρακτηριστικών τις εγγραφής· η μετατροπή του σε αντικείμενο `RowData` είναι υπόθεση της κλάσης `DataSet`.

```
import jaif.dm.DataSet;

public DataSet(String[] attnames);
public DataSet(int initCap, String[] attnames);

public void addRecord(double[] rowData);
```

Με το σύνολο δεδομένων συσχετίζεται πάντοτε και ένας *δρομέας (cursor)*. Ο δρομέας μπορεί να μετακινείται μέσα στο σύνολο δεδομένων και να επιτελεί διάφορες ενέργειες όπως είναι η προσθήκη, διαγραφή ή ανάκτηση μίας εγγραφής. Κάθε σύνολο δεδομένων, επομένως, θα πρέπει να παρέχει και ένα δρομέα για να είναι δυνατή η διαχείριση των εγγραφών του από τον χρήστη. Οι λειτουργίες ενός δρομέα ορίζονται από την διεπιφάνεια `Cursor` η οποία ορίζει 8 μεθόδους.

```
import jaif.dm.Cursor;

public boolean  hasNext();
public boolean  hasPrevious();
public RowData  nextRecord();
public RowData  previousRecord();
public void     addRecord(RowData elem);
public RowData  removeRecord();
public int      getPosition();
public void     setPosition(int pos);
```

Οι πρώτες 4 μέθοδοι φροντίζουν για την ασφαλή μετακίνηση του δρομέα εντός του συνόλου δεδομένων. Η μετακίνηση του δρομέα βασίζεται στις ίδιες αρχές

που βασίζεται η μετακίνηση ενός `ListIterator`. Η `nextRecord()` μετακινεί τον δρομέα μία θέση και επιστρέφει το αντικείμενο που μόλις πέρασε, ενώ αντιστρόφως λειτουργεί η `previousRecord()`. Οι δύο αυτές μέθοδοι μπορεί, γενικά, να εγείρουν μία εξαίρεση `NoSuchElementException`, στην περίπτωση που ο δρομέας μετακινηθεί εκτός των ορίων του συνόλου δεδομένων. Επιπλέον, για τις δύο αυτές μεθόδους θα πρέπει να ισχύουν:

- Εάν ο δρομέας βρίσκεται στην τελευταία εγγραφή του συνόλου δεδομένων, δηλαδή η `hasNext()` επιστρέφει `false`, τότε η επόμενη κλήση στην `nextRecord()` θα πρέπει να προκαλεί σφάλμα.
- Εάν ο δρομέας βρίσκεται στην πρώτη εγγραφή του συνόλου δεδομένων, δηλαδή η `hasPrevious()` επιστρέφει `false`, τότε η επόμενη κλήση στην `previousRecord()` θα πρέπει να προκαλεί σφάλμα.

Επομένως, μπορεί κανείς να διατρέξει ένα ολόκληρο σύνολο δεδομένων χρησιμοποιώντας τον κώδικα:

```
for (Cursor cur = dataSet.cursor(); cur.hasNext(); )
    use(cur.nextRecord());
```

Οι μέθοδοι `addRecord()` και `removeRecord()` προσθέτουν και αφαιρούν την εγγραφή που βρίσκεται στην τρέχουσα θέση του δρομέα. Για τις δύο αυτές μεθόδους ισχύουν οι αντίστοιχες συμβάσεις που ισχύουν για τις μεθόδους `add()` και `remove()` ενός `ListIterator`:

- Η νέα εγγραφή εισάγεται ακριβώς πριν από την εγγραφή που θα επιστρεφόταν από την `nextRecord()` και ακριβώς μετά από την εγγραφή που θα επιστρεφόταν από την `previousRecord()`. Με άλλα λόγια, η `addRecord()` δεν θα πρέπει να επηρεάζει την επόμενη κλήση της `nextRecord()`.
- Η εγγραφή που διαγράφεται είναι πάντα η εγγραφή που επιστράφηκε από την τελευταία κλήση της `nextRecord()` ή της `previousRecord()`, όποια έγινε τελευταία.

Τέλος, η μέθοδος `setPosition()` θέτει ρητά την θέση του δρομέα μέσα στο σύνολο δεδομένων, ενώ η `getPosition()` ανακτά την θέση του δρομέα. Στο Πρόγραμμα Α.5 παρουσιάζεται η υλοποίηση του δρομέα της κλάσης `DataSet`, όπου τα δεδομένα βρίσκονται στην μνήμη σε ένα πίνακα μεταβλητού μεγέθους (`ArrayList`). Μέσω των μεθόδων `cursor()` ο χρήστης μπορεί να ανακτήσει ένα δρομέα για το συγκεκριμένο σύνολο δεδομένων.

```
package jaif.dm;

import java.util.ArrayList;
import java.util.NoSuchElementException;

public class DataSet {

  private ArrayList<RowData>    data;
  // More declarations come here...

  // Constructors and the rest methods are omitted...

  public Cursor cursor() { return( cursor(0) ); }
```

```
public Cursor cursor(int pos) {

  return( new CursorImpl(pos) );
}

/**
 *    A cursor for this DataSet.
 */
private class CursorImpl implements Cursor {

  /** The current position in the dataset. */
  private int position;
  /** Index of the last returned element. */
  private int returnIndex;

  public CursorImpl(int initPos) { position = initPos; }

  public int getPosition() { return position; }

  public void setPosition(int pos) { position = pos; }

  public boolean hasNext() {

    return( position < data.size() );
  }

  public boolean hasPrevious() {

    return( position-1 >= 0 );
  }

  public RowData nextRecord() {

    try {
      returnIndex = position;
      return( data.get(position++) );
    } catch (IndexOutOfBoundsException e) {
      throw new NoSuchElementException("invalid index: " +
                                       position);
    }
  }

  public RowData previousRecord() {

    try {
      returnIndex = --position;
      return( data.get(position) );
    } catch (IndexOutOfBoundsException e) {
      throw new NoSuchElementException("invalid index: " +
                                       position);
    }
  }
```

```
    public void addRecord(RowData elem) {

      data.add(position++, elem);
    }

    public RowData removeRecord() {

      return( data.remove(returnIndex) );
    }
  }
}
```

Πρόγραμμα A.5: Η υλοποίηση του δρομέα της κλάσης `DataSet` του JAIF.

A.4.3 Κλάσεις δεδομένων

Οι κλάσεις δεδομένων απεικονίζονται χρησιμοποιώντας την έννοια των αντιγόνων κλάσεων δεδομένων (ΑΚΔ), που παρουσιάστηκε στην §5.4.1. Για κάθε κλάση δεδομένων ορίζεται ένα ξεχωριστό ΑΚΔ, το οποίο περιέχει όλα τα μέλη της κλάσης. Ένα ΑΚΔ στο JAIF αντιπροσωπεύεται από την κλάση `DataClassAntigen`, η οποία επεκτείνει την αφηρημένη κλάση `Antigen`. Το μόνο στοιχείο που απαιτεί η κλάση `Antigen` από τις υποκλάσεις της, είναι να οριστεί το μήκος του αντιγόνου. Έτσι, το μήκος ενός ΑΚΔ ορίζεται ως το πλήθος των μελών της κλάσης δεδομένων που αντιπροσωπεύει. Εάν μάλιστα οριστεί και κάποιος κατάλληλος εκτιμητής ποιότητας σύνδεσης, ο οποίος θα μπορεί να αναγνωρίζει το ΑΚΔ και να αξιολογεί την ποιότητα σύνδεσης του με κάποιο συγκεκριμένο τύπο αντισώματος, τότε η έννοια των ΑΚΔ θα έχει ενσωματωθεί πλήρως στην δομή και την φιλοσοφία του JAIF. Αυτό πρακτικά σημαίνει, ότι ο ίδιος αλγόριθμος που επιλύει ένα πρόβλημα αναγνώρισης προτύπων, όπως είναι ο `ImmuneAlgorithm`, μπορεί να χρησιμοποιηθεί ως έχει, χωρίς *καμία* αλλαγή, και για την επίλυση ενός προβλήματος εξόρυξης από δεδομένα. Το παράδειγμα των ΑΚΔ καταδεικνύει με πολύ εύγλωττο τρόπο την μεγάλη ευελιξία που παρέχει η βασική προγραμματιστική διεπιφάνεια του Java Artificial Immune Framework, η οποία παρέχει την δυνατότητα στα διάφορα εργαλεία ανοσολογικής μηχανικής να επιτελούν τις διάφορες λειτουργίες τους, εντελώς ανεξάρτητα από τις υλοποιήσεις των διαφόρων προγραμματιστικών στοιχείων που χρησιμοποιούν.

Ένα `DataClassAntigen` συσχετίζεται μέ ένα συγκεκριμένο `DataSet` και διατηρεί ένα πίνακα με τους αριθμούς των εγγραφών που ανήκουν στην κλάση που αντιπροσωπεύει. Αυτή η υλοποίηση προϋποθέτει ότι η σειρά των εγγραφών στο σύνολο δεδομένων δεν θα πρέπει να αλλάξει, αφότου δημιουργηθεί ένα ΑΚΔ που σχετίζεται με αυτό το σύνολο δεδομένων. Η υλοποίηση αυτή είναι σχετικά περιοριστική και ενδεχομένως όχι τόσο αποδοτική, οπότε μπορεί στο μέλλον να αλλάξει.

```
import jaif.dm.*;

public DataClassAntigen(DataSet dataSet,
                        String classId);
```

Πέρα από το σύνολο δεδομένων, κατά την κατασκευή ενός `DataClassAntigen` θα πρέπει να προσδιορίζεται και ένα αναγνωριστικό, *classId*, για την κλάση

δεδομένων που αυτό αντιπροσωπεύει. Το αναγνωριστικό αυτό θα πρέπει να είναι μοναδικό για κάθε διαφορετική κλάση δεδομένων.

Η κλάση `DataClassAntigen` παρέχει επίσης ένα σύνολο μεθόδων για τον χειρισμό και την ανάκτηση των εγγραφών της κλάσης δεδομένων.

```
import jaif.dm.*;

public boolean      addMember(int dataSetIndex);
public boolean      isMember(int dataSetIndex);
public int[]        getMembers();
public RowData[]    getClassData();
```

Η μέθοδος `addMember()` προσθέτει την εγγραφή του συνόλου δεδομένων με αριθμό *dataSetIndex* στην κλάση δεδομένων, ενώ η μέθοδος `isMember()` ελέγχει εάν μία εγγραφή με ένα συγκεκριμένο αριθμό ανήκει στην κλάση δεδομένων. Τέλος, οι `getMembers()` και `getClassData()` επιστρέφουν τους αριθμούς των εγγραφών της κλάσης δεδομένων, είτε τις ίδιες τις εγγραφές.

A.4.4 Κανόνες και σύνολα κανόνων

Απαραίτητο συστατικό της εξόρυξης από δεδομένα είναι ο ορισμός της αναπαράστασης των κανόνων. Για την αναπαράστασή τους, το πακέτο `jaif.dm` παρέχει μία διεπιφάνεια την οποία θα πρέπει να υλοποιούν οι κανόνες που χρησιμοποιούνται κατά την εξόρυξη από τα δεδομένα. Η διεπιφάνεια αυτή ορίζει 4 μεθόδους.

```
import jaif.Antibody;
import jaif.dm.DataRule;

public String   getClassId();
public String   getExpression();
public double   getFitness();
public Antibody getRuleImpl();
```

Η πρώτη μέθοδος θα πρέπει να επιστρέφει το αναγνωριστικό της κλάσης στην οποία αναφέρεται ο κανόνας, η δεύτερη μία αναγνώσιμη αναπαράσταση του κανόνα, η τρίτη ένα μέτρο ποιότητας του κανόνα, ενώ η τέταρτη θα πρέπει να επιστρέφει την υλοποίηση του κανόνα στη μορφή ενός αφηρημένου αντισώματος.

A.5 Μελλοντικές βελτιώσεις και επεκτάσεις

Στην παρούσα του έκδοση το Java Artificial Immune Framework παρέχει ήδη ένα μεγάλο φάσμα λειτουργιών, όπως έγινε φανερό από την περιγραφή που προηγήθηκε στο Παράρτημα αυτό. Εξακολουθούν, ωστόσο, να υπάρχουν στοιχεία ή λειτουργίες, που χρήζουν βελτίωσης ή εμπλουτισμού. Ανάμεσα στις μελλοντικές βελτιώσεις βρίσκεται η τροποποίηση των μεθόδων επιλογής αντισωμάτων της κλάσης `AntibodyPool`, έτσι ώστε η επιλογή των καλυτέρων ή των χειροτέρων αντισωμάτων να γίνεται σε χρόνο $O(n)$ και όχι $O(n \lg n)$, όπως συμβαίνει τώρα. Αυτό θα επιτευχθεί αποφεύγοντας την ταξινόμηση των αντισωμάτων· στην θέση της θα χρησιμοποιείται επαναληπτικά ένας αλγόριθμος $O(n)$ επιλογής του

k-οστού καλυτέρου αντισώματος (Cormen et al., 2001). Μικρές βελτιώσεις και επεκτάσεις μπορούν να γίνουν και στο πακέτο `jaif.dm`, έτσι ώστε να βελτιωθεί η υποστήριξη της εξόρυξης γνώσης από δεδομένα.

Παρ΄ όλα αυτά το κυριώτερο πρόβλημα του JAIF είναι, ότι προς το παρόν δεν παρέχει κάποιον ενοποιημένο και καλά καθορισμένο τρόπο, τόσο για την καταγραφή διαφόρων συμβάντων (event logging), όσο και για την εύκολη και πλήρως παραμετροποιήσιμη αρχικοποίηση των αλγορίθμων που προσφέρει. Επομένως, ως άμεση προοπτική ανάπτυξης τίθεται η ενσωμάτωση των δύο αυτών λειτουργιών στην υποδομή του JAIF, έτσι ώστε να γίνει ακόμα πιο φιλικό στον τελικό χρήστη-προγραμματιστή που θα θελήσει να το ενσωματώσει σε ένα γενικότερο και μεγαλύτερο σύστημα λογισμικού. Τέλος, άλλο ένα σημείο που χρήζει αναθεώρησης είναι το σύστημα εξαιρέσεων, έτσι ώστε να γίνει ακόμα πιο ευέλικτο._

Βιβλιογραφία

I. Aleksander and H. Morton. *An Introduction to Neural Computing*. Chapman and Hall, 1990.

C. Austin. J2SE 5.0 in a Nutshell, May 2004. URL http://java.sun.com/developer/technicalArticles/releases/j2se15.

H. Bersini. The immune and the chemical crossover. *IEEE Transactions on Evolutionary Computation*, 6(3):306–313, June 2002.

G. Bracha. Generics in the Java Programming Language, July 2004. URL http://java.sun.com/j2se/1.5.0/docs/guide/language.

D. W. Bradley and A. M. Tyrrell. Immunotronics–novel finite-state-machine architectures with built-in self-test using self-nonself discrimination. *IEEE Transactions on Evolutionary Computation*, 6:227–238, June 2002.

F. M. Burnet. *The Clonal selection theory of acquired immunity*. Vanderbilt Univ. Press, Nashville TN, 1959.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, second edition, 2001.

N. L. Cramer. A representation for the adaptive generation of simple programs. In *International Conference on Genetic Algorithms and Their Applications.*, pages 183–187, July 1985.

C. Darwin. *The Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life*. Murray, London, 1859.

D. Dasgupta. Artificial neural networks and artificial immune systems: Similarities and differences, 1997.

D. Dasgupta. *Artifical Immune Systems and their Applications*. Springer Verlag, Berlin, 1998.

D. Dasgupta and F. González. An immunity-based technique to characterize intrusions in computer networks. *IEEE Transactions on Evolutionary Computation*, 6:281–291, June 2002.

L. N de Castro and F. J. Von Zuben. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6:239–251, June 2002.

C. Ferreira. Gene Expression Programming: A new adaptive algorithm for solving problems. *Complex Systems*, 13(2):87–129, 2001α.

C. Ferreira. GEP tutorial. WSC6 tutorial, September 2001β.

L. M. Gambardella and M. Dorigo. Ant colonies for the traveling salesman problem, March 17 1997.

P. K. Harmer, P. D. Williams, G. H. Gunsch, and G. B. Lamont. An artificial immune system architecture for computer security applications. *IEEE Transactions on Evolutionary Computation*, 6:252–280, June 2002.

S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, New Jersey, USA, 1999.

J. H. Holland. *Adaptation in natural artificial systems*. University of Michigan Press, Ann Arbor, 1975.

C. S. Horstmann and G. Cornell. *Core Java 2*, volume I–Fundamentals. Sun Microsystems Press, 2001.

C. S. Horstmann and G. Cornell. *Core Java 2*, volume II–Advanced Features. Sun Microsystems Press, 2002.

J. E. Hunt and D. E. Cooke. Learning using an artificial immune system. *Journal of Network and Computer Applications*, 19:189–212, 1996.

J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks (ICNN'95)*, volume 4, pages 1942–1947, Perth, Western Australia, November-December 1995. IEEE.

J. R. Koza. *Genetic programming: On the programming of computers by natural selection*. MIT Press, Cambridge, Mass., 1992.

L. Lamport. *LaTeX: A Document Preparation System (User's guide and Reference Manual)*. Addison Wesley, second edition, 1994.

R. P. Lippman. An introduction to computing with neural nets. *Computer Architecture News ACM*, 16(1):7–25, March 1988.

W. S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

R. Z. Michalski and K. A. Kaufman. A measure of description quality for data mining and its implementation in the AQ18 Learning System. In *International ICSC Symposium on Advances in Intelligent Data Analysis (AIDA)*, June 1999.

T. M. Mitchell. *Machine learning*. McGraw Hill, New York, US, 1996.

F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, and C. Rowley. *The LaTeX Companion: Tools and Techniques for Computer Typesetting*. Addison Wesley, second edition, 2004.

D. J. Newman, S. Hettich, C. L. Blake, and C. Z. Merz. UCI repository of machine learning databases, 1998. URL http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html.

M. C. Nussenzweig. Immune receptor editing: revise and select. *Cell*, 95(7):875–878, December 1998.

A. S. Perelson and G. F. Oster. Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-non-self discrimination. *Journal of Theoretical Biology*, 81(4):645–670, December 1979.

R. J. Roiger and M. W. Geatz. *Data Mining: A tutorial-based primer*. Addison Wesley (International Edition), 2003.

F. Rosenblatt. The perceptron: A probabilistic model for information storage in the brain. *Psychological review*, 65:386–408, 1958.

F. Rosenblatt. The perceptron: A perceiving and recognizing automaton (project PARA). Technical Report 85-460-1, Cornell Aeronautical Laboratory, January 1957.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagation. *Nature*, 323(99):533–536, 1986.

C. E Shannon and W. Weaver. *The mathematical theory of communication*. University of Illinois Press, 1949.

S. B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. De Jong, S. Džeroski, S. E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R. S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, and J. Zhang. The MONK's Problems: A performance comparison of different learning algorithms. Technical Report CMU-CS-91-97, Carnegie Mellon University, December 1991.

F. J. Von Zuben and L. N. De Castro. Artificial Immune Systems: Part I - Basic theory and Applications, December 1999.

H. Xavier. Genetic algorithms for optimization: Background and applications, February 1997. URL http://www.epcc.ed.ac.uk/overview/publications/training_material/tech_watch/97_tw/techwatch-ga/.

C. Zhou, W. Xiao, T. M. Tirpak, and P. C. Nelson. Evolving accurate and compact classification rules with Gene Expression Programming. *IEEE Transactions on Evolutionary Computation*, 7(6):519–531, December 2003.

Β. Αλεπόρου-Μαρίνου, Αλ. Αργυροκαστρίτης, Αικ. Κομητοπούλου, Περ. Πιαλόγλου, και Β. Σγουρίτσα. *Βιολογία Θετικής Κατεύθυνσης Γ΄ Τάξης Ενιαίου Λυκείου*. Οργανισμός Εκδόσεων Διδακτικών Βιβλίων, Αθήνα, 1999.

Γ. Κοκολάκης και Ι. Σπηλιώτης. *Εισαγωγή στην Θεωρία Πιθανοτήτων και Στατιστική*. Εκδόσεις Συμεών, 1991.

Φ. Μπαρώνα-Μάμαλη, Ι. Μπότσαρης, Ι. Μπουρμπουχάκης, και Β. Περάκη. *Βιολογία Γενικής Παιδείας Γ΄ Τάξης Ενιαίου Λυκείου*. Οργανισμός Εκδόσεων Διδακτικών Βιβλίων, Αθήνα, 1999.

Σπ. Τζαφέστας. *Υπολογιστική Νοημοσύνη*. Εκδόσεις ΕΜΠ, Αθήνα, 2002.