

Αναγνώριση Προτύπων Θέμα Εξαμήνου

Αναγνωστόπουλος Βασίλης - Θάνος (ΜΠΠΛ 13002)
Βελισσαρίου Κυριάκος (ΜΠΠΛ 13005)

Αθήνα, 2014

Περιεχόμενα

1 Άσκηση 1η - Artificial Immune System	1
1.1 Εκφώνηση	1
1.2 Λύση	1
2 Άσκηση 2η - Swarm Intelligence	3
2.1 Εκφώνηση	3
2.2 Λύση	3
2.2.1 Εισαγωγή	3
2.2.2 Ορισμός - Αρχές	3
2.2.3 Κυριότεροι αλγόριθμοι σμήνους	4
3 Άσκηση 3η	8
3.1 Εκφώνηση	8
3.2 Λύση	8
3.2.1 Εισαγωγή	8
3.2.2 Εξέλιξη και Εξελικτική Υπολογιστική	8
3.2.3 Γενετικοί Αλγόριθμοι	10
4 Άσκηση 4η - Fuzzy Logic Systems	10
4.1 Εκφώνηση	10
4.2 Ασαφής Λογική	10
4.3 Συστήματα ασαφούς λογικής	11
5 Άσκηση 5η	13
5.1 Εκφώνηση	13
5.2 Λύση	14
Βιβλιογραφία	20

Κατάλογος σχημάτων

4.1 Παραδείγματα συναρτήσεων μελών [6]	12
4.2 Σύστημα ασαφούς λογικής [9, 23]	12
4.3 Σύστημα ασαφούς λογικής (2) [6]	13
5.1 Τα δεδομένα εκφράσεων προσώπου	17
5.2 Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την μέση σύνδεση και την ευκλείδεια απόσταση	18
5.3 Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την μέση σύνδεση και την απόσταση mahalanobis	18
5.4 Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την μέση σύνδεση και την απόσταση manhattan	18
5.5 Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την μονή σύνδεση και την ευκλείδεια απόσταση	19
5.6 Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την πλήρη σύνδεση και την ευκλείδεια απόσταση	19

Κατάλογος αλγορίθμων

1	Βασικός αλγόριθμος ΤΑΣ [6]	2
2	Ψευδοκώδικας αλγόριθμου pso	5
3	Ψευδοκώδικας αλγόριθμου aco	7
4	Γενική περίπτωση Εξελικτικού Αλγόριθμου	9

Κατάλογος πηγαίου κώδικα

1	Ο κώδικας του matlab	14
---	----------------------	----

1 Άσκηση 1η - Artificial Immune System

1.1 Εκφώνηση

Να γίνει πλήρης βιβλιογραφική έρευνα με βάση τις λέξεις κλειδιά "Artificial Immune System".

1.2 Λύση

Το φυσικό ανοσοποιητικό σύστημα είναι ένα πολύπλοκο σύστημα για την λειτουργία του οποίου έχουν γραφτεί αρκετές δημοσιεύσεις για το πως λειτουργεί [16]. Εκτός από την ικανότητα του να καταπολεμά ξένα κύτταρα και μολύνσεις προς τον οργανισμό, διαθέτει μνήμη, μία ιδιαίτερα σημαντική ιδιότητα, καθ' ότι του επιτρέπει να αναγνωρίζει και να αντιμετωπίζει πιο άμεσα σε μία εισβολή από παθογόνα, που έχουν προσβάλει και παλιότερα τον οργανισμό.

Ένα τεχνητό ανοσοποιητικό σύστημα (ΤΑΣ - αγγλ. Artificial Immune System) μοντελοποιεί την ικανότητα του φυσικού ανοσοποιητικού συστήματος των σπονδυλωτών να ανιχνεύει κύτταρα ξένα προς τον οργανισμό. Το αποτέλεσμα είναι ένα νέο υπολογιστικό μοντέλο το οποίο είναι ικανό να αναγνωρίζει πρότυπα και εφαρμόζεται κυρίως στην ανίχνευση ανωμαλιών [6, 13].

Ο ορισμός και η ανάπτυξη ενός πλήρους ΤΑΣ περιλαμβάνει, γενικά, μία πληθώρα θεμάτων, μεταξύ των οποίων είναι [6, 22]:

- υβριδικές δομές και αλγόριθμοι, οι οποίοι λαμβάνουν υπ' όψιν τους μηχανισμούς του ανοσοποιητικού συστήματος όπως την ανίχνευση ξένων προτύπων με μία ορισμένη συγγένεια και την αποθήκευση πληροφορίας και την επαναχρησιμοποίηση της.
- 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.

Χρησιμοποιώντας τα παραπάνω ο Engelbrecht παραθέτει τον βασικό αλγόριθμο για την δημιουργία ΤΑΣ (βλέπε αλγόριθμο 1).

Κάθε κομμάτι του αλγορίθμου αναλύεται πιο κάτω [6]:

Αρχικοποίηση C και καθορισμός D_T : Ο πληθυσμός C μπορεί να είναι είτε δημιουργημένος από τυχαία δημιουργημένα τεχνητά λεμφοκυττάρων ¹ (ΤΛ) ή από κάποια άλλη μέθοδο η οποία εξαρτάται από τον αλγόριθμο του ΤΑΣ.

Συνθήκη τερματισμού του while: Στα περισσότερα μοντέλα των ΤΑΣ, η συνθήκη τερματισμού βασίζεται στην σύγκλιση του πληθυσμού των ΤΛ ή από ένα συγκεκριμένο αριθμό επαναλήψεων.

Επιλογή του υποσυνόλου S των ΤΛ: Το υποσύνολο S μπορεί να είναι ολόκληρο το σύνολο P ή ένα τυχαίος αριθμός ΤΛ από το P .

¹Το λεμφοκύτταρο αποτελεί είδος λευκού αιμοσφαιρίου το οποίο το συναντιόνται στο φυσικό ανοσοποιητικό σύστημα και είναι επιφορτισμένα με την άμυνα του οργανισμού έναντι σε λοιμώξεις [17].

Αλγόριθμος 1 Βασικός αλγόριθμος ΤΑΣ [6]

Αρχικοποίηση ενός σύνολου τεχνικών λεμφοκυττάρων (ΤΛ) ως πληθυσμός C ;
Καθορισμός των προτύπων των αντιγόνων ως σύνολο εκπαίδευσης D_T ;
while κάποια συνθήκη τερματισμού είναι αναληθείς do
 for κάθε πρότυπο αντιγόνου $z_p \in D_T$ do
 Επιλογή ενός υποσυνόλου ΤΛ για έκθεση στο z_p , σαν πληθυσμός $S \leq C$;
 for για κάθε ΤΛ $x_i \in S$ do
 Υπολογισμός την ομοιότητα του αντιγόνου μεταξύ z_p, x_i ;
 end
 Επιλογή ενός υποσυνόλου ΤΛ που έχουν την μεγαλύτερη ομοιότητα
 αντιγόνων σαν πληθυσμός $H \leq S$;
 Προσαρμογή των ΤΛ με κάποια μέθοδο επιλογής, με βάση την
 υπολογισμένη ομοιότητα και/ή την ομοιότητα του δικτύου
 των ΤΛ στο H ;
 Ανανέωση του βαθμού ομοιότητας των ΤΛ στο H ;
 end
end

Υπολογισμός της ομοιότητας του αντιγόνου: Η ομοιότητα του αντιγόνου (αγγλ. antigen affinity) είναι η μέτρηση της "συγγένειας" που υπάρχει μεταξύ των ΤΛ και των προτύπων των αντιγόνων.

Επιλογή του υποσυνόλου H των ΤΛ: Σε κάποια από τα μοντέλα των ΤΑΣ, η επιλογή της καλύτερης ομοιότητας των ΤΛ βασίζεται σε κάποιο κατώφλι ομοιότητας. Έτσι το υποσύνολο H μπορεί να είναι ολόκληρο το S , αναλόγως ποιο είναι το κατώφλι ομοιότητας.

Υπολογισμός της ομοιότητας του δικτύου: Είναι η μέτρηση της ομοιότητας μεταξύ δύο ΤΛ. Τα διάφορα μέτρα ομοιότητας ενός δικτύου είναι τα ίδια με εκείνα των αντιγόνων.. Ένα καθορισμένο κατώφλι ομοιότητας προσδιορίζει αν δύο ή περισσότερα ΤΛ συνδέονται για να σχηματίσουν ένα δίκτυο.

Ανανέωση του βαθμού ομοιότητας των ΤΛ στο H : Είναι η διαδικασία με την οποία τα ΤΛ ωριμάζουν. Η διαδικασία ωρίμανσης αλλάζει ανάλογα με το μοντέλο του ΤΑΣ.

Έχουν προταθεί αρκετοί αλγόριθμοι για την επίλυση των ΤΑΣ όπως [6]:

το κλασσικό μοντέλο: σε αυτό το μοντέλο το ΤΑΣ "εκπαιδεύει" τα ΤΛ σε ένα σύνολο προτύπων του εαυτού του έτσι ώστε να είναι "αυτο-ανεκτικά", δηλαδή να έχουν την ικανότητα να αναγνωρίζουν τα πρότυπα μεταξύ του εαυτού και του μη εαυτού [6]. Βασίζεται στα Τ-λεμφοκύτταρα του φυσικού ανοσοποιητικού συστήματος τα οποία αναγνωρίζουν τα κύτταρα του οργανισμού και επιτίθενται μόνο στα ξένα κύτταρα. Ένα από τα μοντέλα των ΤΑΣ που βασίζονται στο κλασσικό μοντέλο είναι το μοντέλο της αρνητικής επιλογής.

το μοντέλο επιλογής του κλώνου: σε αυτό το μοντέλο των ΤΑΣ η επιλογή ενός συνόλου ΤΛ γίνεται με βάση εκείνα τα οποία έχουν τον υψηλότερο

βαθμό ομοιότητας μαζί με ένα πρότυπο μη εαυτού. Έπειτα τα επιλεγμένα ΤΛ κλωνοποιούνται και μεταλλάσσονται σε μία προσπάθεια να έχουν υψηλότερη ομοιότητα με το πρότυπο μη εαυτού.

το μοντέλου του δικτύου: σε αυτό το μοντέλο των ΤΑΣ τα ΤΛ αλληλεπιδρούν μεταξύ τους έτσι ώστε να μάθει το ένα από το άλλο την μορφή του προτύπου μη εαυτού, με αποτέλεσμα να δημιουργείται ένα δίκτυο από ΤΛ.

το μοντέλο του κινδύνου: σε αυτό το μοντέλο, σε αντίθεση με το κλασσικό μοντέλο, το μοντέλο του κινδύνου διαφοροποιείται αναγνωρίζοντας το τί είναι επικίνδυνο και τί είναι μη-επικίνδυνο αντί να βρίσκει πρότυπα εαυτού και μη-εαυτού.

Τα τεχνητά ανοσοποιητικά συστήματα έχουν εφαρμοστεί με επιτυχία σε πολλές περιοχές, όπως στην ανίχνευση ανωμαλιών, στην ταξινόμηση δεδομένων, στην ανίχνευση ιών, στην αναγνώριση προτύπων κ.λ.π. [13, 6, 22].

2 Άσκηση 2η - Swarm Intelligence

2.1 Εκφώνηση

Να γίνει πλήρης βιβλιογραφική έρευνα με βάση τις λέξεις κλειδιά "Swarm Intelligence".

2.2 Λύση

2.2.1 Εισαγωγή

Στην παρούσα ερώτηση θα επιχειρηθεί μια σύντομη έκθεση η οποία θα αφορά τον όρο Νοημοσύνη Σμήνους (Swarm Intelligence). Η δομή που ακολουθείται είναι η εξής: Αρχικά, θα δοθεί ο ορισμός της Νοημοσύνης Σμήνους και θα αναφερθούν οι γενικές αρχές της. Στην συνέχεια, θα γίνει μια σύντομη επισκόπηση στις κατηγορίες αλγορίθμων νοημοσύνης σμήνους. Τέλος, στον επίλογο, θα αναφερθούν μια σειρά από προβλήματα, στην επίλυση των οποίων βρίσκουν εφαρμογή οι προαναφερθέντες αλγόριθμοι.

2.2.2 Ορισμός - Αρχές

Σύμφωνα με τους Gerardo Beni και Jim Wang [2], η Νοημοσύνη Σμήνους είναι η συλλογική συμπεριφορά αποκεντρωμένων αυτο-οργανωμένων συστημάτων, φυσικών ή τεχνητών.

Όπως είναι προφανές, τα φυσικά συστήματα νοημοσύνης σμήνους προϋπήρχαν των τεχνητών και μάλιστα, στάθηκαν η αφορμή για την δημιουργία των τελευταίων. Τα φυσικά συστήματα που μελετήθηκαν αρχικά περιελάμβαναν κοπάδια ψαριών, σμήνη πτηνών και μελισσών καθώς και αποικίες μυρμηγκιών. Αυτό το οποίο εντυπωσίασε του ερευνητές και που προσπάθησαν να εκμεταλλευτούν με την δημιουργία τεχνητών συστημάτων νοημοσύνης σμήνους, είναι η δυνατότητα των συστημάτων αυτών, να πετυχαίνουν πολύπλοκους στόχους ενώ τα ίδια είναι αποτελούμενα από απλές οντότητες περιορισμένων δυνατοτήτων. Ένα ακόμα εντυπωσιακό στοιχείο, σύμφωνα και με τον ορισμό που

δόθηκε, είναι ότι οι απλές αυτές οντότητες δεν κατευθύνονται από κάποια κεντρική οντότητα [1], αλλά δρουν αποκεντρωμένα. Για παράδειγμα, στις αποικίες μυρμηγκιών, τα μυρμηγκία είναι ικανά όχι μόνο να βρουν τον δρόμο, αλλά και το συντομότερο μονοπάτι προς την τροφή τους. Αυτό το πετυχαίνουν με την έκκριση μια φερομόνης. Συνεπώς, τα μυρμηγκία θα ακολουθούν μονοπάτια τα οποία έχουν έντονη ποσότητα φερομόνης για να τα διεγείρει. Τελικά, το συντομότερο μονοπάτι προς την τροφή θα συγκεντρώνει όλο και μεγαλύτερο αριθμό μυρμηγκιών, έτσι η φερομόνη σε αυτό το μονοπάτι θα είναι εντονότερη κοκ [4].

Στο επόμενο υποκεφάλαιο παρουσιάζονται κάποιοι από τους σημαντικότερους αλγόριθμους Νοημοσύνης Σμήνους οι οποίοι προήλθαν από μελέτη φαινομένων σαν αυτά που αναφέρθηκαν στην προηγούμενη παράγραφο.

2.2.3 Κυριότεροι αλγόριθμοι σμήνους

Αλγόριθμος PSO

Ο αλγόριθμος PSO είναι εμπνευσμένος από τον τρόπο με τον οποίο λειτουργούν τα σμήνη πουλιών και μελισσών για να προσανατολιστούν στον χώρο. Πρόκειται για μεταεωριστικό αλγόριθμο καθώς δεν κάνει υποθέσεις σχετικά με το πρόβλημα που πρόκειται να επιλυθεί. Οι αρχική σύλληψη του αλγορίθμου αποδίδεται στον Kennedy [8].

Ο αλγόριθμος βρίσκει πεδίο εφαρμογής σε προβλήματα βελτιστοποίησης τα οποία έχουν μεγάλους χώρους πιθανών λύσεων. Ωστόσο, ο PSO δεν εγγυάται την εύρεση της βέλτιστης λύσης, αλλά μια πολύ καλή προσέγγιση αυτής. Το γεγονός της εύκολης υλοποίησής του σε συνδιασμό με το ότι δεν απαιτεί το πρόβλημα προς βελτιστοποίηση να είναι παραγωγίσιμο [8] τον καθιστά μια καλή επιλογή για προβλήματα που μεταβάλλονται στον χρόνο ή ενέχουν θόρυβο.

Ο αλγόριθμος PSO υπάρχει σε πολλές παραλλαγές, ωστόσο η κύρια μορφή του είναι αυτή κατά την οποία αναζητείται το ολικό μέγιστο. Επιπλέον, σε ότι αφορά την τοπολογία του σμήνους, έχουμε ένα πλήρως συνδεδεμένο σμήνος, δηλαδή κάθε μέλος του σμήνους γνωρίζει τις διαθέσιμες πληροφορίες για όλα τα υπόλοιπα μέλη. Στον πίνακα 2 φαίνεται ψευδοκώδικας για τον γενικό αλγόριθμο PSO όπως αυτός ανακτήθηκε από τον ιστότοπο [11]. Ακολουθεί λεκτική περιγραφή του αλγορίθμου.

Αρχικά ορίζεται η λειτουργία της αρχικοποίησης του σμήνους. Δηλαδή, για κάθε σωματίδιο του σμήνους, καθώς και για κάθε συνιστώσα της λύσης δίνεται μια τυχαία τιμή μέσα στις επιτρεπτές τιμές. Είναι φανερό, ότι οι επιτρεπτές τιμές καθώς και οι συνιστώσες στις οποίες έχει αποκωδικοποιηθεί η λύση εξαρτώνται από την δομή του προβλήματος. Στην συνέχεια αρχικοποιείται η καλύτερη θέση κάθε σωματιδίου, η οποία είναι η τρέχουσα θέση καθώς ο αλγόριθμος δεν έχει ξεκινήσει. Τέλος, βρίσκεται το ολικό μέγιστο, δηλαδή η θέση η οποία έχει την μεγαλύτερη αξία. Η αξία δίνεται από την συνάρτηση f η δομή της οποίας εξαρτάται από την φύση του προβλήματος.

Αφού γίνει η αρχικοποίηση, ορίζουμε μια καθολική μεταβλητή η οποία δείχνει τον μέγιστο αριθμό των επαναλήψεων της εκτέλεσης του αλγορίθμου. Σε κάθε μια από αυτές τις επαναλήψεις γίνονται τα εξής: Για κάθε σωματίδιο ανανεώνεται η καλύτερή του θέση. Επίσης, ανανεώνεται το ολικό μέγιστο. Στην συνέχεια για κάθε σωματίδιο και κάθε διάσταση της λύσης υπολογίζεται

Αλγόριθμος 2 Ψευδοκώδικας αλγόριθμου pso

```
//Αρχικοποίηση;
for κάθε σωματίδιο  $i \in S$  do
  for κάθε διάσταση  $d \in D$  do
    //Αρχικοποίηση της ταχύτητας και της θέσης των σωματιδίων
     $x_{i,d} = Rnd(x_{min}, x_{max})$ 
     $u_{i,d} = Rnd(-u_{max}/3, u_{max}/3)$ 
  end
  //Αρχικοποίηση της καλύτερης θέσης κάθε σωματιδίου
   $pb_i = x_i$ 
  //Ενημέρωση της καθολικά καλύτερης θέσης
  if  $f(pb_i) < f(gb)$ 
    then  $gb = pb_i$ 
  fi
end
(Αρχικοποίηση)
while  $it < MAX\_ITERATIONS$  do
  for κάθε σωματίδιο  $i \in S$  do
    //Ενημέρωση της καλύτερης θέσης κάθε σωματιδίου
    if  $f(x_i) < f(pb_i)$ 
      then  $pb_i = x_i$ 
    fi
    //Ενημέρωση του καθολικού μέγιστου
    if  $f(pb_i) < f(gb)$ 
      then  $gb = pb_i$ 
    fi
  end
  //Ενημέρωση της ταχύτητας και της θέσης των σωματιδίων
  for κάθε σωματίδιο  $i \in S$  do
    for κάθε διάσταση  $d \in D$  do
       $u_{i,d} = u_{i,d} + C_1 * Rnd(0, 1) * [pb_{i,d} - x_{i,d}] + C_2 * Rnd(0, 1) * [gb_d - x_{i,d}]$ 
       $x_{i,d} = x_{i,d} + u_{i,d}$ 
    end
  end
  //Επόμενη επανάληψη
   $it = it + 1$ 
end
επέστρεψε  $gb$ 
```

η νέα ταχύτητα και η νέα θέση. Η νέα θέση προκύπτει από την προηγούμενη θέση συν την ταχύτητα. Σε ότι αφορά την ενημέρωση της ταχύτητας αυτή προκύπτει από το άθροισμα διαφορετικών συστατικών. Το πρώτο προκύπτει από την διαφορά της τρέχουσας θέσης από την καλύτερη θέση που έχει επισκεφθεί το συγκεκριμένο σωματίδιο. Το δεύτερο προκύπτει από την διαφορά της τρέχουσας θέσης του σωματιδίου από το ολικό μέγιστο. Παρατηρούμε ότι και τα δύο παραπάνω είναι πολλαπλασιασμένα από δύο επιταχυντές C_1 και C_2 αντίστοιχα, καθώς και από μια στοχαστική μεταβλητή η οποία λαμβάνει κάθε φορά μια τιμή ανάμεσα στο 0 και το 1. Η χρησιμότητα της στοχαστικής μεταβλητής έγκειται στην αποφυγή της παγίδευσης του αλγορίθμου σε τοπικό μέγιστο. Το τελευταίο συστατικό της στο άθροισμα είναι η τρέχουσα ταχύτητα. Τέλος, αυξάνεται ο μετρητής επαναλήψεων κατά 1.

Όταν ολοκληρωθεί ο αριθμός των επαναλήψεων που έχει οριστεί, ο αλγόριθμος τερματίζει και επιστρέφεται η τιμή που υπάρχει στην μεταβλητή gb , η οποία εκφράζει την καλύτερη λύση που έχει βρεθεί συνολικά.

Αλγόριθμος ACO

Ο αλγόριθμος ACO (Ant Colony Optimization) είναι εμπνευσμένος, όπως αναφέρθηκε παραπάνω, από τον τρόπο με τον οποίο τα μυρμήγκια βρίσκουν τον συντομότερο δρόμο προς την τροφή τους χρησιμοποιώντας τις φερομόνες. Δημιουργοί του αλγορίθμου είναι οι Dorigo and Birattari [5]. Η κυριότερη εφαρμογή του αλγορίθμου ACO βρίσκεται σε προβλήματα συνδυαστικής φύσης και πιο συγκεκριμένα σε εύρεση βέλτιστων μονοπατιών σε γράφους.

Σε μια πραγματική αποικία μυρμηγκιών, εάν ένα σύντομο μονοπάτι προς τροφή έχει βρεθεί με την διαδικασία που περιγράφηκε παραπάνω, τότε η πλειοψηφία των μυρμηγκιών θα κινείται σε αυτό. Εάν η τροφή τελειώσει τότε θα αρχίσουν να αναζητούνται άλλα μονοπάτια, δηλαδή άλλη βέλτιστη λύση και καθώς θα γίνεται αυτό οι φερομόνες στο προηγούμενο μονοπάτι θα εξασθενούν προσελκύνοντας όλο και λιγότερα μυρμήγκια. Εκμεταλλευόμενος αυτό ακριβώς το χαρακτηριστικό, ο αλγόριθμος ACO αποδεικνύεται ιδιαίτερα χρήσιμος σε προβλήματα γράφων των οποίων η τοπολογία αλλάζει δυναμικά στον χρόνο. Στον πίνακα 3 καθώς και στην παράγραφο που ακολουθεί περιγράφεται η λειτουργία του βασικού αλγορίθμου. Σημειώνεται πως εκτός από τον βασικό αλγόριθμο, υπάρχουν διάφορες παραλλαγές του όπως των Merkle et al. [10] και των Yaseen and Al-Slamy [18].

Στον αλγόριθμο, όπως αυτός περιγράφεται στον πίνακα 3[12] συμβαίνουν τα εξής: Αρχικά, για κάθε κόμβο αρχικοποιείται ο βαθμός ελκυστικότητάς του και η ορατότητά του η οποία είναι ένα είδος μετρικής, συνήθως η απόσταση. Στην συνέχεια κάθε μυρμήγκι επιλέγει τον επόμενο κόμβο τον οποίο θα επισκεφθεί προχωρώντας προς την κατασκευή μια πλήρους λύσης. Η επιλογή γίνεται βάση της εξίσωσης 1.

$$Prob(choose_available_edge) = \frac{\tau(e) \cdot n(e)}{\sum available_edges(\tau(e') \cdot n(e'))} \quad (1)$$

Όπως φαίνεται από την εξίσωση, όσο μεγαλύτερη είναι η τιμή της φερομόνης τόσο μεγαλύτερη είναι η πιθανότητα να επιλεγεί ο συγκεκριμένος κόμβος. Επίσης η πιθανότητα επιλογής μεγαλώνει όσο μικρότερη είναι η απόσταση του κόμβου. Από αυτό συμπεραίνουμε ότι η $n(e)$ εκφράζει το αντίστροφο της απόστασης.

Αλγόριθμος 3 Ψευδοκώδικας αλγόριθμου aco

```
Αρχικοποίηση της ελκυστικότητας  $\tau$  και της ορατότητας  $\eta$  για κάθε κόμβο;  
while  $i < MaxIterations$  do  
  for κάθε  $ant \in C$  do  
    Επιλογή, βάσει εξίσωσης, του επόμενο κόμβου μετακίνησης;  
    Καταχώρηση του κόμβου στην λίστα επισκεπτόμενων  
    κόμβων του μυρμηγκιού;  
    Επανάληψη μέχρι κάθε μυρμήγκι να έχει κατασκευάσει μια λύση;  
  end  
  for κάθε  $ant \in C$  do  
    Ενημέρωση της ελκυστικότητας κάθε κόμβου από την λύση;  
  end  
  if  $lb$  καλύτερη από  $gb$   
    then  $gb = lb$   
  fi  
   $i = i + 1$ ;  
end
```

Στην συνέχεια, και αφού όλα τα εικονικά μυρμήγκια έχουν καταλήξει σε κάποια λύση, για κάθε έναν από τους κόμβους που συμμετέχουν σε αυτή, ενημερώνεται η τιμή της φερομόνης (μέτρο της ελκυστικότητας του κόμβου) μέσω της εξίσωσης 2.

$$\tau(e) = \begin{cases} (1 - p) \cdot \tau(e), & \text{εάν ο κόμβος δεν έχει επισκεφθεί} \\ (1 - p) \cdot \tau(e) + \text{νέα φερομόνη}, & \text{εάν ο κόμβος έχει επισκεφθεί} \end{cases} \quad (2)$$

Η παράμετρος p , $0 < p < 1$ εκφράζει τον ρυθμό εξάτμισης, δηλαδή το πόσο γρήγορα μειώνεται η επίδραση της φερομόνης. Συνεπώς, όσο μεγαλύτερος είναι ο αριθμός, τόσο μεγαλύτερος είναι ο ρυθμός εξάτμισης γεγονός που αυξάνει την προσαρμοστικότητα του συστήματος. Αντίστοιχα στην περίπτωση που το p είναι μικρό. Ο όρος *νέα φερομόνη* συνήθως αντιστοιχεί σε έναν λόγο της μορφής $\frac{Q}{length_of_tour}$, όπου Q μια σταθερά ελκυστικότητας. Ο παρονομαστής αποτελεί το μέγεθος που κατά περίπτωση θέλουμε να βελτιστοποιήσουμε.

Στην συνέχεια ελέγχεται εάν το τοπικό μέγιστο που έχει βρεθεί αποτελεί καλύτερη λύση από την καθολική λύση μέχρι εκείνη την στιγμή. Εάν η συνθήκη ισχύει τότε η καθολική λύση ανανεώνεται. Τέλος, αυξάνεται ο αριθμός των επαναλήψεων κατά μια.

Εφαρμογές Αλγόριθμων Νοημοσύνης Σμήνους

Τα χαρακτηριστικά που συγκεντρώνουν οι αλγόριθμοι Νοημοσύνης Σμήνους, όπως αναλύθηκαν παραπάνω, κάνει την εφαρμογή τους ιδιαίτερα δημοφιλή σε σύγχρονα προβλήματα.

Γενικότερα, οι παραπάνω αλγόριθμοι εφαρμόζονται σε προβλήματα που ενέχουν κάποιου είδους βελτιστοποίηση. Μια σημαντική εφαρμογή τους είναι στο πεδίο των δικτύων τηλεπικοινωνιών, καθώς χρησιμοποιούνται για την εύρεση συντομότερων διαδρομών για την μεταφορά των πακέτων. Επίσης, χρη-

σιμοποιούνται για προσομοίωση οδικών δικτύων προκειμένου να αναδυθούν κυκλοφοριακά πρότυπα, καθώς και σε πλήθος στρατιωτικών εφαρμογών [7].

Τέλος, ένας τομέας με σημαντικότερη διεξόδυση την Νοημοσύνης Σμήνους αποτελεί το πεδίο της Ρομποτικής Σμήνους (Swarm Robotics). Η Ρομποτική Σμήνους αναφέρεται στην εφαρμογή τεχνικών Νοημοσύνης Σμήνους για την ανάλυση δραστηριοτήτων που προέρχονται από ρομποτικές συσκευές οι οποίες μπορούν να επηρεάσουν το περιβάλλον τους μέσω ευφυών μηχανισμών λήψης αποφάσεων [7].

Όπως γίνεται αντιληπτό, η Νοημοσύνη Σμήνους αποτελεί ένα σημαντικό πεδίο της Τεχνητής Νοημοσύνης το οποίο θα συνεχίζει να συγκεντρώνει ερευνητικές προσπάθειες λόγω της ευκολίας (σε σχέση με άλλες μεθόδους) με την οποία μπορεί να λύνει δύσκολα μη δομημένα και δυναμικά προβλήματα συνδυαστικής φύσεως.

3 Άσκηση 3η

3.1 Εκφώνηση

Να γίνει πλήρης βιβλιογραφική έρευνα με βάση τις λέξεις κλειδιά "Evolutionary Computing" και "Genetic Algorithms".

3.2 Λύση

3.2.1 Εισαγωγή

Η δομή της απάντησης έχει ως εξής: Αρχικά, θα γίνει η εισαγωγή της έννοιας της εξέλιξης και στην συνέχεια θα συνδυαστεί με αυτή της Εξελικτικής Υπολογιστικής (Evolutionary Computing). Στην συνέχεια, θα παρουσιαστούν τα βασικότερα χαρακτηριστικά της Εξελικτικής Υπολογιστικής και θα γίνει σύντομη αναφορά στους κυριότερους αλγόριθμους που την απαρτίζουν.

Στο επόμενο μέρος, θα αναλυθούν περαιτέρω οι Γενετικοί Αλγόριθμοι (Genetic Algorithms) οι οποίοι αποτελούν ένα υποσύνολο των αλγόριθμων Εξελικτικής Υπολογιστικής. Τέλος, θα γίνει μια επισκόπηση στην παρούσα κατάσταση σε ότι αφορά το πεδίο της Εξελικτικής Υπολογιστικής.

3.2.2 Εξέλιξη και Εξελικτική Υπολογιστική

Σύμφωνα με τον Darwin [3], η εξέλιξη είναι μια διαδικασία βελτιστοποίησης, στόχος της οποίας είναι η βελτίωση της ικανότητας ενός οργανισμού να επιβιώσει σε ένα δυναμικά μεταβαλλόμενο και ανταγωνιστικό περιβάλλον. Σημειώνεται πως ο όρος εξέλιξη μπορεί να αναφέρεται σε συστήματα διαφορετικής φύσης όπως βιολογικά, κοσμικά, οργανικά, χημικά κλπ. Βασικό χαρακτηριστικό της εξέλιξης είναι η έννοια της φυσικής επιλογής (natural selection). Σύμφωνα με την φυσική επιλογή, σε ένα ανταγωνιστικό περιβάλλον κάθε οργανισμός ανταγωνίζεται για την επιβίωσή του. Οι οργανισμοί με τα καλύτερα χαρακτηριστικά είναι πιθανότερο να επιβιώσουν και να αναπαραχθούν και συνεπώς να κληρονομήσουν τα χαρακτηριστικά που τους έκαναν ισχυρούς στους απογόνους τους.

Η Εξελικτική Υπολογιστική αναφέρεται σε υπολογιστικές μεθόδους επίλυσης προβλημάτων οι οποίες χρησιμοποιούν διαδικασίες εμπνευσμένες από την θεωρία της εξέλιξης όπως η φυσική επιλογή, η επικράτηση του ισχυρού και η αναπαραγωγή [6].

Σύμφωνα με τον Engelbrecht [6] και όπως φαίνεται στον πίνακα 4, η γενική περίπτωση Εξελικτικού Αλγόριθμου λειτουργεί ως εξής: Αρχικά ορίζεται ένας μετρητής γενεών. Στην συνέχεια, δημιουργείται και αρχικοποιείται ο πληθυσμός πλήθους n_s . Κάθε στοιχείο του πληθυσμού εκφράζει μια λύση n_x διαστάσεων. Ο τρόπος απεικόνισης της λύσης αποτελεί σχεδιαστική επιλογή του προγραμματιστή και εξαρτάται από το εκάστοτε πρόβλημα προς αντιμετώπιση. Όσο το κριτήριο τερματισμού δεν πληρούται, υπολογίζεται μια συνάρτηση προσαρμοστικότητας για κάθε μέλος του πληθυσμού. Στην συνέχεια επιτελείται η λειτουργία της αναπαραγωγής για την δημιουργία απογόνων. Μετά επιλέγονται τα μέλη της νέας γενιάς βάσει συγκεκριμένης συνάρτησης. Τα μέλη του πληθυσμού των οποίων η συνάρτηση προσαρμοστικότητας έχει υψηλή τιμή ενέχουν και μεγαλύτερη πιθανότητα επιλογής. Τέλος αυξάνεται ο μετρητής γενεών.

Αλγόριθμος 4 Γενική περίπτωση Εξελικτικού Αλγόριθμου

```
Έστω  $t = 0$  ο μετρητής γενεών  
Αρχικοποίηση  $n_x$ -διάστατου πληθυσμού αποτελούμενο από  $n_s$  μέλη  
while κριτήριο τερματισμού ΨΕΥΔΕΣ do  
    Υπολογισμός τιμής της  $f(x_i(t))$  για κάθε μέλος  $x_i(t)$   
    Λειτουργία αναπαραγωγής  
    Επιλογή νέου πληθυσμού  
    Αύξηση μετρητή γενεών  $t = t + 1$   
end
```

Ανάλογα με τον τρόπο με τον οποίο υλοποιείται ένας εξελικτικός αλγόριθμος μπορεί να ταξινομηθεί στους παρακάτω τομείς [6]:

- Γενετικοί Αλγόριθμοι (Genetic Algorithms)
- Γενετικός Προγραμματισμός (Genetic programming)
- Εξελικτικός Προγραμματισμός (Evolutionary Programming)
- Εξελικτική Στρατηγική (Evolution Strategies)
- Διαφορική Εξελικτική (Differential Evolution)
- Εξελικτική Κουλτούρας (Cultural Evolution)
- Συν Εξέλιξη (Co-evolution)

Τα βασικά σχεδιαστικά θέματα τα οποία πρέπει να αποφασίσει ο σχεδιαστής ενός εξελικτικού αλγορίθμου είναι τα εξής:

- **Αναπαράσταση λύσης:** Η αναπαράσταση της λύσης γίνεται με το χρωμόσωμα. Κάθε στοιχείο του χρωμοσώματος αναπαριστά μια διάσταση του προβλήματος. Οι συνηθέστεροι τρόποι αναπαράστασης λύσεων είναι μέσω ακολουθιών δυαδικών ψηφίων και αλφαριθμητών ακολουθιών.

- **Αρχικός Πληθυσμός:** Ο αρχικός πληθυσμός αποτελείται από τα άτομα καθένα από τα οποία φέρει μια υποψήφια λύση. Η αρχικοποίηση του πληθυσμού γίνεται με την απόδοση τυχαίων τιμών σε κάθε διάσταση. Ωστόσο ο σχεδιαστής πρέπει να φροντίσει ώστε αυτές οι τιμές να αναπαριστούν έγκυρη λύση.
- **Συνάρτηση Προσαρμογής:** Η συνάρτηση προσαρμογής δέχεται σαν είσοδο μια υποψήφια λύση και επιστρέφει μια τιμή η οποία δείχνει πόσο ικανοποιητική είναι. Ο ρόλος της συνάρτησης προσαρμογής είναι βασικής σημασίας για την λειτουργία του αλγορίθμου, καθώς χρησιμοποιείται ως βασικό κριτήριο από την διαδικασία της επιλογής.
- **Διαδικασία Επιλογής:** Η διαδικασία της επιλογής επιτελείται προκειμένου να επιλεγεί ο πληθυσμός που θα στελεχώνει την επόμενη γενιά των λύσεων. Ο τρόπος με τον οποίο γίνεται η επιλογή μπορεί να είναι τυχαίος. Ωστόσο, η συνηθέστερη πρακτική είναι η επιλογή να γίνεται βάσει της συνάρτησης προσαρμογής. Ως εκ τούτου, μεγαλύτερη τιμή της συνάρτησης προσαρμογής συνεπάγεται μεγαλύτερη πιθανότητα επιλογής του συγκεκριμένου ατόμου.
- **Διαδικασία Αναπαραγωγής:** Οι τελεστές αναπαραγωγής χρησιμοποιούν τις ήδη υπάρχουσες λύσεις για την κατασκευή νέων. Οι δύο κυριότεροι τελεστές είναι οι διασταύρωση και η μετάλλαξη. Η διασταύρωση χρησιμοποιεί δύο χρωμοσώματα. Αφού επιλεγεί με τυχαίο τρόπο ένα σημείο διασταύρωσης, γίνεται ο συνδυασμός των δύο λύσεων. Κατά την μετάλλαξη, επιλέγεται ένα χρωμόσωμα. Στην συνέχεια με τυχαίο τρόπο αλλάζουν κάποια κομμάτια από την αναπαράσταση της λύσης. Για παράδειγμα, εάν η λύση αναπαρίσταται δυαδικά τότε κάποια 0 θα γίνουν 1 και το αντίστροφο. Η κυριότερη λειτουργία της διαδικασίας αναπαραγωγής είναι ότι μέσω αυτής αποφεύγονται φαινόμενα παγίδευσης σε τοπικό μέγιστο.
- **Κριτήρια Τερματισμού:** Τα βασικότερα κριτήρια προκειμένου να περατωθεί ένας εξελικτικός αλγόριθμος είναι ο μέγιστος αριθμός γενεών, η περίπτωση κατά την οποία δεν έχει παρατηρηθεί αλλαγή στον πληθυσμό και η περίπτωση κατά την οποία έχει βρεθεί μια ικανοποιητική λύση. Το τελευταίο καθορίζεται μέσω ενός μέγιστου περιθωρίου λάθους.

3.2.3 Γενετικοί Αλγόριθμοι

4 Άσκηση 4η - Fuzzy Logic Systems

4.1 Εκφώνηση

Να γίνει πλήρης βιβλιογραφική έρευνα με βάση τις λέξεις κλειδιά "Fuzzy Logic Systems".

4.2 Ασαφής Λογική

Η ασαφής λογική πρόκειται για μία γενίκευση της συμβατικής Θεωρίας Συνόλων και είναι μία πλειότιμη λογική η οποία ασχολείται με την λογική σαν μία προσέγγιση και όχι σαν κάτι σταθερό και ακριβές [6, 23, 14, 21].

Η ασαφής λογική βασίζεται στα ασαφή σύνολα. Σε σύγκριση με τα δυαδικά σύνολα (οι μεταβλητές των οποίων μπορούν να λάβουν την τιμή "αλήθεια" ή "ψευδές"), τα ασαφή σύνολα περιέχουν αντικείμενα τα οποία ικανοποιούν ανακριβείς ιδιότητες και μπορεί να έχουν μία τιμή αληθείας η οποία κυμαίνεται στο βαθμό μεταξύ 0 και 1 σε αυτό το σύνολο (βλ. εξίσωση (3)) [19].

$$\mu_A : X \rightarrow [0, 1] \quad (3)$$

όπου X ο χώρος των αντικειμένων. Αν x ένα γενικό αντικείμενο του X με $x \in X$ τότε το $\mu_A(x)$ δείχνει την βεβαιότητα με την οποία το στοιχείο x ανήκει στο ασαφή σύνολο A .

Ο βαθμός συμμετοχής σε ένα ασαφές σύνολο υποδηλώνει την βεβαιότητα (ή την αβεβαιότητα) ότι το στοιχείο ανήκει σε αυτή την ομάδα. Η ασαφής λογική χρησιμοποιείται για να χειριστεί την έννοια της μερικής αλήθειας, όπου η τιμή της αλήθειας μπορεί να κυμαίνεται μεταξύ εντελώς αλήθειας και εντελώς ψευδής. Τα ασαφή σύνολα επιτρέπουν την μοντελοποίηση των αβεβαιοτήτων της φυσικής γλώσσας και μπορούν να χρησιμοποιηθούν τόσο για διακριτούς ή/και συνεχόμενους χώρους [6, 23, 14].

Για τον πλήρη ορισμό ενός ασαφές συνόλου χρειαζόμαστε ακόμα και την συνάρτηση μέλος του ασαφούς συνόλου, η οποία χρησιμοποιείται για να συνδέσει τον βαθμό συμμετοχής κάθε στοιχείου x του χώρου των αντικειμένων με το αντίστοιχο ασαφές σύνολο. Οι συναρτήσεις μέλη μπορεί να έχουν οποιοδήποτε σχήμα ή τύπο (βλ. σχήμα 4.1). [6].

Τέλος στην ασαφή λογική μπορούν να χρησιμοποιηθούν και λεκτικές μεταβλητές (αγγλ. linguistic variables). Με τον όρο λεκτική μεταβλητή εννοούμε τις μεταβλητές των οποίων οι τιμές είναι λέξεις σε μία φυσική ή τεχνική γλώσσα [20]. Οι λεκτικές μεταβλητές επιτρέπουν την μετατροπή της φυσικής γλώσσας σε λογικές ή αριθμητικές παραστάσεις, οι οποίες παρέχουν τα εργαλεία για την προσεγγιστική λογική [6].

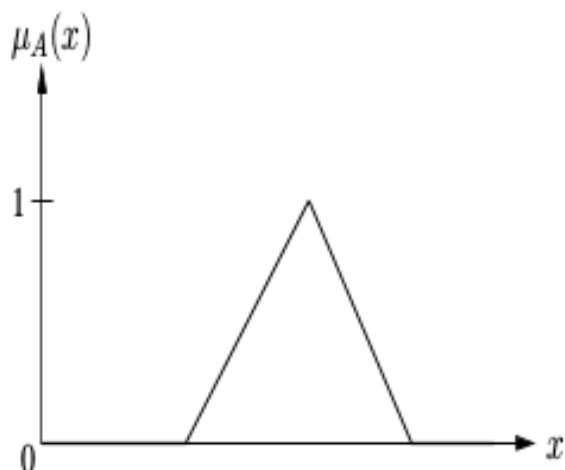
4.3 Συστήματα ασαφούς λογικής

Η σχεδίαση συστημάτων ασαφούς λογικής (ΣΑΛ) είναι ένας από τους μεγαλύτερους τομείς εφαρμογής της ασαφούς λογικής [6]. Ένα σύστημα ασαφούς λογικής (αγγλ. Fuzzy Logic Systems - FLS) είναι μία γραμμική απεικόνιση ενός διανύσματος δεδομένων εισόδου (χαρακτηριστικά) σε μία βαθμωτή έξοδο (η οποία αποσυντίθεται σε μία συλλογή από ανεξάρτητες πολλαπλές εισόδου/μονής εξόδου σύστημα - βλέπε σχήμα 4.2) [9, 23].

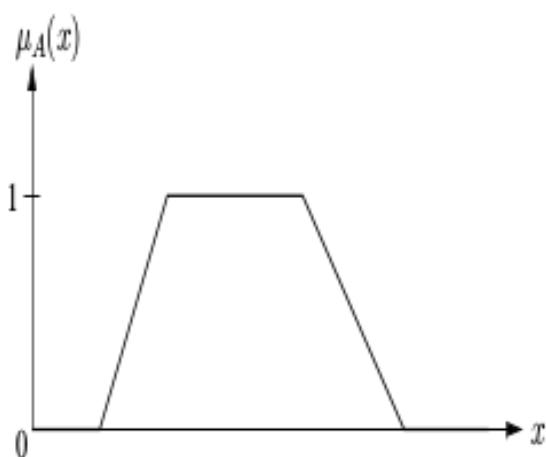
Στα ΣΑΛ, η δυναμική συμπεριφορά του συστήματος περιγράφεται από ένα σύνολο λεκτικών ασαφών κανόνων της μορφής AN - TOTE [6, 23]. Σε συνδυασμό, τα ασαφή σύνολα και οι λεκτικοί κανόνες αποτελούν την γνωσιακή βάση ενός ΣΑΛ. Ακόμα ένα ΣΑΛ αποτελείται από άλλα 3 στοιχεία:

ασαφοποιητής (αγγλ. fuzzifier): ο σκοπός του ασαφοποιητή είναι η εύρεση μίας ασαφούς τιμής από μη ασαφές τιμές εισόδου. Αυτό επιτυγχάνεται χρησιμοποιώντας την συνάρτηση μέλους που σχετίζεται με κάθε ασαφές σύνολο στα δεδομένα εισόδου. Δηλαδή, οι τιμές στα δεδομένα εισόδου αντιστοιχούνται σε βαθμούς συμμετοχής σε ασαφή σύνολα [6].

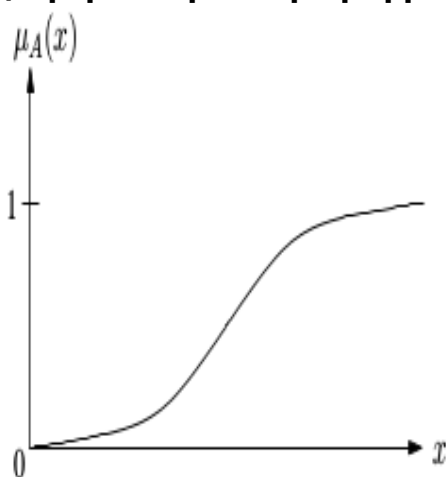
συμπερασματοποιητής (αγγλ. inference): ο σκοπός του συμπερασματοποιητή (αγγλ. inference) είναι να χαρτογραφήσει τις ασαφές εισόδους



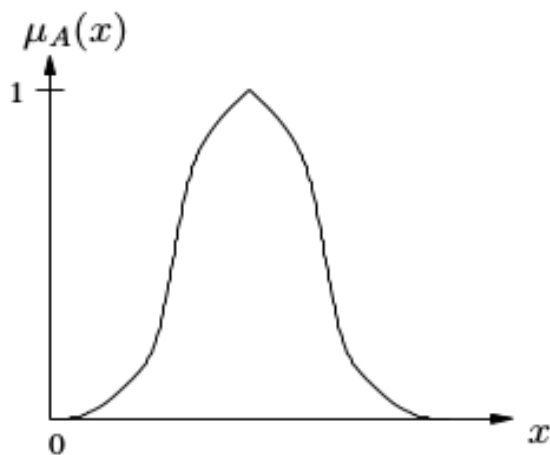
(α') Τριγωνική συνάρτηση μέλους



(β') Τραπεζοειδής συνάρτηση μέλους

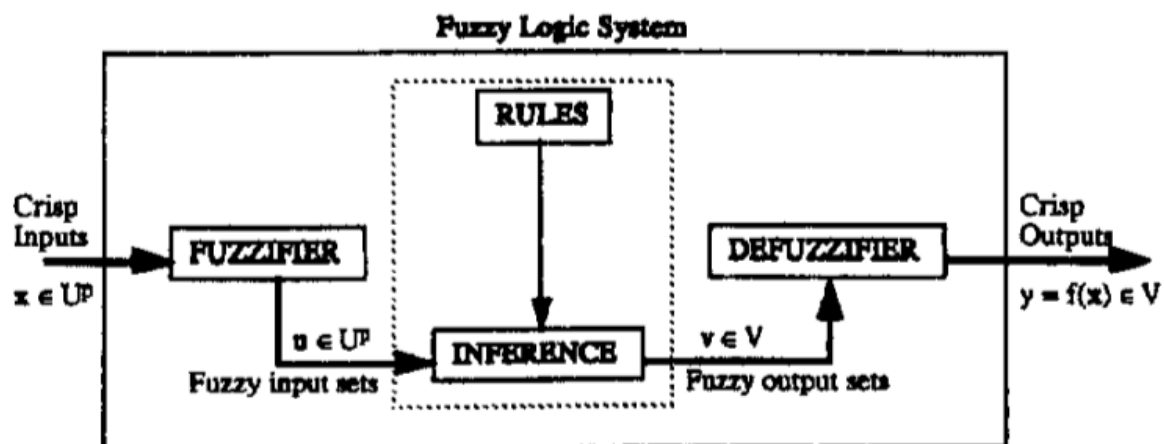


(γ') Λογιστική συνάρτηση μέλους

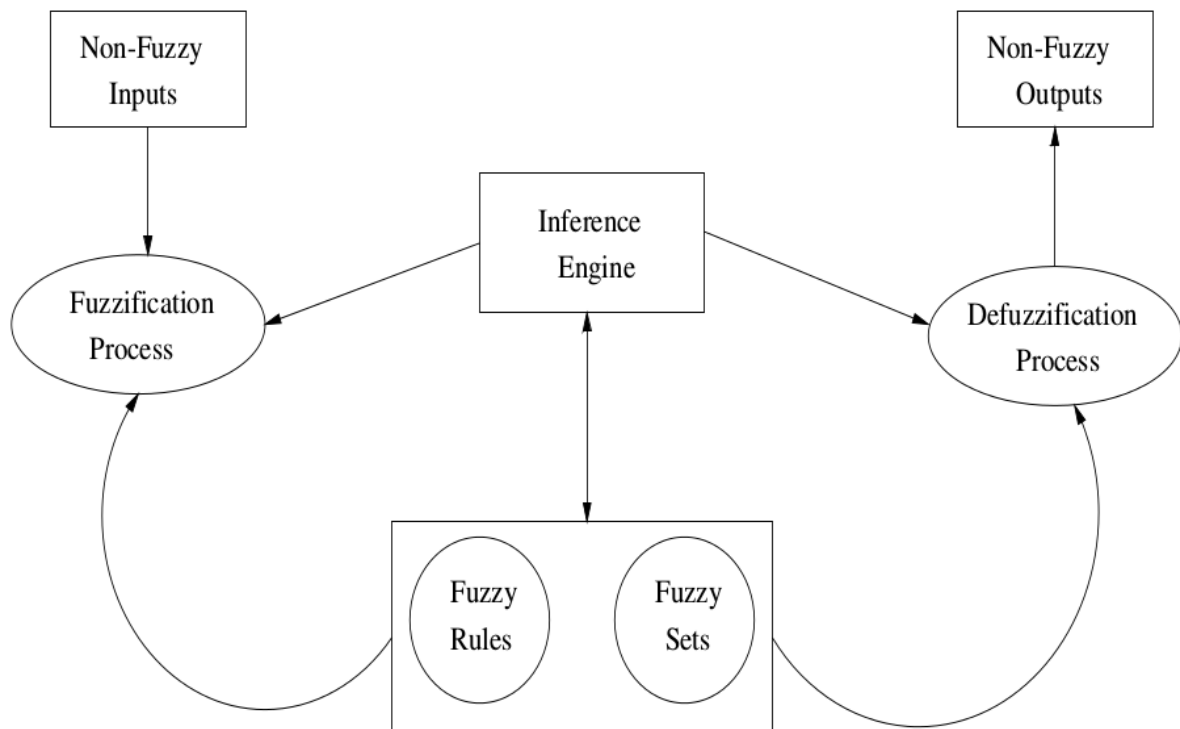


(δ') Γκαουσιανή συνάρτηση μέλους

Σχήμα 4.1: Παραδείγματα συναρτήσεων μελών [6]



Σχήμα 4.2: Σύστημα ασαφούς λογικής [9, 23]



Σχήμα 4.3: Σύστημα ασαφούς λογικής (2) [6]

(όπως λαμβάνονται από την διαδικασία της ασαφοποίησης (αγγλ. *fuzzifier*) στους λεκτικούς κανόνες και να παράγουν μία αποσαφηνισμένη (αγγλ. *fuzzified*) έξοδο για κάθε κανόνα [6].

αποσαφηνιστής(αγγλ. *defuzzifier*): ο σκοπός του αποσαφηνιστή (αγγλ. *defuzzifier*) είναι η παραγωγή μίας σαφούς εξόδους για το ΣΑΛ από το ασαφές σύνολο που προέκυψε ως έξοδος από τον συμπερασματοποιητή. Δηλαδή παραγάγει σε αυτό το στάδιο γίνεται η μετατροπή των ασαφών κανόνων σε μία βαθμωτή τιμή ή γενικά σε μή ασαφή μεταβλητή [23, 6]. Στην βιβλιογραφία έχουν προταθεί αρκετοί αποσαφηνιστές για την εύρεση της βαθμωτής τιμής που αναπαριστά την ενέργεια που πρέπει να πραγματοποιηθεί.

Καθένα από αυτά τα στοιχεία εκτελεί μία συγκεκριμένη εργασία κατά την διαδικασία συλλογισμού (αγγλ. *reasoning process*). Τα διαφορετικά στοιχεία ενός ΣΑΛ φαίνονται στο σχήμα 4.3.

5 Άσκηση 5η

5.1 Εκφώνηση

Να υλοποιηθούν αλγόριθμοι ιεραρχικής ομαδοποίησης δεδομένων και να εφαρμοστούν στα δεδομένα εκφράσεων προσώπου που θα σας παρασχεθούν.

5.2 Λύση

Η ιεραρχική ομαδοποίηση (αγγλ. hierarchical clustering) είναι μία μέθοδος ανάλυσης κλάσεων η οποία αναζητά να κατασκευάσει μία ιεραρχία ομάδων. Η ιεραρχική ομαδοποίηση έχει την ιδιότητα ότι τα δείγματα που ανήκουν στην ίδια ομάδα σε κάποιο επίπεδο να παραμένουν στην ίδια ομάδα σε υψηλότερα επίπεδα [15, 23].

Οι στρατηγικές ιεραρχικής ομαδοποίησης διακρίνονται σε δύο κατηγορίες [15, 23]:

συγχωνευτικές (αγγλ. agglomerative): Οι συγχωνευτικές (από κάτω προς τα πάνω) ξεκινούν από n ομάδες και δημιουργούν μία ακολουθία από διαδοχικές συγχωνεύσεις ομάδων.

διαιρετικές (αγγλ. divisive): Οι διαιρετικές (από πάνω προς τα κάτω) ξεκινούν με όλα τα δείγματα σε μία ομάδα και δημιουργούν μία ακολουθία από διαδοχικές διαιρέσεις ομάδων.

Τα αποτελέσματα της ιεραρχικής ομαδοποίησης συνήθως απεικονίζονται σε ένα δενδρόγραμμα (αγγλ. dendrogram).

Για να διαπιστωθεί ποιες ομάδες θα πρέπει να συγχωνευτούν (ή αντιστοίχως να διαιρεθούν), θα πρέπει να υπάρχει ένα μέτρο της ανομοιογένειας μεταξύ των ομάδων των παρατηρήσεων. Στην πλειοψηφία των ιεραρχικών ομαδοποιήσεων αυτό επιτυγχάνετε με την χρήση ενός κατάλληλου μέτρου (ένα μέτρο της απόστασης μεταξύ των ζευγών των παρατηρήσεων - βλ. πίνακα 5.1), καθώς και ένα κριτήριο συνδέσεως (βλ. πίνακα 5.2) η οποία καθορίζει την ανομοιογένεια των συνόλων ως συνάρτηση των αποστάσεων κατά ζεύγη των παρατηρήσεων στα σύνολα [15].

Όνομα	Τύπος
Ευκλείδεια απόσταση	$\ a - b\ _2 = \sqrt{\sum_i (a_i - b_i)^2}$
Απόσταση manhattan	$\ a - b\ _1 = \sum_i a_i - b_i $
Μέγιστη απόσταση	$\ a - b\ _\infty = \max_i a_i - b_i $
Απόσταση mahalanobis	$\sqrt{(a - b)^T S^{-1} (a - b)}$, όπου S είναι ο πίνακας συ- μεταβλητότητας

Πίνακας 5.1: Ο πίνακας με τις αποστάσεις [15].

Όνομα	Τύπος
Πλήρης σύνδεσης	$\max \{ d(a, b) : a \in A, b \in B \}$
Μονή σύνδεσης	$\min \{ d(a, b) : a \in A, b \in B \}$
Μέση σύνδεσης	$\frac{1}{AB} \sum_{a \in A} \sum_{b \in B} d(a, b)$

Πίνακας 5.2: Ο πίνακας με τα κριτήρια σύνδεσης [15].

Στον πηγαίο κώδικα 1 φαίνεται το script που γράφτηκε για τον υπολογισμό των ιεραρχικών ομαδοποιήσεων στο matlab.

Πηγαίος κώδικας 1: Ο κώδικας του matlab

```

1  % This script demonstrates the hierarchical clustering algorithm on a set of
2  % given data points.
3
4  %Initialize workspace.
5  clc
6  clear all
7
8  % Load data
9  X1 = csvread('features_1.csv');
10 X2 = csvread('features_2.csv');
11 X3 = csvread('features_3.csv');
12
13 % Store both sets of points in a single matrix.
14 X = [X1;X2;X3];
15
16 % Plot the labeled data points
17 figure('Name','Labeled Data Popints')
18 hold on
19 plot(X1(:,1),X1(:,2),'*r','LineWidth',1.4);
20 plot(X2(:,1),X2(:,2),'*b','LineWidth',1.4);
21 plot(X3(:,1),X3(:,2),'*g','LineWidth',1.4);
22 xlabel('x1');
23 ylabel('x2');
24 grid on
25 hold off
26
27 %Plot the unlabeled data points.
28 figure('Name','Unlabeled Data Popints')
29 hold on
30 plot(X(:,1),X(:,2),'*k','LineWidth',1.4);
31 xlabel('x1');
32 ylabel('x2');
33 grid on
34 hold off
35
36 % Create the hierarchical clustering dendrogram with euclidean distance
37 % and average linkage.
38 Y = pdist(X,'euclidean');
39 Z = linkage(Y,'average');
40 figure('Name','Hierarchical Dendrogram (average,euclidean)')
41 [H,T] = dendrogram(Z,0,'colorthreshold','default');
42 set(H,'LineWidth',2)
43
44 % Cluster data in three clusters.
45 T = clusterdata(X,'linkage','average','distance','euclidean',3);
46 figure('Name','Identified Clusters (average,euclidean)')
47 scatter(X(:,1),X(:,2),20,T,'filled')
48 grid on
49
50 % Create the hierarchical clustering dendrogram with mahalanobis distance

```

```

51 % and average linkage.
52 Y = pdist(X,'mahalanobis');
53 Z = linkage(Y,'average');
54 figure('Name','Hierarchical Dendrogram (average,mahalanobis)')
55 [H,T] = dendrogram(Z,0,'colorthreshold','default');
56 set(H,'LineWidth',2)
57
58 % Cluster data in three clusters.
59 T = clusterdata(X,'linkage','average','distance','mahalanobis',3);
60 figure('Name','Identified Clusters (average,mahalanobis)')
61 scatter(X(:,1),X(:,2),20,T,'filled')
62 grid on
63
64 % Create the hierarchical clustering dendrogram with cityblock distance
65 % and average linkage.
66 Y = pdist(X,'cityblock');
67 Z = linkage(Y,'average');
68 figure('Name','Hierarchical Dendrogram (average,cityblock)')
69 [H,T] = dendrogram(Z,0,'colorthreshold','default');
70 set(H,'LineWidth',2)
71
72 % Cluster data in three clusters.
73 T = clusterdata(X,'linkage','average','distance','cityblock',3);
74 figure('Name','Identified Clusters (average,cityblock)')
75 scatter(X(:,1),X(:,2),20,T,'filled')
76 grid on
77
78 % Create the hierarchical clustering dendrogram with euclidean distance
79 % and single linkage.
80 Y = pdist(X,'euclidean');
81 Z = linkage(Y,'single');
82 figure('Name','Hierarchical Dendrogram (single,euclidean)')
83 [H,T] = dendrogram(Z,0,'colorthreshold','default');
84 set(H,'LineWidth',2)
85
86 % Cluster data in three clusters.
87 T = clusterdata(X,'linkage','single','distance','euclidean',3);
88 figure('Name','Identified Clusters (single,euclidean)')
89 scatter(X(:,1),X(:,2),20,T,'filled')
90 grid on
91
92 % Create the hierarchical clustering dendrogram with euclidean distance
93 % and complete linkage.
94 Y = pdist(X,'euclidean');
95 Z = linkage(Y,'complete');
96 figure('Name','Hierarchical Dendrogram (complete,euclidean)')
97 [H,T] = dendrogram(Z,0,'colorthreshold','default');
98 set(H,'LineWidth',2)
99
100 % Cluster data in three clusters.

```

```

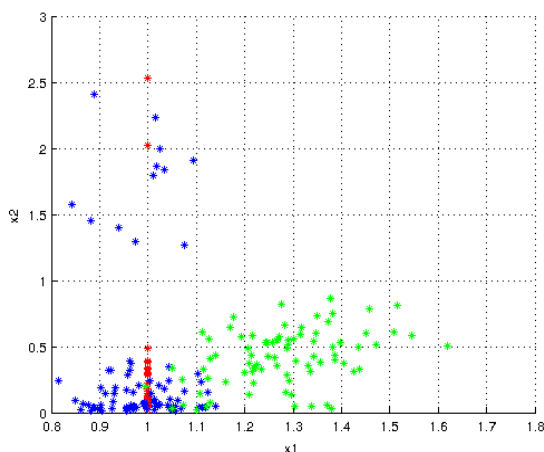
101 T = clusterdata(X,'linkage','complete','distance','euclidean',3);
102 figure('Name','Identified Clusters (complete,euclidean)')
103 scatter(X(:,1),X(:,2),20,T,'filled')
104 grid on

```

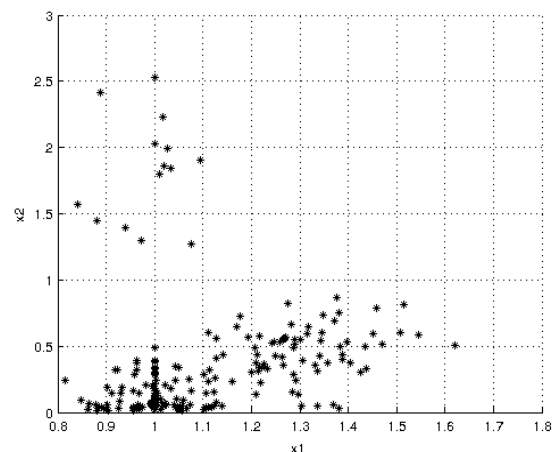
Στο σχήμα 5.1 φαίνονται τα χαρακτηριστικά όπως είναι ταξινομημένα στην πραγματικότητα και όπως τα αντιλαμβάνεται ο αλγόριθμος.

Για την καλύτερη κατανόηση της επίδρασης του μέτρου της απόστασης (βλ. πίνακα 5.1) και της απόστασης μεταξύ των ζευγών παρατηρήσεων (βλ. πίνακα 5.2) πραγματοποιήθηκαν μία σειρά από ιεραρχικές ταξινομήσεις. Στις 3 πρώτες (βλ. σχήμα 5.2 έως 5.4) διατηρήθηκε σταθερή η απόσταση μεταξύ των ζευγών (και συγκεκριμένα μέση σύνδεση) και άλλαζε ο τύπος της απόστασης ενώ στα 2 τελευταία (βλ. σχήμα 5.5 έως 5.6) διατηρήθηκε ο τύπος της απόστασης σταθερός (και συγκεκριμένα ευκλείδεια) και άλλαζε η απόσταση μεταξύ των ζευγών.

Παρατηρούμε ότι οι συστάδες που δημιουργήθηκαν δεν αλλάξαν παρά μόνο στο σχήμα 5.3. Άρα οι συστάδες επηρεάζονται περισσότερο από την γεωμετρία του ίδιου του προβλήματος παρά από τις παραμέτρους που βάζουμε. Αντιθέτως το δένδρογραμμα άλλαζε κάθε φορά πράγμα που το καθιστά και ευαίσθητο στις παραμέτρους που τοποθετούμε στο πρόβλημα.

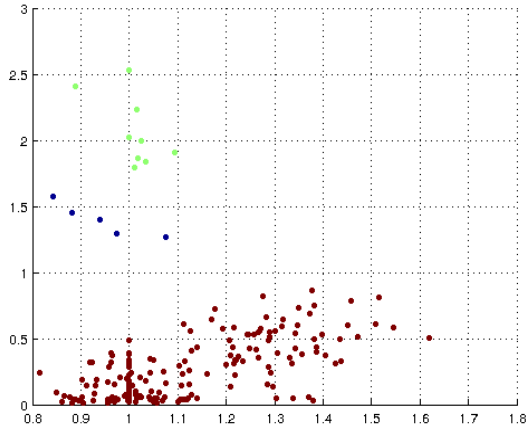
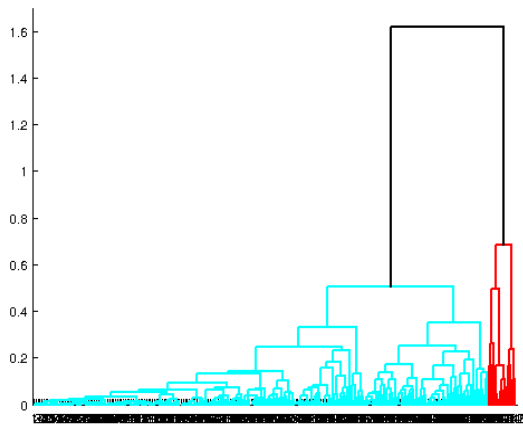


(α') Τα ταξινομημένα δεδομένα



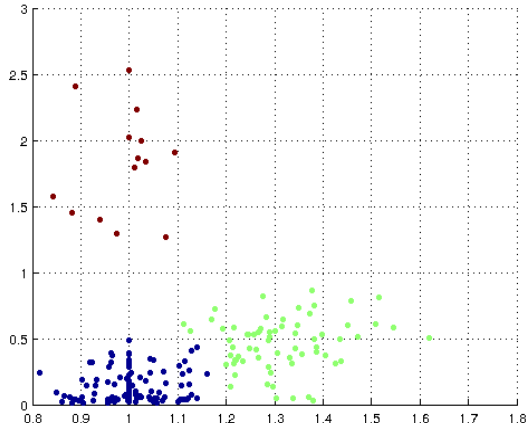
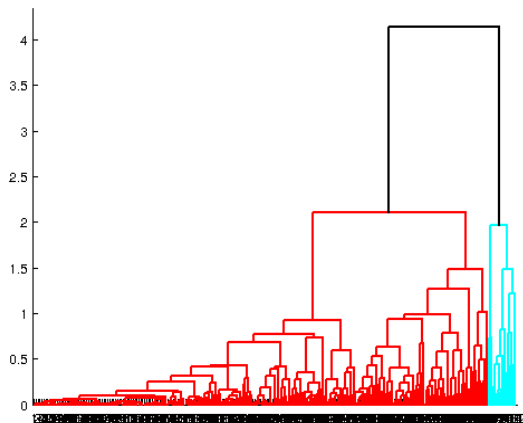
(β') Τα αταξινομημένα δεδομένα

Σχήμα 5.1: Τα δεδομένα εκφράσεων προσώπου



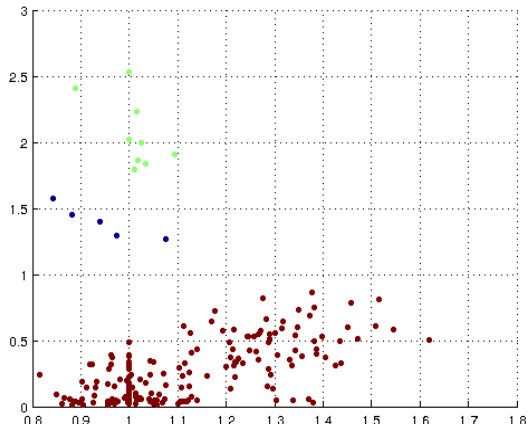
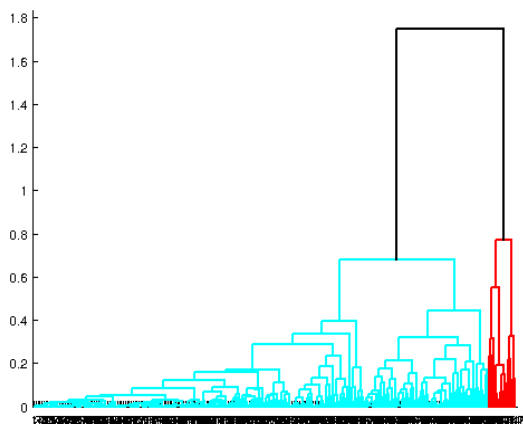
(α') Τα δενδρογράμματα των δεδομένων (β') Οι συστάδες που δημιουργήθηκαν

Σχήμα 5.2: Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την μέση σύνδεση και την ευκλείδεια απόσταση



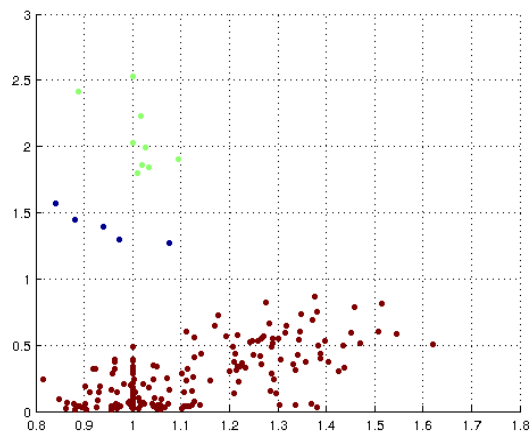
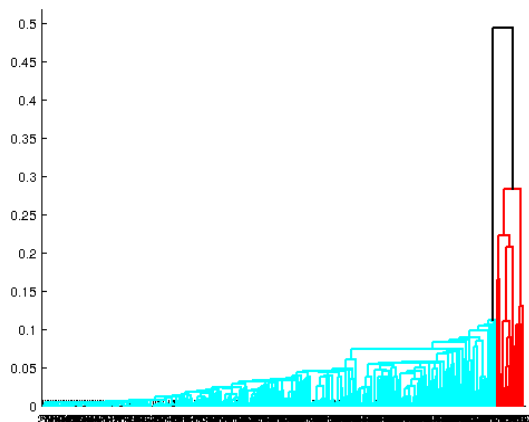
(α') Τα δενδρογράμματα των δεδομένων (β') Οι συστάδες που δημιουργήθηκαν

Σχήμα 5.3: Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την μέση σύνδεση και την απόσταση mahalanobis



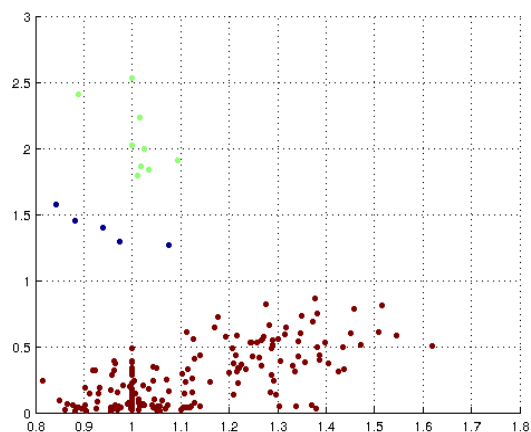
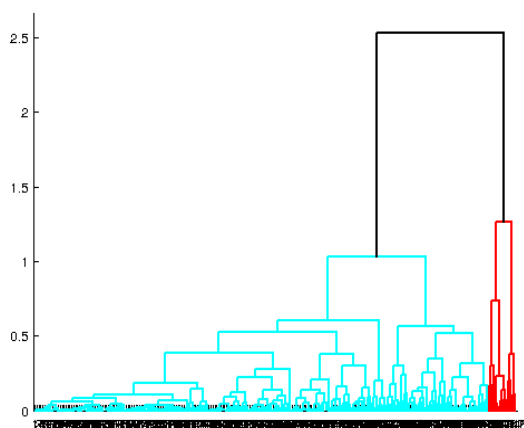
(α') Τα δενδρογράμματα των δεδομένων (β') Οι συστάδες που δημιουργήθηκαν

Σχήμα 5.4: Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την μέση σύνδεση και την απόσταση manhattan



(α') Τα δένδρογραμμα των δεδομένα (β') Οι συστάδες που δημιουργήθηκαν

Σχήμα 5.5: Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την μονή σύνδεση και την ευκλείδεια απόσταση



(α') Τα δένδρογραμμα των δεδομένα (β') Οι συστάδες που δημιουργήθηκαν

Σχήμα 5.6: Η ιεραρχική ομαδοποίηση των δεδομένων χρησιμοποιώντας την πλήρη σύνδεση και την ευκλείδεια απόσταση

Αναφορές

- [1] Hazem Ahmed and Janice Glasgow. Swarm intelligence: concepts, models and applications. *School of Computing, Queen's University*, 2012.
- [2] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics?*, pages 703--712. Springer, 1993.
- [3] Charles Darwin. On the origins of species by means of natural selection. *London: Murray*, 1859.
- [4] Marco Dorigo. *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006, Brussels, Belgium, September 4-7, 2006, Proceedings*, volume 4150. Springer Science & Business Media, 2006.
- [5] Marco Dorigo and Mauro Birattari. Ant colony optimization. In *Encyclopedia of Machine Learning*, pages 36--39. Springer, 2010.
- [6] Andries P. Engelbrecht. *Computational Intelligence: An Introduction*. Wiley Publishing, 2nd edition, 2007. ISBN 0470035617.
- [7] Aleksandar Jevtić and Diego Andina de la Fuente. Swarm intelligence and its applications in swarm robotics. 2007.
- [8] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760--766. Springer, 2010.
- [9] J.M. Mendel. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345--377, Mar 1995. ISSN 0018-9219. doi: 10.1109/5.364485.
- [10] Daniel Merkle, Martin Middendorf, and Hartmut Schmeck. Ant colony optimization for resource-constrained project scheduling. *Evolutionary Computation, IEEE Transactions on*, 6(4):333--346, 2002.
- [11] unknown. Particle swarm optimization (pso), 2015. URL <http://web.ist.utl.pt/gdgp/VA/pso.htm#global>. [Πρόσβαση στις 22 Μαρτίου 2015].
- [12] unknown. Ant colony optimization (pso), 2015. URL <http://www.math.ucla.edu/~wittman/10c.1.11s/Lectures/Raids/AC0.pdf>. [Πρόσβαση στις 23 Μαρτίου 2015].
- [13] Wikipedia. Artificial immune system --- Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Artificial_immune_system, 2015. URL http://en.wikipedia.org/wiki/Artificial_immune_system. [Πρόσβαση στις 20 Ιανουαρίου 2015].
- [14] Wikipedia. Fuzzy logic --- Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Fuzzy_logic, 2015. URL http://en.wikipedia.org/wiki/Fuzzy_logic. [Πρόσβαση στις 18 Ιανουαρίου 2015].

- [15] Wikipedia. Hierarchical clustering --- Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Hierarchical_clustering, 2015. URL http://en.wikipedia.org/wiki/Hierarchical_clustering. [Πρόσβαση στις 25 Ιανουαρίου 2015].
- [16] Wikipedia. Immune system --- Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Immune_system, 2015. URL http://en.wikipedia.org/wiki/Immune_system. [Πρόσβαση στις 24 Ιανουαρίου 2015].
- [17] Wikipedia. Immune system --- Wikipedia, the free encyclopedia. <http://el.wikipedia.org/wiki/%CE%9B%CE%B5%CE%BC%CF%86%CE%BF%CE%BA%CF%8D%CF%84%CF%84%CE%B1%CF%81%CE%BF>, 2015. URL <http://el.wikipedia.org/wiki/%CE%9B%CE%B5%CE%BC%CF%86%CE%BF%CE%BA%CF%8D%CF%84%CF%84%CE%B1%CF%81%CE%BF>. [Πρόσβαση στις 24 Ιανουαρίου 2015].
- [18] Saad Ghaleb Yaseen and Nada MA Al-Slamy. Ant colony optimization. *IJCSNS*, 8(6):351, 2008.
- [19] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 -- 353, 1965. ISSN 0019-9958. doi: [http://dx.doi.org/10.1016/S0019-9958\(65\)90241-X](http://dx.doi.org/10.1016/S0019-9958(65)90241-X). URL <http://www.sciencedirect.com/science/article/pii/S001999586590241X>.
- [20] Lotfi A Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information sciences*, 8(3):199--249, 1975.
- [21] Lotfi A. Zadeh. Fuzzy logic, neural networks, and soft computing. *Commun. ACM*, 37(3):77--84, March 1994. ISSN 0001-0782. doi: 10.1145/175247.175255. URL <http://doi.acm.org/10.1145/175247.175255>.
- [22] Βασίλειος Κ. Καρακάσης. Μοντέλα τεχνητών ανοσοποιητικών συστημάτων για την εξόρυξη γνώσης από σύνολα δεδομένων. Master's thesis, Εθνικό Μετσόβιο Πολυτεχνείο, 2005. URL http://artemis.cslab.ntua.gr/el_thesis/artemis.ntua.ece/DT2005-0229/DT2005-0229.pdf.
- [23] Γεώργιος Α. Τσιχριντζής. Σημειώσεις διδασκαλίας για το μάθημα "Αναγνώριση Προτύπων" - Εισαγωγή στην Αναγνώριση Προτύπων. Σημειώσεις μαθήματος, 2014.