



Σχολή Επιστήμης της Πληροφορίας &
Πληροφορικής

Τμήμα Πληροφορικής

Διπλωματική Εργασία

**«Ευρετικές και Υπολογιστικές Μέθοδοι
Επίλυσης Προβλημάτων»**

Παναγιώτης Μπούρας (ΠΜ 201704)

ΦΕΒΡΟΥΑΡΙΟΣ 2019



Σχολή Επιστήμης της Πληροφορίας &
Πληροφορικής

Τμήμα Πληροφορικής

Διπλωματική Εργασία

**«Ευρετικές και Υπολογιστικές Μέθοδοι
Επίλυσης Προβλημάτων»**

Παναγιώτης Μπούρας (ΠΜ 201704)

Τριμελής Επιτροπή	
Παναγιώτης Βλάμος	Καθηγητής (Επιβλέπων)
Φοίβος Μυλωνάς	Αναπληρωτής Καθηγητής (Μέλος Τριμελούς)
Θεμιστοκλής Έξαρχος	Επίκουρος Καθηγητής (Μέλος Τριμελούς)

ΦΕΒΡΟΥΑΡΙΟΣ 2019

Ευχαριστίες

Πρώτα απ' όλα, θέλω να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, Καθηγητή και Πρόεδρο του Τμήματος Πληροφορικής κ. Παναγιώτη Βλάμο, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μου. Επίσης, είμαι ευγνώμων στα υπόλοιπα μέλη της εξεταστικής επιτροπής της διπλωματικής εργασίας μου, Αναπληρωτή Καθηγητή κ. Φοίβο Μυλωνα και Επίκουρο Καθηγητή Θεμιστοκλή Έξαρχο για την προσεκτική ανάγνωση της εργασίας μου και για τις πολύτιμες υποδείξεις τους. Ευχαριστώ τους συναδέλφους μου Καραγεωργίου Νικόλαο, Πορτοκαλά Χρήστο, Παπαζή Νικόλαο και Βαρσάμη Δημήτριο για την πολύτιμη βοήθειά τους καθώς και την ηθική τους υποστήριξη. Επίσης, ευχαριστώ την Μαρία Καραϊνδρου για την κατανόησή της, ιδιαίτερα κατά τη διάρκεια των τελευταίων μηνών της προσπάθειάς μου. Πάνω απ' όλα, είμαι ευγνώμων στους γονείς μου για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια. Αφιερώνω αυτή την εργασία στην μητέρα μου και στον πατέρα μου.

Παναγιώτης Μπούρας.

Περιεχόμενα

Περίληψη	9
Abstract.....	10
1. Επίλυση Προβλημάτων.....	11
1.1. Αναζήτηση λύσης προβλημάτων	11
1.1.1 Προβλήματα και τύποι προβλημάτων	12
1.1.2 Προβλήματα που επιλύονται με αναζήτηση.....	14
1.2 Ορισμός Προβλήματος.....	17
1.2.1 Το μέγεθος του χώρου αναζήτησης	20
1.3 Μοντελοποίηση του προβλήματος	29
2. Ευρετικές Μέθοδοι επίλυσης προβλημάτων	37
2.1. Άπληστοι Αλγόριθμοι (Greedy Algorithms)	37
2.1.1. Ο Αλγόριθμος του Dijkstra	38
2.2. Διαιρεί και βασίλευε (Divideandconquer)	40
2.3. Διακλάδωση και Οριοθέτηση (BranchandBound)	42
2.4. Αλγόριθμος A^*	45
3. Υπολογιστικές Μέθοδοι για επίλυση προβλημάτων	47
3.1. Εισαγωγή	47
3.2. Κατηγορίες Μηχανικής Μάθησης.....	51
3.3. Κατηγοριοποίηση.....	54
3.3.1. Τεχνικές Ταξινόμησης.....	56
3.3.2. Μηχανές Διανυσμάτων Υποστήριξης (SVM).....	58
3.4. Προβλήματα Ομαδοποίησης	60
3.4.1. Εισαγωγή.....	60
3.4.2. Μαθηματικός Ορισμός	62
3.5. Διαδικασία Ομαδοποίησης.....	63
3.6. Αλγόριθμοι ομαδοποίησης.....	65
4. Εξελικτικοί Υπολογισμοί	69
4.1. Εισαγωγή	69
4.2. Εξελικτικοί αλγόριθμοι και ομαδοποίηση.....	74
Βιβλιογραφία.....	78

Πίνακας Εικόνων

Εικόνα 1: Γραφική αναπαράσταση του κόσμου του προβλήματος του πωλητή σε 6 πόλεις.	18
Εικόνα 2: Ένα πρότυπο TSP. Στα TSP των σχολικών βιβλίων, συνήθως, επιτρέπονται διαδρομές από κάθε πόλη προς κάθε άλλη πόλη, όμως, στα προβλήματα του πραγματικού κόσμου δεν δίνονται πάντα τέτοιες ευκαιρίες.	24
Εικόνα 3: Γράφημα συνάρτησης $G2$ για $n = 2$, όπου η τιμή μηδέν αποδόθηκε στις ανέφικτες λύσεις.	28
Εικόνα 4: Η γραφική αναπαράσταση της συνάρτησης f_{23} για το κόστος μεταφοράς.	32
Εικόνα 5: Γράφημα προσεγγιστικής συνάρτησης f'_{23} με την αρχική συνάρτηση f_{23}).	35
Εικόνα 6: Επαναληπτική διαδικασία Διαιρεί και Βασίλευε για επίλυση του προβλήματος P	41
Εικόνα 7: Μια διαίρεση του χώρου αναζήτησης S του TSP πέντε πόλεων, ανάλογα με το εάν μία δεδομένη ακμή (ij) ανήκει ή όχι στην διαδρομή. ...	43
Εικόνα 8: Αλγόριθμος διακλάδωσης και οριοθέτησης	44
Εικόνα 9: Αναζήτηση πρώτα-στο-καλύτερο (Best-firstsearch).	46
Εικόνα 10: Φάσεις Μηχανικής Μάθησης	53
Εικόνα 11: Στάδια εξόρυξης γνώσης	54
Εικόνα 12: Από τις 2 στις 3 διαστάσεις	60
Εικόνα 13: Παραδείγματα ομάδων στο επίπεδο	62
Εικόνα 14: Διαδικασία Ομαδοποίησης	63
Εικόνα 15: Κατηγορίες Ομαδοποίησης	66
Εικόνα 16: Ομαδοποίηση Μονοθετικού αλγόριθμου	69
Εικόνα 17: Διάγραμμα ροής Τυπικού Εξελικτικού Αλγόριθμου	72
Εικόνα 18: Κατηγορίες Εξελικτικού Υπολογισμού.....	73

Πίνακας Συντομεύσεων

ΓΑ	ΓΕΝΕΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ
ΔΑ	ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ
ΕΠ	ΕΞΕΛΙΚΤΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ
ΕΥ	ΕΞΕΛΙΚΤΙΚΟΙ ΥΠΟΛΟΓΙΣΜΟΙ
ΕΑ	ΕΞΕΛΙΚΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ
ΕΣ	ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ
ΜΜ	ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ
ΝΔ	ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ
ΣΤ	ΣΤΑΤΙΣΤΙΚΗ ΤΑΞΙΝΟΜΗΣΗ
ΤΝΔ	ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ
ΤΝ	ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
ΥΝ	ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ
ΥΣ	ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

List of Abbreviations

SVM	SUPPORT VECTOR MACHINES
RBS	RULE-BASED SYSTEMS
PU	PROCESSING UNITS
FCM	FUZZY C-MEANS

Περίληψη

Η επίλυση προβλημάτων, η οποία είναι τμήμα της διαδικασίας της σκέψης, αποτελεί έναν σημαντικό ερευνητικό κλάδο, στον οποίον αρκετές υπολογιστικές και ευρετικές μέθοδοι έχουν προταθεί προς αυτή την κατεύθυνση. Στην παρούσα διπλωματική εργασία, αφού ορίσουμε και αναλύσουμε το θέμα της επίλυσης ενός προβλήματος, πραγματοποιείται μια βιβλιογραφική ανασκόπηση γνωστών και καθιερωμένων υπολογιστικών και ευρετικών μεθόδων που χρησιμοποιούνται για την επίλυση προβλημάτων.

Abstract

Problem solving, which is part of the process of thinking, is an important research field in which several computational and heuristic methods have been proposed towards this direction. In this diploma thesis, after defining and analyzing the issue of solving a problem, a literature review of well-known and well-established computational and heuristic methods used to solve problems is carried out.

1. Επίλυση Προβλημάτων

1.1. Αναζήτηση λύσης προβλημάτων

Ο τρόπος επίλυσης ενός προβλήματος εκκινεί από τον καθορισμό αυτού του προβλήματος και εν συνεχεία τη δημιουργία ενός αλγορίθμου. Ο αλγόριθμος πρέπει να είναι ο κατάλληλος, ώστε να επιλύει το πρόβλημα δια μέσου πολλαπλών άμεσων επιλογών ανά βήμα. Η ορθή επιλογή του αλγορίθμου μπορεί να γίνεται τυχαία ή μέσω κάποιου ποιοτικού κριτηρίου. Δύο ορισμοί-κλειδιά που σχετίζονται με τη διαδικασία επίλυσης προβλημάτων είναι η *αναζήτηση* (search) και ο *αλγόριθμος αναζήτησης* (search algorithm). Η *αναζήτηση* αφορά μια διαδικασία όπου εντοπίζεται μια σειρά καθορισμένων επιλογών που έχουν ένα ορισμένο στόχο. Από την άλλη, ο *αλγόριθμος αναζήτησης* αναγνωρίζει ένα πρόβλημα και του επιτρέπει να εισαχθεί σε αυτόν, ώστε να προκύψει ο τρόπος λύσης του προβλήματος μέσω της αναπαράστασης από σειρά ενεργειών. Ωστόσο, για να εισαχθεί πρέπει το πρόβλημα να έχει οριστεί με επάρκεια και σαφήνεια (Funke, 2010).

Επιπρόσθετα, υπάρχουν κάποια συστήματα που εντοπίζουν λύσεις μέσω του καθορισμού πολλαπλών αλγοριθμικών εναλλακτικών, όπως αυτά της ΤΝ. Το παρόν έγγραφο παρουσιάζει τις πιο βασικές αλγοριθμικές προσεγγίσεις. Ωστόσο, πριν από αυτή την αναφορά, πρέπει να αναφερθούν και να αναλυθούν οι ποικίλοι τύποι προβλημάτων και η σειρά των λύσεων που επιδέχονται. Αυτό συμβαίνει γιατί κάθε τύπος προβλήματος για να επιλυθεί χρειάζεται διαφορετικό αλγόριθμο κάθε φορά.

1.1.1 Προβλήματα και τύποι προβλημάτων

Υπάρχουν τύποι προβλημάτων, όπου εξ αρχής ο στόχος τους είναι γνωστοποιημένος και κατ' επέκταση η σειρά βημάτων που πρέπει να ακολουθεί, ώστε να επιλυθεί το πρόβλημα και να αποφέρει το στόχο τους. Υπάρχουν και περιπτώσεις προβλημάτων, όπου δύναται να μην επηρεάζει σε κάτι η γνώση –ή η μη γνώση- των βημάτων. Για να γίνει καλύτερα κατανοητή αυτή η άποψη, μπορεί να αναλογιστεί κάποιος ότι κατά τη διαδικασία ενός παιχνιδιού τάβλι, ο βασικός στόχος είναι η νίκη του ενός από τους δύο παίκτες και όχι οι τρόποι που φτάνει στη νίκη κάποιος παίχτης.

Ωστόσο, από την άλλη, η σειρά βημάτων μπορεί να είναι εξέχουσας σημασίας, όπως σε περιπτώσεις πωλητών που πουλάνε τα προϊόντα από τη μια πόλη στην άλλη. Σε αυτήν την περίπτωση, ο στόχος είναι να υπάρξει η καλύτερη δυνατή μικρότερη απόσταση από το ένα σημείο στο άλλο, ώστε να μειωθεί ο χρόνος μεταφοράς τους. Άλλοι τύποι προβλημάτων μπορεί να μην έχουν γνωστοποιημένο στόχο. Αυτό σημαίνει ότι η επίλυση επικεντρώνεται στην εύρεση του στόχου μέσω των βημάτων. Χαρακτηριστικό παράδειγμα αποτελούν οι ιατρικές διαγνώσεις, οι οποίες επιτελούνται κυρίως μέσω των συμπτωμάτων μιας ασθένειας (Jonassenetal., 2003).

Αποτελεί γεγονός ότι οι αλγόριθμοι με γνωστοποιημένο στόχο δύνανται να μεγιστοποιούν τις αποδόσεις τους συγκριτικά με αυτούς που δεν έχουν γνωστό στόχο. Γενικότερα, οι στόχοι ενός προβλήματος μπορεί να εμπερικλείουν την τελική κατάσταση ενός προβλήματος ή διάφορες άλλες τελικές καταστάσεις. Δηλαδή, στο παράδειγμα με τον πωλητή

παρουσιάζεται μια μόνο τελική κατάσταση που είναι η άφιξη σε μια πόλη, ενώ στο παράδειγμα με το τάβλι παρουσιάζονται πολλές εναλλακτικές τελικές καταστάσεις, ανάλογα με την εξέλιξη του παιχνιδιού. Στη δεύτερη περίπτωση, ο συνδυασμός αυτών θα οδηγήσει σε μια τελική κατάσταση.

Επομένως, οι τύποι των προβλημάτων συμπυκνώνονται σε **α.** προβλήματα με γνωστοποιημένο στόχο και γνωστοποιημένες τελικές καταστάσεις, όπου αναζητείται ένας κατάλληλος αλγόριθμος για μια εξ αυτών. Αυτό συμβαίνει, για παράδειγμα, σε επιλύσεις προβλημάτων σχεδιασμού δράσεων, καθώς και πλοήγησης. Εν συνέχεια, υπάρχουν **β.** προβλήματα όπου είναι γνωστός ο στόχος και οι διάφορες τελικές καταστάσεις, όπου ο αλγόριθμος προσπαθεί να εντοπίσει την καταλληλότερη τελική κατάσταση όπως συμβαίνει με τα προβλήματα βελτιστοποίησης. Επίσης, ο **γ.** τύπος προβλημάτων αφορά τα προβλήματα όπου είναι γνωστοποιημένα κάποια χαρακτηριστικά των τελικών καταστάσεων και, έτσι, ο στόχος είναι να εντοπιστεί ένα επαρκές πλάνο της τελικής κατάστασης (Leung, 2004). Χαρακτηριστικά παραδείγματα είναι τα προβλήματα χρονοπρογραμματισμού ή αλλιώς προβλήματα που ικανοποιούν τους περιορισμούς. Τέλος, ο **δ.** τύπος προβλημάτων σχετίζεται με αυτά που είναι παντελώς άγνωστος ο στόχος τους, αλλά η επίλυσή τους αφορά στην εύρεση μιας τελικής κατάστασης, όπως συμβαίνει με αυτά της εξαγωγής συμπερασμάτων.

Ο προσανατολισμός του παρόντος κεφαλαίου επικεντρώνεται στον **α.** τύπο προβλημάτων. Ένας άξιος ενδιαφέροντος ορισμός είναι ο **στόχος προβλήματος** (problem goal) που αποτελεί το σύνολο των καταστάσεων του προβλήματος, όπου το ζητούμενο πραγματοποιείται και όπου οι

καταστάσεις χαρακτηρίζονται ως τελικές του προβλήματος. Όπως προαναφέρθηκε, η επίλυση των προβλημάτων αναδύει ως πιο σημαντικό και προέχον βήμα την επαρκή και σαφή διατύπωση ενός στόχου. Σημαντικό είναι ότι αυτή η κατηγορία προβλημάτων επιδέχεται διάφορες αλγοριθμικές προσεγγίσεις. Για παράδειγμα, το πρόβλημα των Πύργων του Ανόι, το οποίο μπορεί να επιλυθεί με πολλούς τρόπους χωρίς να επιβάλλεται η γνώση των βημάτων επίλυσης και χωρίς την πραγμάτωση αναδρομής. Η αναδρομή άλλων όμοιων αλγοριθμικών προσεγγίσεων παρουσιάζει κάποια αρνητικά ως προσέγγιση, αφού δε δηλώνεται η σειρά των ενεργειών του αλγορίθμου και δεν πραγματοποιείται αναζήτηση μέσω των ποιοτικών κριτηρίων (Anderson et al., 2001). Βάσει αυτής της θέσης, τα συστήματα της ΤΝ επιβάλλουν μια προσέγγιση αναζήτησης, η οποία καταγράφει και γνωστοποιεί τη σειρά των ενεργειών ενός αλγορίθμου που θα οδηγήσουν στο στόχο.

1.1.2 Προβλήματα που επιλύονται με αναζήτηση

Οι βασικές έννοιες, που περιβάλλουν την προσέγγιση επίλυσης ενός προβλήματος βάσει αναζήτησης, είναι η *κατάσταση ενός προβλήματος* (problem state) και ο *χώρος καταστάσεων* (state space), όπου η δεύτερη έννοια ορίζεται ως το σύνολο των έγκυρων καταστάσεων που μπορεί να επέλθει ένα πρόβλημα, συμπεριλαμβανομένων της αρχικής και των πιθανών τελικών καταστάσεων του (Friedland, 2012).

Η προσέγγιση επίλυσης ενός προβλήματος βάσει αναζήτησης περιγράφει πως δημιουργείται ένας αλγόριθμος αναζήτησης, ο οποίος σχεδιάζεται βάσει της κατάστασης του προβλήματος. Επίσης, τα προβλήματα που επιδέχονται τέτοια επίλυση αναγνωρίζονται και ως προβλήματα που

επιδέχονται επίλυση με χώρο καταστάσεων. Για παράδειγμα, στο πρόβλημα με τον πωλητή η κατάσταση αφορά μια καθορισμένη ενέργεια, όπου το αποτέλεσμα της μετακίνησης του πωλητή βασίζεται στην κατάσταση του προβλήματος. Η κατάσταση του προβλήματος είναι η ακριβής τοποθεσία του πωλητή.

Το ενδιαφέρον σε τέτοια προβλήματα στρέφεται στον εντοπισμό της πιο αποδοτικής και ποιοτικής επίλυσης. Πιο συγκεκριμένα, η αναζήτηση θεωρείται ως ένα πρόβλημα βελτιστοποίησης. Δηλαδή, στο πρόβλημα με τον πωλητή πρέπει να εντοπιστεί η πιο γρήγορη διαδρομή που θα τον οδηγήσει στην επίλυση και στο στόχο του προβλήματος. Φυσικά, η βέλτιστη λύση συνδέεται άρρηκτα με το πρόβλημα. Ειδικότερα, αναδύεται η έννοια της μεγιστοποίησης μιας μέτρησης, δηλαδή το κέρδος ή η ελάττωση ενός μεγέθους, δηλαδή το κόστος. Η βέλτιστη λύση αποτελεί μια πολυσύνθετη έννοια, η οποία πηγάζει θετικά σχετικά με το πόσο ποιοτική είναι η τελική κατάσταση, ενώ προκύπτει αρνητικά από τις ενέργειες που είχαν κόστος, όπως το κόστος μιας διαδρομής. Τα παραπάνω συνοψίζονται στη σχέση:

Συνολικό κόστος = κόστος - κέρδος(κατάστασης στόχου)

Εν συνεχεία, παρουσιάζονται προβλήματα που λειτουργούν ως προσδιοριστικά παραδείγματα. Πιο απλά, αυτά τα προβλήματα απαιτούν αλγορίθμους όπου προσδιορίζονται οι ενέργειες τους ανά βήμα από τους εν γένει περιορισμούς του προβλήματος και όχι από εξωγενείς συνθήκες. Παρόλα αυτά, στα συστήματα της TN υπάρχουν προβλήματα που αφορούν σε διεξαγωγή παιχνιδιών (gameplaying). Σε αυτά, οι εξωγενείς παράγοντες διαδραματίζουν σημαντικό ρόλο στον καθορισμό

της τρέχουσας κατάστασης ενός προβλήματος, αφού έχουν πραγματωθεί συγκεκριμένες επιλογές από αυτούς τους παράγοντες. Αυτή η διαπίστωση αποδεικνύεται και από παιχνίδια που χρειάζονται δύο παίκτες, όπως το προαναφερθέν τάβλι αλλά και το σκάκι, όπου ο ένας παίχτης είναι ένας άνθρωπος και ο δεύτερος παίχτης είναι ο υπολογιστής.

Αυτό το παράδειγμα καθίσταται ιδιαίτερα σημαντικό, αφού η ύπαρξη ενός αντίπαλου παίχτη δύναται να αλλάζει την προσέγγιση της αναζήτησης. Δηλαδή, ο ένας παίχτης, που είναι ο υπολογιστής, χρησιμοποιεί έναν αλγόριθμο που έχει στόχο την εύρεση της καλύτερης/βέλτιστης λύσης για να κερδίσει το παιχνίδι. Αντίθετα, ο αντίπαλος, που είναι το φυσικό πρόσωπο, προσπαθεί να ανατρέψει αυτή την πορεία. Ωστόσο, στο τάβλι υπάρχει ένας πολύ σημαντικός εξωγενής παράγοντας, που επηρεάζει το αποτέλεσμα, ο οποίος είναι η τύχη. Δηλαδή, το αποτέλεσμα των ζαριών είναι τυχαίο. Αυτό έχει ως αποτέλεσμα να διαμορφώνεται ένας μη προσδιοριστικός τρόπος για την επόμενη κατάσταση του προβλήματος.

Παρόλα αυτά, ο αλγόριθμος αναζήτησης εστιάζει μόνο στην τρέχουσα κατάσταση του προβλήματος. Αν αναλογιστεί κάποιος, άλλα παιχνίδια όπως το μπριτζ, που παίζεται με τράπουλα, ένας παίχτης είναι σημαντικό να ανακαλεί τις προγενέστερες κινήσεις και του ίδιου, αλλά και των αντιπάλων, καθώς έτσι καθορίζεται η επόμενη επιλογή κίνησής του. Το ίδιο συμβαίνει και σε περιπτώσεις φυσικής αλληλεπίδρασης. Δηλαδή, σε αυτές τις περιπτώσεις, οι εξωτερικές συνθήκες δεν είναι γνωστοποιημένες εξ αρχής. Επομένως, υπάρχει ένας αλγόριθμος που προσπαθεί να ασκήσει επίδραση στον αληθινό κόσμο και κατ' επέκταση έρχεται αντιμέτωπος με

νέες πληροφορίες και συνθήκες, που δύνανται να διευκολύνουν ή να περιορίσουν την προγραμματισμένη διαδικασία.

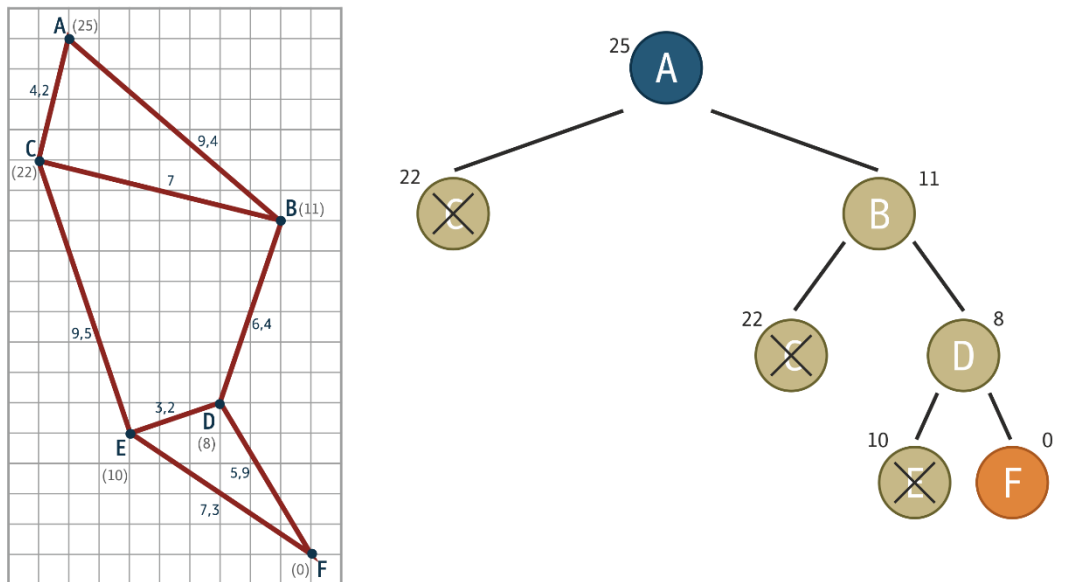
Εν προκειμένω, ο βασικός προσανατολισμός τίθεται στην προσέγγιση της προσδιοριστικής αναζήτησης επιλύσεων σε ένα ορισμένο χώρο καταστάσεων, ο οποίος αποτελείται από επιμέρους υποδειγματικά προβλήματα. Τέτοια υποδειγματικά προβλήματα μπορεί να είναι προβλήματα περιορισμένου φυσικού κόσμου, ατομικών παιχνιδιών και άλλων όπως τα κουίζ. Ως σημαντικότερο και πρωτογενές βήμα έχει αναφερθεί η σαφής και ακριβής **διατύπωση ενός προβλήματος** (problem formulation), η οποία ορίζεται ως η διαδικασία όπου εξηγούνται οι ενέργειες και οι καταστάσεις που θα εξεταστούν βάσει ενός καθορισμένου στόχου. Τέλος, πρέπει να αναφερθεί ότι η διατύπωση για να είναι σαφής πρέπει να διατυπωθεί με τυποποιημένο τρόπο.

1.2 Ορισμός Προβλήματος

Ο ορισμός του προβλήματος είναι μια σύνθετη έννοια η οποία επιβάλλει την εισαγωγή μιας άλλης έννοιας, αυτής του **κόσμου του προβλήματος** (problem world). Αρχικά, είναι σημαντικό να παρουσιαστεί και να οριστεί η παραπάνω έννοια. Ο **κόσμος του προβλήματος** ορίζεται ως το υποσύνολο του πραγματικού κόσμου και περιλαμβάνει αυστηρά μόνο τα αντικείμενα που σχετίζονται με το πρόβλημα. Δηλαδή, τα χαρακτηριστικά των προβλημάτων και τις σχέσεις που τα περιβάλλουν και τα συνδέουν.

Για να κατανοηθεί καλύτερα η έννοια πρέπει να επανέλθει το πρόβλημα με τον πωλητή. Αν ο πωλητής περιπλανηθεί σε έξι πόλεις, τα

χαρακτηριστικά του προβλήματος θα αφορούν τις ιδιότητες των πόλεων, που είναι η απόσταση που έχουν οι πόλεις μεταξύ τους και οι σχέσεις που συνδέουν αυτές τις ιδιότητες θα είναι οι τρέχοντες δρόμοι που διέρχεται ο πωλητής. Αυτή η σχέση απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 1: Γραφική αναπαράσταση του κόσμου του προβλήματος του πωλητή σε 6 πόλεις.

Ο κόσμος του προβλήματος συγκεντρώνει κάποια χαρακτηριστικά και, έτσι, κατηγοριοποιείται σε **κλειστό** (closed world) και **ανοιχτό** (open world). Ο κλειστός κόσμος του προβλήματος αφορά την περίπτωση, όπου δεν εισέρχεται και δεν εξάγεται κανένα νέο αντικείμενο, ιδιότητα ή σχέση από ή προς τον εξωτερικό κόσμο (Applegate et al., 2006). Από την άλλη, ο ανοιχτός κόσμος του προβλήματος σχετίζεται με την περίπτωση όπου υπάρχει εισαγωγή αντικειμένων από τον εξωτερικό κόσμο, τα οποία ασκούν επίδραση στην τρέχουσα εξέλιξη του κόσμου του προβλήματος. Ωστόσο, αυτό συνεπάγεται ότι η εξέλιξη του κόσμου του προβλήματος δε θα είναι εύκολα προβλέψιμη.

Σχετικά με τη διατύπωση του προβλήματος πρέπει να αναφερθεί ότι περικλείεται από τέσσερα στοιχεία, τα οποία συμπληρώνουν τον ορισμό του. Το πρώτο στοιχείο είναι η *αρχική κατάσταση* (start state ή input state) που αρχίζει η αναζήτηση. Το δεύτερο στοιχείο είναι ο *στόχος* (goal), ο οποίος περιλαμβάνει έγκυρες καταστάσεις, με συγκεκριμένα ενδεχόμενα χαρακτηριστικά, όπου η αναζήτηση προσπαθεί να τις εντοπίσει. Ως μια τέτοια κατάσταση θεωρείται η *κατάσταση - στόχος* (goal state) του προβλήματος.

Το τρίτο στοιχείο είναι η *περιγραφή των ενεργειών* που διαθέτει ο αλγόριθμος. Αν η συνάρτηση που δέχεται ένα σύνολο ενεργειών ονομαστεί *συνάρτηση διαδόχων* (successor function), τότε με δεδομένη μια συγκεκριμένη κατάσταση x του προβλήματος, η συνάρτηση διαδόχων θα επιστρέφει ένα σύνολο διατεταγμένων ζευγών (ενέργεια, διάδοχος), όπου κάθε ενέργεια είναι μία από τις επιτρεπτές ενέργειες στην κατάσταση x . Εν συνεχεία, *διάδοχος* (successor) είναι μια κατάσταση στην οποία μπορούμε να οδηγηθούμε από την κατάσταση x με την εκτέλεση αυτής της ενέργειας. Οι ενέργειες αυτές εφαρμόζονται σε μια κατάσταση, για να παράγουν διάδοχες καταστάσεις. Οι ενέργειες μπορούν να ονομαστούν *τελεστές μετάβασης* (transition operators) του προβλήματος (Gutinet al., 2006).

Τέλος, το τέταρτο στοιχείο είναι ο *χώρος καταστάσεων* (state space) που μπορεί να οριστεί έμμεσα από την *αρχική κατάσταση* και τη συνάρτηση διαδόχων. Ο *χώρος καταστάσεων* μπορεί να αναπαρασταθεί με ένα γράφημα, του οποίου οι κόμβοι είναι καταστάσεις και τα τόξα μεταξύ αυτών είναι ενέργειες. Για παράδειγμα, σε προβλήματα διάσχισης δρόμου, το γράφημα του χώρου καταστάσεων ταυτίζεται με τη γραφική

αναπαράσταση του κόσμου του προβλήματος, όπως φαίνεται στο Σχήμα 1. Επιπρόσθετα, και σχετικά με τα προβλήματα που σχετίζονται με την ποιότητα λύσης επιβάλλεται η αναφορά του στοιχείου *συνάρτησης κόστους διαδρομής* (function of path cost).

Σύμφωνα με όσα έχουν αναφερθούν, προκύπτει ότι ένα πρόβλημα (P) θα οριστεί από τη σχέση: $P=(I,G,T,S)$. Όπου:

I: η αρχική κατάσταση,

G: το σύνολο των καταστάσεων που περιλαμβάνει ο στόχος,

T: το σύνολο των τελεστών μετάβασης,

S: ο χώρος των καταστάσεων

Ιδιαίτερα σημαντική είναι η παρουσίαση και η λεκτική περιγραφή των κλειστών κόσμων των υποδειγματικών προβλημάτων, η οποία θα παρουσιαστεί στη συνέχεια με παραδείγματα προβλημάτων, όπως αυτών των 3 κύβων, του αγρότη με τη βάρκα κοκ. Επίσης, οι τεχνικές της αναπαράστασης της γνώσης για ένα πρόβλημα προκύπτουν μέσα από την προαναφερθείσα λεκτική περιγραφή (Bektas, 2006).

1.2.1 Το μέγεθος του χώρου αναζήτησης

Αναφορικά με τα προβλήματα στη λογική, προκύπτει το πρόβλημα ικανοποιησιμότητας των λογικών προτάσεων (Satisfiability Problem ή SAT). Στόχος είναι, για μια σύνθετη πρόταση Boolean μεταβλητών, να γίνει ανάθεση τιμών στις μεταβλητές, ώστε η πρόταση να ισοδυναμεί με ΑΛΗΘΕΣ. Αν υποθέσουμε ότι υπάρχουν 100 μεταβλητές σε συζευκτική κανονική μορφή, προκύπτει:

$$F(x) = (x_{17} \vee \bar{x}_{37} \vee x_{73}) \wedge (\bar{x}_{11} \vee \bar{x}_{56}) \wedge \dots \wedge (x_2 \vee x_{43} \vee \bar{x}_{77} \vee \bar{x}_{89} \vee \bar{x}_{97}).$$

Συνεπώς, αναζητείται η ανάθεση τιμής για κάθε μεταβλητή x_i , για όλα τα $i = 1, \dots, 100$, έτσι ώστε $F(x) = \text{ΑΛΗΘΕΣ}$. Δύναται να χρησιμοποιηθεί 1 και 0 ως συνώνυμα για το ΑΛΗΘΕΣ και ΨΕΥΔΕΣ, με $\bar{x}_i$ αρνητικό του x_i (δηλαδή, εάν x_i είναι ΑΛΗΘΕΣ ή 1, τότε το $\bar{x}_i$ θα είναι ΨΕΥΔΕΣ ή 0). Ανεξάρτητα από το πρόβλημα που πρέπει να επιλυθεί, είναι σημαντικό να εξετάζεται ο *χώρος των πιθανών λύσεων*.

Στην παραπάνω περίπτωση, κάθε δυαδική συμβολοσειρά μήκους 100 αποτελεί μία πιθανή λύση στο πρόβλημα. Κάθε μεταβλητή περιλαμβάνει δυο επιλογές και, έτσι, εάν υπάρχουν 100 μεταβλητές, τότε προκύπτουν 2^{100} πιθανότητες. Επομένως, το μέγεθος του χώρου αναζήτησης είναι $|S| = 2^{100} \approx 10^{30}$, ο οποίος είναι ιδιαίτερα μεγάλος αριθμός. Αυτό συνεπάγεται ότι είναι ανέφικτο να δοκιμαστούν όλες αυτές οι εναλλακτικές. Για παράδειγμα, εάν υπάρχει ένας υπολογιστής που δύναται να δοκιμάσει 1000 συμβολοσειρές ανά δευτερόλεπτο και λειτουργούσε από την αρχή του κόσμου, πριν από 15 δισεκατομμύρια έτη, θα είχε εξεταστεί λιγότερο από ένα τοις εκατό όλων των πιθανοτήτων έως σήμερα (Mézard et al., 2005).

Από την άλλη, η επιλογή της συνάρτησης αξιολόγησης που θα χρησιμοποιηθεί δεν είναι σαφής. Πιο ειδικά, το στοιχείο, που είναι απαραίτητο από μία συνάρτηση αξιολόγησης, είναι κάποιες οδηγίες που σχετίζονται με την ποιότητα της προτεινόμενης λύσης. Δηλαδή, οι λύσεις που πλησιάζουν τη σωστή απάντηση θα πρέπει να συμπυκνώνουν καλύτερες αξιολογήσεις από αυτές που δεν πλησιάζουν την σωστή απάντηση. Σε αυτό το σημείο, ωστόσο, πρέπει να υπολογιστεί το $F(x)$ το

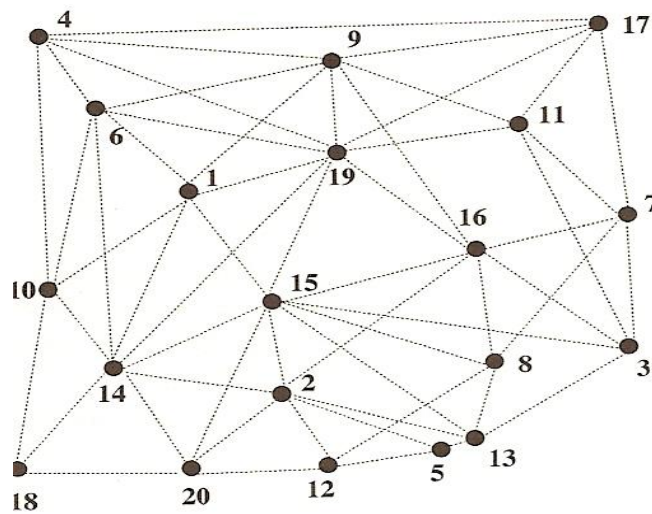
οποίο μπορεί να είναι είτε ΑΛΗΘΕΣ είτε ΨΕΥΔΕΣ. Εάν δοκιμαστεί μία συμβολοσειρά x και το $F(x)$ επιφέρει τιμή ΑΛΗΘΕΣ, τότε προκύπτει το επιθυμητό αποτέλεσμα, δηλαδή η εύρεση της απάντησης (Gottlieb et al., 2002).

Ωστόσο, εάν το $F(x)$ επιφέρει τιμή ΨΕΥΔΕΣ, δεν προκύπτει το ίδιο αποτέλεσμα. Το ερώτημα που προκύπτει είναι το πώς μπορούν να διακριθούν οι καλύτερες από τις χειρότερες πιθανές λύσεις, εφόσον σχεδόν κάθε πιθανή συμβολοσειρά από 0 και 1 που μπορεί να δοκιμαστεί, συνήθως επιστρέφει τιμή ΨΕΥΔΕΣ. Σύμφωνα με αυτά, εάν χρησιμοποιηθεί μια απαριθμητική αναζήτηση, δεν παρουσιάζεται κάποιο πρόβλημα, γιατί εξετάζεται κάθε πιθανότητα μέχρι να βρεθεί κάποια ενδιαφέρουσα. Από την άλλη, εάν επιθυμείται, μέσα από την συνάρτηση αξιολόγησης, να εντοπιστούν οι βέλτιστες λύσεις ταχύτερα από ότι με την απαρίθμηση, δεν είναι επαρκής μια λύση «σωστή» ή «λανθασμένη». Επιπρόσθετα, ο τρόπος που μπορεί να επιτευχθεί το παραπάνω για το πρόβλημα SAT δεν είναι εξαρχής σαφής.

Γενικότερα, υπάρχουν προβλήματα που φαίνεται να είναι ευκολότερα από τα SAT. Αυτό συμβαίνει, γιατί αυτά τα προβλήματα προτείνουν μια πιθανή συνάρτηση αξιολόγησης. Το αρνητικό στοιχείο έγκειται ότι και σε αυτά τα προβλήματα το μέγεθος του χώρου αναζήτησης υπάρχει η πιθανότητα να είναι ιδιαίτερα μεγάλο. Αν χρησιμοποιηθεί πάλι το παράδειγμα του προβλήματος με τον περιπλανώμενο πωλητή, θα πρέπει να αναλογιστεί κάποιος ότι ο πωλητής θα επισκεφτεί κάθε προκαθορισμένη πόλη/τοποθεσία μια φορά μονάχα και στην επιστροφή θα επιλέξει την πιο σύντομη απόσταση.

Παρόλα αυτά, υπάρχουν πανομοιότυπα προβλήματα που χρειάζεται να χρησιμοποιηθούν διαφορετικά κριτήρια για την επίλυσή τους. Ως τέτοια μπορεί να θεωρηθούν προβλήματα αναζήτησης μιας διαδρομής με την ελάχιστη χρονική διάρκεια ταξιδιού ή το μικρότερο κόστος καυσίμων, καθώς και άλλα κριτήρια. Παρά αυτές τις διαφορές, το γενικό πλαίσιο παραμένει το ίδιο. Επίσης, στο παράδειγμα με το πρόβλημα του πωλητή προκύπτει το ερώτημα πώς πρέπει να προγραμματιστεί το δρομολόγιο του εάν θεωρηθεί ως κριτήριο το κόστος ταξιδιού από την μια πόλη στην άλλη και συνεπώς πρέπει να εντοπιστεί το ελάχιστο κόστος διάνυσης απόστασης.

Για παράδειγμα, αν παρατηρηθεί το Σχήμα 2, διαφαίνεται ένα συμμετρικό TSP 20 πόλεων, όπου η απόσταση μεταξύ κάθε ζεύγους πόλεων i και j είναι ίδια προς κάθε κατεύθυνση. Δηλαδή, $dist(i,j)=dist(j,i)$. Παρότι οι πραγματικές αποστάσεις δεν αναφέρονται στο σχήμα, θεωρείται ότι έτσι είναι. Το εναλλακτικό σχήμα που μπορεί να προκύψει είναι ότι μπορεί να σχεδιαστεί ένα μη συμμετρικό TSP, όπου $dist(i,j) \neq dist(j,i)$ για ένα ορισμένο i και j . Συμπληρωματικά, αυτές οι δύο κατηγορίες του TSP παρουσιάζουν διάφορα προβλήματα στην αναζήτηση των διαδρομών ελάχιστου κόστους.



Εικόνα 2: Ένα πρότυπο TSP. Στα TSP των σχολικών βιβλίων, συνήθως, επιτρέπονται διαδρομές από κάθε πόλη προς κάθε άλλη πόλη, όμως, στα προβλήματα του πραγματικού κόσμου δεν δίνονται πάντα τέτοιες ευκαιρίες.

Σύμφωνα με τα παραπάνω, πρέπει να εντοπιστεί ο χώρος αναζήτησης για το TSP. Ο τρόπος για να γίνει αυτό, θα μπορούσε να ήταν ο ορισμός μιας πιθανότητας που θα όριζε ως χώρο αναζήτησης το σύνολο μεταθέσεων n πόλεων. Αυτό σημαίνει ότι η κάθε μετάθεση n πόλεων αποδίδει μία ταξινομημένη λίστα που καθορίζει την ακολουθία των πόλεων, που θα επισκεφθεί ο αντιπρόσωπος πωλητής. Δηλαδή, θα ξεκινήσει από τη βάση του και θα κατευθυνθεί προς τον τελικό προορισμό του προτού ακριβώς ξεκινήσει την επιστροφή του στη βάση του. Σε αυτήν την περίπτωση, η καλύτερη/βέλτιστη επίλυση είναι η μετάθεση που θα δίνει τη διαδρομή με το ελάχιστο κόστος.

Γενικότερα, διαδρομές όπως $2 - \dots - 6 - 15 - 3 - 11 - 19 - 17, 15 - 3 - 11 - 19 - 17 - 2 - \dots - 6, 3 - 11 - 19 - 17 - 2 - \dots - 6 - 15$ είναι πανομοιότυπες, καθώς η περίμετρος που δημιουργείται από την καθεμία είναι ίδια, ανεξάρτητα από την πόλη-αφετηρία. Επίσης, υπάρχουν n διαδρομές σαν αυτήν για

Διαφαίνεται ότι προκύπτει ξανά ιδιαίτερα μεγάλος αριθμός. Δηλαδή, για κάθε $n > 6$, ο αριθμός των πιθανών λύσεων στο TSP με n πόλεις είναι μεγαλύτερος από τον αριθμό των πιθανών λύσεων στο πρόβλημα SAT με n μεταβλητές. Επιπρόσθετα, η διαφορά μεταξύ των μεγεθών των δύο χώρων αναζήτησης θα αυξηθεί άμεσα καθ' όλη τη διάρκεια αύξησης του n . Ειδικά, για $n = 6$, υπάρχουν $5!/2 = 60$ διαφορετικές λύσεις στο TSP και $2^6 = 64$ λύσεις σε ένα SAT, αλλά για $n = 7$ αυτοί οι αριθμοί είναι 360 και 128 αντίστοιχα.

[illegible]

{ 25 }

αντιληφθεί τέτοια ποσοστά αριθμών. Ακόμα και στην περίπτωση που το TSP έχει έναν απίστευτα μεγάλο χώρο αναζήτησης, η συνάρτηση αξιολόγησης που μπορεί να χρησιμοποιηθεί, για να υπολογιστεί η ποιότητα κάθε συγκεκριμένης διαδρομής πόλεων, είναι πολύ πιο απλή από αυτήν που χρησιμοποιείται για ένα πρόβλημα SAT.

Σε αυτό το σημείο, κρίνεται σημαντική η αναφορά ενός πίνακα που εμφανίζει όλες τις αποστάσεις μεταξύ κάθε ζεύγους πόλεων, και μετά από n προσθέσεις υπολογίζεται η απόσταση κάθε υποψήφιας διαδρομής. Η απόσταση χρησιμοποιείται για να αξιολογηθεί το πλεονέκτημά της. Αναλυτικότερα, το κόστος της διαδρομής: $15 - 3 - 11 - 19 - 17 - 2 - \dots - 6$, είναι:

$Cost = dist(15,3) + dist(3,11) + dist(11,19) + \dots + dist(6,15)$. Σε αυτή την περίπτωση θα έπρεπε να θεωρείται ότι η πιο φυσική συνάρτηση αξιολόγησης θα μπορούσε να υποβοηθήσει στην αναζήτηση χρήσιμων επιλύσεων στο TSP ανεξάρτητα από το μέγεθος του χώρου αναζήτησης.

Επιπρόσθετα, ιδιαίτερα χρήσιμη καθίσταται η ανάλυση ενός συγκεκριμένου προβλήματος μη γραμμικού προγραμματισμού (NLP). Αυτό το πρόβλημα είναι ιδιαίτερα σύνθετο και έχει απασχολήσει τους επιστήμονες ιδιαίτερα, καθώς δεν έχει βρεθεί κανένα ικανοποιητικό αποτέλεσμα με την εφαρμογή τυπικών μεθόδων βελτιστοποίησης. Το παράδειγμα που παρουσιάζεται είναι η μεγιστοποίηση της συνάρτησης:

$$G2(x) = \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}} \right|, \text{ για}$$

$$\prod_{i=1}^n x_i \geq 0.75 \text{ και } \sum_{i=1}^n x_i \leq 7.5n \text{ και όρια } 0 \leq x_i \leq 10 \text{ για } 1 \leq i \leq n.$$

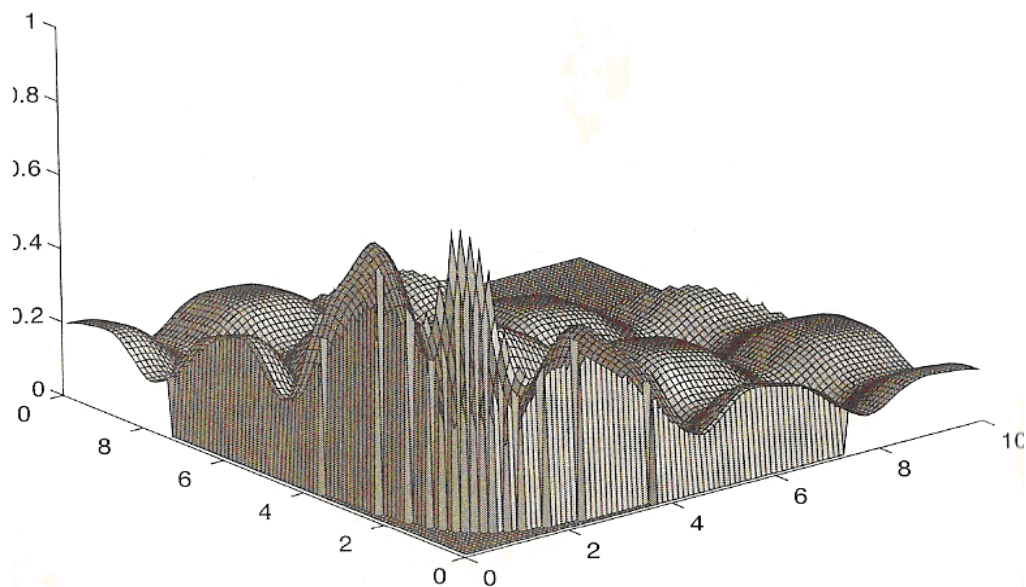
Η παραπάνω συνάρτηση $G2$ ενέχει μη γραμμικό προγραμματισμό και ολικό μέγιστό της άγνωστο, αλλά βρίσκεται κοντά στην αρχή των αξόνων (origin). Έτσι, το πρόβλημα βελτιστοποίησης θέτει έναν μη γραμμικό περιορισμό και έναν γραμμικό περιορισμό, όπου ο γραμμικός είναι ανενεργός κοντά στην αρχή των αξόνων.

Σε αυτό το παράδειγμα πρέπει πάλι να εντοπιστεί το μέγεθος του χώρου αναζήτησης. Μπορεί εκ πρώτης όψεως να θεωρηθεί ότι το μέγεθος του χώρου αναζήτησης εξαρτάται από την διαστασιμότητα του προβλήματος, που είναι ο αριθμός των μεταβλητών. Αν αντιμετωπιστεί ως ένα ξεκάθαρα μαθηματικό πρόβλημα, με n μεταβλητές, κάθε διάσταση δύναται να ενσωματώνει άπειρο αριθμό πιθανών τιμών και κατ' επέκταση θα προκύψει ένας αφάνταστα μεγάλος χώρος.

Παρόλα αυτά, εάν χρησιμοποιηθεί ένας αλγόριθμος για τον εντοπισμό του βέλτιστου $G2$ ε έναν υπολογιστή, που όλα είναι πεπερασμένα και ψηφιακά, τότε πρέπει να αναλογιστεί κάποιος τη διαθέσιμη ακρίβεια του υπολογιστή. Δηλαδή, εάν υπάρχει ακρίβεια με έξι δεκαδικά ψηφία, τότε κάθε μεταβλητή μπορεί να λάβει 10.000.000 διαφορετικές τιμές. Αυτό συνεπάγεται ότι το μέγεθος του χώρου αναζήτησης θα είναι $|S| = 10.000.000^n = 10^{7n}$. Αυτός ο αριθμός είναι αισθητά μεγαλύτερος από τον αριθμό των επιλύσεων για το TSP. Επιπλέον, ακόμα και για $n = 50$ υπάρχουν 10^{350} λύσεις στο NLP με μόνο έξι δεκαδικά ψηφία ακρίβειας, οι περισσότεροι υπολογιστές έχουν τη δυνατότητα να δίνουν δύο φορές αυτήν την ακρίβεια.

Σχετικά με τη συνάρτηση αξιολόγησης αναδύεται το ερώτημα κατά πόσον μπορεί να υπολογιστεί η ποιότητα των εναλλακτικών λύσεων. Ένας από τους τρόπους θα ήταν να χρησιμοποιηθεί η παραπάνω

συνάρτηση $G2$ ως συνάρτηση αξιολόγησης. Με αυτόν τον τρόπο θα προέκυπταν λύσεις που θα παρείχαν υψηλότερες τιμές από την τιμή της $G2$. Αυτές οι υψηλότερες τιμές θα ήταν προτιμότερες και καλύτερες από αυτές που θα είχαν αποδοθεί ως χαμηλότερες. Πρόκειται για μια σύνθετη διαδικασία που παρουσιάζει κάποιες δυσκολίες. Αυτές οι δυσκολίες αναπαριστώνται στο Σχήμα 3, όπου υπάρχουν *ανέφικτες περιοχές* (infeasible regions), ενώ σε όλα τα *ανέφικτα σημεία* (infeasible points) είναι αντιστοιχισμένη η τιμή μηδέν.



Εικόνα 3: Γράφημα συνάρτησης $G2$ για $n = 2$, όπου η τιμή μηδέν αποδόθηκε στις ανέφικτες λύσεις.

Επίσης, το ενδιαφέρον όριο ανάμεσα σε εφικτές και ανέφικτες περιοχές δίνεται από την εξίσωση:

$\prod_{i=1}^n x_i = 0.75$, ενώ το βέλτιστο αποτέλεσμα εδράζει ή βρίσκεται κοντά σε αυτό το όριο. Ειδικότερα, ο εντοπισμός των ορίων ενός εφικτού τμήματος του χώρου αναζήτησης είναι ιδιαίτερα σύνθετος και δύσκολος. Αυτό αιτιολογείται, καθώς η αναζήτηση των ορίων επιβάλλει ειδικούς τελεστές,

οι οποίοι είναι προσαρμοσμένοι για τον σκοπό αυτό, καθώς και για το συγκεκριμένο πρόβλημα.

Από τα παραπάνω, προκύπτει ότι παρουσιάζεται ένα πρόσθετο επίπεδο δυσκολίας που δεν υπήρχε στις περιπτώσεις του SAT ή του TSP, παρόλο που και στο TSP, όπου δεν επιτρέπονταν η μετάβαση μεταξύ όλων των πιθανών ζευγών πόλεων, κάποιες μεταθέσεις θα ήταν εξίσου ανέφικτες. Συνεπώς, ακόμα και χωρίς αυτήν την πρόσθετη πληροφορία, είναι εμφανές ότι και τα φαινομενικά απλά προβλήματα ενέχουν κομβικές δυσκολίες, λόγω του μεγάλου αριθμού των εναλλακτικών λύσεων. Έτσι προκύπτει πως το μέσο που χρησιμοποιούνται για να δημιουργηθούν οι ποικίλοι τρόποι αξιολόγησης των επιλύσεων δεν είναι ακριβής.

1.3 Μοντελοποίηση του προβλήματος

Αρχικά, η επίλυση ενός προβλήματος είναι ουσιαστικά η λύση ενός μοντέλου του προβλήματος. Πιο συγκεκριμένα, τα μοντέλα των προβλημάτων αναπαριστούν τον φυσικό κόσμο με τον πιο απλό τρόπο, ούτως ώστε να αποφεύγεται η πολυπλοκότητα του πραγματικού κόσμου. Γενικότερα, για να επιλυθεί ένα πρόβλημα προτιμούνται δύο διακριτά βήματα. Το πρώτο αφορά το σχεδιασμό του μοντέλου του εκάστοτε προβλήματος και το δεύτερο αφορά την εφαρμογή αυτού του μοντέλου, ώστε να γενικευθεί η λύση του. Συνοψίζεται σε: Πρόβλημα $\rightarrow$ Μοντέλο $\rightarrow$ Λύση. Η λύση που θα προκύψει είναι ουσιαστικής και χρηστικής, εφόσον το μοντέλο έχει υψηλό βαθμό αξιοπιστίας. Σε περίπτωση που δεν έχει υψηλό βαθμό αξιοπιστίας, δηλαδή ενσωματώνει μη ικανοποιητικές υποθέσεις

και μη καλά σχεδιασμένες προσεγγίσεις, θα προκύψει μια λύση χωρίς νόημα.

Οι τρεις πιο γνωστές μαθηματικές μορφές μοντέλων είναι τα SAT, TSP και NLP, οι οποίες εφαρμόζονται σε ποικίλες διακριτές περιπτώσεις προβλημάτων. Για την καλύτερη κατανόηση πρέπει να αναλογιστεί κάποιος ότι ένα εργοστάσιο οχημάτων που τα παρασκευάζει σε ποικίλα χρώματα. Αν υπάρχουν n χρώματα συνολικά, τότε πρέπει να εντοπιστεί ο βέλτιστος σχεδιασμός παραγωγής που θα μειώσει στο ελάχιστο το συνολικό κόστος βαψίματος των οχημάτων (Michalewicz et al., 2004). Σε αυτή την περίπτωση, θεωρείται ότι, κατά τη διάρκεια των εργασιών, τα μηχανήματα της γραμμής παραγωγής αλλάζουν ανάλογα με το χρώμα που πρέπει να επιτευχθεί. Αυτές οι αλλαγές – ή αλλιώς μεταβολές – ενέχουν κάποιο κόστος, το οποίο εξαρτάται άμεσα από τα δύο χρώματα που θα χρησιμοποιηθούν και τη σειρά χρήσης τους.

Για παράδειγμα, αν θεωρηθεί ότι το κόστος εναλλαγής από το κόκκινο στο μαύρο χρώμα είναι 30 μονάδες, θα υπολογιστεί ανάλογα τα σχετικά διαθέσιμα μέτρα όπως ευρώ ή λεπτά. Ωστόσο, το κόστος μεταβολής από μαύρο σε κόκκινο μπορεί να είναι 80 μονάδες, δηλαδή 50 περισσότερες από πριν. Αυτό σημαίνει ότι τα κόστη εναλλαγής μπορεί να διαφέρουν αρκετά. Με τον ίδιο τρόπο, μπορεί να θεωρηθεί ότι το κόστος εναλλαγής από κόκκινο σε πράσινο είναι 35 μονάδες.

Η βέλτιστη λειτουργία της παραγωγής με σκοπό την ελαχιστοποίηση του κόστους επιβάλλει τον εντοπισμό μιας σειράς εργασιών, οι οποίες θα εξυπηρετούν όλες τις ανάγκες και τις απαιτήσεις παραγωγής ανάλογα με τον αριθμό των οχημάτων που θα βαφτούν. Παράλληλα, καθ' όλη τη διαδικασία πρέπει να διατηρηθεί το κόστος παραγωγής σε ιδιαίτερα

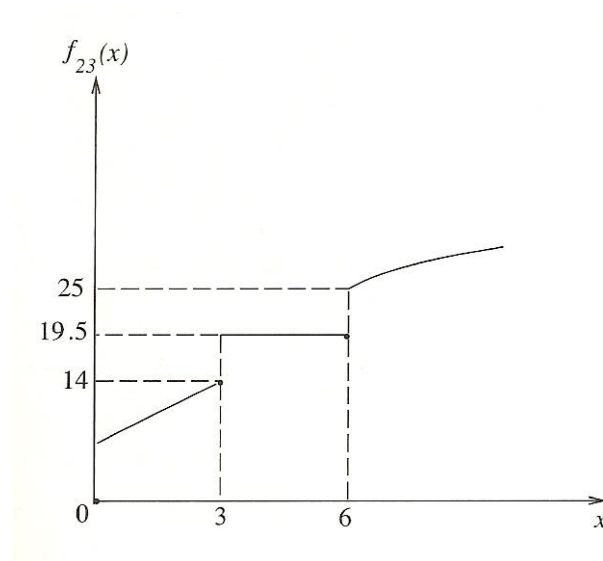
χαμηλό επίπεδο. Η διαδικασία που έχει περιγραφεί, ομοιάζει με ένα TSP, όπου σύμφωνα με το παραπάνω παράδειγμα του πωλητή, η κάθε πόλη αντιστοιχεί στην κάθε εργασία που γίνεται για το βάψιμο ενός αυτοκινήτου με ένα μόνο χρώμα. Αντίστοιχα, η απόσταση ανάμεσα στις πόλεις ομοιάζουν με τα κόστη αλλαγής/ μεταβολής των εργασιών. Στην τελευταία διαδικασία, το TSP δε δύναται να είναι συμμετρικό.

Σε αυτό το σημείο, είναι χρήσιμο να περιγραφεί ένα σενάριο του τρόπου που θα βοηθήσει να προκύψουν απλοποιήσεις στη μοντελοποίηση. Η υπόθεση που θα χρησιμοποιηθεί είναι η ύπαρξη ενός εργοστασίου που ενσωματώνει n αποθήκες για τη φύλαξη χαρτιού. Τα αποθέματα χαρτιού προορίζονται για παράδοση σε k κέντρα διανομής. Συνεπώς, θεωρείται ότι οι αποθήκες του εργοστασίου είναι προελεύσεις και οι τόποι διανομής οι προορισμοί. Κάθε πιθανή διαδρομή διανομής μεταξύ μίας αποθήκης i και ενός κέντρου διανομής j έχει ένα μετρήσιμο κόστος μεταφοράς, που θα οριστεί από την συνάρτηση $f(i,j)$.

Υπάρχει ένα πλήθος παραγόντων, το οποίο καθορίζει το σχήμα που θα πάρει η συνάρτηση. Οι παράγοντες αυτοί μπορούν να αφορούν την απόσταση που υπάρχει από την προέλευση στον προορισμό, την ποιότητα των δρόμων διαδρομής, την κυκλοφοριακή κίνηση, το ποσοστό των στάσεων, το μέσο όριο ταχύτητας από το ένα σημείο στο άλλο κοκ. Η συνάρτηση κόστους μεταφοράς μεταξύ της αποθήκης 2 και του κέντρου διανομής 3 θα οριστεί βάσει αυτής της υπόθεσης ως εξής:

$$f_{23} = \begin{cases} 0 & \text{αν } x = 0 \\ 4 + 3.33x & \text{αν } 0 < x \leq 3 \\ 19.5 & \text{αν } 3 < x \leq 6 \\ 0.5 + 10\sqrt{x} & \text{αν } 6 < x \end{cases}, \text{ όπου το } x \text{ είναι η ποσότητα των}$$

αποθεμάτων που παραδόθηκαν από το σημείο 2 στο σημείο 3. Η συνάρτηση αναπαριστάται στο παρακάτω σχήμα:



Εικόνα 4: Η γραφική αναπαράσταση της συνάρτησης f_{23} για το κόστος μεταφοράς.

Εκ πρώτης όψεως η παραπάνω συνάρτηση είναι ασυνήθιστη. Ωστόσο, είναι χρήσιμη για το σχεδιασμό του μοντέλου αυτού του προβλήματος μεταφοράς. Πιο ειδικά, αναφέρεται ότι εάν δεν πραγματοποιηθεί διανομή, τότε το κόστος θα είναι μηδενικό, ενώ εάν παραδοθούν λιγότες δεσμίδες χαρτιού (έως τρεις), τότε χρησιμοποιείται μια συγκεκριμένη συσκευασία μεταφοράς. Βάσει αυτού, προκύπτει ότι το συνολικό κόστος είναι τέσσερις μονάδες για τη συσκευασία και ότι το επιπλέον κόστος είναι 3.33 μονάδες ανά μια δεσμίδα. Συμπληρωματικά, σε αυτή την περίπτωση παρατηρείται γραμμική αύξηση του κόστους.

Από την άλλη, εάν μεταφερθούν παραπάνω δεσμίδες από τις προαναφερόμενες, αλλά έως έξι, τότε χρησιμοποιείται ένα κουτί που έχει ένα μεταλλικό πλέγμα. Έτσι, προκύπτει ένα σταθερό κόστος 19.5 μονάδων, που θα είναι ανεξάρτητο από την ποσότητα δεσμίδων χαρτιού μεταφοράς. Αν ο αριθμός μεταφοράς υπερβεί τις έξι δεσμίδες, τότε θα χρησιμοποιηθεί ένα ενισχυμένο κιβώτιο, που θα είναι μεγαλύτερο από τα υπόλοιπα. Αυτό σημαίνει ότι το κόστος της μεταφοράς θα εξαρτάται από την ποσότητα των δεσμίδων που μεταφέρονται και θα είναι αυξανόμενος. Η αύξηση αυτή ορίζεται ως η τετραγωνική ρίζα αυτής της ποσότητας συν ένα μικρό κόστος 0.5 μονάδες (Michalewicz et al., 2004).

Όλα αυτά τα στοιχεία λειτουργούν ως δεδομένα που χρησιμοποιούνται για να σχεδιαστεί το μοντέλο αυτού του προβλήματος. Έτσι, προκύπτει:

Ελαχιστοποίηση της

$$\sum_{i=1}^n \sum_{j=1}^k f_{ij} (x_{ij})$$

για

$$\begin{aligned} \sum_{j=1}^k x_{ij} &\leq \text{sour}(i), & \text{for } i = 1, 2, \dots, n \\ \sum_{i=1}^n x_{ij} &\leq \text{dest}(j) & \text{for } j = 1, 2, \dots, k \\ x_{ij} &\geq 0 & \text{for } i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, k \end{aligned}, \text{ όπου sour είναι η}$$

προέλευση και dest ο προορισμός. Επιπλέον, οι περιορισμοί του προβλήματος καθορίζουν μία εφικτή λύση. Η εφικτή λύση σχετίζεται με το ότι καμία μεταφορά από οποιαδήποτε αποθήκη δεν υπερβαίνει τον αριθμό των διαθέσιμων δεσμίδων της αποθήκης, και η συνολική μεταφορά σε οποιοδήποτε κέντρο διανομής πρέπει να ικανοποιεί την

ζήτησή του. Δηλαδή, η συνολική μεταφορά είναι τουλάχιστον ίση με τον αριθμό των δεσμίδων παραγγελίας.

Η παραπάνω συνάρτηση κόστους αναπαριστά με αξιοπιστία –ακρίβεια– το φυσικό περιβάλλον χωρίς να περιλαμβάνει τα κόστη που μπορεί να προκύψουν από άλλους παράγοντες που προαναφέρθηκαν. Αυτό ωθεί στην άποψη ότι υπάρχει η δυνατότητα δημιουργίας συναρτήσεων που θα περιγράφουν τα κόστη μεταφοράς των δεσμίδων χαρτιού από κάθε αποθήκη σε κάθε κέντρο διανομής. Παρόλα αυτά, ακόμα και ένα τέτοιο μοντέλο του προβλήματος που εκφράζει πιστότητα, δεν μπορεί να έχει γενικευμένη εφαρμογή. Αυτό συμβαίνει γιατί οι συναρτήσεις είναι φύσει πολύπλοκες για πολλούς από τους τυπικούς αλγόριθμους βελτιστοποίησης.

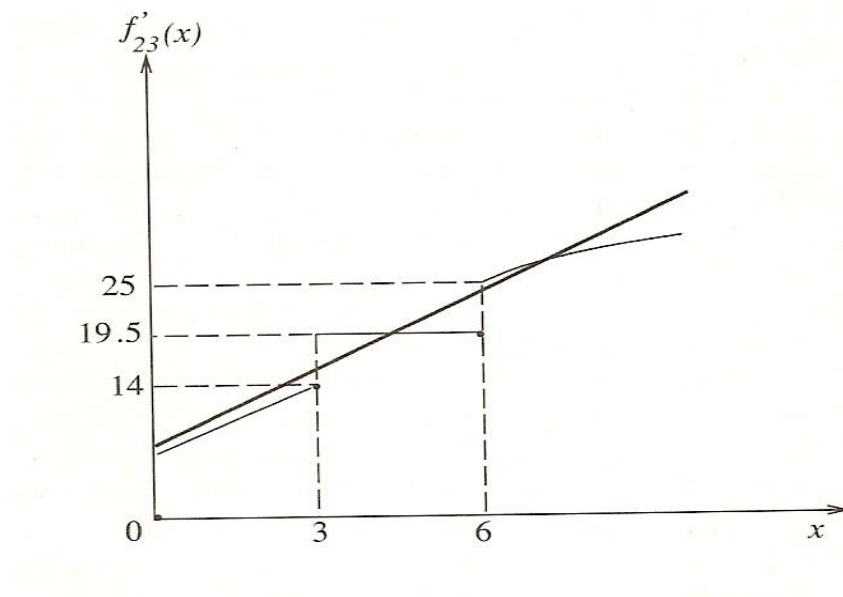
Αν οι συναρτήσεις χρησιμοποιηθούν από κάποιους άπειρους, θα παρουσιάζουν ασυνέχειες που με τη σειρά τους θα επιφέρουν κομβικά προβλήματα. Για παράδειγμα, τα αποτελέσματα που θα προέκυπταν από ορισμένες μεθόδους κλίσεις (gradient-based methods) σε αυτές τις συναρτήσεις δεν θα ήταν αξιόπιστα. Συνεπώς, δε δύναται να αναζητηθεί μία λύση βάσει αυτού του μοντέλου. Αυτό επιδεικνύει ότι όσο καλό και βέλτιστο και αν είναι το μοντέλο, δεν μπορεί να βοηθήσει ουσιαστικά.

Βάσει των παραπάνω προκύπτουν δύο επιλογές που μπορούν να ακολουθηθούν. Η *πρώτη επιλογή* σχετίζεται με την απλοποίηση του μοντέλου κατά τέτοιο τρόπο ώστε οι παραδοσιακές μέθοδοι βελτιστοποίησης να αποδώσουν τις καλύτερες δυνατές απαντήσεις. Η *δεύτερη επιλογή* είναι να χρησιμοποιηθεί το μοντέλο ως έχει και να

ακολουθηθεί μια μη παραδοσιακή προσέγγιση ώστε να αναζητηθεί η βέλτιστη επίλυση του προβλήματος.

Η αλήθεια είναι ότι η πρώτη επιλογή είναι προτιμότερη, γιατί μπορεί να προσεγγιστεί η συνάρτηση f_{23} ως εξής:

$f'_{23}(x) = 2.66x + 8.25$, όπου x είναι ο αριθμός των δεσμίδων που μεταφέρθηκαν από το σημείο 2 στο σημείο 3. Το κόστος μεταφοράς απεικονίζεται στην έντονη μαύρη γραμμή.



Εικόνα 5: Γράφημα προσεγγιστικής συνάρτησης f'_{23} με την αρχική συνάρτηση f_{23} .
Αυτή η συνάρτηση κόστους μεταφοράς f_{23} απλοποιήθηκε, γεγονός που αποκαλύπτει ότι υπάρχει η δυνατότητα απλοποίησης και άλλων συναρτήσεων. Πρέπει, όμως, να αναφερθεί ότι εάν όλα τα f'_{ij} s ήταν γραμμικά, θα προέκυπτε ένα γραμμικό μοντέλο του προβλήματος, το οποίο θα επιλυόταν με σωστό τρόπο μέσω μιας μεθόδου γραμμικού προγραμματισμού. Όμως, αυτή η περίπτωση θα έδινε μια λύση για το απλοποιημένο πρόβλημα και όχι για το πραγματικό.

Από την άλλη, η δεύτερη επιλογή είναι να διατηρηθεί το μοντέλο ως έχει και να χρησιμοποιηθεί μια μη παραδοσιακή μέθοδος για την εύρεση της βέλτιστης επίλυσης του προβλήματος. Ανάλογες μη παραδοσιακές μέθοδοι θεωρούνται η προσομοιωμένη ανόπτηση ή ένας εξελικτικός αλγόριθμος. Τα ποσοστά των πειραμάτων έχουν επιδείξει ότι η τελευταία μη παραδοσιακή προσέγγιση παρουσιάζει τα περισσότερα πλεονεκτήματα σε σχέση με άλλες.

Σε αυτό το σημείο μπορεί να θεωρηθεί ότι υπάρχουν δύο εναλλακτικές προσεγγίσεις. Η μια αφορά τη χρήση ενός προσεγγιστικού μοντέλου, το $Model_a$ ενός προβλήματος, και έπειτα βρίσκει τη σωστή λύση $Solution_p$ ($Model_a$) για αυτό το προσεγγιστικό μοντέλο. Δηλαδή:

$$Pr oblem \Rightarrow Model_a \Rightarrow Solution_p(Model_a)$$

Η δεύτερη εναλλακτική προσέγγιση χρησιμοποιεί ένα ακριβές μοντέλο, το $Model_p$ του προβλήματος, και έπειτα βρίσκει μία προσεγγιστική λύση $Solution_a$ ($Model_p$) αντί της σωστής για αυτό το ακριβές μοντέλο:

$$Pr oblem \Rightarrow Model_p \Rightarrow Solution_a(Model_p)$$

Είναι σημαντικό να αναφερθεί ότι ανάμεσα στις δύο προσεγγίσεις, προτιμάται η δεύτερη για να βρεθεί λύση στο πρόβλημα προς εξέταση, δηλαδή $Solution_a$ ($Model_p$) είναι καλύτερη από $Solution_p$ ($Model_a$) ως λύση στο αρχικό πρόβλημα. Παρότι η επιλογή είναι ελεύθερη, προκύπτει ότι όποια και από τις δύο και αν ακολουθηθεί, θα παρουσιαστούν νέες δυσκολίες στην επίλυση των προβλημάτων. Αυτό συμβαίνει, καθώς η προσέγγιση της βέλτιστης λύσης επιβάλλει την καλύτερη δυνατή προσέγγιση του μοντέλου ή της λύσης του προβλήματος.

2. Ευρετικές Μέθοδοι επίλυσης προβλημάτων

Οι Ευρετικές Μέθοδοι περιλαμβάνουν κάποιους αλγόριθμους που σχετίζονται με τις μερικές ή τις μη ολοκληρωμένες λύσεις. Αυτό συνεπάγεται ότι μπορούν να επιλύουν προβλήματα μέσω ενός τρόπου που «σπάει» τη λύση ανά ένα κομμάτι κάθε φορά.

2.1. Άπληστοι Αλγόριθμοι (Greedy Algorithms)

Οι *Άπληστοι Αλγόριθμοι* χρησιμοποιούνται για την επίλυση ενός προβλήματος διά μέσου που αντιτίθεται το σύστημα ολοκληρωμένων λύσεων βάσει μια καθορισμένης σειράς βημάτων. Πρόκειται για μια ιδιαίτερα γνωστή και πολυχρησιμοποιούμενη κατηγορία, γιατί χαρακτηρίζονται από απλότητα. Γίνεται λόγος, λοιπόν, για μια άπληστη προσέγγιση εξαιτίας αυτής της απλότητας που τα χαρακτηρίζει. Πιο συγκεκριμένα, η διαδικασία που επιτελείται αφορά στην μια προς μια αντιστοίχιση των τιμών σε όλες τις μεταβλητές αποφάσεων, όπου λαμβάνεται η βέλτιστη απάντηση για κάθε μια από αυτές τις αντιστοιχίσεις (Barron et al., 2008).

Η διαδικασία που γίνεται με τη χρήση άπληστων αλγόριθμων επιβάλλει τη χρήση μιας Ευρετικής Μεθόδου, ούτως ώστε να εξασφαλίζεται ότι η κάθε κίνηση του εκάστοτε βήματος θα είναι η βέλτιστη. Δηλαδή, θα αποφέρει το πιο υψηλό όφελος και για αυτό το λόγο η προσέγγιση ονομάζεται *άπληστη*. Βέβαια, αν παρατηρηθεί πιο προσεκτικά αυτή η προσέγγιση, διαφαίνεται πως δεν είναι απαραίτητο να προκύπτει η βέλτιστη τελική λύση από την βέλτιστη επιλογή κινήσεων σε κάθε βήμα στη διαδικασία της επίλυσης.

2.1.1. Ο Αλγόριθμος του Dijkstra

Ο αλγόριθμος Dijkstra χρησιμοποιείται με σκοπό να εντοπιστούν οι πιο σύντομες διαδρομές (Single-Source Shortest Path Problem) από το σημείο εκκίνησης σε έναν γράφο με μη αρνητικά βάρη στις ακμές είτε αυτός είναι κατευθυνόμενος είτε δεν είναι. Ο Dijkstra ανήκει στους άπληστους αλγορίθμους, αφού κάθε βήμα του χαρακτηρίζεται από την βέλτιστη λύση σε τοπικό επίπεδο μέχρι να φτάσει στο τελευταίο βήμα που θα δώσει τη γενικά βέλτιστη λύση του προβλήματος. Ειδικότερα, αν ο γράφος αποτελείται από αρνητικά βάρη, δε θα προκύψει σωστό αποτέλεσμα από αυτόν τον αλγόριθμο (Deng et al., 2012).

Γενικότερα, πρέπει να παρουσιαστεί η διαδικασία που ακολουθούν οι λειτουργίες αυτού του αλγορίθμου, οι οποίες ενσωματώνουν κάποια ορισμένα βήματα. Αρχικά, πρέπει να δοθεί σε κάθε κόμβο μια ετικέτα απόστασης *ετικέτα απόστασης* ($d[*]$) με τιμή 0 στον αρχικό κόμβο και τιμή *άπειρο* σε όλους τους υπόλοιπους. Εν συνεχεία, πρέπει να δοθεί μια *ετικέτα προηγούμενου κόμβου* ($prev[*]$), όπου θα λάβει την κενή τιμή για όλους τους κόμβους. Η ετικέτα είναι απόλυτα χρήσιμη, αφού μέσω αυτής γίνεται ο υπολογισμός της ζητούμενης διαδρομής στο τέλος της διαδικασίας.

Το δεύτερο βήμα σχετίζεται με την επισήμανση όλων των κόμβων ως μη επεξεργασμένων, ενώ παράλληλα ο κόμβος που βρίσκεται σε χρήση θα είναι ο αρχικός. Αυτός ο τρέχων κόμβος επιβάλλει τον έλεγχο των μη επεξεργασμένων γειτόνων συνολικά για να υπολογιστεί, εν συνεχεία, το συνολικό άθροισμα της απόστασης που έχουν οι μη επεξεργασμένοι γείτονες από τον αρχικό κόμβο. Για να κατανοηθεί καλύτερα αυτό, πρέπει να παρουσιαστεί μια υπόθεση. Δηλαδή, εάν ο τρέχων κόμβος βρίσκεται σε

απόσταση 6 από τον αρχικό και εφόσον ο γείτονας με τον τρέχοντα κόμβο βρίσκεται σε απόσταση 2, τότε το συνολικό άθροισμα της απόστασης μεταξύ γείτονα και αρχικού κόμβου, θα είναι $6+2=8$. Αν η επισήμανση της ετικέτας είναι μεγαλύτερη από αυτή την τελική απόσταση, τότε θα πρέπει να επισημανθεί ξανά με την προκύπτουσα νέα τιμή και εν συνεχεία ο τρέχων κόμβος θα καταγραφεί στην *ετικέτα προηγούμενου κόμβου*.

Έπειτα, το επόμενο βήμα περιλαμβάνει τον τελικό έλεγχο των γειτόνων του τρέχοντος κόμβου, ο οποίος πρέπει να επισημανθεί ως ο πλέον επεξεργασμένος. Από τη στιγμή που θα χαρακτηριστεί επεξεργασμένος, δεν επανεξετάζεται από τον αλγόριθμο και η ετικέτα απόστασης που έχει δοθεί παραμένει ως ελάχιστη και σταθερή. Εν συνεχεία, ακολουθείται η ίδια διαδικασία για τον επόμενο τρέχοντα κόμβο, ο οποίος τώρα θα είναι ο μη επεξεργασμένος κόμβος και θα λάβει τη μικρότερη ετικέτα απόστασης. Το επόμενο βήμα συντελείται, εφόσον όλοι οι κόμβοι έχουν τεθεί ως επεξεργασμένοι. Έτσι, το επόμενο βήμα εκκινεί από τον τελευταίο τρέχοντα κόμβο, ο οποίος ορίζεται ως κόμβος-προορισμός. Αυτός θα εκτυπωθεί βάσει της επισήμανσης της *ετικέτας προηγούμενου κόμβου* και η διαδικασία θα συνεχίζεται μέχρι να «αδειάσει» η *ετικέτα προηγούμενου κόμβου*.

Αναφέρεται ξανά ότι ο αλγόριθμος Dijkstra είναι άπληστος, καθώς η γενικά τελική λύση προκύπτει από τον κάθε υπολογισμό της τοπικά βέλτιστης λύσης. Με άλλα λόγια, ενοποιεί τις τοπικές λύσεις για να παρέχει την τελική γενική λύση. Ωστόσο, επισημαίνεται ξανά πως η τελική λύση δε συνεπάγεται και τη βέλτιστη λύση, εφόσον δεν είναι γνωστό και ακριβές εάν οι τοπικές λύσεις ήταν οι βέλτιστες. Ως εκ τούτου,

θα πρέπει να αποδειχθεί η αξιοπιστία του αλγορίθμου ως προς τα αποτελέσματα που έδωσε.

2.2. Διαίρει και βασίλευε (Divide and conquer)

Σε πολλές περιπτώσεις, η επίλυση ενός πολύπλοκου προβλήματος προτιμάται να γίνεται με τη διάσπασή του σε μικρότερα προβλήματα, τα οποία θα είναι και πιο απλά από το αρχικό πρόβλημα. Σκοπός είναι να επιλυθεί το κάθε πρόβλημα χωριστά και να παραχθεί μια μόνο απάντηση για καθένα από αυτά τα μικρότερα προβλήματα. Αυτή η διαδικασία μπορεί να επιτευχθεί μέσω της προσέγγισης *Διαίρει και Βασίλευε*, η οποία αποκτά αξία και χαμηλό κόστος υπό ορισμένες προϋποθέσεις (Saxe et al., 2006).

Δηλαδή, πρέπει ο χρόνος, που θα σπαταληθεί για τη διεξαγωγή της επίλυσης όλων των μικρότερων προβλημάτων και την εξαγωγή τελικής απάντησης, να είναι σαφώς λιγότερος από αυτόν που θα χρειαστεί για να επιλυθεί συνολικά το πρώτο πολύπλοκο πρόβλημα. Συνήθως, η ανάλυση των μικρότερων προβλημάτων επιφέρει τα επιθυμητά αποτελέσματα και λύσεις, αλλά, παράλληλα, αυτή η διάσπαση του αρχικού προβλήματος μπορεί να απομακρύνει την εξαγωγή του συνολικά βέλτιστου αποτελέσματος. Η παρουσίαση αυτής της προσέγγισης φαίνεται σχηματικά παρακάτω:

procedure D&C (P)

begin

 ανάλυσε το πρόβλημα P στα υπό-προβλήματα $P_1 \dots P_k$

for $i=1$ **to** k **do**

if $\text{size}(P_i) < P$ **then** επίλυσε το P_i (παίρνοντας λύση s_i)

else $s_i \leftarrow \text{D\&C}(P_i)$

 συνδύασε τις λύσεις s_i ώστε να πάρεις την τελική λύση

end

Εικόνα 6: Επαναληπτική διαδικασία Διαίρει και Βασίλευε για επίλυση του προβλήματος P .

Αναλυτικά, το αρχικό πρόβλημα ορίζεται ως P και βάσει αυτού δημιουργούνται μικρότερα προβλήματα, τα οποία δύναται να διασπώνται ξανά και ξανά σε άλλα μικρότερα. Η διαδικασία πραγματώνεται επαναληπτικά μέχρι τα διασπώμενα προβλήματα να γίνουν όσο πιο απλά γίνεται. Δηλαδή, να γίνουν μικρότερα από μια εννοιολογική σταθερά P . Ο βασικός στόχος είναι να μπορεί κάποιος να τα λύσει χωρίς τη χρήση ηλεκτρονικού μέσου. Η λειτουργία του αλγορίθμου επιτελείται ανοδικά, καθώς πραγματώνει ένα συνδυασμό των λύσεων για να προκύψει μια επίλυση για τα πιο μεγάλα προβλήματα, που, όμως, είναι ακόμη μικρότερα του αρχικού προβλήματος.

Αποτελεί γεγονός ότι αυτή η προσέγγιση χρησιμοποιείται από αρκετούς αλγορίθμους **Ταξινόμησης** όπως σε:

- Αλγόριθμους Ταχείας Ταξινόμησης (quick sort)
- Αλγόριθμους Ταξινόμησης με συγχώνευση (merge sort)

Τέλος, αυτή η προσέγγιση χρησιμοποιείται και από πολωνυμικούς πολλαπλασιασμούς ή πολλαπλασιασμούς πίνακα.

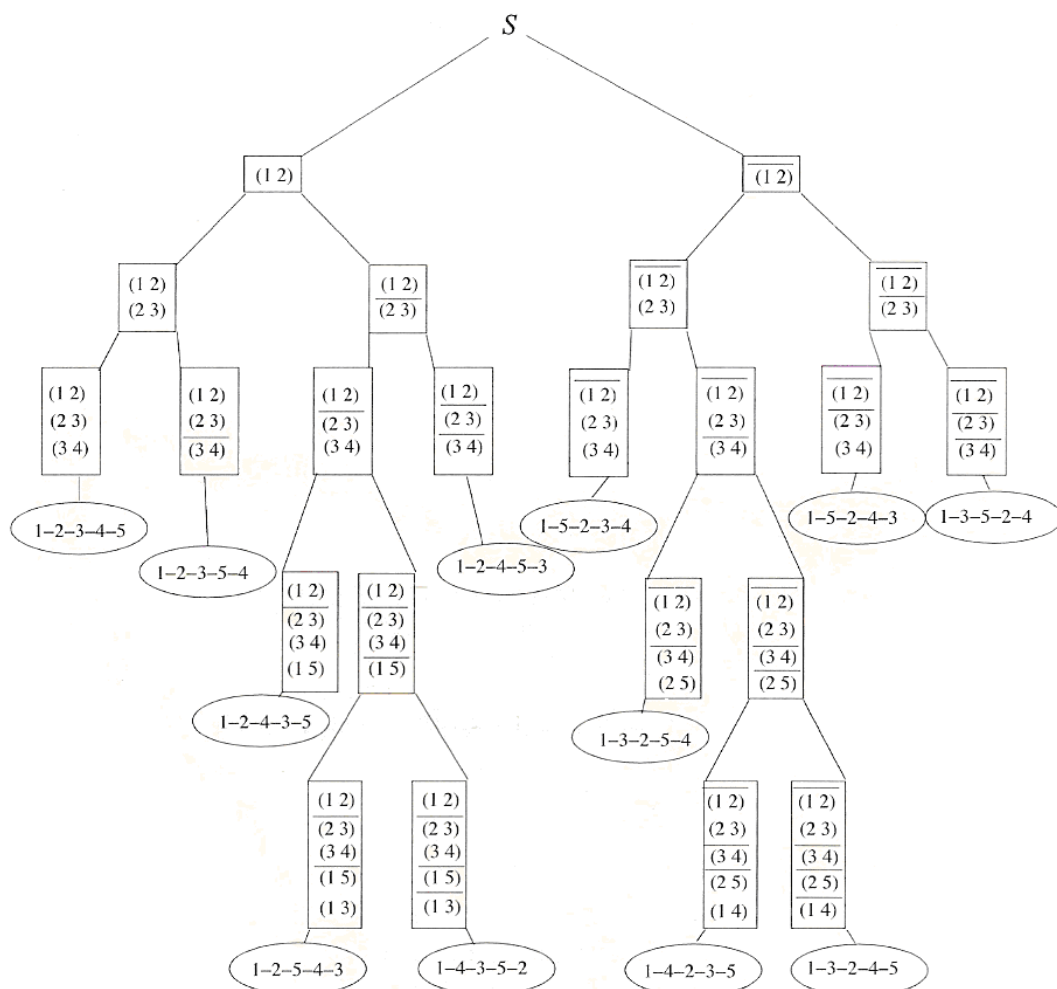
2.3. Διακλάδωση και Οριοθέτηση (Branch and Bound)

Αποτελεί γεγονός ότι η αύξηση των μεταβλητών ενός προβλήματος επηρεάζουν και το μέγεθος –δηλαδή την αύξησή του- των προβλημάτων του φυσικού περιβάλλοντος. Αρχικά, υπάρχουν $(n - 1)!/2$ διαφορετικές λύσεις για το συμμετρικό TSP. Αυτό σημαίνει ότι δεν μπορεί να πραγματοποιηθεί ολοκληρωτική διερεύνηση για $n > 20$. Συνεπώς, κρίνεται σημαντικό να γίνει χρήση ενός ευρετικού που θα μπορεί να διαγράφει τα τμήματα του χώρου αναζήτησης, τα οποία δεν παρέχουν τη βέλτιστη λύση που επιθυμείται.

Μια τέτοια ευρετική μέθοδος θεωρείται η προσέγγιση της Διακλάδωσης και Οριοθέτησης, η οποία διαιρεί τον χώρο αναζήτησης με διαδοχικό τρόπο. Πιο συγκεκριμένα, απαιτείται ένα εργαλείο που θα μπορεί να εντοπίσει το πιο χαμηλό όριο κόστους για μια εκάστοτε λύση ή το πιο υψηλό όριο. Η επιλογή αυτή σχετίζεται με την επιθυμία ελαχιστοποίησης ή μεγιστοποίησης (D'ariano et al., 2007). Πιο αναλυτικά, αν πραγματώνεται ελαχιστοποίηση, και εφόσον υπάρχει μια λύση με κόστος c μονάδων και είναι γνωστοποιημένο ότι η επόμενη λύση έχει κάτω όριο μεγαλύτερο των c , τότε δεν απαιτείται έλεγχος ως προς το εάν είναι λανθασμένη η λύση. Παύει να ενδιαφέρει αυτή η λύση και η διαδικασία προχωρά στην επόμενη πιθανή λύση.

Επιπρόσθετα, ο χώρος αναζήτησης αναπαριστάται με ένα δενδροδιάγραμμα. Η χρήση της ευρετικής μεθόδου θα αποβάλλει τα κλαδιά που δε βρίσκονται στον πυρήνα του ενδιαφέροντος. Αυτή η άποψη μπορεί να αποσαφηνιστεί από την υπόθεση ύπαρξης ενός συμμετρικού TSP με πέντε πόλεις. Σε αυτό το παράδειγμα, ο χώρος αναζήτησης

ορίζεται ως S , και διαιρείται βάσει του εάν η ακμή $(1\ 2)$ αποτελεί ή όχι μέρος μιας διαδρομής. Ταυτόχρονα, ο χώρος αναζήτησης δύναται να διαιρείται περαιτέρω βάσει του εάν η ακμή $(2\ 3)$ αποτελεί ή όχι μέρος της διαδρομής. Η παραπάνω διαδικασία απεικονίζεται στο παρακάτω δενδροδιάγραμμα, του οποίου τα κλαδιά αναπαριστούν τις πιθανές διαδρομές ως $(5-1)!/2 = 12$. Η οριζόντια γραμμή πάνω στην ακμή αποτελεί την τελευταία περίπτωση, ενώ παράλληλα επισημαίνεται ότι οι ακμές προέκυψαν από τυχαία επιλογή σειράς.



Εικόνα 7:Μια διαίρεση του χώρου αναζήτησης S του TSP πέντε πόλεων, ανάλογα με το εάν μία δεδομένη ακμή (ij) ανήκει ή όχι στην διαδρομή.

Επίσης, το κόστος του ταξιδιού μεταξύ των πόλεων δίνεται από την παρακάτω υπόθεση:

$$\begin{bmatrix} 0 & 7 & 12 & 8 & 11 \\ 7 & 0 & 10 & 7 & 13 \\ 12 & 10 & 0 & 9 & 12 \\ 8 & 7 & 9 & 0 & 10 \\ 11 & 13 & 12 & 10 & 0 \end{bmatrix}$$

, όπου η κάθε καταγραφή αποτελεί το κόστος του ταξιδιού από τον έναν προορισμό στην i σειρά σε έναν προορισμό στην j στήλη. Η παρουσία μηδενικών αριθμών διευκρινίζουν ότι δεν είναι δυνατή η μεταφορά από τη μια πόλη στην ίδια πόλη. Επιπλέον, εφόσον ο χώρος αναζήτησης αναπαριστάται με τη μορφή δενδροδιαγράμματος, επιβάλλεται η χρήση ενός ευρετικού με σκοπό τον υπολογισμό του κάτω ορίου κόστους για την εκάστοτε τελική λύση. Ο υπολογισμός έγκειται και στον κάθε υπαρκτό κόμβο. Συνεπώς, αν το κάτω όριο εντοπιστεί να είναι μεγαλύτερο από το κόστος της μέχρι τώρα βέλτιστης λύσης, τότε υπάρχει η δυνατότητα να αναιρεθεί η καινούρια λύση. Αυτό σημαίνει ότι πρέπει να συνεχιστεί η αναζήτηση ανεξάρτητα από το αν θα υπολογιστεί το ακριβές κόστος.

Η αρχιτεκτονική της προσέγγισης Διακλάδωσης και Οριοθέτησης δίνεται παρακάτω στην Εικόνα 8.

procedure branch and bound

begin

αρχικοποίησε το C

αρχικοποίησε το f_{bound}

while(συνθήκη μη-τερματισμού) **do**

αφαίρεσε καλύτερο πλαίσιο $C \rightarrow D_i$

μείωσε ή υποδιαίρεσε $D_i \rightarrow D'_i$

ενημέρωσε το f_{bound}

```

 $C \leftarrow C \cup D'_i$ 
for all  $D_i \in C$  do
    if (κάτω όριο της  $f(D_i)$ )  $> f_{bound}$  then αφαίρεσε το  $D_i$  από
    το  $C$ 
end

```

Εικόνα 8:Αλγόριθμος διακλάδωσης και οριοθέτησης

2.4. Αλγόριθμος A^*

Επισημάνθηκε πολλές φορές ότι η βέλτιστη κάθε φορά κίνηση, παρότι διευκολύνει, μπορεί να παρουσιάσει διάφορες αδυναμίες. Το βασικό γεγονός είναι ότι η χρήση των Άπληστων Αλγορίθμων δύναται να εξυπηρετεί προσωρινά, αλλά δε συνεπάγεται την τελικά βέλτιστη απόδοση. Η ορθή χρήση τέτοιων αλγορίθμων επιβάλλει τη δημιουργία μιας συνάρτησης αξιολόγησης, η οποία θα λειτουργεί ενημερωτικά ώστε να αποφεύγονται τα αναξιόπιστα αποτελέσματα. Αυτή η άποψη εκπληρώνεται από τον Αλγόριθμο A^* , ο οποίος αποτελεί τη γνωστή αναζήτηση *πρώτα- στο- καλύτερο* (best-first search) (Gil-García et al., 2003).

Η κατασκευή ενός δενδροδιαγράμματος για τον χώρο αναζήτησης αναπαριστά τις περιπτώσεις κατευθυνόμενων γραφημάτων. Σε αυτό το δέντρο μπορεί να εντοπιστεί ο τρόπος επιλογής αναζήτησης. Εν προκειμένω, η χρήση εστιάζεται στην αναζήτηση *πρώτα-κατά-βάθος* (depth-first search), όπου, έπειτα, η αναζήτηση συνεχίζεται καθοδικά στα επίπεδα του δέντρου για να επέλθει μια αξιολόγηση και να πραγματοποιηθεί και η αντίστοιχη διαδρομή.

Επιπλέον, παρέχεται η δυνατότητα, μέσω ενός ευρετικού, να τοποθετηθούν οι διαθέσιμοι κόμβοι βάσει των αναμενόμενων ή προσδοκώμενων απαντήσεων στις λύσεις. Αυτό συμβαίνει όταν η

αναζήτηση θα έχει φτάσει στο βαθύτερο επίπεδο. Σημαντικό είναι ότι αναζητούνται και εντοπίζονται οι κόμβοι, που θα παρέχουν τον εντοπισμό της βέλτιστης λύσης. Η αναζήτηση *πρώτα-στο-καλύτερο* παρουσιάζεται στην Εικόνα 9:

```
procedure best-first(v)  
begin  
    επισκέψου τον κόμβο  
    for κάθε διαθέσιμο w do  
        όρισε μια ευρετική τιμή για το w  
        q ← τον καλύτερο διαθέσιμο κόμβο  
        best-first(q)  
end
```

Εικόνα 9: Αναζήτηση πρώτα-στο-καλύτερο (Best-first search).

3. Υπολογιστικές Μέθοδοι για επίλυση προβλημάτων

3.1. Εισαγωγή

Υπάρχουν κάποιες προσεγγίσεις για επίλυση προβλημάτων που ονομάζονται ευφυείς. Ο χαρακτηρισμός αυτός σχετίζεται με τη λειτουργία των συστημάτων να μαθαίνουν νέες προσεγγίσεις. Πριν την αναλυτική περιγραφή αυτών των προσεγγίσεων, πρέπει να αποσαφηνιστούν κάποιες βασικές έννοιες και ορισμοί.

❖ *Μάθηση* (Learning)

Η *Μάθηση* αποτελεί μια εν γένει ιδιότητα του ανθρώπου, που σχετίζεται με την επιθυμία του να μαθαίνει. Η Γνωστική Ψυχολογία και η Φιλοσοφία πραγματεύονται αυτή την έννοια επί χρόνια και παρά τις προσπάθειές τους συνεχίζει να μην έχει περιγραφεί στο πλήρες εύρος της. Βάσει αυτής της διαπίστωσης, αναδύεται το ερώτημα του κατά πόσον οι ερευνητές της ΤΝ δύνανται να κατασκευάσουν υπολογιστικά συστήματα που θα βασίζονται στη *Μηχανική Μάθηση*, από τη στιγμή που δεν είναι πλήρως κατανοητή η έννοια της *Μάθησης*.

❖ *Μηχανική Μάθηση* (Machine Learning)

Η *Μηχανική Μάθηση* ορίζεται ως το φαινόμενο, όπου ένα σύστημα βελτιώνεται αποδοτικά κατά την επιτέλεση μιας ορισμένης εργασίας. Στον ορισμό ενσωματώνεται και η απουσία ανάγκης εκ νέου προγραμματισμού (Nasrabadi et al., 2007). Πιο αναλυτικά, η *Μηχανική*

Μάθηση στοχεύει στην κατασκευή ικανών μηχανών που μπορούν να μαθαίνουν ώστε να βελτιώνουν κάθε φορά την απόδοσή τους σε ορισμένους τομείς. Δηλαδή, να δύνανται να εκμεταλλεύονται ορθά την προγενέστερη εμπειρία και γνώση. Ο ορισμός που δίνεται από τον Mitchell (1997), αναφέρει: «Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P , αν η απόδοσή του σε εργασίες από το T , όπως μετριέται από το P , βελτιώνεται μέσω της εμπειρίας E ».

❖ Επαγωγική Μάθηση (Inductive Learning)

Η Επαγωγική Μάθηση ενσωματώνει τη διαδικασία της *επαγωγής* (induction), κατά την οποία ένας άνθρωπος μπορεί να μάθει δια μέσου της κατανόησης του φυσικού περιβάλλοντος. Αυτή η κατανόηση επιτελείται από τις παρατηρήσεις που πράττει, όπου εν συνεχεία δια μέσου της αφαιρετικής σκέψης παράγει μια απλοποιημένη εκδοχή του πραγματικού κόσμου. Όταν το κάνει αυτό, ο άνθρωπος έχει κατασκευάσει ένα νοητικό μοντέλο (mental model). Παράλληλα, ο άνθρωπος δύναται να βάζει σε οργάνωση τις όποιες εμπειρίες του και να πραγματώνει συσχετίσεις ανάμεσα σε αυτές. Επιπλέον, οι πολλαπλές παρατηρήσεις δημιουργούν αλληπάλληλες νέες δομές, δηλαδή τα νοητικά πρότυπα (mental patterns). Αυτά τα νοητικά πρότυπα δημιουργούνται δια μέσου επαγωγικών και απαγωγικών συλλογισμών.

Τα νέα πρότυπα βασίζονται σε ήδη υπάρχοντα, δηλαδή παλαιά πρότυπα. Συνεπώς, δημιουργούνται τρόποι μάθησης που βασίζονται σε προγενέστερες γνώσεις για την επίλυση προβλημάτων. Για παράδειγμα, τρόποι μάθησης θεωρούνται οι επεξηγήσεις και η μάθηση από περιπτώσεις. Η προαναφερόμενη φύσει ικανότητα του ανθρώπου να

μαθαίνει, έχει οδηγήσει τους φιλοσόφους στο ερώτημα του κατά πόσον ένας επαγωγικός συλλογισμός που θα επιφέρει μάθηση μπορεί να κριθεί για το αν είναι ορθός και κατάλληλος. Επιπρόσθετα, η ψυχολογία έχει αναρωτηθεί για τους τρόπους που ο ανθρώπινος νους αποθηκεύει τα αποτελέσματα που λαμβάνει από τη μάθηση (Michalski et al., 2013). Αυτά τα αποτελέσματα είναι τα νοητικά μοντέλα και τα νοητικά πρότυπα. Από την άλλη, οι επιστήμονες της ΤΝ αναρωτιούνται για τους τρόπους που μια μηχανή δύναται να σχεδιάζει νέα πρότυπα και μοντέλα μάθησης, τα οποία προκύπτουν από εξειδικευμένα παραδείγματα. Παράλληλα, αναζητούν το ποσοστό της αξιοπιστίας τους στην πράξη.

Σύμφωνα με τα παραπάνω, επιστρέφει η έννοια της *Μηχανικής Μάθησης* για να δοθεί ένας εναλλακτικός ορισμός για αυτήν. Έτσι, μπορεί να λεχθεί ότι η *Μηχανική Μάθηση* αποτελεί την ικανότητα ενός υπολογιστικού συστήματος, όπου η άντληση από ένα σύνολο δεδομένων, οδηγεί στην κατασκευή νέων μοντέλων και προτύπων. Ειδικότερα, η *Μηχανική Μάθηση* αποτελεί κλάδο της ΤΝ και απασχολεί τη μελέτη αλγορίθμων που μπορούν να βελτιώνονται καθ' όλη τη διάρκεια επιτέλεσης μιας εργασίας βάσει της ανάκλησης εμπειριών και γνώσεων.

Τα συστήματα της Μηχανικής Μάθησης αποτελούν κομμάτι της Συμβολικής ΤΝ. Σε αυτά τα συστήματα η δυνατότητα μάθησης αφορά την ικανότητα απόκτησης και κατάκτησης περαιτέρω γνώσεων. Η εισαγωγή αυτών των γνώσεων τροποποιεί τις ήδη υπάρχουσες γνώσεις με το να μεταβάλλει τα χαρακτηριστικά τους ή να προσθέτει ή να αφαιρεί πληροφορίες. Από την άλλη, τα συστήματα που ανήκουν στη Μη Συμβολική ΤΝ, όπως τα Τεχνητά Νευρωνικά Δίκτυα, παρουσιάζουν τη μάθηση ως μια δυνατότητα των ίδιων των συστημάτων να τροποποιούν

την δομή τους εσωτερικά. Δηλαδή, δεν προσανατολίζονται στο να τροποποιούν τη γνώση κατά το σχεδιασμό τους.

Είναι εύλογο πως η κατασκευή τέτοιων μηχανών δεν μπορεί να φτάσει το ανθρώπινο επίπεδο μάθησης. Ωστόσο, έχουν σχεδιαστεί αλγόριθμοι που σχετίζονται με ορισμένες περιοχές μάθησης. Η δημιουργία τέτοιων αλγορίθμων έχει οδηγήσει στην παρουσία νέων σύγχρονων εμπορικών εφαρμογών, οι οποίες είναι επιτυχείς. Οι τρόποι εφαρμογών της ΤΝ έχουν οδηγήσει σε εμφανή αποτελέσματα, που προσφέρουν απαντήσεις σε ερωτήματα που ήταν άλυτα ως τώρα. Επίσης, προσφέρονται απαντήσεις σε ερωτήματα διαφόρων κλάδων που ασχολούνται με την ανθρώπινη ικανότητα μάθησης.

❖ *Εξελικτική Μάθηση (Evolutionary Learning)*

Συμπληρωματικά, πρέπει να αναφερθεί ότι η *Μηχανική Μάθηση* ως τομέας της ΤΝ, ενσωματώνει και εξελίσσει την *Εξελικτική Μάθηση*. Η τελευταία προσπαθεί να αναπαριστά μιμητικά τη διαδικασία της φυσικής αναπαραγωγικής σε έμβια όντα και εφαρμόζεται σε προβλήματα βελτιστοποίησης κατά κύριο λόγο. Σημαντικό είναι ότι οι αλγόριθμοι που χρησιμοποιούνται στην *Εξελικτική Μάθηση* είναι οι γενετικοί (Ponsen et al., 2006).

Τέλος, πέρα από τον κλάδο της ΤΝ, η *Μηχανική Μάθηση* απασχολεί και άλλους επιστημονικούς τομείς όπως:

- ✓ Πιθανότητες και Στατιστική
- ✓ Αριθμητική Βελτιστοποίηση
- ✓ Εξόρυξη Δεδομένων
- ✓ Θεωρία της Πληροφορίας
- ✓ Θεωρία της Πολυπλοκότητας

- ✓ Προσαρμοστική Θεωρία ή αλλιώς Θεωρία Ελέγχου
- ✓ Γνωστική και Εξελικτική Ψυχολογία
- ✓ Γνωστική Γλωσσολογία
- ✓ Νευροεπιστήμες όπως Νευροβιολογία, Νευρογλωσσολογία και Βιογλωσσολογία

3.2. Κατηγορίες Μηχανικής Μάθησης

Η *Μηχανική Μάθηση* περιλαμβάνει τρεις διαδικασίες μάθησης. Αυτές σχετίζονται με τους τρόπους μάθησης του ανθρώπινου νου και συνοπτικά είναι η *Επιβλεπόμενη Μάθηση*, η *Μη Επιβλεπόμενη Μάθηση* και η *Ενισχυτική Μάθηση*. Παρακάτω περιγράφονται αναλυτικά οι τρεις κατηγορίες μάθησης.

❖ *Επιβλεπόμενη Μάθηση* (Supervised Learning)

Αυτός ο τρόπος μάθησης αφορά τη διαδικασία, όπου χρησιμοποιείται ένας αλγόριθμος ο οποίος δημιουργεί μια συνάρτηση. Η συνάρτηση αυτή αναπαριστά δεδομένες εισόδους (σύνολο εκπαίδευσης) σε σχέση με τις γνωστές επιθυμητές εξόδους. Ο βασικός στόχος είναι να γενικευτεί η συνάρτηση ώστε να αναπαριστά εισόδους με άγνωστες εξόδους. Αυτός ο τύπος μάθησης εφαρμόζεται σε προβλήματα *Ταξινόμησης* (Classification), *Πρόγνωσης* (Prediction) και *Διερμηνείας* (Interpretation) (Caruana et al., 2006).

❖ *Μη Επιβλεπόμενη Μάθηση* (Unsupervised Learning)

Η *Μη Επιβλεπόμενη Μάθηση* αποτελεί μια διαδικασία όπου ο επιλεγμένος αλγόριθμος θα δημιουργήσει ένα μοντέλο για ένα ορισμένο σύνολο εισόδων, που θα έχει τη μορφή παρατηρήσεων. Σημαντικό είναι

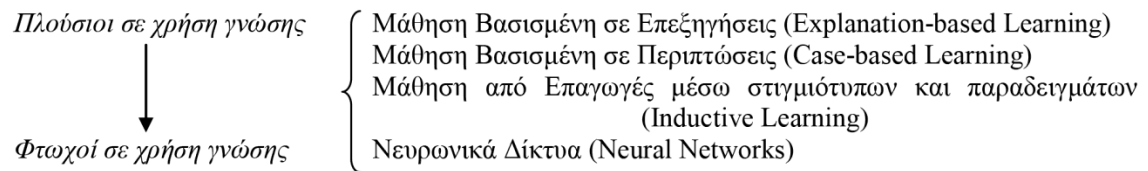
ότι σε αυτή την περίπτωση το κατασκευασμένο μοντέλο δε γνωρίζει τις εξόδους. Η Μη Επιβλεπόμενη Μάθηση εφαρμόζεται σε προβλήματα *Ανάλυσης Συσχετισμών* (Association Analysis) και *Ομαδοποίησης* (Clustering).

❖ *Ενισχυτική Μάθηση* (Reinforcement Learning)

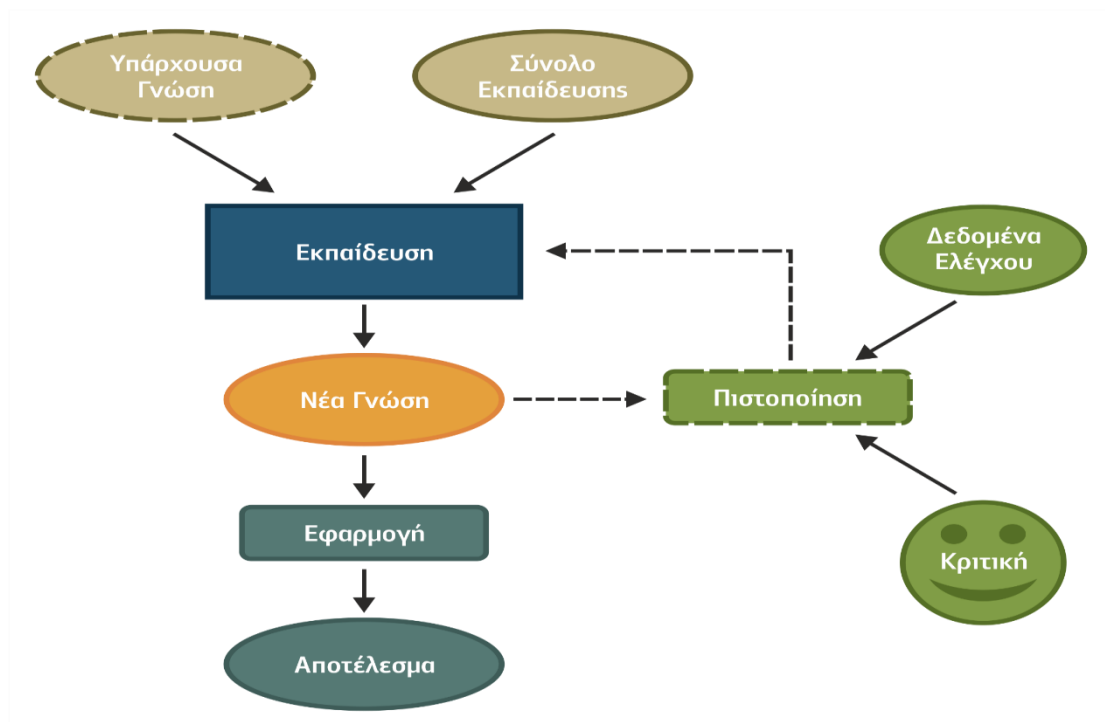
Η Ενισχυτική Μάθηση χρησιμοποιεί έναν αλγόριθμο, ο οποίος έχει την ικανότητα να μάθει μια σειρά από ενέργειες μέσω της ενεργής αλληλεπίδρασής του με το περιβάλλον. Αυτός ο τύπος μάθησης εφαρμόζεται σε προβλήματα *Σχεδιασμού* (Planning). Χαρακτηριστικά τέτοια προβλήματα είναι ο έλεγχος κίνησης των ρομπότ, καθώς και η βελτιστοποίηση εργασιών σε τόπους εργοστασίων.

Επιπρόσθετα, πρέπει να αναφερθεί ότι το εκάστοτε πρόβλημα, που διεξάγεται στο χώρο της Μηχανικής Μάθησης, επιδέχεται έναν ορθό τρόπο μάθησης. Εν συνεχεία, ο κάθε τρόπος μάθησης χρησιμοποιεί έναν συγκεκριμένο κατάλληλο αλγόριθμο. Γενικότερα, οι αλγόριθμοι της Μηχανικής Μάθησης χρησιμοποιούν τη γνώση με τέτοιο τρόπο, ώστε να την αναπαριστούν κατάλληλα. Αυτό σημαίνει ότι την αναπαριστούν με τρόπους μαθηματικής φύσης, οι οποίοι επιλέγονται ως οι πιο κατάλληλοι από τον αλγόριθμο για να εκφραστεί η προσδοκώμενη γνώση.

Από την άλλη, υπάρχουν άλλοι αλγόριθμοι που επιδέχονται μόνο την είσοδο παρατηρήσεων, όπου χρησιμοποιούν την προγενέστερη γνώση περισσότερο ή λιγότερο. Η ταξινόμηση των αλγορίθμων βάσει του ποσοστού που προσεγγίζουν την προϋπάρχουσα γνώση (λιγότερο ή περισσότερο) συνοψίζεται στο σχήμα:



Η εικόνα δείχνει γραφικά τον πιο γενικό τρόπο που εφαρμόζονται και λειτουργούν οι αλγόριθμοι στη Μηχανική Μάθηση. Πρέπει να επισημανθεί ότι το σημαντικότερο βήμα των αλγορίθμων γενικότερα είναι η *εκπαίδευση*. Σε αυτό το στάδιο, ο αλγόριθμος δέχεται ως είσοδο το *σύνολο δεδομένων εκπαίδευσης* (training set) με σκοπό να κατασκευάσει και να αποκτήσει τη νέα γνώση και κατ' επέκταση να επιτύχει το σκοπό του. Εν προκειμένω, ο αλγόριθμος έχει τη δυνατότητα να αξιοποιήσει λιγότερο ή περισσότερο την προγενέστερη γνώση, καθώς δύναται να μην την αξιοποιήσει καθόλου.



Εικόνα 10: Φάσεις Μηχανικής Μάθησης

Εν συνεχεία, το βήμα που έπεται την εκπαίδευση είναι αυτό της *πιστοποίησης* της νέας γνώσης που παράγεται κατά τη διαδικασία. Επίσης, ο ίδιος αλγόριθμος πραγματώνει το βήμα της *πιστοποίησης* μέσα

από τις διαδικασίες ανάκλησης (recall), οι οποίες υποστηρίζονται από τα δεδομένα ελέγχου (test data). Επιπλέον υποστήριξη παρέχεται από τον ίδιο το χρήστη και τις γνώσεις αυτού για το πρόβλημα που προσπαθεί να επιλύσει ο αλγόριθμος. Η γνώση που θα προκύψει από αυτή τη διαδικασία προσφέρεται γενικότερα σε εφαρμογές που θα μπορούν να την αξιοποιήσουν, ώστε να επέλθουν λύσεις πραγματικών προβλημάτων.

3.3. Κατηγοριοποίηση

Η *Κατηγοριοποίηση* αποτελεί το μετέπειτα στάδιο της *Μηχανικής Μάθησης*, όπου η δεύτερη σχετίζεται με τους τρόπους που οι αλγόριθμοι επιτρέπουν στα ΥΣ να μαθαίνουν. Η Κατηγοριοποίηση ονομάζεται αλλιώς και *Ταξινόμηση* (Classification), όπου ολοκληρώνονται τα στάδια της εκπαίδευσης και της κατηγοριοποίησης των δεδομένων, όπως φαίνεται παρακάτω:



Εικόνα 11: Στάδια εξόρυξης γνώσης

Η *Ταξινόμηση* (Στατιστική Ταξινόμηση - ΣΤ) αφορά τη διαδικασία όπου η τοποθέτηση ανεξάρτητων αντικειμένων ανά ομάδες εξαρτάται από την ποσοτική πληροφορία ενός ή πολλαπλών χαρακτηριστικών των ανεξάρτητων αντικειμένων. Τα αντικείμενα αυτά λογίζονται ως μεταβλητές ή ιδιότητες, που βασίζονται σε ένα σύνολο δεδομένων που

αφορά την εκπαίδευση αυτών ανάλογα με την ομάδα στην οποία έχουν κατηγοριοποιηθεί. Το σημαντικό είναι ότι από το σύνολο των δεδομένων, παράγεται ένας ταξινομητής, ο οποίος θα κατηγοριοποιεί ένα ανεξάρτητο αντικείμενο στην πραγματική ετικέτα ταξινόμησης. Η ετικέτα ταξινόμησης με τη σειρά της χαρακτηρίζεται από μια άγνωστη αντιστοίχιση.

Γενικότερα, η ΣΤ αποτελεί μια από τις μεθόδους *Επιβλεπόμενης Μάθησης*. Εκκινώντας από ένα σύνολο παραδειγμάτων, που είναι γνωστοποιημένο σε ποια ομάδα ανήκουν, το ΥΣ εκπαιδεύεται με τέτοιο τρόπο, ώστε να δύναται να ταξινομεί άγνωστα αντικείμενα στις ήδη υπάρχουσες παραγόμενες ομάδες. Επιπλέον, το σύνολο των παραδειγμάτων μπορεί να αποτελεί θετικό ή αρνητικό στιγμιότυπο μιας έννοιας.

Επιπρόσθετα, η αξιοπιστία ενός συστήματος ελέγχεται, εφόσον διατηρηθεί ένα μέρος του training set, το οποίο δεν έχει συμμετοχή στη διαδικασία της εκπαίδευσης. Δηλαδή, όταν η εκπαίδευση πραγματοποιηθεί εισάγεται το training set και εν συνεχεία αξιολογούνται τα αποτελέσματα (test set). Η διαδικασία είναι έλλογη, καθώς ένα ΥΣ έχει εκπαιδευτεί να ταξινομεί αποδοτικά ένα σύνολο δεδομένων. Στόχος είναι να μην υπερ-εκπαιδευτεί μόνο σε ένα σύνολο, καθώς θα αδυνατεί να λειτουργεί αποδοτικά σε άγνωστα σύνολα δεδομένων. Συνεπώς, η διατήρηση ενός μέρους του training set εξυπηρετεί να αξιολογηθούν τα αποτελέσματα της **Ταξινόμησης** είτε έχουν προκύψει από γνωστά είτε από άγνωστα δεδομένα.

3.3.1. Τεχνικές Ταξινόμησης

Τα δείγματα/δεδομένα επιδέχονται διάφορες μεθόδους Ταξινόμησης.

Συνοπτικά αναφέρεται ότι:

- i. Γραμμικοί Ταξινομητές*
 - Logistic regression
 - Naive Bayes classifier
 - Αισθητήρας (Perceptron)
 - Μηχανές Διανυσμάτων Υποστήριξης (Support vector machines)
 - Fisher's linear discriminant
- ii. Quadratic classifiers*
- iii. Κ-πλησιέστερων γειτόνων (k-nearest neighbor)*
- iv. Ενίσχυση (Boosting)*
- v. Δέντρα Απόφασης- ΔΑ (Decision trees)*
 - Random forests
- vi. Νευρωνικά Δίκτυα (Neural networks)*
- vii. Bayesian networks*
- viii. Hidden Markov models*

Τα νευρωνικά δίκτυα αποτελούν μια από τις πιο χρηστικές μεθόδους της ΜΜ, ενώ παράλληλα χρησιμοποιούνται καθολικά στην κατηγοριοποίηση των μαστογραφιών. Εκκινούν από την ιδέα των Τεχνητών Νευρωνικών Δικτύων (ΤΝΔ) ή Νευρωνικών Δικτύων (ΝΔ), τα οποία σχετίζονται με τους τρόπους που λειτουργεί ο ανθρώπινος εγκέφαλος βάσει των βιολογικών νευρωνικών δικτύων. Αποτελεί κοινή παραδοχή ότι ο ανθρώπινος εγκέφαλος έχει διαφορετικές λειτουργίες από αυτές των Ηλεκτρονικών Υπολογιστών ως προς την πραγμάτωση υπολογισμών. Αυτό οδήγησε την έρευνα στην προσπάθεια να αναπαραστήσει τις μεθόδους μάθησης, αλλά και λειτουργίας των Ηλεκτρονικών Υπολογιστών.

Γενικά, τα ΤΝΔ/ΝΔ σχετίζονται με την προσπάθεια που γίνεται από μια υπολογιστική μηχανή να προσεγγίσει τις λειτουργικές διαδικασίες του ανθρώπινου εγκεφάλου. Οι υπολογισμοί, που πραγματώνονται, γίνονται με καθολικό παράλληλο τρόπο και η δομή τους χαρακτηρίζεται από τη δομή των Βιολογικών Νευρωνικών Δικτύων. Πιο ειδικά, αποτελούν μια συλλογή από νευρώνες (Processing Units - PUs), οι οποίοι αλληλοσυνδέονται. Κάθε PU αποτελείται από διάφορες εισόδους και μια έξοδο. Η έξοδος, ωστόσο, μπορεί να είναι η είσοδος για άλλες PUs, ενώ οι συνδέσεις που υφίστανται μεταξύ τους ενέχουν διαφορές σχετικές με τη σημαντικότητά τους. Αυτή η σημαντικότητα ορίζεται από τον συντελεστή βάρους, που ονομάζεται σύναψη.

Επιπλέον, η επεξεργασία του εκάστοτε PU εξαρτάται από την συνάρτηση μεταφοράς, καθώς με αυτήν καθορίζεται η έξοδος συγκριτικά με άλλες εισόδους και με τους συντελεστές βάρους. Η χρήση ενός ΤΝΔ εξαρτάται άμεσα από την εκπαίδευσή του στο να «μαθαίνει». Αυτό σημαίνει πως πρέπει η μάθησή του να περιλαμβάνει τον προσδιορισμό καθορισμένων συντελεστών βάρους, ούτως ώστε το ΤΝΔ να παράγει προσδοκώμενα αποτελέσματα. Παράλληλα, αυτή η διαδικασία πραγματώνεται από αλγορίθμους μάθησης ή εκπαίδευσης, ενώ οι συντελεστές βάρους συντελούν στην ερμηνεία και κατανόηση της αποθηκευμένης γνώσης. Η γνώση έχει αντληθεί και εξαχθεί από το σύνολο των παραδειγμάτων. Όλα τα παραπάνω περιγράφουν τον τρόπο που τα ΝΔ εκπαιδεύονται για μάθουν το περιβάλλον, το οποίο αποτελείται από τα δεδομένα που κατασκευάζουν το φυσικό αυτό μοντέλο.

3.3.2. Μηχανές Διανυσμάτων Υποστήριξης (SVM)

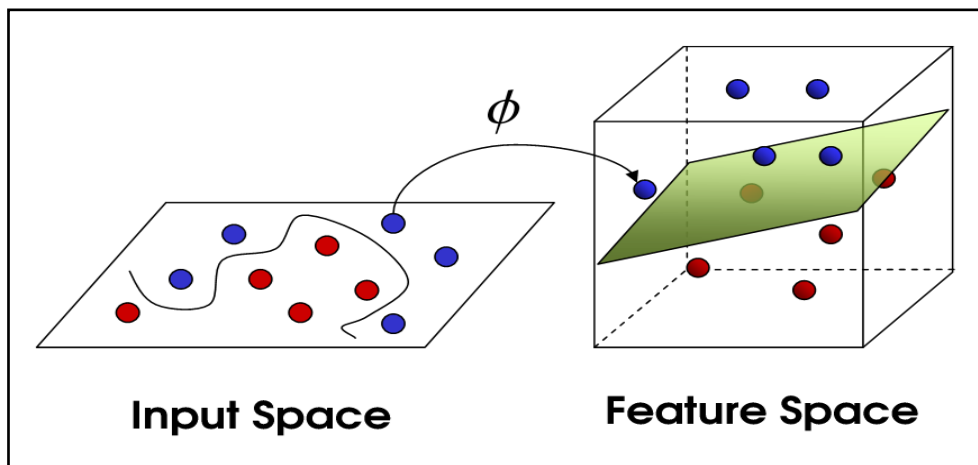
Μια από τις σύγχρονες προσεγγίσεις ΜΜ είναι η μέθοδος Support Vector Machines (SVM), η οποία μπορεί να γενικεύει και κατ' επέκταση χρησιμοποιείται σε σύνθετα δεδομένα ερευνών. Τα SVM μπορούν να χαρακτηριστούν ως μια μέθοδος Ταξινόμησης, και επιδέχονται έναν πίνακα έκφρασης γονιδίων ως είσοδο. Ο πίνακας έκφρασης γονιδίων συμβολίζεται ως $n*m$, όπου n είναι ο αριθμός των γονιδίων και m ο αριθμός των πραγματοποιμένων πειραμάτων.

Τα SVM στοχεύουν στο διαχωρισμό των γονιδίων σε δυο κατηγορίες στον εκάστοτε m -διάστατο χώρο, στον οποίο η πρώτη κατηγορία θα αναπαριστά τα μέλη μιας κλάσης – δηλαδή, θετικά παραδείγματα- ενώ η δεύτερη κατηγορία θα αναπαριστά τα μη μέλη – δηλαδή, αρνητικά παραδείγματα. Δεν είναι ιδιαίτερα εύκολο να πραγματοποιηθεί το παραπάνω στο m -διάστατο χώρο, και για αυτό το λόγο είναι αναγκαίο να βρεθεί ένας χώρος με μεγαλύτερες διαστάσεις. Αυτός ο χώρος θα ονομαστεί feature space και θα χρησιμοποιηθεί για να επιτευχθεί ο διαχωρισμός. Αυτή η διαδικασία με τη σειρά της δημιουργεί υπερ-επίπεδα, τα οποία είναι πολλαπλά και διαχωρίζουν τα δεδομένα. Ωστόσο, είναι σημαντικό να επιλεχθεί το καλύτερο/ βέλτιστο υπερ-επίπεδο (Suykens et al., 1999).

Ειδικότερα, αυτή η μέθοδος κάνει την υπόθεση ότι τα πρότυπα έχουν ταξινομηθεί βάσει μιας ήδη γνωστής πληροφορίας ταξινόμησης. Επομένως, τα πρότυπα πηγαίνουν από τον πρώτο χώρο μικρότερων διαστάσεων σε αυτόν με μεγαλύτερες διαστάσεις, όπου προσπαθούν να εντοπίσουν σε αυτόν το βέλτιστο υπερ-επίπεδο. Η επιλογή του βασίζεται με τέτοιο τρόπο, ώστε να μπορεί να μεγιστοποιηθεί η απόσταση ανάμεσα

στα κοντινότερα πρότυπα των δυο διαφορετικών κατηγοριών. Αυτό εξηγείται παραστατικά στην Εικόνα 12, όπου τα δεδομένα είναι στον 2-διαστατο χώρο, ενώ ο βέλτιστος διαχωρισμός υλοποιείται στον 3-διαστατο χώρο.

Το γεγονός ότι ο διαχωρισμός πραγματώνεται στον 3-διαστατο χώρο συμβαίνει, καθώς οι μεταβλητές (ή ιδιότητες) των δεδομένων χαρακτηρίζονται από σύνθετες μη γραμμικές σχέσεις. Σύμφωνα με αυτό, ο μη-γραμμικός διαχωρισμός οδηγεί συχνά στην τροποποίηση ενός μη γραμμικά διαχωρίσιμου χώρου καταστάσεων σε έναν πιο απλό γραμμικά διαχωρίσιμο χώρο χαρακτηριστικών υψηλών διαστάσεων (high dimensional feature space). Έτσι, όποιος αλγόριθμος και αν έχει χρησιμοποιηθεί, έχει τη δυνατότητα να παράγει μια συνάρτηση πυρήνα, που ονομάζεται αλλιώς kernel trick. Αυτό μπορεί να συμβεί, εφόσον μπορεί να εκφραστεί με τέτοιο τρόπο ούτως ώστε το εκάστοτε διάνυσμα δεδομένων εισόδου (input data) να μπορεί να εμφανιστεί αποκλειστικά σε πράξεις εσωτερικού γινομένου (dot product). Έπειτα από αυτό, κατασκευάζεται η μη γραμμική εκδοχή του χρησιμοποιούμενου αλγορίθμου με αντικατάσταση του εσωτερικού γινομένου από τη συνάρτηση πυρήνα.



Εικόνα 12: από τις 2 στις 3 διαστάσεις

3.4. Προβλήματα Ομαδοποίησης

3.4.1. Εισαγωγή

Η Ομαδοποίηση ονομάζεται και Συσταδοποίηση (Clustering) και αφορά τη διαδικασία όπου διακρίνεται το σύνολο από κάποια «αντικείμενα». Ιδιαίτερα ενδιαφέρον στοιχείο αποτελούν οι μαρτυρίες ότι η διαδικασία της ομαδοποίησης εμφανίζεται και στην Αρχαία Ελλάδα. Έχει αναφερθεί πως τον 4ο αιώνα π.Χ. ο Αριστοτέλης και ο Θεόφραστος ομαδοποιούσαν τα φυτά ανάλογα με τα μορφολογικά και τα δομικά τους γνωρίσματα. Αυτά τα αντικείμενα διαχωρίζονται σε υποσύνολα, τα οποία αποτελούνται από κάποια στοιχεία που είναι υποχρεωτικό να ομοιάζουν σε σχέση με άλλα υποσύνολα, όπως φαίνεται στην Εικόνα 13. Αυτή η ομαδοποίηση των αντικειμένων εντός μιας ομάδας ερμηνεύεται ως η ομοιότητα αυτών και αντίστροφα. Δηλαδή, αφορά και αντικείμενα διαφορετικών ομάδων, που είναι ανόμοιες μεταξύ τους.

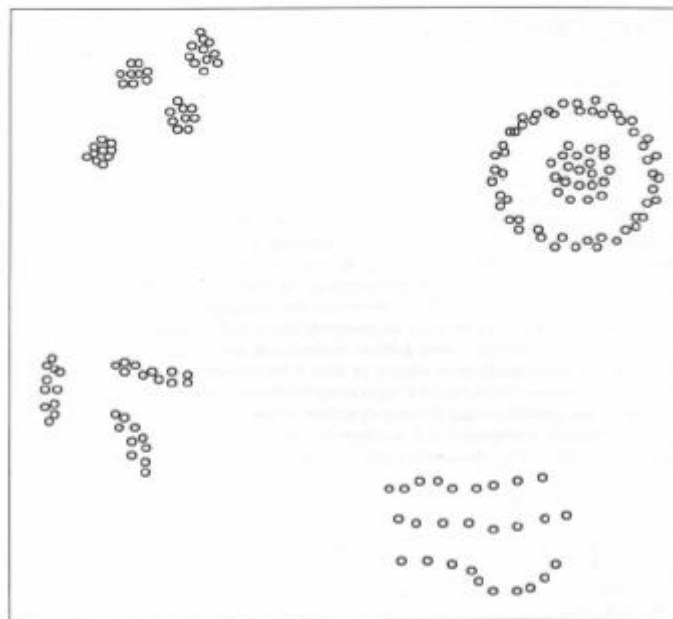
Σημαντικό είναι ότι η προαναφερθείσα ομοιότητα των αντικειμένων πηγάζει από ένα ορισμένο πρόβλημα, αλλά και από τη μορφή αυτών των αντικειμένων. Γενικότερα, η ομοιότητα έχει αναφερθεί και ως:

- Ανάλυση Ομάδων (Cluster Analysis)
- Αυτόματη Κατηγοριοποίηση (Automatic Classification)
- Αριθμητική Ταξινόμηση (Numerical Taxonomy)
- Τυπολογική Ανάλυση (Typological Analysis)

Η διαδικασία της ομαδοποίησης χρησιμοποιείται με απώτερο στόχο τον εντοπισμό υποσυνόλων όμοιων αντικειμένων σε ένα σύνολο δεδομένων. Τα υποσύνολα θεωρούνται ομάδες (Clusters). Παρότι έχουν προταθεί πολλές ερμηνείες, δεν έχει οριστεί επακριβώς η έννοια της ομάδας (Wilks, 2011). Πληροφοριακά, το Tryon (1939) είχε συγγράψει την πρώτη συνοπτική έρευνα για την ομαδοποίηση. Σε αυτήν αναφέρεται πως η διαδικασία της ομαδοποίησης αποτελείται από έναν αριθμό διαφορετικών αλγόριθμων οι οποίοι ταξινομούν με στόχο να δομήσουν την παραγόμενη πληροφορία μέσα σε αντίστοιχες ομάδες. Επιπλέον, ο Everitt (1974) αναφέρει πως μια ομάδα είναι μια περιοχή ενός χώρου που θα χαρακτηρίζεται από μια υψηλή πυκνότητα σημείων. Παράλληλα, αυτή η περιοχή θα έχει χωρίσματα λόγω της παρουσίας άλλων περιοχών με σχετικά μεγάλη πυκνότητα από σημεία ή με περιοχές χαμηλής πυκνότητας σημείων.

Επιπρόσθετα, οι Kendall et al. (1950) περιγράφουν την ομάδα ως το σύνολο συνεχόμενων στοιχείων, που προκύπτει από έναν στατιστικό πληθυσμό. Τα παραδείγματα, που δίνονται, αφορούν ένα σύνολο ανθρώπων που συγκατοικούν, μια συνεχή ροή παρατηρήσεων σε μια ιεραρχημένη σειρά,

καθώς και ένα σύνολο γειτονικών μερών εντός ενός πεδίου. Επίσης, οι Wallace et al. όρισαν την ομάδα ως το υποσύνολο οντοτήτων. Αυτές οι οντότητες συνήθως χρησιμοποιούνται ως ισοδύναμες. Τέλος, ο Gergrelli κάνει λόγο για ένα σύνολο σημείων εντός ενός χώρου ελέγχου. Με αυτόν τον τρόπο, η απόσταση ανάμεσα σε δύο τυχαία σημεία της ομάδας μπορεί να είναι μικρότερη σε σχέση με την απόσταση ανάμεσα σε ένα τυχαίο σημείο της ομάδας και σε ένα σημείο που βρίσκεται εκτός αυτής.



Εικόνα 13: Παραδείγματα ομάδων στο επίπεδο

3.4.2. Μαθηματικός Ορισμός

Αν θεωρηθεί ένα σύνολο διανυσμάτων $X = \{X_1, X_2, X_3, \dots, X_n\}$, ζητούνται m σύνολα-ομάδες $C_1, C_2, \dots, C_m$ με $m \ll n$, ούτως ώστε: $C_i \neq \emptyset \forall i=1,2,3,\dots,m$, όπου m οι ομάδες που αποτελούν διαμέριση του συνόλου X .

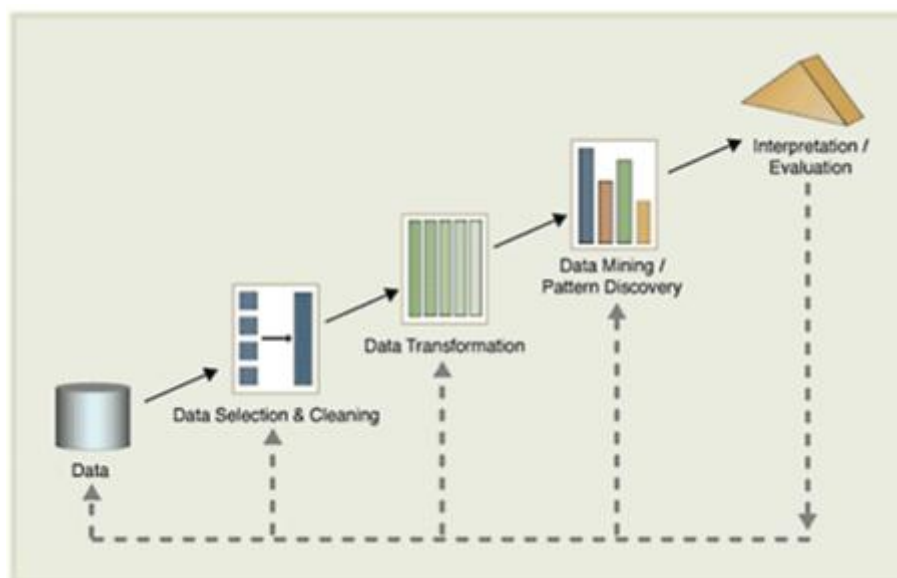
Ο παραπάνω ορισμός σχετίζεται με την αυστηρή συσταδοποίηση, καθώς κάθε διάνυσμα από το σύνολο αντιστοιχεί σε μια μόνο ομάδα. Ωστόσο, η ασαφής συσταδοποίηση μπορεί να ερμηνευτεί με εναλλακτικό τρόπο.

Δηλαδή, πιο ειδικά, αν χρησιμοποιηθούν *ασαφή σύνολα*, οι m συναρτήσεις συμμετοχής ορίζονται ως $U_j: X \rightarrow [0,1]$ για $j = 1,2, \dots, m$. Οι συναρτήσεις $U(\cdot)$ ποσοτικοποιούν την βεβαιότητα που μπορεί να υπάρξει σχετικά με το αν κάποιο διάνυσμα i αντιστοιχεί στην ομάδα j .

3.5. Διαδικασία Ομαδοποίησης

Η διαδικασία της Ομαδοποίησης ενσωματώνει τέσσερα βήματα, όπως δείχνει γραφικά η Εικόνα 14. Συνοπτικά, τα βήματα είναι:

- ✓ η επιλογή των χαρακτηριστικών
- ✓ η επιλογή ενός αλγόριθμου ομαδοποίησης
- ✓ η αξιοπιστία
- ✓ ο έλεγχος των παραγόμενων αποτελεσμάτων



Εικόνα 14: Διαδικασία Ομαδοποίησης

Το πρώτο στάδιο είναι η *επιλογή των χαρακτηριστικών*. Σε αυτό το στάδιο, πρέπει να επιλεχθεί ένα σύνολο δεδομένων, αλλά και τα αντικείμενα που θα ταξινομηθούν σε ομάδες και θα αποτελούν το δείγμα. Μετά από αυτό το βήμα, είναι αναγκαίο να επιλεχθούν προσεκτικά τα χαρακτηριστικά που θα λάβουν μέρος στη διαδικασία της ομαδοποίησης. Πιο ειδικά, τα χαρακτηριστικά είναι αυτά που δύνανται να κωδικοποιήσουν με τον πιο αποδοτικό τρόπο την πληροφορία που σχετίζεται με το πρόβλημα επίλυσης. Το σύνολο δεδομένων που έχει επιλεχθεί θα αναπαρίσταται ως n -διάστατα διανύσματα χαρακτηριστικών, ενώ συνήθως έχει υποστεί προ-επεξεργασία.

Το δεύτερο βήμα είναι η *επιλογή αλγορίθμου ομαδοποίησης*. Αυτό το στάδιο περιλαμβάνει την επιλογή ενός αλγοριθμικού σχήματος, που χρησιμεύει για τον εντοπισμό των ομάδων που υπάρχουν στο σύνολο δεδομένων. Βάσει αυτού, προκύπτει η *επιλογή του μέτρου εγγύτητας* που είναι το μέτρο που δείχνει, κάθε φορά, ποσοτικά την ομοιότητα - ή τη μη ομοιότητα- δυο συγκεκριμένων οντοτήτων από τα δεδομένα. Επιπλέον, προκύπτει η *επιλογή του κριτηρίου ομαδοποίησης*, η οποία βασίζεται σε πολλούς παράγοντες. Τέτοιοι παράγοντες μπορούν να είναι ο τύπος των δεδομένων (συνεχή, διακριτά ή δυαδικά) και η δομή των ομάδων εντός των δεδομένων (συμπαγείς ή επιμήκεις ομάδες κοκ).

Το τρίτο βήμα είναι η *αξιοπιστία*. Η αξιοπιστία είναι το βήμα που πρέπει να ελεγχθεί η ποιότητα και η ορθότητα των ομάδων που εν τέλει προέκυψαν. Υπάρχουν διάφορες τεχνικές αξιοπιστίας (validation techniques), που χρησιμοποιούνται για να διακριθούν οι αληθινές από τις μη-αληθινές δομές. Οι τεχνικές αξιοπιστίας περικλείονται από τρεις

μεθόδους ελέγχου πιστότητας, που είναι τα **Εξωτερικά Κριτήρια**, τα **Εσωτερικά Κριτήρια** και τα **Σχετικά Κριτήρια**.

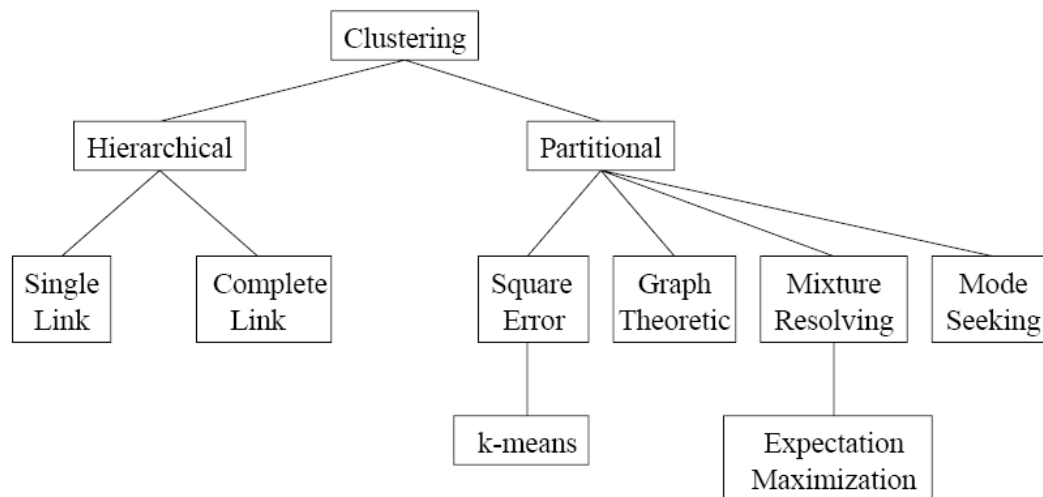
Τα **Εξωτερικά Κριτήρια** (External Criteria) βασίζονται σε τεστ σημαντικότητας (significant tests) βάσει εξωτερικών παραγόντων. Από την άλλη, τα **Εσωτερικά Κριτήρια** (Internal Criteria) αφορούν τεστ σημαντικότητας βάσει εσωτερικών μεταβλητών. Δηλαδή, μεταβλητών των συγκεκριμένων χαρακτηριστικών των οντοτήτων που συλλέχθηκαν. Η τρίτη μέθοδος αφορά τα **Σχετικά Κριτήρια** (Relative Criteria), όπου η ομαδοποίηση τίθεται σε σύγκριση με αποτελέσματα από άλλα υπάρχοντα σχήματα ομαδοποίησης. Τα άλλα σχήματα έχουν προέλθει από την επιτέλεση λειτουργίας του ίδιου αλγορίθμου με διαφορετικές παραμέτρους από τις τρέχουσες.

Το τέταρτο και τελευταίο βήμα της διαδικασίας ομαδοποίησης αφορά την **ερμηνεία των αποτελεσμάτων**. Αυτό το βήμα σχετίζεται με την ερμηνεία και ανάλυση των αποτελεσμάτων της ομαδοποίησης. Σε αυτό περιλαμβάνονται τα συμπεράσματα της παραγόμενης διαδικασίας. Όπως είναι εύλογο, η ανάλυση αυτή πραγματοποιείται από τους αντίστοιχους επιστήμονες του κλάδου που εφάρμοσαν τη διαδικασία ομαδοποίησης.

3.6. Αλγόριθμοι ομαδοποίησης

Η Εικόνα 15 αναπαριστά διαγραμματικά τον τρόπο που μπορούν να ταξινομηθούν οι ποικίλες τεχνικές ομαδοποίησης. Έτσι, παρατηρείται ότι διακρίνονται σε πρώτο στάδιο οι μέθοδοι ομαδοποίησης σε **Ιεραρχικές** (Hierarchical) και **Διαμεριστικές** (Partitional). Είναι εμφανές ότι οι Ιεραρχικές μέθοδοι ενσωματώνουν μια αλληλουχία διαμερίσεων που

βρίσκεται η μία μέσα στην άλλη. Αυτή η αλληλουχία μπορεί να χαρακτηριστεί ως δένδροδιάγραμμα. Από την άλλη, οι διαμεριστικές μέθοδοι ενσωματώνουν μόνο μια διαμέριση.



Εικόνα 15: Κατηγορίες Ομαδοποίησης

- **Συγκεντρωτικοί και Διαχωριστικοί (Agglomerative vs Divisive)**

Αυτές οι κατηγορίες παρουσιάζουν μια βασική διαφοροποίηση σχετικά με τη λειτουργία και τις βασικές δομές ενός αλγορίθμου. Οι Συγκεντρωτικοί αφορούν τους αλγόριθμους που εκκινούν τη λειτουργία τους με την ταξινόμηση του κάθε προτύπου σε μια διακριτή ομάδα. Παράλληλα, αρχίζει να συμπυκνώνει σταθερά και διαδοχικά τις ομάδες μέχρι να ικανοποιηθεί ένα κριτήριο που έχει τεθεί. Οι Διαχωριστικοί αφορούν τις μεθόδους όπου ο αλγόριθμος ταξινομεί όλα τα πρότυπα σε μια ομάδα και εν συνεχεία αυτή η ομάδα διαχωρίζεται ξανά σε υπο- ομάδες για να καταλήξει στην ικανοποίηση ενός κριτηρίου τερματισμού (Gowda, 1978).

- **Μονοθετικοί και Πολυθετικοί (Monothetic vs Polythetic)**

Αυτές οι κατηγορίες ταξινομούν τους αλγορίθμους ανάλογα με το αν χρησιμοποιούν τα χαρακτηριστικά των προτύπων το ένα μετά το άλλο ή τα χρησιμοποιούν ταυτόχρονα κατά τη διάρκεια της διαδικασίας της

ομαδοποίησης. Αποτελεί γεγονός ότι η πλειοψηφία των αλγορίθμων υπάγεται στους Πολυθετικούς. Αυτό σημαίνει ότι τα χαρακτηριστικά του προτύπου χρησιμοποιούνται παράλληλα για τον υπολογισμό των αποστάσεων ανάμεσα στα πρότυπα. Από την άλλη, ένας Μονοθετικός αλγόριθμος χρησιμοποιεί ένα χαρακτηριστικό κάθε φορά και οι ομαδοποιήσεις, που πράττει, βασίζονται σε αυτό το χαρακτηριστικό. Όταν επαναλαμβάνεται η διαδικασία, επιλέγεται άλλο χαρακτηριστικό, ενώ παράλληλα διαχωρίζονται ξανά οι τρέχουσες ομάδες.¹

Στην Εικόνα 16 τα δεδομένα ταξινομούνται σε δύο ομάδες βάσει του χαρακτηριστικού x_1 . Αυτή η ταξινόμηση που αποτελεί ένα διαχωρισμό συμβολίζεται με την κάθετη γραμμή V. Έπειτα, η κάθε ομάδα διαχωρίζεται βάσει του χαρακτηριστικού x_2 , όπως προκύπτει από τις γραμμές H1 και H2. Ωστόσο, παρουσιάζεται ένα αρνητικό στοιχείο σε τέτοιους αλγορίθμους, που είναι η παραγωγή 2d ομάδων, όπου d ορίζεται η διάσταση των προτύπων ή αλλιώς το πλήθος των χαρακτηριστικών των προτύπων. Επίσης, για μεγάλες τιμές του d ($d > 100$), ο αριθμός των παραγόμενων ομάδων είναι τόσο μεγάλος, ώστε το σύνολο των δεδομένων να διαχωρίζεται σε πολύ μικρές και με ελάχιστη σημασία ομάδες.

▪ *Αυστηροί και Ασαφείς (Hard vs Fuzzy)*

Η Αυστηρή ομαδοποίηση επιτρέπει σε έναν αλγόριθμο να αποδίδει κάθε πρότυπο σε μια μόνο ομάδα. Αυτή η διαδικασία πραγματώνεται κατά την επιτέλεση της επεξεργασίας των δεδομένων, αλλά και στην τελική ομαδοποίηση. Από την άλλη, η Ασαφής ομαδοποίηση επιτρέπει στον

¹ Βλ. Εικόνα 16.

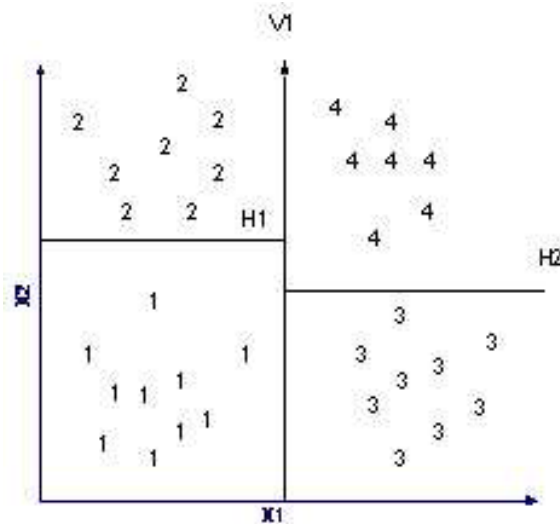
αλγόριθμο να αποδίδει ένα βαθμό σε κάθε πρότυπο για κάθε μια ομάδα. Ο βαθμός αυτός δείχνει το ποσοστό που αυτό το πρότυπο ταιριάζει στην ομάδα αυτή. Επιπρόσθετα, ο αλγόριθμος της Ασαφούς μεθόδου μπορεί να γίνει Αυστηρός με το να αναθέτει κάθε πρότυπο στην ομάδα που έχει τον μεγαλύτερο βαθμό.

- **Ντετερμινιστικοί και Στοχαστικοί (Deterministic vs Stochastic)**

Το χαρακτηριστικό αυτών των αλγορίθμων είναι ότι είναι διαιρετικοί. Επίσης, αφορούν την βελτιστοποίηση μιας τετραγωνικής συνάρτησης, που πραγματώνεται είτε με παραδοσιακές τεχνικές βελτιστοποίησης είτε με μια τυχαία αναζήτηση στο χώρο. Με αυτόν τον τρόπο, προκύπτει η βέλτιστη διαμέριση των προτύπων εντός των ομάδων προορισμών τους.

- **Αυξητικοί και Μη Αυξητικοί (Incremental vs Non Incremental)**

Αυτές οι μέθοδοι παρουσιάζονται στις περιπτώσεις που το σύνολο των προτύπων, που πρέπει να ομαδοποιηθεί, είναι ιδιαίτερα μεγάλο. Παράλληλα, εμφανίζονται και κάποιοι περιορισμοί, οι οποίοι σχετίζονται με τον χρόνο εκτέλεσης του αλγορίθμου και με τον χώρο της μνήμης. Αυτοί οι περιορισμοί έχουν τη δυνατότητα να ασκούν επίδραση στην αρχιτεκτονική του αλγορίθμου. Αν τα δεδομένα, στα πρώτα στάδια ομαδοποίησης, δεν είναι πολλά, τότε οι πρώτοι προς χρήση αλγόριθμοι δεν δύνανται να επεξεργαστούν μεγάλα σύνολα προτύπων. Όσο η διαδικασία εξελίσσεται, το εύρος των δεδομένων μεγαλώνει και κατ' επέκταση δημιουργείται η ανάγκη να σχεδιαστούν κατάλληλοι αλγόριθμοι που θα μειώσουν στο ελάχιστο τον αριθμό των σαρώσεων του συνόλου των προτύπων.



Εικόνα 16:Ομαδοποίηση Μονοθετικού αλγόριθμου

4. Εξελικτικοί Υπολογισμοί

4.1. Εισαγωγή

Η περιγραφή των Εξελικτικών Υπολογισμών επιβάλλει την αναφορά στη θεωρία της Εξέλιξης των Ειδών (Evolution of Species) του Δαρβίνου. Λόγω αυτής της θεωρίας προέκυψαν αρκετές αντιδράσεις, προερχόμενες κυρίως από διάφορες τρέχουσες θρησκευτικές πεποιθήσεις που έκαναν λόγο για την δική τους οπτική ως προς τη δημιουργία της ανθρώπινης ύπαρξης και εξέλιξης. Παρόλα αυτά, η θεωρία της Εξέλιξης των Ειδών έχει αντιμετωπιστεί θετικά από την πλειοψηφία της επιστημονικής κοινότητας, καθώς έχει προσφέρει ένα σύνολο καθολικών απαντήσεων σε κομβικά ερωτήματα που ταλάντευαν την ανθρωπότητα. Ο βασικός στόχος αυτής της θεωρίας είναι να προσφέρει μια βασική ερμηνεία για το φαινόμενο της ζωής, τη δημιουργία της, αλλά και τις κομβικές ιδιότητές της.

Οι *Εξελικτικοί Υπολογισμοί* (Evolutionary Computations) εκκινούν από την θεωρία της Εξέλιξης των Ειδών και κατά συνέπεια μπορούν να

θεωρηθούν το αποτέλεσμα της προσομοίωσής τους με αυτήν. Πιο συγκεκριμένα, οι ΕΥ εφαρμόζουν υπολογιστικά μοντέλα διαδικασιών εξέλιξης, τα οποία λειτουργούν ως κεντρικά στοιχεία για τον σχεδιασμό και την πραγματοποίηση των υπολογιστικών συστημάτων για επίλυση προβλημάτων. Επιπρόσθετα, αποτελούν μια μίμηση των καταστάσεων βιολογικής εξέλιξης, υπό την έννοια ότι έχουν τη δυνατότητα να πραγματώνουν την ιδέα της φυσικής επιλογής, καθώς και την ιδέα της επικράτησης του ισχυρότερου. Με αυτό τον τρόπο, δύνανται να προσφέρουν τις καλύτερες δυνατές απαντήσεις στα προβλήματα βελτιστοποίησης και αναζήτησης (Bäck et al., 1997).

Αναλυτικότερα, οι ΕΥ αποτελούνται από έναν πληθυσμό ατόμων (population of individuals). Έχουν τη δυνατότητα να τον αναπτύσσουν δια μέσου κάποιων κανόνων επιλογής (selection rules) και ορισμένων τελεστών (operators). Τέτοιοι τελεστές θεωρούνται ο ανασυνδυασμός (recombination) και η μετάλλαξη (mutation). Το κάθε άτομο του εκάστοτε πληθυσμού αποτελεί μια αναπαράσταση ενός σημείου του χώρου, που σχετίζεται με τις πιθανές λύσεις ενός προβλήματος προς επίλυση. Σημαντικό είναι ότι περιλαμβάνονται σε αυτούς και κάποιες πληροφορίες σχετικές με τους κανόνες του περιβάλλοντος εντός ενός προβλήματος. Επίσης, υπάρχει ένα μέτρο αξιολόγησης (fitness), το οποίο προσκολλά σε κάθε ένα άτομο του πληθυσμού και παρέχεται κάθε φορά σε ένα συγκεκριμένο περιβάλλον του προβλήματος. Το περιβάλλον έχει κωδικοποιηθεί από κάποια συνάρτηση αξιολόγησης (fitness function). Παράλληλα, η επιλογή εξαρτάται από τα άτομα που έχουν δημιουργήσει την πιο κατάλληλη συνάρτηση αξιολόγησης, ενώ ο ανασυνδυασμός και η μετάλλαξη επιδρούν

στη δομή των ατόμων. Αυτό συνεπάγεται ότι δημιουργούνται κάποιες δυνατότητες ώστε να διερευνηθεί ο χώρος.

Πρέπει, επίσης, να επισημανθεί ότι ο αρχικός πληθυσμός του Εξελικτικού Υπολογισμού ταξινομείται σε τυχαίες τιμές και αναπτύσσεται με διαδοχικό τρόπο σε προτιμότερες περιοχές του χώρου αναζήτησης. Αυτή η εξέλιξη σημειώνεται δια μέσου των διαδικασιών της επιλογής, που είναι ο ανασυνδυασμός και η μετάλλαξη. Ταυτόχρονα, το περιβάλλον θα προσφέρει τις πληροφορίες που σχετίζονται με την ποιότητα των νέων σημείων αναζήτησης, ενώ η διαδικασία επιλογής θα προωθήσει τα άτομα με τη βέλτιστη ποιότητα ώστε να αυξηθεί η αναπαραγωγή τους συγκριτικά με άλλα άτομα. Η διαδικασία του ανασυνδυασμού αναμειγνύει την πληροφορία που διαδίδεται από τους προγόνους στους απογόνους, ενώ η διαδικασία της μετάλλαξης παρουσιάζει καινούρια στοιχεία, που θα αποτελέσουν καινοτομίες στον πληθυσμό.

Οι ΕΑ περιγράφουν όλων των ειδών τους αλγορίθμους, οι οποίοι εφαρμόζονται και λειτουργούν βάσει των διαδικασιών φυσικής εξέλιξης. Οι αλγόριθμοι αυτοί δεν είναι ίδιοι μεταξύ τους, καθότι αποτελούνται από διαφορετικές λεπτομέρειες ως προς τα στοιχεία της λειτουργίας τους, όπως για παράδειγμα, οι χρησιμοποιούμενοι τελεστές εξέλιξης, η αναπαράσταση του πληθυσμού κλπ. Παρακάτω, διαφαίνεται η δομή ενός εξελικτικού αλγορίθμου βάσει της κατασκευής μιας ψευδογλώσσας, ενώ στην Εικόνα 17 παρουσιάζεται το Διάγραμμα Ροής του Τυπικού Εξελικτικού Αλγορίθμου.

Εξελικτικός Αλγόριθμος

$G := 0;$

Αρχικοποίησε και αξιολόγησε $[P(G)]$;

ΟΣΟ η κατάσταση δεν αλλάζει **KANE**

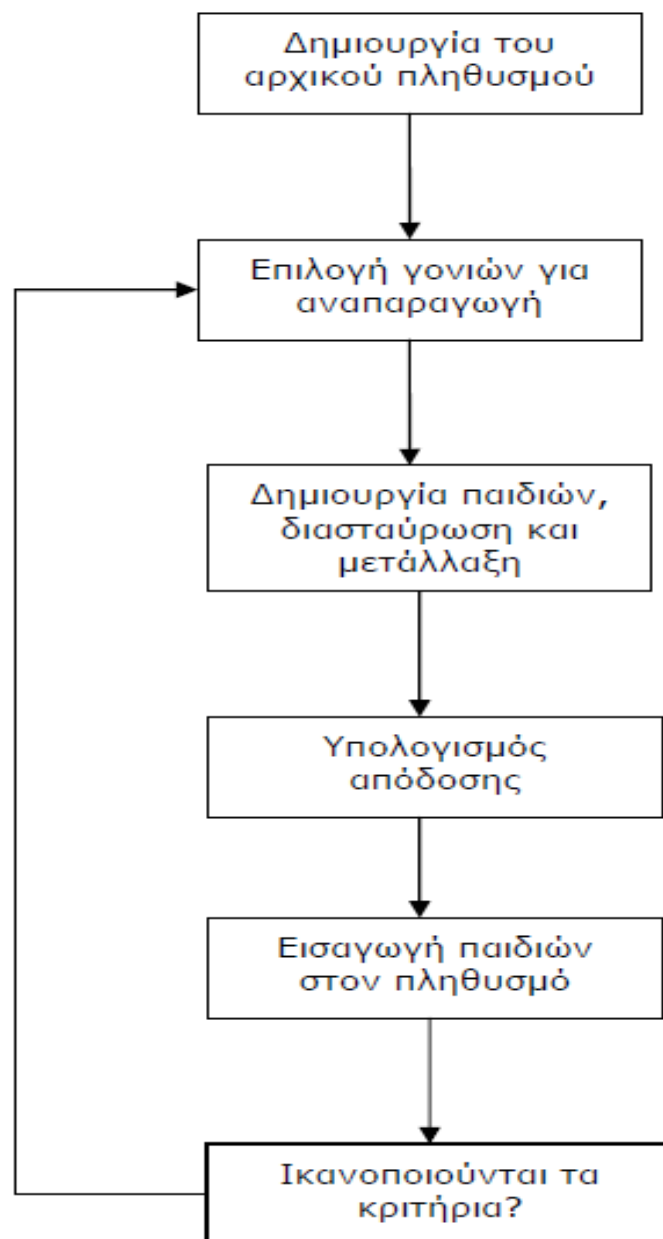
$P'(G) := \text{παραλλαγή } [P(G)]$

Αξιολόγησε $[P'(G)]$;

$P(G + 1) := \text{επέλεξε } [(P'(G), P(G))]$

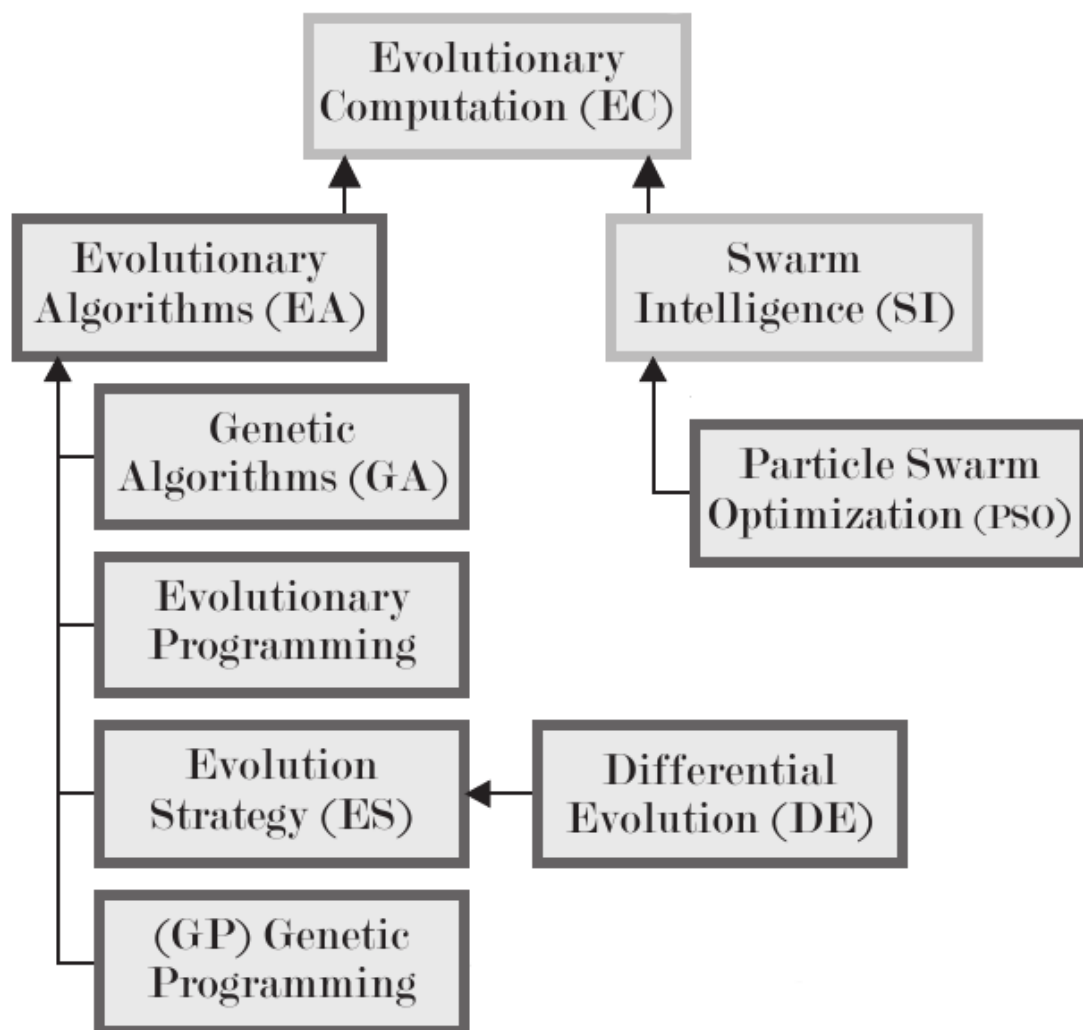
$G := G + 1;$

ΤΕΛΟΣ ΟΣΟ



Εικόνα 17: Διάγραμμα ροής Τυπικού Εξελικτικού Αλγόριθμου

Γενικά, οι *ΕΑ* αποτελούν μια κατηγορία του Εξελικτικού Υπολογισμού, όπως είναι και ο Διαφοροεξελικτικός Αλγόριθμος. Η δομή η οποία διαχωρίζει τον Εξελικτικό Υπολογισμό, καθώς και οι πιο βασικοί Εξελικτικοί Αλγόριθμοι, παρουσιάζονται στην Εικόνα 18. Οι βασικοί Εξελικτικοί Αλγόριθμοι είναι: οι *Γενετικοί Αλγόριθμοι* - ΓΑ (Genetic Algorithm - GA), οι *Εξελικτικές Στρατηγικές* - ΕΣ (Evolutionary Strategies - ES), ο *Εξελικτικός Προγραμματισμός* - ΕΠ (Evolutionary Programming - EP), ο *Γενετικός Προγραμματισμός* - ΓΠ (Genetic Programming - GP), οι *Διαφοροεξελικτικοί Αλγόριθμοι* - ΔΕ (Differential Evolution - DE) και οι *Αλγόριθμοι Νοημοσύνης Σμηνών* - ΑΝΣ (Particle Swarm Intelligence - PSO).



Εικόνα 18: Κατηγορίες Εξελικτικού Υπολογισμού

4.2. Εξελικτικοί αλγόριθμοι και ομαδοποίηση

Πολύ συχνά για να αντιμετωπιστεί το πρόβλημα της ομαδοποίησης χρησιμοποιείται ο τρόπος που η φύση διαχειρίζεται το πρόβλημα της εξέλιξης των ειδών. Η διαδικασία σχετίζεται με τη μίμηση και αντιγραφή της φύσης μέσω της χρήσης κάποιων τελεστών εξέλιξης και ενός πληθυσμού από ομαδοποιήσεις. Με αυτόν τον τρόπο, πραγματοποιείται η αναζήτηση της συνολικά βέλτιστης ομαδοποίησης για το επιλεγμένο σύνολο δεδομένων. Οι εναλλακτικές λύσεις του προβλήματος ομαδοποίησης θεωρούνται και κωδικοποιούνται ως χρωμοσώματα.

Από την άλλη, οι τελεστές χρήσης, που είναι οι πιο συχνοί, είναι η επιλογή (selection), ο ανασυνδυασμός (recombination) και η μετάλλαξη (mutation). Ο κάθε τελεστής σχεδιάζει ξανά ένα ή πιο πολλά χρωμοσώματα εισόδου σε ένα ή περισσότερα χρωμοσώματα εξόδου. Σχετικά με το κάθε χρωμόσωμα υπολογίζεται η τιμή μιας συνάρτησης που ονομάζεται fitness function. Αυτή η τιμή επιδεικνύει τις πιθανότητες, όπου ένα χρωμόσωμα μπορεί να επιβιώσει σε επόμενη γενεά.

Ο Εξελικτικός Αλγόριθμος Ομαδοποίησης περιλαμβάνει τρία βήματα διαδικασιών. Αρχικά, το πρώτο βήμα αφορά την τυχαία επιλογή ενός πληθυσμού από κάποιες λύσεις/ομαδοποιήσεις. Εν συνεχεία του ίδιου βήματος, περιγράφεται η ανάθεση μιας τιμής που θα αναπαριστά τη φυσική κατάσταση για τη συνάρτηση Fitness Function σε κάθε λύση ή ομαδοποίηση. Η ανάθεση της τιμής μπορεί να είναι αντιστρόφως ανάλογη της τιμής του τετραγωνικού σφάλματος (Maulik, 2000). Αυτό σημαίνει ότι μια λύση/ομαδοποίηση με μικρό τετραγωνικό σφάλμα, θα

έχει μεγάλη τιμή για την Fitness Function. Το δεύτερο βήμα αφορά τη χρήση των εξελικτικών τελεστών, δηλαδή της επιλογής, του ανασυνδυασμού και της μετάλλαξης με σκοπό την παραγωγή μιας νέας γενιάς λύσεων/ομαδοποιήσεων που θα πηγάζει από την προγενέστερη γενιά. Το τρίτο βήμα αφορά την επανάληψη του δεύτερου βήματος έως ότου να πραγματοποιηθεί το κριτήριο τερματισμού.

Επιπρόσθετα, αξίζει να τονιστεί ότι στη διαδικασία ομαδοποίησης προτιμάται η χρήση των Γενετικών Αλγορίθμων. Οι ΓΑ κωδικοποιούν τις πιθανές λύσεις σαν δυαδικά strings. Παράλληλα, υπάρχει ο τελεστής επιλογής, ο οποίος διαδίδει τις λύσεις από τη μια γενιά στην άλλη. Αυτό πραγματοποιείται βάσει της τιμής που έχει η συνάρτηση για την επιβίωσή τους. Ο τελεστής επιλογής θα χρησιμοποιήσει ένα πιθανοτικό μοντέλο με σκοπό να αναπαραχθούν οι λύσεις/ομαδοποιήσεις με τη μεγαλύτερη τιμή «επιβίωσης». Επιπλέον, αυτοί οι αλγόριθμοι έχουν τη δυνατότητα να αναπαριστούν τα σημεία του χώρου λύσεων σαν δυαδικά strings και να πραγματοποιούν διερεύνηση του χώρου με τον τελεστή crossover.

Από την άλλη, ο τελεστής μετάλλαξης τίθεται σε χρήση όταν πρέπει να διασφαλιστεί η πλήρης εξερεύνηση του χώρου. Επιπλέον, οι ΕΣ και ο ΕΠ έχουν διαφορές από τους Γενετικούς Αλγορίθμους ως προς το πώς αναπαριστούν τις λύσεις/ομαδοποιήσεις, αλλά και ως προς τον τελεστή μετάλλαξης που χρησιμοποιούν. Πιο ειδικά, ο ΕΠ κάνει χρήση τελεστών μετάλλαξης και επιλογής, ενώ αφήνει απέξω τη χρήση τελεστών ανασυνδυασμού. Και οι τρεις προσεγγίσεις στοχεύουν στην επίλυση προβλημάτων ομαδοποίησης μέσω της επιδίωξης να ελαχιστοποιηθεί το κριτήριο τετραγωνικού σφάλματος.

Επιπρόσθετα, οι ΓΑ είναι υπεύθυνοι για το ολικό βέλτιστο της λύσης/ομαδοποίησης. Αντίθετα, άλλες μέθοδοι ομαδοποίησης εστιάζουν στο τοπικό βέλτιστο. Για τις δεύτερες περιπτώσεις, η λύση πηγάζει από συνεχή επανάληψη του αλγορίθμου και το τοπικό βέλτιστο θα υπάρχει στην περιοχή της υπάρχουσας λύσης.

Σύμφωνα με τα παραπάνω, προκύπτει ότι ο αλγόριθμος K-means και οι ασαφείς αλγόριθμοι είναι οι αλγόριθμοι που χρησιμεύουν για τη διαδικασία ομαδοποίησης και αναζητούν το τοπικό βέλτιστο. Εν αντιθέσει, οι ΓΑ, οι τελεστές crossover και μετάλλαξης αναζητούν και προσφέρουν καινούριες λύσεις/ομαδοποιήσεις που δε σχετίζονται με τις προγενέστερες. Παρόλα αυτά, υπάρχει η δυνατότητα να κριθεί το πρόβλημα της ομαδοποίησης ως ένα πρόβλημα βελτιστοποίησης. Σε τέτοιες περιπτώσεις, αναζητούνται τα βέλτιστα κέντρα των ομάδων και όχι η βέλτιστη διαμέριση μέσω της χρήσης Γενετικών Αλγορίθμων.

Σύμφωνα με το τελευταίο, προκύπτει η χρήση των ΕΣ και ΕΠ, καθώς αυτές οι μέθοδοι δύνανται να αναπαριστούν, με εύληπτο τρόπο, τα κέντρα των ομάδων σε διανύσματα πραγματικών αριθμών. Επίσης, τα ΕΣ μπορούν να χρησιμοποιηθούν σε προβλήματα αυστηρής, αλλά και ασαφούς ομαδοποίησης. Από την άλλη, ο ΕΠ μπορεί να χρησιμοποιηθεί για προβλήματα ασαφών min-max ομάδων. Αποτελεί γεγονός ότι με αυτόν τον τρόπο προσφέρονται προτιμότερα αποτελέσματα από αυτά που προκύπτουν από παρόμοιους αλγόριθμους χρήσης, όπως K-means και FCM. Τέλος, οι προαναφερόμενες προσεγγίσεις παρουσιάζουν ένα πρόβλημα στην ορθή επιλογή των παραμέτρων, καθώς το κάθε πρόβλημα επιβάλλει έναν διαφορετικό προσδιορισμό παραμέτρων.

Βιβλιογραφία

- Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). *The traveling salesman problem: a computational study*. Princeton university press.
- Babu, G. P., & Murty, M. N. (1994). Clustering with evolution strategies. *Pattern Recogn*, 27, 321329.
- Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997). *Handbook of evolutionary computation*. CRC Press.
- Barron, A. R., Cohen, A., Dahmen, W., & DeVore, R. A. (2008). Approximation and learning by greedy algorithms. *The annals of statistics*, 36(1), 64-94.
- Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3), 209-219.
- Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York NY.
- Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168). ACM.
- D'ariano, A., Pacciarelli, D., & Pranzo, M. (2007). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2), 643-657.
- Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*, 12(3), 1231-1237.

- Dunn, J. C. (1973). A fuzzy Relative of the ISODATA Process and its use in detecting compact well separated clusters. *Journal of Cybernetics*, 3, 32-57.
- Fogel, D. B., & Simpson, P. K. (1993). Evolving fuzzy clusters. In *Proceedings of the International Conference on Neural Networks* (San Francisco, CA), 18291834.
- Friedland, B. (2012). *Control system design: an introduction to state-space methods*. Courier Corporation.
- Funke, J. (2010). Complex problem solving: A case for complex cognition?. *Cognitive Processing*, 11(2), 133-142.
- Gil-García, R. J., Badía-Contelles, J. M., & Pons-Porrata, A. (2003, November). Extended star clustering algorithm. In *Iberoamerican Congress on Pattern Recognition* (pp. 480-487). Springer, Berlin, Heidelberg.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading, MA.
- Gottlieb, J., Marchiori, E., & Rossi, C. (2002). Evolutionary algorithms for the satisfiability problem. *Evolutionary Computation*, 10(1), 35-50.
- Gowda, K. C., & Krishna, G. (1978). Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2), 105-112.
- Gutin, G., & Punnen, A. P. (Eds.). (2006). *The traveling salesman problem and its variations* (Vol. 12). Springer Science & Business Media.
- Jain, A. K., Murty, M. N., & Flynn P. J. (1999). Data Clustering: A Review.

ACM Computing Surveys, vol. 31, No. 3.

Jonassen, D. H., Howland, J., Moore, J., &Marra, R. M. (2003). Learning to solve problems with technology: A constructivist perspective.

Leung, J. Y. (Ed.). (2004). *Handbook of scheduling: algorithms, models, and performance analysis*. CRC Press.

Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern recognition*, 33(9), 1455-1465.

Mézard, M., Mora, T., &Zecchina, R. (2005). Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 94(19), 197205.

Michalewicz, Z., & Fogel, D. B. (2004). Traditional Methods—Part 1. In *How to Solve It: Modern Heuristics* (pp. 55-81). Springer, Berlin, Heidelberg.

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4), 049901.

Ponsen, M., Munoz-Avila, H., Spronck, P., & Aha, D. W. (2006). Automatically generating game tactics through evolutionary learning. *AI Magazine*, 27(3), 75.

Ruspini, E. H. (1969). A new approach to clustering. *Inf. Control*, 15, 22-32.

Saxe, R., Brett, M., & Kanwisher, N. (2006). Divide and conquer: a defense of functional localizers. *Neuroimage*, 30(4), 1088-1096.

Sebastiani, F. (2002). Machine learning in automated text categorization.

ACM computing surveys (CSUR), 34(1), 1-47.

Storn, R. & Price, K. (1997). Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces.

Journal of Global Optimization, vol. 11, 341–359.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3), 293-300.

Wilks, D. S. (2011). Cluster analysis. In *International geophysics* (Vol. 100, pp. 603-616). Academic press.