

ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ  
ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ Τ.Τ

ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:

«Μελέτη , αποτίμηση και υλοποίηση ευρετικών και μεθευρευτικών αλγορίθμων , για την βελτιστοποίηση του χρονισμού των φωτεινών σηματοδοτών αστικών περιοχών»

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: κος Δρόσος Χρήστος

ΣΠΟΥΔΑΣΤΕΣ : ΚΑΜΠΑΣ ΛΑΜΠΡΟΣ Α.Μ: 44627

ΚΑΡΙΚΑΣ ΝΙΚΟΛΑΟΣ Α.Μ:44637

ΑΙΓΑΛΕΩ

ΝΟΕΜΒΡΙΟΣ 2017

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / η ..... Καρίμας Νικόλαος .....  
του Ιωάννη ..... με αριθμό μητρώου 44637 ..... φοιτητής / τριά του  
Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την  
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του  
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και  
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται  
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη  
αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα  
του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος  
φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα  
του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η  
Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του  
αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα  
καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός  
ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα  
προβλεπόμενα στο άρθρο 13, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ο Δηλών  


Ημερομηνία 28/11/2017

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / η Καλνός Λάκης,  
του Χρυσόφορου, με αριθμό μητρώου 44692 φοιτητής / τριά του  
Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την  
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του  
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και  
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται  
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη  
αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα  
του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος  
φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα  
του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η  
Συνέλευση του Τμήματος με νέα απόφασής της, μετά από αίτηση του ενδιαφερόμενου, του  
αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα  
καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός  
ημερολογιακού έτους από την ημερομηνία ανάθεσής της. Κατά τα λοιπά εφαρμόζονται τα  
προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ο Δηλών  
Καλνός Λάκης

Ημερομηνία

28/11/2017

## Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1 – Η ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ .....	6
1.1. Εισαγωγή.....	6
1.2 Βασικές έννοιες γύρω από την Τεχνητή Νοημοσύνη.....	8
1.2.1. Ορισμός της Νοημοσύνης .....	8
1.2.2. Ορισμός της Τεχνητής Νοημοσύνης .....	10
1.2.3. Διαχωρισμός της τεχνητής από τη φυσική Νοημοσύνη .....	12
1.2.4. Υπολογιστική Νοημοσύνη .....	13
1.3. Η Εξέλιξη της Τεχνητής Νοημοσύνης .....	18
1.4. Η Τεχνητή Νοημοσύνη σήμερα .....	20
ΚΕΦΑΛΑΙΟ 2 – ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ.....	23
2.1. Εισαγωγή.....	23
2.2. Περιγραφή Προβλημάτων.....	24
2.2.1. Περιγραφή με Χώρο Καταστάσεων .....	25
2.2.2. Περιγραφή με Αναγωγή .....	30
2.3. Αλγόριθμοι αναζήτησης προβλημάτων.....	31
2.3.1. Αναζήτηση Πρώτα σε Βάθος (Depth First Search - DFS) .....	38
2.3.2. Αναζήτηση Πρώτα σε Πλάτος ( <i>Breadth First Search - BFS</i> ).....	39
2.3.3. Αναζήτηση Διπλής Κατεύθυνσης ( <i>Bidirectional Search - BiS</i> ) .....	41
ΚΕΦΑΛΑΙΟ 3 – ΕΥΡΕΤΙΚΟΙ ΚΑΙ ΜΕΤΑΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ.....	46
3.1. Ευρετικοί αλγόριθμοι .....	46
3.2. Μεταευρετικοί αλγόριθμοι.....	48
3.3. Παραδείγματα για τους ευρετικούς αλγορίθμους.....	49
3.3.1. Αλγόριθμοι κατασκευής .....	49
3.3.2. Lagrangian Relaxation.....	49
3.3.3. Γενετικοί Αλγόριθμοι.....	50
3.3.4. Αλγόριθμος Tabu Search.....	51
3.3.5. Αλγόριθμος Simulated Annealing .....	51
3.3.6. Νοημοσύνη των σμηνών .....	52
3.4. Παραδείγματα μεταευρετικών αλγορίθμων .....	53
3.4.1. Βελτιστοποίηση με Σμήνος Σωματιδίων .....	53

3.4.2. Διαφοροεξελικτικός Αλγόριθμος .....	56
ΚΕΦΑΛΑΙΟ 4 – ΜΕΘΟΔΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΦΩΤΕΙΝΩΝ ΣΗΜΑΤΟΔΟΤΩΝ .....	58
4.1. Συστήματα ελέγχου κυκλοφορίας .....	58
4.2. Έλεγχος σταθερού χρόνου.....	61
4.3. Δυναμικός έλεγχος .....	62
4.4. Συντονισμένος έλεγχος.....	64
ΚΕΦΑΛΑΙΟ 5 – ΧΡΗΣΗ ΕΥΡΕΤΙΚΩΝ ΚΑΙ ΜΕΤΑΕΥΡΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΤΗΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΩΝ ΦΩΤΕΙΝΩΝ ΣΗΜΑΤΟΔΟΤΩΝ .....	67
5.1. Εισαγωγή.....	67
5.2. Υπόβαθρο .....	68
5.3. Εξέταση της χρήσης ευρετικών αλγορίθμων.....	70
5.4. Η αναλογική ευρετική μέθοδος.....	73
5.5. Πιθανά οφέλη από μια μεθευρετική προσέγγιση .....	74
5.6. Εφαρμογή της μεθευρετικής προσέγγισης .....	75
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	78
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	79

## ΚΕΦΑΛΑΙΟ 1 – Η ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

### 1.1. Εισαγωγή

Η Τεχνητή Νοημοσύνη - TN (Artificial Intelligence -AI) αν και συμπλήρωσε μισό αιώνα ζωής εξακολουθεί να είναι μια από τις πιο μοντέρνες ερευνητικές περιοχές. Τυπικά ξεκίνησε το 1956 στη συνάντηση μερικών επιφανών επιστημόνων, όπως ο John McCarty, Marvin Minsky, Claude Shannon και άλλοι, αν και η έρευνα είχε ξεκινήσει 5 χρόνια πριν. Από την άλλη, η μελέτη της νοημοσύνης είναι ένα από τα πιο παλιά θέματα της ανθρώπινης αναζήτησης. Για περισσότερο από 2000 χρόνια, οι φιλόσοφοι προσπάθησαν να ερμηνεύσουν το μηχανισμό της όρασης, της μάθησης, της απομνημόνευσης, της αντίληψης και του συλλογισμού, όπως ο Αριστοτέλης, ο Ηράκλειτος, ο Descartes κλπ.

Η TN περικλείει ένα πλήθος ερευνητικών πεδίων, από γενικού σκοπού όπως η αντίληψη και η συλλογιστική έως πιο συγκεκριμένων, όπως το παίξιμο σκακιού, η απόδειξη θεωρημάτων, η διάγνωση ασθενειών, κλπ. Συχνά ερευνητές από άλλες επιστημονικές περιοχές καταφεύγουν στην TN με σκοπό να βρουν εργαλεία για να αυτοματοποιήσουν τα λογικά βήματα που χρησιμοποιούν στην εργασία τους. Όμοια, ερευνητές της TN εφαρμόζουν τις μεθόδους τους σε διάφορες περιοχές όπου απαιτείται ανθρώπινη ευφυή προσπάθεια.

Η TN χωρίζεται σε δύο κατηγορίες, τη Συμβολική και τη Μη-Συμβολική TN. Η *Συμβολική Τεχνητή Νοημοσύνη (symbolic AI-Artificial Intelligence)* προσομοιώνει τον τρόπο σκέψης του ανθρώπου, χρησιμοποιώντας ως δομικές μονάδες τα σύμβολα. Ένα σύμβολο μπορεί να αναπαριστά μία έννοια ή μία σχέση ανάμεσα σε έννοιες. Παραδείγματα αυτής της κατηγορίας είναι οι εφαρμογές της TN που χρησιμοποιούν αναπαράσταση γνώσης με λογική, κανόνες, πλαίσια, κλπ. Η *Μη Συμβολική Τεχνητή Νοημοσύνη (non symbolic AI- Artificial Intelligence)* προσομοιώνει βιολογικές διεργασίες, όπως τη διαδικασία εξέλιξης των ειδών ή τη λειτουργία του εγκεφάλου.

Παραδείγματα τέτοιων τεχνικών αποτελούν τα νευρωνικά δίκτυα και οι γενετικοί αλγόριθμοι [1].

## 1.2 Βασικές έννοιες γύρω από την Τεχνητή Νοημοσύνη

Ένας από τους πρώτους ορισμούς που διατυπώθηκαν από τους Barr και Feigenbaum αναφέρει ότι, *"TN είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφρών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με τη νοημοσύνη στην ανθρώπινη συμπεριφορά"* [1].

Αυτό που προκύπτει από τον παραπάνω ορισμό αλλά και από τους άλλους που θα παρατεθούν στη συνέχεια, είναι ότι για να ορισθεί τι μπορεί να κάνει ένα "νοήμον" υπολογιστικό σύστημα πρέπει πρώτα να ορισθεί ο όρος "νοημοσύνη".

### 1.2.1. Ορισμός της Νοημοσύνης

Στο ερμηνευτικό λεξικό αναφέρεται ότι: *"Νοημοσύνη είναι η ποιότητα, άσκηση ή το παράγωγο της ενεργής διανόησης, η ικανότητα να ασκεί κάποιος τις υψηλότερες διανοητικές λειτουργίες, η ετοιμότητα στην κατανόηση"*, ενώ *"Διανόηση είναι η δύναμη ή η αντίληψη ή η σκέψη, αυτό που κάνει τους ανθρώπους να ξεχωρίζουν από τα ζώα, οι υψηλότερες διανοητικές λειτουργίες όπως αυτές διαχωρίζονται από τις αισθήσεις και τη μνήμη, η νοημοσύνη* [1].

Εύκολα κάποιος εμπλέκεται σε έναν ατέρμονα βρόχο αλληπάληλων και αμοιβαία αναδρομικών ορισμών, με αποτέλεσμα ο ορισμός της "νοημοσύνης" να μην είναι απόλυτα δυνατός.

Η έλλειψη, όμως, ορισμού δε σημαίνει ότι κάποιος δεν αναγνωρίζει τη νοημοσύνη όταν τη "δει". Κι αυτό γιατί η νοημοσύνη έχει κάποια χαρακτηριστικά και μία συγκεκριμένη δομή. Έτσι θα μπορούσαμε να πούμε ότι νοημοσύνη είναι:

- Η δυνατότητα λόγου, δηλαδή η δυνατότητα αντίληψης, κατανόησης και παραγωγής του.
- Η δυνατότητα συλλογισμών.
- Η δυνατότητα μάθησης.



Βέβαια, η νοημοσύνη υποβοηθείται από την ύπαρξη των αισθήσεων και της μνήμης, αλλά σε καμία περίπτωση αυτά μόνα τους δεν αρκούν. Τα επόμενα χαρακτηριστικά προτείνονται από τον Douglas Hofstadter σε μια λίστα από θεμελιώδεις δυνατότητες νοημοσύνης στο βιβλίο του που τιμήθηκε με το βραβείο Pulitzer το 1979: "Gödel, Escher, Bach: An Eternal Golden Braid " :

- Να ανταποκρίνεσαι σε καταστάσεις με ελαστικότητα, δηλαδή να μην αντιδράς με τον ίδιο τρόπο σε ταυτόσημα προβλήματα (μηχανική συμπεριφορά).
- Να αντιλαμβάνεσαι (κατανοείς) τα ασαφή ή αντιφατικά μηνύματα από τα συμφραζόμενα.
- Να αναγνωρίζεις και να ιεραρχείς τα διάφορα δεδομένα με βάση τη σπουδαιότητα τους.
- Να βρίσκεις ομοιότητες μεταξύ καταστάσεων οι οποίες εκ πρώτης όψεως μοιάζουν διαφορετικές.
- Να βρίσκεις διαφορές μεταξύ καταστάσεων οι οποίες εκ πρώτης όψεως μοιάζουν παρόμοιες.

Οι ικανότητες αυτές έχουν τουλάχιστον ένα κοινό χαρακτηριστικό: αποκτώνται εύκολα από τους ανθρώπους και συνήθως τις αποκαλούμε κοινή λογική (common sense). Η επιστήμη που ερευνά τους μηχανισμούς της ανθρώπινης ευφυΐας αναφέρεται ως γνωστική ή γνωσιολογική επιστήμη (cognitive science).

Έχει αποδειχθεί ότι όσο πιο απλή και αυτονόητη για τους ανθρώπους είναι μια λειτουργία τόσο πιο δύσκολα μπορεί να μεταφερθεί σε έναν υπολογιστή, δηλαδή να περιγραφεί με ένα πρόγραμμα. Και αυτό γιατί συνήθως οι απλές λειτουργίες εκτελούνται μηχανικά χωρίς ιδιαίτερη σκέψη και συνεπώς είναι πολύ δύσκολο να περιγράψουμε πώς τις πραγματοποιήσαμε. Τέτοιες λειτουργίες είναι για παράδειγμα η διαδικασία ανάγνωσης χαρακτήρων, ο σχηματισμός λέξεων και η κατανόηση του νοήματος κειμένων ή ομιλιών καθώς και η διαδικασία της μετακίνησης από το σπίτι στο γραφείο περπατώντας, η επιλογή της διαδρομής, η αναγνώριση γνωστών στο δρόμο, κλπ.

Αντίθετα αν μια δουλειά είναι δύσκολη και απαιτεί σχεδιασμό των βημάτων που πρέπει να εκτελεστούν για να πετύχουμε το επιθυμητό αποτέλεσμα, τότε δεν θα ήταν τόσο δύσκολο να περιγράψουμε αυτά τα βήματα με κάποιο πρόγραμμα, όπως για

παράδειγμα η επίλυση ενός μαθηματικού προβλήματος ή η επεξεργασία μιας φορολογικής δήλωσης, κλπ.

### 1.2.2. Ορισμός της Τεχνητής Νοημοσύνης

Η Τεχνητή Νοημοσύνη είναι το γενικό όνομα το οποίο δόθηκε στο πεδίο της επιστήμης των υπολογιστών αφιερωμένο στην ανάπτυξη των προγραμμάτων τα οποία τείνουν να αντιγράψουν πιστά την ανθρώπινη νοημοσύνη [10]. Κατά καιρούς έχουν διατυπωθεί διάφοροι ορισμοί της Τεχνητής Νοημοσύνης (TN), από τους οποίους άλλοι επικεντρώνονται στη διαδικασία σκέψης και συλλογισμού και άλλοι στη συμπεριφορά. Έτσι υπάρχουν ορισμοί σύμφωνα με τους οποίους στόχος της TN είναι να φτιάξει συστήματα που:

- Σκέφτονται όπως οι άνθρωποι :

Η προσπάθεια να κατασκευάσουμε υπολογιστές με διανοητική ικανότητα με την πλήρη και κυριολεκτική έννοια του όρου (Haugeland).

- Συμπεριφέρονται όπως οι άνθρωποι :

Η μελέτη του πώς να κάνουμε τους υπολογιστές να κάνουν πράγματα στα οποία αυτήν τη στιγμή οι άνθρωποι είναι καλύτεροι (Rich και Knight).

- Σκέφτονται λογικά :

Η μελέτη των υπολογισμών που καθιστούν εφικτή την αντίληψη, τη λογική σκέψη και την αντίδραση (Winston).

- Αντιδρούν λογικά :

Ο τομέας της επιστήμης των υπολογιστών που ασχολείται με την αυτοματοποίηση της ευφυούς συμπεριφοράς (Luger και Stubblefield).

Ένας γενικός ορισμός που περιλαμβάνει τα περισσότερα στοιχεία από τους ανωτέρω θα μπορούσε να είναι ο εξής :

*“Τεχνητή Νοημοσύνη είναι ο τομέας της Επιστήμης των Υπολογιστών που ασχολείται με τη σχεδίαση και την υλοποίηση προγραμμάτων τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες, εμφανίζοντας έτσι χαρακτηριστικά που αποδίδουμε*

*συνήθως σε ανθρώπινη συμπεριφορά,, όπως για παράδειγμα η επίλυση προβλημάτων, η αντίληψη μέσω της όρασης, η μάθηση, η εξαγωγή συμπερασμάτων, η κατανόηση φυσικής γλώσσας, κλπ.”*

Άμεση συνέπεια των παραπάνω ορισμών είναι η αποδοχή ότι η TN είναι ένας συνεχώς εξελισσόμενος τομέας την Επιστήμης των Υπολογιστών που προσπαθεί να κάνει πραγματικότητα, ότι αυτή η επιστήμη δεν έχει καταφέρει μέχρι στιγμής. Κι αυτό είναι μία μεγάλη αλήθεια, αν σκεφτεί κανείς ότι η τεχνολογία του σήμερα δεν έχει προσφέρει στην υπηρεσία του ανθρώπου, τίποτε άλλο εκτός από μηχανές οι οποίες αποθηκεύουν τεράστιες ποσότητες πληροφορίας και έχουν την ικανότητα να τις προσπελάσουν σε ελάχιστο χρόνο. Ωστόσο, οι μηχανές αυτές απέχουν ακόμη από το να μην απαιτούν ειδικές γνώσεις για το χειρισμό τους, να προσαρμόζονται στις ανάγκες του χρήστη, να μαθαίνουν από τα λάθη τους και να επιλύουν πραγματικά, δύσκολα, καθημερινά προβλήματα και όχι μόνον αριθμητικά.

Οι μεθοδολογίες που αναπτύχθηκαν για την TN έχουν αποδώσει καρπούς σε πολλές από τις επιμέρους περιοχές αυτής της επιστήμης, όπως:

- Απόδειξη θεωρημάτων
- Επεξεργασία φυσικής γλώσσας
- Τεχνητή όραση
- Μηχανική μάθηση
- Σχεδιασμός ενεργειών και χρονοπρογραμματισμός
- Αυτόνομα robot
- Έμπειρα συστήματα και συστήματα γνώσης

Υπάρχουν δύο προσεγγίσεις για την TN. Η *κλασική ή συμβολική* προσέγγιση, που βασίζεται στην κατανόηση των νοητικών διεργασιών και ασχολείται με τη προσομοίωση της ανθρώπινης νοημοσύνης προσεγγίζοντας την με αλγορίθμους και συστήματα που βασίζονται στη γνώση και η *συνδετική* (connectionist approach) ή *μη-συμβολική* προσέγγιση που βασίζεται στη μίμηση της βιολογικής λειτουργίας του εγκεφάλου προσεγγίζοντας το θέμα με τα λεγόμενα νευρομορφικά ή νευρωνικά δίκτυα.

### 1.2.3. Διαχωρισμός της τεχνητής από τη φυσική Νοημοσύνη

Η εφαρμογή της Τεχνητής Νοημοσύνης σε μηχανές καθιστούν τις μηχανές αυτές ευφυείς και τις κάνουν να συμπεριφέρονται σχεδόν όπως ο άνθρωπος. Ο διαχωρισμός των μηχανών αυτών ως ευφυείς ή όχι καθορίζεται με τη βοήθεια του κριτηρίου που έθεσε ο Turing (δοκιμασία Turing).

Ο Alan Turing (1912-1953), ο οποίος θεωρείται ο πατέρας της ΤΝ, εμπνεύστηκε το 1950 μία δοκιμασία η οποία πήρε και το όνομα του (Turing Test - Δοκιμασία Turing), για το χαρακτηρισμό των μηχανών. Αυτό βασίζεται σε μία σειρά από ερωτήσεις που υποβάλει κάποιος ταυτόχρονα σε έναν άνθρωπο και μία μηχανή χωρίς να γνωρίζει εκ των προτέρων ποιος είναι τι. Αν στο τέλος δεν καταφέρει να ξεχωρίσει τον άνθρωπο από τη μηχανή, τότε η μηχανή πετυχαίνει στη δοκιμασία και θεωρείται ευφυής [1]. Αν και η αποτελεσματικότητα της Δοκιμασίας Turing εξαρτάται από πολλούς παράγοντες, θεωρείται μέχρι σήμερα ένα καλό μέτρο σύγκρισης της φυσικής με την τεχνητή νοημοσύνη και πολλοί διαγωνισμοί διοργανώνονται σε ετήσια βάση, χωρίς όμως ιδιαίτερα σοβαρά ή τουλάχιστο χρήσιμα αποτελέσματα.

Ο προγραμματισμός ενός υπολογιστή για να περάσει τη δοκιμασία Turing απαιτεί τη συμμετοχή αρκετών επιστημονικών περιοχών, όπως της επεξεργασίας φυσικής γλώσσας (natural language processing) για επικοινωνία σε φυσική γλώσσα, της αναπαράστασης γνώσης (knowledge representation) για την αποθήκευση της γνώσης πριν και κατά τη διάρκεια της δοκιμασίας, της αυτοματοποιημένης συλλογιστικής (automated reasoning) για τη χρήση της αποθηκευμένης πληροφορίας και την εξαγωγή συμπερασμάτων, της μηχανικής μάθησης (machine learning) για προσαρμογή σε νέες περιπτώσεις, κλπ.

Στην αρχική της μορφή, η δοκιμασία Turing δεν προέβλεπε φυσική επαφή ανθρώπου μηχανής. Ωστόσο μια επέκταση της (Πλήρης Δοκιμασία Turing) περιλαμβάνει και την αναγνώριση εικόνων και αντικειμένων που ανταλλάσσονται μέσα από κάποια θυρίδα ώστε να μην υπάρχει οπτική επαφή με το δοκιμαζόμενο για τον έλεγχο των δυνατοτήτων αντίληψης του. Για το σκοπό αυτό απαιτείται η συμμετοχή και άλλων δύο επιστημονικών περιοχών, της μηχανικής όρασης (machine vision) για την αναγνώριση και της ρομποτικής (robotics) για τη μετακίνηση τους.

#### 1.2.4. Υπολογιστική Νοημοσύνη

Η Τεχνητή Νοημοσύνη έχει ως κύριο στόχο της να γίνει ο υπολογιστής πιο έξυπνος και κατ' επέκταση πιο χρήσιμος, αφού θα είναι σε θέση να ανταποκρίνεται πολύ καλύτερα στις ανάγκες και τις επιθυμίες του ανθρώπου. Από αυτήν τη σκοπιά, Τεχνητή Νοημοσύνη (TN) ονομάζεται η μελέτη των τεχνικών και των διεργασιών που δίνουν σε ένα υπολογιστή τη δυνατότητα να αποκτά διανοητικές ικανότητες, ανάλογες με αυτές που διαθέτει ο ανθρώπινος εγκέφαλος. *Υπολογιστική Νοημοσύνη (YN) ή μη συμβολική είναι το τμήμα της TN, το οποίο προέρχεται από την προσπάθεια εξομοίωσης νοητικών και γενετικών λειτουργιών των έμβιων όντων.* Στην κατηγορία αυτή ανήκουν τα *Νευρωνικά Δίκτυα* (Neural Networks - NN) και οι *Γενετικοί αλγόριθμοι* (Genetic Algorithms - GA) που είναι οι δύο περισσότερο μελετημένοι τομείς της Τεχνητής Νοημοσύνης [18].

Στα παρακάτω κεφάλαια λέγοντας Τεχνητή Νοημοσύνη εννοούμε το τμήμα της TN που αναφέρθηκε στην πάνω παράγραφο, δηλαδή την Υπολογιστική Νοημοσύνη (YN). Σ' αυτή την πτυχιακή θα ασχοληθούμε μόνο με τα *Νευρωνικά Δίκτυα*, αλλά οφείλουμε να αναφέρουμε και λίγα πράγματα γύρω από τη θεωρία των *Γενετικών Αλγορίθμων*, όπως βλέπουμε παρακάτω.

Οι Γενετικοί Αλγόριθμοι (Genetic Algorithms, GAs) βασίζονται σε ένα αφηρημένο πρότυπο της φυσικής εξέλιξης, έτσι ώστε η ποιότητα των ατόμων να χτίζεται στο πιο υψηλό επίπεδο που είναι συμβατό με το περιβάλλον (περιορισμοί του προβλήματος) (Holland 1975, Goldberg 1989) [15].

Η φιλοσοφία των γενετικών αλγορίθμων είναι η εξής: Αρχικά δημιουργείται με τυχαίο τρόπο ένα σύνολο P από υποψήφιες λύσεις του προβλήματος. Έστω N το πλήθος των στοιχείων του συνόλου P. Οι λύσεις αυτές βαθμολογούνται από μια *συνάρτηση καταλληλότητας (fitness function)*. Η βαθμολόγηση τους συνίσταται στην αντιστοίχιση σε κάθε υποψήφια λύση ενός αριθμού, ο οποίος δηλώνει την εγγύτητα της υποψήφιας μη αποδεκτής λύσης ως προς κάποια αποδεκτή. Στη συνέχεια, από τον αρχικό πληθυσμό σχηματίζονται N/2 ζευγάρια όχι απαραίτητα μοναδικών γονέων, δίνοντας μεγαλύτερη προτεραιότητα στις πλέον κατάλληλες λύσεις. Κάθε ζευγάρι *ζευγαρώνει (mates)*, δίνοντας δύο νέες λύσεις, τους *απογόνους (offsprings)*. Ο νέος πληθυσμός P' αποτελείται από το σύνολο των απογόνων και συνήθως αποτελεί βελτίωση του

προηγούμενου πληθυσμού. Η διαδικασία επαναλαμβάνεται για το νέο πληθυσμό  $P'$ . Η γενική μορφή ενός γενετικού αλγόριθμου είναι η εξής:

1. Δημιούργησε έναν αρχικό πληθυσμό  $P$ , με  $N$  υποψήφιες λύσεις.
2. Υπολόγισε την καταλληλότητα κάθε λύσης.
3. Όσο ο πληθυσμός  $P$  δε συγκλίνει σε μια λύση:  
Επανάλαβε  $N/2$  φορές τα ακόλουθα βήματα:
  - α. Επέλεξε δύο λύσεις από τον πληθυσμό  $P$ .
  - β. Συνδύασε τις δύο λύσεις για να βγάλεις δύο απογόνους,
  - γ. Υπολόγισε την καταλληλότητα των δύο απογόνων,
  - δ. Εισήγαγε τους δύο απογόνους στο νέο πληθυσμό.

#### *Αναπαράσταση Υποψήφιων Λύσεων*

Στην κλασική προσέγγιση των γενετικών αλγορίθμων, κάθε υποψήφια λύση αναπαρίσταται με μία **συμβολοσειρά** (*string*) ενός πεπερασμένου αλφάβητου. Κάθε στοιχείο της συμβολοσειράς ονομάζεται και **γονίδιο** (*gene*). Συνήθως η συμβολοσειρά αναφέρεται και σαν **χρωμόσωμα** (*chromosome*), δανειζόμενη όρους από τη βιολογία. Στο πραγματικό DNA το αλφάβητο έχει μήκος τέσσερα και αποτελείται από τα στοιχεία A, G, T και C (adenine, guanine, thymine και cytosine), ενώ στους γενετικούς αλγόριθμους χρησιμοποιείται συνήθως το δυαδικό αλφάβητο, οπότε οι συμβολοσειρές ονομάζονται και *bit-strings*. Ωστόσο υπάρχουν περιπτώσεις γενετικών αλγορίθμων που χρησιμοποιούν πιο σύνθετες μορφές αναπαράστασης.

Στα περισσότερα προβλήματα οι λύσεις περιγράφονται με μεταβλητές διαφόρων τύπων δεδομένων, επομένως η διαδικασία της κωδικοποίησης περιλαμβάνει τη μετατροπή των τιμών αυτών των μεταβλητών στις αντίστοιχες δυαδικές. Στη γενετική, το σύνολο των παραμέτρων που αναπαρίστανται από ένα συγκεκριμένο χρωμόσωμα αναφέρονται σαν **γονότυπος** (*genotype*). Ο γονότυπος περιλαμβάνει όλες εκείνες τις πληροφορίες που είναι απαραίτητες για την αναπαράσταση του οργανισμού, ο οποίος αποτελεί το **φαινότυπο** (*phenotype*) του χρωμοσώματος [15].

## Συνάρτηση Καταλληλότητας

Η *συνάρτηση καταλληλότητας* δέχεται ως είσοδο ένα χρωμόσωμα και επιστρέφει έναν αριθμό που υποδηλώνει το πόσο κατάλληλο είναι. Το πεδίο τιμών της συνάρτησης καταλληλότητας είναι συνήθως το διάστημα των πραγματικών αριθμών από το 0 ως το 1, αν και ανάλογα με την υλοποίηση αυτό θα μπορούσε να διαφέρει. Η τιμή 1 υποδηλώνει ότι το συγκεκριμένο χρωμόσωμα είναι τέλειο, δηλαδή ικανοποιεί όλες τις προϋποθέσεις του προβλήματος και αποτελεί αποδεκτή λύση. Ο τρόπος υλοποίησης της συνάρτησης βέβαια εξαρτάται από το εκάστοτε πρόβλημα.

Ανάλογα με το πρόβλημα, η κατασκευή της συνάρτησης καταλληλότητας μπορεί να είναι από απλή, έως εξαιρετικά πολύπλοκη. Η ιδανική συνάρτηση καταλληλότητας θα πρέπει να είναι συνεχής και μονότονη. Ωστόσο αυτό σπάνια συμβαίνει (εάν συνέβαινε, θα αρκούσε μια απλή αναζήτηση αναρρίχησης λόφου για να βρεθεί η βέλτιστη λύση), οπότε αυτό που επιζητείται είναι μια συνάρτηση καταλληλότητας που δεν έχει πολλά τοπικά μέγιστα ή ένα απομονωμένο ολικό μέγιστο.

Ο γενικός κανόνας στην κατασκευή της συνάρτησης καταλληλότητας είναι ότι αυτή πρέπει να αντικατοπτρίζει ρεαλιστικά την αξία του χρωμοσώματος. Ωστόσο η έννοια της αξίας ενός χρωμοσώματος δεν είναι πάντα ιδιαίτερα χρήσιμη. Για παράδειγμα, σε προβλήματα βελτιστοποίησης με περιορισμούς υπάρχουν πολλά χρωμοσώματα, δηλαδή πολλές υποψήφιες λύσεις που παραβιάζουν περιορισμούς, άρα η αξία τους είναι μηδενική.

Μια άλλη προσέγγιση που ακολουθείται πολλές φορές είναι αυτή της *προσεγγιστικής συνάρτησης καταλληλότητας* (*approximate fitness function*). Το κρίσιμο θέμα είναι η επιθυμητή ακρίβεια της συνάρτησης καταλληλότητας και το υπολογιστικό κόστος που θεωρείται αποδεκτό για μια συνάρτηση καταλληλότητας που δίνει ενδεχομένως άριστα αποτελέσματα. Εάν λοιπόν υπάρχουν διαθέσιμες δύο συναρτήσεις καταλληλότητας, μια ακριβής και μια προσεγγιστική, όπου όμως η ακριβής έχει δεκαπλάσιο χρόνο υπολογισμού από την προσεγγιστική, ενδεχομένως να συμφέρει η χρήση της προσεγγιστικής, αφού αυτή θα επιτρέψει την πραγματοποίηση δεκαπλάσιας ποσότητας υπολογισμών (δηλαδή αναπαραγωγών) στον ίδιο χρόνο, σε σχέση με τη χρονοβόρα συνάρτηση.

## Διαδικασία Επιλογής Γονέων

Η διαδικασία επιλογής γονέων σχετίζεται με την απόδοση πιθανοτήτων επιλογής προς αναπαραγωγή στα μέλη ενός πληθυσμού υποψηφίων λύσεων. Κατά τη διαδικασία αυτή, κάποιοι γονείς με υψηλή τιμή στη συνάρτηση καταλληλότητας ενδέχεται να επιλεγούν προς αναπαραγωγή περισσότερες από μία φορές, ενώ κάποιοι γονείς με χαμηλή καταλληλότητα ενδέχεται να μην επιλεγούν καθόλου.

Κατά τη διαδικασία επιλογής, αρχικά οι υποψήφιες λύσεις αντιγράφονται σε μια δεξαμενή ζευγαρώματος (*mating pool*). Η δεξαμενή αυτή έχει μέγεθος ίσο με τον αρχικό πληθυσμό, και σε αυτήν αντιγράφονται μέλη του αρχικού πληθυσμού, με πιθανότητα ανάλογη της καταλληλότητάς τους.

Για την επιλογή των χρωμοσωμάτων που θα αντιγραφούν στη δεξαμενή ζευγαρώματος χρησιμοποιούνται αρκετές τεχνικές. Μία από τις πιο συνηθισμένες είναι η *τεχνική της ρουλέτας*, η οποία συνοψίζεται στα ακόλουθα βήματα:

1. Παράγεται το άθροισμα όλων των τιμών αξιολόγησης των υποψηφίων λύσεων, έστω  $\delta$ .
2. Επιλέγεται ένας τυχαίος αριθμός  $\pi$ , από το 0 μέχρι το  $\delta$ , χρησιμοποιώντας συνάρτηση ομοιόμορφης κατανομής για τη δημιουργία των τυχαίων αριθμών.
3. Επαναληπτικά εξετάζεται κάθε υποψήφια λύση και η τιμή του προστίθεται σε έναν καταχωρητή  $K$ .
4. Αν η τιμή του  $K$  γίνει μεγαλύτερη ή ίση του  $\pi$ , η λύση επιλέγεται και ο  $K$  μηδενίζεται. Στην αντίθετη περίπτωση εκτελείται πάλι το 3.
5. Αν δεν έχει επιλεγεί ικανοποιητικός αριθμός υποψηφίων λύσεων εκτελείται το 2, αλλιώς ο αλγόριθμος τερματίζει.

Μετά την ολοκλήρωση της παραπάνω διαδικασίας, από τα μέλη της δεξαμενής ζευγαρώματος δημιουργούνται ζευγάρια με τυχαίο τρόπο και οι απόγονοι που προκύπτουν αποτελούν το νέο πληθυσμό.

Ο τρόπος επιλογής των γονέων που θα ζευγαρώσουν επηρεάζει σημαντικά την απόδοση των γενετικών αλγορίθμων. Δυο προβλήματα που εμφανίζονται συχνά στους γενετικούς αλγόριθμους και τα οποία μπορούν να αντιμετωπιστούν με τροποποίηση της διαδικασίας επιλογής, είναι η *πρόωρη σύγκλιση* (*premature convergence*) και η



*αργή σύγκλιση (slow convergence)*. Με τον όρο "*σύγκλιση*" εννοούμε την επικράτηση ενός χρωμοσώματος ή μικρών παραλλαγών του, σε μεγάλο ποσοστό στον πληθυσμό.

Με έναν αποδοτικό γενετικό αλγόριθμο, ο πληθυσμός θα πρέπει μετά από αρκετές επαναλήψεις να συγκλίνει προς το ολικό μέγιστο. Η σύγκλιση αυτή αφορά είτε το καλύτερο στοιχείο του πληθυσμού ή το μέσο όρο αυτού. Ένα γονίδιο θεωρείται ότι έχει συγκλίνει, όταν έχει την ίδια τιμή στο 95% των χρωμοσωμάτων. Ένας πληθυσμός θεωρείται ότι έχει συγκλίνει, όταν όλα τα γονίδια του έχουν συγκλίνει. Κατά την πρόωρη σύγκλιση, ο πληθυσμός πολύ γρήγορα συγκλίνει γύρω από κάποιο χρωμόσωμα, το οποίο όμως αποτελεί τοπικό μέγιστο. Το αποτέλεσμα είναι να μην μπορεί πλέον ο γενετικός αλγόριθμος να ξεφύγει από αυτό το τοπικό μέγιστο, παρά μόνο με τη διαδικασία της μετάλλαξης (αναφέρεται παρακάτω), το οποίο όμως έχει πρακτικά μηδενική πιθανότητα να συμβεί. Το φαινόμενο αυτό εμφανίζεται σε "περιπτώσεις που η συνάρτηση καταλληλότητας εμφανίζει πολύ απότομες μεταβολές έντονα τοπικά μέγιστα και μπορεί να αντιμετωπιστεί με δύο τρόπους. Ο πρώτος η απεικόνιση της συνάρτησης καταλληλότητας σε μία νέα συνάρτηση, λιγότερο απότομη (*fitness remapping*). Ο δεύτερος είναι ο καθορισμός ελάχιστων και μέγιστων ορίων, όσον αφορά το πόσες φορές επιλέγεται ένα χρωμόσωμα προς αναπαραγωγή σε κάθε κύκλο ανανέωσης του πληθυσμού.

Η *αργή σύγκλιση* είναι ουσιαστικά το ακριβώς αντίθετο φαινόμενο της πρόωρης. Κατά αυτήν, μετά από ένα μεγάλο αριθμό επαναλήψεων, ο πληθυσμός να μην έχει συγκλίνει. Το φαινόμενο εμφανίζεται όταν η έχει μικρές κλίσεις, με αποτέλεσμα τα μέγιστα και τα ελάχιστα της να έχουν μικρές διαφορές. Η λύση είναι και πάλι η απεικόνιση της συνάρτησης σε μια νέα, η οποία πρέπει να έχει πιο έντονες διακυμάνσεις.

Η μέθοδος της μερικής ανανέωσης προσεγγίζει πιο πολύ στην πραγματικότητα, αφού εκεί συνυπάρχουν πάντα σε κάποιο βαθμό οι διαφορετικές γενεές. Μάλιστα δίνεται η δυνατότητα στους απογόνους να ανταγωνιστούν τους γονείς τους, επικρατώντας και πάλι ο καλύτερος.

## Αναπαραγωγή

Αναπαραγωγή είναι η διαδικασία δημιουργίας δύο απογόνων από δύο γονείς-χρωμοσώματα. Η αναπαραγωγή μπορεί να γίνει με διάφορες τεχνικές, η σημαντικότερη από τις οποίες είναι ο συνδυασμός της *διασταύρωσης* (*cross-over*) και της *μετάλλαξης* (*mutation*) [1].

Κατά τη διασταύρωση, για κάθε ζευγάρι επιλεγμένων γονέων επιλέγεται με τυχαίο τρόπο ένα σημείο διασταύρωσης (*cross-point*), το οποίο είναι ένας αριθμός  $N$ , από το 1 μέχρι το μήκος  $L$  των bit-strings. Με βάση το σημείο διασταύρωσης, από δύο γονείς  $A$  και  $B$  παράγονται δύο απόγονοι με τον εξής τρόπο: Ο πρώτος απόγονος δημιουργείται από τα πρώτα  $N$  γονίδια του  $A$  και τα τελευταία  $L-N$  του  $B$ , ενώ ο δεύτερος δημιουργείται από τα πρώτα  $N$  γονίδια του  $B$  και τα τελευταία  $L-N$  του  $A$ .

Έστω για παράδειγμα οι γονείς  $A=011001$  και  $B=110010$  και σημείο διασταύρωσης το 3. Οι δύο απόγονοι θα είναι οι: **011010** και **110001**.

### 1.3. Η Εξέλιξη της Τεχνητής Νοημοσύνης

Η πρώτη εργασία που θεωρείται ότι ανήκει στο πεδίο της Τεχνητής Νοημοσύνης αναμφίβολα είναι αυτή των Warren McCulloch και Walter Pitts το 1933, που συσχέτιζε τους βιολογικούς νευρώνες του εγκεφάλου με απλά υπολογιστικά στοιχεία και πρότεινε ένα μοντέλο τεχνητών νευρώνων που είχε τη δυνατότητα να μαθαίνει και να υπολογίζει κάθε υπολογίσιμη συνάρτηση. Ο Donald Hebb το 1939 πρότεινε έναν απλό κανόνα της μεταβολής δύναμης της σύνδεσης μεταξύ των νευρώνων έτσι ώστε να επιτυγχάνεται η μάθηση. Το 1951, δύο μεταπτυχιακοί φοιτητές του Μαθηματικού Τμήματος του Princeton, ο Marvin Minsky και ο Dean Edmonds υλοποίησαν το πρώτο νευρωνικό δίκτυο (neural network computer) με 30 νευρώνες, το Snarc, το οποίο χρησιμοποιούσε 3.000 λυχνίες και ηλεκτρονικά εξαρτήματα από άχρηστα αεροπλάνα.

Καθοριστική στη γέννηση της ΤΝ θεωρείται το συνέδριο που διοργανώθηκε το 1956 στο Dartmouth, Μασαχουσέτη από τους McCarthy, Minsky, Claude, Shannon και Nathaniel Rochester. Το κυριότερο ίσως αποτέλεσμα του συνεδρίου ήταν η αποδοχή του ονόματος που πρότεινε ο McCarthy, για τη νέα ερευνητική περιοχή: *Τεχνητή Νοημοσύνη*. Η γλώσσα Lisp ορίστηκε το 1958 από το McCarthy σαν η επίσημη γλώσσα της ΤΝ η οποία και κυριάρχησε για μεγάλο χρονικό διάστημα (μέχρι τις αρχές του '70 που προτείνεται η Prolog). Την ίδια χρονιά προτάθηκε από τον Friedberg η τεχνική γενετικών αλγορίθμων (genetic algorithms).

Η δεκαετία του '60 ήταν επίσης δημιουργική και στην περιοχή των νευρωνικών δικτύων με βελτιώσεις της μεθόδου μάθησης του Hebb από τον Rosenblatt (1962). Ο τελευταίος απέδειξε το περίφημο θεώρημα σύγκλισης του perceptron αποδεικνύοντας ότι ο αλγόριθμος μάθησης του μπορούσε να προσαρμόζει τα βάρη στην είσοδο του νευρώνα έτσι ώστε να ταιριάζει με κάθε δεδομένο εισόδου (με την προϋπόθεση ότι υπήρχε τέτοια σύγκλιση).

Τον ενθουσιασμό της πρώτης δεκαετίας της ΤΝ διαδέχθηκε (στα τέλη της δεκαετίας του '60) η εποχή της κριτικής, με αιχμή ότι τα συστήματα ήταν κατάλληλα μόνο για παιχνίδια (toy problems). Το χαρακτηριστικό των συστημάτων της εποχής ήταν ότι περιείχαν ελάχιστη ή καθόλου γνώση για το πεδίο του προβλήματος που σχεδιάζονταν να λύσουν. Αυτό είχε σαν αποτέλεσμα η Βρετανική κυβέρνηση το 1973 να διακόψει την υποστήριξη της έρευνας στην ΤΝ. Αλλά και τα συστήματα της συνδετικής προσέγγισης της ΤΝ δηλαδή τα νευρωνικά δίκτυα, δέχθηκαν κριτική μέσα από το βιβλίο των Minsky και Papert (Perceptron, 1969) ότι είχαν περιορισμένες δυνατότητες.

Το Dendral, που αναπτύχθηκε στο Stanford το 1969 από τους Feigenbaum, Lendeborg και Buchanan, ήταν το πρώτο από τα συστήματα της επόμενης γενιάς που περιείχε σημαντική ποσότητα γνώσης εκφρασμένη με τη μορφή κανόνων. Σαν πρώτο πετυχημένο εμπορικό έμπειρο σύστημα μπορεί να θεωρηθεί το R1/XCON που αναπτύχθηκε για την εταιρεία Digital Equipments Corporation (DEC) με σκοπό τη διαμόρφωση (configuration) των παραγγελιών της με βάση τις ανάγκες των πελατών, τους αμοιβαίους περιορισμούς των διαφόρων εξαρτημάτων αλλά και τα διαθέσιμα αποθέματα τους. Στα τέλη της δεκαετίας του '80 κάθε μεγάλη επιχείρηση της Αμερικής είχε τη δικιά της ομάδα ΤΝ και χρησιμοποιούσε ή ανέπτυσσε έμπειρα συστήματα [1].

Το 1981 οι Ιάπωνες ανακοίνωσαν το πρόγραμμα της 5ης γενιάς, ένα δεκαετές πρόγραμμα για την κατασκευή υπολογιστών που θα είχαν σαν γλώσσα μηχανής την prolog και θα ήταν σε θέση να εκτελούν εκατομμύρια λογικών συμπερασμάτων το δευτερόλεπτο. Στα μέσα της δεκαετίας του '80 επανεμφανίστηκαν τα νευρωνικά δίκτυα, εμφανίστηκε πάλι ο *αλγόριθμος μάθησης με οπισθοδρόμηση (back-propagation)* που είχε προταθεί το 1969 από τους Bryson και Ho και ο οποίος σε συνδυασμό με τα νευρωνικά δίκτυα πολλών επιπέδων, εφαρμόστηκε με μεγάλη επιτυχία σε πολλά προβλήματα.

Με βάση τα προηγούμενα πολλοί συγγραφείς διακρίνουν στην ιστορία της Τεχνητής Νοημοσύνης (TN) τέσσερις περιόδους. Την προϊστορική, όπου η TN ουσιαστικά προαναγγέλλεται σε διηγήματα επιστημονικής φαντασίας, την κλασική (μέχρι τα μέσα της δεκαετίας του 1960) όπου αναπτύχθηκαν συστήματα που έπαιζαν παιχνίδια και έλυναν γρίφους, τη ρομαντική (μέχρι τα μέσα της δεκαετίας του 1970) κατά την οποία, οι προσπάθειες επικεντρώνονται στην ανάπτυξη συστημάτων που κατανοούν ιστορίες και διάλογους σε φυσική γλώσσα, και τη μοντέρνα (μέχρι τα τέλη της δεκαετίας του 1980) η οποία χαρακτηρίζεται από την ανάπτυξη συστημάτων που βασίζονται στη γνώση και την εμπορική εκμετάλλευση των αποτελεσμάτων της έρευνας γύρω από την Τεχνητή Νοημοσύνη.

Αυτήν τη στιγμή βιώνουμε τη μετά-μοντέρνα περίοδο στην οποία η TN καλείται να παίξει ένα σημαντικό ρόλο σε ένα νέο πληροφοριακό περιβάλλον του οποίου κύριο χαρακτηριστικό είναι η εξάπλωση του διαδικτύου. Ιδιαίτερη έμφαση δίνεται στην ανάπτυξη προγραμμάτων και τεχνικών (όπως τα προγράμματα πράκτορες) που διευκολύνουν τη χρήση του διαδικτύου (αναζήτηση πληροφοριών) ή την ανάπτυξη εφαρμογών που σχετίζονται με αυτό (όπως το ηλεκτρονικό εμπόριο).

#### **1.4. Η Τεχνητή Νοημοσύνη σήμερα**

Τα τελευταία χρόνια υπήρξαν σημαντικές εξελίξεις σε εφαρμογές της TN όπως η *ρομποτική, η μηχανική όραση, η μηχανική μάθηση* και ο *σχεδιασμός ενεργειών*. Αυτή τη στιγμή υπάρχουν ευφυή συστήματα τα οποία βοηθούν το χρήστη στη λειτουργία ορισμένων προγραμμάτων ευρείας καθημερινής χρήσης (π.χ. Office Assistant) ή στην

αναζήτηση στο διαδίκτυο. Ο σχεδιασμός ενεργειών έχει εφαρμοστεί με επιτυχία σε προγραμματισμό παραγωγής σε βιομηχανίες, σε αποστολές στο διάστημα, σε αντιμετώπιση κρίσεων, κλπ. Επίσης υπάρχουν συστήματα αναγνώρισης φωνής τα οποία κάνουν κρατήσεις για αεροπορικές θέσεις τηλεφωνικά βρίσκοντας τις βέλτιστες πτήσεις με κριτήρια το κόστος και το χρόνο (π.χ. το σύστημα Pegasus) ή δίνουν διάφορες πληροφορίες γενικού ενδιαφέροντος (π.χ. η φωνητική πύλη My Cosmos).

Επιπλέον υπάρχουν έμπειρα συστήματα πραγματικού χρόνου (π.χ. το σύστημα Marvel) που επεξεργάζονται τα δεδομένα που μεταδίδονται από διαστημόπλοια διεκπεραιώνοντας συνηθισμένες εργασίες και ενεργοποιώντας το συναγερμό σε δύσκολες καταστάσεις. Ακόμη υπάρχουν ρομποτικά συστήματα που οδηγούν αυτοκίνητα σε αυτοκινητόδρομο χρησιμοποιώντας βιντεοκάμερες και αποστασιόμετρα, συστήματα που διεξάγουν ιατρικές διαγνώσεις, συστήματα που ελέγχουν και ρυθμίζουν την κυκλοφορία αυτοκινήτων σε πολυσύχναστους δρόμους, συστήματα που ελέγχουν την εναέρια κυκλοφορία των αεροπλάνων και πολλά άλλα. Επίσης άνοιξη γνωρίζουν κυρίως λόγω της διάδοσης του διαδικτύου, τα *προγράμματα πράκτορες* (agents) και οι αρχιτεκτονικές συστημάτων που βασίζονται σε πράκτορες με πιο γνωστό το σύστημα Soar.

Όσον αφορά την έρευνα πάνω στην Τεχνητή Νοημοσύνη, εκφράζεται μέσω ενός αριθμού συνεδρίων και επιστημονικών περιοδικών στα οποία κάθε χρόνο δημοσιεύονται τα τελευταία επιτεύγματα στις διάφορες επιμέρους περιοχές. Τα κυριότερα συνέδρια είναι τα:

- The International Joint Conference on Artificial Intelligence (IJCAI)
- The European Conference on Artificial Intelligence (ECAI)
- The Conference of the American Association for Artificial Intelligence (AAAI)

Ενώ μερικά από τα πλέον έγκυρα περιοδικά είναι τα:

- Journal of Artificial Intelligence (Elsevier)
- IEEE Intelligent Systems
- AI-Magazine (American Association of Artificial Intelligence)
- Journal of Artificial Intelligence Research (Morgan Kaufmann)

- IEEE Transactions of Pattern Analysis & Machine Intelligence
- AI Review (Kluwer)
- Intelligence SIGART (ACM)

Επίσης στις περισσότερες χώρες υπάρχουν επιστημονικές ενώσεις με σκοπό την ανταλλαγή απόψεων και την προώθηση συνεργασιών στην περιοχή της ΤΝ, με πιο γνωστή την Αμερικάνικη ΑΑΑΙ και την ευρωπαϊκή ECCAL. Στην Ελλάδα ο αντίστοιχος φορέας είναι η Ελληνική Εταιρεία Τεχνητής Νοημοσύνης (EETN).

## ΚΕΦΑΛΑΙΟ 2 – ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ

### 2.1. Εισαγωγή

Η επίλυση προβλημάτων αποτελεί ένα από τα βασικά χαρακτηριστικά γνωρίσματα της νοημοσύνης. Η έρευνα κατά τη διάρκεια των πρώτων χρόνων της Τεχνητής Νοημοσύνης (TN) έπρεπε να καταλήξει σε τυποποιημένες μεθοδολογίες με τις οποίες να είναι δυνατή, αφενός η αναπαράσταση προβλημάτων, και αφετέρου η επίλυση τους. Για την αυτοματοποίηση της διαδικασίας επίλυσης προτάθηκαν αλγόριθμοι οι οποίοι αναζητούν λύση σε κάποιο πρόβλημα, εξετάζοντας με κάποια σειρά τα διαφορετικά στιγμιότυπα στα οποία μπορούν να περιέλθουν οι διάφορες ιδιότητες που καθορίζουν τον κόσμο ενός προβλήματος. Οι αλγόριθμοι αναζήτησης (search algorithms), όπως ονομάστηκαν, μαζί με την αναπαράσταση γνώσης (knowledge representation), συνθέτουν τον πυρήνα κάθε εφαρμογής της TN. Η επιλογή ενός αλγόριθμου αναζήτησης για ένα συγκεκριμένο πρόβλημα είναι σημαντική, διότι οι αλγόριθμοι αυτοί διαφέρουν μεταξύ τους σε αρκετά χαρακτηριστικά, όπως η αποδοτικότητα (efficiency) σε μνήμη και χρόνο, η πολυπλοκότητα (complexity), η πληρότητα (completeness) και η ευκολία υλοποίησης. Ειδικότερα, σε πολλές περιπτώσεις όπου οι λύσεις των προβλημάτων πρέπει να ικανοποιούν περιορισμούς, η επιλογή του κατάλληλου αλγορίθμου ικανοποίησης περιορισμών (constraint satisfaction) μπορεί να αποδειχτεί κρίσιμη.

Στο κεφάλαιο αυτό θα πραγματευτούμε τον ορισμό προβλημάτων, καθώς επίσης και ένα σύνολο αλγορίθμων αναζήτησης που μπορούν να τα επιλύσουν. Άλλοι από αυτούς τους αλγορίθμους αναζητούν τη λύση τυφλά, ενώ άλλοι εκμεταλλεύονται κάποια ιδιαίτερη γνώση που υπάρχει για κάθε πρόβλημα ή και περιορισμούς που πρέπει να ικανοποιηθούν έτσι ώστε η αναζήτηση να οδεύει γρηγορότερα προς τη λύση.

## 2.2. Περιγραφή Προβλημάτων

Η έννοια του προβλήματος είναι διαισθητικά γνωστή σε όλους: υπάρχει μία δεδομένη κατάσταση, υπάρχει μία επιθυμητή κατάσταση και διαθέσιμες ενέργειες που πρέπει να γίνουν ώστε να φτάσουμε στην επιθυμητή [1]. Στην καθημερινότητα, πολλές φορές χρησιμοποιείται με λάθος σημασία η λέξη "πρόβλημα", ακόμη και όταν δεν είναι γνωστό ένα από τα παραπάνω τρία βασικά συστατικά. Αν, λόγου χάρη, κάποιος δεν έχει στόχους ή δε γνωρίζει τι ενέργειες μπορεί να εκτελέσει, τότε δεν υφίσταται πρόβλημα, τουλάχιστον με την τυπική έννοια του όρου. Όμως, η επίλυση προβλημάτων που επιδιώκεται από την Τεχνητή Νοημοσύνη απαιτεί το σαφή και τυποποιημένο ορισμό τους. Επιπλέον, ο ορισμός ενός προβλήματος πρέπει να είναι ανεξάρτητος από την πολυπλοκότητα επίλυσης αυτού του προβλήματος. Ο ορισμός ή περιγραφή του προβλήματος καθορίζεται από την αναπαράσταση του, ενώ η πολυπλοκότητα καθορίζεται από την πολυπλοκότητα του αλγορίθμου αναζήτησης που εφαρμόζεται για την επίλυση του.

Η ΤΝ καλείται να επιλύσει πραγματικά και πολύπλοκα προβλήματα (real world problems). Τα προβλήματα αυτά δε διαφέρουν ουσιαστικά στον τρόπο αντιμετώπισης από όλα τα υπόλοιπα, γιατί ακολουθούν τα γενικά χαρακτηριστικά των απλών προβλημάτων (toy problems) και τις βασικές αρχές και έννοιες που τα διέπουν. Άλλωστε η μεθοδολογία που εφαρμόζεται στην αναπαράσταση και την επίλυση είναι παρόμοια και στις δύο ακραίες περιπτώσεις προβλημάτων. Αναφερόμαστε λοιπόν προς το παρόν μόνον σε απλά προβλήματα, όπως για παράδειγμα είναι οι κύβοι (blocks), το N-puzzle, η τρίλιζα (tic-tac-toe), ο λαβύρινθος (maze).

Η περιγραφή ενός προβλήματος μπορεί να γίνει με δύο βασικούς τρόπους: με περιγραφή με χώρο καταστάσεων (state space) και περιγραφή με αναγωγή (reduction). Στη συνέχεια θα εξεταστούν αναλυτικότερα οι δύο αυτοί βασικοί τρόποι.



### 2.2.1. Περιγραφή με Χώρο Καταστάσεων

Για να περιγραφεί ένα πρόβλημα με χώρο καταστάσεων, θα πρέπει να αναπαρασταθεί πρώτα ο κόσμος του προβλήματος και στη συνέχεια τα συγκεκριμένα χαρακτηριστικά ενός προβλήματος, όπως η αρχική κατάσταση, η τελική κατάσταση και οι τελεστές μετάβασης ή ενέργειες που μπορούν να γίνουν ώστε να εξευρεθεί η λύση του. Σημαντικά χαρακτηριστικά σε ένα τέτοιου είδους πρόβλημα είναι το όνομα του, η κατάστασή του καθώς και οι τελεστές μετάβασής του, όπως περιγράφονται παρακάτω.

#### i) Κόσμος προβλήματος

Ο κόσμος ενός προβλήματος (problem world) αποτελείται μόνον από τα αντικείμενα που υπάρχουν σε αυτόν, τις ιδιότητες των αντικειμένων και τις σχέσεις που τα συνδέουν. Δηλαδή, ο κόσμος ενός προβλήματος είναι υποσύνολο του πραγματικού κόσμου και περιέχει μόνον όσα αντικείμενα έχουν άμεση σχέση με το πρόβλημα [1]. Ο κόσμος χαρακτηρίζεται ως κλειστός (closed world) όταν κανένα νέο αντικείμενο, καμία νέα ιδιότητα ή καμία νέα σχέση δεν εισάγεται ή εξάγεται από ή προς έναν άλλο κόσμο. Αντίθετα ένας κόσμος χαρακτηρίζεται ως ανοιχτός (open world) όταν δέχεται εξωτερικές παρεμβάσεις που αλλάζουν την περιγραφή του. Οι ανοιχτοί κόσμοι αντιμετωπίζονται πολύ δύσκολα επειδή δεν είναι προβλέψιμοι. Παρόλο που οι ανοιχτοί κόσμοι είναι πιο κοντά στα πραγματικά προβλήματα, για λόγους απλότητας στη συνέχεια θα θεωρηθούν μόνον κλειστοί κόσμοι. Για παράδειγμα, στον κόσμο των κύβων δε μπορεί ξαφνικά να εμφανιστεί ένας νέος κύβος ή να εξαφανιστεί ένας υπάρχων. Όμοια στον κόσμο του λαβυρίνθου δεν είναι δυνατόν να μην υπάρχει έξοδος αφού κάτι τέτοιο δεν προβλέπεται από το χώρο του προβλήματος.

#### ii) Όνομα και περιγραφή

Λέγοντας όνομα εννοούμε ένα επίθετο ή ουσιαστικό κυρίως από το οποίο μπορούμε να καταλάβουμε αρκετές φορές (αν όχι τις περισσότερες) περίπου το περιεχόμενο του προβλήματος. Όσον αφορά την περιγραφή είναι η λεπτομερής παρουσίαση του προβλήματος. Τέσσερα σύντομα παραδείγματα παρουσιάζονται

παρακάτω, όπου πρώτα δίνεται το όνομα και κατόπιν η περιγραφή του κάθε προβλήματος.

Κύβοι:

Τρεις κύβοι βρίσκονται σε τυχαία διάταξη πάνω στο τραπέζι. Σκοπός είναι να μετακινηθούν οι κύβοι ώστε να σχηματιστεί μια άλλη επιθυμητή διάταξη. Ένας κύβος μπορεί να μετακινηθεί, εφόσον έχει ελεύθερη την πάνω έδρα του.

N-Puzzle:

Ένα πλαίσιο περιέχει N πλακίδια, έστω 8, αριθμημένα από το 1 έως το 8. Τα πλακίδια μπορούν να μετακινηθούν σε γειτονική θέση, εφόσον η θέση αυτή είναι κενή. Σκοπός είναι να έρθουν τα πλακίδια σε κάποια επιθυμητή διάταξη.

Τρίλιζα:

Δύο παίκτες σημειώνουν εναλλάξ X ή O στις εννέα θέσεις του αρχικά κενού ταμπλό με σκοπό κάποιος να έχει τελικά τρία X ή τρία O στη σειρά, οριζόντια, κατακόρυφα ή διαγώνια.

Λαβύρινθος:

Αναζήτηση μίας διαδρομής σε ένα λαβύρινθο από την είσοδο έως την έξοδο.

iii) Κατάσταση προβλήματος

Κατάσταση ενός προβλήματος είναι ένα στιγμιότυπο (instance) ή φωτογραφία (snapshot) μίας συγκεκριμένης χρονικής στιγμής της εξέλιξης του προβλήματος [1].

Η τυπική αναπαράσταση μίας κατάστασης δε μπορεί να γίνει με απεικόνιση, αλλά μόνο με μία περιγραφή των χαρακτηριστικών της. Η σχηματική περιγραφή είναι διαισθητική, αλλά όχι αρκετή για την αυτοματοποίηση της επίλυσης προβλημάτων. Για το λόγο αυτόν η περιγραφή πρέπει να γίνει με τέτοιο τρόπο που να είναι αποδεκτή και αξιοποιήσιμη από ένα υπολογιστικό σύστημα. Για παράδειγμα, μία τέτοια περιγραφή είναι η προτασιακή ή κατηγορηματική λογική. Η κατάσταση ως σύνολο αντικειμένων, ιδιοτήτων και σχέσεων μεταξύ τους αναπαριστάται με ευκολία στη γλώσσα της λογικής.

Κατάσταση (state) του κόσμου ενός προβλήματος είναι μία επαρκής αναπαράσταση του προβλήματος σε μία δεδομένη χρονική στιγμή. Με τη λέξη "επαρκής" εννοείται ότι οι χαρακτηριστικές ιδιότητες μιας κατάστασης του κόσμου του προβλήματος πρέπει να

επιλεγούν με τέτοιο τρόπο, ώστε διαφορετικές τιμές των ιδιοτήτων να αντικατοπτρίζουν τις πραγματικές διαφορές των στιγμιότυπων του κόσμου. Αυτό επιτυγχάνεται με τη λειτουργία της αφαίρεσης (abstraction), σύμφωνα με την οποία για να περιγραφεί μία κατάσταση αφαιρούνται όλες εκείνες οι λεπτομέρειες οι οποίες δεν είναι απαραίτητες για την μετέπειτα επίλυση ενός προβλήματος. Ο Πίνακας 1 δείχνει τη λεκτική περιγραφή της κατάστασης του κόσμου των κύβων.

**Πίνακας 1. Λεκτική περιγραφή μίας κατάστασης του κόσμου των κύβων.**

Αντικείμενα	Ιδιότητες	Σχέσεις
Κύβος Α	Κύβος Α είναι ελεύθερος	Κύβος Α πάνω στον κύβο Β
Κύβος Β	Κύβος Γ είναι ελεύθερος	Κύβος Β πάνω στο Τ
Κύβος Γ	Τ έχει αρκετό ελεύθερο χώρο	Κύβος Γ πάνω στο Τ
Τ είναι τραπέζι	Κύβος Β δεν είναι ελεύθερος	

Στην παραπάνω περιγραφή δεν αναφέρονται κάποιες λεπτομέρειες λόγω της αφαίρεσης τους, όπως για παράδειγμα ότι "ο κύβος Γ βρίσκεται αριστερά του Α", ή ότι "το Τραπέζι είναι οριζόντιο", κλπ. Αυτές οι λεπτομέρειες δεν ενδιαφέρουν τη μετέπειτα επίλυση του προβλήματος, χωρίς βέβαια αυτό να σημαίνει ότι αν υπήρχε συγκεκριμένος λόγος αυτές δε θα αναφέρονταν.

#### **iv) Τελεστές μετάβασης**

Οι καταστάσεις του κόσμου ενός προβλήματος συνδέονται μεταξύ τους, με την έννοια ότι από μία κατάσταση μπορεί να προκύψει μία νέα κατάσταση. Η παραγωγή νέων καταστάσεων οφείλεται στους τελεστές μετάβασης ή ενέργειες που μπορούν να γίνουν σε κάποια κατάσταση του κόσμου του προβλήματος. Τελεστής μετάβασης (transition operator) είναι μια αντιστοίχιση μίας κατάστασης του κόσμου σε νέες καταστάσεις. Αν δηλαδή εφαρμοστεί ένας τελεστής μετάβασης σε μία κατάσταση του κόσμου, επιστρέφει άλλες νέες καταστάσεις. Τυπικά, το σύνολο των τελεστών μετάβασης  $T$  ορίζεται ως η απεικόνιση (relation)  $T:S \rightarrow S$  και αντίθετα. Το πεδίο ορισμού και τιμών είναι το  $S$ , δηλαδή το σύνολο όλων των έγκυρων (valid) καταστάσεων. Εναλλακτικά, οι τελεστές μετάβασης μπορούν να οριστούν και ως συνάρτηση που επιστρέφει σύνολο από καταστάσεις, δηλαδή  $T:S \rightarrow \text{Powerset}(S)$  [1].

Στο παράδειγμα του κόσμου των κύβων οι τελεστές μετάβασης είναι:

Βάλε τον κύβο A πάνω στον κύβο Γ. Βάλε τον κύβο A πάνω στον κύβο B. Βάλε τον κύβο A πάνω στον τραπέζι T.

Δηλαδή όλες οι πιθανές ενέργειες που μπορεί να υπάρξουν. Οι τελεστές μετάβασης μπορεί να περιέχουν μεταβλητές, για να αποφευχθεί ο μεγάλος αριθμός ενεργειών που προκύπτει από κάθε πιθανό συνδυασμό. Έτσι, δύο από τους παραπάνω τελεστές είναι στιγμιότυπα ενός γενικότερου τελεστή:

Βάλε κάποιον κύβο X πάνω σε κάποιον κύβο Y.

Οι τελεστές επιτυγχάνουν να αλλάξουν την κατάσταση του κόσμου του προβλήματος μέσω της αλλαγής της περιγραφής αυτής της κατάστασης. Νέες τιμές ιδιοτήτων και νέα στιγμιότυπα σχέσεων αντικαθιστούν τις τιμές ιδιοτήτων και τα στιγμιότυπα σχέσεων της προηγούμενης κατάστασης. Για παράδειγμα, τα αποτελέσματα της παραπάνω ενέργειας είναι ότι ο κύβος X βρίσκεται πάνω στον Y, ο Y δεν είναι πλέον ελεύθερος, ο X δεν είναι πάνω στον κύβο Z που βρισκόταν προηγουμένως και ο κύβος Z είναι πλέον ελεύθερος.

Οι τελεστές έχουν συνήθως προϋποθέσεις εφαρμογής (preconditions), που πρέπει να ισχύουν για να μπορούν να εφαρμοστούν. Για παράδειγμα, η εφαρμογή του παραπάνω τελεστή γίνεται μόνο όταν ο X και ο Y είναι ελεύθεροι, ο X βρίσκεται πάνω στο Z και ο X και Y είναι δύο διαφορετικοί κύβοι.

#### ***v) Χώρος Καταστάσεων***

Χώρος καταστάσεων (state space ή domain space) ενός προβλήματος ονομάζεται το σύνολο όλων των έγκυρων καταστάσεων. Ο χώρος καταστάσεων απεικονίζεται συνήθως διαισθητικά με ένα γράφο (graph). Κάθε κόμβος είναι μία κατάσταση και κάθε ακμή που ενώνει δύο κόμβους είναι ένας τελεστής. Τυπικά, ο χώρος καταστάσεων μπορεί να αναπαρασταθεί με περιγραφή σε μία κατάλληλη γλώσσα, όπως για παράδειγμα τη λογική [1].

### **vi) Αρχικές και τελικές καταστάσεις**

Ένα πρόβλημα χαρακτηρίζεται από δύο ειδικού τύπου καταστάσεις που εκφράζουν το δεδομένο και το ζητούμενο. Αυτές οι καταστάσεις ονομάζονται αρχική (initial state) και τελική (final ή goal state) αντίστοιχα.

Η τελική κατάσταση σε κάποια προβλήματα είναι πλήρως γνωστή, δηλαδή είναι γνωστή η πλήρης περιγραφή της, όπως για παράδειγμα στο 8-Puzzle που η τελική μορφή πρέπει να βρίσκει στη σειρά από το 1 έως το 8 διατεταγμένα τα στοιχεία του κύβου. Σε άλλα προβλήματα, η τελική κατάσταση δεν είναι πλήρως ορισμένη, αλλά είναι γνωστές κάποιες επιθυμητές ιδιότητες της, όπως για παράδειγμα στο σκάκι, όπου η τελική κατάσταση (νίκη - ισοπαλία) καθορίζεται από τους κανόνες ματ.

### **vii) Ορισμός προβλήματος**

Ένα πρόβλημα (Problem) ορίζεται από την αρχική κατάσταση, το σύνολο των τελικών καταστάσεων, το σύνολο των τελεστών μετάβασης και το χώρο καταστάσεων. Τυπικά, ένα πρόβλημα ορίζεται ως η τετράδα  $P = (I, G, T, S)$  όπου:

- $I$  είναι η αρχική κατάσταση,  $I \in S$
- $G$  είναι το σύνολο των τελικών καταστάσεων,  $G \subseteq U$
- $T$  είναι το σύνολο των τελεστών μετάβασης.  $T: S \rightarrow S$
- $S$  είναι ο χώρος καταστάσεων

### **viii) Λύση προβλήματος**

Λύση (Solution) σε ένα πρόβλημα  $(I, G, T, S)$ , είναι μία ακολουθία από τελεστές μετάβασης:  $t_1, t_2, t_3, \dots, t_n \in T$

με την ιδιότητα:  $g = t_n(\dots(t_2(t_1(I))))$ , όπου  $g \in G$

δηλαδή, αν εφαρμοστεί πρώτα το  $t_1$  στην αρχική κατάσταση, μετά το  $t_2$  στην κατάσταση που προκύπτει από την εφαρμογή του  $t_1$  κ.ο.κ., το αποτέλεσμα θα είναι μία τελική κατάσταση. Από τον παραπάνω ορισμό προκύπτει ότι λύση σε ένα πρόβλημα είναι η ακολουθία τελεστών που εφαρμόζονται στην αρχική κατάσταση για να προκύψει η τελική κατάσταση.

Δηλαδή είναι εμφανές ότι σε προβλήματα που είναι πλήρως γνωστές οι τελικές καταστάσεις ενδιαφέρει η ακολουθία των ενεργειών που έγιναν για να επιτευχθεί μία τελική κατάσταση. Έτσι, για παράδειγμα στο 8-puzzle λύση είναι οι κινήσεις που έγιναν ώστε να διαμορφωθεί το τελικό puzzle. Σε άλλα προβλήματα όμως, όπου αναζητούνται οι τιμές των ιδιοτήτων ή τα στιγμιότυπα των σχέσεων που εκφράζουν πλήρως, μία τελική κατάσταση, λύση θεωρείται η ίδια η τελική κατάσταση και όχι η σειρά με την οποία οι τελεστές μετάβασης ανέθεσαν αυτές τις τιμές ή βρήκαν τις σχέσεις. Για παράδειγμα, λύση σε ένα σταυρόλεξο είναι όλες οι λέξεις τοποθετημένες στη σωστή τους θέση. Μία τέτοια κατηγορία προβλημάτων είναι τα προβλήματα με ικανοποίηση περιορισμών.

### 2.2.2. Περιγραφή με Αναγωγή

Αντίθετα με την περιγραφή του χώρου καταστάσεων, όπου η βασική δομή είναι η κατάσταση, στην περιγραφή με αναγωγή (reduction) η βασική δομή είναι η ίδια η περιγραφή του προβλήματος. Η γενική ιδέα της αναγωγής είναι πως αν δοθεί μια αρχική περιγραφή ενός προβλήματος, τότε μία ακολουθία από τελεστές μπορούν να ανάγουν την περιγραφή σε υποπροβλήματα τα οποία είναι άμεσα επιλύσιμα. Τέτοια προβλήματα ονομάζονται αρχέγονα (Primitive Problems) [1].

Επίσης αντί για αρχική κατάσταση υπάρχει η αρχική περιγραφή του προβλήματος (initial problem description). Αντίστοιχα, τελική κατάσταση είναι η τελική περιγραφή του προβλήματος (final problem description).

Ένας τελεστής αναγωγής (reduction operator) ανάγει ένα πρόβλημα σε υποπροβλήματα. Αυτό σημαίνει πως για να λυθεί το πρόβλημα θα πρέπει να λυθούν όλα τα υποπροβλήματα. Μπορεί να υπάρχουν πολλοί διαφορετικοί τελεστές αναγωγής που να εφαρμόζονται σε ένα πρόβλημα.

### **Ορισμός προβλήματος**

Στην αναπαράσταση με αναγωγή, ένα πρόβλημα ορίζεται τυπικά ως η τετράδα:

$$P = (ID, GD, TR, PP) ,$$

όπου ID είναι η αρχική περιγραφή, GD είναι ένα σύνολο από τελικές περιγραφές, TR είναι ένα σύνολο τελεστών αναγωγής και PP είναι ένα σύνολο από αρχέγονα προβλήματα.

Προβλήματα με χωρική διάσπαση (spatial decomposition) είναι προβλήματα που λύνονται κατ' εξοχήν με αναγωγή. Ένα παράδειγμα τέτοιου προβλήματος είναι η εύρεση διαδρομής σε ένα λαβύρινθο που αποτελείται από έναν αριθμό διαμερισμάτων. Το πρόβλημα μπορεί να αναχθεί σε υποπροβλήματα, δηλαδή στην εύρεση διαδρομής από την είσοδο έως την έξοδο κάθε διαφορετικού διαμερίσματος.

Μία πολύ χρήσιμη τεχνική για την επίλυση προβλημάτων με αναγωγή είναι η αναδρομή (recursion), στην οποία οι σχέσεις ορίζονται με βάση τον εαυτό τους, π.χ.  $A^v = A^{v-1} \times A$ . Έτσι και στην περιγραφή προβλημάτων με αναγωγή, η επίλυση τους βασίζεται στο ότι αν θεωρηθεί ότι έχει επιλυθεί το ίδιο, αλλά κατά μία τάξη μικρότερης κλίμακας πρόβλημα, η συνολική επίλυση του προβλήματος έγκειται στην (άμεση) επίλυση ενός αρχέγονου προβλήματος. Η αναγωγή είναι μία αρκετά διαδεδομένη μέθοδος επίλυσης προβλημάτων. Για παράδειγμα η δημοφιλής γλώσσα προγραμματισμού Prolog βασίζεται στην αναγωγή. Τα αρχέγονα προβλήματα αντιστοιχούν είτε σε ενσωματωμένα κατηγορήματα είτε ή σε γεγονότα που είναι άμεσα επιλύσιμα. Η αρχική περιγραφή του προβλήματος είναι η ερώτηση που πρέπει να επιλυθεί, ενώ η τελική περιγραφή είναι η ερώτηση μαζί με τις τιμές που έχουν πάρει οι μεταβλητές του κατηγορήματος της ερώτησης. Ο τελεστής αναγωγής είναι η αντικατάσταση ενός στόχου με το σώμα της πρότασης που ενοποιείται με αυτόν το στόχο, όπως ορίζεται στο μηχανισμό εκτέλεσης της Prolog.

Καταλήγοντας, υπάρχουν αντιστοιχίες μεταξύ των δύο τύπων περιγραφής προβλημάτων που αναφέρθηκαν. Συνήθως ένα πρόβλημα ταιριάζει καλύτερα στον ένα τύπο ή στον άλλο. Για παράδειγμα ένα πρόβλημα που εκ φύσεως ταιριάζει στην αναγωγή, δε σημαίνει πως δε μπορεί να αναπαρασταθεί και με χώρο καταστάσεων.

### **2.3. Αλγόριθμοι αναζήτησης προβλημάτων**

Δοθέντος ενός προβλήματος που περιγράφηκε με χώρο καταστάσεων ή με αναγωγή στόχος είναι να βρεθεί η λύση του. Όπως αναφέρθηκε, η τυποποίηση της περιγραφής ενός προβλήματος διευκολύνει την επίλυσή του. Κατ' αντιστοιχία η τυποποίηση των βημάτων επίλυσης διευκολύνει την αυτοματοποίηση, δηλαδή την υλοποίηση του τρόπου επίλυσης σε ένα υπολογιστικό σύστημα. Η τυποποίηση αυτή επιτυγχάνεται μέσω αυστηρά προκαθορισμένων βημάτων, δηλαδή αλγορίθμων, που πρέπει να εφαρμοστούν για να επιλυθεί ένα πρόβλημα. Επειδή οι αλγόριθμοι αυτοί αναζητούν τη λύση στο πρόβλημα

ονομάζονται αλγόριθμοι αναζήτησης (search algorithms). Τέτοιοι υπάρχουν και πολλές φορές συγκεκριμένα προβλήματα δίνουν την αφορμή για το σχεδιασμό νέων αλγορίθμων. Ο Πίνακας 2 παρουσιάζει μερικούς αλγόριθμους αναζήτησης, καθώς και τις συντομογραφίες που χρησιμοποιούνται [1].

Οι αλγόριθμοι αναζήτησης δεν αντανακλούν πάντα τον τρόπο της ανθρώπινης σκέψης, αποτελούν όμως μία σαφή μαθηματική μεθοδολογία. Ο λόγος για την ύπαρξη τόσων αλγορίθμων είναι ότι κάθε ένας έχει διαφορετική πολυπλοκότητα (complexity) που τον καθιστά περισσότερο ή λιγότερο αποδοτικό (efficient) σε απαίτηση μνήμης ή χρόνο εκτέλεσης από άλλους αλγόριθμους κατά τη διάρκεια της αναζήτησης λύσεων σε συγκεκριμένη κατηγορία προβλημάτων.

**Πίνακας 2. Αλγόριθμοι Αναζήτησης και Συντομογραφίες τους**

Όνομα Αλγορίθμου	Συντομογραφίες	Ελληνική Ορολογία
Depth-First Search	DFS	Αναζήτηση Πρώτα σε Βάθος
Breadth-First Search	BFS	Αναζήτηση Πρώτα σε Πλάτος
Branch and Bound	B&B	Επέκταση και Οριοθέτηση
Hill Climbing	HS	Αναρρίχηση Λόφων
Beam Search	BS	Ακτινωτή Αναζήτηση
Best-First Search	BestFS	Αναζήτηση Πρώτα στο Καλύτερο
A*(A-star)	A*	A* (Άλφα Άστρο)
Iterative Deepening	ID	Επαναληπτική Εκβάθυνση
Bi-directional	BiS	Αναζήτηση Διπλής Κατεύθυνσης
Minimax	Minimax	Αναζήτηση Μεγίστου-Ελαχίστου
Alpha-Beta	AB	Άλφα-Βήτα

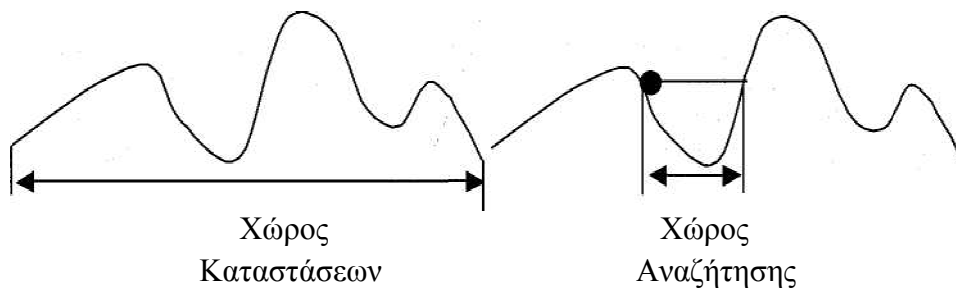
### Χώρος Αναζήτησης

Ένα πρόβλημα ορίστηκε προηγουμένως ως η τετράδα (I, G, T, S), όπου I είναι η αρχική κατάσταση, G είναι το σύνολο των τελικών καταστάσεων, T είναι το σύνολο των τελεστών μετάβασης και S είναι ο χώρος καταστάσεων. Σκοπός ενός αλγορίθμου αναζήτησης είναι να προσπαθήσει να βρει τη λύση μέσα στο χώρο καταστάσεων. Φυσιολογικά ένας τέτοιος αλγόριθμος πρέπει να εξετάσει μόνον το υποσύνολο του χώρου καταστάσεων στο οποίο ανήκει και η αρχική κατάσταση. Δοθέντος ενός προβλήματος (I, G, T, S), χώρος αναζήτησης (search space-SP) είναι το σύνολο όλων των καταστάσεων που είναι προσβάσιμες από την αρχική κατάσταση. Τυπικά, μία κατάσταση S ονομάζεται



προσβάσιμη (accessible) αν υπάρχει μια ακολουθία τελεστών μετάβασης  $t_1, t_2, \dots, t_k \in T$  τέτοια ώστε  $S = t_k(\dots(t_2(t_1(I))))$ .

Η διαφορά μεταξύ του χώρου καταστάσεων και του χώρου αναζήτησης είναι λεπτή. Ο χώρος αναζήτησης είναι υποσύνολο του χώρου καταστάσεων, δηλαδή  $SP \subseteq S$ . Η διαφορά έγκειται στο γεγονός ότι εξ' ορισμού ο χώρος αναζήτησης εξαρτάται από την αρχική κατάσταση, ενώ ο χώρος καταστάσεων όχι. Μόνον όταν όλες οι καταστάσεις του χώρου καταστάσεων είναι προσβάσιμες από την αρχική κατάσταση, ο χώρος αναζήτησης και ο χώρος καταστάσεων ταυτίζονται. Για παράδειγμα, στο Σχήμα 2.1, η μπίλια μπορεί να βρεθεί σε οποιοδήποτε σημείο της επιφάνειας (χώρος καταστάσεων). Αν όμως αυτή αφηθεί ελεύθερη σε ένα σημείο της επιφάνειας, οι καταστάσεις οι οποίες είναι προσβάσιμες είναι αυτές που ενεργειακά είναι δυνατές (χώρος αναζήτησης).

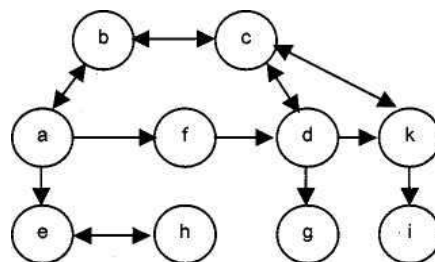


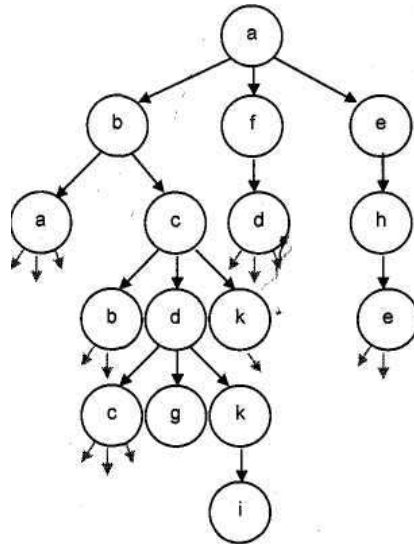
*Ο χώρος αναζήτησης είναι υποσύνολο του χώρου καταστάσεων.*

Όπως ο χώρος καταστάσεων, έτσι και ο χώρος αναζήτησης μπορεί να αναπαρασταθεί με γράφο. Είναι πάντα εφικτό να μετατραπεί ο γράφος σε δένδρο αναζήτησης (search tree), το οποίο όμως μπορεί να έχει μονοπάτια απείρου μήκους. Τα δένδρα αυτά είναι OR-δένδρα (OR-trees) γιατί σε κάθε κόμβο υπάρχουν εναλλακτικοί τελεστές που μπορεί να εφαρμοστούν. Ο Πίνακας 3 περιέχει την αντιστοιχία μίας δομής δένδρου με το χώρο αναζήτησης ενός προβλήματος.

**Πίνακας 3. Αντιστοιχία δομής δένδρου και χώρου αναζήτησης**

<b>Τμήμα Δένδρου</b>	<b>Αναπαράσταση</b>
Κόμβος (Node)	Κατάσταση
Ρίζα (Root)	Αρχική Κατάσταση
Φύλλο (Tip, Leaf)	Τελική Κατάσταση ή Αδιέξοδο ( <i>Dead Node</i> ), δηλαδή κατάσταση στην οποία δεν μπορεί να εφαρμοστεί κανένας τελεστής μετάβα-
Κλαδί (Branch)	Τελεστής Μετάβασης που μετατρέπει μια κατάσταση-Γονέα (Parent State) σε μία άλλη κατάσταση-Παιδί (Child State)
Λύση (Solution)	Μονοπάτι (Path) που ενώνει την αρχική με μία τελική κατάσταση
Επέκταση (Expansion)	Η διαδικασία παραγωγής όλων των καταστάσεων-παιδιών ενός κόμβου
Παράγοντας Διακλάδωσης (Branching Factor)	Ο αριθμός των καταστάσεων-παιδιών που προκύπτουν από μία επέκταση. Επειδή δεν είναι σταθερός αριθμός, αναφέρεται και ως Μέσος Παράγοντας Διακλάδωσης (Average Branching Factor)





Στο παραπάνω σχήμα απεικονίζεται ένας χώρος αναζήτησης και το αντίστοιχο OR-δένδρο. Οι κόμβοι α μέχρι k είναι καταστάσεις. Η ρίζα α είναι η αρχική κατάσταση. Τα φύλλα g και i είναι αντίστοιχα η τελική κατάσταση και ένα αδιέξοδο. Οι συνδέσεις μεταξύ κόμβων είναι τα κλαδιά του δένδρου. Μία λύση στο πρόβλημα αναζήτησης αποτελούν τα κλαδιά που συνδέουν με τη σειρά τους κόμβους a-b, b-c, c-d, d-g. Ο παράγοντας διακλάδωσης (branching factor) εκφράζει τον αριθμό των καταστάσεων που προκύπτουν από μία άλλη κατάσταση. Για παράδειγμα, ο παράγοντας διακλάδωσης είναι 3 για τον κόμβο α, 2 για τον κόμβο f, 1 για τον e, κ.ο.κ. Όπως προαναφέρθηκε, το αντίστοιχο δένδρο μπορεί να έχει και ακολουθία κλαδιών απείρου μήκους, όπως φαίνεται από στο παραπάνω σχήμα.

Το πρόβλημα που παρουσιάζεται στους χώρους αναζήτησης πραγματικών προβλημάτων είναι ο γρήγορος ρυθμός με τον οποίον αναπτύσσεται το δένδρο. Έστω, για παράδειγμα, ένα τυπικό πρόβλημα μεσαίου μεγέθους, όπου το δένδρο έχει μέσο παράγοντα διακλάδωσης 10 και η λύση βρίσκεται σε βάθος 20, τότε ο χώρος αναζήτησης μπορεί να φτάσει την τάξη του  $10^{20}$ . Το φαινόμενο αυτό της εκθετικής αύξησης του αριθμού των κόμβων του δένδρου ονομάζεται συνδυαστική έκρηξη (combinatorial explosion). Λόγω της συνδυαστικής έκρηξης αυξάνονται εκθετικά και οι απαιτήσεις ενός αλγορίθμου σε μνήμη και χρόνο, με αποτέλεσμα να είναι πρακτικά αδύνατον να βρεθεί λύση σε πραγματικό χρόνο.

## Χαρακτηριστικά Αλγορίθμων

Ένας αλγόριθμος είναι μία αυστηρά καθορισμένη ακολουθία βημάτων-εντολών, η οποία επιδιώκει να λύσει ένα πρόβλημα [1]. Μετά την εφαρμογή κάποιου αλγορίθμου στο χώρο αναζήτησης ενός προβλήματος  $P=(I, G, T, S)$  προκύπτει το λυμένο πρόβλημα (solved problem), το οποίο ορίζεται ως μία τετράδα  $P_s=(V, A, F, G^s)$  όπου:

- $V$  είναι το σύνολο των καταστάσεων που εξέτασε ο αλγόριθμος αναζήτησης,
- $A$  είναι ο αλγόριθμος που χρησιμοποιήθηκε,
- $F$  είναι το σύνολο των λύσεων που βρέθηκαν, και
- $G^s$  είναι το σύνολο των τελικών καταστάσεων που εξετάστηκαν.

Η αντιπαράθεση της τετράδας  $P$  ενός προβλήματος, του χώρου αναζήτησης και της παραπάνω τετράδας  $P_s$  ενός λυμένου προβλήματος επιτρέπει κάποιες συγκρίσεις, οι οποίες με τη σειρά τους οδηγούν σε ορισμούς νέων εννοιών. Κατ' αρχήν, το σύνολο  $V$  είναι υποσύνολο του χώρου αναζήτησης,  $V \subseteq SP$ . Ο πληθάριθμος του  $V$  (ο αριθμός των καταστάσεων που περιέχει) είναι ένα από τα χαρακτηριστικά της αποδοτικότητας του αλγορίθμου. Όταν το σύνολο των καταστάσεων που εξετάζει ο αλγόριθμος για να βρει τις απαιτούμενες λύσεις είναι ίσο με το χώρο αναζήτησης, δηλαδή  $V=SP$ , τότε ο αλγόριθμος ονομάζεται εξαντλητικός (exhaustive).

Ένας αλγόριθμος δε λύνει πάντα κάποιο πρόβλημα, έστω και αν υπάρχει κάποια λύση. Τότε τα σύνολα  $G^s$  και  $F$  είναι κενά. Ένας αλγόριθμος αναζήτησης ονομάζεται πλήρης (complete) αν εγγυάται ότι θα βρει μία λύση για οποιαδήποτε τελική κατάσταση, αν τέτοια λύση υπάρχει. Σε αντίθετη περίπτωση, ο αλγόριθμος ονομάζεται ατελής (incomplete).

Η πληρότητα ενός αλγορίθμου μπορεί να αποδειχθεί μόνο με μαθηματικές μεθόδους. Ωστόσο για ορισμένες περιπτώσεις είναι σίγουρο πως αν ο αλγόριθμος δε βρει λύση, τότε οπωσδήποτε δεν υπάρχει λύση στο πρόβλημα, όπως για παράδειγμα στην περίπτωση που ένας αλγόριθμος είναι εξαντλητικός. Η μη εύρεση λύσης από έναν πλήρη αλγόριθμο είναι εξίσου σημαντική με την εύρεση λύσης στο ίδιο πρόβλημα.

Μερικά προβλήματα έχουν διατεταγμένο το σύνολο των τελικών καταστάσεων, σύμφωνα με τη σημαντικότητα - αξία της κάθε τελικής κατάστασης. Μία λύση ονο-

μάζεται βέλτιστη (optimal) αν οδηγεί στην καλύτερη, σύμφωνα με τη διάταξη, τελική κατάσταση. Όταν δεν υπάρχει διάταξη, μία λύση ονομάζεται βέλτιστη αν είναι η συντομότερη (shortest), δηλαδή αν περιέχει το μικρότερο αριθμό τελεστών μετάβασης που οδηγούν σε κάποια τελική κατάσταση. Διαφορετικοί αλγόριθμοι βρίσκουν λύσεις διαφορετικής ποιότητας. Ένας αλγόριθμος αναζήτησης καλείται αποδεκτός (admissible) αν εγγυάται ότι θα βρει τη βέλτιστη λύση, αν μια τέτοια λύση υπάρχει.

### **Διαδικασία Επιλογής ενός Αλγορίθμου Αναζήτησης**

Δοθέντος ενός προβλήματος, είναι σημαντικό να επιλεγεί ο καταλληλότερος αλγόριθμος για την επίλυση του. Η επιλογή αυτή γίνεται βάσει κάποιων κριτηρίων τα οποία όμως δε μπορούν να τυποποιηθούν. Η επιλογή εξαρτάται κυρίως από τη φύση του προβλήματος και σε μεγάλο βαθμό από τους συμβιβασμούς που πρέπει να γίνουν. Για παράδειγμα, μπορεί κάποιος να είναι διατεθειμένος να θυσιάσει την αποδοτικότητα σε χώρο ή χρόνο προς χάριν της καλύτερης λύσης ή να θυσιάσει την πληρότητα, επιδιώκοντας τη γρήγορη εύρεση οποιασδήποτε λύσης. Εν συντομία, η επιλογή ενός αλγορίθμου βασίζεται στα εξής κριτήρια :

- τον αριθμό των καταστάσεων που αυτός επισκέπτεται
- τη δυνατότητα εύρεσης λύσεων εφόσον αυτές υπάρχουν
- τον αριθμό των λύσεων
- την ποιότητα των λύσεων
- την αποδοτικότητα του σε χρόνο
- την αποδοτικότητα του σε χώρο (μνήμη)
- την ευκολία υλοποίησης του

Στα παραπάνω κριτήρια εντάσσεται και η έννοια του κλαδέματος ή αποκοπής καταστάσεων (pruning) του χώρου αναζήτησης. Αποκοπή είναι η διαδικασία κατά την οποία ο αλγόριθμος απορρίπτει, κάτω από ορισμένες συνθήκες, κάποιες καταστάσεις και μαζί με αυτές όλο το υποδένδρο που εκτυλίσσεται κάτω από τις καταστάσεις αυτές. Η αποκοπή μπορεί να βασίζεται είτε σε αντικειμενικά κριτήρια όταν είναι σίγουρο ότι δεν υπάρχει λύση από εκεί και κάτω ή σε αυθαίρετα, ευριστικά όπως αναφέρεται στη συνέχεια, κριτήρια. Για παράδειγμα, η συνέχιση της αναζήτησης από μία κατάσταση μπορεί να οδηγήσει σε λύση, αλλά το κόστος υπολογισμού της να είναι τόσο μεγάλο,

ώστε να αποφασιστεί να κλαδευτεί ο χώρος αναζήτησης που συνδέεται με αυτήν την κατάσταση.

Οι αλγόριθμοι αναζήτησης χωρίζονται σε δύο μεγάλες κατηγορίες, τους τυφλούς και τους ευριστικούς. Οι τυφλοί διατάσσουν το μέτωπο της αναζήτησης βάσει της χρονικής δημιουργίας των νέων καταστάσεων. Οι ευριστικοί αλγόριθμοι όμως, διατάσσουν το μέτωπο αναζήτησης σύμφωνα με κάποια κριτήρια που αξιολογούν τις νέες καταστάσεις ως "καλύτερες" ή "χειρότερες" από κάποιες άλλες. Για παράδειγμα, ένα κριτήριο για το αν μία κατάσταση είναι "καλύτερη" από κάποια άλλη είναι η εκτιμώμενη απόσταση της από την τελική κατάσταση.

### **2.3.1. Αναζήτηση Πρώτα σε Βάθος (Depth First Search - DFS)**

Όπως φανερώνει η ονομασία του αλγορίθμου πρώτα σε βάθος (Depth-First Search- DFS), η αναζήτηση επιλέγει προς επέκταση την κατάσταση που βρίσκεται πιο βαθιά στο δένδρο. Στην περίπτωση που υπάρχουν περισσότερες από μία καταστάσεις στο ίδιο βάθος ο DFS επιλέγει τυχαία μία από αυτές και για ευκολία θεωρείται ότι επιλέγεται η αριστερότερη. Ο αλγόριθμος DFS περιγράφεται ως εξής:

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
3. Βγάλε την πρώτη κατάσταση από το μέτωπο της αναζήτησης.
4. Αν είναι η κατάσταση μέλος του κλειστού συνόλου τότε πήγαινε στο δεύτερο βήμα.
5. Αν η κατάσταση είναι μία από τις τελικές, τότε ανέφερε τη λύση.
6. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
7. Εφάρμοσε τους τελεστές μετάβασης για να βρεις τις καταστάσεις-παιδιά.
8. Βάλε τις καταστάσεις-παιδιά στην αρχή του μετώπου της αναζήτησης.
9. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
10. Πήγαινε στο βήμα 2.

Το μέτωπο της αναζήτησης είναι μια δομή στοίβας (Stack LIFO - Last In First Out), δηλαδή οι νέες καταστάσεις τοποθετούνται πάντα στην αρχή της στοίβας και η

αναζήτηση συνεχίζεται με μία από αυτές. Το βασικό μειονέκτημα του αλγορίθμου αναζήτησης πρώτα σε βάθος (Depth - First Search - DFS) είναι ότι δεν εγγυάται ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη (μονοπάτι με το μικρότερο μήκος ή με μικρότερο κόστος). Επίσης, αν δεν υπάρχει έλεγχος βρόχων ή αν ο χώρος αναζήτησης είναι μη πεπερασμένος, ο αλγόριθμος DFS μπορεί να μπλεχτεί σε κλαδιά μεγάλου μήκους ή ατέρμονα κλαδιά του δένδρου (κλαδιά με άπειρο μήκος). Ο αλγόριθμος αυτός μπορεί να μη βρει ποτέ μια τελική κατάσταση αν και μπορεί να περάσει από πολύ κοντά της. Συνεπώς, ο DFS εν γένει θεωρείται ατελής. Στις περιπτώσεις όμως που ο χώρος αναζήτησης είναι πεπερασμένος και χρησιμοποιείται κλειστό σύνολο, ο αλγόριθμος αναζήτησης πρώτα σε βάθος θα βρει λύση, εάν μια τέτοια υπάρχει.

### 2.3.2. Αναζήτηση Πρώτα σε Πλάτος (*Breadth First Search - BFS*)

Ο αλγόριθμος αναζήτησης πρώτα σε πλάτος (Breadth First Search - BFS) εξετάζει πρώτα τις καταστάσεις που βρίσκονται στο ίδιο βάθος και μόνον όταν τις εξετάσει όλες συνεχίζει στην επέκταση καταστάσεων στο αμέσως επόμενο επίπεδο. Η αναζήτηση μοιάζει να προχωρά κατά κύματα, όπου το πρώτο κύμα είναι οι καταστάσεις του πρώτου επιπέδου του δένδρου, το δεύτερο κύμα οι καταστάσεις του δεύτερου επιπέδου, κ.ο.κ.

1. Περιγραφικά ο αλγόριθμος είναι:
2. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
3. Αν το μέτωπο της αναζήτησης είναι κενό τότε σταμάτησε.
4. Βγάλε την πρώτη κατάσταση από το μέτωπο της αναζήτησης.
5. Αν είναι η κατάσταση μέλος του κλειστού συνόλου τότε πήγαινε στο βήμα 2.
6. Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση.
7. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
8. Εφάρμοσε τους τελεστές μεταφοράς για να βρεις τις καταστάσεις-παιδιά.
9. Βάλε τις καταστάσεις-παιδιά στο τέλος του μετώπου της αναζήτησης.
10. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
11. Πήγαινε στο βήμα 2.

Η διαφορά του αλγορίθμου BFS (Breadth First Search) από τον αλγόριθμο DFS (Depth First Search) εντοπίζεται στην περιγραφή του καθενός. Και συγκεκριμένα σε μία μόνο λέξη, "τέλος" αντί "αρχή". Εδώ το μέτωπο της αναζήτησης είναι μια δομή ουράς (Queue FIFO, δηλαδή First In First Out) και όχι στοίβας, ποτέ δεν επεκτείνεται μία κατάσταση αν δεν επεκταθούν πρώτα όλες οι καταστάσεις που βρίσκονται σε μικρότερο βάθος, γιατί απλά οι τελευταίες μπήκαν στο μέτωπο της αναζήτησης νωρίτερα.

Βασικό πλεονέκτημα του αλγορίθμου αναζήτησης πρώτα σε πλάτος (BFS - Breadth First Search) είναι ότι βρίσκει πάντα την καλύτερη λύση (μικρότερη σε μήκος). Είναι πλήρης, δηλαδή θα βρει λύση σε κάποιο πρόβλημα, αν τέτοια υπάρχει. Σε μία ακραία περίπτωση, όταν δηλαδή ένα δένδρο αναζήτησης έχει άπειρο πλάτος, (υπάρχουν άπειροι τελεστές που εφαρμόζονται σε μία κατάσταση), τότε ο BFS έχει πρόβλημα στην ανάπτυξη του δένδρου αναζήτησης. Τέτοιου είδους προβλήματα όμως είναι σπάνια και για αυτό ο BFS (Breadth First Search) θεωρείται δίκαια πλήρης.

#### **4.3.3 Αλγόριθμος Επαναληπτικής Εκβάθυνσης (*Iterative Deepening - ID*)**

Ο αλγόριθμος επαναληπτικής εκβάθυνσης (Iterative Deepening - ID) συνδυάζει με τον καλύτερο τρόπο τους DFS (Depth First Search) και BFS (Breadth First Search). Ο ID είναι κατά βάση DFS, αλλά η αναζήτηση γίνεται σε στάδια. Κάθε στάδιο είναι η εφαρμογή του DFS σε ορισμένο μόνο βάθος, ανεξάρτητα από το συνολικό βάθος του δένδρου αναζήτησης. Όταν η αναζήτηση στο προκαθορισμένο βάθος ολοκληρωθεί, ο DFS επαναλαμβάνεται με την ίδια αρχική κατάσταση, αλλά σε μεγαλύτερο βάθος. Αναλυτικότερα η περιγραφή του ID είναι:

1. Όρισε το αρχικό βάθος αναζήτησης (συνήθως 1).
2. Εφάρμοσε τον αλγόριθμο DFS μέχρι αυτό το βάθος αναζήτησης.
3. Αν έχεις βρει λύση σταμάτησε.
4. Αύξησε το βάθος αναζήτησης (συνήθως κατά 1).
5. Πήγαινε στο βήμα 2.

Σοβαρό μειονέκτημα του Αλγορίθμου Επαναληπτικής Εκβάθυνσης (ID) είναι ότι όταν αρχίζει ο αλγόριθμος αναζήτησης πρώτα σε βάθος (DFS) με διαφορετικό βάθος δε θυμάται τίποτα από την προηγούμενη αναζήτηση. Αυτό όμως αντισταθμίζεται



από τα πλεονεκτήματα του ID, δηλαδή ότι δεν κινδυνεύει να χαθεί σε κάποιο κλαδί με άπειρο μήκος, αφού το βάθος αναζήτησης είναι προκαθορισμένο. Άρα ο ID είναι πλήρης. Επίσης, αν το βάθος αυξάνεται κατά 1 σε κάθε κύκλο και ο ID βρει λύση, τότε αυτή η λύση θα είναι η καλύτερη. Και αυτό γιατί αν υπήρχε άλλη καλύτερη λύση αυτή θα βρισκόταν σε προηγούμενο κύκλο αναζήτησης.

### 2.3.3. Αναζήτηση Διπλής Κατεύθυνσης (*Bidirectional Search - BiS*)

Η ιδέα της αναζήτησης διπλής κατεύθυνσης (*Bidirectional Search - BiS*) πηγάζει από τη δυνατότητα του παραλληλισμού (*parallelism*) στα υπολογιστικά συστήματα. Αν υπάρχουν για παράδειγμα δύο επεξεργαστές, ο ένας μπορεί να αρχίσει να αναζητά τη λύση από την αρχική προς μία τελική κατάσταση και ο άλλος από μία τελική προς την αρχική, μοιράζοντας τη δουλειά και (θεωρητικά) μειώνοντας το χρόνο αναζήτησης [1]. Ο BiS έχει δύο προϋποθέσεις κάτω από τις οποίες μπορεί να εφαρμοστεί:

- οι τελεστές μετάβασης σε ένα πρόβλημα είναι αντιστρέψιμοι (*reversible*), δηλαδή αν για κάθε τελεστή που εφαρμόζεται σε μια κατάσταση *S1* και προκύπτει η κατάσταση *S2*, υπάρχει ο αντίστροφος τελεστής που εφαρμόζεται στην *S2* και προκύπτει η *S1*, και
- είναι πλήρως γνωστή η τελική κατάσταση (δηλαδή δεν είναι γνωστά μόνον μερικά χαρακτηριστικά της).

Στα μειονεκτήματα του BiS (*Bidirectional Search*) συγκαταλέγεται το γεγονός ότι υπάρχει επιπλέον κόστος που οφείλεται στην επικοινωνία μεταξύ των δύο αναζητήσεων. Η αναζήτηση προχωρά και από τις δύο πλευρές ταυτόχρονα με την ελπίδα ότι κάποια από τις καταστάσεις που θα επεκταθεί θα είναι κοινή και στις δύο πλευρές. Αυτό προϋποθέτει ότι ο αλγόριθμος πρέπει να θυμάται ποιες καταστάσεις έχει επισκεφθεί και από τις δύο πλευρές έτσι ώστε να αναγνωρίσει την ύπαρξη κοινής κατάστασης.

Αν βρεθεί κοινή κατάσταση και από τις δύο αναζητήσεις, η λύση είναι η ένωση των μονοπατιών από την κοινή κατάσταση έως την αρχική και έως την τελική κατάσταση. Δεν υπάρχει περιορισμός όσον αφορά τον αλγόριθμο με τον οποίο γίνονται οι

αναζητήσεις, μάλιστα μπορεί να είναι και διαφορετικοί για κάθε νήμα της αναζήτησης (thread). Πρέπει να σημειωθεί, ωστόσο, ότι ο BiS δε μειώνει την πολυπλοκότητα, αλλά κληρονομεί την πολυπλοκότητα των αλγορίθμων τους οποίους χρησιμοποιεί.

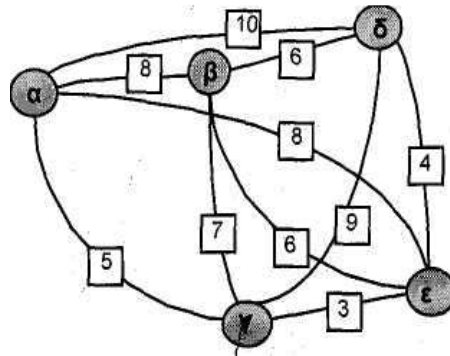
#### 4.3.5 Επέκταση και Οριοθέτηση (*Branch and Bound – B&B*)

Στις προηγούμενες ενότητες, καλύτερες λύσεις θεωρούνται όπως διαπιστώνεται, αυτές που περιείχαν λιγότερους σε αριθμό τελεστές. Αυτό υπονοεί ότι όλοι οι τελεστές μετάβασης έχουν το ίδιο κόστος, κάτι το οποίο δε συμβαίνει σε όλα τα προβλήματα. Για παράδειγμα, το κόστος μετακίνησης από μία πόλη σε μία άλλη εξαρτάται από την, απόσταση μεταξύ των δύο πόλεων. Αν ένας αλγόριθμος αναζητάει τη συντομότερη διαδρομή από μία αρχική πόλη σε μία άλλη πόλη-προορισμό, θα πρέπει να ληφθεί υπόψη και το συνολικό κόστος μετακίνησης. Αυτό εκφράζεται από τη συνολική απόσταση μεταξύ της αρχικής πόλης και της πόλης προορισμού, δηλαδή από το άθροισμα του επιμέρους κόστους κάθε τελεστή. Φυσικά, ένας απλός αλλά αφελής τρόπος θα ήταν να εφαρμοστεί ένας αλγόριθμος που βρίσκει όλες τις λύσεις, να τις αξιολογεί και να καταλήγει σε αυτή με το μικρότερο συνολικά κόστος. Κάτι τέτοιο όμως θα απαιτούσε την εξαντλητική αναζήτηση σε όλο το χώρο αναζήτησης. Μία καλύτερη προσέγγιση στο πρόβλημα θα ήταν κατά τη διάρκεια της αναζήτησης να μην εξετάζονται καταστάσεις οι οποίες είναι σίγουρο ότι δε θα οδηγήσουν στην καλύτερη λύση. Έτσι, κλαδεύεται μεγάλο (πιθανά) τμήμα του χώρου αναζήτησης.

Ο αλγόριθμος επέκτασης και οριοθέτησης (Branch and Bound – B&B) εφαρμόζεται σε προβλήματα όπου αναζητείται η βέλτιστη λύση, δηλαδή εκείνη με το ελάχιστο κόστος όπως ο χρονοπρογραμματισμός εργασιών. Σε αυτήν την κατηγορία των προβλημάτων οι τελεστές μετάβασης δεν είναι όλοι του ίδιου κόστους. Επιπλέον το κόστος είναι μία μονότονη συνάρτηση, δηλαδή αυξάνει συνεχώς καθώς εξελίσσεται η αναζήτηση. Αυτό οφείλεται στο ότι όλοι οι τελεστές έχουν πάντα θετικό κόστος.

Η λειτουργία του B&B επιδεικνύεται μέσω ενός προβλήματος. Το πρόβλημα παριστάνεται με ένα γράφο που αποτελείται από  $N$  κόμβους οι οποίοι αντιπροσωπεύουν αντίστοιχα  $N$  πόλεις. Κάποιος πρέπει να ξεκινήσει από μία πόλη, να επισκεφτεί μία φορά την κάθε πόλη και να επιστρέψει στην αρχική πόλη με το λιγότερο συνολικό κόστος, διανύοντας δηλαδή την ελάχιστη δυνατή απόσταση. Για απλότητα

όλοι οι κόμβοι ενώνονται μεταξύ τους (πλήρης γράφος) και η αρχή είναι δεδομένη, για παράδειγμα η πόλη α. Ο χώρος αναζήτησης του προβλήματος απεικονίζεται με γράφο στο παρακάτω σχήμα. Αν ο γράφος αυτός αναπτυσσόταν σε δένδρο, το δένδρο θα είχε 3 επιλογές από την αρχική κατάσταση, 3 επιλογές από κάθε κατάσταση που προκύπτει από την αρχική κ.ο.κ., δηλαδή  $3!$  διαφορετικές λύσεις με το αντίστοιχο κόστος, από τις οποίες μία θα είναι η βέλτιστη.



Πλήρης γράφος για το TSP με 5 πόλεις.

Το πρόβλημα ανήκει στην κατηγορία προβλημάτων με λύση μη-πολυωνυμικού χρόνου (NP-Complete). Η εξαντλητική εξέταση  $(N-1)!$  μονοπατιών για  $N$  πόλεις είναι ένας πολύ μεγάλος αριθμός ακόμα και για μικρό αριθμό πόλεων. Για παράδειγμα, αν  $N=20$ , τότε  $19! = 1,216351003088 \times 10^{17}$ , που σημαίνει ότι ακόμη και ο πιο γρήγορος υπολογιστής που βρίσκει 1 εκατομμύριο λύσεις το δευτερόλεπτο θα χρειαζόταν 385 χρόνια για να λύσει το πρόβλημα.

Τα προβλήματα ελαχιστοποίησης κόστους έχουν πολλές εφαρμογές. Για παράδειγμα, η αυτόματη συναρμολόγηση ψηφιακών πλακετών, όπου ένας μηχανικός βραχίονας μεταφέρει τα ολοκληρωμένα κυκλώματα και τα εναποθέτει στη σωστή τους θέση, είναι πρόβλημα ελαχιστοποίησης κόστους. Μία μη βέλτιστη διαδρομή του βραχίονα μπορεί να έχει τεράστιο χρονικό και κατά συνέπεια οικονομικό κόστος σε μία εταιρία συναρμολόγησης.

Η λειτουργία του B&B βασίζεται στο κλάδεμα καταστάσεων (pruning) και κατά συνέπεια στην ελάττωση του χώρου αναζήτησης. Όταν το πρόβλημα είναι τέτοιο ώστε να εγγυάται ότι η αναζήτηση σε ένα υποδένδρο, ως προς τη σημαντικότητα της λύσης, δε θα είναι χρήσιμη, τότε το υποδένδρο αυτό κλαδεύεται, δηλαδή η αναζήτηση δεν προχωρά σε αυτό. Για παράδειγμα, έστω ότι σε ένα πρόβλημα έχει βρεθεί μία

λύση, της οποίας το κόστος είναι 26, ενώ καθώς η αναζήτηση συνεχίζεται, σε μία τρέχουσα κατάσταση η διαδρομή κοστολογείται μέχρι στιγμής ως 27. Τότε δεν υπάρχει λόγος να γίνει η επέκταση αυτής της κατάστασης, αφού είναι σίγουρο πως η επέκταση αυτή δε θα οδηγήσει στην ανεύρεση μιας καλύτερης λύσης. Συνεπώς, ο αλγόριθμος μπορεί να κλαδέψει την κατάσταση αυτή και μαζί της φυσικά όλο το υποδένδρο που ξεκινά από αυτήν. Η περιγραφή του B&B είναι:

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αρχική τιμή της καλύτερης λύσης είναι το  $+\infty$  (όριο).
3. Αν το μέτωπο της αναζήτησης είναι κενό, τότε σταμάτησε. Η καλύτερη μέχρι τώρα λύση είναι και η βέλτιστη.
4. Βγάλε την πρώτη σε σειρά κατάσταση από το μέτωπο της αναζήτησης.
5. Αν η κατάσταση ανήκει στο κλειστό σύνολο, τότε πήγαινε στο 3.
6. Αν η κατάσταση είναι τελική, τότε ανανέωσε τη λύση ως την καλύτερη μέχρι τώρα και ανανέωσε την τιμή του ορίου με την τιμή που αντιστοιχεί στην τελική κατάσταση. Πήγαινε στο 3.
7. Εφάρμοσε τους τελεστές μεταφοράς για να παράγεις τις καταστάσεις-παιδιά και την τιμή που αντιστοιχεί σε αυτές.
8. Βάλε τις καταστάσεις-παιδιά, των οποίων η τιμή δεν υπερβαίνει το όριο, μπροστά στο μέτωπο της αναζήτησης.
9. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
10. Πήγαινε στο 3.

Υπάρχουν διάφορες παραλλαγές του B&B, ανάλογα με το ποια κατάσταση επεκτείνεται πρώτη. Μία άλλη βελτίωση που επιδέχεται ο B&B είναι να επεκταθεί αυτή η κατάσταση που κατά εκτίμηση οδηγεί στη λύση με το μικρότερο κόστος.

Τέλος, ο B&B μπορεί να συνδυαστεί με δυναμικό προγραμματισμό (dynamic programming) όπου το κλάδεμα δε γίνεται μόνο σε σύγκριση με το τρέχον όριο, δηλαδή τη βέλτιστη λύση μέχρι εκείνη τη στιγμή, αλλά γίνεται και για κάθε κατάσταση που είναι περιττή. Για παράδειγμα, έστω δύο μονοπάτια το  $αβ\epsilon 13$  και το  $α\epsilon\delta$ . Εφόσον υπάρχει διαδρομή από το  $\alpha$  στο  $\epsilon$  με κόστος 8 δεν υπάρχει νόημα να επεκταθεί το μονοπάτι  $αβ\epsilon$  με κόστος 13, που είναι μεγαλύτερο, γιατί ποτέ το  $αβ\epsilon$  δε θα καταλήξει σε βέλτιστη λύση.

Η πρώτη εφαρμογή της μεθόδου BB στην επίλυση του προβλήματος του job shop προτάθηκε από τον Balas, και στη συνέχεια ακολούθησαν πολλοί άλλοι μεταξύ των οποίων και οι Carlier, Carlier και Pinson, Appelgate και Cook, Brucker, Perregaard και Clausen, Boyd και Burlingame και Martin. Αν και η μελέτη των μεθόδων BB δείχνει ότι επιτεύχθηκαν σημαντικές βελτιώσεις στη επίλυση του προβλήματος με τη χρήση τους, αυτό θα πρέπει κυρίως να αποδοθεί στη βελτίωση της τεχνολογίας των ηλεκτρονικών υπολογιστών, παρά στις τεχνικές που χρησιμοποιήθηκαν [13].

## ΚΕΦΑΛΑΙΟ 3 – ΕΥΡΕΤΙΚΟΙ ΚΑΙ ΜΕΤΑΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

### 3.1. Ευρετικοί αλγόριθμοι

Η εύρεση της καλύτερης λύσης σε ένα δεδομένο πρόβλημα είναι ένα δυσεπίλυτο πρόβλημα. Δεδομένου ότι τα προβλήματα που παρουσιάζονται στις διάφορες σύγχρονες επιστημονικές κοινότητες είναι προβλήματα συνδυαστικά και πολύπλοκα, τότε καταλαβαίνει κανείς ότι η μελέτη όλων των πιθανών λύσεων με σκοπό την εύρεση της καλύτερης, είναι τις περισσότερες φορές απαγορευτική αν όχι αδύνατη. Αυτή η αδυναμία ώθησε στην ανάπτυξη των λεγόμενων ευρετικών αλγορίθμων. Τα δυο κυριότερα στοιχεία ενός αλγορίθμου είναι η δυνατότητα εύρεσης της βέλτιστης λύσης και ο υπολογιστικός χρόνος που χρειάζεται για την εύρεση αυτή. Οι ευρετικοί αλγόριθμοι προσανατολίζονται στην εύρεση μιας λύσης σε κάποιο υπολογιστικό χρόνο θυσιάζοντας πολλές φορές τον έναν από τους δυο παράγοντες. Βρίσκουν έτσι, αρκετά καλές λύσεις στα προβλήματα στα οποία εφαρμόζονται αλλά χωρίς απόδειξη ότι η λύσεις αυτές είναι πραγματικά καλές αλλά και δεν υπάρχει εγγύηση ότι θα δοθεί λύση σε σύντομο υπολογιστικό χρόνο. Η έρευνα έχει δείξει ότι ο υπολογιστικός χώρος αναζήτησης λύσεων για ρεαλιστικά προβλήματα είναι πολύ μεγάλος, ώστε να παρθούν λύσεις σε μικρούς σχετικά υπολογιστικούς χρόνους.

Οι ευρετικοί αλγόριθμοι εκμεταλλεύονται μια αρχική προσέγγιση ενός τυχαίου αριθμού του πληθυσμού και τυχαίας επιλογής του αρχικού χώρου λύσεων ώστε να φτάσουν με ευρετικούς μηχανισμούς σε μια καλύτερη περιοχή λύσεων μέσω του ανασυνδυασμού και της μετάλλαξης. Αυτό επιτυγχάνεται με την αξιολόγηση των ενδιάμεσων καταστάσεων ως προς την εκτιμώμενη απόστασή τους από μια τελικά κατάσταση και επεκτείνουν τις λύσεις που βρίσκονται κοντά σε αυτή, ενώ ταυτόχρονα μειώνουν την ισχύ αυτών που απέχουν πολύ ή και τις διαγράφουν εντελώς από το δέντρο των λύσεων. Στην ουσία καταγωγή των αλγορίθμων είναι η θεωρία της μαθηματικής βελτιστοποίησης η οποία εντοπίζει το ελάχιστο ή και το μέγιστο μιας συνάρτησης διακριτής μεταβλητής. Το ρόλο της πραγματικής συνάρτησης στους αλγορίθμους αυτούς παίζει η (μετα) ευρετική συνάρτηση και ο χώρος των λύσεων

παρουσιάζεται σαν ένας χώρος στον οποίο οι λύσεις μεταξύ τους διαφέρουν με βάση τη μεταξύ τους απόσταση, ενώ όσο καλύτερη ευρετική τιμή έχει μια λύση, τόσο υψηλότερα σε σχέση με τις άλλες τοποθετείται σε αυτό το χώρο του συνόλου των λύσεων. Μπορούμε να παρουσιάσουμε αυτό το χώρο των λύσεων σαν ένα γράφο με κορυφές τις καταστάσεις και ακμές τους τελεστές μετάβασης.

Σε μεγάλους χώρους αναζήτησης μπορούμε να πετύχουμε πολύ καλύτερη απόδοση αν εκμεταλλευτούμε τη γνώση που έχουμε σχετικά με το συγκεκριμένο πρόβλημα που προσπαθούμε να επιλύσουμε (domain specific knowledge). Οι ευρετικοί μηχανισμοί (heuristics) είναι κανόνες βασισμένοι σε τέτοιου είδους γνώση σύμφωνα με τους οποίους επιλέγουμε τον επόμενο κόμβο για επέκταση κατά τη διάρκεια της αναζήτησης. Ένα βασικό συστατικό αυτών των αλγορίθμων είναι μια ευρετική συνάρτηση (heuristic function)  $h$ , τέτοια ώστε:

- $h(n)$  είναι το εκτιμώμενο κόστος του συντομότερου μονοπατιού από την κατάσταση στον κόμβο  $n$  ως μια κατάσταση στόχου
- Η  $h$  μπορεί να είναι οποιαδήποτε συνάρτηση τέτοια ώστε  $h(n) = 0$  αν το  $n$  αντιστοιχεί στην κατάσταση-στόχο

Για να βρούμε όμως μια καλή ευρετική συνάρτηση χρειαζόμαστε πληροφορία σχετικά με το συγκεκριμένο πρόβλημα. Ένα παράδειγμα στο οποίο θα μπορούσε να εφαρμοστεί ένας τέτοιος αλγόριθμος είναι η συσκευασία στοιχείων που δεν έχουν κανονισμένο και το ίδιο σχήμα σε έναν περιορισμένο χώρο. Η εύρεση της καλύτερης λύσης είναι ένα πολύ δύσκολο πρόβλημα με αποτέλεσμα, αν κάποιος θέλει να βρει την τέλεια λύση να πρέπει να δοκιμάσει όλους τους πιθανούς συνδυασμούς. Αυτό για μικρό αριθμό αντικειμένων ίσως να φαντάζει απλό αλλά αν ο αριθμός αυτός αυξηθεί έστω και λίγο τότε το πρόβλημα που προκύπτει είναι δυσεπίλυτο. Η λογική που μπορεί να χρησιμοποιηθεί σε αυτήν την περίπτωση είναι να τοποθετηθούν πρώτα τα μεγαλύτερα αντικείμενα και προχωρώντας προς τα μικρότερα να μπορέσουμε να τα τοποθετήσουμε ανάλογα με το χώρο που διαθέτουμε. Αυτός είναι και ο τρόπος που θα λειτουργούσε μια ευρετική συνάρτηση σε αυτήν την περίπτωση με τη διαφορά ότι η λύση που θα παίρναμε θα ήταν πιο άμεση.

Οι ευρετικές μέθοδοι είναι βασισμένες σε ευφυείς στρατηγικές αναζήτησης που χρησιμοποιούνται για την επίλυση υπολογιστικών προβλημάτων χρησιμοποιώντας διάφορες εναλλακτικές προσεγγίσεις. Υπάρχουν στην πραγματικότητα και περιπτώσεις

στις οποίες ο ευρετικός αλγόριθμος τελικά θα επιστρέψει χαμηλής ποιότητας αποτελέσματα ή θα τα επιστρέψει με μεγάλη καθυστέρηση, παρόλα αυτά γίνεται αυτές οι δυσλειτουργίες να διορθωθούν για κάθε περίπτωση προβλήματος και ο αλγόριθμος τελικά να γίνει αποδοτικός. Η χρήση των ευρετικών αλγορίθμων είναι πλέον ευρέως διαδεδομένη και στην πραγματικότητα σε πολλές από τις περιπτώσεις προβλημάτων είναι και ο μόνος τρόπος προσέγγισης λύσεων σε ένα λογικό χρονικό διάστημα.

### **3.2. Μεταευρετικοί αλγόριθμοι**

Οι μεταευρετικοί αλγόριθμοι είναι μια υποκατηγορία και παράλληλα μια εξέλιξη των ευρετικών αλγορίθμων και οι οποίοι δε βελτιώνουν μια λύση απευθείας, αλλά διαχειρίζονται την επιλογή ευρετικών υπο-μεθόδων για τη βελτίωση της τρέχουσας λύσης –«μια ευρετική που επιλέγει ευρετικές». Αυτές οι λεγόμενες «χαμηλού επιπέδου» ευρετικές αναλαμβάνουν την αναζήτηση στον χώρο των λύσεων.

Ο όρος μεταευρετικός αποτελεί σύνθεση των λέξεων «μετά» και «ευρίσκω» και χρησιμοποιείται έχοντας την έννοια της αναζήτησης σε υψηλότερο επίπεδο. Οι μεταευρετικές τεχνικές στηρίζονται στην παραδοχή ότι ενδεχομένως δεν θα εντοπίσουν την βέλτιστη λύση αλλά στην πράξη έχει αποδειχθεί ότι συχνά αποτελούν τον πλέον ενδεδειγμένο τρόπο για την αντιμετώπιση δύσκολων προβλημάτων συνδυαστικής βελτιστοποίησης. Οι Osman και Laporte περιγράφουν τις μεταευρετικές τεχνικές ως εξής: “Μια μεταευρετική τεχνική ορίζεται ως μια επαναληπτική διαδικασία δημιουργίας λύσεων, η οποία καθοδηγεί μια δευτερεύουσα ευρετική τεχνική συνδυάζοντας με ευφυή τρόπο διαφορετικούς τρόπους εξερεύνησης και ανάλυσης του χώρου αναζήτησης, ενώ στρατηγικές μάθησης χρησιμοποιούνται προκειμένου να εντοπίσουν με αποδοτικό τρόπο λύσεις οι οποίες βρίσκονται κοντά στις βέλτιστες.”

Οι μεταευρετικές τεχνικές φαίνεται να έχουν ισχυρή δυναμική για την περαιτέρω εξάπλωση και χρήση τους. Σε αυτό το συμπέρασμα συνηγορεί το γεγονός ότι εκτός από τις παραδοσιακές μεταευρετικές τεχνικές (Simulated Annealing, Tabu Search, κ.α.) τα τελευταία χρόνια έχουν αναπτυχθεί νέες τεχνικές (Variable Neighborhood Search, Guided Local Search, κ.α.). Οι μεταευρετικές τεχνικές μπορούν να λειτουργήσουν είτε ανεξάρτητα είτε σε υβριδικούς συνδυασμούς είτε οργανωμένες σε διάφορα επίπεδα αφαίρεσης (Hyperheuristics). Θα πρέπει να τονιστεί ότι για



ορισμένες τεχνικές τα όρια ανάμεσα στην τεχνητή νοημοσύνη, την υπολογιστική νοημοσύνη και τις μεταερευτικές τεχνικές δεν είναι σαφή. Για παράδειγμα οι γενετικοί αλγόριθμοι αναφέρονται στην βιβλιογραφία ως παράδειγμα τεχνικής και στις τρεις προαναφερθείσες κατηγορίες.

### **3.3. Παραδείγματα για τους ευρετικούς αλγορίθμους**

#### **3.3.1. Αλγόριθμοι κατασκευής**

Οι αλγόριθμοι αυτοί επιχειρούν να δώσουν λύσεις στο υπό εξέταση πρόβλημα ξεκινώντας από μια κενή αρχική λύση και λειτουργώντας επαυξητικά και επαναληπτικά προσθέτει καίρια καθορισμένα τμήματα της λύσης χωρίς να επιστρέφει σε προηγούμενα βήματα της διαδικασίας μέχρι να δώσει μια ολοκληρωμένη λύση. Στην πιο απλή μορφή του αλγόριθμου οι λύσεις προστίθενται με τυχαία σειρά. Οι «άπληστοι» (greedy) ευρετικοί αλγόριθμοι κατασκευής προσθέτουν σε κάθε βήμα ένα κομμάτι λύσης, και έτσι επιτυγχάνεται το μέγιστο «μυωπικό» όφελος όπως αυτό υπολογίζεται από κάποια συγκεκριμένη ευρετική πληροφορία.

#### **3.3.2. Lagrangian Relaxation**

Η μέθοδος προσεγγίζει ένα δύσκολο πρόβλημα της βελτιστοποίησης υπό περιορισμούς μέσω ενός απλούστερου προβλήματος. Η μέθοδος Lagrangian relaxation (LR) είναι μια μαθηματική προγραμματιστική τεχνική για βελτιστοποίηση προβλημάτων που περιλαμβάνουν περιορισμούς (constrained optimization). Όμοια με τον μηχανισμό των τιμών σε μια αγορά, η μέθοδος αντικαθιστά τους “σκληρούς” περιορισμούς, όπως περιορισμούς στην διαθεσιμότητα των μηχανών, με την πληρωμή ορισμένων τιμών, δηλαδή των πολλαπλασιαστών Lagrange, βασιζόμενη στην “ζήτηση” για τη χρήση μιας μηχανής σε κάθε μονάδα χρόνου. Το αρχικό πρόβλημα μπορεί με αυτό τον τρόπο να αναλυθεί σε πολλά μικρότερα και ευκολότερα στην επίλυση υποπροβλήματα.

Πιο συγκεκριμένα, η μέθοδος αυτή είναι καλή για προβλήματα στα οποία μπορούμε να διαχωρίσουμε τους περιορισμούς σε δυο κατηγορίες: στους “καλούς” και “κακούς”. Με την πρώτη κατηγορία το πρόβλημα λύνεται εύκολα ενώ με την προσθήκη των δεύτερων το πρόβλημα γίνεται πολύ περίπλοκο στη λύση του. Αυτό που γίνεται στη συνέχεια είναι ότι απομακρύνονται οι «κακοί» περιορισμοί και τοποθετούνται στην αντικειμενική συνάρτηση με αντίστοιχα βάρη τα οποία αντιπροσωπεύουν μια ποινή που λέγεται πολλαπλασιαστής Lagrange και που προστίθεται σε μια λύση που δεν ικανοποιεί το συγκεκριμένο περιορισμό.

### 3.3.3. Γενετικοί Αλγόριθμοι

Οι γενετικοί αλγόριθμοι αναφέρονται σε όλες τις διαδικασίες αναζήτησης λύσεων οι οποίες προσομοιώνουν τη φυσική εξελικτική διαδικασία επιλογής. Αναφέρονται λοιπόν σε συγκεκριμένα χαρακτηριστικά και μεθόδους εξέλιξης πληθυσμών ανάλογα με αυτούς που συναντάμε σε πληθυσμούς (συνήθως εντόμων) στη φύση. Οι γενετικοί αλγόριθμοι εφαρμόζονται σε πεπερασμένους πληθυσμούς υποψήφιων λύσεων και ιδιαίτερα, όταν εφαρμόζονται στον χρονοπρογραμματισμό, θεωρούν τις ακολουθίες ή τους διάφορους χρονοπρογραμματισμούς ως άτομα (individuals) ή μέλη ενός πληθυσμού (population). Κάθε άτομο του πληθυσμού έχει κάποια χαρακτηριστικά όπως είναι η υγεία του (fitness). Η υγεία ενός ατόμου του πληθυσμού μετράται από την αντικειμενική συνάρτηση. Η διαδικασία δουλεύει επαναληπτικά και κάθε επανάληψη είναι μια γενιά (generation).

Όσον αφορά στις λύσεις που αντιμετωπίζονται από τον αλγόριθμο, σε κάθε επανάληψη του αλγορίθμου οι χειρότερες λύσεις μετακινούνται από τον πληθυσμό και αντικαθίστανται από νέες υποψήφιες λύσεις που παράγονται με: (1) εφαρμογή μεταλλάξεων (mutations) σε ορισμένες λύσεις του πληθυσμού, (2) εφαρμογή διασταυρώσεων (crossover) σε ζευγάρια λύσεων του πληθυσμού. Η ίδια διαδικασία συνεχίζεται για έναν προκαθορισμένο αριθμό επαναλήψεων και ο πιο επιτυχημένος υποψήφιος επιλέγεται ως η καλύτερη λύση. Όπως η εξελικτική διαδικασία στη φύση, ο πληθυσμός συνήθως παραμένει σταθερός, χάρη στον αποκλεισμό των πιο αδύναμων λύσεων που παράγονται (weaker child solutions), στην αναπαραγωγή ή τη μετάλλαξη

των μελών του πληθυσμού της προηγούμενης γενιάς και στο γενικό κανόνα της επιβίωσης του υγιέστερου (survival of the fittest).

### **3.3.4. Αλγόριθμος Tabu Search**

Στην αναζήτηση tabu ο αλγόριθμος αναζήτησης μπορεί να μετακινηθεί από μία τρέχουσα υποψήφια λύση σε οποιαδήποτε «γειτονική»-καλύτερη ή χειρότερη-στηριζόμενος μόνο στο ότι η μετάβαση στην επόμενη λύση δεν περιέχεται στην λίστα-tabu (tabu list). Σε οποιοδήποτε στάδιο της διαδικασίας κρατείται μια tabu list των μεταβολών (mutations) που η διεργασία δεν επιτρέπεται να επανεκτελέσει. Οι μεταβολές αυτές μπορεί να είναι για παράδειγμα ζεύγη διεργασιών που δεν μπορούν να υποστούν αμοιβαία ανταλλαγή. Οι πληροφορίες που μεταφέρονται κατά τη διάρκεια της εκτέλεσης της μεθόδου tabu search, συνίστανται από την tabu list καθώς και τη βέλτιστη λύση που λαμβάνεται μέσω της διαδικασίας αναζήτησης.

Όταν μία μετατροπή εφαρμόζεται σε μία τρέχουσα υποψήφια λύση αυτή η μετατροπή προστίθεται στην κεφαλή της λίστας και στη συνέχεια η ουρά της λίστας απομακρύνεται. Με αυτό τον τρόπο ο αλγόριθμος οδηγεί τη διαδικασία αναζήτησης μακριά από λύσεις που (με βάση τις διαθέσιμες πληροφορίες) εμφανίζονται να επαναλαμβάνονται ή να μοιάζουν σε λύσεις που έχουν βρεθεί σε προηγούμενα στάδια.

### **3.3.5. Αλγόριθμος Simulated Annealing**

Η μέθοδος simulated annealing είναι μια μέθοδος αναζήτησης που έχει την καταγωγή της στα πεδία της φυσικής και της επιστήμης των υλικών. Αναπτύχθηκε πρώτα σαν μοντέλο προσομοίωσης για την περιγραφή της σκλήρυνσης με πυράκτωσης συμπτυκνωμένου μετάλλου. Η τεχνική αυτή είναι μία παραλλαγή της τοπικής αναζήτησης. Στη μέθοδο simulated annealing επιτρέπονται μετακινήσεις προς χειρότερες λύσεις (ανηφορικές-uphill μετακινήσεις). Ο λόγος που επιτρέπονται τέτοιες μετακινήσεις είναι ότι δίνεται στη διαδικασία η δυνατότητα να μετακινηθεί μακριά από ένα τοπικό ελάχιστο και να βρει μια καλύτερη λύση αργότερα. Αν η τιμή της αντικειμενικής συνάρτησης της γειτονικής λύσης είναι καλύτερη από αυτή της

τρέχουσας υποψήφιας λύσης τότε επιλέγεται η γειτονική. Αν δεν είναι, τότε επιλέγεται και πάλι με τη διαφορά ότι στην επόμενη επανάληψη μειώνεται η πιθανότητα να επιλεγεί η επόμενη χειρότερη γειτονική λύση. Με αυτό τον τρόπο, αν κατά τη διάρκεια της αναζήτησης βρίσκουμε συνεχώς χειρότερες γειτονικές λύσεις τότε κάποια στιγμή η πιθανότητα να επιλεγεί μία επόμενη χειρότερη θα είναι μηδέν και ο αλγόριθμος θα σταματήσει εκεί και μάλιστα σε σύντομο χρονικό διάστημα έχοντας βρει ένα τοπικό βέλτιστο. Διαφορετικά βρίσκοντας καλύτερες λύσεις συνεχώς, φθάνει στο ολικό μέγιστο.

### **3.3.6. Νοημοσύνη των σμηνών**

Η Νοημοσύνη των σμηνών (Swarm Intelligence-SI) είναι μια τεχνική τεχνητής νοημοσύνης που βασίζεται γύρω από τη μελέτη της συμπεριφοράς συστημάτων συνόλων πληθυσμών. Η έκφραση "νοημοσύνη σμηνών" εισήχθη από τους Beni & WANG το 1989, για τη μελέτη των κυψελοειδών ρομποτικών συστημάτων. Ο όρος νοημοσύνη των σμηνών αναφέρεται σε ένα κλάδο της τεχνητής ζωής, ο οποίος στηρίζεται στη μελέτη της συλλογικής συμπεριφοράς σε μη κατανεμημένα, αυτό-οργανωμένα σύνολα ατόμων όπως μυρμηγκία, μέλισσες κ.α. που εκτελούν διάφορες εργασίες ζωτικής σημασίας για το σύνολο του πληθυσμού που ταυτόχρονα σαν ξεχωριστά άτομα δε θα μπορούσαν να πετύχουν. Τα συστήματα αυτά αποτελούνται από έναν πληθυσμό απλών πρακτόρων που αλληλεπιδρούν ατομικά και συλλογικά μεταξύ τους και με το περιβάλλον τους. Γενικά, δεν υπάρχει κάποια συγκεντρωτική δομή ελέγχου του τρόπου της αλληλεπίδρασης, ενώ αυτό γίνεται αντιληπτό από τη συλλογική συμπεριφορά. Παραδείγματα τέτοιων συστημάτων μπορούν να βρεθούν στη φύση είναι οι αποικίες των μυρμηγκιών, τα σμήνη των πουλιών, οι αποικίες βακτηριδίων και τα κοπάδια των ψαριών.

Από τις πιο επιτυχημένες τέτοιες τεχνικές που έχουν αναπτυχθεί μέχρι τώρα είναι η Βελτιστοποίηση με τη μέθοδο των αποικιών μυρμηγκιών (Ant Colony Optimization-ACO) και η Βελτιστοποίηση με τη μέθοδο σμηνών μορίων (Particle swarm optimization-PSO). Οι ACO αλγόριθμοι είναι μεταευρετικοί αλγόριθμοι που δίνουν κατά προσέγγιση λύσεις σε δύσκολα συνδυαστικά προβλήματα βελτιστοποίησης. Οι αλγόριθμοι PSO αποτελούν τεχνικές ελαχιστοποίησης για τα

προβλήματα στα οποία μια καλύτερη λύση μπορεί να αντιπροσωπευθεί ως σημείο ή επιφάνεια σε ένα n-διάστατο σύστημα. Μια ακόμα τεχνική που αξίζει να δούμε και ανήκει στις τεχνικές των σμηνών είναι και τεχνική αποικίας τεχνητών μελισσών (Artificial Bee Colony -ABC). Η πρώτη φορά που μελετήθηκε μια τέτοια τεχνική ήταν από τον Karaboga το 2005. Σε αυτήν περίπτωση γίνεται προσομοίωση της διαδικασίας εύρεσης τροφής των μελισσών. Οι αλγόριθμοι που παρουσιάζουν αυτήν την τεχνική έχουν τρεις φάσεις: της μέλισσας εργάτριας, της μέλισσας ανιχνευτή και της μέλισσας επιστάτη.

### **3.4. Παραδείγματα μεταερευνητικών αλγορίθμων**

#### **3.4.1. Βελτιστοποίηση με Σμήνος Σωματιδίων**

Ο αλγόριθμος Βελτιστοποίηση με Σμήνος Σωματιδίων (Particle Swarm Optimization PSO) είναι ένας στοχαστικός μεταερευνητικός αλγόριθμος βασισμένος στην έννοια του πληθυσμού. Προτάθηκε από τους Eberhart και Kennedy (1995) ως μία εναλλακτική λύση στους κυρίαρχους μέχρι τότε Εξελικτικούς Αλγορίθμους (Evolutionary Algorithms - EA) για την επίλυση προβλημάτων αριθμητικής βελτιστοποίησης. Από την δημιουργία του, ο αλγόριθμος PSO έχει αποκτήσει αυξανόμενο ερευνητικό ενδιαφέρον. Αυτό οφείλεται στην επιβεβαιωμένη αποδοτικότητά του σε πληθώρα δύσκολων προβλημάτων βελτιστοποίησης, καθώς και στην εύκολη υλοποίησή του. Σήμερα, ο αλγόριθμος PSO κατατάσσεται ανάμεσα στους κορυφαίους μεταερευνητικούς αλγορίθμους βελτιστοποίησης, μετρώντας ένα μεγάλο πλήθος εφαρμογών σε διάφορα επιστημονικά πεδία, εκτενή βιβλιογραφία, καθώς και δημοσιευμένο ερευνητικό λογισμικό. Ο κύριος μηχανισμός του αλγορίθμου PSO αποτελείται από μία ομάδα συνεργαζόμενων πρακτόρων αναζήτησης, οι οποίοι επαναληπτικά εξετάζουν τον χώρο αναζήτησης. Πιο συγκεκριμένα, θεωρούμε το n-διάστατο πρόβλημα ολικής ελαχιστοποίησης:

$$\min_{x \in \mathcal{X} \subset \mathbb{R}^n} C(x).$$

Ένα σμήνος (swarm) είναι ένα σύνολο από σημεία αναζήτησης και ορίζεται ως:

$$S = \{x_1, x_2, \dots, x_N\}, \quad x_i \in X, \quad i \in I = \{1, 2, \dots, N\}.$$

Κάθε σημείο αναζήτησης καλείται σωματίδιο (particle) και είναι ένα -διάστατο διάνυσμα που ορίζεται ως:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{in})^T \in X, \quad i \in I.$$

Κάθε σημείο αναζήτησης μπορεί να κινηθεί μέσα στον χώρο αναζήτησης. Για τον σκοπό αυτό χρησιμοποιείται μια προσαρμοζόμενη μετατόπιση θέσης, η οποία καλείται ταχύτητα (velocity) του σωματιδίου και ορίζεται ως ακολούθως:

$$v_i = (v_{i1}, v_{i2}, \dots, v_{in})^T, \quad i \in I.$$

Επιπλέον, κάθε σωματίδιο διατηρεί στην μνήμη την καλύτερη θέση (best position) που βρήκε στον χώρο αναζήτησης, δηλαδή αυτή με την μικρότερη συναρτησιακή τιμή:

$$p_i = (p_{i1}, p_{i2}, \dots, p_{in})^T, \quad i \in I.$$

Αν συμβολίσουμε τον μετρητή επαναλήψεων του αλγορίθμου, τότε ισχύει ότι:

$$p_i(t) = x_i(\tau), \quad \tau = \arg \min_{\kappa \in \{0, 1, \dots, t\}} \{C(x_i(\kappa))\}.$$

Οι καλύτερες θέσεις αποτελούν ένα είδος εμπειρίας (experience) για τα σωματίδια. Διαμοιράζοντας την εμπειρία αυτή, επιτυγχάνεται συνεργασία ανάμεσα στα σωματίδια κατευθύνοντας την αναζήτησή τους προς τις πιο ελπιδοφόρες περιοχές του χώρου αναζήτησης. Ο διαμοιρασμός της πληροφορίας βασίζεται στην έννοια της γειτονιάς (neighborhood). Η γειτονιά του -οστού σωματιδίου ορίζεται από το σύνολο δεικτών των σωματιδίων με τα οποία ανταλλάσσει πληροφορίες:

$$NB_{i,s} = \{j_1, j_2, \dots, j_3\} \subseteq I, \quad i \in NB_{i,s}.$$

Η καλύτερη θέση της γειτονιάς, όπου:

$$g_i = \arg \min_{j \in NB_{i,s}} \{C(p_j)\},$$

χρησιμοποιείται μαζί με το  $p_i$  για να ανανεωθεί το  $i$ -οστό σωματίδιο σε κάθε επανάληψη. Η παράμετρος  $s$  ορίζεται από τον χρήστη και καλείται μέγεθος γειτονιάς

(neighborhood size) ενώ έχει άμεσο αντίκτυπο στο εύρος ανταλλαγής πληροφορίας ανάμεσα στα σωματίδια. Ως εκ τούτου, μπορεί να επηρεάσει τις ιδιότητες εξερεύνησης (exploration)/εκμετάλλευσης (exploitation) του αλγορίθμου. Στην ειδική περίπτωση όπου  $s = N$ , ολόκληρο το σμήνος αποτελεί την γειτονιά κάθε σωματιδίου. Αυτή η περίπτωση ορίζει το λεγόμενο ολικό μοντέλο (global model) της PSO, το οποίο συμβολίζεται και ως *gbest*. Αντίθετα, αυστηρά μικρότερες γειτονιές ορίζουν το τοπικό μοντέλο (local model), το οποίο συμβολίζεται και ως *lbest*.

Τα σωματίδια που θα αποτελέσουν κάθε γειτονιά συνήθως καθορίζονται διαμέσου γενικών σχημάτων, στα οποία θεωρείται μια χωρική οργάνωση των δεικτών των σωματιδίων. Αυτά τα σχήματα ονομάζονται τοπολογίες γειτονιών (neighborhood topologies) και μπορεί να έχουν σημαντικό αντίκτυπο στην απόδοση της PSO επειδή καθορίζουν την ροή της πληροφορίας ανάμεσα στα σωματίδια. Μια ευρέως χρησιμοποιούμενη τοπολογία γειτονιών είναι ο δακτύλιος (ring), στην οποία όλοι οι δείκτες σωματιδίων θεωρούνται διατεταγμένοι σε αύξουσα σειρά, με τους δείκτες να ανακυκλώνονται μετά το  $n$ . Σύμφωνα με αυτή την τοπολογία, κάθε σωματίδιο θεωρεί ως γειτονιά τα σωματίδια με γειτονικούς σε αυτό δείκτες.

Μέχρι σήμερα, έχουν προταθεί στην βιβλιογραφία διάφορες παραλλαγές του αλγορίθμου PSO. Μία από τις πιο δημοφιλείς είναι η παραλλαγή με χρήση συντελεστή περιορισμού (constriction coefficient) που προτάθηκε από τους Clerc και Kennedy (2002). Σύμφωνα με αυτή, η κίνηση του σωματιδίου περιγράφεται από τις ακόλουθες σχέσεις:

$$v_{ij}(t+1) = \chi [v_{ij}(t) + c_1 R_1 (p_{ij}(t) - x_{ij}(t)) + c_2 R_2 (p_{g_{ij}}(t) - x_{ij}(t))],$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1),$$

Όπου  $i=1,2,\dots,n$  και  $j=1,2,\dots,n$ , η παράμετρος  $\chi$  είναι ο συντελεστής περιορισμού, τα  $c_1$  και  $c_2$  και είναι σταθερές γνωστές και ως γνωσιακή και κοινωνική σταθερά αντίστοιχα, και τα  $R_1$  και  $R_2$  είναι τυχαίες μεταβλητές ομοιόμορφα κατανομημένες στο διάστημα  $[0,1]$ . Θα πρέπει να σημειωθεί πως χρησιμοποιείται διαφορετική τιμή των  $R_1$  και  $R_2$  για κάθε  $i$  και  $j$  της παραπάνω σχέσης σε κάθε επανάληψη. Οι καλύτερες θέσεις των σωματιδίων ανανεώνονται σε κάθε επανάληψη με βάση την παρακάτω σχέση:

$$p_i(t+1) = \begin{cases} x_i(t+1), & \text{αν } C(x_i(t+1)) < C(p_i(t)), \\ p_i(t), & \text{διαφορετικά,} \end{cases} \quad i \in I.$$

Η παραπάνω μορφή της PSO προέκυψε από την ανάλυση ευστάθειας του αλγορίθμου. Από την ίδια ανάλυση προέκυψε και το σύνολο τιμών των παραμέτρων,  $\chi=0.729$ ,  $c_1=c_2=2.05$ , το οποίο θεωρείται εν γένει ικανοποιητικό, παρέχοντας στον αλγόριθμο σταθερότητα και ταχύτητα. Εναλλακτικά σύνολα παραμέτρων έχουν δοθεί στην βιβλιογραφία.

### 3.4.2. Διαφοροεξελικτικός Αλγόριθμος

Ο Διαφοροεξελικτικός Αλγόριθμος (Differential Evolution - DE) είναι ένας μεταευρετικός αλγόριθμος βασισμένος στην έννοια του πληθυσμού και έχει πολλές ομοιότητες με την PSO. Προτάθηκε το 1997 από τους Storn και Price και, ομοίως με την PSO, χρησιμοποιεί μία ομάδα από σημεία αναζήτησης που καλείται πληθυσμός (population) και ορίζεται ως:

$$S = \{x_1, x_2, \dots, x_N\},$$

για να εξερευνήσει τον χώρο αναζήτησης. Κάθε σημείο αναζήτησης καλείται άτομο (individual) και ορίζεται ως ένα n-διάστατο διάνυσμα:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{in})^T \in X, \quad i \in I$$

το οποίο ανήκει στο χώρο αναζήτησης, σε άμεση συμφωνία με τα σωματίδια της PSO. Επίσης, ο πληθυσμός αρχικοποιείται τυχαία, συνήθως με βάση την ομοιόμορφη κατανομή στο χώρο αναζήτησης. Ωστόσο, η κύρια διαδικασία ανανέωσης του DE διαφέρει από της PSO. Συγκεκριμένα, αντί της υπόθεσης πως τα άτομα κινούνται μέσα στον χώρο αναζήτησης, η ανανέωση γίνεται με χρήση τριών τελεστών που δειγματοληπτούν άτομα από τον πληθυσμό, τα συνδυάζουν παράγοντας νέες πιθανές λύσεις και επιλέγοντας τα καλύτερα ανάμεσα στα νέα και τα ήδη υπάρχοντα άτομα. Οι τελεστές αυτοί καλούνται μετάλλαξη (mutation), διασταύρωση (crossover) και επιλογή (selection), και αναλύονται στην συνέχεια.



Ο τελεστής της μετάλλαξης παράγει ένα νέο διάνυσμα,  $v_i$ , για κάθε άτομο  $x_i$ ,  $i=1,2,\dots,N$ , συνδυάζοντας κάποια από τα άτομα του πληθυσμού. Υπάρχει πλήθος τελεστών μετάλλαξης για τον DE που αναφέρονται στη βιβλιογραφία. Οι παρακάτω πέντε τελεστές είναι οι επικρατέστεροι:

$$(DE1) \quad v_i(t+1) = x_g(t) + F(x_{r_1}(t) - x_{r_2}(t)),$$

$$(DE2) \quad v_i(t+1) = x_{r_1}(t) + F(x_{r_2}(t) - x_{r_3}(t)),$$

$$(DE3) \quad v_i(t+1) = x_i(t) + F(x_g(t) - x_i(t) + x_{r_1}(t) - x_{r_2}(t)),$$

$$(DE4) \quad v_i(t+1) = x_g(t) + F(x_{r_1}(t) - x_{r_2}(t) + x_{r_3}(t) - x_{r_4}(t)),$$

$$(DE5) \quad v_i(t+1) = x_{r_1}(t) + F(x_{r_2}(t) - x_{r_3}(t) + x_{r_4}(t) - x_{r_5}(t)),$$

## ΚΕΦΑΛΑΙΟ 4 – ΜΕΘΟΔΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΦΩΤΕΙΝΩΝ ΣΗΜΑΤΟΔΟΤΩΝ

Η κανονική λειτουργία των φωτεινών σηματοδοτών απαιτεί περισσότερα από έναν απλό έλεγχο και συντονισμό για να διασφαλιστεί ότι η κυκλοφορία πραγματοποιείται όσο πιο ομαλά και με ασφάλεια και ότι οι πεζοί προστατεύονται όταν διασχίζουν τους δρόμους. Χρησιμοποιείται μια ποικιλία διαφορετικών συστημάτων ελέγχου, τα οποία κυμαίνονται από απλούς μηχανισμούς ρολογιού μέχρι εξειδικευμένα ηλεκτρονικά συστήματα ελέγχου και συντονισμού που προσαρμόζονται αυτομάτως ώστε να ελαχιστοποιούν την καθυστέρηση στους ανθρώπους που χρησιμοποιούν το δρόμο.

### 4.1. Συστήματα ελέγχου κυκλοφορίας

Ένα σήμα κυκλοφορίας ελέγχεται συνήθως από έναν ελεγκτή τοποθετημένο μέσα σε ένα ερμάριο [3]. Ορισμένοι ηλεκτρομηχανικοί ελεγκτές εξακολουθούν να χρησιμοποιούνται (η Νέα Υόρκη είχε ακόμα 4.800 από το 1998, αν και ο αριθμός είναι χαμηλότερος τώρα λόγω της επικράτησης των κουτιών ελέγχου σήματος [4]). Ωστόσο, οι σύγχρονοι ελεγκτές κυκλοφορίας είναι στερεά. Το ντουλάπι περιέχει συνήθως έναν πίνακα ισχύος, για τη διανομή της ηλεκτρικής ενέργειας στο ντουλάπι, ένα πάνελ διεπαφής ανιχνευτή, για σύνδεση με ανιχνευτές βρόχων και άλλους ανιχνευτές, ενισχυτές ανιχνευτή, τον ίδιο τον ελεγκτή, μια μονάδα παρακολούθησης των συγκρούσεων, φλας μεταγωγής, ένα πάνελ της αστυνομίας, προκειμένου να επιτρέψει στην αστυνομία να απενεργοποιήσει το σήμα, και άλλα στοιχεία. [3]

Στις Ηνωμένες Πολιτείες, οι ελεγκτές τυποποιούνται από το NEMA, το οποίο ορίζει πρότυπα για συνδετήρες, λειτουργικά όρια και διαστήματα [3]. Το πρότυπο TS-1 εισήχθη το 1976 για την πρώτη γενιά ελεγκτών στερεάς κατάστασης [5].

Οι ελεγκτές στερεάς κατάστασης πρέπει να διαθέτουν ανεξάρτητη μονάδα παρακολούθησης σύγκρουσης (CMU), η οποία εξασφαλίζει ασφαλή λειτουργία σε περίπτωση βλάβης. Η μονάδα CMU παρακολουθεί τις έξοδοι του ελεγκτή και εάν εντοπιστεί σφάλμα, η CMU χρησιμοποιεί τα ρελέ μεταφοράς φλας για να θέσει τη

διασταύρωση σε FLASH, με όλες τις κόκκινες λυχνίες να αναβοσβήνουν, αντί να εμφανίζει δυνητικά επικίνδυνο συνδυασμό σημάτων. Η μονάδα CMU προγραμματίζεται με τους επιτρεπόμενους συνδυασμούς φωτεινών σημείων και, για παράδειγμα, θα ανιχνεύσει εάν ο ελεγκτής δίδει στις κατευθυντήριες γραμμές αντίθετες κατευθύνσεις ένα πράσινο σήμα.



**Εικόνα 1. Μηχανοποιημένο κιβώτιο ελέγχου κυκλοφορίας**

Στα τέλη της δεκαετίας του 1990, μια προσπάθεια εθνικής τυποποίησης γνωστή ως Advanced Controller (ATC) πραγματοποιήθηκε στις Ηνωμένες Πολιτείες από το Ινστιτούτο Μηχανικών Συγκοινωνιών [5]. Το έργο επιχειρεί να δημιουργήσει ένα ενιαίο εθνικό πρότυπο για τους ελεγκτές φωτεινών σηματοδοτών. Η προσπάθεια τυποποίησης αποτελεί μέρος του εθνικού προγράμματος ευφών μεταφορικών συστημάτων που χρηματοδοτείται από διάφορους λογαριασμούς αυτοκινητοδρόμων, ξεκινώντας από το ISTEA το 1991, ακολουθούμενο από το TEA-21 και τους επόμενους λογαριασμούς. Οι υπεύθυνοι επεξεργασίας θα επικοινωνούν με το Εθνικό Σύστημα Επικοινωνιών για το πρωτόκολλο ITS (NTCIP), βασισμένο σε πρωτόκολλο Internet, ISO / OSI και ASN.1 [5]

Τα αντίγραφα ασφαλείας των μπαταριών εγκαθίστανται σε ξεχωριστό περίβλημα από τον πίνακα ελέγχου κυκλοφορίας στην κορυφή.

Οι φωτεινοί σηματοδότες πρέπει να καθοδηγούνται ως προς το πότε πρέπει να αλλάξουν το στάδιο και συνήθως συντονίζονται έτσι ώστε οι αλλαγές σταδίων να

συμβαίνουν σε κάποια σχέση με άλλα κοντινά σήματα ή με το πάτημα ενός πλήκτρου πεζών ή με τη δράση ενός χρονοδιακόπτη ή ενός αριθμού άλλων εισόδων.

Στις περιοχές που είναι επιρρεπείς σε διακοπή ρεύματος, η προσθήκη εφεδρικών μπαταριών στα συστήματα κυκλοφορίας μπορεί να ενισχύσει την ασφάλεια των οδηγών και των πεζών. Στο παρελθόν, θα χρειαζόταν μεγαλύτερη χωρητικότητα αδιάλειπτης τροφοδοσίας για να συνεχιστεί η πλήρης λειτουργία των σημάτων οδικής κυκλοφορίας με τη χρήση λαμπτήρων πυρακτώσεως. Το κόστος για ένα τέτοιο σύστημα θα ήταν απαγορευτικό. Μετά τις νέες γενιές σημάτων κυκλοφορίας που χρησιμοποιούν φώτα LED που καταναλώνουν 85-90% λιγότερη ενέργεια, είναι πλέον δυνατή η ενσωμάτωση των εφεδρικών μπαταριών στα συστήματα φωτεινών σηματοδοτών. Τα αντίγραφα ασφαλείας των μπαταριών θα εγκατασταθούν στον πίνακα ελέγχου κυκλοφορίας ή στο δικό τους περίβλημα δίπλα στον ελεγκτή.



**Εικόνα 2. Τα αντίγραφα ασφαλείας των μπαταριών εγκαθίστανται σε ξεχωριστό περίβλημα από τον πίνακα ελέγχου κυκλοφορίας στην κορυφή.**

Τα αντίγραφα ασφαλείας της μπαταρίας μπορούν να λειτουργήσουν τον ελεγκτή σε λειτουργία έκτακτης ανάγκης με το κόκκινο φως να αναβοσβήνει ή σε πλήρη λειτουργία. Το 2004, η Επιτροπή Ενέργειας Καλιφόρνιας συνέστησε στις τοπικές κυβερνήσεις να μετατρέψουν τα φανάρια τους σε LED με εφεδρεία μπαταριών. Αυτό θα μειώσει την κατανάλωση ενέργειας και θα ενισχύσει την ασφάλεια σε μεγάλες

διασταυρώσεις. Η σύσταση ήταν για ένα σύστημα που παρέχει πλήρως λειτουργικά σήματα κυκλοφορίας για δύο ώρες μετά την διακοπή ρεύματος. Στη συνέχεια, τα σήματα θα έχουν αναβοσβήνει κόκκινο φάτα για άλλες δύο ώρες. [6]

## 4.2. Έλεγχος σταθερού χρόνου

Στον έλεγχο κυκλοφορίας, οι απλές και παλαιές μορφές ελεγκτών σήματος είναι αυτές που είναι γνωστές ως ηλεκτρομηχανικοί ελεγκτές σημάτων. Σε αντίθεση με τους ηλεκτρονικούς ελεγκτές σήματος, οι ηλεκτρομηχανικοί ελεγκτές σήματος αποτελούνται κυρίως από κινητά μέρη (έκκεντρα, πηνία και άξονες) που ελέγχουν τα σήματα που συνδέονται σωστά με αυτά. Εκτός από τα κινητά μέρη, χρησιμοποιούνται επίσης ηλεκτρικά ρελέ. Γενικά, οι ηλεκτρομηχανικοί ελεγκτές σήματος χρησιμοποιούν χρονόμετρα επιλογής τα οποία έχουν σταθερά, σηματοδοτημένα χρονοδιαγράμματα διασταύρωσης. Τα μήκη κύκλου σηματοδοτημένων διασταυρώσεων καθορίζονται από μικρά γρανάζια που βρίσκονται μέσα στους χρονομετρητές κλήσης. Τα κυκλικά γρανάζια, όπως είναι συνήθως γνωστά, κυμαίνονται από 35 δευτερόλεπτα έως 120 δευτερόλεπτα. Αν ένας κύλινδρος σε έναν χρονιστή επιλογής προκαλεί βλάβη, μπορεί να αντικατασταθεί με άλλο κιβώτιο ταχυτήτων που θα ήταν κατάλληλο για χρήση. Δεδομένου ότι ο χρονομετρητής περιλαμβάνει μόνο ένα χρονοδιάγραμμα τομής, αυτό μπορεί να ελέγχει τις φάσεις σε μια σηματοδοτούμενη διασταύρωση μόνο με έναν τρόπο. Πολλές παλιές σηματοδοτούμενες διασταυρώσεις εξακολουθούν να χρησιμοποιούν ηλεκτρομηχανικούς ελεγκτές σήματος και τα σήματα που ελέγχονται από αυτά είναι αποτελεσματικά σε πλέγματα μονής κατεύθυνσης όπου συχνά είναι δυνατόν να συντονιστούν τα σήματα στο μεταγεγραμμένο όριο ταχύτητας. Είναι ωστόσο μειονεκτικές όταν ο χρονισμός σήματος μιας διασταύρωσης θα επωφελείτο από την προσαρμογή στις κυρίαρχες ροές που αλλάζουν κατά τη διάρκεια της ημέρας [7].

### **4.3. Δυναμικός έλεγχος**

Ο ελεγκτής χρησιμοποιεί είσοδο από ανιχνευτές, οι οποίοι είναι αισθητήρες που ενημερώνουν τον επεξεργαστή του ελεγκτή εάν υπάρχουν οχήματα ή άλλοι χρήστες του δρόμου, για να ρυθμίσουν το χρονισμό του σήματος και τη σταδιακή κατάργηση εντός των ορίων που ορίζονται από τον προγραμματισμό του ελεγκτή. Μπορεί να δώσει περισσότερο χρόνο σε μια προσέγγιση διασταύρωσης που αντιμετωπίζει έντονη κυκλοφορία, ή να συντομεύσει ή ακόμα και να παραλείψει μια φάση που έχει μικρή ή καθόλου κίνηση που περιμένει ένα πράσινο φως. Οι ανιχνευτές μπορούν να ομαδοποιηθούν σε τρεις κατηγορίες: ανιχνευτές πεζοδρομίων, μη παρεμβατικοί ανιχνευτές και ανίχνευση για χρήστες μη οδικών οδών.

#### ***Ανιχνευτές πεζοδρομίων***

Αυτοί οι ανιχνευτές είναι θαμμένοι μέσα ή κάτω από το οδόστρωμα. Οι επαγωγικοί βρόχοι ανιχνευτών είναι ο πιο συνηθισμένος τύπος. Είναι αισθητήρες που είναι θαμμένοι στο δρόμο για να ανιχνεύσουν την παρουσία κίνησης που περιμένει στο φως και έτσι μπορεί να μειώσει το χρόνο που ένα πράσινο σήμα δίνεται σε έναν άδειο δρόμο. Ένας χρονοδιακόπτης χρησιμοποιείται συχνά ως προεπιλογή κατά τη διάρκεια πολύ χαμηλής πυκνότητας κυκλοφορίας και ως εφεδρικό σε περίπτωση αποτυχίας των αισθητήρων. Οι βρόχοι αισθητήρων τυπικά λειτουργούν με τον ίδιο τρόπο όπως και οι ανιχνευτές μετάλλων. Συνεπώς, μικρά οχήματα και ποδήλατα ή οχήματα με χαμηλή περιεκτικότητα σε μέταλλα ενδέχεται να μην ανιχνεύονται προκαλώντας τους να περιμένουν επ' αόριστον, εκτός εάν υπάρχει επίσης ένας χρονοδιακόπτης ως μέρος του συστήματος ελέγχου.

#### ***Μη παρεμβατικοί ανιχνευτές***

Μερικές φορές είναι πιο συμφέρουσα και οικονομικά αποδοτική η εγκατάσταση αισθητήρων υπεράνω του οδοστρώματος παρά η κοπή του δρόμου και η ενσωμάτωση των επαγωγικών βρόχων. Αυτές οι τεχνολογίες περιλαμβάνουν τη σωστή διαχείριση βίντεο, την ανταγωνιστική τιμολόγηση για την εγκατάσταση σε επίπεδο νομισματικής και εργατικής φύσης και τον κίνδυνο για το προσωπικό εγκατάστασης και την ικανότητα να ενεργούν ως συσκευές διαχείρισης της κυκλοφορίας σε

πραγματικό χρόνο. Επίσης, λειτουργούν ως ανιχνευτές πολλαπλών λωρίδων και συλλέγουν τύπους δεδομένων που δεν είναι διαθέσιμοι από αισθητήρες εσωτερικής οδού. [8]

### ***Μη μηχανική ανίχνευση χρηστών***

Ορισμένα φώτα κυκλοφορίας στις διαβάσεις πεζών, ειδικά εκείνα που βρίσκονται εκτός διασταυρώσεων, περιλαμβάνουν ένα κουμπί το οποίο πρέπει να πατηθεί για να ενεργοποιηθεί το σύστημα χρονισμού. Αυτό γενικά συνοδεύεται από μια μεγάλη οθόνη με την ένδειξη "wait", η οποία ανάβει όταν πατηθεί το κουμπί. αυτό το σήμα σβήνει όταν τα φώτα των οχημάτων εισέλθουν στην "κόκκινη" φάση. Στις Ηνωμένες Πολιτείες, τα σήματα πεζών συνεχίζουν να εμφανίζουν ένα σταθερό κόκκινο σήμα "χέρι" ή "Μην περπατάτε" όταν πιέζετε το κουμπί, σε ένα λευκό σήμα "άνθρωπος" ή "Περπατάτε" στο τέλος της φάσης των οχημάτων. Συχνά υπάρχουν άλλες οθόνες, όπως η αντίστροφη μέτρηση ή οι πράσινοι και οι κόκκινοι φώτα πεζών, που περιλαμβάνονται σε αυτόν τον πίνακα. Με την εμφάνιση των φωτεινών σηματοδοτών που ελέγχονται από υπολογιστή σε πολλές χώρες, τα κουμπιά ενεργοποίησης έχουν καταστεί άνευ αντικειμένου. Στην πραγματικότητα, οι περισσότεροι στη Νέα Υόρκη έχουν αποσυνδεθεί. [9] [10] Αντίθετα, οι νέες εγκαταστάσεις κουμπιών ενεργοποίησης προσφέρουν όλο και περισσότερες ειδικές ομάδες χρηστών, συμπεριλαμβανομένων ακουστικών κουμπιών και σημάτων για χρήστες με προβλήματα όρασης και τα λεγόμενα πλήκτρα διασταύρωσης Pegasus [11] για χρήστες με άλογο [12]. [13]

Οι τυπικοί ανιχνευτές σήματος έχουν έναν σκληρό χρόνο ανίχνευσης ποδηλάτων, λόγω της χαμηλής περιεκτικότητας σε σιδηρούχα μέταλλα των τυπικών ποδηλάτων. Εάν ένας ποδηλάτης βόλτα απευθείας πάνω από τα καλώδια ενός βρόχου ανιχνευτή, μπορεί να ανιχνεύσει τον ποδηλάτη. Ωστόσο, δεν λειτουργεί πάντα, και λίγοι ποδηλάτες ξέρουν να το κάνουν. Σε περιοχές όπου οι ποδηλάτες είναι συνηθισμένοι, μπορεί να χρησιμοποιηθεί ένας ειδικός βρόχος ανιχνευτή για τους ποδηλάτες. Ένα μικρό σύμβολο ποδηλάτου σημειώνεται συχνά στο πεζοδρόμιο για να ενημερώσει τον ποδηλάτη από πού πρέπει να σταματήσει για να ενεργοποιήσει το σήμα. [14] Άλλα μέρη τοποθετούν απλά ένα επιπλέον κουμπί πεζών κοντά στο περίπτερο όπου μπορεί να φτάσει ένας ποδηλάτης.

#### 4.4. Συντονισμένος έλεγχος

Προσπαθούν συχνά να τοποθετούν σήματα κυκλοφορίας σε ένα συντονισμένο σύστημα, έτσι ώστε οι οδηγοί να αντιμετωπίζουν ένα πράσινο κύμα, μια μακρά σειρά πράσινων φώτων (ο τεχνικός όρος είναι εξέλιξη). Η διάκριση μεταξύ συντονισμένων σημάτων και συγχρονισμένων σημάτων είναι πολύ σημαντική. Τα συγχρονισμένα σήματα αλλάζουν ταυτόχρονα και χρησιμοποιούνται μόνο σε ειδικές περιπτώσεις ή σε παλαιότερα συστήματα. Τα συντονισμένα (προχωρημένα) συστήματα ελέγχονται από έναν κύριο ελεγκτή και ρυθμίζονται έτσι ώστε τα φώτα "cascade" (progress) να ακολουθούν, έτσι ώστε οι διμοιρίες των οχημάτων να μπορούν να προχωρήσουν μέσα από μια συνεχή σειρά πράσινων φώτων. Μια γραφική αναπαράσταση της κατάστασης φάσης σε ένα επίπεδο δύο αξόνων της απόστασης σε σχέση με το χρόνο δείχνει σαφώς μια "πράσινη ζώνη" που έχει καθοριστεί με βάση την σηματοδοτημένη απόσταση μεταξύ των διασταυρώσεων και τις αναμενόμενες ταχύτητες του οχήματος. [15] Σε ορισμένες χώρες (π.χ. Γερμανία, Γαλλία και Κάτω Χώρες), αυτό το σύστημα "πράσινης ζώνης" χρησιμοποιείται για τον περιορισμό της ταχύτητας σε ορισμένες περιοχές. Τα φώτα είναι χρονομετρημένα κατά τέτοιο τρόπο ώστε οι οδηγοί να μπορούν να οδηγούν χωρίς να σταματούν εάν η ταχύτητά τους είναι χαμηλότερη από ένα δεδομένο όριο, συνήθως 50 km / h (30 mph) σε αστικές περιοχές. Αυτό το σύστημα είναι γνωστό ως "grüne Welle" στα γερμανικά, "vague verte" στα γαλλικά, ή "groene golf" στα ολλανδικά (αγγλικά: "πράσινο κύμα"). Τέτοια συστήματα χρησιμοποιούνται συνήθως στις αστικές περιοχές των Ηνωμένων Πολιτειών από τη δεκαετία του 1940, αλλά είναι λιγότερο κοινά σήμερα. Στο Ηνωμένο Βασίλειο, το Slough στο Berkshire είχε μέρος του A4 που πειραματίστηκε με αυτό. Πολλές πόλεις των ΗΠΑ έβαλαν το πράσινο κύμα σε δρόμους διπλής κατεύθυνσης για να λειτουργούν προς την κατεύθυνση που ταξίδευαν περισσότερο, αντί να προσπαθούν να προχωρήσουν στην κυκλοφορία και στις δύο κατευθύνσεις. Αλλά η πρόσφατη εισαγωγή του κίτρινου βέλους που αναβοσβήνει (βλέπε σηματοδότηση και λειτουργία φώτων κυκλοφορίας) καθιστά το σήμα καθυστέρησης μολύβδου, ένα βοήθημα για την πρόοδο, διαθέσιμο με προστατευμένες / επιτρεπτικές στροφές [15]. [16]

Στα σύγχρονα συντονισμένα συστήματα σήματος, είναι δυνατό οι οδηγοί να ταξιδεύουν σε μεγάλες αποστάσεις χωρίς να συναντήσουν κόκκινο φως. Αυτός ο συντονισμός γίνεται εύκολα μόνο σε δρόμους μονής κατεύθυνσης με σχετικά σταθερά



επίπεδα κυκλοφορίας. Οι δρόμοι δύο δρόμων είναι συχνά διατεταγμένοι ώστε να αντιστοιχούν στις ώρες αιχμής για να επιταχύνουν την βαρύτερη κατεύθυνση του όγκου. Η συμφόρηση μπορεί συχνά να αποβάλει κάθε συντονισμό. Από την άλλη πλευρά, ορισμένα σήματα κυκλοφορίας συντονίζονται για να αποτρέψουν τους οδηγούς να συναντήσουν μια μακρά σειρά πράσινων φώτων. Αυτή η πρακτική αποθαρρύνει τους μεγάλους όγκους κίνησης προκαλώντας καθυστέρηση, αποτρέποντας παράλληλα την συμφόρηση. Η ταχύτητα ρυθμίζεται αυτορυθμιζόμενη σε συντονισμένα συστήματα σήματος, οι οδηγοί που ταξιδεύουν πολύ γρήγορα θα φτάσουν σε μια κόκκινη ένδειξη και θα σταματήσουν να φτάνουν, οι οδηγοί που ταξιδεύουν πολύ αργά δεν θα φτάσουν στο επόμενο σήμα εγκαίρως για να χρησιμοποιήσουν την πράσινη ένδειξη. Σε συγχρονισμένα συστήματα, ωστόσο, οι οδηγοί συχνά χρησιμοποιούν υπερβολική ταχύτητα για να περάσουν όσο το δυνατόν περισσότερα φώτα.

Πιο πρόσφατα έχουν χρησιμοποιηθεί ακόμα πιο εξελιγμένες μέθοδοι. Τα φώτα πορείας μερικές φορές ελέγχονται κεντρικά από οθόνες ή από ηλεκτρονικούς υπολογιστές, ώστε να μπορούν να συντονίζονται σε πραγματικό χρόνο για να αντιμετωπίζουν τα μεταβαλλόμενα μοντέλα κυκλοφορίας [17]. Οι βιντεοκάμερες ή οι αισθητήρες που είναι θαμμένοι στο οδόστρωμα μπορούν να χρησιμοποιηθούν για την παρακολούθηση της κυκλοφορίας σε μια πόλη. Οι μη συντονισμένοι αισθητήρες παρεμποδίζουν περιστασιακά την κυκλοφορία ανιχνεύοντας μια ηρεμία και γυρίζοντας κόκκινο ακριβώς όπως τα αυτοκίνητα φθάνουν από το προηγούμενο φως. Τα συστήματα υψηλής τεχνολογίας χρησιμοποιούν δεκάδες αισθητήρες και κοστίζουν εκατοντάδες χιλιάδες δολάρια ανά διασταύρωση, αλλά μπορούν να ελέγξουν πολύ προσεκτικά τα επίπεδα κυκλοφορίας. Αυτό ανακουφίζει την ανάγκη για άλλα μέτρα (όπως νέους δρόμους) που είναι ακόμη πιο ακριβά.

Τα οφέλη περιλαμβάνουν: [18] [19]

- Αύξηση της ικανότητας χειρισμού κυκλοφορίας των οδών
- Μείωση των συγκρούσεων και του χρόνου αναμονής τόσο για τα οχήματα όσο και για τους πεζούς [20]
- Ενθάρρυνση του ταξιδιού εντός του ορίου ταχύτητας για την κάλυψη των πράσινων φώτων

- Μείωση της περιττής διακοπής και εκκίνησης της κυκλοφορίας - αυτό με τη σειρά του μειώνει την κατανάλωση καυσίμου, την ατμοσφαιρική και ηχητική ρύπανση και τη φθορά των οχημάτων
- Μείωση του χρόνου ταξιδιού
- Μείωση της απογοήτευσης του οδηγού και της οργής στο δρόμο

#### Παραδείγματα:

Νέα Υόρκη: 7.660 (από συνολικά 12.460) σηματοδοτούμενες διασταυρώσεις ελέγχονται από ένα κεντρικό δίκτυο υπολογιστών [4] [21]

Τορόντο: Το 83% των σημάτων του ελέγχονται από το σύστημα κύριων σημάτων κυκλοφορίας (MTSS). Το 15% χρησιμοποιεί επίσης το SCOOT (κύκλος διαχωρισμού και τεχνική βελτιστοποίησης αντιστάθμισης), ένα προσαρμοστικό σύστημα ελέγχου σήματος. [22]

Σίδνεϊ: 3.400 σήματα κυκλοφορίας που συντονίζονται από το Σύστημα Διακυβερνητικής Διακυβερνητικής Συγκέντρωσης (SCATS). Σχεδιασμένο και αναπτυγμένο από την RTA, το σύστημα εισήχθη για πρώτη φορά το 1963 και σταδιακά αναπτύχθηκε έκτοτε. Μέχρι τον Οκτώβριο του 2010, η SCATS είχε λάβει άδεια σε 33.200 διασταυρώσεις σε 144 πόλεις σε 24 χώρες παγκοσμίως, όπως η Σιγκαπούρη, το Χονγκ Κονγκ, το Δουβλίνο, η Τεχεράνη και η Μινεάπολη και το Ντιτρόιτ [19] [23]

Μελβούρνη: 3.200 φανάρι σε όλη τη Βικτώρια, συμπεριλαμβανομένων των περιφερειακών περιοχών όπως το Geelong και το Ballarat, χρησιμοποιώντας το SCATS. Περίπου 500 διασταυρώσεις έχουν επίσης προτεραιότητα τραμ και λεωφορείων. [25]

Αδελαΐδα: 580 σύνολα συντονισμένων φανών σε ολόκληρη την μητροπολιτική περιοχή που διαχειρίζεται το Σύστημα συντονισμού της κυκλοφορίας Αδελαΐδας (ACTS) [18].

## ΚΕΦΑΛΑΙΟ 5 – ΧΡΗΣΗ ΕΥΡΕΤΙΚΩΝ ΚΑΙ ΜΕΤΑΕΥΡΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΤΗΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΩΝ ΦΩΤΕΙΝΩΝ ΣΗΜΑΤΟΔΟΤΩΝ

### 5.1. Εισαγωγή

Ο περιορισμός της εισερχόμενης ροής κίνησης είναι μια από τις προσεγγίσεις της διαχείρισης της ζήτησης ταξιδιού (TDM) για την αποτροπή της απόφραξης στις διασταυρώσεις και τις κοινές τους συνδέσεις. Αυτή η προσέγγιση μπορεί να εφαρμοστεί τόσο σε απομονωμένες όσο και σε ολοκληρωμένες διασταυρώσεις. Σε πυκνά δίκτυα με ομοιόμορφη κατανομή συμφόρησης, η διαχείριση της εισερχόμενης κυκλοφοριακής ροής ενδέχεται να έχει ως αποτέλεσμα την κατανομή της πράσινης ώρας, η οποία τελικά οδηγεί σε αυξημένη απόδοση δικτύου.

Υπάρχουν πολλές μελέτες στον τομέα της "βελτιστοποίησης του σήματος κυκλοφορίας" που πραγματοποίησαν οι ερευνητές τις τελευταίες δύο δεκαετίες. Από την άποψη αυτή, οι Gal-Tzur et al. προσέφερε μια μέθοδο ελέγχου σήματος βασισμένη στη μέτρηση της κυκλοφορίας. Παρουσίασαν ένα μοντέλο για τον σχεδιασμό του χρονισμού του σήματος για το συμφορημένο αστικό δίκτυο, στο οποίο μία ή περισσότερες διασταυρώσεις μπορούν να φτάσουν σε υπερκορεσμένες συνθήκες σε σύγκριση με άλλες διασταυρώσεις [1]. Η βασική ιδέα πίσω από την προσέγγιση ήταν να περιοριστεί ο εισερχόμενος όγκος στην ικανότητα κρίσιμης τομής, προκειμένου να αποφευχθεί η αδιέξοδο στο αστικό δίκτυο μεταφορών. Η διαδικασία μέτρησης επιτρέπει στους χειριστές του συστήματος να ανιχνεύσουν τη θέση και το μήκος των ουρών για να ρυθμίσουν τις διαδρομές για τη δημιουργία κατάλληλων buffer αποκατάστασης. Σε γενικές γραμμές, η στρατηγική αυτή ονομάζεται "Διαχείριση θέσης ουράς". Τα προγράμματα προσομοίωσης της κυκλοφορίας και βελτιστοποίησης του χρονισμού σήματος όπως το TRANSYT χρησιμοποιούν το μαθηματικό μοντέλο που σχεδιάστηκε και ενσωματώθηκε για την εφαρμογή της προσέγγισης.

ύμφωνα με τη θεωρία Daganzo [3] όταν το δίκτυο είναι σε κατάσταση συμφόρησης, ακόμη και η μικρότερη αύξηση του όγκου εισόδου πέρα από την κρίσιμη τιμή μπορεί να έχει σημαντικό αντίκτυπο στην κατάσταση της κυκλοφορίας και να έχει ως αποτέλεσμα την επιδείνωση της απόδοσης του δικτύου. Σε τέτοιες περιπτώσεις,

όπου το σύστημα είναι ασταθές, ο έλεγχος της ροής εισόδου με την πρόσληψη στρατηγικών μέτρησης στην περιοχή των ορίων είναι απαραίτητη για να καταστεί δυνατή η διαχείριση της συμφόρησης.

Τα οχήματα που χρησιμοποιούν σηματοδοτούμενη διασταύρωση μπορούν να εκτιμήσουν την απόδοση των ταξιδιών τους βάσει της καθυστερημένης καθυστέρησης τους σε αυτή τη διασταύρωση. Ωστόσο, η βελτιστοποίηση της λειτουργίας του σήματος κυκλοφορίας βάσει αυτού του μέτρου αποτελεσματικότητας, καθώς και άλλων, είναι πολύπλοκη καθώς ο αριθμός των οχημάτων που φθάνουν και ο αριθμός των φάσεων για το σήμα. Αυτοί οι παράγοντες μπορούν να δημιουργήσουν ένα εξαιρετικά μεγάλο χώρο λύσης για τη βελτιστοποίηση. Κατά την εκτέλεση αυτής της βελτιστοποίησης για σηματοδοτούμενη διασταύρωση, ο χρόνος υπολογισμού είναι καθοριστικός για την επιτυχία της εφαρμογής της λύσης βραχυπρόθεσμα (π.χ. από τη φάση στη φάση ή από τον κύκλο στον κύκλο). Αυτή η πρόταση ασχολείται με την ανάπτυξη μιας ευρετικής προσέγγισης, μέσω της μεθόδου Webster για τη διάσπαση σήματος, και με μια μεθευρετική προσέγγιση, μέσω της αναζήτησης tabu, στην εξεύρεση λύσης στο πρόβλημα σηματοδοτούμενης βελτιστοποίησης των διασταυρώσεων. Οι προσεγγίσεις που επινοήθηκαν στο παρόν βρέθηκαν, με πειραματισμό πάνω από αρκετές εκατοντάδες σειρές, για να μειωθούν οι χρόνοι υπολογισμού κατά μέσο όρο 44% και έως 76%, σε ορισμένες περιπτώσεις, σε σύγκριση με μια εξαντλητική αναζήτηση του χώρου διαλύματος. Διαπιστώθηκε επίσης ότι ανήλθαν σε μέσο όρο 10% της βέλτιστης λύσης και μείωσαν τη διακύμανση στο τελικό διάλυμα που βρέθηκε από το τρέξιμο για να τρέξει κατά μέσο όρο 33%. Τα αποτελέσματα αυτά θα επιτρέψουν τη βελτίωση της σκοπιμότητας για την εφαρμογή πιο σύνθετων στρατηγικών ελέγχου της κυκλοφορίας με μεγαλύτερο αντίκτυπο στη μείωση της συμφόρησης στις σηματοδοτούμενες διασταυρώσεις.

## **5.2. Υπόβαθρο**

Όταν τα οχήματα χρησιμοποιούν σηματοδοτούμενη διασταύρωση κυκλοφορίας, η απόδοση της εν λόγω διασταύρωσης μπορεί να μετρηθεί αντικειμενικά. Αυτό επιτυγχάνεται μέσω ποικίλων μέτρων αποτελεσματικότητας (MOE), τα οποία μπορεί να περιλαμβάνουν [Pignataro (1973) και May (1990)]:

- τον αριθμό των στάσεων των οχημάτων
- τα οχήματα χρονομέτρησης πρέπει να σταματήσουν ή να σταματήσουν την καθυστέρηση (σύνολο ή μέση τιμή ανά
- όχημα)
- τα χρονικά οχήματα που περνούν στο σύστημα
- χρόνος ταξιδιού
- η διαφορά μεταξύ του χρόνου ταξιδιού και του ελάχιστου χρόνου που μπορούν να δαπανήσουν τα οχήματα στο σύστημα ή της συνολικής καθυστέρησης

Τα συστήματα ελέγχου της κυκλοφορίας έχουν σχεδιαστεί τα τελευταία 40 περίπου χρόνια, τα οποία έχουν αρχίσει να εκμεταλλεύονται την ευκολότερη και πιο εύκολη μέτρηση αυτών των MOE, είτε μέσω της χρήσης ανιχνευτών, εξοπλισμού βίντεο είτε RFID (π.χ. αναμεταδότες διοδίων). Αυτά τα συστήματα, αν μπορούν να ανταποκριθούν άμεσα στις αλλαγές σε αυτά τα MOE με αλλαγές στο κυκλοφοριακό ρεύμα, ονομάζονται προσαρμοστικά συστήματα σήματος κυκλοφορίας. Οι πιο σημαντικές από πλευράς χρήσης και εφαρμογής είναι:

- Βελτιστοποιημένες πολιτικές για προσαρμοστικό έλεγχο (OPAC) [Gartner (1983)]
- Τεχνική Βελτιστοποίησης Offset Cycle Offset (SCOOT) [Greenough and Kelman (1998)]
- Το Σύστημα Συντονισμένου Κυκλοφοριακού Συστήματος του Σίδνεϊ (SCATS) [Lowrie 2001]
- Ιεραρχικό, βελτιστοποιημένο, διανεμημένο, αποτελεσματικό σύστημα σε πραγματικό χρόνο (RHODES) [Mirchandani and Head (2001)]

These all use, to some degree, a means of evaluating traffic streams in either recently-past ή επερχόμενες βραχυπρόθεσμες χρονικές περιόδους (που κυμαίνονται από μερικά δευτερόλεπτα έως πέντε λεπτά σε μήκος). Οι αξιολογήσεις τους είναι είτε εξαντλητικές σε σχέση με τα βασικά κριτήρια (π.χ. παρουσία ή απουσία οχημάτων σε δεδομένη χρονική περίοδο), είτε εξετάζουν χρονικές περιόδους με καθορισμένα μήκη (π.χ., καθορισμένη χρονική υστέρηση μεταξύ διασταυρώσεων ή μετατόπιση). Ωστόσο, αυτές οι μέθοδοι δεν αξιοποιούν πλήρως το δυναμικό των συστημάτων συλλογής δεδομένων για την παροχή πληροφοριών σχετικά με τις ροές κυκλοφορίας. Πιο πρόσφατα, οι μεθοδολογίες προσαρμοζόμενου ελέγχου σήματος στην ανάπτυξη

χρησιμοποιούν τα δεδομένα άφιξης και αναχώρησης σχετικά με μεμονωμένα οχήματα για τον προσδιορισμό της απόδοσης μέσω των MOE. αυτά περιλαμβάνουν το Cai et. al. (2009) και Fang και Ελευθεριδαδου (2010). Τέτοια συστήματα μπορούν να λειτουργούν σε χρονικές περιόδους με βάση τα μοτίβα ταξιδιού των οχημάτων, έτσι ώστε ούτε τα χρονικά διαστήματα ούτε οι θέσεις να καθορίζονται σε έναν ορίζοντα αξιολόγησης. Τα σήματα κυκλοφορίας σε τέτοια συστήματα ελέγχονται από τη βελτιστοποίηση ενός MOE που μπορεί να υπολογιστεί απευθείας από αυτά τα δεδομένα άφιξης ή / και αναχώρησης.

Εδώ, το σύστημα που επινοήθηκε από τους Shenoda και Machemehl (2006) θα εξεταστεί για βελτιστοποίηση. Σε αυτό, η βασική ιδέα είναι ότι η καθυστέρηση διακοπής υπολογίζεται με βάση τη διαφορά μεταξύ του χρονικού σημείου άφιξης κάθε οχήματος και του χρονικού σημείου κατά το οποίο το σήμα κυκλοφορίας αλλάζει σε κόκκινο. Το χρονικό σημείο μετατοπίζεται έπειτα έως ότου ελαχιστοποιηθεί το άθροισμα των καθυστερημένων οχημάτων. Μια εξέταση των καταστάσεων και των συνθέσεων του σκευάσματος και του δείγματος δείχνει ότι ο αλγόριθμος περιλαμβάνει μια μη κυρτή βελτιστοποίηση.

### **5.3. Εξέταση της χρήσης ευρετικών αλγορίθμων**

Ακόμη και σε έναν περιορισμένο χρονικό ορίζοντα αξιολόγησης, μια τέτοια βελτιστοποίηση φαίνεται να έχει άπειρο αριθμό πιθανών λύσεων και, επομένως, μια αποτελεσματική διαδικασία αναζήτησης θα ήταν ικανή να εκπληρώσει το έργο. Η εύρεση του βέλτιστου χρονικού σημείου δεν θα ήταν εγγυημένη, αλλά δεδομένου του αρκετού χρόνου λειτουργίας και μιας κατάλληλης μεθόδου, θα μπορούσε να επιτευχθεί λύση μέσα σε κάποιο αποδεκτό περιθώριο σφάλματος του βέλτιστου. Οι αριθμητικές μέθοδοι αναζήτησης σε μη κυρτά διαστήματα συνήθως περιλαμβάνουν τη χρήση μιας μονοδιάστατης αναζήτησης (π.χ., απότομη κάθοδο, χρυσή τομή κλπ.) Της περιοχής σε διαστήματα μη μονόδρομου. Οι ιδιαιτερότητες αυτών των τύπων μεθόδων δεν θα συζητηθούν στο παρόν, αλλά αναφέρονται εύκολα μέσω της βιβλιογραφίας για τη βελτιστοποίηση. Ο Fletcher (1987) έκανε μια καλή αναφορά για τις ιδιαιτερότητες αυτών των μεθόδων. Υπάρχουν όμως μειονεκτήματα όσον αφορά τη χρήση αυτών,

δεδομένου ότι, ανάλογα με το αποδεκτό σφάλμα και την αρχική λύση που χρησιμοποιείται, μπορούν να χρονοβόρα και να εμποδίσουν την επίτευξη μιας αποδεκτής λύσης σε ένα πρακτικό χρονικό πλαίσιο. Άλλοι περιοριστικοί παράγοντες στη χρήση αυτού του τύπου προσέγγισης είναι το χρονικό βήμα κατά το οποίο επαναλαμβάνεται η βελτιστοποίηση και ο αριθμός των φάσεων πάνω από τις οποίες το σύστημα πρέπει να βελτιστοποιηθεί, γεγονός που μπορεί να επεκτείνει εκθετικά τον χρόνο υπολογισμού.

Υπό το πρίσμα αυτό, θα μπορούσε κανείς να σκεφτεί ότι δεν μπορεί να υπάρχει άπειρος αριθμός πιθανών χρονικών σημείων, αλλά ότι ο αριθμός αυτός είναι πρακτικά περιορισμένος. Είναι απλούστερο να θεωρούμε ότι τα χρονικά σημεία, κατά το διαχωρισμό του ορίζοντα σε τμήματα "πράσινου σήματος" και "κόκκινου σήματος", θα πρέπει να έχουν μια περιοριστική ανάλυση βασισμένη στην απόδοση του συστήματος. Το ένα άκρο του φάσματος των ψηφιοποιήσεων μπορεί να είναι, για παράδειγμα, η ταχύτητα μετάδοσης μεταξύ του επεξεργαστή και του ελεγκτή σήματος, ο οποίος μπορεί να είναι της τάξης των 10-3 δευτερολέπτων ή χαμηλότερος [AASHTO et al. (2004)] και η άλλη μπορεί να είναι η ελάχιστη απόσταση μεταξύ οχημάτων με ταχύτητα ελεύθερης ροής, η οποία γενικά υποτίθεται ότι βασίζεται σε Greenshields et al. (1947), για περίπου 2 δευτερόλεπτα. Αυτό θέτει τον αριθμό των πρακτικά πιθανών χρονικών σημείων στην περιοχή μικρότερη από 102 έως 104 ανά φάση. Όταν επιβάλλουμε αυτό το όριο στον αριθμό των χρονικών σημείων, η δυνατότητα εξαντλητικής αξιολόγησης μπορεί να επανεξεταστεί. Εντούτοις, ένα δυνητικό μειονέκτημα για τον περιορισμό των χρονικών σημείων με αυτή την προσέγγιση είναι ότι εάν η ανάλυση δεν είναι αρκετά μικρή, μπορεί να μην είναι δυνατή η καταγραφή της χρονικής διαφοράς μεταξύ αφίξεων οχημάτων σε πολλαπλές προσεγγίσεις.

Αυτό μπορεί να εξαλείψει την ικανότητα να χρησιμοποιήσει πιθανά χρονικά σημεία που θα μπορούσαν να βελτιώσουν την καθυστέρηση. (Αυτό είναι, στην πραγματικότητα, ένα πρόσθετο μειονέκτημα στις μεθόδους αριθμητικής αναζήτησης, οι οποίες μπορούν να παρουσιάσουν αυτό το σφάλμα αν το αποδεκτό σφάλμα στη λύση χρονικού σημείου δεν είναι αρκετά μικρό.) Ένα άλλο μειονέκτημα είναι, όπως με τις αριθμητικές μεθόδους αναζήτησης, ότι ως τομή βελτιστοποιημένη γίνεται πιο περίπλοκη, ο χρόνος υπολογισμού επίσης. Αν κάποιος εξετάσει τον τρόπο με τον οποίο υπολογίζεται η καθυστέρηση, με βάση τις αφίξεις οχημάτων, μπορεί κάποιος να δει ότι αυτή η καθυστέρηση μπορεί να βελτιστοποιηθεί πιο εύκολα κάνοντας μεταγωγείς στις

καταστάσεις σήματος μόνο μέσα σε κάποιο διάστημα αυτών των κρίσιμων σημείων κατά μήκος του ορίζοντα. Αυτό σημαίνει ότι η καθυστέρηση μπορεί να προσεγγίσει μόνο ένα ελάχιστο με την αλλαγή της κατάστασης σήματος όσο το δυνατόν γρηγορότερα, αφού το όχημα περάσει από μια προσέγγιση που επί του παρόντος εξυπηρετείται από ένα πράσινο σήμα. Όταν εξετάζουμε αυτό το πρόβλημα, το πρόβλημα γίνεται περισσότερο σαν ένα συνδυαστικό πρόβλημα βελτιστοποίησης. Σε αυτό, το χρονικό σημείο για το οποίο πρέπει να αλλάξει η κατάσταση του σήματος καθορίζει την τοποθέτηση ενός οχήματος σε ένα από τα τρία σύνολα: ένα σετ "κόκκινο πριν από το πράσινο", ένα σετ "πράσινο" και ένα σετ "κόκκινο μετά από το πράσινο".

Η καθυστέρηση για κάθε όχημα είναι μια καθορισμένη τιμή για κάθε ένα από τα τώρα περιορισμένα χρονικά σημεία που πρέπει να ληφθούν υπόψη. ο προσδιορισμός τότε γίνεται ένας από τους οποίους χρονικά σημεία θα επιτρέψει ποιες τιμές καθυστέρησης πρέπει να ληφθούν υπόψη σε ποια ομάδα, καθορίζοντας εάν η εν λόγω τιμή καθυστέρησης θα συμβάλει στη συνολική καθυστέρηση. Είναι επίσης πιθανό, με τον αριθμό των πιθανών χρονικών σημείων να μειώνεται ουσιαστικά στον αριθμό των οχημάτων που φθάνουν στη διασταύρωση κατά τη διάρκεια του ορίζοντα αξιολόγησης, να διεξάγει μια εξαντλητική απαρίθμηση των πιθανών χρονικών σημείων για να βρεθεί η βέλτιστη λύση. Μπορεί επίσης να είναι δυνατή η κατασκευή μιας διαδικασίας αριθμητικής αναζήτησης μέσω μιας προσέγγισης προγραμματισμού ακέραιων αριθμών μετά από αυτή τη μείωση. Ο συνολικός αριθμός συνδυασμών που χρησιμοποιούν μία από αυτές τις προσεγγίσεις θα είναι της τάξης του μέγιστου αριθμού οχημάτων που επιθυμούν να χρησιμοποιήσουν μια φάση κατά τη διάρκεια ενός δεδομένου ορίζοντα ως προς τη δύναμη του αριθμού των φάσεων. Ωστόσο, καθώς αυξάνεται ο αριθμός των οχημάτων και ο αριθμός των φάσεων, ο χρόνος υπολογισμού για μια εξαντλητική απαρίθμηση θα αυξηθεί εκθετικά. Για παράδειγμα, σε μια διασταύρωση τεσσάρων φάσεων όπου χρειάζεται να υποβληθούν σε επεξεργασία έως και 30 οχήματα για μια φάση, ο αριθμός πιθανών συνδυασμών χρονικού σημείου είναι  $30^4$  ή περίπου 810.000 συνδυασμοί. Αυτός ο τύπος προβλήματος θα μπορούσε εύκολα να αντιμετωπιστεί όταν μια φάση επεξεργάζεται περισσότερες από μία προσεγγίσεις (π.χ. μία φάση και για τις δύο κατευθύνσεις ενός δρόμου).



#### 5.4. Η αναλογική ευρετική μέθοδος

Μια βασική ευρετική προσέγγιση ή μια προσέγγιση επίλυσης προβλημάτων μπορεί να εφαρμοστεί στο πρόβλημα. Ο στόχος αυτού του ευρετικού είναι περιορισμένος, διότι θα κάνει μόνο τα εξής:

α) παρέχουν ένα χρήσιμο σημείο εκκίνησης για βελτιστοποίηση

β) παρέχουν μια βασική τιμή για την ταξινόμηση διαδοχικών χρονικών σημείων ή συνδυασμών αυτών για αξιολόγηση

Η αξιολόγηση μπορεί στη συνέχεια να αποκοπεί με βάση τους χρονικούς περιορισμούς του σήματος κυκλοφορίας, όπως συζητήθηκε προηγουμένως.

Σε αυτή την περίπτωση, θα χρησιμοποιηθεί μια λύση που βασίζεται σε αυτό που μπορεί να θεωρηθεί ως "ανάλογη ευρετική". Η ιδέα απαιτεί την εκκίνηση με ένα "αναλογικό χρονικό σημείο" για κάθε φάση, υπολογιζόμενο χρησιμοποιώντας την αναλογία οχημάτων που επιθυμούν να χρησιμοποιήσουν αυτή τη φάση κατά τη διάρκεια ενός ορίζοντα ως προς τον συνολικό αριθμό των οχημάτων που φθάνουν στη διασταύρωση κατά τη διάρκεια αυτού του ορίζοντα, όπως και η μέθοδος του Webster για την επίλυση των μηκών φάσης σε ένα κύκλο (Pignataro (1973) και May (1990)). Χρησιμοποιώντας αυτό το σύνολο αναλογικών χρονικών σημείων, λαμβάνεται αρχική λύση, με το χρονικό σημείο,  $\lambda$ , για κάθε φάση  $k$  που έχει τη μικρότερη απόκλιση από το αναλογικό χρονικό σημείο για αυτή τη φάση. Αυτό μπορεί να θεωρηθεί σε σχέση με το σύνολο των χρονικών σημείων για μια φάση που παράγεται από κάθε όχημα  $j$  (δηλαδή  $\lambda_{kj}$ ). Μαθηματικά, αυτή η αρχική λύση είναι::

$$\lambda_k = \lambda_{kj}$$

Για την πλήρη αξιοποίηση του αναλογικού ευρετικού, οι πιθανές χρονικές στιγμές των συνδυασμών πρέπει να απαριθμηθούν εξαντλητικά και στη συνέχεια να ταξινομηθούν για διαδοχική χρήση. Αυτή η κατάταξη γίνεται με αύξουσα σειρά του αθροίσματος των διαφορών μεταξύ του αναλογικού χρονικού σημείου και της τιμής  $\lambda_k$  στο συνδυασμό για κάθε φάση. Επομένως, οι λύσεις που αποκλίνουν λιγότερο από το αναλογικό σύστημα λάμδα θα ταξινομηθούν υψηλότερα και διαθέσιμες για χρήση στον αλγόριθμο σε προηγούμενες επαναλήψεις. Χρησιμοποιώντας τις λύσεις που βασίζονται σε αυτό το σύστημα κατάταξης, ο αλγόριθμος θα πρέπει να είναι σε θέση να φτάσει σε μια καλή, αν όχι τη βέλτιστη λύση, πολύ πιο γρήγορα από την κανονική εξαντλητική

απαρίθμηση. Πρέπει να σημειωθεί ότι αυτή η κατάσταση δεν είναι εγγυημένη, καθώς οι ιδιαιτερότητες στο σύνολο των ωρών άφιξης (π.χ. ομαδοποίηση αφίξεων στα άκρα του ορίζοντα) μπορεί να παράγουν ένα βέλτιστο χρονικό σημείο μακριά από το αναλογικό λάμδα για μια συγκεκριμένη φάση. Επιπλέον, η χρήση του αναλογικού ευρετικού είναι προτιμότερη μόνο σε προβλήματα μικρότερης κλίμακας, καθώς η διαδικασία εφαρμογής του ευρετικού για την πλήρη επίλυση του αλγορίθμου απαιτεί επί του παρόντος την εξαντλητική ταξινόμηση των λύσεων και στη συνέχεια την εξαντλητική αναζήτηση τους. καθώς αυξάνεται ο αριθμός των λύσεων, η διαδικασία τελικά απαιτεί περισσότερο υπολογιστικό χρόνο από ό, τι εξοικονομείται για την επίτευξη της βέλτιστης ή καλής λύσης σε μια προηγούμενη επανάληψη.

### **5.5. Πιθανά οφέλη από μια μεθευρετική προσέγγιση**

Σαφώς, μια βασική ευρετική, όπως αυτή που περιγράφεται παραπάνω, είναι πολύ χρήσιμη σε ορισμένες περιπτώσεις μικρής κλίμακας, αλλά μπορεί να μην είναι τόσο χρήσιμη σε άλλες. Υπάρχουν επίσης περιπτώσεις όπου, ακόμη και σε μια κολοβωμένη αναζήτηση όπου μια καλή λύση μπορεί να αναμένεται μετά από περιορισμένο αριθμό επαναλήψεων, η ευρετική παραπάνω μπορεί να είναι προβληματική. Αυτές θα συνεπαγόταν την "διπλασιασμό" ή την ομαδοποίηση οχημάτων σε διάφορες χρονικές θέσεις εντός του ορίζοντα αξιολόγησης. Είναι δυνατό για μια μικρή διμοιρία οχημάτων να μετατοπίσουν το βέλτιστο χρονικό σημείο σε απόσταση μεγαλύτερη από το μέγεθός τους σε σχέση με το συνολικό κυκλοφοριακό ρεύμα, απλώς λόγω των εκτεταμένων καθυστερήσεων. Κάτι τέτοιο θα καθιστούσε την αναλογική αξία μη καταλληλότερη ως κριτήριο για εξέταση από οποιοδήποτε άλλο χρονικό σημείο. Το ευρετικό μπορεί να προσαρμοστεί για διασταυρώσεις όπου αναμένεται τέτοια συμπεριφορά με βάση τα διδάγματα, αλλά αυτό δεν θα ήταν στο πνεύμα της "προσαρμοστικότητας" μιας μεθοδολογίας ελέγχου. Ένα ευρετικό που επιτρέπει μια μετατόπιση του κριτηρίου αξιολόγησης με βάση προηγούμενα πρότυπα άφιξης μπορεί να αποτελέσει τη βάση για περαιτέρω έρευνα.

Μπορεί να υπάρχουν άλλα μειονεκτήματα στη χρήση ενός βασικού ευρετικού για το σενάριο προσαρμοστικού ελέγχου. Μια κατάσταση όπου υπάρχει ένα υψηλό

καθεστώς ζήτησης προς επεξεργασία από μία φάση και η χαμηλότερη ζήτηση προς επεξεργασία από τις άλλες θα δημιουργούσε πολλούς συνδυασμούς όπου πιθανές μηδενικές τιμές χρονικού σημείου κατέχουν τους χώρους στο χώρο λύσης όπου οι χρόνοι άφιξης του οχήματος σε αυτές τις φάσεις είναι ανύπαρκτοι . Οι δυνατότητες εξορθολογισμού μιας αναζήτησης που εμπλέκει αυτό το ζήτημα θα είναι η δημιουργία μιας εναλλακτικής αναπαράστασης του χώρου λύσης (π.χ. προς τα εμπρός ή προς τα πίσω αστέρι) για την εξάλειψη των μηδενικών θέσεων και τη μείωση του αριθμού των συνδυασμών. Ωστόσο, χωρίς να αποκλείεται αυτό ως δυνατότητα περαιτέρω εξερεύνησης, ειδικά για λιγότερο σύνθετες εφαρμογές, τα οφέλη μιας τέτοιας προσέγγισης θα ήταν σημαντικά μόνο σε μια εξαιρετικά διαφορετική κατάσταση ζήτησης. Αυτό δεν είναι εγγυημένο να υπάρχει, από φάση σε φάση, ακόμη και για την ίδια διασταύρωση. Σε κάθε περίπτωση διανομής ζήτησης, η αύξηση της πολυπλοκότητας και των συνολικών απαιτήσεων των οχημάτων σε διασταυρώσεις που θα αντιμετωπιστούν με τη μεθοδολογία ελέγχου θα ξεπερνούσε πιθανώς τα οφέλη μιας τέτοιας προσέγγισης. Επιπλέον, η αύξηση αυτή εξετάζεται μόνο σε μια μονοκατοικία. η λειτουργία σε αρτηριακή ή δικτυακή βάση μπορεί να καταστήσει αρκετά εξαντλητική την απαρίθμηση. Ένας σημαντικός περιορισμός, η διαθεσιμότητα του χρόνου υπολογισμού με βάση το χρονικό βήμα των επαναλήψεων και οι απαιτήσεις "πραγματικού χρόνου" της λειτουργίας του αλγορίθμου οδηγεί στην εξέταση άλλων εναλλακτικών λύσεων εξοικονόμησης χρόνου.

## **5.6. Εφαρμογή της μεθευρετικής προσέγγισης**

Η αναζήτηση Tabu είναι μια μεθευρετική διαδικασία αναζήτησης που αντιμετωπίζει πολλά από τα σχετικά ζητήματα παρέχοντας ταυτόχρονα μια σαφή λογική στη μετάβαση μεταξύ των πιθανών συνδυασμών χρονικού σημείου που πρέπει να ληφθούν υπόψη. Μια συζήτηση για την αναζήτηση tabu μπορεί να βρεθεί στο Glover (1989), αλλά η διαδικασία μπορεί να συνοψιστεί ως εξής:

1. Δημιουργείται μια αρχική λύση και η αντικειμενική της τιμή λαμβάνεται ως η τρέχουσα καλύτερη τιμή.

2. Ένας γείτονας (ή ένας υποψήφιος, από μια λίστα πιο περιοριστική από τη γειτονιά) στην αρχική λύση επιλέγεται με βάση μια κίνηση.
3. Η αντικειμενική του αξία καθορίζεται:
  1. Αν η αντικειμενική τιμή του γείτονα είναι μικρότερη από εκείνη της αρχικής λύσης, τότε αυτή η τιμή γίνεται η τρέχουσα καλύτερη τιμή.
  2. Εάν η αντικειμενική τιμή του γείτονα είναι μεγαλύτερη από εκείνη της αρχικής λύσης, τότε κάποια ιδιότητα της κίνησης γίνεται ένας περιορισμός tabu οποιασδήποτε κίνησης (δηλαδή μια κίνηση προς έναν γείτονα που συνεπάγεται αυτό το χαρακτηριστικό δεν μπορεί να γίνει για κάποιο αριθμό επαναλήψεων).
4. Εάν το 3 (a) είναι αληθές, τότε γίνεται μια κίνηση χωρίς ταμπού από την αρχική λύση. εάν το 3 (b) είναι αληθές, γίνεται κίνηση από τον γείτονα χωρίς ταμπού.
5. Οι κινήσεις εκτός του tabu μπορούν να γίνουν μέχρι:
  1. ο κατάλογος της γειτονιάς ή των υποψηφίων έχει εξαντληθεί.
  2. ένα κριτήριο αναρρόφησης (ένα που επιτρέπει μια κίνηση που θεωρείται tabu που πρέπει να γίνει) εφαρμόζεται.
  3. ένα κριτήριο στάσης (π.χ. τέλος του διαθέσιμου χρόνου υπολογισμού, ελάχιστη βελτίωση της λύσης, κλπ.) πληρούται.

Η αναζήτηση Tabu έχει εφαρμοστεί σε μια ευρεία ποικιλία ερευνητικών προβλημάτων και έχει πραγματοποιήσει αρκετές βελτιώσεις σε αυτήν, οι οποίες της επέτρεψαν να αντιμετωπίσει μάλλον περίπλοκα προβλήματα σε αυτόν τον τομέα. Μια μάλλον απλή εφαρμογή αναζήτησης tabu χρησιμοποιήθηκε ως πλαίσιο για τη βελτιστοποίηση του προσαρμοστικού ελέγχου σήματος κυκλοφορίας στην τρέχουσα ρύθμισή του. αυτό είναι παρόμοιο με αυτό των Laguna et al. (1990) για το πρόβλημα προγραμματισμού ενός μηχανήματος, το οποίο ουσιαστικά περιλαμβάνει την βασική διαδικασία που περιγράφηκε παραπάνω.

Η αναζήτηση μπορεί να ξεκινήσει με τη σειρά των τιμών χρονικών σημείων, η οποία έχει αριθμό σειρών ίσων με το μέγιστο αριθμό οχημάτων που φθάνουν σε μια φάση και με αριθμό στηλών ίσο με τον αριθμό των φάσεων που πρέπει να ληφθούν υπόψη. Επιλέγεται ένα χρονικό σημείο για κάθε  $k$  ή φάση στη συστοιχία. Στη συνέχεια, μια γειτονιά αυτής της λύσης θα μπορούσε να έχει όλες τις τιμές χρονικού σημείου που κρατούνται εκτός από μία, η οποία μπορεί να ποικίλει με βάση μια κίνηση προς τα

πάνω και προς τα κάτω σε μια συγκεκριμένη στήλη. Μια τέτοια κίνηση θα ορίζεται ως μια κίνηση χαμηλής επιρροής. Η γειτονιά θα μπορούσε επίσης να επιτρέψει στις στήλες να αλλάξουν το σύνολο των τιμών των χρονικών σημείων, έτσι ώστε οι πράσινοι χρόνοι να μπορούν να εμφανιστούν με διαφορετική σειρά κατά τη διάρκεια του ορίζοντα. Αυτό θα οριστεί ως μεγάλη κίνηση επιρροής. Η τρέχουσα υλοποίηση χρησιμοποιεί μόνο κινήσεις χαμηλής επιρροής, ακολουθώντας τον περιορισμό των αυξανόμενων διαδοχικών τιμών χρονικού σημείου. Στην τρέχουσα περίπτωση, δημιουργείται το σύνολο αναλογικών ευρετικών λύσεων (όπως περιγράφεται παρακάτω) και επιλέγεται από την αρχική λύση. Με βάση την εμφάνιση μιας μη μειούμενης καθυστέρησης, ένα σύνολο τιμών χρονικού σημείου έχει ένα από τα στοιχεία του εκχωρημένα ως tabu. Το στοιχείο και η θέση του στο καθορισμένο χρονικό σημείο τοποθετούνται σε συστοιχίες όπου παραμένουν για το μήκος της θητείας. Εάν στη συνέχεια επιλεγεί ένα χρονικό σημείο με το στοιχείο tabu στην αποθηκευμένη θέση πριν ολοκληρωθεί η θητεία του tabu, αυτό το σύνολο χρονικών σημείων απορρίπτεται και ένα άλλο χρονικό σημείο έχει μεταπηδήσει στο χώρο αναλογικής λύσης, που είναι η γειτονιά αυτής της συγκεκριμένης εφαρμογής. Η διάταξη tabu μπορεί να κρατήσει όσα ζεύγη στοιχείων-θέσης ως η αξία της κατοχής των tabu.

Τα κριτήρια έχουν μεγάλη επίδραση στην ικανότητα εξερεύνησης λύσεων, καθώς είναι πολύ πιθανό ότι μια κίνηση tabu ή μη γειτονιάς μπορεί να οδηγήσει σε μια σημαντικά βελτιωμένη λύση σε σύγκριση με την τρέχουσα βέλτιστη λύση. Τα δύο βασικά ισχύοντα κριτήρια αναρρόφησης είναι τα εξής:

- ένα κριτήριο "απόσβεση από προεπιλογή", όπου, εάν όλες οι κινήσεις είναι tabu, θα επιτραπεί μια κίνηση που οδήγησε στην αντικειμενική τιμή που βρίσκεται πιο κοντά στην τρέχουσα καλύτερη τιμή
- ένα κριτήριο "αναρρόφησης με επιρροή", όπου, εάν έχει πραγματοποιηθεί μια κίνηση "υψηλής επιρροής" από τότε που ένα χαρακτηριστικό έγινε tabu, επιτρέπονται πλέον μετακινήσεις "χαμηλής επιρροής" που συνεπάγονται αυτό το χαρακτηριστικό.

Χρησιμοποιείται εδώ μια έκδοση του κριτηρίου "απορρόφησης από προεπιλογή", όπου, αν όλες οι κινήσεις είναι tabu, επιλέγεται η επιλογή που ήταν αρχικά επόμενη στον χώρο αναλογικής λύσης.

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Η χρήση ευρετικών και μεθευρετικών προσεγγίσεων δείχνει σαφείς δυνατότητες μείωσης του χρόνου υπολογισμού για τους μεγάλους χώρους λύσεων που εμπλέκονται στη βελτιστοποίηση του ελέγχου κυκλοφορίας. Μπορούν επίσης να "σφίξουν" τις αναζητήσεις σε ένα μεγάλο χώρο λύσης σε σχετικά σύντομο χρόνο υπολογισμού. Αυτά τα οφέλη είναι κρίσιμα για να επιτρέπεται η λειτουργία περίπλοκων αλγορίθμων για τον έλεγχο του κυκλοφοριακού σήματος εντός των χρονικών περιορισμών που απαιτούνται τόσο για τη λειτουργία του εξοπλισμού όσο και για τις αλλαγές στην κυκλοφοριακή ροή. Υπάρχουν επίσης πολύτιμες για την προσαρμογή του ελέγχου προσαρμοστικού σήματος κυκλοφορίας για την ενσωμάτωση πιο ακριβών και ολοκληρωμένων δεδομένων κίνησης. Περαιτέρω έρευνα θα επιτρέψει τέτοιες ευρετικές και μεταχειριστικές προσεγγίσεις να είναι ακόμα πιο ισχυρές και βιώσιμες στο περιβάλλον ελέγχου κυκλοφοριακού σήματος.

## BIBΛIOΓΡΑΦΙΑ

1. R. Z. Farahani and M. Hekmatfar, *Facility Location: Concepts, Models, Algorithms and Case Studies*, Springer-Verlag, Berlin, Germany, 2009.
2. Weber, *Ueber den Standort der Industrien, Erster Teil: Reine Theorie des Standortes*, J.C.B. Mohr, Tübingen, Germany, 1909.
3. G. Wesolowsky, "The Weber problem: history and perspectives," *Location Science*, vol. 1, pp. 5–23, 1993.
4. Z. Drezner, K. Klamroth, A. Schöbel, and G. O. Wesolowsky, "The Weber problem," in *Facility Location*, pp. 1–36, Springer, Berlin, Germany, 2002.
5. P. S. Stanimirović, M. S. Ćirić, L. A. Kazakovtsev, and I. A. Osinuga, "Single-facility Weber location problem based on the lift metric," *Facta Universitatis. Series: Mathematics and Informatics*, vol. 27, no. 2, pp. 175–190, 2012.
6. R. F. Love, W. G. Truscott, and J. Walker, "Facilities Location: Models and Methods," *International Journal of Machine Learning and Cybernetics*, North-Holland, New York, 1988.
7. E. Weiszfeld, "Sur le point sur lequel la somme des distances de  $n$  points donnés est minimum," *Tohoku Mathematical Journal*, vol. 43, no. 1, pp. 335–386, 1937.
8. Y. Vardi and C.-H. Zhang, "A modified Weiszfeld algorithm for the Fermat-WEber location problem," *Mathematical Programming*, vol. 90, no. 3, Ser. A, pp. 559–566, 2001.
9. C. Szegedy, *Some Applications of the Combinatorial Laplacian*, University of Bonn, 2005.
10. P. Hansen, N. Mladenović, and E. Taillard, "Heuristic solution of the multisource Weber problem as a  $p$ -median problem," *Operations Research Letters*, vol. 22, no. 2-3, pp. 55–62, 1998.

11. S. Gonzalez-Martin, A. Ferrer, A. A. Juan, and D. Riera, "Solving non-smooth arc routing problems throughout biased-randomized heuristics," *Advances in Intelligent Systems and Computing*, vol. 262, pp. 451–462, 2014.
12. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic algorithms in modeling and optimization," in *Metaheuristic Applications in Structures and Infrastructures*, A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, Eds., pp. 1–24, Elsevier, 2013.
13. R. Martí and G. Reinelt, *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*, vol. 175, Springer, New York, NY, USA, 2011.
14. Afshar, F. Massoumi, A. Afshar, and M. A. Mariño, "State of the art review of ant colony optimization applications in water resource management," *Water Resources Management*, vol. 29, no. 11, pp. 3891–3904, 2015.
15. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. I. Background and development," *Natural Computing. An International Journal*, vol. 6, no. 4, pp. 467–484, 2007.
16. Fister Jr., M. Perc, S. M. Kamal, and I. Fister, "A review of chaos-based firefly algorithms: perspectives and research challenges," *Applied Mathematics and Computation*, vol. 252, pp. 155–165, 2015.
17. Fister, X. S. Yang, and D. Fister, "Cuckoo search: a brief literature review," in *Cuckoo Search and Firefly Algorithm: Theory and Applications*, Xin-She Yang, Ed., pp. 49–62, Springer International Publishing.
18. Fister, I. Fister Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, no. 1, pp. 34–46, 2013.
19. D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014.



20. M. Bilal, M. Shams-ur-Rehman, and M. A. Jaffar, "Reconstruction: Image restoration for space variant degradation," *Smart Computing Review*, vol. 3, no. 4, pp. 220–232, 2013.
21. P. Siarry, *Optimisation in Signal and Image Processing*, Wiley-ISTE, 2009.
22. X. Yang, S. F. Chien, and T. O. Ting, "Computational Intelligence and Metaheuristic Algorithms with Applications," *The Scientific World Journal*, vol. 2014, Article ID 425853
23. Fister Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *Elektrotehniški Vestnik*, vol. 80, no. 3, pp. 1–7, 2013.
24. M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
25. C. Mohan and R. Baskaran, "A survey: ant colony optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4618–4627, 2012.
26. R. F. Tavares Neto and M. Godinho Filho, "Literature review regarding ant colony optimization applied to scheduling problems: guidelines for implementation and directions for future research," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 150–161, 2013.
27. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.
28. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. {II}. Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications," *Natural Computing. An International Journal*, vol. 7, no. 1, pp. 109–124, 2008.

29. Khare and S. Rangnekar, "A review of particle swarm optimization and its applications in Solar Photovoltaic system," *Journal Applied Soft Computing*, vol. 13, no. 5, pp. 2997–3006, 2013.
30. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, pp. 120–142, 2012.
31. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization," Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
32. Baykasoglu, L. Ozbakir, and P. Tapkan, "Artificial bee colony algorithm and its application to generalized assignment problem," in *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, F. T. S. Chan and M. K. Tiwari, Eds., pp. 113–144, I-Tech Education and Publishing, Vienna, Austria, 2007
33. S. Zhang and S. Liu, "A novel artificial bee colony algorithm for function optimization," *Mathematical Problems in Engineering*, vol. 2015, Article ID 129271
34. Y. Liang, Z. Wan, and D. Fang, "An improved artificial bee colony algorithm for solving constrained optimization problems," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 3, pp. 739–754, 2017.
35. T. K. Sharma and M. Pant, "Shuffled Artificial Bee Colony Algorithm," *Soft Computing*, pp. 1–20, 2016.
36. X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Computer Science*, pp. 169–178, Springer, Berlin, 2009.
37. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," *Computers and Structures*, vol. 89, no. 23-24, pp. 2325–2336, 2011

38. M. Huang, J. Yuan, and J. Xiao, "An adapted firefly algorithm for product development project scheduling with fuzzy activity duration," *Mathematical Problems in Engineering*, vol. 2015, Article ID 973291
39. X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, Coimbatore, India, December 2009.
40. Y. Lin, C. Zhang, and Z. Liang, "Cuckoo search algorithm with hybrid factor using dimensional distance," *Mathematical Problems in Engineering*, vol. 2016.