



**ΓΕΩΠΟΝΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ:**

**ΘΕΤΙΚΕΣ ΕΠΙΣΤΗΜΕΣ ΣΤΗ ΓΕΩΠΟΝΙΑ ΚΑΙ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ**

**ΚΛΑΔΟΣ Ι: ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ**

**Προσωποποιημένη πλοήγηση  
σε εξωτερικούς χώρους:  
ανασκόπηση αλγορίθμων  
και μεθόδων επιλογής μονοπατιού**

**ΚΥΒΕΛΗ ΚΩΝ/ΝΑ ΔΙΑΡΕΜΕ**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:**

**ΘΕΟΔΩΡΟΣ ΤΣΙΛΙΓΚΙΡΙΔΗΣ**

**ΑΘΗΝΑ 2013**

## **ΕΠΙΤΡΟΠΗ**

**Επιβλέπων καθηγητής: ΤΣΙΛΙΓΚΙΡΙΔΗΣ ΘΕΟΔΩΡΟΣ**

**ΓΙΑΛΟΥΡΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ**

**ΚΩΣΤΟΠΟΥΛΟΥ ΚΩΝΣΤΑΝΤΙΝΑ**

## Ευχαριστίες

Ευχαριστώ πολύ τους καθηγητές του Μεταπτυχιακού προγράμματος «ΘΕΤΙΚΕΣ ΕΠΙΣΤΗΜΕΣ ΣΤΗ ΓΕΩΠΟΝΙΑ ΚΑΙ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ, ΚΛΑΔΟΣ Ι: ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ» για τις γνώσεις που μου έδωσαν, τα μέλη της επιτροπής μου, κα. Κωστοπούλου και κ. Γιαλούρη, καθώς και τα μέλη του Εργαστηρίου της Πληροφορικής του Γεωπονικού Πανεπιστημίου Αθηνών.

Κυρίως θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Τσιλιγκιρίδη, για την εμπιστοσύνη του και τη συνεχή βοήθεια, καθοδήγηση και υποστήριξη που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης της Μεταπτυχιακής Μελέτης μου.

Τέλος, ευχαριστώ θερμά τη μητέρα μου για την υποστήριξη της.

Μέρος της παρούσας μελέτης θα δημοσιευθεί στα Proceedings του συνεδρίου “2<sup>nd</sup> International Symposium and 24<sup>th</sup> National Conference on Operational Research, Athens 2013”.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	6
1. ΕΙΣΑΓΩΓΗ.....	7
2. ΥΠΟΒΑΘΡΟ: ΕΝΝΟΙΕΣ ΚΑΙ ΟΡΙΣΜΟΙ .....	9
2.1 ΠΡΟΒΛΗΜΑΤΑ ΑΠΟΦΑΣΗΣ - ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ .....	9
2.2 ΣΥΣΤΗΜΑΤΑ ΥΠΟΣΤΗΡΙΞΗΣ ΑΠΟΦΑΣΕΩΝ .....	9
2.3 ΣΥΣΤΗΜΑΤΑ ΠΡΟΤΑΣΕΩΝ - (Recommender Systems) .....	10
2.4 ΥΠΗΡΕΣΙΕΣ ΘΕΣΗΣ - (Location Based Services) .....	10
2.5 ΘΕΩΡΙΑ ΤΩΝ ΓΡΑΦΩΝ .....	11
HAMILTONIAN ΜΟΝΟΠΑΤΙ.....	13
2.6 ΘΕΩΡΙΑ ΤΗΣ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ .....	14
2.7 ΑΛΓΟΡΙΘΜΟΙ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ.....	16
ΑΚΡΙΒΕΙΣ ΑΛΓΟΡΙΘΜΟΙ - (Exact algorithms) .....	16
ΠΡΟΣΕΓΓΙΣΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ - (Approximate algorithms).....	17
ΑΠΛΗΣΤΟΙ ΑΛΓΟΡΙΘΜΟΙ (Greedy algorithms) .....	18
ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΜΕΘΟΔΟΙ ΕΠΙΛΟΓΗΣ ΜΟΝΟΠΑΤΙΟΥ.....	20
3. ΠΡΟΣΩΠΟΠΟΙΗΜΕΝΗ ΠΛΟΗΓΗΣΗ.....	25
3.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ .....	25
3.2 ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΛΑΝΟΔΙΟΥ ΠΩΛΗΤΗ.....	28
ΟΡΙΣΜΟΣ ΤΟΥ TSP ΣΤΟΝ ΓΡΑΜΜΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ...	29
3.3 ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ (ΜΕ ΕΝΑ ΠΑΙΧΤΗ, ΟΜΑΔΙΚΟ) .....	30
ΠΡΟΒΛΗΜΑ ΤΟΥ ΟΜΑΔΙΚΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ .....	30
ΟΡΙΣΜΟΣ ΤΟΥ OP ΣΤΟΝ ΓΡΑΜΜΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ.....	32
3.4 ΑΝΑΣΚΟΠΗΣΗ ΠΡΟΗΓΟΥΜΕΝΗΣ ΕΡΕΥΝΑΣ.....	33
3.5 ΑΠΛΟ ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΣΩΠΟΠΟΙΗΜΕΝΗΣ ΠΛΟΗΓΗΣΗΣ.....	36
4. ΕΠΙΣΚΟΠΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΚΑΙ ΔΙΑΔΙΚΑΣΙΩΝ .....	42
4.1 ΕΥΡΕΣΗ ΒΕΛΤΙΣΤΗΣ ΔΙΑΔΡΟΜΗΣ .....	42

ΚΑΤΑΣΚΕΥΗ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ .....	42
ΒΕΛΤΙΩΣΗ ΤΗΣ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ.....	46
4.2 ΑΛΓΟΡΙΘΜΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ .....	50
ΣΤΟΧΑΣΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ .....	50
ΑΛΓΟΡΙΘΜΟΣ ΒΑΣΗ ΤΗΣ ΠΟΛΥΚΡΙΤΗΡΙΑΚΗΣ ΑΝΑΛΥΣΗΣ ..	51
ΑΛΓΟΡΙΘΜΟΣ ΒΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ ΣΑΚΙΔΙΟΥ (Knapsack) .....	52
5. ΠΡΟΤΕΙΝΟΜΕΝΟΣ ΑΛΓΟΡΙΘΜΟΣ.....	53
5.1 ΠΑΡΟΥΣΙΑΣΗ ΑΛΓΟΡΙΘΜΟΥ .....	54
ΚΑΤΑΣΚΕΥΗ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ .....	55
ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ.....	58
5.2 ΑΞΙΟΛΟΓΗΣΗ ΑΛΓΟΡΙΘΜΟΥ .....	61
ΣΥΝΟΛΑ ΣΗΜΕΙΩΝ ΓΙΑ ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ..	62
ΠΙΝΑΚΕΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....	63
5.3 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	67
6. ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ - ΠΡΟΤΑΣΕΙΣ .....	68
7. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	69
8. ΣΥΝΔΕΣΜΟΙ.....	73
9. ΠΑΡΑΡΤΗΜΑ.....	74

## ΠΕΡΙΛΗΨΗ

Η προσωποποιημένη πλοήγηση αφορά στην επιλογή και χάραξη της βέλτιστης διαδρομής βάσει των προτιμήσεων ενός ατόμου. Σήμερα η διαδικασία πραγματοποιείται μέσα από διαδικτυακές εφαρμογές συστημάτων λήψης απόφασης που μπορεί να τις χειρίζεται ο χρήστης από έναν ηλεκτρονικό υπολογιστή ή κινητό τηλέφωνο. Με βάση τη θέση του χρήστη ή το σημείο έναρξης της διαδρομής, τα διαθέσιμα σημεία ενδιαφέροντος (Point of Interest (POIs)) αλλά και τις δηλωθείσες προτιμήσεις του, ο χρήστης πληροφορείται για τη διαδρομή που πρέπει να ακολουθήσει. Η διαδρομή αυτή θα παρέχει τη μέγιστη χρησιμότητα μέσα στους περιορισμούς που το ίδιο το άτομο θέτει. Η παρούσα εργασία επικεντρώνεται στη χρήση της προσωποποιημένης πλοήγησης για τουρίστες.

Η προσωποποιημένη πλοήγηση δεν στοχεύει μόνο στη χάραξη μιας βέλτιστης διαδρομής αλλά επιτρέπει σε κάθε χρήστη να προσθέτει τις προτιμήσεις του και τους περιορισμούς του. Προβλήματα αυτής της κατηγορίας δεν μπορούν να λυθούν από ακριβείς αλγόριθμους όμως πλήθος ευρετικών, μεταευρετικών αλγορίθμων και αλγορίθμων τοπικής αναζήτησης που απαντώνται στην βιβλιογραφία μπορούν να χρησιμοποιηθούν για να δώσουν ως αποτέλεσμα μια καλή, αν όχι βέλτιστη, προσωποποιημένη διαδρομή.

Σκοπός της παρούσας μελέτης είναι η παρουσίαση της προσωποποιημένης πλοήγησης, στοχευμένη σε τουριστικές εφαρμογές, καθώς και των βασικών εννοιών που συνδέονται με αυτή. Παρουσιάζεται το ειδικότερο πρόβλημα της χάραξης της βέλτιστης διαδρομής που πραγματεύεται η παρούσα μελέτη και των μεθόδων και αλγορίθμων που μπορούν να χρησιμοποιηθούν για να προσεγγίσουν τη βέλτιστη λύση. Τέλος εξηγείται η συνάφειά του προβλήματος με το πρόβλημα του Προσανατολισμού (Orienteering Problem (OP)), καθώς και με το συνηθέστερο πρόβλημα του Ομαδικού Προσανατολισμού (Team Orienteering Problem (TOP)). Ως βάση για μελλοντική εργασία προτείνεται ένας νέος αλγόριθμος για το OP και για την ειδικότερη περίπτωση που ένας τουρίστας ξεκινά από ένα, όχι πάντα το ίδιο, σημείο και καταλήγει πίσω σε αυτό. Ο αλγόριθμος χρησιμοποιεί μεθοδολογία που βασίζεται στο TOP για την κατασκευή μιας αρχικής διαδρομής και στη συνέχεια μεθοδολογία που βασίζεται στο OP για να βελτιωθεί η διαδρομή. Ο αλγόριθμος εφαρμόζεται σε τρία πολύ γνωστά σετ δεδομένων και συγκρίνεται με 4 αλγόριθμους της βιβλιογραφίας.

## 1. ΕΙΣΑΓΩΓΗ

Ο τομέας του Τουρισμού, ή κατά άλλους οι «βιομηχανία» του τουρισμού, θεωρείται από τους πιο παραγωγικούς και τους πιο κερδοφόρους παγκοσμίως. Σύμφωνα με την τελευταία έκδοση του Παγκόσμιου Βαρομέτρου Τουρισμού του Παγκόσμιου Οργανισμού Τουρισμού (ΠΟΤ) των Ηνωμένων Εθνών (World Tourism Organization UNWTO) το 2012, παρά την οικονομική κρίση που ταλαιπωρεί μεγάλο κομμάτι του παγκόσμιου πληθυσμού και κυρίως την Ευρωζώνη, οι αφίξεις σε παγκόσμιο επίπεδο αυξήθηκαν κατά 4% και ξεπέρασαν το ένα δισεκατομμύριο, αριθμός ρεκόρ για τα μέχρι σήμερα δεδομένα. Το 2012 ταξίδεψαν τον κόσμο 1.034 εκατομμύρια τουρίστες και ο ΠΟΤ προβλέπει ότι οι διεθνείς τουριστικές αφίξεις θα ενισχυθούν από 3% έως 4% μέχρι το τέλος του 2013, συμβαδίζοντας σε μεγάλο βαθμό με την μακροπρόθεσμη πρόβλεψη του κατά την οποία προσδοκάται ετήσια αύξηση 3,8% κατά μέσο όρο μεταξύ 2010 και 2020. Περίπου το 13% του Παγκόσμιου Ακαθάριστου Προϊόντος οφείλεται στον τουρισμό.

Από τα παραπάνω στοιχεία γίνονται αντιληπτά τα οικονομικά οφέλη που μπορεί να αποφέρει ο τομέας του τουρισμού σε μία περιοχή και ειδικά σε μία χώρα που αποτελεί κατεξοχήν τουριστικό προορισμό, με μεγάλο πολιτισμό, όπως η Ελλάδα. Επίσης, τα παραπάνω αιτιολογούν την αύξηση του αριθμού των παρεχόμενων υπηρεσιών στους τουρίστες αλλά και την προσπάθεια εύρεσης νέων καινοτόμων λύσεων που στοχεύουν στο να διευκολύνουν, ομαλοποιούν, ομορφαίνουν τις διακοπές των τουριστών και τους βοηθούν να αξιοποιούν καλύτερα τον χρόνο τους.

Όταν οι τουρίστες βρίσκονται σε ένα προορισμό, συνήθως αναζητούν πληροφορίες στα τουριστικά πρακτορεία. Εκεί το προσωπικό καλείται να αποφασίσει το «προφίλ» του τουρίστα και τους περιορισμούς του. Συνδυάζοντας αυτές τις πληροφορίες με τις γνώσεις τους για την περιοχή, τα θεάματα, τα μέσα μαζικής μεταφοράς και άλλες σχετικές πληροφορίες, οι υπάλληλοι των πρακτορείων, προτείνουν μία προσωποποιημένη διαδρομή στον τουρίστα. Μία διαδρομή που θα ικανοποιήσει τις ανάγκες του και θα ταιριάζει στις επιθυμίες του.

Στις μοντέρνες κοινωνίες και ειδικότερα σε μεγάλες πόλεις που αποτελούν πόλους έλξης τουριστών η πληθώρα των παρεχόμενων υπηρεσιών αλλά και η ραγδαία αύξηση του όγκου της πληροφορίας πολλές φορές δεν επιτρέπουν σε ένα άτομο – καταναλωτή – τουρίστα να βρει εύκολα αυτό που ψάχνει και φυσικά δεν μπορεί να επισκεφτεί όλα τα μέρη, σημεία ενδιαφέροντος λόγω του περιορισμένου χρόνου και χρήματος. Επίσης οι

διαφορετικοί περιορισμοί του κάθε τουρίστα αλλά και οι διαφορετικές προτιμήσεις είναι πάρα πολλές (π.χ. το σημείο που βρίσκεται, το σημείο που θέλει να καταλήξει, η ώρα της μέρας, ο καιρός, ο χρόνος και τα χρήματα που έχει στην διάθεση του). Οι Kramer et al. το 2006 προσπάθησαν να ομαδοποιήσουν τους περιορισμούς των τουριστών και να καταλήξουν σε συγκεκριμένα προφίλ. Μέσα από αυτή την προσπάθεια κατέληξαν στο συμπέρασμα ότι οι εν δυνάμει περιορισμοί και προτιμήσεις του κάθε ατόμου είναι τόσοι πολλοί και διαφορετικοί που πλέον είναι προτιμότερο να προσαρμόζεται μία διαδρομή στις ανάγκες ενός ατόμου παρά να χαράζονται γενικές διαδρομές που ο κάθε τουρίστας θα καλείται να ακολουθήσει. Έτσι, σήμερα, αναπτύσσονται, ολοένα και περισσότερο γεωπληροφοριακά συστήματα στα οποία ενσωματώνονται έξυπνες εφαρμογές και υπηρεσίες που αναλαμβάνουν τον παραπάνω ρόλο, κάνοντας χρήση διαφόρων τεχνολογιών, για να παρέχουν την καλύτερη δυνατή πληροφόρηση, με εγκυρότητα και πληρότητα στο συντομότερο δυνατό χρονικό διάστημα.



## **2. ΥΠΟΒΑΘΡΟ: ΕΝΝΟΙΕΣ ΚΑΙ ΟΡΙΣΜΟΙ**

Στο κεφάλαιο αυτό παρουσιάζονται εισαγωγικές έννοιες, τεχνολογίες και ορισμοί που σχετίζονται με την προσωποποιημένη πλοήγηση, αναφέρονται σε πολλά σημεία του κειμένου και η κατανόηση τους θα διευκολύνει τον αναγνώστη.

### **2.1 ΠΡΟΒΛΗΜΑΤΑ ΑΠΟΦΑΣΗΣ - ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ**

Στη θεωρία της επιστήμης των υπολογιστών ως πρόβλημα απόφασης χαρακτηρίζεται κάθε πρόβλημα που μπορεί να πάρει μόνο δύο τιμές, «ΝΑΙ» ή «ΟΧΙ».

Ως πρόβλημα βελτιστοποίησης αναφέρεται κάθε πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης της συνάρτησης μίας μεταβλητής ή πολλών μεταβλητών. Τα προβλήματα αυτά είτε υπόκειται σε περιορισμούς, ισότητας ή ανισότητας, είτε είναι χωρίς περιορισμούς.

### **2.2 ΣΥΣΤΗΜΑΤΑ ΥΠΟΣΤΗΡΙΞΗΣ ΑΠΟΦΑΣΕΩΝ (Decision Support Systems)**

Τα συστήματα υποστήριξης αποφάσεων είναι αλληλεπιδραστικά συστήματα λογισμικού που αποσκοπούν στη βελτίωση της αποτελεσματικότητας των διαδικασιών λήψης αποφάσεων σε χώρους προβλημάτων που χαρακτηρίζονται από χαμηλό βαθμό δόμησης. Στα πλαίσια ενός τέτοιου συστήματος, ο χρήστης υποστηρίζεται από αναλυτικές μεθόδους και μοντέλα για να θέτει στόχους και να ορίζει εναλλακτικά σενάρια (λύσεις), να αναλύει τις επιπτώσεις τους, να αξιολογεί τις εναλλακτικές λύσεις και τελικά να επιλέγει την κατάλληλη λύση που θα εφαρμοστεί.

Σχεδιάζονται για να διευκολύνουν και να ενισχύουν τη συμμετοχή του χρήστη στις διαδικασίες λήψης αποφάσεων και υποστηρίζουν χωρίς να αντικαθιστούν την κρίση του, παρέχοντάς πρόσβαση σε βάσεις δεδομένων και αναλυτικά-ποσοτικά μοντέλα. Μία συνηθισμένη χρήση των συστημάτων αυτών είναι για πρόβλεψη αποτελεσμάτων ή καταστάσεων μέσω χρονοσειρών.

### **2.3 ΣΥΣΤΗΜΑΤΑ ΠΡΟΤΑΣΕΩΝ - (Recommender Systems)**

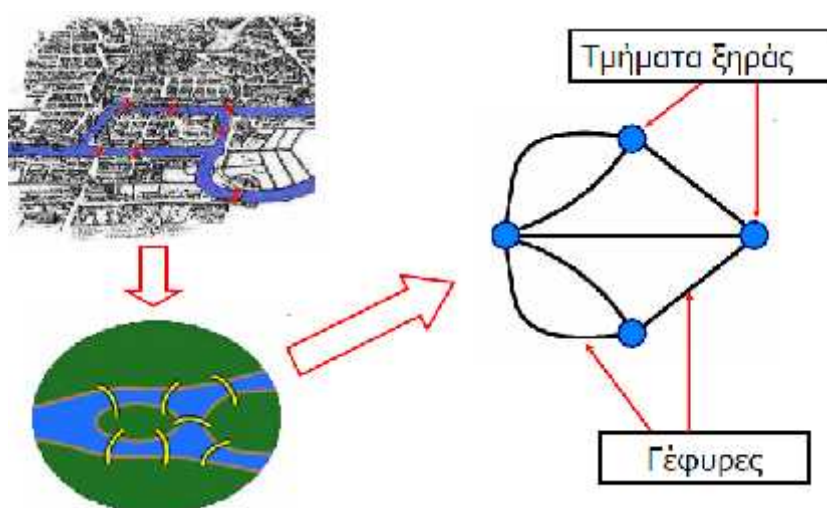
Πρόκειται για συστήματα που διευκολύνουν τη λήψη απόφασης μέσα στην πληθώρα πληροφορίας που υπάρχει σήμερα. Οι χρήστες διαδικτυακών υπηρεσιών και πλατφορμών συχνά έρχονται αντιμέτωποι με τεράστια ποσότητα προϊόντων που τους παρέχονται ή έρχονται αντιμέτωποι με πληθώρα δυνατών λύσεων. Τα συστήματα προτάσεων παρέχουν στον χρήστη πληροφορίες για να μπορέσουν να τον διευκολύνουν στην επιλογή του. Οι πληροφορίες αυτές μπορεί να είναι μεταδεδομένα, αξιολογήσεις των πιθανών λύσεων από άλλους χρήστες. Τα συστήματα προτάσεων χρησιμοποιούνται ευρέως στο ηλεκτρονικό εμπόριο από την δεκαετία του '90.

### **2.4 ΥΠΗΡΕΣΙΕΣ ΘΕΣΗΣ - (Location Based Services)**

Οι Υπηρεσίες Θέσης είναι υπηρεσίες που ενσωματώνουν την γεωγραφική θέση του χρήστη. Μέσα από τα δίκτυα κινητής τηλεφωνίας ή και μέσω δορυφόρων, οι υπηρεσίες αυτές, μπορούν να κάνουν χρήση της θέσης της κινητής συσκευής του χρήστη ενώ διάφορες εφαρμογές μπορούν να χρησιμοποιήσουν την πληροφορία αυτή για να παρέχουν απαντήσεις. Για παράδειγμα ο χρήστης μπορεί να δει που ακριβώς βρίσκεται, τι υπάρχει γύρω του, να αλληλεπιδράσει με άλλους χρήστες, να μάθει που μπορούν να βρει κάτι που θέλει, όπως ένα φαρμακείο ή εστιατόριο. Ακόμη είναι δυνατόν να παρέχονται και πληροφορίες όπως το τι ώρα είναι, το ιστορικό θέσεων του χρήστη κ.α.

## 2.5 ΘΕΩΡΙΑ ΤΩΝ ΓΡΑΦΩΝ

Η θεωρία των γραφών είναι ένα γνωστικό πεδίο των διακριτών μαθηματικών, με εφαρμογές στην πληροφορική, στις επιστήμες των μηχανικών, στη χημεία, στην κοινωνιολογία κ.α. Με γραφούς μπορούν να μοντελοποιηθούν πολλές διαφορετικές φυσικές ή τεχνολογικές δομές, όπως τα δίκτυα υπολογιστών, το οδικό δίκτυο μίας πόλης, οι γέφυρες μίας πόλης (Εικόνα 1) κ.α. Η θεωρία των γραφών ασχολείται με τη μελέτη των γραφών (γραφημάτων) και των σχέσεών τους.

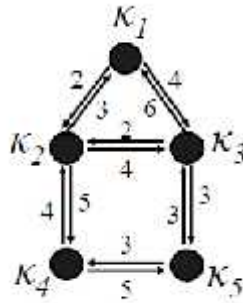


Εικόνα 1: Μεταφορά σε γράφο των γεφυρών του Königsberg

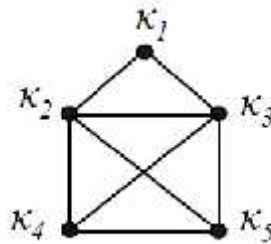
Στην Εικόνα 1 απεικονίζεται η μεταφορά σε γράφο των γεφυρών του Königsberg. Για την ιστορία της θεωρίας των γραφών θεωρείται σημαντική η μελέτη του Euler (1736) για τις Επτά Γέφυρες του Königsberg. Ο τύπος του Euler, σχετικά με τον αριθμό των ακμών, των κορυφών και των εδρών ενός κυρτού πολυέδρου μελετήθηκε από τους Augustin Louis Cauchy, Simon Antoine Jean L'Huilier και αποτελεί την αρχή της τοπολογίας, του κλάδου των μαθηματικών που μελετά, ανάμεσα σε άλλα, τις ιδιότητες εκείνες των γεωμετρικών σχημάτων οι οποίες παραμένουν αναλλοίωτες κατά τις τοπολογικές απεικονίσεις.

Ο γράφος είναι ένα σύνολο από κόμβους (κορυφές) που ενώνονται μεταξύ τους με ακμές και ορίζεται από τον τρόπο με τον οποίο συνδέονται οι

κορυφές. Αν οι ακμές προσανατολίζονται οριζόμενες από διατεταγμένα ζεύγη κόμβων, τότε ο γράφος αποκαλείται κατευθυνόμενος ή δίγραφος (Εικόνα 2). Αν οι ακμές δεν προσανατολίζονται, οριζόμενες απλώς από διμελή σύνολα και όχι διατεταγμένα ζεύγη, τότε αποκαλείται μη κατευθυνόμενος (Εικόνα 3). Επιπλέον στοιχεία για τον ορισμό ενός γράφου είναι η σύνδεση των ακμών του με κάποια αξία, οπότε αποκαλείται σταθμισμένος (Εικόνα 2). Τέλος ένας γράφος ονομάζεται συνδεδεμένος αν υπάρχει μονοπάτι μεταξύ δύο οποιονδήποτε κόμβων του.



Εικόνα 2: Κατευθυνόμενος σταθμισμένος γράφος με 5 κόμβους.



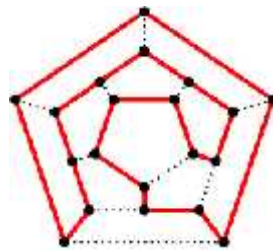
Εικόνα 3: Μη κατευθυνόμενος γράφος, όχι σταθμισμένος με 5 κόμβους.

Ακόμη στη θεωρία των γράφων μπορούμε να διακρίνουμε τις παρακάτω έννοιες (R. Diestel 2010):

- Διαδρομή: ακολουθία από κορυφές και ακμές, που αρχίζει και τελειώνει με κορυφή.
- Ίχνος: διαδρομή όπου κάθε ακμή δεν εμφανίζεται πάνω από μία φορά.
- Μονοπάτι: ίχνος όπου μια κορυφή δεν εμφανίζεται πάνω από μία φορά (δεν τέμνεται με τον εαυτό του), πχ. το Hamiltonian μονοπάτι.

## HAMILTONIAN ΜΟΝΟΠΑΤΙ

Με τον όρο «Hamiltonian μονοπάτι» αναφερόμαστε σε ένα μονοπάτι που σε ένα γράφο με κατευθύνσεις θα επισκεφτεί κάθε κόμβο μόνο μία φορά. Δεν γνωρίζουμε ένα γρήγορο τρόπο για να βρούμε ένα τέτοιο μονοπάτι αλλά μπορούμε να ελέγξουμε εύκολα εάν ένα δοθέν μονοπάτι είναι Hamiltonian. Στην Εικόνα 4 παρουσιάζεται ένα τυπικό μονοπάτι τέτοιου είδους για ένα δωδεκάεδρο.



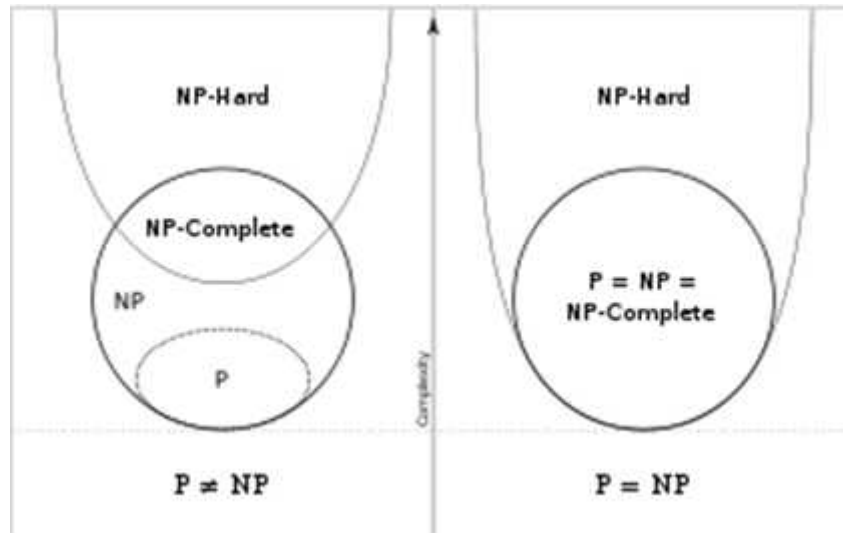
Εικόνα 4: Hamiltonian μονοπάτι για δωδεκάεδρο.

## 2.6 ΘΕΩΡΙΑ ΤΗΣ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ

Κάθε πρόβλημα αποτελείται από ένα σύνολο δεδομένων. Συνδυάζοντας τα δεδομένα αυτά προκύπτει η βέλτιστη λύση. Πολλά, όμως, προβλήματα καθίστανται δυσεπίλυτα όσο μεγαλώνει ο αριθμός των διαφορετικών συνδυασμών που πρέπει να εξεταστούν και είναι αδύνατο να λυθούν σε λογικό χρόνο (πολυωνυμικό χρόνο).

Η θεωρία της πολυπλοκότητας ασχολείται με την κοστολόγηση των πόρων που απαιτούνται για την αλγοριθμική επίλυση ενός προβλήματος. Αποτελεί βασικό δομικό λίθο της ανάλυσης αλγορίθμων και κεντρικό γνωστικό πεδίο της επιστήμης υπολογιστών. Οι συνηθέστεροι πόροι για τους οποίους ενδιαφερόμαστε είναι ο χρόνος, οπότε μιλάμε για τη χρονική πολυπλοκότητα του αλγορίθμου, δηλαδή πόσα «βήματα» χρειάζεται να εκτελέσει ο αλγόριθμος συναρτήσει της εισόδου του, και ο χώρος, οπότε μιλάμε για τη χωρική πολυπλοκότητα, δηλαδή πόσο χώρο (μνήμη) χρειάζεται ο αλγόριθμος συναρτήσει της εισόδου του. Εκτός από αυτούς τους πόρους, κατά περίπτωση, μπορεί να ενδιαφερόμαστε και για άλλους, όπως για παράδειγμα πόσοι παράλληλοι επεξεργαστές χρειάζονται για να λυθεί ένα πρόβλημα.

Η θεωρία πολυπλοκότητας ταξινομεί τα προβλήματα σε κλάσεις (σύνολα) ισοδυναμίας που ορίζουν ότι τα προβλήματα στην ίδια κλάση έχουν την ίδια δυσκολία. Ιδιαίτερου ενδιαφέροντος στο πλαίσιο αυτό είναι οι κλάσεις P (Deterministic Polynomial Time) και NP (Non Deterministic Polynomial Time). Γενικά θα μπορούσαμε να πούμε ότι η κλάση P περιλαμβάνει τα περισσότερα προβλήματα της NP (Εικόνα 5). Θα πρέπει να τονίσουμε ότι οι κλάσεις P και NP ορίζονται ως προς προβλήματα απόφασης, δηλαδή προβλήματα στα οποία καλούμαστε να απαντήσουμε μια συγκεκριμένη ερώτηση με ΝΑΙ ή ΟΧΙ.



Εικόνα 5: Διάγραμμα του Euler για P, NP, NP - complete, NP - hard προβλήματα. Το αριστερό διάγραμμα ισχύει για P διάφορο του NP ενώ το δεξί διάγραμμα ισχύει για P=NP

Στην Εικόνα 5 παρουσιάζεται το διάγραμμα του Euler για P, NP, NP - complete, NP - hard προβλήματα. Ένα πρόβλημα απόφασης είναι NP - πλήρες (NP-complete) εάν ανήκει στην κλάση των NP και ταυτόχρονα στην κλάση των NP - δύσκολων προβλημάτων. Βασικό χαρακτηριστικό των προβλημάτων αυτών είναι ότι καμία γρήγορη λύση δεν είναι γνωστή. Χαρακτηριστικό πρόβλημα της κατηγορίας αυτής: Hamiltonian μονοπάτι.

NP πλήρες (NP - complete): προβλήματα απόφασης που ανήκουν στην κλάση των NP και ταυτόχρονα στην κλάση των NP - δύσκολων προβλημάτων. Βασικό χαρακτηριστικό των προβλημάτων αυτών είναι ότι καμία γρήγορη λύση δεν είναι γνωστή. Χαρακτηριστικό πρόβλημα της κατηγορίας αυτής είναι η εύρεση ενός Hamiltonian μονοπατιού.

NP δύσκολο (NP - hard): προβλήματα όχι πάντα απόφασης, είναι τουλάχιστον τόσο δύσκολα όσο το δυσκολότερο πρόβλημα στην κλάση NP ενώ το ίδιο το πρόβλημα μπορεί να μην ανήκει στην κλάση NP.

NP εύκολο (NP - easy): προβλήματα όχι πάντα απόφασης, είναι το πολύ τόσο δύσκολα όσο το δυσκολότερο πρόβλημα στην κλάση NP ενώ το ίδιο το πρόβλημα μπορεί να μην ανήκει στην κλάση NP.

## 2.7 ΑΛΓΟΡΙΘΜΟΙ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ

Δοθέντος ενός προβλήματος, ένας αλγόριθμος είναι μία καλά προσδιορισμένη διαδικασία που παρέχει ακριβείς οδηγίες σύμφωνα με τις οποίες τα δεδομένα του προβλήματος μετασχηματίζονται και συνδυάζονται για να προκύψει η λύση του.

### ΑΚΡΙΒΕΙΣ ΑΛΓΟΡΙΘΜΟΙ - (Exact algorithms)

Οι ακριβείς μέθοδοι επίλυσης προβλημάτων ή αλγόριθμοι προσδιορίζουν τη βέλτιστη λύση ενός προβλήματος ελέγχοντας όλες τις πιθανές λύσεις του και αποφασίζοντας ποιά είναι η καλύτερη. Συνήθως επιλύουν προβλήματα περιορισμένων διαστάσεων και μικρής πολυπλοκότητας ενώ σε διαφορετικές περιπτώσεις μπορούν να οδηγήσουν σε απαγορευτικούς για τη χρήση τους χρόνους. Έστω ένα πρόβλημα με  $k$  στοιχεία τότε υπάρχουν  $k!$  διαφορετικές τοποθετήσεις των στοιχείων που θα πρέπει να εξεταστούν και να αξιολογηθούν για να προκύψει η βέλτιστη λύση.

Χαρακτηριστικό παράδειγμα αποτελεί ο αλγόριθμος διακλάδωσης και αποκοπής (branch and cut). Στην ουσία πρόκειται για έναν αλγόριθμο διακλάδωσης και φραγής (branch and bound) που μπορεί να χρησιμοποιηθεί σε περιπτώσεις ακέραιου γραμμικού προγραμματισμού καθώς χρησιμοποιεί την μέθοδο Simplex.

#### 2.7.1.1 ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΚΛΑΔΩΣΗΣ ΚΑΙ ΦΡΑΓΗΣ (*branch and bound algorithm*)

Ο αλγόριθμος διακλάδωσης και φραγής χρησιμοποιείται για να βρίσκει βέλτιστες λύσεις σε προβλήματα βελτιστοποίησης. Απαριθμεί με συστηματικό τρόπο όλες τις πιθανές λύσεις και απορρίπτει συλλήβδην λύσεις που βρίσκονται κάτω ή πέρα από προκαθορισμένα όρια. Η βασική του ιδέα είναι η απαλοιφή τμημάτων του χώρου αναζήτησης μέσα στα οποία γνωρίζουμε ότι δεν μπορεί να υπάρξει λύση.

Κατά την εφαρμογή του αλγορίθμου το σύνολο των εφικτών λύσεων διαμερίζεται σε πολλά απλούστερα υποσύνολα. Σε κάθε στάδιο του αλγορίθμου, κάποιο υποσύνολο επιλέγεται και γίνεται προσπάθεια να βρεθεί



η καλύτερη εφικτή λύση που περιλαμβάνεται σε αυτό. Εάν αυτή βρεθεί τότε το υποσύνολο λέγεται πως είναι αξιολογημένο. Εάν δεν βρεθεί, τότε το υποσύνολο διαμερίζεται εκ νέου σε δύο (ή και περισσότερα) απλούστερα υποσύνολα και η διαδικασία επαναλαμβάνεται. Κάθε σύνολο χαρακτηρίζεται από ένα άνω και ένα κάτω όριο. Εάν η λύση που βρίσκεται σε ένα σύνολο είναι μεγαλύτερη από το κάτω όριο ενός υποσυνόλου του τότε διαγράφεται το σύνολο αυτό από τον χώρο των λύσεων και συνεχίζεται η αναζήτηση σε άλλο σύνολο. Η διαδικασία επαναλαμβάνεται μέχρις ότου να βρεθεί μία μοναδική λύση, μείνει μόνο ένα σύνολο.

### **ΠΡΟΣΕΓΓΙΣΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ - (Approximate algorithms)**

Σε αντίθεση με τους ακριβείς αλγορίθμους οι προσεγγιστικοί δεν εξετάζουν όλους τους πιθανούς συνδυασμούς για να βρουν τη βέλτιστη λύση αλλά μέσα από διαδικασίες προσπαθούν να προσεγγίσουν την καλή, βέλτιστη ή σχεδόν βέλτιστη λύση. Ένας προσεγγιστικός αλγόριθμος υπολογίζει σε πολυωνυμικό χρόνο μία λύση η οποία είναι εγγυημένα «κοντά» στη βέλτιστη λύση. Οι προσεγγιστικοί αλγόριθμοι χρησιμοποιούνται σε σύνθετα και μεγάλα προβλήματα όπου η συστηματική εξέταση όλων των πιθανών καταστάσεων θα οδηγούσε σε απαγορευτικούς χρόνους. Διακρίνονται στους ευρετικούς, μετά - ευρετικούς και υπερ - ευρετικούς αλγόριθμους.

#### **2.7.1.2 ΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ - (Heuristics)**

Οι ευρετικοί ή αλλιώς ευρεστικοί ή ευρεστικοί αλγόριθμοι είναι προσεγγιστικοί αλγόριθμοι που χρησιμοποιούν πρακτικούς κανόνες για την επίλυση προβλημάτων αναζητώντας ποιοτικές εφικτές λύσεις που ικανοποιούν τους περιορισμούς ενός προβλήματος. Αν και οι ευρετικοί αλγόριθμοι δίνουν απλές και ικανοποιητικές λύσεις σε μερικά προβλήματα, τίποτα δεν εγγυάται ότι αυτές οι λύσεις είναι οι καλύτερες δυνατές. Οι αλγόριθμοι αυτοί εξερευνούν μέρος ή το μεγαλύτερο μέρος του χώρου λύσεων ώστε να καταλήξουν σε καλή λύση σε αποδεκτό χρονικό διάστημα και συνήθως τερματίζουν με την εύρεση κάποιας τοπικής βέλτιστης λύσης.

### 2.7.1.3 ΜΕΤΑΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ - (Metaheuristics)

Οι μεταευρετικοί αλγόριθμοι αποτελούν γενικές στρατηγικές αναζήτησης και σκοπός τους είναι να κατευθύνουν με κατάλληλο τρόπο ευρετικές μεθόδους ώστε να μπορούν να εφαρμοστούν σε διάφορα προβλήματα και όχι μόνο σε ειδικές, συγκεκριμένες περιπτώσεις. Οι μεταευρετικοί αλγόριθμοι ερευνούν λύσεις και σε μη εφικτές περιοχές προς αποφυγήν του εγκλωβισμού του αλγόριθμου σε τοπικά ελάχιστα.

Από τους πιο δημοφιλείς μεταευρετικούς αλγορίθμους είναι οι αλγόριθμοι τοπικής αναζήτησης και οι γενετικοί αλγόριθμοι και παρουσιάζονται στη συνέχεια του κεφαλαίου.

Εκτός των μεταευρετικών αλγορίθμων υπάρχουν και οι υπερεευρετικοί (hyper-heuristics) οι αποτελούν γενικές στρατηγικές αναζήτησης και σκοπός τους είναι να κατευθύνουν με κατάλληλο τρόπο μεταευρετικές μεθόδους.

### ΑΠΛΗΣΤΟΙ ΑΛΓΟΡΙΘΜΟΙ (Greedy algorithms)

Οι άπληστοι αλγόριθμοι εφαρμόζονται συνήθως για την επίλυση προβλημάτων βελτιστοποίησης, όπως π.χ. η εύρεση του συντομότερου μονοπατιού μεταξύ δύο κορυφών ενός γράφου ή η εύρεση της βέλτιστης σειράς για την εκτέλεση από έναν υπολογιστή ενός συνόλου έργων.

Με τον όρο «άπληστος αλγόριθμος» αναφερόμαστε σε αλγόριθμους που σε κάθε βήμα κάνουν την επιλογή που φαίνεται ως καλύτερη την κάθε δεδομένη στιγμή ελπίζοντας ότι η τελική λύση που θα δώσουν θα είναι η καλύτερη δυνατή. Ένας άπληστος αλγόριθμος διαθέτει μία γενικά απλή δομή που αποτελείται από τα εξής στοιχεία:

- ένα σύνολο υποψήφιων επιλογών (π.χ. οι κορυφές ενός γράφου),
- ένα σύνολο επιλογών που έχουν ήδη χρησιμοποιηθεί,
- μία συνάρτηση ελέγχου, που απαντά στο ερώτημα αν ένα συγκεκριμένο σύνολο υποψήφιων επιλογών αποδίδει μία λύση, όχι απαραίτητα τη βέλτιστη για τη στιγμή που εξετάζεται,

- μία συνάρτηση που ελέγχει αν ένα σύνολο υποψήφιων επιλογών είναι εφικτό, με την έννοια ότι μπορεί αυτό να συμπληρωθεί με τέτοιο τρόπο, ώστε να δώσει μία λύση στο πρόβλημα,
- μία συνάρτηση επιλογής, που ανά πάσα στιγμή δείχνει ποια επιλογή έχει την καλύτερη προοπτική για να είναι μέρος της λύσης,
- μία αντικειμενική συνάρτηση, που δίνει την τιμή της λύσης και είναι αυτή που επιθυμούμε να βελτιστοποιήσουμε.

Οι άπληστοι αλγόριθμοι επικεντρώνονται στην εύρεση, κάθε φορά κάτω από τους περιορισμούς που έχει το πρόβλημα, της τοπικά βέλτιστης επιλογής που θα μεγιστοποιεί την αντικειμενική συνάρτηση που εξετάζεται με την ελπίδα να βρεθεί μία τοπικά βέλτιστη επιλογή που θα οδηγεί σε μία ολικά βέλτιστη λύση. Γενικά οι αλγόριθμοι αυτοί σχεδόν ποτέ δεν παράγουν μια ολικά βέλτιστη λύση, αρκετές φορές δίνουν καλά αποτελέσματα μέσα σε λίγο υπολογιστικό χρόνο και άλλες δίνουν την χειρότερη δυνατή λύση. Χαρακτηριστικό παράδειγμα είναι η επιλογή του κοντινότερου γείτονα για επίλυση του προβλήματος του πλανόδιου πωλητή. Πολύ γνωστό παράδειγμα άπληστου αλγορίθμου αποτελεί ο αλγόριθμος Dijkstra που αναλύεται παρακάτω.

## ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΜΕΘΟΔΟΙ ΕΠΙΛΟΓΗΣ ΜΟΝΟΠΑΤΙΟΥ

### 2.7.1.4 ΑΛΓΟΡΙΘΜΟΙ ΣΥΝΤΟΜΟΤΕΡΗΣ ΔΙΑΔΡΟΜΗΣ

#### *(Shortest Path Routing)*

Η βασική ιδέα των αλγορίθμων αυτών είναι η αναπαράσταση του δικτύου σε ένα γράφο και η εύρεση της καταλληλότερης σύνδεσης μεταξύ των κόμβων μέσω του υπολογισμού της συντομότερης διαδρομής (Shortest Path). Πρόκειται για στατικούς αλγόριθμους που δεν λαμβάνουν υπόψη τους τον φόρτο του δικτύου. Κάθε κόμβος του δικτύου αναπαρίσταται με μια κορυφή στο γράφο, ενώ οι ακμές του γράφου αναπαριστούν τη φυσική σύνδεση μεταξύ αυτών. Σε κάθε κορυφή αντιστοιχίζεται μία ετικέτα (label) στην οποία αναγράφεται το κόστος μετάβασης σε αυτήν, από τον αρχικό κόμβο, μέσω της συντομότερης διαδρομής και ο προηγούμενος κόμβος της διαδρομής.

Στην απλούστερη μορφή τους, για τον υπολογισμό της συντομότερης διαδρομής λαμβάνεται υπόψη μόνο ο αριθμός των κόμβων που παρεμβάλλονται μεταξύ του κόμβου αποστολέα και κόμβου παραλήπτη (hops). Σε πιο εξελιγμένη μορφή μπορεί να λαμβάνουν υπόψη τους την φυσική απόσταση, το εύρος ζώνης (bandwidth), τον μέσο φόρτο του δικτύου (average traffic), το κόστος της επικοινωνίας, τη καθυστέρηση (delay), και άλλους παράγοντες.

Υπάρχουν αρκετοί αλγόριθμοι στην βιβλιογραφία για τον υπολογισμό της βέλτιστης διαδρομής μεταξύ δύο κόμβων ενός γραφήματος. Ο πιο γνωστός από αυτούς είναι ο αλγόριθμος του Dijkstra (1959).

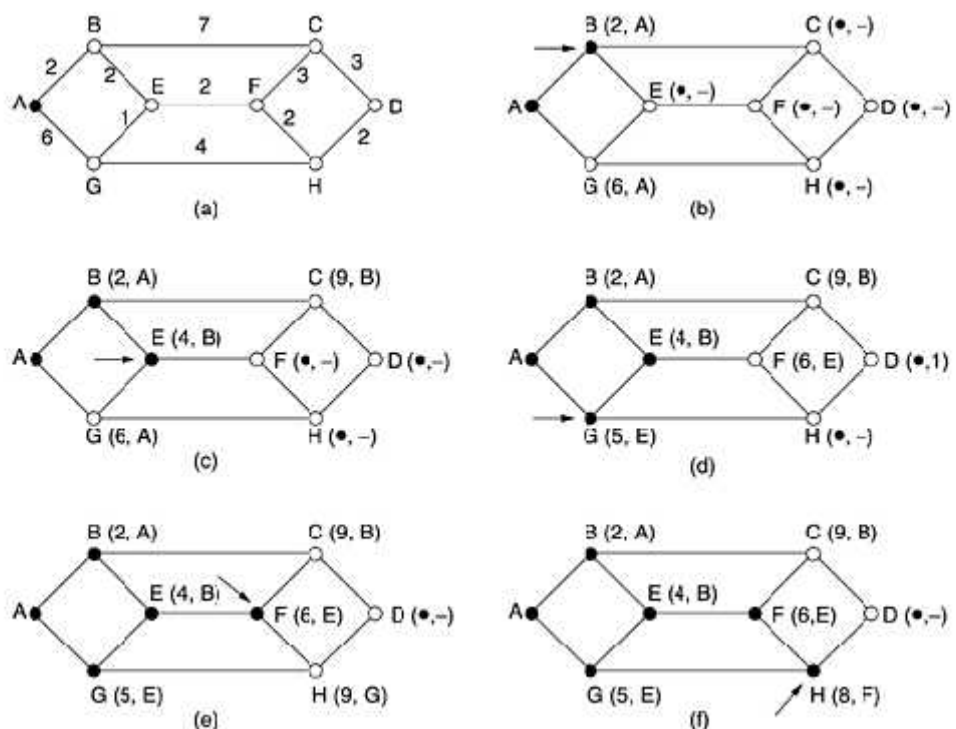
## ΑΛΓΟΡΙΘΜΟΣ ΤΟΥ DIJKSTRA (1959)

Πρόκειται για ένα άπληστο αλγόριθμο επιλογής της συντομότερης διαδρομής που συνδέει δύο σημεία πάνω σε ένα γράφο όπως εμφανίζεται στην Εικόνα 6.

Τα βήματα του αλγορίθμου περιγραφικά μπορούν να διατυπωθούν ως ακολούθως:

- Αρχικά δίνεται η τιμή απόστασης 0 στον αρχικό κόμβο και τιμή άπειρο σε όλους τους υπόλοιπους.
- Χαρακτηρίζονται όλοι οι κόμβοι ως μη-επεξεργασμένοι. Ο «τρέχων κόμβος» είναι ο αρχικός.
- Για τον τρέοντα κόμβο, εξετάζονται όλοι οι μη-επεξεργασμένοι γείτονές του και υπολογίζεται το συνολικό άθροισμα απόστασής τους από τον αρχικό κόμβο. Για παράδειγμα, αν ο τρέχων κόμβος έχει απόσταση 6 από τον αρχικό και ο γείτονας του τρέχοντος κόμβου, που εξετάζει αυτή τη στιγμή ο αλγόριθμος, έχει απόσταση 2 από τον τρέχων, το συνολικό άθροισμα απόστασης του γείτονα από τον αρχικό κόμβο είναι  $6+2=8$ . Αν αυτή η απόσταση είναι μικρότερη από την ετικέτα απόστασης που είχε σημειωθεί, αντικαθίσταται με τη νέα υπολογισμένη τιμή και σημειώνεται ο τρέχων κόμβος στην ετικέτα του γείτονα ως ετικέτα προηγούμενου κόμβου.
- Όταν τελειώσει η εξέταση όλων των γειτόνων του τρέχοντος κόμβου, σημειώνεται ο τρέχων κόμβος ως επεξεργασμένος. Ένας επεξεργασμένος κόμβος δεν εξετάζεται ποτέ ξανά από τον αλγόριθμο. Η ετικέτα απόστασής του είναι η ελάχιστη και θα παραμείνει σταθερή.
- Ο επόμενος τρέχων κόμβος θα είναι ο μη-επεξεργασμένος κόμβος με τη μικρότερη ετικέτα απόστασης.

- Αν όλοι οι κόμβοι έχουν σημειωθεί ως επεξεργασμένοι, ο αλγόριθμος προχωράει στο επόμενο βήμα. Διαφορετικά συνεχίζει από το βήμα 3.
- Ξεκινώντας από τον κόμβο-προορισμό (ο οποίος είναι ο τελευταίος τρέχων κόμβος) εκτυπώνεται ο κόμβος που αναγράφεται στην ετικέτα προηγούμενου κόμβου. Η διαδικασία αυτή επαναλαμβάνεται μέχρι η ετικέτα προηγούμενου κόμβου που θα συναντήσει ο αλγόριθμος να είναι άδεια.



Εικόνα 6: Αναπαράσταση της λειτουργίας του Αλγορίθμου Dijkstra.

Στην Εικόνα 6 απεικονίζεται η λειτουργία του αλγορίθμου Dijkstra. Συμπερασματικά μπορούμε να πούμε ότι, με βάση τις πληροφορίες από τον κάτω δεξιό γράφο, η συντομότερη διαδρομή για να μεταβούμε από τον αρχικό κόμβο A στον κόμβο H είναι η A-B-E-F-H.

Παράδειγμα του αλγορίθμου του Dijkstra δίνεται στο Παράρτημα.

### 2.7.1.5 ΑΛΓΟΡΙΘΜΟΙ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

#### *(Local Search Algorithms)*

Οι αλγόριθμοι τοπικής αναζήτησης λειτουργούν βελτιώνοντας μία τρέχουσα κατάσταση ενώ μετακινούνται μόνο σε γειτονικές καταστάσεις. Πρόκειται για μεταερευνητικούς αλγορίθμους όπου σε κάθε βήμα του αλγορίθμου έχουμε μία πλήρη αλλά ατελή λύση στο υπό μελέτη πρόβλημα. Οι αλγόριθμοι αυτοί χρειάζονται σταθερό χώρο αλλά μπορούν να βρουν ικανοποιητικές λύσεις σε μεγάλους ή άπειρους χώρους λύσεων όπου εξαντλητική αναζήτηση όλων των πιθανών συνδυασμών θα ήταν απαγορευτική.

### 2.7.1.6 ΑΛΓΟΡΙΘΜΟΣ ΑΠΑΓΟΡΕΥΜΕΝΗΣ ΑΝΑΖΗΤΗΣΗΣ

#### *(Tabu search)*

Πρόκειται για μεταερευνητικό αλγόριθμο αναζήτησης. Ο αλγόριθμος της απαγορευμένης αναζήτησης βελτιώνει τη λειτουργία των αλγορίθμων τοπικής αναζήτησης αποθηκεύοντας, κατά την εκτέλεση του, σε μία δομή, αποτελέσματα από προηγούμενες αναζητήσεις. Κατά συνέπεια εάν μία πιθανή λύση είχε απορριφθεί ως μη βέλτιστη ή παραβίαζε παραμέτρους του προβλήματος τότε αυτή χαρακτηρίζεται ως «απαγορευμένη» και ο αλγόριθμος δεν την επανεξετάζει.

### 2.7.1.7 ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ *(Genetic Algorithms)*

Είναι μια τεχνική προγραμματισμού που εισήγαγε στα τέλη της δεκαετίας του 1960 ο Τζον Χόλαντ, ερευνητής του Ινστιτούτου της Σάντα Φε, αν και γενετικοί αλγόριθμοι είχαν εμφανιστεί από την δεκαετία του 50 στον χώρο της βιολογίας.

Ο τρόπος λειτουργίας των Γενετικών Αλγορίθμων είναι εμπνευσμένος από την βιολογία. Χρησιμοποιούν την ιδέα της εξέλιξης μέσω γενετικής μετάλλαξης, φυσικής επιλογής και διασταύρωσης και προσπαθούν να βρουν τη λύση ενός προβλήματος με το να προσομοιώνουν την εξέλιξη ενός πληθυσμού «λύσεων» του προβλήματος.

Στην πράξη ο αλγόριθμος ξεκινά μ' ένα σύνολο λύσεων - ονομάζονται γονιδιώματα, δανειζόμενες το όνομά τους από τη βιολογία- οι οποίες συνιστούν τον «πληθυσμό». Κατόπιν ζητείται από τον υπολογιστή να δημιουργήσει μια σειρά τυχαίων ανασυνδυασμών και μεταλλάξεων των «γονιδιωμάτων». Οι πιο ικανές λύσεις για ένα συγκεκριμένο πρόβλημα συνεχίζουν να εξελίσσονται και ανασυνδυάζονται τυχαία, μέχρις ότου επιβιώσουν οι καλύτερες. Συνήθως, όσο περισσότερες γενιές περνούν τόσο καλύτερες λύσεις βρίσκονται.

#### 2.7.1.8 ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ

##### *(Simulated Annealing)*

Ο αλγόριθμος της προσομοιωμένης απόπτωσης είναι εμπνευσμένος από την θερμή κατεργασία των μετάλλων και έχει εφαρμοστεί σε μεγάλο βαθμό σε προβλήματα βελτιστοποίησης και ειδικότερα στο πρόβλημα του πλανόδιου πωλητή, που αναλύεται στην συνέχεια.

Πρόκειται για μεταερευτικό αλγόριθμο που επιτρέπει, με συγκεκριμένη πιθανότητα, τη μετάβαση σε λύσεις που αντιστοιχούν σε χειρότερες τιμές της αντικειμενικής συνάρτησης του προβλήματος. Με αυτό τον τρόπο επιτρέπεται ο απεγκλωβισμός του αλγορίθμου επίλυσης από τοπικά ελάχιστα. Η πιθανότητα μετάβασης καθορίζεται από την «θερμοκρασία» του αλγορίθμου. Όσο υψηλότερη είναι η θερμοκρασία τόσο μεγαλύτερη είναι η πιθανότητα για ανοδική κίνηση στον χώρο των λύσεων. Παράδειγμα του αλγορίθμου δίνεται στο Παράρτημα.



### 3. ΠΡΟΣΩΠΟΠΟΙΗΜΕΝΗ ΠΛΟΗΓΗΣΗ

Στο κεφάλαιο αυτό παρουσιάζονται το πρόβλημα της προσωποποιημένης πλοήγησης, τρία πολύ γνωστά προβλήματα που απαντώνται στη βιβλιογραφία και γίνεται σύντομη αλλά περιεκτική ανασκόπηση της προηγούμενης έρευνας. Ακόμη, για να γίνει κατανοητός ο τρόπος λειτουργίας ενός συστήματος προσωποποιημένης πλοήγησης σε εξωτερικούς χώρους για τουρίστες και το πρόβλημα της χάραξης της διαδρομής δίνεται ένα απλό παράδειγμα. Μέσα από το παράδειγμα αυτό παρουσιάζεται η ομοιότητα του υπό μελέτη προβλήματος με το πρόβλημα του Προσανατολισμού και γίνεται σαφές γιατί το δεύτερο χρησιμοποιήθηκε κατά την υλοποίηση του αλγορίθμου που προτείνεται στην παρούσα μελέτη.

#### 3.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Όπως έχει ήδη αναφερθεί η διαδικασία της προσωποποιημένης πλοήγησης πραγματοποιείται μέσα από συστήματα λήψης απόφασης ή διαδικτυακές εφαρμογές που μπορεί να τις χειρίζεται ο χρήστης από έναν ηλεκτρονικό υπολογιστή ή κινητό τηλέφωνο και αφορά την επιλογή και χάραξη μίας βέλτιστης διαδρομής βάση των προτιμήσεων ενός ατόμου. Τα δεδομένα εισόδου είναι συνήθως τα σημεία ενδιαφέροντος και οι περιορισμοί του χρήστη, στην περίπτωση μας ενός τουρίστα που βρίσκεται σε μία δεδομένη περιοχή. Η έξοδος είναι η διαδρομή που καλείται να ακολουθήσει. Η προσωποποιημένη πλοήγηση μπορεί να αφορά πλοήγηση σε εσωτερικούς ή εξωτερικούς χώρους.

Χαρακτηριστικό παράδειγμα προσωποποιημένης πλοήγησης σε εσωτερικούς χώρους είναι η εύρεση διαδρομών για τη πλοήγηση ατόμων με ειδικές ανάγκες μέσα σε κτήρια (J. Candy 2007, H. A Karimi et al. 2010). Ενώ η πλοήγηση σε εξωτερικούς χώρους αφορά την επιλογή της βέλτιστης ή συντομότερης διαδρομής, η πλοήγηση σε εσωτερικούς χώρους έχει ως βασικό στόχο την εύρεση της ασφαλέστερης διαδρομής (ή της διαδρομής με τα λιγότερα εμπόδια) από ένα σημείο σε ένα άλλο μέσα σε ένα κλειστό χώρο.

Θα μπορούσαμε να χωρίσουμε τη λειτουργία μιας εφαρμογής προσωποποιημένης πλοήγησης μπορεί σε τρία στάδια:

- I. Προτάσεις (Recommendation)
- II. Χάραξη διαδρομής (Route generation) και
- III. Προσωποποίηση της διαδρομής (Route customization)

Το πρώτο στάδιο αφορά τις πληροφορίες που παρέχονται σε έναν συνδεδεμένο χρήστη και τα προτεινόμενα σημεία που μπορεί αυτός να επισκεφτεί. Αναφέρεται στην στιγμή κατά την οποία ο χρήστης συνδέεται στο σύστημα ή εφαρμογή και εμφανίζονται σε αυτόν τα σημεία ενδιαφέροντος που υπάρχουν στην περιοχή του αλλά και πληροφορίες για αυτά.

Τα επόμενα δύο στάδια αναφέρονται στη χάραξη της βέλτιστης διαδρομής βάση των προτιμήσεων και περιορισμών του χρήστη και είναι τα στάδια στα οποία επικεντρώνεται η παρούσα μελέτη.

Τα βασικά προβλήματα που πρέπει να αντιμετωπισθούν κατά το σχεδιασμό μιας εφαρμογής προσωποποιημένης πλοήγησης είναι τα ακόλουθα:

- Ο τρόπος και οι τεχνικές με τις οποίες θα λαμβάνονται τα δεδομένα.
- Ο τρόπος με τον οποίο θα λαμβάνονται έγκυρα- επικαιροποιημένα δεδομένα.
- Τα κριτήρια με τα οποία θα ιεραρχούνται τα σημεία ενδιαφέροντος.
- Τα κριτήρια - προτιμήσεις της διαδρομής που θα μπορεί να μεταβάλει ο χρήστης.
- Η μεθοδολογία με την οποία θα γίνεται ο **υπολογισμός της βέλτιστης διαδρομής (optimal route)**.
- Ο τρόπος με τον οποίο θα γίνεται η θέαση των αποτελεσμάτων.

Σημαντικό στοιχείο στην προσωποποιημένη πλοήγηση είναι λήψη των δεδομένων. Στην απλούστερη τους μορφή τα δεδομένα αποτελούνται από τα σημεία ενδιαφέροντος, πληροφορίες για την γεωγραφική τους θέση (γεωγραφικό πλάτος και μήκος) και την βαθμολογία (score) που φέρουν. Η

βαθμολογία ουσιαστικά αντικατοπτρίζει το βαθμό ελκυστικότητας ενός σημείου ή αλλιώς την χρησιμότητα που έχει για τον τουρίστα που θα το επισκεφτεί. Σε πιο πολύπλοκα μοντέλα τα δεδομένα μπορεί να αφορούν και πληροφορίες για την κίνηση στο οδικό δίκτυο, πληροφορίες για τον καιρό κ.α.

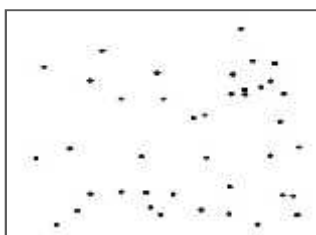
Αν και η παρούσα μελέτη επικεντρώνεται στον τρόπο χάραξης και επιλογής μίας καλής αν όχι της βέλτιστης διαδρομής αξίζει να αναφέρουμε ότι στις περισσότερες των περιπτώσεων τα αρχικά δεδομένα, όπως αναφέρθηκαν προηγουμένως, εισάγονται σε ένα κείμενο το οποίο με τη σειρά του μετατρέπεται σε διανύσματα που περιέχουν τις κατάλληλες πληροφορίες. Αρχικά από το κείμενο αφαιρούνται οι λέξεις «και», «μετά», «αυτό», έπειτα οι λέξεις αναλύονται και αντικαθίστανται από την ρίζα τους ώστε ο υπολογιστής να μπορεί να τις αναγνωρίσει. Η διαδικασία αυτή πραγματοποιείται συνήθως με ειδικούς αλγορίθμους όπως π.χ. ο Porter Stemming Algorithm (Porter 1980). Με αυτόν τον τρόπο το λεξιλόγιο του κειμένου μειώνεται κατά 33% και είναι πιο εύκολη η δημιουργία διανυσμάτων που περιέχουν σωστή πληροφορία (Souffriau et al. 2008).

Ένα από τα βασικά στοιχεία του προβλήματος της προσωποποιημένης πλοήγησης αποτελεί ο υπολογισμός της βέλτιστης διαδρομής. Υπάρχουν αρκετές προσεγγίσεις σε αυτό το πρόβλημα. Το πρόβλημα της εύρεσης της βέλτιστης διαδρομής έχει απασχολήσει την επιχειρησιακή έρευνα, τους μαθηματικούς και τους πληροφορικούς. Για παράδειγμα το ίδιο πρόβλημα μπορεί να μοντελοποιηθεί σαν πρόβλημα γραμμικού ή μη γραμμικού προγραμματισμού αλλά και σαν πρόβλημα πχ. του πλανόδιου πωλητή. Αν και το στάδιο αυτό αναλύεται εκτενώς σε επόμενο κεφάλαιο αξίζει να πούμε ότι ο τρόπος υπολογισμού της βέλτιστης διαδρομής επηρεάζει άμεσα την αποτελεσματικότητα και την αξία της όποιας λύσης στο πραγματικό περιβάλλον καθώς επηρεάζει την ταχύτητα λειτουργίας του όποιου συστήματος.

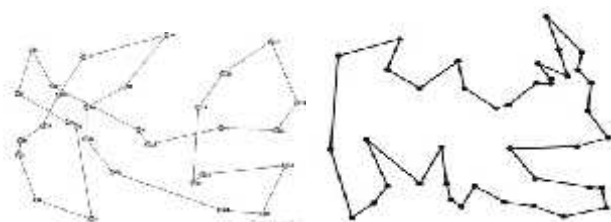
Στην συνέχεια παρουσιάζονται τρία πολύ γνωστά προβλήματα που απαντώνται στην βιβλιογραφία και έχουν χρησιμοποιηθεί στα πλαίσια της προσωποποιημένης πλοήγησης και της επιλογής της βέλτιστης διαδρομής. Αυτά είναι το πρόβλημα του Πλανόδιου Πωλητή (Keller 1988) , το πρόβλημα του Προσανατολισμού (Tsiligirides 1984, Geem 2005, Chao 2006, Souffriau et al. 2008) και το πρόβλημα του Ομαδικού Προσανατολισμού (Vansteenwegen et al. 2008).

### 3.2 ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΛΑΝΟΔΙΟΥ ΠΩΛΗΤΗ

Στο πρόβλημα του Πλανόδιου Πωλητή (Travelling salesman problem – TSP) με δεδομένο ένα σύνολο από πόλεις και γνωρίζοντας την μεταξύ τους απόσταση θέλουμε να βρούμε την διαδρομή που θα ακολουθήσει ένας πλανόδιος πωλητής. Ο πωλητής θα πρέπει να περάσει από όλες τις πόλεις (Εικόνα 7) ακριβώς μία φορά και να καταλήξει σε αυτή από την οποία ξεκίνησε ενώ θα πρέπει να διανύσει τα λιγότερα δυνατά χιλιόμετρα.



Εικόνα 7: Σημεία ενδιαφέροντος που πρέπει να επισκεφτεί ο πωλητής.



Εικόνα 8: Δύο διαδρομές για το πρόβλημα του πλανόδιου πωλητή..

Στην Εικόνα 8 απεικονίζονται δύο διαδρομές που έχουν χαραχθεί για το πρόβλημα του πλανόδιου πωλητή, μία αποδέκτη και μία που πρέπει να απορριφθεί. Η διαδρομή της αριστερής εικόνας περιέχει πολλές διασταυρώσεις, αυτό σημαίνει ότι ο χρήστης ίσως να διανύει μεγαλύτερη απόσταση από ότι πραγματικά θα έπρεπε και άρα είναι σχεδόν βέβαιο ότι υπάρχει καλύτερη λύση. Οι αποδεκτές λύσεις έχουν την μορφή της δεξιάς εικόνας. Δεν έχουν διασταυρώσεις και δεν αναγκάζουν τον χρήστη να πηγαίνει πίσω-μπρος γύρω από ένα σημείο σπαταλώντας χρόνο.

Πρόκειται για ένα NP – Hard πρόβλημα ευρέως διαδεδομένο στην θεωρία των υπολογιστών αλλά και στην επιχειρησιακή έρευνα. Έχει μελετηθεί στα πλαίσια των προβλημάτων βελτιστοποίησης περισσότερο από κάθε άλλο

ενώ βρίσκει εφαρμογές σε ένα ευρύ φάσμα προβλημάτων (από μεταφορά εμπορευμάτων έως και προβλήματα που αφορούν την αλληλουχία του DNA).

Στην γενικευμένη του μορφή (generalized TSP) τα σημεία εκκίνησης και τερματισμού διαφέρουν. Μία πολύ γνωστή παραλλαγή του TSP αποτελεί το πρόβλημα του Προσανατολισμού που μελετάται στη συνέχεια, ενώ υπάρχουν και ειδικές μορφές όπως το delta - TSP (οι αποστάσεις μεταξύ των πόλεων επιβεβαιώνουν την τριγωνική ανισότητα), το Euclidean - TSP και asymmetric - TSP (μεταξύ δύο σημείων A,B η αποστάσεις AB και BA είναι διαφορετικές, συνήθως εμφανίζεται σε προβλήματα όπου η διαδρομή χαράζεται με βάση το οδικό δίκτυο). Ακόμη, υπάρχει και το πολλαπλό TSP (mTSP) κατά το οποίο υπάρχουν παραπάνω από ένας πωλητές που πρέπει να επισκεφθούν το σύνολο των πόλεων. Κάθε πόλη μπορεί να την επισκεφθεί μόνο ένας πωλητής και κάθε διαδρομή πρέπει να πραγματοποιείται στον ελάχιστο δυνατό χρόνο.

### ΟΡΙΣΜΟΣ ΤΟΥ TSP ΣΤΟΝ ΓΡΑΜΜΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Έστω A ένα σύνολο με n πόλεις, d η απόσταση μεταξύ των πόλεων i, j. Το σύνολο αυτό μπορεί να αναπαρασταθεί σε ένα μη κατευθυνόμενο γράφο όπου οι κόμβοι είναι οι πόλεις του συνόλου A και οι ακμές των γράφων αναπαριστούν τις αποστάσεις μεταξύ των πόλεων, συνεπώς πρέπει να βρεθεί ο συντομότερος κύκλος Hamilton. Το πρόβλημα του Πλανόδιου Πωλητή μπορεί να γραφεί σαν απλό πρόβλημα ελαχιστοποίησης:

$$\min \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}$$

$$\sum_{i=0}^n x_{ij} = 1, \quad j = 0, \dots, n$$

$$\sum_{j=0}^n x_{ij} = 1, \quad i = 0, \dots, n$$

Η μεταβλητή x, είναι ακέραιος αριθμός και παίρνει τιμές 1 αν το υπό μελέτη μονοπάτι πηγαίνει από τη πόλη i στην j και 0 για κάθε άλλη περίπτωση.

### 3.3 ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ (ΜΕ ΕΝΑ ΠΑΙΧΤΗ, ΟΜΑΔΙΚΟ)

Το πρόβλημα του Προσανατολισμού (Orienteering Problem - OP) βασίζεται στο «orienteering». Το «orienteering» είναι ένα υπαίθριο άθλημα που παίζεται συνήθως σε βουνά ή σε περιοχές με αυξημένη βλάστηση. Σε μία προκαθορισμένη περιοχή επιλέγονται συγκριμένα σημεία τα οποία οι παίκτες μπορούν να επισκεφτούν και τους αποδίδεται μία βαθμολογία. Οι παίκτες ξεκινούν από ένα συγκεκριμένο σημείο και οπλισμένοι με χάρτες και πυξίδες πρέπει να περάσουν από όσα το δυνατόν περισσότερα σημεία μπορούν μέσα σε συγκεκριμένο χρόνο, άρα κάθε παίκτης μπορεί να μην έχει αρκετό χρόνο να περάσει από όλα τα σημεία. Επειδή κάθε σημείο τους αποδίδει και ένα συγκεκριμένο σκορ/βαθμολογία σκοπός του παιχνιδιού είναι να μεγιστοποιήσουν τη τελική βαθμολογία που θα συγκεντρώσουν ενώ κάθε σημείο μπορούν να το επισκεφτούν μόνο μία φορά. Εάν κάποιος φτάσει στο τερματισμό μετά από τον προκαθορισμένο χρόνο αποκλείεται από το παιχνίδι, Tsiligirides (1984).

Η παραπάνω περιγραφή αφορά το πρόβλημα του Προσανατολισμού με ένα παίκτη (Single-Competitor Orienteering Problem (OP)) ή όπως αναφέρεται σπανιότερα το Selective Traveling Salesman Problem (STSP) και είναι ένα NP - hard πρόβλημα, Golden, Levy and Vohra (1987). Συνοπτικά θα μπορούσαμε να πούμε ότι το OP μπορεί να μοντελοποιηθεί σαν ένα πολυεπίπεδο πρόβλημα βελτιστοποίησης. Αρχικά πρέπει να επιλεγούν τα σημεία που ένας παίκτης μπορεί να επισκεφτεί και έπειτα το πρόβλημα μπορεί να αναχθεί σε πρόβλημα του πλανόδιου πωλητή ή σε εύρεση της μικρότερης Hamiltonian διαδρομής, Chao et al. (1966b).

### ΠΡΟΒΛΗΜΑ ΤΟΥ ΟΜΑΔΙΚΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ

Στο πρόβλημα του Ομαδικού Προσανατολισμού (Team Orienteering Problem - TOP) οι συνθήκες του παιχνιδιού είναι ίδιες μόνο που δεν διαγωνίζεται κάθε ένας για τον εαυτό του αλλά ανά ομάδες. Κάθε μέλος της ομάδας μέσα σε προκαθορισμένο χρόνο περνάει από όσα περισσότερα σημεία μπορεί ενώ μόνο ο πρώτος παίκτης της κάθε ομάδας που θα περάσει από ένα σημείο λαμβάνει τη σχετική βαθμολογία. Οι ομάδες που τερματίζουν έγκαιρα κρίνονται με βάση τη συνολική βαθμολογία που έχουν συλλέξει. Έτσι, προτιμούν να χωρίζουν την περιοχή στην οποία εκτυλίσσεται το παιχνίδι σε

υποπεριοχές και κάθε μέλος της ομάδας να ασχολείται μόνο με μία ώστε να μην υπάρχουν αλληλεπικαλύψεις και άρα σπατάλη χρόνου, δηλαδή η λύση να είναι τύπου «μαργαρίτας».

Το TOP μπορεί να μοντελοποιηθεί ως ένα «πολυπεπίπεδο πρόβλημα βελτιστοποίησης», Chao et al. (1996a), όπου αρχικά από το σύνολο των σημείων πρέπει να επιλεγεί το υποσύνολο που η ομάδα προλαβαίνει να χρησιμοποιήσει, έπειτα το υποσύνολο αυτό πρέπει να χωρισθεί σε επιμέρους υποσύνολα που κάθε παίχτης μπορεί να χρησιμοποιήσει και τέλος πρέπει να αποφασισθεί η διαδρομή που θα ακολουθήσει κάθε παίχτης.

Το OP έχει χρησιμοποιηθεί εκτενώς, σε αντίθεση με το TOP, σε πολλές έρευνες για την επίλυση διαφόρων προβλημάτων. Από πρακτικά προβλήματα όπως η διάθεση καυσίμων θέρμανσης όπου η βαθμολογία του κάθε σημείου είναι η αμεσότητα με την οποία πρέπει να εξυπηρετηθεί ο πελάτης, Golden, Assad and Dahl (1984), μέχρι και σε παραλλαγές με χρονικά παράθυρα που μπορούν να εφαρμοστούν σε τράπεζες, Kantor, Rosenwein (1992).

Αν και διάφοροι ερευνητές έχουν διαφορετικές απόψεις, στα πλαίσια της παρούσας μελέτης, θεωρείται ότι τη γενικευμένη μορφή του προβλήματος αποτελεί το TOP και ειδικότερη μορφή αυτού, δηλαδή με έναν παίχτη σε κάθε ομάδα, αποτελεί το Single - Competitor Orienteering Problem (OP).

## ΟΡΙΣΜΟΣ ΤΟΥ ΟΡ ΣΤΟΝ ΓΡΑΜΜΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Έστω  $A$  ένα σύνολο με  $n$  σημεία ενδιαφέροντος,  $d$  η απόσταση μεταξύ των σημείων  $i, j$ . Το σύνολο αυτό μπορεί να αναπαρασταθεί σε ένα κατευθυνόμενο γράφο όπου οι κόμβοι είναι τα σημεία του συνόλου  $A$  και οι ακμές των γράφων αναπαριστούν τις αποστάσεις  $d$  μεταξύ των σημείων. Κάθε κόμβος φέρει μία βαθμολογία  $S$  και κάθε ακμή συνδέεται και με ένα χρόνο μετάβασης από το ένα σημείο στο άλλο,  $t_{ij}$ . Το πρόβλημα του Πλανόδιου Πωλητή μπορεί να γραφεί σαν πρόβλημα μεγιστοποίησης της βαθμολογίας χωρίς όμως να γίνεται υπέρβαση του προκαθορισμένου χρονικού περιθωρίου ( $T_{max}$ ):

$$\text{Max} \quad \sum_{i=2}^{n-1} \sum_{j=2}^n S_i x_{ij}$$

s.t.:

$$\sum_{j=2}^n x_{1j} = \sum_{i=1}^{n-1} x_{in} = 1 \quad (1)$$

$$\sum_{j=2}^n x_{kj} = \sum_{i=1}^{n-1} x_{ik} \leq 1 \quad k = 2, \dots, n-1 \quad (2)$$

$$\sum_{i=1}^{n-1} \sum_{j=2}^n t_{ij} x_{ij} \leq T_{max} \quad (3)$$

$$2 \leq u_i \leq n \quad i = 2, \dots, n \quad (4)$$

$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}) \quad i, j = 2, \dots, n \quad (5)$$

$$x_{ij} \in \{0,1\} \quad 1 \leq i, j \leq n, \quad u_i \in Z \quad i = 2, \dots, n$$

Η μεταβλητή  $x$ , είναι ακέραιος αριθμός και παίρνει τιμές 1 αν μετά το σημείο ενδιαφέροντος  $i$  ο χρήστης επισκέπτεται το σημείο  $j$  και 0 για κάθε άλλη περίπτωση. Η μεταβλητή  $u$  είναι ακέραιος αριθμός και δείχνει τη θέση του  $i$  σημείου μέσα στη διαδρομή.

Ο περιορισμός (1) καθορίζει ότι η διαδρομή θα ξεκινάει από το σημείο 1 και θα τελειώνει σε κάποιο άλλο και ο (2) καθορίζει ότι όλα τα σημεία θα χρησιμοποιούνται το πολύ μία φορά στη διαδρομή. Ο περιορισμός (3) αφορά τον χρονικό περιορισμό ενώ οι περιορισμοί (4) και (5) συμπληρώνουν τη διατύπωση του προβλήματος, Vansteenwegen et al. (2011).



### 3.4 ΑΝΑΣΚΟΠΗΣΗ ΠΡΟΗΓΟΥΜΕΝΗΣ ΕΡΕΥΝΑΣ

Σήμερα μεγάλη μερίδα του πληθυσμού ανατρέχει για να βρει κάθε είδους πληροφορία στο διαδίκτυο. Είτε αυτό γίνεται από το σπίτι είτε από αλλού οι άνθρωποι αναζητούν πληροφορίες στο διαδίκτυο από τους υπολογιστές τους, τα κινητά τους τηλέφωνα ή τις ταμπλέτες τους. Πρωταγωνιστικό ρόλο σε αυτή την διαδικασία έχουν οι εφαρμογές, οι διάφορες προσωποποιημένες διαδικτυακές υπηρεσίες και τα συστήματα λήψης απόφασης. Αυτή η τάση φυσικά δεν έχει αφήσει ανεπηρέαστο τον τομέα του τουρισμού.

Με τον όρο Προσωποποιημένες Τουριστικές Διαδικτυακές Υπηρεσίες αναφερόμαστε σε διαδικτυακές υπηρεσίες που κάνουν χρήση των υπηρεσιών θέσης με σκοπό να παρέχουν πληροφορίες σε τουρίστες και να εξυπηρετούν με διάφορους τρόπους την μετάβαση στον προορισμό που έχουν επιλέξει ή τις δραστηριότητές τους. Χαρακτηριστικό των υπηρεσιών αυτών είναι ότι δεν κάνουν μόνο χρήση της θέσης του χρήστη, της ώρας του καιρού κτλ αλλά και των προσωπικών προτιμήσεων του.

Την τελευταία εικοσαετία έχουν γίνει αρκετές προσπάθειες και έχουν υλοποιηθεί αρκετές υπηρεσίες και συστήματα με σκοπό την παροχή λύσεων για τουρίστες όπως συστήματα αεροπορικών κρατήσεων, υπηρεσίες για επιλογή εστιατορίων. Μία από τις πρώτες απόπειρες για τη δημιουργία ενός φορητού τουριστικού οδηγού αποτελεί το Cyberguide (1995-1997), του Τεχνολογικού Ινστιτούτου της Τζόρτζια, που παρείχε πληροφορίες βασιζόμενες στην θέση και το προσανατολισμό του χρήστη με βασικό χαρακτηριστικό τη θέαση σε ψηφιακό περιβάλλον κλασικών τουριστικών οδηγών. Έκτοτε αναπτύχθηκαν και άλλα συστήματα όπως το Gulliver's Genie των O' Hare, O' Grandy (2002) και το GUIDE των K. Cheverst et al. (2000). Όλα όμως λαμβάνουν υπόψη την θέση του χρήστη, παρέχουν πληροφορίες για τα σημεία που βρίσκονται γύρω του και τον βοηθούν στο να επιλέξει ποιά σημεία θέλει να επισκεφθεί ενώ για την χάραξη μίας προσωποποιημένης διαδρομής ο χρήστης πρέπει να επιλέξει ο ίδιος τα σημεία που τον ενδιαφέρουν, W. Souffriau et al. (2008).

Την τελευταία δεκαετία έχουν αναπτυχθεί συστήματα και εφαρμογές οι οποίες αντί να προτείνουν έτοιμες διαδρομές ή να ταξινομούν απλά τα διάφορα σημεία ενδιαφέροντος, προτείνουν ένα συνδυασμό από σημεία με σκοπό την παροχή του καλύτερου δυνατού αποτελέσματος, μίας καλής αν όχι της βέλτιστης διαδρομής.

Οι Soo και Liang (2001) παρουσιάζουν ένα πράκτορα λογισμικού (software agent) που προτείνει ολοκληρωμένες διαδρομές, για την πόλη της Ταϊπέι στην Κίνα, μέσω διαλόγου με τον χρήστη. Αρχικά ο τουρίστας επιλέγει το ξενοδοχείο του και βάση του διαθέσιμου χρόνου επιλέγονται, με την μέθοδο του πλησιέστερου γείτονα, τα σημεία ενδιαφέροντος που μπορεί να επισκεφτεί. Για την ίδια πόλη το 2009 οι Yu και Chang υλοποίησαν ένα σύστημα που προτείνει προσωποποιημένες διαδρομές για ξενοδοχεία, εστιατόρια και σημεία ενδιαφέροντος. Συνδυάζοντας αυτές τις τρεις λειτουργίες παρέχεται στον χρήστη μία προσωποποιημένη διαδρομή ανάλογα με την ώρα, την τοποθεσία και τις προτιμήσεις του.

Οι Ardissono et al. (2002, 2003) παρουσιάζουν το INTRIGUE ένα σύστημα προτάσεων (recommender system) που κάνει χρήση της ασαφούς λογικής και παρέχει πληροφορίες και διαδρομές για την πόλη του Τορίνο στην Ιταλία.

Οι Suna και Lee (2004) παρουσιάζουν ένα multi-agent σύστημα που χρησιμοποιεί διανύσματα για να υπολογίσει το ενδιαφέρον του χρήστη απέναντι σε διάφορες γεωγραφικές οντολογίες. Με χρήση ενός αλγορίθμου συντομότερης διαδρομής και βάση του προβλήματος του πλανόδιου πωλητή χαράζονται διαδρομές από σημείο σε σημείο και τέλος παρέχεται στον χρήστη μία διαδρομή που μπορεί να ακολουθήσει.

Οι Maruyama et al. (2004) παρουσιάζουν το P-Tour, μια συσκευή πλοήγησης που υπολογίζει τουριστικές διαδρομές. Αφού ο χρήστης εισάγει μόνος του τη βαθμολογία για κάθε σημείο ενδιαφέροντος το P-Tour χρησιμοποιεί μια παραλλαγή του TSP με κέρδη με σκοπό την εύρεση, με άπληστο τρόπο, του μικρότερου κλειστού κυκλώματος που ελαχιστοποιεί το κόστος ταξιδιού. Το P-Tour επεκτείνεται το 2005 από τους Shiraishi et al. με δύο τρόπους. Πρώτον, εντοπίζουν ανεπιθύμητες καταστάσεις, πχ. λάθος διαδρομή, κατά την εκτέλεση της προγραμματισμένης διαδρομής και προειδοποιούν τον χρήστη με κατάλληλο τρόπο. Δεύτερον, η μηχανή αναζήτησης επεκτείνεται με συναρτήσεις που μπορούν να επιλύσουν περισσότερες συγκρούσεις κατά την επεξεργασία των δεδομένων και τη χάραξη της διαδρομής. Το 2006 οι Nagata et al. επεκτείνουν το P-Tour, προκειμένου να προγραμματίζουν διαδρομές για ομάδες τουριστών ενώ το 2009 οι Wu et al. προσθέτουν στο ίδιο σύστημα στοιχεία, που αφορούν τον καιρό, με τέτοιο τρόπο που η βαθμολογία που αποδίδει κάθε σημείο ενδιαφέροντος να αλλάζει ανάλογα με τον εάν έχει λιακάδα, συννεφιά ή αν βρέχει.

Οι ten Hagen et al. (2005) παρουσιάζουν το Dynamic Tour Guide. Όλα τα σημεία ενδιαφέροντος χωρίζονται σε κατηγορίες (μουσεία, εστιατόρια κ.α). Κάθε κατηγορία αποτελεί μία δομική μονάδα της διαδρομής. Όλες μαζί οι δομικές μονάδες σχηματίζουν μία οντολογία σε μορφή δέντρου και η διαδρομή χαράσσεται επιλέγοντας τυχαία σημεία από τις δομικές μονάδες βάση των προτιμήσεων του χρήστη.

Το 2007 οι Lee et al. παρουσιάζουν ένα σύστημα για περιήγηση στην επαρχία Jeju στην Κορέα. Επιχειρούν να κάνουν συσχετισμούς με τη βοήθεια διανυσμάτων μεταξύ των σημείων ενδιαφέροντος και του χρήστη και στην πραγματικότητα χαράσσουν διαδρομές με την μέθοδο του Orienteering Problem αλλά με τρόπο αρκετά χρονοβόρο, Souffriau, Vansteenwegen (2010). Ακόμη, οι Lee et al., το 2009 παρουσιάζουν ένα σύστημα που επιτρέπει τον σχεδιασμό προσωποποιημένων διαδρομών για την πόλη Ταϊβάν της Κίνας. Πρόκειται για ένα multi-agent σύστημα που αρχικά επιλέγει τα σημεία που ταιριάζουν στις προτιμήσεις του χρήστη, χρησιμοποιεί ασαφή λογική για να επιλέξει 8 σημεία από αυτά (3 αρχαιολογικά σημεία και 5 εστιατόρια) και έπειτα χαράσσει μία διαδρομή ανάμεσα στα σημεία με βάση το πρόβλημα του πλανόδιου πωλητή.

Τέλος οι Souffriau, Vansteenwegen (2010) δημιούργησαν το City Trip Planner, ένα διαδικτυακό σύστημα λήψης απόφασης για τουρίστες. Η χάραξη της διαδρομής γίνεται με βάση το Orienteering Problem μέσω ευρετικών αλγορίθμων ενώ χρησιμοποιούνται και χρονικά παράθυρα που μπορούν να διαφέρουν από μέρα σε μέρα.

### 3.5 ΑΠΛΟ ΠΑΡΑΔΕΙΓΜΑ ΠΡΟΣΩΠΟΠΟΙΗΜΕΝΗΣ ΠΛΟΗΓΗΣΗΣ

Για να γίνει κατανοητός ο τρόπος λειτουργίας μίας εφαρμογής αλλά και το βασικό πρόβλημα της χάραξης της διαδρομής δίνεται το παρακάτω απλό παράδειγμα.

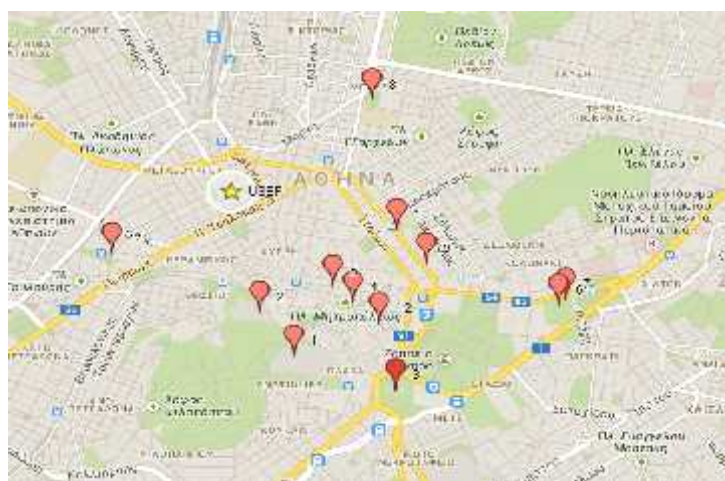
Έστω ότι ο χρήστης συνδέεται μέσω του κινητού του τηλεφώνου σε μία υπηρεσία που σκοπό έχει να του παρέχει μία προσωποποιημένη διαδρομή και δέχεται ως δεδομένα από τον χρήστη μόνο τον χρόνο που έχει στη διάθεση του για να περιηγηθεί στην περιοχή αυτή. Υποθέτουμε πως ο χρήστης βρίσκεται στο κέντρο της Αθήνας και γύρω του υπάρχουν 12 σημεία. Επιλέχθηκαν για το παράδειγμα αυτό 12 σημεία αρχαιολογικού και πολιτιστικού ενδιαφέροντος που βρίσκονται στο κέντρο της πόλης (Πίνακας 1). Για το κάθε σημείο ενδιαφέροντος δίνεται η θέση του (γεωγραφικό μήκος και πλάτος) καθώς και η βαθμολογία η οποία αντιστοιχεί σε αυτό και αντικατοπτρίζει το βαθμό χρησιμότητας που αποδίδει το σημείο αυτό στον χρήστη. Η Εικόνα 9 παρουσιάζει την εικόνα που θα έβλεπε ο χρήστης στο κινητό του.

Το σύνολο των σημείων είναι γνωστό και δίνεται εξαρχής καθώς και οι αποστάσεις μεταξύ των σημείων. Εδώ οι αποστάσεις των σημείων δίνονται από την ευκλείδεια απόσταση. Θεωρητικά θα μπορούσαμε να πούμε ότι ο χρήστης θα πραγματοποιήσει τη διαδρομή με τα πόδια ενώ εναλλακτικά περίπτωση θα μπορούσαν να υπολογίζονται οι αποστάσεις μέσα από το πραγματικό οδικό δίκτυο δοσμένες σε ένα πίνακα.

LAT	LON	SCORE	NAME
37.971775	23.725916	100	Ακρόπολη
37.974177	23.722744	67	Αρχαία Αγορά
37.969322	23.732890	46	Ναός του Ολυμπίου Διός
37.975344	23.730092	33	Μητρόπολη
37.976439	23.728639	54	Καινικαρέα
37.974715	23.744888	22	Βυζαντινό και Χριστιανικό Μουσείο
37.975327	23.745339	34	Πολεμικό Μουσείο
37.988925	23.730984	77	Αρχαιολογικό Μουσείο
37.977906	23.735510	44	Νομισματικό Μουσείο
37.981048	23.733005	65	Εθνική Βιβλιοθήκη
37.977746	23.713431	90	Τεχνόπολις Δήμου Αθηνών
37.972942	23.732451	45	Μουσείο Ελληνικής Παιδικής Τέχνης

**Πίνακας 1: 12 σημεία ενδιαφέροντος στο κέντρο της Αθήνας.**

Στον Πίνακα 1 παρουσιάζονται 12 πολύ γνωστά σημεία ενδιαφέροντος που βρίσκονται στο κέντρο της Αθήνας. Δίνονται οι συντεταγμένες τους και η βαθμολογία που τους αποδίδεται. Πρόκειται για σημεία αρχαιολογικού και πολιτιστικού ενδιαφέροντος και έχουν επιλεγεί τυχαία για να βοηθήσουν στο παράδειγμα της παρούσας ενότητας.

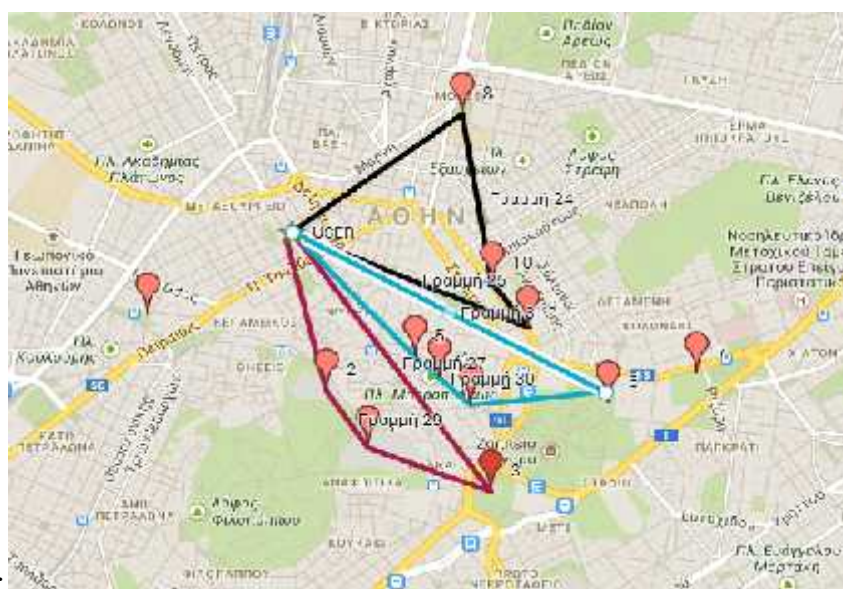


**Εικόνα 9: Αρχική εικόνα που θα έβλεπε ο τουρίστας που συνδέεται σε μία εφαρμογή προσωποποιημένης πλοήγησης πριν τη χάραξη της διαδρομής που θα ακολουθήσει.**

Στην Εικόνα 9 βλέπουμε την αρχική εικόνα που θα έβλεπε ένας χρήστης- τουρίστας όταν θα συνδεόταν σε μία εφαρμογή προσωποποιημένης πλοήγησης. Με το κίτρινο αστέρι εμφανίζεται η θέση του χρήστη ενώ τα κόκκινα σημεία αποτελούν τα σημεία ενδιαφέροντος που βρίσκονται στην περιοχή του και μπορεί να επισκεφθεί.

Το επόμενο στάδιο είναι να χαραχθεί η βέλτιστη διαδρομή βάση των περιορισμών του χρήστη. Στην περίπτωση μας, που είναι και η απλούστερη, οι περιορισμοί είναι μόνο ένας, ο χρονικός περιορισμός. Για να χαραχθεί η διαδρομή πρέπει να αποφασίσουμε τι είδος διαδρομής θέλουμε. Μπορούμε να διακρίνουμε τρία είδη διαδρομών:

1. Στο αρχικό σημείο βρίσκονται όχι ένας μόνο τουρίστας αλλά ένα γκρουπ τουριστών που θέλουν είτε να χωριστούν σε υποομάδες και να πραγματοποιήσουν διαφορετικές διαδρομές - ξεναγήσεις είτε ο κάθε ένας μόνος του να ακολουθήσουν από μία διαδρομή. Όποια και αν είναι η περίπτωση καταλήγουμε στην ίδια διαδικασία. Με αφετηρία ένα σημείο και τερματισμό το ίδιο σημείο (ώστε όλο το γκρουπ να ξανασυναντηθεί μετά τις ξεναγήσεις) πρέπει να χαραχθούν πολλαπλές διαδρομές. Η λύση θα είναι τύπου «μαργαρίτας» και σχηματικά αυτό αναπαρίσταται στην Εικόνα 10.

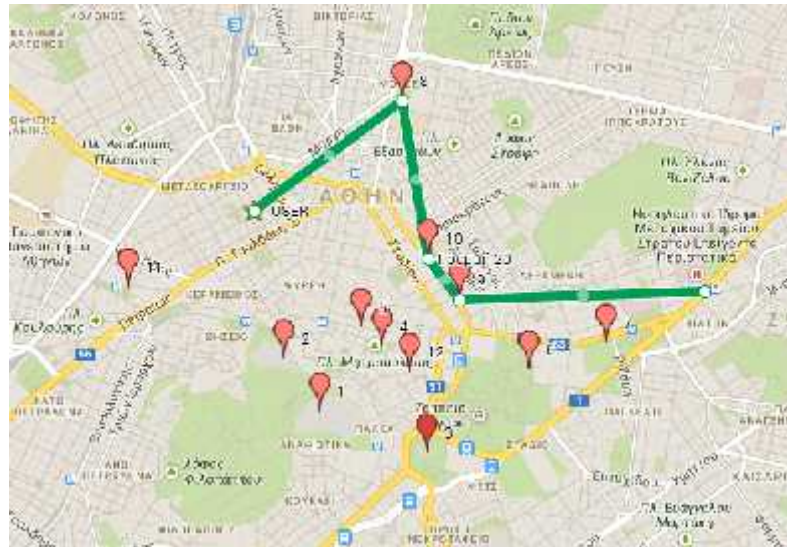


**Εικόνα 10: Χάραξη περισσότερων των μία διαδρομών και προβολή τους πάνω στον χάρτη που εμφανίζεται στους χρήστες.**

Στην Εικόνα 10 απεικονίζονται οι τελικές διαδρομές που θα είχαν χαραχθεί για την περίπτωση των πολλαπλών τουριστών που ξεκινούν από ένα σημείο και καταλήγουν πίσω στο ίδιο. Το πρόβλημα αυτό είναι ανάλογο με το πρόβλημα του Ομαδικού Προσανατολισμού (TOP).



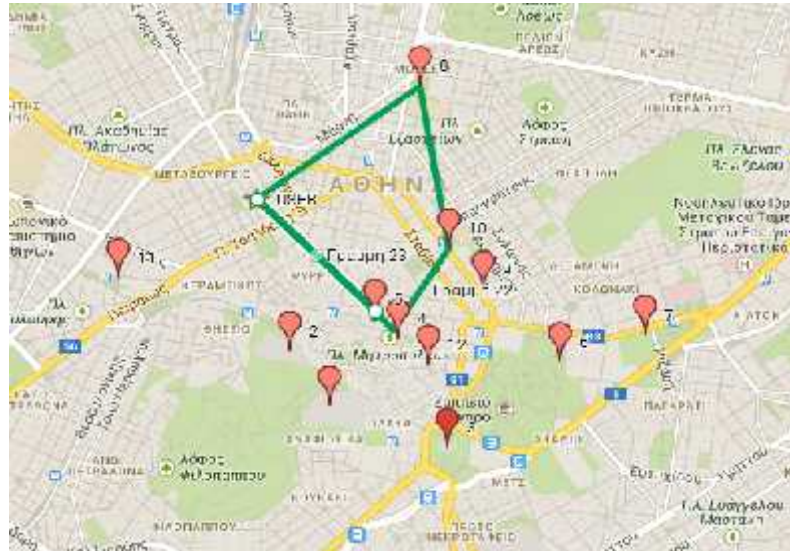
2. Στο αρχικό σημείο βρίσκεται μόνο ένας τουρίστας, θέλει να ακολουθήσει μία διαδρομή και να καταλήξει στο σημείο τερματισμού που είναι διαφορετικό από το σημείο από όπου ξεκίνησε. Σχηματικά αυτό αναπαριστάται στην Εικόνα 11.



Εικόνα 11: Χάρση μιας διαδρομής για ένα τουρίστα με διαφορετικά σημεία εκκίνησης και τερματισμού και προβολή της πάνω στον χάρτη που εμφανίζεται στον χρήστη.

*Στην Εικόνα 11 απεικονίζεται η τελική διαδρομή που θα είχε χαραχθεί για ένα τουρίστα που ξεκινά από ένα σημείο και καταλήγει σε ένα διαφορετικό.*

3. Στο αρχικό σημείο βρίσκεται μόνο ένας τουρίστας και θέλει να ακολουθήσει μία διαδρομή και να επιστρέψει στο σημείο από όπου ξεκίνησε. Σχηματικά αυτό αναπαρίσταται στην Εικόνα 12.



**Εικόνα 12:** Χάρση μιας διαδρομής για ένα τουρίστα όταν τα σημεία εκκίνησης και τερματισμού ταυτίζονται και προβολή της πάνω στον χάρτη που εμφανίζεται στον χρήστη.

Στην Εικόνα 12 απεικονίζεται η τελική διαδρομή που θα είχε χαραχθεί για ένα τουρίστα που ξεκινά από ένα σημείο και καταλήγει πίσω στο ίδιο. Το πρόβλημα αυτό είναι ανάλογο με το πρόβλημα του Προσανατολισμού με ένα παίχτη (Single-Competitor OP).

Συνεπώς, στην γενικευμένη περίπτωση που υπάρχουν περισσότεροι του ενός χρήστη σε ένα σημείο εκκίνησης π.χ. ένα γκρουπ τουριστών που ο κάθε ένας θέλει να πραγματοποιήσει μία διαδρομή μόνος του, θεωρούμε ότι ο κάθε χρήστης/τουρίστας δεν μπορεί να επισκεφτεί όλα τα σημεία ενδιαφέροντος που βρίσκονται στην περιοχή του λόγω του περιορισμένου χρόνου που διαθέτει. Έτσι πρέπει να επιλεγεί η διαδρομή αυτή, που μέσα στον χρονικό περιορισμό, θα του αποδίδει τη μέγιστη χρησιμότητα, ωφέλεια από την επίσκεψη των σημείων που προτείνονται από τη διαδρομή. Το πρόβλημα αυτό είναι παρόμοιο με το πρόβλημα του Ομαδικού Προσανατολισμού (TOP) κατά το οποίο με δεδομένο ένα σύνολο σημείων κάθε μέλος της ομάδας μέσα στον προκαθορισμένο χρόνο περνάει από όσα περισσότερα σημεία μπορεί και λαμβάνει και τη σχετική βαθμολογία που του αποδίδει το κάθε σημείο.



Στην παρούσα μελέτη μελετάται η περίπτωση κατά την οποία υπάρχει μόνο ένας μεμονωμένος χρήστης που θα ακολουθήσει μία διαδρομή και θα καταλήξει στο σημείο από όπου ξεκίνησε. Το πρόβλημα αυτό μοντελοποιείται με χρήση της ειδικότερης περίπτωσης του TOP, του προβλήματος του Προσανατολισμού με ένα παίχτη (Single-Competitor OP).

Στη συνέχεια η μελέτη επικεντρώνεται στην ανάλυση των αλγορίθμων και διαδικασιών που υπεισέρχονται στο στάδιο χάραξης της βέλτιστη διαδρομής, παρουσιάζονται αλγοριθμικές προσεγγίσεις ενώ προτείνεται ένας αλγόριθμος για το OP που υλοποιήθηκε στα πλαίσια της παρούσας μελέτης.

## 4. ΕΠΙΣΚΟΠΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΚΑΙ ΔΙΑΔΙΚΑΣΙΩΝ

Όπως έχουμε ήδη αναφέρει στο πλαίσιο της προσωποποιημένης πλοήγησης βασικό στοιχείο αποτελεί η κατασκευή της βέλτιστης διαδρομής που θα ακολουθήσει ο χρήστης. Συγκεκριμένα, όπως έχει ήδη αναφερθεί, στην παρούσα μελέτη το πρόβλημα προσεγγίζεται με χρήση του OP.

Σε αυτό το κεφάλαιο παρουσιάζονται τα στάδια που υπεισέρχονται στην εύρεση της βέλτιστης διαδρομής, οι διαδικασίες και οι αλγόριθμοι που εμφανίζονται στη βιβλιογραφία και βρίσκουν εφαρμογή στα TOP και OP αλλά και στο TSP.

### 4.1 ΕΥΡΕΣΗ ΒΕΛΤΙΣΤΗΣ ΔΙΑΔΡΟΜΗΣ

Η διαδικασία μέσα από την οποία ένας ευρετικός αλγόριθμος βρίσκει μια βέλτιστη διαδρομή μεταξύ ενός δοθέντος συνόλου από σημεία ενδιαφέροντος και κάτω από συγκεκριμένους περιορισμούς διακρίνεται σε δύο βασικά στάδια. Πρώτον κατασκευή της αρχικής διαδρομής και δεύτερον βελτιστοποίηση της. Σε κάθε στάδιο υπάρχουν διαφορετικοί αλγόριθμοι που μπορούν να λάβουν χώρα.

#### ΚΑΤΑΣΚΕΥΗ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ

Η κατασκευή της αρχικής διαδρομής πραγματοποιείται από κατασκευαστικούς αλγόριθμους (construction heuristics), οι αλγόριθμοι αυτοί κατασκευάζουν μία διαδρομή βάση ορισμένων κανόνων και δεν προσπαθούν να τη βελτιώσουν.

Βασικά ερωτήματα που πρέπει να απαντηθούν για το σχεδιασμό ενός τέτοιου αλγορίθμου είναι:

- Με ποιο τρόπο θα κατασκευαστεί η αρχική διαδρομή.
- Με ποιο τρόπο θα επιλέγεται κάθε φορά το επόμενο σημείο που θα εισαχθεί στη διαδρομή.
- Σε ποιο σημείο της διαδρομής θα μπει το επόμενο σημείο.

Ένα πολύ απλό παράδειγμα θα ήταν η χρήση του αλγορίθμου του κοντινότερου γείτονα για επίλυση του TSP. Ο αλγόριθμος αυτός πολλές φορές στην προσπάθεια αναζήτησης του πλησιέστερου γείτονα, παραβλέπει κάποια σημεία τα οποία πρέπει να εισάγει στο τέλος της διαδρομής με μεγάλο κόστος. Αν και τα αποτελέσματα που δίνουν αυτοί οι αλγόριθμοι δεν είναι τα βέλτιστα, οι διαδρομές συνήθως δεν περιέχουν σοβαρά λάθη και έτσι αποτελούν καλή βάση για την εύρεση μία τελικώς βέλτιστης λύσης.

Άλλος τρόπος κατασκευής μίας αρχικής διαδρομής είναι μέσω των αλγορίθμων εισαγωγής (insertion heuristics). Για παράδειγμα για το TSP η κατασκευή της αρχικής διαδρομής θα μπορούσε να γίνει ξεκινώντας από μία διαδρομή που ενώνει ένα υποσύνολο των σημείων και μετά να επεκταθεί προσθέτοντας σε αυτή τα σημεία που απομένουν, το ένα μετά το άλλο.

Συνήθως διακρίνονται δύο είδη εισαγωγής σημείων:

- Cheapest insertion: Με τον τρόπο αυτό, κάθε φορά επιλέγεται το σημείο που θα προκαλέσει την μικρότερη αύξηση στο μήκος, διάρκεια της διαδρομής. Συνήθως τέτοιες μέθοδοι χρησιμοποιούνται μαζί με άπληστους αλγόριθμους.
- Farthest insertion: Με τον τρόπο αυτό, κάθε φορά επιλέγεται το πιο απομακρυσμένο σημείο.

Για την κατασκευή της αρχικής διαδρομής μπορούν να χρησιμοποιηθούν στοχαστικές μέθοδοι, να κατασκευαστούν διαδρομές επιλέγοντας τυχαία σημεία, να κατασκευαστούν διαδρομές με την χρήση παλινδρόμησης ή πολυκριτηριακής ανάλυσης.

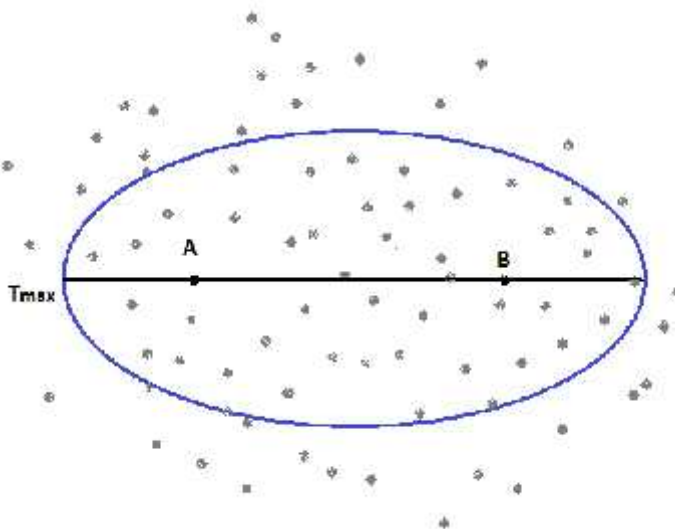
Στην συνέχεια παρουσιάζεται ο αλγόριθμος construct. Πρόκειται για έναν άπληστο ευρετικό αλγόριθμο που επιλέγει μία αρχική διαδρομή μέσα σε μικρό χρονικό διάστημα και έχει χρησιμοποιηθεί με μία μικρή παραλλαγή και στον αλγόριθμο που υλοποιήθηκε στα πλαίσια της παρούσας μελέτης.

## CONSTRUCT

Ο αλγόριθμος παρουσιάζεται για πρώτη φορά από τους Chao, Golden and Wasil (1996a,b) σε δύο εκδοχές μία για το TOP και μια για το OP άλλωστε το orienteering problem όπως περιγράφηκε από τους Chao, Golden and Wasil (1996) μπορεί να θεωρηθεί ως μία ειδική περίπτωση του TOP, δηλαδή ως TOP με έναν παίχτη, Chao et al. (1996a).

Ο Chao et al. (1996a) παρουσιάζουν έναν ευρετικό αλγόριθμο επίλυσης του TOP που αποτελείται από δύο διακριτά τμήματα, το πρώτο επιλέγει μία αρχική διαδρομή και το δεύτερο αφορά την βελτιστοποίηση της αρχικής διαδρομής μέσα από διάφορους μηχανισμούς. Το πρώτο κομμάτι που το ονομάζουν «initialization» επιλέγει μία αρχική διαδρομή και έχει χρησιμοποιηθεί και σε άλλες εργασίες από τους ίδιους συγγραφείς και άλλους.

Για να γίνει κατανοητός ο τρόπος λειτουργίας του αλγορίθμου ας θεωρήσουμε την περιοχή με τα σημεία ως ένα γεωμετρικό τόπο. Βασισμένοι στον στοχαστικό αλγόριθμο του Tsiligirides (1984) για επίλυση του OP, ο αλγόριθμος που προτείνουν οι συγγραφείς αρχικά κατασκευάζει μία έλλειψη πάνω από τον τόπο όπου περιέχεται το σύνολο των σημείων. Τα σημεία έναρξης και τερματισμού αποτελούν τις εστίες της έλλειψης και ο χρονικός περιορισμός αποτελεί τον μεγάλο άξονα της έλλειψης όπως φαίνεται στην Εικόνα 13.



Εικόνα 13: Δημιουργία έλλειψης

Εν συνεχεία μόνο τα σημεία που βρίσκονται μέσα στην έλλειψη θα χρησιμοποιηθούν στην επίλυση του προβλήματος αφού θεωρείται ότι μόνο

αυτά προλαβαίνουν οι παίχτες να επισκεφθούν. Στόχος είναι να κατασκευαστούν τόσες διαδρομές όσες και τα άτομα της ομάδας. Υποθέτουμε ότι περιέχονται  $N$  σημεία μέσα στην έλλειψη, και η ομάδα που λαμβάνει μέρος στο TOP αποτελείται από  $M$  άτομα. Αρχικά επιλέγονται  $L = \min(5, N)$  σημεία έτσι ώστε να είναι τα πιο απομακρυσμένα από τα σημεία εκκίνησης και τερματισμού.  $M$  από το  $L$  σημεία επιλέγονται ως τα πρώτα σημεία που θα εισαχθούν σε κάθε διαδρομή. Για να βρεθεί η  $L$ -η λύση, το μονοπάτι κατασκευάζεται βρίσκοντας το  $L$  σημείο που είναι πιο μακριά από τα σημεία εκκίνησης και τερματισμού και δημιουργείται ένα μονοπάτι με αυτά τα 3 σημεία. Τα υπόλοιπα σημεία εισάγονται στην διαδρομή με άπληστο τρόπο, εισάγονται πρώτα τα σημεία που βρίσκονται πιο κοντά είτε βάση της απόστασης είτε βάση του χρόνου μετάβασης που απαιτείται από το ένα σημείο στο άλλο μέχρις ότου η προσθήκη ενός ακόμα σημείου να παραβιάζει τον χρονικό περιορισμό. Με αυτό τον τρόπο κατασκευάζονται  $M$  διαδρομές.

Αν υπάρχουν σημεία που δεν έχουν χρησιμοποιηθεί σε καμία διαδρομή κατασκευάζονται εκ νέου καινούριες διαδρομές μέχρι ότου όλα τα σημεία να ανήκουν σε κάποια διαδρομή. Τέλος επιλέγονται οι  $M$  διαδρομές που αν τις ακολουθήσει κάποιος παίχτης θα λάβει την υψηλότερη δυνατή βαθμολογία και άρα θα δίνουν την υψηλότερη βαθμολογία και στην ομάδα. Οι διαδρομές αυτές στο επόμενο μέρος του αλγορίθμου βελτιώνονται με διάφορες μεθόδους που δεν μας απασχολούν στο σημείο αυτό.

Να σημειωθεί ότι αν για ένα πρόβλημα  $L > M$  τότε υπάρχουν  $(LM)$  διαφορετικές λύσεις ενώ στην περίπτωση που  $L \leq M$  σημαίνει ότι  $L = N$  και τότε ο κάθε παίχτης θα επισκεφτεί το πολύ ένα σημείο, άρα η βέλτιστη λύση μπορεί να βρεθεί.

Αργότερα οι Chao et al. (1996b) χρησιμοποιούν την προηγούμενη διαδικασία στην προσπάθειά τους να προτείνουν μία λύση για το OP με την διαφορά ότι κατασκευάζονται για κάθε πρόβλημα  $L = \min(10, N)$  διαδρομές. Τον ίδιο αλγόριθμο συναντούμε και στους Vasteenwegen, Souffriau, Vanden Berghe, Van Oudheusden (2008) αλλά και οι Souffriau, Vasteenwegen, Vertommen, Vanden Berghe, Van Oudheusden (2008) προτείνουν ένα μεταυρετικό αλγόριθμο για το OP και χρησιμοποιούν τον αλγόριθμο αυτό με την διαφορά ότι τα σημεία εκκίνησης και τερματισμού ταυτίζονται, η έλλειψη αντικαθίσταται από κύκλο και από τις διαδρομές που κατασκευάζονται επιλέγεται αυτή με την μεγαλύτερη βαθμολογία για περαιτέρω βελτιστοποίηση.

## ΒΕΛΤΙΩΣΗ ΤΗΣ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ

Οι διαδρομές που δίνονται από το προηγούμενο στάδιο δεν είναι οι βέλτιστες αλλά αποτελούν τη βάση για την εύρεση της βέλτιστης ή σχεδόν βέλτιστης λύσης. Οι αλγόριθμοι που υπεισέρχονται σε αυτή τη διαδικασία είναι αρκετοί και σχεδόν ποτέ δεν μπορούμε να περιοριστούμε μόνο στην χρήση ενός για την βελτίωση της διαδρομής. Η εύρεση της βέλτιστης διαδρομής γίνεται μέσα από τον συνδυασμό αρκετών διαδικασιών.

Οι διαδικασίες αυτές μπορούν να χωριστούν σε τρεις κατηγορίες (Vasteenwegen 2009):

1. Αλγόριθμοι και διαδικασίες για αύξηση της συνολικής βαθμολογίας που αποδίδει η διαδρομή.
2. Αλγόριθμοι και διαδικασίες για μείωση του κόστους της διαδρομής.
3. Διαδικασίες διαφοροποίησης.

### 4.1.1.1 ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΔΙΑΔΙΚΑΣΙΕΣ ΓΙΑ ΑΥΞΗΣΗ ΤΗΣ ΣΥΝΟΛΙΚΗΣ ΒΑΘΜΟΛΟΓΙΑΣ ΠΟΥ ΑΠΟΔΙΔΕΙ Η ΔΙΑΔΡΟΜΗ

Όπως έχει ήδη αναφερθεί μία καλή αν όχι βέλτιστη διαδρομή πρέπει να αποδίδει υψηλή βαθμολογία στον χρήστη. Στην λογική αυτή για τη βελτιστοποίηση μίας αρχικής διαδρομής μπορούν να χρησιμοποιηθούν, όσο δεν παραβιάζεται ο χρονικός περιορισμός, οι μέθοδοι που παρουσιάζονται στην συνέχεια.

**Insert, one in - zero out:** Τα σημεία που δεν έχουν χρησιμοποιηθεί στην κατασκευή της αρχικής διαδρομής εισάγονται σε αυτή. Η εισαγωγή συνήθως γίνεται βρίσκοντας την καλύτερη θέση για κάθε ένα από αυτά, τη θέση δηλαδή μεταξύ των ήδη υπαρχόντων σημείων της διαδρομής που θα οδηγήσει στην μικρότερη δυνατή αύξηση του κόστους της διαδρομής.

**TwoInsert:** Είναι διαδικασία παρόμοια με την προηγούμενη μόνο που εδώ εισάγονται κάθε φορά δύο γειτονικά ή μη σημεία που δεν μετέχουν στην διαδρομή.

**Replace:** Κάθε σημείο που δεν μετέχει στην διαδρομή αξιολογείται για το αν θα προστεθεί σε αυτή. Στην απλή της μορφή για κάθε σημείο της διαδρομής βρίσκεται ο κοντινότερος κόμβος που δεν μετέχει σε αυτή. Έπειτα εξετάζεται εάν η αντικατάσταση των δύο αυτών σημείων είναι συμφέρουσα. Με τον όρο «συμφέρουσα» εννοούμε ότι η αντικατάσταση μπορεί να οδηγήσει σε αύξηση της βαθμολογίας, μείωση του κόστους της διαδρομής ή και τα δύο.

#### 4.1.1.2 ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΔΙΑΔΙΚΑΣΙΕΣ ΓΙΑ ΜΕΙΩΣΗ ΤΟΥ ΚΟΣΤΟΥΣ ΤΗΣ ΔΙΑΔΡΟΜΗΣ

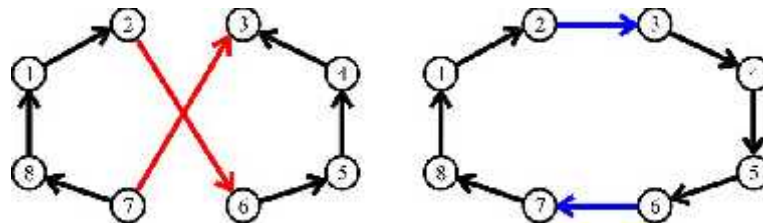
Όσο μικρότερο είναι το κόστος μία διαδρομής τόσο περισσότερα σημεία μπορούν να προστεθούν σε αυτή και άρα να αυξηθεί η βαθμολογία της. Απλή αφαίρεση σημείων από την διαδρομή δεν έχει νόημα αφού μία τέτοια κίνηση θα οδηγούσε ενδεχομένως σε μείωση της βαθμολογίας. Οι διαδικασίες που υπεισέρχονται σε αυτό το στάδιο συνήθως μειώνουν το κόστος της διαδρομής επιλύοντας τις διασταυρώσεις που τυχόν να έχουν προκύψει.

**Swap:** Αναδιατάσσονται, εναλλάσσονται τα σημεία που μετέχουν στη διαδρομή κατά τέτοιο τρόπο ώστε να επιλύονται οι διασταυρώσεις. Η βασική ιδέα παρουσιάστηκε πρώτη φορά από τον Croes το 1958 και έκτοτε έχει εφαρμοστεί σε πολλές περιπτώσεις με διάφορους όρους όπως transpose, interchange, ενώ αρκετά συχνή είναι η λανθασμένη αναφορά του ως 2-opt.

Έστω 4 διαδοχικά σημεία **a,b,c,d** μίας διαδρομής, εάν διασταυρώνονται τότε η εναλλαγή από **a,b,c,d** σε **a,c,b,d** θα οδηγήσει σε πραγματοποίηση μικρότερης διαδρομής. Η βαθμολογία της διαδρομής παραμένει αμετάβλητη αφού δεν αλλάζουν τα σημεία που μετέχουν στη διαδρομή.

Ο αλγόριθμος ελέγχει στη διαδρομή κάθε διατεταγμένη 4άδα ξεκινώντας από το σημείο εκκίνησης της διαδρομής και επιλέγεται ως τελική διαδρομή αυτή με το μικρότερο κόστος.

**2-opt:** Κατά την διαδικασία αυτή επιλέγονται με διάφορους τρόπους (τυχαία, σε συγκεκριμένες αποστάσεις κ.α.) δύο τόξα της διαδρομής. Με τον όρο τόξο αναφερόμαστε στην διαδρομή που ενώνει δύο σειριακά σημεία ενδιαφέροντος. Στην συνέχεια γίνεται θραύση των δύο αυτών τόξων και επανασύνδεση των σημείων με όλους τους δυνατούς τρόπους και επιλέγεται η διαδρομή που οδηγεί στο μικρότερο κόστος (Εικόνα 14). Συχνά εφαρμόζεται και ένα ακόμα βήμα. Με την θραύση των δύο τόξων η διαδρομή σπάει σε τρία κομμάτια, το μεσαίο αναδιατάσσεται με σκοπό να επιλυθούν ακόμη περισσότερες διασταυρώσεις και έπειτα εξετάζονται οι δυνατοί συνδυασμοί επανασύνδεσης της διαδρομής. Εκτός από το 2-opt υπάρχουν, στην ίδια λογική, και διαδικασίες 3-opt κ.οκ.



**Εικόνα 4: Παράδειγμα εφαρμογής 2-opt.**

Στην Εικόνα 14 δίνεται ένα παράδειγμα εφαρμογής της 2-opt διαδικασίας. Τα τόξα μεταξύ των σημείων 2-6 και 7-3 “σπάνε” και η διαδρομή συνδέεται με τέτοιο τρόπο ώστε να γίνεται αφαίρεση των διασταυρώσεων (η διαδρομή θα δίνει το μικρότερο κόστος). Στο αριστερό σχήμα απεικονίζεται η διαδρομή πριν την εφαρμογή της διαδικασίας ενώ στο δεξιό σχήμα παρουσιάζεται το τελικό αποτέλεσμα.



#### 4.1.1.3 ΔΙΑΔΙΚΑΣΙΕΣ ΔΙΑΦΟΡΟΠΟΙΗΣΗΣ

Επειδή οι ευρετικοί αλγόριθμοι συνήθως τερματίζουν με την εύρεση κάποιας τοπικά βέλτιστης λύσης εφαρμόζονται διαδικασίες διαφοροποίησης της διαδρομής με σκοπό να απεγκλωβιστεί ο αλγόριθμος από αυτή την κατάσταση ώστε να είναι εφικτή η εύρεση μίας ολικά βέλτιστης ή σχεδόν βέλτιστης λύσης. Στην προσπάθεια αυτή μέρος της διαδρομής αφαιρείται και στην συνέχεια η διαδρομή βελτιστοποιείται χρησιμοποιώντας κάποιες από τις προηγούμενες διαδικασίες που αναφέρθηκαν. Στην πράξη αφαιρούνται είτε ποσοστό των σημείων είτε συγκεκριμένος αριθμός σημείων από την αρχή της διαδρομής ή το τέλος και έπειτα οι αλγόριθμοι βελτιστοποίησης της διαδρομής ξεκινούν από την αρχή (προστίθενται σημεία, επιλύονται διασταυρώσεις κ.ο.κ).

## 4.2 ΑΛΓΟΡΙΘΜΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ

Όπως εξηγήθηκε σε προηγούμενο κεφάλαιο, το πρόβλημα που μελετάται στην παρούσα εργασία μπορεί να προσεγγιστεί μέσω του ΟΡ. Στην παρούσα ενότητα παρουσιάζονται συνοπτικά τέσσερις διαφορετικές αλγοριθμικές προσεγγίσεις για το ίδιο πρόβλημα, δύο αλγόριθμοι του Tsiligirides (1984), ένας του Keller(1985) και ένας του Golden (1985). Τα αποτελέσματα των αλγορίθμων αυτών συγκρίνονται, σε επόμενο κεφάλαιο, με τον αλγόριθμο που υλοποιήθηκε στο πλαίσιο της παρούσας μελέτης με σκοπό να αξιολογηθεί ο αλγόριθμος αυτός. Για την σύγκριση των αποτελεσμάτων χρησιμοποιούνται τρία σύνολα σημείων που δίνονται από τον Tsiligirides (1984), προέρχονται από πραγματικό Orienteering παιχνίδι και παρουσιάζονται στη συνέχεια.

### ΣΤΟΧΑΣΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ

Ο Tsiligirides (1984) παρουσιάζει δύο ευρετικούς αλγορίθμους για την κατασκευή της αρχικής διαδρομής για λύση του ΟΡ. Ο πρώτος βασίζεται σε στοχαστική προσέγγιση με χρήση Monte Carlo τεχνικής, για την παραγωγή μεγάλου αριθμού λύσεων, κατά την οποία παράγονται 3000 λύσεις/διαδρομές και επιλέγεται η καλύτερη. Ο δεύτερος βασίζεται σε αλγόριθμο των Wren και Holliday (1972) για το vehicle scheduling problem και οι διαδρομές κατασκευάζονται με τα σημεία που βρίσκονται κάθε φορά μέσα σε δύο ομόκεντρους δακτυλίους και επιλέγεται η διαδρομή με τη μεγαλύτερη βαθμολογία.

Τέλος προτείνεται και ένας τρίτος αλγόριθμος για βελτίωση των αρχικών διαδρομών που αποτελείται από τρεις διαδικασίες, αντιμετάθεση σημείων για επίλυση των διασταυρώσεων, εισαγωγή σημείων και τέλος εισαγωγή ενός σημείου-εξαγωγή ενός σημείου. Η πλήρης ανάλυση του αλγορίθμου παρέχεται στο Tsiligirides (1984).

## ΑΛΓΟΡΙΘΜΟΣ ΒΑΣΗ ΤΗΣ ΠΟΛΥΚΡΙΤΗΡΙΑΚΗΣ ΑΝΑΛΥΣΗΣ

Ο Keller (1988) παρουσιάζει μία γενίκευση του προβλήματος του πλανόδιου πωλητή, το Multidjective Vending Problem, κατά την οποία με δεδομένο ένα σύνολο σημείων δεν είναι απαραίτητο να μετέχουν όλα τα σημεία ενός συνόλου σε μία διαδρομή ενώ προτείνεται και ένας αλγόριθμος που επιλύει το πρόβλημα.

Στο MVP κάθε σημείο αποδίδει μία βαθμολογία ενώ η μετάβαση από ένα σημείο σε ένα άλλο αποδίδει ένα πέναλτι. Το πρόβλημα αντιμετωπίζεται στα πλαίσια του πολυκριτηριακού προγραμματισμού, χωρίς να είναι γνωστός ο οριακός λόγος υποκατάστασης μεταξύ βαθμολογίας και πέναλτι, όπου ένας ευρετικός αλγόριθμος προσπαθεί να βρει κάθε φορά την σχέση μεταξύ αυτών των δύο που θα οδηγήσει στην εύρεση της βέλτιστης διαδρομής. Το OP μπορεί να θεωρηθεί ότι είναι μία ειδική περίπτωση του MVP εφόσον η απόσταση μεταξύ των σημείων έναρξης και τερματισμού είναι μηδενική, δηλαδή ή ταυτίζονται ή στην πραγματικότητα τα δύο αυτά σημεία είναι πολύ κοντά.

Αναλυτικότερα, βασικό στοιχείο της μεθοδολογίας του Keller αποτελεί η αύξηση της βαθμολογίας μίας διαδρομής ενώ παράλληλα χρησιμοποιούνται σε αυτή όσο το δυνατόν λιγότερα σημεία, άρα η ύπαρξη δύο αντικειμενικών συναρτήσεων που πρέπει να βελτιστοποιηθούν. Μία συνάρτηση βελτιστοποίησης για την συνολική βαθμολογία της διαδρομής όπου ο αριθμός των σημείων που μετέχουν σε αυτή θα πρέπει να είναι όσο το δυνατόν μεγαλύτερος και μία δεύτερη για την ελαχιστοποίηση του συνολικού πέναλτι που προκύπτει από τη διαδρομή, όπου τα σημεία που μετέχουν σε αυτή θα πρέπει να είναι όσο το δυνατόν λιγότερα. Ο ευρετικός αλγόριθμος που προτείνεται έχει ικανοποιητική απόδοση όταν εφαρμόζεται σε προβλήματα που περιέχουν 25 σημεία, Keller (1988). Η πλήρης ανάλυση του αλγορίθμου παρέχεται στο Keller (1988). Τα αποτελέσματα του αλγορίθμου, όπως παρουσιάζονται σε επόμενο κεφάλαιο και με βάση τα οποία γίνεται η σύγκρισή, δίνονται στην δημοσίευση του ίδιου το 1989 και προέρχονται από την αδημοσίευτη διατριβή του που ολοκληρώθηκε το 1985.

Με τον όρο «πολυκριτηριακός προγραμματισμός» ή «πολυκριτηριακή ανάλυση» ή αλλιώς «αριστοποίηση κατά Pareto» αναφερόμαστε στην επίλυση μαθηματικών προβλημάτων βελτιστοποίησης με περισσότερες της μίας αντικειμενικές συναρτήσεις υποκείμενες η κάθε μία στους περιορισμούς της. Ο πολυκριτηριακός προγραμματισμός εφαρμόζεται εκτενώς σε πολλές επιστήμες όπως η μηχανική και η οικονομία όπου η λήψη μίας απόφασης βασίζεται στον καθορισμό της σχέσης μεταξύ δύο ή περισσότερων

παραγόντων π.χ. ο καθορισμός της παραγόμενης ποσότητας ενός αγαθού με τον μικρότερο αριθμό εργατοωρών και την κατανάλωση όσο το δυνατόν λιγότερης ενέργειας που θα οδηγήσει σε μεγιστοποίηση του κέρδους για μία εταιρεία.

### **ΑΛΓΟΡΙΘΜΟΣ ΒΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΤΟΥ ΣΑΚΙΔΙΟΥ (Knapsack)**

Οι Golden et al. (1985, 1987) παρουσιάζουν έναν αλγόριθμο που προτείνει λύση για τον ΟΡ και βασίζεται στο πρόβλημα Knapsack και αποτελείται από πέντε βήματα. Κατασκευή αρχικών διαδρομών εκ των οποίων επιλέγεται η καλύτερη, δύο ρουτίνες βελτίωσης εκ των οποίων η μία βασίζεται στον Tsiligirides (1985) και η δεύτερη εισάγει σημεία στη διαδρομή με τη μέθοδο cheapest insertion, ένα βήμα εισαγωγής σημείων στη διαδρομή που βασίζεται στο πρόβλημα του σακιδίου και μία ρουτίνα για διαφοροποίηση της διαδρομής.

Τα αποτελέσματα του αλγορίθμου, όπως παρουσιάζονται στην επόμενη ενότητα και με βάση τα οποία γίνεται η σύγκρισή, δίνονται από τον Keller (1989).

Αξίζει να αναφερθεί ότι στο πρόβλημα του σακιδίου (Knapsack) θεωρούμε ότι υπάρχουν  $n$  διαφορετικά αντικείμενα με συγκεκριμένη αξία και βάρος και ένας σάκος με συγκεκριμένη χωρητικότητα. Σκοπός είναι να βρεθεί ποια αντικείμενα πρέπει να τοποθετηθούν μέσα στον σάκο χωρίς να γίνει υπέρβαση του μέγιστου συνολικού φορτίου που μπορεί να φέρει ενώ παράλληλα θα μεγιστοποιείται η συνολική αξία.

## 5. ΠΡΟΤΕΙΝΟΜΕΝΟΣ ΑΛΓΟΡΙΘΜΟΣ

Ένας από τους σκοπούς της παρούσας μελέτης είναι και η ανάλυση του προβλήματος εύρεσης μίας καλής αν όχι της βέλτιστης διαδρομής στο πλαίσιο της προσωποποιημένης πλοήγησης σε εξωτερικούς χώρους. Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο το πρόβλημα αυτό μπορεί να περιγραφεί με τη χρήση του TOP. Ως βάση για περαιτέρω εργασία και έρευνα αναπτύχθηκε ένας αλγόριθμος για το OP που θεωρείται ειδικότερη περίπτωση του TOP. Στο παρόν κεφάλαιο παρουσιάζεται ο προτεινόμενος αλγόριθμος και αξιολογείται η λειτουργία του βάση των τεσσάρων αλγοριθμικών προσεγγίσεων που παρουσιάστηκαν σε προηγούμενο κεφάλαιο.

Ο αλγόριθμος είναι εμπνευσμένος από τον αλγόριθμο Guided local Search (GLS). Ο GLS είναι ένας μεταερευτικός αλγόριθμος που δίνει λύση στο OP και σε κάποιες επεκτάσεις του. Δημιουργήθηκε από τους Wouter Souffriau, Pieter Vasteeuwegen, Greet Vander Berghe, Dirk Van Oudheusden και παρουσιάζεται στο “A personalized Tourist Trip Design Algorithm for Mobile Tourist Guides” και αναλυτικότερα στο “A guided local search metaheuristic for the team orienteering problem”. Ο GLS συνδυάζει διαφορετικούς ευρέως διαδεδομένους αλγόριθμους και στόχος του ήταν μέσα από την λειτουργία του οι αρχικοί ευρετικοί αλγόριθμοι να συνδυαστούν με τέτοιο τρόπο ώστε να δίνουν τα καλύτερα δυνατά αποτελέσματα στο λιγότερο χρόνο.

## 5.1 ΠΑΡΟΥΣΙΑΣΗ ΑΛΓΟΡΙΘΜΟΥ

Ο αλγόριθμος αναπτύχθηκε με την χρήση του MatLab. Στον Πίνακα 3 παρουσιάζεται ο ευρετικός αλγόριθμος με την μορφή ψευδοκώδικα προς διευκόλυνση του αναγνώστη ενώ αναλυτικότερα ο αλγόριθμος δίνεται στο Παράρτημα.

Για τη λειτουργία του αλγορίθμου θεωρείται δεδομένο το σύνολο των σημείων ενδιαφέροντος που βρίσκονται στην περιοχή του χρήστη, ο μέγιστος χρόνος που έχει στη διάθεση του αλλά και ότι τα σημεία εκκίνησης και τερματισμού είναι πολύ κοντά μεταξύ τους.

```
Loop=0

ΚΑΘΟΡΙΣΜΟΣ ΕΦΙΚΤΟΥ ΣΥΝΟΛΟΥ
ΚΑΤΑΣΚΕΥΗ ΔΙΑΔΡΟΜΩΝ
ΕΠΙΛΟΓΗ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ

ΕΠΑΝΑΛΑΒΕ ΜΕΧΡΙ ΝΑ ΜΗΝ ΥΠΑΡΧΕΙ ΚΑΛΥΤΕΡΗ ΛΥΣΗ ΚΑΙ Loop<5

    ΕΠΙΛΥΣΗ ΔΙΑΣΤΡΑΥΡΩΣΕΩΝ
    ΑΝΤΙΚΑΤΑΣΤΑΣΗ ΣΗΜΕΙΩΝ
    ΕΙΣΑΓΩΓΗ ΣΗΜΕΙΩΝ

ΤΕΛΟΣ

ΕΜΦΑΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΟΣ
```

Πίνακας 2: Ο προτεινόμενος αλγόριθμος για το OP.

## ΚΑΤΑΣΚΕΥΗ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ

Για την κατασκευή της αρχικής διαδρομής χρησιμοποιείται ένας άπληστος ευρετικός αλγόριθμος ο οποίος βασίζεται, με ορισμένες αλλαγές, στον αλγόριθμο «construct» που έχει παρουσιαστεί σε προηγούμενο κεφάλαιο.

Τα βήματα που ακολουθεί ο αλγόριθμος για την κατασκευή της αρχικής διαδρομής είναι τα εξής:

- Αφαιρούνται όλα τα σημεία που βρίσκονται έξω από την μέγιστη ακτίνα της διαδρομής ( $t_{max}$ ). Αφαιρούνται, δηλαδή, τα σημεία που βάση του χρονικού περιορισμού του χρήστη δεν θα προλάβει εξ ορισμού να επισκεφτεί.
- Κατασκευάζονται  $L$  διαδρομές και επιλέγονται  $L$  κοντινότερα σημεία στο σημείο εκκίνησης και τοποθετούνται στις  $L$  διαδρομές, ένα σημείο σε κάθε διαδρομή. Πειραματικά επιλέχθηκε ότι για  $L=3$  προκύπτουν τα καλύτερα αποτελέσματα.
- Τα υπόλοιπα σημεία, που δεν έχουν χρησιμοποιηθεί στο προηγούμενο βήμα, εισάγονται με άπληστο τρόπο στις διαδρομές λαμβάνοντας υπ' όψιν μόνο την απόσταση (nearest neighbor).
- Εάν όλες οι  $L$  διαδρομές είναι πλήρεις, και παραμένουν σημεία που δεν ανήκουν σε κάποια διαδρομή, κατασκευάζονται και άλλες με άπληστο τρόπο μέχρις ότου όλα τα σημεία να ανήκουν σε κάποια διαδρομή.
- Η διαδρομή με την μεγαλύτερη βαθμολογία (score) επιλέγεται ως αρχική λύση.

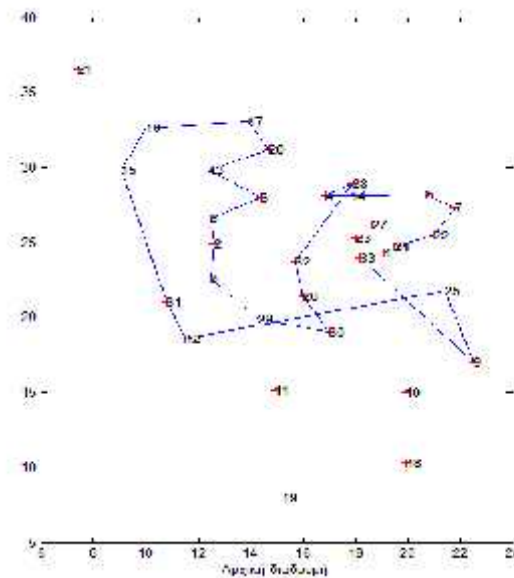
πχ. Όταν εφαρμόστηκε ο αλγόριθμος για το πρόβλημα 3 (βλέπε ενότητα 6.2),  $t_{max} = 85$ , και σημείο εκκίνησης το “1” και σημείο τερματισμού το “33”, κατασκευάστηκαν οι αρχικές διαδρομές:

I. diadromi = 1 24 22 7 5 14 4 28 32 26 30 29 8  
 2 6 3 13 20 17 16 15 31 12 25 9 33  
 με tScore: 590, cost: 83.0856

II. diadromi = 1 23 10 11 18 19 21 33  
 με tScore: 200, cost: 75.2737

III. diadromi = 1 27 33  
 με tScore: 10, cost: 4.1981

Η αρχική διαδρομή που επιλέγεται για να βελτιστοποιηθεί είναι η πρώτη (Εικόνα 15) αφού είναι αυτή με την μεγαλύτερη βαθμολογία (tScore: 590).



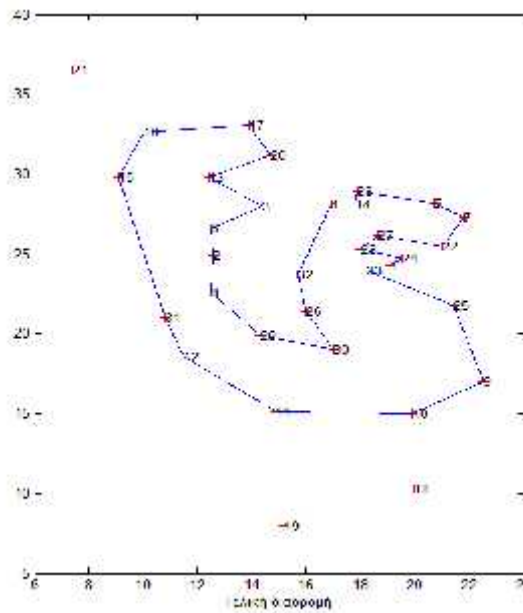
Εικόνα 15: Αρχική διαδρομή για το πρόβλημα 3 και για  $t_{max}=80$ .

Στην Εικόνα 15 απεικονίζεται, όπως προκύπτει από το Matlab, η αρχική διαδρομή που κατασκευάστηκε με χρήση του προτεινόμενου αλγορίθμου για το πρόβλημα 3,  $t_{max}=85$ , με σημείο εκκίνησης το “1” και σημείο τερματισμού το “33”.



Αφού επιλεγεί η αρχική διαδρομή τότε ο αλγόριθμος εισέρχεται στο στάδιο της βελτιστοποίησης.

Η βελτιωμένη διαδρομή (Εικόνα 16), για το προηγούμενο παράδειγμα, αποτελείται από 30 σημεία, έναντι των 26 της αρχικής διαδρομής και έχει tScore: 670 και cost: 83.4124 (η αρχική διαδρομή είχε tScore: 590, cost: 83.0856). Χαρακτηριστικό στοιχείο της τελικής διαδρομής είναι η μη ύπαρξη διασταυρώσεων αφού αυτές έχουν επιλυθεί με τρόπο που παρουσιάζεται στην επόμενη ενότητα.



Εικόνα 16 Τελική διαδρομή για το πρόβλημα 3 και tmax=80.

Στην Εικόνα 16 απεικονίζεται, όπως προκύπτει από το Matlab, η τελική διαδρομή που κατασκευάστηκε με χρήση του αλγορίθμου για το πρόβλημα 3, tmax=85, και σημείο εκκίνησης το "1" και σημείο τερματισμού το "33".

## ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΑΡΧΙΚΗΣ ΔΙΑΔΡΟΜΗΣ

Το στάδιο της βελτιστοποίησης περιλαμβάνει τις παρακάτω διαδικασίες οι οποίες επαναλαμβάνονται μέχρις ότου να μην μπορεί να βρεθεί καλύτερη λύση.

### 5.1.1.1 ΕΠΙΛΥΣΗ ΔΙΑΣΤΑΥΡΩΣΕΩΝ (Swap)

Για την επίλυση των διασταυρώσεων της διαδρομής χρησιμοποιείται ένας αλγόριθμος τοπικής αναζήτησης που σκοπό έχει την μείωση του κόστους της διαδρομής αναδιατάσσοντας τα σημεία που μετέχουν σε αυτή κατά τέτοιο τρόπο ώστε να επιλύονται οι διασταυρώσεις.

Η βασική ιδέα πίσω από τον αλγόριθμο είναι η εξής: έστω 4 διαδοχικά σημεία a,b,c,d μίας διαδρομής εάν αυτά τα σημεία διασταυρώνονται τότε η εναλλαγή από a,b,c,d σε a,c,b,d θα οδηγήσει σε πραγματοποίηση μικρότερης διαδρομής.



Αρχικά ο αλγόριθμος ελέγχει στη διαδρομή κάθε διατεταγμένη 4άδα ξεκινώντας από το σημείο εκκίνησης. Για έλεγχο περισσότερων λύσεων αλλάζει σειριακά η θέση των σημείων της διαδρομής και για κάθε νέα διαδρομή που προκύπτει ελέγχονται εκ νέου όλες οι 4άδες.

Η αλλαγή των σημείων της διαδρομής προκύπτει ως εξής:

Έστω διαδρομή με σημεία: 1, 7, 9, 13, 5 με το σημείο 1 να είναι το σημείο εκκίνησης και το 5 το σημείο τερματισμού. Η αναδιάταξη ξεκινά από το τελευταίο μη σταθερό σημείο (εδώ το 13) και επαναλαμβάνεται μέχρις ότου όλα τα σημεία να έχουν περάσει από κάθε δυνατή θέση στην διαδρομή.

π.χ. 1, 7, 13, 9, 5

1, 13, 7, 9, 5

1, 13, 9, 7, 5

1, 9, 13, 7, 5 κ.ο.κ

Σε κάθε βήμα ελέγχονται όλες οι 4άδες (στο παράδειγμά μας είναι πάντα 2) ξεκινώντας από το σημείο εκκίνησης και επιλέγεται ως τελική διαδρομή αυτή με το μικρότερο κόστος. Η βαθμολογία της διαδρομής παραμένει αμετάβλητη, σε αυτό το στάδιο, αφού δεν αλλάζουν τα σημεία που μετέχουν στη διαδρομή.

#### 5.1.1.2 ΕΙΣΑΓΩΓΗ ΣΗΜΕΙΩΝ (*Insert*)

Η εισαγωγή σημείων έχει ως σκοπό την αύξηση της συνολικής βαθμολογίας της διαδρομής.

- Για κάθε σημείο που μετέχει στη διαδρομή,
- επιλέγεται το κοντινότερο γειτονικό και
- εισάγεται στη διαδρομή εφόσον η κίνηση αυτή δεν παραβιάζει τον χρονικό περιορισμό.

Η διαδικασία επαναλαμβάνεται μέχρι να μην υπάρχουν άλλα σημεία να εισαχθούν στη διαδρομή ή να έχει εξαντληθεί ο χρόνος.

#### 5.1.1.3 ΑΝΤΙΚΑΤΑΣΤΑΣΗ ΣΗΜΕΙΩΝ (*one in- one out*)

Η αντικατάσταση σημείων ή αλλιώς η διαδικασία Replace έχει ως σκοπό τη βελτίωση της διαδρομής. Επειδή η διαδικασία αυτή επαναλαμβάνεται σε μία διαδρομή μέχρις ότου να μην μπορεί να τη βελτιστοποιήσει περαιτέρω, είναι δυνατόν το τελικό αποτέλεσμα να είναι διαδρομή με μεγαλύτερη βαθμολογία και ταυτόχρονα μικρότερο κόστος.

Αναλυτικότερα, για κάθε σημείο της διαδρομής, εκτός των σημείων εκκίνησης και τερματισμού που είναι σταθερά, επιλέγεται το κοντινότερο γειτονικό σημείο και ελέγχεται εάν είναι συμφέρουσα η αντικατάσταση. Αυτό γίνεται με δύο κριτήρια, είτε μετά την αντικατάσταση θα μειώνεται το κόστος και η βαθμολογία θα παραμένει σταθερή είτε θα αυξάνεται η βαθμολογία χωρίς να παραβιάζεται ο χρονικός περιορισμός (το κόστος μπορεί να παραμείνει σταθερό ή να αυξηθεί).

Οι παραπάνω διαδικασίες εφαρμόζονται από τον αλγόριθμο, επαναληπτικά, με την σειρά που παρουσιάστηκαν έως ότου να μην μπορεί να βελτιωθεί περαιτέρω η διαδρομή. Δηλαδή η εφαρμογή και των τριών διαδικασιών να μην μπορεί να παράγει διαδρομή με μεγαλύτερη βαθμολογία.

## 5.2 ΑΞΙΟΛΟΓΗΣΗ ΑΛΓΟΡΙΘΜΟΥ

Η αξιολόγηση του προτεινόμενου αλγορίθμου, όπως αυτός περιγράφεται στην προηγούμενη ενότητα, έγινε με βάση τα τρία πολύ γνωστά σύνολα (σετ) σημείων όπως παρουσιάζονται από τον Tsiligirides (1984) . Τα αποτελέσματα που προκύπτουν για τα σημεία των προβλημάτων αυτών συγκρίνονται με τα αποτελέσματα των αλγορίθμων της ενότητας 4.2 της παρούσας μελέτης. Οι αλγόριθμοι αυτοί επιλέχθηκαν κυρίως για τη σχετική συνάφεια που εμφανίζουν με τον αλγόριθμο αλλά και επειδή και για τους τέσσερις αλγόριθμους τα αποτελέσματά τους είναι γνωστά και δημοσιευμένα χωρίς να υπάρχουν κενά ή ελλείψεις.

## ΣΥΝΟΛΑ ΣΗΜΕΙΩΝ ΓΙΑ ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Ο Tsiligirides (1984) αξιολογεί τους αλγορίθμους που προτείνει εφαρμόζοντας τους σε τρία προβλήματα όπως παρουσιάζονται συνοπτικά στην συνέχεια. Αναλυτικότερη περιγραφή των προβλημάτων, με τις συνταγμένες των σημείων και την βαθμολογία που αποδίδει το κάθε ένα από αυτά δίνονται στα Tsiligirides (1984), Keller (1989).

### ➤ Πρόβλημα 1:

- 32 σημεία ενδιαφέροντος,
- η βαθμολογία κυμαίνεται σε 3 κατηγορίες τιμών,
- τα σημεία εκκίνησης και τερματισμού είναι τοποθετημένα στο κέντρο σχεδόν του χώρου.

### ➤ Πρόβλημα 3:

- 33 σημεία ενδιαφέροντος,
- η βαθμολογία κυμαίνεται σε 4 κατηγορίες τιμών,
- τα σημεία εκκίνησης και τερματισμού είναι τοποθετημένα στο κέντρο σχεδόν του χώρου.

### ➤ Πρόβλημα 2:

- 22 σημεία ενδιαφέροντος,
- η βαθμολογία μπορεί να πάρει 7 διαφορετικές τιμές,
- τα σημεία εκκίνησης και τερματισμού είναι τοποθετημένα στην άκρη του χώρου.

## ΠΙΝΑΚΕΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Στην παρούσα ενότητα δίνονται οι πίνακες με τα αποτελέσματα των πέντε αλγορίθμων, Tsiligirides (δύο αλγόριθμοι), Keller, Golden και του αλγορίθμου που αναπτύχθηκε στα πλαίσια της παρούσας μελέτης, για τα τρία σύνολα σημείων και παρουσιάζεται η αξιολόγηση του προτεινόμενου αλγορίθμου όπως προκύπτει από την σύγκριση των αποτελεσμάτων.

Στον Πίνακα 4 δίνονται τα αποτελέσματα των τεσσάρων αλγορίθμων, βαθμολογίες διαδρομής και κόστη, για το Πρόβλημα 1. Στον Πίνακα 5 δίνονται τα αποτελέσματα των τεσσάρων αλγορίθμων, βαθμολογίες διαδρομής και κόστη, για το Πρόβλημα 2. Στον Πίνακα 6 δίνονται τα αποτελέσματα των πέντε αλγορίθμων, βαθμολογίες διαδρομής και κόστη, για το Πρόβλημα 3.

Οι χρόνοι λειτουργίας του αλγορίθμου, στο στάδιο αυτό, είναι τόσο μικροί που μπορούν να θεωρηθούν αμελητέοι και συνεπώς παραλείπονται.

ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ 1										
TMAX	ΤΣΙΛΙΓΚΙΡΙΔΗΣ				GOLDEN KNAPSCACK		KELLER MVP		ΔΙΑΡΕΜΕ	
	SCORE	DIST	SCORE	DIST	SCORE	DIST	SCORE	DIST	SCORE	DIST
5					10	4.14	10	4.14	10	4.1426
10					15	6.87	15	6.87	15	6.8665
15					45	14.26	45	14.28	30	12.6698
20	65	19.60	65	19.60	65	19.85	65	19.85	50	19.6068
25	90	24.82	90	24.65	90	24.88	90	24.88	65	24.9560
30	110	28.80	110	28.80	110	29.88	110	28.80	80	29.2154
35	135	34.08	135	34.08	125	33.60	130	33.69	115	34.4180
40	150	38.02	150	38.02	140	39.87	155	39.91	130	39.1110
46	175	44.51	175	44.51	165	45.85	175	45.86	155	45.1242
50	190	44.53	190	49.78	180	49.92	185	49.54	160	49.5277
55	200	52.86	205	54.08	200	54.38	200	54.32	185	54.8789
60	220	59.65	220	58.93	205	59.40	225	59.89	180	58.5153
65	240	63.82	240	63.82	220	64.69	240	63.93	200	64.7446
70	260	69.13	245	69.13	245	69.91	260	69.53	215	69.3325
73	265	70.73	265	70.73	255	72.27	265	72.65	220	72.7013
75	275	74.66	275	74.66	265	74.61	270	73.80	230	74.3392
80	280	77.73	280	78.34	275	79.57	280	78.18	245	79.2492
85	285	81.33	285	81.82	285	81.78	285	81.78	255	84.3756

Πίνακας 3: Αποτελέσματα αλγορίθμων για το Πρόβλημα 1.



ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ 2										
TMAX	ΤΣΙΛΙΓΚΙΡΙΔΗΣ				GOLDEN		KELLER		ΔΙΑΠΕΜΕ	
	D(R-1)		S(R-1)		KNAPSCACK		MVP		SCORE	DIST
	SCORE	DIST	SCORE	DIST	SCORE	DIST	SCORE	DIST	SCORE	DIST
15	120	14.25	120	14.88	120	14.64	120	14.37	110	14.6023
20	200	19.88	200	19.88	200	19.88	200	19.88	180	19.4637
23	210	22.65	210	22.65	210	23.00	210	22.65	200	20.6459
25	230	24.13	230	24.26	230	24.13	230	24.13	230	24.8947
27	230	24.13	230	24.26	230	24.13	230	24.13	230	24.8947
30	265	29.85	260	29.49	260	29.22	260	29.22	240	29.8787
32	300	31.65	300	31.63	275	31.69	300	31.62	260	29.9835
35	320	34.51	320	34.51	305	34.51	320	34.51	260	33.8839
38	355	37.62	355	37.62	355	37.79	360	37.84	260	33.8839
40	385	39.56	495	39.78	380	39.89	380	39.86	280	39.4382
45	450	44.44	450	44.44	450	44.44	450	44.89	370	44.9113

Πίνακας 4: Αποτελέσματα αλγορίθμων για το Πρόβλημα 2.

ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ 3										
TMAX	ΤΣΙΛΙΓΚΙΡΙΔΗΣ		S(R-1)	GOLDEN KNPSCACK		KELLER MVP		ΔΙΑΠΕΜΕ		
	D(R-1)			SCORE	DIST	SCORE	DIST	SCORE	DIST	SCORE
15	100	13.82	100	13.82	170	14.47	170	14.47	170	14.5727
25	190	24.66	190	24.66	250	23.61	260	24.95	200	24.5326
30	240	29.60	240	29.60	320	29.19	320	29.47	230	29.4266
35	280	34.15	290	34.93	380	34.69	370	34.48	280	34.5153
40	340	39.70	330	39.65	420	38.19	430	39.69	320	39.6099
45	370	44.04	370	44.04	450	44.96	460	44.80	390	44.7489
50	420	49.58	420	49.58	500	49.33	520	48.94	420	49.6095
55	440	54.61	460	53.97	520	52.81	550	53.56	490	54.0368
60	500	59.07	500	59.07	580	58.82	570	59.26	550	58.5213
65	530	63.81	530	64.65	600	63.72	610	64.95	540	64.8184
70	560	68.75	560	69.39	640	69.19	640	69.51	600	67.5893
75	600	74.32	590	74.78	650	71.00	670	74.62	630	73.6331
80	640	79.42	640	79.80	700	79.43	700	79.66	660	79.6291
85	670	83.61	670	83.61	720	83.82	740	84.86	670	83.4124
90	700	89.14	700	89.13	770	89.31	760	88.76	720	89.9326
95	740	94.96	730	94.67	790	92.79	790	92.58	740	94.7499
100	770	99.09	760	99.10	800	97.08	800	96.99	760	96.5991
105	790	103.65	790	104.89	800	97.77	800	97.77	720	104.6504
110	800	106.19	800	108.31	800	97.08	800	96.99	800	107.0317

Πίνακας 5: Αποτελέσματα αλγορίθμων για το Πρόβλημα 3.

### 5.3 ΣΥΜΠΕΡΑΣΜΑΤΑ

Σκοπός της παρούσας μελέτης δεν ήταν να βρεθεί ένας αλγόριθμος που θα παράγει τα βέλτιστα αποτελέσματα, καθώς κάτι τέτοιο θα ξέφευγε από τα πλαίσια της. Σκοπός ήταν να βρεθεί ένας αλγόριθμος που θα λειτουργεί σωστά, θα παράγει μία καλή αν όχι τη βέλτιστη λύση καθώς επίσης και να εξεταστεί κατά πόσο αυτό θα ήταν εφικτό χρησιμοποιώντας αρκετά κλασικές μεθόδους και διαδικασίες. Ο αλγόριθμος που υλοποιήθηκε ενσωματώνει απλές και βασικές διαδικασίες που απαντώνται στη βιβλιογραφία και θα χρησιμοποιηθεί ως βάση για μελλοντική εργασία.

Από τη σύγκριση των αποτελεσμάτων (Πίνακες 3 έως 5) προκύπτει η ορθότητα της λειτουργίας του αλγορίθμου. Τα αποτελέσματα που παράγει ορισμένες φορές υπερέρχουν, άλλες φορές υπολείπονται των άλλων τεσσάρων αλγορίθμων της βιβλιογραφίας, σε κάθε περίπτωση όμως είναι συγκρίσιμα. Αυτό μας οδηγεί στο συμπέρασμα ότι αυτός ο σχετικά απλός αλγόριθμος έχει την ικανότητα να παράγει καλές αν όχι τις βέλτιστες λύσεις και συνεπώς κρίνεται σκόπιμο να βελτιωθεί περαιτέρω προκειμένου να ενσωματωθεί σε μία ολοκληρωμένη εφαρμογή πλοήγησης σε εξωτερικούς χώρους που θα χρησιμοποιείται από τουρίστες,

## 6. ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ - ΠΡΟΤΑΣΕΙΣ

Από την εφαρμογή του αλγορίθμου για τα τρία προβλήματα και τη σύγκριση των αποτελεσμάτων με γνωστούς αλγορίθμους της βιβλιογραφίας, τα αποτελέσματα που προκύπτουν για την ποιότητα του αλγορίθμου και την ικανότητα του να παράγει καλές αν όχι τις βέλτιστες διαδρομές είναι ενθαρρυντικά. Συνεπώς κρίνεται σκόπιμο να βελτιωθεί μελλοντικά ο αλγόριθμος.

Στη συνέχεια παρουσιάζονται επιγραμματικά οι στόχοι της μελλοντικής εργασίας που πρόκειται να γίνει.

- ⊙ Περαιτέρω βελτιστοποίηση του αλγορίθμου.
- ⊙ Εφαρμογή τυχαιοποιημένης τοπικής αναζήτησης.
- ⊙ Εφαρμογή του λόγου απόσταση/βαθμολογία κατά την τοπική αναζήτηση.
- ⊙ Επέκταση για λύση προβλημάτων με διαφορετικά σημεία εκκίνησης και τερματισμού.
- ⊙ Επέκταση για λύση του TOP ώστε να μπορούν να εξυπηρετηθούν ταυτόχρονα ομάδες τουριστών.
- ⊙ Επέκταση αλγορίθμου για λειτουργία στο πραγματικό οδικό δίκτυο.
- ⊙ Εισαγωγή περισσότερων περιορισμών (π.χ. χρονικά παράθυρα, κίνηση στους δρόμους).
- ⊙ Υλοποίηση εφαρμογής για λειτουργία μέσω διαδικτύου.

## 7. ΒΙΒΛΙΟΓΡΑΦΙΑ

1. L. Ardissono, A. Goy, G. Petrone, M. Signan, and P. Torasso. Intrigue: personalized recommendation of tourism attractions for desktop and handset devices. *Applied Artificial Intelligence*, 17(8{9}):687-714, 2003.
2. J. Candy, A Mobile Indoor Location-based GIS Application, in Proc of the 5<sup>th</sup> International Symposium on Mobile Mapping Technology, 2007.
3. M. Chao, B. L. Golden, E. A. Wasil, The team orienteering problem, *European Journal of Operational Research*, 2006a.
4. M. Chao, B. L. Golden, E. A. Wasil, A fast an effective heuristic for the orienteering problem, *European Journal Of Operational Research*, 2006b.
5. K. Cheverst, N. Davies, K. Mitchell, A. Friday, Experiences of developing and deploying a context-aware tourist guide: the GUIDE project. In Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00). ACM, New York, NY, USA, pp. 20-31, 2000.
6. R. Diestel, *Graph Theory*, Springer-Verlag, Heidelberg Graduate Texts in Mathematics, Volume 173, 2010.
7. Z. W. Geem, C. Tseng, Y. Park, Harmony Search for Generalized Orienteering Problem: Best Touring in China, Springer ICNS, LNCS 3612, pp. 741- 750, 2005.
8. B. Golden, A. Assad, R. Dahl, Analysis of a Large Scale Vehicle Routing Problem with an Inventory Component, *Large Scale Systems* 7, pp.181-190, 1984.
9. B. L. Golden, L. Levy, R. Vohra, The orienteering problem, *Naval Research Logistics*, pp. 307-318, 1987.

10. G. Gutin, A. Yeo, A. Zverovich, Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP, *Discrete Applied Mathematics* 117, 2002.
11. K. ten Hagen, R. Kramer, M. Hermkes, B. Schumann, P. Mueller. Semantic matching and heuristic search for a dynamic tour guide, In *Information and Communication Technologies in Tourism*, Springer 2005.
12. O' Hare, M.J. O' Grady, Gulliver's Genie: a multi-agent system for ubiquitous and intelligent content delivery, *Computer Communications*, Volume 26, Issue 11, pp. 1177-1187, 2003.
13. K. Kabassi, Personalizing recommendations for tourists, *Telematics and Informatics*, 27(1):51-66, February 2010.
14. H. A. Karimi, M. Ghafourian, Indoor Routing for Individuals with Special Needs and Preferences, *Transactions in GIS*, 2010.
15. C. P Keller, M. F Goodchild, The multiobjective vending problem: a generalization of the travelling salesman problem, *Environment and Planning B: Planning and Design*, pp. 447-480, 1988.
16. C. P. Keller, Algorithms to solve the orienteering problem: A comparison\* , *European Journal of Operational Research*, pp. 224-231, 1989.
17. A. Küpper, *Location-Based Services: Fundamentals and Operation*, Wiley, October 2005.
18. R. Kramer, M. Modsching, K. ten Hagen. A city guide agent creating and adapting individual sightseeing tours based on field trial results, *International Journal of Computational Intelligence Research*, 2(2):191-206, 2006.

19. C.-S. Lee, Y.-C. Chang, and M.-H. Wang. Ontological recommendation multi-agent for tainan city travel, *Expert Systems with Applications*, 36:6740-6753, 2009.
20. J. Lee, E. Kang, G.-L. Park, *Computational Science and Its Applications ICCSA*, chapter Design and Implementation of a Tour Planning System for Telematics Users, pp. 179-189. Springer Berlin/Heidelberg, 2007.
21. A. Maruyama, N. Shibata, Y. Murata, K. Yasumoto, M. Ito, P-tour: A personal navigation system for tourism. In *Proceedings of 11th World Congress on ITS*, pp. 18-21, 2004a.
22. M. Nagata, Y. Murata, N. Shibata, K. Yasumoto, M. Ito, *Simulated Evolution and Learning*, volume 4247 of LNCS, chapter A Method to Plan Group Tours with Joining and Forking, pp. 881-888, Springer Berlin/Heidelberg, 2006.
23. T. Shiraishi, M. Nagata, N. Shibata, Y. Murata, K. Yasumoto, M. Ito, A personal navigation system with a schedule planning facility based on multi-objective criteria, In *Proceedings of 2nd International Conference on Mobile Computing and Ubiquitous Networking*, pp. 104-109, 2005a.
24. T. Shiraishi, M. Nagata, N. Shibata, Y. Murata, K. Yasumoto, M. Ito. A personal navigation system with functions to compose tour schedules based on multipleconnecting criteria, *IPSJ Digital Courier*, 1:528-536, 2005b.
25. V.-W. Soo, S.-H. Liang, *Cooperative Information Agents V*, volume 2182 of *Lecture Notes in Computer Science*, chapter Recommending a Trip Plan by Negotiation with a Software Travel Agent, pp. 32-37, Springer Berlin/ Heidelberg, 2001.
26. W. Souffriau, P. Vasteenwegen, J. Vertommen, G. V. Berghe, D. Van Oudheusden, A PERSONALIZED TOURIST TRIP DESIGN ALGORITHM FOR MOBILE TOURIST GUIDES, *Applied Artificial Intelligence: An International Journal*, 2008.

27. W. Souffriau, P. Vasteenwegen, Tourist Trip Planning Functionalities: State-of-the-Art and Future, 10<sup>th</sup> ICWE, pp. 474-485, 2010.
28. Y. Suna, L. Lee, Agent-based personalized tourist route advice system, In ISPRS, Congress Istanbul, Proceedings of Commission II, pp. 319-324, 2004.
29. T. Tsiligirides, Heuristic Methods Applied to Orienteering, The Journal of the Operational Research Society, Vol. 35, No. 9, pp. 797-809, 1984.
30. P. Vansteenwegen, W. Souffriau, G. V. Berghe, D. Van Oudheusden, A guided local search metaheuristic for the team orienteering problem, European Journal of Operational Research 196, pp. 118-127, 2008.
31. P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The orienteering problem: A survey, European Journal of Operational Research 209, pp. 1-10, 2011.
32. K. Virrantaus, H. Tirri, J. Veijalainen, J. Markkula, A. Katanosov, A. Garmash, V. Terziyan, Developing gissupported location-based services, Web Information Systems Engineering, 2001.
33. M. Wium, Design and Evaluation of an Personalized Mobile Tourist Application, NTNU, 2010.



## 8. ΣΥΝΔΕΣΜΟΙ

1. Παγκόσμιος Οργανισμός Τουρισμού των Ηνωμένων εθνών (UNWTO)  
- <http://www2.unwto.org>
2. Cyberguide - <http://www.cc.gatech.edu/fce/cyberguide/>
3. Porter Stemming Algorithm -  
<http://tartarus.org/martin/PorterStemmer/>
4. the GUIDE project- <http://www.guide.lancs.ac.uk/overview.html>
5. <http://www.theprojectspot.com>

## 9. ΠΑΡΑΡΤΗΜΑ

### I. Υλοποίηση Αλγορίθμου Dijkstra σε γλώσσα προγραμματισμού C

```
#define MAX_NODES 1024          /* maximum number of nodes */
#define INFINITY 1000000000    /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES]; /* dist[i][j] is the distance from i to j */
void shortest_path(int s, int t, int path[])
{ struct state {
    int predecessor;          /* the path being worked on */
    int length;              /* previous node */
    enum {permanent, tentative} label; /* length from source to this node */
} state[MAX_NODES];

int i, k, min;
struct state *p;

for (p = &state[0]; p < &state[n]; p++) { /* initialize state */
    p->predecessor = -1;
    p->length = INFINITY;
    p->label = tentative;
}
state[t].length = 0; state[t].label = permanent;
k = t;
do {
    for (i = 0; i < n; i++)
        if (dist[k][i] != 0 && state[i].label == tentative) {
            if (state[k].length + dist[k][i] < state[i].length) {
                state[i].predecessor = k;
                state[i].length = state[k].length + dist[k][i];
            }
        }

    /* Find the tentatively labeled node with the smallest label. */
    k = 0; min = INFINITY;
    for (i = 0; i < n; i++)
        if (state[i].label == tentative && state[i].length < min) {
            min = state[i].length;
            k = i;
        }
    state[k].label = permanent;
} while (k != s);

/* Copy the path into the output array. */
i = 0; k = s;
do {path[i++] = k; k = state[k].predecessor; } while (k >= 0);
}
```

## II. Υλοποίηση αλγορίθμου Simulated Annealing σε Matlab

```
function [minimum,fval] = anneal(loss, parent, options)
% ANNEAL Minimizes a function with the method of simulated annealing
% (Kirkpatrick et al., 1983)

def = struct(...

    'CoolSched',@(T) (.8*T),...
    'Generator',@(x) (x+(randperm(length(x))==length(x))*randn/100),...
    'InitTemp',1,...
    'MaxConsRej',1000,...
    'MaxSuccess',20,...
    'MaxTries',300,...
    'StopTemp',1e-8,...
    'StopVal',-Inf,...
    'Verbosity',1);

% Check input
if ~nargin %user wants default options, give it and stop
    minimum = def;
    return
elseif nargin<2, %user gave only objective function, throw error
    error('MATLAB:anneal:noParent','You need to input a first guess. ');
elseif nargin<3, %user gave no options structure, use default
    options=def;
else %user gave all input, check if options structure is complete
    if ~isstruct(options)
        error('MATLAB:anneal:badOptions',...
            'Input argument "options" is not a structure')
    end
    fs = {'CoolSched','Generator','InitTemp','MaxConsRej',...
        'MaxSuccess','MaxTries','StopTemp','StopVal','Verbosity'};
    for nm=1:length(fs)
        if ~isfield(options,fs{nm}), options.(fs{nm}) = def.(fs{nm}); end
    end
end

% main settings
newsol = options.Generator;    % neighborhood space function
Tinit = options.InitTemp;     % initial temp
minT = options.StopTemp;     % stopping temp
cool = options.CoolSched;    % annealing schedule
minF = options.StopVal;
max_consec_rejections = options.MaxConsRej;
max_try = options.MaxTries;
max_success = options.MaxSuccess;
report = options.Verbosity;
k = 1;                        % boltzmann constant
```

```

% counters etc
itry = 0;
success = 0;
finished = 0;
consec = 0;
T = Tinit;
initenergy = loss(parent);
oldenergy = initenergy;
total = 0;
if report==2, fprintf(1,'\n T = %7.5f, loss = %10.5f\n',T,oldenergy); end

while ~finished;
    itry = itry+1; % just an iteration counter
    current = parent;

    % % Stop / decrement T criteria
    if itry >= max_try || success >= max_success;
        if T < minT || consec >= max_consec_rejections;
            finished = 1;
            total = total + itry;
            break;
        else
            T = cool(T); % decrease T according to cooling schedule
            if report==2, % output
                fprintf(1,' T = %7.5f, loss = %10.5f\n',T,oldenergy);
            end
            total = total + itry;
            itry = 1;
            success = 1;
        end
    end

    newparam = newsol(current);
    newenergy = loss(newparam);

    if (newenergy < minF),
        parent = newparam;
        oldenergy = newenergy;
        break
    end

    if (oldenergy-newenergy > 1e-6)
        parent = newparam;
        oldenergy = newenergy;
        success = success+1;
        consec = 0;
    else
        if (rand < exp( (oldenergy-newenergy)/(k*T) ));
            parent = newparam;

```

```
        oldenergy = newenergy;
        success = success+1;
    else
        consec = consec+1;
    end
end
end

minimum = parent;
fval = oldenergy;

if report;
    fprintf(1, '\n Initial temperature:  \t%g\n', Tinit);
    fprintf(1, ' Final temperature:    \t%g\n', T);
    fprintf(1, ' Consecutive rejections: \t%i\n', consec);
    fprintf(1, ' Number of function calls:\t%i\n', total);
    fprintf(1, ' Total final loss:      \t%g\n', fval);
end
```

### III. Περιγραφή του προτεινόμενου αλγόριθμου σε Matlab

Ισχύει ότι s=σημείο εκκίνησης, e=σημείο τερματισμού, lroutes=3.

```
% δημιουργία array of structure με τα σημεία ενδιαφέροντος και πληροφορίες
poisAll =Pois();
poisFeas=poisAll;

% poisAll θα περιέχει, μέχρι τέλους, όλα τα Pois,
% poisFeas τροποποιείται και περιλαμβάνει μόνο τα Pois που
% βρίσκονται μέσα στη μέγιστη ακτίνα της διαδρομής

% δημιουργία πίνακα 'remain'
% περιέχει τα σημεία που ΔΕΝ έχουν εισαχθεί σε διαδρομές
remain = (1:length(poisAll));

% αφαίρεση από τον πίνακα των σημείων εκκίνησης και τερματισμού
remain=remove(remain,s);
remain=remove(remain,e);

% δημιουργία πίνακα Ευκλείδειας Απόστασης
for i = 1: length(poisFeas)
    for j = 1: length(poisFeas)
        E(i,j)= sqrt((poisFeas(j).x-poisFeas(i).x)^2 + (poisFeas(j).y-poisFeas(i).y)^2);
    end
end

% anonymous function που βάση των σημείων εκκίνησης και τερματισμού
υπολογίζει το κόστος της διαδρομής για ένα σημείο x
dista = @(x) E(s,x)+E(x,e);

% αφαίρεση των σημείων που είναι έξω από την μέγιστη ακτίνα της διαδρομής
for i= length(poisFeas):-1:1
    if dista(i) > tmax
        poisFeas(i)=[];
        remain=remove(remain,i);
    end
end

% επιλογή των l σημείων που είναι πιο απομακρυσμένα από τα σημεία εκκίνησης
και τερματισμού
j = 1;
furth = struct ('id',{},'score',{},'dist',{}); με χρήση της @distance

for i = 1: length (poisFeas)
    if (poisFeas(i).id ~= s && poisFeas(i).id ~= e)
        furth(j)=struct('id',poisFeas(i).id,'score',poisFeas(i).score,
            'dist',dista(poisFeas(i).id));
        j=j+1;
    end
end
```

```

    end
end

[~, order] = sort([furth(:).dist],'ascend');
sortedStruct = furth(order);

% δημιουργία L διαδρομών (Lroutes)
% -----
% αρχικοποίηση διαδρομών
L=struct('tScore',(0), 'cost',(0));
L.b=struct ('poi',{});

% προσθήκη των l σημείων στις L διαδρομές
for i=1:lroutes
    numpoi=1;

    % εισαγωγή του σημείου εκκίνησης στις διαδρομές
    L(i).tScore= poisAll(s).score;
    L(i).cost=0;
    L(i).b(numpoi).poi=s;
    numpoi=numpoi+1;

    % προσθήκη των l σημείων στις L διαδρομές
    % εισαγωγή σημείου μόνο εάν ισχύει η τριγωνική ιδιότητα
    % αφαίρεση σημείου από τον πίνακα για να μην ξαναχρησιμοποιηθεί
    for q=1:length(sortedStruct)
        if (E(s,sortedStruct(q).id)+E(sortedStruct(q).id,e))>E(s,e)
            L(i).b(numpoi).poi=sortedStruct(q).id;
            L(i).tScore = L(i).tScore + sortedStruct(q).score;
            L(i).cost = L(i).cost + E(s,sortedStruct(q).id);
            % αφαίρεση των σημείων που χρησιμοποιήθηκαν
            remain=remove(remain,sortedStruct(q).id);
            sortedStruct(q)=[];
            break
        end
    end
end
end

% προσθήκη σημείων στις διαδρομές, όσο δεν παραβιάζεται ο χρ. περιορισμός
for i=1:lroutes
    if isempty(remain), break,end
    x=numpoi;
    while L(i).cost<=tmax
        n=nextPoi2(L(i).b(x).poi,e,remain,E);
        if ((L(i).cost + E(L(i).b(x).poi,n)+ E(n,e))<=tmax)
            if (E(L(i).b(x).poi,n)+E(n,e)>E(L(i).b(x).poi,e))
                L(i).b(x+1).poi=n;
                L(i).tScore = L(i).tScore + poisAll(n).score;
                L(i).cost = L(i).cost + E(L(i).b(x).poi,n);
                remain=remove(remain,n);
            end
        end
    end
end

```

```

        x=x+1;
    end
    else break
    end
    if isempty(remain), break, end
end
end

% εάν απομένουν σημεία που δεν έχουν χρησιμοποιηθεί δημιουργούνται και άλλες
διαδρομές με άπληστο τρόπο
j=lroutes;
while ~isempty(remain)
    x=1;
    j = j+1;
    L(j).b(x).poi=s;
    L(j).tScore= poisAll(s).score;
    L(j).cost=0;
    while L(j).cost<=tmax
        n=nextPoi(L(j).b(x).poi,remain,E);
        if ((L(j).cost + E(L(j).b(x).poi,n)+E(n,e))<=tmax)
            if (E(L(i).b(x).poi,n)+E(n,e)>E(L(i).b(x).poi,e))
                L(j).b(x+1).poi=n;
                L(j).tScore = L(j).tScore + poisAll(n).score;
                L(j).cost = L(j).cost + E(L(j).b(x).poi,n);
                remain=remove(remain,n);
                x=x+1;
            end
        else break
        end
        if isempty(remain), break, end
    end
end

% προσθήκη του σημείου τερματισμού στις διαδρομές
for i=1:j
    L(i).b(end+1).poi=e;
    L(i).tScore = L(i).tScore + poisAll(e).score;
    L(i).cost = L(i).cost + E(L(i).b(end-1).poi,e);
end

% ταξινόμηση των L διαδρομών βάση το συνολικό score της κάθε διαδρομής
[~, order] = sort([L(:).tScore],'descend');
initialRoutes = L(order);

% εάν υπάρχουν διαδρομές με το ίδιο score τότε πρέπει να ταξινομηθούν βάση του
cost, διαδρομές με μικρό cost είναι επιθυμητές
j=0;
for i=1:length(initialRoutes)
    if initialRoutes(i).tScore==initialRoutes(1).tScore; j=j+1; end
end
end

```



```

[~,order]=sort([initialRoutes(1:j).cost], 'ascend');
initialRoutes = initialRoutes(order);
newRoute= initialRoutes(1);
c= cell2mat({poisAll(:).score});
r= cell2mat({poisFeas(:).id});

% ΒΕΛΤΙΩΣΗ ΔΙΑΔΡΟΜΗΣ
% -----
bestScore=0;
bestcost=0;
rep=0;

while ((bestScore<newRoute.tScore | bestcost>newRoute.cost) & rep <6)
    bestScore=newRoute.tScore;
    bestcost=newRoute.cost;

    newRoute= swap(newRoute,E);
    newRoute= oneOpt(newRoute,r,E,tmax,c);
    newRoute= insert(newRoute,r,E,tmax,c);

    rep=rep+1;

end

display('ΒΕΛΤΙΩΜΕΝΗ ΔΙΑΔΡΟΜΗ')
newRoute
finalRoute=cell2mat({newRoute.b.poi})

```