# Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»


## Μεταπτυχιακή Διατριβή

| | |
|---|---|
| Τίτλος Διατριβής | **Προσεγγιστικοί Αλγόριθμοι του Προβλήματος του Περιοδεύοντος Πωλητή**<br>**Approximation Algorithms for the Travelling Salesman Problem** |
| Ονοματεπώνυμο Φοιτητή | **Παναγιώτης Λαμπάτος** |
| Πατρώνυμο | **Γεώργιος** |
| Αριθμός Μητρώου | **ΜΠΠΛ/08013** |
| Επιβλέπων | **Χαράλαμπος Κωνσταντόπουλος, Επίκουρος Καθηγητής** |


Ημερομηνία Παράδοσης **Φεβρουάριος 2014**

**Τριμελής Εξεταστική Επιτροπή**

| | | |
|---|---|---|
| (υπογραφή) | (υπογραφή) | (υπογραφή) |
| Γεώργιος Τσιχριτζής<br>Καθηγητής | Χαράλαμπος Κωνσταντόπουλος<br>Επίκουρος Καθηγητής | Άγγελος Πικράκης<br>Λέκτορας |

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή μου, κ. Χαράλαμπο Κωνσταντόπουλο για την καθοδήγηση και την υποστήριξη του στην εκπόνηση της παρούσας εργασίας.

Επιπλέον οφείλω ένα μεγάλο ευχαριστώ στην οικογένεια μου, για τη στήριξη που μου παρείχε όλα τα χρόνια των σπουδών μου.

## Περίληψη

Στην εργασία αυτή παρουσιάζεται το πρόβλημα του Περιοδεύοντος Πωλητή (Travelling Salesman Problem, TSP) και οι προσεγγιστικοί αλγόριθμοι επίλυσης του. Το πρόβλημα συνίσταται στην προσπάθεια εύρεσης της οικονομικότερης – άριστης περιοδείας, που πρέπει να ακολουθήσει ένας περιοδεύων πωλητής, ώστε να επισκεφθεί κάθε πόλη, ενός συνόλου πόλεων, ακριβώς μια φορά, και να επιστέψει στην αρχική πόλη. Η επίλυση του προβλήματος έχει αποδειχτεί πολύ δύσκολη και το πρόβλημα κατηγοριοποιείται στα NP-Πλήρη (Np-Complete) προβλήματα, καθώς για μεγάλο αριθμό πόλεων ο χρόνος που απαιτείται για όλους τους υπολογισμούς είναι κυριολεκτικά ασύλληπτος.

Η εργασία ξεκινά με την ιστορική παρουσίαση του προβλήματος και των εφαρμογών του. Ακολουθεί παρουσίαση της πολυπλοκότητας του προβλήματος και των διάφορων παραλλαγών του. Στην συνέχεια παρουσιάζονται οι ακριβείς (exact) αλγόριθμοι επίλυσης του TSP οι οποίοι υπολογίζουν πάντα την άριστη λύση του προβλήματος αλλά αποδίδουν ικανοποιητικά μόνο σε προβλήματα λίγων πόλεων. Κατόπιν παρουσιάζονται οι ευρετικοί (heuristics) αλγόριθμοι, οι οποίοι δίνουν αποδεδειγμένα λύσεις που προσεγγίζουν την άριστη σε πολύ μικρότερο χρόνο και ανάλογα με την τεχνική που χρησιμοποιούν κατηγοριοποιούνται σε Κατασκευαστικούς και Βελτιωτικούς. Στη συνέχεια παρουσιάζονται οι προσεγγιστικοί αλγόριθμοι σταθερού λόγου προσέγγισης και τέλος τα Προσεγγιστικά Σχήματα Πολυωνυμικού Χρόνου (Polynomial Time Approximation Scheme, PTAS) για το πρόβλημα του Ευκλείδειου-TSP.

## Abstract

This paper presents the Travelling Salesman Problem (TSP) and the approximation algorithms solving it. The problem consists in trying to find the most economical - optimal tour, to be followed by a traveling salesman in order to visit each city from a set of cities, exactly once, starting and ending at the same city. Solving the problem has proven to be very difficult and the problem is classified as NP-hard due to the fact that for a large number of cities the time required for all calculations are virtually inconceivable.

The paper begins with the historical presentation of the problem and its applications. It is followed by a presentation of the complexity of the problem and its variants. Further on, we present the exact algorithms for TSP, which find the optimal tour for the problem and for this reason they perform well only in few cities problems. Then we present the heuristics algorithms, which provide solutions that are proven close to optimal in a much shorter time. Depending on the technique they use they are categorized as construction and improvement heuristics. Next we present constant ratio approximation algorithms and finally the Polynomial Time Approximation Schemes, PTAS, for the Euclidean-TSP problem.

## Εισαγωγή

Ένα από τα μαθήματα επιλογής που είχα επιλέξει ως προπτυχιακός φοιτητής του Μαθηματικού Τμήματος ήταν η Θεωρία Γραφημάτων. Έχοντας πολύ καλή σχέση με τους υπολογιστές, τις γλώσσες προγραμματισμού και γενικότερα τα μαθήματα της εφαρμοσμένης κατεύθυνσης ήταν μια λογική επιλογή. Η πρώτη επαφή με τη θεωρία Γραφημάτων μου άφησε μια πολύ ευχάριστη εντύπωση καθώς οι διαδρομές πάνω στις Γέφυρες του Königsberg, οι «περίπατοι» του Euler και οι «κύκλοι» του Hamilton φαίνονταν ιδανική απασχόληση για έναν αναγνώστη των βιβλίων «διασκεδαστικών μαθηματικών» του Martin Gardner.

Η πρώτης εργασία στα πλαίσια του μαθήματος ήταν το πρόβλημα «Ο γύρος του Ιππότη» (Knight's Tour), σύμφωνα με το οποίο με ένα αλογάκι και το παράξενο βηματισμό του (Γ), πρέπει να επισκεφθούμε όλα τα τετράγωνα της σκακιέρας. Ουσιαστικά πρόκειται για ένα πρόβλημα εύρεσης κύκλου Hamilton σε ένα γράφημα 64 σημείων, πρόβλημα αρκετά δύσκολο όπως θα δούμε στη συνέχεια της παρούσας εργασίας και συγγενικό με το TSP.

Η υλοποίηση της εργασίας καταρχήν ήταν εύκολη, όμως είτε αργούσε, είτε αποτύχαινε να διασχίσει πάντα ολόκληρη την σκακιέρα, καθώς όπως ανέφερε και ο καθηγητής μου κ. Σπύρου στο βιβλίο του «..ο αριθμός τέτοιων περιπάτων είναι από συνδυαστικής άποψης απαγορευτικός.» και «…υπορουτίνες που θα κατασκευάσουν μια ακολουθία τέτοιων περιπάτων, ελπίζοντας ότι θα προκύψει ο γύρος του αλόγου.» [1, p. 206]. Ένα παιδικό παιχνίδι, το σκάκι με ένα μόνο πιόνι, με εισήγαγε στον κόσμο της Θεωρίας Υπολογιστικής Πολυπλοκότητας (Computational complexity theory) και της Συνδυαστική Βελτιστοποίηση (Combinatorial Optimization).

Η συγκεκριμένη κατηγορία προβλημάτων με ενδιέφερε πάντα, είτε σαν απλό αναγνώστη ψυχαγωγικών μαθηματικών, είτε σαν φοιτητή, οπότε ήταν η πρώτη μου επιλογή κατά την επιλογή θέματος για την Μεταπτυχιακή μου Διατριβή. Ολοκληρώνοντας την εισαγωγή παραθέτω το σχόλιο του καθηγητή Χρήστου Παπαδημητρίου για το Πρόβλημα του Περιοδεύοντος Πωλητή:

*Christos Papadimitriou told me that the traveling salesman problem is not a problem, it's an addiction. Jon Bentley, 1991* [2]

## Σύντομη Περιγραφή της Εργασίας

Το Πρόβλημα του Περιοδεύοντος Πωλητή (Travelling Salesman Problem - TSP) είναι ένα από τα δυσκολότερα προβλήματα Συνδυαστικής Βελτιστοποίησης, λόγω της μεγάλης Υπολογιστικής Πολυπλοκότητας του. Αποτελεί επέκταση του προβλήματος του κύκλου Hamilton, επίσης δύσκολου προβλήματος, καθώς εισέρχεται η παράμετρος του κόστος για κάθε μετακίνηση και η απαίτηση να βρεθεί η διαδρομή ελαχίστου κόστους. Επομένως δεν μας αρκεί απλώς η εύρεση μια τέτοιας διαδρομής, αλλά του συνόλου τους ώστε να γίνουν οι απαραίτητες συγκρίσεις και να επιλεγεί η μικρότερη – άριστη.

Στην διάρκεια των τελευταίων δύο αιώνων και ειδικά των τελευταίων πενήντα χρόνων έχουν γίνει προσπάθειες για την λύση του με διάφορα μαθηματικά εργαλεία και τρόπους. Προσπαθώντας να ξεπεράσουμε το πρόβλημα, μια από τις στρατηγικές που επιλέξαμε είναι και να μην απαιτήσουμε την άριστη λύση αλλά έστω μια σχετικά καλή λύση που προσεγγίζει τη άριστη και μπορούμε να την βρούμε σε πολύ καλύτερο χρόνο. Οι προσεγγιστικές αυτές μέθοδοι καθώς και τα μαθηματικά γύρω από αυτές, που μας υποδεικνύουν το αν και το κατά πόσο γρήγορα συγκλίνουν σε μια καλή – αποδεκτή λύση αποτελούν το αντικείμενο της παρούσας εργασίας.

## Δομή της Εργασίας

Στο $1^{ο}$ Κεφάλαιο της Εργασίας παρουσιάζεται η ιστορία του προβλήματος καθώς και οι εφαρμογές του.

Στο $2^{ο}$ Κεφάλαιο δίνεται η μαθηματική τυποποίηση του προβλήματος και παρουσιάζονται οι διάφορες παραλλαγές του.

Στο $3^{ο}$ Κεφάλαιο δίνονται ορισμοί, αξιώματα και θεωρήματα της Θεωρίας Γραφημάτων που απαιτούνται στα επόμενα κεφαλαία.

Στο $4^{ο}$ Κεφάλαιο γίνεται μια εισαγωγή στην Θεωρία Πολυπλοκότητας, στις κλάσεις των πολύπλοκων προβλημάτων και η απόδειξη της πολυπλοκότητας του προβλήματος.

Στο $5^{ο}$ Κεφάλαιο παρουσιάζονται οι Ακριβείς Αλγόριθμοι εύρεσης λύσης του προβλήματος, εξετάζεται η πολυπλοκότητα και η απόδοση τους και δίνονται παραδείγματα υλοποίηση τους.

Στο $6^{ο}$ Κεφάλαιο γίνεται μια εισαγωγή στους Προσεγγιστικού Αλγορίθμους και στην κατηγοριοποίηση των διάφορων προβλημάτων ανάλογα με την δυνατότητα προσέγγισης τους.

Στο $7^{ο}$ Κεφάλαιο παρουσιάζονται οι Κατασκευαστικοί Ευρετικοί Αλγόριθμοι, οι οποίοι κατασκευάζουν σταδιακά μια λύση του προβλήματος που προσεγγίζει την άριστη, εξετάζεται η πολυπλοκότητα και η απόδοση τους και δίνονται παραδείγματα υλοποίηση τους.

Στο $8^{ο}$ Κεφάλαιο παρουσιάζονται οι Βελτιωτικοί Ευρετικοί Αλγόριθμοι οι οποίοι βελτιώνουν μια περιοδεία που έχουμε ήδη βρει με κάποιον κατασκευαστικό αλγόριθμο. Κατόπιν εξετάζεται η πολυπλοκότητα και η απόδοση τους και δίνονται παραδείγματα υλοποίηση τους.

Στο $9^{ο}$ Κεφάλαιο παρουσιάζονται οι Προσεγγιστικοί Αλγόριθμοι Σταθερού Λόγω Προσέγγισης, εξετάζεται η πολυπλοκότητα και η απόδοση τους και δίνονται παραδείγματα υλοποίηση τους.

Στο $10^{ο}$ Κεφάλαιο παρουσιάζονται τα Προσεγγιστικά Σχήματα Πολυωνυμικού Χρόνου (PTAS) που είναι οι αλγόριθμοι με την καλύτερα εγγυημένη απόδοση.

Η εργασία ολοκληρώνεται με το $11^{ο}$ Κεφάλαιο όπου συνοψίζονται σε πίνακα η πολυπλοκότητα και η απόδοση των αλγορίθμων που παρουσιάστηκαν και δίνονατι τα συμπεράσματα της εργασίας.

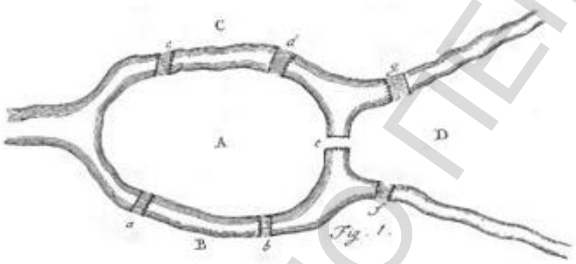
# Περιεχόμενα

## Πίνακας Εικόνων

# 1. Ιστορική Παρουσίαση του προβλήματος και εφαρμογές

## 1.1. Ιστορική Παρουσίαση

Η θεωρία γραφημάτων σαν αυτοτελές μαθηματικό πεδίο θεωρείται ότι ξεκίνησε από τον Leonard Euler και την εργασία του για τις γέφυρες του ποταμού Pregel στο Koenigsberg το 1736. Στην εργασία αυτή ο Euler μελετούσε την δυνατότητα να διασχίσει και τις επτά γέφυρες του ποταμού, χωρίς να χρειαστεί να περάσει δεύτερη φορά από κάποια. Το πρόβλημα αυτό ονομάστηκε Κύκλος του Euler, προς τιμήν του, και αποτελεί ένα από τα βασικά προβλήματα της θεωρίας γραφημάτων. Ένας κύκλος του Euler είναι μια περιοδεία σε όλες τις κορυφές ενός γραφήματος όπου κάθε ακμή χρησιμοποιείται μόνο μια φορά.




**Εικόνα 1 Σχέδιο του Euler με τις γέφυρες του ποταμού Pregel στο Koenigsberg**

Ο Κύκλος του Hamilton αποτελεί το δεύτερο διάσημο πρόβλημα της θεωρίας γραφημάτων. Ένας κύκλος του Hamilton είναι μια περιοδεία σε όλες τις κορυφές ενός γραφήματος όπου από κάθε κορυφή μπορούμε να περάσουμε μόνο μια φορά. Παρουσιάστηκε πρώτη φορά από τον Γάλλο χημικό Alexandre Théophile Vandermonde στη μελέτη του Remarques sur des problèmes de situation (1771) για τις κινήσεις του Ίππου στην σκακιέρα. Το πρόβλημα πήρε το όνομα του μαθηματικού William Rowan Hamilton, ο οποίος δημιούργησε το 1857 το παιχνίδι Icosian [3, pp. 150-156], στο οποίο πρέπει να βρεθεί ένας κύκλος Hamilton από τις ακμές ενός δωδεκάεδρου. Το πρόβλημα του Κύκλου του Hamilton είναι ιδιαίτερα δύσκολο καθώς δεν γνωρίζουμε ικανή και αναγκαία συνθήκη για την ύπαρξη του σε ένα γράφημα.


**Εικόνα 2 Το παιχνίδι Icosian**

Το πρόβλημα του Περιοδεύοντος Πωλητή (Travelling Salesman Problem, TSP) συνίσταται στην προσπάθεια εύρεσης της οικονομικότερης – άριστης περιοδείας, που πρέπει να ακολουθήσει ένας περιοδεύων πωλητής, ώστε να επισκεφθεί κάθε πόλη ενός συνόλου πόλεων ακριβώς μια φορά, ξεκινώντας και επιστρέφοντας στην αρχική πόλη. Είναι προφανής η σύνδεση του TSP με το πρόβλημα του Κύκλου του Hamilton καθώς, η λύση του TSP προβλήματος είναι ο μικρότερος – άριστος κύκλος Hamilton του αντίστοιχου γραφήματος.

Ιστορικά η πρώτη αναφορά στο TSP γίνεται σε ένα βιβλίο για περιοδεύοντες πωλητές το 1832 [4] όπου αναφέρετε το πρόβλημα με εκπληκτική ακρίβεια και περιλαμβάνονται πέντε παραδείγματα περιοδειών σε περιοχές της Γερμανίας και της Ελβετίας χωρίς όμως να υπάρχει κάποια μαθηματική εφαρμογή.

Η πρώτη μαθηματική αναφορά στο πρόβλημα εντοπίζεται το 1920, όταν ο μαθηματικός Karl Menger στη Βιέννη, παρουσιάζει το πρόβλημα του ταχυδρόμου (messenger problem), πολύ κοντινό με το TSP. Το πρόβλημα αναφέρεται στην εύρεση της πιο σύντομης διαδρομής ανάμεσα σε δεδομένα σημεία με γνωστές τις μεταξύ τους αποστάσεις, χωρίς όμως την επιστροφή στο αρχικό σημείο. Η μετατροπή του προβλήματος του ταχυδρόμου σε TSP μπορεί να γίνει με την προσθήκη ενός βοηθητικού σημείου μηδενικής απόστασης από όλα τα άλλα σημεία ώστε να μην επηρεάζει τη λύση του προβλήματος.

Το πρόβλημα, απασχόλησε τη μαθηματική κοινότητα του Princeton με πρωτοπόρο τον Hassler Whitney κατά τη διάρκεια της δεκαετίας του 1930. Στη συνέχεια, το 1948, ο μαθηματικός Merrill Meeks Flood, εισήγαγε την ονομασία TSP, παρουσιάζοντάς το πρόβλημα στη RAND Corporation. Η RAND Corporation είναι ένας μη κερδοσκοπικός οργανισμός και θεωρείται το επίκεντρο της πνευματικής έρευνας και ανάπτυξης των Ηνωμένων Πολιτειών. Η RAND το 1949 συμπεριέλαβε το TSP στα προβλήματα που προσφερόταν χρηματικό βραβείο για την επίλυση τους. Το πρόβλημα είχε γίνει πλέον ευρέως γνωστό και αρχίσαν να εμφανίζονται σχετικές δημοσιεύσεις.

Η δυσκολία του προβλήματος οδήγησε τους μαθηματικούς στην έρευνα της ποιότητας των λύσεων καθώς και των ορίων τους. Το 1948 ο Eli Marks απέδειξε ότι για την παραλλαγή του προβλήματος όπου οι $n$ πόλεις επιλέγονται εντός μοναδιαίου τετραγώνου (δηλ. οι συντεταγμένες τους (x,y)∈ [0,1]x[0,1]) η άριστη διαδρομή έχει μήκος τουλάχιστον $\frac{1}{\sqrt{2}}(\sqrt{n}-\frac{1}{\sqrt{n}})$ . Ένα χρόνο αργότερα το 1949 ο M. N. Ghosh απέδειξε ότι η άριστη διαδρομή για το ίδιο πρόβλημα έχει μήκος το πολύ $1.27\sqrt{n}$ .

Οι προσπάθειες για την επίλυση του TSP δεν σημείωσαν κάποια σημαντική πρόοδο μέχρι 1953, όταν οι Dantzig, Fulkerson, και Johnson [5, pp. 393-410] παρουσίασαν έναν αλγόριθμο επίλυσης του TSP (Cutting planes) αντιμετωπίζοντας το ως πρόβλημα ακέραιου γραμμικού προγραμματισμού. Με τον αλγόριθμο αυτό λύθηκε το πρόβλημα 49 πόλεων, το οποίο και παρέμεινε το μεγαλύτερο λυμένο TSP για 17 χρόνια.

Το 1958 ο Croes [6] παρουσίασε την ευρετική μέθοδο βελτιστοποίησης 2-opt για το Ευκλείδειο-TSP.

Οι Held και Karp [7]παράλληλα με τον Bellman [8], το 1962 διατύπωσαν έναν ακριβή αλγόριθμο εκθετικής τάξεως, ενώ το 1971, έλυσαν ένα πρόβλημα 64 σημείων καταρρίπτοντας το ρεκόρ των Dantzig, Fulkerson, και Johnson.

Το 1972 ο Karp [9] έδειξε ότι ο κύκλος του Hamilton είναι ένα NP-Πλήρες πρόβλημα, από το οποίο προκύπτει ότι το TSP είναι και αυτό NP- Πλήρες πρόβλημα. Έτσι αποδείχθηκε και θεωρητικά η μεγάλη υπολογιστική πολυπλοκότητα και δυσκολία του προβλήματος.

Το 1973 οι Lin και Kernighan [10] παρουσίασαν την μέθοδο τους η οποία αποτελεί γενίκευση της 2-opt για το Ευκλείδειο-TSP και μια από τις αποδοτικότερες που έχουν παρουσιασθεί.

Το 1975, ο Παναγιώτης Μηλιώτης [11, p. 177–188] βασιζόμενος στον αλγόριθμο των Dantzig, Fulkerson, και Johnson έδωσε λύση σε ένα πρόβλημα που περιείχε 80 τυχαία σημεία.

Ο Νίκος Χριστοφίδης [12] παρουσίασε το 1976 τον προσεγγιστικό αλγόριθμο που φέρει το όνομα του, για το Μετρικό-TSP με λόγο προσέγγισης χειρότερης περίπτωσης 3/2 ως προς την άριστη περιοδεία (worst case ratio 3/2), ο οποίος παραμένει ο καλύτερος που γνωρίζουμε μέχρι σήμερα για το συγκεκριμένο πρόβλημα.

Τη ίδια χρονιά οι Sahni και Gonzalez [13], απόδειξαν ότι δεν υπάρχει $\rho$ −προσεγγιστικός αλγόριθμος για το γενικό TSP, επιβεβαιώνοντας έτσι ότι το TSP εκτός από δύσκολο, είναι και δύσκολα προσεγγίσιμο.

Οι Grotschel, Padberg, Rinaldi [14], την δεκαετία του 1980, βρήκαν άριστες λύσεις σε προβλήματα μέχρι και 2392 πόλεις με την μέθοδο των Cutting planes.

Από την δεκαετία του 1990, οι Applegate, Bixby, Chvatal και Cook αναπτύσσουν το λογισμικό Concorde που θεωρείτε το καλύτερο για την επίλυση προβλημάτων TSP και έχει αποδώσει άριστες περιοδείες σε μεγάλα TSP.

Το 1991 ο Reinelt ξεκίνησε τη δημοσίευση του TSPLIB, μία βιβλιοθήκη που περιλαμβάνει λυμένα TSP διαφόρων τύπων καθώς και τις άριστες περιοδείες τους.

Το 1991 επίσης οι Olivier Martin , Steve W. Otto και Edward W. Felten [15] βελτίωσαν την μέθοδο Lin - Kernighan παρουσιάζοντας την Chained Lin-Kernighan μέθοδο η οποία περιλήφθηκε στο λογισμικό Concorde.

Το 1996 ο Arora [16] και παράλληλα ο Mitchell [17], ανακάλυψαν το πρώτο γνωστό προσεγγιστικό σχήμα πολυωνυμικού χρόνου (Polynomial Time Approximation Scheme, PTAS) για το πρόβλημα του Ευκλείδειου-TSP. Και οι δύο μοιράστηκαν το βραβείο Gödel 2010 για αυτήν την ανακάλυψη.

Το 2000 ο Helsgaun [18]παρουσίασε μια βελτιωμένη υλοποίηση της μεθόδου Lin - Kernighan που θεωρείται η πιο αποδοτική μέχρι και σήμερα.

Το 2005 [19] με το λογισμικό Concorde TSP Solver βρέθηκε η άριστη περιοδεία ενός TSP 33.810 πόλεων, που αντιστοιχούσαν σε σημεία ηλεκτρονικής πλακέτας.

Το 2006 [19] επιλύθηκε πρόβλημα 85.900 πόλεων που προέκυψε από την ανάγκη βελτιστοποίησης της διαδρομής ενός laser κατά την δημιουργία ενός κυκλώματος, χρησιμοποιώντας το λογισμικό Concorde TSP Solver.

## 1.2. Εφαρμογές

Το πρόβλημα του Περιοδεύοντος Πωλητή και η ανάγκη επίλυσης του, εμφανίζονται σε διάφορες εφαρμογές στην πραγματική ζωή. Από την παράδοση της αλληλογραφίας ενός ταχυδρόμου, μέχρι το σχέδιο ενός αρχιτέκτονα που προσπαθεί να σχεδιάσει την πιο αποτελεσματική τοπολογία δικτύου ενός κτιρίου, που θα συνδέει εκατοντάδες ηλεκτρονικούς υπολογιστές.

Η πιο κοινή εφαρμογή του προβλήματος είναι σε προβλήματα δρομολόγησης μεταφοράς. Η διαδρομή που θα κάνει ένας πωλητής, ένας ταχυδρόμος, ένας μεταφορέας ή ένα λεωφορείο καθημερινά μπορεί να βελτιστοποιηθεί λύνοντας το αντίστοιχο TSP πρόβλημα.

Το TSP συναντάται στον προγραμματισμό μηχανήματων. Υπάρχουν μηχανές που για παράδειγμα ανοίγουν με τρυπάνι διάφορες τρύπες σε ένα συγκεκριμένο κομμάτι του υλικού. Το υλικό μπορεί να είναι μία πλακέτα κυκλώματος, το πλαίσιο ενός οχήματος, ή ακόμα και ένα κομμάτι ξύλου που θα χρησιμοποιηθεί ως ράφι. Το μηχάνημα κινείται κατά μήκος των διαδρομών, έτσι ώστε το τρυπάνι να τρυπήσει μια συγκεκριμένη περιοχή. Θα χρειαστεί κάποιο χρονικό διάστημα για να επανατοποθετηθεί το τρυπάνι, ανάλογα με την απόσταση που πρέπει να προχωρήσει. Μια λύση για το αντίστοιχο TSP όπου οι κορυφές αντιπροσωπεύουν τις θέσεις που πρέπει να γίνουν οι τρύπες και οι ακμές είναι οι αποστάσεις μεταξύ τους, μπορεί να χρησιμοποιηθεί για να προγραμματιστεί η άριστη σειρά με την οποία θα ανοιχτούν οι τρύπες. Εξοικονομείται έτσι παραγωγικός χρόνος αλλά και η μηχανή καταπονείται λιγότερο.

Παράδειγμα τέτοιων μηχανών είναι τα μεγάλα τηλεσκόπια τα οποία για τις ανάγκες των ερευνητών πρέπει να παρακολουθήσουν διάφορα φαινόμενα σε διαφορετικά σημεία του ουρανού. Καθώς η μετακίνηση τους είναι αργή και κοστοβόρα προγραμματίζονται βάσει μιας λύσης του αντίστοιχου TSP με κορυφές του προβλήματος η περιοχή στόχος του ουρανού και ακμές με κόστος την αντίστοιχη μετακίνηση που απαιτείται.

Άλλη εφαρμογή του TSP η ηλεκτρονική ή μηχανική τοποθέτηση συνδέσεων. Η καλωδίωση ενός κυκλώματος, η ηλεκτρική καλωδίωση μέσα σε ένα μεγάλο κτίριο ή ακόμη η διάταξη των υδραυλικών εγκαταστάσεων μέσα σε ένα κτίριο μπορούν να αντιμετωπιστούν ως πρόβλημα TSP. Σε αυτές τις περιπτώσεις, οι συνδέσεις πρέπει να είναι διατεταγμένες έτσι ώστε όλα τα συστατικά να είναι συνδεδεμένα σε ένα κύκλο. Στην περίπτωση ενός κυκλώματος, οι συνδέσεις είναι τα καλώδια και τα εξαρτήματα όπως οι κρυσταλλολυχνίες, οι αντιστάσεις, κλπ. Όταν μιλάμε για την ηλεκτρική εγκατάσταση του κτιρίου, οι συνδέσεις είναι τα καλώδια και τα στοιχεία είναι οι διακόπτες, πρίζες, φωτιστικά, κλπ. Τέλος, για τη διαμόρφωση στα υδραυλικά του κτιρίου, οι συνδέσεις είναι οι σωλήνες και τα εξαρτήματα είναι οι βρύσες. Σε όλες αυτές τις περιπτώσεις , ο σχεδιαστής προσπαθεί να βρει την άριστη διαδρομή, προκειμένου να εξοικονομήσει υλικό και να βελτιστοποιήσει τη ροή ή την κατανάλωση μειώνοντας το μήκος του κύκλου. Στη σύνδεση κυκλωμάτων ή καλωδίωση ηλεκτρονικών εξαρτημάτων, το ρεύμα πρέπει να ταξιδέψει την όσο το δυνατόν μικρότερη απόσταση για να αυξήσει τη συνολική απόδοση.

Ένα από τα ζωτικότερα κομμάτια της σύγχρονης οικονομίας είναι Διοίκηση Συστημάτων Εφοδιασμού (logistics). Το TSP εμφανίζεται σε πάρα πολλά σημεία της αλυσίδας εφοδιασμού πέραν της τελικής δρομολόγησης προς τον πελάτη που είναι και το πρωταρχικό TSP.Η εκτέλεση μιας παραγγελίας απαιτεί τη συλλογή όλων των αντικειμένων της παραγγελίας από την αποθήκη που βρίσκονται. Η άριστη διαδρομή για την συλλογή όλης της παραγγελίας, είναι προφανές ότι είναι η λύση ενός προβλήματος TSP με κορυφές τις θέσεις των αντικειμένων και ακμές με κόστος το κόστος (συνήθως χρονικό) μετακίνησης του συλλέκτη ανθρώπου ή μηχανήματος. Το πρόβλημα επεκτείνεται δε, όταν έχουμε παραγγελίες με προϊόντα από πολλές αποθήκες σε διαφορετικά σημεία όπως στην πραγματικότητα συμβαίνει στις μεγάλες σύγχρονες επιχειρήσεις. Το TSP εμφανίζεται επίσης ως δευτερεύον πρόβλημα σε εφαρμογές πρόβλεψης της ζήτησης των προϊόντων και της άριστης τοποθέτησης των αποθηκών.

Στις σύγχρονες επιχειρήσεις προφανώς δεν υπάρχει μόνο ένας πωλητής ή συλλέκτης αντικειμένων από τις αποθήκες. Η διαχείριση πολλών πωλητών περιγράφεται από το πολλαπλό TSP (m-TSP), όπου έχουμε πολλούς πωλητές που πρέπει να επισκεφθούν ένα σύνολο πόλεων και κάθε πόλη πρέπει να την επισκεφτεί ακριβώς μία φορά οποιοσδήποτε από τους πωλητές. Κάθε ένας από τους πωλητές πρέπει να ολοκληρώσει μια περιοδεία και ο συνδυασμός όλων των περιοδειών περιλαμβάνει την επίσκεψη σε μια πόλη ακριβώς μία φορά. Οι πωλητές αρχίζουν και τελειώνουν την περιοδεία τους στο ίδιο σημείο (τοποθεσία βάσης). Το αντικείμενο του προβλήματος είναι να καθορίσουμε πόσους πωλητές θα πρέπει προσλάβουμε και ποιες είναι οι περιοδείες που θα ακολουθήσουν ώστε να ελαχιστοποιηθεί η συνολική απόσταση και το κόστος μίσθωσης των πωλητών. Το m-TSP είναι άμεσα συνδεδεμένο με το Πρόβλημα της Δρομολόγησης Οχημάτων (Vehicle Route Problem, VRP).

# 2. Μαθηματική τυποποίηση και Παραλλαγές του Προβλήματος

Όπως είδαμε στις εφαρμογές του προβλήματος το TSP εμφανίζεται σε διάφορες καταστάσεις και προβλήματα και με διαφορετικές παραλλαγές. Στο κεφάλαιο αυτό θα παρουσιάσουμε τις σημαντικότερες από αυτές.

## 2.1. Μαθηματική τυποποίηση του Προβλήματος

Ένας πλανόδιος πωλητής θέλει να επισκεφθεί όλους τους πελάτες του μια φορά και στο τέλος να επιστρέψει στο σημείο εκκίνησης της περιοδείας του. Το πρόβλημα έγκειται στη σχεδίαση της διαδρομής ούτως ώστε να ελαχιστοποιηθεί το μήκος της περιοδείας.

Η μαθηματική τυποποίηση του προβλήματος ως πρόβλημα ακέραιου γραμμικού προγραμματισμού σύμφωνα με τους Dantzig, Fulkerson και Johnson [5],είναι η ακόλουθη:

Δοθέντος πλήρους κατευθυνόμενου γραφήματος $G(V,A)$ όπου $V = \{1,2..n\}$ το σύνολο κορυφών και $A = \{(i,j)|\ i,j = 1 \dots n\}$ το σύνολο ακμών και $nXn$ πίνακα αποστάσεων C = ($c_{ij} \geq 0$), πρέπει ελαχιστοποιηθεί η συνάρτηση:

$$\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \qquad (1)$$

με τους ακόλουθους περιορισμούς

$$\sum_{i \in V} x_{ij} = 1, i \in V$$
$$\sum_{j \in V} x_{ij} = 1, j \in V \qquad (2)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1\,, \forall\, S \subset V, S \neq \emptyset \qquad (3)$$

$$x_{ij} = \begin{cases} 1 \text{ αν συμμετέχει η ακμή } (i,j) \text{ στη λύση} \\ 0 \end{cases} \quad i,j \in V \qquad (4)$$

Ο περιορισμός (2) επιβάλει ο πωλητής να αφιχθεί σε μια πόλη- κορυφή του γραφήματος μία και μόνο μία φορά και να αναχωρήσει από αυτήν μία και μόνο μία φορά.

Ο περιορισμός (3) δεν επιτρέπει τη δημιουργία περιοδειών (subtour elimination) με λιγότερες από $n$ πόλεις.

Επειδή το G είναι πλήρες δεν υπάρχει κάποιος περιορισμός για τις ακμές. Αν θέλουμε να αγνοήσουμε κάποια μπορούμε να της θέσουμε άπειρο κόστος ($c_{ij}$=∞). Σε κάθε περίπτωση θέτουμε $c_{ii}$=∞.

## 2.2. Ασύμμετρο TSP

Το ασύμμετρο TSP ή a-TSP είναι η γενικότερη παραλλαγή του προβλήματος και αντιστοιχεί σε ένα κατευθυνόμενο βεβαρημένο γράφημα. Κάθε ακμή ανάλογα με τον προσανατολισμό της έχει και διαφορετικό βάρος. Τα περισσότερα πραγματικά προβλήματα είναι συνήθως ασύμμετρα. Η μαθηματική τυποποίηση της προηγούμενης παραγράφου αναφέρεται στο Ασύμμετρο TSP

## 2.3. Συμμετρικό TSP

Το συμμετρικό (symmetric) TSP αντιστοιχεί σε ένα μη κατευθυνόμενο βεβαρυμμένο γράφημα. Κάθε ακμή μεταξύ δύο πόλεων – σημείων του προβλήματος έχει ακριβώς το ίδιο κόστος ανεξαρτήτως της κατεύθυνσης που θα κινηθεί ο πωλητής. Παρατηρούμε ότι ο πίνακας κόστους του αντίστοιχου γραφήματος είναι συμμετρικός.

Η μαθηματική τυποποίηση του προβλήματος ως πρόβλημα ακέραιου γραμμικού προγραμματισμού σύμφωνα με τους Dantzig, Fulkerson και Johnson [5],είναι η ακόλουθη:

Δοθέντος πλήρους μη κατευθυνόμενου γραφήματος $G(V,A)$ όπου $V = \{1,2..n\}$ το σύνολο κορυφών και $A = \{(i,j)|\ i,j = 1 \dots n\}$ το σύνολο ακμών και $nXn$ πίνακα αποστάσεων C = ($c_{ij} \geq 0$), πρέπει ελαχιστοποιηθεί η συνάρτηση:

$$\sum_{i \in V} \sum_{j > i} c_{ij} x_{ij} \tag{1}$$

με τους ακόλουθους περιορισμούς

$$\sum_{j<i} x_{ij} + \sum_{j>i} x_{ij} = 2,\ i \in V \tag{2}$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j>i}} x_{ij} \leq |S| - 1\,, \forall\, S \subset V, S \neq \emptyset \tag{3}$$

$$x_{ij} = \begin{cases} 1 \ \text{αν συμμετέχει η ακμή } (i,j) \text{ στη λύση} \\ 0 \end{cases} \quad i,j \in V, j > i \tag{4}$$

Ο περιορισμός (2) επιβάλει ο πωλητής να αφιχθεί σε μια πόλη- κορυφή του γραφήματος μία και μόνο μία φορά και να αναχωρήσει από αυτήν μία και μόνο μία φορά.

Ο περιορισμός (3) δεν επιτρέπει τη δημιουργία περιοδειών (subtour elimination) με λιγότερες από $n$ πόλεις.

Επειδή το G είναι πλήρες δεν υπάρχει κάποιος περιορισμός για τις ακμές. Αν θέλουμε να αγνοήσουμε κάποια μπορούμε να της θέσουμε άπειρο κόστος ($c_{ij}$=∞). Σε κάθε περίπτωση θέτουμε $c_{ii}$=∞.

## 2.4. Μετρικό - TSP

Στο Μετρικό-TSP (Metric-TSP ή *delta-TSP* ή Δ-TSP), τα βάρη των ακμών είναι συμμετρικά και παράλληλα ικανοποιούν την τριγωνική ανισότητα (triangle inequality).

Η τριγωνική ανισότητα είναι χαρακτηριστική ιδιότητα μιας Μετρικής συνάρτησης. Υπενθυμίζουμε τον ορισμό της Μετρικής:

Μετρική ονομάζεται μια συνάρτηση d:V→R, όπου V≠∅ σύνολο, η οποία ικανοποιεί τις παρακάτω ιδιότητες για κάθε x,y,z∈V:

- d(x,y)=0 αν και μόνο αν x=y
- d(x,y)= d(y,x)
- d(x,y)≤ d(x,z)+ d(z,y) (τριγωνική ανισότητα)

Η πρώτη ιδιότητα αποκλείει TSP που μπορεί το γράφημα τους να έχει Βρόγχους.

Η δεύτερη ιδιότητα απαιτεί το γράφημα του προβλήματος να είναι συμμετρικό.

Γνωρίζουμε πολλές συναρτήσεις που ικανοποιούν τις ιδιότητες ώστε να είναι μετρικές. Η πιο συνηθισμένη μετρική είναι η Ευκλείδια, την οποία θα εξετάσουμε σε επόμενη παράγραφο. Άλλες γνωστές μετρικές είναι η Manhattan, η maximum και άλλες. Η μετρική Manhattan εμφανίζεται στην εφαρμογή του TSP που περιγράψαμε στην παράγραφο 1.2 για τον προγραμματισμό μηχανήματων που ανοίγουν τρύπες σε πλακέτες κυκλώματος.

## 2.5. Ευκλείδειο - TSP

Το Ευκλείδειο-TSP (Euclidean TSP) είναι ένα Μετρικό- TSP στο οποίο Μετρική είναι η Ευκλείδεια Μετρική. Το πρόβλημα έχει αποδειχθεί ότι είναι NP-Πλήρες [20, p. 237–244 ]. Η Ευκλείδεια μετρική ορίζεται για d-διάστατους χώρους αν x,y∈V με x = (x1, x2, . . . , xd) και y = (y1, y2, . . . , yd) τότε η απόσταση τους είναι d(x,y)=$\sum_{i=1}^{d}\sqrt{(x_i - y_i)}$

Αν d=2 τότε το πρόβλημα βρίσκεται στο συνήθη δισδιάστατο χώρο και καλείται επίπεδο-TSP (planar).

## 2.6. Μέγιστο - TSP

Το Μέγιστο TSP (Max – TSP) είναι μια παραλλαγή στην οποία αναζητούμε την περιοδεία με το μεγαλύτερο κόστος. Το πρόβλημα ανάγετε εύκολα σε κανονικό TSP με την χρήση αρνητικών βαρών, ενώ αν η μέθοδος επίλυσης που επιλέγουμε απαιτεί θετικά βάρη στο γράφημα μας, μπορούμε να προσθέσουμε μια μεγάλη σταθερά χωρίς βλάβη της άριστης περιοδείας.

## 2.7. bottleneck TSP

Στο TSP λαιμού μπουκαλιού (bottleneck TSP) σκοπός μας είναι να βρούμε μια περιοδεία τις οποίας το μεγαλύτερο βάρος μιας ακμή να είναι όσο το δυνατόν μικρότερο. Το πρόβλημα έχει τις ρίζες του στην κίνηση των αυτοκινήτων, καθώς επιθυμούμε να ελαχιστοποιήσουμε το χρόνο οδήγησης από μια πόλη σε μια άλλη και όχι κατ’ ανάγκην ολόκληρης της διαδρομής.

# 3. Στοιχεία Θεωρίας Γραφημάτων

Στο Κεφάλαιο αυτό παραθέτουμε βασικούς ορισμούς, αξιώματα και θεωρήματα της Θεωρίας Γραφημάτων που απαιτούνται για την παρακολούθηση των επόμενων κεφαλαίων.

## 3.1. Βασικοί Ορισμοί Γραφημάτων

- **Μη Κατευθυνόμενο Γράφημα**

Γράφημα είναι ένα διατεταγμένο ζεύγος *G* = (*V*, *E*) αποτελούμενο από το σύνολο *V* των κορυφών και ένα σύνολο *E* από ακμές οι οποίες είναι υποσύνολα δύο στοιχείων του *V*. Μια ακμή σχετίζεται με δύο κορυφές και η σχέση απεικονίζεται ως <u>μη διατεταγμένο</u> ζεύγος των κορυφών σε σχέση με τη συγκεκριμένη ακμή. Κάθε γράφημα που ορίζεται σύμφωνα με τον παραπάνω ορισμό καλείται Μη Κατευθυνόμενο (undirected) γράφημα

Βρόγχος καλείται μια ακμή με αρχή και πέρας την ίδια κορυφή.

Μηδενικό Γράφημα καλείται κάθε γράφημα χωρίς ακμές ή κορυφές

- **Κατευθυνόμενο Γράφημα**

Κατευθυνόμενο (directed) Γράφημα είναι ένα διατεταγμένο ζεύγος *G* = (*V*, *E*) αποτελούμενο από το σύνολο *V* των κορυφών και ένα σύνολο *E* από ακμές Μια ακμή σχετίζεται με δύο κορυφές και η σχέση απεικονίζεται ως <u>διατεταγμένο</u> ζεύγος των κορυφών σε σχέση με τη συγκεκριμένη ακμή.

- **Υπογραφήμα**

Καλούμε Υπογράφημα (sub graph) H(V',E') ενός γραφήματος G (V,E), το γράφημα H όταν V'⊆V και E'⊆ E.

- **Γεννητικό Υπογραφήμα**

Καλούμε Γεννητικό Υπογράφημα (spanning sub graph) H(V',E') ενός γραφήματος G (V,E), το γράφημα H όταν V'=V και E'⊆ E. Επομένως το γεννητικό υπογράφημα περιέχει το σύνολο των κορυφών του γραφήματος.

- **Συμπλήρωμα Γραφήματος**

Συμπλήρωμα Γραφήματος $G(V,E)$ καλείται το γράφημα $\bar{G}(V,\bar{E})$ το οποίο περιέχεις τις ίδιες κορυφές με το G και κάθε ακμή μεταξύ κορυφών που δεν ανήκει στο G. Δηλ. αν $u,v\ \in V\ \eta\ \alpha\kappa\mu\acute{\eta}\ (u,v) \notin E\ \tau\acute{o}\tau\varepsilon\ (u,v)\ \in \bar{E}$

- **Πλήρες Γράφημα**

Πλήρες (complete) Γράφημα ή Κλίκα καλούμε ένα γράφημα G όταν κάθε ζεύγος κορυφών του συνδέεται με μια ακμή .

- **Διμερές Γράφημα**

Διμερές (bipartite) γράφημα G(V,E) καλείται ένα γράφημα όταν το σύνολο των κορυφών του V μπορεί να χωριστεί σε δύο ξένα μεταξύ τους υποσύνολα $V_1$ και $V_2$ ($V= V_1 \cup V_2$) τέτοια ώστε κάθε ακμή του να έχει ένα άκρο κορυφή στο $V_1$ και άλλο άκρο κορυφή στο $V_2$.

- **Πλήρες Διμερές Γράφημα**

Πλήρες Διμερές (complete bipartite) γράφημα G (V,E) καλείται ένα διμερές γράφημα όταν περιέχει κάθε δυνατή ακμή που μπορούμε να σχηματίσουμε με ένα άκρο κορυφή στο $V_1$ και άλλο άκρο κορυφή στο $V_2$.

- **Βαθμός Κορυφής**

Βαθμό Κορυφή (vertex degree) καλούμε τον αριθμό των ακμών που περιέχουν την κορυφή. Στην περίπτωση βρόγχου θεωρούμε περιέχει την κορυφή δύο φορές.

- **Επίπεδο Γράφημα**

Ένα επίπεδο (planar) γράφημα είναι ένα γράφημα του οποίου οι κορυφές και οι ακμές μπορούν να τοποθετηθούν στο επίπεδο και να μην υπάρχουν δύο ακμές οι οποίες να τέμνονται.

- **Βεβαρημένο ή Σταθμισμένο Γράφημα**

Βεβαρημένο (Weighted) ή Σταθμισμένο Γράφημα είναι ένα γράφημα στο οποίο σε κάθε ακμή έχει συσχετισθεί ένας αριθμός (βάρος).

- **Μονοπάτι**

Μονοπάτι (path) καλούμε μια ακολουθία ακμών $e_1,e_2...e_k \in E$ τέτοια ώστε η κορυφή που καταλήγει μια ακμή να είναι η αρχή της επόμενη ακμής, πλην της τελευταίας. Απλό μονοπάτι ονομάζουμε ένα μονοπάτι για το οποίο σε κάθε κορυφή προσπίπτουν μόνο δύο ακμές, πλην της αρχικής και της τελικής. Δηλαδή το απλό μονοπάτι περνάει μια και μόνο μια φορά από κάθε κορυφή. Μονοπάτι u-v καλούμε ένα Μονοπάτι που ξεκινά από την κορυφή u και καταλήγει στην κορυφή v

Κόστος ή Μήκος Μονοπατιού σε ένα Βεβαρυμμένο Γράφημα καλούμε το άθροισμα των βαρών των ακμών του.

- **Κύκλος**

Κύκλο (tour, cycle) καλούμε ένα μονοπάτι που η τελευταία του ακμή καταλήγει στην κορυφή που ξεκινά η πρώτη ακμή του.

- **Συνεκτικό – Μη Συνεκτικό Γράφημα**

Συνεκτικό γράφημα (connected) καλείται ένα γράφημα G (V,E) όταν για κάθε ζεύγος κορυφών του $u,v \in V$ υπάρχει u-v μονοπάτι στο G.

Μη Συνεκτικό Γράφημα καλείται ένα γράφημα G (V,E) όταν υπάρχει τουλάχιστον ένα ζεύγος κορυφών του $u,v \in V$ για το οποίο δεν υπάρχει u-v μονοπάτι στο G.

- **Κύκλος του Euler.**

Κύκλο του Euler ονομάζουμε ένα κύκλο που περιέχει όλες τις κορυφές και κάθε ακμή ακριβώς μία φορά. Ένα γράφημα έχει κύκλο του Euler αν και μόνο αν είναι συνεκτικό και κάθε κορυφή έχει άρτιο βαθμό.

- **Κύκλος του Hamilton.**

Κύκλο του Hamilton ονομάζουμε ένα κύκλο που περιέχει και όλες τις κορυφές του γραφήματος μόνο μία φορά.

- **Πίνακας Γειτνίασης και Πίνακας Κόστους ή Αποστάσεων**

Σε έναν γράφημα με n κορυφές ο πίνακας γειτνίασης n*n περιέχει 1 στις θέσεις (u,v) όπου υπάρχει ακμή που συνδέει την κορυφή u με την κορυφή v και 0 διαφορετικά. Αν το γράφημα είναι μη κατευθυνόμενο ο πίνακας είναι συμμετρικός.

Ο Πίνακας Κόστους ή Αποστάσεων στη θέση στις θέσεις (u,v) εφόσον υπάρχει ακμή που συνδέει την κορυφή u με την κορυφή v περιέχει το βάρος, διαφορετικά το 0.

## 3.2. Βασικοί Ορισμοί και Αλγόριθμοι Δένδρων

### 3.2.1. Δένδρο - Γεννητικό Δένδρο

Δέντρο (tree) καλείται ένα συνεκτικό γράφημα το οποίο δεν περιλαμβάνει κύκλους.

Ένα δέντρο Τ είναι γεννητικό δέντρο (spanning tree) του γραφήματος G, αν το Τ είναι γεννητικό υπογράφημα του G.

### 3.2.2. Ελάχιστο Γεννητικό Δένδρο

Ελάχιστο Γεννητικό Δέντρο (minimum spanning tree) Τ του συνεκτικού βεβαρυμμένου γραφήματος G(V,E) είναι ένα γεννητικό δέντρο του οποίου το άθροισμα των βαρών των ακμών είναι ελάχιστο . Για την εύρεση του ελάχιστου γεννητικού δέντρου ενός γραφήματος υπάρχουν δύο αλγόριθμοι.

### 3.2.3. Ο Αλγόριθμος Kruskal

Ο Αλγόριθμος του Kruskal [21, pp. 48-50] παρουσιάστηκε το 1956 και αποτελεί έναν άπληστο (greedy) κατασκευαστικό αλγόριθμο για την εύρεση ενός ελάχιστου γεννητικά δέντρου ενός γραφήματος. Τα βήματα του αλγόριθμου είναι τα παρακάτω:

1. Ταξινομούμε τις ακμές του γραφήματος βάσει κόστους κατά αύξουσα σειρά.
2. Κατασκευάζουμε ένα υπό - γράφημα S του γραφήματος G έτσι ώστε:
    a. Για κάθε ακμή e στον πίνακα μας εξετάζουμε αν οι κορυφές της συνδέονται στο S και αν όχι την προσθέτουμε στο S.
    b. Ο Αλγόριθμος τερματίζει όταν το S έχει n-1 ακμές
3. Το S είναι ένα ελάχιστο γεννητικό δέντρο του G

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(m + nlogn)$ όπου $n$ ο αριθμός των ακμών του γραφήματος και O($m$) η πολυπλοκότητα του αλγορίθμου ταξινόμησης. Αναλυτικό παράδειγμα εφαρμογής του αλγορίθμου παρουσιάζεται στην παράγραφο 0

### 3.2.4. Ο Αλγόριθμος Prim

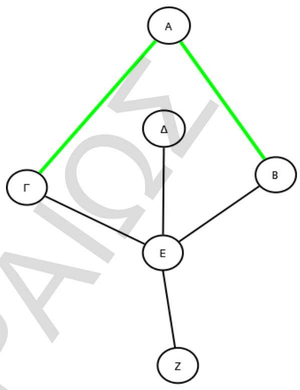
Ο Αλγόριθμος του Prim [22, p. 1389–1401] παρουσιάστηκε το 1957 και αποτελεί επίσης έναν άπληστο (greedy) κατασκευαστικό αλγόριθμο για ελάχιστα γεννητικά δέντρα. Τα βήματα του αλγόριθμου είναι τα παρακάτω:

1. Επιλέγουμε τυχαία μια κορυφή του γραφήματος και την προσθέτουμε στο υπό - γράφημα S του γραφήματος G.
2. Προσθέτουμε στο S μια ακμή επιλέγοντας αυτήν με το μικρότερο κόστος ανάμεσα σε αυτές που συνδέουν το S με τις κορυφές που δεν ανήκουν στο S
3. Επαναλαμβάνουμε το δεύτερο βήμα μέχρι όλες οι κορυφές να ανήκουν στο S.
4. Το S είναι ένα ελάχιστο γεννητικό δέντρο του G

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(m + nlogn)$ όπου m ο αριθμός των κορυφών και n ο αριθμός των ακμών του γραφήματος .

#### 3.2.5. 1-Δέντρο (1-tree)

Δοθείσης κορυφής γραφήματος, την οποία καλούμε κορυφή 1, 1-Δέντρο (1-Tree) ονομάζουμε ένα δέντρο των κορυφών {2,3,…n} του γραφήματος μαζί με δύο διακριτές ακμές που προσπίπτουν στην κορυφή 1.




**Εικόνα 3 1-Δέντρο**

Ένα 1-Δέντρο περιέχει πάντα ένα κύκλο.

Ελάχιστο 1-Δέντρο είναι το 1-Δέντρο του οποίου το άθροισμα των βαρών των ακμών είναι ελάχιστο ανάμεσα στα δυνατά 1-Δέντρα.

Για την εύρεση ενός ελαχίστου 1-Δέντρου βρίσκουμε το ελάχιστο γεννητικό δέντρο του υπογραφήματος των κορυφών {2,3,…n} και κατόπιν προσθέτουμε τις δύο «φθηνότερες» προσπίπτουσες ακμές στην κορυφή 1.

Ένας κύκλος Hamilton είναι και 1-Δέντρο.

Ένα Ελάχιστο 1-Δέντρο εφόσον είναι κύκλος του Hamilton είναι και ο ελάχιστος.

### 3.3. Ταιριάσματα

Ταίριασμα (matching) ενός γραφήματος G = (V , E) καλείται ένα υποσύνολο M ⊆ E του συνόλου των ακμών του, τέτοιο ώστε κάθε κορυφή του γραφήματος G να ανήκει <u>το πολύ</u> σε μια ακμή του M. Μέγεθος του ταιριάσματος είναι ο πληθικός αριθμός |M|.

#### 3.3.1. Μέγιστικό Ταίριασμα

Μεγιστικό Ταίριασμα (maximal matching) ενός γραφήματος G = (V , E) καλείται ένα ταίριασμα, στο οποίο δεν μπορούμε να προσθέσουμε καμία άλλη ακμή.

#### 3.3.2. Μέγιστο Ταίριασμα

Μέγιστο Ταίριασμα (maximum matching) ενός γραφήματος G = (V , E) καλείται ένα ταίριασμα, εφόσον δεν μπορούμε να βρούμε μεγαλύτερο από αυτό.

Αριθμός ταιριάσματος (matching number) ενός γραφήματος είναι το μέγεθος ενός μέγιστου ταιριάσματος του.

#### 3.3.3. Πλήρες Ταίριασμα

Πλήρες ή τέλειο Ταίριασμα (perfect matching) λέγεται ένα ταίριασμα αν κάθε κορυφή του γραφήματος G να ανήκει <u>ακριβώς</u> σε μια ακμή του M.

Ένα τέλειο ταίριασμα είναι και μέγιστο.

- Ένα γράφημα περιττού πλήθος κορυφών δεν μπορεί να έχει τέλειο ταίριασμα.
- Ένα γράφημα έχει τέλειο ταίριασμα αν και μόνο αν ο αριθμός των κορυφών του ισούται με το διπλάσιο του Αριθμού ταιριάσματος του γραφήματος.

### 3.3.4. Πλήρες Ταίριασμα Ελαχίστου Κόστους

Πλήρες Ταίριασμα Ελαχίστου Κόστους (minimum weight perfect matching) λέγεται ένα πλήρες ταίριασμα που το κόστος του είναι μικρότερο οποιοδήποτε άλλου.

Το 1965 ο Edmonds [23, p. 449–467 ] παρουσίασε τον αλγόριθμο Blossom («άνθος») ο οποίος υπολογίζει ένα Πλήρες Ταίριασμα Ελαχίστου Κόστους ενός μη κατευθυνόμενου βεβαρυμμένου γραφήματος σε πολυωνυμικό χρόνο. Η πολυπλοκότητα του αλγορίθμου είναι $O(n^2m)$ όπου $n$ ο αριθμός των κορυφών του γραφήματος και $m$ ο αριθμός των ακμών. Από τότε έχουν υπάρξει σημαντικές βελτιώσεις και η πιο πρόσφατη υλοποίηση προτάθηκε από τον Gabow (1990) με πολυπλοκότητα $O(n(m + nlogn))$ .[7].

# 4. Πολυπλοκότητα του προβλήματος

## 4.1. Θεωρία Πολυπλοκότητας - Οι κλάσεις P και NP

Η θεωρία πολυπλοκότητας αναπτύχθηκε από την δεκαετία του 1970 και μετά, με σκοπό την κατηγοριοποίηση των διαφόρων προβλημάτων και των αλγορίθμων επίλυσης τους ως προς τις απαιτήσεις τους σε υπολογιστικό χρόνο και χώρο (μνήμη). Με βάση τη θεωρία πολυπλοκότητας ο χώρος όλων των προβλημάτων ιεραρχείται σε κλάσεις χρονικής και χωρικής πολυπλοκότητας (complexity classes).

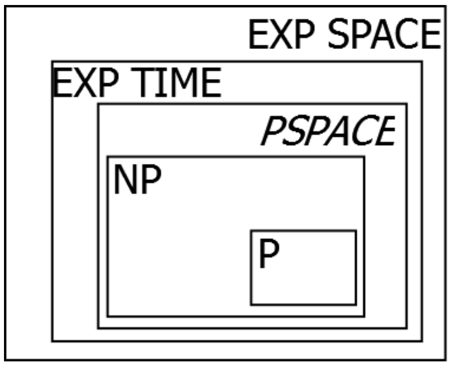
Η πρώτη κλάση που εξετάζουμε είναι η P (ντετερμινιστικού πολυωνυμικού χρόνου - Deterministic Polynomial Time), στην οποία ανήκουν τα προβλήματα που μπορούν να λυθούν σε πολυωνυμικό χρόνο. Τα προβλήματα της κλάσης P χαρακτηρίζονται ως «εύκολα», αφού λύνονται γρήγορα, καθώς υπάρχει αλγόριθμος επίλυσης που ο αριθμός των πράξεων του είναι πολυωνυμικής τάξης $O(n^k)$, όπου n το μέγεθος των δεδομένων της εισόδου του προβλήματος και k σταθερά.

Η δεύτερη κλάση είναι η NP ( μη ντετερμινιστικού πολυωνυμικού χρόνου – Non deterministic Polynomial Time), στην οποία ανήκουν τα προβλήματα που αν μας δοθεί μια λύση μπορούμε να ελέγξουμε την ορθότητα της με έναν αλγόριθμο σε πολυωνυμικό χρόνο. Ο όρος «μη ντετερμινιστικός» αναφέρεται στην «αυθαίρετη» επιλογή του επόμενου βήματος ενός αλγορίθμου στην περίπτωση πολλών επιλογών και προέρχεται από αρχικό ορισμό της πολυπλοκότητας αναφορικά με τις μηχανές Turing. Η εξήγηση ενός προβλήματος κλάσης NP γίνεται με την μέθοδο της «πιστοποίησης». Η μη ντετερμινιστική μηχανή Turing επιτρέπεται να μαντέψει («oracle») μια λύση σ' ένα δεδομένο πρόβλημα, αλλά μετά πρέπει να δώσει ένα «πιστοποιητικό» για την λύση, σε πολυωνυμικό χρόνο.

Οι κλάσεις P και NP ορίζονται για προβλήματα απόφασης, δηλαδή προβλήματα στα οποία καλούμαστε να απαντήσουμε μια συγκεκριμένη ερώτηση με ΝΑΙ ή ΟΧΙ. Η βασική διαφορά των δύο κλάσεων είναι ότι ενώ στην μεν P κάθε βήμα υπολογισμού του αλγορίθμου καθορίζεται πλήρως από το προηγούμενο, στην NP δεν υπάρχει αιτιολογία η οποία δικαιολογεί μια συγκεκριμένη διαδοχή βημάτων. Η κατηγορία P είναι υποσύνολο της κατηγορίας NP ( $P \subseteq NP$ ). Το πιο διάσημο πρόβλημα της Θεωρίας Πολυπλοκότητας αλλά και της Θεωρητικής Πληροφορικής είναι να αποδειχθεί αν ταυτίζονται ή όχι οι δύο κλάσεις P και NP.

Άλλες κλάσεις πολυπλοκότητας είναι:

- η PSPACE που περιέχει τα προβλήματα απόφασης που απαιτούν πολυωνυμικής τάξης χώρο.
- η EXP TIME και περιέχει τα προβλήματα απόφασης που απαιτούν εκθετικής τάξης χρόνο $O(2^{kn})$,
- η EXP SPACE που περιέχει τα προβλήματα απόφασης που απαιτούν εκθετικής τάξης χώρο.




**Εικόνα 4 : Ιεράρχηση Κλάσεων Πολυπλοκότητας**

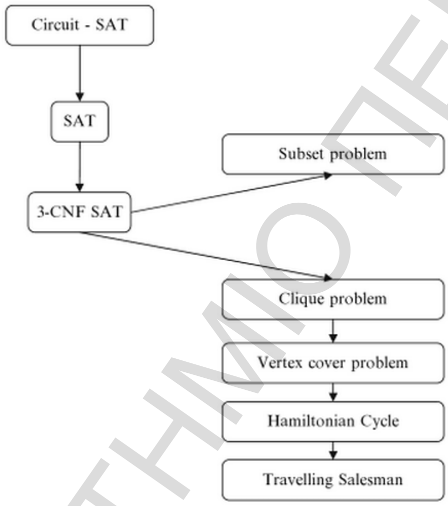
Η ιεραρχία των κλάσεων ακολουθεί την παρακάτω σχέση υποσυνόλων:

**$P \subseteq NP \subseteq PSPACE \subseteq EXP\ TIME \subseteq EXP\ SPACE$**

## 4.2. NP - Πληρότητα

Ένα πρόβλημα ονομάζεται NP-Πλήρες (NP-Complete) αν ανήκει στην κλάση NP και κάθε άλλο πρόβλημα που ανήκει στην NP μπορεί να αναχθεί πολυωνυμικά σε αυτό. Αυτά τα προβλήματα είναι και τα δυσκολότερα της κατηγορίας NP. Το 1971 ο Stephen Cook έδειξε ότι το πρόβλημα ικανοποιησιμότητας (Satisfiability Problem - SAT) είναι NP πλήρες [24]. Το 1972 ο Karp εξέδωσε μια λίστα με 21 επιπλέον NP-Πλήρη προβλήματα [25].

Η διαδικασία της μετάθεσης από ένα πρόβλημα σε ένα άλλο καλείται αναγωγή (reduction) και είναι μεταβατική. Έτσι για να δείξουμε τη NP-πληρότητα ενός δεδομένου προβλήματος Α, το μόνο που πρέπει να κάνουμε είναι να δείξουμε ότι το πρόβλημα ανήκει στην κλάση NP και να ανάγουμε ένα πρόβλημα που είναι γνωστό ότι είναι NP-Πλήρες (π.χ. SAT) στο πρόβλημα Α.




**Εικόνα 5: Δένδρο με αναγωγές προβλημάτων για απόδειξη της NP-πληρότητας [26, p. 33]**

## 4.3. NP - Δυσκολία

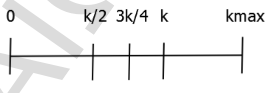
Ένα πρόβλημα Χ ονομάζεται NP-Δύσκολο (NP-Hard) αν για κάθε πρόβλημα Π ∈ NP μπορεί γίνει αναγωγή από το Χ και Χ∉NP. Αν επιπλέον Χ ∈ NP τότε το Χ είναι NP – πλήρες. Η διαφορά του ορισμού της NP-Δυσκολίας από αυτόν της NP-Πληρότητας είναι ότι αναφέρεται σε προβλήματα τα οποία δεν ανήκουν στην κλάση NP. [27] [28, p. 397].

## 4.4. Πρόβλημα Απόφασης – Πρόβλημα Βελτιστοποίησης

Η θεωρία πολυπλοκότητας, όπως είδαμε, επεξεργάζεται μόνο προβλήματα απόφασης. Τα περισσότερα προβλήματα, Συνδυαστικής Βελτιστοποίησης που μας απασχολούν, όπως και το πρόβλημα TSP, δεν είναι προβλήματα απόφασης. Μπορούμε όμως να τα μετασχηματίσουμε σε προβλήματα απόφασης με την εισαγωγή νέων παραμέτρων. Με τον τρόπο αυτό είναι δυνατόν να τα κατηγοριοποιήσουμε ως προς τη δυσκολία τους.

Στο TSP ορίζουμε το σύνολο S με όλους τους δυνατούς κύκλους του γραφήματος, οι οποίοι περνούν μία φορά από όλους τους κόμβους του γραφήματος. Ορίζουμε συνάρτηση f ως αντικειμενική συνάρτηση (objective function), η οποία μας δίνει μια αποτίμηση της κάθε εφικτής λύσης $c \in S$, δηλαδή το κόστος του κάθε κύκλου. Επομένως μας ενδιαφέρει να βελτιστοποιήσουμε την f στο σύνολο των εφικτών λύσεων. Ζητάμε δηλαδή εφικτή λύση $c^* \in S$ τέτοια ώστε: $f(c^*)=\min f(c)$. Εισάγοντας έναν αριθμό k στα δεδομένα μετατρέπουμε το πρόβλημα σε πρόβλημα απόφασης, και ζητάμε $c \in S$, $f(c) \leq k$.

Εάν γνωρίζουμε εάν αλγόριθμο για το πρόβλημα βελτιστοποίησης τότε προφανώς μπορούμε να επιλύσουμε και το πρόβλημα απόφασης. Αντίστροφα αν γνωρίζουμε έναν αλγόριθμο για το πρόβλημα απόφασης τότε με δυαδική αναζήτηση μπορούμε να βρούμε την άριστη λύση για το πρόβλημα βελτιστοποίησης, επιβαρύνοντας τον αλγόριθμο, κατά πολυωνυμικό παράγοντα πολυπλοκότητας $\log_2(n)$.




**Εικόνα 6 Δυαδική Αναζήτηση**

## 4.5. Πολυπλοκότητα του TSP

Για να αποδείξουμε ότι το TSP είναι NP-Πλήρες απαιτείται να δείξουμε ότι το TSP-Απόφασης ανήκει στην κλάση NP και κατόπιν ότι ένα ήδη γνωστό NP-Πλήρες πρόβλημα μπορεί να γίνει αναγωγή σε TSP.

Πρώτα θα αποδείξουμε ότι το πρόβλημα TSP-Απόφασης (TSP-Decide) ανήκει στην κλάση NP. Δεδομένου συνεκτικού γραφήματος G, ενός κόστους k και μιας περιοδείας C στο G, μπορούμε να ελέγξουμε σε πολυωνυμικό χρόνο αν η C περιέχει όλες τις κορυφές του G και αν το κόστος της είναι μικρότερο ή ίσο του k. Η διαδικασία ολοκληρώνεται προφανώς σε πολυωνυμικό χρόνο και συνεπώς το TSP-Απόφασης ανήκει στην κλάση NP.

Στη συνέχεια θα κάνουμε αναγωγή από Κύκλο Hamilton σε πρόβλημα TSP, καθώς ο κύκλος Hamilton είναι NP-Πλήρες πρόβλημα [29, p. 47].

Θα χρησιμοποιήσουμε τον παρακάτω μετασχηματισμό:

Έστω γράφημα G=(V,E) με κύκλο Hamilton h. Κατασκευάζουμε ένα πλήρες γράφημα G′=(V,E′) του προβλήματος TSP, όπου $E' = \{(i, j): i, j \in V \text{ και } i \neq j\}$. Επίσης ορίζουμε μια συνάρτηση κόστους ως εξής:

$$c(i, j) := \begin{cases} 0 \ \varepsilon\acute{\alpha}\nu \ (i,j) \in \mathrm{E} \\ 1 \ \varepsilon\acute{\alpha}\nu \ (i,j) \notin \mathrm{E} \end{cases}$$

Κάθε ακμή που ανήκει στο κύκλο Hamilton h έχει κόστος 0 στο G′ καθώς ανήκει και στο E. Επομένως ο κύκλος h θα έχει συνολικό κόστος 0 στο G′.

Αντίστροφα έστω ότι το G′ έχει κύκλο h′ με κόστος το πολύ 0. Το κόστος της ακμής στο E′ από τον ορισμό είναι 0 και 1. Εφόσον το κόστος του κύκλου h′ είναι 0 τότε και κάθε ακμή πρέπει να έχει κόστος 0. Συνεπώς ο h′ απαρτίζεται μόνο από ακμές του E άρα ο h′ είναι κύκλος Hamilton στο γράφημα G.

Αποδείξαμε λοιπόν ότι το γράφημα G έχει κύκλο Hamilton αν και μόνο αν το γράφημα G′ έχει κύκλο – περιοδεία κόστους 0

Συνεπώς το TSP είναι NP-Πλήρες [30, p. 120] [31, p. 756].

# 5. Ακριβείς Αλγόριθμοι ( Exact Algorithms )

Ένας αλγόριθμος ονομάζεται ακριβής (exact) όταν μας αποδίδει την άριστη λύση για κάθε είσοδο του προβλήματος. Στην περίπτωση ενός TSP προβλήματος άριστη λύση είναι η συντομότερη (shortest) περιοδεία όλων των πόλεων – σημείων του προβλήματος. Στη συνέχεια θα παρουσιάσουμε του βασικούς Ακριβείς αλγορίθμους για το πρόβλημα TSP και θα τους αξιολογήσουμε.

## 5.1. Εξαντλητική Αναζήτηση (Brute force)

Αλγόριθμος Εξαντλητικής ( exhaustive ) αναζήτησης ονομάζεται ένας αλγόριθμος όταν το σύνολο των καταστάσεων που εξετάζει για να βρει τις απαιτούμενες λύσεις είναι ίσο με το χώρο αναζήτησης. Σε ένα πρόβλημα TSP αυτό ισοδυναμεί με τον υπολογισμό όλων των δυνατών περιοδειών του προβλήματος και την εύρεση της περιοδείας με το ελάχιστο κόστος.

Για να εξετάσουμε τον αριθμό των καταστάσεων που θα ελέγξει ο αλγόριθμος θεωρούμε ένα πρόβλημα TSP 33 πόλεων. Επειδή δεν μας ενδιαφέρουν οι διατάξεις μιας διαδρομής ( πχ 1-2-3-…33 είναι ίδιου κόστους με την 2-3-…33-1) υπολογίζουμε τις μεταθέσεις των 33 πόλεων ξεκινώντας από την 1. Σύμφωνα με την πολλαπλασιαστική αρχή έχουμε 32x31x30x…x3x2x1= 32! μεταθέσεις. Επειδή επίσης δεν μας ενδιαφέρει η κατεύθυνση με την οποία θα διασχίσουμε την περιοδεία (δηλ. 1-2-3…-33 είναι ίδια με 33-32-…-3-2-1) καταλήγουμε στο συμπέρασμα ότι υπάρχουν συνολικά 32! / 2 περιοδείες για να εξετάσουμε.

Ενδεικτικά το 32!/2 = 131,565,418,466,846,765,083,609,006,080,000,000.

Αν υποθέσουμε ότι έχουμε στην διάθεση μας έναν υπερυπολογιστή με δυνατότητα 1,5 τρισεκατομμύριού πράξεων το δευτερόλεπτο (ένας απλός υπολογιστής i7 έχει δυνατότητα 100 εκατομμυρίων πράξεων περίπου το δευτερόλεπτο), καθώς επίσης και ότι ο έλεγχος κάθε διαδρομής είναι μία και μόνο πράξη, τότε χρειαζόμαστε περίπου 28 τρισεκατομμύρια χρόνια για να λύσουμε το πρόβλημα μας. Για σύγκριση υπενθυμίζουμε ότι η ηλικία του σύμπαντος υπολογίζεται στα 14 δισεκατομμύρια χρόνια.

Είναι προφανής η μη αποδοτικότητα του αλγορίθμου αλλά και το μέγεθος του προβλήματος καθώς ο αλγόριθμος είναι τάξης πολυπλοκότητας O((n-1)!).

## 5.2. Ο Αλγόριθμος Bellman – Held - Karp

Ο Bellman [8] και ανεξάρτητα οι Held M. και Karp R. [7]. , το 1962 διατύπωσαν έναν αλγόριθμο εκθετικής τάξεως $O(n^2 2^n)$. Η βελτίωση είναι σημαντική για μεγάλα n, σε σχέση με το (n-1)! της Εξαντλητικής αναζήτησης, αλλά και πάλι οι χρόνοι ολοκλήρωσης των υπολογισμών παραμένουν τεράστιοι για τα ανθρώπινα δεδομένα.

Για να περιγράψουμε την κεντρική ιδέα του αλγορίθμου, θεωρούμε μια άριστη περιοδεία d ελαχίστου κόστους που ξεκινάει από το σημείο 1 και συνεχίζει στο σημείο m. Το κόστος της d είναι το κόστος της διαδρομής από το σημείο 1 έως το m και το κόστος της υπόλοιπης διαδρομής από το m έως το 1. Δηλαδή $d=d_{(1,m)}+d_{(m,1)}$. Επειδή η d είναι άριστη περιοδεία τότε το μονοπάτι $d_{(m,1)}$ για τα δεδομένα σημεία που περνάει (εκτός αυτών του μονοπατιού (1,m)) είναι και αυτό βέλτιστο – ελάχιστο. Βασιζόμενοι σε αυτή την παρατήρηση θα διατυπώσουμε έναν αναδρομικό αλγόριθμο εύρεσης άριστης λύσης.

Έστω γράφημα G=(V,A) που αντιστοιχεί στο TSP. Ορίζουμε την c(i,S) συνάρτηση υπολογισμού κόστους του συντομότερου μονοπατιού από το $i$ στο 1, το οποίο διέρχεται από όλες τις κορυφές του S ακριβώς μία φορά:

$$c(1, V - \{1\}) = \min_{2\le k\le n}\{\mathrm{cost}(1,k) + c(k, V - \{1,k\})\}$$

και γενικότερα:

$$c(i, S) = \min_{j\in S}\{\mathrm{cost}(i,j) + c(j, S - \{j\})\}$$

Τέλος

$$c(i,\emptyset) = \mathrm{cost}(i,1)$$

Η συνάρτηση cost ( $V \times V \to Z$ ) μας δίνει το κόστος κάθε ακμής του προβλήματος. Χωρίς βλάβη της γενικότητας θεωρούμε τις αποστάσεις των σημείων μεταξύ τους θετικές και την απόσταση κάθε σημείου από τον εαυτό του μηδενική.

Σε κάθε βήμα i υπολογίζουμε το $c(k, S - \{1\})$ για n-1 διαφορετικά k. Τα υπόλοιπα n-2 στοιχεία του S πρέπει να συνδυαστούν ανά $i$ δηλαδή $\binom{n-2}{i}$. Ο χώρος που απαιτεί ο αλγόριθμος σε θέσεις μνήμης είναι $(n-1)\cdot 2^{n-2}$, λόγω της αναδρομικότητας, για τον υπολογισμό της άριστης περιοδείας. [32, p. 189]

Ο αλγόριθμος Bellman - Held-Karp παραμένει μέχρι σήμερα ο καλύτερος ακριβής αλγόριθμος από πλευράς υπολογιστικού κόστους και οποιαδήποτε προσπάθεια βελτίωση του κόστους έχει αποτύχει. Ενδεικτικά αναφέρουμε ότι παραμένει ανοικτό πρόβλημα αν υπάρχει ακριβής αλγόριθμος πολυωνυμικής τάξης $O(1.99999^n)$ [33]

Στην συνέχεια εφαρμόζουμε τον αλγόριθμο σε ένα πρόβλημα ασύμμετρο TSP 4 πόλεων. Έστω C ο πίνακας κόστους των ακμών του αντίστοιχου γραφήματος.

$$C=\begin{pmatrix} - & 2 & 9 & 10 \\ 1 & - & 6 & 4 \\ 15 & 7 & - & 8 \\ 6 & 3 & 12 & - \end{pmatrix}$$

Θέλουμε να υπολογίσουμε την c(1,{2,3,4}).

Υπολογίζουμε όλα τα κόστη για |V|=∅

c(2,∅)=c21=1, c(3,∅)=c31=15, c(4,∅)=c41=6

Υπολογίζουμε όλα τα κόστη για |V|=1

S= {2}:

c(3,{2}) = c32 + c(2, ∅ ) = c32 + c21 = 7 + 1 = 8

c(4,{2}) = c42 + c(2, ∅ ) = c42 + c21 = 3 + 1 = 4

$S = \{3\}$:
$c(2,\{3\}) = c23 + c(3, \emptyset) = c23 + c31 = 6 + 15 = 21$
$c(4,\{3\}) = c43 + c(3, \emptyset) = c43 + c31 = 12 + 15 = 27$
$S = \{4\}$:
$c(2,\{4\}) = c24 + c(4, \emptyset) = c24 + c41 = 4 + 6 = 10$
$c(3,\{4\}) = c34 + c(4, \emptyset) = c34 + c41 = 8 + 6 = 14$
Υπολογίζουμε όλα τα κόστη για $|V|=2$
$S=\{2,3\}$:
$c(4,\{2,3\}) = \min \{c42 + c(2,\{3\}), c43 + c(3,\{2\})\} = \min \{3+21, 12+8\} =$
$= \min \{24, 20\} = 20$
$S=\{2,4\}$:
$c(3,\{2,4\}) = \min \{c32 + c(2,\{4\}), c34 + c(4,\{2\})\} = \min \{7+10, 8+4\} =$
$= \min \{17, 12\} = 12$
$S=\{3,4\}$:
$c(2,\{3,4\}) = \min \{c23 + c(3,\{4\}), c24 + g(4,\{3\})\} = \min \{6+14, 4+27\} =$
$= \min \{20, 31\} = 20$
Υπολογίζουμε το κόστος της άριστης περιοδείας.
$f = c(1,\{2,3,4\}) = \min \{ c12 + c(2,\{3,4\}), c13 + c(3,\{2,4\}), c14 + c(4,\{2,3\}) \}$
$= \min \{2 + 20, 9 + 12, 10 + 20\} = \min \{22, 21, 30\} = 21$
Επειδή $f = c13 + c(3,\{2,4\}) = c13 + c34 + c(4,\{2\}) = c13 + c34 + c42$ συνεπάγεται ότι η άριστη περιοδεία του TSP είναι: $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1$

## 5.3. Διακλάδωση και Οριοθέτηση (Branch & Bound)

Η μέθοδος Διακλάδωσης και Οριοθέτησης (Branch & Bound) θεωρείται από τις αποτελεσματικότερες για την επίλυση προβλημάτων Επιχειρησιακής Έρευνας και ακέραιου προγραμματισμού (IP). Παρουσιάστηκε το 1960 από τους Ailsa H. Land και Alison G. Doig [34] , έχοντας τις ρίζες του στην έρευνα για το πρόβλημα TSP. Η ονομασία του μάλιστα εμφανίζεται σε εργασίες για το TSP ήδη από την δεκαετία του 1950. [2, p. 41]

Ο μέθοδος βασίζεται στην τεχνική «διαίρει και βασίλευε» (divide and conquer) και αποτελεί έναν «οργανωμένο» τρόπο για να κάνουμε εξαντλητική αναζήτηση της άριστης λύσης ενός προβλήματος. Πρώτα χωρίζουμε το σύνολο των επιτρεπτών λύσεων του προβλήματος σε μικρότερα υποσύνολα (branch - διακλάδωση) χτίζοντας ουσιαστικά ένα δέντρο αναζήτησης (search tree).

Στη συνέχεια χρησιμοποιούμε κανόνες (bound – οριοθέτηση) ώστε:

α) να προσδιορισθούν τα υποσύνολα που με τη μεγαλύτερη πιθανότητα περιέχουν άριστη λύση.

β) να προσδιορισθούν τα υποσύνολα που δεν χρειάζεται να διερευνηθούν περαιτέρω, καθώς έχουν μικρή πιθανότητα να περιέχουν άριστη λύση.

Ένας αλγόριθμος Branch & Bound επηρεάζεται από τρείς στρατηγικές επιλογές:

- Την στρατηγική αναζήτησης (search strategy)

  Συνηθέστερες επιλογές είναι η Αναζήτηση Πρώτα σε Πλάτος (breadth first search) η οποία έχει μεγάλο κόστος σε χώρο μνήμης, η Αναζήτηση Πρώτα κατά Κόστος (Best-First search) και η Αναζήτηση Πρώτα σε Βάθος ( depth - First search).
- Την στρατηγική επιλογής φραγμάτων (Bound strategy)
- Την στρατηγική επιλογής ακμών (node selection strategy)

Τα βήματα ενός αλγορίθμου Branch & Bound είναι τα εξής [32, p. 287]:

*Ξεκινάμε με ένα πρόβλημα* $P_0$

*Θέτουμε* $S=\{P_0\}$ *το σύνολο των υπό διερεύνηση υποπροβλημάτων*

*Θέτουμε την μεταβλητή με την τιμή της καλύτερης λύσης* $UB=\infty$

*Επαναλαμβάνουμε όσο το S δεν είναι άδειο:*

> *Επιλέγουμε ένα υποπρόβλημα από το* $P \in S$ *και το αφαιρούμε από το S*
>
> *Αναλύουμε το P σε μικρότερα υποπροβλήματα* $P_1, P_2, \ldots P_k$
>
> *Για κάθε* $P_i$ , $i=1..k$
>
> > *Εάν το* $P_i$ *είναι λύση του προβλήματος τότε ενημέρωσε το UB*
> >
> > *αλλιώς αν το κάτω φράγμα* $LB(P_i) < UB$ *τοποθέτησε το* $P_i$ *στο S*

*Επέστρεψε την τιμή της καλύτερης λύσης UB.*

Η διαδικασία μπορεί να αναπαρασταθεί με ένα δέντρο αναζήτησης ( search tree), του οποίου κάθε κόμβος είναι και ένα υποπρόβλημα, ενώ ο αρχικός κόμβος είναι το ίδιο το πρόβλημα.

Ο Eastman το 1958 [35], παρουσίασε έναν αλγόριθμο που τυποποιεί το TSP σαν πρόβλημα ανάθεσης (assignment problem - AP) με επιπρόσθετους περιορισμούς. Στο Πρόβλημα Ανάθεσης πρέπει να ζευγαρώσουμε κάθε πόλη με μια μόνο άλλη πόλη, ώστε το συνολικό κόστος των ακμών ζευγαρώματος να ελαχιστοποιηθεί. Με όρους Γραμμικού Προγραμματισμού το Πρόβλημα Ανάθεσης είναι ένα «χαλαρό» (relaxation) πρόβλημα TSP. Το πρόβλημα ανάθεσης μπορεί να λυθεί με τον Ουγγρική μέθοδο (Hungarian Method [36] ) με υπολογιστική πολυπλοκότητα τάξης $O(n^3)$.

Αν η λύση του Προβλήματος Ανάθεσης είναι μια περιοδεία τότε έχουμε βρει την άριστη λύση του TSP. Αν όχι την χρησιμοποιήσουμε σαν κάτω φράγμα. Στη συνέχεια προχωράμε σε διακλαδώσεις, αναζητώντας περιοδεία, εξαιρώντας ακμές εντός της λύσης του AP. Εάν το κόστος μιας διακλάδωσης ξεπερνά την λύση του AP τότε αυτή απορρίπτεται.

Διάφοροι αλγόριθμοι Διακλάδωσης και Οριοθέτησης, έχουν προταθεί για το TSP βασιζόμενοι στο πρόβλημα ανάθεσης. Οι σημαντικότεροι είναι των Shapiro (1966) [37], Garfinkel (1973) [38], Smith, Srinivasan and Thompson (1977) [39], Balas και Christofides (1981) [40] and Miller and Pekny (1991) [41].

Γενικά οι αλγόριθμοι Branch & Bound στην χειρότερη περίπτωση δεν είναι καλύτεροι από την εξαντλητική αναζήτηση. Η απόδοση τους εξαρτάται από την στρατηγική για την διακλάδωση και την οριοθέτηση που υλοποιούν, καθώς και την στρατηγική επιλογής ακμών. Ειδικά η επιλογή ενός «καλού» αρχικού φράγματος, έχει σημαντικές επιπτώσεις στο μέγεθος του δέντρου αναζήτησης και επομένως στην πολυπλοκότητα. [42, p. 585]

Στη συνέχεια παρουσιάζουμε αλγόριθμο Branch & Bound Best-First για συμμετρικό TSP. [43, p. 329] [32, p. 273]. Για τον υπολογισμό του κάτω φράγματος (LB) κάθε στιγμιότυπου του προβλήματος, αθροίζουμε το κόστος των ακμών που ήδη συμμετέχουν στο στιγμιότυπο και επιλέγουμε να αθροίσουμε τις ελαχίστου κόστους ακμές για κάθε κορυφή που δεν συμμετέχει ήδη στο στιγμιότυπο.

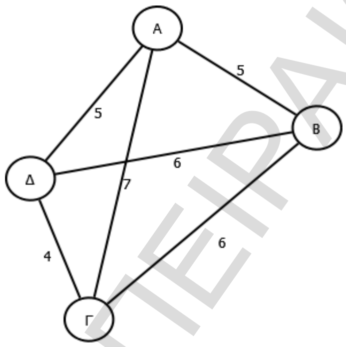
Τα βήματα του αλγορίθμου είναι:

1. Θέτουμε άνω Φράγμα (UB) το άπειρο ή το κόστος γνωστής περιοδείας.
2. Επιλέγουμε κορυφή εκκίνησης.
3. Για κάθε κορυφή που γειτνιάζει με την κορυφή που επιλέξαμε, υπολογίζουμε το κάτω φράγμα LB του στιγμιότυπου του προβλήματος. Από την διαδικασία εξαιρούνται οι ακμές που συνδέουν κορυφές εντός του στιγμιότυπου και άρα δημιουργείται κύκλος, εκτός αν ήδη συμμετέχουν n-1 κορυφές στο στιγμιότυπο όποτε έχουμε μια περιοδεία.
4. Κριτήρια Διακοπής και Διακλάδωσης
    a. Εάν με την προσθήκη της ακμής έχουμε μια περιοδεία τότε θέτουμε UB=LB της ακμής.
    b. Τα στιγμιότυπα με LB≥UB απορρίπτονται για διακλάδωση καθώς δεν μπορούν να δώσουν καλύτερη περιοδεία. Εάν απορρίπτονται όλα τα εναπομείναντα στιγμιότυπα, τότε έχουμε την άριστη περιοδεία και ο αλγόριθμος τερματίζει.
5. Επιλέγουμε το στιγμιότυπο με το μικρότερο LB, ως αυτό με την μεγαλύτερη πιθανότητα να περιέχεται στη άριστη λύση, για να συνεχίσουμε την

διακλάδωση. Σε περίπτωση ισότητας επιλέγουμε αυτό που βρίσκεται στο μεγαλύτερο επίπεδο του δέντρου αναζήτησης.

6. Επιλέγουμε την επόμενη κορυφή και συνεχίζουμε από το βήμα 3

Θα εφαρμόσουμε τον αλγόριθμο σε συμμετρικό πρόβλημα TSP 4 πόλεων. Έστω C ο πίνακας κόστους των ακμών του αντίστοιχου γραφήματος.

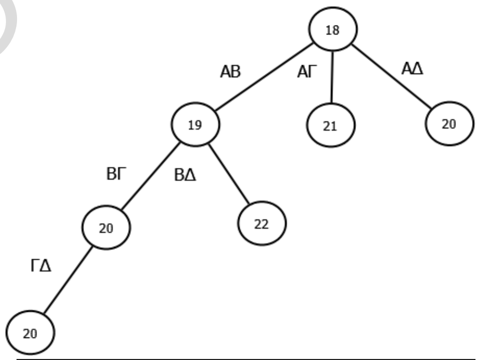
| ΠΙΝΑΚΑΣ ΚΟΣΤΟΥΣ | ΓΡΑΦΗΜΑ ΠΡΟΒΛΗΜΑΤΟΣ |
|---|---|
| $C=\begin{pmatrix} - & 5 & 7 & 5 \\ 5 & - & 6 & 6 \\ 7 & 6 & - & 4 \\ 5 & 6 & 4 & - \end{pmatrix}$ |  |

1. Θέτουμε UB=∞ και LB=5+5+4+4=18

$$\begin{pmatrix} - & 5 & 7 & 5 \\ 5 & - & 6 & 6 \\ 7 & 6 & - & 4 \\ 5 & 6 & 4 & - \end{pmatrix}$$

2. Επιλεγούμε σαν κορυφή εκκίνησης την Α

3. Υπολογίζουμε το ελάχιστο κόστος μιας περιοδείας που μπορεί να συμμετέχει κάθε κορυφή που συνδέεται με την Α.




**Εικόνα 7 Δέντρο Branch & Bound**

| | |
|---|---|
| Για την ΑΒ υπολογίζουμε LB=5+6+4+4=19<br>Η ΒΑ δεν υπολογίζεται καθώς δημιουργεί κύκλο (ΑΒΑ). | $\begin{pmatrix} - & 5 & - & - \\ - & - & 6 & 6 \\ 7 & - & - & 4 \\ 5 & - & 4 & - \end{pmatrix}$ |
| Για την ΑΓ υπολογίζουμε LB=7+5+4+5=21<br>Η ΓΑ δεν υπολογίζεται καθώς δημιουργεί κύκλο (ΑΓΑ) | $\begin{pmatrix} - & - & 7 & - \\ 5 & - & - & 6 \\ - & 6 & - & 4 \\ 5 & 6 & - & - \end{pmatrix}$ |
| Για την ΑΔ υπολογίζουμε LB=5+5+6+4=20<br>Η ΔΑ δεν υπολογίζεται καθώς δημιουργεί κύκλο (ΑΔΑ) | $\begin{pmatrix} - & - & - & 5 \\ 5 & - & 6 & - \\ 7 & 6 & - & - \\ - & 6 & 4 & - \end{pmatrix}$ |

4. Έλεγχος Κριτηρίων
    a. Δεν σχηματίζεται περιοδεία
    b. Δεν υπάρχει ακμή με LB≥UB
5. Επιλέγουμε το στιγμιότυπο με την ακμή AB για να συνεχίσουμε την διακλάδωση καθώς έχει το μικρότερο LB=19
6. Επιλέγουμε την κορυφή B η οποία δεν έχει ακμή εξόδου
7. Υπολογίζουμε το ελάχιστο κόστος μιας περιοδείας που μπορεί να συμμετέχει κάθε κορυφή που συνδέεται με την B.Η BA απορρίπτεται γιατί λόγω της συμμετρικότητας συμμετέχει ήδη στην περιοδεία.

| | |
|---|---|
| Για την ΒΓ υπολογίζουμε LB=5+6+4+5=20 | $\begin{pmatrix} - & \mathbf{5} & - & - \\ - & - & \mathbf{6} & - \\ 7 & - & - & 4 \\ 5 & - & - & - \end{pmatrix}$ |
| Για την ΒΔ υπολογίζουμε LB=5+6+7+4=22 | $\begin{pmatrix} - & \mathbf{5} & - & - \\ - & - & - & \mathbf{6} \\ 7 & - & - & - \\ 5 & - & 4 & - \end{pmatrix}$ |

8. Έλεγχος Κριτηρίων
    a. Δεν σχηματίζεται περιοδεία
    b. Δεν υπάρχει ακμή με LB≥UB
9. Έχουμε δύο στιγμιότυπα με το μικρότερο LB=20. Επιλέγουμε αυτό με τις ακμές A→B→Γ για να συνεχίσουμε την διακλάδωση καθώς είναι σε μεγαλύτερο επίπεδο του δέντρου αναζήτησης.
10. Επιλέγουμε την κορυφή Γ η οποία δεν έχει ακμή εξόδου
11. Υπολογίζουμε το ελάχιστο κόστος μιας περιοδείας που μπορεί να συμμετέχει κάθε ακμή.

Η ΓΑ αποκλείεται γιατί δημιουργεί κύκλο.

Η ΓΒ απορρίπτεται γιατί λόγω της συμμετρικότητας συμμετέχει ήδη στην περιοδεία.

Για την ΓΔ $\begin{pmatrix} - & \mathbf{5} & - & - \\ - & - & \mathbf{6} & - \\ - & - & - & \mathbf{4} \\ 5 & - & - & - \end{pmatrix}$ LB=5+6+4+5=20

12. Έλεγχος Κριτηρίων
    a. Με την ΓΔ σχηματίζεται περιοδεία (1-2-3-4-1) με κόστος 20.
    b. Θέτουμε UB=20
    c. Δεν υπάρχει ακμή με LB<UB. Συνεπώς η περιοδεία είναι η άριστη και ο αλγόριθμος τερματίζει.

# 6. Προσεγγιστικοί Αλγόριθμοι – Κλάσεις Προσεγγισιμότητας

Οι Ακριβείς Αλγόριθμοι υπολογίζουν πάντα την άριστη λύση ενός προβλήματος TSP αλλά αποδίδουν ικανοποιητικά για προβλήματα πολύ μικρού μεγέθους όπως είδαμε στο προηγούμενο κεφάλαιο. Συνήθως η απόδοση τους για προβλήματα δέκα και άνω πόλεων είναι απαγορευτική, καθώς ο όγκος των υπολογισμών αυξάνεται δραματικά.

Προσπαθώντας να ξεπεράσουμε το πρόβλημα, επιλέξαμε να μην απαιτήσουμε την άριστη λύση, αλλά έστω μια σχετικά καλή λύση που προσεγγίζει τη άριστη, την οποία όμως μπορούμε να βρούμε σε πολύ καλύτερο χρόνο. Στα επόμενα κεφάλαια θα εξετάσουμε αλγόριθμους για το TSP, που προσεγγίζουν την άριστη λύση και θα τους ονομάσουμε προσεγγιστικούς.

Στο κεφάλαιο αυτό θα παρουσιάσουμε την μεθοδολογία αξιολόγησης των προσεγγιστικών αλγορίθμων και την κατηγοριοποίηση των προβλημάτων, ανάλογα με την ποιότητα των λύσεων που αποδίδουν.

## 6.1. Απόδοση ενός Προσεγγιστικού Αλγορίθμου

Για να μετρήσουμε την απόδοση ενός προσεγγιστικού αλγορίθμου θα πρέπει να εισάγουμε ένα δείκτη αξιολόγησης του. Αν γνωρίζαμε μία άριστη λύση του προβλήματος τότε θα μπορούσαμε να την συγκρίνουμε με την προσεγγιστική λύση.

Εισάγουμε έτσι την έννοια του λόγου προσέγγισης (approximation ratio). Όσο πιο κοντά στο 1 είναι ο λόγος προσέγγισης τόσο καλύτερη είναι η προσεγγιστική λύση. Επισημαίνουμε ότι για πρόβλημα ελαχιστοποίησης, όπως το TSP, ο λόγος είναι μεγαλύτερος της μονάδας, ενώ για προβλήματα μεγιστοποίησής μικρότερος. Έστω $C_{opt}$ το κόστος της άριστης λύσης του προβλήματος και $C_{apr}$ το κόστος της προσεγγιστικής λύσης. Ορίζουμε ως:

- λόγο προσέγγισης $\rho(n) = \dfrac{C_{apr}}{C_{opt}}$

  όπου $n$ το μέγεθος εισόδου του προβλήματος

Ο αλγόριθμος με λόγο προσέγγισης $\rho(n)$ καλείται και $\rho(n)$-προσεγγιστικός αλγόριθμος.

Ένας $\rho(n)$-προσεγγιστικός αλγόριθμος εξαρτάται από το μέγεθος εισόδου του προβλήματος $n$. Συνήθως όσο μεγαλώνει το μέγεθος του προβλήματος τόσο αποκλίνει και ο αλγόριθμος από το 1 άρα έχουμε χειρότερη προσέγγιση της λύσης.

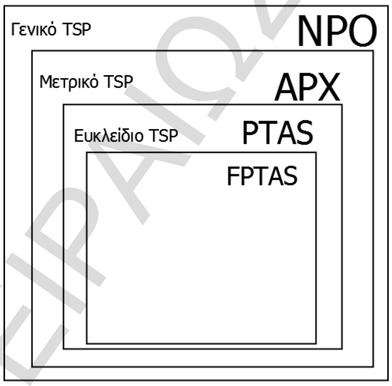
Αν ο λόγος προσέγγισης είναι ανεξάρτητος του μεγέθους εισόδου του προβλήματος τότε χρησιμοποιούμε το όρο $\rho$-προσεγγιστικός αλγόριθμος. Για παράδειγμα ένας 1-προσεγγιστικός αλγόριθμος δίνει πάντα άριστη λύση.

## 6.2. Κλάσεις Προσεγγισιμότητας

Τα προβλήματα βελτιστοποίησης μπορούμε να τα κατατάξουμε σε τέσσερις κύριες κλάσεις, ανάλογα με την απόδοση που έχουν οι αλγόριθμοι που τα προσεγγίζουν.

### 6.2.1. NPO

Στην κλάση προσεγγισιμότητας NPO (NP-optimization problem) ανήκουν όλα τα προβλήματα βελτιστοποίησης των οποίων τα αντίστοιχα προβλήματα απόφασης είναι NP. Το TSP στην γενική του μορφή ανήκει στην NPO κλάση.




**Εικόνα 8 Κλάσεις Προσέγγισιμοτητας και TSP**

### 6.2.2. APX

Η κλάση APX (approximable) είναι υποσύνολο των NPO προβλημάτων, για τα οποία υπάρχει πολυωνυμικού χρόνου $\rho$-προσεγγιστικός αλγόριθμος. Τα προβλήματα αυτά είναι μερικώς προσεγγίσιμα καθώς το $\rho$ δε μπορεί να γίνει οσοδήποτε μικρό. Το Μετρικό - TSP ανήκει στην APX κλάση, καθώς όπως θα δούμε στην παράγραφο 9.2 υπάρχει $1\frac{1}{2}$-προσεγγιστικός αλγόριθμος.

### 6.2.3. Προσεγγιστικό Σχήμα Πολυωνυμικού Χρόνου (PTAS)

Προσεγγιστικό Σχήμα Πολυωνυμικού Χρόνου (Polynomial Time Approximation Scheme - PTAS) ενός προβλήματος, ονομάζουμε ένα προσεγγιστικό αλγόριθμο, ο οποίος για κάθε $\varepsilon > 0$ δίνει λύση με λόγο προσέγγισης $(1 + \varepsilon)$ , σε πολυωνυμικό χρόνο ως προς το μέγεθος της εισόδου του προβλήματος $n$. Ένας σχήμα PTAS καλείται και $(1 + \varepsilon)$- προσεγγιστικός αλγόριθμος.

Η πολυπλοκότητα ενός σχήματος PTAS εξαρτάται από το $n$ και συνήθως το $\varepsilon$ (όχι όμως υποχρεωτικά). Μεγαλώνοντας το μέγεθος $(n)$ του προβλήματος αυξάνει και ο χρόνος εύρεσης της προσεγγιστικής λύσης, καθώς έχουμε δεσμεύσει την ποιότητα της λύσης $(\varepsilon)$. Το ίδιο συμβαίνει αυξάνοντας την ποιότητα δηλαδή μικραίνοντας το $\varepsilon$.

Για παράδειγμα θα εξετάσουμε στην παράγραφο 10 έναν PTAS αλγόριθμο για το Ευκλείδειο –TSP με πολυπλοκότητα $O\left(n(\log n)^{O(\frac{\sqrt{d}}{\varepsilon})^{d-1}}\right)$. Παρατηρούμε ότι ο αλγόριθμος έχει πολυωνυμική τάξη ως προς το μέγεθος εισόδου του προβλήματος $(n)$, αλλά ως προς την ποιότητα της λύσης είναι εκθετικής τάξης αφού το $\varepsilon$ βρίσκεται στον παρονομαστή του εκθέτη. Ζητώντας καλύτερη προσέγγιση, δηλαδή μικραίνοντας το $\varepsilon$, αυξάνουμε εκθετικά το κόστος πολυπλοκότητας του αλγορίθμου και άρα το χρόνο εκτέλεσης του.

Θα πρέπει να σημειώσουμε ότι το Μετρικό – TSP, στην γενική του μορφή έχει αποδειχθεί ότι δεν έχει PTAS [44], ενώ το Ευκλείδειο TSP που είναι υποπερίπτωση του έχει PTAS.

### 6.2.4. Προσεγγιστικό Σχήμα Πλήρως Πολυωνυμικού Χρόνου (FPTAS)

Προσεγγιστικό Σχήμα Πλήρως Πολυωνυμικού Χρόνου (Fully Polynomial Time Approximation Scheme - FPTAS) ονομάζουμε ένα σχήμα PTAS το οποίο έχει πολυωνυμική πολυπλοκότητα ως προς το μέγεθος εισόδου του προβλήματος $n$ αλλά και ως προς το $1/_{\varepsilon}$

Από το ορισμό προκύπτει ότι FPTAS ⊂ PTAS. Η βασική διαφορά των σχημάτων FPTAS είναι ο χρόνος εκτέλεσης καθώς πλέον λόγω της πολυωνυμικής πολυπλοκότητας ως προς το $1/_{\varepsilon}$. Ζητώντας καλύτερη προσέγγιση ο αλγόριθμος αυξάνει την πολυπλοκότητα του πολυωνυμικά.

Παράδειγμα FPTAS αλγόριθμου έχουμε για το Πρόβλημα Σακιδίου (Knapsack) όπου η πολυπλοκότητα του είναι $O\left(n^3/_{\varepsilon}\right)$. Ο αλγόριθμος έχει πολυωνυμική τάξη ως προς το μέγεθος εισόδου του προβλήματος ($n$), αλλά και ως προς την ποιότητα της λύσης το $1/_{\varepsilon}$.

## 6.3. Μη Προσεγγισιμότητα του TSP

Έχοντας αποδείξει ότι το TSP είναι ένα NP-Complete πρόβλημα, γνωρίζουμε ότι δεν μπορούμε να το λύσουμε σε πολυωνυμικό χρόνο. Για το λόγο αυτό στραφήκαμε προς τις προσεγγιστικές μεθόδους. Τα δύο θεωρήματα που ακολουθούν, αποδεικνύουν ότι το TSP δεν είναι απλώς δύσκολο , είναι και μη προσεγγίσιμο.

Το 1976 οι Sahni και Gonzalez [13, pp. 555-565], απόδειξαν ότι δεν υπάρχει $\rho$ –προσεγγιστικός αλγόριθμος για το TSP για κάθε $\rho > 0$ ,εκτός αν P=NP.

Για την απόδειξη θεωρούμε ότι υπάρχει $\rho$-προσεγγιστικός αλγόριθμος για το TSP και θα δείξουμε ότι μπορεί να λύσει σε πολυωνυμικό χρόνο το πρόβλημα του Κύκλου του Hamilton, το οποίο είναι NP-Πλήρες και συνεπώς θα καταλήγουμε στο P=NP. Η απόδειξή γίνεται με εις άτοπο απαγωγή.

Έστω γράφημα G=(V,E) προβλήματος Κύκλου Hamilton. Κατασκευάζουμε ένα πλήρες γράφημα G′=(V,E′), όπου E′= {(i, j):i, j ∈ V και i ≠ j.}.

Επίσης ορίζουμε μια συνάρτηση κόστους ως εξής:c(i, j):=$\begin{cases} 1 \; \varepsilon\acute{\alpha}\nu \; (i,j) \in \mathrm{E} \\ \rho\, n+1 \;\; \varepsilon\acute{\alpha}\nu \; (i,j) \notin \mathrm{E} \end{cases}$

- Εάν υπάρχει κύκλος Hamilton στο G τότε επειδή κάθε ακμή που ανήκει στο G έχει κόστος 1 στο G′, το G′ έχει περιοδεία TSP κόστους $n$.
- Εάν δεν υπάρχει κύκλος Hamilton στο G τότε μια περιοδεία στο G′ θα πρέπει να χρησιμοποιεί τουλάχιστον μια ακμή που δεν υπάρχει στο E. Επομένως η περιοδεία θα έχει κόστος τουλάχιστον ($\rho\, n+1$ )+( $n-1$ )= $\rho\, n+n > \rho n$

Εφαρμόζοντας τον $\rho$-προσεγγιστικό αλγόριθμο στο G′ έχουμε πάλι δυο περιπτώσεις:

- Εάν υπάρχει κύκλος Hamilton στο G τότε πρέπει να μας δώσει μια περιοδεία στο G′ μήκους το πολύ $\rho\, n$ καθώς είναι $\rho$-προσεγγιστικός
- Εάν δεν υπάρχει κύκλος Hamilton στο G τότε θα μας δώσει περιοδεία μεγαλύτερη από $\rho\, n$

Η διαδικασία αυτή όμως λύνει το πρόβλημα του Κύκλου του Hamilton σε πολυωνυμικό χρόνο, γεγονός άτοπο εκτός αν P=NP

Οι Παπαδημητρίου και Γιαννακάκης [45] το 1993 παρουσιάσανε μια αναγωγή του προβλήματος Max 3-SAT-d σε Μετρικό – TSP, η οποία σε συνδυασμό με το PCP (probabilistically checkable proof) θεώρημα, των S. Arora, C. Lund, R. Motwani, M. Sudan και M. Szegedy [44], οδηγεί στην απόδειξη της μη ύπαρξης PTAS για το Μετρικό – TSP εκτός αν P=NP.

## 6.4. Φράγμα Held – Karp

Στην παράγραφο 6.1 ο λόγος προσέγγισης ενός προσεγγιστικού αλγορίθμου ορίζεται σαν κλάσμα με αριθμητή το κόστος της λύσης που αποδίδει ο αλγόριθμος και παρονομαστή το κόστος της άριστης λύσης. Όπως ήδη γνωρίζουμε το κόστος της άριστης λύσης είναι δύσκολο να υπολογιστεί, ενώ στην πλειονότητα των προβλημάτων παραμένει άγνωστο. Για να αντιμετωπίσουμε το πρόβλημα και να μπορέσουμε να μελετήσουμε την απόδοση των ευρετικών αλγορίθμων, συγκρίνουμε τις λύσεις τους με το Φράγμα Held – Karp το οποίο μπορούμε να υπολογίσουμε.

Το Φράγμα Held – Karp αποτελεί μια πολύ καλή προσέγγιση της άριστης λύσης ενός προβλήματος TSP. Στην χειρότερη περίπτωση δεν μπορεί να είναι μικρότερο από ${}^{2}/_{3} \cdot OPT$ [46, pp. 121-134] [47, pp. 281-285]. Παρουσιάστηκε το 1970 από τους Held και Karp [48] [49] και μπορεί να υπολογιστεί λύνοντας ένα πρόβλημα γραμμικού προγραμματισμού (linear programming relaxation of the TSP).

Για προβλήματα μερικών εκατοντάδων πόλεων μπορεί να υπολογιστεί ακριβώς, αλλά η πολυπλοκότητα του είναι εκθετική ως προς τον αριθμό των πόλεων n. Για μεγαλύτερο αριθμό πόλεων αναπτύχθηκε μια τεχνική που επιλύει διαδοχικά προβλήματα γραμμικού προγραμματισμού με περιορισμούς (restricted linear programs) από τον Reinelt [50] με την μέθοδο Simplex. Με την τεχνική αυτή υπολογίστηκε το φράγμα για προβλήματα μέχρι και 33,810 πόλεων [51].

Οι παραπάνω μέθοδοι δεν αποδίδουν για μεγαλύτερα προβλήματα και καταφεύγουμε και πάλι σε προσεγγιστικές μεθόδους Οι ίδιοι Held και Karp στην εργασία που παρουσίασαν το φράγμα [48] [49], έδωσαν και μια επαναληπτική μέθοδο προσέγγισης του. Οι Karger και Stein [52], το 1993 παρουσίασαν ένα προσεγγιστικό σχήμα με πολυπλοκότητα $O(n^4 log^6 n)$.

Αναπόδεικτη παραμένει η εικασία ότι το κόστος της άριστης λύσης ενός προβλήματος TSP, είναι το πολύ ${}^{4}/_{3}$ μεγαλύτερο του φράγματος Held – Karp. Η απόδειξη της εικασίας θα οδηγούσε σε έναν ${}^{4}/_{3}$–προσεγγιστικό αλγόριθμο για το Μετρικό–TSP [53, p. 519].

# 7. Κατασκευαστικοί Ευρετικοί Αλγόριθμοι του TSP.

Οι Κατασκευαστικοί Ευρετικοί αλγόριθμοι κατασκευάζουν σταδιακά μια περιοδεία που προσεγγίζει την άριστη. Συνήθως αποδίδουν καλές περιοδείες σε ικανοποιητικό υπολογιστικό χρόνο. Η ποιότητα της λύσης που αποδίδουν δεν είναι «εγγυημένη» και συνήθως εξαρτάται από τον αριθμό των πόλεων του προβλήματος.

## 7.1. Πλησιέστερος Γείτονας (Nearest Neighbord)

Ο απλούστερος ευρετικός αλγόριθμος είναι αυτός του Πλησιέστερου Γείτονα (Nearest Neighbor). Η κεντρική ιδέα πάνω στην οποία βασίζεται ο αλγόριθμος είναι σε κάθε βήμα να επιλέγουμε την πλησιέστερη πόλη – γείτονα.

Ο αλγόριθμος του Πλησιέστερου Γείτονα έχει τάξη πολυπλοκότητας $O(n^2)$ και λόγο προσέγγισης χειρότερης περίπτωσης ${}^{1}\!/_{2} \cdot (log_2(n) + 1)$ [54].Το μήκος των περιοδειών που μας δίνει, είναι κατά μέσο όρο 23% μεγαλύτερο του φράγματος Held-Karp. [55]

Τα βήματα του αλγορίθμου είναι:

Επιλέγουμε τη πόλη εκκίνησης

1. Βρίσκουμε την πλησιέστερη πόλη και προχωράμε σε αυτή
2. Αν έχουν μείνει πόλεις που δεν έχουμε επισκεφθεί πηγαίνουμε στο Βήμα 1. Αν όχι επιστρέφουμε στην πόλη εκκίνησης για να ολοκληρώσουμε την περιοδεία.

Θα εφαρμόσουμε τον αλγόριθμο στο προηγούμενο πρόβλημα TSP 4 πόλεων όπου C ο πίνακας κόστους των ακμών του αντίστοιχου γραφήματος. Έχουμε ήδη δει ότι μια άριστη περιοδεία είναι η 1 → 3 → 4 → 2 → 1 με κόστος 21

$$C=\begin{pmatrix} - & 2 & 9 & 10 \\ 1 & - & 6 & 4 \\ 15 & 7 & - & 8 \\ 6 & 3 & 12 & - \end{pmatrix}$$

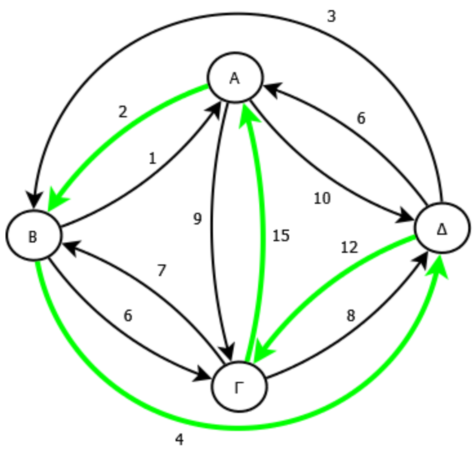
Επιλέγουμε την πόλη 1 για να ξεκινήσουμε.

Η κοντινότερη πόλη είναι η 2 με c12=2.

Από την πόλη 2 η κοντινότερη είναι η 1 με κόστος c21=1 αλλά απορρίπτεται καθώς την έχουμε ήδη επισκεφθεί. Η επόμενη κοντινότερη είναι η 4 με c24=4.

Από την 4 πηγαίνουμε στην πόλη 3 καθώς είναι η τελευταία που δεν έχουμε επισκεφθεί με c43=12.

Επομένως έχουμε την περιοδεία 1 → 2 → 4 →3 → 1 με κόστος 2+4+12+15=33.




**Εικόνα 9 Λύση με τον Αλγόριθμο του Πλησιέστερου Γείτονα**

Αν ξεκινάγαμε από την πόλη 2 θα είχαμε την περιοδεία 2→1→3→4→2 με κόστος 1+9+8+3=21 που ουσιαστικά είναι μετάθεση της γνωστής άριστης περιοδείας.

Αν ξεκινάγαμε από την πόλη 3 θα είχαμε την περιοδεία 3→2→1→4→3 με κόστος 7+1+10+12=30

Αν ξεκινάγαμε από την πόλη 4 θα είχαμε την περιοδεία 4→2→1→3→4 με κόστος 3+1+9+8=21 και πάλι μετάθεση της γνωστής άριστης περιοδείας.

## 7.2. Άπληστος Αλγόριθμος (Greedy)

Ο Άπληστος Αλγόριθμος κατασκευάζει μια περιοδεία επιλέγοντας την πλησιέστερη πόλη όπως και ο Αλγόριθμος του Πλησιέστερου Γείτονα, με τις παρακάτω όμως προϋποθέσεις :

- ότι δε σχηματίζει κύκλο με τις ακμές, που έχουν ήδη επιλεγεί, εκτός από την ακμή που κλείνει την περιοδεία του πωλητή (n ακμές)
- αν επιλεγεί, δε θα αποτελεί την τρίτη προσπίπτουσα ακμή στην ίδια κορυφή

Τα βήματα του αλγορίθμου είναι:

1. Κατασκευάζουμε πίνακα με ταξινομημένες ανά κόστος όλες τις ακμές σε αύξουσα σειρά.
2. Επιλέγουμε την ακμή με το μικρότερο κόστος, εφόσον ικανοποιεί τις παραπάνω προϋποθέσεις.
3. Αν έχουν μείνει πόλεις που δεν έχουμε επισκεφθεί συνεχίζουμε από το Βήμα 2. Αν όχι επιστρέφουμε στην πόλη εκκίνησης για να ολοκληρώσουμε την περιοδεία.

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(n^2 log_2(n))$ και λόγο προσέγγισης χειρότερης περίπτωσης ${}^{1}/_{2} \cdot (log_2(n) + 1)$ [56] [57]. Το μήκος των περιοδειών που μας δίνει, είναι κατά μέσο όρο 14% μεγαλύτερο του φράγματος Held-Karp. [55]

Θα εφαρμόσουμε τον αλγόριθμο στο προηγούμενο πρόβλημα TSP 4 πόλεων. Έχουμε ήδη δει ότι μια άριστη περιοδεία είναι η 1 → 3 → 4 → 2 → 1 με κόστος 21.

| Πίνακας Ακμών ανά Κόστος | | | ΒΗΜΑ 1 | | ΒΗΜΑ 2 | | ΒΗΜΑ 3 | | ΒΗΜΑ 4 | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | c21 | 1 | c21 | 1 | c21 | 1 | c21 | 1 | c21 | 1 |
| 2 | c12 | 2 | c12 | 2 | c12 | 2 | c12 | 2 | c12 | 2 |
| 3 | c42 | 3 | c42 | 3 | c42 | 3 | c42 | 3 | c42 | 3 |
| 4 | c24 | 4 | c24 | 4 | c24 | 4 | c24 | 4 | c24 | 4 |
| 5 | c23 | 6 | c23 | 6 | c23 | 6 | c23 | 6 | c23 | 6 |
| 6 | c41 | 6 | c41 | 6 | c41 | 6 | c41 | 6 | c41 | 6 |
| 7 | c32 | 7 | c32 | 7 | c32 | 7 | c32 | 7 | c32 | 7 |
| 8 | c34 | 8 | c34 | 8 | c34 | 8 | c34 | 8 | c34 | 8 |
| 9 | c13 | 9 | c13 | 9 | c13 | 9 | c13 | 9 | c13 | 9 |
| 10 | c14 | 10 | c14 | 10 | c14 | 10 | c14 | 10 | c14 | 10 |
| 11 | c43 | 12 | c43 | 12 | c43 | 12 | c43 | 12 | c43 | 12 |
| 12 | c31 | 15 | c31 | 15 | c31 | 15 | c31 | 15 | c31 | 15 |

**Εικόνα 10 Πίνακας Βημάτων Άπληστου Αλγόριθμου**

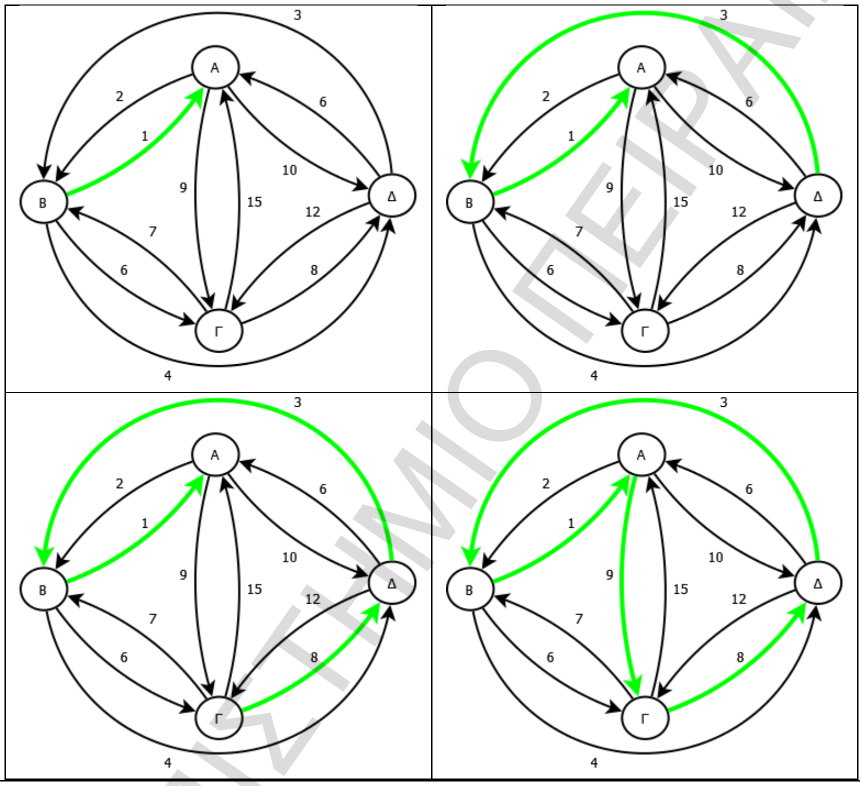
Στο πρώτο βήμα επιλέγεται η διαδρομή 2 → 1 ως φτηνότερη, απορρίπτονται οι 2 → 4 και 2 → 3 ως ακριβότερες καθώς επίσης και η διαδρομή 1 → 2 γιατί δημιουργεί πλέον κύκλο (2 → 1 → 2).

Στο δεύτερο βήμα επιλέγετε η 4→2 ως φτηνότερη, απορρίπτονται οι 4 → 1 και 4 → 3 ως ακριβότερες Η διαδρομή 1 → 4 δημιουργεί κύκλο (2 → 1 → 4 → 2). Η 3 → 2 απορρίπτεται γιατί τότε θα υπήρχε και Τρίτη προσπίπτουσα στην 2.

Στο τρίτο βήμα επιλέγετε 3→4 η ως φτηνότερη, απορρίπτεται η 3 → 1 ως ακριβότερη.

Στο τέταρτο βήμα επιλέγετε 1→3 η ως φτηνότερη και ολοκληρώνεται η περιοδεία.

Παρατηρούμε ότι η περιοδεία 2→1→3→4→2 που βρίσκουμε είναι μετάθεση της άριστης.




**Εικόνα 11 Στιγμιότυπα Βημάτων Άπληστου Αλγορίθμου**

## 7.3. Ευρετικοί Αλγόριθμοι Παρεμβολής

Οι Ευρετικοί Αλγόριθμοι Παρεμβολής ξεκινούν από μια περιοδεία ενός υποσυνόλου των πόλεων του προβλήματος και χρησιμοποιώντας διάφορα κριτήρια την επεκτείνουν μέχρι να περάσει από όλες τις πόλεις. Η αρχική υπο - περιοδεία συνήθως αποτελείται από τρεις τουλάχιστον πόλεις ή από το κυρτό περίβλημα (convex hull) που περιέχει όλες τις πόλεις, ενώ σε εκφυλισμένες περιπτώσεις μπορεί να είναι μια ακμή ή ένα σημείο- πόλη. Ανάλογα με τα κριτήρια παρεμβολής έχουμε και τους αντίστοιχους αλγόριθμους.

### 7.3.1. Κοντινότερη Εισαγωγή (Nearest Insertion)

Στον αλγόριθμο της κοντινότερης εισαγωγής, εισάγουμε στην την πόλη που βρίσκετε πιο κοντά σε κάποια πόλη της υπο-περιοδείας.

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(n^2)$ και λόγο προσέγγισης χειρότερης περίπτωσης $2-\frac{2}{n}$ [54] Το μήκος των περιοδειών που μας δίνει, είναι κατά μέσο όρο 27% μεγαλύτερο του φράγματος Held-Karp.

Τα βήματα είναι τα ακόλουθα:

1.Επιλέγουμε την πόλη που δεν συμμετέχει στην υπό - περιοδεία και έχει την μικρότερη απόσταση από τις πόλεις που συμμετέχουν στην υπό - περιοδεία.

2. Βρίσκουμε ακμή της υπό – περιοδείας τέτοια ώστε η εισαγωγή της πόλης, ανάμεσα στις δύο πόλεις της ακμής, να έχει ελάχιστο κόστος. Εισάγουμε την πόλη στην υπό – περιοδεία.

3. Επαναλαμβάνουμε από το βήμα 1 μέχρι να συμμετάσχουν όλες οι πόλεις στην περιοδεία.

Θα εφαρμόσουμε τον αλγόριθμο στο πρόβλημα TSP 4 πόλεων.

$$C=\begin{pmatrix} - & 2 & 9 & 10 \\ 1 & - & 6 & 4 \\ 15 & 7 & - & 8 \\ 6 & 3 & 12 & - \end{pmatrix}$$

Έστω η αρχική υπό – περιοδεία 2 → 1 → 2 με κόστος c21+c12=1+2=3 (εκφυλισμένη για τις ανάγκες του παραδείγματος).

Στο πρώτο βήμα παρατηρούμε ότι η κοντινότερη πόλη για την 1 είναι η 3 με κόστος c13=9. Η κοντινότερη πόλη στην 2 είναι η 4 με κόστος c24=4.

Επομένως επιλέγουμε να εισάγουμε την πόλη 4.

Στο δεύτερο βήμα θα επιλέξουμε ανάμεσα σε πια ακμή θα παρεμβάλουμε την νέα πόλη.

Για την ακμή 2→1 γίνεται 2→4→1→2 με κόστος c24+c41=4+6=10

Για την ακμή 1→2 γίνεται 2→1→4→2 με κόστος c14+c42=10+3=13

Επομένως επιλέγουμε την ακμή 2→1 και η υπό -περιοδεία γίνεται 2→4→1→2

Επαναλαμβάνουμε την διαδικασία και στο πρώτο βήμα προφανώς επιλέγουμε την πόλη 3 που έχει απομείνει.

Στο δεύτερο βήμα θα επιλέξουμε ανάμεσα σε πια ακμή θα παρεμβάλουμε την νέα πόλη.

Για την ακμή 2→4 γίνεται 2→3→4→1→2 με κόστος c23+c34=6+8=14

Για την ακμή 4→1 γίνεται 2→4→3→1→2 με κόστος c43+c31=12+15=27

Για την ακμή 1→2 γίνεται 2→4→1→3→2 με κόστος c13+c32=9+7=16

Επομένως επιλέγουμε την ακμή 2→4 και η υπό -περιοδεία γίνεται 2→3→4→1→2

Επειδή συμμετέχουν όλες οι πόλεις έχουμε πλέον μια περιοδεία κόστους c23+c34+c41+c12=6+8+6+2=22 η οποία είναι μια πολλή καλή προσέγγιση της άριστης που έχει κόστος 21

## 7.3.2. Εισαγωγή Ελαχίστου Κόστους (Cheapest Insertion)

Στον αλγόριθμο της Εισαγωγής Ελαχίστου Κόστους, εισάγουμε στην αρχική υπό – περιοδεία την πόλη που μπορεί να εισαχθεί με την ελάχιστη αύξηση του κόστους.

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(n^2 log_2(n))$ και λόγο προσέγγισης χειρότερης περίπτωσης $2 - \frac{2}{n}$ [54]. Το μήκος των περιοδειών που μας δίνει, είναι κατά μέσο όρο 22% μεγαλύτερο του φράγματος Held-Karp.

Τα βήματα είναι τα ακόλουθα:

1. Για κάθε πόλη εκτός υπό –περιοδείας βρίσκουμε ακμή της υπό – περιοδείας τέτοια ώστε η εισαγωγή της πόλης, ανάμεσα στις δύο πόλεις της ακμής, να έχει ελάχιστο κόστος Επιλέγουμε την πόλη με το ελάχιστο κόστος εισαγωγής και την εισάγουμε στη υπό – περιοδεία.
2. Επαναλαμβάνουμε μέχρι να συμμετάσχουν όλες οι πόλεις στην περιοδεία.

Ο αλγόριθμος επιβαρύνεται υπολογιστικά σε σχέση με τον αλγόριθμο της Κοντινότερης Απόστασης καθώς σε κάθε επανάληψη πρέπει να κάνουμε τους ελέγχους του δεύτερου βήματος για κάθε πόλη εκτός υπό – περιοδείας και όχι μόνο για μία.

## 7.3.1. Απομακρυσμένη Εισαγωγή (Farthest Insertion)

Στον αλγόριθμο της απομακρυσμένης εισαγωγής, εισάγουμε στην αρχική υπό – περιοδεία την πόλη που βρίσκετε πιο μακριά σε κάποια πόλη της υπο-περιοδείας.

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(n^2)$ , ενώ δεν έχει αποδειχθεί κάποιο συγκεκριμένο φράγμα για το μήκος της περιοδείας που αποδίδει, αν και στην πράξη αποδίδει καλύτερες περιοδείες από τους δύο προηγούμενους αλγορίθμους Εισαγωγής [54].

## 7.4. Clark & Wright

Οι Clarke και Wright [58] το 1964 παρουσίασαν έναν αλγόριθμο για προβλήματα δρομολόγησης οχημάτων, ο οποίος μπορεί να χρησιμοποιηθεί για την λύση του TSP, θεωρώντας το ως ειδική περίπτωση δρομολόγησης ενός μόνο οχήματος με απεριόριστη χωρητικότητα και τους πελάτες ως τις πόλεις που πρέπει να επισκεφθούμε.

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(n^2 log_2(n))$, αντίστοιχη του Άπληστου Αλγορίθμου. Οι Ong και Moore [57, pp. 273-277] απέδειξαν, για το συμμετρικό Μετρικό –TSP, ότι έχει λόγο προσέγγισης χειρότερης περίπτωσης $log_2(n) + 1$. Το μήκος των περιοδειών που μας δίνει, είναι κατά μέσο όρο 12% μεγαλύτερο του φράγματος Held-Karp

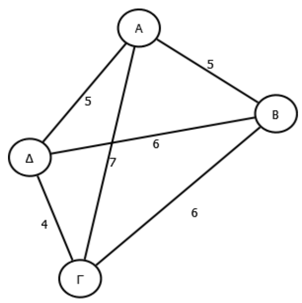
Τα βήματα του αλγορίθμου είναι ως εξής:

1. Στο πρώτο βήμα, επιλέγουμε την πόλη εκκίνησης και την αριθμούμε με το 1.
2. Στο δεύτερο βήμα υπολογίζουμε για τις υπόλοιπες n-1 πόλεις τις διαδρομές (1,i,1) με i=2,…,n δηλ τα $c_{i1} + c_{1i}$.
3. Στο τρίτο βήμα υπολογίζουμε την συνάρτηση κέρδους (savings), $s_{ij} = c_{i1} + c_{1j} - c_{ij}$ μετάβασης από την i πόλη στη j πόλη (i,j= 2,3,…,n (i ≠ j)) συγκρίνοντας την απευθείας διαδρομή με αυτήν που περνάει από την αρχική πόλη. Οι τιμές ταξινομούνται κατά φθίνουσα σειρά.
4. Κατά την φθίνουσα σειρά των $s_{ij}$ αντικαθιστούμε τις δύο ακμές i1 και j1 με την ακμή ij. Η αντικατάσταση δεν γίνεται όταν οι κορυφές i και j ανήκουν ήδη στην περιοδεία που δημιουργούμε.
5. Επαναλαμβάνουμε το τέταρτο βήμα μέχρι όλες οι πόλεις να συμμετέχουν στην περιοδεία.

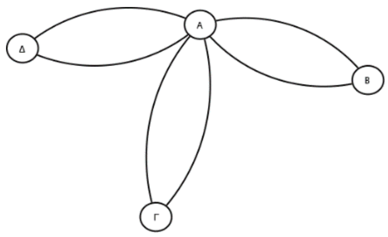
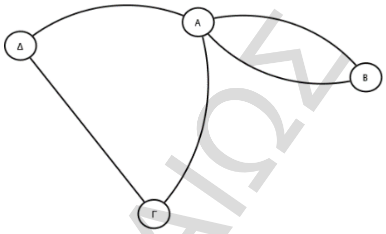
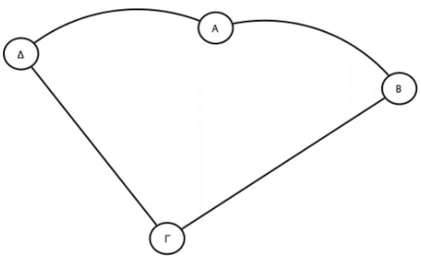
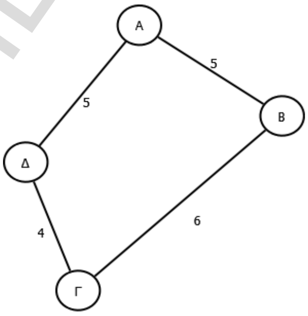
Θα εφαρμόσουμε τον αλγόριθμο σε συμμετρικό πρόβλημα TSP 4-πόλεων, όπου C ο πίνακας κόστους των ακμών του αντίστοιχου γραφήματος, για το οποίο ισχύει η τριγωνική ανισότητα.

1. Επιλέγουμε σαν πόλη εκκίνησης την Α i=1
2. Υπολογίζουμε όλα τα $c_{i1} + c_{1j}$

   $c_{21} + c_{12}=10$, $c_{31} + c_{13}=14$, $c_{41} + c_{14}=10$
3. Υπολογίζουμε την συνάρτηση κέρδους (savings), $s_{ij} = c_{i1} + c_{1j} - c_{ij}$ και χρησιμοποιούμε τον πίνακα C για την αποθήκευση του καθώς λόγω συμμετρικού TSP είναι άνω τριγωνικός. Επίσης $s_{ij} = s_{ji}$

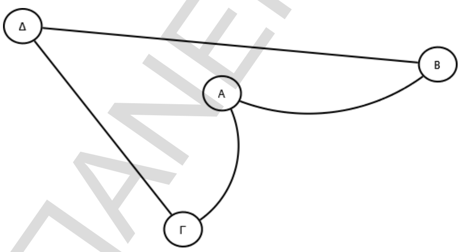
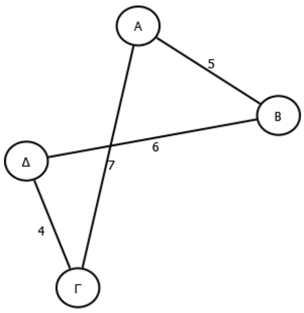
| ΠΙΝΑΚΑΣ ΚΟΣΤΟΥΣ | ΠΙΝΑΚΑΣ ΚΟΣΤΟΥΣ ΜΕ ΣΥΝΑΡΤΗΣΗ ΚΕΡΔΟΥΣ | ΓΡΑΦΗΜΑ ΠΡΟΒΛΗΜΑΤΟΣ |
|---|---|---|
| $C=\begin{pmatrix} - & 5 & 7 & 5 \\ 5 & - & 6 & 6 \\ 7 & 6 & - & 4 \\ 5 & 6 & 4 & - \end{pmatrix}$ | $C=\begin{pmatrix} - & 5 & 7 & 5 \\ 5 & - & 6 & 6 \\ 7 & 4 & - & 4 \\ 5 & 4 & 10 & - \end{pmatrix}$ |  |

4. Ταξινομούμε κατά φθίνουσα σειρά τα $s_{ij}$: $s_{34}=10$, $s_{23}=4$, $s_{24}=4$

<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>5. Αρχική περιοδεία.</td>
<td>6. Προσθέτουμε την ακμή 34=ΓΔ και αφαιρούμε της 31=ΑΓ και 41=ΑΔ</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>7. Προσθέτουμε την ακμή 23=ΒΓ και αφαιρούμε της 21=ΑΒ και 31=ΑΓ</td>
<td>Η περιοδεία έχει κόστος 20 και είναι η άριστη</td>
</tr>
</table>

Επειδή $s_{23}=4=s_{24}$ εξετάζουμε την εναλλακτική:

4. Ταξινομούμε κατά φθίνουσα σειρά τα $s_{ij}$: $s_{34}=10$, $s_{24}=4$, $s_{23}=4$
5. Ίδιο με προηγουμένως
6. Ίδιο με προηγουμένως

<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>7. Προσθέτουμε την ακμή 24=ΒΔ και αφαιρούμε της 21=ΑΒ και 41=ΑΔ</td>
<td>Η περιοδεία έχει κόστος 22</td>
</tr>
</table>

# 8. Βελτιωτικοί Ευρετικοί Αλγόριθμοι του TSP

Οι Κατασκευαστικοί αλγόριθμοι που εξετάσαμε στο προηγούμενο κεφαλαίο κατασκευάζουν σταδιακά μια περιοδεία που προσεγγίζει την άριστη. Οι Βελτιωτικοί (Improvement) Ευρετικοί Αλγόριθμοι που θα εξετάσουμε σε αυτό το κεφάλαιο ξεκινούν με δεδομένη μια περιοδεία την οποία βελτιώνουν, με τη χρήση τεχνικών βελτιστοποίησης και για αυτό το λόγο η πολυπλοκότητα τους εξαρτάται από τον κατασκευαστικό αλγόριθμο της αρχικής περιοδείας.

## 8.1. 2-opt

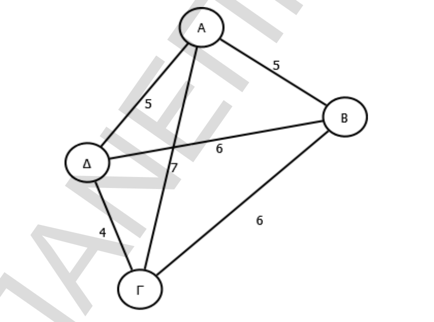
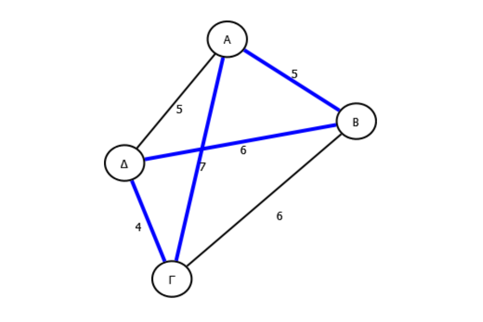
Ο απλούστερος βελτιωτικός αλγόριθμος παρουσιάστηκε το 1958 από τον Croes [6] και έχει σαν βασική ιδέα την παρατήρηση ότι συνήθως δύο ακμές που διασταυρώνονται σε μια περιοδεία μπορούν να αντικατασταθούν από δύο άλλες με μικρότερο κόστος.

Ο αλγόριθμος 2-opt εξετάζει όλες τις ακμές μιας περιοδείας, ανά δύο χωρίς κοινά σημεία, και ελέγχει αν η αντικατάστασή κάθε ζεύγους ακμών, από δύο άλλες, που συνδέουν τις αντίστοιχές κορυφές εναλλάξ, έχει μικρότερο κόστος. Με την ολοκλήρωση του αλγορίθμου ονομάζουμε τη νέα περιοδεία 2-optimal.

Αναλυτικότερα σε κάθε βήμα βελτίωσης επιλέγουμε δύο ακμές της περιοδείας u1→u2 και v1→v2 , με u1,u2,v1,v2 μοναδικές και διακριτές κορυφές , με την σειρά που εμφανίζονται στην περιοδεία και τις αντικαθιστούμε με τις ακμές u1→v1 και u2→v2 και εξετάζουμε αν αποδίδει καλύτερου κόστους περιοδεία.

Ο αλγόριθμος έχει τάξη πολυπλοκότητας O($n^2$). Οι περιοδείες που αποδίδει είναι κατά μέσο όρο 6,9% πάνω από το φράγμα Held-Karp [55] ενώ δεν είναι ποτέ μεγαλύτερες από $4 \cdot \sqrt{n}$ OPT [59, p. 570].

Θα εφαρμόσουμε τον αλγόριθμο στο συμμετρικό πρόβλημα TSP της προηγούμενής παραγράφου. Θεωρούμε αρχική περιοδεία η οποία δεν είναι άριστη και μπορεί να βελτιωθεί (βλ. αλγόριθμο Double MST της παραγράφου 9.1).

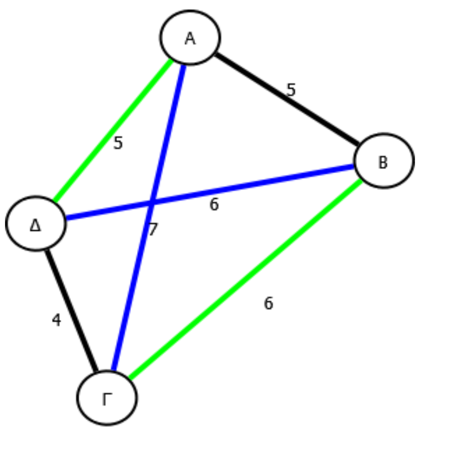
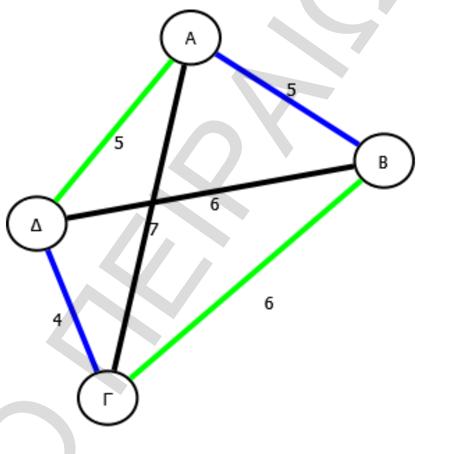
| ΓΡΑΦΗΜΑ ΠΡΟΒΛΗΜΑΤΟΣ | ΑΡΧΙΚΗ ΠΕΡΙΟΔΕΙΑ S (Double MST) |
|---|---|
|  |  |

Η αρχική περιοδεία S={A→B →Δ→Γ→A} έχει κόστος 22.

Έστω το σύνολο των ακμών του περιοδείας ανά δύο χωρίς κοινά σημεία: Z={(AB,ΓΔ), (AΓ,BΔ)}.

Για το ζεύγος (ΑΒ,ΔΓ) ελέγχουμε το κόστος αντικατάστασης του από το ζεύγος (ΑΔ,ΒΓ). Παρατηρούμε ότι cΑΒ+cΔΓ=5+4=9 ενώ cΑΔ+cΒΓ=5+6=11.

Η νέα περιοδεία {Α→Δ→Β→Γ→Α} έχει κόστος 24 μεγαλύτερο της αρχικής και συνεπώς η αντικατάσταση απορρίπτεται.

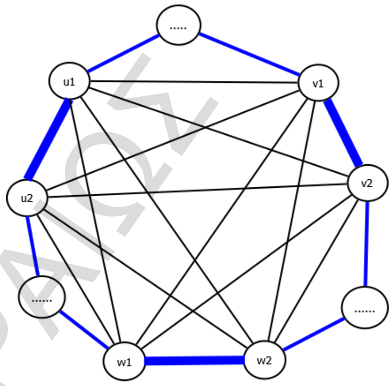
| 1$^{η}$ Αντικατάσταση | 2$^{η}$ Αντικατάσταση |
|---|---|
|  |  |

Για το ζεύγος (ΑΓ,ΒΔ) ελέγχουμε το κόστος αντικατάστασης του από το ζεύγος (ΑΔ,ΒΓ). Παρατηρούμε ότι cΑΓ+cΒΔ=7+6=13 ενώ cΑΔ+cΒΓ=5+6=11.

Η νέα περιοδεία {Α→ Β→Γ→Δ→Α} έχει κόστος 20 μικρότερο της αρχικής και συνεπώς η αντικατάσταση είναι αποδεκτή. Η συγκεκριμένη διαδρομή είναι και η άριστη.

Το 1990 ο Bentley παρουσίασε την «fast 2-opt» [60, pp. 387-411] μέθοδο χρησιμοποιώντας k-d δέντρα (k-d trees) για να βρει πιθανά ζευγάρια για βελτίωση. Η μέθοδος του δεν συγκρίνει ακμές, αλλά ερευνά τις κορυφές βρίσκοντας πλησιέστερους γείτονες που μπορεί να χρησιμοποιήσει για βελτίωση.

## 8.2. 3-opt

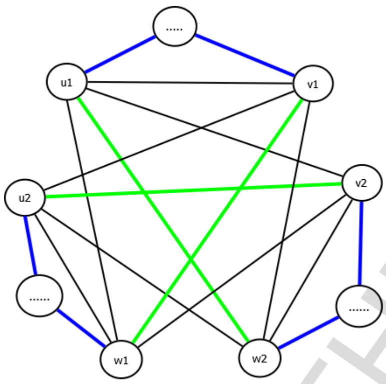
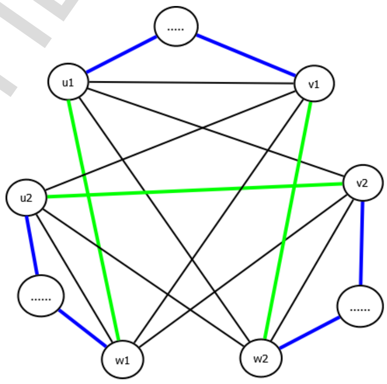
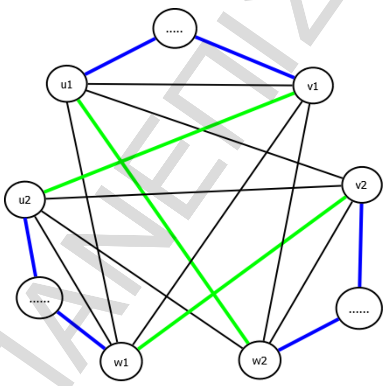
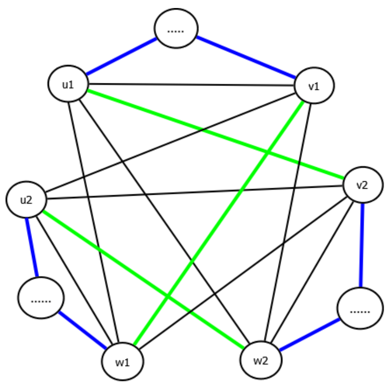



Εικόνα 12 Παράδειγμα 3-opt

Ο αλγόριθμος 3-opt προτάθηκε από τους Bock και Lin [61] και σε αντιστοιχία με τον 2-opt εξετάζει όλες τις ακμές μιας περιοδείας, ανά τρείς χωρίς κοινά σημεία, και ελέγχει αν η αντικατάστασή τους από τρείς άλλες, που συνδέουν τις αντίστοιχές κορυφές εναλλάξ, έχει μικρότερο κόστος.

Στη συνέχεια παρουσιάζουμε γράφημα προβλήματος συμμετρικού TSP όπου επιλέγουμε τις ακμές u1→u2, v1→v2 και w1→w2 της αρχικής περιοδείας u1→u2→…→w1→w2→…→v2→v1→…→u1 για παράδειγμα 3-opt αντικατάστασης.

Στο γράφημα για λόγους απλοποίησης εμφανίζονται μόνο οι ακμές που συνδέουν ανά δύο τα u1,u2,v1,v2,w1,w2. Οι δυνατές 3-opt αντικαταστάσεις είναι:

| | |
|---|---|
|  |  |
| u1→w2→…→v2→u2→…→w1→v1→…→u1 | u1→w1→…→u2→v2→…→w2→v1→…→u1 |
|  |  |
| u1→w2→…→v2→w1→…→u2→v1→…→u1 | u1→v2→…→w2→u2→…→w1→v1→…→u1 |

Επισημαίνουμε ότι δεν μπορούμε να συνδέσουμε διαδοχικές στην περιοδεία κορυφές πχ u2→w1 καθώς κλείνει με αυτό τον τρόπο κύκλος. Κάθε μία αντικατάσταση

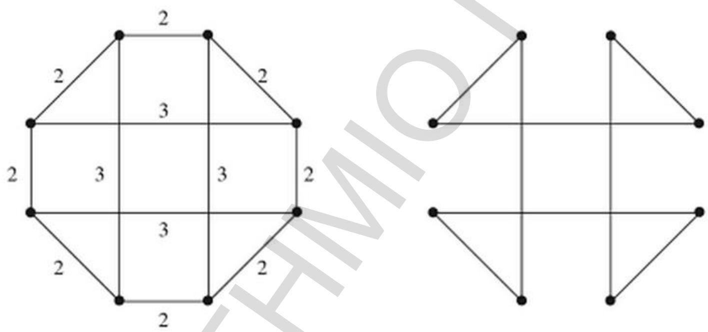
3-opt μπορούμε να την δούμε και σαν δύο διαδοχικές 2-opt αντικαταστάσεις και για αυτό μια 3-optimal περιοδεία είναι και 2-optimal.

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(n^3)$ και αποδίδει περιοδείες κατά μέσο όρο 4,6% πάνω από το φράγμα Held-Karp [55].

## 8.3. k-opt

Η γενίκευση των δύο προηγούμενων αλγορίθμων είναι η μέθοδος k-opt όπου επιλέγουμε να αντικαταστήσουμε k>3 ακμές του προβλήματος ερευνώντας για μικρότερου κόστους περιοδείες. Στην γενική περίπτωση ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(n^K)$. Επίσης οι περιοδείες που αποδίδει είναι δυνατόν να είναι μεγαλύτερες ακόμα και $\frac{1}{4}n^{\frac{1}{2k}}$ φορές της άριστης [59, p. 570] .

Αν και θεωρητικά η αύξηση του k θα έπρεπε να αποδώσει καλύτερες περιοδείες, η μεγάλη πολυπλοκότητα που επιφέρει , συνδυασμένη με την μη ύπαρξη κάποιου ποιοτικού φράγματος όπως είδαμε, οδηγούν συνήθως στην χρήση των μεθόδων 2-opt και 3-opt.




**Εικόνα 13 Παράδειγμα 4-opt [59]**

Στην Εικόνα 13 έχουμε ένα παράδειγμα 4-opt αντικατάστασης. Η συγκεκριμένη αντικατάσταση καλείται "the crossing bridges" και δεν μπορεί να κατασκευαστεί με διαδοχικές 2-opt ή 3-opt αντικαταστάσεις.

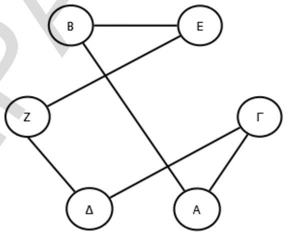
## 8.4. Lin-Kernighan (v-opt)

Οι Lin και Kernigham παρουσίασαν έναν αλγόριθμο βελτιστοποίησης το 1972 [10] βασισμένο στην τεχνική k-opt αλλά με κεντρική ιδέα τη δυναμική επιλογή αριθμού ακμών για αντικατάσταση. Ο αλγόριθμος δεν έχει σταθερό κ, όπως ο k-opt, αλλά μεταβλητό (variable , v-opt), το οποίο και αποφασίζει σε κάθε βήμα.

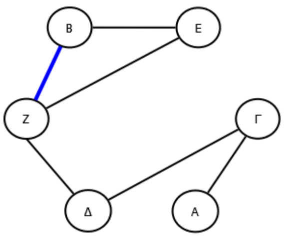
Στην γενική περίπτωση ο αλγόριθμος έχει τάξη πολυπλοκότητας O(n$^2$) και αποδίδει περιοδείες κατά μέσο όρο 1-2% μεγαλύτερες της άριστης [55, p. 4] [62, p. 123].

Τα βήματα του αλγορίθμου είναι τα εξής:

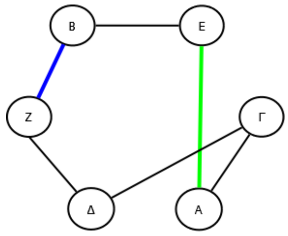
1. Επιλεγούμε την αρχική περιοδεία. (Α→Β→Ε→Ζ→Δ→Γ→Α)



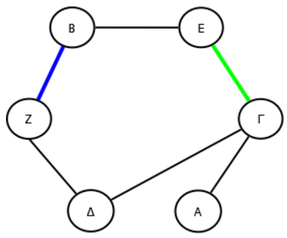

2. Θέτουμε $G_0^* = 0$.
   2.1. Επιλέγουμε την αρχική πόλη Α της περιοδείας μας και μία ακμή για αντικατάσταση, από τις δύο της περιοδείας που προσπίπτουν(Α→Β,Γ→Α) σε αυτή, πχ Α→Β.
   2.2. Θέτουμε τον δείκτη p=1.
3. Από την δεύτερη πόλη Β επιλέγουμε μια ακμή που δεν ανήκει στην αρχική περιοδεία πχ Β→Ζ τέτοια ώστε $g_1 = c_{AB} - c_{BZ} > 0$. Η ακμή επιλέγετε με κριτήριο τη μεγιστοποίηση του $g_1$ δηλαδή η Β→Ζ έχει το μικρότερο κόστος από όλες τις προσπίπτουσες ακμές στην Β που δεν συμμετέχουν στην αρχική περιοδεία.




4. Στο προηγούμενο βήμα έχουμε επιλέξει να αφαιρέσουμε την Α→Β από την περιοδεία και να προσθέσουμε την Β→Ζ, επομένως η επόμενη ακμή που θα αφαιρεθεί πρέπει να είναι η μία από τις δύο ακμές που προσπίπτουν στην Ζ και συμμετέχουν στην αρχική περιοδεία, της οποίας η αφαίρεση αφήνει τις πόλεις συνδεδεμένες.
   4.1. Στο παράδειγμα μας η επιλέγεται να αφαιρεθεί η Ζ→Ε καθώς η αφαίρεση της Ζ→Δ αφήνει τις πόλεις ασύνδετες.
   4.2. Για να έχουμε ξανά περιοδεία προσθέτουμε την Ε→Α
   4.3. Θέτουμε $G_1^* = g_1 + c_{ZE} - c_{EA}$
   4.4. Αυξάνουμε το p κατά 1.




5. Η ακμή Ε→Α που προσθέσαμε για να έχουμε ξανά περιοδεία δεν είναι κατ' ανάγκην αυτή που θα επιλέξουμε. Πρέπει να βρούμε την πόλη x που μεγιστοποιεί το $g_p = c_{ZE} - c_{Ex}$. Έστω ότι x=Γ και η ακμή που προσθέτουμε είναι η Ε→Γ
   5.1. Υπολογίζουμε το $G_p = \sum_{s=1}^{p} g_s$ και το $G_p^* = G_p + c_{ZE} - c_{E\Gamma}$
   5.2. Θέτουμε $G^* = \max\{ G_1^* \, G_2^* \dots G_p^*\}$
   5.3. Αυξάνουμε το p κατά 1.



5.4. Επαναλαμβάνουμε το βήμα 5 έως ότου:

- Δεν υπάρχει άλλη εφικτή αντικατάσταση.
- Έχουμε μια νέα περιοδεία. (όπως είδαμε στο βήμα 4.2)
- $G_p \leq 0$
- $G_p \leq G^*$

Εφόσον κάποιο από τα παραπάνω κριτήρια ισχύσει έχουμε την νέα περιοδεία η οποία αντιστοιχεί στην ακολουθία $\{\ G_1^*\ G_2^* \ldots G_p^*\}$.

Ολόκληρη η διαδικασία επαναλαμβάνεται ώστε κάθε πόλη να χρησιμοποιηθεί σαν αρχική πόλη και να μην υπάρχει άλλη εφικτή αντικατάσταση.

Για να πετύχουν έναν αποδοτικό αλγόριθμο οι Lin και Kernighan έθεσαν τα εξής κριτήρια για την επιλογή των ακμών προς αντικατάσταση:

- Το κριτήριο της διαδοχικής αντικατάστασης: Η ακμή $x_i$ και η $y_i$ που την αντικαθιστά, πρέπει να έχουν μία κοινή κορυφή και το ίδιο και η $y_i$ με την $x_{i+1}$. Η συνθήκη αυτή είναι αναγκαία, αλλά όχι ικανή, ώστε μετά τις αντικαταστάσεις να έχουμε πάλι μια περιοδεία. Γενικά η βελτίωση μιας περιοδείας μπορεί να γίνει με μια σειρά διαδοχικών αντικαταστάσεων 2-opt και 3-opt. Είδαμε όμως στην Εικόνα 13 παράδειγμα αντικατάστασης που δεν μπορεί να γίνει με διαδοχικές 2-opt και 3-opt αντικαταστάσεις.
- Το κριτήριο της εφικτότητας: Απαιτούμε η ακμή $x_i = t_{2i-1} \rightarrow t_{2i}$ να επιλεγεί κατά τέτοιο τρόπο ώστε, αν η κορυφή $t_{2i}$ συνδέεται με την $t_1$, την πρώτη κορυφή στην αρχική περιοδεία, να σχηματίζεται περιοδεία. Το κριτήριο αυτό εξασφαλίζει την δημιουργία περιοδείας και ουσιαστικά επιτρέπει να γίνουν μόνο k-opt αντικαταστάσεις που είναι σειρά διαδοχικών 2-opt και 3-opt.
- Το κριτήριο του κέρδους: Η ακμή $y_i$ επιλέγεται με κριτήριο το συνολικό κέρδος $G_p$ να παραμένει θετικό. Το κριτήριο αυτό θεωρείται το σημαντικότερο για την απόδοση του αλγορίθμου.
- Το κριτήριο του διαχωρισμού των ακμών: Το σύνολο των ακμών που φεύγουν από την αρχική περιοδεία με το σύνολο των ακμών που μπαίνουν στην περιοδεία πρέπει να είναι ξένα μεταξύ τους.
- Το κριτήριο του υποψήφιου συνόλου: Οι Lin και Kernighan πρότειναν να περιοριστεί η αναζήτηση για υποψήφια προς ένταξη ακμή σε αυτές των πέντε κοντινότερων γειτόνων (nearest neighborhood ) της κάθε πόλης.

### 8.4.1. Lin-Kernighan (LKH-1)

Ο Helgaun το 2000 [18] παρουσίασε μια τροποποίηση του αλγορίθμου των Lin-Kernighan, αναθεωρώντας τα κριτήρια που είχαν θέσει. Συνοπτικά παρουσιάζουμε τις αλλαγές:

- Το κριτήριο της διαδοχικής αντικατάστασης έγινε πιο «χαλαρό» ώστε όταν δεν μπορεί να γίνει πλέον άλλη αντικατάσταση από διαδοχικές 2-opt και 3-opt αντικαταστάσεις εξετάζονται και μη διαδοχικές 4-opt και 5-opt.
- Το κριτήριο της εφικτότητας: Η βασική αντικατάσταση είναι πλέον η 5-opt και όχι οι διαδοχικές 2-opt και 3-opt. Με αυτή την αλλαγή ο αλγόριθμος αποδίδει καλύτερης ποιότητας λύσεις με μεγαλύτερο υπολογιστικό κόστος όμως.

- Το κριτήριο του κέρδους παραμένει ως έχει.
- Το κριτήριο του διαχωρισμού των ακμών δεν απαιτείται να ικανοποιείται πλέον. Για να αποφύγουμε όμως ατέρμονες αντικαταστάσεις ακολουθείται ο εξής κανόνας: Η τελευταία ακμή που αφαιρείται από μια 5-opt αντικατάσταση δεν μπορεί να έχει προστεθεί με μια προηγούμενη.
- Το κριτήριο του υποψήφιου συνόλου άλλαξε καθώς έγινε εισαγωγή του μέτρου «α-measure». Δοθέντος ενός ελάχιστου 1-tree του γραφήματος, για κάθε ακμή προκύπτει η τιμή «α-value» που είναι η αύξηση του κόστους του ελαχίστου 1-tree όταν απαιτείται να την περιέχει.

Ο Helgaun το 2009 [62]επέκτεινε και γενίκευσε τον LKH-1 παρουσιάζοντας τον LKH-2 στον οποίο επιτρέπονται όλες οι αντικαταστάσεις που μπορούν να γίνουν με διαδοχικές k-opt όπου 2≤κ≤n καθώς και οι μη διαδοχικές. Στα συμπεράσματα του ο Helgaun καταλήγει ότι ο αλγόριθμος επιβαρύνεται σημαντικά υπολογιστικά όταν επιλέγει k>4 σημειώνοντας τελικά ότι οι  k-opt αντικαταστάσεις επιλέγονται σε πολύ συγκεκριμένες περιπτώσεις.

# 9. Προσεγγιστικοί Αλγόριθμοι Σταθερού Λόγω Προσέγγισης

Για το Μετρικό - TSP έχουν προταθεί δύο προσεγγιστικοί $\rho$-προσεγγιστικοί αλγόριθμοι τους οποίους παρουσιάζουμε στη συνέχεια. Και οι δύο χρησιμοποιούν ως σημείο εκκίνησης ένα ελάχιστο γεννητικό δέντρο (MST) του προβλήματος για αυτό και αναφέρονται στη βιβλιογραφία και ως MST αλγόριθμοι για το TSP.

## 9.1. Αλγόριθμος Διπλού Ελάχιστου Γεννητικού Δέντρου (Double MST)

Ο Αλγόριθμος Διπλού Ελάχιστου Γεννητικού Δέντρου (Double MST) χρησιμοποιεί σαν αφετηρία της αναζήτησης του για μια προσεγγιστική λύση του TSP ένα ελάχιστο γεννητικό δέντρο του γραφήματος που αντιστοιχεί στο πρόβλημα.

Το κόστος ενός ελάχιστου γεννητικού δέντρου είναι κάτω φράγμα του κόστους της άριστης περιοδείας. Η απόδειξη είναι στοιχειώδης καθώς αν αφαιρέσουμε από το γράφημα μιας άριστης περιοδείας, οποιαδήποτε ακμή, έχουμε ένα ελάχιστο γεννητικό δέντρο. Συνεπώς ισχύει ότι cost(MST)≤OPT.

Αν διπλασιάσουμε τις ακμές του ελάχιστου γεννητικού δέντρου τότε δημιουργούμε ένα γράφημα το οποίο έχει ένα κύκλο Euler. Επομένως cost(Euler)=2·cost(MST)

Για να βρούμε έναν κύκλο Hamilton, και μια περιοδεία για το πρόβλημα μας, η ιδέα είναι να αντικαταστήσουμε δύο ακμές του MST, που περνάνε από πόλη A που έχουμε ήδη επισκεφθεί (μια που έρχεται X→A και μια που φεύγει A→Y) με την ακμή X→Y. Λόγω της τριγωνικής ανισότητας η παράκαμψη είναι μικρότερη του κόστους των δύο ή περισσότερων ακμών του κύκλου του Euler που αντικαθιστά:

cost(S)≤ cost(Euler)

Επομένως ο αλγόριθμος θα μας δώσει μια περιοδεία που στην χειρότερή περίπτωση θα έχει κόστος 2 φορές αυτό της άριστης ( worst case ratio 2).

cost(S)≤ cost(Euler) ≤2·cost(MST) )≤2·OPT

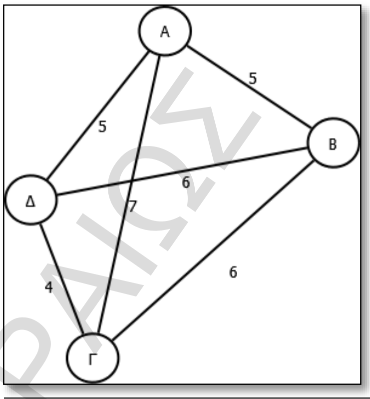
Τα βήματα του αλγορίθμου είναι τα ακόλουθα:

1. Κατασκευάζουμε ένα ελάχιστο γεννητικό δέντρο με κορυφές τις πόλεις του προβλήματος.
2. Διπλασιάζουμε όλες τις ακμές του ελάχιστου γεννητικού δέντρου και κατασκευάζουμε ένα κύκλο του Euler.
3. Διασχίζουμε τον κύκλο του Euler. Κάθε σημείο που συναντάμε για πρώτη φορά συμμετέχει στην λύση, ενώ όταν συναντάμε σημείο που έχουμε ήδη επισκεφθεί το προσπερνάμε και συνεχίζουμε.
4. Ο αλγόριθμος ολοκληρώνεται όταν συμμετέχουν όλες οι κορφές στην νέα περιοδεία.

Ο αλγόριθμος έχει τάξη πολυπλοκότητας $O(n^2 log_2(n))$.

Θα εφαρμόσουμε τον αλγόριθμο στο επόμενο πρόβλημα TSP 4 πόλεων όπου C ο πίνακας κόστους των ακμών του αντίστοιχου γραφήματος, για το οποίο ισχύει η τριγωνική ανισότητα.

$$C=\begin{pmatrix} - & 5 & 7 & 5 \\ 5 & - & 6 & 6 \\ 7 & 6 & - & 4 \\ 5 & 6 & 4 & - \end{pmatrix}$$




**Εικόνα 14 Γράφημα προβλήματος Αλγορίθμου Double MST**

Στο πρώτο βήμα πρέπει να βρούμε το ελάχιστο γεννητικό δέντρο του γραφήματος με τον αλγόριθμο του Kruskal

1. Ταξινομούμε τις ακμές του γραφήματος βάσει κόστους κατά αύξουσα σειρά.

| | | |
|---|---|---|
| 1 | cΓΔ | 4 |
| 2 | cAB | 5 |
| 3 | cAΔ | 5 |
| 4 | cBΓ | 6 |
| 5 | cBΔ | 6 |
| 6 | cAΓ | 7 |

2. Κατασκευάζουμε ένα υπό - γράφημα S έτσι ώστε:

   Για κάθε ακμή e στον πίνακα μας εξετάζουμε αν οι κορυφές της συνδέονται στο S και αν όχι την προσθέτουμε στο S.

   2.1. S={∅}. Επιλέγουμε την επόμενη ακμή Γ→Δ.
   Οι κορυφές Γ και Δ δεν συνδέονται στο S και προσθέτουμε την Γ→Δ στο S.

   2.2. S={Γ→Δ}. Επιλέγουμε την επόμενη ακμή AB.
   Οι κορυφές A και B δεν συνδέονται στο S και προσθέτουμε την A→B στο S.

   2.3. S={Γ→Δ,A→B}. Επιλέγουμε την επόμενη ακμή A→Δ.
   Οι κορυφές A και Δ δεν συνδέονται στο S και προσθέτουμε την A→Δ στο S.

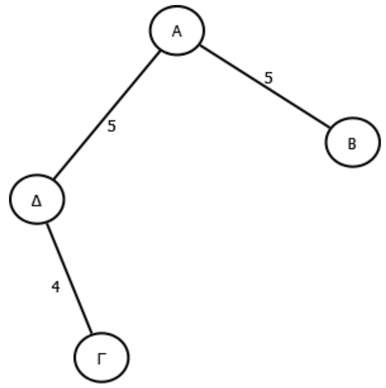
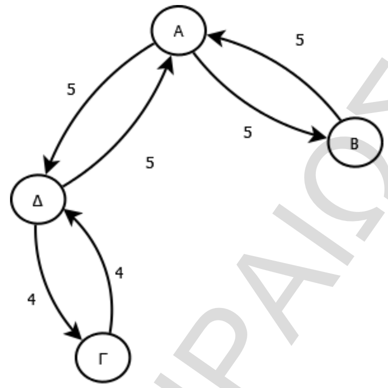
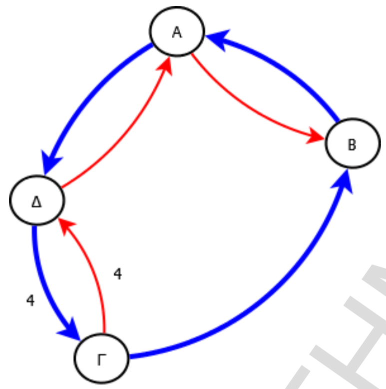
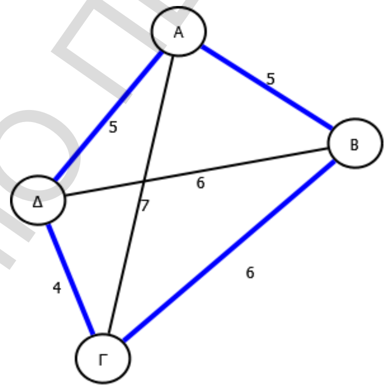
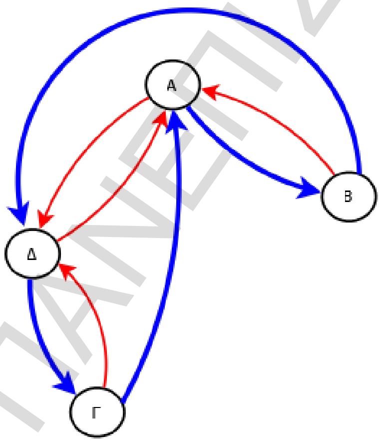
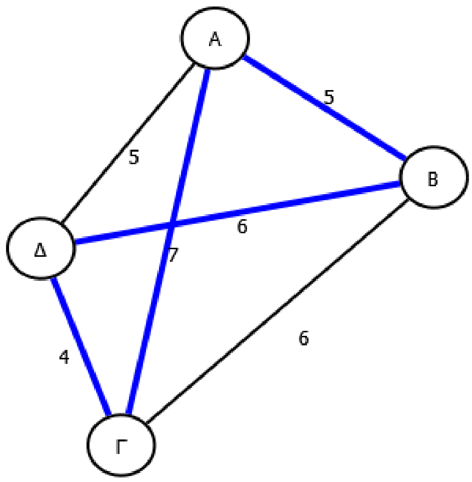
   2.4. S={Γ→Δ,A→B,A→Δ}. Επιλέγουμε την επόμενη ακμή B→Γ.
   Οι κορυφές B και Γ συνδέονται στο S με B→A→Δ→Γ καθώς Γ→Δ,A→B,A→Δ ∈ S.

   2.5. S={Γ→Δ,A→B,A→Δ}. Επιλέγουμε την επόμενη ακμή B→Δ.
   Οι κορυφές B και Δ συνδέονται στο S με B→A→Δ καθώς A→B,A→Δ ∈ S

   2.6. S={Γ→Δ,A→B,A→Δ}. Επιλέγουμε την επόμενη ακμή A→Γ.
   Οι κορυφές A και Γ συνδέονται στο S με A→Δ→Γ καθώς Γ→Δ,A→Δ ∈ S

   Επομένως το S={Γ→Δ,A→B,A→Δ} είναι ένα ελάχιστο γεννητικό δέντρο.

Στο δεύτερο βήμα διπλασιάζουμε τις ακμές του S και κατασκευάζουμε τον κύκλο Euler.

| ΒΗΜΑ 1 | ΒΗΜΑ 2 |
|---|---|
|  |  |
| ΒΗΜΑ 3 | ΒΗΜΑ 4 |
|  |  |
| ΒΗΜΑ 3 (εναλλακτικό) | ΒΗΜΑ 4(εναλλακτικό) |
|  |  |

**Εικόνα 15 Πίνακας Βημάτων Αλγορίθμου Double MST**

Στο τρίτο βήμα διασχίζουμε τον κύκλο του Euler και δημιουργούμε την περιοδεία C. Ξεκινάμε από το A C={A}

Επόμενο σημείο το Δ που δεν συμμετέχει στο C και το προσθέτουμε. C={A,Δ}

Επόμενο σημείο το Γ που δεν συμμετέχει στο C και το προσθέτουμε. C={A,Δ,Γ}

Επόμενο σημείο το Δ που συμμετέχει στο C. Παραμένει C={A,Δ,Γ}

Επόμενο σημείο το A που συμμετέχει στο C. Παραμένει C={A,Δ,Γ}

Επόμενο σημείο το B που δεν συμμετέχει στο C και το προσθέτουμε. C={A,Δ,Γ,B}

Έχοντας ολοκληρώσει τον κύκλο του Euler ο αλγόριθμος μας δίνει την περιοδεία C: A→Δ →Γ→B→A η οποία έχει κόστος 20 και είναι και η άριστη.

Αν επιλέξουμε να κινηθούμε προς το B έχουμε μια δεύτερη λύση.

Ξεκινάμε από το A C={A}

Επόμενο σημείο το B που δεν συμμετέχει στο C και το προσθέτουμε. C={A,B}

Επόμενο σημείο το A που συμμετέχει στο C. Παραμένει C={A,B}

Επόμενο σημείο το Δ που δεν συμμετέχει στο C και το προσθέτουμε. C={A,B,Δ}

Επόμενο σημείο το Γ που δεν συμμετέχει στο C και το προσθέτουμε. C={A,B,Δ,Γ}

Έχοντας ολοκληρώσει τον κύκλο του Euler ο αλγόριθμος μας δίνει την περιοδεία C: A→B →Δ→Γ→A η οποία έχει κόστος 22.

## 9.2. Αλγόριθμος Χριστοφίδη (Christofides)

Στον Αλγόριθμο Διπλού Ελάχιστου Γεννητικού Δέντρου (Double MST) χρησιμοποιήσαμε ένα ελάχιστο γεννητικό δέντρο του γραφήματος του προβλήματος και διπλασιάζοντας το, δημιουργήσαμε ένα κύκλο Euler. Ο Νίκος Χριστοφίδης (Christofides) το 1976 [12] παρουσίασε μια τροποποίηση του αλγορίθμου αντικαθιστώντας τον διπλασιασμό του ελάχιστου γεννητικού δέντρου με ένα ελάχιστου κόστους Πλήρες Ταίριασμα (minimum weight perfect matching -MWPM) των περιττού βαθμού κορυφών του MST.

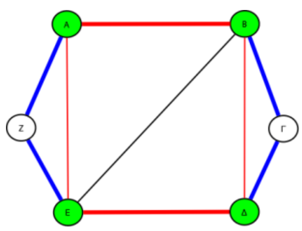
Η επιλογή του Πλήρους Ταιριάσματος επιβαρύνει υπολογιστικά τον αλγόριθμο, ανεβάζοντας την τάξη πολυπλοκότητας σε $O(n^3)$ [28], όμως παράλληλα ο λόγος προσέγγισης χειρότερης περίπτωσης κατεβαίνει στο $^3/_2$. Ο αλγόριθμος παραμένει ο καλύτερος που γνωρίζουμε για το Μετρικό - TSP μέχρι σήμερα. Το μήκος των περιοδειών που μας δίνει, είναι κατά μέσο όρο 10% μεγαλύτερο του φράγματος Held-Karp. [55]

Θα δείξουμε ότι ο αλγόριθμος είναι $^3/_2$-προσεγγιστικός.

Όπως είδαμε στην προηγούμενη παράγραφο το κόστος του MST είναι κάτω φράγμα για το κόστος της άριστης διαδρομής . Έχουμε λοιπόν cost(MST)≤OPT.

Θα δείξουμε ότι το κόστος του MWPM των περιττού βαθμού κορυφών είναι μικρότερο από το μισό κόστος της άριστης διαδρομής.

Θεωρούμε H τον της τριγωνικής ανισότητας cost(H′) ≤ cost(H)=OPT. Ο H′ αποτελείται από την ένωση δύο ταιριασμάτων των κορυφών του συνόλου V′, το καθένα το οποίο αποτελείται από τις ακμές του H εναλλάξ. Το ταίριασμα με το μικρότερο κόστος από τα δύο είναι μικρότερο από το $^1/_2$ cost(H′) ≤ $^1/_2$ OPT.



**Εικόνα 16 Άθροισμα δύο Ελάχιστου κόστους Πλήρες Ταιριασμάτων**

Το MWPM που ζητάμε για τον αλγόριθμο μας είναι πλήρες (perfect) και επομένως είναι μικρότερου κόστους από οποιοδήποτε άλλο ταίριασμα.

Συνεπώς cost(MWPM) ≤ $^1/_2$ cost(H′) ≤ $^1/_2$ OPT

Ενώνοντας το ελάχιστο συνεκτικό δέντρο MST και το ελάχιστου κόστους πλήρες ταίριασμα MWPM έχουμε ένα γράφημα στο οποίο μπορούμε να βρούμε κύκλο Euler και κύκλο Hamilton όπως στον προηγούμενο αλγόριθμο. Λόγω της τριγωνικής ανισότητας η παράκαμψη είναι μικρότερη του κόστους των δύο ή περισσότερων ακμών του κύκλου του Euler που αντικαθιστά και τελικά έχουμε:

$$\text{cost(C)} \leq \text{cost(Euler)} \leq \text{cost(MWPM)} + \text{cost(MST)}\ ) \leq 1 + {}^1/_2\ \text{OPT}.$$

Αναλυτικά ο αλγόριθμος έχει ως εξής:

1. Κατασκευάζουμε ένα ελάχιστο γεννητικό δέντρο με κορυφές τις πόλεις του προβλήματος (MST).
2. Βρίσκουμε ένα Πλήρες Ταίριασμα για τις περιττού βαθμού κορυφές του MST.
3. Δημιουργούμε ένα κύκλο Euler με τις ακμές από τα MST και PM.
4. Διασχίζουμε τον κύκλο του Euler. Κάθε σημείο που συναντάμε για πρώτη φορά συμμετέχει στην λύση, ενώ όταν συναντάμε σημείο που έχουμε ήδη επισκεφθεί το προσπερνάμε και συνεχίζουμε.

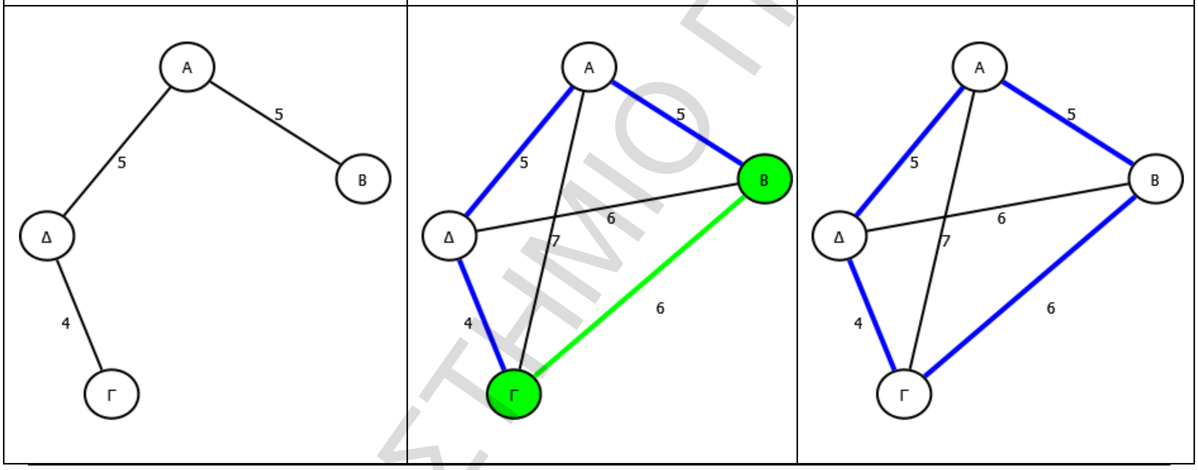
Θα εφαρμόσουμε τον αλγόριθμο στο πρόβλημα TSP 4 πόλεων (6.4.4) της προηγούμενης παραγράφου.

Στο πρώτο βήμα πρέπει να βρούμε το ελάχιστο γεννητικό δέντρο MST του γραφήματος όπως το έχουμε υπολογίσει στην προηγούμενη παράγραφο.

Στο δεύτερο βήμα θα βρούμε ένα Πλήρες Ταίριασμα ελαχίστου κόστους MWPM για τις περιττού βαθμού κορυφές. Οι κορυφές Γ και Β είναι περιττού βαθμού στο ελάχιστο γεννητικό δέντρο και υπολογίζουμε το Πλήρες Ταίριασμα που είναι η ακμή ΒΓ.

Στο τρίτο βήμα ενώνουμε τα δύο γραφήματα (MST + MWPM) και διασχίζουμε τον κύκλο του Euler δημιουργώντας την περιοδεία C αντίστοιχα με την προηγούμενη παράγραφο. Στο παράδειγμα μας η διαδικασία είναι προφανής.

| ΒΗΜΑ 1<br>ΕΛΑΧΙΣΤΟ ΓΕΝΝΗΤΙΚΟ ΔΕΝΤΡΟ | ΒΗΜΑ 2<br>ΠΛΗΡΕΣ ΤΑΙΡΙΑΣΜΑ ΕΛΑΧΙΣΤΟΥ ΚΟΣΤΟΥΣ ΠΕΡΙΤΤΟΥ ΒΑΘΜΟΥ ΚΟΡΥΦΩΝ | ΒΗΜΑ 3 |
|---|---|---|




**Εικόνα 17 Πίνακας Βημάτων Αλγορίθμου Χριστοφίδη**

# 10. Προσεγγιστικά Σχήματα Πολυωνυμικού Χρόνου (PTAS) του Ευκλείδειου TSP

Ο Arora το 1996 [16] παρουσίασε έναν PTAS για το Ευκλείδειο-TSP στο επίπεδο (d=2), ενώ σχεδόν ταυτόχρονα ο Mitchell [17] παρουσίασε τον δικό του. Η παρουσίαση PTAS και μάλιστα δύο διαφορετικών, ήταν από τα πιο σημαντικά γεγονότα στην ιστορία του TSP, ειδικά μετά την απόδειξη της μη ύπαρξης PTAS για το Μετρικό-TSP στο οποίο ανήκει και το Ευκλείδειο-TSP. Για την ανακάλυψη αυτή οι Arora και Mitchell μοιράστηκαν το βραβείο Gödel 2010.

## 10.1. Ο Αλγόριθμος Arora

Η κεντρική ιδέα του αλγορίθμου του Arora ήταν να διαμερίσει το επίπεδο και κατόπιν να χρησιμοποιήσει δυναμικό προγραμματισμό για να βρει μια περιοδεία η οποία διασχίζει τις ευθείες τις διαμέρισης το πολύ $O(\varepsilon)$ φορές, όπου $\varepsilon > 0$ η ζητούμενη ποιότητα της προσέγγισης για το PTAS. Η ιδέα, όπως ο ίδιος αναφέρει, δεν είναι πρωτότυπη αλλά είχε ήδη χρησιμοποιηθεί από τους Karp και Smith [16, p. 3] για έναν ακριβή αλγόριθμο πολυπλοκότητας $2^{O(\sqrt{n})}$.

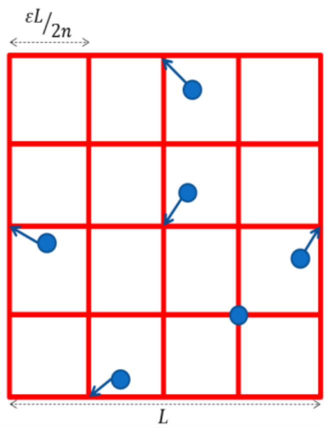
Ο PTAS που παρουσίασε Arora το 1996 είχε πολυπλοκότητα τάξης $n^{O(1/\varepsilon)}$ [63] Ένα χρόνο αργότερα ο ίδιος βελτίωσε την πολυπλοκότητα του αλγορίθμου σε $O\left(n(\log n)^{O(\varepsilon)}\right)$. [16]Παράλληλα γενίκευσε τον αλγόριθμο για Ευκλειδείους χώρους μεγαλύτερων διαστάσεων $(d \geq 2)$ με σχεδόν γραμμική τάξη πολυπλοκότητας $O\left(n(\log n)^{O(\frac{\sqrt{d}}{\varepsilon})^{d-1}}\right)$.

Εδώ θα περιγράψουμε τον PTAS του Arora για το Ευκλείδειο-TSP σε ***2***-διάστατο Ευκλείδειο χώρο. Χαρακτηριστικό του αλγορίθμου είναι ότι δεν είναι ντετερμινιστικός αλλά Πιθανοθεωρητικός (Probabilistic).

Ορίζουμε ως περιβάλλον τετράγωνο (bounding box) το μικρότερο παράλληλο προς τους άξονες τετράγωνο πλευράς L το οποίο περιέχει και τα $n$ σημεία του προβλήματος. Έχουμε ότι:

$L$=max (max$_i$(x$_i$)- min$_i$(x$_i$), max$_i$(y$_i$)- min$_i$(y$_i$)).

Τουλάχιστον 2 σημεία βρίσκονται στις απέναντι πλευρές του τετραγώνου και συνεπώς το μήκος της πλευράς του περιβάλλοντος τετραγώνου είναι μικρότερο της άριστης περιοδείας $L \leq OPT$. Θεωρούμε σύστημα συντεταγμένων επί του περιβάλλοντος τετραγώνου με μοναδιαία απόσταση ${}^{\varepsilon L}/_{2n}$ .




**Εικόνα 18 Διατάραξη του στιγμιότυπου του προβλήματος**

Στη συνέχεια προχωρούμε σε μια διατάραξη του στιγμιότυπου του προβλήματος (perturbation), μετατοπίζοντας κάθε ένα από τα $n$ σημεία πάνω σε σημείο τομής του πλέγματος των συντεταγμένων. Επειδή κάθε σημείο θα μετακινηθεί το πολύ κατά ${}^{\varepsilon L}/_{2n}$ , η απόσταση μεταξύ δύο σημείων θα μεταβληθεί κατά $\pm{}^{2\varepsilon L}/_{2n}$ και άρα συνολικά το κόστος μιας περιοδείας το πολύ κατά $\pm\varepsilon L$ .

Κατόπιν αυξάνουμε την μοναδιαία απόσταση του συστήματος συντεταγμένων ώστε να είναι ίση με 4=${}^{\varepsilon L}/_{2n} \cdot {}^{8n}/_{\varepsilon L}$ και μεταφέρουμε το περιβάλλον τετράγωνο ώστε η κάτω αριστερή γωνία του να συμπίπτει με την αρχή των αξόνων. Με την διαδικασία αυτή εξασφαλίζουμε ότι κάθε σημείο έχει θετικές ακέραιες συντεταγμένες και η ελάχιστη απόσταση μεταξύ δύο σημείων είναι τουλάχιστον 4. Επίσης ενώ στο αρχικό στιγμιότυπο του προβλήματος η μέγιστη απόσταση δύο σημείων είναι το πολύ $2L$ , μετά τη διατάραξη είναι το πολύ $2L \cdot {}^{8n}/_{\varepsilon L} = O(n)$. Επομένως αν $C$ το κόστος μιας περιοδείας και $C'$ το κόστος της ίδιας μετά την διατάραξη έχουμε ότι:

$$ {}^{8n}/_{\varepsilon L} \cdot (C - \varepsilon L) \le C' \le {}^{8n}/_{\varepsilon L} \cdot (C + \varepsilon L) \quad (1) $$

Έστω $OPT$ το κόστος της άριστης περιοδείας και $C$ της περιοδείας που επιστρέφει ο PTAS αλγόριθμος μας για το αρχικό στιγμιότυπο του προβλήματος. Επίσης $OPT'$ και $C'$ τα αντίστοιχα κόστη μετά την διατάραξη του προβλήματος. Προφανώς ισχύει ότι $C' \le (1+\varepsilon)OPT'$ και λόγω της (1) $OPT' \le {}^{8n}/_{\varepsilon L} \cdot (OPT + \varepsilon L)$ .

Επομένως έχουμε:

$$ {}^{8n}/_{\varepsilon L} \cdot (C - \varepsilon L) \le C' \le (1+\varepsilon)OPT' \le (1+\varepsilon)\, {}^{8n}/_{\varepsilon L} \cdot (OPT + \varepsilon L) \Rightarrow $$

$$ {}^{8n}/_{\varepsilon L} \cdot (C - \varepsilon L) \le (1+\varepsilon)\, {}^{8n}/_{\varepsilon L} \cdot (OPT + \varepsilon L) \Rightarrow $$

$$ C - \varepsilon L \le (1+\varepsilon) \cdot (OPT + \varepsilon L) \Rightarrow $$

Όπως είδαμε ισχύει $L \le OPT$ και άρα έχουμε:

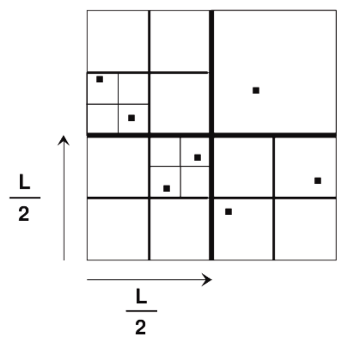
$$ \boxed{C \le (1 + 3\varepsilon + \varepsilon^2)OPT} $$

Επιλέγοντας κατάλληλο $\varepsilon$ μπορούμε να βρούμε περιοδεία $C$ με κόστος το πολύ $(1+\varepsilon')OPT \ \forall \varepsilon' > 0$ .

Με την διαδικασία της διατάραξης έχουμε πλέον ένα «καλό» στιγμιότυπο του προβλήματος με θετικές ακέραιες συντεταγμένες και ελάχιστη απόσταση μεταξύ δύο σημείων τουλάχιστον 4. Για να εφαρμόσουμε τεχνικές δυναμικού προγραμματισμού θα διαμερίσουμε το επίπεδο σε τετράγωνα ώστε να «σπάσουμε» το πρόβλημα σε επιμέρους υποπροβλήματα.

Θεωρούμε $L'$ την μικρότερη δύναμη του 2 που είναι μεγαλύτερη από $2L$. Επιλέγουμε τετράγωνο πλευράς $L'$ που περιέχει το στιγμιότυπο του προβλήματος μας. Όπως είδαμε η μέγιστη απόσταση δύο σημείων μετά τη διατάραξη είναι το πολύ $2L \cdot {}^{8n}/_{\varepsilon L} = O(n)$ , επομένως $L' = O(n)$.

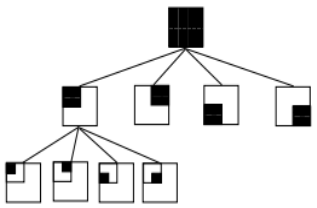
Ονομάζουμε *βασική διαμέριση (dissection)* του τετραγώνου την διαμέριση του σε μικρότερα τετράγωνα. Έτσι το $L'$ x $L'$ τετράγωνο χωρίζεται σε τέσσερα $L'$/2 x $L'$/2 τετράγωνα και ούτω καθεξής. Η διαμέριση αυτή μπορεί να παρουσιαστεί ευκολότερα, με την δομή ενός τετραδικού (quad tree) δέντρου, *T* , του οποίου η ρίζα είναι το περιβάλλον τετράγωνο. Τα τετραδικά δέντρα (quad trees), είναι μία ιεραρχική δομή δεδομένων, που χρησιμοποιείται συχνά σε προβλήματα που απαιτούν διαχωρισμό του χώρου.




**Εικόνα 19 Βασική διαμέριση του περιβάλλοντος τετραγώνου [16]**

Κάθε κόμβος του δέντρου Τ αντιστοιχίζεται σε ένα μοναδικό επίπεδο. Η ρίζα βρίσκεται στο επίπεδο 0, τα παιδιά στο επίπεδο 1 κλπ. Κάθε τετράγωνο αποκτά το επίπεδο του κόμβου του. Συνεπώς ένα τετράγωνο στο επίπεδο *i* έχει διαστάσεις $L'/2^i$ x $L'/2^i$. Η διαμέριση συνεχίζεται μέχρι να πετύχουμε μοναδιαία τετράγωνα δηλαδή τετράγωνα που περιέχουν το πολύ ένα σημείο του προβλήματος. Το Τ έχει βάθος κ= $O(\log L') = O(\log n)$.

Με την έννοια *χρήσιμο τετράγωνο* αναφερόμαστε σε ένα τετράγωνο που αντιπροσωπεύεται από ένα κόμβο στο δέντρο Τ.



**Εικόνα 20 Τετραδικό Δέντρο**

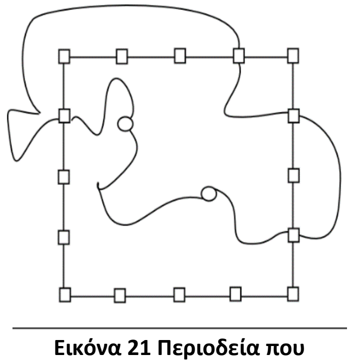
Στην συνέχεια, ορίζουμε τα επίπεδα των οριζοντίων και καθέτων ευθειών οι οποίες αποτελούν την βασική διαμέριση, (δηλαδή όλες οι ευθείες που αποτελούν το δίκτυο συντεταγμένων με το οποίο ορίσαμε το περιβάλλον τετράγωνο). Οι δύο ευθείες που χωρίζουν το περιβάλλον τετράγωνο σε τέσσερα τετράγωνα έχουν επίπεδο 1. Γενικεύοντας, κάθε μία από τις $2^{i}$ ευθείες, οι οποίες χωρίζουν τα επιπέδου ***i***-1 τετράγωνα σε επιπέδου *i* τετράγωνα, έχουν επίπεδο *i* . Ως εκ τούτου μια ευθεία επιπέδου *i* διαμορφώνει την πλευρά των χρήσιμων τετραγώνων επιπέδου *i, i*+1,….κλπ, το μεγαλύτερο από τα οποία έχει διαστάσεις $L'/2^i$ x $L'/2^i$.

Ορίζουμε σε κάθε ευθεία ένα σύνολο σημείων τα οποία ονομάζουμε *πύλες (portals)*. Η περιοδεία, την οποία αναζητούμε, επιτρέπεται να διασταυρωθεί με μια ευθεία, μόνο σε μία πύλη.

Οι πύλες μιας ευθείας είναι σημεία τα οποία ισαπέχουν μεταξύ τους. Σε μια γραμμή επιπέδου *i* , υπάρχουν $L'/(2^i m)$ πύλες, όπου η παράμετρος *m* ορίζεται ως δύναμη του 2 στην περιοχή του [k/ε,2k/ε]. Προφανώς $m = O(\log n/\varepsilon)$.

Επειδή το μεγαλύτερο χρήσιμο τετράγωνο στο επίπεδο *i* έχει διαστάσεις $L'/2^i$ x $L'/2^i$ ,κάθε χρήσιμο τετράγωνο έχει συνολικά μέχρι 4*m* πύλες στις τέσσερις πλευρές του. Επιλέξαμε το m να είναι δύναμη του 2 έτσι ώστε μια πύλη που βρίσκεται σε χαμηλότερου επιπέδου τετράγωνο είναι ταυτόχρονα και πύλη για όλα τα υψηλότερου επιπέδου τετράγωνα στα οποία ανήκει.

Η περιοδεία ***τ*** λέμε ότι *συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση* εάν περνά από $n$ σημεία και οποιοδήποτε υποσύνολο πυλών. Επιπρόσθετα, η περιοδεία, επιτρέπεται να επισκέπτεται πύλες πολλαπλές φορές, αλλά δεν επιτρέπεται να τέμνει με τον εαυτό της.

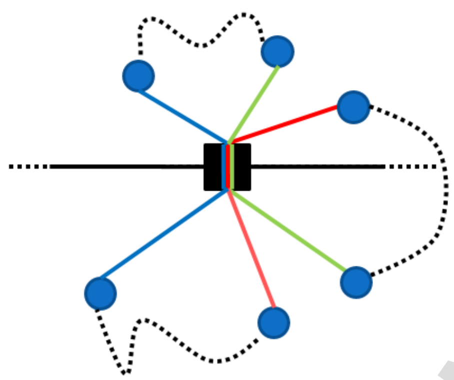
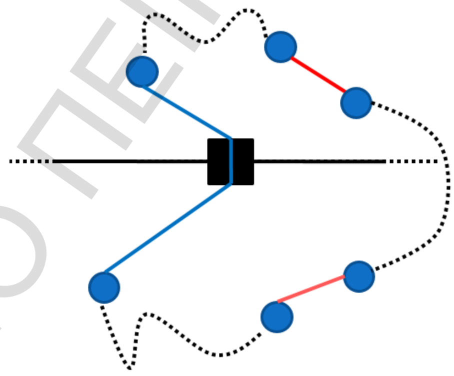
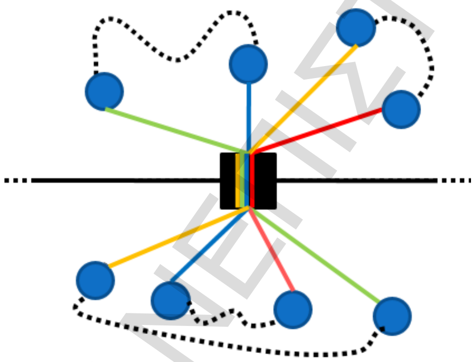
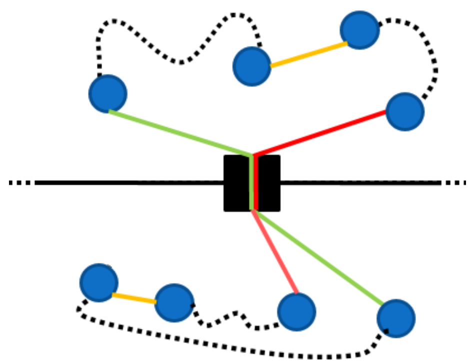


**Εικόνα 21 Περιοδεία που συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση και έχει ελάχιστες διασταυρώσεις**

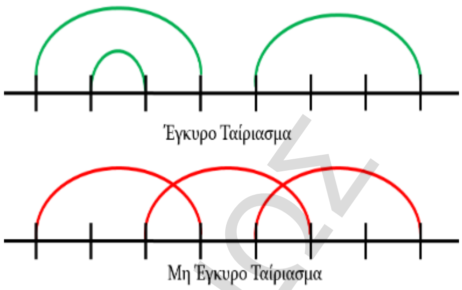
Η περιοδεία ***τ*** *συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση και έχει ελάχιστες διασταυρώσεις* εάν συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση και επιπλέον επισκέπτεται κάθε πύλη το πολύ δύο φορές.

Εάν η περιοδεία ***τ*** *συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση* τότε υπάρχει περιοδεία που *συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση, με ελάχιστες διασταυρώσεις*, τις οποίας το μήκος δεν είναι μεγαλύτερο από αυτό της ***τ*** , καθώς η αφαίρεση των διασταυρώσεων μπορεί να μας δώσει μικρότερη περιοδεία γιατί βρισκόμαστε σε Ευκλείδειο χώρο και ισχύει η τριγωνική ανισότητα.

Εάν η τ χρησιμοποιεί μια πύλη στην γραμμή l περισσότερες από δύο φορές, μπορούμε να συνεχίσουμε να «κόβουμε δρόμο» στις δύο πλευρές της, μέχρι η πύλη να χρησιμοποιείται το πολύ δύο φορές. Εάν αυτή η διαδικασία εισάγει νέες τομές της περιοδείας με τον εαυτό της, τότε και αυτές μπορούν με τον ίδιο τρόπο να αφαιρεθούν.

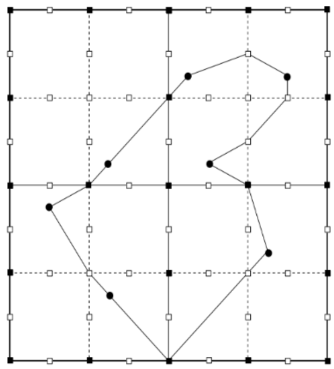
<table>
<tr><td colspan="2">Παράδειγμα ελαχιστοποίησης περιττού αριθμού (3) διασταυρώσεων</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Παράδειγμα ελαχιστοποίησης άρτιου αριθμού (4) διασταυρώσεων</td></tr>
<tr><td></td><td></td></tr>
</table>

Θα δείξουμε ότι *η άριστη περιοδεία, που συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση, με ελάχιστες διασταυρώσεις,* μπορεί να υπολογιστεί σε χρόνο $2^{O(m)} = n^{O(1/\varepsilon)}$.




**Εικόνα 22 Έγκυρο και Μη Έγκυρο Ταίριασμα**

Έστω τ η άριστη περιοδεία που ψάχνουμε. Είναι προφανές ότι η περιοδεία τ μπορεί να εισέλθει και να εξέλθει, από ένα χρήσιμο τετράγωνο S το πολύ 8m φορές. Το κομμάτι τις τ που βρίσκεται μέσα στο τετράγωνο S είναι ένα σύνολο το πολύ 4m μονοπατιών που καλύπτουν όλα τα σημεία εσωτερικά του τετραγώνου. Το κάθε ένα μονοπάτι μπορεί να εισέλθει και να εξέλθει από το S μέσω των πυλών. Αυτό σημαίνει ότι ένα *έγκυρο ταίριασμα* των σημείων εισόδου και εξόδου των μονοπατιών αντιστοιχεί σε μια ισοσταθμισμένη διάταξη παρενθέσεων. Επιπλέον, τα μονοπάτια πρέπει εσωτερικά να μην τέμνονται με τον εαυτό τους, πχ δύο μονοπάτια μπορούν να διασταυρωθούν μόνο στα σημεία εισόδου ή εξόδου του τετραγώνου.



**Εικόνα 23 Συμβατή έγκυρη επίσκεψη [72, p. 262]**

Καλούμε *έγκυρη επίσκεψη (valid visit)* ενός χρήσιμου τετραγώνου, μια ακολουθία άρτιου αριθμού διασταυρώσεων των πυλών του τετραγώνου από την περιοδεία, μαζί με τα ταιριάσματα τους ως σημεία εισόδου και εξόδου μιας περιοδείας που *συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση, με ελάχιστες διασταυρώσεις*.

Ονομάζουμε *συμβατή* μια έγκυρη επίσκεψη στο τετράγωνο $S_{i+1}$ επιπέδου i+1 με μια επίσκεψη στο τετράγωνο $S_i$ που το περιέχει όταν εισέρχεται και εξέρχεται στο τετράγωνο μόνο από πύλες του $S_{i+1}$ .

Ο αριθμός των χρήσιμων τετραγώνων είναι $O(n)$. Θα δείξουμε πρώτα ότι το πλήθος των έγκυρων επισκέψεων σε ένα χρήσιμο τετράγωνο είναι το πολύ $n^{O(1/\varepsilon)}$.

Θεωρούμε ένα χρήσιμο τετράγωνο *S*. Κάθε πύλη του *S* , χρησιμοποιείται 0,1, ή 2 φορές, συνολικά δηλαδή $3^{4m} = n^{O(1/\varepsilon)}$ ενδεχόμενα. Κρατάμε μόνο αυτά που χρησιμοποιούν την πύλη άρτιες φορές (δηλ. έχουν είσοδο και έξοδο). Θεωρούμε ένα ενδεχόμενο, και υποθέτουμε ότι κάνει χρήση 2r πυλών, Στη συνέχεια θεωρούμε όλα τα δυνατά ζεύγη αυτών των πυλών που σχηματίζουν μια ισορροπημένη ακολουθία από παρενθέσεις. Ο αριθμός αυτών των διατάξεων είναι ο r αριθμός του Catalan[1], ο οποίος φράσσεται από το $2^{2r} = n^{O(1/\varepsilon)}$. Επομένως ο συνολικός αριθμός επισκέψεων στο S έχει άνω όριο το $n^{O(1/\varepsilon)}$.

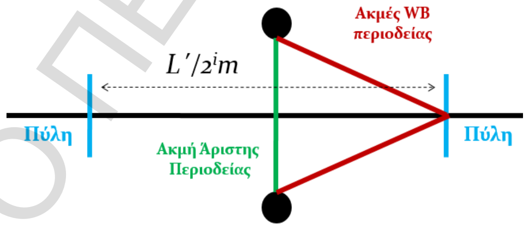
[1] Ο τύπος της ακολουθίας των αριθμών Catalan είναι: $\boldsymbol{C_n} = \frac{\binom{2n}{n}}{\boldsymbol{n+1}} = \frac{(\boldsymbol{2n})!}{(\boldsymbol{n+1})!\,\boldsymbol{n}!} \leq \frac{4^n}{n^{3/2}\sqrt{\pi}} \leq 4^n = 2^{2n}$

Θεωρούμε μια έγκυρη επίσκεψη V σε ένα τετράγωνο S επιπέδου i. Έχουμε ήδη καθορίσει τις εισόδους και τις εξόδους στο τετράγωνο S. Το S έχει τέσσερα παιδιά - τετράγωνα επιπέδου i+1, τα οποία έχουν τέσσερίς εσωτερικές πλευρές (που δεν εφάπτονται με τις πλευρές του S) και άρα έχουν το πολύ 4m ακόμα πύλες. Κάθε μια από αυτές τις πύλες, χρησιμοποιείται 0,1, ή 2 φορές, οδηγώντας και πάλι σε $n^{O(1/\varepsilon)}$ ενδεχόμενα.

Κατόπιν βρίσκουμε όλα τα έγκυρα ταιριάσματα όλων των πυλών και των i+1 επιπέδου τετράγωνων ώστε οι επισκέψεις σε αυτά να είναι συμβατές με την V. Χρησιμοποιώντας όπως και προηγουμένως τους αριθμούς Catalan καταλήγουμε και πάλι ότι αριθμός τους είναι φραγμένος από το $n^{O(1/\varepsilon)}$. Κάθε ένα από τα ταιριάσματα οδηγεί σε έγκυρες επισκέψεις των τεσσάρων τετραγώνων επόμενου επιπέδου.

Επομένως με εφαρμογή δυναμικού προγραμματισμού έχουμε υπολογίσει το κόστος των έγκυρων επισκέψεων. Υπολογίζουμε το άθροισμά τους. Το μικρότερο από αυτά τα αθροίσματα είναι αυτό του βέλτιστου τρόπου υλοποίησης της επίσκεψης V στο τετράγωνο S.

Έχοντας βρει την άριστη περιοδεία, που συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση, με ελάχιστες διασταυρώσεις *(WB)* θα εξετάσουμε πόσο αυτή διαφέρει από την άριστη (*OPT*). Έστω ότι έχουμε την άριστη περιοδεία του προβλήματος μας. Θα την μετατρέψουμε σε WB και θα μετρήσουμε την μεταβολή της. Στην Εικόνα 24 βλέπουμε ότι σε κάθε «πέρασμα» από κάποια πύλη η περιοδεία επιβαρύνεται το πολύ κατά $L'/(2^i m)$. Επισημαίνουμε ότι όσο χαμηλότερο το επίπεδο του τετραγώνου που ανήκει ο κόμβος τόσο μεγαλύτερη η επιβάρυνση.




**Εικόνα 24 Μεταβολή Κόστους Άριστης Περιοδείας**

Στη συνέχεια θα δείξουμε με αντιπαράδειγμα ότι μια WB περιοδεία δεν είναι πάντα μια $(1+\varepsilon)-$ προσέγγιση της άριστης λύσης. Στην Εικόνα 25 παραθέτουμε Ευκλείδειο –TSP. Με χρήση του πυθαγορείου θεωρήματος είναι εμφανές ότι η άριστη περιοδεία (πράσινο χρώμα) έχει μήκος:

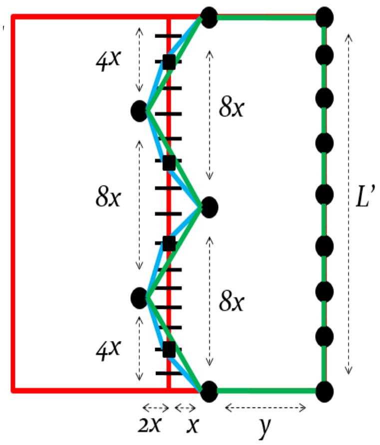
$| OPT | = 4 \cdot \sqrt{(3x)^2 + (4x)^2} + 2y + L'$

Μετατρέποντας την άριστη περιοδεία σε WB (γαλάζιο χρώμα) έχουμε:

$| WB | = 4 \cdot (\sqrt{(3x)^2 + (4x)^2} + \sqrt{x^2 + x^2}) + 2y + L'$

Επειδή στο παράδειγμα ισχύει $y \leq {}^{L'}/_2$ έχουμε ότι ${}^{|WB|}/_{|OPT|} \geq 1.0015$

και επομένως για $\varepsilon < 0.0015$ η διαδικασία που περιγράψαμε αποτυγχάνει.

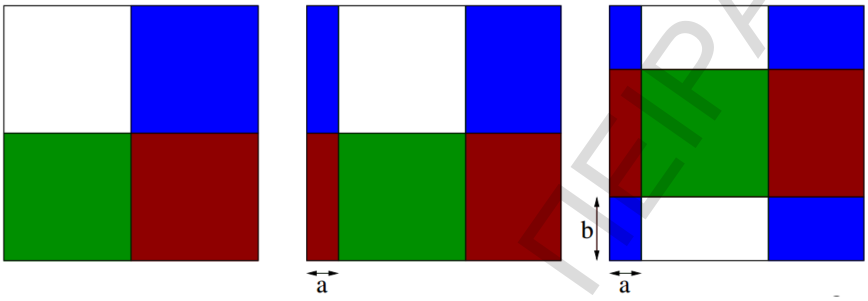



**Εικόνα 25 Αντιπαράδειγμα αποτυχίας προσεγγιστικού αλγορίθμου**

Για να ξεπεράσουμε το πρόβλημα θα μετατρέψουμε τον αλγόριθμο σε Πιθανοθεωρητικό τυχαιοποιόντας (randomize) την διαδικασία της διαμέρισης.

- Επιλέγουμε τυχαία δύο ακεραίους ώστε 0 ≤ a, b < $L'$.
- Μεταφέρουμε κάθε κάθετη ευθεία της διαμέρισης από την θέση x στην θέση (x+a)mod $L'$ .
- Μεταφέρουμε κάθε οριζόντια ευθεία της διαμέρισης από την θέση y στην θέση (y+b)mod $L'$ .

  Καλούμε την νέα διαμέριση *(a,b)-μετατοπισμένη διαμέριση*.




**Εικόνα 26 (a,b)-μετατοπισμένη διαμέριση**

Με την μετατόπιση κάθε ευθεία της διαμέρισης έχει πλέον τυχαίο επίπεδο. Η κάθετη ευθεία 1$^{ου}$ επιπέδου της αρχικής διαμέρισης, δεν είναι 1$^{ου}$ επιπέδου αλλά εξαρτάται από την επιλογή του α. Το επίπεδο της θα είναι το ($L'/2$ + a) mod $L'$ της αρχικής διαμέρισης.

Υπάρχουν $2^1$ ευθείες 1$^{ου}$ επιπέδου,$2^2$ ευθείες 2$^{ου}$ επιπέδου, …, $2^i$ ευθείες i-επιπέδου

Επειδή έχουμε k επίπεδα, συνολικά έχουμε $2^{k+1}-2$ ευθείες στην διαμέριση.

Η πιθανότητα να επιλέξουμε μια ευθεία i-επιπέδου είναι:

$$p(i-\varepsilon\pi\iota\pi\acute{\varepsilon}\delta o\upsilon)=\frac{2^i}{2^{k+1}-2}=\frac{2^i}{2L'-2}\leq\frac{2^i}{L'}$$

Δείξαμε ότι όταν η τ διασταυρώνεται με ευθεία της διαμέρισης i-επιπέδου κατά τη διαδικασία δημιουργίας της WB αυτή επιβαρύνεται το πολύ κατά $x(i)=\frac{L'}{2^i m}$.

Θέτουμε X τυχαία μεταβλητή της πιθανότητας αύξησης του μήκους της WB όταν <u>διασταυρώνεται με μια ευθεία της διαμέρισης</u>. Η αναμενόμενη αύξηση δίνεται από τον τύπο:

$$E(X)=\sum_{i=1}^{k}p(i)\cdot x(i)\leq\sum_{i=1}^{k}\frac{2^i}{L'}\cdot\frac{L'}{2^i m}=\frac{k}{m}\leq\varepsilon$$

με $\frac{k}{\varepsilon}\leq m\leq 2\frac{k}{\varepsilon}$

Για να υπολογίσουμε τη συνολική αύξηση από όλες τις διασταυρώσεις με τις ευθείες τις διαμέρισης θα βρούμε ένα φράγμα για τον συνολικό αριθμό των διασταυρώσεων. Έστω $\boldsymbol{\tau}$ άριστη περιοδεία με $|\tau| = OPT$ και $N(\boldsymbol{\tau})$ ο αριθμός των διασταυρώσεων της με τις ευθείες της διαμέρισης. Θα δείξουμε ότι $N(\tau) \le \sqrt{2} \cdot OPT$.

Έστω ευθύγραμμο τμήμα $c$ της άριστης περιοδείας $\boldsymbol{\tau}$ από το σημείο $(x_1, y_1)$ στο σημείο $(x_2, y_2)$. Το $c$ μπορεί να διασταυρώνεται το πολύ $|x_2 - x_1|$ φορές με τις κάθετες ευθείες της διαμέρισης και $|y_2 - y_1|$ και άρα $N(c) \le |x_2 - x_1| + |y_2 - y_1|$

Το τμήμα λόγω του πυθαγορείου θεωρήματος έχει μήκος $D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

και συνεπώς έχουμε ότι: $D^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \ge {((x_2 - x_1) + (y_2 - y_1))^2}/{2} \Leftrightarrow\ 2 \cdot D^2 \ge ((x_2 - x_1) + (y_2 - y_1))^2 \Leftrightarrow\ \sqrt{2} \cdot D\ \ge |x_2 - x_1| + |y_2 - y_1| \ge N(c)$

Αθροίζοντας τα επιμέρους τμήματα της άριστη περιοδείας έχουμε ότι $N(\tau) \le \sqrt{2} \cdot OPT$

Θέτουμε Y τυχαία μεταβλητή της πιθανότητας αύξησης του μήκους της WB για το σύνολο των διασταυρώσεών της με της ευθείες της διαμέρισης. Η αναμενόμενη αύξηση δίνεται από τον τύπο:

$$E(Y) = N(\tau) \cdot E(X) \le \sqrt{2} \cdot OPT \cdot \varepsilon$$

Η ανισότητα Markov για τυχαία μεταβλητή $X$ και $a > 0$ είναι $P(X > a) \le \frac{E(X)}{a}$

Εφαρμόζοντας την ανισότητα Markov για την τυχαία μεταβλητή Y έχουμε ότι:

$$P\big(Y > 2 \cdot \sqrt{2} \cdot OPT \cdot \varepsilon\big) \le \frac{E(Y)}{2 \cdot \sqrt{2} \cdot OPT \cdot \varepsilon} \le \frac{1}{2}$$

Συνεπώς θέτοντας $\varepsilon' = 2 \cdot \sqrt{2} \cdot \varepsilon$ έχουμε την επιθυμητή προσέγγιση με πιθανότητα τουλάχιστον $\frac{1}{2}$.

Συνοψίζουμε τα βήματα του αλγορίθμου του Arora:

1. Βρίσκουμε το περιβάλλον τετράγωνο (bounding box) του προβλήματος.
2. Διαταράσσουμε (perturbation) το αρχικό πρόβλημα ώστε να έχουμε πλέον ένα «καλό» στιγμιότυπο του προβλήματος με θετικές ακέραιες συντεταγμένες και ελάχιστη απόσταση μεταξύ δύο σημείων τουλάχιστον 4.
3. Διαμερίζουμε το περιβάλλον τετράγωνο με μια *(a,b)-μετατοπισμένη διαμέριση* για τυχαίους ακέραιους 0≤a,b<L’
4. Εφαρμόζουμε τεχνικές δυναμικού προγραμματισμού για να υπολογίσουμε μια άριστη περιοδεία, *που συμπεριφέρεται καλώς αναφορικά με την βασική διαμέριση, με ελάχιστες διασταυρώσεις (WB)*
5. Με πιθανότητα τουλάχιστον $\frac{1}{2}$ η WB είναι μια $(1 + \varepsilon) -$ προσέγγιση της άριστης περιοδείας (OPT) αναφορικά με την βασική διαμέριση.

Ο αλγόριθμος μπορεί να γίνει ντετερμινιστικός εφαρμόζοντας τον αλγόριθμο για όλες τις πιθανές ολισθήσεις της διαμέρισης του περιβάλλοντος τετραγώνου, οι οποίες είναι $L'^4 = O(n^4)$ και επιλέγοντας την μικρότερη περιοδεία WB.

## 10.2. Ο Αλγόριθμος Mitchell

Ο Mitchell [17] παρουσίασε το 1996 έναν PTAS για το Ευκλείδειο-TSP, ανεξάρτητα από τον Arora. Οι ιδέες του αλγορίθμου του είναι παρόμοιες με του Arora, καθώς διαμερίζει το επίπεδο σε μικρότερα παραλληλόγραμμα και χρησιμοποιεί δυναμικό προγραμματισμό για την εύρεση λύσης.

Για τον διαμερισμό του επιπέδου, ο Mitchell εισάγει την μέθοδο της «m-guillotine» διαμέρισης, μια μέθοδο πολυγωνικής υποδιαίρεσης (polygonal subdivision) του επιπέδου.

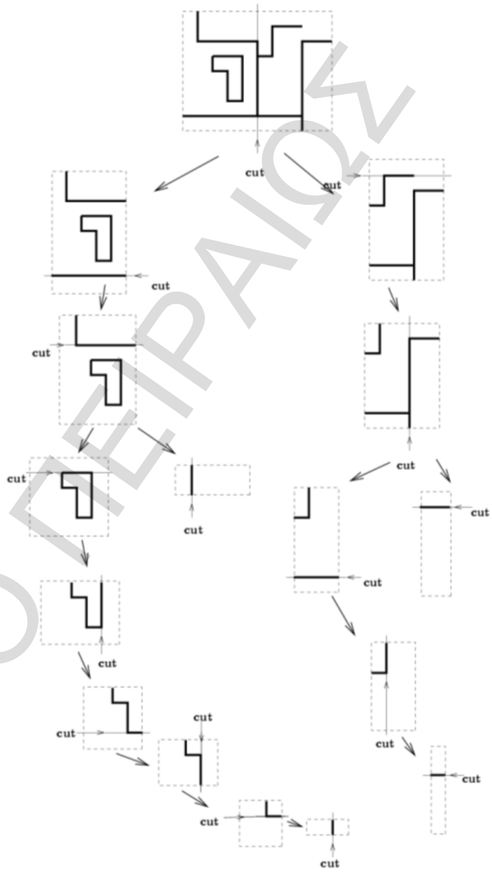
Οι υποδιαιρέσεις του επιπέδου γίνονται με *κοψίματα* (cuts), όπου ένα κόψιμο είναι μια ευθεία που χωρίζει ένα παραλληλόγραμμο R του προβλήματος σε δύο μικρότερα. Μια διαμέριση του επιπέδου είναι «m-guillotine», αν υπάρχει κόψιμο το οποίο διασταυρώνεται με υπάρχοντα κοψίματα, δημιουργώντας το πολύ $O(m)$ συνδεδεμένα τμήματα και τα δύο νέα παραλληλόγραμμα είναι επίσης «m-guillotine».

Μια γέφυρα (bridge) είναι ένα ευθύγραμμο τμήμα του κοψίματος, που τα άκρα του είναι τομές το κοψίματος με την περιοδεία του προβλήματος.

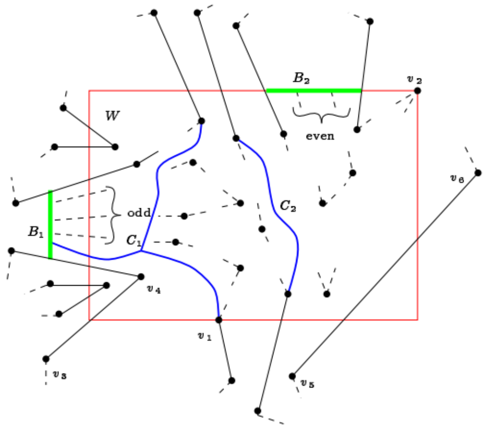
Το βασικό θεώρημα που αποδεικνύει ο Mitchell είναι ότι κάθε πολυγωνική υποδιαίρεση του επιπέδου μπορεί να μετατραπεί σε «m-guillotine» λειτουργώντας επαναληπτικά πάνω σε παραλληλόγραμμα R, όπου προσπαθεί να βρει γέφυρες.

Με το θεώρημα αυτό και κάνοντας χρήση δυναμικού προγραμματισμού δημιουργεί έναν $1 + {}^{2\sqrt{2}}/_{m}$-προσεγγιστικό αλγόριθμο ο οποίος έχει πολυπλοκότητα $O(n^{20m+5})$.

Ο αλγόριθμος PTAS του Mitchell δεν έχει γίνει δυνατόν να επεκταθεί σε περισσότερες διαστάσεις Ευκλείδειου χώρου, ούτε έχει δεχθεί βελτιώσεις όπως ο αντίστοιχος του Arora.




**Εικόνα 27 Παράδειγμα 1-guillotine υποδιαίρεσης [17]**




**Εικόνα 28 Περιβάλλον παραλληλόγραμμο και γέφυρες του Mitchell [17]**

# 11. Συμπεράσματα – Περίληψη

| **Πρόβλημα** | **Αλγόριθμος** | **Πολυπλοκότητα** | **Λόγος προσέγγισης χειρότερης περίπτωσης** | **Held Karp Φράγμα** |
|---|---|---|---|---|
| *ΚΑΤΗΓΟΡΙΑ ΑΛΓΟΡΙΘΜΟΥ* | | | | |
| *ΑΚΡΙΒΕΙΣ (EXACT)* | | | | |
| TSP | *Εξαντλητικής αναζήτηση (exhaustive )* | $O(n!)$ | 1 | |
| TSP | *Bellman – Held - Karp* | $O(n^2 2^n)$ | 1 | |
| TSP | *Branch & Bound* | $O(2^n)$ | 1 | |
| *ΚΑΤΑΣΚΕΥΑΣΤΙΚΟΙ ΕΥΡΕΤΙΚΟΙ ( CONTRUCTION HEURISTICS)* | | | | |
| TSP | *Πλησιέστερου Γείτονα (Nearest Neighbor).* | $O(n^2)$ | $1/2 \cdot (log_2(n) + 1)$ | 23% |
| TSP | *Άπληστος (Greedy)* | $O(n^2 log_2(n))$ | $1/2 \cdot (log_2(n) + 1)$ | 14% |
| *ΕΥΡΕΤΙΚΟΙ ΠΑΡΕΜΒΟΛΗΣ ( INSERTION HEURISTICS)* | | | | |
| TSP | *Κοντινότερη Εισαγωγή (Nearest Insertion)* | $O(n^2)$ | $2 - \frac{2}{n}$ | 27% |
| TSP | *Εισαγωγή Ελαχίστου Κόστους (Cheapest Insertion)* | $O(n^2 log_2(n))$ | $2 - \frac{2}{n}$ | 22% |
| TSP | *Απομακρυσμένη Εισαγωγή (Farthest Insertion)* | $O(n^2)$ | - | |
| TSP | *Clark & Wright* | $O(n^2 log_2(n))$ | $log_2(n) + 1$ | 12% |
| *ΒΕΛΤΙΩΤΙΚΟΙ ΕΥΡΕΤΙΚΟΙ (IMPROVEMENT HEURISTICS)* | | | | |
| Ευκλείδειο-TSP | *2-opt* | $O(n^2)$ | $<4 \cdot \sqrt{n}$ | 6,9% |
| Ευκλείδειο-TSP | *3-opt* | $O(n^3)$ | | 4,6% |
| Ευκλείδειο-TSP | *k-opt* | $O(n^k)$ | $>\frac{1}{4} n^{\frac{1}{2k}}$ | |
| Ευκλείδειο-TSP | *Lin-Kernighan(v-opt)* | $O(n^2)$ | | 1-2% |
| *ΣΤΑΘΕΡΟΥ ΛΟΓΟΥ ΠΡΟΣΕΓΓΙΣΗΣ* | | | | |
| Μετρικό –TSP | *Διπλού Ελάχιστου Γεννητικού Δέντρου (Double MST)* | $O(n^2 log_2(n))$ | 2 | |
| Μετρικό –TSP | *Χριστοφίδη (Christofides)* | $O(n^3)$ | $3/2$ | 10% |
| *ΠΡΟΣΕΓΓΙΣΤΙΚΑ ΣΧΗΜΑΤΑ ΠΟΛΥΩΝΥΜΙΚΟΥ ΧΡΟΝΟΥ (PTAS)* | | | | |
| Ευκλείδειο-TSP $(d \geq 2)$ | Arora | $O\left(n(\log n)^{O\left(\frac{\sqrt{d}}{\varepsilon}\right)^{d-1}}\right)$ | $1 + \varepsilon$ | |
| Ευκλείδειο-TSP $(d = 2)$ | Mitchell | $O(n^{20m+5})$ | $1 + 2\sqrt{2}/m$ | |

Συνοψίζοντας την παρουσίαση των σημαντικότερων προσεγγιστικών αλγορίθμων επίλυσης του προβλήματος του Περιοδεύοντος πωλητή, παραθέτουμε πίνακα με τις πολυπλοκότητες, τους λόγους προσέγγισης χειρότερης περίπτωσης καθώς και τους μέσους όρους απόκλισης των αποτελεσμάτων τους σε σχέση με το φράγμα Held -Karp.

Όπως είδαμε στην ιστορική αναδρομή η προσπάθεια επίλυσης του προβλήματος έχει εντατικοποιηθεί τις τελευταίες τέσσερις δεκαετίες και έχει αποδώσει πληθώρα αλγορίθμων και τεχνικών. Παράλληλα όμως η θεωρητική έρευνα έχει θέσει και τα όρια των δυνατοτήτων μας, ήδη από την δεκαετία του 1970, με την απόδειξη της NP-Πληρότητας και της μη προσεγγισιμότητας του γενικού προβλήματος.

Η δυσκολία του προβλήματος και της προσεγγισιμότητας του, που απέδειξαν τα θεωρητικά αποτελέσματα, δικαιολογεί και την δυσκολία στην παρουσίαση αποδοτικότερων αλγορίθμων. Χαρακτηριστικά αναφέρουμε ότι ο αλγόριθμος του Χριστοφίδη (1976) παρέμεινε για περισσότερο από δύο δεκαετίες ο καλύτερος από πλευράς θεωρητικής απόδοσης γνωστός αλγόριθμος. Η παρουσίαση PTAS για το Ευκλείδιο –TSP, από τους Arora και Mitchell το 1996, ήταν ένα από τα σημαντικότερα αποτελέσματα όσο αναφορά το TSP, με μεγάλη θεωρητική αξία αλλά και δύσκολη πρακτική εφαρμογή.

Από πλευράς πρακτικών εφαρμογών οι Ακριβείς χρησιμοποιούνται για πολύ μικρού μεγέθους προβλήματα (n<5). Για μεγαλύτερα προβλήματα οι Κατασκευαστικοί Ευρετικοί αλγόριθμοι επιλέγονται λόγω της ευκολίας υλοποίησης, ενώ για n>10 οι Ευρετικοί Παρεμβολής έχουν αντίστοιχη πολυπλοκότητα και καλύτερο λόγο προσέγγισης. Οι αλγόριθμοι του Διπλού Γεννητικού Δέντρου και του Χριστοφίδη χρησιμοποιούνται όταν απαιτείται εγγυημένος λόγος προσέγγισης.

Για μεγάλου μεγέθους προβλήματα οι Βελτιωτικοί είναι καλύτερη επιλογή και ειδικά ο αλγόριθμος Lin-Kernighan είναι ο αποδοτικότερος, όπως φαίνεται από την σύγκριση των αποδόσεων σε σχέση με το φράγμα Held-Karp. Το λογισμικό Concorde TSP Solver των Applegate, Bixby, Chvatal και Cook, που συνδυάζει αλγορίθμους και τεχνικές, είναι σήμερα το σημαντικότερο εργαλείο επίλυσης μεγάλων προβλημάτων TSP.

Εκτός των αλγορίθμων που παρουσιάστηκαν σε αυτή την εργασία, πρέπει να σημειώσουμε ότι υπάρχει πλειάδα άλλων τεχνικών και αλγορίθμων για την επίλυση του TSP, οι οποίοι έχουν αποδώσει σημαντικά αποτελέσματα. Αναφέρουμε ενδεικτικά τους αλγόριθμους αναζήτησης με tabu (tabu search), τους αλγορίθμους προσομοίωσης ανόπτησης (simulated annealing) και τους γεννητικούς αλγόριθμους. Επίσης πρέπει να αναφέρουμε ότι στα πλαίσια της στρατηγικής «διαίρει και βασίλευε», έχουν παραχθεί εξειδικευμένοι αλγόριθμοι και τεχνικές με σημαντικά αποτελέσματα για ειδικές περιπτώσεις του προβλήματος.

## 12. Βιβλιογραφία


[1] Σπύρου Π.Γ., Θεωρία Γραφημάτων, 1999.

[2] L. Apllegate, R. Bixby, W. Cook και V. Chvatal, The Traveling Salesman Problem A Computational Study, Princeton University Press, 2006.

[3] M. Gardner, Mathematical Games: About the Remarkable Similarity between the Icosian Game and the Towers of Hanoi., Sci. Amer. 196, 1957.

[4] B. F. Voigt, Der Handlungsreisende—wie er sein soll und was er zu thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiss zu sein Von einem alten Commis-Voyageur, Ilmenau, 1832.

[5] G. Dantzig, R. Fulkerson και S. Johnson, Solution of a large-scale traveling-salesman problem., Journal of the operations research society of America 2.4, 1954.

[6] G. Croes, A method for solving traveling salesman problems. Operations Res. 6 (1958) , pp., 791-812., 1958.

[7] M. Held και R. Karp, A Dynamic Programming Approach to Sequencing Problem., 1962.

[8] R. Bellman, Dynamic programming treatment of the travelling salesman problem, Journal of Assoc. Computing Mach. 9. 1962., 1962.

[9] R. M. Karp, Reducibility among combinatorial problems, Springer, 1972.

[10] S. Lin και B. W. Kernighan, An Effective Heuristic Algorithm for the Traveling-Salesman Problem, 1973.

[11] P. Miliotis, Using cutting planes to solve the symmetric travelling salesman problem., salesman problem. Mathematical Programming 15, 1978.

[12] N. Christofides, Worst-case analysis of a new heuristic for the traveling salesman problem. Report 388, Graduate School of Industrial Administration, CMU, 1976, 1976.

[13] S. Sahni και T. Gonzalez, P-Complete Approximation Problems. J., ACM 23, 1976.

[14] M. Gotschel και M. W. Padberg, On the symmetric traveling salesman problem I: Inequalities., Mathematical Programming 16, 1979.

[15] M. Olivier, S. W. Otto και E. W. Felten, Large-step Markov chains for the traveling salesman problem, Oregon Graduate Institute of Science and Technology, Department of Computer Science and Engineering, 1991.

[16] S. Arora, Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems, Foundations of Computer Science, Proceedings., 37th Annual Symposium on. IEEE, 1996., 1996.

[17] J. S. Mitchell, Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems., Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms. SIAM, 1996.

[18] K. Helsgaun, An effective implementation of the Lin–Kernighan traveling salesman heuristic, EJOR 12, 2000.

[19] D. Apllegate, R. Bixby, V. Chvatal και W. Cook, Concorde TSP solver, 2006.

[20] C. H. Papadimitriou, The Euclidean traveling salesman problem is NP-complete, Theoretical Computer Science 4, 1977.

[21] J. Kruskal, Proceedings of the American Mathematical Society, 1956.

[22] R. C. Prim, Shortest connection networks and some generalizations. In: Bell System Technical Journal, 36, 1957.

[23] J. Edmonds, Path, trees, and flowers., Can. J. Math.17, 1965.

[24] S. Cook, The Complexity of Theorem Proving Procedures, 1971.

[25] R. Karp, Reducibility Among Combinatorial Problems, 1972.

[26] R. Z. Farahani και M. Hekmatfar, Facility Location: Concepts, Models, Algorithms and Case Studies, Springer, 2009.

[27] Ζησιμόπουλος Β, Αλγόριθμοι και Πολυπλοκότητα, 2008.

[28] C. H. Papadimitriou και K. Steiglitz, Combinatorial Optimization Algorithms and Complexity, Dover, 1998.

[29] M. Garey και D. Johnson, Computers and intractability: A guide to the theory of NP-completeness, Freeman, 1979.

[30] M. S. Daskin, Network and Discrete Location: Models, Algorithms, and Applications, John Wiley & Sons, 2013.

[31] G. Gutin και A. Punnen, The Traveling Salesman Problem And Its Variations, Kluwer, 2004.

[32] C. H. Papadimitriou, V. Vazirani και S. Dasgupta, Algorithms, McGraw-Hill, 2006.

[33] Woeginger G J, Exact Algorithms for NP-Hard Problems: A Survey, Combinatorial Optimization – Eureka, You Shrink! Lecture notes in computer science, 2003.

[34] A. Land και A. Doig, An automatic method of solving discrete programming problems, 1960.

[35] W. Eastman, Linear programming with pattern constraints Ph.D. Thesis, Harvard University, 1958.

[36] H. Kuhn, The Hungarian method for the assignment problem., Naval Res. Legist. Quart. 2, 83-97, 1955.

[37] D. Shapiro, Algorithms for the solution of the optimal cost and bottleneck traveling salesman problems Sc.D. Thesis, Washington University, St. Louis, MO, 1966.

[38] R. Garfinkel, On partitioning the feasible set in a branch-and-bound algorithm for the asymmetric traveling-salesman problem, Operations Research 21, 1973.

[39] T. Smith, V. Srinivasan και G. Thompson, Computational performance of three subtour elimination algorithms for solving asymmetric traveling salesman problems, Annals of Discrete Mathematics 1, 1977.

[40] E. Balas και N. Christofides, A restricted lagrangean approach to the traveling salesman problem, Mathematical Programming 21, 1981.

[41] D. Miller και J. Pekny, Exact solution of large asymmetric traveling salesman problems, Science 251, 1991.

[42] B. Korte και J. Vygen, Combinatorial Optimization: Theory and Algorithms, Springer, 2012.

[43] G. Srinivasan, Operations Research: Principles And Applications, PHI Learning Pvt. Ltd, 2007.

[44] S. Arora, C. Lund, R. Motwani, M. Sudan και M. Szegedy, Proof Verification and the Hardness of Approximation Problems, Journal of the ACM, Vol. 45, No. 3, 1998.

[45] C. H. Papadimitriou και M. Yannakakis, The traveling salesman problem with distances one and two, Math. Oper. Res. 18, 1993.

[46] L. A. Wolsey, Heuristic analysis, linear programming and branch and bound. Combinatorial Optimization II., Berlin Heidelberg, 1980. 121-134.: Springer, 1980..

[47] D. B. Shmoys και D. P. Williamson., Analyzing the Held-Karp TSP bound: A monotonicity property with application., Information Processing Letters 35.6, 1990.

[48] M. Held και R. M. Karp, The traveling-salesman problem and minimum spanning trees, Operations Res. 18 1138-1162., 1970.

[49] M. Held και R. Karp, The traveling-salesman problem and minimum spanning trees:Part II, Math. Programming 1, 6-25, 1971.

[50] G. Reinelt, The Traveling Salesman Problem: Computational Solutions for TSP Applications, Lecture Notes in Computer Science 840, Berlin: Springer-Verlag, 1994.

[51] D. S. Johnson, L. A. McGeoch και E. E. Rothberg, Asymptotic experimental analysis for the Held-Karp traveling salesman bound. Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms., Society for Industrial and Applied Mathematics, 1996.

[52] D. R. Karger και S. Clifford, An Õ (n 2) algorithm for minimum cuts., Proceedings of the twenty-fifth annual ACM symposium on Theory of computing. ACM, , 1993.

[53] M.-Y. Kao, Encyclopedia of algorithms, Springer, 2008.

[54] D. Rosenkrantz, R. Stearns και P. Lewis, An analysis ofseveral heuristics for the traveling salesman problem., SIAM J. Comput.,6:563–581, 1977.

[55] D. Johnson και L. McGeoch, The Traveling Salesman Problem: A Case Study in Local Optimization, 1995.

[56] A. M. Frieze, Worst-case analysis of algorithms for travelling salesman problems, Methods of Operations Research 32, 1979.

[57] H. Ong και M. J.B., Worst-case analysis of two travelling salesman heuristics., Oper. Res. Lett. 2, 1984.

[58] G. U. Clarke και J. W. Wright, Scheduling of vehicles from a central depot to a number of delivery points., Operations research 12.4 :568-581., 1964.

[59] J. V. Bernhard Korte, Combinatorial Optimization: Theory and Algorithms, Springer, 2012.

[60] J. J. Bentley, Fast Algorithms for Geometric Traveling Salesman Problems, INFORMS Journal on Computing 4, 1992.

[61] F. Bock, An algorithm for solving ‘‘traveling-salesman’’ and related network optimization problems, unpublished manuscript associated withtalk presented at the 14th ORSA National Meeting, 1958.

[62] K. Helsgaun, General k-opt submoves for the Lin–Kernighan TSP, Springer and Mathematical Programming Society 2009, 2009.

[63] S. Arora, M. Grigni, D. Karger, P. Klein, Woloszyn και A., A Polynomial-Time Approximation Scheme for Weighted Planar Graph TSP., Society for Industrial and Applied Mathematics, , 1998.

[64] V. V. Vazirani, Approximation algorithms, Springer, 2001.

[65] R. Graham, L. Grotschel και Lovasz, Handbook of Combinatorics, τόμ. 2, MIT Press, 1995.

[66] Laporte G, European Journal of Operational Research 59, 1992.

[67] Z. Weixiong, Branch and Bound Search Algorithms and their Computational Complexity, 1996.

[68] J. W. Cook, In pursuit of the traveling salesman, Princeton University Press, 2012.

[69] V. Kolmogorov, Blossom V: A new implementation of a minimum cost perfect matching algorithm, Mathematical Programming Computation 1 , 2009.

[70] D. Johnson και C. H. Papadimitriou, Performance guarantees for heuristics, Lawler, 1985.

[71] S. Lin, Computer solutions of the traveling salesman problem, Bell Syst. Tech., 1965.

[72] D. P. Williamson και D. B. Shmoys, The Design of Approximation Algorithms, Cambridge University Press, 2011.