

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ****Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Κυβερνοασφάλεια και Επιστήμη Δεδομένων»****Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης</b> <b>Graph Neural Networks for Solving the Traveling Salesman Problem: Combining Mathematical Programming and Machine Learning</b>
Όνοματεπώνυμο Φοιτητή	<b>Ιωάννης Μούγιος</b>
Πατρώνυμο	<b>Παντελεήμων</b>
Αριθμός Μητρώου	<b>ΜΠΚΕΔ2327</b>
Επιβλέπων	<b>Κορωνάκος Γρηγόριος , Επίκουρος Καθηγητής</b>

Ημερομηνία Παράδοσης **Δεκέμβριος 2025**

**Τριμελής Εξεταστική Επιτροπή**

Κορωνάκος Γρηγόριος  
Επίκουρος Καθηγητής

Διονύσιος Σωτηρόπουλος  
Αναπληρωτής Καθηγητής

Δημήτρης Αποστόλου  
Καθηγητής

## Πίνακας Περιεχομένων

### 1 Εισαγωγή

- 1.1 Προσεγγιστικές Τεχνικές για το πρόβλημα του Περιοδεύοντος Πωλητή
- 1.2 Μηχανική Μάθηση για το πρόβλημα του Περιοδεύοντος Πωλητή
- 1.3 Συνεισφορές

### 2 Επίλυση του TSP με Γραμμικό Προγραμματισμό

- 2.1 Προσεγγίσεις Μαθηματικής Μοντελοποίησης
  - 2.1.1 Μαθηματική Διατύπωση του Μοντέλου Miller, Tucker και Zemlin
  - 2.1.2 Μαθηματική Διατύπωση του Μοντέλου Dantzig–Fulkerson–Johnson
- 2.2 Ακέραιος Γραμμικός Προγραμματισμός (ILP)
- 2.3 Ανασκόπηση Μεθόδων Αποκατάστασης Ακεραιότητας
  - 2.3.1 Μέθοδος της Εξάντλησης και υπολογιστικής πολυπλοκότητας
  - 2.3.2 Διακλάδωση και Φραγμός (Branch and Bound)
  - 2.3.3 Τέμνοντα Επίπεδα (Cutting Planes)

### 3 Εισαγωγή στα Νευρωνικά Δίκτυα Γράφων

- 3.1 Βασικές Αρχές Νευρωνικών Δικτύων
- 3.2 Θεμελιώδης Περιορισμοί και Γενίκευση της Συνέλιξης
  - 3.2.1 Κίνητρο για Γενίκευση της Συνέλιξης
  - 3.2.2 Συνέλιξη σε μη Ευκλείδειες δομές
  - 3.2.3 Δίκτυα Μηχανισμών Προσοχής
- 3.3 Κατηγορίες Εκπαίδευσης Νευρωνικών Δικτύων Γράφων
  - 3.3.1 Επιτηρούμενη και Ημι-επιτηρούμενη Μάθηση σε Γράφους
  - 3.3.2 Επαγωγική και Μεταγωγική Μάθηση σε Γράφους
- 3.4 Μοντελοποίηση του Προβλήματος του Πλανόδιου Πωλητή με GNN
  - 3.4.1 Αναπαράσταση του Συμμετρικού TSP ως Μη Κατευθυνόμενος Γράφος
  - 3.4.2 Νευρωνικό Δίκτυο για την Εκτίμηση της Λύσης του TSP

### 4 Μεθοδολογία – Πειραματισμός – Αξιολόγηση

- 4.1 Δημιουργία Βέλτιστων Λύσεων με Ακέραιο Γραμμικό Προγραμματισμό
- 4.2 Παραγωγή συνόλων δεδομένων
- 4.3 Εκπαίδευση Νευρωνικού Δικτύου και Παραμετροποίηση
- 4.4 Αξιολόγηση απόδοσης στο σύνολο ελέγχου
- 4.5 Συμπεράσματα

## Περίληψη

Η παρούσα διατριβή εξετάζει τη χρήση Νευρωνικών Δικτύων Γράφων (Graph Neural Networks - GNNs) για την επίλυση του κλασικού προβλήματος του Πλανόδιου Πωλητή (Traveling Salesman Problem - TSP). Το TSP αποτελεί ένα από τα σημαντικότερα προβλήματα συνδυαστικής βελτιστοποίησης και έχει χαρακτηριστεί ως NP-hard. Η μεθοδολογία που αναπτύχθηκε συνδυάζει κλασικές τεχνικές βελτιστοποίησης με σύγχρονες προσεγγίσεις μηχανικής μάθησης, αναδεικνύοντας τον ρόλο των Νευρωνικών Δικτύων Γράφων στη μοντελοποίηση δομών γράφων και στην επίλυση προβλημάτων συνδυαστικής φύσης. Ειδικότερα, η εφαρμογή Νευρωνικών Δικτύων Γράφων στο συγκεκριμένο πρόβλημα στοχεύει στην πρόβλεψη των ακμών που συνθέτουν τη βέλτιστη διαδρομή σε πλήρως συνδεδεμένους γράφους. Η εκπαίδευση και η αξιολόγηση μοντέλων Νευρωνικών Δικτύων Γράφων διεξάχθηκε με τη δημιουργία συνθετικών δεδομένων. Τα πειραματικά αποτελέσματα αναδεικνύουν ότι τα μοντέλα Νευρωνικών Δικτύων Γράφων μπορούν να εκπαιδευτούν αποτελεσματικά στα μοτίβα που αποτυπώνουν τις βέλτιστες λύσεις και να γενικεύσουν σε νέα παραδείγματα, γεγονός που επιβεβαιώνει τη δυναμική τους για την αντιμετώπιση πολύπλοκων προβλημάτων βελτιστοποίησης.

## Abstract

This dissertation examines the use of Graph Neural Networks (GNNs) for solving the classical Traveling Salesman Problem (TSP). The TSP is one of the well-known combinatorial optimization problems and is classified as NP-hard. The methodology developed in this work combines traditional optimization techniques with modern machine learning approaches, highlighting the role of Graph Neural Networks in modeling graph structures and addressing combinatorial problems. Specifically, the application of GNNs to the TSP aims at predicting the edges that form the optimal route in fully connected graphs. The training and evaluation of the Graph Neural Network models were conducted using synthetically generated data. The experimental results demonstrate that GNN models can be effectively trained to capture the patterns underlying optimal solutions and generalize to new instances, confirming their potential for handling complex optimization problems.

## 1 Εισαγωγή

Το Πρόβλημα του Περιπλανώμενου Πωλητή (Traveling Salesperson Problem - TSP) αποτελεί μία από τις πιο εντατικά μελετημένες προκλήσεις στον τομέα της συνδυαστικής βελτιστοποίησης, τόσο στην επιστήμη των υπολογιστών όσο και στην έρευνα επιχειρησιακών διαδικασιών. Στην ουσία, το πρόβλημα θέτει την εξής απλή ερώτηση. Δίνοντας μια λίστα πόλεων και τις αποστάσεις μεταξύ τους, ποια είναι η συντομότερη δυνατή διαδρομή που επισκέπτεται κάθε πόλη ακριβώς μία φορά και επιστρέφει στην αρχική πόλη; Παρά την απλή του διατύπωση, η υπολογιστική πολυπλοκότητα του TSP το καθιστά μια διαρκή πρόκληση και σημείο αναφοράς για την ανάπτυξη αλγορίθμων. Συγκεκριμένα, ο χώρος των εφικτών λύσεων του TSP αυξάνεται παραγοντικά με τον αριθμό των πόλεων ακόμη και στην συμμετρική εκδοχή του προβλήματος όπου απαιτείται η εξέταση  $\frac{(n-1)!}{2}$  πιθανών διαδρομών. Με τον χαρακτηρισμό συμμετρική εκδοχή ενός TSP δεχόμαστε την υπόθεση ότι το κόστος μετάβασης από το σημείο A στο σημείο B ισούται με το κόστος μετάβασης από το σημείο B στο σημείο A. Μολονότι η υπόθεση αυτή συνεπάγεται μείωση των μεταβλητών απόφασης στο μισό, ο χώρος των εφικτών λύσεων συνεχίζει να αυξάνεται παραγοντικά ως προς τον αριθμό των πόλεων, καθιστώντας την ακριβή επίλυση μέσω πλήρους εξάντλησης πρακτικά αδύνατη ακόμα και για σχετικά μικρούς αριθμούς πόλεων. Για να αποδώσουμε το μέγεθος του προβλήματος, στην συμμετρική μορφή του TSP, ο αριθμός των εφικτών διαδρομών για 10 πόλεις είναι 181.440, ενώ για 15 πόλεις αυξάνεται σε 43.589.145.600. Μέθοδοι γραμμικού προγραμματισμού, όπως η προσέγγιση με την μέθοδο των τεμνόντων επιπέδων και η δυναμική αποκοπή διαδρομών που υλοποιούμε στην παρούσα εργασία, προσφέρουν πιο αποδοτικά πλαίσια επίλυσης. Ωστόσο, ακόμα και αυτές οι μαθηματικές τεχνικές δεν ξεφεύγουν από τη βασική NP-hard φύση του προβλήματος. Όπως δείχνουν οι Vercesi & Buchanan (2024), το μοντέλο Dantzig–Fulkerson–Johnson (DFJ) παραμένει θεωρητικά NP-hard, ακόμη και όταν ο αριθμός των περιορισμών αποκοπής επιμέρους υπό-διαδρομών (*Subtour Elimination Constraints (SEC)*) αυξάνεται πολυωνυμικά ως προς το πλήθος των πόλεων  $n$ . Επομένως, ακόμη και όταν οι SEC εισάγονται δυναμικά κατά τη διάρκεια της επίλυσης — όπως στη μέθοδο των τεμνόντων επιπέδων — το DFJ διατηρεί τη θεωρητική του εκθετική πολυπλοκότητα.

### 1.1 Προσεγγιστικές Τεχνικές για το πρόβλημα του Περιοδεύοντος Πωλητή

Παρά την εντυπωσιακή πρόοδο των ακριβών μεθόδων βελτιστοποίησης, η αντιμετώπιση του TSP σε μεγαλύτερη κλίμακα εξακολουθεί να βασίζεται σε προσεγγιστικές τεχνικές όπως γενετικούς ή ευρετικούς αλγορίθμους, οι οποίοι επιτρέπουν την εξαγωγή λύσεων υψηλής ποιότητας μέσα σε περιορισμένο χρόνο. Οι τεχνικές αυτές δεν στοχεύουν στην εγγυημένη εύρεση της βέλτιστης λύσης, αλλά στην παραγωγή διαδρομών που βρίσκονται «κοντά» στο βέλτιστο, εκμεταλλευόμενες στοχαστικούς μηχανισμούς εξερεύνησης του χώρου λύσεων. Στον πυρήνα τους, βασίζονται στην παρατήρηση ότι το TSP παρουσιάζει πλούσια τοπική δομή, η οποία μπορεί να αξιοποιηθεί μέσω στρατηγικών τοπικής αναζήτησης και επαναληπτικής βελτίωσης (Chandra et al., 2022).

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

Μεταξύ των πλέον διαδεδομένων προσεγγιστικών αλγορίθμων για το TSP συγκαταλέγονται οι Γενετικοί Αλγόριθμοι (Genetic Algorithms – GA), οι Αλγόριθμοι Αποικίας Μυρμηγκιών (Ant Colony Optimization – ACO) και οι Αλγόριθμοι Σμήνους Σωματιδίων (Particle Swarm Optimization – PSO). Αν και προέρχονται από διαφορετικά θεωρητικά πλαίσια, όλοι επιδιώκουν την προσέγγιση ποιοτικών λύσεων μέσω μηχανισμών εξερεύνησης και σταδιακής βελτίωσης.

Οι Γενετικοί Αλγόριθμοι βασίζονται σε αρχές της εξελικτικής βιολογίας — εξού και ο όρος «γενετικοί». Υποψήφιος διαδρομές αναπαρίστανται ως «άτομα» ενός πληθυσμού, ο οποίος εξελίσσεται διαδοχικά μέσω τελεστών όπως ο ανασυνδυασμός (συνδυασμός τμημάτων από διαφορετικές λύσεις) και η μετάλλαξη (εισαγωγή μικρών τυχαίων αλλαγών στη σειρά των πόλεων). Μέσω της επαναληπτικής διαδικασίας επιλογής και αναπαραγωγής, ο αλγόριθμος ενισχύει σταδιακά τις πιο αποδοτικές λύσεις.

Οι Αλγόριθμοι Αποικίας Μυρμηγκιών εμπνέονται από τη συλλογική συμπεριφορά πραγματικών μυρμηγκιών, τα οποία αφήνουν φερομόνες για να σχηματίσουν μονοπάτια επικοινωνίας. Στο πλαίσιο του TSP, οι φερομόνες αποτελούν τεχνητά μεγέθη που τοποθετούνται στις ακμές του γραφήματος και λειτουργούν ως ενδείξεις προτίμησης. Όσο μια διαδρομή επιλέγεται από τον αλγόριθμο, η συγκέντρωση φερομόνης αυξάνεται, κατευθύνοντας έτσι την αναζήτηση προς υποσχόμενα μονοπάτια.

Τέλος, στους Αλγορίθμους Σμήνους Σωματιδίων, κάθε υποψήφια λύση αναπαρίσταται ως «σωματίδιο» που κινείται στον χώρο λύσεων. Η κίνησή του καθορίζεται από δύο βασικές πηγές πληροφορίας: την καλύτερη λύση που έχει εντοπίσει το ίδιο το σωματίδιο και την καλύτερη λύση που έχει εντοπίσει συνολικά το σμήνος. Έτσι, ο αλγόριθμος επιδιώκει ισορροπία μεταξύ εξερεύνησης νέων περιοχών και σύγκλισης προς ήδη γνωστά αποδοτικά μοτίβα.

Παρότι οι παραπάνω ευρετικοί αλγόριθμοι δεν εγγυώνται την εύρεση της βέλτιστης λύσης, η βιβλιογραφία καταγράφει πλήθος περιπτώσεων όπου επιτυγχάνουν ιδιαίτερα ανταγωνιστικά αποτελέσματα. Για παράδειγμα, στην εργασία του Gao (2020), μια βελτιωμένη εκδοχή του ACO πέτυχε μέση απόκλιση μόλις 2,07% από τη βέλτιστη λύση σε οκτώ περιπτώσεις δεδομένων που από τη βιβλιοθήκη TSPLIB<sup>1</sup>, ενώ η καλύτερη λύση ανά εκτέλεση απείχε μόλις 0,14% από το βέλτιστο. Αντίστοιχα, ο Liu (2014) ανέπτυξε έναν ενισχυμένο Γενετικό Αλγόριθμο που κατάφερε να εντοπίσει βέλτιστες λύσεις σε προβλήματα έως και 16.862 πόλεις. Στο πεδίο των PSO-προσεγγίσεων, η υβριδική μέθοδος PSO-LK των Goldberg et al. (2008) παρουσίασε εξαιρετικά υψηλή ακρίβεια, επιτυγχάνοντας μέσες αποκλίσεις της τάξης του 0,005% σε κλασικά TSP με 51 και 52 πόλεις σε σύνολα δεδομένων της βιβλιοθήκης TSPLIB. Ωστόσο, παρά τα εντυπωσιακά αυτά αποτελέσματα, οι ευρετικές μέθοδοι παρουσιάζουν σημαντικούς περιορισμούς: απαιτούν προσεκτικό και συχνά χρονοβόρο χειρισμό πλήθους υπερπαραμέτρων, εμφανίζουν αστάθεια από εκτέλεση σε εκτέλεση και, κυρίως, η απόδοσή τους δεν γενικεύεται εύκολα. Ένας αλγόριθμος που έχει ρυθμιστεί ώστε να αποδίδει καλά σε μικρά ή μεσαίου μεγέθους προβλήματα συχνά χάνει σε αποτελεσματικότητα όταν εφαρμόζεται σε μεγαλύτερα ή διαφορετικά τοπολογικά στιγμιότυπα του TSP.

## 1.2 Μηχανική Μάθηση για το πρόβλημα του Περιοδεύοντος Πωλητή

---

<sup>1</sup> Βιβλιοθήκη με σύνολα δεδομένων TSPLIB <https://github.com/mastqe/tsplib>

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

Η παρούσα εργασία εξετάζει αν τα νευρωνικά δίκτυα γράφων (Graph Neural Networks - GNNs) μπορούν να μάθουν τα μοτίβα των βέλτιστων λύσεων του TSP από μικρά παραδείγματα και να γενικεύσουν αυτή τη γνώση σε μεγαλύτερα προβλήματα. Η προσέγγιση μας βασίζεται σε τρεις βασικές παρατηρήσεις:

- **Δομική Ομοιότητα:** Μικρά και μεγάλα παραδείγματα του TSP παρουσιάζουν κοινά τοπολογικά χαρακτηριστικά στις βέλτιστες λύσεις τους.
- **Τοπικά Μοτίβα:** Οι βέλτιστες διαδρομές συχνά εμφανίζουν αναγνωρίσιμα τοπικά μοτίβα επιλογής ακμών, τα οποία τα GNN μπορούν να μάθουν.
- **Ανοχή σε υποβέλτιστες λύσεις:** Αν και δεν εγγυώνται βέλτιστες λύσεις, οι νευρωνικές προσεγγίσεις μπορεί να προσφέρουν κοντινές στη βέλτιστη λύσεις με σημαντικά μειωμένο χρόνο υπολογισμού.

Αυτό υλοποιείται μέσω ενός **υβριδικού πλαισίου μάθησης και αναζήτησης**, το οποίο αποτελείται από τα εξής στάδια:

- **Δημιουργία συνόλου εκπαίδευσης:** Λύνονται μικρά TSP instances (20–50 κόμβοι) με Ακέραιο Γραμμικό Προγραμματισμό (ILP) με δυναμική εισαγωγή περιορισμών αποκοπή υποκύκλων (SECs), ώστε να εξαχθούν οι βέλτιστες ακμές κάθε διαδρομής.
- **Εκπαίδευση Γραφικού Νευρωνικού Δικτύου:** Το GNN εκπαιδεύεται να προβλέπει την πιθανότητα επιλογής κάθε ακμής ως μέρος της βέλτιστης λύσης, χρησιμοποιώντας τα δεδομένα του προηγούμενου βήματος.
- **Καθοδηγούμενη αναζήτηση:** Οι προβλέψεις πιθανοτήτων του GNN χρησιμοποιούνται για να καθοδηγήσουν έναν beam search αλγόριθμο, ο οποίος κατασκευάζει εφικτές διαδρομές, δίνοντας προτεραιότητα σε ακμές με υψηλότερη πιθανότητα επιλογής.

### 1.3 Συνεισφορά

Η συνεισφορά της συγκεκριμένης εργασίας συνοψίζεται στα ακόλουθα:

- Υλοποίηση μοντέλων Νευρωνικών Δικτύων Γράφων παρέχοντας ελεύθερη πρόσβαση στον κώδικα.
- Υβριδική μεθοδολογία που δείχνει πώς η μηχανική μάθηση μπορεί να συμπληρώσει τον μαθηματικό προγραμματισμό στην επίλυση συνδυαστικών προβλημάτων.
- Εμπειρική τεκμηρίωση μέσω συστηματικής αξιολόγησης, η οποία αποδεικνύει ότι τα GNN συλλαμβάνουν ουσιαστικά μοτίβα λύσεων και θέτουν θεμέλια για περαιτέρω έρευνα.

## 2 Επίλυση του TSP με Γραμμικό Προγραμματισμό

Το Πρόβλημα του Περιπλανώμενου Πωλητή αποτελεί ένα κλασικό και ευρέως μελετημένο πρόβλημα ακέραιου προγραμματισμού και μπορεί να διατυπωθεί ως εξής: Δίνεται ένα σύνολο πόλεων και οι αποστάσεις μεταξύ κάθε πιθανού ζεύγους αυτών. Στόχος είναι να βρεθεί η συντομότερη διαδρομή που:

- επισκέπτεται κάθε πόλη ακριβώς μία φορά (1)
- αποτελεί έναν ενιαίο κύκλο που περικλείει όλες τις πόλεις (2)

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

- και ελαχιστοποιεί το συνολικό μήκος ή κόστος της διαδρομής (3)

## 2.1 Προσεγγίσεις Μαθηματικής Μοντελοποίησης

Στη βιβλιογραφία έχουν αναπτυχθεί διάφορες μαθηματικές διατυπώσεις για τη μοντελοποίηση του Προβλήματος του Πλανόδιου Πωλητή, με δύο από τις πιο διαδεδομένες να είναι εκείνες των Miller–Tucker–Zemlin (MTZ) (Miller, Tucker & Zemlin, 1960) και Dantzig–Fulkerson–Johnson (DFJ) (Dantzig, Fulkerson & Johnson, 1954). Αμφότερες οι προσεγγίσεις ξεκινούν εισάγοντας δυαδικές μεταβλητές  $x_{ij}$  οι οποίες λαμβάνουν τιμή 1 αν η ακμή από την πόλη  $i$  προς την πόλη  $j$  περιλαμβάνεται στη λύση, και 0 διαφορετικά. Με αυτόν τον τρόπο, το πρόβλημα μετασχηματίζεται σε ένα πρόβλημα ακέραιου γραμμικού προγραμματισμού, όπου στόχος είναι η ελαχιστοποίηση του συνολικού κόστους των επιλεγμένων διαδρομών. Ωστόσο, παρά τον κοινό αυτό κορμό, οι δύο μοντελοποιήσεις διαφέρουν σημαντικά στον τρόπο με τον οποίο διασφαλίζουν την εγκυρότητα της δεύτερης συνθήκης του προβλήματος, δηλαδή την ύπαρξη ενός και μόνο ενιαίου κύκλου που να περιλαμβάνει όλες τις πόλεις.

Σε αυτό το σημείο είναι κρίσιμο να αποσαφηνιστεί ο όρος «επιμέρους κύκλοι», καθώς πρόκειται να χρησιμοποιηθεί συστηματικά στη συνέχεια της εργασίας. Ως επιμέρους κύκλοι ορίζονται ανεξάρτητοι κλειστοί κύκλοι που ενδέχεται να προκύψουν σε μία λύση. Αυτοί οι κύκλοι μπορεί να ικανοποιούν τη συνθήκη (1) —δηλαδή κάθε πόλη να επισκέπτεται μία μόνο φορά— αλλά παραβιάζουν τη συνθήκη (2), αφού το σύνολο των ακμών δεν σχηματίζει ένα συνεκτικό μονοπάτι που διατρέχει όλες τις πόλεις. Στο πλαίσιο της θεωρίας γράφων, η παρουσία επιμέρους κύκλων συνεπάγεται ότι η επιλεγμένη υποδομή ακμών αντιστοιχεί σε μη συνεκτικό γράφο.

Η προσέγγιση MTZ επιχειρεί να εξαλείψει τις επιμέρους διαδρομές εισάγοντας βοηθητικές μεταβλητές  $u_i$  και  $u_j$ , οι οποίες εκφράζουν τις «θέσεις» των πόλεων  $i$  και  $j$  εντός της συνολικής διαδρομής. Μεταξύ αυτών των μεταβλητών θέσεων επιβάλλεται ο ανισοτικός περιορισμός της μορφής:

$$u_i - u_j + 1 \leq (n - 1)(1 - x_{ij}) \quad \forall i, j \in [2, n] \text{ και } i \neq j$$

οι οποίες διασφαλίζουν την αδυναμία σχηματισμού κυκλικών ακολουθιών όπως  $u_1 < u_2 < \dots < u_i$ , οι οποίες θα οδηγούσαν σε επιμέρους κύκλους. Οι τιμές των  $u_i$  περιορίζονται εντός ενός καθορισμένου εύρους, εξασφαλίζοντας έτσι ότι δεν μπορεί να σχηματιστεί κύκλος σε υποσύνολο πόλεων  $S$ .

Αντίθετα, η προσέγγιση DFJ βασίζεται περισσότερο στη συνδεσιμότητα του γράφου. Σε αυτήν, η ύπαρξη ενός ενιαίου κύκλου που διατρέχει όλες τις πόλεις αποτυπώνεται στο γεγονός ότι το πλήθος των επιλεγμένων ακμών πρέπει να είναι ίσο με το πλήθος των πόλεων. Η απουσία συνδεσιμότητας αντιμετωπίζεται μέσω της κατασκευής περιορισμών-επιπέδων που περιορίζουν τον χώρο των εφικτών λύσεων (*cutting planes*), αποτρέποντας οποιοδήποτε κυκλικό μονοπάτι μήκους μικρότερου από τον αριθμό των πόλεων.

Συνοψίζοντας, και οι δύο προσεγγίσεις επιδιώκουν τη δημιουργία ενός μοναδικού, συνεκτικού κύκλου. Η μέθοδος MTZ επιτυγχάνει αυτόν τον στόχο με τη χρήση επιπλέον μεταβλητών και περιορισμών τύπου θέσης, ενώ η μέθοδος DFJ υιοθετεί μια πιο δυναμική προσέγγιση, εξαλείφοντας τις ανεπιθύμητες λύσεις με βάση το μήκος του κυκλικού μονοπατιού που

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

αντιπροσωπεύει την λύση. Οι διαφορές αυτές έχουν σημαντικές επιπτώσεις στην αποδοτικότητα και την πολυπλοκότητα του μοντέλου, όπως θα παρουσιαστεί αναλυτικότερα στη συνέχεια της εργασίας.

### 2.1.1 Μαθηματική Διατύπωση του Μοντέλου Miller, Tucker και Zemlin

Μία από τις πρώτες και πιο κλασικές μαθηματικές διατυπώσεις του προβλήματος του Πλανόδιου Πωλητή προτάθηκε από τους Miller, Tucker και Zemlin (MTZ). Η συγκεκριμένη προσέγγιση βασίζεται στην ιδέα ότι μπορούμε να περιγράψουμε το πρόβλημα μέσω ενός μικρού αριθμού γραμμικών περιορισμών και δυαδικών μεταβλητών, με σκοπό να αποκλείσουμε τους λεγόμενους υποκύκλους χωρίς να χρειαστεί να τους καταγράψουμε ρητά.

Ας θεωρήσουμε ότι έχουμε ένα πλήρως συνδεδεμένο γράφημα  $G = (V, E)$  όπου κάθε κόμβος  $i \in V$  αντιπροσωπεύει μία πόλη και κάθε ακμή  $(i, j) \in E$  έχει ένα αντίστοιχο κόστος μετακίνησης  $c_{ij}$ . Ο στόχος είναι να επισκεφθούμε κάθε πόλη ακριβώς μία φορά και να επιστρέψουμε στο σημείο εκκίνησης, ελαχιστοποιώντας το συνολικό κόστος. Στο μοντέλο MTZ χρησιμοποιούνται οι εξής μεταβλητές:

- $x_{ij} \in \{0, 1\}$ : η μεταβλητή λαμβάνει την τιμή 1 αν ο πωλητής μετακινηθεί από την πόλη  $i$  στην πόλη  $j$ , αλλιώς 0.
- $u_i \in \mathcal{R}$ : βοηθητικές μεταβλητές που εισάγονται για την εξάλειψη των επιμέρους κύκλων (μόνο για κόμβους  $i=2, \dots, n$ ).

Το μοντέλο MTZ διατυπώνεται ως:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (1.1)$$

υπό τους περιορισμούς:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall i \in \{1, \dots, n\} \quad (1.2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall j \in \{1, \dots, n\} \quad (1.3)$$

$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}) \quad \forall i, j \in [2, n] \text{ και } i \neq j \quad (1.4)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in 1, \dots, n, i \neq j \quad (1.5)$$

Κάθε σύνολο περιορισμών (1.2) και (1.3) αποτελείται από  $n$  περιορισμούς, οι οποίοι επιβάλλουν τα εξής: κάθε πόλη να αποτελεί ακριβώς μία φορά πόλη αναχώρησης και μία φορά πόλη άφιξης, αντίστοιχα. Δηλαδή, από κάθε πόλη περνά ακριβώς μία διαδρομή και προς κάθε πόλη καταλήγει ακριβώς μία διαδρομή. Ο τελευταίος περιορισμός (1.4) έχει ως στόχο να αποτρέψει την ύπαρξη επιμέρους διαδρομών — δηλαδή, κύκλων που δεν περιλαμβάνουν όλες τις πόλεις. Γι' αυτόν τον λόγο, οι δείκτες σε αυτόν τον περιορισμό ξεκινούν από το 2 και φτάνουν έως το  $n-1$ .

Αναλυτικότερα:

- Η πόλη 1 επιλέγεται ως αρχική πόλη εκκίνησης της διαδρομής. Δεν χρειάζεται να συμμετέχει στους περιορισμούς υποκύκλων (1.4), καθώς οι υπόλοιπες πόλεις  $2, \dots, n$  καθορίζουν τη σειρά επισκέψεων και η πόλη 1 θα αποτελέσει τη φυσική επιστροφή στο τέλος της διαδρομής.

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

- Οι βοηθητικές μεταβλητές  $u_i$  ορίζουν τη θέση επίσκεψης κάθε πόλης στη διαδρομή.
- Με αυτόν τον περιορισμό διασφαλίζουμε ότι η σειρά των επισκέψεων δεν σχηματίζει επιμέρους κύκλους, επιτρέποντας όμως την επιστροφή από την τελευταία πόλη πίσω στην αρχική.

Για παράδειγμα, αν έχουμε 4 πόλεις, ο περιορισμός αυτός επιτρέπει μια διαδρομή της μορφής  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ , δηλαδή επιστροφή από την τελευταία πόλη στην πρώτη, χωρίς να επιτρέψει κύκλους όπως  $2 \rightarrow 3 \rightarrow 2$  που δεν περνούν από όλες τις πόλεις.

### 2.1.2 Μαθηματική Διατύπωση του Μοντέλου Dantzig–Fulkerson–Johnson

Η προσέγγιση του Dantzig–Fulkerson–Johnson (DFJ) αποτελεί μία από τις κλασικές και ακριβείς μαθηματικές διατυπώσεις του προβλήματος του Πλανόδιου Πωλητή. Όπως και στην περίπτωση της MTZ, το πλήρως συνδεδεμένο γράφημα  $G=(V,E)$  έχει κόμβους  $i \in V$  που αντιστοιχούν σε πόλεις, και ακμές  $(i, j) \in E$  με αντίστοιχο κόστος μετακίνησης  $c_{ij}$ . Οι δυαδικές μεταβλητές ορίζονται ως εξής:

- $x_{ij} \in \{0,1\}$ : είναι 1 αν ο πωλητής μετακινηθεί από την πόλη  $i$  στην πόλη  $j$ , αλλιώς 0.

Το μοντέλο διατυπώνεται ως:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2.1)$$

Υπό τους περιορισμούς:

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in \{1, \dots, n\} \quad (2.2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in \{1, \dots, n\} \quad (2.3)$$

$$\sum_{i \in Q} \sum_{j \in Q, j \neq i} x_{ij} \leq |Q| - 1 \quad \forall Q \subsetneq \{1, \dots, n\}, \quad |Q| \geq 2 \quad (2.4)$$

Οι περιορισμοί (2.2) και (2.3) διασφαλίζουν από κοινού ότι κάθε κόμβος έχει ακριβώς μία εισερχόμενη και μία εξερχόμενη ακμή, ενώ ο περιορισμός (2.4) αποτρέπει τη δημιουργία επιμέρους διαδρομών (subtours) μέσω της απαρίθμησης όλων των πιθανών υποσυνόλων κόμβων που θα μπορούσαν να σχηματίσουν υποκύκλους.

Η διαφορά με το μοντέλο MTZ είναι ο τρόπος αντιμετώπισης των επιμέρους κύκλων, ενώ η MTZ χρησιμοποιεί βοηθητικές μεταβλητές για τη σειρά επίσκεψης, το μοντέλο DFJ τους αποτρέπει ρητά επιβάλλοντας το μήκος του κύκλου που αντιπροσωπεύει την βέλτιστη λύση να ισούται με τον αριθμό των πόλεων που πρέπει να επισκεφτεί ο πωλητής. Ωστόσο, ακόμη και σε αυτή την προσέγγιση, ο αριθμός των περιορισμών που απαιτούνται για την αποτροπή όλων των επιμέρους κύκλων αυξάνεται εκθετικά με τον αριθμό των πόλεων. Συγκεκριμένα, για ένα πρόβλημα με  $n$  πόλεις, θα πρέπει να εισαχθούν περιορισμοί για όλους τους δυνατούς κύκλους που περιλαμβάνουν από 2 έως  $n-1$  πόλεις, δηλαδή για κάθε υποσύνολο κόμβων που ενδέχεται να σχηματίσει ανεξάρτητο κύκλο. Για παράδειγμα, σε ένα πρόβλημα με 5 πόλεις, απαιτείται η αποτροπή όλων των δυνατών κύκλων με 2, 3 και 4 κόμβους, γεγονός που οδηγεί σε έναν εκθετικά αυξανόμενο αριθμό περιορισμών καθώς μεγαλώνει το πλήθος των πόλεων.

Στην παρούσα εργασία, χρησιμοποιήθηκε η DFJ μοντελοποίηση για την επίλυση προβλημάτων TSP, τα οποία αποτέλεσαν τη βάση εκπαίδευσης του Νευρωνικού Δικτύου Γράφων. Λεπτομέρειες

σχετικά με την υλοποίηση καθώς και την αρχιτεκτονική και τα χαρακτηριστικά του νευρωνικού μοντέλου αναλύονται παρακάτω.

## 2.2 Ακέραιος Γραμμικός Προγραμματισμός

Ο Ακέραιος Γραμμικός Προγραμματισμός (Integer Linear Programming - ILP) αποτελεί μια υποκατηγορία του γραμμικού προγραμματισμού, στην οποία επιλύεται ένα πρόβλημα βελτιστοποίησης με γραμμική αντικειμενική συνάρτηση και γραμμικούς περιορισμούς, με τη διαφορά ότι ορισμένες ή όλες οι μεταβλητές απόφασης πρέπει να λάβουν ακέραιες τιμές.

Αν όλες οι μεταβλητές είναι ακέραιες, τότε το πρόβλημα χαρακτηρίζεται ως αμιγώς ακέραιος γραμμικός προγραμματισμός (Pure ILP ή PILP), ενώ αν κάποιες μεταβλητές επιτρέπεται να λάβουν συνεχείς τιμές, τότε πρόκειται για μικτό ακέραιο γραμμικό προγραμματισμό (Mixed-Integer Linear Programming - MILP).

Μια ειδική περίπτωση του ILP είναι ο δυαδικός ακέραιος προγραμματισμός, κατά τον οποίο οι μεταβλητές μπορούν να λάβουν μόνο τις τιμές 0 ή 1. Η μορφή αυτή είναι ιδιαίτερως χρήσιμη σε προβλήματα όπου απαιτούνται δυαδικές αποφάσεις, όπως επιλογή/απόρριψη, ενεργοποίηση/απενεργοποίηση, ή ύπαρξη/απουσία σχέσης μεταξύ οντοτήτων. Το Πρόβλημα του Πλανόδιου Πωλητή, το οποίο αποτελεί το αντικείμενο της παρούσας εργασίας, ανήκει ακριβώς σε αυτήν την κατηγορία.

Η γενική μαθηματική διατύπωση ενός ILP προβλήματος είναι η εξής:

$$\min_{\{x\}} c^T x$$

Υπό τους περιορισμούς:  $Ax \leq b$

$$x_i \in \mathbb{Z}, \forall i \in I$$

$$x_j \in \mathbb{R}, \forall j \notin I$$

Σε περίπτωση δυαδικού ILP, ισχύει:

$$x_i \in \{0,1\}, \forall i \in I$$

Η κύρια πρόκληση του ακέραιου γραμμικού προγραμματισμού έγκειται στο ότι ο περιορισμός της ακεραιότητας κάποιων μεταβλητών απόφασης καθιστά το σύνολο εφικτών λύσεων μη κυρτό. Στον κλασικό γραμμικό προγραμματισμό, οι λύσεις βρίσκονται σε ένα κυρτό πολύεδρο – δηλαδή, οποιοδήποτε σημείο μεταξύ δύο εφικτών λύσεων είναι επίσης εφικτό. Ωστόσο, όταν απαιτείται οι λύσεις να είναι ακέραιες, το σύνολο εφικτών λύσεων αποτελείται από διακριτά σημεία, γεγονός που αναιρεί την ιδιότητα της κυρτότητας.

Πρακτικά, αυτό σημαίνει ότι αλγόριθμοι όπως ο Simplex ή οι μέθοδοι εσωτερικού σημείου (Interior Point Methods), παρότι επιλύουν αποδοτικά προβλήματα γραμμικού προγραμματισμού, δεν εγγυώνται ότι η λύση τους θα ανήκει στο σύνολο των εφικτών λύσεων του αρχικού προβλήματος ILP. Ο λόγος είναι ότι το αποτέλεσμα αυτών των μεθόδων ενδέχεται να είναι μερικώς ή πλήρως μη ακέραιο, καθώς επιλύουν το συνεχές χαλαρωμένο πρόβλημα και όχι το διακριτό.

Η τυπική διαδικασία επίλυσης ενός ILP προβλήματος ακολουθεί επομένως δύο στάδια:

1. Χαλάρωση (Relaxation): Επίλυση της εκδοχής του προβλήματος (LP Relaxation), όπου οι περιορισμοί ακεραιότητας αγνοούνται. Το πρόβλημα επιλύεται μέσω καθιερωμένων αλγορίθμων γραμμικού προγραμματισμού (π.χ. Simplex, Interior Point), και η βέλτιστη τιμή

του παρέχει ένα φράγμα (κάτω φράγμα σε προβλήματα ελαχιστοποίησης ή άνω φράγμα σε προβλήματα μεγιστοποίησης) για το αρχικό ακέραιο πρόβλημα.

2. Αποκατάσταση της Ακεραιότητας (Restoring Integrality): Εφόσον η λύση του LP Relaxation δεν ικανοποιεί τους ακέραιους περιορισμούς, απαιτείται περαιτέρω διαδικασία ώστε να εντοπιστεί η βέλτιστη λύση εντός του συνόλου των ακέραιων εφικτών σημείων.

Έστω το γενικό πρόβλημα Ακέραιου Γραμμικού Προγραμματισμού (ILP):

$$\min \{ c^T x : Ax \leq b, x_i \in \mathbb{Z}, \forall x_i \in I, x_j \in \mathbb{R}, x_j \quad \forall j \notin I \}$$

Ορίζεται το πολύεδρο εφικτών λύσεων του LP Relaxation:

$$P = \{ x \in \mathbb{R}^n : Ax \leq b \}$$

Το σύνολο των ακέραιων εφικτών λύσεων είναι:

$$X = P \cap \mathbb{Z}^n$$

Η κυρτή θήκη των ακέραιων λύσεων είναι:

$$P_x = \text{conv}(X)$$

Τότε ισχύει:

$$X \subseteq P_x \subseteq P$$

και για τη βέλτιστη τιμή ενός προβλήματος ακέραιου προγραμματισμού:

$$\min\{c^T x : x \in P\} \leq \min\{c^T x : x \in X\}$$

$$\min\{c^T x : x \in P_x\} \leq \min\{c^T x : x \in X\}$$

Αυτό σημαίνει ότι το LP Relaxation παρέχει:

- ένα κάτω φράγμα (σε προβλήματα ελαχιστοποίησης), και
- μία καλή προσέγγιση της γεωμετρίας του συνόλου των ακέραιων λύσεων.

Τα παραπάνω αποτελούν αναγκαία στοιχεία για όλες τις σύγχρονες μεθόδους αποκατάστασης της ακεραιότητας. Είτε πρόκειται για δέντρα διακλάδωσης και φραγμών (Branch and Bound) είτε για τέμνοντα επίπεδα (Cutting Planes), οι μέθοδοι αυτές βασίζονται στα αποτελέσματα του LP Relaxation τόσο για την παραγωγή φραγμάτων όσο και για την παραγωγή πρόσθετων περιορισμών ή την επιλογή υπό-προβλημάτων.

## 2.3 Ανασκόπηση Μεθόδων Αποκατάστασης Ακεραιότητας

Παρόλο που το βήμα της αποκατάστασης της ακεραιότητας είναι ένα φαινομενικά απλό βήμα, στην πραγματικότητα αποτελεί κρίσιμο και πολύπλοκο στάδιο, καθώς μετατρέπει προβλήματα που αρχικά μπορεί να έχουν πολυωνυμική χρονική πολυπλοκότητα σε NP-hard προβλήματα. Για το λόγο αυτό, η εύρεση αποδοτικών μεθόδων για την αποκατάσταση της ακεραιότητας είναι θεμελιώδης για την πρακτική επίλυση προβλημάτων Ακέραιου Γραμμικού Προγραμματισμού.

### 2.3.1 Μέθοδος της Πλήρους Απαρίθμησης

Η πλέον άμεση —και θεωρητικά ακριβής— προσέγγιση για την επίλυση ενός προβλήματος Ακέραιου Γραμμικού Προγραμματισμού (ILP) είναι η πλήρης απαρίθμηση όλων των δυνατών ακέραιων λύσεων και η επιλογή της βέλτιστης μεταξύ αυτών. Ωστόσο, η μέθοδος αυτή καθίσταται

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

υπολογιστικά μη εφαρμόσιμη στην πράξη, εξαιτίας της παραγοντικής αύξησης του χώρου των εφικτών λύσεων. Η υπολογιστική πολυπλοκότητα είναι συνάρτηση του πλήθους των τιμών που μπορεί να λάβει κάθε ακέραια μεταβλητή, αλλά κυρίως από το πλήθος των μεταβλητών που υπόκεινται σε ακέραιο περιορισμό. Αναλυτικότερα αν ένα πρόβλημα ακέραιου προγραμματισμού έχει:

- $n$  ακέραιες μεταβλητές
- και κάθε μεταβλητή  $x_i$  μπορεί να πάρει  $k_i$  διακριτές τιμές

Το πλήθος των λύσεων είναι:

$$\prod_{(i=1)}^n k_i$$

Ακόμη και σε προβλήματα δυαδικού προγραμματισμού, όπου κάθε μεταβλητή μπορεί να λάβει μόνο τις τιμές 0 ή 1, το πλήθος των δυνατών λύσεων αυξάνεται ραγδαία με τον αριθμό των ακέραιων μεταβλητών. Για παράδειγμα, σε ένα συμμετρικό TSP με  $n$  πόλεις, το πλήθος των πιθανών διατάξεων είναι  $(n-1)!/2$ , αριθμός που καθιστά την εξαντλητική απαρίθμηση απαγορευτική ήδη για  $n > 15$ . Για τον λόγο αυτό, έχουν αναπτυχθεί αποδοτικότερες μέθοδοι αποκατάστασης της ακεραιότητας, οι οποίες βασίζονται στην προηγούμενη επίλυση του αντίστοιχου συνεχούς γραμμικού προβλήματος (LP Relaxation). Οι μέθοδοι αυτές αξιοποιούν τη λύση του συνεχούς προβλήματος είτε για την κατασκευή φραγμάτων είτε για την καθοδήγηση της αναζήτησης κατάλληλων πρόσθετων περιορισμών. Δύο από τις σημαντικότερες τεχνικές είναι:

- η μέθοδος Διακλάδωσης και Φραγμού (Branch and Bound) και
- η μέθοδος Τεμνόντων Επιπέδων (Cutting Planes).

### 2.3.2 Μέθοδος Διακλάδωσης και Φραγμού

Η μέθοδος Διακλάδωσης και Φραγμού (Branch and Bound) αποτελεί μία από τις πιο διαδεδομένες τεχνικές για την επίλυση προβλημάτων ακέραιου γραμμικού προγραμματισμού. Η βασική ιδέα της βασίζεται στην επαναληπτική και αναδρομική επίλυση ολοένα και λιγότερο χαλαρωμένων υποπροβλημάτων, δηλαδή προβλημάτων που προκύπτουν όταν αγνοούνται προσωρινά οι περιορισμοί ακεραιότητας.

Υπό το πλαίσιο της μεθόδου, ο όρος υποπρόβλημα αναφέρεται σε ένα νέο πρόβλημα που προκύπτει από το αρχικά χαλαρωμένο γραμμικό πρόβλημα μέσω της προσθετικής προσθήκης ενός επιπλέον περιορισμού. Κατά τη διαδικασία της διακλάδωσης (branching), δημιουργούνται τέτοια υποπροβλήματα, με στόχο τη σταδιακή επιβολή της ακεραιότητας σε όλες τις μεταβλητές απόφασης. Κάθε υποπρόβλημα ορίζει έναν μικρότερο υπόχωρο εφικτών λύσεων του αρχικού προβλήματος και επιλύεται ως ένα νέο χαλαρωμένο γραμμικό πρόβλημα. Στην περίπτωση μη ακέραιας λύσης, η λύση του χαλαρωμένου προβλήματος παρέχει φράγματα (bounds) που καθοδηγούν τη διαδικασία αναζήτησης. Σε προβλήματα ελαχιστοποίησης, η καλύτερη ακέραιη λύση που έχει εντοπιστεί μέχρι εκείνο το σημείο παρέχει ένα άνω φράγμα. Το αντίστροφο ισχύει σε προβλήματα μεγιστοποίησης. Αν η λύση του χαλαρωμένου προβλήματος είναι ακέραια, τότε αποτελεί βέλτιστη λύση και η διαδικασία ολοκληρώνεται. Διαφορετικά, επιλέγεται μια μεταβλητή με μη ακέραιη τιμή και δημιουργούνται δύο νέα υποπροβλήματα με τους παρακάτω περιορισμούς:

- $x_i \leq \lfloor x_i^* \rfloor$
- $x_j \geq \lceil x_j^* \rceil$

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

Η διαδικασία επαναλαμβάνεται αναδρομικά έως ότου ισχύσει ένα από τα παρακάτω:

- Το υποπρόβλημα είναι ανέφικτο.
- Η βέλτιστη λύση του υποπροβλήματος είναι χειρότερη από το άνω φράγμα, άρα μπορεί να απορριφθεί.
- Η λύση είναι ακέραια και βελτιώνει το άνω φράγμα.

Η αποτελεσματικότητα της μεθόδου έγκειται στην εισαγωγή νέων περιορισμών που αποκλείουν μεγάλες περιοχές του εφικτού χώρου στις οποίες οι μεταβλητές λαμβάνουν συνεχείς τιμές. Η εύρεση της βέλτιστης ακέραιας λύσης επιτυγχάνεται, περιορίζοντας το εύρος αναζήτησης μόνο στις υποπεριπτώσεις που παρέχουν προοπτική βελτίωσης. Τα παραπάνω μπορούν να γίνουν κατανοητά με την χρήση του παρακάτω παραδείγματος. Έστω το πρόβλημα ελαχιστοποίησης:

$$\min_{\{x_1, x_2\}} f(x_1, x_2) = 7x_1 + 8x_2 \quad (3.1)$$

υπό τους περιορισμούς:

$$-x_1 - 3x_2 \leq 14 \quad (3.2)$$

$$-12x_1 - 5x_2 \leq -28 \quad (3.3)$$

$$x_1 + 2x_2 \leq 10 \quad (3.4)$$

$$x_1, x_2 \geq 0 \quad (3.5)$$

$$x_1, x_2 \in \mathbb{Z} \quad (3.6)$$

Αρχικά, επιλύεται το χαλαρωμένο πρόβλημα παραλείποντας τον περιορισμό ακεραιότητας (3.6). Η βέλτιστη λύση είναι:

$$(x_1, x_2) = (2.3334, 0)$$

Με τιμή αντικειμενικής συνάρτησης:

$$f(x_1, x_2) = 16.331$$

Εφόσον η μεταβλητή  $x_1$  δεν έχει λάβει ακέραια τιμή, ξεκινά η διαδικασία διακλάδωσης με αυτήν και την εισαγωγή των ακόλουθων περιορισμών στη διαδικασία:

- $x_1 \leq 2$
- $x_1 \geq 3$

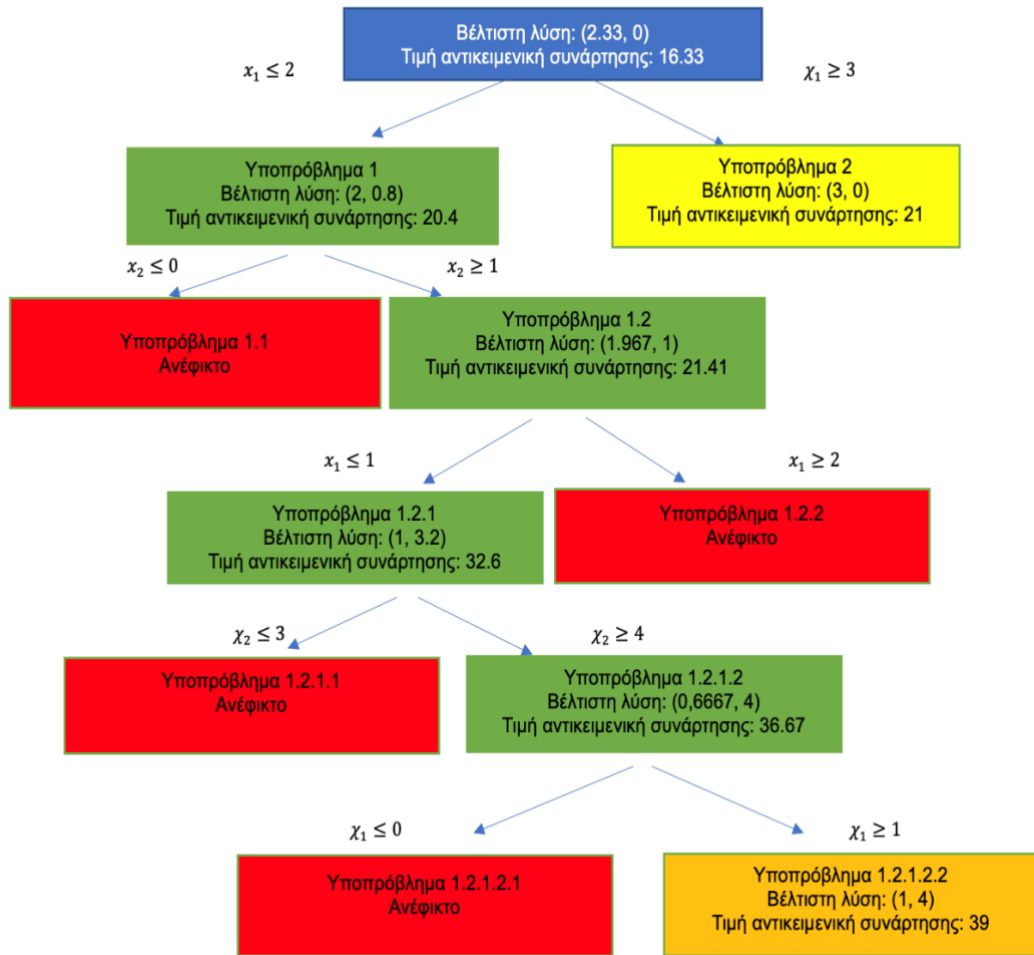
Στο αντίστοιχο δέντρο διακλάδωσης, οι καταστάσεις απεικονίζονται χρωματικά:

Πράσινο: συνεχίζεται η διαδικασία.

Κόκκινο: τερματισμός λόγω ανεφικτότητας ή χειρότερης λύσης.

Πορτοκαλί: εφικτή ακέραια λύση, αποτελεί άνω φράγμα

Κίτρινο: τελική βέλτιστη λύση.



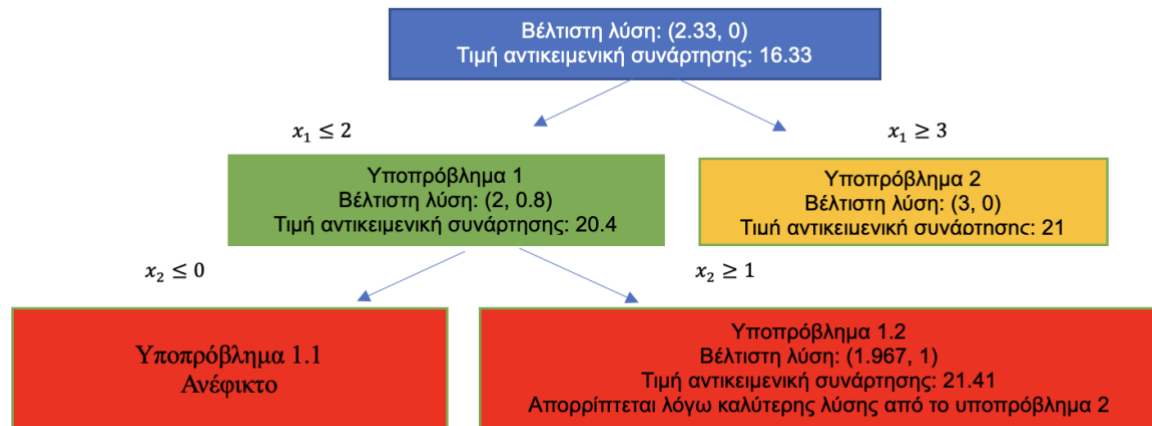
**Σχήμα 2.1: Δέντρο διακλάδωσης για την επίλυση του προβλήματος με τη μέθοδο Branch and Bound**

Αξίζει να σημειωθεί ότι η διαδικασία της διακλάδωσης εκτείνεται καθ' ύψος, με την έννοια ότι πριν εξεταστεί το επόμενο υποπρόβλημα στο ίδιο επίπεδο του δέντρου (οριζόντια διακλάδωση), θα πρέπει η αναδρομική επίλυση του προηγούμενου κλάδου να έχει πλήρως ολοκληρωθεί. Η ολοκλήρωση αυτή μπορεί να προκύψει είτε λόγω ανεφικτότητας του υποπροβλήματος είτε λόγω εύρεσης ακέραιης λύσης — ανεξαρτήτως αν αυτή γίνεται αποδεκτή ή απορρίπτεται με βάση τα υπάρχοντα φράγματα. Το χαρακτηριστικό αυτό διασφαλίζει ότι η εξερεύνηση του δέντρου πραγματοποιείται κατά βάθος (depth-first search), επιτρέποντας την έγκαιρη ενημέρωση του άνω φράγματος σε περίπτωση εύρεσης έγκυρων ακέραιων λύσεων.

Ωστόσο, η ταχύτητα σύγκλισης προς τη βέλτιστη λύση επηρεάζεται σημαντικά και από τη σειρά με την οποία επιλύονται τα υποπροβλήματα σε κάθε επίπεδο. Στο συγκεκριμένο παράδειγμα, αν η επίλυση είχε ξεκινήσει από το υποπρόβλημα 2, ο αλγόριθμος θα μπορούσε να είχε τερματίσει νωρίτερα. Η λύση του υποπροβλήματος 2 είναι ακέραιη και θα μπορούσε εξαρχής να λειτουργήσει

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

ως άνω φράγμα για το αρχικό πρόβλημα. Η ύπαρξη ενός ισχυρού άνω φράγματος ήδη από τα πρώτα στάδια της διαδικασίας θα είχε οδηγήσει στην αποφυγή επίλυσης πολλών υποπροβλημάτων, επιταχύνοντας σημαντικά τη σύγκλιση προς τη βέλτιστη λύση. Παρακάτω παρατίθεται το δέντρο διακλάδωσης σε περίπτωση που ξεκινούσαμε τον αλγόριθμο από την επίλυση του υποπροβλήματος 2 αντί του υποπροβλήματος 1.



**Σχήμα 2.2:** Δέντρο διακλάδωσης όταν προηγείται το υποπρόβλημα 2 κατά την διαδικασία επίλυσης.

Στο δέντρο διακλάδωσης, ξεκινώντας από την επίλυση του υποπροβλήματος 2, προκύπτει μια εφικτή λύση με τιμή αντικειμενικής συνάρτησης ίση με 21. Η τιμή αυτή λειτουργεί ως άνω φράγμα για το πρόβλημα. Στη συνέχεια, όταν εξετάζουμε το υποπρόβλημα 1.2 και διαπιστώνουμε ότι η τιμή της αντικειμενικής συνάρτησης είναι μεγαλύτερη από 21, ο αλγόριθμος τερματίζει, καθώς δεν μπορεί να βρεθεί καλύτερη λύση.

Η μέθοδος του Διακλάδωσης και Φραγμών (Branch and Bound) επιδιώκει τη βέλτιστη ακέραια λύση μέσω του συστηματικού περιορισμού του χώρου των εφικτών λύσεων, αξιοποιώντας μια αναδρομική προσέγγιση τύπου «διαίρει και βασίλευε». Με τη διαδοχική διαίρεση του προβλήματος σε υποπροβλήματα και την αξιολόγησή τους βάσει άνω και κάτω φραγμάτων, η μέθοδος επιτρέπει την αποκοπή μεγάλων περιοχών του χώρου λύσεων που δεν μπορούν να περιέχουν καλύτερες λύσεις. Από την άλλη πλευρά, η μέθοδος Branch and Cut επεκτείνει τη βασική προσέγγιση εισάγοντας τέμνοντα επίπεδα (cutting planes) τα οποία στοχεύουν στον περαιτέρω περιορισμό του χώρου των εφικτών λύσεων του χαλαρωμένου προβλήματος, χωρίς όμως να αποκλείουν καμία ακέραια λύση. Η προσέγγιση αυτή προσφέρει σημαντική ενίσχυση στη διαδικασία σύγκλισης, όπως θα αναλυθεί στην επόμενη ενότητα.

### 2.3.3 Μέθοδος Τεμνότων Επιπέδων(Cutting planes)

Η μέθοδος των τεμνότων επιπέδων αποτελεί θεμελιώδη τεχνική στον Ακέραιο Προγραμματισμό, με στόχο την εξάλειψη μη ακέραιων βέλτιστων λύσεων που προκύπτουν από τη χαλαρωμένη γραμμική μορφή του προβλήματος. Η βασική ιδέα είναι η επαναληπτική προσθήκη έγκυρων ανισοτήτων (τομών), οι οποίες αποκλείουν τις μη ακέραιες λύσεις χωρίς να αφαιρούν καμία εφικτή Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

ακέραιη λύση. Σε κάθε επανάληψη το πρόβλημα επιλύεται επαναληπτικά μέχρις ότου προκύψει ακέραια βέλτιστη λύση ή αποδειχθεί ότι δεν υπάρχει ακέραιη λύση.

Στην πράξη, ο αριθμός των απαραίτητων τομών δεν είναι γνωστός εκ των προτέρων και η διαδικασία μπορεί να γίνει υπολογιστικά απαιτητική. Για αυτό, η μέθοδος των τεμνόντων επιπέδων συνδυάζεται συχνά με τη μέθοδο Διακλάδωσης και Φραγμού (Branch-and-Bound), σχηματίζοντας τη μέθοδο branch-and-cut.

Στο πρόβλημα του Πλανόδιου Πωλητή προκύπτουν δύο βασικά χαρακτηριστικά που δυσκολεύουν την επίλυση:  
(α) η επιβολή ακεραιότητας στις δυαδικές μεταβλητές  $x_{ij}$  και  
(β) η αποτροπή υποκύκλων (subtours).

Η ακεραιότητα υποστηρίζεται μέσω μεθόδων MIP (π.χ. branch-and-cut). Παράλληλα, απαιτείται η δυναμική εισαγωγή περιορισμών εξάλειψης υποκύκλων (Subtour Elimination Constraints, SECs), καθώς η τυπική γραμμική χαλάρωση του TSP επιτρέπει μη ακέραιες λύσεις και υποκύκλους.

Οι SECs δεν επιβάλλουν άμεσα ακεραιότητα λύσεων, αλλά μειώνουν προοδευτικά το πολύτοπο της γραμμικής χαλάρωσης ώστε, αφού προστεθούν όλες οι απαραίτητες τομές, κάθε βέλτιστη λύση να είναι ακέραιη και να αντιστοιχεί σε έναν έγκυρο κύκλο Hamilton. Ειδικότερα, προστίθενται μόνο οι SECs που αντιστοιχούν σε υποκύκλους που εντοπίστηκαν σε προηγούμενη επανάληψη, αποφεύγοντας την εισαγωγή του πλήρους αριθμού περιορισμών που απαιτείται για την αποφυγή όλων των υποκύκλων.

## 2.4 Συμμετρική μορφή TSP: Μαθηματική Διατύπωση και Υλοποίηση Επίλυσης σε Python

Στο πλαίσιο της παρούσας εργασίας, υιοθετήθηκε και υλοποιήθηκε η συμμετρική εκδοχή του προβλήματος του Πλανόδιου Πωλητή (Symmetric TSP), στην οποία θεωρούμε ότι το κόστος μετάβασης μεταξύ δύο πόλεων είναι το ίδιο ανεξαρτήτως κατεύθυνσης, δηλαδή  $c_{ij} = c_{ji}$ . Η παραδοχή αυτή απλοποιεί τη μαθηματική μοντελοποίηση, επιτρέποντας την αναπαράσταση του προβλήματος ως ένα πλήρη μη κατευθυνόμενο γράφο. Σε όρους γραμμικού προγραμματισμού διατυπώνεται ως εξής:

$$\min_{\{y_e\}} \sum_{e \in E} c_e \cdot y_e \quad (4.1)$$

Υπό τους περιορισμούς:

$$\sum_{e \in N(u)} y_e = 2, \quad \forall u \in V \quad (4.2)$$

$$\sum_{e \in E(S)} y_e \leq |S| - 1, \quad \forall S \subset V, \quad 2 \leq |S| \leq |V| - 1 \quad (4.3)$$

$$y_e \in \{0, 1\}, \quad \forall e \in E \quad (4.4)$$

Όπου:

- $E$ : Σύνολο ακμών.
- $V$ : Σύνολο κορυφών (πόλεων).
- $y_e$ : Δυαδική μεταβλητή που δηλώνει αν η ακμή  $e$  περιλαμβάνεται στη λύση.
- $c_e$ : Κόστος/απόσταση της ακμής  $e$ .
- $N(v)$ : Σύνολο ακμών που συνδέονται με την κορυφή  $v$ .

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

- $E(S)$ : Υποσύνολο των ακμών που συνδέουν τις κορυφές του υποσυνόλου  $S \subseteq V$ .

Ο πρώτος περιορισμός της συμμετρικής διατύπωσης του TSP:

$$\sum_{e \in N(u)} y_e = 2 \quad \forall i \in V \quad (4.2)$$

προκύπτει ως αποτέλεσμα της μετάβασης από τη κατευθυνόμενη (ασύμμετρη) μορφή του προβλήματος σε μια μη κατευθυνόμενη (συμμετρική) εκδοχή. Στην ασύμμετρη περίπτωση, η πορεία του πωλητή αναπαρίσταται μέσω κατευθυνόμενων ακμών, και κάθε κόμβος πρέπει να έχει μία εισερχόμενη και μία εξερχόμενη ακμή, κάτι που απαιτεί δύο ξεχωριστούς ισοτικούς περιορισμούς (in-degree και out-degree) για κάθε κόμβο:

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

Στη συμμετρική εκδοχή όμως, οι ακμές είναι μη κατευθυνόμενες. Καθώς δεν υπάρχει πλέον έννοια κατεύθυνσης, δεν έχει σημασία ποια είναι η “είσοδος” ή η “έξοδος” από μια πόλη —σημασία έχει μόνο το αν η πόλη συμμετέχει ακριβώς σε δύο ακμές της τελικής διαδρομής. Έτσι, οι δύο προηγούμενοι περιορισμοί συγχωνεύονται σε έναν ενιαίο περιορισμό βαθμού, ο οποίος δηλώνει ότι κάθε κόμβος έχει βαθμό 2 στο γράφο της λύσης.

Αξίζει να σημειωθεί ότι η μετάβαση από τη γενική (κατευθυνόμενη) εκδοχή του Προβλήματος του Πλανόδιου Πωλητή στη συμμετρική (μη κατευθυνόμενη) μορφή συνοδεύεται από σημαντικά υπολογιστικά οφέλη. Συγκεκριμένα, ενώ στη γενική περίπτωση το σύνολο των ακμών περιλαμβάνει και τις δύο κατευθύνσεις για κάθε ζεύγος πόλεων  $(i,j)$  και  $(j,i)$  στη συμμετρική μορφή λαμβάνεται υπόψη μόνο μία από αυτές. Ως αποτέλεσμα, ο συνολικός αριθμός δυαδικών μεταβλητών  $y_e$  που χρησιμοποιούνται στη μαθηματική μοντελοποίηση μειώνεται σχεδόν στο μισό. Αυτή η μείωση επιφέρει σημαντική συρρίκνωση του χώρου των εφικτών λύσεων και συμβάλλει καθοριστικά στο χρόνο σύγκλισης προς την βέλτιστη λύση.

### 3 Εισαγωγή στα Νευρωνικά Δίκτυα Γράφων (GNNs)

Ένας μαθηματικός γράφος είναι μια δομή δεδομένων που περιγράφει σχέσεις μεταξύ οντοτήτων. Τυπικά, ένας γράφος ορίζεται ως ένα ζεύγος  $G = (V, E)$  όπου το σύνολο  $V$  περιέχει τους κόμβους (ή κορυφές) και το σύνολο  $E \subseteq V \times V$  περιέχει τις ακμές (ή συνδέσεις) μεταξύ των κόμβων. Οι ακμές μπορεί να έχουν φορά και στην περίπτωση αυτή ο γράφος χαρακτηρίζεται ως κατευθυνόμενος ή μπορεί να περιγράφουν αμφίδρομες σχέσεις όπου η φορά της σύνδεσης δεν έχει σημασία και τότε ο γράφος χαρακτηρίζεται ως μη κατευθυνόμενος. Επιπλέον, οι ακμές μπορούν να συνοδεύονται από —όπως αριθμητικά βάρη (π.χ. αποστάσεις, κόστη) ή κατηγορικές ετικέτες που δηλώνουν τον τύπο της σχέσης (π.χ. «φιλική», «επαγγελματική»).

Αυτή η εννοιολογική ευελιξία καθιστά τους γράφους εξαιρετικά ισχυρό εργαλείο μοντελοποίησης. Επιτρέπουν την αναπαράσταση μιας ευρείας γκάμας πραγματικών συστημάτων: από κοινωνικά και οδικά δίκτυα, μέχρι χημικές ενώσεις, πληροφοριακές ροές, και αλληλεπιδράσεις σε φυσικά και βιολογικά περιβάλλοντα. Χάρη σε αυτή την εκφραστική δύναμη, οι γράφοι έχουν αποκτήσει

κεντρική θέση σε πολλά επιστημονικά πεδία — όπως η πληροφορική, η φυσική, η βιολογία και τα οικονομικά.

Τα τελευταία χρόνια, η σημασία των γράφων έχει αναδειχθεί ιδιαίτερα και στο πεδίο της μηχανικής μάθησης, καθώς προσφέρουν έναν ευέλικτο τρόπο οργάνωσης των δεδομένων που υπερβαίνει τις παραδοσιακές δομές, όπως οι μήτρες(πίνακες). Η αναπαράσταση των δεδομένων με την χρήση γράφων επιτρέπει τη μοντελοποίηση πολύπλοκων συσχετίσεων και αλληλεπιδράσεων, καθιστώντας την ιδανική για μια πληθώρα εφαρμογών.

Σε αυτό το πλαίσιο, έχουν διαμορφωθεί ορισμένες γενικές κατηγορίες προβλημάτων που προσεγγίζονται μέσω της δομής των γράφων:

- Κατηγοριοποίηση κόμβων (node classification): Πρόβλεψη ετικέτας για έναν μεμονωμένο κόμβο, όπως ο προσδιορισμός του τύπου ενός κυττάρου σε ένα βιολογικό δίκτυο.
- Πρόβλεψη ακμών (link prediction): Εκτίμηση της πιθανότητας εμφάνισης νέας σύνδεσης μεταξύ δύο κόμβων, π.χ. πρόβλεψη νέας φιλίας σε κοινωνικά δίκτυα.
- Κατηγοριοποίηση γράφων (graph classification): Ανάθεση μιας συνολικής ετικέτας σε έναν γράφο, όπως η ταξινόμηση ενός μορίου ανάλογα με τη φαρμακολογική του δράση.
- Συστήματα συστάσεων (recommendation systems): Ειδική περίπτωση πρόβλεψης ακμών, όπου χρήστες και προϊόντα αναπαρίστανται ως κόμβοι σε διμερείς γράφους, και το ζητούμενο είναι η πρόβλεψη πιθανών αλληλεπιδράσεων (π.χ. αξιολογήσεις ή αγορές).

Για να κατανοηθεί σε βάθος πώς επιτυγχάνεται η μάθηση από τέτοιου είδους δομημένα δεδομένα, είναι χρήσιμο να προηγηθεί μια σύντομη ανασκόπηση των βασικών αρχών των νευρωνικών δικτύων. Στην επόμενη ενότητα, θα παρουσιαστούν τα κύρια χαρακτηριστικά τους, ώστε να τεθεί το υπόβαθρο για την κατανόηση των Νευρωνικών Δικτύων Γράφων (Graph Neural Networks – GNNs).

### 3.1 Βασικές Αρχές Νευρωνικών Δικτύων

Τα κλασικά νευρωνικά δίκτυα (Artificial Neural Networks – ANNs) έχουν ως θεμελιώδες δομικό στοιχείο τον τεχνητό νευρώνα — μια υπολογιστική μονάδα που δέχεται είσοδο, πολλαπλασιάζει κάθε τιμή με ένα αντίστοιχο βάρος (weight), εφαρμόζει μία μη γραμμική συνάρτηση ενεργοποίησης (activation function) και παράγει την τελική έξοδο. Η διαδικασία εκπαίδευσης βασίζεται στη μέθοδο της «προς τα πίσω διάδοσης» (backpropagation), κατά την οποία τα βάρη ενημερώνονται με βάση το σφάλμα πρόβλεψης.

Στις παραδοσιακές αρχιτεκτονικές, οι είσοδοι οργανώνονται ως δισδιάστατοι πίνακες διαστάσεων  $N \times F$ , όπου κάθε γραμμή αντιστοιχεί σε μία παρατήρηση (sample), και κάθε στήλη σε μία μεταβλητή χαρακτηριστικό (feature) του χώρου εισόδου. Σε αυτά τα πλαίσια, η διάταξη (σειρά) των γραμμών και των στηλών δεν φέρει σημασιολογική πληροφορία· δηλαδή, μπορούμε να αναδιατάξουμε είτε τις παρατηρήσεις είτε τα χαρακτηριστικά, αρκεί να διατηρείται η αντιστοιχία μεταξύ των στοιχείων. Είναι ωστόσο κρίσιμο να επισημανθεί ότι, για ένα συγκεκριμένο νευρωνικό δίκτυο, ο αριθμός των χαρακτηριστικών (στηλών  $F$ ) παραμένει σταθερός, ενώ ο αριθμός των παρατηρήσεων (γραμμών  $N$ ) μπορεί να ποικίλλει ανάλογα με το σύνολο δεδομένων. Η ιδιότητα αυτή προσφέρει μεγάλη ευελιξία στα παραδοσιακά μοντέλα μάθησης. Αντιθέτως, όταν επιχειρούμε να περιγράψουμε δομές όπως γράφοι μέσω πίνακα γειννίας (adjacency matrix), η

εισαγωγή ή η αφαίρεση κόμβων τροποποιεί άμεσα τις διαστάσεις του πίνακα, γεγονός που διαταράσσει τη δομή εισόδου και δημιουργεί δυσκολίες στη γενίκευση του μοντέλου.

Αξίζει επιπλέον να σημειωθεί ότι υπάρχουν περιπτώσεις στις οποίες η διάταξη των στοιχείων στον πίνακα εισόδου είναι κρίσιμη, καθώς φέρει χωρική ή χρονική πληροφορία. Αυτή είναι η περίπτωση της δομής δεδομένων πλέγματος (grids) π.χ. εικόνες ή σήματα, όπου η σχετική θέση των παρατηρήσεων επιτρέπει την αξιοποίηση της τοπικότητας μέσω της διαδικασίας της συνέλιξης (convolution). Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks – CNNs) έχουν βασιστεί σε αυτή την αρχή και έχουν φέρει επανάσταση σε εφαρμογές όπως η αναγνώριση εικόνων και η επεξεργασία φυσικής γλώσσας (Zhou et al., 2018; Wu et al., 2019).

Στην περίπτωση των γραφημάτων, η ενσωμάτωση της έννοιας της συνέλιξης αποτελεί θεμελιώδη πρόκληση, κυρίως λόγω δύο βασικών χαρακτηριστικών:

1. Οι γράφοι δεν διαθέτουν εγγενή διάταξη μεταξύ των κόμβων — δεν υπάρχει φυσική σειρά, όπως στις εικόνες.
2. Ο ίδιος γράφος μπορεί να έχει πολλαπλές ισομορφικές αναπαραστάσεις, δηλαδή διαφορετικές αλγεβρικές μορφές που περιγράφουν ακριβώς την ίδια δομή.

Αυτά τα εμπόδια οδήγησαν στην ανάπτυξη νέων μεθόδων και αρχιτεκτονικών που γενικεύουν τη συνέλιξη σε δομές δεδομένων όπως οι γράφοι — δίνοντας έτσι τη βάση για τα Νευρωνικά Δίκτυα Γράφων (Graph Neural Networks – GNNs).

### 3.2 Θεμελιώδης Περιορισμοί και Γενίκευση της Συνέλιξης σε Γράφους

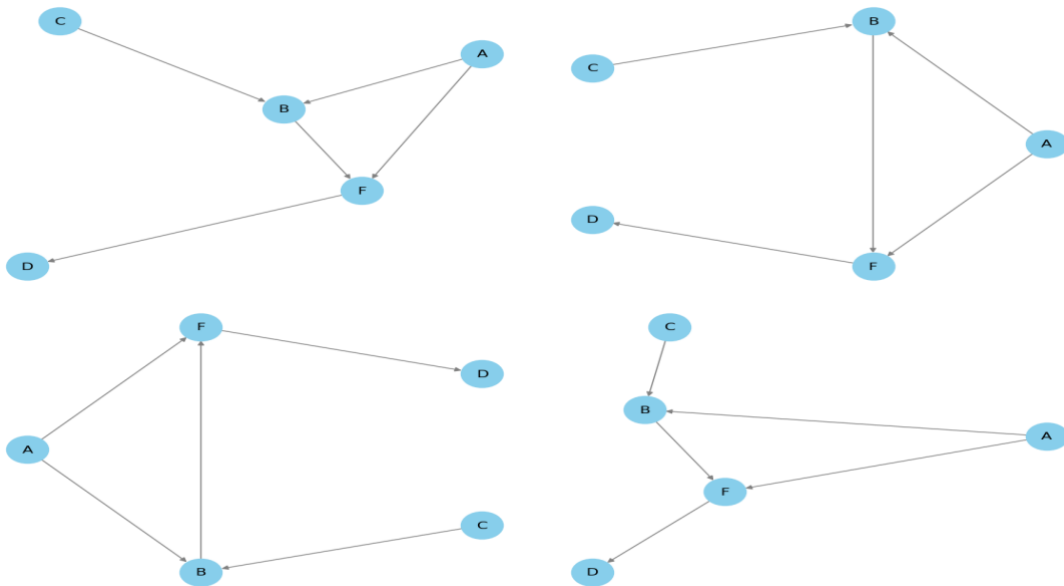
Η επέκταση της έννοιας της συνέλιξης από δεδομένα οργανωμένα σε δομές πλέγματος (όπως εικόνες ή ηχητικά σήματα) σε δομές γράφων αποτελεί μία από τις σημαντικότερες θεωρητικές και πρακτικές προκλήσεις στο πεδίο των Νευρωνικών Δικτύων Γράφων (Graph Neural Networks – GNNs) (Zhou et al., 2018; Wu et al., 2019). Σε αντίθεση με τα Συνελικτικά Νευρωνικά Δίκτυα (CNNs), τα οποία επωφελούνται από την σταθερή διάταξη των εισερχόμενων δεδομένων, οι γράφοι συνιστούν ετερογενείς δομές με δύο βασικά χαρακτηριστικά:

- Δεν υφίσταται εγγενής ή φυσική σειρά μεταξύ των κόμβων.
- Ο ίδιος γράφος μπορεί να έχει πολλαπλές ισομορφικές αναπαραστάσεις, δηλαδή διαφορετικές αλγεβρικές μορφές που περιγράφουν την ίδια τοπολογία.

Για να κατανοήσουμε την πολυπλοκότητα αυτή, ας εξετάσουμε τον παρακάτω πίνακα γειτνίασης:

	A	B	C	D	F
A	0	1	0	0	1
B	0	0	0	0	1
C	0	1	0	0	0
D	0	0	0	0	0
F	0	0	0	1	0

Ο παραπάνω πίνακας μπορεί να αναπαριστά πολλούς φαινομενικά διαφορετικούς γράφους, όπως φαίνεται στο Σχήμα 3.1. Παρότι η γραφική απεικόνιση αλλάζει, η υποκείμενη δομή παραμένει η ίδια.



**Σχήμα 3.1 Διαφορετικές διατάξεις του ίδιου γράφου**

Αντιστρόφως, ο ίδιος γράφος μπορεί να παρασταθεί από διαφορετικούς πίνακες γεινίασης με απλή αναδιάταξη της σειρά των κόμβων. Το φαινόμενο αυτό είναι γνωστό ως *πρόβλημα της μετάθεσης (permutation problem)* και αποτελεί ειδική περίπτωση του προβλήματος του ισομορφισμού. Για παράδειγμα, ο παρακάτω πίνακας γεινίασης περιγράφει την ίδια τοπολογία, αλλά με διαφορετική διάταξη των κορυφών A και B:

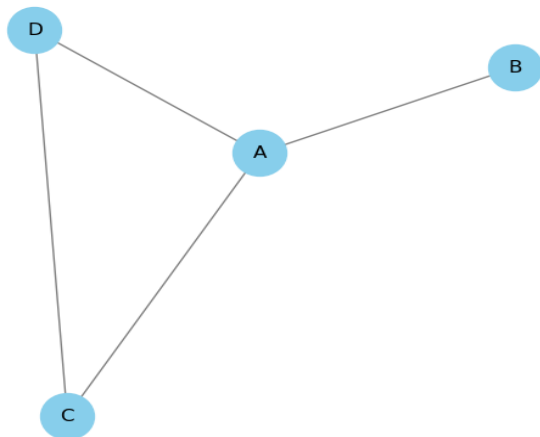
	B	A	C	D	F
B	0	0	0	0	1
A	1	0	0	0	1
C	1	0	0	0	0
D	0	0	0	0	0
F	0	0	0	1	0

Εν κατακλείδι, στην περίπτωση της μετάθεσης του γράφου  $G(V,E)$  τα σύνολα  $E$  και  $V$  παραμένουν αμετάβλητα. Αξίζει ωστόσο να σημειωθεί ότι ο ισομορφισμός συνιστά σημαντική πρόκληση στην εφαρμογή τεχνικών μηχανικής μάθησης σε γράφους, καθώς δύο διαφορετικά αλγεβρικά μοντέλα μπορεί να αναπαριστούν την ίδια τοπολογία. Δύο γράφοι  $G = (V, E)$  και  $G' = (V', E')$  θεωρούνται ισομορφικοί όταν υπάρχει μια αμφιμονοσήμαντη αντιστοίχιση  $f: V \rightarrow V'$  τέτοια ώστε κάθε ακμή

$(u, v) \in E$  να αντιστοιχεί στην ακμή  $(f(u), f(v)) \in E'$ . Με άλλα λόγια, η δομή των συνδέσεων διατηρείται, ακόμη κι αν αλλάξουν οι ονομασίες ή η σειρά των κόμβων.

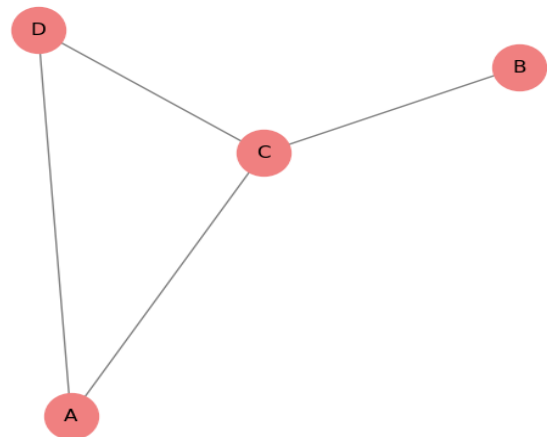
	A	B	C	D
A	0	1	1	1
B	1	0	0	0
C	1	0	0	1
D	1	0	1	0

Γράφος 1



	A	B	C	D
A	0	0	1	1
B	0	0	1	0
C	1	1	0	1
D	1	0	1	0

Γράφος 2



**Σχήμα 3.2: Ισομορφισμός λόγω ανταλλαγής θέσης κορυφών A και C**

Το Σχήμα 3.2 απεικονίζει ένα παράδειγμα ισομορφισμού μεταξύ δύο γράφων.

Τα σύνολα ακμών είναι:

- $E_1 = \{(D, A), (A, B), (A, C), (C, D)\}$
- $E_2 = \{(D, C), (C, B), (C, A), (A, D)\}$ .

Αν και τα σύνολα ακμών είναι διαφορετικά, η συνδυαστική δομή είναι ουσιαστικά η ίδια. Η δεύτερη περίπτωση προκύπτει από την ανταλλαγή των ονομάτων των κορυφών A και C στον αρχικό γράφο.

Εκτός από το ζήτημα του ισομορφισμού, η ετερογένεια των γράφων εμφανίζεται και σε άλλες περιπτώσεις όπως όταν:

- Ο αριθμός των κόμβων και ακμών διαφέρει μεταξύ γράφων.
- Οι κόμβοι μπορεί να έχουν διαφορετικό αριθμό γειτόνων (μεταβλητός βαθμός).
- Οι γράφοι μπορεί να περιέχουν πολλαπλούς τύπους κόμβων και ακμών, όπως π.χ. άτομα, προϊόντα, αξιολογήσεις, συναλλαγές κ.λπ.

Σε αντίθεση με τα εικονοστοιχεία (pixels) μιας εικόνας:

- Υπάρχει σταθερή διάταξη και 1-προς-1 αντιστοίχιση θέσης.
- Οι διαστάσεις είναι σταθερές (π.χ., 28×28).
- Κάθε pixel έχει τον ίδιο αριθμό γειτόνων.
- Η συνέλιξη εφαρμόζεται ομοιόμορφα σε όλο το πεδίο.

Αντιθέτως, στους γράφους δεν υπάρχει τέτοια κανονικότητα, γεγονός που καθιστά την εφαρμογή της συνέλιξης μη τετριμμένη.

### 3.2.1 Κίνητρο για γενίκευση της συνέλιξης

Τα συνελκτικά νευρωνικά δίκτυα (Convolutional Neural Networks, CNNs) έχουν καταφέρει εντυπωσιακά αποτελέσματα σε προβλήματα υπολογιστικής όρασης, επεξεργασία φυσικής γλώσσας και γενικά σε δεδομένα που παρουσιάζουν ευκλείδεια δομή: εικόνες, κείμενο, ήχος. Σε τέτοιες περιπτώσεις, η τοπολογία των δεδομένων είναι σταθερή, και η γειτονιά κάθε σημείου (π.χ. pixel) είναι σαφώς ορισμένη και σταθερών διαστάσεων.

Ωστόσο, σε πλήθος πραγματικών προβλημάτων, η δομή των δεδομένων δεν ακολουθεί αυστηρά ευκλείδεια γεωμετρία. Παραδείγματα περιλαμβάνουν:

- Κοινωνικά δίκτυα (οι κόμβοι είναι χρήστες, οι ακμές φιλικές σχέσεις),
- Μόρια στην επιστήμη της βιολογίας (οι κόμβοι είναι άτομα, οι ακμές χημικοί δεσμοί),
- Δίκτυα μεταφορών, επικοινωνίας, ή πληροφορίας,

- Τεχνουργημένες περιπτώσεις όπως ο πλήρης γράφος στην περίπτωση του TSP.

Αυτές οι δομές περιγράφονται με γράφους, οι οποίοι είναι μη ευκλείδειες, ανομοιογενείς, γεωμετρικές. Ο αριθμός και η τοπολογία των γειτόνων κάθε κόμβου διαφέρει, καθιστώντας την απευθείας εφαρμογή της συνέλιξης μέσω σταθερής μάσκας ανέφικτη. Για να γίνει πιο κατανοητό ως προσπαθήσουμε να ορίσουμε την **‘γειτονιά’** σε μια εικόνα και σε ένα γράφο.

Σε έναν δισδιάστατο πίνακα εικόνας, κάθε ρixel έχει σταθερούς γείτονες (π.χ. 8 σε  $3 \times 3$  πυρήνα(kernel)). Σε έναν γράφο, κάθε κόμβος  $u$  έχει γείτονες  $N(u)$  με μη σταθερό πλήθος και χωρίς συγκεκριμένη διάταξη. Επομένως:

- Δεν μπορούμε να εφαρμόσουμε πυρήνες συνέλιξης με σταθερή μάσκα.
  - Δεν υπάρχει έννοια "γειτονικού ρixel προς τα πάνω ή κάτω".
  - Η διάταξη των γειτόνων είναι ασύμμετρη και αταξινόμητη.
- Αυτό οδηγεί στην ανάγκη για νέα, γενικευμένη έννοια συνέλιξης.

### 3.2.2 Συνέλιξη σε Μη Ευκλείδειες Δομές

Η συνέλιξη σε γράφους παύει πλέον να ορίζεται ως σταθερή γραμμική πράξη μιας μάσκας σε δεδομένα πλέγματος, αλλά να αποτελεί μια αλγεβρική διαδικασία συσσώρευσης πληροφορίας από την γειτονία κάθε κόμβου. Ο βασικός στόχος παραμένει ο ίδιος. Να κατασκευάσουμε πλούσιες, πολυεπίπεδες αναπαραστάσεις των κόμβων και των ακμών (embeddings), ενσωματώνοντας την δομή και τα χαρακτηριστικά των γειτόνων τους.

Αυτό επιτυγχάνεται με την **μηχανική διάδοση μηνυμάτων(message passing)**, που στην ουσία, κάθε κόμβος συλλέγει πληροφορία από την γειτονία του, την ενσωματώνει στην αναπαράστασή του και τη μεταδίδει ξανά στους γείτονές του σε επόμενο επίπεδο. Σε κάθε επίπεδο  $\ell$  του δικτύου, ο στόχος είναι να υπολογιστεί μια νέα αναπαράσταση (embedding) για κάθε κόμβο  $v$ , η οποία εξαρτάται:

- από την τρέχουσα αναπαράσταση του κόμβου  $v$ ,
- από τις αναπαραστάσεις των γειτόνων του  $u \in N(v)$ ,
- και από πιθανές πληροφορίες που φέρουν οι ακμές (features)  $e_{uv}$ .

Η διαδικασία αυτή είναι αναδρομική: οι αναπαραστάσεις του επιπέδου  $\ell$  χτίζονται με βάση τις αναπαραστάσεις του επιπέδου  $\ell-1$ . Ο γενικός μαθηματικός τύπος που περιγράφει την ενημέρωση των αναπαραστάσεων των κορυφών σε γράφους είναι:

$$h_u^\ell = \psi^\ell \left( h_u^{\ell-1}, \text{AGG}^\ell \left( \{ \varphi^\ell ( h_u^{\ell-1}, h_v^{\ell-1}, e_{uv} ) \mid v \in N(u) \} \right) \right)$$

Όπου:

- $h_u^\ell$  η αναπαράσταση(embedding) του κόμβου  $u$  στο επίπεδο  $\ell$
- $\psi^\ell$  είναι συνάρτηση ενημέρωσης,
- $\text{AGG}^\ell$  συνάρτηση συσσώρευσης (aggregation function),
- $\varphi^\ell$  είναι η συνάρτηση δημιουργίας μηνυμάτων
- $N(u)$  το σύνολο των γειτόνων του κόμβου  $u$ .

Η εξίσωση που παρουσιάστηκε περιγράφει τη θεμελιώδη πράξη συνελικτικής επεξεργασίας σε γράφους και αποτελεί την καρδιά της διαδικασίας δημιουργίας Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

αναπαραστάσεων (embeddings). Όπως αναφέρθηκε και προηγουμένων η διαδικασία αυτή είναι αναδρομική: η αναπαράσταση κάθε κόμβου  $h_u^\ell$  στο επίπεδο  $\ell$  εξαρτάται από την αναπαράστασή του στο προηγούμενο επίπεδο  $\ell - 1$  καθώς και από τις αναπαραστάσεις των γειτόνων του. Αυτό σημαίνει ότι, μέσω της διαδοχικής εφαρμογής του ίδιου κανόνα ανά επίπεδο, οι αναπαραστάσεις (embeddings) "βαθαίνουν", ενσωματώνοντας πληροφορία από όλο και ευρύτερες περιοχές του γράφου.

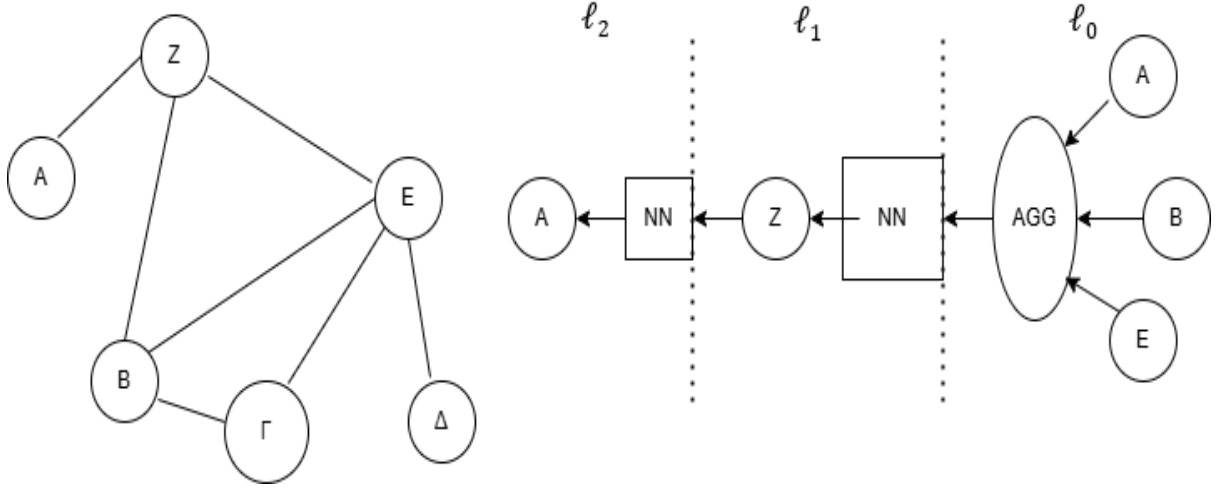
Κρίσιμος παράγοντας στη γενίκευση της συνέλιξης είναι τα προ-απαιτούμενά της συνάρτησης **AGG** (Aggregation), η οποία είναι υπεύθυνη για τη συσσώρευση πληροφορίας από τη γειτονιά  $N(u)$  του κόμβου  $u$ . Εφόσον οι γείτονες ενός κόμβου σε γράφο δεν έχουν εγγενή διάταξη, οποιαδήποτε λειτουργία συσσώρευσης πρέπει να είναι αντιμεταθετική (*permutation-invariant*), δηλαδή να μην εξαρτάται από τη σειρά με την οποία προσκομίζονται οι πληροφορίες των γειτόνων.

Αυτός ο περιορισμός είναι απαραίτητος προκειμένου η διαδικασία να είναι συνεπής ως προς το πρόβλημα του ισομορφισμού: δύο κόμβοι σε ισόμορφους γράφους με ίδιες τοπικές δομές (ανεξαρτήτως αρίθμησης ή διάταξης γειτόνων) πρέπει να καταλήγουν σε ίδιες αναπαραστάσεις. Εάν η έξοδος της συνάρτησης συσσώρευσης (AGG) εξαρτάται από τη σειρά των εισόδων, τότε θα μπορούσε να καταστραφεί αυτή η ισότητα και η μοντελοποίηση να είναι ευαίσθητη σε μεταθέσεις.

Για τον λόγο αυτό, οι πιο συχνά χρησιμοποιούμενες συναρτήσεις AGG είναι οι εξής:

- Πρόσθεση
- Μέσο όρος
- Μέγιστο-Ελάχιστο

Όλες οι παραπάνω ικανοποιούν την απαίτηση της αντιμεταθετικότητας, εξασφαλίζοντας ότι η τελική αναπαράσταση εξαρτάται μόνο από το *ποιοι* είναι οι γείτονες και *τι χαρακτηριστικά φέρουν*, αλλά όχι από το *πώς είναι διατεταγμένοι*. Για την καλύτερη κατανόηση της εξίσωσης που περιεγράφηκε προηγουμένως, παρατίθεται στο Σχήμα 3.3 ο υπολογιστικός γράφος της κορυφής A, ο οποίος αποτυπώνει σχηματικά τη διαδικασία συλλογής και συνδυασμού πληροφορίας από τη γειτονιά της.



Σχήμα 3.3 Υπολογιστικός γράφος για την αναπαράσταση της κορυφής A

### 3.2.3 Δίκτυα Μηχανισμών Προσοχής

Αφού παρουσιάστηκε η γενική αρχιτεκτονική των συνελικτικών νευρωνικών δικτύων σε γράφους (GCNs), αξίζει να εξεταστεί μια ιδιαίτερα αποδοτική και δημοφιλή παραλλαγή τους: οι Μηχανισμοί Προσοχής σε Γράφους (Graph Attention Networks, GAT). Η βασική καινοτομία των GAT έγκειται στην ενσωμάτωση μηχανισμών προσοχής (attention mechanisms), οι οποίοι επιτρέπουν στο δίκτυο να μαθαίνει δυναμικά τη σημαντικότητα κάθε γείτονα για την αναπαράσταση ενός κόμβου.

Με τον τρόπο αυτό, αντί να συμμετέχουν όλοι οι γείτονες ισότιμα — όπως συμβαίνει, για παράδειγμα, με τον μέσο όρο ή το απλό άθροισμα στα παραδοσιακά GCNs — το μοντέλο αποδίδει προσαρμοσμένες βαρύτητες στις πληροφορίες που προέρχονται από κάθε γείτονα, ανάλογα με τη σχετική σημασία τους για τον κόμβο-στόχο (Velicković et al., 2018).

Ανακεφαλαιώνοντας, η γενική εξίσωση για την ανανέωση της αναπαράστασης ενός κόμβου  $u$  στο επίπεδο  $\ell$  έχει τη μορφή:

$$h_u^\ell = \psi^\ell \left( h_u^{\ell-1}, \text{AGG}^\ell \left( \left\{ \varphi^\ell \left( h_u^{\ell-1}, h_v^{\ell-1}, e_{uv} \right) \mid v \in N(u) \right\} \right) \right)$$

Στο πλαίσιο των Νευρωνικών Δικτύων Γράφων, αυτή η γενική μορφή εξειδικεύεται ως εξής:

$$h_u^\ell = \sigma \left( \sum_{v \in N(u)} a_{uv}^\ell \mathbf{W}^\ell h_v^{\ell-1} \right)$$

Όπου:

- $\mathbf{h}_u^\ell$  είναι η αναπαράσταση(embedding) του κόμβου  $u$  στο επίπεδο  $\ell$ ,
- $\mathbf{h}_v^{\ell-1}$  είναι η αναπαράσταση(embedding) του γείτονα  $v$  στο προηγούμενο επίπεδο  $\ell-1$ ,
- $W^\ell$  είναι πίνακας γραμμικού μετασχηματισμού (shared weight matrix), κοινός για όλους τους κόμβους στο επίπεδο  $\ell$ ,
- $a_{uv}^\ell$  είναι ο συντελεστής προσοχής, που εκφράζει τη σχετική σημασία του κόμβου  $v$  για την ενημέρωση του κόμβου  $u$ ,
- $\sigma$  είναι μια μη-γραμμική συνάρτηση ενεργοποίησης, όπως LeakyReLU.

Επομένως, στην ορολογία της γενικής εξίσωσης:

- Η συνάρτηση  $\varphi^\ell$  υλοποιείται ως  $\varphi^\ell(\mathbf{h}_u, \mathbf{h}_v) = \alpha_{uv}^\ell W^\ell \mathbf{h}_v$
- Ο τελεστής  $AGG^\ell$  γίνεται ένα σταθμισμένο άθροισμα πάνω στους γείτονες,
- Και η  $\psi^\ell$  αντιστοιχεί στην συνάρτηση ενεργοποίησης(activation function).

Οι συντελεστές προσοχής  $a_{uv}^\ell$  προκύπτουν ως εξής:

$$a_{uv} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T \left[ W^\ell \mathbf{h}_u^{\ell-1} \parallel W^\ell \mathbf{h}_v^{\ell-1} \right] \right)\right)}{\sum_{k \in \mathcal{N}(u)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T \left[ W^\ell \mathbf{h}_u^{\ell-1} \parallel W^\ell \mathbf{h}_v^{\ell-1} \right] \right)\right)}$$

Στην παραπάνω σχέση παρατηρούμε ότι οι συντελεστές σημαντικότητας βασίζονται στην συνένωση των μετασχηματισμένων αναπαραστάσεων των γειτονικών μεταξύ τους κόμβων  $u$  και  $v$ .

Το διάνυσμα  $\mathbf{a}$  και ο πίνακας  $\mathbf{W}$  είναι παράμετροι που μαθαίνονται κατά την εκπαίδευση. Με αυτό τον τρόπο, το GAT επιτρέπει στο δίκτυο να εστιάζει σε “σημαντικούς” γείτονες, αντί να αντιμετωπίζει όλους τους κόμβους της τοπικής γειτονιάς με ίση βαρύτητα.

Αξίζει όμως να σημειωθεί ότι, παρόλο που στις παραπάνω εξισώσεις για τη δημιουργία των αναπαραστάσεων (embeddings) μέσω των Δικτύων Μηχανισμών Προσοχής (GATs) οι ακμές δεν φαίνεται να παίζουν άμεσα ρόλο —καθώς η συνάρτηση δημιουργίας μηνύματος  $\varphi^\ell(\mathbf{h}_u, \mathbf{h}_v)$  εξειδικεύεται μόνο ως συνάρτηση των κόμβων— στην πράξη, αυτό δεν αποτελεί περιορισμό. Η υλοποίηση των μηχανισμών προσοχής σε σύγχρονες βιβλιοθήκες όπως το PyTorch Geometric επιτρέπει την ενσωμάτωση των ιδιοτήτων των ακμών (edge attributes) τόσο στον υπολογισμό των συντελεστών προσοχής όσο και στη φάση δημιουργίας των μηνυμάτων.

Αυτό το χαρακτηριστικό είναι εξαιρετικά σημαντικό όταν εξετάζουμε προβλήματα όπως το Traveling Salesman Problem, στα οποία οι ακμές του γράφου φέρουν κρίσιμη σημασιολογική πληροφορία. Συγκεκριμένα, στην περίπτωση του TSP, οι ακμές αντιστοιχούν σε ευκλείδειες αποστάσεις μεταξύ κόμβων (πόλεων), και αυτή η πληροφορία είναι θεμελιώδης για την εκτίμηση της βέλτιστης διαδρομής. Μέσω της κατάλληλης τροποποίησης της συνάρτησης προσοχής ή της συνάρτησης δημιουργίας μηνύματος, οι αποστάσεις αυτές μπορούν να ενσωματωθούν στην αναπαράσταση, προσφέροντας στο GNN πρόσβαση σε σημαντική γνώση για την τοπολογία του προβλήματος. Στην πράξη, η συνάρτηση προσοχής μπορεί να τροποποιηθεί ώστε να εξαρτάται από τα χαρακτηριστικά των ακμών, δηλαδή:

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

$$\alpha_{uv}^{\ell} = \text{Attention}(W_{h_u}^{\ell-1}, W_{h_v}^{\ell-1}, e_{uv})$$

όπου  $e_{uv}$  το διάνυσμα με τις ιδιότητες κάθε ακμής .

Συνοψίζοντας, τα Γραφονευρωνικά Δίκτυα ενισχύουν το βασικό μοντέλο των GNNs με μηχανισμούς προσοχής, επιτρέποντας στο δίκτυο να επικεντρώνεται σε πιο σημαντικούς γείτονες κατά την παραγωγή των αναπαραστάσεων. Αν και οι αρχικές εξισώσεις επικεντρώνονται κυρίως στις αναπαραστάσεις (embeddings) των κόμβων, στην πράξη οι GAT μπορούν να αξιοποιήσουν και χαρακτηριστικά των ακμών, κάτι που αποδεικνύεται κρίσιμο σε προβλήματα όπως το Traveling Salesman Problem, όπου η πληροφορία των αποστάσεων ενσωματώνεται μέσω των ακμών.

Πριν όμως περάσουμε στην πρακτική εφαρμογή των GAT για την εκτίμηση της βέλτιστης λύσης του TSP, είναι χρήσιμο να παρουσιαστούν οι διαφορετικοί τρόποι με τους οποίους τα GNNs μπορούν να χρησιμοποιηθούν στο πλαίσιο της μηχανικής μάθησης σε γράφους. Οι έννοιες της εποπτευόμενης και ημι-εποπτευόμενης μάθησης, καθώς και της επαγωγικής (inductive) και μεταγωγικής (transductive) προσέγγισης, είναι ουσιώδεις για την κατανόηση των επιλογών κατά τον σχεδιασμό ενός μοντέλου βασισμένου σε γράφους.

### 3.3 Κατηγορίες Εκπαίδευσης Νευρωνικών Δικτύων Γράφων

Κατά την εφαρμογή Νευρωνικών Δικτύων Γράφων (GNNs), υπάρχουν διαφορετικοί τρόποι εκπαίδευσης, οι οποίοι εξαρτώνται τόσο από τη διαθεσιμότητα τιμών στις μεταβλητές στόχους (ετικέτες) στον γράφο όσο και από τη φύση της γενίκευσης που επιδιώκουμε να επιτύχουμε. Η πρώτη διάκριση αφορά το αν οι κόμβοι ή ακμές του γράφου συνοδεύονται ή όχι από ετικέτες, οδηγώντας σε σενάρια επιβλεπόμενης ή ημι-επιβλεπόμενης μάθησης. Η δεύτερη διάκριση σχετίζεται με το αν θέλουμε να κάνουμε προβλέψεις σε νέους, άγνωστους γράφους ή εντός του γράφου εκπαίδευσης, κάτι που διαφοροποιεί την επαγωγική (inductive) από τη μεταγωγική (transductive) μάθηση.

Στις υποενότητες που ακολουθούν, θα εξετάσουμε αναλυτικά κάθε μία από αυτές τις διακρίσεις, με έμφαση στο πώς συνδέονται με την αρχιτεκτονική και τις δυνατότητες των GNNs και ειδικότερα των Νευρωνικών Δικτύων Γράφων με μηχανισμούς προσοχής (GATs).

#### 3.3.1 Επιτηρούμενη και Ημι-επιτηρούμενη Μάθηση σε Γράφους

Στην εφαρμογή νευρωνικών δικτύων σε γράφους, ο τρόπος με τον οποίο εκπαιδεύεται ένα μοντέλο εξαρτάται σε μεγάλο βαθμό από το πόσες πληροφορίες είναι διαθέσιμες για την μεταβλητή που επιθυμούμε να προβλέψουμε, καθώς και από τον τύπο της πρόβλεψης που επιδιώκουμε. Στο πλαίσιο αυτό, διακρίνονται δύο βασικές μορφές μάθησης: η επιτηρούμενη (supervised) και η ημι-επιτηρούμενη (semi-supervised).

Με τον όρο "διαθέσιμη ετικέτα" εννοούμε ότι, κατά την εκπαίδευση, γνωρίζουμε ήδη την τιμή της μεταβλητής που θέλουμε να εκτιμήσουμε. Αυτό μπορεί να αφορά ένα χαρακτηριστικό κόμβου, μια ιδιότητα ακμής ή ακόμα και μια τιμή που σχετίζεται με ολόκληρο τον γράφο.

Αντίστοιχα, τα προβλήματα μάθησης σε γράφους μπορούν να αφορούν την πρόβλεψη είτε διακριτών κατηγοριών είτε συνεχών ποσοτήτων και ταξινομούνται συνήθως σε τρεις βασικές κατηγορίες, ανάλογα με το επίπεδο του γράφου στο οποίο εστιάζει το πρόβλημα:

- Σε επίπεδο κόμβων: επιδιώκεται η πρόβλεψη μιας ιδιότητας κάθε κόμβου, όπως η κατηγορία στην οποία ανήκει ή κάποια ποσοτική τιμή (π.χ. πιθανότητα ή βαθμολογία).
- Σε επίπεδο ακμών: στόχος είναι να προβλεφθεί είτε η ύπαρξη ή ο τύπος της σχέσης μεταξύ δύο κόμβων (σε προβλήματα κατηγοριοποίησης), είτε κάποιο συνεχές χαρακτηριστικό της (όπως το βάρος, η απόσταση ή το κόστος μετάβασης).
- Σε επίπεδο γράφων: πρόκειται για την εκτίμηση μιας ιδιότητας που χαρακτηρίζει ολόκληρο τον γράφο, είτε πρόκειται για διακριτή (όπως η κατηγορία ενός μορίου) είτε για συνεχή τιμή (π.χ. η συνολική ενεργειακή σταθερότητα).

Στην επιτηρούμενη μάθηση, όλες οι τιμές της μεταβλητής στόχου είναι γνωστές κατά την εκπαίδευση. Το μοντέλο εκπαιδεύεται ώστε να ελαχιστοποιήσει το σφάλμα μεταξύ των πραγματικών και των προβλεπόμενων τιμών, χρησιμοποιώντας συνήθως μια συνάρτηση κόστους όπως η "διεντροπία"(cross entropy) στην περίπτωση προβλημάτων κατηγοριοποίησης. Η αξιολόγηση γίνεται σε σύνολα δεδομένων που δεν χρησιμοποιήθηκαν στην εκπαίδευση, ώστε να εκτιμηθεί η ικανότητα γενίκευσης του μοντέλου.

Ένα παράδειγμα επιτηρούμενης μάθησης αφορά έναν γράφο άρθρων, όπου κάθε κόμβος αντιστοιχεί σε μία δημοσίευση και οι ακμές δηλώνουν παραπομπές. Αν γνωρίζουμε για όλα τα άρθρα το επιστημονικό πεδίο στο οποίο ανήκουν, μπορούμε να εκπαιδεύσουμε ένα GNN ώστε να προβλέπει την κατηγορία νέων άρθρων με βάση τις συνδέσεις τους.

Αντίθετα, στην ημι-επιτηρούμενη μάθηση μόνο ένα μέρος των κόμβων (ή ακμών ή γράφων) φέρει τιμές για τη μεταβλητή-στόχο. Το μοντέλο βασίζεται σε αυτές τις περιορισμένες πληροφορίες και ταυτόχρονα αξιοποιεί τη δομή του γράφου, ώστε να διαχύσει τη διαθέσιμη γνώση και να παράγει προβλέψεις για τα μη επισημασμένα στοιχεία. Η συνάρτηση κόστους υπολογίζεται μόνο για τα δείγματα με γνωστή τιμή στόχου, ενώ η γενίκευση επιτυγχάνεται κυρίως μέσω της τοπικής πληροφορίας και της γειτονικότητας.

Ένα παράδειγμα ημι-επιτηρούμενης μάθησης προερχόμενο από τον ακαδημαϊκό χώρο: σε έναν γράφο φοιτητών και μαθημάτων, όπου οι ακμές υποδεικνύουν εγγραφή σε κάποιο μάθημα, μπορεί να επιδιώκεται η πρόβλεψη του τελικού βαθμού. Αν γνωρίζουμε τους βαθμούς μόνο για ένα υποσύνολο φοιτητών, μπορούμε να χρησιμοποιήσουμε ημι-επιτηρούμενη μάθηση ώστε να εκτιμήσουμε τους βαθμούς των υπολοίπων.

Σε κάθε περίπτωση, η μορφή μάθησης που επιλέγεται πρέπει να λαμβάνει υπόψη τόσο τη διαθεσιμότητα δεδομένων όσο και τη φύση του προβλήματος. Επιπλέον, η αξιολόγηση της προγνωστικής ικανότητας του μοντέλου γίνεται πάντα σε δεδομένα που δεν έχουν χρησιμοποιηθεί στην εκπαίδευση, ώστε να διασφαλίζεται ότι το μοντέλο μαθαίνει ουσιώδεις συσχετίσεις και όχι απλώς να απομνημονεύει περιπτώσεις.

### 3.3.2 Επαγωγική και Μεταγωγική Μάθηση σε Γράφους

Μια άλλη διάκριση της μηχανικής μάθησης σε δομές γράφων μπορεί να γίνει ανάλογα με τον τρόπο διάρθρωσης των δεδομένων και το είδος της γενίκευσης που επιδιώκεται από το μοντέλο. Δύο βασικές στρατηγικές σε αυτό το πλαίσιο είναι η επαγωγική (inductive) και η μεταγωγική (transductive) μάθηση.

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

Στην επαγωγική μάθηση, κάθε παρατήρηση αντιστοιχεί σε έναν ανεξάρτητο γράφο. Το μοντέλο εκπαιδεύεται σε ένα σύνολο τέτοιων γραφών και επιδιώκει να μάθει γενικούς κανόνες που μπορούν να εφαρμοστούν σε άγνωστους γράφους κατά την διαδικασία εκπαίδευσης. Η γενίκευση εδώ συντελείται από το ειδικό στο γενικό: από συγκεκριμένες, γνωστές περιπτώσεις επιχειρείται η εξαγωγή ενός γενικού κανόνα. Ένα αντιπροσωπευτικό παράδειγμα είναι ένα σύστημα ανάλυσης μοριακών δομών, το οποίο εκπαιδεύεται πάνω σε ένα πλήθος από διαφορετικά μόρια (θεωρούμενα ως γράφοι) ώστε να προβλέπει την τοξικότητα ή άλλες φυσικοχημικές ιδιότητες για νέα, άγνωστα μόρια.

Αντίθετα, στη μεταγωγική μάθηση, το πρόβλημα διατυπώνεται πάνω σε έναν ενιαίο γράφο, ο οποίος περιλαμβάνει τόσο τα δεδομένα εκπαίδευσης όσο και εκείνα για τα οποία θέλουμε να πραγματοποιήσουμε προβλέψεις. Σκοπός είναι η διάδοση πληροφορίας **εντός του ίδιου γραφήματος**, δηλαδή η γενίκευση επιτελείται **από το ειδικό στο ειδικό**. Τυπικό παράδειγμα αποτελεί ένα κοινωνικό δίκτυο όπου, γνωρίζοντας τις προτιμήσεις ή ιδιότητες ενός υποσυνόλου χρηστών (κόμβων), επιδιώκουμε να εκτιμήσουμε τις αντίστοιχες ιδιότητες για τους υπόλοιπους χρήστες, αξιοποιώντας τη δομή των μεταξύ τους σχέσεων.

Ένας εμπειρικός κανόνας είναι ότι η επαγωγική μάθηση συναντάται συχνότερα σε επιτηρούμενα σενάρια, όπου η τιμή της μεταβλητής-στόχου (όπως κατηγορία, αριθμητική τιμή ή κάποιο χαρακτηριστικό) είναι γνωστή για όλα τα δείγματα του εκπαιδευτικού συνόλου. Αντίθετα, η μεταγωγική μάθηση εφαρμόζεται συνήθως σε ημι-επιτηρούμενα περιβάλλοντα, στα οποία η πληροφορία-στόχος είναι διαθέσιμη μόνο για ένα υποσύνολο των κόμβων ή ακμών, και το μοντέλο καλείται να εκμεταλλευτεί τη δομή του γράφου ώστε να τη μεταφέρει στο υπόλοιπο δίκτυο.

Αξίζει να υπογραμμιστεί ότι, παρόλο που στην επαγωγική προσέγγιση είναι συνηθισμένο το εκπαιδευτικό σύνολο να αποτελείται από πολλαπλούς, διακριτούς γράφους ως παρατηρήσεις, αυτό δεν συνιστά αυστηρό ή εγγενή περιορισμό. Σε ορισμένα σενάρια, ακόμη και ένας ενιαίος γράφος μπορεί να διαμεριστεί σε κατάλληλα υποσύνολα, διαμορφώνοντας έτσι ένα επαγωγικό πλαίσιο. Συνεπώς, ο διαχωρισμός μεταξύ επαγωγικής και μεταγωγικής μάθησης βασίζεται κυρίως στον στόχο γενίκευσης του μοντέλου, και όχι αποκλειστικά στη δομή των δεδομένων.

### **3.4 Μοντελοποίηση του Προβλήματος του Πλανόδιου Πωλητή με Νευρωνικό Δίκτυο**

Η ενότητα αυτή επικεντρώνεται στη μοντελοποίηση του Προβλήματος του Πλανόδιου Πωλητή με όρους θεωρίας γραφημάτων, ως βάση για την υλοποίηση ενός νευρωνικού δικτύου το οποίο επιχειρεί να εκτιμήσει βέλτιστες λύσεις. Αρχικά, περιγράφεται η αναπαράσταση του προβλήματος ως γράφος με συγκεκριμένα χαρακτηριστικά και περιορισμούς. Στη συνέχεια, παρουσιάζεται η επιλογή του τύπου του γραφήματος και των δεδομένων που τροφοδοτούν το νευρωνικό δίκτυο.

#### **3.4.1 Αναπαράσταση του Συμμετρικού TSP ως Μη Κατευθυνόμενος Γράφος**

Στη συγκεκριμένη εργασία μοντελοποιείται η συμμετρική εκδοχή του TSP, όπου η απόσταση μεταξύ δύο πόλεων είναι ίδια και στις δύο κατευθύνσεις. Το πρόβλημα περιγράφεται ως εξής:

- Θεωρούμε έναν μη κατευθυνόμενο πλήρες γράφο  $G = (V, E)$  όπου:
  - $V = \{v_1, v_2, \dots, v_n\}$  το σύνολο των κορυφών, αντιστοιχεί στα σημεία (πόλεις) που πρέπει να επισκεφθούμε,

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

$E = \{\{v_i, v_j\}: v_i, v_j \in V, i \neq j\}$  το σύνολο όλων των ακμών μεταξύ των κορυφών, καθώς το γράφημα είναι πλήρες.

- Κάθε κορυφή  $v_i$  διαθέτει ως χαρακτηριστικό τις δυσδιάστατες συντεταγμένες  $(x_i, y_i)$  στο επίπεδο, που αντιστοιχούν στη γεωγραφική θέση της πόλης.
- Το βάρος κάθε ακμής  $e_{ij} = \{v_i, v_j\} \in E$  υπολογίζεται ως η ευκλείδεια απόσταση μεταξύ των αντίστοιχων κορυφών:

$$c_{ij} = \|v_i - v_j\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

- Η λύση του προβλήματος είναι ένας Hamiltonian κύκλος, δηλαδή ένας κύκλος που επισκέπτεται κάθε κορυφή ακριβώς μία φορά και επιστρέφει στην αρχική, με στόχο τη ελαχιστοποίηση του συνολικού βάρους του κύκλου:

$$\min_{C \in H} \sum_{\{v_i, v_j\} \in C} c_{ij}$$

όπου  $H$  το σύνολο όλων των Hamiltonian κύκλων στον γράφο  $G$ .

### 3.4.2 Νευρωνικό Δίκτυο για την Εκτίμηση της Λύσης του TSP

Για την προσέγγιση του προβλήματος του Πλανόδιου Πωλητή, σχεδιάστηκε και υλοποιήθηκε ένα Νευρωνικό Δίκτυο Γράφου (Graph Neural Network – GNN), με στόχο να προβλέψει ποιες ακμές από έναν πλήρως συνδεδεμένο γράφο είναι πιθανό να συμμετέχουν στη βέλτιστη λύση. Η βασική ιδέα είναι ότι, μέσω της κατάλληλης επεξεργασίας της σχετικής θέσης των κορυφών και την αξιοποίηση όλων των αποστάσεων μεταξύ τους, το μοντέλο μπορεί να μάθει να εκτιμά τη σημασία κάθε ακμής ως προς τη διαδρομή ελάχιστου κόστους που επισκέπτεται κάθε κορυφή ακριβώς μία φορά.

#### Δομή του Νευρωνικού Δικτύου

Η αρχιτεκτονική του δικτύου αποτελείται από τέσσερις βασικές ενότητες:

1. Κωδικοποιητής Κόμβων (Node Encoder): Κάθε κόμβος στον γράφο αντιστοιχεί σε ένα γεωμετρικό σημείο στο επίπεδο, με τις συντεταγμένες του να παρέχουν πληροφορία για τη θέση του. Ο κωδικοποιητής κόμβων (node encoder) προβάλλει αυτές τις δυσδιάστατες συντεταγμένες σε ένα πιο εκφραστικό, πολυδιάστατο διανυσματικό χώρο.
2. Κωδικοποιητής Ακμών (Edge Encoder): Για κάθε ακμή, χρησιμοποιείται ως είσοδος η ευκλείδεια απόσταση μεταξύ των δύο άκρων της. Ο **κωδικοποιητής ακμών** λειτουργεί ανάλογα με τον κωδικοποιητή κόμβων: μετατρέπει το μονοδιάστατο χαρακτηριστικό της απόστασης σε μια πλουσιότερη διανυσματική αναπαράσταση.
3. Στρώσεις Μηχανισμού Προσοχής σε Γράφους (Graph Attention Layers): Ο πυρήνας του μοντέλου είναι δύο διαδοχικές στρώσεις τύπου Νευρωνικών Δικτύων Γράφων με μηχανισμούς προσοχής (GAT). Αυτές οι στρώσεις επιτρέπουν στο κάθε κόμβο να ενημερώνει τα χαρακτηριστικά του με βάση τα χαρακτηριστικά των γειτονικών του κόμβων και των ακμών που τους συνδέουν, δίνοντας μεγαλύτερη "προσοχή" (βάρος) σε εκείνους που φαίνεται να είναι πιο σημαντικοί.

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

4. Μονάδα Πρόβλεψης Ακμών (Edge Prediction Module): Στο τελικό στάδιο, για κάθε ακμή του γράφου συνενώνονται τα χαρακτηριστικά των δύο ακριανών κόμβων της μαζί με το ενισχυμένο χαρακτηριστικό της ίδιας της ακμής. Αυτά εισάγονται σε έναν νέο πολυεπίπεδο αντιληπτικό μηχανισμό (MLP -Multi-Layer Perceptron), ο οποίος επιστρέφει έναν αριθμό στο διάστημα [0, 1] που αντιπροσωπεύει την πιθανότητα η συγκεκριμένη ακμή να συμμετέχει στη βέλτιστη διαδρομή.

```
TSPGNN(
  (node_encoder): Sequential(
    (0): Linear(in_features=2, out_features=64, bias=True)
    (1): LeakyReLU(negative_slope=0.2)
    (2): LayerNorm((64,), eps=1e-05, elementwise_affine=True)
  )
  (edge_encoder): Sequential(
    (0): Linear(in_features=1, out_features=32, bias=True)
    (1): LeakyReLU(negative_slope=0.2)
    (2): Linear(in_features=32, out_features=16, bias=True)
    (3): LayerNorm((16,), eps=1e-05, elementwise_affine=True)
  )
  (conv1): GATConv(64, 64, heads=4)
  (norm1): LayerNorm((256,), eps=1e-05, elementwise_affine=True)
  (conv2): GATConv(256, 64, heads=1)
  (norm2): LayerNorm((64,), eps=1e-05, elementwise_affine=True)
  (edge_mlp): Sequential(
    (0): Linear(in_features=144, out_features=64, bias=True)
    (1): LeakyReLU(negative_slope=0.1)
    (2): Linear(in_features=64, out_features=1, bias=True)
  )
)
```

Σχήμα 3.4 Αναπαράσταση Αρχιτεκτονικής Νευρωνικού Δικτύου στην python

#### Επεξεργασία της εξόδου του Νευρωνικού Δικτύου

Η ροή δεδομένων έχει ως εξής:

1. Δίνεται ως είσοδος ένας πλήρης γράφος.

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

2. Το νευρωνικό δίκτυο υπολογίζει μια πιθανότητα για κάθε ακμή να ανήκει στη βέλτιστη λύση του TSP.
3. Οι πιθανότητες αυτές, ωστόσο, δεν εγγυώνται την παραγωγή ενός έγκυρου κύκλου Hamilton, δηλαδή μιας διαδρομής που επισκέπτεται όλους τους κόμβους ακριβώς μία φορά και επιστρέφει στο αρχικό σημείο.

Για να μετατραπούν οι ανεξάρτητες προβλέψεις του μοντέλου σε μια πλήρη και έγκυρη διαδρομή, χρησιμοποιείται ένας ευρετικός αλγόριθμος εξερεύνησης πιθανών κύκλων Hamilton (beam search).

#### *Beam Search: Δημιουργία Κύκλου Hamilton*

Ο beam search είναι ένας ευρετικός αλγόριθμος εξερεύνησης χώρου καταστάσεων. Ξεκινώντας από έναν αρχικό κόμβο, κατασκευάζει επαναληπτικά μερικές διαδρομές (beams), κάθε φορά προσθέτοντας έναν νέο μη επισκέψιμο κόμβο με βάση την πιθανότητα μετάβασης από τον τρέχοντα. Σε κάθε βήμα κρατάει μόνο τις καλύτερες διαδρομές (π.χ. τις 5 πιο υποσχόμενες), ώστε να περιορίσει τον υπολογιστικό φόρτο. Η διαδικασία συνεχίζεται έως ότου προκύψουν πλήρεις διαδρομές που περιλαμβάνουν όλους τους κόμβους. Τέλος, η διαδρομή με τη μεγαλύτερη συνολική πιθανότητα επιλέγεται ως η εκτιμώμενη λύση.

Ουσιαστικά, ο αλγόριθμος beam search λειτουργεί σαν ένας έξυπνος «οδηγός» που χρησιμοποιεί τις πιθανότητες του μοντέλου για να εντοπίσει βήμα προς βήμα έναν κύκλο Hamilton, χωρίς να εξετάσει εξαντλητικά όλες τις δυνατές διαδρομές.

## **4. Μεθοδολογία, Πειραματισμός και Αξιολόγηση**

Στο παρόν κεφάλαιο παρουσιάζεται αναλυτικά η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη, την εκπαίδευση και την αξιολόγηση του υβριδικού μοντέλου επίλυσης – εκτίμησης του προβλήματος του Πλανόδιου Πωλητή. Αρχικά, περιγράφεται η διαδικασία παραγωγής βέλτιστων λύσεων μέσω ενός μοντέλου μεικτού ακέραιου-γραμμικού προγραμματισμού, βασισμένου στη διατύπωση DFJ, το οποίο υλοποιήθηκε μέσω της βιβλιοθήκης SciPy και επιλύθηκε με την χρήση του solver HiGHS.

Στη συνέχεια, αναλύεται η διαδικασία δημιουργίας του συνόλου εκπαίδευσης, καθώς και οι αρχές σχεδιασμού της κατανομής των δεδομένων, προκειμένου να διασφαλιστεί η ποικιλομορφία του δείγματος και η ικανότητα γενίκευσης του μοντέλου. Ακολούθως, παρουσιάζεται η διαδικασία εκπαίδευσης του Νευρωνικού Δικτύου Γράφου (Graph Neural Network – GNN), συνοδευόμενη από μια σύντομη αναφορά στα συμπεράσματα που προέκυψαν από τον πειραματισμό με διαφορετικές υπερπαραμέτρους.

Έμφαση δίνεται στην τελική φάση πρόβλεψης, όπου το νευρωνικό δίκτυο εκτιμά πιθανότητες για κάθε ακμή του γραφήματος, και στην τεχνική κατασκευής κύκλων Hamilton μέσω αλγορίθμου beam search. Τέλος, ακολουθεί σχολιασμός των αποτελεσμάτων και σύντομη περιγραφή της υλοποίησης σε Python, συνοδευόμενη από παραπομπή στο δημόσιο αποθετήριο του έργου στο GitHub.

Για μεγαλύτερη σαφήνεια, η συνολική ροή συνοψίζεται στα εξής στάδια:

- **Παραγωγή βέλτιστων λύσεων:** Μικρά μεγέθους TSP προβλήματα επιλύονται μέσω μοντέλου αμιγούς ακέραιου-γραμμικού προγραμματισμού (MILP) βασισμένου στη διατύπωση DFJ, χρησιμοποιώντας τον solver HiGHS.

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

- **Δημιουργία συνόλου εκπαίδευσης:** Από τις βέλτιστες λύσεις του προηγούμενου βήματος εξάγονται οι ακμές που συμμετέχουν στις βέλτιστες διαδρομές και συγκροτούνται τα δεδομένα εκπαίδευσης του GNN.
- **Εκπαίδευση GNN:** Το νευρωνικό δίκτυο γράφου μαθαίνει να προβλέπει την πιθανότητα κάθε ακμής να ανήκει στη βέλτιστη διαδρομή, αξιοποιώντας τα δεδομένα από το προηγούμενο βήμα.
- **Καθοδηγούμενη αναζήτηση:** Οι πιθανότητες που παράγει το GNN χρησιμοποιούνται για να καθοδηγήσουν έναν αλγόριθμο *beam search*, ο οποίος κατασκευάζει εφικτές διαδρομές.

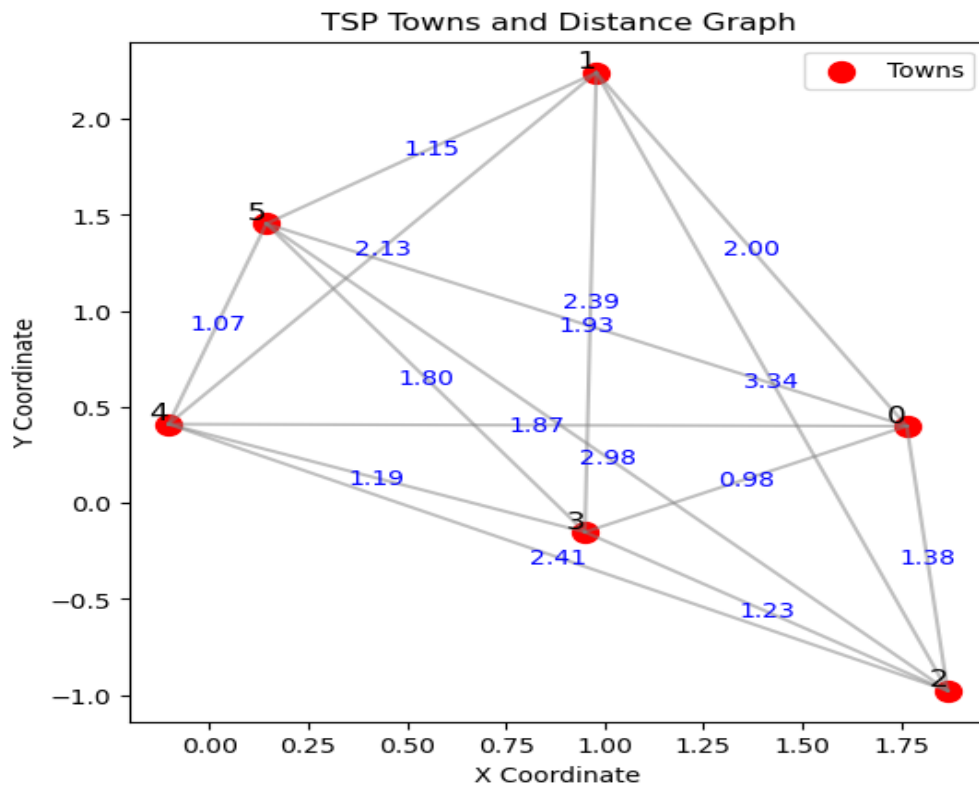
#### 4.1 Δημιουργία Βέλτιστων Λύσεων με Ακέραιο Γραμμικό Προγραμματισμό

Η παραγωγή των βέλτιστων λύσεων για τα προβλήματα εκπαίδευσης βασίστηκε στην μαθηματική διατύπωση DFJ (Dantzig-Fulkerson-Johnson), η οποία έχει αναλυθεί στην ενότητα 2.1.2. Η υλοποίηση πραγματοποιήθηκε με τη βοήθεια της συνάρτησης `linprog` από τη βιβλιοθήκη **SciPy**, αξιοποιώντας την προεπιλεγμένη μέθοδο `branch-and-bound` θέτοντας ρυθμίζοντας την παράμετρο `integrality=1`.

Για την αποφυγή ανεπιθύμητων λύσεων με πολλαπλούς υποκύκλους (*subtours*), εφαρμόστηκε η τεχνική **branch - and - cut**, η οποία ενισχύει δυναμικά το μοντέλο με **περιορισμούς αποκοπής** (*subtour elimination constraints*) κάθε φορά που εντοπίζεται μη αποδεκτή λύση. Οι περιορισμοί αυτοί δεν είναι σταθερά προκαθορισμένοι, αλλά προστίθενται κατά την πορεία της επίλυσης, με στόχο την κατασκευή ενός έγκυρου κύκλου Hamilton.

Για την παρουσίαση της μεθοδολογίας, εξετάζεται ένα απλό παράδειγμα με **6 πόλεις**, στο οποίο παρουσιάζονται:

- Το γράφημα με τις ακμές και τις αντίστοιχες αποστάσεις (Σχήμα 4.1),
- Η τελική βέλτιστη διαδρομή και η αντίστοιχη ακολουθία επισκέψεων,
- Η μήτρα ισοτικών περιορισμών (*equality constraints*), η οποία διασφαλίζει πως κάθε κόμβος συμμετέχει ακριβώς σε δύο μεταβάσεις (μία εισερχόμενη και μία εξερχόμενη ακμή),
- Η μήτρα ανισοτικών περιορισμών (*inequality constraints*), η οποία προστέθηκε δυναμικά κατά την επίλυση για την εξάλειψη υποκύκλων,
- Απόσπασμα Python κώδικα που δημιουργεί και επιλύει το μοντέλο.



Σχήμα 4.1 Πόλεις και ζεύγη

Παράδειγμα επίλυσης με 6 πόλεις

Κορυφή 1	Κορυφή 2	Απόσταση
0	1	2.00
0	2	1.38
0	3	0.98
0	4	1.87
0	5	1.93
1	2	3.34
1	3	2.39
1	4	2.13
1	5	1.15
2	3	1.23
2	4	2.41
2	5	2.98
3	4	1.19

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

Κορυφή 1	Κορυφή 2	Απόσταση
3	5	1.80
4	5	1.07

Πίνακας 4.1: Μετασχηματισμένα δεδομένα εισόδου TSP με 6 σημεία

\*\* Οι αποστάσεις είναι συμμετρικές:  $d(i, j) = d(j, i)$ , και δεν εμφανίζονται διπλότυπα.)

#### Ισοτικοί Περιορισμοί (Equality Constraints)

Η παρακάτω μήτρα (6 x 15) εξασφαλίζει ότι κάθε πόλη συμμετέχει σε ακριβώς δύο ακμές (μία εξερχόμενη και μία εισερχόμενη). Οι στήλες αντιστοιχούν στις μεταβλητές  $x_{ij}$ , που υποδηλώνουν εάν η ακμή μεταξύ των πόλεων  $i$  και  $j$  περιλαμβάνεται στη λύση.

Πόλη \ Μετάβαση	$x_{01}$	$x_{02}$	$x_{03}$	$x_{04}$	$x_{05}$	$x_{12}$	$x_{13}$	...	$x_{45}$
0	1	1	1	1	1	0	0	...	0
1	1	0	0	0	0	1	1	...	0
2	0	1	0	0	0	1	0	...	0
3	0	0	1	0	0	0	1	...	1
4	0	0	0	1	0	0	0	...	1
5	0	0	0	0	1	0	0	...	1

Πίνακας 4.2: Ισοτικοί περιορισμοί TSP του παραδείγματος του πίνακα 4.1

#### Ανισοτικοί Περιορισμοί (Inequality Constraints)

Κατά την επίλυση, ο επιλυτής εντόπισε μια αρχική λύση που περιλάμβανε **δύο υποκύκλους**, γεγονός που ενεργοποίησε τη δυναμική προσθήκη τεμνόντων επιπέδων (cutting planes). Συγκεκριμένα, προστέθηκαν οι εξής δύο ανισοτικοί περιορισμοί:

Περιορισμός	$x_{01}$	$x_{02}$	$x_{03}$	$x_{04}$	$x_{05}$	$x_{12}$	$x_{13}$	$x_{14}$	$x_{15}$	$x_{23}$	$x_{24}$	$x_{34}$	$x_{35}$	$x_{45}$
1	0	1	1	0	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	1	1	0	0	0	0	1

Πίνακας 4.3: Δυναμικά εισηγμένοι ανισοτικοί περιορισμοί TSP του παραδείγματος του πίνακα 4.1

Επεξήγηση:

- Κάθε γραμμή υποδηλώνει έναν περιορισμό της μορφής:

$$x_{ij} + x_{jk} + \dots \leq |Q| - 1 \quad \forall Q \subseteq \{1, \dots, n\}, \quad |Q| \geq 2$$

όπου οι μηδενικές τιμές σημαίνουν ότι η αντίστοιχη ακμή **δεν** συμμετέχει στον υποκύκλο, ενώ οι μονάδες υποδεικνύουν συμμετοχή της ακμής στον κύκλο που πρέπει να αποκοπεί.

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

Ερμηνεύοντας τον πίνακα που δίνετε πιο πάνω συμπεραίνουμε ότι οι κύκλοι που αφαιρέθηκαν από το σύνολο των εφικτών λύσεων όπως ορίζεται από τους υπόλοιπους περιορισμούς είναι :

0 - 2 - 3 - 0 ή 0 - 3 - 2 - 0  
1 - 4 - 5 - 1 ή 1 - 5 - 4 - 1

\*Λόγω συμμετρίας μπορούμε να διαβάσουμε την ακολουθία κορυφών από τα δεξιά προς τα αριστερά.

Για τη δημιουργία του παραδείγματος χρησιμοποιήθηκε ο παρακάτω κώδικας στην γλώσσα προγραμματισμού python:

```
import sys
from pathlib import Path
base_dir = Path.cwd().parent.resolve()
sys.path.append(str(base_dir))
from tsp import TSP
import numpy as np

np.random.seed(0)
num_towns = 6
towns = list(range(num_towns))
coords = np.random.randn(num_towns,2)
tsp = TSP(coordinates=coords, distance_metric='euclidean')
tsp.solve(verbose=True)

print(f"Equality Constraint Matrix:
{tsp.equality_constraints_matrix}")
print(f"Inequality Constraint Matrix:
{tsp.inequality_constraints_matrix}")
print(f"Optimal tour: {tsp.optimal_tour}")
tsp.plot_optimal_tour()
```

Σχήμα 4.2: Απόσπασμα κώδικα python για το TSP του παραδείγματος του πίνακα 4.1

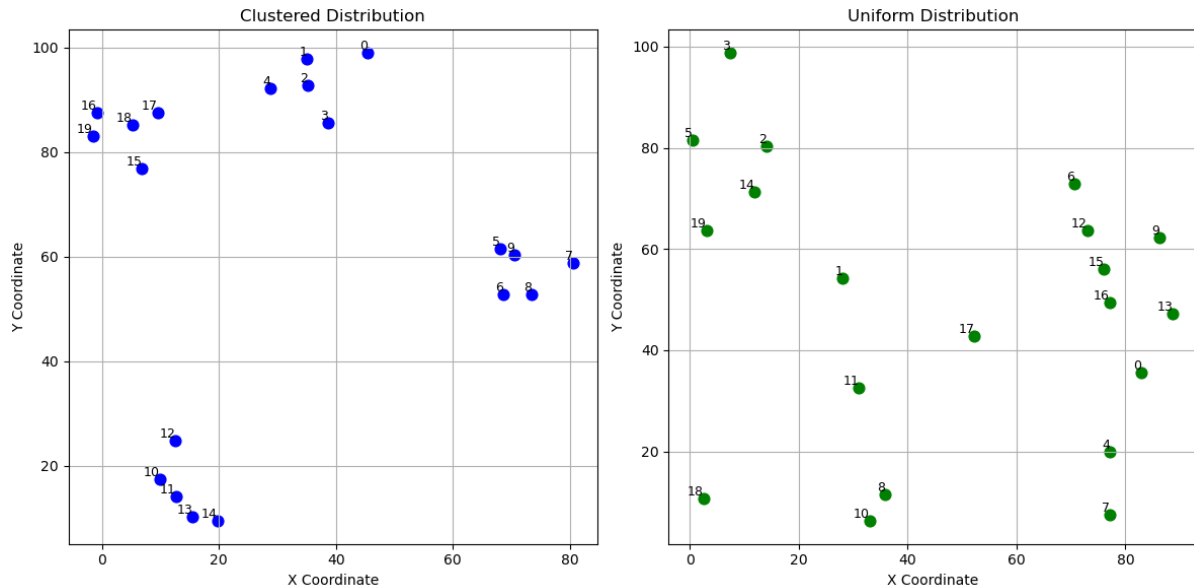
## 4.2 Παραγωγή συνόλων δεδομένων

Η διαδικασία παραγωγής δεδομένων εκπαίδευσης αποτελεί κρίσιμο βήμα για την επιτυχία υλοποίησης του νευρωνικού δικτύου. Συνολικά παράχθηκαν 30.000 παραδείγματα του προβλήματος TSP, για διαφορετικές κατανομές γεωγραφικής τοποθέτησης των πόλεων:

- 15.000 παραδείγματα δημιουργήθηκαν με βάση την ομοιόμορφη (uniform) κατανομή σε δισδιάστατο χώρο.
- 15.000 παραδείγματα αφορούσαν συστάδες πόλεων (clustered distribution), με αριθμό συστάδων από 2 έως 5. Η προσέγγιση αυτή προσομοιώνει σενάρια όπου οι πόλεις ομαδοποιούνται γύρω από φυσικά εμπόδια, όπως βουνά ή θάλασσες.

- Για κάθε κατανομή, δημιουργήθηκαν 5.000 παραδείγματα με αριθμό κόμβων 20, 35 και 50 αντίστοιχα.

Ο σκοπός της διαφοροποίησης αυτής ήταν να διασφαλιστεί η ποικιλομορφία στο σύνολο εκπαίδευσης και αξιολόγησης, ενισχύοντας την ικανότητα γενίκευσης του μοντέλου και περιορίζοντας τον κίνδυνο μεροληψίας προς συγκεκριμένες τοπολογικές δομές.



**Σχήμα 4.3** ομοιόμορφης - συσταδοποιημένης κατανομής

Κάθε παράδειγμα επιλύθηκε χρησιμοποιώντας τον ILP solver που περιεγράφηκε στην προηγούμενη ενότητα, και τα αποτελέσματα αποθηκεύτηκαν σε συλλογή MongoDB. Η επιλογή αυτή επέτρεψε την αποδοτική αποθήκευση, ανάκτηση και διαχείριση των δεδομένων.

```

{
  _id: ObjectId('68028e2172cb27455151e3fe'),
  number_of_points: 20,
  distance_metric: 'euclidean',
  coordinates: [
    [ 22.035257719233492, 46.881231688206405 ],
    [ 17.556910672546643, 26.213026103020376 ],
    ....
  ],
  decision_variables: [0,0,1,0, ....],
  optimal_tour: [0,2,3 ...],
  minimum_distance: 378.41193424785644,
  subtour_revisions: 2,
  elapsed_time: 0.00012679166672266244
}

```

Σχήμα 4.4: Ενδεικτική εγγραφή λυμένου προβλήματος TSP σε συλλογή MongoDB

Το σύνολο χωρίστηκε σε τρία υποσύνολα:

- Εκπαίδευση: 70% των δειγμάτων
- Επαλήθευση (validation): 10% των δειγμάτων κατά τη διάρκεια εκπαίδευσης
- Αξιολόγηση (test set): 20% των δειγμάτων, πλήρως άγνωστο στο μοντέλο κατά την εκπαίδευση

### 4.3 Εκπαίδευση Νευρωνικού Δικτύου και Παραμετροποίηση

Η εκπαίδευση του νευρωνικού δικτύου πραγματοποιείται πάνω σε σύνολο δεδομένων όπου η μεταβλητή στόχος είναι γνωστή κατά τη φάση εκπαίδευσης (labeled data). Τα δεδομένα αυτά ανακτώνται από τη βάση δεδομένων MongoDB, και στην Εικόνα 4.4 παρουσιάζεται μια τυπική εγγραφή από τη συλλογή, η οποία περιλαμβάνει τα απαραίτητα χαρακτηριστικά του γράφου εισόδου — όπως οι Ευκλείδειες αποστάσεις μεταξύ κόμβων — καθώς και τη βέλτιστη λύση που έχει προκύψει μέσω του επιλυτή που περιεγράφηκε στην Ενότητα 4.1.

Αξίζει να διευκρινιστεί ότι, καθώς το πρόβλημα διατυπώνεται ως δυαδική κατηγοριοποίηση ακμών, η μεταβλητή στόχος είναι μια ακολουθία από 0 και 1, η οποία υποδηλώνει αν κάθε ακμή ανήκει ή όχι στη βέλτιστη διαδρομή. Οι τιμές αυτές αντιστοιχούν στο πεδίο **"decision variables"** της εικόνας 4.4.

Η έξοδος του μοντέλου δεν αποτελεί μια απόλυτη επιλογή ακμών, αλλά μια **πιθανότητα** για κάθε ακμή να ανήκει στη βέλτιστη λύση. Για τη δημιουργία μιας τελικής, αποδεκτής λύσης του TSP, εφαρμόζεται ο αλγόριθμος beam search πάνω στις προβλεπόμενες πιθανότητες.

Ο στόχος του παραπάνω αλγορίθμου είναι η κατασκευή έγκυρων και κοντά στο βέλτιστο λύσεων χωρίς την εξάντληση όλων των αποδεκτών λύσεων.

Η διαδικασία εκπαίδευσης περιλαμβάνει τα εξής βασικά στάδια:

1. Ανάκτηση και απεικόνιση δεδομένων: Τα δεδομένα φορτώνονται από τη MongoDB (όπως φαίνεται στην εικόνα 4.4) και διαμορφώνονται καταλλήλως για είσοδο στο δίκτυο.
2. Προ-επεξεργασία δεδομένων: Κατασκευή των εισόδων του μοντέλου με βάση τα χαρακτηριστικά ακμών (όπως οι Ευκλείδειες αποστάσεις), καθώς και διαχωρισμός του συνόλου σε σύνολα εκπαίδευσης, επαλήθευσης και αξιολόγησης.
3. Ρύθμιση παραμέτρων για την εκπαίδευση: Χρήση του συνόλου επαλήθευσης για βελτιστοποίηση παραμέτρων εκπαίδευσης όπως ο ρυθμός εκμάθησης, η επιλογή του βελτιστοποιητή και της συνάρτησης κόστους καθώς και το βάρος της θετικής κατηγορίας.
4. Εκπαίδευση: Βελτιστοποίηση των παραμέτρων του νευρωνικού δικτύου ώστε να ελαχιστοποιείται η απόκλιση μεταξύ των προβλεπόμενων πιθανοτήτων και των πραγματικών τιμών της μεταβλητής στόχου, χρησιμοποιώντας κατάλληλη συνάρτηση απωλειών (loss function).

Σε αυτό το σημείο, είναι κρίσιμο να αναγνωριστεί μια σημαντική ιδιαιτερότητα του προβλήματος: η έντονη ανισορροπία στη μεταβλητή στόχο. Καθώς ο γράφος εισόδου είναι πλήρως συνεκτικός, περιέχει συνολικά  $\frac{n(n-1)}{2}$  ακμές για  $n$  κόμβους. Ωστόσο, η βέλτιστη λύση περιλαμβάνει ακριβώς  $n$  ακμές (μία για κάθε μετάβαση του κύκλου Hamilton), με αποτέλεσμα οι θετικές περιπτώσεις να είναι αριθμητικά πολύ λιγότερες από τις **αρνητικές**. Συγκεκριμένα, το πλήθος των αρνητικών ακμών είναι  $\frac{n(n-1)}{2} - n$ , που αυξάνεται τετραγωνικά σε σχέση με το  $n$ .

Αυτή η ανισορροπία επηρεάζει άμεσα την αξιολόγηση της απόδοσης του μοντέλου. Ένα σύστημα μπορεί να εμφανίζει υψηλή ακρίβεια (accuracy) προβλέποντας σχεδόν όλες τις ακμές ως αρνητικές, όμως ένα τέτοιο μοντέλο αποτυγχάνει να αναγνωρίσει τις λίγες αλλά κρίσιμες θετικές περιπτώσεις. Για τον λόγο αυτό, χρησιμοποιήθηκαν μετρικές αξιολόγησης που αποτυπώνουν την ποιότητα του μοντέλου υπό συνθήκες ανισορροπίας.

Στο πλαίσιο αυτής της εργασίας, η PR-AUC σε συνδυασμό με την Ανάκληση (Εμβαδόν Καμπύλης Precision-Recall) θεωρήθηκαν πιο αντιπροσωπευτικά μέτρα απόδοσης από την απλή ακρίβεια. Το εμβαδόν της καμπύλης PR - AUC αποτυπώνει τη συνολική απόδοση του μοντέλου ως προς την ισορροπία μεταξύ ακρίβειας (precision) και ανάκλησης (recall) για διαφορετικές τιμές κατώφλιού ταξινόμησης. Το κατώφλι ταξινόμησης είναι μια αριθμητική τιμή (συνήθως μεταξύ 0 και 1) που καθορίζει πότε μια προβλεπόμενη πιθανότητα χαρακτηρίζεται ως θετική: αν η πιθανότητα μιας ακμής να ανήκει στη βέλτιστη λύση ξεπερνά αυτό το όριο, τότε η ακμή προβλέπεται ως θετική, διαφορετικά ως αρνητική.

Μεταβάλλοντας το κατώφλι, μεταβάλλονται και οι τιμές precision και recall. Η καμπύλη Precision-Recall καταγράφει αυτή τη μεταβολή και το PR-AUC είναι το εμβαδόν κάτω από την καμπύλη, με τιμές μεταξύ 0 και 1. Όσο μεγαλύτερο το PR-AUC, τόσο καλύτερη η ικανότητα του μοντέλου να εντοπίζει τις θετικές ακμές με λίγα ψευδώς θετικά. Σε αντίθεση με την ROC-AUC, η οποία λαμβάνει υπόψη και τις αρνητικές περιπτώσεις, η PR-AUC εστιάζει αποκλειστικά στην απόδοση του μοντέλου στην αναγνώριση της θετικής κατηγορίας — γεγονός που την καθιστά ιδανική για προβλήματα έντονης ανισορροπίας, όπως το TSP.

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

### Πειραματισμός

Στην πρώτη φάση, για την αξιολόγηση της συνεισφοράς των υπερπαραμέτρων εκπαίδευσης, χρησιμοποιήθηκε ένα υποσύνολο των διαθέσιμων δεδομένων, ώστε να είναι δυνατή η ταχεία αξιολόγηση διαφορετικών συνδυασμών παραμέτρων χωρίς υπερβολική υπολογιστική δαπάνη. Σε αυτό το πλαίσιο διερευνήθηκαν διάφοροι βελτιστοποιητές (AdamW, SGD, RMSprop), διαφορετικά μεγέθη batch (8, 16, 32) καθώς και επιλογές συναρτησών κόστους: η BCEWithLogitsLoss με βάρη θετικής κατηγορίας 3 και 5 και η Focal Loss. Για κάθε συνδυασμό υπερπαραμέτρων καταγράφηκαν οι μετρικές απόδοσης του μοντέλου, όπως η ανάκληση (Recall) και το PR-AUC, προκειμένου να αξιολογηθεί η ικανότητα του δικτύου να εντοπίζει τις θετικές ακμές. Τα αποτελέσματα καταγράφηκαν σε συγκεντρωτικούς πίνακες, προκειμένου να διευκολυνθεί η συστηματική σύγκριση των πειραματικών παραμέτρων και των δεικτών απόδοσης.

	criterion	PR_AUC		Recall		Precision	
		BCE	FocalLoss	BCE	FocalLoss	BCE	FocalLoss
optimizer	batch_size						
AdamW	8	0.54788	0.55734	0.70971	0.28229	0.53777	0.71072
	16	0.55481	0.56820	0.68457	0.24000	0.51979	0.76648
	32	0.55968	0.56793	0.62114	0.26000	0.56323	0.73649
RMSprop	8	0.56092	0.56121	0.67371	0.30971	0.56439	0.77707
	16	0.56243	0.56667	0.70343	0.32629	0.48507	0.70147
	32	0.56469	0.56554	0.70057	0.34800	0.50771	0.71672
SGD	8	0.55575	0.54149	0.66514	0.36000	0.55549	0.61477
	16	0.56048	0.54552	0.67029	0.37543	0.57647	0.61690
	32	0.56185	0.55257	0.66457	0.38400	0.55425	0.62570

Πίνακας 4.4: Αποτελέσματα πειραματισμού και υπερπαραμέτροι εκπαίδευσης

Έπειτα δίνοντας ιδιαίτερη βαρύτητα στην συνάρτηση κόστους με βάση τα αποτελέσματα του πρώτου πειραματισμού πραγματοποιήθηκε επανάληψη της εκπαίδευσης χρησιμοποιώντας βελτιστοποιητή RMSprop και μέγεθος παρτίδας 16, με μοναδική διαφοροποίηση τη συνάρτηση κόστους, αλλά πλέον εφαρμόζοντας το πλήρες διαθέσιμο σύνολο δεδομένων. Η πρώτη περίπτωση χρησιμοποιεί τη **Focal Loss** (υλοποιημένη στο αρχείο `tsp_hybrid_solver/gnn/gat.py`), η οποία δοκιμάστηκε κυρίως για την αντιμετώπιση της ανισορροπίας μεταξύ θετικών και αρνητικών ακμών. Στο υποκείμενο πρόβλημα TSP, η αναλογία θετικών προς αρνητικές ακμές μειώνεται δραστικά καθώς αυξάνεται ο αριθμός των σημείων. Η Focal Loss δίνει προοδευτικά μεγαλύτερη βαρύτητα στα σπάνια θετικά δείγματα, προσπαθώντας να βελτιώσει την εκμάθηση των κρίσιμων ακμών. Παρά τη θεωρητική της δυνατότητα, πειραματικά απέδωσε χειρότερα αποτελέσματα.

Η δεύτερη περίπτωση χρησιμοποιεί την **BCEWithLogitsLoss** της βιβλιοθήκης PyTorch με σταθερό βάρος στις ακμές της βέλτιστης διαδρομής και, στην πράξη, απέδωσε καλύτερα αποτελέσματα όσον αφορά την ανάκληση (Recall) και την PR-AUC, που έχουν καθοριστική σημασία για το πρόβλημα του TSP.

Δείκτης Απόδοσης	Σύνολο Εκπαίδευσης	Σύνολο Επαλήθευσης
Απώλεια (Τιμή Συνάρτηση Κόστους)	0.0078	0.0083
Ακρίβεια (Precision)	69.94%	76.40%
Ανάκληση (Recall)	36.49%	<b>24.44%</b>
ROC-AUC	0.9384	0.9299
PR-AUC	0.5789	0.5487

Πίνακας 4.5: Μέτρα απόδοσης με συνάρτηση κόστους FocalLoss

Δείκτης Απόδοσης	Σύνολο Εκπαίδευσης	Σύνολο Επαλήθευσης
Απώλεια (Τιμή Συνάρτηση Κόστους)	0.3376	0.3277
Ακρίβεια (Precision)	36.06%	45.21%
Ανάκληση (Recall)	69.21%	<b>62.26%</b>
ROC-AUC	0.9249	0.9330
PR-AUC	0.4911	0.5507

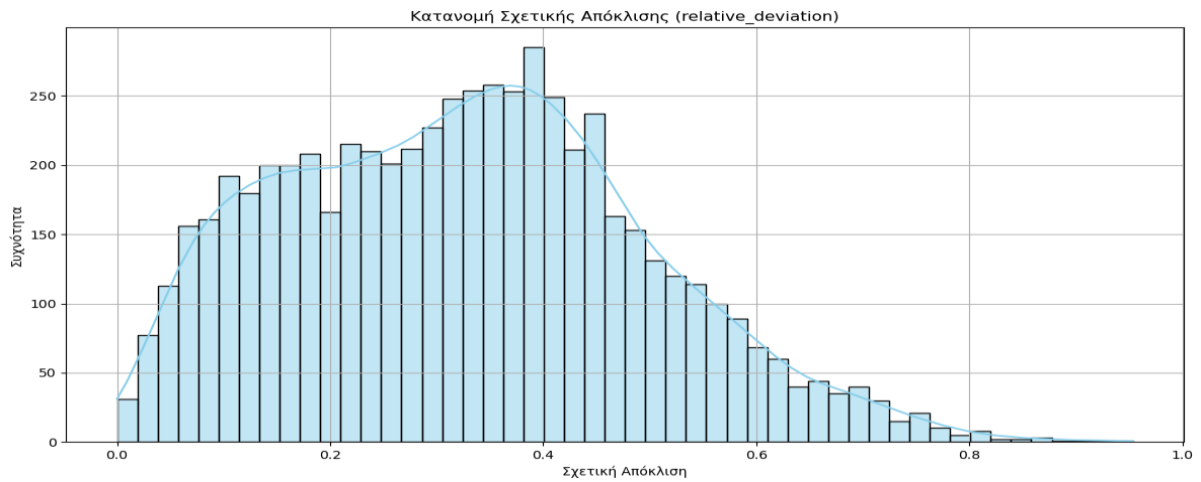
Πίνακας 4.6: Μέτρα απόδοσης με συνάρτηση κόστους BCEWithLogitsLoss

Υπό το πρίσμα των παραπάνω, όπου η **ανάκληση** και η **PR-AUC** έχουν καθοριστική σημασία για το πρόβλημα του TSP, επιλέχθηκε τελικά η συνάρτηση κόστους BCEWithLogitsLoss. Αξίζει να σημειωθεί ότι αλλαγές σε υπερπαραμέτρους — όπως ο ρυθμός μάθησης ή το μέγεθος παρτίδας — δεν παρουσίασαν σημαντική διαφοροποίηση στα αποτελέσματα. Τέλος, λαμβάνοντας υπόψη την περιορισμένη δυνατότητα πειραματισμού τόσο με την αρχιτεκτονική του μοντέλου όσο και με τις παραμέτρους εκπαίδευσης, τα αποτελέσματα κρίνονται ενθαρρυντικά και δικαιολογούν τη συνέχιση της παρούσας προσέγγισης για την επίλυση του TSP.

#### 4.4 Αξιολόγηση απόδοσης στο σύνολο ελέγχου

Μετά την ολοκλήρωση της διαδικασίας εκπαίδευσης, το μοντέλο αξιολογήθηκε σε ένα ανεξάρτητο σύνολο ελέγχου (*test set*), το οποίο περιλαμβάνει προβλήματα του TSP με διαφορετικό αριθμό κορυφών. Η απόδοση μετρήθηκε ως προς τη σχετική απόκλιση (*relative deviation*), η οποία ορίζεται ως το ποσοστό διαφοράς μεταξύ της προβλεπόμενης διαδρομής (προερχόμενης από τον συνδυασμό του νευρωνικού μοντέλου και του αλγορίθμου *beam search*) και της πραγματικά βέλτιστης λύσης που υπολογίστηκε μέσω ακέραιου γραμμικού προγραμματισμού. Χαμηλότερες τιμές σχετικής απόκλισης υποδηλώνουν πιο ακριβείς προβλέψεις.

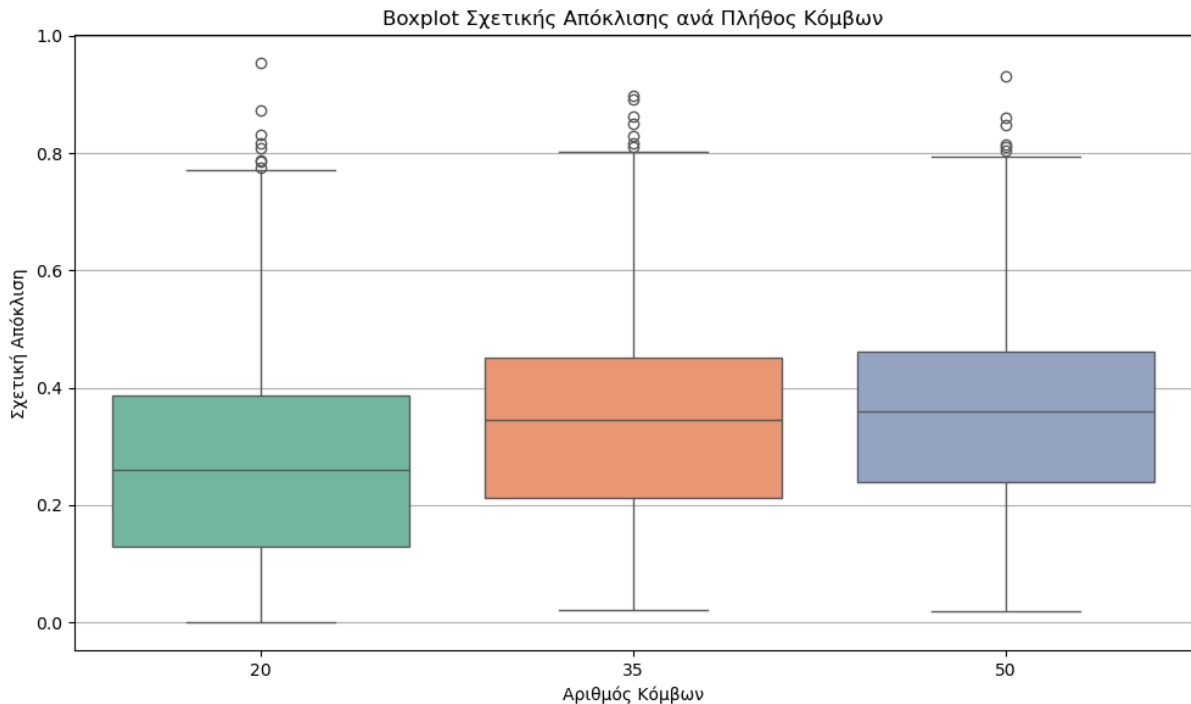
Η μέση σχετική απόκλιση στο σύνολο ελέγχου ανήλθε στο 32.49%, κάτι που καταδεικνύει ότι το μοντέλο καταφέρνει να προσεγγίσει τη βέλτιστη λύση, αλλά εξακολουθεί να υπάρχει σημαντικό περιθώριο βελτίωσης. Η **ελάχιστη** τιμή της απόκλισης ήταν 0.00%, γεγονός που δείχνει πως σε κάποιες περιπτώσεις το μοντέλο βρήκε την απολύτως βέλτιστη λύση. Ωστόσο, η **μέγιστη** απόκλιση έφτασε στο 95.30%, κάτι που φανερώνει αδυναμία γενίκευσης σε ορισμένα προβλήματα. Τα παραπάνω αποτελέσματα απεικονίζονται διαγραμματικά στο σχήμα 4.5.



**Σχήμα 4.7: Κατανομή Σχετικής Απόκλισης Προβλεπόμενης-Βέλτιστης Διαδρομής**

Αναλύοντας τα αποτελέσματα ανά αριθμό κόμβων:

- Για γραφήματα με 20 κόμβους, η μέση απόκλιση ήταν 27.20%
- Για γραφήματα με 35 κόμβους, η μέση απόκλιση αυξάνεται στο 34.15%
- Για γραφήματα με 50 κόμβους, φτάνει το 36.10%



**Σχήμα 4.8: Κατανομές Σχετικής Απόκλισης ανά αριθμό κόμβων**

Νευρωνικά Δίκτυα Γράφων για την επίλυση του Προβλήματος του Περιπλανώμενου Πωλητή: Συνδυασμός Μαθηματικού Προγραμματισμού και Μηχανικής Μάθησης

Η αύξηση της σχετικής απόκλισης με τον αριθμό κόμβων είναι αναμενόμενη, καθώς το πρόβλημα του TSP γίνεται εκθετικά πιο πολύπλοκο. Παράλληλα, από την **κατανομή των τιμών** που απεικονίζεται στο παρακάτω διάγραμμα, παρατηρείται μεγάλη διακύμανση στις τιμές απόκλισης, υποδεικνύοντας ότι το μοντέλο αποδίδει καλά σε ορισμένα παραδείγματα, αλλά αποτυγχάνει σε άλλα.

Η γενική εικόνα δείχνει αρκετά υποσχόμενη προσέγγιση, αλλά χρειάζεται περαιτέρω βελτιώσεις. Πιθανές κατευθύνσεις θα μπορούσαν να περιλαμβάνουν:

- Επέκταση του εύρους των δεδομένων εκπαίδευσης
- Δοκιμή διαφορετικών αρχιτεκτονικών GNN
- Αντικατάσταση ή βελτίωση του αλγορίθμου beam search

## 4.5 Συμπεράσματα

Η παρούσα εργασία παρουσίασε μια υβριδική προσέγγιση επίλυσης του προβλήματος του Πλανόδιου Πωλητή, συνδυάζοντας ένα νευρωνικό δίκτυο γραφήματος με μια ευρετική μέθοδο αναζήτησης. Το δίκτυο εκπαιδεύτηκε ώστε να εκτιμά ποιες συνδέσεις μεταξύ πόλεων είναι πιθανότερο να ανήκουν στη βέλτιστη διαδρομή, ενώ η διαδικασία αναζήτησης χρησιμοποιήθηκε για να συνθέσει μια έγκυρη και όσο το δυνατόν πιο καλή λύση βασισμένη σε αυτές τις εκτιμήσεις.

Επιπλέον, η εργασία αυτή συνοδεύεται και από την υλοποίηση ενός επιλύτη βασισμένου στον Ακέραιο Γραμμικό Προγραμματισμό (ILP), μέσω του οποίου υπολογίστηκαν οι πραγματικά βέλτιστες λύσεις. Παρότι οι ILP-solver προσφέρουν ακρίβεια, καθίστανται αναποτελεσματικοί ή και αδύνατο να εφαρμοστούν σε προβλήματα μεγάλου μεγέθους λόγω του εκθετικού χρόνου επίλυσης. Ο απώτερος στόχος αυτής της μελέτης είναι να αναπτυχθεί μια μεθοδολογία εκτίμησης που να μπορεί να λειτουργεί σε τέτοιες περιπτώσεις, προσφέροντας λύσεις σε προβλήματα όπου οι παραδοσιακοί επιλυτές αποτυγχάνουν λόγω υπολογιστικής πολυπλοκότητας.

Τα πειραματικά αποτελέσματα ανέδειξαν ένα υποσχόμενο μοντέλο εκτίμησης της βέλτιστης λύσης, με μέση σχετική απόκλιση στο 33.66% ενώ σε κάποιες σπάνιες περιπτώσεις την εντόπισε ακριβώς. Πρέπει επίσης να σημειωθεί ότι η ακριβής πρόβλεψη της βέλτιστης διαδρομής είναι εξαιρετικά δύσκολη και σε καμία περίπτωση δεν μπορεί να θεωρηθεί τυχαίο καθώς ο χώρος των εφικτών λύσεων μεγεθύνετε παραγοντικά σε σχέση με τον αριθμό των κορυφών. Ωστόσο, υπήρξε σημαντική διακύμανση στις προβλέψεις, ενώ η απόδοση μειωνόταν σταθερά με την αύξηση του αριθμού κόμβων. Αυτό ήταν αναμενόμενο, αλλά ταυτόχρονα υποδεικνύει ότι απαιτείται περαιτέρω ενίσχυση της ικανότητας γενίκευσης του μοντέλου.

Τέλος, λόγω περιορισμένων υπολογιστικών πόρων, τα σύνολα εκπαίδευσης και ελέγχου περιορίστηκαν σε γράφους από 20 έως 50 κορυφές. Παρά τον περιορισμό αυτό, τα αποτελέσματα δείχνουν υποσχόμενα και ίσως με καλύτερη αρχιτεκτονική και επαρκείς πόρους, ένα νευρωνικό δίκτυο που εκπαιδεύεται σε μικρά γραφήματα ίσως δυνητικά να μπορεί να γενικεύσει και σε μεγαλύτερα, ανοίγοντας το δρόμο για πρακτικές εφαρμογές σε μεγάλης κλίμακας προβλήματα δρομολόγησης. Να σημειωθεί επίσης ότι για να διατηρηθεί ο χρόνος υπολογισμού σε αποδεκτά επίπεδα, περιορίστηκε

- το πλήθος των πιθανών διαδρομών που εξετάζονται κατά τη διάρκεια της αναζήτησης beam search

- ο πειραματισμός με διαφορετικές παραλλαγές του νευρωνικού δικτύου και με τις παραμέτρους εκπαίδευσης.

#### Μελλοντικές Κατευθύνσεις

Η εργασία αυτή θέτει τις βάσεις για περαιτέρω ερευνητική διερεύνηση και εξέλιξη. Πιθανές κατευθύνσεις για μελλοντική εργασία περιλαμβάνουν:

- Δοκιμή διαφορετικών παραλλαγών νευρωνικών δικτύων: Η χρήση εναλλακτικών μοντέλων μπορεί να οδηγήσει σε καλύτερη κατανόηση της δομής του προβλήματος και σε ακριβέστερες εκτιμήσεις.
- Καλύτερη παραμετροποίηση και ρύθμιση της εκπαίδευσης: Πιο στοχευμένη επιλογή υπερπαραμέτρων (learning rate, batch size κ.ά.) μπορεί να ενισχύσει σημαντικά την απόδοση του μοντέλου.
- Πλουσιότερο και μεγαλύτερο σύνολο δεδομένων: Η εκπαίδευση σε μεγαλύτερο όγκο δεδομένων και η ενσωμάτωση πιο πολύπλοκων παραδειγμάτων (με περισσότερες πόλεις ή διαφορετικές κατανομές) αναμένεται να βελτιώσει την ικανότητα γενίκευσης.
- Εναλλακτική μετα-επεξεργασία των προβλέψεων: Η έξοδος του νευρωνικού δικτύου μπορεί να αξιοποιηθεί με άλλους τρόπους για την κατασκευή διαδρομών, πέραν του beam search, ίσως με τη χρήση διαφορετικών ευρετικών αλγορίθμων.
- Βελτίωση ή αντικατάσταση του beam search: Ο επανασχεδιασμός της διαδικασίας αναζήτησης ή η προσαρμογή του σε δυναμικά χαρακτηριστικά του προβλήματος μπορεί να οδηγήσει σε πιο ποιοτικές και σταθερές λύσεις.

Το υλικό της παρούσας μεταπτυχιακής διατριβής έχει δημοσιευθεί δημόσια στο GitHub, με στόχο τη διευκόλυνση της αναπαραγωγής των αποτελεσμάτων, και είναι διαθέσιμο στον ακόλουθο σύνδεσμο:

<https://github.com/I-Mougiotis/tsp-ilp-gnn-thesis>

#### Βιβλιογραφικές Αναφορές

Vercesi, N. & Buchanan, I. (2024). *The Dantzig–Fulkerson–Johnson TSP formulation is easy to solve for few subtour constraints*. Optimization Online. <https://optimization-online.org/2024/09/the-dantzig-fulkerson-johnson-tsp-formulation-is-easy-to-solve-for-few-subtour-constraints>

Vercesi, E. (2024). The Dantzig-Fulkerson-Johnson TSP formulation is easy to solve for few subtour constraints. Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). *Integer programming formulation of traveling salesman problems*. Journal of the ACM, 7(4), 326–329. <https://doi.org/10.1145/321043.321046>

Dantzig, G. B., Fulkerson, D. R., & Johnson, S. M. (1954). *Solution of a large-scale traveling-salesman problem*. Journal of the Operations Research Society of America, 2(4), 393–410. <https://doi.org/10.1287/opre.2.4.393>

A. Chandra & A. Naro (2022). *A Comparative Study of Metaheuristics Methods for Solving Traveling Salesman*

- Gao, W. (2020). *New Ant Colony Optimization Algorithm for the Traveling Salesman Problem*. *International Journal of Computational Intelligence Systems*, 13(1), 44–55.
- Liu, S. (2014). *A Powerful Genetic Algorithm for Traveling Salesman Problem*. arXiv:1402.4699.
- Goldbarg, E. F., et al. (2008). *Particle Swarm Optimization Algorithm for the Traveling Salesman Problem*.
- Huang, L., Chen, X., Huo, W., Wang, J., Zhang, F., Bai, B., & Shi, L. (2021). *Branch and Bound in Mixed Integer Linear Programming Problems: A Survey of Techniques and Trends*. arXiv:2111.06257.
- Department of Computer Science, University of Copenhagen. Retrieved from <https://web.archive.org/web/20150923214803/http://www.diku.dk/OLD/undervisning/2003e/datV-optimer/JensClausenNoter.pdf>
- G. Kobeaga, M. Merino, J. A. Lozano (2020). *On Solving Cycle Problems with Branch-and-Cut: Extending Shrinking and Exact Subcycle Elimination Separation Algorithms*. arXiv:2004.14574
- Balcan, M.-F., Dick, T., Sandholm, T., & Vitercik, E. (2022). *Cutting Planes via Machine Learning*. arXiv:2204.07312.
- Paul, M., Barzilay, R., & Jaakkola, T. (2020). *Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search*. arXiv:2004.14574.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2019). *A Comprehensive Survey on Graph Neural Networks*. arXiv:1812.08434.
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2018). *Graph Neural Networks: A Review of Methods and Applications*. arXiv:1812.08434.
- Velicković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph Attention Networks* (ICLR). arXiv preprint arXiv:1710.10903.