### **Πίνακας Περιεχομένων**

### **1. Εισαγωγή**

1.1 Το Πρόβλημα του Περιπλανώμενου Πωλητή (TSP)  
1.2 Μια Προσέγγιση με Μηχανική Μάθηση  
1.3 Συνεισφορές

### **2****. Επίλυση του TSP με Γραμμικό Προγραμματισμό**

Διατύπωση του Προβλήματος  
 2.1 Προσεγγίσεις Μαθηματικής Μοντελοποίησης  
  2.1.1 Μαθηματική Διατύπωση του Μοντέλου MTZ  
  2.1.2 Μαθηματική Διατύπωση του Μοντέλου DFJ

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 Τεμνόντα Επίπεδα (Cutting Planes)

### **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 (n), ακόμη και στην συμμετρική εκδοχή του προβλήματος, όπου απαιτείται η εξέταση (n−1)!/2​ πιθανών διαδρομών. Με τον χαρακτηρισμό **συμμετρική εκδοχή ενός TSP** δεχόμαστε την υπόθεση ότι το κόστος μετάβασης από το σημείο Α στο σημείο Β ισούται με το κόστος μετάβασης από το σημείο Β στο σημείο Α. Μολονότι η υπόθεση αυτή συνεπάγεται μείωση των μεταβλητών απόφασης στο μισό, ο χώρος τον 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 cutting-plane και δυναμική αποκοπή διαδρομών που υλοποιούμε στην παρούσα εργασία, προσφέρουν πιο αποδοτικά πλαίσια επίλυσης. Ωστόσο, ακόμα και αυτές οι προηγμένες μαθηματικές τεχνικές δεν ξεφεύγουν από τη βασική NP-hard φύση του προβλήματος. Ο χειρότερος χρόνος εκτέλεσης παραμένει εκθετικός, περιορίζοντας τη χρήση τους σε μεσαίου μεγέθους προβλήματα παρά τις δεκαετίες βελτιώσεων.

#### **1.1 Μια Προσέγγιση με Μηχανική Μάθηση**

Η παρούσα εργασία εξετάζει ένα εναλλακτικό παράδειγμα: Μπορούν τα νευρωνικά δίκτυα γράφων (Graph Neural Networks - GNNs) να μάθουν τα μοτίβα των βέλτιστων λύσεων του TSP από μικρά παραδείγματα και να γενικεύσουν αυτή τη γνώση σε μεγαλύτερα προβλήματα; Η προσέγγισή 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 όπου:

* Ένα GNN μαθαίνει να προβλέπει τις πιθανότητες επιλογής ακμών από βέλτιστες λύσεις μικρών παραδειγμάτων (20-50 κόμβοι).
* Οι προβλέψεις αυτές καθοδηγούν έναν αλγόριθμο beam search για την κατασκευή εφικτών διαδρομών.

#### **1.2 Συνεισφορές**

Η εργασία αυτή προσφέρει τρεις βασικές συνεισφορές:

* **Υλοποίηση:** Ανάπτυξη τόσο παραδοσιακών λύσεων γραμμικού προγραμματισμού όσο και νέων GNN-based λύσεων με ανοικτή πρόσβαση στον κώδικα.
* **Υβριδική Μεθοδολογία:** Επίδειξη του πώς η μηχανική μάθηση μπορεί να συμπληρώσει τον μαθηματικό προγραμματισμό στην επίλυση συνδυαστικών προβλημάτων.
* **Εμπειρική Επιβεβαίωση:** Συστηματική αξιολόγηση που αποδεικνύει ότι τα GNN μπορούν να συλλάβουν ουσιαστικά μοτίβα λύσεων, θέτοντας θεμέλια για μελλοντική 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)
* και ελαχιστοποιεί το συνολικό μήκος ή κόστος της διαδρομής (3)

**2.1 Προσεγγίσεις Μαθηματικής Μοντελοποίησης**

Στη βιβλιογραφία έχουν αναπτυχθεί διάφορες μαθηματικές διατυπώσεις για τη μοντελοποίησης του Προβλήματος του Πλανόδιου Πωλητή (TSP), με δύο από τις πιο διαδεδομένες να είναι εκείνες των **Miller–Tucker–Zemlin (MTZ)** και **Dantzig–Fulkerson–Johnson (DFJ)**. Αμφότερες οι προσεγγίσεις ξεκινούν εισάγοντας **δυαδικές μεταβλητές** xij 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** ui και uj, 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:

ui - uj+1 ≤ (n -1) (1 – xij) ∀ i ε [2, n-1] και ι ≠j

οι οποίες διασφαλίζουν την αδυναμία σχηματισμού κυκλικών ακολουθιών όπως ui<uj<⋯<uι, οι οποίες θα οδηγούσαν σε επιμέρους κύκλους. Οι τιμές των ui 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 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 MTZ**

Μία από τις πρώτες και πιο κλασικές μαθηματικές διατυπώσεις του προβλήματος του Πλανόδιου Πωλητή (TSP) 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 ∈ V αντιπροσωπεύει μία πόλη και κάθε ακμή (i,j) ∈ E έχει ένα αντίστοιχο κόστος μετακίνησης cij. Ο στόχος είναι να επισκεφθούμε κάθε πόλη ακριβώς μία φορά και να επιστρέψουμε στο σημείο εκκίνησης, ελαχιστοποιώντας το συνολικό κόστος.

Η διατύπωση MTZ χρησιμοποιεί τις εξής μεταβλητές:

* xij ∈ {0,1): είναι 1 αν ο πωλητής μετακινηθεί από την πόλη i στην πόλη j, αλλιώς 0.
* ui ∈ ℜ: βοηθητικές μεταβλητές που εισάγονται για την εξάλειψη των **επιμέρους κύκλων** (μόνο για κόμβους i=1,…,n).

Η διατύπωση του προβλήματος γίνεται ως εξής:

**Στόχος:**

(1)

**υπό τους περιορισμούς:**

(2)

(3)

(4)

Οι περιορισμοί **(2)** και **(3)** αποτελούνται από σύνολο n περιορισμών ο καθένας και επιβάλλουν τα εξής: κάθε πόλη να επισκέπτεται ακριβώς μία φορά ως πόλη αναχώρησης και μία φορά ως πόλη άφιξης, αντίστοιχα. Δηλαδή, από κάθε πόλη φεύγει ακριβώς μία διαδρομή και προς κάθε πόλη καταλήγει ακριβώς μία διαδρομή.

Ο τελευταίος περιορισμός **(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 θεωρείται η αρχική πόλη εκκίνησης και δεν χρειάζεται να περιοριστεί εδώ.
* Οι βοηθητικές μεταβλητές ui​ 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→2→3→4→1, δηλαδή επιστροφή από την τελευταία πόλη στην πρώτη, χωρίς να επιτρέψει κύκλους όπως 1→2→3→2 που δεν περνούν από όλες τις πόλεις.

**2.1.2 Μαθηματική Διατύπωση του Μοντέλου DFJ**

Η προσέγγιση του Dantzig–Fulkerson–Johnson (DFJ) αποτελεί μία από τις κλασικές και ακριβείς μαθηματικές διατυπώσεις του προβλήματος του Πλανόδιου Πωλητή (TSP). Όπως και στην περίπτωση της MTZ, το πλήρως συνδεδεμένο γράφημα G=(V,E) έχει κόμβους i ∈ V που αντιστοιχούν σε πόλεις, και ακμές (i, j) ∈ E με αντίστοιχο κόστος μετακίνησης cij​.

Οι δυαδικές μεταβλητές ορίζονται ως εξής:

* xij ∈ {0,1): είναι 1 αν ο πωλητής μετακινηθεί από την πόλη i στην πόλη j, αλλιώς 0.

Το πρόβλημα διατυπώνεται ως εξής:

(1)

**Υπό τους περιορισμούς:**

Κάθε πόλη πρέπει να επισκεφθεί ακριβώς μία φορά ως πόλη άφιξης

(2)

Κάθε πόλη πρέπει να επισκεφθεί ακριβώς μία φορά ως πόλη αναχώρησης

(3)

Αποτροπή επιμέρους κύκλων

(4)

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.

Για την παρούσα εργασία, ο επιλυτής του προβλήματος TSP που υλοποιήθηκε σε γλώσσα Python πραγματοποιήθηκε με χρήση τεχνικών άμεσης ανίχνευσης επιμέρους κύκλων και προσθήκης των αντίστοιχων περιορισμών. Το συγκεκριμένο μοντέλο χρησιμοποιήθηκε για την παραγωγή βέλτιστων λύσεων, οι οποίες αποτέλεσαν τη βάση εκπαίδευσης του Νευρωνικού Δικτύου Γράφων(GNN). Η λεπτομέρειες αναφορικά με την υλοποίηση του επιλυτή καθώς η αρχιτεκτονική και τα χαρακτηριστικά του νευρωνικού μοντέλου θα αναλυθούν σε επόμενο τμήμα της εργασίας.

**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 ιδιαιτέρως χρήσιμη σε προβλήματα όπου απαιτούνται **δυαδικές αποφάσεις**, όπως επιλογή/απόρριψη, ενεργοποίηση/απενεργοποίηση, ή ύπαρξη/απουσία σχέσης μεταξύ οντοτήτων. Το Πρόβλημα του Πλανόδιου Πωλητή (TSP), το οποίο αποτελεί το αντικείμενο της παρούσας εργασίας, ανήκει ακριβώς σε αυτήν την κατηγορία.

Η γενική μαθηματική διατύπωση ενός ILP προβλήματος είναι η εξής:

Ελαχιστοποίηση:

Υπό τους περιορισμούς:

Σε περίπτωση δυαδικού ILP, ισχύει:

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 συνεχούς 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 **θεωρητικό 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.

Μαθηματικά, έστω το γενικό πρόβλημα Ακέραιου Γραμμικού Προγραμματισμού (ILP):

(1)

Ορίζουμε το πολύεδρο εφικτών λύσεων του LP Relaxation:

Το σύνολο των ακέραιων εφικτών λύσεων είναι:

*X =* P ∩

H κυρτή θήκη των ακέραιων λύσεων είναι:

Px = conv(X)

Τότε ισχύει:

X ⊆ Px​ ⊆ P

και για τις βέλτιστες τιμές των προβλημάτων:

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 φαινομενικα 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 Εξάντλησης και υπολογιστικής πολυπλοκότητας**

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

* n ακέραιες μεταβλητές
* και κάθε μεταβλητή xι​ μπορεί να πάρει κι​ διακριτές τιμές

Το πλήθος των λύσεων είναι:

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 Διακλάδωση και Φραγμός**

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.

Ύπό το πλαίσιο της μεθόδου, ο όρος **υποπρόβλημα** αναφέρεται σε ένα νέο πρόβλημα που προκύπτει από το **αρχικά χαλαρωμένο γραμμικό πρόβλημα** μέσω της προοδευτικής 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 υποχώρο εφικτών λύσεων του αρχικού προβλήματος και επιλύεται ως ένα νέο χαλαρωμένο γραμμικό πρόβλημα.

Η λύση κάθε χαλαρωμένου προβλήματος παρέχει **φράγματα (bounds)** που καθοδηγούν τη διαδικασία αναζήτησης:

* Η **καλύτερη ακέραιη λύση** που έχει εντοπιστεί μέχρι εκείνο το σημείο παρέχει ένα **άνω φράγμα(πρ**οβλημα ελαχιστοποίησης)

Το αντίστροφο ισχύει σε προβλήματα μεγιστοποίησης.

Αν η λύση του χαλαρωμένου προβλήματος είναι ήδη 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:

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

* Το υποπρόβλημα είναι **ανεφικτό**.
* 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 δύναμη της μεθόδου έγκειται στο γεγονός ότι, μέσω της διαδικασίας του φραγμού, **αποκλείονται μεγάλες περιοχές του εφικτού χώρου** χωρίς να απαιτείται η εξαντλητική αναζήτηση όλων των δυνατών λύσεων.

Μέσω της διαδικασίας του φραγμού διασφαλίζεται η εύρεση της **βέλτιστης ακέραιας λύσης** χωρίς να απαιτείται η εξαντλητική εξέταση όλων των δυνατών συνδυασμών. Αυτό επιτυγχάνεται, περιορίζοντας το εύρος αναζήτησης μόνο στις υποπεριπτώσεις που έχουν προοπτική βελτίωσης της βέλτιστης λύσης. Τα παραπάνω μπορούν να γίνουν κατανοητά με την χρήση του παρακάτω παραδείγματος.

Έστω το πρόβλημα ελαχιστοποίησης της αντικειμενικής συνάρτησης:

f(x1, x2) = 7x1 + 8x2

υπό τους περιορισμούς:

(1)

(2)

(3)

(4)

(5)

Αρχικά, επιλύεται το χαλαρωμένο πρόβλημα παραλείποντας τον περιορισμό ακεραιότητας (5).

Η βέλτιστη λύση είναι:

Με τιμή αντικειμενικής συνάρτησης:

Εφόσον x1 δεν είναι ακέραια, ξεκινά η διαδικασία διακλάδωσης με βάση τον περιορισμό:

Ακολουθεί αναδρομικά νέα διακλάδωση με βάση την τιμή του x2​ κ.ο.κ.

Στο αντίστοιχο **δέντρο διακλάδωσης**, οι καταστάσεις απεικονίζονται χρωματικά:

* **Πράσινο**: συνεχίζεται η διαδικασία.
* **Κόκκινο**: τερματισμός λόγω ανεφικτότητας ή χειρότερης λύσης.
* **Πορτοκαλί**: εφικτή ακέραια λύση (όχι απαραίτητα τελική).
* **Κίτρινο**: τελική βέλτιστη λύση.

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

Βέλτιστη λύση: (2.33, 0)

Τιμή αντικειμενική συνάρτησης: 16.33

Υποπρόβλημα 2

Βέλτιστη λύση: (3, 0)

Τιμή αντικειμενική συνάρτησης: 21

Υποπρόβλημα 1

Βέλτιστη λύση: (2, 0.8)

Τιμή αντικειμενική συνάρτησης: 20.4

Υποπρόβλημα 1.2

Βέλτιστη λύση: (1.967, 1)

Τιμή αντικειμενική συνάρτησης: 21.41

Υποπρόβλημα 1.1

Ανέφικτο

Υποπρόβλημα 1.2.1

Βέλτιστη λύση: (1, 3.2)

Τιμή αντικειμενική συνάρτησης: 32.6

Υποπρόβλημα 1.2.2

Ανέφικτο

Υποπρόβλημα 1.2.1.1

Ανέφικτο

Υποπρόβλημα 1.2.1.2

Βέλτιστη λύση: (0,6667, 4)

Τιμή αντικειμενική συνάρτησης: 36.67

Υποπρόβλημα 1.2.1.2.2

Βέλτιστη λύση: (1, 4)

Τιμή αντικειμενική συνάρτησης: 39

Υποπρόβλημα 1.2.1.2.1

Ανέφικτο

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

Η μέθοδος του φραγμού (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 τεμνόντα 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 στο επόμενο τμήμα.

Σχημα 2.2: Απεικόνιση της διαδικασίας διακλάδωσης όταν προηγείται το υποπρόβλημα

Βέλτιστη λύση: (2.33, 0)

Τιμή αντικειμενική συνάρτησης: 16.33

Υποπρόβλημα 2

Βέλτιστη λύση: (3, 0)

Τιμή αντικειμενική συνάρτησης: 21

Υποπρόβλημα 1

Βέλτιστη λύση: (2, 0.8)

Τιμή αντικειμενική συνάρτησης: 20.4

Υποπρόβλημα 1.2

Βέλτιστη λύση: (1.967, 1)

Τιμή αντικειμενική συνάρτησης: 21.41

**Απορρίπτεται χρησιμοποιώντας ως άνω φράγμα την βέλτιστη λύση από το υποπρόβλημα 2**

Υποπρόβλημα 1.1

Ανέφικτο

**2.3.3 Τεμνόντων Επιπέδων**

Η μέθοδος των τεμνόντων επιπέδων (cutting planes) αποτελεί θεμελιώδη τεχνική στον Ακέραιο Γραμμικό Προγραμματισμό (ILP), με κύριο στόχο την αποκατάσταση της ακεραιότητας των μεταβλητών σε προβλήματα που αρχικά επιλύονται στη χαλαρωμένη (γραμμική) τους μορφή. Η βασική ιδέα της μεθόδου είναι η επαναληπτική προσθήκη νέων γραμμικών περιορισμών — γνωστών ως *τομές* — οι οποίοι περικόπτουν τον χώρο των εφικτών λύσεων του χαλαρωμένου προβλήματος, αποκλείοντας τις μη ακέραιες λύσεις, χωρίς όμως να εξαιρούν καμία από τις πιθανές ακέραιες λύσεις του αρχικού προβλήματος.

Η αποτελεσματικότητα αυτής της τεχνικής βασίζεται στην παρατήρηση ότι η βέλτιστη ακέραια λύση ενός προβλήματος ILP συχνά βρίσκεται κοντά στη λύση της χαλαρωμένης μορφής. Όταν η τελευταία δεν είναι ακέραια, κατασκευάζονται κατάλληλες τομές που περικόπτουν το συγκεκριμένο τμήμα του πολυέδρου που περιέχει τη μη αποδεκτή λύση. Οι τομές αυτές σχεδιάζονται προσεκτικά ώστε να περνούν από το εσωτερικό του εφικτού πολυέδρου, αφαιρώντας μόνο τις ανεπιθύμητες (μη ακέραιες) κορυφές, διατηρώντας όμως ανέπαφες όλες τις ακέραιες λύσεις. Η διαδικασία επαναλαμβάνεται: κάθε νέα τομή συνοδεύεται από εκ νέου επίλυση του προβλήματος με τους επικαιροποιημένους περιορισμούς, μέχρι είτε την επίτευξη ακέραιας λύσης είτε την απόδειξη αδιεξόδου.

Στην πράξη, ωστόσο, η μέθοδος μπορεί να είναι υπολογιστικά απαιτητική. Ο ακριβής αριθμός τομών που απαιτείται δεν είναι εκ των προτέρων γνωστός, ενώ επαναλαμβανόμενοι υπολογισμοί ενδέχεται να οδηγήσουν σε αριθμητικές αστάθειες. Παρόλα αυτά, η μέθοδος παραμένει ένας ισχυρός μηχανισμός επιβολής ακεραιότητας, ιδιαίτερα όταν συνδυάζεται με άλλες τεχνικές, όπως η μέθοδος Διακλάδωσης και Φραγμών (*Branch-and-Bound*).

Στο Πρόβλημα του Πλανόδιου Πωλητή (TSP), η μέθοδος των τεμνόντων επιπέδων αποκτά έναν ειδικό, εναλλακτικό ρόλο. Στη συγκεκριμένη περίπτωση, οι μεταβλητές ​ είναι δυαδικές, και η ακεραιότητα μπορεί να εξασφαλιστεί αποτελεσματικά μέσω της τεχνικής Branch-and-Bound. Το ουσιαστικό πρόβλημα εδώ εντοπίζεται στην εμφάνιση **επιμέρους κύκλων (subtours)** — λύσεων που καλύπτουν μόνο ένα υποσύνολο πόλεων, χωρίς να σχηματίζουν έναν ενιαίο, πλήρη κύκλο. Ο αριθμός των πιθανών υποκύκλων αυξάνεται εκθετικά με τον αριθμό των πόλεων, καθιστώντας αδύνατη την εκ των προτέρων εισαγωγή όλων των απαιτούμενων περιορισμών.

Για την αντιμετώπιση αυτού του προβλήματος εφαρμόζεται η τεχνική **δυναμικής παραγωγής περιορισμών** (*lazy constraints*), μέσω των λεγόμενων **Subtour Elimination Constraints (SECs)**. Αρχικά, το πρόβλημα επιλύεται με έναν βασικό μόνο κορμό περιορισμών. Εάν η λύση περιέχει **επιμέρους κύκλους**, εντοπίζονται και εισάγονται οι αντίστοιχοι SECs, και η διαδικασία επαναλαμβάνεται. Έτσι, η μέθοδος **Branch-and-Cut** εισάγει μόνο τους απολύτως απαραίτητους περιορισμούς, εξοικονομώντας σημαντικά υπολογιστικούς πόρους.

Συγκεκριμένα:

* Δεν απαιτείται η εκ των προτέρων κατασκευή του εκθετικού συνόλου όλων των SECs.
* Οι περιορισμοί προστίθενται *μόνο* όταν εντοπιστούν παραβιάσεις τους στη λύση.

Στην περίπτωση του TSP η μέθοδος των τεμνόντων επιπέδων δεν χρησιμοποιείται για την επιβολή της ακεραιότητας στις μεταβλητές απόφασης αλλά αποτελεί έναν στρατηγικό μηχανισμό ενίσχυσης της δομής της λύσης. Με την αφαίρεση δομικά ανύπαρκτων μονοπατιών και τη δυναμική ενσωμάτωση SECs, βελτιώνεται ριζικά η απόδοση του αλγορίθμου και καθίσταται δυνατή η επίλυση προβλημάτων TSP σχετικά μεγάλης κλίμακας.

**2.4 Συμμετρικό TSP: Μαθηματική Διατύπωση και Υλοποίηση Επιλυτή 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 ​. 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:

Ελαχιστοποίηση:

(1)

Υπό τους περιορισμού:

(2)

*(3)*

*(4)*

Όπου:

* E: Σύνολο ακμών.
* V: Σύνολο κορυφών (πόλεων).
* ​: Δυαδική μεταβλητή που δηλώνει αν η ακμή e περιλαμβάνεται στη λύση.
* ​: Κόστος/απόσταση της ακμής e.
* N(v): Σύνολο ακμών που συνδέονται με την κορυφή v.
* E(S): Υποσύνολο των ακμών που συνδέουν τις κορυφές του υποσυνόλου S⊂V.

O πρώτος περιορισμός της συμμετρικής διατύπωσης του TSP:

**V (2)**

προκύπτει ως αποτέλεσμα της μετάβασης από τη κατευθυνόμενη (ασύμμετρη) μορφή του προβλήματος σε μια ακατεύθυντη (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:

Στη συμμετρική εκδοχή όμως, οι ακμές είναι ακατεύθυντες. 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.

**3.1 Εισαγωγή στα Νευρωνικά Δίκτυα Γραφών (GNNs)**

Ένας μαθηματικός γράφος (graph) 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 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 **ιδιότητες** — 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).

**3.2 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×F**, όπου κάθε γραμμή αντιστοιχεί σε μία παρατήρηση (sample), και κάθε στήλη σε μία μεταβλητή ή 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 **Ν**) 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 γειτνίασης (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 δομή 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.

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).

**3.3 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). 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:

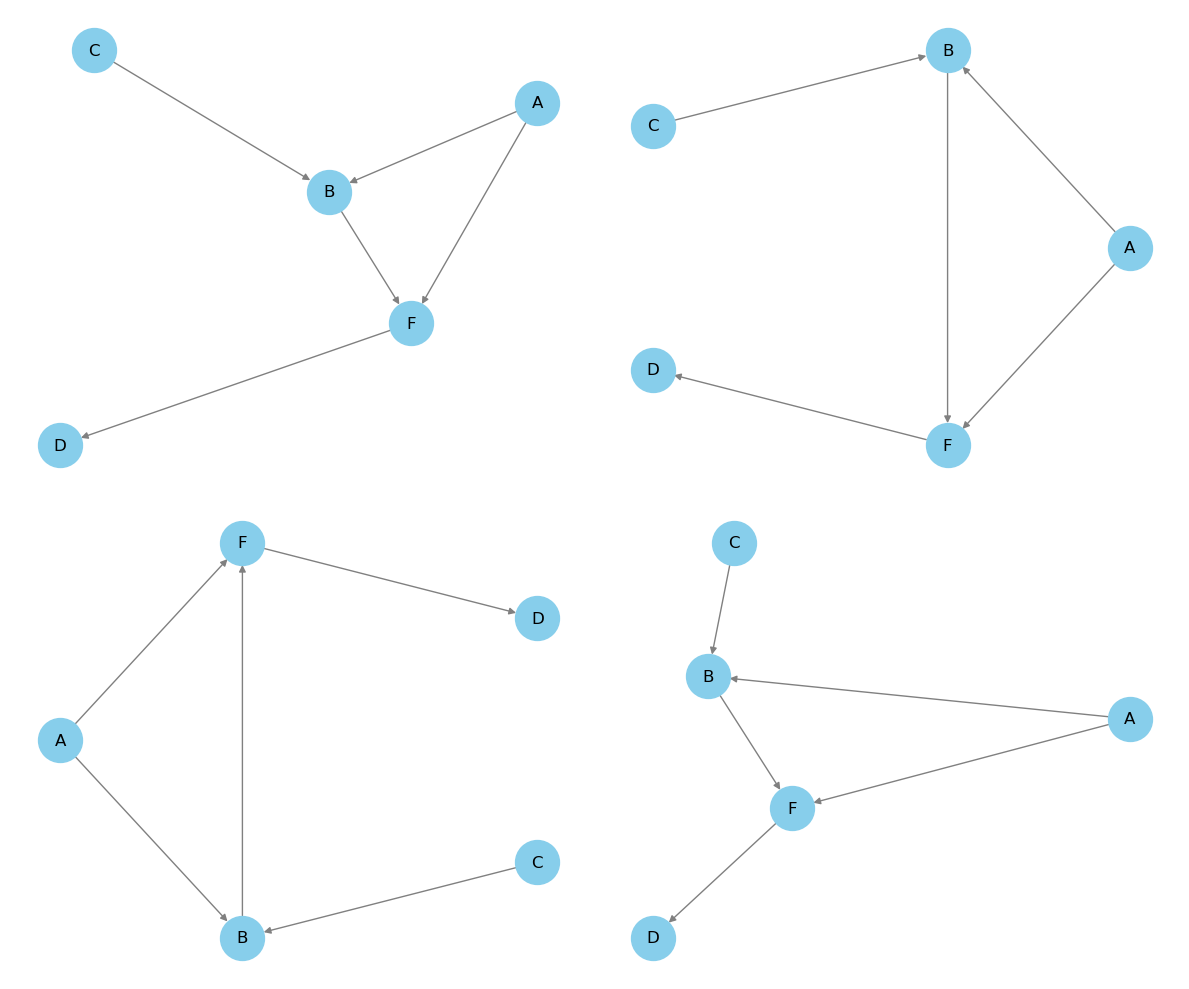
* Δεν υφίσταται εγγενής ή φυσική σειρά μεταξύ των κόμβων.
* Ο ίδιος γράφος μπορεί να έχει πολλαπλές ισομορφικές αναπαραστάσεις, δηλαδή διαφορετικές αλγεβρικές μορφές που περιγράφουν την ίδια τοπολογία.

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

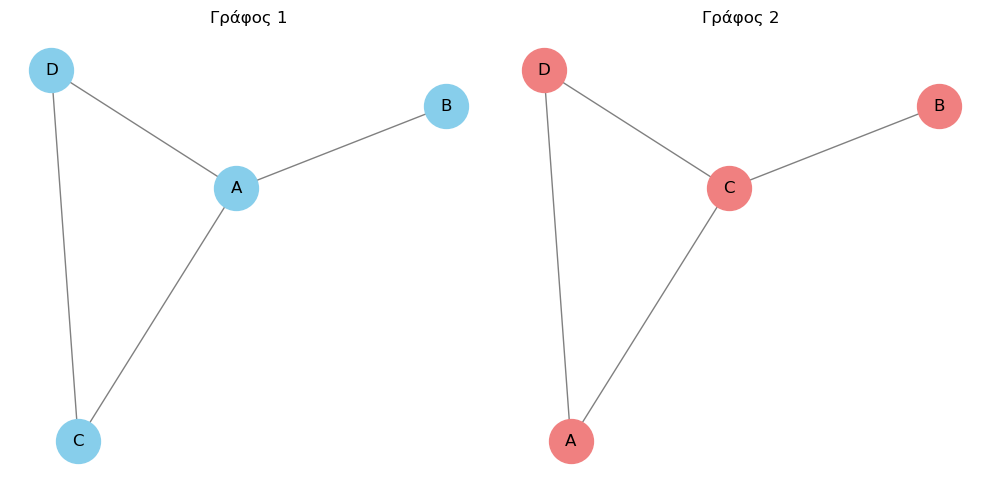


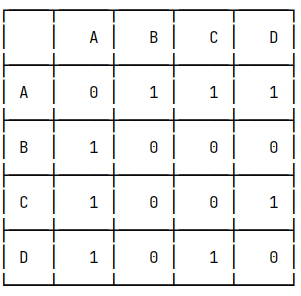
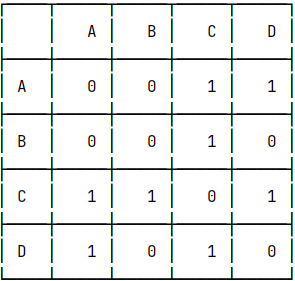
Αντιστρόφως, ο ίδιος γράφος μπορεί να παρασταθεί από **διαφορετικούς πίνακες γειτνίασης** με απλή αναδιάταξη της σειρά των κόμβων. Το φαινόμενο αυτό είναι γνωστό ως πρόβλημα της μετάθεσης (permutation problem) και αποτελεί ειδική περίπτωση του προβλήματος του ισομορφισμού. Για παράδειγμα, ο παρακάτω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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Β | Α | 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 μετάθεση του γράφου **G(V,E)** τα σύνολα  **Ε και V** παραμένουν αμετάβλητα. Αξίζει ωστόσο να σημειωθεί ότι ό ισομορφισμός συνιστά σημαντική πρόκληση στην εφαρμογή τεχνικών μηχανικής μάθησης σε γράφους, καθώς δύο διαφορετικά αλγεβρικά μοντέλα μπορεί να αναπαριστούν την ίδια τοπολογία. Δύο γράφοι θεωρούνται ισομορφικοί όταν υπάρχει μια αμφιμονοσήμαντη αντιστοίχιση τέτοια ώστε κάθε ακμή να αντιστοιχεί στην ακμή. Με άλλα λόγια, η δομή των συνδέσεων διατηρείται, ακόμη κι αν αλλάξουν οι ονομασίες ή η σειρά των κόμβων.

Σχήμα 3.2

****

****

Το Σχήμα 3.2 απεικονίζει ένα παράδειγμα ισομορφισμού μεταξύ δύο γράφων. Τα σύνολα ακμών είναι:

* .

Αν και τα σύνολα ακμών φαίνονται διαφορετικά, η συνδετική 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**.

Μια ακόμη πτυχή των γράφων που δυσκολεύει σημαντικά την εφαρμογή της συνέλιξης είναι ο **ισομορφισμός.** Την περίπτωση αυτή την συναντάμε όταν δύο γράφοι δομικά ταυτόσημο(ίδιο σύνολο κορυφών και ακμών) έχουν διαφορετικές γραφικές και αλγεβρικές αναπαραστάσεις. Η περίπτωση αυτή αποτυπώνεται στην παρακάτω εικόνα:

**Βιβλιογραφικές Αναφορές**

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](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.