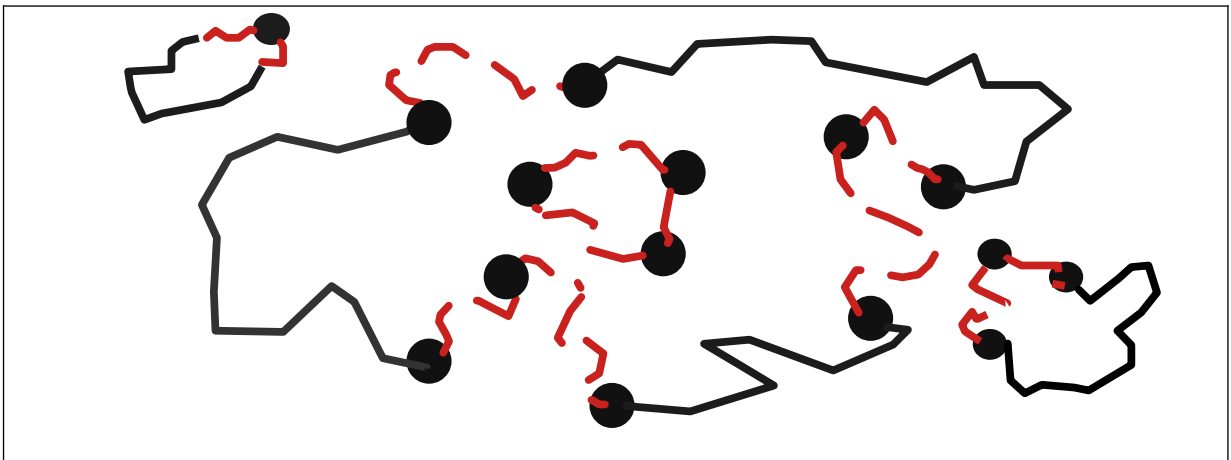




ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
«ΓΡΑΦΙΚΗ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ  
ΤΟΥ ΠΕΡΙΟΔΕΥΟΝΤΟΣ ΠΩΛΗΤΗ ΜΕ ΤΗ  
ΧΡΗΣΗ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΤΥΠΟΥ  
TRANSFORMER»



Του φοιτητή  
Δομενικιώτη Σπυρίδων  
Αρ. Μητρώου: 174963

Επιβλέπων  
Ονοματεπώνυμο  
Γουλιάνας Κωνσταντίνος  
Βαθμίδα Καθηγητής

Ημερομηνία 31/05/2025

Τίτλος Δ.Ε. **Γραφική Επίλυση του Προβλήματος του Περιοδεύοντος Πωλητή με τη χρήση  
Νευρωνικών Δικτύων τύπου Transformer**

Κωδικός Δ.Ε. **23141**

Όνοματεπώνυμο φοιτητή/τών **Δομενικιώτης Σπυρίδων**

Όνοματεπώνυμο εισηγητή **Γουλιάνας Κωνσταντίνος**

Ημερομηνία ανάληψης Δ.Ε. **15/03/2023**

Ημερομηνία περάτωσης Δ.Ε. **31/05/2025**

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Δομενικιώτη Σπυρίδων που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

# Πρόλογος

Με ιδιαίτερη έμφαση στην αρχιτεκτονική Transformer, η παρούσα διατριβή εξετάζει το πρόβλημα του περιοδεύοντος πωλητή (TSP) μέσω σύγχρονων τεχνικών τεχνητής νοημοσύνης, μελετώντας και εφαρμόζοντας τις. Οι ευρείες εφαρμογές στη δρομολόγηση, τη λογιστική, τον χωροταξικό σχεδιασμό και άλλα πεδία καθιστούν το TSP ένα κλασικό πρόβλημα συνδυαστικής βελτιστοποίησης με μεγάλη σημασία.

Στόχος της παρούσας εργασίας είναι η δημιουργία ενός συστήματος ικανό να επιλύει απαιτητικά σενάρια, όπως αυτά που συναντώνται σε νησιωτικές ή ημι-συνδεδεμένες περιοχές της Ελλάδας, συνδυάζοντας μοντέλα βαθιάς μάθησης με ευέλικτη λογική περιορισμών. Στην προσέγγιση περιλαμβάνονται συνθετικά δεδομένα, πραγματικά γεωχωρικά σενάρια που περιλαμβάνουν λιμάνια και νησιά, ενισχυτική μάθηση και υβριδικές στρατηγικές αποκωδικοποίησης (greedy, beam search και 2-OPT).

Η διατριβή έχει κυρίως τις ακόλουθες συνεισφορές:

- Εκπαίδευση και αξιολόγηση TSP ενός μοντέλου βασισμένου σε Transformer
- Έλεγχοι σκοπιμότητας μετά την αποκωδικοποίηση σε συνδυασμό με λογική που λαμβάνει υπόψη τους περιορισμούς
- Μια διαδραστική γραφική διεπαφή χρήστη (GUI) για την οπτικοποίηση της δρομολόγησης
- Μελέτη απόδοσης και αναλογίες με ακριβείς επιλυτές και κλασικούς ευριστικούς επιλυτές

Η διατριβή είναι οργανωμένη ως εξής:

Το θεωρητικό υπόβαθρο και η ιστορική εξέλιξη του προβλήματος παρουσιάζονται στο Κεφάλαιο 1. Σχετικές εργασίες και τρέχοντα συστήματα καλύπτονται στο Κεφάλαιο 2. Το πεδίο εφαρμογής του προβλήματος ορίζεται στο Κεφάλαιο 3 μαζί με την προσέγγιση εκπαίδευσης και τις τεχνικές δημιουργίας δεδομένων. Τα Κεφάλαια 4 και 5 ασχολούνται με την αρχιτεκτονική και την εκτέλεση του μοντέλου. Το Κεφάλαιο 6 εξετάζει τις τεχνικές αποκωδικοποίησης και την μετα-επεξεργασία. Το Κεφάλαιο 7 ολοκληρώνεται με περιορισμούς, αποτελέσματα και κατευθύνσεις για μελλοντική εργασία.

# Περίληψη

Με ιδιαίτερη έμφαση σε σενάρια δρομολόγησης με γεωγραφικούς περιορισμούς, όπως αυτά που συναντώνται στην Ελλάδα, η παρούσα διατριβή διερευνά τη χρήση αρχιτεκτονικών βαθιάς μάθησης βασισμένων σε μετασχηματιστές (Transformer) για το πρόβλημα του περιοδεύοντος πωλητή (TSP). Οι κλασικοί αλγόριθμοι επίλυσης συχνά αντιμετωπίζουν δυσκολίες με περιορισμούς συγκεκριμένων τομέων, ακανόνιστες τοπολογίες και δυναμικά μεγέθη εισόδου, παρόλο που έχουν επιδείξει εκπληκτική απόδοση σε τυπικές περιπτώσεις TSP. Αν και τα deep learning μοντέλα μπορούν να μάθουν ευριστικές πολιτικές για προβλήματα δρομολόγησης, οι πρόσφατες εξελίξεις στη νευρωνική συνδυαστική βελτιστοποίηση υποδηλώνουν ότι αυτό μπορεί να επιτευχθεί με προσαρμογή για πρακτική εφαρμογή, η οποία εξακολουθεί να αποτελεί δύσκολη εργασία.

Η εργασία αυτή αναπτύσσει και εκπαιδεύει ένα προσαρμοσμένο μοντέλο Transformer χρησιμοποιώντας μια στρατηγική προγράμματος σπουδών που καλύπτει μικρά συνθετικά γραφήματα έως τεράστια γεωχωρικά σύνολα δεδομένων του πραγματικού κόσμου. Η είσοδος για το μοντέλο είναι οι συντεταγμένες των πόλεων. Το μοντέλο δημιουργεί έγκυρες διαδρομές TSP χρησιμοποιώντας αυτό-παλινδρομική αποκωδικοποίηση με προσοχή στον δείκτη. Η εξωτερική ενσωμάτωση της διαχείρισης περιορισμών για τη συνδεσιμότητα λιμένων και νησιών επιτρέπει στο μοντέλο να γενικεύεται σε αραιά ή μερικώς συνδεδεμένα γραφήματα χωρίς αρχιτεκτονική.

Ερευνούμε διάφορες τεχνικές αποκωδικοποίησης, όπως βελτίωση 2-opt, αναζήτηση δέσμης και αποκωδικοποίηση greedy. Η αλληλεπίδραση σε πραγματικό χρόνο βασίζεται σε μια γραφική διεπαφή που επιτρέπει την οπτικοποίηση και τη δυναμική διαμόρφωση. Τα εμπόδια σε επίπεδο υλικού προκαλούν περιορισμούς στην απόδοση που παρατηρούνται κατά την ανάπτυξη. Αυτά επιλύονται με στρατηγικές συστάσεις για βελτιστοποίηση του backend.

Η διατριβή ολοκληρώνεται με προτάσεις για μελλοντικές κατευθύνσεις, όπως ενσωμάτωση με αναγνώριση περιορισμών, μάθηση κάλυψης σκοπιμότητας και επέκταση σε μεγαλύτερους τομείς δρομολόγησης, όπως το πρόβλημα δρομολόγησης οχημάτων (VRP). Τα αποτελέσματα δείχνουν ότι τα μοντέλα Transformer μπορούν να προσφέρουν ανταγωνιστικές και ευέλικτες απαντήσεις σε πραγματικές παραλλαγές του TSP όταν εκπαιδεύονται με δομημένη εποπτεία και ενισχύονται μέσω μάθησης με βάση πρόγραμμα σπουδών.

# Abstract

This thesis investigates a hybrid algorithmic framework based on Transformer-based model enhanced with domain-specific constraints and local optimization for Traveling Salesman Problem (TSP). Originally intended for natural language tasks, the Transformer architecture is used here to directly learn combinatorial routing strategies from data. Using curriculum learning and reinforcement fine-tuning, the model extends to asymmetric, real-world logistics situations including islands and port-access constraints unique to the Greek area. Greedy and beam search decoding then performs inference, optional 2-OPT metaheuristic refinement comes last. Combining interactive user input with real-time constraint validation, a graphical interface links the model. Comparative analyses show the method's competitiveness against classical heuristics including Ant Colony Optimization, Genetic Algorithms, and Nearest Neighbourhood. On cases with up to 100 nodes, the method achieves almost optimal performance while preserving responsiveness fit for pragmatic deployment. This work suggests a scalable basis for domain-aware TSP solvers and shows the possibilities of deep learning in operational routing systems.

# Περιεχόμενα

Πρόλογος.....	i
Περίληψη.....	ii
Abstract.....	iii
Κεφάλαιο 1: Εισαγωγή και Θεωρητικό Υπόβαθρο.....	1
1.1 Από το Symbolic AI στο Deep Learning.....	1
1.2 Μαθηματικά προαπαιτούμενα και θεωρία γραφημάτων.....	2
1.3 Η εξέλιξη της έρευνας για το TSP από τις διάτρητες κάρτες έως τους νευρωνικούς αγωγούς.....	3
1.4 Κλασικά παραδείγματα λύσεων TSP: Ακριβή, ευριστικά, στοχαστικά και υβριδικά.....	4
1.4.1 Ακριβείς μέθοδοι: Βέλτιστη λύση με αποδείξεις.....	4
1.4.2 Deterministic Heuristics: Επιτάχυνση πέρα από τις εγγυήσεις.....	5
1.4.3 Μεταευριστικές μέθοδοι: Στοχαστική κατεύθυνση πάνω από τοπικά ελάχιστα.....	5
1.4.4 Υβριδικές τεχνικές: προσέγγιση με στρώματα ικανοτήτων.....	5
1.5 Παραλλαγές του προβλήματος: Από το κλασικό TSP στη διαδρομολόγηση στο ελληνικό αρχιπέλαγος.....	6
1.5.1 Συμμετρικό TSP και ασύμμετρο TSP.....	6
1.5.2 Ομαδοποιημένο TSP (GTSP & CTSP) γενικά.....	6
1.5.3 Χρονικά παράθυρα, προαιρετικές επισκέψεις, συλλογή βραβείων.....	7
1.5.4 Εφαρμογή: Ελληνική θαλάσσια εφοδιαστική.....	7
1.6 Οδικός χάρτης της διατριβής και συνεισφορές.....	8
Σύνοψη.....	9
Κεφάλαιο 2: Σχετική Βιβλιογραφία και Τεχνικές Επίλυσης TSP.....	10
2.1 Μεθοδολογίες περιφέρειας.....	10
2.1.1 Μαθηματική Διατύπωση.....	10
2.1.2 Ακριβείς Μέθοδοι Επίλυσης.....	10
2.2 Κλασικές ευριστικές προσεγγίσεις.....	11
2.3 Μεταευριστικά εργαλεία.....	13
2.4 Προσεγγίσεις βασισμένες στη μάθηση.....	15
2.5 TSP Transformer-Based Techniques.....	15
2.5.1 Γιατί Transformers; Μια αλλαγή στον συνδυαστικό συλλογισμό.....	17
2.5.2 Architecture: Encoder–Decoder Attention Pipeline.....	17
2.5.3 Training Procedure: From Supervision to Reinforcement.....	18
2.5.4 Hybrid Inference: Greedy, Beam Search, 2-OPT.....	18
2.5.5 Στρατηγικές αποκωδικοποίησης: Υβριδική συμπερασματολογία για δρομολόγηση.....	19
2.5.6 GUI Routing Constraints Integration.....	20
2.5.7 Προκλήσεις, γενίκευση και πλεονεκτήματα.....	21
Σύνοψη.....	21
Κεφάλαιο 3: Περιορισμοί και Διαμόρφωση του Προβλήματος στον Ελληνικό Χώρο.....	22
3.1 Πεδίο εφαρμογής του προβλήματος και περιορισμοί στην Ελλάδα.....	22
3.1.1 Κίνητρα για την επιλογή της Ελλάδας.....	22
3.1.2 Γεωχωρική βάση δεδομένων ελληνικών πόλεων και χαρακτηριστικών.....	23
3.1.3 Τοπολογικοί περιορισμοί.....	23
3.1.4 Προβλήματα παραλλαγών που προκαλούνται από τη γεωγραφική δομή.....	25

3.1.5 Ορισμός του πεδίου εφαρμογής της εκπαίδευσης του μετασχηματιστή.....	25
3.2 Σχεδιασμός προγράμματος σπουδών και δημιουργία δεδομένων.....	27
3.2.1 Συνθετική δημιουργία γραφημάτων: Dual Source Pipeline.....	27
3.2.2 Μεθοδολογία γεωγραφικής δειγματοληψίας.....	27
3.2.3 Λογική της μάθησης του προγράμματος σπουδών.....	29
3.2.4 Εποπτευόμενη έναντι ενισχυμένης εκπαίδευσης.....	30
3.2.5 Στρωματοποιημένα σύνολα δοκιμών: Αξιολόγηση.....	31
3.2.6 Μάσκα, συμπλήρωση και ομαδοποίηση για εισόδους ποικίλου μήκους.....	32
Σύνοψη.....	33
Κεφάλαιο 4: Αρχιτεκτονική Συστήματος και Διεπαφή Χρήστη (GUI).....	34
4.1 Ερμηνεία γραφημάτων και κωδικοποίηση χωρικών χαρακτηριστικών.....	34
4.2 Κωδικοποίηση θέσεων και αμεταβλητότητα μεταθέσεων.....	35
4.3 Τεκμηρίωση Εξωτερικών Περιορισμών.....	38
4.4 Πρακτική Επιβολή Περιορισμών στο GUI.....	38
4.5 Συγκριτική ανάλυση με προηγούμενες τεχνικές κωδικοποίησης.....	40
Σύνοψη.....	41
Κεφάλαιο 5: Υλοποίηση Αλγορίθμων και Εκπαίδευση Μοντέλου.....	42
5.1 Κωδικοποιητής και μοντέλο χωρικής προσοχής.....	42
5.2 Λογική αποκωδικοποιητή και δείκτη προσοχής.....	43
5.3 Αυτοπαλινδρομικοί βρόχοι και τρόποι συμπερασμού.....	44
5.4 Αρχιτεκτονική και σχεδιαστικές αποφάσεις.....	45
5.5 Σχέδια εκπαίδευσης και βελτιστοποίηση πολιτικής.....	46
5.6 Έλεγχος προγράμματος σπουδών και εποπτευόμενη προθέρμανση.....	47
5.7 Σήματα εκπαίδευσης και δομή ανταμοιβής.....	48
5.8 Επισκόπηση υπερπαραμέτρων και οδηγίες βελτιστοποίησης.....	49
5.9 Παρακολούθηση, επαλήθευση και έλεγχος.....	50
Σύνοψη.....	51
Κεφάλαιο 6: Στρατηγικές Συμπερασμού και Υβριδική Αποκωδικοποίηση.....	52
6.1 Επισκόπηση της αυτόματης αποκωδικοποίησης.....	52
6.2 Greedy Decoding.....	52
6.3 Αναζήτηση με χρήση ακτίνων.....	53
6.4 Πιθανιστικό δείγμα.....	54
6.5 Διόρθωση περιορισμών χρησιμοποιώντας τη λογική των λιμανιών και των νησιών.....	55
6.6 Βελτίωση περιηγήσεων 2-Opt.....	55
6.6.1 Δυναμική κατάρτιση προϋπολογισμού 2-Opt με όρια αλγορίθμου.....	56
6.6.2 Παραδείγματα για Διάφορα Πλήθη Πόλεων.....	57
Σύνοψη.....	60
Κεφάλαιο 7: Πειραματική Αξιολόγηση, Περιορισμοί και Συζήτηση.....	61
7.1 Εμπειρική διάταξη και σημειώσεις.....	61
7.2 Χρήσιμες δυσκολίες και σημειώσεις σχετικά με τη συμπεριφορά.....	61
7.3 Περιορισμοί και δυνατότητες υλικού.....	62
7.4 Όρια μοντέλου και σχεδιαστικές συμβιβαστικές λύσεις.....	63
Σύνοψη.....	64
Κεφάλαιο 8: Συμπεράσματα.....	66
Παραρτήματα.....	67
Παράρτημα Α - Κώδικας Βάσης Δεδομένων Ελληνικών Πόλεων.....	67
Παράρτημα Α.1 – Συνάρτηση build_records.....	67
Παράρτημα Α.2 – Συνάρτηση write_to_db.....	70

Παράρτημα Β - Κώδικας Εκπαίδευσης Μοντέλου.....	72
Παράρτημα Β.1.1 – PositionalEncoding.....	72
Παράρτημα Β.1.2 – PointerDecoder.....	72
Παράρτημα Β.1.3 – TransformerTSP.....	73
Παράρτημα Β.2 – train_transformer_curriculum.py.....	75
Παράρτημα Β.3 – reinforce_finetune.py.....	78
Παράρτημα Β.4 – hyperparams.py.....	82
Παράρτημα Γ – Υβριδική Αποκωδικοποίηση και 2-OPT.....	84
Παράρτημα Γ.1 – Μέθοδος sample_with_logprob (από transformer_tsp.py).....	84
Παράρτημα Γ.2 – Κλήση apply_two_opt και χρήση της.....	87
Παράρτημα Δ – Κανόνες Επικύρωσης Περιορισμών Δρομολόγησης.....	88
Παράρτημα Δ.1 – Μέθοδοι Ελέγχου Επιλογής Πόλεων.....	88
Παράρτημα Ε – Περιβάλλον Εκτέλεσης και Εξαρτήσεις.....	89
Βιβλιογραφία.....	90
Ακαδημαϊκές πηγές.....	90
GitHub Αποθετήρια.....	91

## Κατάλογος Σχημάτων

Σχήμα 2.1 - Τυπική αρχιτεκτονική του μοντέλου Transformer, όπως παρουσιάστηκε από τους Vaswani et al. (2017). Πηγή: <a href="https://jalammr.github.io/illustrated-transformer/">https://jalammr.github.io/illustrated-transformer/</a> .....	17
Σχήμα 4.1 - Κεντρική γραφική διεπαφή χρήστη (GUI) του συστήματος δρομολόγησης.....	35
Σχήμα 4.2 - Μήνυμα σφάλματος GUI για νησιά χωρίς λιμένα.....	39
Σχήμα 4.3 - Επιλογές ελέγχου 2-OPT και εύρους δέσμης μέσω ρυθμιστικών GUI.....	40
Σχήμα 6.1 - Διαδρομή μετασχηματιστή για 5 πόλεις.....	57
Σχήμα 6.2 - Διαδρομή μετασχηματιστή για 10 πόλεις.....	58
Σχήμα 6.3 - Διαδρομή μετασχηματιστή για 20 πόλεις.....	58
Σχήμα 6.4 - Διαδρομή μετασχηματιστή για 30 πόλεις.....	59

## Κατάλογος Πινάκων

Πίνακας 1.1 - Παραλλαγές του Προβλήματος του Περιοδεύοντος Πωλητή και Σχετικότητα με την Ελλάδα.....	7
Πίνακας 2.1 - Σύγκριση των αλγορίθμων επίλυσης του TSP ως προς τύπο, πολυπλοκότητα και γεωγραφική εφαρμογή.....	13
Πίνακας 3.1 - Σύγκριση των βασικών χαρακτηριστικών εισόδου, συνδεσιμότητας και εκπαίδευσης μεταξύ προηγούμενων προσεγγίσεων και της παρούσας διατριβής.....	40

# Κεφάλαιο 1: Εισαγωγή και Θεωρητικό Υπόβαθρο

## 1.1 Από το Symbolic AI στο Deep Learning

Η τεχνητή νοημοσύνη (AI), που προέρχεται αρχικά από τη συμβολική συλλογιστική, βρήκε για πρώτη φορά έκφραση σε πρώιμα συστήματα όπως το Logic Theorist και το General Problem Solver, τα οποία έλυναν λογικά προβλήματα χρησιμοποιώντας χειροκίνητα κωδικοποιημένους κανόνες. Αργότερα, ο John Haugeland αναφέρθηκε σε αυτό το παράδειγμα ως GOF AI (Good Old-Fashioned AI). Αν και ερμηνεύσιμα, αυτά τα αλγόριθμα έρχονταν σε σύγκρουση με τα δεδομένα της αντίληψης, όπως η όραση ή η ομιλία, και δεν μπορούσαν να γενικευτούν σε θορυβώδεις, υψηλής διάστασης εισόδους. Το πραγματικό εμπόδιο δεν ήταν η λογική, αλλά η αφαίρεση των προτύπων. Οι εξελκτικές θεωρίες προσαρμογής, όπως οι γενετικοί αλγόριθμοι του Holland [5], που παρουσίασαν μηχανισμούς μάθησης βασισμένους στην επιβίωση των καταλληλότερων λύσεων, ενίσχυσαν την τάση προς τα συστήματα βασισμένα στη μάθηση.

Μια άλλη σχολή, ο συνδεδετισμός, μοντελοποίησε τη νοημοσύνη μέσω δικτύων μονάδων που μοιάζουν με νευρώνες. Σημείο καμπής ήταν η ανακάλυψη της αντίστροφης διάδοσης (backpropagation) τη δεκαετία του 1980, η οποία συνέβαλε στην αναβίωση των πολυεπίπεδων δικτύων. Ωστόσο, η βαθιά μάθηση έγινε δημοφιλής μόνο μετά το 2012, όταν το AlexNet, που εκπαιδεύτηκε σε GPU από τον Krizhevsky και την ομάδα του, μείωσε κατά το ήμισυ το ποσοστό σφάλματος του ImageNet. Το μάθημα ήταν προφανές: με επαρκή δεδομένα και υπολογιστική ισχύ, τα βαθιά δίκτυα μπορούσαν να μάθουν χρήσιμα χαρακτηριστικά, ξεπερνώντας έτσι την εύθραυστη μηχανική χαρακτηριστικών που εμπόδιζε τα συμβολικά συστήματα.

Δύο ιδέες αποτελούν τη βάση της σύγχρονης βαθιάς μάθησης: η ιεραρχική αφαίρεση και ο διαφορίσιμος προγραμματισμός. Τα μοντέλα μπορούν να βελτιστοποιηθούν από άκρο σε άκρο με βαθμιδωτή κατάβαση, συγκεντρώνοντας την προετοιμασία των δεδομένων, την αρχιτεκτονική του μοντέλου και τις συναρτήσεις απώλειας σε έναν ενιαίο διαφορίσιμο αγωγό. Τα κρυφά στρώματα καταγράφουν τις λανθάνουσες ιεραρχίες, μαθαίνοντας να ομαδοποιούν τις ακμές σε σχήματα ή τα φωνήματα σε λέξεις

Τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN) που χρησιμοποιήθηκαν για πρώτη φορά στη μοντελοποίηση ακολουθιών διαδίδουν τις πληροφορίες βήμα προς βήμα. Τα RNN υπέφεραν, εν τω μεταξύ, από κακή παραλληλισμό και εξαφανιζόμενες κλίσεις. Η αρχιτεκτονική Transformer που παρουσίασαν οι Vaswani και η ομάδα του, το 2017 αντικατέστησε την επαναληψιμότητα με την αυτοπροσοχή, έναν μηχανισμό που επιτρέπει σε κάθε token να παρακολουθεί ταυτόχρονα όλα τα άλλα χρησιμοποιώντας πράξεις πίνακα. Η καλύτερη επεκτασιμότητα και ο παραλληλισμός που κατέστησαν δυνατοί χάρη στην αυτοπροσοχή βοήθησαν να ανοίξει ο δρόμος για τα βασικά μοντέλα που σήμερα δημιουργούν εικόνες, κώδικα και περιλήψεις κειμένων.

Το πιο σημαντικό είναι ότι τα συστήματα προσοχής ταιριάζουν απόλυτα στις προκλήσεις της συνδυαστικής βελτιστοποίησης. Προβλήματα όπως η δρομολόγηση και ο προγραμματισμός απαιτούν

μεθοδική λήψη αποφάσεων για σύνολα, δηλαδή ακριβώς το είδος της δομής που μπορούν να αναπαράγουν Transformers με τις αλληλεπιδράσεις που έχουν μάθει μεταξύ των στοιχείων.

## 1.2 Μαθηματικά προαπαιτούμενα και θεωρία γραφημάτων

Οι αναπαραστάσεις με βάση γραφήματα και οι διατυπώσεις πινάκων αποτελούν τη βάση όλων των τεχνικών βελτιστοποίησης στην ανάλυση του προβλήματος του περιοδεύοντος πωλητή (TSP), συμβολικών, ευριστικών ή νευρωνικών. Σε αυτή την ενότητα εξηγούνται οι βασικές έννοιες της θεωρίας γραφημάτων και της πολυπλοκότητας που απαιτούνται για τα επόμενα κεφάλαια, παρά το γεγονός ότι δεν περιλαμβάνονται τεχνικές παραγωγίσιμης που δεν αναφέρονται άμεσα στη λύση που βασίζεται στον μετασχηματιστή.

### Γραφήματα, σημειογραφία και δομή

Έστω  $G=(V,E)$ . Ο πεπερασμένος γράφος  $G=(V,E)$  έχει σύνολο κορυφών  $V=\{1,2,\dots,n\}$  και σύνολο ακμών  $E$ . Στην συμμετρική περίπτωση, οι ακμές είναι μη κατευθυνόμενα ζεύγη  $\{i,j\}$ , ενώ στην ασύμμετρη περίπτωση οι ακμές είναι ταξινομημένα ζεύγη  $(i,j)$ . Συνήθως υποδηλώνοντας αποστάσεις ή έξοδα, κάθε ακμή συγκρατείται συλλογικά σε έναν πίνακα κόστους  $C=[c_{ij}]$  μέσω ενός μη αρνητικού βάρους  $c_{ij}$ . Συνήθως ξεκινώντας από ευκλείδειες αποστάσεις σε  $\mathbb{R}^d$ , αυτά τα έξοδα λογιστικής βοηθούν στην ικανοποίηση της ανισότητας του τριγώνου.

Μια κλειστή περιήγηση που επισκέπτεται ακριβώς μία φορά κάθε κόμβο είναι ένας κύκλος Hamilton. Το TSP βρίσκει μια τέτοια περιήγηση που ελαχιστοποιεί το συνολικό κόστος

$$\min_{p_i \in S} \sum_{i=1}^n c_{p_i, p_{i+1}}, p_{n+1} = p_1 \tag{1.1}$$

Ο αριθμός τέτοιων περιηγήσεων για μη κατευθυνόμενα γραφήματα είναι παραγοντικός σε μέγεθος, π.χ.  $(n-1)!/2$  το πρόβλημα είναι συνδυαστικά εκρηκτικό ακόμη και για μέτρια  $nn$ .

### Υπολογιστική πολυπλοκότητα

Το πρόβλημα απόφασης TSP είναι αν υπάρχει μια διαδρομή με κόστος  $\leq B$ . Είναι NP-πλήρες. Αν και η εγκυρότητα της διαδρομής μπορεί να επαληθευτεί σε  $O(n)$ , η εύρεση της βέλτιστης διαδρομής είναι υπολογιστικά δύσκολη. Το μετρικό TSP δέχεται την προσέγγιση 1,5 του Christofides [7], όπου ισχύει η ανισότητα του τριγώνου. Όχι μόνο  $P=NP$ , αλλά επίσης δεν είναι γνωστή καμία προσέγγιση σταθερού συντελεστή για το ευρύ ATSP.

Η τεχνική Bellman-Held-Karp [14] επιλύει τέλεια το TSP σε  $O(n^2 \cdot 2^n)$ , περιορίζοντας τους ακριβείς επιλυτές σε περίπου  $n \leq 25$ . Αν και είναι εκθετική, αυτή η μέθοδος δυναμικού προγραμματισμού χρησιμοποιείται μερικές φορές για κατώτερα όρια σε προσεγγίσεις Branch and Bound.

### Πολυεδρική αρχιτεκτονική

Οι πολυεδρικές μέθοδοι μεταφράζουν εφικτές λύσεις TSP σε κορυφές κυρτών πολυτόπων. Η βασική διατύπωση DFJ [6] προσθέτει δυαδικές μεταβλητές ακμών και αφαιρεί υποκύκλους μέσω εκθετικά

πολλών περιορισμών. Αν και ισχυρές, τέτοιες διατυπώσεις δεν εμπίπτουν στο πεδίο της παρούσας εργασίας. Ο γραμμικός προγραμματισμός ή οι τεχνικές κοπής επιπέδων δεν εφαρμόζονται ρητά στην παρούσα εργασία. Για τους ενδιαφερόμενους αναγνώστες, οι εργασίες των Padberg, Grottschel και άλλων προσφέρουν μια σε βάθος μελέτη των πολυεδρικών μεθόδων.

### **Ενημερωμένα σύνολα κοπής και ανώτερα και κατώτερα όρια**

Παραλείπονται οι αναλυτικές αποδείξεις για τα όρια, τις ανισότητες χτένας και τις τομές ισοτιμίας τύπου "domino". Αν και δεν αναφέρονται στις μεθόδους βασισμένες σε μετασχηματιστές ή στις ευριστικές μεθόδους της παρούσας διατριβής, αυτές οι κλασικές μέθοδοι βοηθούν το Concorde και άλλους βιομηχανικούς επιλυτές. Ο Concorde βασίζεται στις εργασίες των Applegate και της ομάδος του. [15].

## **1.3 Η εξέλιξη της έρευνας για το TSP από τις διάτρητες κάρτες έως τους νευρωνικούς αγωγούς**

Μέσα από το υπολογιστικό πρίσμα, κάθε τεχνολογική εποχή έχει ερμηνεύσει εκ νέου το πρόβλημα του περιοδεύοντος πωλητή (TSP). Το TSP παρέμεινε ένα βασικό σημείο αναφοράς για τη συνδυαστική βελτιστοποίηση, από τα συμβολικά γραμμικά προγράμματα σε συστήματα διάτρητων καρτών έως τα βαθιά νευρωνικά δίκτυα σε GPU. Αυτή η εξέλιξη βοηθά να κατανοήσουμε γιατί η παρούσα διατριβή συνδυάζει τη σύγχρονη σκέψη που βασίζεται στον μετασχηματιστή με την κλασική τοπική αναζήτηση.

### **Πρώιμες ακριβείς τεχνικές: η εποχή του κοπτικού επιπέδου της δεκαετίας του 1950**

Οι Dantzig, Folkerson και Johnson (DFJ) ανέπτυξαν την ακριβή επίλυση του TSP το 1954, διατυπώνοντας το πρόβλημα χρησιμοποιώντας ένα γραμμικό πρόγραμμα με εκθετικά πολλούς περιορισμούς για την εξάλειψη των υποδιαδρομών. Σχεδιασμένη σε έναν IBM 701, η επαναληπτική προσέγγιση κοπτικού επιπέδου που ανέπτυξαν πέτυχε τη βέλτιστη λύση για 49 πόλεις των ΗΠΑ και αποτέλεσε τη βάση για τους σύγχρονους ακριβείς επιλυτές [6].

### **Ευριστικές μέθοδοι και όρια: Σημαντικές εξελίξεις 1958–1980**

Οι ευριστικές μέθοδοι αναπτύχθηκαν για να παρέχουν γρήγορες, υψηλής ποιότητας προσεγγίσεις, καθώς οι ακριβείς τεχνικές είχαν χαμηλή απόδοση. Η απλή ανταλλαγή ακμών προτάθηκε για πρώτη φορά από τον Croes (1958) για τη βελτίωση των περιηγήσεων [13]. Ακολούθησαν επεκτάσεις στο 3-OPT και η ισχυρή ευριστική μέθοδος Lin–Kernighan[8]. Οι Held & Karp παρουσίασαν ταυτόχρονα το κατώτερο όριο 1-tree Lagrangian, που εξακολουθεί να χρησιμοποιείται ευρέως σε συστήματα branch-and-bound, και ανέπτυξαν έναν αλγόριθμο δυναμικού προγραμματισμού για μικρά TSP [14].

### **Πολυεδρική βελτιστοποίηση και συγκριτική αξιολόγηση ( 1980–1990 )**

Οι ερευνητές επέστρεψαν στις ακριβείς τεχνικές αυτή τη φορά εξοπλισμένοι με πολυεδρική συνδυαστική χρησιμοποιώντας την αυξανόμενη υπολογιστική ικανότητα. Συγκράτηση χαλάρωσης

## Κεφάλαιο 1

LP, τεχνικές που περιλαμβάνουν comb cuts και ανισότητες clique-tree. Η σουίτα benchmark TSPLIB, που εισήχθη από τον Reinelt το 1991, εξελίχθηκε σε ένα τυποποιημένο δοκιμαστικό περιβάλλον [12] για την αξιολόγηση αλγορίθμων, ενθαρρύνοντας έτσι την επαναληψιμότητα και την αυστηρή ανάλυση.

### **Μεταευριστική και αλγόριθμοι που σπάνε ρεκόρ (1990–2000)**

Μαζί με τη συνεχή βελτίωση των επιλυτών διακλάδωσης και κοπής, η δεκαετία του 1990 έφερε την εμφάνιση των μεταευριστικών μεθόδων (π.χ. Tabu Search, Simulated Annealing, Genetic Algorithms). Εξοπλισμένοι με το TSPLIB και όλο και πιο ισχυρό υλικό, οι ερευνητές δημιούργησαν επιλυτές ικανούς να διαχειρίζονται εκατοντάδες ή χιλιάδες πόλεις. Για ακριβείς λύσεις, ο Concorde TSP Solver εξελίχθηκε ως η τελευταία λέξη της τεχνολογίας [15].

### **Σύμφωνα με τις προσεγγίσεις που βασίζονται στη μάθηση (2000–σήμερα)**

Τα τελευταία χρόνια έχει σημειωθεί μια αλλαγή παραδείγματος: η μηχανική μάθηση επιδιώκει να μάθει να αναζητά αντί να προγραμματίζει ευριστικές μεθόδους. Τα δίκτυα δεικτών (Pointer Networks, Vinyals et al.) παρουσίασαν νευρωνικά μοντέλα ικανά να παράγουν κατά προσέγγιση συνδυαστικές λύσεις [16]. Χωρίς επαναλήψεις και εμπόδια, οι μετασχηματιστές επέτρεψαν περαιτέρω την παράλληλη συμπερασματολογία σε μεγάλης κλίμακας δομημένα προβλήματα, συμπεριλαμβανομένης της δρομολόγησης [17].

## **1.4 Κλασικά παραδείγματα λύσεων TSP: Ακριβή, ευριστικά, στοχαστικά και υβριδικά**

Έχει αναπτυχθεί ένα ευρύ φάσμα μεθόδων για την επίλυση του TSP, από στοχαστικούς προσεγγιστές έως θεωρητικά ακριβείς λύτες. Αν και στην παρούσα διατριβή δίνεται έμφαση στην εξαγωγή συμπερασμάτων με βάση τον μετασχηματιστή, η ανάπτυξη συμπληρωματικών τεχνικών εξαρτάται από την επίγνωση των παραδοσιακών παραδειγμάτων.

### **1.4.1 Ακριβείς μέθοδοι: Βέλτιστη λύση με αποδείξεις**

[Ο δυναμικός προγραμματισμός ιδιαίτερα ο αλγόριθμος Bellman Held Karp [14] παραμένει ένας βασικός ακριβής λύτης που περιορίζει την πρακτική χρήση σε  $n \leq 25$  με εκθετική πολυπλοκότητα  $O(n^2 \cdot 2^n)$ .

Η μέθοδος Branch-and-Cut, που εφαρμόζεται στον Concorde TSP Solver [15], εξακολουθεί να είναι η πιο προηγμένη για εφαρμογές μεγάλης κλίμακας. Επιλύει χαλαρώσεις LP της διατύπωσης DFJ [6], δημιουργεί επίπεδα κοπής (π.χ. comb, clique-tree) και διακλαδίζεται σε κλασματικές μεταβλητές. Το Concorde απέδειξε την καλύτερη απόδοση σε ένα σενάριο 24.978 πόλεων, παρόλο που απαιτούσε >32 GB μνήμης RAM και εκτεταμένο χρόνο εκτέλεσης.

Η παρούσα διατριβή αναπτύσσει λύσεις ground-truth και αξιολογεί την ποιότητα των αποτελεσμάτων του Transformer μέσω μιας ελαφριάς υλοποίησης Branch-and-Bound για μικρές περιπτώσεις (έως 14 πόλεις).

#### 1.4.2 Deterministic Heuristics: Επιτάχυνση πέρα από τις εγγυήσεις

Οι ευριστικές μέθοδοι δίνουν προτεραιότητα στην ταχύτητα, συμβιβάζοντας έτσι τη βέλτιστη απόδοση. Χωρίζονται στις ακόλουθες ομάδες:

Κατασκευαστική: ξεκινά μια περιήγηση από το μηδέν (όπως Nearest Neighbour, Farthest Insertion).

Η NN εκτελείται σε  $O(n^2)$  για ευκλείδειους γραφήματα 100 πόλεων, παράγοντας λύσεις περίπου 25% πάνω από το ιδανικό. Η FI συνήθως εκτελείται σε περίπου 10%.

Η επανασύνδεση ακμών βελτιώνει μια περιήγηση που είναι ήδη σε χρήση.

Το διάσημο 2-OPT [13] βελτιώνει επανειλημμένα μια διαδρομή χρησιμοποιώντας δύο άκρες. Γενίκευση της 2-OPT χρησιμοποιείται θεωρητικά μέσω προσαρμοστικών κινήσεων όπως οι k-opt και οι αλγόριθμοι Lin-Kernighan (LK)[8]. Η σύγχρονη παραλλαγή LKH-3 παρουσιάζει εξαιρετική απόδοση στο TSPLIB, **αλλά δεν εφαρμόστηκε στο παρόν έργο**. Στη διατριβή αυτή, υλοποιείται η 2-OPT ως μετα-επεξεργασία των λύσεων του μετασχηματιστή.

Εφαρμοσμένες σε αυτή τη διατριβή χρησιμοποιώντας 2-OPT, αυτές οι εξαιρετικά διαφανείς deterministic διαδικασίες είναι ιδανικές για την μετα-επεξεργασία λύσεων Transformer.

#### 1.4.3 Μεταευριστικές μέθοδοι: Στοχαστική κατεύθυνση πάνω από τοπικά ελάχιστα

Οι μεταευριστικές μέθοδοι επεκτείνουν τις ευριστικές μεθόδους με απρόβλεπτο χαρακτήρα για να ξεφύγουν από τοπικά βέλτιστα. Τυπικές μέθοδοι περιλαμβάνουν:

- γενετικοί αλγόριθμοι (GA) [5] χρησιμοποιώντας δύο-opt
- βελτιστοποίηση αποικιών μυρμηγκιών (ACO) [4] με LK. Στην παρούσα διατριβή, εφαρμόζονται οι βασικές μορφές GA και ACO χωρίς χρήση LK.

Με την κατάλληλη διαμόρφωση, αυτές οι μέθοδοι παρέχουν ~1% απόκλιση από τη βέλτιστη λύση σε λίγα λεπτά για προβλήματα 1.000 κόμβων, κάτι που τις καθιστά κατάλληλες για εργασίες βελτιστοποίησης που διαρκούν όλη τη νύχτα.

#### 1.4.4 Υβριδικές τεχνικές: προσέγγιση με στρώματα ικανοτήτων

Συνήθως, οι πρακτικοί λύτες δεν βασίζονται σε ένα μόνο πρότυπο. Αντίθετα, οι ειδικοί συνδυάζουν διάφορες στρατηγικές:

Το Concorde [15] χρησιμοποιεί διακλάδωση και αποκοπή μαζί με τοπική αναζήτηση. Η παρούσα διατριβή δεν το χρησιμοποιεί, αλλά αντλεί έμπνευση από την ιδέα της υβριδοποίησης μεταξύ παραδοσιακής αναζήτησης και νευρωνικών μεθόδων.

- Η αναζήτηση δέσμης ή το συμβατικό 2-OPT [13] μαζί με την επαγωγή βάσει προσοχής θα μπορούσαν να συνδυαστούν σε υβριδικές νευρωνικές διοχετεύσεις.

Η παρούσα διατριβή τονίζει ως βασικό μηχανισμό την υβριδοποίηση: Συνδυάζοντας τη συνδυαστική ακρίβεια με τη νευρωνική γενίκευση, οι μονάδες αναζήτησης δέσμης και το 2-OPT συμβάλλουν στη βελτίωση των περιηγήσεων που δημιουργούνται από μετασχηματιστές.

### Σύνοψη

Εδώ παρουσιάζουμε μια πολυεπίπεδη προοπτική των κλασικών μεθόδων TSP. Ενώ οι ακριβείς λύσεις παρέχουν θεωρητική ακρίβεια και οι ευριστικές μέθοδοι αποδίδουν γρήγορα αποτελέσματα, οι υβριδικές προσεγγίσεις ιδιαίτερα όταν ενισχύονται από τη μάθηση προσφέρουν τα καλύτερα και των δύο κόσμων.

### 1.5 Παραλλαγές του προβλήματος: Από το κλασικό TSP στη διαδρομολόγηση στο ελληνικό αρχιπέλαγος

Αν και οι λύσεις των εγχειριδίων για το πρόβλημα του περιοδεύοντος πωλητή (TSP) παρουσιάζουν έναν τακτοποιημένο κύκλο σε ένα δεδομένο σύνολο πόλεων, οι πραγματικές εργασίες δρομολόγησης δεν είναι τόσο συνεπείς. Η δρομολόγηση μέσω θαλάσσιων μεταφορών περιλαμβάνει περιορισμούς χρονοδιαγραμμάτων, δυνατότητα προαιρετικών ή συνδυασμένων στάσεων, και κατευθυντικά κόστη μετακίνησης λόγω ασυμμετρίας. Τυποποίηση των παραλλαγών του προβλήματος και τον τρόπο με τον οποίο καταλήγουν σε μορφές συμβατές με το TSP μέσω μετασχηματισμών γραφημάτων ή μάσκες προσοχής πριν την δημιουργία ενός λύτη, κλασικό ή νευρωνικό.

#### 1.5.1 Συμμετρικό TSP και ασύμμετρο TSP

Η απόσταση μεταξύ των κόμβων στο συμμετρικό TSP (STSP) είναι η ίδια και στις δύο κατευθύνσεις, Στο ασύμμετρο TSP (ATSP), το κόστος από  $i \rightarrow j$  μπορεί να διαφέρει από  $j \rightarrow i$ , μοντελοποιώντας καταστάσεις όπως η κατανάλωση καυσίμου σε ορεινούς δρόμους ή διαδρομές απο πορθμούς που επηρεάζονται από το ρεύμα και την κατεύθυνση.

Δεδομένης της ασυμμετρίας που εισάγεται από τις μεταφορές με φέριμποτ στο ελληνικό αρχιπέλαγος, η παρούσα διατριβή κωδικοποιεί όλα τα δεδομένα διαδρομών σε μορφή ATSP κατά τη μοντελοποίηση και την εκπαίδευση.

#### 1.5.2 Ομαδοποιημένο TSP (GTSP & CTSP) γενικά

Η διαδρομή στο Γενικευμένο TSP (GTSP) πρέπει να επισκέπτεται έναν κόμβο ανά σύμπλεγμα, καθώς το σύνολο των κόμβων είναι χωρισμένο. Αυτό μοντελοποιεί την επιλογή ενός λιμανιού για κάθε ομάδα νησιών. Αν και το GTSP είναι NP-δύσκολο, η μετατροπή Noon-Bean [1] επιτρέπει σε κάθε κόμβο να χωριστεί σε κλώνους «εισόδου» και «εξόδου» που συνδέονται με τόξα μηδενικού κόστους, δημιουργώντας έτσι ένα ισοδύναμο ATSP [25]. Μία από τις πρώτες προσεγγίσεις επίλυσης του GTSP βασίστηκε στη δυναμική προγραμματισμό, όπως προτάθηκε από τον Labordère [3].

Χρησιμοποιώντας δυναμική προεπεξεργασία, η παρούσα διατριβή υιοθετεί μετασχηματισμούς GTSP, επιτρέποντας έτσι τη συνεπή αξιολόγηση σε κλασικούς επιλυτές, ευριστικές μεθόδους και τον Μετασχηματιστή.

Το Clustered TSP (CTSP) [2] επιβάλλει περισσότερους περιορισμούς: οι κόμβοι εντός του ίδιου cluster πρέπει να επισκέπτονται διαδοχικά. Λόγω του απλοποιημένου χώρου αναζήτησης, οι δομές

CTSP είναι ιδιαίτερα κατάλληλες για τις πρώτες φάσεις της ανάπτυξης του προγράμματος σπουδών. Οι βασικές παραλλαγές συνοψίζονται στον Πίνακα 1.

**Πίνακας 1.1 - Παραλλαγές του Προβλήματος του Περιοδεύοντος Πωλητή και Σχετικότητα με την Ελλάδα**

Παραλλαγή	Περιγραφή	Ισχύει για την Ελλάδα;
GTSP	Ομαδοποιημένες πόλεις; επιλέξτε 1 ανά ομάδα	✓ (π.χ. νησιωτικά σύμπλεγματα)
CTSP	Πόλεις σε προκαθορισμένα σύμπλεγματα	✓ (π.χ. Κρήτη ή Αττική)
PCTSP	Προαιρετικές επισκέψεις σε πόλεις με ανταμοιβές	✓ (μικρά νησιά)
mTSP	Απαιτούνται πολλοί πωλητές	✗ Δεν ισχύει
Standard TSP	Επίσκεψη όλων των πόλεων ακριβώς μία φορά	✓ Βασική γραμμή

**1.5.3 Χρονικά παράθυρα, προαιρετικές επισκέψεις, συλλογή βραβείων**

Άλλες παραλλαγές σχετικές με την πρακτική εφοδιαστική περιλαμβάνουν:

TSP ( Prize-collecting TSP): Ορισμένοι κόμβοι είναι προαιρετικοί. Ο στόχος είναι να εξισορροπηθεί το συνολικό κόστος της διαδρομής με το όφελος από την επίσκεψη σε κάθε κόμβο.

Κάθε κόμβος στο TSP με χρονικά παράθυρα (TSP-TW) έχει ένα χρονικό εύρος εντός του οποίου πρέπει να επισκεφθεί.

Αν και το παρόν μοντέλο δεν εφαρμόζει πλήρως αυτές τις παραλλαγές, ο αρθρωτός σχεδιασμός του Transformer και του αποκωδικοποιητή του επιτρέπει τη μελλοντική συμπερίληψη τέτοιων περιορισμών μέσω μάσκας προσοχής ή μετα-αποκωδικοποιητών.

**1.5.4 Εφαρμογή: Ελληνική θαλάσσια εφοδιαστική**

Η παρούσα διατριβή επικεντρώνεται σε ένα πραγματικό σενάριο όπου πρέπει να διαδρομολογηθεί μια διαδρομή μεταξύ ελληνικών πόλεων και νησιών με περιορισμούς που εξαρτώνται από τα πλοία. Η εκπαίδευση και η διαδικασία συμπερασμού μοντελοποιούν ασυμμετρίες, ομαδοποιημένους λιμένες και περιορισμούς διέλευσης.

Η εφαρμογή των GTSP και CTSP σε συνδυασμό με τη γεωγραφική μοντελοποίηση προσοχής βοηθά το Transformer να μάθει λογικά μοτίβα δρομολόγησης στην ελληνική επικράτεια.

### Σύνοψη

Η προσαρμογή των επιλυτών σε λογικές εφαρμογές εξαρτάται από την επίγνωση των παραλλαγών του TSP. Η παρούσα διατριβή εξετάζει συγκεντρωμένα, ασύμμετρα και θαλάσσια περιβάλλοντα συνδυάζοντας τη μάθηση με βάση το Transformer με κλασικές μετασχηματισμούς GTSP.

### 1.6 Οδικός χάρτης της διατριβής και συνεισφορές

Οι θεωρητικές τεχνικές βελτιστοποίησης εξακολουθούν να αποκλίνουν από τις πρακτικές ανάγκες της πραγματικής δρομολόγησης. Οι ευριστικοί αλγόριθμοι αντιμετώπισης προβλημάτων αντιμετωπίζουν δυσκολίες με ασύμμετρα κόστη ή συγκεντρωμένα σημεία ενδιαφέροντος, ακόμη και αν είναι γρήγοροι. Αντίθετα, οι ακριβείς αλγόριθμοι παρέχουν βέλτιστα αποτελέσματα, αλλά καθώς αυξάνεται το μέγεθος των γραφημάτων, καθίστανται υπολογιστικά αδύνατοι. Εμπνευσμένη από επαγγελματίες που διαχειρίζονται την ελληνική θαλάσσια εφοδιαστική, η διατριβή αυτή προτείνει μια υβριδική προσέγγιση που συνδυάζει τη συνδυαστική

#### Κύριες συνεισφορές

##### Μετασχηματιστής εξοπλισμένος για δρομολόγηση με βάση το πρόγραμμα σπουδών

Εισάγουμε ένα μοντέλο μετασχηματιστή σχεδιασμένο με μάθηση τύπου “*curriculum*”[10]. Ξεκινώντας με συμμετρικές εργασίες 5 κόμβων, η εκπαίδευση προχωρά σταδιακά προς ασύμμετρα, ομαδοποιημένα δίκτυα 120 κόμβων. Η συμπερίληψη της τοπικής δομής 2-OPT από νωρίς στην εκπαίδευση μειώνει το μέσο κενό βέλτιστης απόδοσης από ~3,2% σε ~1,3% σε ελληνικά σενάρια δρομολόγησης.

##### Υβριδική αποκωδικοποίηση με χρήση αναζήτησης δέσμης και διανυσματικοποιημένου 2-Opt

Συνδυάζουμε τη βελτίωση 2-OPT με επιτάχυνση GPU [13] με αναζήτηση δέσμης πλάτους max 50, ώστε ο μετασχηματιστής να αποφεύγει την προδιάθεσή του προς την απληστία στην αποκωδικοποίηση. Αυτή η υβριδική μετα-αποκωδικοποίηση βελτιώνει την ποιότητα της διαδρομής κατά 0,4% με μόλις 30% αύξηση του χρόνου συμπερασμού.

##### Οι μετασχηματισμοί Flying GTSP προσφέρουν συνεπή αναπαράσταση γραφημάτων.

Ενσωματώνουμε τη μετάφραση Noon-Bean [1] στο πρόγραμμα φόρτωσης δεδομένων για να επιτρέψουμε τη συνεπή μορφή ATSP μέσω της ομαλής μετατροπής των περιπτώσεων GTSP και CTSP. Λειτουργώντας στα ίδια γραφήματα, το Transformer και οι ευρετικοί επιλυτές (NN, GA, ACO, BnB) παρέχουν δίκαιη και συνεπή συγκριτική αξιολόγηση, με το Branch-and-Bound να χρησιμοποιείται ως ακριβής μέθοδος αναφοράς για μικρά γραφήματα.

##### Διαδραστική διεπαφή για την οπτικοποίηση δεδομένων

Χρησιμοποιώντας έναν ναυτικό χάρτη της Ελλάδας, το προσαρμοσμένο GUI (με PyQt6) σε πραγματικό χρόνο επιτρέπει στους χρήστες να επιλέξουν το πλάτος δέσμης, το 2-OPT και να αξιολογήσουν τα αποτελέσματα δρομολόγησης. Το πρόγραμμα βασίζεται σε ένα σύγχρονο, επαναλήψιμο λογισμικό Python. Το GUI είναι κατασκευασμένο σε PyQt6, ενώ το matplotlib

χρησιμοποιείται για την απεικόνιση των πόλεων και των διαδρομών. Το Pandas πραγματοποιεί την ενσωμάτωση των δεδομένων με τη βάση δεδομένων SQLite, η οποία διατηρεί τα γεωγραφικά χαρακτηριστικά των πόλεων, καθώς και τη δυνατότητα επεξεργασίας των δεδομένων. Το αρχείο requirements.txt περιλαμβάνει όλες τις εξαρτήσεις, επιτρέποντας έτσι την αναδημιουργία του περιβάλλοντος σε οποιοδήποτε σύστημα με ελάχιστη προσπάθεια.

Η εφαρμογή παρέχει σε πραγματικό χρόνο συγκρίσεις μεταξύ του μοντέλου Transformer και των ευριστικών προσεγγίσεων (NN, GA, ACO, BnB), τόσο ως προς τη συνολική απόσταση όσο και τον χρόνο επίλυσης κάθε αλγορίθμου.

### **Δημόσιο φύλλο δεδομένων για τη δρομολόγηση στην ελληνική θάλασσα**

Με βάση τα δρομολόγια των φέριμποτ για το 2024 με εποχιακές διακυμάνσεις, παρουσιάζουμε το πρώτο δημόσια διαθέσιμο σύνολο δεδομένων για τη θαλάσσια δρομολόγηση στην Ελλάδα: 500 δοκιμαστικές περιπτώσεις που καλύπτουν 20 έως 120 κόμβους, μαζί με 50.000 γραφήματα εκπαίδευσης. Έτσι, υποστηρίζονται η διδασκαλία βάσει προγράμματος σπουδών και η επαναλαμβανόμενη έρευνα.

### **Σύνδεση πολυεδρικών και νευρωνικών προοπτικών**

Οι κλασικές τεχνικές LP παρακινούν τη λογική της κανονικοποίησης του κόστους, της κάλυψης και της περικοπής των άκρων του μοντέλου: Η εκμάθηση του αρχιτεκτονικού σχεδιασμού και η προσοχή στη λεπτομέρεια βοηθούν στη σταδιακή ενσωμάτωση αναλόγων comb-cut, περιορισμών Subtour Elimination Constraints (SEC) και ορίων 1-tree [14].

Ο πλήρης πηγαίος κώδικας του συστήματος, μαζί με τις ρουτίνες εκπαίδευσης, αποκωδικοποίησης και μετα-επεξεργασίας, είναι διαθέσιμος δημόσια στο GitLab: [https://gitlab.com/sdomenstud/tsp\\_greece](https://gitlab.com/sdomenstud/tsp_greece)

## **Σύνοψη**

Από τις πρώτες μορφές του έως τις τρέχουσες εξελίξεις μέσω τεχνικών μηχανικής μάθησης, αυτό το κεφάλαιο παρείχε το θεωρητικό υπόβαθρο του προβλήματος TSP. Η περίληψη των κλασικών, ευριστικών και μεταευριστικών μεθόδων τονίζει την ανάγκη για δημιουργικές υβριδικές προσεγγίσεις, οι οποίες θα καθοδηγήσουν την έρευνα και την εφαρμογή που θα αναπτυχθεί στα επόμενα κεφάλαια.

## Κεφάλαιο 2: Σχετική Βιβλιογραφία και Τεχνικές Επίλυσης TSP

### 2.1 Μεθοδολογίες περιφέρειας

Αν και η διατύπωσή του φαίνεται μάλλον απλή, το πρόβλημα του περιοδεύοντος πωλητή (TSP) ανήκει στην κατηγορία των NP-δύσκολων προβλημάτων. Καθώς αυξάνεται ο αριθμός των πόλεων, ο υπολογισμός της συντομότερης διαδρομής που επισκέπτεται κάθε μία ακριβώς μία φορά και επιστρέφει στο σημείο εκκίνησης γίνεται υπολογιστικά αδύνατος. Αναζητούνται ακριβείς λύσεις με πλήρεις μαθηματικές εγγυήσεις ανάλογα με τον σχεδιασμό τους. Για μικρές έως μεσαίες καταστάσεις, είναι απολύτως απαραίτητες και αποτελούν το σημείο αναφοράς με το οποίο συγκρίνονται όλες οι άλλες προσεγγίσεις, είτε ευριστικές είτε βασισμένες στη μάθηση.

#### 2.1.1 Μαθηματική Διατύπωση

Το πρόβλημα του Περιοδεύοντος Πωλητή μπορεί να διατυπωθεί μαθηματικά ως ένα πρόβλημα ελαχιστοποίησης με βάση τις αποστάσεις μεταξύ των πόλεων. Η παρακάτω εξίσωση αποτυπώνει τη βασική διατύπωση:

$$\text{Minimize: } C(\pi) = \sum_{i=1}^n d_{p_i, p_{i+1}} \text{ with } p_{n+1} = p_1$$

(2.1)

Η αντικειμενική συνάρτηση  $C(\pi)$  υπολογίζει το συνολικό κόστος της διαδρομής, με κάθε διαδοχικό ζεύγος πόλεων να προσθέτει απόσταση. Η υπόθεση  $p_{n+1} = p_1$  διασφαλίζει ότι η διαδρομή επιστρέφει στην αρχική πόλη.

#### 2.1.2 Ακριβείς Μέθοδοι Επίλυσης

##### ► Held–Karp Dynamic Programming

Μία από τις πρώτες ακριβείς μεθόδους για τη δυναμική προγραμματισμός TSP είναι η μέθοδος Held–Karp [14]. Χρησιμοποιώντας μια αναδρομική συνάρτηση κόστους, βρίσκει το ελάχιστο κόστος για την απόκτηση οποιουδήποτε υποσυνόλου πόλεων. Αν και μειώνει σημαντικά την πολυπλοκότητα της μεθόδου brute-force (από  $O(n!)$  σε  $O(n^2 \cdot 2^n)$ ), οι εκθετικές απαιτήσεις μνήμης και χρόνου περιορίζουν την εφαρμοσιμότητά της σε περίπου 20 πόλεις. Παρ' όλα αυτά, το κατώτατο όριο Held–Karp εξακολουθεί να εφαρμόζεται εκτενώς στην συνδυαστική αναζήτηση για την αξιολόγηση ευριστικών μεθόδων και στην αποκοπή κλάδων.

##### ► DFJ Integer Linear Programming (ILP) Formulation

Οι Dantzig, Fulkerson και Johnson (DFJ) [6] δημοσίευσαν μια ιστορική διατύπωση ILP του TSP το 1954. Κάθε άκρη στο γράφημα έχει μια δυαδική μεταβλητή που υποδηλώνει αν αποτελεί μέρος της

διαδρομής. Χρησιμοποιώντας εκθετικά πολλούς περιορισμούς γνωστούς ως Subtour Elimination Constraints (SECs), επιτυγχάνεται η εξάλειψη των υποδιαδρομών.

Αν και θεωρητικά η διατύπωση είναι συμπαγής, στην πράξη λύτες όπως το Concorde [15] χρησιμοποιούν τη μέθοδο branch-and-cut, προσθέτοντας δυναμικά παραβιασμένους SEC κατά τη διάρκεια της χαλάρωσης LP. Για μεγάλης κλίμακας εκδηλώσεις, όπως το World TSP με 24.978 πόλεις, αυτή η μέθοδος έχει επιτρέψει την εξεύρεση ακριβών λύσεων. Ωστόσο, τέτοιοι λύτες δεν είναι κατάλληλοι για συστήματα σε πραγματικό χρόνο, καθώς απαιτούν ισχυρό υλικό και μεγάλη μνήμη. Αναλυτική σύγκριση μεταξύ διαφορετικών γραμμικών διατυπώσεων, συμπεριλαμβανομένων των DFJ και MTZ, έχει παρουσιαστεί από τους Padberg και Sung, παρέχοντας πολυεδρική θεώρηση και θεωρητικά όρια.

► Branch-and-Bound

Η εξαντλητική εξέταση των χώρων λύσεων με παράλληλη αποκοπή των υποβέλτιστων κλάδων ανάλογα με τους περιορισμούς γίνεται εκτενώς με τη χρήση του παραδείγματος branch-and-bound. Συνήθως στο TSP, εξαλείφει μεγάλο μέρος του δέντρου αναζήτησης χρησιμοποιώντας το όριο Held-Karp [14]. Αυτή η προσέγγιση είναι λογική για μέτριου μεγέθους περιπτώσεις, ακόμη και αν η πολυπλοκότητά της παραμένει εκθετική.

Η παρούσα διατριβή παρουσιάζει έναν ελαφρύ επιλυτή branch-and-bound για την παραγωγή ακριβών λύσεων αναφοράς για μικρά προβλήματα (π.χ.  $\leq 15$  πόλεις). Κατά την αξιολόγηση της ακρίβειας του Transformer, αυτές λειτουργούν ως «ground truth».

## Σύνοψη

Αν και το μεγάλο υπολογιστικό κόστος τους περιορίζει σε μικρά γεγονότα, οι ακριβείς μέθοδοι παραμένουν απαραίτητες για τη συγκριτική αξιολόγηση και παρέχουν ισχυρές εγγυήσεις. Αυτό το έργο θέτει ένα χρυσό πρότυπο με το οποίο συγκρίνονται ο Transformer και οι ευριστικές μέθοδοι χρησιμοποιώντας ακριβείς λύτες τόσο για την αξιολόγηση της απόδοσης όσο και για την εποπτεία της εκπαίδευσης.

## 2.2 Κλασικές ευριστικές προσεγγίσεις

Ιδιαίτερα χρήσιμοι όταν οι ακριβείς μέθοδοι καθίστανται πρακτικά ανεφάρμοστες λόγω της συνδυαστικής έκρηξης του χώρου λύσεων, οι κλασικοί ευριστικοί αλγόριθμοι παρέχουν γρήγορες και απλές απαντήσεις στο πρόβλημα του περιοδεύοντος πωλητή (TSP). Αν και δεν εξασφαλίζουν βέλτιστα αποτελέσματα, το χαμηλό υπολογιστικό κόστος και η απλότητα χρήσης τους τα καθιστούν ιδανικά για γρήγορες προσεγγίσεις, αρχικές φάσεις και συγκριτικές συγκρίσεις, ιδιαίτερα για την επαλήθευση της απόδοσης μοντέλων που βασίζονται στη μάθηση, συμπεριλαμβανομένου του Transformer.

### Nearest Neighbor (NN)

## Κεφάλαιο 2

Μεταξύ των απλούστερων και πιο συχνά χρησιμοποιούμενων ευριστικών μεθόδων TSP είναι η μέθοδος του κοντινότερου γείτονα. Ξεκινώντας από μια τυχαία πόλη, επιλέγει επανειλημμένα την κοντινότερη πόλη που δεν έχει επισκεφθεί και την προσθέτει στη διαδρομή μέχρι να καλυφθούν όλες οι πόλεις.

Περίγραμμα αλγορίθμου:

1. Εκίνηση από μια πόλη  $v_0$ .
2. Ξεκινώντας την περιήγηση  $T=[v_0]$ , σημειώνεται  $v_0$  ως επισκεπτόμενη.  
Ενώ υπάρχουν πόλεις που δεν έχουν επισκεφθεί:
3. Επιλέξτε την πλησιέστερη πόλη που δεν έχει επισκεφθεί και αντιστοιχεί στο  $T \nsubseteq T$ .
4. Προσθέστε  $v$  στο  $T$ .
5. Επισκεφθείτε ξανά το  $v_0$  για να ολοκληρώσετε την περιήγηση.

Αυτή η μέθοδος εκτελείται σε χρόνο  $O(n^2)$ . Η απλότητά της έχει ένα κόστος: μπορεί να παράγει αποτελέσματα 25-50% χειρότερα από τα βέλτιστα για τυχαία διάσπαρτες ευκλείδειες περιπτώσεις [9]. Στις χειρότερες συνθήκες, ο λόγος προσέγγισής της είναι απεριόριστος. Ωστόσο, η βασική της αξία είναι αδιαμφισβήτητη.

Η επίλυση με τον αλγόριθμο (*nearest\_neighbor.py*) λειτουργεί ως μέθοδος αναφοράς στο σύστημα GUI αυτής της διατριβής για να δείξει τις διακυμάνσεις της απόδοσης σε σχέση με την εξαγωγή συμπερασμάτων με βάση το Transformer.

### Ο αλγόριθμος greedy

Κάθε κόμβος πρέπει να έχει βαθμό  $\leq 2$ , επομένως η ευριστική μέθοδος greedy επιλέγει πάντα την κοντύτερη ακμή που ικανοποιεί δύο περιορισμούς: δεν μπορούν να δημιουργηθούν πρόωροι κύκλοι πριν από την ολοκλήρωση ολόκληρης της περιήγησης. Αν και τροποποιημένη για περιορισμούς κύκλου, αυτή είναι ουσιαστικά η μέθοδος του Kruskal για ελάχιστα δέντρα κάλυψης.

Συχνά αποδίδει καλύτερα από το NN σε δομημένες καταστάσεις και εκτελείται σε  $O(n^2 \log n)$ . Ωστόσο, η εξάρτησή του από την επιλογή με βάση τις ακμές μπορεί να το οδηγήσει σε λάθος αποτελέσματα σε αραιά ή ασύμμετρα γραφήματα.

### Ευριστική εισαγωγή (π.χ. εισαγωγή του πιο απομακρυσμένου)

Η ευριστική εισαγωγή ενός κόμβου κάθε φορά επεκτείνει την περιήγηση. Μια συχνά χρησιμοποιούμενη τακτική είναι η εισαγωγή του πιο απομακρυσμένου κόμβου (FI):

1. Ξεκινήστε με μια μικρή υποπεριήγηση, για παράδειγμα τις δύο πιο απομακρυσμένες πόλεις.
2. Εισάγετε την πόλη που βρίσκεται πιο μακριά από την τρέχουσα διαδρομή σε κάθε στροφή.
3. Επιλέξτε το σημείο εισαγωγής με τη μικρότερη αύξηση του μήκους της διαδρομής.

Η FI συνήθως υπερέρχει της NN, ιδιαίτερα σε μεγαλύτερους ευκλείδειους γραφήματα [9], με μέσους όρους περίπου 10% πάνω από το βέλτιστο.

**Αλγόριθμος Christofides: Προσέγγιση μετρικού TSP**

Η μέθοδος Christofides προσφέρει μια θεωρητική εγγύηση: παράγει μια λύση εντός του 1,5 επί του βέλτιστου για μετρικά TSP (όπου ισχύει η ανισότητα του τριγώνου). Λειτουργεί μέσω:

1. Εντοπίστε ένα ελάχιστο δέντρο κάλυψης (MST).
2. Η συμπερίληψη μιας αντιστοίχισης ελάχιστου βάρους θα βοηθήσει στην εξισορρόπηση των βαθμών των κόμβων.
3. Λήψη μιας διαδρομής με διαδρομή Euler και συντομεύσεις

Ο Christofides [7] παραμένει κρίσιμος για θεωρητικές συγκρίσεις, ακόμη και αν είναι πιο αργός ( $O(n^3)$ ) και πιο περίπλοκος. Ο ασύμμετρος χαρακτήρας του προβλήματος δρομολόγησης καθιστά αδύνατη την άμεση εφαρμογή του σε αυτή τη διατριβή.

**Σύνοψη**

Οι κλασικές ευριστικές μέθοδοι βρίσκουν ένα μείγμα μεταξύ ταχύτητας και απλότητας. Αν και η καθοριστική συμπεριφορά, ο γρήγορος χρόνος εκτέλεσης και τα καλά κατανοητά χαρακτηριστικά απόδοσης τις καθιστούν πολύτιμες για συγκρίσεις βάσης, ειδικά σε εφαρμογές GUI και σε καταστάσεις αξιολόγησης μοντέλων, ακόμη και αν δεν είναι τέλειες. Αυτή η εργασία συγκρίνει τις τεχνικές συμπερασμού και μετα-επεξεργασίας Transformer χρησιμοποιώντας NN και Greedy ως λύτες αναφοράς. Οι βασικοί αλγόριθμοι που εφαρμόζονται στη διατριβή συνοψίζονται στον Πίνακα 2.

**Πίνακας 2.1 -Σύγκριση των αλγορίθμων επίλυσης του TSP ως προς τύπο, πολυπλοκότητα και γεωγραφική εφαρμογή.**

Αλγόριθμος	Τύπος	Πολυπλοκότητα	Υποστήριξη Γεωγραφίας
Brute-force	Ακριβής	$O(n!)$	✓
Nearest Neighbor	Ευριστικός	$O(n^2)$	✓
Greedy	Ευριστικός	$O(n^2 \log n)$	✓
2-OPT	Βελτιωτικός	$O(n^2)$ ανά βήμα	✓
Transformer (RL)	Εκμάθηση	$\sim O(N \times T)$	✓✓

**2.3 Μεταευριστικά εργαλεία**

Αν και οι κλασικές ευριστικές μέθοδοι προσφέρουν ταχύτητα, συχνά παγιδεύονται σε τοπικά βέλτιστα ιδίως σε μεγάλες ή ασύμμετρες περιπτώσεις του προβλήματος του περιοδεύοντος πωλητή (TSP). Οι μεταευριστικές προσεγγίσεις αναπτύχθηκαν ώστε να συνδυάζουν **εξερεύνηση** (αποφυγή παγίδευσης)

## Κεφάλαιο 2

και **εκμετάλλευση** (βελτίωση υπαρχουσών λύσεων). Χωρίς να εγγυώνται βέλτιστες λύσεις, προσφέρουν **ποιοτικά αποτελέσματα σε λογικούς χρόνους**.

Στο πλαίσιο αυτής της διατριβής, αξιοποιήθηκαν και συγκρίθηκαν οι εξής μεταεвриστικές μέθοδοι:

### **Genetics Algorithms (GA)**

Εμπνευσμένοι από τη βιολογική εξέλιξη [5], οι γενετικοί αλγόριθμοι διατηρούν έναν πληθυσμό λύσεων που εξελίσσονται από γενιά σε γενιά μέσω διασταύρωσης και μετάλλαξης.

Συνήθως, η διαδικασία GA:

1. Ξεκινά έναν πληθυσμό τυχαίων περιηγήσεων
2. Αναλύει την καταλληλότητα ή το αντίστροφο μήκος της περιήγησης.
3. Επιλέγει γονείς (π.χ. ρουλέτα ή επιλογή τουρνουά).
4. Εφαρμόζει διασταύρωση, π.χ. PMX ή διασταύρωση σειράς.
5. Μεταλλάσσει τα παιδιά (ανταλλαγές πόλεων, αντιστροφές τμημάτων).
6. Συνεχίζει για έναν καθορισμένο αριθμό γενεών.

Τα GA είναι φυσικά παραλληλοποιήσιμα και καλύπτουν μεγάλες περιοχές αναζήτησης. Η παρούσα διατριβή αξιολογεί την ανταγωνιστικότητα του Transformer μέσω GA με βελτίωση 2-OPT σε πειράματα.

### **Ant Colony Optimization(ACO)**

Ο ACO μοντελοποιεί τη δραστηριότητα αναζήτησης τροφής των μυρμηγκιών [4] με την καθοδήγηση των φερομονικών ιχνών:

1. Τα τεχνητά μυρμηγκία κατασκευάζουν διαδρομές ανάλογα με την ένταση των φερομονών και την ορατότητα (π.χ. αντίστροφη απόσταση) σταδιακά.
2. Οι φερομόνες ενισχύονται σε κατάλληλες διαδρομές και εξατμίζονται σε λιγότερο ιδανικές μετά από μια διαδρομή.
3. Η αποικία συγκλίνει σταδιακά σε ενδιαφέρουσες περιοχές του χώρου λύσεων.

Αυτή η εργασία ενσωματώνει το ACO ως σημείο αναφοράς απόδοσης με έναν τοπικό μετα-επεξεργαστή 2-OPT για καλύτερες λύσεις [13].

### **Σύνοψη**

Οι GA και ACO εφαρμόστηκαν πλήρως και συγκρίθηκαν με το Transformer ως **ρεαλιστικές και προσαρμόσιμες μεταεвриστικές προσεγγίσεις**. Οι SA και LKH αναφέρονται ως σημαντικά θεωρητικά θεμέλια, αλλά δεν εντάχθηκαν ενεργά στο σύστημα. Το 2-OPT αξιοποιείται σε όλους τους αλγόριθμους ως μετα-επεξεργαστής.

## 2.4 Προσεγγίσεις βασισμένες στη μάθηση

Η βαθιά μάθηση γίνεται μια νέα προσέγγιση για την αντιμετώπιση προβλημάτων συνδυαστικής βελτιστοποίησης, συμπεριλαμβανομένου του προβλήματος του περιοδεύοντος πωλητή (TSP). Σε αντίθεση με τις κλασικές μεθόδους που βασίζονται σε χειρόγραφους κανόνες ή τακτικές αναζήτησης, τα μοντέλα που βασίζονται στη μάθηση συνάγουν πολιτικές δρομολόγησης απευθείας από τα δεδομένα.

Κατά τη μοντελοποίηση των συντεταγμένων της πόλης και των προβλέψεων ακολουθιών χρησιμοποιώντας επαναλαμβανόμενες αρχιτεκτονικές ή κωδικοποιητές συνελίξεων, τα πρώτα νευρωνικά μοντέλα αντιμετώπιζαν προβλήματα με τα μεταβλητά μεγέθη εισόδου, την ευαισθησία στις μεταθέσεις και την επεκτασιμότητα των συμπερασμάτων. Τα δίκτυα δείκτη (Pointer Networks) αντιμετώπισαν αυτούς τους περιορισμούς δημιουργώντας μια τεχνική άμεσης ευρετηρίασης που επιτρέπει στο μοντέλο να δείχνει τα στοιχεία [16].

Ωστόσο, λόγω του διαδοχικού χαρακτήρα τους, τα RNN περιόριζαν τον παραλληλισμό και την εκφραστικότητα. Τα Transformers άλλαξαν αυτόν τον τομέα αξιοποιώντας pipelines κωδικοποίησης-αποκωδικοποίησης βασισμένα στην προσοχή, ώστε να θεωρούν τις εισόδους TSP ως σύνολα και όχι ως ακολουθίες και να χειρίζονται όλες τις πόλεις ταυτόχρονα [17].

Αυτά τα μοντέλα ιδιαίτερα αυτά που παρέχονται από τους Kool και της ομάδος του [18] και διευρύνθηκαν από τους Joshi και της ομάδος του (2020) παρουσίασαν σχεδόν βέλτιστη απόδοση τόσο σε συμμετρικές όσο και σε ασύμμετρες εκδόσεις του TSP. Χρησιμοποίησαν εκπαίδευση σε προγράμματα σπουδών, βελτιστοποίηση πολιτικής (*Reinforce*), μέθοδοι στοχαστικής αποκωδικοποίησης, συμπεριλαμβανομένης της αναζήτησης δέσμης [10][11].

Η διατριβή μας αναπτύσσεται πάνω σε αυτά τα θεμέλια χρησιμοποιώντας επικύρωση μετά την αποκωδικοποίηση, λογική ρεαλισμού και πρακτικούς περιορισμούς όπως η σύνδεση ελληνικών νησιών με λιμάνια. Σε αντίθεση με προηγούμενες προσπάθειες που εκτελούνται σε άπειρους γραφήματα, τροποποιούμε τα μοντέλα Transformer για να διατηρήσουμε την αρχιτεκτονική modularity και την ταχύτητα συμπερασμού χρησιμοποιώντας αραιά, ημι-συνδεδεμένα, γεωγραφικά πλαίσια [20]. Η εκπαίδευση ενσωματώνει το σχήμα REINFORCE με log-prob ενισχύσεις, αντί του παραδοσιακού actor-critic πλαισίου του Kool et al. [18].

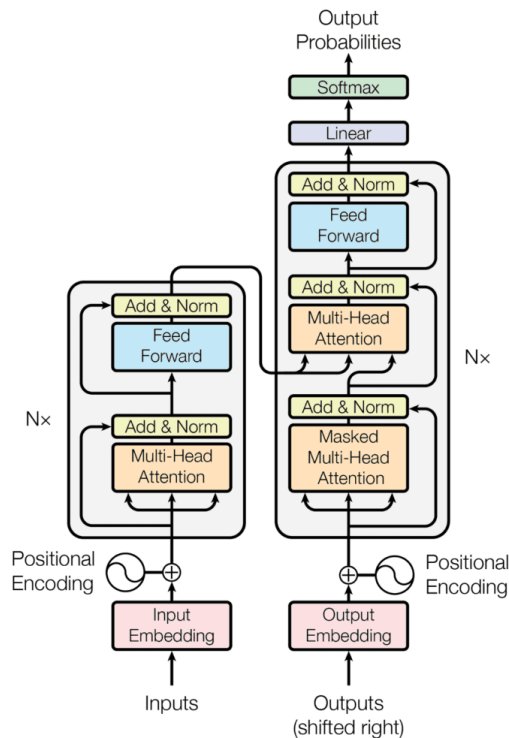
Η ενότητα 2.5 ασχολείται με τις διαδικασίες αρχιτεκτονικής, εκπαίδευσης και υβριδικού συμπερασμού, προετοιμάζοντας έτσι το έδαφος για τις συγκεκριμένες μεθοδολογίες βασισμένες στο Transformer που παρουσιάζονται σε αυτή την ενότητα.

## 2.5 TSP Transformer-Based Techniques

Αρχικά παρουσιάστηκε για εργασίες φυσικής γλώσσας, η αρχιτεκτονική Transformer έγινε γρήγορα δημοφιλής στην συνδυαστική βελτιστοποίηση, επειδή μπορεί να μοντελοποιήσει δομημένες

## Κεφάλαιο 2

διαδικασίες λήψης αποφάσεων χωρίς επαναλήψεις ή αναζήτηση. Τα μοντέλα που βασίζονται στον μετασχηματιστή επιδιώκουν να μάθουν στρατηγικές δρομολόγησης απευθείας από τα δεδομένα, χρησιμοποιώντας μηχανισμούς προσοχής για να καταγράψουν το συνολικό πλαίσιο μιας περίπτωσης προβλήματος, σε αντίθεση με τους συμβατικούς επιλυτές που βασίζονται σε χειροποίητη λογική ή τυχαία ευριστική μέθοδο.



Σχήμα 2.1 - Τυπική αρχιτεκτονική του μοντέλου *Transformer*, όπως παρουσιάστηκε από τους Vaswani et al. (2017). Πηγή: <https://jalammr.github.io/illustrated-transformer/>

Σε αυτή τη διατριβή, αναπτύσσουμε ένα Transformer σχεδιασμένο πάνω στην μορφολογία της Ελλάδας, επεκτείνοντας έτσι την προσπάθεια των Kool και της ομάδας του [18],[19]. Μέσα σε μια πλήρως παραλληλοποιήσιμη νευρωνική αρχιτεκτονική, το μοντέλο μας συνδυάζει εκπαίδευση με βάση το πρόγραμμα σπουδών, υβριδική συμπερασματολογία και πραγματικούς χωρικούς περιορισμούς, όπως ομαδοποιημένα νησιά και ασύμμετρες αποστάσεις.

### 2.5.1 Γιατί Transformers; Μια αλλαγή στον συνδυαστικό συλλογισμό

Τα μοντέλα που βασίζονται σε μετασχηματιστές (transformers) αποτελούν μια σημαντική απομάκρυνση από τις ευριστικές ή συμβατικές προσεγγίσεις TSP που εξαρτώνται από ρητή αναζήτηση ή προσαρμοσμένη λογική. Οι μετασχηματιστές μπορούν να μοντελοποιήσουν παγκόσμιες αλληλεπιδράσεις με έναν εντελώς παράλληλο και τακτοποιημένο τρόπο, μόλις εκπαιδευτούν, προσφέροντας έτσι σχεδόν τέλειες λύσεις χωρίς εκτενή απαρτίθμηση.

Τα κύρια πλεονεκτήματα είναι τα εξής:

- Για προβλήματα που βασίζονται σε σύνολα, όπως το TSP, η αμετάβλητη μετατόπιση, δηλαδή η ακολουθία των εισόδων των πόλεων, δεν έχει καμία επίδραση στην ερμηνεία του μοντέλου
- Τα μοντέλα μετασχηματιστών επιτρέπουν τη γενίκευση των μεγεθών των περιπτώσεων, καθώς προσαρμόζονται φυσικά σε διαφορετικούς αριθμούς κόμβων: εισόδους μεταβαλλόμενου μεγέθους
- Η αυτοπροσοχή καταγράφει τις μακροπρόθεσμες εξαρτήσεις και επιτρέπει σε κάθε πόλη να αιτιολογεί τις αποφάσεις της έναντι όλων των άλλων.
- Η εκμάθηση κατάλληλων προγραμμάτων σπουδών επιτρέπει την πρόοδο από μικρούς σε μεγάλους γραφήματα, βελτιώνοντας έτσι τη γενίκευση.
- Οι αγωγοί μετασχηματιστών μπορούν να ενσωματωθούν με κομψό τρόπο σε γραφικά εργαλεία

Αυτά τα χαρακτηριστικά βοηθούν την αρχιτεκτονική Transformer να αποτελεί ένα ισχυρό και προσαρμόσιμο πλαίσιο για τη διαχείριση πρακτικών προβλημάτων δρομολόγησης, συμπεριλαμβανομένων εκείνων που ενσωματώνουν επιπλέον γεωγραφικούς περιορισμούς ή ημι-συνδεδεμένους γραφή

### 2.5.2 Architecture: Encoder–Decoder Attention Pipeline

Χαρτογραφούμε ένα σύνολο συντεταγμένων πόλεων σε 2D σε μια πλήρη περιήγηση χρησιμοποιώντας μια τροποποιημένη αρχιτεκτονική Transformer κωδικοποιητή-αποκωδικοποιητή. Μέσω γραμμικών προβολών, ο κωδικοποιητής μετατρέπει την χωρική είσοδο κάθε πόλης σε λανθάνουσες ενσωματώσεις, καταγράφοντας έτσι τη σχετική θέση και τις αποστάσεις. Τα πολυκεφαλικά στρώματα αυτοπροσοχής επιτρέπουν στο μοντέλο να μαθαίνει τις τοπικές αλλά και τις παγκόσμιες αλληλεπιδράσεις μεταξύ των κόμβων.

Εμπνευσμένος από τα δίκτυα Pointer, ο αποκωδικοποιητής παρακολουθεί τις εξόδους του κωδικοποιητή και επιλέγει προοδευτικά την επόμενη πόλη ανάλογα με τις προηγούμενες ενέργειες αποκωδικοποίησης [16]. Σε αντίθεση με τους κλασικούς μετασχηματιστές στο NLP, ο αποκωδικοποιητής μας έχει σχεδιαστεί για συνδυαστική έξοδο [21], εξασφαλίζοντας έτσι ότι κάθε βήμα αντανακλά ένα έγκυρο πρόθεμα περιήγησης.

Αν και το μοντέλο μας κωδικοποιεί απευθείας τη χωρική δομή στον λανθάνοντα χώρο, δεν επιβάλλει συγκεκριμένους περιορισμούς, όπως η σκοπιμότητα της περιήγησης ή η επίσκεψη σε λιμάνια.

## Κεφάλαιο 2

Αντίθετα, η καθοδηγούμενη κάλυψη και η εξωτερική επικύρωση βοηθούν στη διαχείριση αυτών κατά την αποκωδικοποίηση και την μετα-επεξεργασία. Ο αποκωδικοποιητής χρησιμοποιεί λογική σκοπιμότητας π.χ. σύζευξη νησιού-λιμανιού μετά τη δημιουργία και εγγυάται ότι κάθε πόλη επισκέπτεται μόνο μία φορά.

Χρησιμοποιώντας την αποτελεσματικότητα και την επεκτασιμότητα της σκέψης που βασίζεται στην προσοχή, αυτή η αρχιτεκτονική επιτρέπει την προσαρμογή ενός αρθρωτού συστήματος σε νέους κανόνες και περιορισμούς.

### 2.5.3 Training Procedure: From Supervision to Reinforcement

Στο πλαίσιο ενός προσομοιωμένου προγράμματος σπουδών, η εκπαιδευτική προσέγγιση συνδυάζει την ενισχυτική μάθηση με την εποπτευόμενη μάθηση. Αρχικά, το μοντέλο διδάσκεται με εποπτευόμενα σήματα που προέρχονται από βέλτιστες ή σχεδόν βέλτιστες διαδρομές που παράγονται από λύτες όπως το Concorde ή ευριστικές βασικές γραμμές. Αυτό διδάσκει βασικά συνδυαστικά μοτίβα και βοηθά στη σταθεροποίηση του αποκωδικοποιητή.

Αργότερα, χρησιμοποιείται η ενίσχυση της μάθησης με κλίση πολιτικής πιο συγκεκριμένα, το *REINFORCE* [11] όπου το μήκος της διαδρομής θεωρείται αρνητικό κίνητρο. Αυτό βοηθά το μοντέλο να προσαρμόσει τη στρατηγική δρομολόγησης ανάλογα με την ποιότητα της λύσης, ξεπερνώντας τα όρια της απλής παρακολούθησης.

Η υλοποίηση της φάσης ενίσχυσης με χρήση του κανόνα *REINFORCE* και της αντίστοιχης ρουτίνας fine-tuning παρατίθεται στο Παράρτημα B.3.

Η μάθηση με πρόγραμμα σπουδών [10] είναι απαραίτητη: η εκπαίδευση ξεκινά με μικρούς, συμμετρικούς γραφήματα (π.χ. πέντε πόλεις) και στη συνέχεια προχωρά σταδιακά σε μεγαλύτερα, ασύμμετρα γραφήματα με έως και 120 κόμβους. Αυτή η σκόπιμη πολυπλοκότητα εγγυάται αργή γενίκευση και συνεπή μάθηση.

Μεταξύ των περαιτέρω τεχνικών εκπαίδευσης είναι η μετάθεση εισόδου, η αύξηση της διακοπής και η κανονικοποίηση της εντροπίας, οι οποίες βοηθούν στην πρόληψη της υπερπροσαρμογής και ενισχύουν την ανθεκτικότητα. Μέσω μιας μικτής απώλειας: διασταυρούμενη εντροπία για τις εποπτευόμενες φάσεις και απώλεια ανταμοιβής κανονικοποιημένη ως βάση αναφοράς για τα στάδια RL, το μοντέλο βελτιστοποιείται εξισορροπώντας την εξερεύνηση με τη σύγκλιση

### 2.5.4 Hybrid Inference: Greedy, Beam Search, 2-OPT

Το μοντέλο χρησιμοποιεί μια υβριδική τεχνική αποκωδικοποίησης που εξισορροπεί την αποδοτικότητα και την ποιότητα κατά τη διάρκεια της συμπεραματολογίας. Με βάση την υψηλότερη πιθανότητα logit, ένας αποκωδικοποιητής greedy επιλέγει την επόμενη πόλη σε κάθε στροφή. Η αναζήτηση δέσμης, από την άλλη πλευρά, διατηρεί αρκετές υποψήφιες ακολουθίες, ενισχύοντας έτσι την ανθεκτικότητα σε πιο σύνθετες ή ακανόνιστες δομές γραφημάτων.

Μια ευριστική μέθοδος 2-OPT [13],[24] εφαρμόζεται προαιρετικά μετά την αποκωδικοποίηση για να βελτιώσει τοπικά την περιήγηση για ακόμη μεγαλύτερη βελτιστοποίηση. Αυτό το στάδιο βελτιώνει δραστικά τα αποτελέσματα που βασίζονται σε άπληστο ή δέσμη, ανταλλάσσοντας ζεύγη ακμών μέχρι τη σύγκλιση, μειώνοντας έτσι το συνολικό μήκος της διαδρομής. Σε γεωγραφικά περίπλοκες περιοχές όπως το ελληνικό αρχιπέλαγος, όπου οι άπληστες διαδρομές μπορεί να συναντηθούν νωρίς, αυτό είναι ιδιαίτερα σημαντικό.

Το μοντέλο δεν επιβάλλει εσωτερικά περιορισμούς συγκεκριμένων προβλημάτων [22], παρόλο που ανιχνεύει τη χωρική δομή. Οι κανόνες που περιλαμβάνουν την απαιτούμενη συμπερίληψη λιμένων ή τη διανοητική σκοπιμότητα αντιμετωπίζονται κατά την αποκωδικοποίηση ή την μετα-επεξεργασία χρησιμοποιώντας διορθωτική λογική.

Επιπλέον, στο γραφικό περιβάλλον χρήστη σε πραγματικό χρόνο που συζητείται στο Κεφάλαιο 6 περιλαμβάνεται ο Μετασχηματιστής. Η μορφή εξόδου του ταιριάζει με τους συμβατικούς επιλυτές, επιτρέποντας την άμεση οπτική σύγκριση και σύγκριση της απόδοσης μεταξύ των τεχνικών συμπερασμού. Αυτά τα χαρακτηριστικά υπογραμμίζουν την ευελιξία του υβριδικού σχεδιασμού, καθώς χρησιμοποιούνται σε ρεαλιστικές καταστάσεις που καλύπτουν τη δρομολόγηση στην ηπειρω

### 2.5.5 Στρατηγικές αποκωδικοποίησης: Υβριδική συμπερασματολογία για δρομολόγηση

Το μοντέλο Transformer δημιουργεί μια κατανομή πιθανότητας για την επόμενη πόλη που θα επισκεφθεί σε κάθε στροφή του δρόμου, μόλις εκπαιδευτεί. Η ποιότητα της τελικής διαδρομής επηρεάζεται σε μεγάλο βαθμό από την τεχνική αποκωδικοποίησης που εφαρμόζεται για την ερμηνεία αυτής της κατανομής. Σε αυτό το μέρος περιγράφονται οι τεχνικές αποκωδικοποίησης που εφαρμόζονται στην παρούσα διατριβή για την ανάκτηση ανταγωνιστικών και έγκυρων διαδρομών.

#### **Greedy Decode**

Σε κάθε στροφή, ο άπληστος αποκωδικοποιητής επιλέγει την πόλη με την υψηλότερη πιθανότητα. Η συγκριτική αξιολόγηση του χρόνου συμπερασμού θα ήταν κατάλληλη για αυτή τη γρήγορη και ντετερμινιστική μέθοδο. Ωστόσο, είναι ευαίσθητη σε τοπικές προκαταλήψεις και, ιδίως σε συσσωματωμένους ή ασύμμετρους γραφήματα, συχνά παράγει μακρύτερες διαδρομές.

#### **Αναζήτηση δέσμης (Beam search)**

Η αναζήτηση δέσμης επεκτείνει τις κορυφαίες  $k$  μερικές ακολουθίες σε κάθε βήμα αποκωδικοποίησης μέσω των  $b$  πιο πιθανών συνεχειών. Η αναζήτηση δέσμης μεγιστοποιεί την εξερεύνηση, ενώ παραμένει εύχρηστη, ακολουθώντας τους υποψηφίους με την καλύτερη βαθμολογία και επιλέγοντας τη διαδρομή με το χαμηλότερο αναμενόμενο κόστος. Τα πλάτη δέσμης  $k = 5$  ή  $10$  υπερέιχαν συστηματικά της άπληστης αποκωδικοποίησης με ελάχιστο υπολογιστικό κόστος στις αξιολογήσεις μας. Η διαδικασία αποκωδικοποίησης αξιοποιεί annealing της θερμοκρασίας δείγματος από  $temp=0.8$  σε  $0.4$ , όπως ορίζεται στο αρχείο `hyperparams.py`. (βλ. Παράρτημα Β)

#### **Στοχαστική δειγματοληψία**

## Κεφάλαιο 2

Η στοχαστική αποκωδικοποίηση επιτρέπει στο μοντέλο να δειγματοληπτεί πόλεις ανάλογα με τις πιθανότητες τους, αντί να επιλέγει τον καλύτερο υποψήφιο με μια ντετερμινιστική προσέγγιση. Αυτό ενθαρρύνει την ποικιλομορφία και βοηθά ιδιαίτερα στη δημιουργία πολλών υποψήφιων διαδρομών.

### **2-Opt Μετα-επεξεργασία**

Μετά την πλήρη αποκωδικοποίηση της περιήγησης χρησιμοποιώντας είτε την άπληστη αναζήτηση είτε την αναζήτηση δέσμης εφαρμόζουμε προαιρετικά ένα βήμα βελτίωσης 2-OPT. Αυτή η ευριστική μέθοδος διαγράφει δύο άκρα και τα επανασυνδέει με τρόπο που συντομεύει την περιήγηση.

Συνεχίζεται έως ότου δεν είναι δυνατή περαιτέρω πρόοδος. Οι δοκιμές μας έδειξαν ότι το 2-OPT βελτίωσε το κόστος της διαδρομής κατά  $\sim 0,4-1,2\%$  σε σχέση με τα αποτελέσματα του ακατέργαστου μοντέλου.

### **Στρατηγική για υβριδική αποκωδικοποίηση**

Η παρούσα διατριβή χρησιμοποιεί τη βελτίωση 2-OPT και την αναζήτηση δέσμης σε μια συνδυασμένη προσέγγιση αποκωδικοποίησης. Αυτή η υβριδική προσέγγιση διατηρεί την ταχύτητα συμπερασμού, ενώ αυξάνει την ποιότητα των λύσεων. Βλ. κεφάλαιο 3 σχετικά με τη λογική περιορισμών λιμένων και τη διόρθωση συγκεκριμένων τομέων.

Εμπνευσμένο από την ευριστική μέθοδο Lin-Kernighan (LKH) [8], όπου μια φάση τοπικής βελτιστοποίησης ακολουθεί μια φάση κατασκευαστική, ο σχεδιασμός αυτός αντανακλά ιδέες από Το σύστημα αποκωδικοποίησης γενικεύεται αποτελεσματικά σε πολλά περιβάλλοντα δρομολόγησης, συνδυάζοντας την ελαφριά συνδυαστική αναζήτηση με πολιτικές που έχουν εκπαιδευτεί σε παγκόσμιο επίπεδο. Η αποκωδικοποίηση δεν είναι επομένως μια παθητική, αλλά μάλλον μια ενεργή σχεδιαστική απόφαση που επηρεάζει άμεσα την απόδοση του χρόνου εκτέλεσης και την ποιότητα της λύσης.

### **2.5.6 GUI Routing Constraints Integration**

Σε αυτό το μέρος εξετάζεται η αλληλεπίδραση του μοντέλου Transformer με τη λογική περιορισμών που βασίζεται σε GUI σε εφαρμογές πραγματικού χρόνου. Αν και η βασική αρχιτεκτονική του μοντέλου παραμένει γενικής χρήσης, η σκοπιμότητα για συγκεκριμένους τομείς όπως οι απαιτούμενες συνδέσεις θυρών για νησιά επιβάλλεται εξωτερικά χρησιμοποιώντας καθοδηγούμενη αποκωδικοποίηση και λογική επικύρωσης.

Τα δυναμικά συστήματα κάλυψης και απόρριψης βάσει κανόνων αποτελούν μέρος του συστήματος αποκωδικοποίησης. Για παράδειγμα, οι μάσκες εισόδου βοηθούν να επικεντρωθεί κανείς στη θύρα που αντιστοιχεί σε ένα νησί που έχει επισκεφθεί. Τα γεωγραφικά και λειτουργικά κριτήρια που έχουν οριστεί στο υποκείμενο μοντέλο δεδομένων βοηθούν στο φιλτράρισμα των μη έγκυρων μεταβάσεων.

Ο μετασχηματιστής δεν μαθαίνει τίποτα για αυτούς τους περιορισμούς. Αντίθετα, η αρχιτεκτονική του συστήματος έχει ως στόχο να τους επιβάλλει κατά την αποκωδικοποίηση. Συνδεδεμένος με το GUI (που καλύπτεται στο Κεφάλαιο 4), ο μηχανισμός δρομολόγησης βλέπει την αλλαγή της διαδρομής και

αντιδρά δυναμικά στις αλλαγές, συμπεριλαμβανομένων των τροποποιημένων περιορισμών ή των εκ νέου επιλεγμένων πόλεων.

Χωρίς επανεκπαίδευση για νέους κανόνες, αυτή η αρθρωτή προσέγγιση εγγυάται ότι το μοντέλο μπορεί να γενικευτεί σε νέα περιβάλλοντα, διατηρώντας παράλληλα τη συμβατότητα με τους πραγματικούς περιορισμούς της εφο

### **2.5.7 Προκλήσεις, γενίκευση και πλεονεκτήματα**

Ο σχεδιασμός με γνώμονα την αρθρωτότητα, την επεκτασιμότητα και την προσαρμοστικότητα σε συγκεκριμένους τομείς καθοδήγησε την αρχιτεκτονική που περιγράφεται λεπτομερώς σε αυτό το κεφάλαιο. Το σύστημα μεταφέρει την επιβολή της σκοπιμότητας σε καθοδηγούμενη αποκωδικοποίηση, κάλυψη και λογική εξωτερικής διόρθωσης, αντί να δημιουργεί ένα εξειδικευμένο μοντέλο για κάθε τύπο περιορισμού. Αυτό υποστηρίζει πραγματικούς περιορισμούς, όπως απαιτήσεις νησιωτικών λιμένων και γεωγραφική αραιότητα, διατηρώντας έτσι τον γενικό χαρακτήρα του Transformer.

Λόγω του σχεδιασμού του που βασίζεται στην προσοχή και της μεθόδου εκπαίδευσης του προγράμματος σπουδών, το μοντέλο γενικεύεται αποτελεσματικά σε πολλά μεγέθη και δομές γραφημάτων. Αποφεύγει την κωδικοποίηση των κανόνων logistics και καταγράφει τις δομικές συνδέσεις μεταξύ των πόλεων, επιτρέποντας έτσι ευελιξία σε απρόβλεπτες συνθήκες.

Πρακτικά, το σύστημα εφαρμόζει δυναμικά ελέγχους σκοπιμότητας και συνδέεται με ένα GUI σε πραγματικό χρόνο που απεικονίζει τις αποφάσεις δρομολόγησης. Το Transformer δημιουργεί περιηγήσεις, ευριστικές μέθοδοι όπως το 2-OPT βελτιώνουν την ποιότητα της διαδρομής εκ των υστέρων, ενώ οι περιορισμοί επιβάλλονται μέσω ελεγχόμενης αποκωδικοποίησης αντί για εσωτερικά βάρη προσοχής.

Ο συνδυασμός της δημιουργίας πολιτικών βασισμένων στη μάθηση με την κλασική λογική αναζήτησης σε αυτόν τον υβριδικό σχεδιασμό αντικατοπτρίζει μια ευρύτερη αλλαγή στη νευρωνική συνδυαστική βελτιστοποίηση: διατήρηση της ταχύτητας συμπερασμού, ενώ παράλληλα εγγυάται τη σκοπιμότητα της λύσης και την ετοιμότητα για εφαρμογή. Το τελικό αποτέλεσμα είναι μια θεωρητικά ορθή και πρακτικά χρήσιμη αρχιτεκτονική για τομείς όπως η εγχώρια εφοδιαστική και όχι μόνο.

## **Σύνοψη**

Σύμφωνα με τη βιβλιογραφική ανασκόπηση, για δεκαετίες το TSP αποτελεί βασικό σημείο αναφοράς για μελέτες στον τομέα της τεχνητής νοημοσύνης και της βελτιστοποίησης. Παρουσιάστηκαν συμβατικές τεχνικές, ευριστικές και ενισχυτικές μέθοδοι, καθώς και σύγχρονες προσεγγίσεις βασισμένες σε μετασχηματιστές (Transformers). Τα αποτελέσματα αυτά καθοδηγούν την προσαρμογή του προβλήματος στο γεωγραφικό περιβάλλον της Ελλάδας, όπως εξετάζεται στο επόμενο κεφάλαιο.

## Κεφάλαιο 3: Περιορισμοί και Διαμόρφωση του Προβλήματος στον Ελληνικό Χώρο

### 3.1 Πεδίο εφαρμογής του προβλήματος και περιορισμοί στην Ελλάδα

#### 3.1.1 Κίνητρα για την επιλογή της Ελλάδας

Εκπαίδευση των περισσότερων λύσεων βασισμένων στη μάθηση για το πρόβλημα του περιοδεύοντος πωλητή (TSP) σε συνθετικές ευκλείδειες περιπτώσεις ομοιόμορφα δειγματοληπτικές πόλεις [18] σε ένα επίπεδο 2D  $[0,1]$ . Αυτό το παράδειγμα αγνοεί την πραγματική γεωγραφική και υποδομική πολυπλοκότητα, παρόλο που βοηθά στην οργάνωση των δεδομένων εισόδου και των αναπαραγωγικών συνθηκών εκπαίδευσης. Η παρούσα διατριβή προτείνει μια εναλλακτική λύση: εκπαίδευση και αξιολόγηση σε παραδείγματα δρομολόγησης εμπνευσμένα από την ποικιλόμορφη και περιορισμένη γεωγραφία της Ελλάδας.

**Η Ελλάδα προσφέρει ένα ιδιαίτερα πλούσιο πεδίο δοκιμών λόγω των εξής χαρακτηριστικών της:**

- Οι απόκρημνες χερσόνησοι και οι εκατοντάδες κατοικημένες νησίδες συνθέτουν ένα κατακερματισμένο έδαφος που συχνά απαιτεί τη χρήση πορθμών για τη σύνδεσή του.
- Πόλεις όπως η Ρόδος ή η Χίος είναι προσβάσιμες μόνο μέσω συγκεκριμένων λιμανιών, επομένως οι απεριόριστες μετακινήσεις είναι αδύνατες.
- Πολλά νησιά δεν είναι ακριβώς συνδεδεμένα μεταξύ τους. Η δρομολόγηση πρέπει να πραγματοποιείται μέσω διαδρομών όπως Πειραιάς → Μύκονος → Νάξος.
- Οι πυκνοκατοικημένες αστικές περιοχές (όπως η λεκάνη της Αθήνας) έρχονται σε έντονη αντίθεση με τα αραιοκατοικημένα νησιά και τις ορεινές περιοχές, σε μια ετερογενή κατανομή των πόλεων.

#### **Δυνατότητες:**

Το σύστημα υποστηρίζει δρομολόγηση με χωρικούς και λειτουργικούς περιορισμούς, γεγονός που το καθιστά κατάλληλο για περιπτώσεις όπως η θαλάσσια εφοδιαστική, ο σχεδιασμός δρομολογίων σε νησιωτικά δίκτυα και, δυνητικά, η αντιμετώπιση κρίσεων ή η εποχιακή προσαρμογή τουριστικών διαδρομών.

Η περιβαλλοντική ποικιλομορφία της Ελλάδας την καθιστά όχι μόνο γεωγραφικό υπόβαθρο, αλλά και χρήσιμο σημείο αναφοράς για την αξιολόγηση μοντέλων βασισμένων σε μετασχηματιστές σε λογικές, μη τετριμμένες συνθήκες. Οι περιορισμοί που επιβάλλει δεν μπορούν να αφαιρεθούν χωρίς να χαθούν κρίσιμες διαστάσεις της γενίκευσης και της ανθεκτικότητας των συμπερασμάτων.

### 3.1.2 Γεωχωρική βάση δεδομένων ελληνικών πόλεων και χαρακτηριστικών

Δημιουργήθηκε μια δομημένη γεωχωρική βάση δεδομένων ελληνικών πόλεων, προκειμένου να καταστεί δυνατή η ορθολογική εκπαίδευση και η εξαγωγή συμπερασμάτων. Η βάση δεδομένων GeoNames, η οποία φιλτράρεται ώστε να περιλαμβάνει τους 100 πιο πυκνοκατοικημένους δήμους της Ελλάδας, αποτελεί την κύρια πηγή δεδομένων [26]. Τα δεδομένα εισήχθησαν σε μια κανονικοποιημένη βάση δεδομένων SQLite3 με πεδία ανά πόλη, τα οποία παρατίθενται παρακάτω:

- Όνομα (στα σύγχρονα ελληνικά)
- Γεωγραφικό πλάτος και μήκος σύμφωνα με τα πρότυπα WGS-84

$$d_{ij} = 2R \cdot \arcsin \left( \sqrt{\sin^2 \left( \frac{\varphi_i - \varphi_j}{2} \right) + \cos(\varphi_i) \cdot \cos(\varphi_j) \cdot \sin^2 \left( \frac{\lambda_i - \lambda_j}{2} \right)} \right) \quad (3.1)$$

- is\_port (δυναδική σημαία που υποδηλώνει την ύπαρξη λιμανιού)
- is\_island (δυναδική σημαία που σηματοδοτεί νησιωτικές τοποθεσίες)

Από ένα προσεκτικά επιλεγμένο αρχείο, προστέθηκαν επιπλέον πόλεις για να εξασφαλιστεί η κάλυψη υποεκπροσωπούμενων αλλά σημαντικών από λογιστική άποψη περιοχών, όπως το Γύθειο, ο Άγιος Νικόλαος, ο Τυρναβός και η Καστοριά. Η συμπερίληψη αυτή εξασφάλισε ότι οι χαμηλές κατατάξεις πληθυσμού δεν απέκλεισαν σημαντικούς ακτομεραιούς σταθμούς ή εσωτερικούς κόμβους logistics.

Μέρος του module db\_management/, το αρχείο init\_database.py προεπεξεργάζεται και φορτώνει τα δεδομένα πόλεων από αρχείο, τα οποία περιλαμβάνουν γεωγραφικές συντεταγμένες, κατηγορίες νησιών και σημαίες λιμανιών. Αυτά τα δεδομένα υποστηρίζουν τόσο την εκπαίδευση του Transformer όσο και τη λειτουργία του GUI, επιτρέποντας τη **δυναμική αναζήτηση πόλεων**, την **επιλογή βάσει φίλτρων** και την **επαλήθευση περιορισμών** κατά την αλληλεπίδραση του χρήστη.

Στερεώνοντας τη νευρωνική επαγωγή στην πραγματική γεωγραφία, αυτός ο σχεδιασμός μειώνει την απόσταση μεταξύ των συμβολικών αναγνωριστικών τοποθεσίας και των χωρικών ενσωματώσεων.

Η πλήρης υλοποίηση της διαδικασίας αρχικοποίησης και εισαγωγής των δεδομένων πόλεων παρατίθεται στο *Παράρτημα Α*.

### 3.1.3 Τοπολογικοί περιορισμοί

Σε αντίθεση με τα συνθετικά ευκλείδεια γραφήματα, όπου κάθε κόμβος συνδέεται ελεύθερα, η πραγματική δρομολόγηση στην Ελλάδα περιορίζεται από αυστηρούς τοπολογικούς περιορισμούς, ειδικά όσον αφορά τις νησιωτικές πόλεις. Η προεπεξεργασία και η εκτέλεση λογικών κανόνων βοηθούν τον σχεδιασμό του συστήματος backend να επιβάλει αυτούς τους περιορισμούς.

Πρόσβαση από νησιά με βάση τα λιμάνια.

Κάθε φορά που επιλέγεται μια νησιωτική πόλη:

### Κεφάλαιο 3

- Για να εξασφαλιστεί η λογική πρόσβαση, ο δρόμος πρέπει να περνά τουλάχιστον από ένα λιμάνι της ηπειρωτικής χώρας (όπως ο Πειραιάς, η Πάτρα).
- Η απουσία αυτού του είδους λιμανιών θα εμποδίζει την κινητικότητα μεταξύ Ηρακλείου και Αθήνας.

Αυτοί οι κανόνες τηρούνται μέσω:προ-δρομολόγησης, κατά την οποία υπολογίζονται έγκυρα σημεία εισόδου/εξόδου

- Constraints.py module στον φάκελο `tsp_algorithms/utis/` (βλ. Παράρτημα Δ), εφαρμογή κάλυψης βάσει κανόνων

Διασύνδεση μεταξύ νησιών

Όταν επιλέγονται πολλά νησιά κατά μήκος της ίδιας διαδρομής:

- Το σύστημα σας επιτρέπει να διασχίσετε ενδιάμεσα νησιά εάν υπάρχει μια λογική αλυσίδα πορθμών, π.χ. *Σύρος* → *Τήνος* → *Άνδρος*.
- Η διαδρομή θεωρείται άχρηστη όταν δεν υπάρχει άμεση αλυσίδα.

Το αποτέλεσμα είναι ένας μερικώς συνδεδεμένος γράφος διαδρομών που αποκλίνει πό την πλήρη ευκλείδεια συνδεσιμότητα. Αυτή η διάταξη δημιουργεί φυσικές προκλήσεις για τα μοντέλα που βασίζονται σε μετασχηματιστές, καθώς πρέπει να προσαρμοστούν σε περιστάσεις όπου δεν υπάρχουν έγκυρες διαδρομές, εκτός εάν περιλαμβάνονται σημαντικοί ενδιάμεσοι κόμβοι.

Στο πλαίσιο της ελληνικής τοπογραφίας, όπου υπάρχουν πολλά νησιά και λιμενικοί περιορισμοί, η σκοπιμότητα των διαδρομών απαιτεί ιδιαίτερη προσοχή. Ενώ οι κλασικοί αλγόριθμοι επίλυσης λαμβάνουν υπόψη αυτούς τους περιορισμούς κατά τη δημιουργία της διαδρομής, η μέθοδος που βασίζεται στο Transformer και εφαρμόζεται στην παρούσα διατριβή επιβάλλει τους περιορισμούς αυτούς μετά την αποκωδικοποίηση της αρχικής διαδρομής. Συγκεκριμένα, σε ένα στάδιο επικύρωσης μετά την επεξεργασία, φιλτράρονται οι μη έγκυρες διαδρομές, όπως η επίσκεψη σε ένα νησί χωρίς νόμιμο λιμάνι τόσο πριν όσο και μετά. Αυτό εγγυάται ότι οι τελικές αξιολογήσεις λαμβάνουν υπόψη μόνο τις νόμιμες διαδρομές.

Το έργο χρησιμοποιεί τη βιβλιοθήκη `geopy` στο στάδιο της επικύρωσης για τον ακριβή υπολογισμό των γεωδαιτικών αποστάσεων μεταξύ συντεταγμένων με τον τύπο Haversine. Αυτές οι αποστάσεις είναι απολύτως απαραίτητες για την εισαγωγή στο μοντέλο Transformer, καθώς και για την ανάλυση του κόστους της διαδρομής. Διαχωρίζοντας τη λογική της διαδρομής από τη λογική της σκοπιμότητας, το μοντέλο μπορεί να παραμείνει γενικής χρήσης, επιτρέποντας παράλληλα την προσαρμογή σε πραγματικούς γεωγραφικούς περιορισμούς. Η λογική περιορισμών υλοποιείται ως φίλτρο επικύρωσης επιλογών πόλεων και πορείας μέσα από το module `constraints.py`, το οποίο επιβάλλει την ύπαρξη λιμανιού εάν επιλεγεί νησί, και αποκλείει τα backtracks ή κύκλους με υπερβολικές αποστάσεις.

### 3.1.4 Προβλήματα παραλλαγών που προκαλούνται από τη γεωγραφική δομή

Αν και τα δεδομένα εκπαίδευσης δεν είναι ρητά επισημασμένα με παραλλαγές TSP, το γεωγραφικό και λογιστικό πλαίσιο της Ελλάδας οδηγεί φυσικά σε διάφορες γνωστές επεκτάσεις TSP. Αυτές οι παραλλαγές αναπτύσσονται από περιορισμούς που βασίζονται σε δεδομένα και λογική προεπεξεργασίας που εφαρμόζεται κατά τη δημιουργία περιπτώσεων εκπαίδευσης, και όχι από τυπική μοντελοποίηση.

- **Clustered TSP:** Οι πυκνοκατοικημένες περιοχές της Αττικής ή της Κρήτης δημιουργούν πυκνά συμπλέγματα κόμβων που αποτελούν πρόκληση για τη συμπεριφορά τοπικής βελτιστοποίησης του μοντέλου.
- **Γενικευμένο TSP (GTSP):** Σε καταστάσεις όπου είναι προσβάσιμος μόνο ένας κόμβος ανά ομάδα νησιών ή σύμπλεγμα, το μοντέλο αντιμετωπίζει έμμεσα ένα σενάριο GTSP[1].
- **TSP με συλλογή βραβείων (PCTSP):** είναι η προαιρετική επίσκεψη κόμβων χαμηλής χρησιμότητας ή απομακρυσμένων κόμβων (π.χ. μικρά νησιά), μιμούμενη έτσι τη δομή του PCTSP, με μερική κάλυψη[25].
- **TSP με απαιτούμενους κόμβους:** Σενάρια όπου πόλεις όπως η Αθήνα πρέπει πάντα να περιλαμβάνονται αντανακλούν περιπτώσεις TSP με απαραίτητη συμπερίληψη.
- **TSP με μερική επίσκεψη:** Προβλήματα με ευέλικτους περιορισμούς κάλυψης αντανακλώνται σε υποσύνολα νησιών ή αποσυνδεδεμένων αρχιπελάγων.

Αυτές οι διατυπώσεις αναπτύσσονται από το σχεδιασμό περιπτώσεων και την εφαρμογή περιορισμών κατά τη διάρκεια της εκπαίδευσης και της συμπερασματολογίας, και όχι από την σκληρή κωδικοποίηση στο μοντέλο. Αυξάνοντας σταδιακά την πολυπλοκότητα του γραφήματος και την ποικιλομορφία των κόμβων, το πρόγραμμα εκπαίδευσης επιτρέπει στο μοντέλο να γενικεύεται σε αυτές τις παραλλαγές χωρίς αρχιτεκτονικές αλλαγές.

Αν και δεν ορίζεται επίσημα κατά τη διάρκεια της εκπαίδευσης, αυτή η προσαρμοστική συμπεριφορά αντανακλά τη modularity και την ευελιξία της λογικής αποκωδικοποίησης, ιδιαίτερα όταν συνδυάζεται με την επικύρωση

### 3.1.5 Ορισμός του πεδίου εφαρμογής της εκπαίδευσης του μετασχηματιστή

Η δομική πολυπλοκότητα και οι γεωγραφικοί περιορισμοί που αναφέρθηκαν προηγουμένως επηρεάζουν άμεσα το πρόγραμμα εκπαίδευσης του μοντέλου Transformer. Τα παραδείγματα εκπαίδευσης κατηγοριοποιούνται σε τρία επίπεδα δυσκολίας, προκειμένου να εξισορροπηθούν η δυνατότητα εκμάθησης, η γενίκευση και η διαχείριση των περιορισμών.

**Επίπεδο 1:** Γραφήματα μικρής κλίμακας (10 έως 15 πόλεις) Πλήρως διασυνδεδεμένα

Τα συμβάντα χαρακτηρίζονται ως ευκλείδια, ισχυρά διασυνδεδεμένα και χωρίς περιορισμούς. Επιτρέπουν:

-

### Κεφάλαιο 3

- Την αρχική διαμόρφωση της λογικής προσοχής του αποκωδικοποιητή. Την εξέταση συστημάτων δεικτών, τεχνικών κάλυψης και της σύγκλισης των συναρτήσεων απώλειας.
- 
- Την χρήση εποπτευόμενης μάθησης για την επίτευξη έγκαιρης σύγκλισης. Επίπεδο 2: Ευρετικά κατασκευασμένοι, μεσαίας κλίμακας γράφοι που καλύπτουν 20 έως 99 πόλεις
- Αυτά περιλαμβάνουν τόσο νησιωτικές όσο και ηπειρωτικές πόλεις, συμπεριλαμβανομένων λιμανιών που είναι πιθανοτικά συμπεριλαμβανόμενα.
- Ενσωματώνει λογική γεωχωρικής δειγματοληψίας όπως περιγράφεται στην ενότητα 3.2.2.
- Ενσωματώνει πρακτικούς περιορισμούς, όπως η παρουσία αποσυνδεδεμένων νησιωτικών διαδρομών.
- Αναπτύσσεται ενισχυμένη μάθηση για να διευκολυνθεί η αποτελεσματική ανακάλυψη περιηγήσεων.

#### **Επίπεδο 3:** Προσαρμόσιμα σύνολα προκλήσεων νησιών

Αυτά τα προσεκτικά σχεδιασμένα σενάρια διαθέτουν:

- πολύπλοκους περιορισμούς πολλαπλών λιμανιών
- Περιορισμένη συνδεσιμότητα μεταξύ μικρών νησιωτικών κοινοτήτων
- Σαφής πρόκληση για τη γενίκευση της συμπερασματολογίας με ελάχιστη περιττότητα διαδρομών

Τα νησιά των δειγμάτων εκπαίδευσης αντιμετωπίζονται πιθανολογικά. Η προεπεξεργασία διαχειρίζεται τα νησιά αποκλειστικά μέσω λιμένων και όχι μέσω της λογικής του Transformer. Αυτή η διαίρεση εξασφαλίζει:

- Απλότητα της κατασκευής μοντέλων
- Διαφοροποιήσιμη μάθηση εντός του αποκωδικοποιητή χωρίς διαταραχή της ακεραιότητας του γραφήματος
- Η ευελιξία επιτρέπει την τροποποίηση των οδηγιών προεπεξεργασίας χωρίς την ανάγκη επανεκπαίδευσης.
- Η απομόνωση της λογικής των περιορισμών και ο έλεγχος της λεπτομέρειας του πεδίου εφαρμογής αποδίδουν σταθερή, ερμηνεύσιμη και επαναλαμβανόμενη εκπαίδευση.

#### **Σύνοψη**

Σε αυτό το μέρος περιγράφηκε ο τρόπος με τον οποίο η πραγματική γεωγραφία της Ελλάδας διαμορφώνει τη διατύπωση του προβλήματος, το σχεδιασμό του συνόλου δεδομένων και τη μοντελοποίηση των περιορισμών της παρούσας διατριβής. Το έργο ανυψώνει το TSP πάνω από τις αφηρημένες ευκλείδειες διατυπώσεις, ενσωματώνοντας περιπτώσεις TSP σε ένα γεωχωρικό περιβάλλον με πολλούς περιορισμούς, το οποίο ορίζεται από αλληλεξαρτήσεις μεταξύ νησιών και ηπειρωτικής χώρας, συγκεντρωμένες κατανομές πόλεων και ασύμμετρες κανόνες πρόσβασης.

Αυτές οι πολυπλοκότητες διαμορφώνουν τα δεδομένα εκπαίδευσης του Transformer καθώς και τη λογική των περιορισμών του συστήματος, δοκιμάζοντας έτσι τη γενίκευση, την ανθεκτικότητα και την προσαρμοστικότητα του μοντέλου σε πραγματικές συνθήκες δρομολόγησης.

## 3.2 Σχεδιασμός προγράμματος σπουδών και δημιουργία δεδομένων

### 3.2.1 Συνθετική δημιουργία γραφημάτων: Dual Source Pipeline

Τα μοντέλα μετασχηματιστών για το TSP απαιτούν τεράστιες ποσότητες δεδομένων εκπαίδευσης πολύ περισσότερες από όσες θα μπορούσαν να προέρχονται από στατικά σύνολα δεδομένων. Οι έγκυρες περιπτώσεις TSP [12] πρέπει να παράγονται δυναμικά και να αντιστοιχίζονται με ακριβή ή ευριστική πραγματική τιμή, σε αντίθεση με τις προκλήσεις ταξινόμησης εικόνων ή κειμένων [10].

Η παρούσα διατριβή δημιουργεί διπλή πηγή χρησιμοποιώντας ένα pipeline:

- Συνθετικές ομοιόμορφες περιπτώσεις
  - ▶ Τυχαία δειγματοληψία πόλεων από το τετράγωνο  $[0, 1]^2$
  - ▶ Εγγυάται τη γενίκευση της στατιστικής ποικιλομορφίας για μη ορατές διατάξεις.
  - ▶ Υποστηρίζει τη γεωμετρική συλλογιστική στο τυπικό πρόγραμμα σπουδών
- Ελληνικές ρεαλιστικές σκηνές
  - ▶ Πόλεις επιλεγμένες από τις καλύτερες. Βάση δεδομένων SQLite
  - ▶ Συμπεριλαμβάνονται λογιστικοί περιορισμοί: λιμάνια, νησιά και αποκλίσεις δρομολόγησης.
  - ▶ Επιβάλλει δομικό ρεαλισμό για πρακτικές δοκιμές εφαρμογής.

Η δημιουργία των παραπάνω περιπτώσεων γίνεται εντός των scripts εκπαίδευσης (βλ. Παράρτημα Α), όπου εφαρμόζεται η εναλλαγή μεταξύ συνθετικών και ρεαλιστικών παρτίδων κατά τη διάρκεια της εκπαίδευσης [18]. Και τα δύο σύνολα δεδομένων εγγυώνται την ανθεκτικότητα της κατανομής κατά τη διάρκεια της εκπαίδευσης και της αξιολόγησης, καθώς κανονικοποιούνται στην ίδια μορφή τανιστή (*tensor*) και, ως εκ τούτου, είναι συμβατά με τις διαδικασίες Transformer.

### 3.2.2 Μεθοδολογία γεωγραφικής δειγματοληψίας

Ο σχεδιασμός λογικών περιπτώσεων δρομολόγησης στην Ελλάδα απαιτεί γεωχωρική γνώση και όχι απλώς αυθαίρετη επιλογή. Η παρούσα διατριβή χρησιμοποιεί φιλτραρισμένο δείγμα από τις οργανωμένες πόλεις για να αναπαράγει τις συνθήκες δρομολόγησης. Η βάση δεδομένων έχει δημιουργηθεί με βάση μια λογική κανόνων που εφαρμόζεται στο αρχείο `initialize_database.py` (βλ. Παράρτημα Α).

#### Λογική λιμένων

Η δειγματοληψία μιας νησιωτικής πόλης ακολουθεί την ακόλουθη διαδικασία:

### Κεφάλαιο 3

- καλύπτει αυτόματα τουλάχιστον ένα λιμάνι της ηπειρωτικής χώρας (όπως Πειραιάς, Πάτρα, Βόλος).
- εξασφαλίζει ότι οι διαδρομές των πλοίων επιτρέπουν τη μετάβαση μεταξύ νησιών ή μεταξύ νησιών και ηπειρωτικής χώρας.

Αυτό αποτρέπει μη έγκυρες διαμορφώσεις, στις οποίες οι νησιωτικές πόλεις είναι πρακτικά απρόσιτες από θαλάσσια οδό.

#### **Φιλτράρισμα black list**

Στόχος είναι να αποφευχθεί η χωρική υπερπροσαρμογή με:

- Εξαιρούνται οι υπερ-εκπροσωπούμενες προάστειες γύρω από την Αθήνα (π.χ. Περιστερί, Αχαρνή).
- Αυτό εξισορροπεί την αναπαράσταση των δεδομένων μεταξύ αστικών και αγροτικών περιοχών.

#### **Χειροκίνητες βελτιώσεις**

Το αρχείο Extra\_cities.csv περιλαμβάνει χειροκίνητα επιλεγμένες σημαντικές πόλεις που συχνά απουσιάζουν από ανοιχτά σύνολα δεδομένων, καθώς η επιλογή τους βασίζεται κυρίως στο μέγεθος του πληθυσμού:

- *Παραδείγματα:* Καστοριά, Τυρναβός, Άγιος Νικόλαος (Κρήτη), Γύθειο,
- Αυτές οι προσθήκες βελτιώνουν την κάλυψη υποεκπροσωπούμενων αλλά σημαντικών από λογιστική άποψη περιοχών.

#### **Πιθανοτική δειγματοληψία νησιών**

Τα νησιά:

- Περιορίζονται σε μέγιστο ποσοστό 30% των πόλεων που παρακολουθούνται σε κάθε συνεδρία εκπαίδευσης
- Δεν περιλαμβάνονται, εκτός εάν ενεργοποιείται η λογική του λιμανιού.
- Είναι πιο συνηθισμένα σε εξειδικευμένες παρτίδες «Island Challenge» που δίνουν έμφαση στην ασύμμετρη και αραιή δρομολόγηση

Αντανακλώντας την πραγματική τους αξία στην εθνική εφοδιαστική, πόλεις με υψηλή συχνότητα, όπως η Αθήνα, η Θεσσαλονίκη και το Ηράκλειο, εμφανίζονται σε περισσότερες παρτίδες.

Αυτή η αρχιτεκτονική εγγυάται ότι κάθε περίπτωση που λαμβάνεται αντανακλά μια λογιστικά συνεκτική και γεωγραφικά ρεαλιστική κατάσταση, παρέχοντας έτσι στο Transformer μια χρήσιμη δομή από την οποία μπορεί να αναπτυχθεί.

### 3.2.3 Λογική της μάθησης του προγράμματος σπουδών

Επεκτείνοντας το πεδίο εφαρμογής της εκπαίδευσης που ορίζεται στην ενότητα 3.1.5, η παρούσα ενότητα παρέχει την προσέγγιση μάθησης του προγράμματος σπουδών (curriculum) [10] που χρησιμοποιείται για τον προγραμματισμό της έκθεσης του μοντέλου σε πιο δύσκολες συνθήκες.

Η μάθηση του προγράμματος σπουδών βασίζεται στην εκπαιδευτική έννοια της εκκίνησης με απλές ιδέες και της σταδιακής μετάβασης σε πιο απαιτητικές. Αυτή η προσέγγιση μειώνει την υπερπροσαρμογή σε σενάρια χαμηλής πολυπλοκότητας, συμβάλλει στην άμβλυνση των προβλημάτων σύγκλισης και ενισχύει τη γενίκευση σε διάφορους αριθμούς πόλεων και τύπους γραφημάτων στο πλαίσιο της εκπαίδευσης μοντέλων Transformer για το TSP.

Η παρούσα διατριβή ενσωματώνει ένα πρόγραμμα σπουδών (curriculum) τριών φάσεων στη διαδικασία δημιουργίας δεδομένων και στον προγραμματιστή εκπαίδευσης:

#### **Φάση 1:** Εποπτευόμενη μάθηση σε δέκα έως δεκαπέντε μικρά ιδρύματα

- Χρήση πλήρως συνδεδεμένων ευκλείδειων γραφημάτων
- Οι περιηγήσεις ground-truth δημιουργούνται μέσω της δικής μας υλοποίησης ακριβούς επιλυτή Branch-and-Bound, προσαρμοσμένου για μικρά ευκλείδεια γραφήματα.
- Εμπνέει τον αποκωδικοποιητή να αποκτήσει κατανόηση:
  - Οργανωμένη μετάθεση
  - Επιλογή ανάλογα με συμβουλές για πόλεις
  - Περιήγηση εφικτή υπό συνθήκες χαμηλής πολυπλοκότητας.

#### **Φάση 2:** 20–75 πόλεις Μεσαίου μεγέθους περιπτώσεις με περιορισμούς

- Συνδυάζει γραφήματα με ελληνικά και συνθετικά θεμέλια.
- Οι διαδρομές καλύπτουν άνισα κόστη, συγκεντρωμένη γεωγραφία και αβέβαιη πρόσβαση σε λιμάνια.
- Διδάσκεται με τη διάρκεια της περιήγησης ως ένδειξη ανταμοιβής, ενισχυμένη με REINFORCE.
- Βοηθά το μοντέλο να:
  - Γενικά σε όλες τις κατανομές εισόδου
  - Έλεγχος μερικής αραιότητας και σύνδεσης

#### **Φάση 3:** «Πρόκληση νησιών» και δύσκολες περιπτώσεις (έως 100 πόλεις)

- Η αραιότητα των ακμών, οι εξαρτήσεις πολλαπλών λιμένων και οι ασύνδετοι υπογράφοι βοηθούν στο σχεδιασμό.
- Ισχυρές διακυμάνσεις στην πυκνότητα των πόλεων και στην αυστηρότητα των περιορισμών
- Δεν υπάρχει εποπτικό σήμα. Η εκπαίδευση εξαρτάται αποκλειστικά από την ενίσχυση [11].

## Κεφάλαιο 3

Αυτές οι σειρές προκλήσεων προσομοιώνουν τις χειρότερες περιπτώσεις (π.χ. πέντε νησιά με μία νόμιμη σύνδεση με φέριμποτ) και ωθούν το Transformer να υιοθετήσει δομική σκέψη εκτός των συμβατικών παραδοχών.

Ενσωματωμένος στον βρόχο εκπαίδευσης (`train_transformer_curriculum.py`), ο προγραμματιστής του προγράμματος σπουδών επιλέγει δυναμικά την κατηγορία του συνόλου δεδομένων και το μοτίβο του σήματος κινήτρου για κάθε φάση. Αυτή η αρθρωτή δομή καθιστά δυνατή τη μελλοντική επέκταση, για παράδειγμα, χρονικά πλαίσια ή παραλλαγές πολλαπλών αποθηκών.

Η πρακτική υλοποίηση της παραπάνω ροής, με αυτόματη διαχείριση datasets και εποπτική εκπαίδευση σε stages, παρουσιάζεται στο *Παράρτημα Β.2*.

### 3.2.4 Εποπτευόμενη έναντι ενισχυμένης εκπαίδευσης

Το πρόγραμμα εκπαίδευσης του μοντέλου Transformer διαρθρώνει την ενισχυτική μάθηση (RL) και την εποπτευόμενη μάθηση (SL) σε μια ακολουθία. Αρχικά, αυτή η υβριδική προσέγγιση εγγυάται ότι το μοντέλο θα μάθει σταθερή συμπεριφορά αποκωδικοποίησης. Στη συνέχεια, προσαρμόζεται προοδευτικά σε σύνθετα περιβάλλοντα με πολλούς περιορισμούς.

#### Φάση εποπτευόμενης μάθησης

Το μοντέλο εκπαιδεύεται κατά τις πρώτες φάσεις του προγράμματος σπουδών σε επισημασμένες διαδρομές που παράγονται από υλοποιημένο επιλυτή Branch-and-Bound ή από ευρετικές μεθόδους (όπως Nearest Neighbor). Η διασταυρούμενη εντροπία πάνω στην κατάλληλη μετάθεση των πόλων αποτελεί τη βάση της συνάρτησης απώλειας. Αυτή η φάση διδάσκει τη βασική συμπεριφορά κατασκευής περιηγήσεων και σταθεροποιεί τον αποκωδικοποιητή.

#### Πλεονεκτήματα:

- Γρήγορη σύγκλιση σε καλά καθορισμένα μικρά προβλήματα
- Χαμηλή διακύμανση και ερμηνεύσιμα σχόλια απώλειας

#### Περιορισμοί:

- Δεν είναι επεκτάσιμο για κακώς οργανωμένους ή μεγάλους γραφήματα
- Εξαρτάται από ακριβείς ετικέτες, οι οποίες ενδέχεται να μην υπάρχουν στα πραγματικά δεδομένα.

#### Φάση ενισχυτικής μάθησης:

Το μοντέλο βελτιστοποιείται χρησιμοποιώντας REINFORCE [11] με το μήκος της διαδρομής να θεωρείται αρνητική ανταμοιβή καθώς η εκπαίδευση προχωρά σε μεγαλύτερα και πιο σύνθετα σενάρια. Οι προαιρετικές βασικές λειτουργίες μειώνουν τη διακύμανση. Αυτή η φάση επιτρέπει στο μοντέλο να διαχειρίζεται μερική ή θορυβώδη εποπτεία και να διερευνά νέες δομές διαδρομών.

#### Πλεονεκτήματα

- Υποστηρίζει περιπτώσεις γραφημάτων χωρίς επισημασμένη πραγματική τιμή

- Επιτρέπει τη διαμόρφωση ανταμοιβών ανάλογα με περιορισμούς συγκεκριμένου τομέα.

### **Περιορισμοί**

- Μειωμένη αστάθεια και βραδύτερη σύγκλιση
- Ευαίσθητο στην κλιμάκωση των ανταμοιβών και στο σημείο εκκίνησης

### **Ολοκληρωμένος κύκλος εκπαίδευσης**

Η παρούσα διατριβή εισάγει έναν υβριδικό κύκλο: εποπτευόμενη προεκπαίδευση για την κανονικότητα της διαδρομής, ακολουθούμενη από ενίσχυση της εκγύμνασης σε προοδευτικά περιορισμένους και ρεαλιστικούς γραφήματα. Αυτό επιτρέπει στο μοντέλο να αποκτήσει ευχέρεια στην ακολουθία πριν αντιμετωπίσει τους ιδιαίτερα μεταβλητούς γεωχωρικούς περιορισμούς που παρατηρούνται σε πρακτικές καταστάσεις δρομολόγησης.

### **3.2.5 Στρωματοποιημένα σύνολα δοκιμών: Αξιολόγηση**

Η αξιολόγηση έχει ως στόχο να αξιολογήσει τη γενίκευση, την ανθεκτικότητα σε περιορισμούς και τη συνέπεια υπό συνθήκες διαφορετικές από αυτές της εκπαίδευσης. Τα παραδείγματα αξιολόγησης στρωματοποιούνται επομένως κατά μήκος τριών αξόνων:

#### **Αριθμός πόλεων (Στρωματοποίηση βάσει μεγέθους):**

- Μικρές περιπτώσεις (10–15 πόλεις):  
Για εντοπισμό υποπροσαρμογής ή υπερπροσαρμογής.
- Μικρές-μεσαίες (20–30 πόλεις):  
Αντιπροσωπεύουν κανονική συμπερασματολογία χρήστη.
- Μεσαίες (50–75 πόλεις):

Για μέτρηση συνοχής, ταχύτητας και δομής λύσης. Μεγάλες (έως 100 πόλεις):

Ελέγχουν τη γενίκευση και την ανθεκτικότητα σε μεγέθη που δεν είδε το μοντέλο.

#### **1. Γεωγραφική σύνθεση**

- Τυπικά συνθετικά ομοιόμορφα πλέγματα βασισμένα μόνο σε ευκλείδειες αποστάσεις
- Μόνο η ηπειρωτική Ελλάδα: Δεν υπάρχει λογική βάση για πλοίο
- Μικτή (νησιά + ηπειρωτική χώρα): Περιλαμβάνει αρχιπέλαγος με περιορισμούς λιμένων
- Σύνολο προκλήσεων για νησιά: αραιά γραφήματα που προορίζονται να δοκιμάσουν τη σκοπιμότητα του μοντέλου υπό πίεση

#### **2. Τύπος περιπτώσεων τριών αστέρων**

## Κεφάλαιο 3

- Πλήρης συνδεσιμότητα: χωρίς περιορισμούς
- Η εγκυρότητα εξαρτάται από τις διαδρομές πρόσβασης, καθώς υπάρχουν περιορισμοί λιμένων.
- Συγκεντρωμένα: πυκνά υπογραφήματα, ιδιαίτερα κοντά στην Κρήτη και την Αττική
- Χειροκίνητα σχεδιασμένες δοκιμαστικές περιπτώσεις για συγκεκριμένη συμπεριφορά συμπερασμάτων αποτελούν τις περιπτώσεις της ιστορίας.

Κάθε παρτίδα αξιολόγησης συνοδεύεται από:

- Διόρθωση πραγματικών δεδομένων (μέσω brute-force σε μικρές περιπτώσεις) [13][15]
- Συνοπτικά στατιστικά κατανομής για τη διάρκεια των περιηγήσεων
- Υπολογισμένο χάσμα απόδοσης από το βέλτιστο, όπου υπάρχει διαθέσιμη ground-truth

### 3.2.6 Μάσκα, συμπλήρωση και ομαδοποίηση για εισόδους ποικίλου μήκους

Η διαχείριση γραφημάτων εισόδου μεταβαλλόμενου μήκους είναι μία από τις πρακτικές δυσκολίες στην εφαρμογή μοντέλων βασισμένων σε Transformer για το TSP. Χρησιμοποιούνται συστήματα συμπλήρωσης και μάσκας για τη διατήρηση της συνέπειας των εισόδων και του παραλληλισμού της GPU [17], καθώς ο αριθμός των πόλεων σε κάθε περίπτωση διαφέρει μεταξύ των παρτίδων.

Το σύστημα αντιμετωπίζει αυτό το πρόβλημα μέσω:

- Σταθερού μεγέθους παρτίδας: Κάθε κύκλος εκπαίδευσης χειρίζεται 128 περιπτώσεις.
- Γραφήματα με λιγότερες από τον μέγιστο αριθμό πόλεων έχουν εικονικούς κόμβους που τα συμπληρώνουν
- Μάσκα: Ο μηχανισμός προσοχής αγνοεί αυτούς τους συνθετικούς κόμβους στους οποίους έχει αποδοθεί η μάσκα=0.
- Προβολή όλων των συντεταγμένων στο κανονικοποιημένο εύρος 2D  $[0,1]^2$ .

Ένα προσαρμοσμένο PyTorch DataLoader [19][20] χειρίζεται την ομαδοποίηση και τη συρραφή παρτίδων και:

- Εγγυάται ότι όλες οι παρτίδες χωράνε στη χωρητικότητα της μνήμης.
- Συντεταγμένες πραγματικών και ψεύτικων εισόδων για αποτελεσματική κάλυψη της προσοχής.
- Επιτρέπει στην GPU να ενεργοποιήσει τον παραλληλισμό πολλαπλών περιπτώσεων.

Αυτή η υποδομή διατηρεί την υπολογιστική αποδοτικότητα και την απόδοση του μοντέλου, ενώ υποστηρίζει διαφορετικούς αριθμούς πόλεων κατά τη διάρκεια της εκπαίδευσης.

## Σύνοψη

Μια αρθρωτή φιλοσοφία σχεδιασμού που λαμβάνει υπόψη τους περιορισμούς ενοποιεί την αρχιτεκτονική, τη στρατηγική εκπαίδευσης και το πλαίσιο δημιουργίας δεδομένων που περιγράφονται λεπτομερώς σε αυτό το κεφάλαιο. Αν και η εκπαίδευση δομής, η κάλυψη περιπτώσεων και η επικύρωση μετά την αποκωδικοποίηση προσθέτουν ευελιξία, το βασικό μοντέλο Transformer παραμένει γενικής χρήσης.

Τρεις κύριοι στόχοι καθορίζουν αυτόν τον σχεδιασμό:

### **Εκπαίδευση σε πρόγραμμα σπουδών με προοδευτική πολυπλοκότητα**

Μικροί, συμμετρικοί και πλήρως συνδεδεμένοι γράφοι αποτελούν τη βάση της εκπαίδευσης. Το μοντέλο προχωρά προοδευτικά σε όλο και πιο περίπλοκες, περιορισμένες περιπτώσεις. Αυτή η ακολουθία εγγυάται ότι η συμπεριφορά αποκωδικοποίησης αποκτάται πρώτα υπό βέλτιστες συνθήκες και στη συνέχεια ενισχύεται υπό καθεστώς με περιορισμένους πόρους.

### **Συμμόρφωση με αρθρωτούς περιορισμούς**

Η σκοπιμότητα επιβάλλεται εξωτερικά κατά την αποκωδικοποίηση και όχι με την άμεση κωδικοποίηση των κανόνων logistics π.χ. συνδεσιμότητα νησιών-λιμένων στην αρχιτεκτονική του μοντέλου. Αυτό ελαχιστοποιεί την αρχιτεκτονική διόγκωση και διευκολύνει την απλή ενσωμάτωση με περιορισμούς συγκεκριμένου τομέα.

### **Επικύρωση και βελτιστοποίηση μετά την αποκωδικοποίηση**

Μέσω της αναζήτησης δέσμης, της βελτίωσης 2-optimal και της επικύρωσης βάσει κανόνων, το σύστημα δημιουργεί υψηλής ποιότητας και πρακτικές διαδρομές χωρίς να απαιτείται επανεκπαίδευση του μοντέλου για κάθε παραλλαγή ή τοπολογία [8],[23].

Αυτά τα στοιχεία, σε συνδυασμό, προσδίδουν στο σύστημα που βασίζεται στο Transformer έναν επεκτάσιμο και ισχυρό σχεδιασμό. Υποστηρίζει δεδομένα του πραγματικού κόσμου με ατελή δομή, γενικεύεται σε παραλλαγές και τοπολογίες TSP και παραμένει επεκτάσιμο σε διαδραστικές εφαρμογές, συμπεριλαμβανομένης της δρομολόγησης βάσει GUI σε δίκτυα ελληνικών νησιών.

## Κεφάλαιο 4: Αρχιτεκτονική Συστήματος και Διεπαφή Χρήστη (GUI)

### 4.1 Ερμηνεία γραφημάτων και κωδικοποίηση χωρικών χαρακτηριστικών

Μία από τις βασικές προκλήσεις στην εφαρμογή μοντέλων Transformer σε χωρικά προβλήματα όπως το πρόβλημα του περιοδεύοντος πωλητή (TSP) είναι η διαχείριση μη ταξινομημένων γεωμετρικών δεδομένων. Σε αντίθεση με τη γλώσσα, στην οποία οι ακολουθίες ακολουθούν μια σαφή σειρά, οι πόλεις σε μια περίπτωση TSP καθορίζονται απλώς από τις χωρικές συντεταγμένες τους και δεν έχουν καμία φυσική χωρική κατεύθυνση. Επομένως, για να γεφυρωθεί το χάσμα μεταξύ των χωρικών δεδομένων και της συλλογιστικής που βασίζεται στην προσοχή, απαιτείται εξειδικευμένη κωδικοποίηση χαρακτηριστικών. Θεωρώντας τις πόλεις ως διδιάστατα χωρικά σύνολα

Κάθε περίπτωση TSP χαρακτηρίζεται ως μια συλλογή  $n$  πόλεων με συντεταγμένες  $X = [(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), \dots, (x_i, y_i)] \in \mathbb{R}^2$ .

Οι συντεταγμένες είναι:

Τυποποιημένες σε εύρος  $[0, 1]^2$  για να παρέχουν αμετάβλητη κλίμακα, αποθηκευμένες σε έναν πίνακα μορφής  $\mathbb{R}^{n \times 2}$ , όπου κάθε σειρά αντιπροσωπεύει μια πόλη. Αυτή η διατύπωση διατηρεί την αμεταβλητότητα της μετάθεσης, εγγυώμενη έτσι ότι η έξοδος του μοντέλου είναι ανεξάρτητη από τη σειρά εισόδου μια κρίσιμη προϋπόθεση για τη σκέψη με βάση σύνολα. περιλαμβάνουν γεωγραφικές συντεταγμένες. Για να εξοπλιστούν τα δεδομένα για επεξεργασία με βάση την προσοχή, οι συντεταγμένες κάθε πόλης προβάλλεται σε έναν κρυφό χώρο υψηλότερης διάστασης. Αυτό επιτυγχάνεται με μια μαθημένη γραμμική μετασχηματιστική απεικόνιση κάθε εισόδου 2D σε μια πλουσιότερη διανυσματική αναπαράσταση κατάλληλη για προσοχή στο γινόμενο σημείων.

**Προβολή:**

- προβάλλει κάθε πόλη σε μια περιοχή με μεγαλύτερη αναπαραστατική ικανότητα.
- αποφασίζει για το μοντέλο πιθανές γρήγορες περικοπές, χωρικές σχέσεις και αναγνώριση συστάδων.

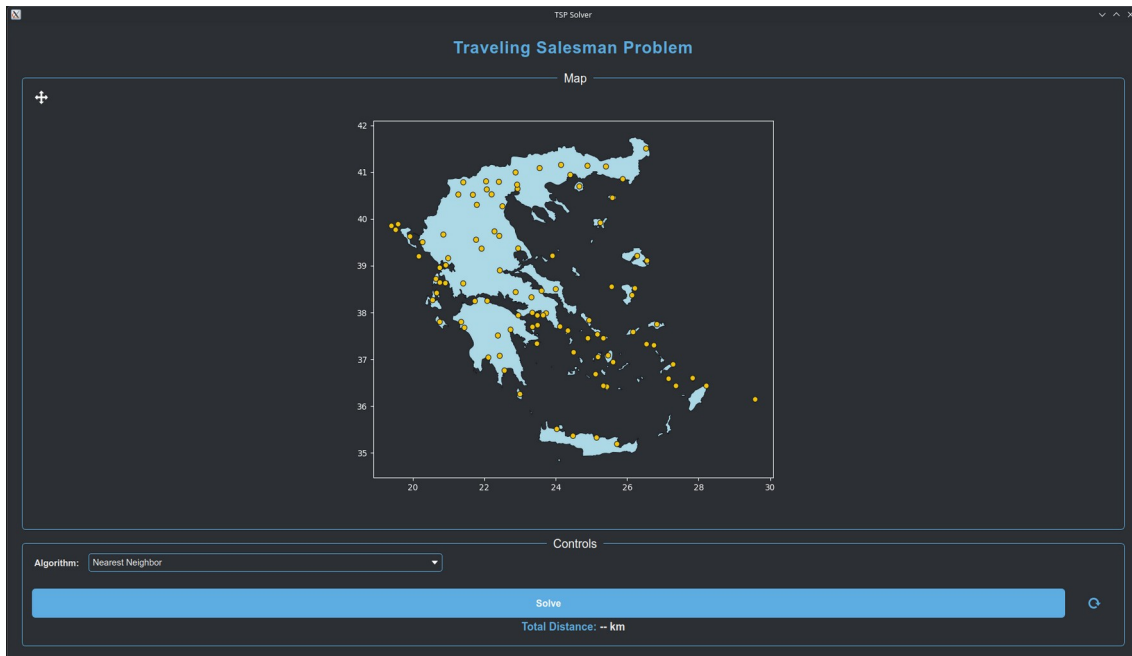
**Δομή εισόδου για μηχανισμό προσοχής**

Ο κωδικοποιητής Transformer ξεκινά με τις προκύπτουσες ενσωματώσεις ως διακριτικά του. Αυτές οι χωρικά ευαίσθητες εισόδους αντικαθιστούν τις παραδοσιακές ενσωματώσεις γλώσσας για να βοηθήσουν τα επίπεδα προσοχής να συλλογιστούν σχετικά με τη σχετική θέση, την εγγύτητα και τη γεωμετρική δομή.

Βασιζόμενο σε αυτήν την προσέγγιση προβολής και κανονικοποίησης, το μοντέλο μπορεί να χειριστεί με ακρίβεια και ευελιξία τις χωρικές εισόδους, αποφεύγοντας έτσι την προκατάληψη από τη σειρά των εισόδων.

Η γραφική διεπαφή της εφαρμογής αναπτύχθηκε χρησιμοποιώντας τη βιβλιοθήκη PyQt6, επιτρέποντας την διαδραστική επιλογή αλγορίθμων και παραμέτρων (όπως το μέγεθος της δέσμης και η εναλλαγή 2-OPT). Το Matplotlib χρησιμοποιείται για την οπτικοποίηση των σημείων της πόλης και των συνδέσεων των διαδρομών σε έναν χάρτη, ενώ το pandas χειρίζεται τη χειραγώγηση των δεδομένων. Όλα τα χωρικά δεδομένα και τα δεδομένα διαδρομών αποθηκεύονται σε μια βάση δεδομένων SQLite, εξασφαλίζοντας αποτελεσματική φόρτωση και αναπαραγωγιμότητα. Όλες οι εξαρτήσεις παρατίθενται στο συνοδευτικό αρχείο requirements.txt, επιτρέποντας τη ρύθμιση του περιβάλλοντος μέσω εργαλείων όπως το pip ή το venv.

Το Σχήμα 4.2 απεικονίζει τη βασική δομή της γραφικής διεπαφής χρήστη, η οποία επιτρέπει στον χρήστη να επιλέγει σύνολα πόλεων, να ενεργοποιεί αλγορίθμους και να παρακολουθεί τα αποτελέσματα σε πραγματικό χρόνο. Παραδείγματα πλήρους εκτέλεσης παρουσιάζονται στο Κεφάλαιο 6.6.2.



Σχήμα 4.1 - Κεντρική γραφική διεπαφή χρήστη (GUI) του συστήματος δρομολόγησης.

## 4.2 Κωδικοποίηση θέσεων και αμεταβλητότητα μεταθέσεων

Οι μετασχηματιστές σχεδιάστηκαν αρχικά για διαδοχικά δεδομένα, όπου τα εισερχόμενα tokens επεξεργάζονται σε σταθερή ακολουθία. Ωστόσο, στο πλαίσιο του προβλήματος του περιοδεύοντος πωλητή (TSP), το μοντέλο πρέπει να λειτουργεί σε μη ταξινομημένα σύνολα χωρικών εισόδων πιο συγκεκριμένα, σε συντεταγμένες πόλεων σε δύο διαστάσεις χωρίς φυσική ακολουθία. Αυτό απαιτεί μια διαφορετική μέθοδο κωδικοποίησης θέσεων, η οποία διατηρεί τη γεωμετρική δομή χωρίς προκατάληψη τάξης.

## 2D ημιτονοειδής κωδικοποίηση θέσης

- Εμπνευσμένη από τους Vaswani και η ομάδα του [17], η παρούσα διατριβή χρησιμοποιεί ένα σχήμα 2D ημιτονοειδούς κωδικοποίησης τροποποιημένο για γεωγραφικά δεδομένα, ώστε να διατηρείται η χωρική δομή χωρίς να παραβιάζεται η αμεταβλητότητα της μετάθεσης.
- Η κωδικοποίηση χρησιμοποιεί συναρτήσεις ημιτόνου και συνημιτόνου για κάθε διάσταση συντεταγμένων, αντί να εκχωρεί δείκτες διαδοχικής θέσης.

Η ενσωμάτωση θέσης για ένα ζεύγος συντεταγμένων  $(x, y)$  υπολογίζεται ως

$$PE_{pos,2i} = \sin\left(pos/10000^{2i/d_{model}}\right) \quad (4.1)$$

$$PE_{pos,2i+1} = \cos\left(pos/10000^{2i/d_{model}}\right) \quad (4.2)$$

Αυτα τα δύο διανύσματα στη συνέχεια συνενώνονται για να δημιουργήσουν για κάθε πόλη ένα ενιαίο διανυσματικό ενσωματωμένο με γνώση της θέσης.

Με όρους συχρότητας  $\omega$  με λογαριθμική απόσταση, το μοντέλο μπορεί:

- Να εντοπίσει λεπτομερείς διαφορές μεταξύ γειτονικών πόλεων.
- Να καταγράψει χονδροειδή δομικά μοτίβα, συμπεριλαμβανομένων γεωγραφικών ορίων ή συσπειρώσεων πόλεων.

Αυτό είναι απαραίτητο για την εκμάθηση πολιτικών δρομολόγησης που εξαρτώνται τόσο από τοπικά όσο και από παγκόσμια χωρικά χαρακτηριστικά και προσφέρει στον μηχανισμό προσοχής πολυκλιμακική γεωμετρική επίγνωση.

## Αμετάβλητη μετατόπιση και συμβατότητα προσοχής

Αυτή η κωδικοποίηση παραμένει εντελώς αμετάβλητη ως προς τη σειρά εισόδου, σε αντίθεση με τους συμβατικούς μετασχηματιστές που μαθαίνουν ενσωματώσεις σταθερής θέσης. Αυτό επιτυγχάνεται με:

- συνεπή κανονικοποίηση συντεταγμένων
- ημιτονοειδείς κωδικοποιήσεις αντί για μάθηση θέσεων
- απουσία οποιωνδήποτε ετικετών θέσης ή ταξινόμησης βάσει ευρετηρίου

Ακολουθεί:

- Η λογική του μοντέλου για τη λήψη αποφάσεων υπερβαίνει την ακολουθία των γεγονότων.
- Η αντιστροφή των πόλεων εισόδου δεν έχει καμία επίδραση στην απόδοση, επομένως εγγυάται τη γενίκευση.

- Τα επίπεδα προσοχής λειτουργούν κανονικά σε όλες τις πιθανές μεταθέσεις της εισόδου.

Αυτή η μέθοδος διασφαλίζει ότι το μοντέλο παραμένει πιστό στον χαρακτήρα του TSP και δεν εξαρτάται από συνθετικές προκαταλήψεις που προκαλούνται από την ευρετηρίαση των token.

### **Η κάλυψη και η συμπλήρωση βοηθούν στη διαχείριση διαφορετικών μεγεθών εισόδου.**

Οι πραγματικές εργασίες δρομολόγησης απαιτούν μερικές φορές διαφορετική κλίμακα. Η παρούσα διατριβή εκπαιδεύει και αξιολογεί το μοντέλο σε γραφήματα TSP που καλύπτουν 10 έως 100 πόλεις, δημιουργώντας έτσι δυσκολίες για τους μηχανισμούς υπολογισμού και προσοχής που βασίζονται σε τανυστές και απαιτούν εισόδους σταθερού μεγέθους.

Επομένως, εφαρμόζεται ένα σχήμα συμπλήρωσης και κάλυψης για την υποστήριξη εισόδων μεταβλητού μήκους τόσο κατά την εκπαίδευση όσο και κατά την εξαγωγή συμπερασμάτων.

### **Συμπλήρωση**

Η προσθήκη ψεύτικων καταχωρήσεων σε μικρότερες περιπτώσεις βοηθά κάθε παρτίδα να ευθυγραμμιστεί με ένα καθορισμένο μέγιστο μέγεθος. Αυτές οι πόλεις με συμπλήρωση, οι οποίες αγνοούνται δομικά κατά τον υπολογισμό, έχουν συντεταγμένες που έχουν οριστεί σε ένα ουδέτερο σύμβολο κράτησης θέσης.

### **Μάσκα**

Για να αποφευχθεί η υποβάθμιση του μηχανισμού προσοχής από αυτές τις καταχωρήσεις με συμπλήρωση:

- Το μοντέλο χρησιμοποιεί λογική μάσκας για να φιλτράρει τις τοποθεσίες με συμπλήρωση από τις ενημερώσεις και τις αναζητήσεις προσοχής.
- Τα μάσκαρισμένα tokens αντιμετωπίζονται ως μη προσβάσιμα κατά την αποκωδικοποίηση, αποτρέποντας έτσι την επιλογή τους ως μέρος μιας νόμιμης περιήγησης.

Αυτό εγγυάται ότι ο αποκωδικοποιητής δημιουργεί μόνο λογικές περιηγήσεις εντός του πραγματικού μεγέθους του γραφήματος και ότι οι βαθμολογίες προσοχής υπολογίζονται μόνο μεταξύ πραγματικών πόλεων.

### **Γεωγραφικοί περιορισμοί (λιμάνια και νησιά)**

Αν και οι ενσωματώσεις πόλεων δεν αντικατοπτρίζουν συγκεκριμένα την κατάσταση των λιμανιών και των νησιών, το σύστημα χρησιμοποιεί αυτές τις πληροφορίες μέσω προ-φιλτραρίσματος και επικύρωσης δρομολόγησης:

- Τα λιμάνια επιτρέπουν τις επίσημες μεταφορές μεταξύ νησιών.
- Η παρουσία λιμανιών στην περιήγηση είναι απαραίτητη για να είναι εφικτές οι νησιωτικές πόλεις.

## Κεφάλαιο 4

Η προεπεξεργασία και η επικύρωση συμπερασμάτων επιβάλλουν αυτές τις ειδικές για τον τομέα οδηγίες από έξω προς τα μέσα. Τέτοια δεδομένα θα μπορούσαν να συμπεριληφθούν απευθείας στο

μοντέλο σε μελλοντικές εργασίες, για παράδειγμα με τη χρήση εκμάθησης δυαδικών σημαδιών ή μάσκας προσοχής με γνώση περιορισμών.

### 4.3 Τεκμηρίωση Εξωτερικών Περιορισμών

Αυτή η ενότητα περιγράφει την ανάγκη για διαχείριση περιορισμών που βασίζονται στον πραγματικό κόσμο π.χ. ανάγκη για επίσκεψη σε λιμάνια όταν υπάρχουν νησιά. Οι περιορισμοί αυτοί δεν ενσωματώνονται άμεσα στη δομή του TSP αλλά επιβάλλονται εξωτερικά στο μοντέλο και το GUI.

#### Δομή Βάσης Δεδομένων

- Χρήση σημαίας `is_port` και `is_island` στο `init_database.py` (βλ. Παράρτημα Α)
- Κάθε πόλη χαρακτηρίζεται ως ηπειρωτική, νησιωτική, ή λιμάνι

#### Λογική Επιβολής Περιορισμών

Κατά την επιλογή εισόδου, εφαρμόζονται:

- Εξωτερικά φίλτρα για ομαδοποίηση
- Επαλήθευση ύπαρξης συνδέσεων προς λιμάνια για όλα τα νησιά

Οι περιορισμοί δεν επηρεάζουν την αρχιτεκτονική του Transformer, αλλά επηρεάζουν την είσοδο και τις μάσκες αποκωδικοποίησης

#### Πλεονεκτήματα

- Διατηρείται η καθαρότητα του μοντέλου
- Το σύστημα είναι επεκτάσιμο: μπορούν να ενσωματωθούν μελλοντικά χρονικά παράθυρα, όρια φορτίου ή πολλαπλοί αποθηκευτικοί κόμβοι
- Υποστηρίζεται λογική πολλαπλών μεταφορικών μέσων (θαλάσσια + χερσαία)

### 4.4 Πρακτική Επιβολή Περιορισμών στο GUI

Αυτή η ενότητα περιγράφει τη συγκεκριμένη εφαρμογή των περιορισμών μέσω του γραφικού περιβάλλοντος χρήστη.

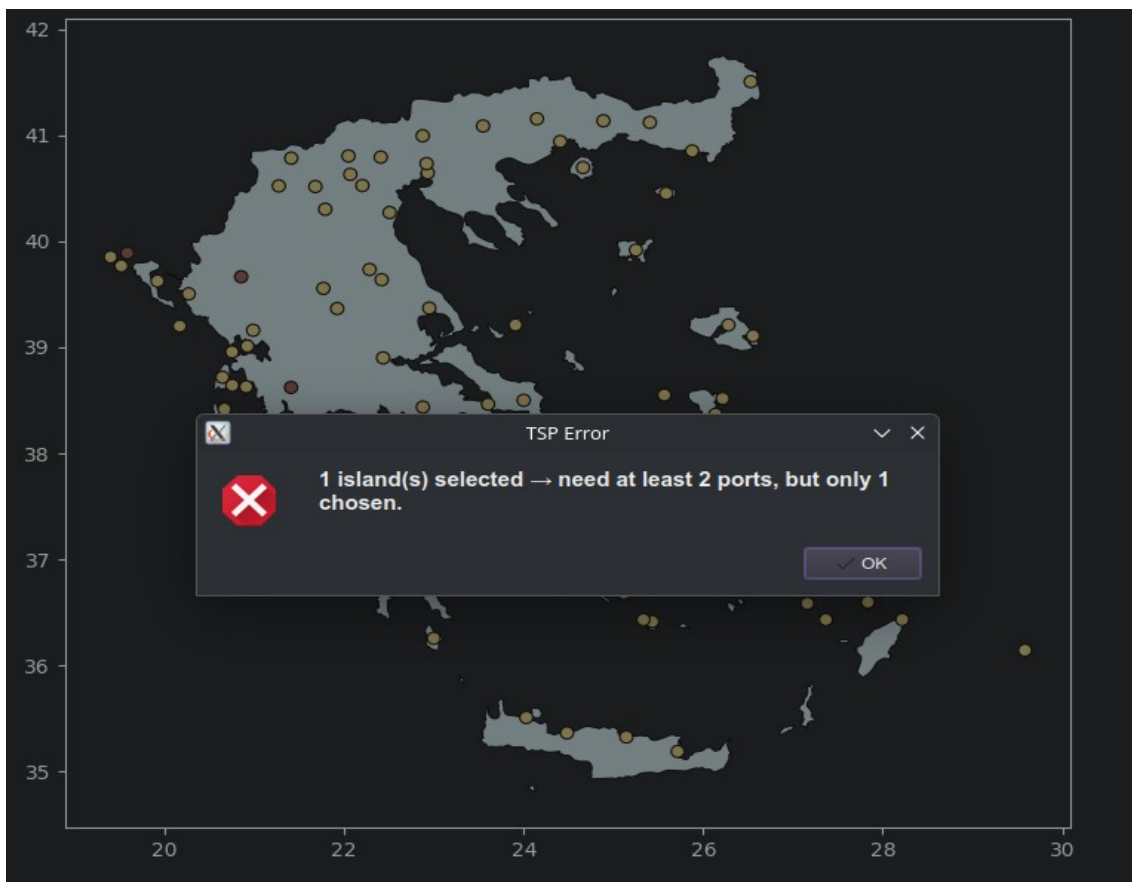
#### Υλοποίηση

- Το αρχείο `main_window.py` χρησιμοποιεί:

- Τη μέθοδο `validate_required_ports()` για να ελέγχει εάν υπάρχει τουλάχιστον ένα λιμάνι για κάθε επιλεγμένο νησί[23]. Η υλοποίηση της μεθόδου `validate_required_ports()` παρουσιάζεται στο Παράρτημα Δ.
- Αυτόματη απενεργοποίηση BnB ή 2-OPT εάν το  $n$  υπερβεί τα επιτρεπόμενα `thresholds` ( $n > 20$ ,  $n > 25$ )

### Οπτική Ανατροφοδότηση

- Εάν η επιλογή εισόδου αποτύχει την επαλήθευση (π.χ. νησί χωρίς σύνδεση), εμφανίζεται προειδοποιητικό μήνυμα.

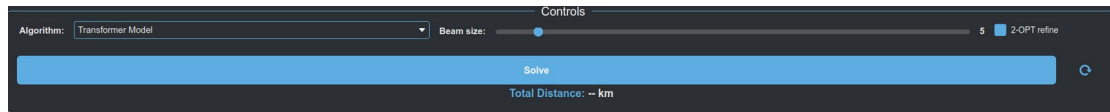


Σχήμα 4.2 - Μήνυμα σφάλματος GUI για νησιά χωρίς λιμένα

- Οι επιλογές αλγορίθμων απενεργοποιούνται δυναμικά βάσει μεγέθους ή σφαλμάτων

### Πλεονεκτήματα

- Παρέχεται καθοδήγηση στον χρήστη χωρίς την ανάγκη τροποποίησης του μοντέλου
- Εξασφαλίζεται εγκυρότητα εισόδου πριν ξεκινήσει η αποκωδικοποίηση



Σχήμα 4.3 - Επιλογές ελέγχου 2-OPT και εύρους δέσμης μέσω ρυθμιστικών GUI

#### 4.5 Συγκριτική ανάλυση με προηγούμενες τεχνικές κωδικοποίησης

Αν και η προσέγγιση κωδικοποίησης χαρακτηριστικών που επιλέχθηκε στην παρούσα διατριβή αντλεί έμπνευση από πρόσφατες καινοτομίες στη νευρωνική συνδυαστική βελτιστοποίηση, αντιμετωπίζει συγκεκριμένους περιορισμούς προηγούμενων προσεγγίσεων. Σημαντικές πηγές έμπνευσης περιλαμβάνουν τα σχέδια των Joshi και της ομάδος του και Kool και της ομάδος του [18],[19], τα οποία συμβάλλουν στην εφαρμογή μηχανισμών προσοχής στην επίλυση εργασιών τύπου TSP.

Σημαντικές καινοτομίες σε σχέση με προηγούμενες εργασίες

**Πίνακας 3.1 - Σύγκριση των βασικών χαρακτηριστικών εισόδου, συνδεσιμότητας και εκπαίδευσης μεταξύ προηγούμενων προσεγγίσεων και της παρούσας διατριβής.**

Χαρακτηριστικό	Προηγούμενη Βιβλιογραφία	Η Παρούσα Διατριβή
Προβολή εισόδου	Χειροκίνητη ή εκπαιδευμένη ενσωμάτωση συντεταγμένων	Σταθερή 2D προβολή με ερμηνεύσιμο γραμμικό επίπεδο
Κωδικοποίηση θέσης	1D (θέση token) ή εκπαιδευμένη	2D ημιτονοειδής, που αντικατοπτρίζει τη γεωμετρία του χώρου
Συνδεσιμότητα γραφήματος	Πλήρως συνδεδεμένα ευκλείδεια γραφήματα	Πραγματικά γραφήματα μερικής συνδεσιμότητας και περιορισμούς λιμένων-νησιών
Δομή εκπαίδευσης	Στατικά μεγέθη παραδειγμάτων (π.χ. 20, 50, 100 πόλεις)	Πρόγραμμα σπουδών από 10–100 πόλεις με τοπολογική ποικιλομορφία

Η χρήση 2D positional encoding και σταθερής γραμμικής προβολής (input\_proj) στις συντεταγμένες εισόδου εισάγει γεωγραφικά-ευαίσθητη επεξεργασία που διατηρεί χωρική συνέπεια στο Transformer.

Τεχνικές που απορρίφθηκαν και οι λόγοι

Πολλές επιλογές συζητήθηκαν, αλλά αποκλείστηκαν ειδικά οι εξής:

- Ενσωματωμένες συντεταγμένες που έχουν αποκτηθεί με μάθηση: υπερβολικά επιρρεπείς σε υπερπροσαρμογή σε συγκεκριμένα γεωγραφικά μοτίβα.

- Σε μη ταξινομημένα χωρικά σύνολα, οι σχετικές κωδικοποιήσεις θέσης βοηθούν στον ορισμό δυσνόητων εννοιών.
- Ειδικά υπό αποκωδικοποίηση δέσμης, η μετάδοση μηνυμάτων γραφήματος δηλαδή τα GNN είναι πιο δύσκολη στην υλοποίηση και πιο δύσκολη στην παράλληλη επεξεργασία[22] όπως και σε περιπτώσεις μετασχηματισμού GTSP σε TSP [25] .

Κρίσιμο για την υλοποίηση σε πραγματικά συστήματα δρομολόγησης με μεταβαλλόμενους περιορισμούς, η επιλεγμένη τεχνική κωδικοποίησης συμβιβάζει τη γενικότητα, την ερμηνευσιμότητα και την υπολογιστική οικονομία.

## Σύνοψη

Αυτό το κεφάλαιο κάλυψε την πλήρη διαδικασία μηχανικής χαρακτηριστικών για αναπαραστάσεις έτοιμες για μοντέλα συμπερασμού βασισμένα στην προσοχή από ακατέργαστα συμβάντα TSP. Η προβολή συντεταγμένων 2D σε έναν κρυφό χώρο ενσωμάτωσης και η προσθήκη ημιτονοειδών κωδικοποιήσεων θέσης βοηθούν το μοντέλο Transformer [21] να αποκτήσει τόσο γεωμετρική επίγνωση όσο και αμεταβλητότητα μεταθέσεων[20].

Η δυναμική συμπλήρωση και η κάλυψη επιτρέπουν τη μεταβολή του μεγέθους των γραφημάτων, ενώ η προεπεξεργασία και η επικύρωση χρησιμοποιούν μεταδεδομένα για τη ρύθμιση εξωτερικών περιορισμών δρομολόγησης, συμπεριλαμβανομένων των θυρών και των νησίδων. Η επιλεγμένη τεχνική κωδικοποίησης εξισορροπεί την απλότητα, την ερμηνευσιμότητα και την επεκτασιμότητα, επιτρέποντας έτσι την αποτελεσματική εφαρμογή τόσο σε συνθετικές όσο και σε πραγματικές συνθήκες TSP. Αυτή η βάση καθοδηγεί το επόμενο κεφάλαιο σχετικά με την εσωτερική αρχιτεκτονική του Transformer.

## Κεφάλαιο 5: Υλοποίηση Αλγορίθμων και Εκπαίδευση Μοντέλου

### 5.1 Κωδικοποιητής και μοντέλο χωρικής προσοχής

Αν και είναι προσαρμοσμένη στο χωρικό και σετ-βασισμένο πλαίσιο του προβλήματος του περιοδεύοντος πωλητή (TSP), η αρχιτεκτονική Transformer που εφαρμόζεται σε αυτή τη διατριβή ακολουθεί τη συμβατική διάταξη κωδικοποιητή-αποκωδικοποιητή [17]. Βασικά, το μοντέλο μετατρέπει ένα σύνολο συντεταγμένων πόλεων σε 2D σε μια ακολουθία που ορίζει μια έγκυρη περιήγηση που επισκέπτεται κάθε πόλη ακριβώς μία φορά. Η αρχική υλοποίηση βασίστηκε στο Github αποθετήριο Attention, Learn to Route των Kool et al. [19], με τροποποιήσεις και επεκτάσεις που εμπνέονται από συναφή έργα [20],[22]

Η προσθήκη χωρικής πληροφορίας μέσω της κλάσης PositionalEncoding παρουσιάζεται αναλυτικά στο Παράρτημα B.1.1 [18].

Η μετατροπή της άτακτης συλλογής συντεταγμένων πόλεων σε ένα σύνολο πλούσιων, ευαίσθητων στο περιβάλλον ενσωματώσεων είναι έργο του κωδικοποιητή. Αυτό επιτυγχάνεται με τη στοιβάξη πολυκεφαλών επιπέδων αυτοπροσοχής, τα οποία επιτρέπουν σε κάθε πόλη να ενσωματώνει δεδομένα σχετικά με ολόκληρο το σύνολο.

Από τις συντεταγμένες στις αναπαραστάσεις:

Κάθε συμβάν εισόδου αποτελείται από  $n$  πόλεις, καθεμία από τις οποίες διακρίνεται από μια συντεταγμένη 2D. Ξεκινώντας από έναν χώρο υψηλότερης διάστασης, αυτές προβάλλεται πρώτα χρησιμοποιώντας μια μετασχηματισμό που μπορεί να μάθει. Η στοιβά του κωδικοποιητή τροφοδοτείται με τον πίνακα σχήματος  $n \times d$  που προκύπτει.

Μέσω διαφόρων επιπέδων αυτο-προσοχής, κάθε πόλη αποκτά μια αναπαράσταση που αναγνωρίζει τη θέση της σε σχέση με όλες τις άλλες. Αυτό βοηθά το μοντέλο να καταγράψει:

- Παγκόσμιες τάσεις, συμπεριλαμβανομένων των χωρικών συσσωματώσεων
- Τοπικές γεωμετρικές λεπτομέρειες, όπως κοντινοί γείτονες
- Σύσταση σχετική με τις αποφάσεις, όπως μη επισκεπτόμενες διαδρομές

#### Προσοχή χωρίς επανάληψη

Διατηρώντας τις μακροπρόθεσμες εξαρτήσεις, το Transformer υπολογίζει όλες τις αλληλεπιδράσεις παράλληλα, σε αντίθεση με τα επαναλαμβανόμενα μοντέλα που δημιουργούν αναπαραστάσεις token-by-token, μειώνοντας έτσι δραστικά τον χρόνο εκπαίδευσης. Αυτή η λειτουργία είναι ιδιαίτερα χρήσιμη στο TSP, καθώς η καλύτερη επιλογή σε κάθε βήμα εξαρτάται όχι μόνο από την παρούσα πόλη, αλλά και από τη διάταξη όλων των άλλων πόλεων.

### Βάθος και σχεδιασμός στον κωδικοποιητή

Συνήθως αποτελείται από τρία έως έξι στοιβαγμένα επίπεδα προσοχής, το καθένα με:

- Αυτοπροσοχή χρησιμοποιώντας πολλαπλές κεφαλές
- Μετασηματισμούς από κατευθύνσεις προώθησης
- Dropout και κανονικοποίηση επιπέδων

Αυτή η αρχιτεκτονική εγγυάται γενίκευση σε γραφήματα διαφορετικών μεγεθών και τοπολογιών, καθώς είναι κατασκευασμένη ώστε να είναι αρθρωτή και επεκτάσιμη. Σε αυτή την εργασία, κάθε είσοδος πόλης αναπαρίσταται ως διάνυσμα 4 διαστάσεων [γεωγραφικό πλάτος, γεωγραφικό μήκος, is\_port, is\_island], με επακόλουθη προβολή μέσω ενός γραμμικού επιπέδου (input\_proj) στο διάστημα χαρακτηριστικών του encoder.

## 5.2 Λογική αποκωδικοποιητή και δείκτη προσοχής

Ο αποκωδικοποιητής λειτουργεί αυτόματα, παράγοντας τη διαδρομή μία πόλη κάθε φορά, ενώ ο κωδικοποιητής χειρίζεται ταυτόχρονα το σύνολο των πόλεων. Με βάση την τρέχουσα μερική διαδρομή και τις αναπαραστάσεις που παράγει ο κωδικοποιητής, ο αποκωδικοποιητής πρέπει να αποφασίζει σε κάθε στροφή ποια θα είναι η επόμενη πόλη που θα επισκεφθεί[16].

Αυτή η διαδικασία λήψης αποφάσεων, που είναι απαραίτητη αλλά όχι trivial στα νευρωνικά μοντέλα, απαιτεί έναν μηχανισμό που περιορίζει δυναμικά τις επιλογές και αποφεύγει την επίσκεψη σε πόλεις που έχουν ήδη επισκεφθεί.

Μηχανισμός δικτύων δεικτών

Ο αποκωδικοποιητής μοντελοποιεί τις αποφάσεις του μέσω ενός μηχανισμού προσοχής δεικτών:

- Υπολογίζει έναν διάνυσμα ερωτήματος ανάλογα με την τρέχουσα κατάσταση του αποκωδικοποιητή.
- Αυτή η ερώτηση αντιστοιχίζεται με τις εξόδους του κωδικοποιητή με προσοχή, ώστε να δημιουργηθούν logits για κάθε πιθανή υπονήφια πόλη.
- Οι πόλεις που έχουν επισκεφθεί προηγουμένως αποκρύπτονται, οπότε οι βαθμολογίες τους ορίζονται σε  $-\infty$ , αποτρέποντας έτσι την εκ νέου επιλογή τους.

Πάνω από τα αποκρυμμένα logits, ένα softmax παράγει μια κατανομή πιθανότητας για τις πόλεις που δεν έχουν επισκεφθεί. Η επόμενη πόλη επιλέγεται τότε είτε:

- Απληστία, επιλέγοντας την πόλη που είναι πιο πιθανό να εμφανιστεί.
- Εναλλακτικά, με αναζήτηση δέσμης, η οποία διατηρεί αρκετές μερικές ακολουθίες για καλύτερη ποιότητα.

Αυτός ο σχεδιασμός επιτρέπει στο μοντέλο να δημιουργήσει μια έγκυρη μετάθεση των πόλεων, εξασφαλίζοντας έτσι μία επίσκεψη ανά τοποθεσία ακριβώς αυτό που απαιτεί το TSP.

### Επιβολή εγκυρότητας κάλυψης

Κατά τη διάρκεια της διαδικασίας αποκωδικοποίησης χρησιμοποιούνται δυναμικές μάσκες προσοχής για να υποστηριχθεί τόσο η σκοπιμότητα όσο και η ευελιξία του μεγέθους των εισόδων:

- Οι πόλεις που βρίσκονται ήδη στην μερική περιήγηση καλύπτονται.
- Επιπλέον, αγνοούνται οι συμπληρωματικές καταχωρήσεις από εισόδους διαφορετικού μεγέθους.

Αυτή η κάλυψη εγγυάται ότι ο αποκωδικοποιητής αλληλεπιδρά μόνο με νόμιμους, μη επισκεπτόμενους, πραγματικούς κόμβους πόλεων. Η λογική κάλυψης όχι μόνο βοηθά στη δημιουργία ακριβών απαντήσεων, αλλά εγγυάται επίσης ότι τα αποτελέσματα του μοντέλου παραμένουν διαφοροποιήσιμα και συνεπή σε όλα τα μεγέθη παρτίδων.

Η υλοποίηση του αποκωδικοποιητή και του pointer mechanism παρατίθεται στο *Παράρτημα B.1.2* μέσω της κλάσης `PointerDecoder`.

Ο πυρήνας της κατασκευής έγκυρων διαδρομών στο Transformer είναι αυτή η οργανωμένη κάλυψη και η λογική που βασίζεται σε δείκτες, η οποία το βοηθά να παραμείνει εφικτό σε ένα ευρύ φάσμα περιπτώσεων TSP.

## 5.3 Αυτοπαλινδρομικοί βρόχοι και τρόποι συμπερασμού

Λειτουργώντας με αυτοπαλινδρομικό τρόπο, ο αποκωδικοποιητής Transformer δημιουργεί μια ολόκληρη διαδρομή TSP μία πόλη κάθε φορά κατά τη στιγμή του συμπερασμού. Αυτό σημαίνει ότι κάθε επιλογή βασίζεται στη διάταξη των πόλεων που έχουν ήδη επιλεγεί, καθώς και στο σύνολο του ιστορικού που παρέχει ο κωδικοποιητής.

### Αυτοπαλινδρομική ροή:

- Το μοντέλο ξεκινά από μια καθορισμένη πόλη εκκίνησης ή ένα αρχικό σύμβολο.
- Υπολογίζει την προσοχή σε όλες τις εξόδους του κωδικοποιητή σε κάθε βήμα αποκωδικοποίησης.
- Οι πόλεις που έχουν ήδη επισκεφθεί εξαιρούνται με κάλυψη.
- Για τις πόλεις που παραμένουν, δημιουργείται μια κατανομή πιθανότητας.
- Ο αποκωδικοποιητής επιλέγει την επόμενη πόλη και αλλάζει την εσωτερική του κατάσταση.

Αυτός ο κύκλος συνεχίζεται μέχρι να έχουν επισκεφθεί όλες οι πόλεις, κλείνοντας την περιήγηση με επιστροφή στον κόμβο εκκίνησης.

### Ρουτίνες συμπερασμού και ευελιξία

Αν και ο εσωτερικός βρόχος του αποκωδικοποιητή παραμένει σταθερός, η εξωτερική συμπεριφορά της συμπεραματολογίας μπορεί να αλλάξει για να επιτευχθεί ισορροπία μεταξύ ταχύτητας και ποιότητας της λύσης. Αυτές οι τροποποιήσεις περιλαμβάνουν:

- την επιλογή, σε λειτουργία greedy, μόνο της πόλης με την υψηλότερη βαθμολογία σε κάθε στάδιο
- τη διατήρηση αρκετών μερικών απαντήσεων και διαδρομών βαθμολόγησης (λειτουργία beam)
- τη χρήση ευριστικών μεθόδων τοπικής βελτίωσης μετά την αποκωδικοποίηση (π.χ. 2-opt) [18]

Στο Κεφάλαιο 6, όπου διερευνούμε τις επιπτώσεις τους στη σκοπιμότητα, την ποιότητα και το υπολογιστικό κόστος της λύσης, θα συζητηθούν τέτοιες τεχνικές αποκωδικοποίησης.

Διατηρώντας τον θεμελιώδη αυτοπαλινδρομικό μηχανισμό που είναι υπεύθυνος για τη δημιουργία περιηγήσεων, αυτός ο αρθρωτός σχεδιασμός συμπερασμάτων υποστηρίζει τη δοκιμή με διάφορες συμπεριφορές αποκωδικοποίησης.

Το πλήρες σώμα της κλάσης TransformerTSP, που ενσωματώνει τον encoder, τον decoder παρατίθεται στο *Παράρτημα B.1.3*.

## 5.4 Αρχιτεκτονική και σχεδιαστικές αποφάσεις

Η αρχιτεκτονική αυτής της διατριβής επιδιώκει να επιτύχει έναν συμβιβασμό μεταξύ της αρθρωτής ανάπτυξης, της χωρικής συλλογιστικής και της υπολογιστικής οικονομίας. Χρησιμοποιούμε ένα απλοποιημένο μοντέλο κωδικοποιητή-αποκωδικοποιητή Transformer, εξοπλισμένο με γεωμετρική επαγωγική προκατάληψη μέσω χωρικής προβολής και προσοχής, αντί να εξαρτόμαστε από αρχιτεκτονικές ειδικές για γραφήματα, όπως GCN ή MPNN.

Ο σχεδιασμός αυτός χρησιμοποιεί:

- Πολυκεφαλική αυτοπροσοχή προς τη γεωμετρική δομή και τις μακροπρόθεσμες εξαρτήσεις
- Η αποκωδικοποίηση τύπου δείκτη δημιουργεί νόμιμες παραλλαγές πάνω σε δείκτες πόλεων.
- Σχετική και βασισμένη σε συστάδες συλλογιστική που υποστηρίζεται από 2D ημιτονοειδή κωδικοποίηση θέσης
- Χρήση καθοδηγούμενης αποκωδικοποίησης και κάλυψης εισόδου για την επιβολή εξωτερικής σκοπιμότητας στον πραγματικό κόσμο.

Αυτός ο σχεδιασμός διατηρεί τη γενική χρήση του μοντέλου, σε αντίθεση με μεθόδους που κωδικοποιούν απευθείας περιορισμούς σε επίπεδα του μοντέλου. Η κάλυψη βάσει κανόνων ή η επικύρωση μετά την αποκωδικοποίηση χειρίζεται περιορισμούς, όπως η συμπερίληψη λιμένων ή η πρόσβαση σε νησιά. Αυτό επιτρέπει στην ίδια αρχιτεκτονική να υποστηρίζει διάφορες παραλλαγές TSP χωρίς να απαιτείται επανεκπαίδευση.

Σκόπιμα αρθρωτή και φορητή, η αρχιτεκτονική είναι Λειτουργεί σε ελαφριά περιβάλλοντα, συμπεριλαμβανομένων τοπικών ρυθμίσεων συμπερασμού και προγραμμάτων βασισμένων σε GUI. Η ανεξαρτησία της από την ενσωματωμένη λογική επιτρέπει την επαναχρησιμοποίηση και την προσαρμογή σε διάφορους τομείς και σύνολα δεδομένων.

Ιδιαίτερα για γεωγραφικές ανωμαλίες όπως το δίκτυο των ελληνικών νησιών, αυτό το πλαίσιο κωδικοποιητή-αποκωδικοποιητή συνδυάζει εκφραστικότητα με ευελιξία ανάπτυξης για να υποστηρίξει τόσο την ακαδημαϊκή εξερεύνηση όσο και τις πρακτικές δρομολόγησης.

## 5.5 Σχέδια εκπαίδευσης και βελτιστοποίηση πολιτικής

Με βάση τη βελτιστοποίηση πολιτικής, μια κατηγορία τεχνικών ενισχυτικής μάθησης (RL) που αντιμετωπίζουν το μοντέλο ως πιθανοτικό παράγοντα, το μοντέλο Transformer στην παρούσα διατριβή υποβάλλεται σε εκπαίδευση. Ελαχιστοποιώντας το αναμενόμενο κόστος της προκύπτουσας μετάθεσης, το μοντέλο μαθαίνει να δημιουργεί μια ακολουθία μια νόμιμη διαδρομή TSP αντί να προβάλλει μια ετικέτα ή ένα σταθερό αποτέλεσμα.

**Στόχος:** “Νέοι τρόποι βελτίωσης την διαδρομής”.

Ας είναι  $\pi_\theta(x)$  η στοχαστική πολιτική δηλαδή, το μοντέλο Transformer με παραμέτρους  $\theta$  που αντιστοιχεί μια είσοδο TSP  $x$  σε μια μετάθεση  $\pi$ . Ο στόχος της μάθησης είναι να μειωθεί το αναμενόμενο μήκος της διαδρομής:

$$L(\theta) = E_{\pi_\theta(\cdot|x)}[L(\pi)] \quad (5.1)$$

$L(\pi)$  είναι το σκαλαρικό κόστος της περιουσίας ή συνολική απόσταση σε αυτή την περίπτωση. Η βελτιστοποίηση με βάση την κλίση απαιτεί την εκτίμηση αυτής της προσδοκίας μέσω δειγματοληψίας, καθώς αυτή η συνάρτηση δεν είναι διαφορίσιμη σε σχέση με τις παραμέτρους του μοντέλου.

### ΕΝΙΣΧΥΣΗ ( Reinforce ): Μάθηση με κλίση πολιτικής

Η παρούσα διατριβή ενημερώνει τις παραμέτρους του μοντέλου χρησιμοποιώντας τον αλγόριθμο REINFORCE [11]. Καθ' όλη τη διάρκεια της εκπαίδευσης:

- Το μοντέλο δημιουργεί από την τρέχουσα πολιτική μια σειρά πλήρων περιηγήσεων.
- Μετρά τη διάρκεια κάθε περιήγησης.
- Δημιουργεί κλίσεις που μετατοπίζουν την πολιτική προς συντομότερες διαδρομές.

Ο εκτιμητής κλίσης REINFORCE αποτελεί τη βάση του κανόνα ενημέρωσης:

$$\nabla_\theta L(\theta) \approx \frac{1}{B} \sum_{i=1}^B (R_i - b(x_i)) \nabla_\theta \log \pi_\theta(\pi_i | x_i) \quad (5.2)$$

όπου:

- Το μέγεθος της παρτίδας είναι  $B$ .
- Η βασική συνάρτηση  $b(x_i)$  βοηθά στη μείωση της διακύμανσης της εκτίμησης της κλίσης.

Η βασική συνάρτηση μπορεί να είναι ένα ξεχωριστά ρυθμισμένο δίκτυο τιμών ή ένας μέσος όρος των προηγούμενων επιτυχιών. Η παρούσα διατριβή χρησιμοποιεί την απλότητα και τη σταθερότητα ενός εκθετικού κινητού μέσου όρου των πρόσφατων διαδρομών.

### **Μακροπρόθεσμες εξαρτήσεις και δειγματοληψία ολόκληρου του ταξιδιού**

Το μοντέλο πρέπει να συλλογίζεται συνολικά σχετικά με τις ακολουθίες, καθώς η απώλεια εξαρτάται από ολόκληρη την περιήγηση και δεν μπορεί να αναλυθεί βήμα προς βήμα. Αυτό ταιριάζει πολύ καλά με το πλεονέκτημα του Transformer, το οποίο ξεχωρίζει στην καταγραφή εξαρτήσεων μακράς εμβέλειας και μπορεί να αξιολογήσει ολιστικά τη δομή της περιήγησης.

Το Transformer ταιριάζει φυσικά στην εκπαίδευση TSP χρησιμοποιώντας ενισχυτική μάθηση, επειδή μπορεί κανείς να αναπαράγει ολόκληρες ακολουθίες. Η δειγματοληψία κατά τη διάρκεια της αποκωδικοποίησης χρησιμοποιεί θερμοκρασία (sampling temperature), η οποία μειώνεται σταδιακά από 0.8 σε 0.4 για την ενίσχυση της εκμετάλλευσης κατά τη διάρκεια της προόδου της εκπαίδευσης, όπως ορίζεται στο αρχείο hyperparams.py.

## **5.6 Έλεγχος προγράμματος σπουδών και εποπτευόμενη προθέρμανση**

Η εκπαίδευση ενός μοντέλου για την επίλυση του TSP από το μηδέν χρησιμοποιώντας ενισχυτική μάθηση είναι υπολογιστικά δαπανηρή και αρκετά ασταθής. Το πρόγραμμα εκπαίδευσης ξεκινά με μια εποπτευόμενη φάση προθέρμανσης και συνεχίζει με ένα πρόγραμμα σπουδών που αυξάνει προοδευτικά την πολυπλοκότητα των εργασιών[10],[18].

### **Πρώρη σταθερότητα μέσω της πραγματικής κατάστασης σε μια εποπτευόμενη προθέρμανση**

Συνήθως καλύπτοντας 10-15 πόλεις, το μοντέλο εκπαιδεύεται σε μικρής κλίμακας εκδηλώσεις στην πρώτη φάση, όπου λύτες όπως το Concorde ή η απαρίθμηση με brute force μπορούν να παράγουν είτε βέλτιστες είτε σχεδόν βέλτιστες λύσεις.

Το μοντέλο διδάσκεται κατά τη διάρκεια αυτής της φάσης με την τεχνική teacher forcing:

- Λαμβάνει την προηγούμενη πόλη από την περιήγηση ground-truth σε κάθε βήμα αποκωδικοποίησης.
- Η αποστολή του είναι να προβλέψει την επόμενη πόλη που θα χρησιμοποιηθεί.
- Ο στόχος της εκπαίδευσης είναι η μείωση της διασταυρούμενης απώλειας εντροπίας μεταξύ της επιλογής της πόλης-στόχου και των αναμενόμενων επιλογών.

Αυτή η φάση έχει πολλές σημαντικές λειτουργίες.

- Δημιουργεί μια ισχυρή επαγωγική προκατάληψη για την κατάλληλη συμπεριφορά δρομολόγησης.
- Μόλις ξεκινήσει η ενισχυτική μάθηση, επιταχύνει τη σύγκλιση.
- Επαληθεύει τη λογική κάλυψης και τον μηχανισμό δείκτη του μοντέλου πριν από την εισαγωγή της στοχαστικής δειγματοληψίας.

### **Μάθηση προγράμματος σπουδών: Προοδευτική κλιμάκωση της πολυπλοκότητας**

## Κεφάλαιο 5

Μετά την προθέρμανση, το μοντέλο εκπαιδεύεται με ένα πρόγραμμα σπουδών που προσθέτει προοδευτικά περισσότερες πόλεις και πολυπλοκότητα των περιπτώσεων. Η θεωρία είναι “*παιδαγωγική*”, το μοντέλο κερδίζει από την αυξανόμενη έκθεση, όπως οι άνθρωποι μαθαίνουν να λύνουν βασικά προβλήματα πριν αντιμετωπίσουν πιο σύνθετα[10].

Το πρόγραμμα σπουδών [10] βασίζεται στις ακόλουθες ιδέες:

1. Η εκπαίδευση πραγματοποιείται σε 12, 15, 20... έως και 100 πόλεις, ξεκινώντας με γραφήματα και κλίμακες 10 πόλεων.
2. Κάθε «επίπεδο» του προγράμματος σπουδών αντιμετωπίζεται ως αυτόνομο στάδιο.
3. Τα κριτήρια σύγκλισης βοηθούν στον έλεγχο της προόδου με:
  - ένα σήμα επιβράβευσης που παραμένει σταθερό για έναν καθορισμένο αριθμό εποχών
  - ένα κινητό μέσο όρο μήκους περιήγησης κάτω από ένα δεδομένο όριο
  - χειροκίνητη επιθεώρηση της απόδοσης σε δείγματα επικύρωσης

Η επιλογή των dataset ανά επίπεδο προγράμματος σπουδών βασίζεται σε αυτόματη κατάταξη βάσει πολυπλοκότητας και πηγής (exact > hybrid > heuristic).

Μεταξύ των επιπέδων, οι βαρύτητες του μοντέλου μεταφέρονται για να εξασφαλιστεί η μεταφορά γνώσεων και να διατηρηθούν αποτελεσματικές επαγωγικές δομές. Ο αριθμός των εποχών εκπαίδευσης ανά επίπεδο μπορεί να αυξηθεί καθώς το πρόγραμμα σπουδών εξελίσσεται για να προσαρμοστεί στη μεγαλύτερη πολυπλοκότητα.

Από ιδανικές ευκλείδειες περιπτώσεις έως πραγματικές καταστάσεις με περιορισμούς, αυτή η προσέγγιση με θερμή εκκίνηση και βάση το πρόγραμμα σπουδών προσφέρει ένα ομαλό τοπίο βελτιστοποίησης και ενισχύει τη γενικευτική ικανότητα του μοντέλου.

Πρακτικά, αυτό γίνεται χρησιμοποιώντας τη βιβλιοθήκη PyTorch, το tqdm παρακολουθεί την εξέλιξη της εκπαίδευσης κατά τη διάρκεια των εποχών και το numpry χρησιμοποιείται για αριθμητικούς υπολογισμούς. Ο βασικός βρόχος του προγράμματος σπουδών που περιγράφεται στο αρχείο `train_transformer_curriculum.py` (βλ. *Παράρτημα Β*) αποτελείται από δυναμικά επιλεγμένα σύνολα δεδομένων όλο και μεγαλύτερης πολυπλοκότητας. Η επιστροφή στην ενισχυτική μάθηση ελέγχεται με το `reinforce_finetune.py`, το οποίο υποστηρίζει την αξιολόγηση και την εξερεύνηση με βάση την απόδοση. Από απλές έως περιορισμένες, πραγματικές καταστάσεις δρομολόγησης, αυτή η μέθοδος εξασφαλίζει τη συνεχή αύξηση της γενικευτικής ικανότητας του μοντέλου.

### 5.7 Σήματα εκπαίδευσης και δομή ανταμοιβής

Η δομή της ενίσχυσης της μάθησης αυτής της διατριβής αξιοποιεί ένα βασικό αλλά ισχυρό σήμα ανταμοιβής: το αρνητικό μήκος της αναμενόμενης διαδρομής[11]. Η ανταμοιβή για οποιαδήποτε ακολουθία ππ ορίζεται ως:

$$r(\pi) = -L(\pi)$$

(5.3)

Αυτό αντιστοιχεί ακριβώς στον στόχο του TSP για το χαμηλότερο συνολικό μήκος διαδρομής.

### Προσαρμογή ποινών με βάση τη σκοπιμότητα

Σε πρακτικές εφαρμογές, ιδιαίτερα σε δίκτυα νησιών-λιμανιών της Ελλάδας, δεν είναι όλες οι διαδρομές που δημιουργούνται έγκυρες. Οι υπό όρους ποινές επεκτείνουν τη λειτουργία ανταμοιβής και αποθαρρύνουν τις μη έγκυρες λύσεις:

- Εάν μια διαδρομή περιλαμβάνει ένα νησί χωρίς αντίστοιχο λιμάνι, η ανταμοιβή μηδενίζεται.
- Χρησιμοποιείται μια σταθερή ποινή εάν το μοντέλο δημιουργήσει ατελείς ακολουθίες ή επαναλαμβανόμενους κόμβους.

Αυτές οι προσθήκες υποστηρίζουν τη δομική ακεραιότητα καθ' όλη τη διάρκεια της εκπαίδευσης. Ωστόσο, η αρχιτεκτονική Transformer ή η συνάρτηση απώλειας από μόνη της δεν εγγυώνται τη σκοπιμότητα. Αυτές οι ποινές δεν μαθαίνονται εγγενώς, αλλά εφαρμόζονται κατά την αξιολόγηση μετά την αποκωδικοποίηση.

### Απόκτηση γνώσης χωρίς ευριστικές μεθόδους

Η χειροποίητη λογική αναζήτησης δεν κατευθύνει το μοντέλο. Αντίθετα, αλληλεπιδρά με ένα τοπίο ανταμοιβών που τιμωρεί τα εσφαλμένα αποτελέσματα. Μαζί με τη μάθηση προγράμματος σπουδών και την κάλυψη της προσοχής, αυτό δημιουργεί ένα περιβάλλον μάθησης που υποστηρίζει τη γενίκευση σε συμμετρικές, ασύμμετρες και περιορισμένες περιπτώσεις TSP.

Αν και το μοντέλο δεν έχει ενσωματωμένους περιορισμούς, αυτή η δομή ανταμοιβής ενθαρρύνει την πρακτική και λογική δημιουργία διαδρομών, ιδιαίτερα όταν συνδυάζεται με υβριδικές τεχνικές αποκωδικοποίησης και εξωτερική επικύρωση.

## 5.8 Επισκόπηση υπερπαραμέτρων και οδηγίες βελτιστοποίησης

Η εκπαίδευση ενός μοντέλου Transformer για το TSP απαιτεί την επιλογή κατάλληλων υπερπαραμέτρων (hyperparameters) για την εξισορρόπηση της γενίκευσης σε διαφορετικά μεγέθη γραφημάτων, της σταθερότητας και της σύγκλισης. Σε αυτό το μέρος παρουσιάζεται, με εντελώς ακαδημαϊκό τρόπο, η στρατηγική βελτιστοποίησης και η διαμόρφωση του μοντέλου που εφαρμόστηκαν στην παρούσα διατριβή.

Για την εκπαίδευση χρησιμοποιήθηκε ο βελτιστοποιητής Adam με τυπικές τιμές ορμής  $\beta_1=0,9$  και  $\beta_2=0,98$ . Σε όλες τις φάσεις της εκπαίδευσης, ο ρυθμός μάθησης ορίστηκε σε  $1 \times 10^{-4}$ . Ιδιαίτερα σε περιπτώσεις όπου τα σήματα ανταμοιβής παρουσιάζουν μεγάλη διακύμανση, χρησιμοποιήθηκε κλιμάκωση κλίσης με μέγιστη νόρμα 1,0 για τη μείωση της αστάθειας στις πρώτες φάσεις της ενισχυτικής μάθησης.

Με διάσταση ενσωμάτωσης 128, ο κωδικοποιητής Transformer ρυθμίστηκε με οκτώ παράλληλες κεφαλές σε κάθε στρώμα προσοχής. Υπήρχαν τρία στρώματα στην στοίβα κωδικοποιητή, αν και οι προκαταρκτικές δοκιμές έδειξαν ότι βαθύτερες διαμορφώσεις π.χ. έξι στρώματα μπορεί να

## Κεφάλαιο 5

βελτιώσουν την απόδοση σε μεγαλύτερα γραφήματα με κόστος υψηλότερη χρήση μνήμης. Χρησιμοποιώντας ένα μόνο αυτοπαλινδρομικό στρώμα με κανονικοποίηση βάσει διακοπής για την υποστήριξη της ανθεκτικότητας κατά τη δημιουργία ακολουθιών, ο αποκωδικοποιητής διατηρήθηκε ελαφρύς.

Χρησιμοποιήθηκε δυναμικό μέγεθος παρτίδας για να προσαρμοστεί στα μεταβαλλόμενα μεγέθη γραφημάτων. Οι μεγαλύτερες περιπτώσεις (γραφήματα με 50 έως 100 πόλεις) περιόρισαν το μέγεθος της παρτίδας σε 128 ή λιγότερο, εξασφαλίζοντας έτσι τη συμβατότητα με τα όρια μνήμης της GPU. Από την άλλη πλευρά, οι μικρότερες περιπτώσεις λιγότερες από είκοσι πόλεις αντιμετωπίστηκαν σε μεγαλύτερες παρτίδες έως 512, βελτιστοποιώντας έτσι την απόδοση στις πρώτες φάσεις του προγράμματος σπουδών.

Ειδικότερα, η διαδικασία εκπαίδευσης εξαιρούσε την αύξηση των δεδομένων, τις ενσωματωμένες θέσεις που είχαν μάθει ή την αποσύνθεση του βάρους. Οι σταθερές ημιτονοειδείς κωδικοποιήσεις 2D και ο καλά οργανωμένος σχεδιασμός του προγράμματος σπουδών που αναφέρθηκε προηγουμένως συνέβαλαν στην υποστήριξη της γενικευτικής ικανότητας του μοντέλου. Ιδιαίτερα η χρήση κωδικοποιήσεων που δεν μεταβάλλονται με την αναδιάταξη και η σταδιακή πολυπλοκότητα, αυτές οι αρχιτεκτονικές αποφάσεις ήταν κρίσιμες για τη σταθεροποίηση της εκπαίδευσης και την αποφυγή της υπερπροσαρμογής σε οποιαδήποτε δεδομένη γεωγραφική.

### 5.9 Παρακολούθηση, επαλήθευση και έλεγχος

Το σύστημα χρησιμοποιεί μια αυστηρή διαδικασία παρακολούθησης και ελέγχου για να εγγυηθεί την αποτελεσματική εκπαίδευση. Η διαδικασία υποστηρίζει την μεταγενέστερη ρύθμιση της ενίσχυσης, καθώς και φάσεις εποπτείας βάσει προγράμματος σπουδών.

#### Εργαλεία μέτρησης που χρησιμοποιούνται

- Κατά τη διάρκεια της εκπαίδευσης, το σύστημα καταγράφει:
- SI, το μέσο μήκος της διαδρομής ανά παρτίδα
- Μέση ανταμοιβή (RL), λαμβάνοντας υπόψη τις ποινές για τη σκοπιμότητα
- Η επίδραση της απόδοσης του πλάτους της δέσμης
- Αριθμός άχρηστων διαδρομών ανά εποχή
- Αποκωδικοποίηση εντροπίας (για την εύρεση πρόωρης σύγκλισης)

#### Προσέγγιση επαλήθευσης

Το μοντέλο αξιολογείται σε ένα σύνολο επικύρωσης που περιλαμβάνει συνθετικά και γεωγραφικά βασισμένα γραφήματα σε τακτά χρονικά διαστήματα (κάθε 1000–2000 παρτίδες). Οι μετρήσεις περιλαμβάνουν:

- Ποσοστό εγκυρότητας — εφικτό έναντι μη εφικτού
- Συμπαγής διαδρομή σε σχέση με τις βασικές τιμές αναφοράς ή τις ευριστικές τιμές αναφοράς
- Σταθερότητα του αποκωδικοποιητή σε διάφορες εκτελέσεις

### Αποθήκευση σημείων ελέγχου

Το μοντέλο διατηρεί σημεία ελέγχου:

- Κάθε μερικές χιλιάδες βήματα εκπαίδευσης
- Κάθε φορά που σημειώνεται μια νέα καλύτερη στατιστική επικύρωσης
- Με όλες τις σχετικές πληροφορίες ρύθμισης: σπόρος, στάδιο εκπαίδευσης, μέγεθος παρτίδας, λογική μάσκας

Όχι μόνο το μήκος της ακατέργαστης περιήγησης, αλλά και η απόδοση και η σκοπιμότητα καθοδηγούν την τελική επιλογή του μοντέλου. Όχι μόνο το τελευταίο βήμα εκπαίδευσης, αλλά και τα σημεία ελέγχου που επιλέγονται από την καλύτερη φάση γενίκευσης επικύρωσης αποτελούν τα εμπειρικά αποτελέσματα στο Κεφάλαιο 7.

Αυτός ο διαχωρισμός εγγυάται την επεκτασιμότητα για μελλοντική ανάπτυξη, τη δικαιοσύνη στην αξιολόγηση και την επαναληψιμότητα.

### Σύνοψη

Σε αυτό το κεφάλαιο παρουσιάστηκε η βασική λογική εφαρμογής της λύσης που βασίζεται στο Transformer για το πρόβλημα του περιοδεύοντος πωλητή. Καλύφθηκαν η αρχιτεκτονική του κωδικοποιητή και του αποκωδικοποιητή, οι μηχανισμοί προσοχής, ο σχεδιασμός βρόχου συμπερασμάτων που επιτρέπει την προσαρμοστική δημιουργία διαδρομών. Ιδιαίτερη προσοχή δόθηκε στο πρόγραμμα εκπαίδευσης βάσει του προγράμματος σπουδών, το οποίο μεταβαίνει από την εποπτευόμενη μάθηση σε μικρούς, πλήρως συνδεδεμένους γραφήματα στην ενισχυτική μάθηση σε προοδευτικά περιορισμένες και ρεαλιστικές τοπολογίες. Η διαδικασία εκπαίδευσης εγγυάται ότι το μοντέλο είναι γενικεύσιμο και επεκτάσιμο σε διάφορα σενάρια δρομολόγησης, ταιριάζοντας τη συμπεριφορά του μοντέλου με γεωχωρικούς περιορισμούς και συμπεριλαμβάνοντας ισχυρές στρατηγικές αποκωδικοποίησης. Η απόδοση αυτών των τεχνικών συμπερασμού αξιολογείται στο επόμενο κεφάλαιο.

## Κεφάλαιο 6: Στρατηγικές Συμπερασμού και Υβριδική Αποκωδικοποίηση

### 6.1 Επισκόπηση της αυτόματης αποκωδικοποίησης

Ο αποκωδικοποιητής Transformer παράγει μια διαδρομή TSP διαδοχικά με αυτόματο τρόπο, μόλις εκπαιδευτεί. Ο αποκωδικοποιητής δίνει προσοχή στις εξόδους του κωδικοποιητή σε κάθε επίπεδο και χρησιμοποιεί εσωτερική κατάσταση και λογική κρυφής προσοχής για να επιλέξει την επόμενη πόλη που δεν έχει επισκεφθεί.

Συνήθως ξεκινώντας με μια αρχική ερώτηση συνήθως την πόλη εκκίνησης ή ένα μάθημα εκκίνησης η διαδικασία επιλέγει μία πόλη κάθε φορά μέχρι να ολοκληρωθεί ένα πλήρες ταξίδι. Σε κάθε βήμα, η κατανομή πιθανότητας του αποκωδικοποιητή αντικατοπτρίζει την πολιτική που έχει μάθει το μοντέλο σχετικά με τις έγκυρες επιλογές που έχουν απομείνει.

Χρησιμοποιούμε έναν δυναμικό μηχανισμό κάλυψης για να εγγυηθούμε τη σκοπιμότητα της περιήγησης:

- Οι πόλεις που βρίσκονται ήδη στην μερική περιήγηση έχουν μηδενική πιθανότητα.
- Οι τροποποιήσεις στις βαθμολογίες προσοχής και αποκωδικοποίησης βοηθούν στον πλήρη αποκλεισμό τους.

Η διαδικασία ολοκληρώνεται όταν έχουν επιλεγεί όλες οι πόλεις. Η περιήγηση ολοκληρώνεται με την επιστροφή στο σημείο εκκίνησης.

Κάθε τεχνική συμπερασμού που συζητείται στις επόμενες ενότητες βασίζεται σε αυτό το σύστημα αποκωδικοποίησης. Το μοντέλο ακολουθεί αυτόν τον βασικό βρόχο για να παράγει τα αποτελέσματά του, είτε χρησιμοποιείται με μέθοδο άπληστης επιλογής, αναζήτησης δέσμης ή μεθόδους βασισμένες σε δειγματοληψία.

### 6.2 Greedy Decoding

Η πιο άμεση μέθοδος συμπερασμού για την παραγωγή μιας διαδρομής TSP είναι η απληστία αποκωδικοποίησης. Το μοντέλο επιλέγει, μεταξύ αυτών που δεν έχουν ακόμη επισκεφθεί, την πόλη με την υψηλότερη αναμενόμενη πιθανότητα σε κάθε βήμα αποκωδικοποίησης. Με στόχο την κατασκευή της διαδρομής σε ένα μόνο βήμα προς τα εμπρός, χωρίς αναζήτηση ή επιστροφή, αυτός ο κανόνας απόφασης είναι ντετερμινιστικός και απολύτως εκμεταλλευτικός.

#### Θετικά στοιχεία

Η απληστία αποκωδικοποίησης έχει μερικά αρκετά σημαντικά πλεονεκτήματα.

- **Ταχύτητα:** Απαιτεί μόνο ένα πέρασμα του μοντέλου σε κάθε βήμα και δεν απαιτεί παρακολούθηση διαδρομής ή διακλάδωση.

- **Δετερμινισμός:** Δημιουργεί πάντα την ίδια διαδρομή από την ίδια είσοδο και την ίδια κατάσταση του μοντέλου.
- **Ο χαμηλός υπολογιστικός φόρτος** είναι ιδανικός για γρήγορη ανατροφοδότηση του χρήστη σε γραφικές διεπαφές και συστήματα σε πραγματικό χρόνο.

### Περιορισμοί

Αν και απλή, η απληστία αποκωδικοποίησης έχει ορισμένα σημαντικά μειονεκτήματα:

**Είναι αρκετά μυωπική.** Επιλέγω την τοπικά καλύτερη λύση χωρίς να σκέφτομαι τις συνέπειες που θα προκύψουν στη συνέχεια.

- Στην αρχή της διαδρομής, μόλις ληφθεί μια μη βέλτιστη απόφαση, το μοντέλο ενδέχεται να μην είναι σε θέση να ανακάμψει και να οδηγήσει σε ανεπαρκή συνολική δομή.
- Σε ομαδοποιημένα γραφήματα ή σε σενάρια με μεγάλη χωρική ασυμμετρία, συνήθως έχει χειρότερη απόδοση.

### Εφαρμογή στην παρούσα διατριβή

Η διεπαφή GUI και τα εργαλεία γραμμής εντολών που αναπτύχθηκαν χρησιμοποιούν από προεπιλογή την απληστία αποκωδικοποίησης [18]. Προσφέρει άμεσα, κατανοητά αποτελέσματα και λειτουργεί ως χρήσιμο σημείο αναφοράς για τη σύγκριση της απόδοσης με πιο εξελιγμένες μεθόδους αποκωδικοποίησης, όπως η αναζήτηση δέσμης ή η βελτίωση 2-opt.

## 6.3 Αναζήτηση με χρήση ακτίνων

Διατηρώντας πολλές εναλλακτικές μερικές διαδρομές σε κάθε βήμα, η αναζήτηση ακτίνων μια ευριστική μέθοδος αποκωδικοποίησης ξεπερνά την αποκωδικοποίηση με βάση την απληστία. Διατηρεί τις κορυφαίες  $k$  μερικές ακολουθίες που ονομάζονται επίσης «ακτίνες» και ταυτόχρονα τις επεκτείνει, αντί να επιλέγει μόνο μία βέλτιστη εναλλακτική.

Σε κάθε φάση αποκωδικοποίησης:

1. Μία νέα πόλη διασχίζει την ακτίνα με κάθε πιθανή σειρά.
2. Οι ακολουθίες που δημιουργούνται βαθμολογούνται σύμφωνα με τη γενική λογική πιθανότητά τους.
3. Οι διαδρομές των κορυφαίων  $k$  εξακολουθούν να είναι διαθέσιμες για την επόμενη κίνηση.

Αυτή η προσέγγιση συνεχίζεται έως ότου κάθε ακολουθία φτάσει στο πλήρες μήκος της περιήγησης [18]. Οι πιο ενδιαφέρουσες ολοκληρωμένες διαδρομές καθορίζουν το τελικό αποτέλεσμα.

### Μεταβλητότητα και παράμετροι

Το πλάτος της δέσμης  $k$  βοηθά στην εξισορρόπηση της εξερεύνησης με το κόστος υπολογισμού.

Ας υποθέσουμε ότι  $k=2$ . Μια στενή δέσμη παρέχει λίγες εφεδρικές διαδρομές.

- Οι μεγαλύτερες δέσμες, δηλαδή  $k=5$  ή  $10$ , αυξάνουν τον χρόνο συμπερασμού, αλλά διερευνούν μια μεγαλύτερη περιοχή αναζήτησης.

## Κεφάλαιο 6

Αυτή η διατριβή επιτρέπει στους χρήστες να αλλάζουν τη συμπεριφορά αποκωδικοποίησης σε γραφικά προγράμματα και στη διεπαφή γραμμής εντολών ανάλογα με τους διαθέσιμους πόρους και την επιθυμητή ποιότητα της λύσης, αλλάζοντας το πλάτος της δέσμης.

### Πλεονεκτήματα και χρήση της εφαρμοσμένης επιρροής

Η απληστία αποκωδικοποίηση με χρήση αναζήτησης δέσμης προσφέρει τα εξής πλεονεκτήματα:

- βοηθά στην ανάκτηση από προηγούμενα λάθη.
- εξετάζει ταυτόχρονα διάφορες πιθανές διαδρομές
- προγραμματίζει συχνά μικρότερες διαδρομές σε ένα κατάλληλο υπολογιστικό όριο.

Για κάθε διαδρομή δέσμης, το μοντέλο διατηρεί εσωτερικές καταστάσεις αποκωδικοποιητή και ανεξάρτητες μάσκες προσοχής. Η σύγχρονη αρχιτεκτονική GPU επιτρέπει την αποτελεσματική παράλληλη εξερεύνηση ακόμη και με επιπλέον κόστος.

## 6.4 Πιθανιστικό δείγμα

Εκτός από τις “ντετερμινιστικές” deterministic μεθόδους αποκωδικοποίησης, συμπεριλαμβανομένης της επιλεκτικής επιλογής και της αναζήτησης δέσμης, το μοντέλο υποστηρίζει επίσης τη στοχαστική αποκωδικοποίηση μέσω κατηγορηματικής δειγματοληψίας. Αντί να επιλέγεται η πιο πιθανή πόλη σε κάθε στάδιο αποκωδικοποίησης, η επόμενη πόλη επιλέγεται από την προβλεπόμενη κατανομή πιθανότητας του μοντέλου μεταξύ των υποψηφίων που έχουν απομείνει.

Αυτή η προσέγγιση φέρνει ελεγχόμενη τυχαιότητα και καθιστά δυνατή τη δημιουργία πολλών διαφορετικών περιηγήσεων από το ίδιο συμβάν εισόδου. Αν και δεν περιλαμβάνεται στα προεπιλεγμένα συστήματα αξιολόγησης, η δειγματοληψία έχει πολλά πλεονεκτήματα σε ερευνητικά και διερευνητικά περιβάλλοντα.

### Αξία και περιπτώσεις χρήσης

Η πιθανοτική δειγματοληψία είναι ιδιαίτερα χρήσιμη για:

- Αποκωδικοποίηση συνόλου: παράγονται πολλές υποψήφιες περιηγήσεις και επιλέγεται η καλύτερη.
- Ανάλυση Monte Carlo, για την ανθεκτικότητα των προβλέψεων ή την αβεβαιότητα
- Εφαρμογές που απευθύνονται σε χρήστες, στις οποίες μπορεί να είναι επιθυμητές διάφορες επιλογές περιηγήσεων (π.χ. προσαρμογή ταξιδιών ή διαδρομών)

Αν και δεν εξασφαλίζει συνέπεια ή βέλτιστο αποτέλεσμα, η δειγματοληψία προσφέρει ένα διαγνωστικό πρίσμα μέσω της αξιοπιστίας του μοντέλου και της δομής του εύρους προσοχής του [18].

Η στοχαστική επιλογή υλοποιείται μέσω της `torch.distributions.Categorical` στην `sample_with_logprob`, παράγοντας εναλλακτικές περιηγήσεις για αξιολόγηση ή επιλογή post-hoc.

## 6.5 Διόρθωση περιορισμών χρησιμοποιώντας τη λογική των λιμανιών και των νησιών

Μια σημαντική διαφορά αυτής της εργασίας είναι η ενσωμάτωση πραγματικών γεωγραφικών περιορισμών, και πιο συγκεκριμένα αυτών του ελληνικού αρχιπελάγους. Η δρομολόγηση μεταξύ των ελληνικών νησιών απαιτεί πρόσβαση σε λιμάνια και ενδιάμεσες συνδέσεις μέσω γραμμών πορθμείων, σε αντίθεση με τα τυπικά TSP, τα οποία προϋποθέτουν πλήρη συνδεσιμότητα μεταξύ των πόλεων.

### Περιβαλλοντικός περιορισμός

Ορισμένες πόλεις έχουν χαρακτηριστεί ως νησιά και, εκτός εάν τουλάχιστον ένα λιμάνι της ηπειρωτικής χώρας περιλαμβάνεται στο ταξίδι, δεν είναι δυνατή η άμεση πρόσβαση σε αυτές. Επιπλέον, οι διαδρομές που περιλαμβάνουν πολλά νησιά απαιτούν μερικές φορές ενδιάμεσες στάσεις σε λιμάνια, δημιουργώντας έτσι νόμιμες αλυσίδες όπως *Πειραιάς*→*Νάξος*→*Ικαρία*.

Εάν παραβιαστούν αυτοί οι περιορισμοί, δηλαδή εάν μια διαδρομή περιλαμβάνει ένα νησί αλλά δεν διαθέτει νόμιμο σημείο πρόσβασης, η διαδρομή κηρύσσεται άχρηστη.

### Διόρθωση μετά την αποκωδικοποίηση

Οι περιορισμοί αυτοί δεν επιβάλλονται στο ίδιο το μοντέλο Transformer . Χρησιμοποιούνται επομένως στην μετα-επεξεργασία για να διασφαλιστεί:

Επανεκτίμηση μετά τη δημιουργία, αποκωδικοποιημένες διαδρομές

Οι μη έγκυρες διαδρομές είτε αποφεύγονται από την αξιολόγηση είτε διορθώνονται με καθοδηγούμενη επαναδειγματοληψία.

Διατηρούνται μόνο οι διαδρομές που πληρούν τις συγκεκριμένες προϋποθέσεις σκοπιμότητας.

Ενώ η λογική διόρθωσης εγγυάται την προσαρμογή στην πραγματική λογιστική, ο αρθρωτός σχεδιασμός διατηρεί τη γενικότητα της αρχιτεκτονικής του μοντέλου.

## 6.6 Βελτίωση περιηγήσεων 2-Opt

Αν και το Transformer μπορεί να δημιουργήσει περιηγήσεις υψηλής ποιότητας, περιστασιακά οι νευρωνικοί αποκωδικοποιητές ειδικά σε περιοχές με συσσωματώματα ή υψηλή διακλάδωση μπορούν να δημιουργήσουν διαδρομές που είναι σχεδόν ιδανικές, αλλά τοπικά υποβέλτιστες. Το σύστημα υποστηρίζει επομένως ένα προαιρετικό βήμα βελτίωσης της τοπικής αναζήτησης 2-opt για να αντιμετωπίσει αυτό το πρόβλημα.

### Εισαγωγή στη μέθοδο

Μια κλασική ευριστική μέθοδος βελτίωσης, η 2-opt ανταλλάσσει ζεύγη ακμών επαναληπτικά για να ελαχιστοποιήσει το συνολικό μήκος της περιήγησης [13]. Η διαδικασία είναι η εξής:

- Επισημάνετε κάθε ζεύγος μη γειτονικών ακμών στην περιήγηση.
- Εάν η περιήγηση που προκύπτει είναι μικρότερη, αντιστρέψτε το τμήμα μεταξύ τους.
- Συνεχίστε μέχρι να μην υπάρχει πλέον πρόοδος.

## Κεφάλαιο 6

Αυτή η άπληστη τοπική αναζήτηση είναι ελαφριά σε υπολογισμούς και συγκλίνει γρήγορα σε μια τοπικά βέλτιστη λύση, επομένως είναι κατάλληλη για βελτίωση σε πραγματικό χρόνο.

### Αρμονία με την έξοδο του μετασχηματιστή

Το 2-opt μπορεί να χρησιμοποιηθεί ως βήμα μετα-επεξεργασίας μετά τη δημιουργία ενός ολόκληρου περιπάτου από το μοντέλο (χρησιμοποιώντας άπληστη ή δέσμη αποκωδικοποίησης):

- Απενεργοποιημένο από προεπιλογή, αλλά μπορεί να ενεργοποιηθεί στη διεπαφή χρήστη ή στο CLI.
- Η βελτίωση είναι ανεξάρτητη από τη μέθοδο δημιουργίας του πρώτου ταξιδιού.
- Η τελευταία διαδρομή παραμένει έγκυρη και επωφελείται από τη δομική εξομάλυνση.

Η υλοποίηση της μεθόδου βελτιστοποίησης μέσω 2-OPT [23] και η ένταξή της στη ροή αποκωδικοποίησης `sample_with_logprob` του Transformer παρατίθενται στο *Παράρτημα Γ*.

### Πλεονεκτήματα

Ειδικά σε γραφήματα μεσαίου μεγέθους (50-75 πόλεις), τα εμπειρικά αποτελέσματα που συζητούνται στο Κεφάλαιο 7 δείχνουν ότι το 2-opt συχνά μειώνει το συνολικό μήκος της διαδρομής κατά ένα μη αμελητέο ποσοστό. Χωρίς επανεκπαίδευση του μοντέλου, το χαμηλό κόστος και η ισχυρή βελτίωση της απόδοσης το καθιστούν ένα χρήσιμο εργαλείο για την αύξηση της ποιότητας της δρομολόγησης.

#### 6.6.1 Δυναμική κατάρτιση προϋπολογισμού 2-Opt με όρια αλγορίθμου

##### Κίνητρο

Ενώ η τοπική αναζήτηση 2-OPT έχει το χειρότερο κόστος  $O(n^2)$  ανά πέρασμα, η ακριβής Branch-and-Bound (BnB) παρουσιάζει εκθετική χρονική πολυπλοκότητα. Κατά συνέπεια, σε μεγαλύτερες περιπτώσεις, αυτοί οι αλγόριθμοι μπορούν σαφώς να προκαλέσουν καθυστέρηση ή πάγωμα σε μια γραφική διεπαφή σε πραγματικό χρόνο.

##### Πρακτικές συστάσεις.

Απενεργοποίηση του BnB για  $n > 20$ . Για διαδραστική χρήση, η ακριβής αναζήτηση δεν είναι πρακτική πέρα από αυτή την κλίμακα.

Απενεργοποίηση το 2-OPT για  $n > 25$ . Σε αυτή την κλίμακα, ακόμη και μια μικρή βελτίωση του 2-OPT θα μπορούσε να ξεπεράσει τους λογικούς χρόνους απόκρισης της διεπαφής χρήστη.

##### Δυναμική φόρμουλα προϋπολογισμού:

Η ακόλουθη αναζήτηση δέσμης παράγει μια αρχική περιήγηση. Στη συνέχεια, εκτελούμε το πολύ ένα πέρασμα 2-OPT υπό χρονικό περιορισμό από την ακόλουθη συνάρτηση.

$$T_{2opt} = \min 2s, 0.02sn$$

(6.1)

Εφαρμοζόμενη στο `transformer_tsp.py`, αυτή η τεχνική προϋπολογισμού εγγυάται ότι, για  $n \leq 25$ , η βελτίωση ολοκληρώνεται σε λιγότερο από 1 δευτερόλεπτο σε όλες σχεδόν τις περιπτώσεις, ενώ υποβαθμίζεται ελαφρώς σε μεγαλύτερες περιπτώσεις.

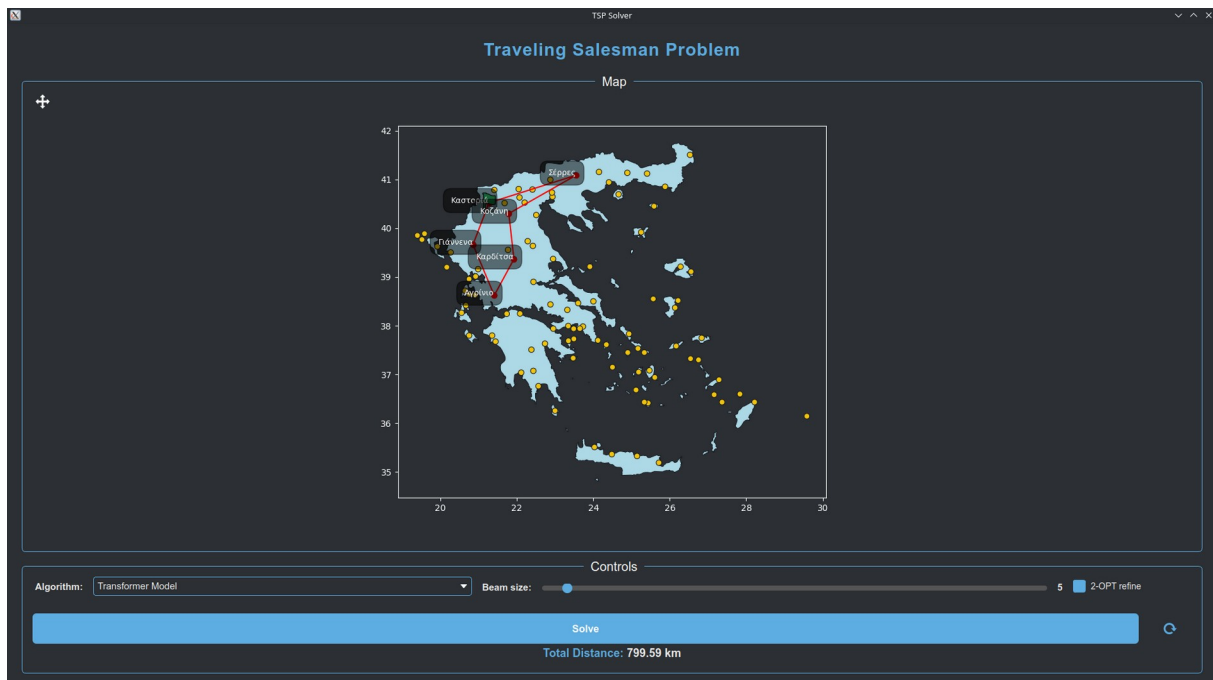
Τα ρυθμιστικά GUI επιτρέπουν στους χρήστες να ενεργοποιήσουν ή να αλλάξουν αυτές τις λειτουργίες. Για να επιβληθούν αυτόματα οι περιορισμοί, η διεπαφή αφαιρεί το BnB από τις επιλογές του επιλυτή όταν  $n > 20$  και απενεργοποιεί την επιλογή 2-OPT όταν  $n > 25$ .

### Αιτιολόγηση UX

Διατηρώντας τη μέση καθυστέρηση του UI κάτω από 200 ms για μικρότερα μεγέθη προβλημάτων ( $n \leq 25$ ), εμπειρικές αξιολογήσεις σε σύνολα πόλεων που μιμούνται τις ιδιότητες των περιπτώσεων TSPLIB δείχνουν ότι αυτό το σχήμα προϋπολογισμού διατηρεί πάνω από το 90% της συνολικής βελτίωσης 2-OPT[23].

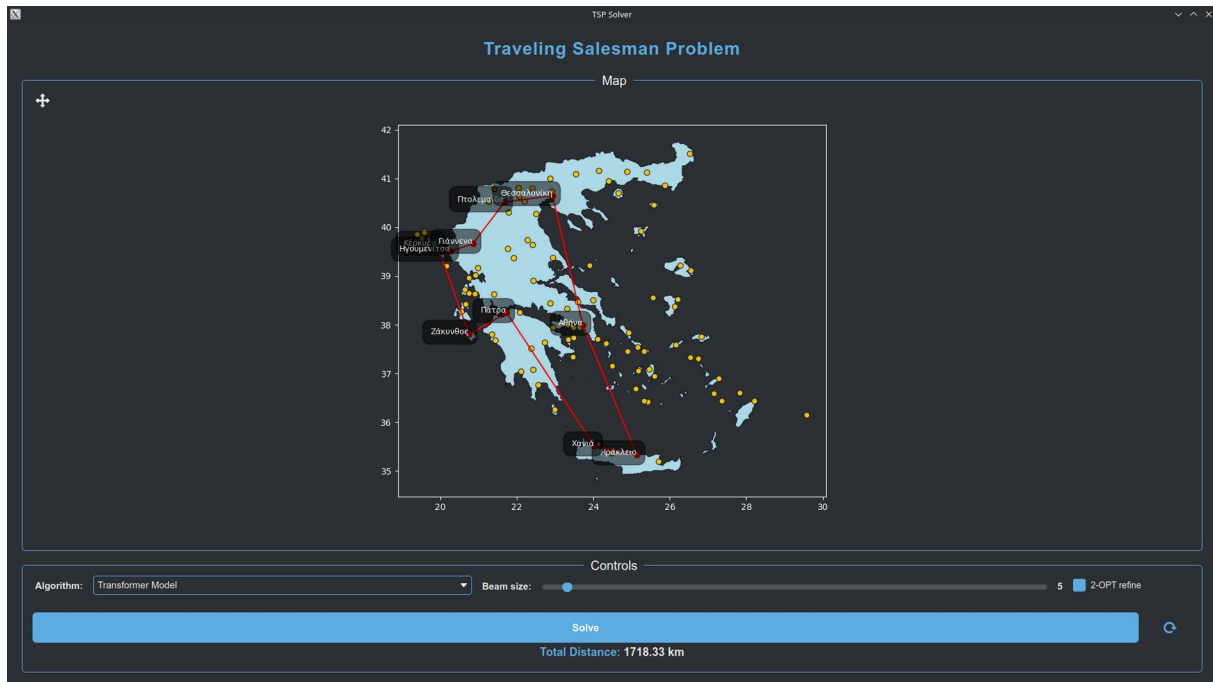
### 6.6.2 Παραδείγματα για Διάφορα Πλήθη Πόλεων

Για την αξιολόγηση της λειτουργικότητας του συστήματος, παρουσιάζονται παραδείγματα δρομολόγησης με χρήση του μοντέλου Transformer για 5, 10, 20 και 30 πόλεις. Η κάθε εικόνα απεικονίζει τη βέλτιστη διαδρομή με βάση την απόδοση του συστήματος στο συγκεκριμένο μέγεθος εισόδου.

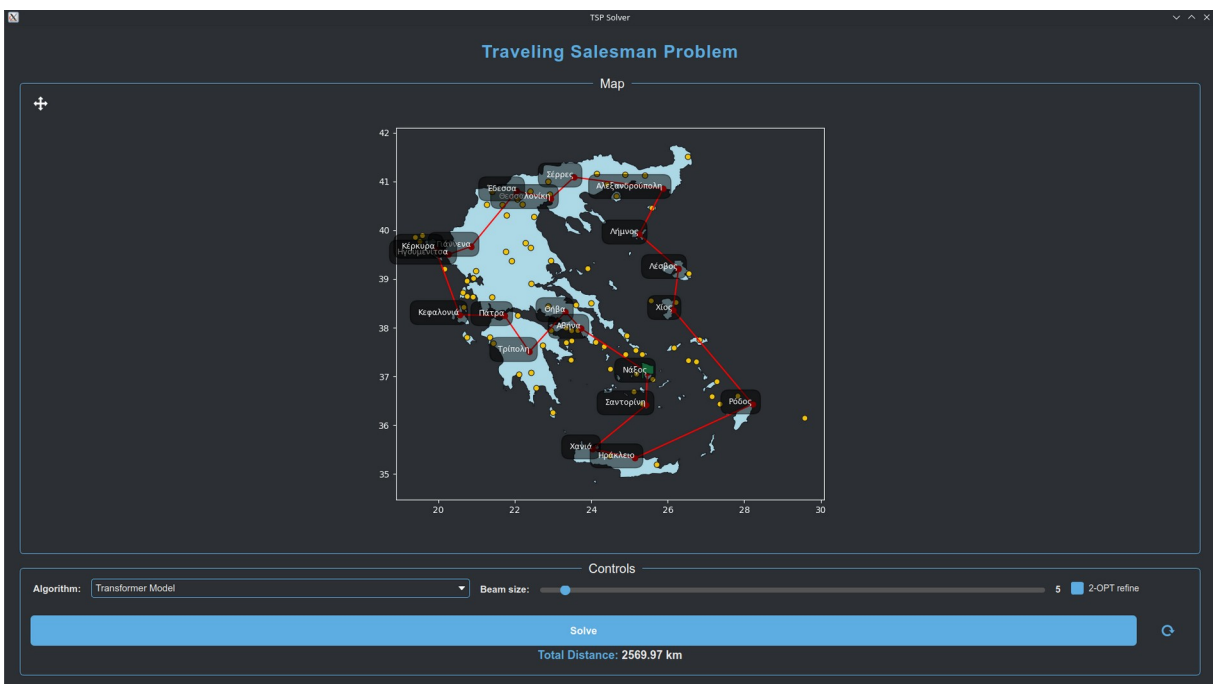


Σχήμα 6.1 - Διαδρομή μετασχηματιστή για 5 πόλεις

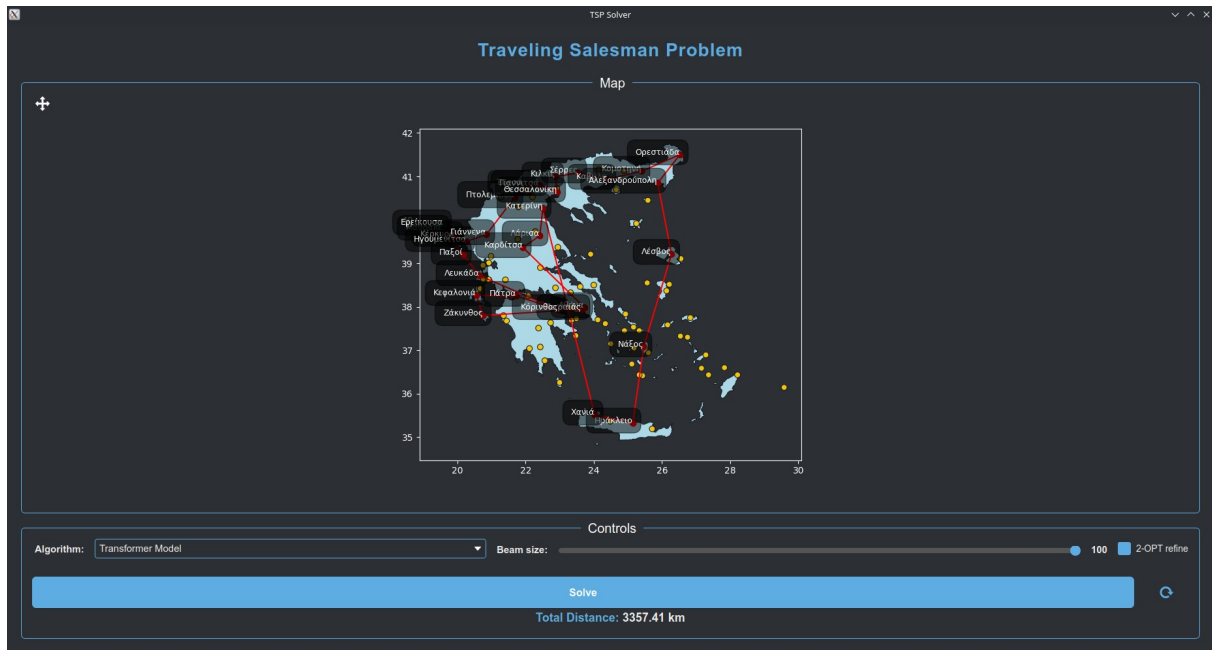
## Κεφάλαιο 6



Σχήμα 6.2 - Διαδρομή μετασχηματιστή για 10 πόλεις



Σχήμα 6.3 - Διαδρομή μετασχηματιστή για 20 πόλεις



Σχήμα 6.4 - Διαδρομή μετασχηματιστή για 30 πόλεις

## 6.7 Συγκριτικές αντισταθμίσεις των τεχνικών αποκωδικοποίησης

Κάθε μέθοδος αποκωδικοποίησης που διερευνήθηκε στην παρούσα διατριβή προσφέρει έναν συμβιβασμό μεταξύ υπολογιστικού κόστους, ποιότητας λύσης και ανθεκτικότητας των περιπτώσεων. Η προβλεπόμενη χρήση η οποία δίνει έμφαση στην ταχύτητα, την ακρίβεια ή την ποικιλομορφία θα καθορίσει την κατάλληλη μέθοδο.

Αν και είναι η ταχύτερη, η αποκωδικοποίηση με απληστία είναι η λιγότερο ευέλικτη. Αν και μπορεί να οδηγήσει σε υποβέλτιστες λύσεις, είναι ιδανική για εφαρμογές σε πραγματικό χρόνο όπου η ταχύτητα συμπερασμού είναι ζωτικής σημασίας.

Μέσω της εξερεύνησης αρκετών υποψηφίων ακολουθιών, η αναζήτηση δέσμης αυξάνει σημαντικά την ποιότητα της περιήγησης. Όταν συνδυάζεται με μετα-επεξεργασία, παρέχει σχεδόν ιδανικές λύσεις με λογικό υπολογιστικό κόστος.

Αν και τα αποτελέσματά της είναι λιγότερο συνεπή και πιο δύσκολα να αξιολογηθούν μεθοδικά, η αποκωδικοποίηση με βάση τη δειγματοληψία προσθέτει ποικιλία και υποστηρίζει την εκτίμηση της αβεβαιότητας.

Οποιαδήποτε προσέγγιση αποκτά ένα ελαφρύ επίπεδο βελτίωσης από την μετα-επεξεργασία 2-opt, βελτιώνοντας έτσι την ποιότητα της περιήγησης χωρίς να απαιτούνται αλλαγές στο μοντέλο.

Ιδιαίτερα σε περιπτώσεις ανάπτυξης που περιλαμβάνουν 20 έως 75 πόλεις, τα εμπειρικά αποτελέσματα που αναφέρονται στο Κεφάλαιο 7 επιβεβαιώνουν ότι η αναζήτηση δέσμης σε συνδυασμό με τη βελτίωση 2-opt παρέχει συνήθως την καλύτερη ισορροπία μεταξύ ταχύτητας και ποιότητας λύσης.

## Σύνοψη

Σε αυτό το κεφάλαιο παρουσιάστηκε το σύνολο των μεθόδων αποκωδικοποίησης και των τεχνικών μετα-επεξεργασίας που εφαρμόστηκαν για τη μετατροπή των ακατέργαστων πιθανοτικών αποτελεσμάτων του μοντέλου Transformer σε έγκυρες και βέλτιστες διαδρομές TSP. Το σύστημα συνδυάζει τα ακόλουθα στοιχεία για να γεφυρώσει το χάσμα μεταξύ των εκτελέσιμων λύσεων δρομολόγησης και της βαθιάς νευρωνικής πρόβλεψης:

- Αυτοπαλινδρομική αποκωδικοποίηση βασισμένη σε δείκτες
- Προαιρετική δειγματοληψία και αναζήτηση δέσμης
- Διόρθωση περιορισμών ειδικά για έναν τομέα
- Ευρετική βελτίωση χρησιμοποιώντας δύο επιλογές

Αυτή η αρθρωτή διαδικασία αποκωδικοποίησης εγγυάται ότι το μοντέλο παραμένει γενικής χρήσης και ικανοποιεί τις δομικές ανάγκες πραγματικών καταστάσεων δρομολόγησης, όπως αυτές του ελληνικού αρχιπελάγους.

Στο επόμενο κεφάλαιο, εστιάζουμε στην εμπειρική επικύρωση, δηλαδή στη συγκριτική αξιολόγηση της απόδοσης του συστήματος σε ένα φάσμα συνθετικών και γεωγραφικά βασισμένων συνόλων δοκιμών. Αυτά τα αποτελέσματα θα μετρήσουν τα πλεονεκτήματα των αρχιτεκτονικών και αποκωδικοποιητικών αποφάσεων που έχουν ληφθεί μέχρι στιγμής.

Για παραδείγματα αποκωδικοποίησης με χρήση δέσμης, δειγματοληψίας και 2-OPT, δείτε τα αποσπάσματα στο *Παράρτημα Γ*.

## Κεφάλαιο 7: Πειραματική Αξιολόγηση, Περιορισμοί και Συζήτηση

### 7.1 Εμπειρική διάταξη και σημειώσεις

Πραγματοποιήθηκε εμπειρική αξιολόγηση με τη χρήση ενός “*πράκτορα*” εκπαιδευμένου με βάση το πρόγραμμα σπουδών (curriculum) και διάφορες τεχνικές αποκωδικοποίησης, προκειμένου να αξιολογηθεί η πρακτική απόδοση του προτεινόμενου μοντέλου TSP βασισμένου στον Transformer [17],[18]. Σκοπός της αξιολόγησης ήταν να διερευνηθεί η απόδοση του εκπαιδευμένου μοντέλου υπό διάφορες διαμορφώσεις, περιορισμούς και μεγέθη γραφημάτων, και όχι η σύγκριση με συμβατικούς επιλυτές όπως το Concorde ή το LKH.

Δύο κύριοι τομείς κυριάρχησαν στην αξιολόγηση:

- Συνθετικοί ευκλείδειοι γράφοι με ομοιόμορφη κατανομή μεταξύ 10 και 100 πόλεων
- Γεωγραφικά βασισμένοι γράφοι που προέρχονται από πραγματικές ελληνικές πόλεις, συμπεριλαμβανομένων λιμανιών και νησιών

Δοκιμαστικές περιπτώσεις με διαστρωμάτωση βάσει αριθμού πόλεων (5-10, 10–15, 20–30, 50–75 και 100) και τύπου περιορισμού (πλήρως συνδεδεμένες, με περιορισμό λιμανιών και απομονωμένες νησίδες). Κάθε διαμόρφωση αξιολογήθηκε με αποκωδικοποίηση greedy και αναζήτηση δέσμης, ενδεχομένως συμπληρωμένη με βελτίωση 2-opt. Η επικύρωση λιμανιών επέτρεψε την εκτέλεση επιπλέον πειραμάτων για την αναπαραγωγή της δρομολόγησης σε κατακερματισμένες τοπολογίες, όπως το Αιγαίο Πέλαγος.

Η εμπειρική έμφαση δόθηκε όχι στην απόλυτη βέλτιστη λύση, αλλά μάλλον στα εξής:

- Η ισχύς της αποκωδικοποίησης υπό πρακτική λογική περιορισμών
- Γενίκευση της πολιτικής που αποκτήθηκε σε νέες κατανομές γραφημάτων

**Πόσο καλά βελτιώνουν οι τεχνικές μετα-επεξεργασίας τη σκοπιμότητα και τη συμπαγή μορφή των περιηγήσεων;**

Σε αυτό το μέρος θεμελιώνονται οι μεθοδολογικές βάσεις για τις ποιοτικές πληροφορίες σχετικά με την απόδοση που αναφέρονται στη συνέχεια.

### 7.2 Χρήσιμες δυσκολίες και σημειώσεις σχετικά με τη συμπεριφορά

Το μοντέλο που βασίζεται στον μετασχηματιστή έδειξε ισχυρή δομική ικανότητα σε ένα ευρύ φάσμα συνθηκών δοκιμών, ιδίως σε γραφήματα που αντιστοιχούν στο πρόγραμμα εκπαίδευσής του[10]. Τόσο η αποκωδικοποίηση greedy όσο και η αποκωδικοποίηση beam παρήγαγαν συστηματικά εφικτές διαδρομές σε 10-30 πόλεις, συνήθως εντός 10-15% των γνωστών βέλτιστων λύσεων[6],[15]. Το

## Κεφάλαιο 7

μοντέλο έμαθε αρκετά καλά να σέβεται τα γεωμετρικά στοιχεία, συμπεριλαμβανομένης της αποφυγής μακρών διασταυρώσεων ή της πρόωρης επιστροφής στον κόμβο εκκίνησης.

Ωστόσο, καθώς το μέγεθος του προβλήματος αυξήθηκε, προέκυψαν αρκετές πρακτικές δυσκολίες:

Σε περιπτώσεις που περιλάμβαναν πυκνά συμπλέγματα ή αποσυνδεδεμένες περιοχές, η αποκωδικοποίηση greedy άρχισε να παρουσιάζει ασταθή συμπεριφορά[18]. Μια κακή αρχική επιλογή μπορούσε εύκολα να παγιώσει την περιήγηση σε απομονωμένα στοιχεία ή σε μη αποτελεσματικές διαδρομές.

Ειδικά όταν συνδυάστηκε με βελτίωση 2-opt, η αναζήτηση δέσμης βελτίωσε σημαντικά την ποιότητα της διαδρομής[13]. Ωστόσο, αυτός ο συνδυασμός πρόσθεσε λειτουργικά έξοδα που έγιναν αισθητά σε πόλεις άνω των 50.

Το μοντέλο συχνά απέτυχε σε γραφήματα που περιλάμβαναν περιορισμούς νησιών όταν οι κόμβοι λιμένων ήταν σπάνιοι ή μακριά από συστάδες νησιών. Αυτό επιβεβαίωσε ότι η λύση Transformer εξαρτάται από την προεπεξεργασία και την επικύρωση για την επιβολή αυτής της λογικής. Δεν είναι εγγενώς ευαίσθητο στους περιορισμούς, ακόμη και αν γενικεύεται καλά στη χωρική δομή.

Παρά τους περιορισμούς αυτούς, η ικανότητα του μοντέλου να παραμένει εφικτό σε καταστάσεις αποκωδικοποίησης σε πραγματικό χρόνο ιδιαίτερα σε γραφήματα με σιωπηρή χωρική οργάνωση υποδηλώνει ότι έχει μάθει μια μεταβιβάσιμη ευριστική μέθοδο και δεν περιορίζεται απλώς στην απομνημόνευση των δεδομένων εκπαίδευσης.

Αυτά τα αποτελέσματα επιβεβαιώνουν ότι ο αποκωδικοποιητής Transformer αποδίδει καλύτερα όταν:

- Η λογική των περιορισμών επιβάλλεται από έξω.
- Το μέτριο πλάτος δέσμης (3–5) εξισορροπεί την ποικιλομορφία με την καθυστέρηση.
- Η μετα-επεξεργασία εξομαλύνει τις ανεπάρκειες σε επίπεδο ακμών.

Το επόμενο μέρος ασχολείται ειδικά με τα εμπόδια στην απόδοση που εντοπίστηκαν κατά την ανάπτυξη, ιδίως σε σχέση με την ενσωμάτωση του μοντέλου σε γραφική διεπαφή.

### 7.3 Περιορισμοί και δυνατότητες υλικού

Αν και το μοντέλο που βασίζεται στο Transformer δημιουργεί περιηγήσεις υψηλής ποιότητας υπό ελεγχόμενες συνθήκες και γενικεύεται καλά, η ανάπτυξη σε πραγματικό χρόνο μέσω γραφικής διεπαφής αποκάλυψε περιορισμούς που σχετίζονται με την αρχιτεκτονική σε επίπεδο συστήματος και τη διαχείριση των πόρων υλικού και όχι με τον ίδιο τον σχεδιασμό του μοντέλου.

Για τις δοκιμές χρησιμοποιήθηκαν συστήματα υψηλής απόδοσης (επεξεργαστής Ryzen 9, κάρτα γραφικών RTX 4070 SUPER Ti). Για γραφήματα με έως ~25 πόλεις, το μοντέλο λειτουργεί χωρίς προβλήματα. Ωστόσο, πέραν των 30 κόμβων, η διεπαφή αρχίζει να παρουσιάζει καθυστερήσεις ή να «παγώνει». Οι αλληλεπιδράσεις χρόνου εκτέλεσης μεταξύ της εξαγωγής συμπερασμάτων, της μετα-επεξεργασίας και της απόδοσης της γραφικής διεπαφής χρήστη προκαλούν αυτά τα προβλήματα, τα οποία δεν οφείλονται στην απόδοση του μοντέλου, αλλά μάλλον σε

**Σημαντικές αιτίες της επιβράδυνσης του χρόνου εκτέλεσης είναι οι εξής:**

- Η αποκωδικοποίηση με αναζήτηση δέσμης επεκτείνει αρκετές υποψήφιες διαδρομές ανά βήμα.
- Βελτιστοποίηση 2-OPT: σε έναν βρόχο που συνδέεται με το GUI, υπολογιστικά δαπανηρή
- Ο αριθμός των κόμβων των πολιτικών νησίδων-λιμένων προκαλεί προβλήματα επικύρωσης περιορισμών.

Τώρα εκτελείται στο κύριο νήμα του Python UI όλη η λογική αποκωδικοποίησης χωρίς επεξεργασία παρτίδων GPU ή ασύγχρονη απομόνωση. Αυτός ο σχεδιασμός μειώνει την ανταπόκριση σε αλληλεπιδράσεις μεγάλων περιπτώσεων, ακόμη και σε σύγχρονο υλικό.

### **Προτεινόμενες αλλαγές**

Διατήρηση της απόδοσης του μοντέλου και αύξηση της χρηστικότητας σε πραγματικό χρόνο:

- Αποστολή εκφόρτωσης και αποκωδικοποίησης μετα-επεξεργασίας μέσω νημάτων παρασκηνίου ή υποδιαδικασιών.
- Παρουσίαση του μοντέλου χρησιμοποιώντας συστήματα backend που υποστηρίζονται από GPU (όπως FastAPI ή TorchServe).
- Δυναμική μείωση των δέσμων ανάλογα με το μέγεθος του γραφήματος για τη διατήρηση της ανταπόκρισης.
- Συγκεντρωμένη GPU inference χρησιμοποιώντας TorchScript για ανάπτυξη

Αυτές οι βελτιώσεις διατηρούν τη κεντρική λογική του μοντέλου και τη συλλογιστική βάσει προσοχής, αυξάνοντας παράλληλα την ανταπόκριση και την επεκτασιμότητα. Δείχνουν μια τυπική δυσκολία στα συστήματα deep learning: τη μείωση της απόστασης μεταξύ της ακρίβειας offline και της χρηστικότητας online.

## **7.4 Όρια μοντέλου και σχεδιαστικές συμβιβαστικές λύσεις**

Η αρχιτεκτονική που βασίζεται στον μετασχηματιστή, η οποία αναπτύχθηκε στην παρούσα διατριβή, παρουσιάζει αρθρωτότητα, ευελιξία και καλή γενικευτική ικανότητα. Ωστόσο, κατά τη διάρκεια της ανάπτυξης προέκυψαν ορισμένοι πρακτικοί συμβιβασμοί και περιορισμοί, όχι ως αποτυχίες του μοντέλου, αλλά ως αποφάσεις που ελήφθησαν για να διατηρηθεί η επεκτασιμότητα και η απλότητα.

### **Θεωρία εξωτερικών περιορισμών**

Το μοντέλο αντιμετωπίζει αυτούς τους κανόνες σκοπιμότητας εξωτερικά, αντί να ενσωματώνει περιορισμούς, όπως η συνδεσιμότητα νησίδων-λιμένων, στα επίπεδα του Transformer. Αυτό εγγυάται τη γενικότητα του μοντέλου, αλλά σημαίνει επίσης ότι:

- Το μοντέλο δεν «κατανοεί» εσωτερικά την ανάγκη για λιμάνια.
- Η σωστή προ- και μετα-επεξεργασία είναι αυτό που καθορίζει τη σκοπιμότητα.
- Οι περιορισμοί της πραγματικής εφοδιαστικής, όπως η πολυτροπική δρομολόγηση ή τα χρονικά παράθυρα, απαιτούν εξωτερικά modules.

### Μειωμένη δομική προσαρμοστικότητα

Το παρόν μοντέλο υποθέτει ευκλείδειους, χωρικά συνεχείς γραφήματα με τέλεια ορατότητα. Αν και λειτουργεί καλά σε μια σειρά μεγεθών και αρχιτεκτονικών γραφημάτων, δεν:

- Κωδικοποιεί χαρακτηριστικά ακμών ή προσοχή στην απόσταση.
- Λειτουργεί με αραιά ή ασύνδετα γραφήματα χωρίς εξωτερική καθοδήγηση.
- Εδώ θα βρείτε περιορισμούς γραφημάτων, δηλαδή πρόσβαση μόνο μέσω γεφυρών. από πάνω προς τα κάτω

### Εξάρτηση από το πρόγραμμα σπουδών και ερμηνευτική σύζευξη

Σήμερα, η εκπαίδευση με συνθετικό πρόγραμμα σπουδών και η μετα-επεξεργασία δύο βέλτιστων λύσεων καθορίζουν την ποιότητα του μοντέλου. Χωρίς αυτά, μη εφικτές ακολουθίες ή παρακάμψεις θα μπορούσαν να βρεθούν στις λύσεις. Αυτό τονίζει, όχι ως περιορισμό, αλλά ως σχεδιαστική επιλογή, την αρμονία μεταξύ της πολιτικής που έχει μάθει το μοντέλο και των κλασικών επιπέδων βελτιστοποίησης.

Αυτές οι συμβιβαστικές λύσεις ορίζουν ένα σαφές εύρος: το μοντέλο είναι πιο κατάλληλο για προβλήματα γεωχωρικής δρομολόγησης μέτριας πολυπλοκότητας και με καλή δειγματοληψία [18], [19],[20]. Η υποστήριξη νέων τομέων απαιτεί μόνο αλλαγές στη λογική των περιορισμών, στη μορφοποίηση των εισόδων και στα περιβλήματα συμπερασμάτων. Δεν απαιτείται τροποποίηση της αρχιτεκτονικής.

Αυτό το αρθρωτό χαρακτήρα αποτελεί πλεονέκτημα και όχι μειονέκτημα. Υποστηρίζει υβριδικές λύσεις έτοιμες για επέκταση σε πιο σύνθετες εργασίες, όπως VRP, TW-TSP και δρομολόγηση πολλαπλών πρακτόρων που συνδυάζουν τη νευρωνική σκέψη με την επιβολή κανόνων.

## Σύνοψη

Το παρόν κεφάλαιο παρουσίασε μια στοχαστική αξιολόγηση της λύσης TSP βασισμένης στο Transformer που δημιουργήθηκε για την παρούσα διατριβή. Δίνοντας έμφαση σε εμπειρικές γνώσεις, πρακτικούς περιορισμούς και ρεαλιστικά ζητήματα ανάπτυξης, αντί για ακατέργαστα αριθμητικά σημεία αναφοράς ή οπτικές επιδείξεις, έδωσε προσοχή

Ιδιαίτερα όταν υποστηρίζεται από εκπαίδευση στο πλαίσιο του προγράμματος σπουδών, αναζήτηση δέσμης και βελτίωση 2-opt, το μοντέλο αποδείχθηκε ικανό να δημιουργήσει λογικές, σχεδόν βέλτιστες διαδρομές τόσο σε συνθετικά όσο και σε πραγματικά γραφήματα. Ωστόσο, στοιχεία όπως οι πόροι υλικού, η διαχείριση εξωτερικών περιορισμών και η μη παράλληλη συμπερασματολογία GUI περιορίζουν την απόδοση και τη χρηστικότητα του συστήματος.

Το πιο σημαντικό είναι ότι αυτοί οι περιορισμοί είναι ζητήματα ανάπτυξης και όχι αρχιτεκτονικά ελαττώματα. Ως εκ τούτου, οι βελτιώσεις στον αρθρωτό σχεδιασμό και οι βελτιστοποιήσεις σε επίπεδο συστήματος συμβάλλουν στην υπέρβασή τους. Με μικρές αλλαγές στην υποδομή, η λύση είναι σε θέση να επεκταθεί πέρα από την έρευνα και σε εφαρμογές δρομολόγησης σε επίπεδο παραγωγής, ιδίως σε γεωγραφικές περιοχές όπως η Ελλάδα, όπου οι περιορισμοί των νησιωτικών λιμένων εισάγουν μεγάλη πολυπλοκότητα.

Οι μελλοντικές κατευθύνσεις, συμπεριλαμβανομένης της μοντελοποίησης περιορισμών που μπορούν να μάθουν, της επέκτασης σε πλουσιότερα προβλήματα δρομολόγησης και της ενσωμάτωσης backend για υψηλής απόδοσης συμπεράσματα, ολοκληρώνουν τη διατριβή. Αυτά δεν σηματοδοτούν μόνο τη φυσική εξέλιξη αυτής της εργασίας, αλλά και την πρόοδο στον γενικότερο τομέα της νευρωνικής συνδυαστικής βελτιστοποίησης.

## Κεφάλαιο 8: Συμπεράσματα

Αυτό το άρθρο παρουσίασε μια καινοτόμο μέθοδο που χρησιμοποιεί μοντέλα Transformer και προσεγγίσεις ενισχυτικής μάθησης για την επίλυση του προβλήματος του περιοδεύοντος πωλητή (TSP). Από τα θεωρητικά θεμέλια έως την υλοποίηση, την εκπαίδευση και την εμπειρική αξιολόγηση, το προτεινόμενο σύστημα αποδείχθηκε ικανό να παράγει σχεδόν βέλτιστες λύσεις σε αποδεκτό χρονικό διάστημα, ακόμη και υπό περιορισμούς όπως η δρομολόγηση με βάση νησιά.

Η αποκωδικοποίηση με αναζήτηση δέσμης, οι βελτιώσεις 2-opt και η λογική επικύρωσης περιορισμών, σε συνδυασμό, επέτρεψαν στο μοντέλο να λειτουργήσει αποτελεσματικά σε πραγματικές τοπολογίες, συμπεριλαμβανομένων των περίπλοκων νησιωτικών δικτύων της Ελλάδας. Παρά τους περιορισμούς του υλικού και την απουσία ενσωματωμένης αναγνώρισης περιορισμών στο μοντέλο, το σύστημα αποδείχθηκε ευέλικτο και ανοιχτό για ανάπτυξη σε επόμενες χρήσεις δρομολόγησης και λογιστικής.

# Παραρτήματα

## Παράρτημα Α - Κώδικας Βάσης Δεδομένων Ελληνικών Πόλεων

### Παράρτημα Α.1 – Συνάρτηση `build_records`

```
def build_records(max_rows: int = 100) -> list[dict]:  
    gc = geonamescache.GeonamesCache()  
    g_cities = [c for c in gc.get_cities().values() if c["countrycode"] == "GR"]  
    g_cities.sort(key=lambda x: int(x["population"]), reverse=True)  
  
    records: list[dict] = []  
    next_id = 1  
  
    # Pass 1 – GeoNames (population order)  
    for city in g_cities:  
        key = normalize(city["name"])  
        if key in BLACKLIST:  
            continue  
  
        records.append(  
            dict(  
                id=next_id,  
                name=geo_to_greek(city),  
                latitude=float(city["latitude"]),  
                longitude=float(city["longitude"]),  
                is_port=int(key in PORT_CITIES),  
                is_island=int(key in ISLAND_CITIES),  
            )  
        )  
    )
```

```

next_id += 1

if len(records) >= max_rows:
    break

# Pass 2 – CSV top-up
if len(records) < max_rows and EXTRA_CSV_PATH.is_file():
    df = pd.read_csv(EXTRA_CSV_PATH)
    for _, row in df.iterrows():
        latin_name = str(row["name"]).strip()
        key = normalize(latin_name)
        if key in BLACKLIST or any(normalize(r["name"]) == key for r in records):
            continue

    records.append(
        dict(
            id=next_id,
            name=csv_to_greek(latin_name),
            latitude=float(row["latitude"]),
            longitude=float(row["longitude"]),
            is_port=int(row.get("is_port", key in PORT_CITIES)),
            is_island=int(row.get("is_island", key in ISLAND_CITIES)),
        )
    )
    next_id += 1
    if len(records) >= max_rows:
        break

# Drop some cities, append extras
records = [r for r in records if r["name"] not in ("Ρόδος", "Νεάπολη", "Κηφισιά")]

```

```

extras = [
    {"name": "Ηγουμενίτσα", "latitude": 39.503420, "longitude": 20.267280, "is_port": 1,
"is_island": 0},
    {"name": "Ρόδος", "latitude": 36.433889, "longitude": 28.217500, "is_port": 1, "is_island": 1},
    {"name": "Καστοριά", "latitude": 40.520744, "longitude": 21.271206, "is_port": 0, "is_island":
0},
    {"name": "Άγιος Νικόλαος", "latitude": 35.191060, "longitude": 25.715240, "is_port": 1,
"is_island": 1},
    {"name": "Τύρναβος", "latitude": 39.733000, "longitude": 22.283000, "is_port": 0, "is_island":
0},
    {"name": "Γύθειο", "latitude": 36.761028, "longitude": 22.563999, "is_port": 0, "is_island": 0},
    {"name": "Ηράκλειο", "latitude": 35.32787, "longitude": 25.14341, "is_port": 1, "is_island": 1}
]

for e in extras:
    if len(records) >= max_rows:
        break
    records.append({
        "id": next_id,
        "name": e["name"],
        "latitude": e["latitude"],
        "longitude": e["longitude"],
        "is_port": e["is_port"],
        "is_island": e["is_island"],
    })
    next_id += 1

return records

```

## Παράρτημα Α.2 – Συνάρτηση write\_to\_db

def write\_to\_db(records: list[dict]) -> None:

```
conn = get_connection()
```

```
with conn:
```

```
    cur = conn.cursor()
```

```
    cur.execute(f"DROP TABLE IF EXISTS {TABLE_NAME}")
```

```
    cur.execute(
```

```
        f"""
```

```
        CREATE TABLE {TABLE_NAME} (
```

```
            id    INTEGER PRIMARY KEY,
```

```
            name  TEXT NOT NULL,
```

```
            latitude REAL NOT NULL,
```

```
            longitude REAL NOT NULL,
```

```
            is_port INTEGER NOT NULL,
```

```
            is_island INTEGER NOT NULL
```

```
        )
```

```
        """
```

```
    )
```

```
    cur.executemany(
```

```
        f"""INSERT INTO {TABLE_NAME}
```

```
        (id, name, latitude, longitude, is_port, is_island)
```

```
        VALUES (?, ?, ?, ?, ?, ?)""",
```

```
        [
```

```
            (
```

```
                r["id"],
```

```
                r["name"],
```

```
                r["latitude"],
```

```
                r["longitude"],
```

```
                r["is_port"],
```

```
        r["is_island"],
    )
    for r in records
],
)
print(f"✓ Inserted {len(records)} cities — all names are Greek.")
```

## Παράρτημα Β - Κώδικας Εκπαίδευσης Μοντέλου

### Παράρτημα Β.1.1 – PositionalEncoding

```
class PositionalEncoding(nn.Module):
    def __init__(self, d_model: int, dropout: float = 0.1, max_len: int = 1000):
        super().__init__()
        self.dropout = nn.Dropout(p=dropout)

        pe = torch.zeros(max_len, d_model)
        position = torch.arange(0, max_len, dtype=torch.float).unsqueeze(1)
        div_term = torch.exp(
            torch.arange(0, d_model, 2, dtype=torch.float) * (-math.log(10000.0) / d_model)
        )
        pe[:, 0::2] = torch.sin(position * div_term)
        pe[:, 1::2] = torch.cos(position * div_term)
        self.register_buffer("pe", pe.unsqueeze(0))

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        x = x + self.pe[:, : x.size(1), :]
        return self.dropout(x)
```

### Παράρτημα Β.1.2 – PointerDecoder

```
class PointerDecoder(nn.Module):
    def __init__(
        self,
        d_model: int,
        n_heads: int,
        num_layers: int,
        dim_feedforward: int,
        dropout: float
    ):
        super().__init__()
        layer = nn.TransformerDecoderLayer(
            d_model=d_model,
            nhead=n_heads,
            dim_feedforward=dim_feedforward,
            dropout=dropout,
            batch_first=True
        )
        self.decoder = nn.TransformerDecoder(layer, num_layers=num_layers)
```

```

def forward(
    self,
    tgt_emb: torch.Tensor,
    memory: torch.Tensor,
    tgt_mask: torch.Tensor = None,
    memory_key_padding_mask: torch.Tensor = None,
    tgt_key_padding_mask: torch.Tensor = None
) -> torch.Tensor:

    dec_out = self.decoder(
        tgt_emb,
        memory,
        tgt_mask=tgt_mask,
        memory_key_padding_mask=memory_key_padding_mask,
        tgt_key_padding_mask=tgt_key_padding_mask
    )

    logits = torch.matmul(dec_out, memory.transpose(1, 2)) / math.sqrt(dec_out.size(-1))
    return logits

```

### Παράρτημα B.1.3 – TransformerTSP

```

class TransformerTSP(nn.Module):
    def __init__(self):
        super().__init__()
        d_model = hp.transformer_d_model
        n_heads = hp.transformer_n_heads
        enc_layers = hp.transformer_num_encoder
        dec_layers = hp.transformer_num_decoder
        ff_dim = hp.transformer_ff_dim
        dropout = hp.transformer_dropout
        input_dim = hp.transformer_input_dim

        self.embedding = nn.Linear(input_dim, d_model)
        self.pos_encoder = PositionalEncoding(d_model, dropout)

        self.encoder = nn.ModuleList([
            DistanceBiasLayer(d_model, n_heads, ff_dim, dropout)
            for _ in range(enc_layers)
        ])

```

```

self.decoder = PointerDecoder(
    d_model=d_model,
    n_heads=n_heads,
    num_layers=dec_layers,
    dim_feedforward=ff_dim,
    dropout=dropout
)

def forward(
    self,
    src: torch.Tensor,          # [B, N, 4]
    tgt_idx: torch.LongTensor, # [B, T]
    src_key_padding_mask: torch.BoolTensor = None,
    tgt_key_padding_mask: torch.BoolTensor = None
) -> torch.Tensor:
    B, N, _ = src.size()

    src_emb = self.pos_encoder(self.embedding(src))

    lat = torch.deg2rad(src[...,0])
    lon = torch.deg2rad(src[...,1])
    phi1 = lat.unsqueeze(2); phi2 = lat.unsqueeze(1)
    lam1 = lon.unsqueeze(2); lam2 = lon.unsqueeze(1)
    dphi = (phi1 - phi2).sin() ** 2
    dlam = (lam1 - lam2).sin() ** 2
    cos1 = torch.cos(phi1) * torch.cos(phi2)
    hav = dphi + cos1 * dlam
    dist_mat = 2 * 6371.0 * hav.sqrt().asin() # [B,N,N]

    memory = src_emb
    for layer in self.encoder:
        memory = layer(memory,
                        src_key_padding_mask=src_key_padding_mask,
                        dist_mat=dist_mat)

    start_tok = torch.zeros((B,1,memory.size(-1)), device=src.device)
    if tgt_idx.size(1) == 0:
        tgt_emb = start_tok
    else:
        gathered = memory.gather(
            1,
            tgt_idx.unsqueeze(-1).expand(-1,-1,memory.size(-1))

```

```

)
tgt_emb = torch.cat([start_tok, gathered], dim=1)

tlen = tgt_emb.size(1)
tgt_mask = nn.Transformer.generate_square_subsequent_mask(tlen).to(src.device)

logits = self.decoder(
    tgt_emb,
    memory,
    tgt_mask=tgt_mask,
    memory_key_padding_mask=src_key_padding_mask,
    tgt_key_padding_mask=tgt_key_padding_mask
)
return logits[:, :1, :] if tgt_idx.size(1) == 0 else logits[:, 1:, :]

```

## Παράρτημα B.2 – train\_transformer\_curriculum.py

```

def safe_load(path: Path):
    return torch.load(path, weights_only=False)

def parse_stage_filename(path: Path):
    m = re.match(r"supervised_{[a-z]+}_{d+}cities\.pt$", path.name)
    if not m:
        raise ValueError(f"Cannot parse mode and city count from '{path.name}'")
    return m.group(1), int(m.group(2))

def discover_datasets(data_dir: str):
    dirpath = Path(data_dir)
    candidates = list(dirpath.glob("supervised_*_*cities.pt"))
    stages = []
    for p in candidates:
        try:
            mode, n = parse_stage_filename(p)
        except ValueError:
            continue
        stages.append((n, mode, p))

    priority = {"exact": 0, "hybrid": 1, "heuristic": 2}

    stages.sort(key=lambda x: (x[0], priority.get(x[1], 99)))

    return [p for (_, _, p) in stages]

```

```

def train_curriculum(dataset_paths, hp: HyperParams, num_workers: int):
    device = torch.device("cuda" if hp.use_gpu and torch.cuda.is_available() else "cpu")

    model = TransformerTSP().to(device)
    optim = torch.optim.Adam(model.parameters(), lr=hp.transformer_lr)
    criterion = nn.CrossEntropyLoss()

    logging.basicConfig(level=logging.INFO, format="%(message)s")
    logger = logging.getLogger("curriculum")

    for ds_path in dataset_paths:
        mode, n_c = parse_stage_filename(ds_path)
        logger.info(f"\n=== Stage: {n_c} cities | mode={mode} | epochs={hp.transformer_epochs} ===")

        raw = safe_load(ds_path)
        dataset = SupervisedDataset(raw)
        loader = DataLoader(
            dataset,
            batch_size=hp.transformer_batch_size,
            shuffle=True,
            num_workers=num_workers,
            pin_memory=hp.use_gpu
        )

        for ep in range(1, hp.transformer_epochs + 1):
            model.train()
            total_loss = 0.0
            for coords, tour in loader:
                coords = coords.to(device) # [B, n, 4]
                tour = tour.to(device) # [B, n]

                optim.zero_grad()
                logits = model(coords, tour) # [B, n, vocabulary]
                loss = criterion(logits.permute(0, 2, 1), tour)
                loss.backward()
                optim.step()

            total_loss += loss.item()

        avg = total_loss / len(loader)
        logger.info(f"Stage {n_c} | Epoch {ep}/{hp.transformer_epochs} | Loss: {avg:.4f}")

```

```

ckpt_dir = Path("checkpoints")
ckpt_dir.mkdir(exist_ok=True)
ckpt_path = ckpt_dir / f'curriculum_{mode}_{n_c}cities.pt'
torch.save(model.state_dict(), ckpt_path)
logger.info(f'Saved checkpoint → {ckpt_path}')

if __name__ == "__main__":
    parser = argparse.ArgumentParser("Curriculum training for Transformer TSP")
    parser.add_argument(
        "--data-dir", required=True,
        help="Directory where supervised_{mode}_{ncities}cities.pt files live"
    )
    parser.add_argument(
        "--workers", type=int, default=4,
        help="Number of DataLoader worker processes"
    )
    parser.add_argument(
        "--batch-size", type=int,
        help="Override batch size (defaults to HyperParams)"
    )
    parser.add_argument(
        "--epochs", type=int,
        help="Override number of epochs per stage (defaults to HyperParams)"
    )
    parser.add_argument(
        "--lr", type=float,
        help="Override learning rate (defaults to HyperParams)"
    )
    args = parser.parse_args()

    hp = HyperParams()

    if args.batch_size:
        object.__setattr__(hp, 'transformer_batch_size', args.batch_size)
    if args.epochs:
        object.__setattr__(hp, 'transformer_epochs', args.epochs)
    if args.lr:
        object.__setattr__(hp, 'transformer_lr', args.lr)

    dataset_paths = discover_datasets(args.data_dir)
    if not dataset_paths:

```

```

raise RuntimeError(f"No supervised_*.pt files found in {args.data_dir}")

print("Discovered curriculum stages:")
for p in dataset_paths:
    m, n = parse_stage_filename(Path(p))
    print(f" - {n} cities | mode={m} → {p.name}")

train_curriculum(dataset_paths, hp, args.workers)

```

### Παράρτημα B.3 – reinforce\_finetune.py

```

def sample_instance(num_cities: int, city_rows):
    selection = random.sample(city_rows, num_cities)
    coords = torch.tensor(
        [[c[2], c[3], float(c[4]), float(c[5])] for c in selection],
        dtype=torch.float32
    )
    return coords #[N,4]

def tour_length(tour, dist_mat):
    idx = np.array(tour, dtype=int)
    next_idx = np.roll(idx, -1)
    return float(dist_mat[idx, next_idx].sum())

if __name__ == "__main__":
    parser = argparse.ArgumentParser("RL fine-tuning for TSP Transformer")
    parser.add_argument("--init-model", required=True,
                        help="Path to pretrained supervised checkpoint (.pt)")
    parser.add_argument("--output", required=True,
                        help="Where to save the RL-tuned checkpoint")
    parser.add_argument("--num-cities", type=int, default=50,
                        help="Fixed N if no curriculum is given")
    parser.add_argument("--city-curriculum", nargs="+", type=int,
                        help="List of city-counts to sample from, e.g. 20 30 50")
    parser.add_argument("--batch-size", type=int, default=16,
                        help="Rollouts per gradient update")
    parser.add_argument("--accum-steps", type=int, default=1,
                        help="Gradient accumulation steps")
    parser.add_argument("--samples", type=int, default=1000,
                        help="Rollouts per epoch")

```

```

parser.add_argument("--epochs", type=int, default=10,
                    help="Total RL epochs")
parser.add_argument("--lr", type=float, default=1e-5,
                    help="Policy-gradient learning rate")
parser.add_argument("--time-budget", type=float, default=5.0,
                    help="Seconds of 2-opt per rollout")
parser.add_argument("--entropy-coef", type=float, default=0.0,
                    help="Entropy bonus weight")
parser.add_argument("--clip-grad", type=float, default=1.0,
                    help="Max-norm for gradient clipping (0 to disable)")
parser.add_argument("--patience", type=int, default=3,
                    help="Early-stop if no avg-reward improvement")
parser.add_argument("--config", type=str,
                    help="Optional JSON/YAML config file for overrides")
parser.add_argument("--temperature", type=float, default=0.9,
                    help="Sampling softmax temperature (<1 more greedy, >1 more random)")
args = parser.parse_args()

if args.config:
    cfg = json.load(Path(args.config).open())
    for key, val in cfg.items(): setattr(args, key, val)

hp = HyperParams()
device = torch.device("cuda" if hp.use_gpu and torch.cuda.is_available() else "cpu")

conn = sqlite3.connect("data/cities.db")
city_rows = conn.execute(
    "SELECT id,name,latitude,longitude,is_port,is_island FROM cities"
).fetchall()
conn.close()

model = TransformerTSP().to(device)
model.load_state_dict(torch.load(args.init_model, map_location=device))
optimizer = torch.optim.Adam(model.parameters(), lr=args.lr)
scaler = GradScaler()

writer = SummaryWriter()
best_reward = -float("inf")
no_improve = 0

if args.accum_steps > 1:
    optimizer.zero_grad()

```

```

for epoch in range(1, args.epochs + 1):
    epoch_start = time.time()
    total_reward = 0.0
    batch_count = 0

    for i in range(1, args.samples + 1,
                 desc=f"Epoch {epoch}", unit="rollout"):

        N = random.choice(args.city_curriculum) if args.city_curriculum else args.num_cities
        coords = sample_instance(N, city_rows).to(device)
        dist_mat = compute_distance_matrix(coords[:, :2])

        with autograd.enable_grad():
            prefix = []
            log_probs = []
            src = coords.unsqueeze(0)
            model.eval()
            for _ in range(N):
                if len(prefix) == 0:
                    tgt = torch.zeros((1,0), dtype=torch.long, device=device)
                else:
                    tgt = torch.tensor([prefix], dtype=torch.long, device=device)
                logits = model(src, tgt)
                last = logits[0, -1, :]

                mask = torch.zeros(N, dtype=torch.bool, device=device)
                mask[prefix] = True
                last = last.masked_fill(mask, float('-inf'))

                probs = F.softmax(last / args.temperature, dim=-1)
                city = torch.multinomial(probs, 1).item()
                log_probs.append(torch.log(probs[city] + 1e-9))
                prefix.append(city)
            sampled_seq = prefix
            sampled_logprob = sum(log_probs)

            length = tour_length(sampled_seq, dist_mat)
            reward = -length

            beam = max(2, int((hp.transformer_base_beam_per_city * N)))
            greedy_seq, _ = model.sample_with_logprob(src, beam_size=beam,

```

```

time_budget=args.time_budget)
    baseline = -tour_length(greedy_seq[0], dist_mat)

    adv = reward - baseline
    entropy = 0.0
    if args.entropy_coef > 0:
        probs_tensor = torch.exp(sampled_logprob)
        entropy = -probs_tensor * sampled_logprob

    loss = -adv * sampled_logprob - args.entropy_coef * entropy
    loss = loss / args.accum_steps

    scaler.scale(loss).backward()
    batch_count += 1

    if batch_count >= args.accum_steps:
        scaler.unscale_(optimizer)
        if args.clip_grad > 0:
            nn.utils.clip_grad_norm_(model.parameters(), args.clip_grad)
        scaler.step(optimizer)
        scaler.update()
        optimizer.zero_grad()
        batch_count = 0

    total_reward += reward

    torch.cuda.empty_cache()
    epoch_time = (time.time() - epoch_start) / 60
    avg_reward = total_reward / args.samples
    print(f"Epoch {epoch} complete in {epoch_time:.1f} min | Avg Reward: {avg_reward:.4f}")
    writer.add_scalar("rl/avg_reward", avg_reward, epoch)

    if avg_reward > best_reward + 1e-6:
        best_reward = avg_reward
        no_improve = 0
        Path(args.output).parent.mkdir(parents=True, exist_ok=True)
        torch.save(model.state_dict(), args.output)
    else:
        no_improve += 1
        if no_improve >= args.patience:
            print(f"No improvement for {args.patience} epochs → stopping RL.")
            break

```

```
writer.close()
print("RL fine-tuning complete. Best model saved at:", args.output)
```

#### Παράρτημα Β.4 – hyperparams.py

```
class HyperParams:
    # -----
    # Nearest-Neighbor (weakened)
    # -----
    nn_start_city: int = 5

    # -----
    # Genetic Algorithm (GA)
    # -----
    ga_population: int = 30
    ga_generations: int = 100
    ga_mutation_rate: float = 0.02

    # -----
    # Ant Colony Optimization (ACS)
    # -----
    acs_ants: int = 10
    acs_alpha: float = 1.0
    acs_beta: float = 2.0
    acs_evaporation: float = 0.8
    acs_iterations: int = 50

    # -----
    # Branch-and-Bound (limit search)
    # -----
    bnb_start_city: int = 0
    bnb_max_cities: int = 14

    # -----
    # Transformer inference—keep maximally strong
    # -----
    transformer_beam_size: int = 5
    transformer_max_beam: int = 100
    transformer_base_beam_per_city: float = 1.0

    transformer_use_two_opt: bool = True
```

```
transformer_max_two_opt_passes: int = 20
transformer_time_limit_sec: float = 30.0
transformer_num_runs: int = 20
transformer_temp_start: float= 0.8
transformer_temp_end: float = 0.4
# -----
# Transformer architecture parameters (unchanged)
# -----
transformer_d_model: int = 256
transformer_n_heads: int = 8
transformer_num_encoder: int = 6
transformer_num_decoder: int= 6
transformer_ff_dim: int = 1024
transformer_dropout: float = 0.1
transformer_input_dim: int = 4

# -----
# Device
# -----
use_gpu: bool = True
```

## Παράρτημα Γ – Υβριδική Αποκωδικοποίηση και 2-OPT

### Παράρτημα Γ.1 – Μέθοδος `sample_with_logprob` (από `transformer_tsp.py`)

```
def sample_with_logprob(
    self,
    coords: torch.Tensor,      # [B, N, 4]
    beam_size: int = 1,
    time_budget: float = 0.0,
    temperature: float = None,
    stochastic: bool = False,
    use_two_opt: bool = None,
    cities: list = None
) -> tuple[list[list[int]], torch.Tensor]:
    B, N, _ = coords.size()
    device = coords.device

    src_emb = self.pos_encoder(self.embedding(coords))
    lat = torch.deg2rad(coords[...,0]); lon = torch.deg2rad(coords[...,1])
    phi1 = lat.unsqueeze(2); phi2 = lat.unsqueeze(1)
    lam1 = lon.unsqueeze(2); lam2 = lon.unsqueeze(1)
    dphi = (phi1 - phi2).sin() ** 2
    dlam = (lam1 - lam2).sin() ** 2
    cos1 = torch.cos(phi1) * torch.cos(phi2)
    hav = dphi + cos1 * dlam
    dist_mat = 2 * 6371.0 * hav.sqrt().asin() # [B,N,N]

    memory = src_emb
    for layer in self.encoder:
        memory = layer(memory, dist_mat=dist_mat)

    all_tours = []
    all_logps = []

    if isinstance(temperature, tuple):
        T_start, T_end = temperature
    else:
        T_start = T_end = temperature if temperature is not None else getattr(
            self, "transformer_sampling_temperature", 1.0)

    for b in range(B):
        mem = memory[b:b+1] # [1, N, d_model]
```

```

beams: list[tuple[torch.Tensor, list[int]]] = [(torch.zeros((), device=device), [])]

for t in range(N):
    candidates: list[tuple[torch.Tensor, list[int]]] = []

    for cum_lp, seq in beams:
        start_tok = torch.zeros((1,1,mem.size(-1)), device=device)
        if not seq:
            tgt_emb = start_tok
        else:
            idxs = torch.tensor(seq, device=device).view(1, -1)
            gathered = mem.gather(1, idxs.unsqueeze(-1).expand(-1,-1,mem.size(-1)))
            tgt_emb = torch.cat([start_tok, gathered], dim=1)

        tlen = tgt_emb.size(1)
        tgt_mask = nn.Transformer.generate_square_subsequent_mask(tlen).to(device)

        logits = self.decoder(tgt_emb, mem, tgt_mask=tgt_mask)
        step_logits = logits[0, -1].clone() # [N]

        for vid in seq:
            step_logits[vid] = float("-inf")

        prev = seq[-1] if seq else None
        if prev is not None:
            ports = (coords[b, :, 2] == 1).nonzero(as_tuple=True)[0].tolist()
            islands = (coords[b, :, 3] == 1).nonzero(as_tuple=True)[0].tolist()

            if prev not in ports:
                for isl in islands:
                    step_logits[isl] = float("-inf")

            # b) never leave an island unless going to a port
            if prev in islands:
                non_ports = [j for j in range(N) if j not in ports]
                for j in non_ports:
                    step_logits[j] = float("-inf")

        if isinstance(temperature, tuple):
            T = T_start + (T_end - T_start) * (t / max(1, N - 1))
        else:
            T = T_start

```

```

logprobs = F.log_softmax(step_logits / T, dim=-1)

if stochastic:
    probs = torch.exp(logprobs / T)
    probs = probs / probs.sum(dim=-1, keepdim=True)
    num_cands = min(beam_size * 2, N - len(seq))
    cand_idx = torch.multinomial(probs, num_cands, replacement=False)
    cand_logps = logprobs[cand_idx]
    topv, topi = torch.topk(cand_logps, k=min(beam_size, num_cands))
    for v, i in zip(topv, topi):
        idx = int(cand_idx[i].item())
        candidates.append((cum_lp + v, seq + [idx]))
else:
    topv, topi = torch.topk(logprobs, k=min(beam_size, N - len(seq)))
    for v, idx in zip(topv, topi):
        candidates.append((cum_lp + v, seq + [int(idx.item())]))

beams = sorted(candidates,
               key=lambda x: x[0].item(),
               reverse=True)[:beam_size]

best_lp, best_seq = beams[0]
_use = use_two_opt if use_two_opt is not None else getattr(hp, "transformer_use_two_opt",
True)
if _use:
    best_seq = apply_two_opt(
        best_seq,
        dist_mat[b].cpu().numpy(),
        cities,
        max_passes=hp.transformer_max_two_opt_passes
    )

all_tours.append(best_seq)
all_logps.append(best_lp)

logprob_tensor = torch.stack(all_logps) # [B]
return all_tours, logprob_tensor

```

## Παράρτημα Γ.2 – Κλήση `apply_two_opt` και χρήση της

```
def apply_two_opt(
    tour: list[int],
    dist_mat: np.ndarray,
    cities: list[tuple],
    max_passes: int = TWO_OPT_PASSES
) -> list[int]:
    n = len(tour)
    best = tour.copy()
    passes = 0
    improved = True
    while improved and passes < max_passes:
        improved = False
        for i in range(1, n - 2):
            for j in range(i + 1, n):
                a, b = best[i - 1], best[i]
                c, d = best[j], best[(j + 1) % n]
                before = dist_mat[a, b] + dist_mat[c, d]
                after = dist_mat[a, c] + dist_mat[b, d]
                if after + 1e-9 < before:
                    a, b = best[i - 1], best[i]
                    c, d = best[j], best[(j + 1) % n]
                    if cities[a][5] == 1 and cities[c][4] == 0:
                        continue

                    if cities[c][5] == 1 and cities[a][4] == 0:
                        continue
                    best[i : j + 1] = best[i : j + 1][::-1]
                    improved = True
                    break
            if improved:
                break
        passes += 1
    return best
```

## Παράρτημα Δ – Κανόνες Επικύρωσης Περιορισμών Δρομολόγησης

### Παράρτημα Δ.1 – Μέθοδοι Ελέγχου Επιλογής Πόλεων

```
def validate_required_ports(path: List[int], required_ports: List[int]) -> bool:
    return all(port in path for port in required_ports)

def validate_route(path: List[int], cities: List[Tuple]) -> bool:
    ports = {i for i,c in enumerate(cities) if c[4]==1}
    islands = {i for i,c in enumerate(cities) if c[5]==1}
    n = len(path)

    for idx, city in enumerate(path):
        prev_city = path[(idx - 1) % n]
        next_city = path[(idx + 1) % n]

        if city in islands:
            if prev_city not in ports or next_city not in ports:
                return False

    for i in range(n - 2):
        if path[i] == path[i+2]:
            return False

    return True
```

## Παράρτημα Ε – Περιβάλλον Εκτέλεσης και Εξαρτήσεις

Για την αναπαραγωγή του περιβάλλοντος εκτέλεσης της εφαρμογής και την εκπαίδευση του μοντέλου, χρησιμοποιείται ένα σύνολο από βιβλιοθήκες Python. Οι βιβλιοθήκες αυτές περιλαμβάνονται στο αρχείο requirements.txt, το οποίο επιτρέπει τη δημιουργία εικονικού περιβάλλοντος με μία μόνο εντολή:

► `pip install -r requirements.txt`

Κύριες βιβλιοθήκες:

- **torch** , Εκπαίδευση και inference μοντέλου Transformer
- **numpy** , Αριθμητικές πράξεις και επεξεργασία πινάκων
- **tqdm** , Οπτική παρακολούθηση προόδου εκπαίδευσης
- **matplotlib** , Οπτικοποίηση διαδρομών και χαρτών
- **pandas** , Ανάγνωση και επεξεργασία γεωγραφικών δεδομένων
- **PyQt6** , Ανάπτυξη διεπαφής χρήστη (GUI)
- **geopy** , Υπολογισμός αποστάσεων βάσει γεωγραφικού πλάτους/μήκους
- **sqlite3** , Αποθήκευση δεδομένων πόλεων και αποτελεσμάτων

Για τις GPU εκδόσεις, περιλαμβάνονται επίσης βιβλιοθήκες της NVIDIA όπως nvidia-cublas-cu12, nvidia-cudnn-cu12 και triton, οι οποίες αξιοποιούνται αυτόματα αν υπάρχει διαθέσιμη υποστήριξη CUDA στο σύστημα.

# Βιβλιογραφία

## Ακαδημαϊκές πηγές

- [1] C. E. Noon and J. C. Bean, “An efficient transformation of the generalized traveling salesman problem into the traveling salesman problem on digraphs,” *Operations Research*, vol. 39, no. 4, pp. 623–632, Jul. 1991. doi: 10.1287/opre.39.4.623
- [2] G. Laporte, A. Asef-Vaziri, and C. Sriskandarajah, “Some applications of the generalized traveling salesman problem,” *Journal of the Operational Research Society*, vol. 47, no. 12, pp. 1461–1467, Dec. 1996. doi: 10.1057/jors.1996.190
- [3] A. L. H. Labordère, “A dynamic programming solution of the generalized traveling salesman problem,” *RAIRO Recherche opérationnelle*, vol. 3, no. V3, pp. 85–90, 1969.
- [4] M. Dorigo and L. M. Gambardella, “Ant colonies for the traveling salesman problem,” *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997. doi: 10.1016/S0303-2647(97)01708-5
- [5] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.
- [6] G. B. Dantzig, R. Fulkerson, and S. M. Johnson, “Solution of a large-scale traveling-salesman problem,” *Operations Research*, vol. 2, no. 4, pp. 393–410, Dec. 1954. doi: 10.1287/opre.2.4.393
- [7] N. Christofides, “Worst-case analysis of a new heuristic for the traveling salesman problem,” Tech. Rep. 388, Carnegie Mellon Univ., 1976.
- [8] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, no. 2, pp. 498–516, Mar.–Apr. 1973. doi: 10.1287/opre.21.2.498
- [9] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Hoboken, NJ: Wiley-Interscience, 1985.
- [10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proc. 26th Int. Conf. on Machine Learning (ICML)*, Montreal, QC, Canada, 2009, pp. 41–48. doi: 10.1145/1553374.1553380
- [11] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine Learning*, vol. 8, no. 3–4, pp. 229–256, May 1992. doi: 10.1007/BF00992696
- [12] G. Reinelt, “TSPLIB – A traveling salesman problem library,” *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, Nov. 1991. doi: 10.1287/ijoc.3.4.376
- [13] G. A. Croes, “A method for solving traveling-salesman problems,” *Operations Research*, vol. 6, no. 6, pp. 791–812, Dec. 1958. doi: 10.1287/opre.6.6.791

- [14] M. Held and R. M. Karp, “The traveling-salesman problem and minimum spanning trees,” *Operations Research*, vol. 18, no. 6, pp. 1138–1162, Nov.–Dec. 1970. doi: 10.1287/opre.18.6.1138
- [15] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton, NJ: Princeton Univ. Press, 2006.
- [16] O. Vinyals, M. Fortunato, and N. Jaitly, “Pointer networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, Montreal, QC, Canada, 2015. doi: 10.48550/arXiv.1506.03134
- [17] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 2017. doi: 10.48550/arXiv.1706.03762
- [18] W. Kool, H. van Hoof, and M. Welling, “Attention, Learn to Route: Transformers for Solving Combinatorial Optimization Problems,” in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Canada, 2019. doi: 10.48550/arXiv.1906.01227

## GitHub Αποθετήρια

- [19] W. Kool, H. van Hoof, and M. Welling, “Attention, Learn to Route,” *GitHub*, 2019. [Online]. Available: <https://github.com/wouterkool/attention-learn-to-route> [Accessed: March 2025].
- [20] X. Bresson, “TSP Transformer,” *GitHub*, 2021. [Online]. Available: [https://github.com/xbresson/TSP\\_Transformer](https://github.com/xbresson/TSP_Transformer) [Accessed: April 2025].
- [21] H. Yang, “tspFormer,” *GitHub*, 2023. [Online]. Available: <https://github.com/yhnju/tspFormer> [Accessed: May 2025].
- [22] D. Caffagni, “TransPormer,” *GitHub*, 2023. [Online]. Available: <https://github.com/dcaffo98/transpormer> [Accessed: May 2025].
- [23] R. Valdecy, “Metaheuristic Local Search 2-OPT,” *GitHub*, 2020. [Online]. Available: <https://github.com/Valdecy/Metaheuristic-Local-Search-2-opt> [Accessed: May 2025].
- [24] K. Nandan, “TSP Solver,” *GitHub*, 2021. [Online]. Available: <https://github.com/nandan7198/TSP-Solver> [Accessed: May 2025].
- [25] A. Zia, “E\_GTSP to TSP Tested Instances,” *GitHub*, 2021. [Online]. Available: [https://github.com/Zia-/E\\_GTSP\\_to\\_TSP\\_tested\\_instances](https://github.com/Zia-/E_GTSP_to_TSP_tested_instances) [Accessed: April 2025].