Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών

Συνδυασμός Τεχνικών Ενισχυτικής Μάθησης και Ευρετικών Αλγορίθμων για την Επίλυση του Προβλήματος του Πλανόδιου Πωλητή με Ανεφοδιασμό

Διπλωματική Εργασία

Λάζαρος Κεϊσίδης
ΑΕΜ: 9765

Επιβλέπων: Κωνσταντίνος Παπαλάμπρου
Επίκουρος Καθηγητής Α.Π.Θ.

Θεσσαλονίκη, Οκτώβριος 2024

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με τη χρήση μεθόδων ενισχυτικής μάθησης σε συνδυασμό με έναν ευρετικό αλγόριθμο βελτίωσης, με στόχο την επίλυση του προβλήματος Traveling Salesman Problem with Refueling (TSPWR), με έμφαση στην εύρεση βέλτιστης λύσης. Το πρόβλημα αυτό αποτελεί παραλλαγή του κλασικού Traveling Salesman Problem (TSP). Δύο εκδοχές του TSPWR (*πρόβλημα A* και *πρόβλημα B*) εξετάζονται αναλυτικά στην εργασία. Αρχικά, γίνεται εισαγωγή στις βασικές έννοιες της Μηχανικής Μάθησης, της Ενισχυτικής Μάθησης, καθώς και στους αλγόριθμους Q-learning και SARSA, που χρησιμοποιούνται για την επίλυση των προβλημάτων. Στη συνέχεια, παρουσιάζεται το Traveling Salesman πρόβλημα και οι δύο παραλλαγές του TSPWR, συνοδευόμενες από μαθηματική μοντελοποίηση, ανάλυση της πολυπλοκότητας των προβλημάτων και περιγραφή των τεσσάρων instances που χρησιμοποιήθηκαν στα πειράματα. Η μεθοδολογία που ακολουθήθηκε συνδυάζει τους αλγόριθμους ενισχυτικής μάθησης με τον ευρετικό αλγόριθμο βελτίωσης, και τα στάδια πειραματισμού αναλύονται σε βάθος. Τα αποτελέσματα των πειραμάτων συγκρίνονται με άλλες παρόμοιες έρευνες, ενώ καταγράφονται και τα συμπεράσματα της εργασίας.

Abstract

This thesis addresses the application of reinforcement learning methods in combination with a heuristic improvement algorithm, with the objective of solving the Traveling Salesman Problem with Refueling (TSPWR), emphasizing the pursuit of optimal solutions. The problem is a variation of the classical Traveling Salesman Problem (TSP). Two versions of the TSPWR (*Problem A* and *Problem B*) are examined in detail within the study. An introduction to the fundamental concepts of Machine Learning, Reinforcement Learning, and the algorithms Q-learning and SARSA—used to address the problems—is provided. Following this, the Traveling Salesman problem and the two variations of the TSPWR are presented, accompanied by mathematical modeling, an analysis of problem complexity, and a description of the four instances utilized in the experiments. The methodology employed combines reinforcement learning algorithms with a heuristic improvement algorithm, and the experimental stages are thoroughly analyzed. The results of the experiments are compared with other similar studies, and the conclusions of the research are documented.

Ευχαριστίες

Τελειώνοντας την διπλωματική μου εργασία κλείνει το κεφάλαιο το οποίο ξεκίνησε το Φθινόπωρο του 2019 με την είσοδο μου στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης. Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Κωνσταντίνο Παπαλάμπρου, τόσο για την ευκαιρία που μου έδωσε να συνεργαστώ μαζί του για την εκπόνηση της εργασίας αυτής, όσο και για την βοήθεια του κάτα την διάρκεια της όλης διαδικασίας. Ένα μεγάλο ευχαριστώ στους γονείς μου, Ηλία και Αναστασία, δίχως την στήριξη και την καθοδόγηση των οποίων δε θα μπορούσα να τα καταφέρω. Όλα μου τα επιτεύγματα είναι δικά τους επιτεύγματα. Στα μικρά μου αδέλφια, Νίκο, Βασίλη, Ροδή, Χρήστο και Αλέξανδρο-Ρωμανό για την παρουσία τους κατά την διάρκεια αυτού του μεγάλου ταξιδιού· ευχαριστώ. Η παρέα τους ήταν καθοριστική. Ακόμη, θα ήθελα να ευχαριστήσω του φίλους μου Μουρμάν και Μιχάλη για τις πολύωρες συζητήσεις μας γύρω από το θέμα της διπλωματικής μου και τον φίλο μου Γιώργο για τους υπολογιστικούς πόρους που μου πρόσφερε κάτα την διάρκεια των πειραμάτων. Τέλος, θα ήθελα να ευχαριστήσω τον αθλητή Taison Barcellos Freda, χάρη στις επιδόσεις του οποίου το καλοκαίρι μου ήταν ευχάριστο με αποτέλεσμα να δουλεύω με όρεξη καθημερινά πάνω στην διπλωματική εργασία.

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

1.1 Προβλήματα μικτού ακέραιου προγραμματισμου

Ένα πρόβλημα μικτού ακέραιου προγραμματισμού (Mixed Integer Programming ή MIP) είναι ένα είδος προβλήματος βελτιστοποίησης που περιλαμβάνει τόσο ακέραιες όσο και συνεχείς μεταβλητές. Ο στόχος είναι η βελτιστοποίηση (ελαχιστοποίηση ή μεγιστοποίηση) μιας γραμμικής αντικειμενικής συνάρτησης, με την επιβολή γραμμικών περιορισμών. Συγκεκριμένα, οι περιορισμοί αυτοί μπορούν να περιλαμβάνουν γραμμικές εξισώσεις ή ανισώσεις. Ορισμένες από τις μεταβλητές πρέπει να παίρνουν ακέραιες τιμές [1].

Τα Mixed Integer Programming (MIP) προβλήματα είναι ένας σημαντικός κλάδος των προβλημάτων συνδυαστικής βελτιστοποίησης (Combinatorial Optimization problems) και ανήκουν στην κατηγορία NP-hard. Επωφελούμενος από την ανάπτυξη της ακαδημαϊκής θεωρίας και του εμπορικού λογισμικού, ο mixed integer προγραμματισμός έχει καταστεί μια σημαντική δυνατότητα που τροφοδοτεί ένα ευρύ φάσμα εφαρμογών, όπως η κατανομή πόρων, το capacity planning, το bin packing, κ.λπ. Η πρόκληση στην επίλυση προβλημάτων MIP είναι το γεγονός πως η περιοχή εφικτών λύσεων είναι διακριτή και μη κυρτή, πράγμα που καθιστά την ανάλυση της δύσκολη και τον σχεδιασμό μεθόδουν βελτιστοποίησης απαιτητικό [2].

1.2 Μηχανική μάθηση (Machine learning)

Η μηχανική μάθηση είναι ένα υποσύνολο της τεχνητής νοημοσύνης (Artificial Intelligence) και περιλαμβάνει αλγόριθμους που επιτρέπουν υπολογιστές να μαθαίνουν και να παίρνουν αποφάσεις από δεδομένα. Ο στόχος της μηχανικής μάθησης είναι η ανάπτυξη μεθόδων ώστε να εντοπίζονται αυτόματα μοτίβα στα δεδομένα και στη συνέχεια η χρησιμοποίηση αυτών των μοτίβων για την πρόβλεψη μελλοντικών δεδομένων ή άλλων αποτελεσμάτων ενδιαφέροντος [3].

1.2.1 Ιστορική αναδρομή

- Το 1952 ο Arthur Samuel δημιουργεί ένα παιχνίδι προγραμματισμένο να παίζει ντάμα, το οποίο φαίνεται πως είναι το πρώτο πρόγραμμα στον κόσμο που μα-

θαίνει από μόνο του, και ως εκ τούτου αποτελεί μια πολύ πρώιμη επίδειξη της θεμελιώδους έννοιας της τεχνητής νοημοσύνης (AI) [4].

- Το 1958 ο Frank Rosenblatt παρουσιάσε έναν υπολογιστή στο μέγεθος ενός δωματίου, τον Perceptron, ο οποίος ήταν μία μηχανή εκμάθησης σχεδιασμένη να προβλέπει αν μία εικόνα ανήκει σε μία από δύο κατηγορίες, θέτοντας έτσι τα θεμέλεια για την τεχνητή νοημοσύνη [5].
- Το 1959 ο Arthur Samuel χρησιμοποιεί για πρώτη φορά τον όρο «μηχανική μάθηση» στο άρθρο του "Some studies in machine learning using the game of checkers". Συγκεκριμένα την ορίζει ως το πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί [6].
- Το 1963 ο Donald Michie δημιούργησε το πρόγραμμα MENACE, το οποίο έμαθε να παίζει το τέλειο παιχνίδι τρίλιζας [7].
- Το 1966 ο Joseph Weizenbaum δημιούργησε την ELIZA, ένα πρόργαμμα ικανό να συμμετάσχει σε συζητήσεις με ανθρώπους, κάνοντας τους να πιστεύουν ότι το λογισμικό έχει ανθρώπινα συναισθήματα. Ως εκ τούτου, η ELIZA ήταν ένα από τα πρώτα chatterbots («chatbot» στις μέρες μας) και ένα από τα πρώτα προγράμματα ικανά να επιχειρήσουν το Turing test [8].
- Το 1967 οι Thomas Cover και Peter Hart εξέλιξαν τον nearest neighbor αλγόριθμο, δίνοντας τη δυνατότητα αναγνώρισης προτύπων σε υπολογιστές. Χρησιμοποιήθηκε από πλανόδιους πωλητές (traveling salesmen) για σχεδιασμό αποτελεσματικών διαδρομών [9].
- Το 1969 ινστιτούτο έρευνας του Stanford ολοκλήρωσε το Shakey, το πρώτο κινητό ευφυές ρομπότ στον κόσμο που συνδύαζε AI, όραση υπολογιστή, δυνατότητες πλοήγησης και επεξεργασία φυσικής γλώσσας [10].
- Το 1973 ο James Lighthill δημοσίευσε την επιστημονική αναφορά ¨Artificial Intelligence: A General Survey¨, η οποία οδήγησε την Βρετανική κυβέρνηση να μειώσει σημαντικά τη συνεισφορά της σε έρευνες στον τομέα της τεχνητής νοημοσύνης [7].
- Το 1985 ο Terry Sejnowski δημιούργησε το πρόγραμμα NetTalk, το οποίο έμαθε να προφέρει λέξεις [7].
- Το 1989 ο Christopher Watkins ανέπτυξε τον αλγόριθμο Q-learning, ένας αλγόριθμος ενισχυτικής μάθησης που αναζητούσε την καλύτερη δυνατή ενέργεια (action) σε κάθε κατάσταση (state) που βρισκόταν [11].
- Το 1997 το πρόγραμμα Deep Blue της IBM νίκησε τον παγκόσμιο πρωταθλητή Gary Kasparov σε μία ιστορική σειρά παιχνιδιών σκάκι [12].
- Το 2010 οι Anthony Goldbloom και Ben Hamner δημιούργησαν την πλατφόρμα Kaggle για διαγωνισμούς μηχανικής μάθησης.
- Το 2014 η Facebook ανέπτυξε το λογισμικό DeepFace, το οποίο είχε την δυνατότητα να αναγνωρίζει ανθρώπινα πρόσωπα σε ψηφιακές εικόνες, με σχεδόν ανθρώπινη ακρίβεια [7].
- Το 2022 η OpenAI κυκλοφόρησε το ChatGPT.

1.2.2 Κατηγορίες μηχανικής μάθησης

Η μηχανική μάθηση κατηγοριοποιείται ανάλογα το είδος εκμάθησης (training) και το είδος των εργασιών που έχει σχεδιαστεί να εκτελεί. Οι βασικές κατηγορίες είναι:

i . **Εποπτευόμενη μάθηση (Supervised learning)**

Το κύριο χαρακτηριστικό της εποπτευόμενης μάθησης είναι η διαθεσιμότητα επισημασμένων δεδομένων εκπαίδευσης (labelled data), δηλαδή η υπάρξη δεδομένων που λειτουργούν ως παραδείγματα, για τα οποία γνωρίζουμε τη σωστή απάντηση ή τη σωστή ετικέτα. Το όνομα αναφέρεται στην ιδέα ενός ¨επόπτη¨ που καθοδηγεί τη μηχανή μάθησης σχετικά με τις ετικέτες που πρέπει να συσχετιστούν με τα παραδείγματα εκπαίδευσης. Συνήθως αυτές οι ετικέτες είναι κλάσεις σε προβλήματα ταξινόμησης. Έτσι η μηχανή μαθαίνει να ταξινομεί δεδομένα χωρίς ετικέτα (unlabelled data). Το Supervised Learning είναι η πιο σημαντική μεθοδολογία στον τομέα του ML [13].

ii . **Μη εποπτευόμενη μάθηση (Unsupervised learning)**

Οι αλγόριθμοι μη εποπτευόμενης μάθησης μπορούν να ανακαλύψουν ενδιαφέροντα και χρήσιμα μοτίβα σε δεδομένα χωρίς ετικέτα (unlabelled data), για αυτό του λόγο έχουν γίνει δημοφιλείς στους ερευνητές. Αυτοί οι αλγόριθμοι έχουν βρει πολλές εφαρμογές, όπως η αναγνώριση προτύπων, ανάλυση καλαθιού αγοράς (Market Basket Analysis), εξόρυξη ιστού (web mining), ανάλυση κοινωνικών δικτύων, ανάκτηση πληροφοριών, συστήματα συστάσεων (recommender systems), έρευνα αγοράς, ανίχνευση εισβολής και ανίχνευση απάτης [14].

iii . **Ενισχυτική μάθηση (Reinforcement learning)**

Στην ενισχυτική μάθηση (RL) ο agent (μηχανή μάθησης) μαθαίνει πως πρέπει να συμπεριφέρεται σε ένα περιβάλλον μέσω αλληλεπιδράσεων δοκιμής και σφάλματος (trial and error). Μέσα από αυτές τις αλληλεπιδράσεις δοκιμής και σφάλματος με το περιβάλλον ο agent μαθαίνει να βελτιώνει την απόδοση του [15].

iv . **Μάθηση με πολλαπλούς στόχους (Multi-task learning)**

Το Multi-Task Learning (MLT) είναι μία κατηγορία της μηχανικής μάθησης με στόχο την αξιοποίηση χρήσιμων πληροφοριών, από τον agent (μηχανή μάθησης), που προέρχονται από εκπαίδευση σε πολλές διαφορετικές δραστηριότητες σχετικές μεταξύ τους, ώστε να βελτιώσει την γενικότερη απόδοση του σε όλες τις δραστηριότητες. Με βάση την υπόθεση ότι όλες οι εργασίες, ή τουλάχιστον ένα υποσύνολο τους, σχετίζονται, η από κοινού εκμάθηση πολλαπλών εργασιών έχει βρεθεί εμπειρικά και θεωρητικά ότι οδηγεί σε καλύτερη απόδοση από την ανεξάρτητη εκμάθηση τους [16].

v . **Μάθηση μέσω μίμησης (Imitation learning)**

Οι Imitation Learning τεχνικές έχουν σκοπό την μίμηση της ανθρώπινης συμπεριφοράς σε μια δεδομένη δραστηριότητα. Ένας agent εκπαιδεύεται να εκτελεί μια εργασία μέσω επιδείξεων, μαθαίνοντας μία χαρτογράφηση μεταξύ παρατηρήσεων και ενεργειών. Η ιδέα του Imitation Learning υπάρχει εδώ και πολλά χρόνια, ωστόσο το πεδίο αυτό κερδίζει προσοχή πρόσφατα εξαιτίας

των τεχνολογικών εξελίξεων, αλλά και εξαιτίας της αυξανόμενης ζήτησης για έξυπνες εφαρμογές [17].

1.2.3 Hyperparameters

Οι αλγόριθμοι μηχανικής μάθησης περιλαμβάνουν υπερπαραμέτρους (hyperparameters) που πρέπει να ρυθμιστούν πριν από την έναρξη της εκμάθησης τους και έχουν σημαντική επιρροή στην απόδοση τους, καθώς ελέγχουν την διαδικασία εκμάθησης (learning proccess). Μερικές επιλογές για τη ρύθμιση υπερπαραμέτρων είναι: οι προεπιλεγμένες τιμές από το πακέτο λογισμικού, η χειροκίνητη διαμόρφωση από τον χρήστη (με βάση συστάσεις από την βιβλιογραφία, εμπειρία ή trial and error) ή η διαμόρφωσή τους για βέλτιστη προβλεπόμενη απόδοση με κάποια διαδικασία συντονισμού (tuning). Οι στρατηγικές συντονισμού υπερπαραμέτρων (hyperparameter tuning) είναι εξαρτημένες από τα δεδομένα. Μία άλλη επιλογή για την επιλογή υπερπαραμέτρων είναι οι διαδικασίες βελτιστοποίησης δεύτερου επιπέδου, οι οποίες προσπαθούν να ελαχιστοποιήσουν το αναμενόμενο σφάλμα γενίκευσης (expected generalization error) του αλγορίθμου σε έναν χώρο αναζήτησης υπερπαραμέτρων των εξεταζόμενων υποψηφίων διαμορφώσεων, συνήθως μέσω της αξιολόγησης προβλέψεων σε ένα ανεξάρτητο σύνολο δοκιμών ή με την εκτέλεση μιας μεθόδου επαναδειγματοληψίας, όπως η τεχνική της διασταύρωσης (cross-validation) [18]. Σε αυτή την διπλωματική εργασία μας απασχολούν δύο υπερπαραμέτροι (hyperparameters), το learning rate (ή αλλιώς step-size) και το discount factor.

***leanrning rate (a)**: Αντιπροσωπεύει τον ρυθμό μάθησης της μηχανής. Καθορίζει πόσο γρήγορα ο agent μαθαίνει από νέες πληροφορίες. Παίρνει τιμές από το 0 έως το 1, με το μηδέν να σημαίνει πως η μηχανή δε μαθαίνει τίποτα από τις νέες πληροφορίες και βασίζεται στις παλιές, και το 1 να σημαίνει πως η μηχανή βασίζεται εξ' ολοκλήρου στα νέα δεδομένα για την αξιολόγηση μίας ενέργειας.*

***discount factor (γ)**: Καθορίζει πόσο σημαντικές θεωρεί τις μελλοντικές ανταμοιβές (future rewards) η μηχανή μάθησης, σε σχέση με τις άμεσες ανταμοιβές (immediate rewards). Παίρνει τιμές από το 0 έως το 1. Για $\gamma = 0$ ο agent επιλέγει ενέργειες βασιζόμενος μονάχα στις άμεσες ανταμοιβές (immediate rewards), αδιαφορώντας για τυχών μελλοντικές. Αντίθετα, για $\gamma = 1$ ο agent δρα με μοναδικό κριτήριο τις μελλοντικές ανταμοιβές (future rewards).*

1.2.4 Συσχέτιση μηχανικής μάθησης με MIP προβλήματα

Γενικά η επίλυση ενός MIP προβλήματος είναι NP-hard. Πολλές τεχνικές έχουν σχεδιαστεί με σκοπό να βρίσκουν υψηλής ποιότητας λύσεις σε περιορισμένο χρόνο. Η μηχανική μάθηση (Machine Learning) έχει ενταχθεί πρόσφατα στις προσπάθειες για την επίλυση προβλημάτων βελτιστοποίησης, ειδικά για προβλήματα συνδυαστικής βελτιστοποίησης (CO). Μέθοδοι μηχανικής μάθησης έχουν δείξει στοιχεία επιτυχίας στην επίλυση ορισμένων NP-hard προβλημάτων συνδυαστικής βελτιστοποίησης. Έχει διαπιστωθεί πως η συνδυαστική χρήση αλγορίθμων μηχανικής μάθησης και ευρετικών (heuristic) αλγορίθμων είναι ένα ανερχόμενο θέμα στον ερευνητικό κόσμο [2].

1.3 Ενισχυτική μάθηση (Reinforcement learning)

Η ενισχυτική μάθηση (RL) χρονολογείται από τις αρχές της κυβερνητικής (cybernetics) και της έρευνας στα πεδία της στατιστικής, της ψυχολογίας, της νευροεπιστήμης και της επιστήμης των υπολογιστών. Όπως σημειώθηκε περιληπτικά παραπάνω, κάτα την ενισχυτική μάθηση (RL) η μηχανή μαθαίνει πως είναι δόκιμο να συμπεριφέρεται σε ένα περιβάλλον μέσω αλληλεπιδράσεων με αυτό. Η βασική ιδέα αυτού του είδους μάθησης είναι απλή, η μηχανή ανταμοίβεται ή τιμωρείται από τις ενέργειες της στο περιβάλλον και μαθαίνει με τον καιρό (training) ποιες ενέργειες είναι θεμιτές και ποιες όχι, ανάλογα την κατάσταση (state) στην οποία βρίσκεται [15]. Σε αυτή την εργασία οι μέθοδοι που θα ακολουθήσουμε για την επίλυση του προβλήματος είναι μέθοδοι RL.

Σε κάθε βήμα αλληλεπίδρασης με το περιβάλλον, η μηχανή λαμβάνει ως είσοδο μια ένδειξη της τρέχουσας κατάστασης, state, του περιβάλλοντος. Στη συνέχεια, επιλέγει μια ενέργεια, action, που παράγεται ως έξοδος. Η ενέργεια αυτή αλλάζει την κατάσταση στην οποία βρίσκεται μέσα στο περιβάλλον, και η τιμή αυτής της μετάβασης κατάστασης επικοινωνείται στον agent μέσω ενός βαθμωτού σήματος ενίσχυσης, reward. Η μηχανή οφείλει να επιλέγει ενέργειες που τείνουν να αυξήσουν τον μακροπρόθεσμο άθροισμα των τιμών του σήματος ενίσχυσης. Μπορεί να μάθει να το κάνει αυτό με τον χρόνο, μέσω συστηματικής δοκιμής και σφάλματος. Στη γενική μορφή του ένα RL μοντέλο περιλαμβάνει:

- Ένα διακριτό σύνολο καταστάσεων περιβάλλοντος, S.
- Ένα διακριτό σύνολο ενεργειών, A, που μπορεί να εκτελέσει η μηχανή μάθησης.
- Ένα σύνολο σημάτων ενίσχυσης, συνήθως τις τιμές 0 ή 1, ή πραγματικούς αριθμούς.

Ο ρόλος του agent είναι να βρει μια πολιτική (policy) π, που αντιστοιχίζει καταστάσεις σε ενέργειες, προκειμένου να μεγιστοποιήσει κάποιο μέτρο μακροπρόθεσμης ενίσχυσης. Συνήθως, αναμένουμε ότι το περιβάλλον θα είναι μη-προσδιοριστικό (non-deterministic), δηλαδή η ίδια ενέργεια στην ίδια κατάσταση σε δύο διαφορετικές περιπτώσεις μπορεί να οδηγήσει σε διαφορετικές επόμενες καταστάσεις και/ή διαφορετικές τιμές ενίσχυσης. Η ενισχυτική μάθηση διαφέρει από το ευρέως μελετημένο πρόβλημα της επιβλεπόμενης μάθησης (supervised learning) σε αρκετούς τομείς. Η πιο σημαντική διαφορά είναι ότι δεν υφίσταται τροφοδότηση της με ζευγάρια εισόδου/εξόδου. Αντ' αυτού, μετά την επιλογή μιας ενέργειας, ο agent ενημερώνεται για την άμεση ανταμοιβή και την επόμενη κατάσταση, αλλά δεν του λέγεται ποια ενέργεια θα ήταν στο μακροπρόθεσμο συμφέρον του. Είναι απαραίτητο για την μηχανή να συγκεντρώσει χρήσιμη εμπειρία σχετικά με τις δυνατές καταστάσεις του συστήματος, τις ενέργειες, τις μεταβάσεις και τις ανταμοιβές για να ενεργεί βέλτιστα. Μια άλλη διαφορά από την επιβλεπόμενη μάθηση είναι ότι η απόδοση σε πραγματικό χρόνο είναι σημαντική: η αξιολόγηση του συστήματος συνήθως συμβαίνει ταυτόχρονα με την μάθηση [15].

1.3.1 Πεπερασμένες μαρκοβιανές δαδικασίες αποφάσεων

Οι Μαρκοβιανές διαδικασίες αποφάσεων (MDPs) είναι μια κλασική μοντελοποίηση της αλληλουχίας λήψης αποφάσεων, όπου οι ενέργειες επηρεάζουν όχι μόνο τις άμεσες ανταμοιβές, αλλά και τις επακόλουθες καταστάσεις, και μέσω αυτών τις μελλοντικές ανταμοιβές. Οι Μαρκοβιανές διαδικασίες αποφάσεων (MDPs) αφορούν καθυστερημένες (μελλοντικές) ανταμοιβές, αλλά και την ανάγκη εξισορρόπησης μεταξύ άμεσων και μελλοντικών ανταμοιβών. Στις MDPs, εκτιμούμε την αξία $q*(s,a)$ κάθε ενέργειας a σε κάθε κατάσταση s, ή εκτιμούμε την αξία $v*(s)$ κάθε κατάστασης με δεδομένες τις βέλτιστες επιλογές ενεργειών. Οι MDPs αποτελούν μια απλή διατύπωση του προβλήματος της μάθησης μέσω αλληλεπίδρασης για επίτευξη κάποιου στόχου. Ο μαθητευόμενος λήπτης αποφάσεων ονομάζεται agent. Το στοιχείο με το οποίο αλληλεπιδρά, που περιλαμβάνει τα πάντα έξω από τον agent, ονομάζεται περιβάλλον. Αυτά αλληλεπιδρούν συνεχώς, με τον agent να επιλέγει ενέργειες και το περιβάλλον να ανταποκρίνεται σε αυτές τις ενέργειες και να παρουσιάζει νέες καταστάσεις στον πράκτορα. Το περιβάλλον επίσης παράγει ανταμοιβές, ειδικές αριθμητικές τιμές που ο πράκτορας επιδιώκει να μεγιστοποιήσει με την πάροδο του χρόνου μέσω της επιλογής των ενεργειών του. Σε μια πεπερασμένη MDP, τα σύνολα καταστάσεων, ενεργειών και ανταμοιβών (S, A και R) έχουν πεπερασμένο αριθμό στοιχείων [19].

1.3.2 Πολιτική (Policy) και συναρτήσεις αξίας (value functions)

Μια πολιτική καθορίζει τον τρόπο συμπεριφοράς του μαθητευόμενου agent σε μια δεδομένη στιγμή. Χονδρικά, μια πολιτική είναι μια αντιστοίχιση από τις αντιληπτές καταστάσεις του περιβάλλοντος στις ενέργειες που πρέπει να επιλεγούν όταν βρισκόμαστε σε αυτές τις καταστάσεις. Αντιστοιχεί σε αυτό που στην ψυχολογία ονομάζεται σύνολο κανόνων ή συνδέσεων ερεθίσματος-αντίδρασης. Σε μερικές περιπτώσεις, η πολιτική μπορεί να είναι μια απλή συνάρτηση ή πίνακας αναζήτησης, ενώ σε άλλες μπορεί να περιλαμβάνει εκτενή υπολογισμό, όπως μια διαδικασία αναζήτησης. Η πολιτική είναι ο πυρήνας ενός agent ενισχυτικής μάθησης, καθώς είναι η αυτή που καθορίζει τη συμπεριφορά του. Γενικά, οι πολιτικές μπορεί να είναι στοχαστικές, καθορίζοντας πιθανότητες για κάθε ενέργεια. Οι μέθοδοι ενισχυτικής μάθησης χωρίζονται σε δύο μεγάλες κατηγορίες, τις μεθόδους on-policy και τις μεθόδους off-policy. Οι μέθοδοι on-policy προσπαθούν να αξιολογήσουν ή να βελτιώσουν την πολιτική που χρησιμοποιείται για τη λήψη αποφάσεων, ενώ οι μέθοδοι off-policy αξιολογούν ή βελτιώνουν μια πολιτική διαφορετική από αυτήν που χρησιμοποιήθηκε για την παραγωγή των δεδομένων.

Οι συναρτήσεις αξίας (value functions) μιας πολιτικής αναθέτουν σε κάθε κατάσταση ή σε κάθε ζεύγος κατάστασης-ενέργειας την αναμενόμενη ανταμοιβή από εκείνη την κατάσταση ή εκείνο το ζεύγος κατάστασης-ενέργειας, δεδομένου ότι ο agent χρησιμοποιεί την πολιτική. Οι βέλτιστες συναρτήσεις αξίας αναθέτουν σε κάθε κατάσταση ή ζεύγος κατάστασης-ενέργειας την μεγαλύτερη αναμενόμενη επιστροφή που μπορεί να επιτευχθεί από οποιαδήποτε πολιτική. Εάν ο πράκτορας ακολουθεί την πολιτική π τη χρονική στιγμή t, τότε $\pi(a|s)$ είναι η πιθανότητα ότι $A_t = a$ αν $S_t = s$. Οι μέθοδοι ενισχυτικής μάθησης καθορίζουν πώς αλλάζει η πολιτική του agent ως αποτέλεσμα της εμπειρίας του [19].

Η συνάρτηση αξίας μιας κατάστασης s υπό μια πολιτική π, συμβολιζόμενη ως $v_\pi(s)$, είναι η αναμενόμενη απόδοση όταν ξεκινάμε από την κατάσταση s και ακολουθούμε

την πολιτική π στη συνέχεια. Για τις MDPs, ορίζουμε την συνάρτηση v_π ως:

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

Η αξία της ενέργειας a στη κατάσταση s σύμφωνα με μια πολιτική π συμβολίζεται ως $q_\pi(s, a)$ και την ορίζουμε ως την αναμενόμενη ανταμοιβή ξεκινώντας από την κατάσταση s, εκτελώντας την ενέργεια a και ακολουθώντας στη συνέχεια την πολιτική π:

$$q_\pi(s, a) = \mathbb{E}_\pi [G_t \mid S_t = s, A_t = a]$$

Όπου G_t αντιπροσωπεύει την άμεση ανταμοιβή ή τη συνολική σωρευμένη ανταμοιβή ξεκινώντας από το χρονικό βήμα τ. Ονομάζουμε το q_π τη συνάρτηση αξίας ενέργειας για την πολιτική π.

Η εξίσωση βελτιστοποίησης του Bellman [20], για την αξία μίας κατάστασης, εκφράζει το γεγονός ότι η αξία αυτής υπό μια βέλτιστη πολιτική πρέπει να ισούται με την αναμενόμενη ανταμοιβή για την καλύτερη ενέργεια σε αυτήν την κατάσταση:

$$v^*(s) = \max_a \sum_{s',r} p(s', r \mid s, a) [r + v^*(s')]$$

Όπου,

- $v^*(s)$: Η αξία της κατάστασης s υπό την βέλτιστη πολιτική.
- $\max_a$: Το μέγιστο, ως προς την ανταμοιβή, μεταξύ όλων των δυνατών ενεργειών a.
- $\sum_{s',r}$: Το άθροισμα για όλες τις επόμενες καταστάσεις s' και ανταμοιβές r.
- $p(s', r \mid s, a)$: Η πιθανότητα μετάβασης στη κατάσταση s' παίρνοντας ανταμοιβή r δεδομένου ότι η ενέργεια a επιλέχθηκε στην κατάσταση s.
- $r + v^*(s')$: Η άμεση ανταμοιβή r συν η αξία της επόμενης κατάστασης s'.

Τέλος, η εξίσωση βελτιστοποίησης του Bellman για την αξία ενός ζεύγους κατάστασης-ενέργειας είναι:

$$q^*(s, a) = \sum_{s',r} p(s', r \mid s, a) \left[r + \max_{a'} q^*(s', a') \right]$$

Όπου,

- $q^*(s, a)$: Η βέλτιση συνάρτηση αξίας ενέργειας για την κατάσταση s και την ενέργεια a.
- $\max_{a'} q^*(s', a')$: Η μέγιστη αξία της επόμενης κατάστασης s' για όλες τις δυνατές ενέργειες a'.

Οι δύο αλγόριθμοι ενισχυτικής μάθησης που θα χρησιμοποιήσουμε στην παρούσα διπλωματική εργασία χρησιμοποιούν την αρχή της εξίσωσης Bellman για την επαναλαμβανόμενη ενημέρωση των τιμών Q για κάθε ζεύγος κατάστασης-ενέργειας.

1.3.3 Εξερεύνηση vs Εκμετάλλευση (Exploration vs Exploitation)

Το trade-off μεταξύ εξερεύνησης και εκμετάλλευσης είναι μία θεμελιώδη έννοια της ενισχυτικής μάθησης. Μία ακόμα διαφορά μεταξύ RL και εποπτευόμενης μάθησης

είναι ότι μία μηχανή της πρώτης κατηγορίας πρέπει οπωσδήποτε να εξερευνήσει το περιβάλλον της. Ο agent ενδέχεται να πιστεύει ότι ένα συγκεκριμένο ζευγάρι κατάστασης-ενέργειας (state-action pair) είναι πολύ συμφέρων, θα επιλέγει πάντα την ίδια ενέργεια όταν βρίσκεται σε αυτή την κατάσταση ή θα πρέπει να δοκιμάσει μία διαφορετική ενέργεια, για την οποία έχει λιγότερες πληροφορίες; [15]

Η εξερεύνηση (exploration) είναι η διαδικασία δοκιμής εντελώς νέων περιοχών ενός χώρου αναζήτησης, ενώ η εκμετάλλευση (exploitation) είναι η διαδικασία επιλογής ενεργειών που προσφέρουν τη μεγαλύτερη ανταμοιβή κατά μέσο όρο. Για να είναι επιτυχής ένας αλγόριθμος πρέπει να καθιερωθεί μια καλή αναλογία μεταξύ εξερεύνησης και εκμετάλλευσης [21].

Μία απλή και αποτελεσματική μέθοδος για ισορρόπηση μεταξύ εξερεύνησης και εκμετάλευσης είναι η μέθοδος ϵ-greedy. Με πιθανότητα 1-ϵ η μηχανή μάθησης επιλέγει την greedy ενέργεια, δηλαδή την εκμετάλλευση, ενώ με πιθανότητα ϵ επιλέγη μία τυχαία ενέργεια, δηλαδή εξερεύνηση. Σε αυτή την μέθοδο ο agent δρα κατά κύριο λόγο με γνώμονα την εκμετάλλευση, παρόλα αυτά αραιά και που με μια μικρή πιθανότητα ϵ επιλέγει τυχαία ποια θα είναι η ενέργεια του.

$$\pi_\epsilon(a|s) = \begin{cases} \arg\max_a Q(s,a) & \text{with probability } 1-\epsilon \\ \text{random action} & \text{with probability } \epsilon \end{cases}$$

1.4 Q-learning

Ο αλγόριθμος Q-learning (Algorithm 1) είναι μια μορφή ενισχυτικής μάθησης, που χρησιμοποιεί την εξίσωση Bellman για να μάθει μια βέλτιστη συνάρτηση αξίας ενεργειών, και θεωρείται ένας model-free αλγόριθμος. Αυτό σημαίνει πως μαθαίνει να λαμβάνει αποφάσεις, αλληλεπιδρώντας άμεσα με το περιβάλλον, χωρίς να κατασκευάζει οπωσδήποτε ένα μοντέλο της δυναμικής του περιβάλλοντος. Αυτοί οι αλγόριθμοι έχουν ως στόχο την εκμάθηση μίας πολιτικής με βασιζόμενοι στις παρατηρούμενες ανταμοιβές και τις μεταβάσεις κατάστασης, αντί να προσπαθούν να προβλέψουν την επόμενη κατάσταση δεδομένες της τρέχουσας κατάστασης και ενέργειας. Παρέχει στις μηχανές μάθησης την ικανότητα να μάθουν να ενεργούν βέλτιστα σε μαρκοβιανού χαρακτήρα περιβάλλον βιώνοντας τις συνέπειες των ενεργειών. Είναι ένας off-policy αλγόριθμος.

Η διαδικασία εκμάθησης προχωρά παρόμοια με τη μέθοδο των χρονικών διαφορών (temporal differences) του Sutton (1984; 1988): ένας agent δοκιμάζει μια ενέργεια σε μια συγκεκριμένη κατάσταση και αξιολογεί τις συνέπειές της αναλόγως της άμεσης ανταμοιβής ή ποινής που λαμβάνει και της εκτίμησης του για την αξία της νέας κατάστασης στην οποία οδηγείται. Δοκιμάζοντας όλες τις ενέργειες σε όλες τις καταστάσεις επανειλημμένα, μαθαίνει ποιες είναι οι καλύτερες συνολικά, κρίνοντας από την μακροπρόθεσμη εκπτωτική (discounted) ανταμοιβή.

Συνοψίζοντας ο αλγόριθμος Q-learning είναι ένας απλός τρόπος ώστε μία μηχανή να μάθει να συμπεριφέρεται σε ένα μαρκοβιανό περιβάλλον [22]. Η μαθηματική εξίσωση που περιγράφει την ενημέρωση της ποιότητας (Q update) μίας ενέργειας σε μία κατάσταση σύμφωνα με τον αλγόριθμο είναι η εξής:

$$Q_{t+1}(\mathrm{s},\mathrm{a}) = Q_t(\mathrm{s},\mathrm{a}) + a\left[r(\mathrm{s},\mathrm{a}) + \gamma \max_{\mathrm{a'}} Q_t(\mathrm{s'},\mathrm{a'}) - Q_t(\mathrm{s},\mathrm{a})\right] \tag{1.1}$$

Όπου:

- $Q_{t+1}(\mathrm{s},\mathrm{a})$: αντιπροσωπεύει την ανανεωμένη (updated) Q-value.
- $Q_t(\mathrm{s},\mathrm{a})$: αντιπροσωπεύει την τωρινή Q-value.
- a: είναι το learning rate.
- $r(\mathrm{s},\mathrm{a})$: είναι η ανταμοιβή που έλαβε η μηχανή με την ενέργεια a στη κατάσταση s.
- γ: είναι το discount factor.
- $\max_{\mathrm{a'}} Q_t(\mathrm{s'},\mathrm{a'})$: η μέγιστη Q-value για την επόμενη κατάσταση s' από όλες τις δυνατές ενέργειες a'.

Algorithm 1 Q-learning Algorithm

1: Set the parameters: a, γ, and ϵ
2: **for** each pair (s, a) **do**
3: Initialize $Q(\mathrm{s},\mathrm{a}) = 0$
4: **end for**
5: Observe the state s
6: **repeat**
7: Select the action a using ϵ-greedy method
8: Take the action a
9: Receive immediate reward $r(\mathrm{s},\mathrm{a})$
10: Observe the new state s'
11: Update $Q(\mathrm{s},\mathrm{a})$ with Eq. (3)
12: Set s = s'
13: **until** the stopping criterion is satisfied

1.5 SARSA (State-Action-Reward-State-Action)

Ο αλγόριθμος SARSA (Algorithm 2) είναι ένας αλγόριθμος ενισχυτικής μάθησης, ο οποίος στοχεύει στο να μάθει η μηχανή μία πολιτική ώστε να παίρνει τη μέγιστη δυνατή συνολική ανταμοιβή σε ένα περιβάλλον. Ο SARSA σχετίζεται στενά με τον Q-learning, άλλα με μία σημαντική διαφορά στον τρόπο με τον οποίο ενημερώνει την συνάρτηση αξίας ενέργειας (action-value function), δηλαδή τον πίνακα Q-values. Ο SARSA είναι ένας αλγόριθμος on-policy, που σημαίνει ότι μαθαίνει την αξία της πολιτικής που ακολουθεί, συμπεριλαμβανομένων των ενεργειών εξερεύνησης. Ενημερώνει τη συνάρτηση αξίας του με βάση τη δράση που πραγματοποίησε η μηχανή, σε αντίθεση με την καλύτερη δυνατή ενέργεια (όπως στο Q-learning):

$$Q_{t+1}(\mathrm{s},\mathrm{a}) = Q_t(\mathrm{s},\mathrm{a}) + a\left[r(\mathrm{s},\mathrm{a}) + \gamma Q_t(\mathrm{s'},\mathrm{a'}) - Q_t(\mathrm{s},\mathrm{a})\right] \tag{1.2}$$

Algorithm 2 SARSA Algorithm

1: Set the parameters: a, γ, and ϵ
2: **for** each pair (s, a) **do**
3: Initialize the matrix Q(s, a) = 0
4: **end for**
5: Observe the state s
6: Select the action a using ϵ-greedy method
7: **repeat**
8: Take the action a
9: Receive immediate reward r(s, a)
10: Observe the new state s'
11: Select the new action a' using ϵ-greedy method
12: Update Q(s, a) with Eq. (2)
13: Set s = s'
14: Set a = a'
15: **until** the stopping criterion is satisfied

Η βασική διαφορά μεταξύ SARSA και Q-learning έγκειται στην ενημέρωση των Q-values. Στο Q-learning, η ενημέρωση γίνεται χρησιμοποιώντας τη μέγιστη δυνατή μελλοντική ανταμοιβή, ανεξάρτητα από την τρέχουσα πολιτική, καθιστώντας την Q-learning μέθοδο off-policy. Στο SARSA, η ενημέρωση πραγματοποιείται χρησιμοποιώντας την ενέργεια που εκτελεί πραγματικά ο agent, καθιστώντας την on-policy.

Κεφάλαιο 2

Πρόβλημα Πλανόδιου Πωλητή

2.1 Σύντομη Περιγραφή

Το πρόβλημα γνωστό και ως Traveling Salesman Problem (TSP) είναι ένα από τα δημοφιλέστερα προβλήματα συνδυαστικής βελτιστοποίησης και συχνά λαμβάνεται υπόψη στον σχεδιασμό διαδρομής αυτόνομων οχημάτων. Σε ένα TSP η ακολουθία κινήσεων του agent θα πρέπει να βελτιστοποιεί μια διαδρομή μεταξύ ενός συνόλου κόμβων. Επίσης, ο agent πρέπει να επισκεφτεί κάθε κόμβο (node) μόνο μία φορά και να επιστρέψει στο σημείο εκκίνησης τελειώνοντας την διαδρομή. Οι γενικεύσεις TSP περιλαμβάνουν διάφορες πτυχές της κινητής ρομποτικής, όπως περιορισμούς του οχήματος, δυναμικά περιβάλλοντα και πολλαπλά οχήματα [23].

Μια σημαντική ερευνητική περιοχή για σχεδιασμούς αυτόνομης διαδρομής οχημάτων λαμβάνει υπόψη τους περιορισμούς στα καύσιμα. Σε αυτές τις περιπτώσεις ο στόχος είναι να καθοριστεί μία διαδρομή που ακολουθώντας την το όχημα θα την τελειώσει δίχως να τελειώνουν τα καύσιμα του. Ακολουθώντας την ίδια λογική τα προβλήματα ανεφοδιασμού επιδιώκουν να βελτιστοποιήσουν τις δαπάνες για την αγορά καυσίμων για οδικές διαδρομές. Αυτό το πρόβλημα είναι μία υποκατηγορία του TSP, και ονομάζεται Traveling Salesman Problem With Refueling ή TSPWR [23].

Άλλες εφαρμογές του TSPWR είναι τα μη επανδρωμένα οχήματα και οι γεωσύγχρονοι δορυφόροι. Είναι σημαντικό να επισημανθεί ότι τα προβλήματα ανεφοδιασμού συνήθως χωρίζονται σε τέσσερις κύριες κατηγορίες: σε αυτά με σταθερή διαδρομή, σε αυτά με μεταβλητή διαδρομή, σε αυτά όπου το κόστος ανεφοδιασμού είναι σε κάθε κόμβο το ίδιο, και σε αυτά όπου το κόστος ανεφοδιασμού διαφέρει σε κάθε σημείο [23].

Στη συγκεκριμένη διπλωματική εργασία θα ασχοληθούμε με το πρόβλημα TSPWR, όπου κάθε σταθμός έχει διαφορετικό κόστους ανεφοδιασμού και η διαδρομή είναι μεταβλητή. Θα ερευνήσουμε δύο διαφορετικές εκδοχές του.

Πρόβλημα Α: *Ο agent ξεκινάει την διαδρομή του με άδεια δεξαμενή καυσίμων (fuel tank) και έχει την δυνατότητα να επιλέξει ο ίδιος ποιο θα είναι το σημείο εκκίνησης.*

Πρόβλημα Β: *Ο agent ξεκινάει την διαδρομή του με γεμάτη δεξαμενή καυσίμων και το σημείο εκκίνησης είναι προκαθορισμένο, η πόλη με δείκτη 0.*

Το πρόβλημα Β εξετάζεται για λόγους σύγκρισης αποτελεσμάτων με τη παρόμοια εργασία [23].

2.2 Μαθηματική μοντελοποίηση

Η μαθηματική μοντελοποίηση του προβλήματος, βασιζόμενη στο [23], περιέχει την μεταβλητή απόφασης $u_{i,j}$, η οποία άμα λάβει την τιμή 1 σημαίνει πως το δρομολόγιο (i,j) αποτελεί μέρος της λύσης, διαφορετικά αν λάβει την τιμή 0 όχι. Επίσης περιέχει την μεταβλητή απόφασης $z_{i,j}$ όπου παίρνει τιμή 1 μόνο όταν ο agent ξεμένει από καύσιμα μεταξή των κόμβων i και j.

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} \left(c_i l_i u_{i,j} + g_{i,j} z_{i,j} \right), \tag{2.1}$$

Υπό τους περιορισμούς:

$$\sum_{i=1}^{N} u_{i,j} = 1, \quad \text{για } j = 1, \ldots, N \tag{2.2}$$

$$\sum_{j=1}^{N} u_{i,j} = 1, \quad \text{για } i = 1, \ldots, N \tag{2.3}$$

$$f_i + l_i \leq L_{\max} u_{i,j}, \quad \text{για } i,j = 1, \ldots, N \tag{2.4}$$

$$l_i = (L_{\max} - f_i) w_i, \quad \text{για } i = 1, \ldots, N \tag{2.5}$$

$$u_{i,j},\ w_i,\ z_{i,j} \in \{0, 1\}, \quad \text{για } i,j = 1, \ldots, N \tag{2.6}$$

$$\{c_i, f_i, g_{i,j}, l_i, L_{\max}\} \geq 0, \quad \text{για } i,j = 1, \ldots, N \tag{2.7}$$

$$U = u_{i,j} \in V, \quad \text{για } i,j = 1, \ldots, N \tag{2.8}$$

Αναλυτικά:

- Το σύνολο N είναι το σύνολο των κόμβων (πόλεων).
- Το κόστος ανεφοδιασμού στην πόλη i είναι c_i.
- l_i είναι η ποσότητα καυσίμου που συμπληρώθηκε στην τοποθεσία i.
- Το επιπλέον κόστος που προστίθεται στο συνολικό κόστος, εφόσον ο agent ξεμείνει από καύσιμα μεταξύ i και j, συμβολίζεται με $g_{i,j}$.
- Η εξίσωση 2.1 είναι η αντικειμενική συνάρτηση, όπου το συνολικό κόστος δίνεται από το άθροισμα του κόστους ανεφοδιασμού και του επιπλέον κόστους εφόσον ξεμείνει από καύσιμα, και πρέπει να ελαχιστοποιηθεί.
- Οι εξισώσεις 2.2 και 2.3 εξασφαλίζουν ότι κάθε τοποθεσία επισκέπτεται μία φορά.

- Η ανίσωση 2.4 εξασφαλίζει πως η ποσότητα καυσίμου στην δεξαμενή του agent δεν ξεπερνά την μέγιστη χωρητικότητα $L_{\max}$, όπου f_i είναι η ποσότητα καυσίμου όταν φτάνει στην πόλη i.
- Η εξίσωση 2.5 εξασφαλίζει πως σε κάθε ανεφοδιασμό γεμίζει εξ ολοκλήρου η δεξαμενή καυσίμου, όπου με w_i συμβολίζεται η μεταβλητή απόφασης που όταν ισούται με 1 σημαίνει πως στην τοποθεσία i πραγματοποιήθηκε ανεφοδιασμός.
- Η εξίσωση 2.6 και η ανίσωση 2.7 εξασφαλίζουν πως οι μεταβλητές $u_{i,j}$, w_i, $z_{i,j}$ είναι δυαδικές και αντίστοιχα πως οι άλλες είναι μη αρνητικές.
- Τέλος, στην εξίσωση 2.8 το σύνολο V αντιπροσωπεύει οποιοδήποτε σύνολο περιορισμών που εξαλείφουν το σχηματισμό υποδιαδρομών (sub-routes).

2.3 Instances

Σε αυτή την εργασία μελετάμε 4 διαφορετικά σύνολα πόλεων (instances), Bahia30D, Minas24D, Minas30D, Minas57D. Στους Πίνακες 2.1, 2.2, 2.3, 2.4 δίνονται τα απαραίτητα δεδομένα κάθε συνόλου πόλεων που εξετάζουμε, τα οποία αποκτήθηκαν έπειτα από επικοινωνία με τους συγγραφείς του επιστημονικού άρθρου [23]. Οι τιμές καυσίμου που δίνονται είναι σε R\$ (βραζιλιάνικο νόμισμα), αλλά τα αποτελέσματα στη πορεία θα μετατραπούν σε € σύμφωνα με την αναλογία $1R\$ = 0.1631€$.

ID	City	Latitude	Longitude	Fuel cost per litre
0	Alagoinhas	-12.13556	-38.41917	3.307
1	Barreiras	-12.15278	-44.99000	3.654
2	Brumado	-14.20361	-41.66528	3.646
3	Caetite	-14.06944	-42.47500	3.688
4	Camaçari	-12.69750	-38.32417	3.358
5	Eunápolis	-16.37750	-39.58028	3.526
6	Feira de Santana	-12.26667	-38.96667	3.346
7	Guanambi	-14.22333	-42.78139	3.620
8	Ilhéus	-14.78889	-39.04944	3.761
9	Ipirá	-12.15833	-39.73722	3.304
10	Irecê	-11.30417	-41.85583	3.650
11	Itabuna	-14.78556	-39.28028	3.599
12	Itamaraju	-17.03917	-39.53111	3.520
13	Jacobina	-11.18056	-40.51833	3.592
14	Jaguaquara	-13.53056	-39.97083	3.336
15	Jequié	-13.85750	-40.08361	3.597
16	Juazeiro	-9.41167	-40.49861	3.668
17	Lauro de Freitas	-12.89444	-38.32722	3.270
18	Livramento de Nossa Senhora	-13.64306	-41.84056	3.721
19	Paulo Afonso	-9.40611	-38.21472	3.683
20	Poções	-14.52972	-40.36528	3.380
21	Porto Seguro	-16.44972	-39.06472	4.067
22	Salvador	-12.97111	-38.51083	3.399
23	Santo Antônio de Jesus	-12.96889	-39.26139	3.340
24	Senhor do Bonfim	-10.46139	-40.18944	3.481
25	Serrinha	-11.66417	-39.00750	3.443
26	Simões Filho	-12.78444	-38.40389	3.367
27	Teixeira de Freitas	-17.53500	-39.74194	3.545
28	Valença	-13.37028	-39.07306	3.532
29	Vitória da Conquista	-14.86611	-40.83944	3.291

2.1: Δεδομένα Bahia30D

ID	City	Latitude	Longitude	Fuel cost per litre
0	Araguari	-18.3850	-48.1114	3.321
1	Belo Horizonte	-19.5515	-43.5616	3.471
2	Betim	-19.5804	-44.1154	3.408
3	Campo Belo	-20.5350	-45.1638	3.433
4	Contagem	-19.5554	-44.0313	3.393
5	Formiga	-20.2752	-45.2535	3.418
6	Governador Valadares	-18.5104	-41.5658	3.366
7	Guaxupé	-21.1819	-46.4246	3.446
8	Itabira	-19.3709	-43.1337	3.476
9	Ituiutaba	-18.5808	-49.2754	3.437
10	Juiz de Fora	-21.4551	-43.2101	3.307
11	Monte Carmelo	-18.4329	-47.2955	3.428
12	Montes Claros	-16.4406	-43.5142	3.458
13	Oliveira	-20.4147	-44.4938	3.361
14	Patos de Minas	-18.3444	-46.3105	3.526
15	Poços de Caldas	-21.4716	-46.3341	3.613
16	Pouso Alegre	-22.1348	-45.5611	3.453
17	Sete Lagoas	-19.2757	-44.1448	3.238
18	Teófilo Otoni	-17.5127	-41.3019	3.443
19	Três Corações	-21.4149	-45.1512	3.735
20	Uberaba	-19.4454	-47.5555	3.510
21	Uberlândia	-18.5507	-48.1638	3.476
22	Unaí	-16.2127	-46.5422	3.486
23	Varginha	-21.3305	-45.2549	3.511

2.2: Δεδομένα Minas24D

ID	City	Latitude	Longitude	Fuel cost per litre
0	Araguari	-18.3850	-48.1114	3.321
1	Araxá	-19.3536	-46.5626	3.399
2	Barbacena	-21.1333	-43.4625	3.475
3	Belo Horizonte	-19.5515	-43.5616	3.471
4	Betim	-19.5804	-44.1154	3.408
5	Campo Belo	-20.5350	-45.1638	3.433
6	Contagem	-19.5554	-44.0313	3.393
7	Divinópolis	-20.0820	-44.5302	3.507
8	Formiga	-20.2752	-45.2535	3.418
9	Governador Valadares	-18.5104	-41.5658	3.366
10	Guaxupé	-21.1819	-46.4246	3.446
11	Ipatinga	-19.2806	-42.3212	3.483
12	Itabira	-19.3709	-43.1337	3.476
13	Ituiutaba	-18.5808	-49.2754	3.437
14	Juiz de Fora	-21.4551	-43.2101	3.307
15	Lavras	-21.1443	-44.5959	3.774
16	Monte Carmelo	-18.4329	-47.2955	3.428
17	Montes Claros	-16.4406	-43.5142	3.458
18	Oliveira	-20.4147	-44.4938	3.361
19	Passos	-20.4308	-46.3635	3.657
20	Patos de Minas	-18.3444	-46.3105	3.526
21	Poços de Caldas	-21.4716	-46.3341	3.613
22	Pouso Alegre	-22.1348	-45.5611	3.453
23	Sete Lagoas	-19.2757	-44.1448	3.238
24	Teófilo Otoni	-17.5127	-41.3019	3.443
25	Três Corações	-21.4149	-45.1512	3.735
26	Uberaba	-19.4454	-47.5555	3.510
27	Uberlândia	-18.5507	-48.1638	3.476
28	Unaí	-16.2127	-46.5422	3.486
29	Varginha	-21.3305	-45.2549	3.511

2.3: Δεδομένα Minas30D

2.4: Δεδομένα Minas57D

ID	City	Latitude	Longitude	Fuel cost per litre
0	Alfenas	-21.4292	-45.9472	3.624
1	Araguari	-18.3850	-48.1114	3.321
2	Araxá	-19.3536	-46.5626	3.399
3	Barbacena	-21.1333	-43.4625	3.475
4	Belo Horizonte	-19.5515	-43.5616	3.471
5	Betim	-19.5804	-44.1154	3.408
6	Bom Despacho	-19.7364	-45.2522	3.249
7	Campo Belo	-20.5350	-45.1638	3.433
8	Caratinga	-19.7897	-42.1392	3.429
9	Congonhas	-20.4997	-43.8578	3.557
10	Conselheiro Lafaiete	-20.6603	-43.7861	3.629
11	Contagem	-19.9317	-44.0536	3.393
12	Coronel Fabriciano	-19.5186	-42.6289	3.668
13	Curvelo	-18.7564	-44.4308	3.288
14	Divinópolis	-20.0820	-44.5302	3.507
15	Formiga	-20.2752	-45.2535	3.418
16	Frutal	-20.0247	-48.9406	3.583
17	Governador Valadares	-18.5104	-41.5658	3.366
18	Guaxupé	-21.1819	-46.4246	3.446
19	Ipatinga	-19.2806	-42.3212	3.483
20	Itabira	-19.3709	-43.1337	3.419
21	Itajubá	-22.4256	-45.4528	3.456
22	Itaúna	-20.0753	-44.5764	3.444
23	Ituiutaba	-18.5808	-49.2754	3.476
24	Janaúba	-15.8025	-43.3089	3.586
25	Januária	-15.4881	-44.3619	3.726
26	João Monlevade	-19.8100	-43.1736	3.421
27	João Pinheiro	-17.7425	-46.1725	3.533
28	Juiz de Fora	-21.4551	-43.2101	3.437
29	Lavras	-21.1443	-44.5959	3.774
30	Leopoldina	-21.5319	-42.6431	3.287
31	Manhuaçu	-20.2581	-42.0336	3.422
32	Monte Carmelo	-18.4329	-47.2955	3.307
33	Montes Claros	-16.4406	-43.5142	3.428
34	Muriaé	-21.1306	-42.3664	3.458
35	Nova Lima	-19.9856	-43.8467	3.724
36	Oliveira	-20.4147	-44.4938	3.361
37	Ouro Preto	-20.2875	-43.5081	3.720
38	Pará de Minas	-19.8603	-44.6083	3.526
39	Paracatu	-17.2222	-46.8747	3.656
40	Passos	-20.4308	-46.3635	3.657
41	Patos de Minas	-18.3444	-46.3105	3.471
42	Patrocínio	-18.9439	-46.9925	3.608
43	Poços de Caldas	-21.4716	-46.3341	3.613
44	Pouso Alegre	-22.1348	-45.5611	3.453
45	Sabará	-19.8864	-43.8067	3.532
46	São João Del Rei	-21.1357	-44.2617	3.712
47	São Sebastião do Paraíso	-20.9169	-46.9914	3.529
48	Sete Lagoas	-19.2757	-44.1448	3.238
49	Teófilo Otoni	-17.5127	-41.3019	3.443
50	Timóteo	-19.5825	-42.6444	3.459
51	Três Corações	-21.4149	-45.1512	3.735
52	Ubá	-21.1200	-42.9428	3.545
53	Uberaba	-19.4454	-47.5555	3.510
54	Uberlândia	-18.5507	-48.1638	3.476
55	Unaí	-16.2127	-46.5422	3.486
56	Varginha	-21.3305	-45.2549	3.511

2.4 Πολυπλοκότητα προβλημάτων

Ο αριθμός όλων των δυνατών αποφάσεων που μπορεί να πάρει η μηχανή μάθησης κάτα την εκπαίδευση είναι εντυπωσιακά μεγάλος. Στο *πρόβλημα A* όλοι οι δυνατοί συνδυασμοί διαδρομών είναι $N!$ (όπου N είναι ο αριθμός πόλεων του εξεταζόμενου συνόλου), σε κάθε σταθμό όμως έχει την δυνατότητα να επιλέξει ανάμεσα στον ανεφοδιασμό και στον μη ανεφοδιασμό, εκτός από την πρώτη πόλη όπου η μόνη επιλογή του είναι να γεμίσει την δεξαμενή καυσίμου, καθώς αυτή είναι άδεια. Όλες οι δυνατές αποφάσεις ανεφοδιασμού ή μη είναι $2^{(N-1)}$. Επομένως στο *πρόβλημα A* όλες οι πιθανές αποφάσεις ή όλα τα πιθανά iterations είναι $N! \times 2^{(N-1)}$. Αντιθέτως στο *πρόβλημα B* όλοι οι δυνατοί συνδυασμοί διαδρομών είναι $(N-1)!$ καθώς η αρχική πόλη είναι σταθερή. Οι πιθανές αποφάσεις ανεφοδιασμού παραμένουν οι ίδιες, καθώς πάλι στην πρώτη πόλη δε του δίνεται η δυνατήτα επιλογής, αφού ξεκινάει με γεμάτη δεξαμενή καυσίμων. Άρα, στο *πρόβλημα B* τα πιθανά iterations είναι $(N-1)! \times 2^{(N-1)}$. Στον Πίνακα 2.5 αναγράφονται όλες οι δυνατές αποφάσεις που μπορεί να πάρει η μηχανή μάθησης στα δύο προβλήματα, ανάλογα τον συνολικό αριθμό πόλεων.

2.5: TSPWR πολυπλοκότητα.

N	**Problem A**	**Problem B**
5	1920	384
10	1.858e+09	1.858e+08
24	5.205e+30	2.169e+29
30	1.424e+41	4.747e+39
57	2.920e+93	5.123e+76
100	5.915e+187	5.915e+95

Σε ένα run των αλγόριθμων ενισχυτικής μάθησης που θα χρησιμοποιήσουμε η πολυπλοκότητα τους μπορεί να εκφραστεί ως το γινόμενο $N \times E$, όπου N ο αριθμός πόλεων του instance και E ο αριθμός επεισοδίων, ο οποίος στα πειράματα μας είναι 10000. Δηλαδή αριθμός ενεργειών που πραγματοποιεί η μηχανή μάθησης μέχρι να παρουσιάσει την τελική λύση είναι κάτα πολύ μικρότερος από όλες τις δυνατές διαφορετικές ενέργειες που θα μπορούσε να εκτελέσει, όπως φαίνεται στον Πίνακα 2.6.

2.6: Διαφορά (diff.) μεταξύ του αριθμού όλων των πιθανών αποφάσεων στο TSPWR και του αριθμού των iteration που εξερευνά ο αλγόριθμος RL-TSPWR σε 10000 επεισόδια.

N	**RL complexity**	**Diff. A**	**Diff. B**
5	50000	−48,080	−49,616
10	100000	1.8579e+09	1.857e+08
24	240000	5.205e+30	2.169e+29
30	300000	1.424e+41	4.747e+39
57	570000	2.920e+93	5.123e+76
100	1000000	5.915e+187	5.915e+95

Κεφάλαιο 3

Μεθοδολογία

Σκοπός της εργασίας είναι να δείξουμε πως η συνδυαστική χρήση αλγορίθμων RL και ευρετικών αλγορίθμων μπορεί να δώσει πολύ καλές λύσεις στο TSPWR. Καθοριστικό ρόλο για την απόδοση ενός αλγορίθμου RL παίζουν οι παράμετροι a, γ, ϵ, reward function οι οποίες ρυθμίζονται πριν την διαδικασία της εκμάθησης. Αφού τεθούν οι παράμετροι RL, ξεκινάει η διαδικασία της εκμάθησης, όταν αυτή τελειώσει παρουσιάζεται η καλύτερη λύση που βρέθηκε από τον αλγόριθμο RL και αυτή η λύση βελτιώνεται περαιτέρω μέσω του ευρετικού αλγόριθμου. Στο πρώτο στάδιο πειραμάτων στόχος είναι να βρεθούν οι συνδυασμοί παραμέτρων που αποδίδουν καλύτερα σε κάθε εξεταζόμενη περίπτωση. Στο δεύτερο στάδιο της εργασίας επικεντρωνόμαστε σε αυτούς τους συνδυασμούς παραμέτρων, ώστε να παρουσιάσουμε στο τέλος τις καλύτερες προτεινόμενες λύσεις αναλυτικά.

3.1 Περιγραφή μοντέλου και περιβάλλοντος

Για την υλοποίηση των πειραμάτων θα πρέπει να θέσουμε κάποιες βασικές παραμέτρους του προβλήματος εξ αρχής. Αυτές οι παράμετροι για λόγους σύγκρισης με την εργασία [23] θα είναι: *μέγιστη χωριτηκότητα δεξαμενής καυσίμου* $L_{\max}$ = 150 l και *μέση κατανάλωση καυσίμου* = 7 km/l. Επίσης το περιβάλλον στο οποίο θα εκπαιδεύονται οι μηχανές μάθησης, θα πρέπει να περιέχει τιμωρίες και ανταμοιβές ώστε να μαθαίνει σιγά σιγά κάθε agent πως οφείλει να δρα μέσα σε αυτό. Στο *πρόβλημα A* οι αλγόριθμοι που θα παρουσιαστούν παρακάτω ελέγχουν κάθε πόλη ενός instance ως πιθανή αρχική πόλη, ενώ στο *πρόβλημα B* η αρχική πόλη είναι σταθερή και είναι αυτή με id = 0 για κάθε instance.

3.1.1 Τιμωρία σε περίπτωση μη αποδεκτής συμπεριφοράς

Ένας από τους βασικότερους, αν όχι ο βασικότερος στόχος της εκπαίδευσης είναι να μάθει η μηχανή να αποφεύγει, όσο αυτό είναι εφικτό, να ξεμένει από καύσιμα στη μέση μίας διαδρομής από μία πόλη i σε μία άλλη πόλη j. Για να πραγματοποιηθεί αυτό πρέπει να εισάγουμε κάποια μορφή τιμωρίας όταν αυτό συμβαίνει, ώστε να μάθει να το αποφεύγει. Στην εργασία [23] αυτή η τιμωρία παίρνει την μορφή σταθερού κόστους 200R$ (βραζιλιάνικο νόμισμα). Κάθε φορά που ο agent ξεμένει από καύσιμα στη μέση της διαδρομής το κόστος αυτό προστίθεται στο συνολικό, γεμίζει την δεξαμενή καυσίμων

και ξανά-πραγματοποιεί την διαδρομή (i,j). Στην εργασία αυτή θα ακολουθήσουμε μία διαφορετική προσέγγιση. Κάθε φορά που ο agent θα ξεμένει από καύσιμα στη μέση της διαδρομής (i,j) θα πραγματοποιεί ανεφοδιασμό με πολύ μεγαλύτερο κόστος ανά λίτρο καυσίμου, ενδιάμεσα των δύο πόλεων, βάζοντας καύσιμα αρκετά ώστε να φτάσει στην πόλη j με άδεια δεξαμενή καυσίμου. Αυτό είναι το επιπλέον κόστος ανεφοδιασμού που περιγράφτηκε στο μαθηματικό μοντέλο του προβλήματος ως $g_{i,j}$ και ισούται με 20R$ / λίτρο.

3.1.2 Υπολογισμός αποστάσεων

Ο υπολογισμός αποστάσεων μεταξύ των πόλεων είναι μεγάλης σημασίας για την πραγματοποίηση του εγχειρήματος. Χρησιμοποιώντας της συντεταγμένες γεωγραφικού πλάτους (latitude) και γεωγραφικού μήκους (longitude) που δίνονται για κάθε πόλη, και χρησιμοποιώντας την μέθοδο Haversine [24] φτιάχνουμε έναν πίνακα αποστάσεων για κάθε instance, ο οποίος περιέχει όλες τις αποστάσεις μεταξύ των εκάστοτε πόλεων.

Αναλυτικά, η μέθοδος Haversine υπολογίζει την απόσταση μεταξύ δύο σημείων στην επιφάνεια μίας σφαίρας, με γνωστά τα γεωγραφικά πλάτη και μήκη. Έστω ότι έχουμε δύο σημεία A και B με τις εξής συντεταγμένες:

$$A = (\text{lat}_1, \text{lon}_1)$$
$$B = (\text{lat}_2, \text{lon}_2)$$

Η απόσταση των δύο σημείων σύμφωνα με αυτή τη μέθοδο δίνεται ως:

$$d = 2R \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\Delta\text{lon}}{2}\right)}\right) \tag{3.1}$$

όπου:

$$\Delta\text{lat} = \text{lat}_2 - \text{lat}_1$$
$$\Delta\text{lon} = \text{lon}_2 - \text{lon}_1$$
$$R = \text{Ακτίνα Γης } (= 6371 \text{ χλμ})$$

Η μέθοδος έχει τα εξής βηματα:

1. Μετατροπή γεωγραφικού πλάτους και μήκους από μοίρες σε ακτίνια.
2. Υπολογισμός των διαφορών Δlat και Δlon ανάμεσα στα γεωγραφικά πλάτη και γεωγραφικά μήκη.
3. Εφαρμογή του τύπου 3.1 για την εύρεση της απόστασης μεταξύ των δύο σημείων.

Algorithm 3 Haversine Distance

1: Convert lat_1, lat_2, lon_1, lon_2 from degrees to radians
2: Set $\Delta\text{lat} = \text{lat}_2 - \text{lat}_1$ and $\Delta\text{lon} = \text{lon}_2 - \text{lon}_1$
3: Set $a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\Delta\text{lon}}{2}\right)$
4: Set $c = 2 \cdot \arcsin\left(\sqrt{a}\right)$
5: Return $d = R \cdot c$

3.2 Μεθοδολογία ενισχυτικής μάθησης

Όπως αναφεραμε παραπάνω, στο πρώτο στάδιο των πειραμάτων, θα ασχοληθούμε με δύο διαφορετικές εκδοχές του non-uniform TSPWR, το *πρόβλημα A* και το *πρόβλημα B*, έτσι όπως τα παρουσιάσαμε. Για κάθε μία εκδοχή θα ερευνήσουμε τα 4 διαφορετικά instances που παρουσιάστηκαν, Bahia30D, Minas24D, Minas30D, Minas57D και σε κάθε ένα από αυτά αυτά θα εφαρμόσουμε δύο διαφορετικούς αλγόριθμους ενισχυτικής μάθησης, τον Q-learning και τον SARSA. Οι παράμετροι που θα εξεταστούν είναι:

$$a = \Big[0.01, \quad 0.15, \quad 0.30, \quad 0.45, \quad 0.60, \quad 0.75, \quad 0.90, \quad 0.99\Big]$$

$$\gamma = \Big[0.01, \quad 0.15, \quad 0.30, \quad 0.45, \quad 0.60, \quad 0.75, \quad 0.90, \quad 0.99\Big]$$

$$\epsilon = \Big[0.01, \quad 0.05, \quad 0.1\Big]$$

Επίσης οι συναρτήσεις ανταμοιβής (reward functions) που θα χρησιμοποιηθούν για την ενισχυτική μάθηση των μηχανών είναι οι εξής:

$$R_1 = -(d_{i,j} + c_i + g_{i,j} z_{i,j}) \tag{3.2}$$

$$R_2 = -(c_i + g_{i,j} z_{i,j}) \tag{3.3}$$

$$R_3 = -(d_{i,j} + c_i + g_{i,j} z_{i,j} + k_{i,j}) \tag{3.4}$$

Όπου $d_{i,j}$ είναι η απόσταση μεταξύ των κόμβων (πόλεων) i και j, c_i το κόστος ανεφοδιασμού (refueling cost) στην πόλη i, $g_{i,j}$ το επιπλέον κόστος ανεφοδιασμού μεταξύ των πόλεων i και j εφόσον ο agent ξεμείνει από καύσιμα σε αυτή τη διαδρομή, δηλαδή όταν η μεταβλητή $z_{i,j}$ ισούται με μονάδα, και τέλος $k_{i,j}$ η ποσότητα καυσίμου που δαπανήθηκε για την μετακίνηση από τον κόμβο i έως τον κόμβο j.

Τα συνολικά διαφορετικά πειράματα που θα εκτελεστούν στο αρχικό στάδιο της εργασίας είναι το σύνολο των δυνατών συνδυασμών όλων των παραπάνω, δηλαδή 2 προβλήματα × 2 αλγόριθμοι RL × 4 instances × 8 διαφορετικά a × 8 διαφορετικά γ × 3 διαφορετικά ϵ × 3 διαφορετικά reward functions. Συνολικά θα εκτελεστούν 9216 διαφορετικά πειράματα, το κάθε ένα θα εκτελεστεί 50 φορές (runs) ώστε τα αποτελέσματα που προκύπτουν να μην είναι τυχαία. Οι αλγόριθμοι ενισχυτικής μάθησης που θα χρησιμοποιήσουμε είναι δύο, Q-learning και SARSA.

3.2.1 Q-learning στο TSPWR

Σε αυτό το κομμάτι θα παρουσιάσουμε και θα αναλύσουμε τον πρώτο αλγόριθμο ενισχυτικής μάθησης (Algorithm 4), ο οποίος βασίζεται στην ενημέρωση των τιμών Q μέσω της σχέσης 1.1. Το κριτήριο διακοπής (σειρά 36) του παρακάτω αλγόριθμου είναι τα 10000 επεισόδια, όπου σε κάθε ένα ενημερώνεται (update) ο πίνακας τιμών Q. Σε κάθε run ο αλγόριθμος ξεκινάει από την αρχή μηδενίζοντας τον πίνακα των τιμών Q. Για κάθε συνδυασμό παραμέτρων κάθε instance πραγματοποιούμε 50 runs.

Ο αλγόριθμος ξεκινάει αρχικοποιώντας τις παραμέτρους a, γ, και ϵ, ενώ επίσης εξ αρχής έχει γίνει η επιλογή συνάρτησης ανταμοιβής (reward function). Στη συνέχεια αρχικοποιεί ως μηδενικό τον πίνακα με τις τιμές Q, αυτός ο πίνακας είναι ένας πίνακας με διαστάσεις *αριθμός πόλεων* × *αριθμός πόλεων* × 2, όπου η τελευταία διάσταση του αφορά τον ανεφοδιασμό. Για παράδειγμα, η τιμή στη θέση [1,2,0] του πίνακα αποθηκεύει την τιμή Q για την διαδρομή (1,2) χωρίς ανεφοδιασμό στην πόλη 1, ενώ η τιμή στη θέση [1,2,1] αποθηκεύει την τιμή για την ίδια διαδρομή αλλά με ανεφοδιασμό στην πόλη 1. Αυτή η τρίτη διάσταση του πίνακα τιμών Q είναι μία σημαντική διαφοροποίηση σε σχέση με την παρόμοια εργασία [23], καθώς ο agent ¨σκέφτεται¨ πότε αξίζει ο ανεφοδιασμός και πότε όχι και δεν επιλέγει να ανεφοδιάσει μηχανικά όταν η στάθμη του πέφτει κάτω από ένα επίπεδο. Με αυτόν τον τρόπο μειώνονται σημαντικά οι περιπτώσεις στις οποίες ξεμένει από καύσιμα στη μέση της διαδρομής. Μετά επιλέγει ποια είναι η πιο συμφέρουσα ενέργεια που μπορεί να πραγματοποιήσει, σύμφωνα με την μέθοδο ϵ-greedy (σειρά 9), δηλαδή, για ϵ = 0.01, 99 φορές στις 100 θα διαλέξει την ενέργεια με την μεγαλύτερη τιμή Q, ενώ με πιθανότητα 1% θα διαλέξει μία τυχαία ενέργεια. Η ενέργεια που επιλέγεται κάθε φορά αποτελείται από δύο τμήματα, την απόφαση ανεφοδιασμού (false, true) και την επόμενη πόλη. Αν η απόφαση ανεφοδιασμού είναι true (σειρά 11) τότε ο agent υπολογίζει το κόστος (σειρές 12-13), γεμίζει την δεξαμενή (σειρά 14) και ξεκινάει το ταξίδι του για την επόμενη πόλη. Διαφορετικά πηγαίνει στην επιλεγμένη πόλη χωρίς ανεφοδιασμό, δηλαδή με μηδενικό κόστος (σειρά 16). Αν τα καύσιμα δεν είναι αρκετά για μία διαδρομή (σειρά 19), τότε συμπληρώνει τα απαραίτητα καύσιμα για το ταξίδι με πολύ ακριβότερο κόστος (σειρά 21), έτσι φτάνει στην πόλη προορισμός με ακριβώς μηδέν λίτρα καυσίμου στη δεξαμενή (σειρά 23). Επιπλέον το συνολικό κόστος του επεισοδίου ανανεώνεται μετά από κάθε ταξίδι από πόλη σε πόλη (σειρά 25), το ίδιο και για την συνολική απόσταση (σειρά 26). Έπειτα, η μηχανή λαμβάνει την ανταμοιβή ανάλογα την reward function που έχει επιλεχθεί στην αρχή του αλγορίθμου (σειρά 27) και πραγματοποιείται το Q update για την ενέργεια που διάλεξε να εκτελέσει ο agent στην αρχή του βρόχου (σειρά 29). Ο εσωτερικός βρόχος ολοκληρώνεται όταν η μηχανή έχει επισκεφθεί όλες τις πόλεις και έχει επιστρέψει στην αρχική, ενώ ο εξωτερικός όταν αυτή η διαδικασία έχει επαναληφθεί για 10000 επεισόδια. Στο τέλος κάθε επεισοδίου συγκρίνεται το συνολικό κόστος με την καλύτερη επίδοση της μηχανής μέχρι στιγμής και μετά μηδενίζεται (σειρά 35) για να αρχίσει το επόμενο επεισόδιο.

Algorithm 4 Q-learning TSPWR

Set the parameters: a, γ, and ϵ
for each pair (s, a) **do**
  Initialize Q(s, a) = 0
end for
Initialize TSPWR variables and constants
Observe the state s_0 : initial city
repeat
  repeat
    Select the action **a** (*refuel decision* and *destination city*) using the ϵ-greedy method
    Take the action **a**
    if *refuel decision* = 1 **then**
      Calculation litres amount for refueling : l_i
      Calculation of the refueling cost : $c_i l_i$
      Maximum tank level : *fuel level* = L_{max}
    else if *refuel decision* = 0 **then**
      The refueling cost is zero : $c_i l_i = 0$
    end if
    Calculation of the *fuel level* in the tank at the arrived city
    if *fuel level* < 0 **then**
      Calculation of missing fuel
      $g_{i,j}$ = 20 * missing fuel
      $z_{i,j}$ = 1
      Reset the tank level : *fuel level* = 0
    end if
    Update the *total cost* of the route : Eq. 2.1
    Update the *total distance* travelled on the route
    Receive immediate reward r(s, a) : Eqs. 3.2, 3.3, 3.4
    Observe the new state s' (new city)
    Update Q(s, a) with Eq. 1.1
    Set s = s'
  until complete the route
  if *total cost* < *lowest cost found so far* **then**
    lowest cost found so far = *total cost*
  end if
  total cost = 0, *total distance* = 0
until the stopping criterion is satisfied

3.2.2 SARSA στο TSPWR

Με κάποιες μικροαλλαγές διαμορφώνεται και ο αλγόριθμος SARSA (Algorithm 5). Η βασική αλλαγή είναι στην εξίσωση ενημέρωσης των τιμών Q, όπου πλέον ενημερώνονται σύμφωνα με την σχέση 1.2. Μία άλλη διαφορά είναι ότι η πρώτη επιλογή ενέργειας της μηχανής συμβαίνει έξω από τον βρόχο.

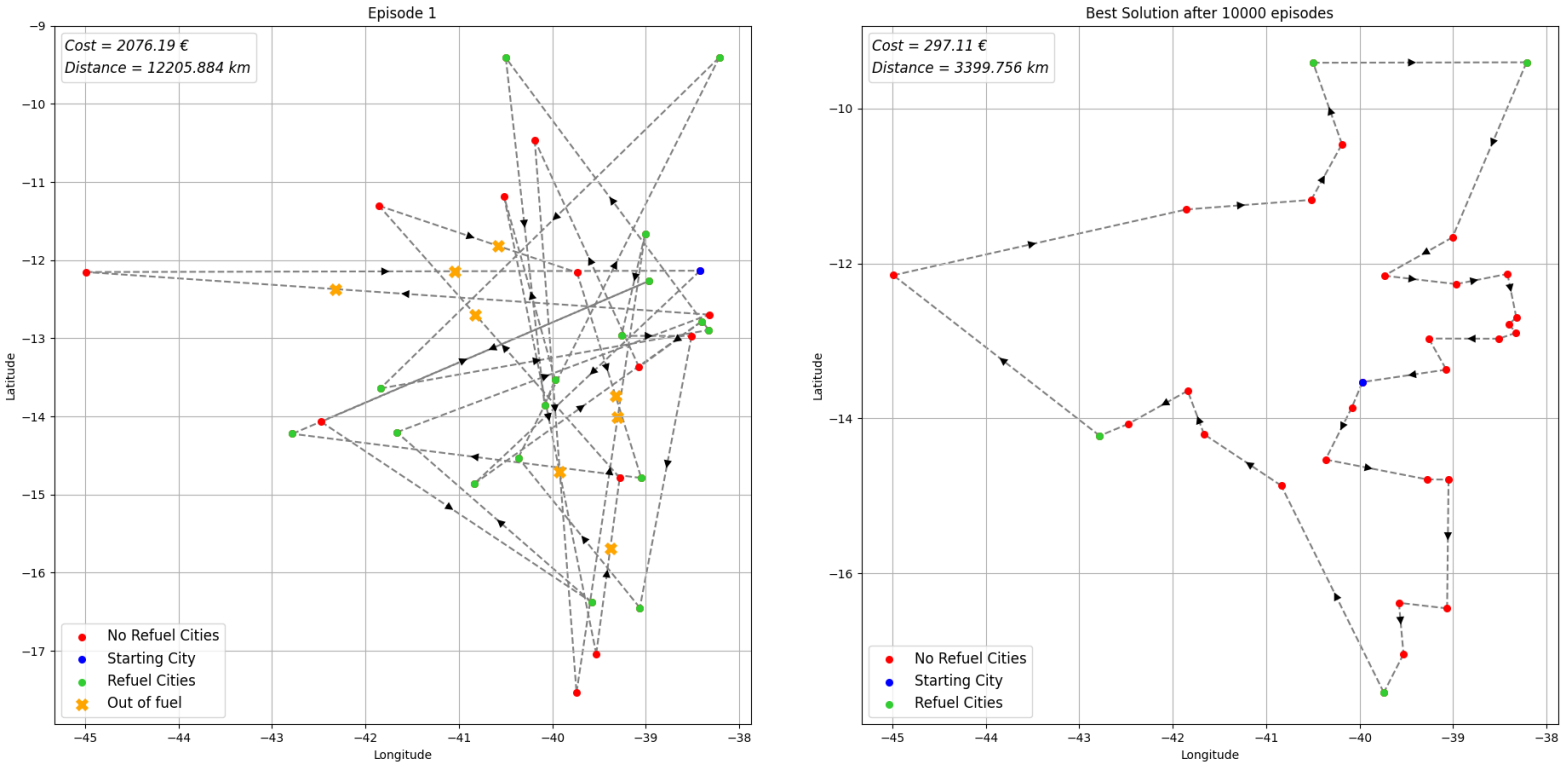
Algorithm 5 SARSA TSPWR

Set the parameters: a, γ, and ϵ
for each pair (s, a) **do**
  Initialize Q(s, a) = 0
end for
Initialize TSPWR variables and constants
Observe the state s_0 : initial city
Select the action **a** (*refuel decision* and *destination city*) using the ϵ-greedy method
repeat
  repeat
    Take the action **a**
    if *refuel decision* = 1 **then**
      Calculation litres amount for refueling : l_i
      Calculation of the refueling cost : $c_i l_i$
      Maximum tank level : *fuel level* = L_{max}
    else if *refuel decision* = 0 **then**
      The refueling cost is zero : $c_i l_i = 0$
    end if
    Calculation of the *fuel level* in the tank at the arrived city
    if *fuel level* < 0 **then**
      Calculation of missing fuel
      $g_{i,j}$ = 20 * missing fuel
      $z_{i,j}$ = 1
      Reset the tank level : *fuel level* = 0
    end if
    Update the *total cost* of the route : Eq. 2.1
    Update the *total distance* travelled on the route
    Receive immediate reward r(s, a) : Eqs. 3.2, 3.3, 3.4
    Observe the new state s' (new city)
    Select the new action a' using the ϵ-greedy method
    Update Q(s, a) with Eq. 1.2
    Set s = s'
    Set a = a'
  until complete the route
  if *total cost* < *lowest cost found so far* **then**
    lowest cost found so far = *total cost*
  end if
  total cost = 0, *total distance* = 0
until the stopping criterion is satisfied

*Αξιοσημείωτο είναι πως στο *πρόβλημα A* σε κάθε επεισόδιο η αρχική πόλη s_0 (σειρά 6 και στους δύο αλγόριθμους) επιλέγεται τυχαία. Στο *πρόβλημα B* σε κάθε επεισόδιο η αρχική πόλη είναι σταθερή.

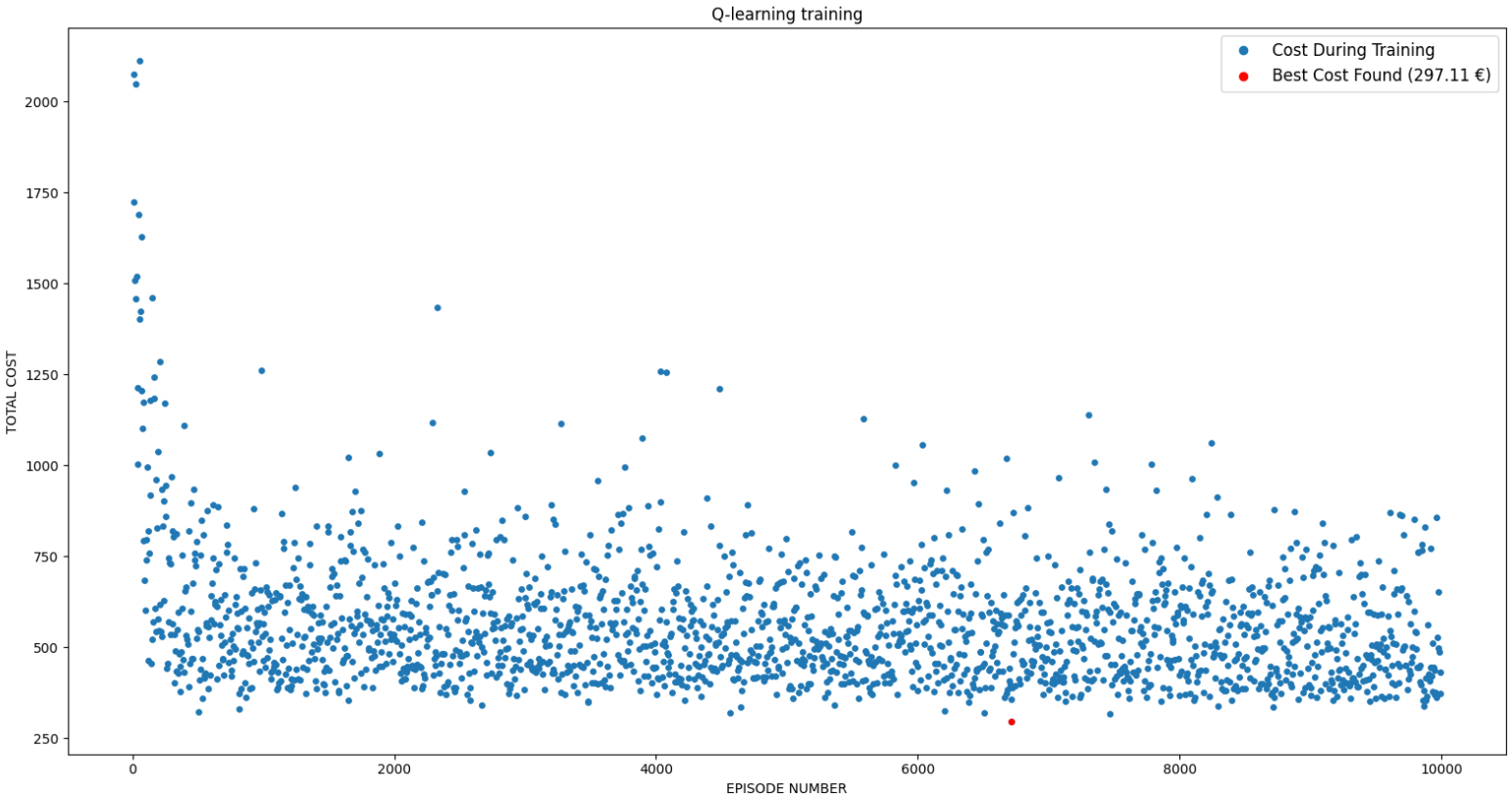
3.2.3 Οπτική αναπαράσταση λειτουργίας αλγορίθμων RL

Η Εικόνα 3.1 αποτελεί ένα οπτικό παράδειγμα για το πως δουλεύουν οι αλγόριθμοι ενισχυτικής μάθησης. Παρατηρούμε πως στο πρώτο επεισόδιο ο agent δε γνωρίζει πως πρέπει να δράσει, ενεργεί τυχαία και αναποτελεσματικά με αποτέλεσμα το κόστος να είναι πολύ υψηλό, όπως και τα συνολικά χλμ που διανύει. Κατά την διάρκεια της εκπαίδευσης όμως μαθαίνει ποιες ενέργειες είναι συμφέρουσες, καταλήγοντας σε πολύ μικρότερο κόστος και πολύ αποδοτικότερη διαδρομή.



Εικόνα 3.1: Οπτική αναπαράσταση της εκπαίδευσης RL για: Πρόβλημα A, Bahia30D, Q-learning.

Στην Εικόνα 3.2 παρουσιάζεται το κόστος κάτα τη διάρκεια του training. Πρόκειται ένα για δείγμα από 2000 επεισόδια, καθώς τα κόστη από όλα τα επεισόδια θα έκαναν το γράφημα δυσανάγνωστο. Η καλύτερη λύση βάση κόστους προστίθεται πάντα στο γράφημα και τονίζεται με κόκκινο χρώμα.



Εικόνα 3.2: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα A, Bahia30D, Q-learning.

3.3 Ευρετικός αλγόριθμος βελτίωσης (Heuristic improvement algorithm)

Μετά το πέρας των 10000 επεισοδίων, δηλαδή μετά το τέλος του αλγόριθμου RL, έχουμε αποθηκευμένη την καλύτερη λύση που βρέθηκε στην μορφή *[total cost, route, refuel states, states that the agent ran out of fuel in between, total distance]*, αυτή η μορφή αποτελεί ένα tuple (μία ομάδα). Όπου το *total cost* είναι η καλύτερη λύση που βρήκε η μηχανή, *route* είναι η διαδρομή που ακολούθησε για να καταλήξει σε αυτή την λύση, *refuel states* είναι όλες οι πόλεις στις οποίες πραγματοποίησε ανεφοδιασμό, *states that the agent ran out of fuel in between* είναι τυχών διαδρομές στις οποίες ο *agent* ξέμεινε από καύσιμα και αναγκάστηκε να κάνει ανεφοδιασμό με πολύ υψηλή τιμή, και τέλος *distance* είναι η τιμή της απόστασης που διένυσε.

Κάθε προτεινόμενη λύση ωστόσο έχει την δυνατότητα να βελτιωθεί περαιτέρω άμα εφαρμόσουμε μερικούς κανόνες λογικής. Σύμφωνα με αυτούς τους κανόνες ίσως αλλάξουν οι πόλεις στις οποίες πραγματοποιείται ανεφοδιασμός, ίσως αλλάξει η αρχική πόλη (μόνο στο *πρόβλημα A*) ή η φορά της διαδρομής, αλλά ο κύκλος της διαδρομής παραμένει ο ίδιος. Για παράδειγμα μία διαδρομή $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$, αλλάζει αρχική πόλη και γίνεται $1 \rightarrow 2 \rightarrow 3 \rightarrow 0 \rightarrow 1$, ή αλλάζει φορά και γίνεται $0 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$. Αυτοί οι κανόνες είναι οι εξής:

- Στο *πρόβλημα Α* εξετάζουμε κάθε πόλη ως πιθανή αρχική πόλη. Εφαρμόζουμε τους παρακάτω κανόνες ξεκινώντας κάθε φορά με διαφορετική αρχική πόλη, διατηρώντας όμως την ίδια διαδρομή (τον ίδιο κύκλο).
- Ξεκινώντας από την αρχική πόλη, ακολουθώντας την διαδρομή που προτείνει ο αλγόριθμος ενισχυτικής μάθησης, ελέγχουμε μέχρι ποια πόλη μπορούμε να φτάσουμε με την γεμάτη δεξαμενή καυσίμου και επιλέγουμε να κάνουμε refuel σε αυτή με το μικρότερο κόστος καυσίμου ανά λίτρο. Στην πόλη που θα κάνουμε ανεφοδιασμό ελέγχουμε πάλι μέχρι που μπορούμε να φτάσουμε και πάλι επιλέγουμε το φθηνότερο refuel. Αυτή τη διαδικασία την ακολουθούμε μέχρι τα καύσιμα μετά από ανεφοδιασμό να επαρκούν για τον τερματισμό της διαδρομής.
- Αν τα καύσιμα αρκούν για να τελειώσει την διαδρομή (να επιστρέψουμε δηλαδή στην αρχική πόλη), τότε δεν πρέπει να κάνουμε κανέναν άλλον ανεφοδιασμό.
- Ο τελευταίος ανεφοδιασμός πρέπει να συμβαίνει σε πόλη όπου με γεμάτη δεξαμενή καυσίμου ο agent έχει την δυνατότητα να φτάσει μέχρι το τέλος. Όμως, σε αντίθεση με τον δεύτερο κανόνα, δεν είναι πάντα η πιο συμφέρουσα επιλογή η οικονομικότερη τιμή καυσίμου ανά λίτρο, άλλα στο τελευταίο refuel μας νοιάζει το κόστος ανεφοδιασμού, δηλαδή η *ποσότητα καυσίμου* × *τιμή ανά λίτρο καυσίμου*.
- Είτε κάνουμε τους παραπάνω ελέγχους για κάθε πόλη ως αρχική (*πρόβλημα Α*), είτε για την σταθερή αρχική πόλη (*πρόβλημα Β*), ελέγχουμε και τις δύο δυνατές κατευθύνσεις της δοσμένης διαδρομής.

Ο αλγόριθμος περιγράφεται για το *πρόβλημα Α* (Algorithm 6), όπου η αλλαγή αρχικής πόλης είναι εφικτή. Η εκδοχή του αλγορίθμου για το *πρόβλημα Β* διαφέρει στο ότι δεν γίνεται έλεγχος για κάθε πόλη της διαδρομής (σειρά 3), άλλα μόνο για την μία και σταθερή αρχική πόλη. Αναλυτικά, θέτουμε το κόστος της λύσης που έδωσε ο agent ως την καλύτερη λύση (σειρά 2), ώστε να έχουμε μέτρο σύγκρισης. Ξεκινάμε με την σειρά και ελέγχουμε κάθε πόλη ως πιθανή αρχική (σειρά 3). Φτιάχνουμε την νέα διαδρομή, μη αλλάζοντας τον κύκλο της διαδρομής, αλλάζοντας απλά την αρχική και τελική πόλη (σειρά 5). Εφόσον δεν έχουμε αρκετά καύσιμα για να ολοκληρώσουμε την διαδρομή χωρίς ανεφοδιασμό, ελέγχουμε ποιες πόλεις είναι σε εύρος μίας γεμάτης δεξαμενής καυσίμου (σειρά 8) και κάνουμε refuel στη πόλη με το φθηνότερο κόστος ανά λίτρο (σειρά 9). Βάζουμε την πόλη στον πίνακα με τις πόλεις ανεφοδιασμού (σειρά 10) που θα παρουσιάσουμε στη τελική λύση αν αυτή είναι συμφέρουσα. Επαναλαμβάνουμε μέχρι να είναι εφικτή η επιστροφή στην αρχική πόλη δίχως ανεφοδιασμό. Στη συνέχεια ψάχνουμε τον ιδανικό τελευταίο ανεφοδιασμό, ο οποίος μπορεί να είναι σε οποιαδήποτε πόλη μεταξύ των δύο τελευταίων ανεφοδιασμών του πίνακα new refuel states, συμπεραλαμβανομένης της πόλης του τελευταίου refuel (σειρές 13-23). Επαναλαμβάνουμε την διαδικασία για την αντίστροφη αρχική διαδρομή (σειρά 25).

Περιληπτικά, ο ευρετικός αλγόριθμος βελτίωσης αποτελείται από 3 στάδια βελτίωσης. Πρώτα ελέγχει όλες τις πόλεις ως πιθανές αρχικές, υπολογίζοντας τα πιθανά κόστη με την λογική ¨κάνε ανεφοδιασμό στη πόλη με τη χαμηλότερη τιμή στην οποία μπορείς να φτάσεις χωρίς να ξεμείνεις από καύσιμα¨, ακολουθώντας την διαδρομή που προτείνει η μηχανή μάθησης. Στο δεύτερο στάδιο ελέγχει πάλι όλες τις πόλεις ως πιθανές αρχικές, άλλα αυτή τη φορά με την αντίστροφη φορά κίνησης στον κύκλο του route. Στο τρίτο και κυριότερο στάδιο, αλλάζει την πόλη του τελευταίου ανεφοδιασμού, κοιτώντας να συνδυάσει φθηνό καύσιμο με λίγη άλλα αρκετή ποσότητα καυσίμου ώστε να φτάσει

στο τέλος της διαδρομής. Μέσα από αυτά τα 3 στάδια επιλέγει σε σύντομο χρονικό διάστημα ίσως την βέλτιστη λύση σύμφωνα με τον κύκλο διαδρομής που προτείνει ο αλγόριθμος ενισχυτικής μάθησης.

Algorithm 6 Heuristic Improvement Algorithm

Require: route, total cost
Ensure: new route, new refuel states, new total cost
1: **function** `heuristic` (route, total cost)
2: best cost = total cost
3: **for** every city in route **do**
4: starting city = city
5: Create the new route with the current starting city as s_0
6: Observe current state (starting city)
7: **while** Not enough fuel to return to the starting city **do**
8: Find all states in fuel range
9: Refuel in the state s' with the lowest cost per litre
10: Add s' in the new refuel states array
11: current state = s'
12: **end while**
13: **for** every city s between the last 2 refuel states, including the last one **do**
14: **if** max fuel tank enough to return to s_0 **then**
15: Change temporarily the last refuel state to s
16: Evaluate the new total cost
17: **if** new total cost < best cost **then**
18: best cost = new total cost
19: Remove the last element in the new refuel states array and add city s
20: Save the new route and the new refuel states
21: **end if**
22: **end if**
23: **end for**
24: **end for**
25: repeat the same process for the reversed route (lines 3-23)
26: **return** new route, new refuel states, best cost
27: **end function**

3.3.1 Πολυπλοκότητα ευρετικού αλγορίθμου

Η πολυπλοκότητα του ευρετικού αλγορίθμου είναι σχεδόν μηδαμινή σε σύγκριση με αυτή των αλγορίθμων ενισχυτικής μάθησης, καθώς ο ευρετικός αλγόριθμος πραγματοποιεί επιπλέον περίπου 180 με 300 iterations σε ένα σύνολο 30 πόλεων. Ο ακριβής αριθμός των επιπλέον iterations δε μπορεί να υπολογιστεί, διότι κοιτώντας τον αλγόριθμο 6 παρατηρούμε ότι αυτός εξαρτάται από τον αριθμό επαναλήψεων του for loop στις σειρές 13 με 23. Ο εξωτερικός βρόχος loop (σειρές 13-24) εκτελείται όσες φορές όσο το πλήθος των πόλεων στο εξεταζόμενο σύνολο, δηλαδή για 30 πόλεις θα εκτελεστεί 30 φορές, αυτόν τον αριθμό τον πολλαπλασιάζουμε με το πόσες πόλεις ενδέχεται να βρίσκονται ανάμεσα στους δύο τελευταίους σταθμούς ανεφοδιασμού (περίπου 3 με 5

πόλεις). Άρα συνολικά πρόκειται για 90 με 150 επιπλέον δοκιμές, ωστόσο ο ευρετικός αλγόριθμος εξετάζει και την αντίστροφη φορά (σειρά 25), επομένως τελικά πρόκειται 180 με 300 επιπλέον δοκιμές. Αν σκεφτούμαι ότι ο αλγόριθμος ενισχυτικής μάθησης εκτελεί 300000 iterations σε ένα σύνολο 30 πόλεων μιλάμε για μία αύξηση χαμηλότερη του 0.1%. Με αυτές τις λίγες παραπάνω δοκιμές στόχος είναι να βελτιωθεί η τελική λύση.

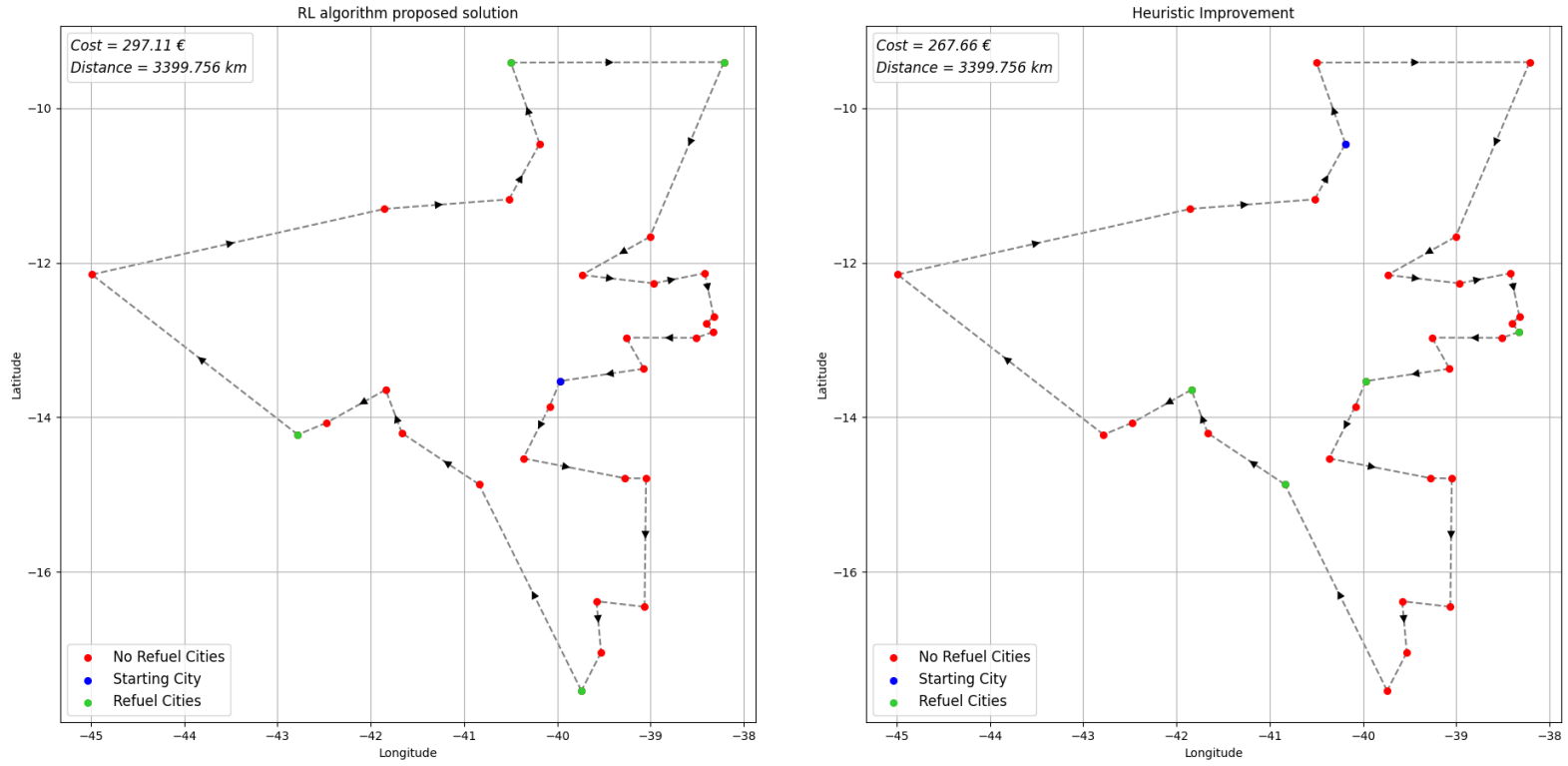
Ο αλγόριθμος που συνδυάζει την ενισχυτική μάθηση και την ευρετική μέθοδο παρουσιάζεται παρακάτω (Algorithm 7). Ουσιαστικά πρόκειται για μία απλή προσθήκη του ευρετικού αλγόριθμου βελτίωσης λύσης στο τέλος του αλγόριθμου ενισχυτικής μάθησης.

Algorithm 7 RL and Heuristic Combination

```
episode number = 1
repeat
    execute the RL algorithm
    if total cost < lowest cost found so far then
        lowest cost found so far = total cost
    end if
    episode number =+ 1
until episode number = 10000
use the heuristic improvement algorithm
heuristic (route, total cost)
```

3.3.2 Οπτική αναπαράσταση λειτουργίας ευρετικού αλγόριθμου βελτίωσης

Στην Εικόνα 3.3 φαίνεται η επίδραση του heuristic improvement algorithm στη λύση που προτείνει ο αλγόριθμος RL. Παρατηρούμε ότι αλλάζει η πόλη εκκίνησης του ταξιδιού, καθώς το συγκεκριμένο παράδειγμα αφορά το *πρόβλημα A*, ενώ αλλάζουν και οι πόλεις στις οποίες γίνεται ανεφοδιασμός καυσίμων, με αποτέλεσμα το κόστος να μειώνεται αισθητά. Η συνολική απόσταση προφανώς μένει ίδια, καθώς ο ευρετικός αλγόριθμος δεν αλλάζει τον κύκλο της διαδρομής.



Εικόνα 3.3: Οπτική αναπαράσταση ευρετικού αλγορίθμου για: Πρόβλημα A, Bahia30D, Q-learning.

3.4 Δεύτερο στάδιο πειραματισμού

Μόλις πραγματοποιηθούν τα 9216 πειράματα από 50 φορές (runs) το καθένα θα παρατηρήσουμε τα αποτελέσματα για κάθε μία από τις 16 εξεταζόμενες περιπτώσεις (πρόβλημα A ή B, αλγόριθμος Q-learning ή SARSA, Bahia30D,Minas24D,Minas30D ή Minas57D) και θα εμβαθύνουμε στους συνδυασμούς παραμέτρων που αποδίδουν καλύτερα σε κάθε περίπτωση, με σκοπό να βρούμε την καλύτερη δυνατή λύση. Συγκεκριμένα, τους 5 καλύτερους συνδυασμούς παραμέτρων a, γ, ϵ και reward function, βάση του μέσου όρο των τελικών λύσεων που προσφέρουν θα τους χρησιμοποιήσουμε κάνοντας επιπλέον 500 δοκιμές σε κάθε εξεταζόμενη περίπτωση. Με το τέλος του δεύτερου σταδίου πειραμάτων θα παρουσιάσουμε σε μορφή χάρτη την καλύτερη λύση που βρέθηκε, ενώ παράλληλα θα κάνουμε σύγκριση αποτελεσμάτων με την εργασία [23].

Κεφάλαιο 4

Αποτελέσματα Πειραμάτων

Η παρουσίαση των αποτελεσμάτων των πειραμάτων θα γίνει σε δύο ενότητες, πρώτα για το *πρόβλημα Α* και μετά για το *πρόβλημα Β*. Κάθε ενότητα θα αποτελείται από δύο υποενότητες, μία για κάθε έναν από τους δύο αλγόριθμους RL που χρησιμοποιήθηκαν, Q-learning και SARSA. Τέλος σε κάθε υποενότητα θα παρουσιάζονται για κάθε instance οι παράμετροι που οδηγούν σε καλύτερο μέσο όρο (καθώς κάθε πείραμα εκτελέστηκε 50 φορές) τελικής λύσης. Ως τελική λύση θεωρούμε την λύση μετά την ευρετική βελτίωση μέσω του heuristic improvement algorithm. Θα παρουσιάσουμε επίσης την καλύτερη λύση που βρήκαμε για κάθε instance.

4.1 Αποτελέσματα στο πρόβλημα Α

Σε αυτή την ενότητα παρουσιάζουμε τα αποτελέσματα για το *πρόβλημα Α*, όπου η αρχική πόλη - και άρα και η τελική - δεν είναι σταθερή, δηλαδή μπορεί να επιλεγεί οποιαδήποτε ως σταθμός εκκίνησης και όπου η δεξαμενή καυσίμου είναι άδεια όταν ξεκινάει το ταξίδι. Για κάθε instance παρουσιάζουμε μία ομάδα τεσσάρων πινάκων. Στον κύριο πίνακα καταγράφουμε τους 10 καλύτερους συνδυασμούς παραμέτρων με βάση τον μέσο όρο αποτελεσμάτων μετά την ευρετική βελτίωση (Post-Heuristic Mean), ενώ στον ίδιο πίνακα φαίνεται το κόστος της καλύτερης λύσης που βρέθηκε με αυτές τις παραμέτρους (Best Solution) και ο μέσος όρος λύσεων πριν την εφαρμογή του heuristic improvement (Pre-Heuristic Mean). Στους δύο επόμενους πίνακες παρουσιάζουμε τον μέσο όρο αποτελεσμάτων για κάθε συνάρτηση ανταμοιβής και για κάθε τιμή του ϵ αντίστοιχα, ενώ ο τελικός πίνακας δείχνει τα αποτελέσματα για κάθε συνδυασμό ϵ και reward function.

4.1.1 Q-learning, Bahia30D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.60	0.01	0.01	3	267.66 €	277.86 €	297.80 €
0.99	0.01	0.01	3	267.21 €	278.12 €	299.54 €
0.60	0.01	0.01	1	266.06 €	278.33 €	298.45 €
0.75	0.01	0.01	1	270.23 €	279.16 €	300.29 €
0.90	0.01	0.01	3	266.76 €	279.78 €	299.72 €
0.75	0.01	0.01	3	268.47 €	279.89 €	298.32 €
0.90	0.01	0.01	1	269.19 €	280.86 €	301.16 €
0.45	0.01	0.01	3	267.66 €	281.05 €	300.58 €
0.75	0.15	0.01	3	266.06 €	281.10 €	305.57 €
0.45	0.01	0.01	1	269.15 €	281.62 €	301.94 €

Πίνακας 4.1: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Bahia30D, Q-learning.

Reward Function	**Post-Heuristic Mean**
1	366.14 €
2	397.12 €
3	365.22 €

Πίνακας 4.2: Αποτελέσματα με βάση την reward function για: Bahia30D, Q-learning.

ϵ	**Post-Heuristic Mean**
0.01	367.54 €
0.05	376.12 €
0.1	384.82 €

Πίνακας 4.3: Αποτελέσματα με βάση το ϵ για: Bahia30D, Q-learning.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	357.46 €
0.01	2	387.89 €
0.01	3	357.28 €
0.05	1	366.20 €
0.05	2	396.91 €
0.05	3	365.26 €
0.1	1	374.77 €
0.1	2	406.56 €
0.1	3	373.12 €

Πίνακας 4.4: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Bahia30D, Q-learning.

4.1.2 Q-learning, Minas24D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.30	0.01	0.01	3	225.71 €	230.51 €	240.11 €
0.45	0.01	0.01	1	225.71 €	230.89 €	242.37 €
0.30	0.01	0.05	3	225.71 €	231.48 €	243.75 €
0.30	0.01	0.01	1	225.71 €	231.59 €	242.82 €
0.45	0.01	0.05	3	225.71 €	232.30 €	245.70 €
0.30	0.01	0.05	1	225.71 €	232.31 €	244.41 €
0.45	0.01	0.01	3	225.71 €	232.31 €	243.49 €
0.60	0.01	0.01	3	225.71 €	232.31 €	245.01 €
0.15	0.01	0.01	3	225.71 €	232.64 €	242.56 €
0.60	0.01	0.01	1	225.71 €	232.71 €	246.27 €

Πίνακας 4.5: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas24D, Q-learning.

Reward Function	**Post-Heuristic Mean**
1	287.94 €
2	318.87 €
3	286.68 €

Πίνακας 4.6: Αποτελέσματα με βάση την reward function για: Minas24D, Q-learning.

ϵ	**Post-Heuristic Mean**
0.01	292.96 €
0.05	296.95 €
0.1	303.58 €

Πίνακας 4.7: Αποτελέσματα με βάση το ϵ για: Minas24D, Q-learning.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	283.56 €
0.01	2	312.84 €
0.01	3	282.47 €
0.05	1	287.26 €
0.05	2	317.74 €
0.05	3	285.85 €
0.1	1	293.01 €
0.1	2	326.02 €
0.1	3	291.71 €

Πίνακας 4.8: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas24D, Q-learning.

4.1.3 Q-learning, Minas30D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.99	0.01	0.01	1	237.38 €	244.68 €	258.92 €
0.99	0.01	0.01	3	237.38 €	245.14 €	257.94 €
0.60	0.01	0.01	3	237.20 €	245.87 €	257.66 €
0.75	0.01	0.01	3	237.04 €	246.06 €	259.58 €
0.75	0.01	0.01	1	237.21 €	246.19 €	260.55 €
0.45	0.01	0.01	1	237.21 €	246.22 €	258.35 €
0.90	0.01	0.01	3	237.21 €	246.41 €	260.43 €
0.90	0.01	0.01	1	237.21 €	246.95 €	260.69 €
0.45	0.01	0.01	3	237.21 €	247.23 €	258.52 €
0.60	0.01	0.01	1	239.19 €	247.33 €	261.07 €

Πίνακας 4.9: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas30D, Q-learning.

Reward Function	**Post-Heuristic Mean**
1	327.65 €
2	365.39 €
3	325.96 €

Πίνακας 4.10: Αποτελέσματα με βάση την reward function για: Minas30D, Q-learning.

ϵ	**Post-Heuristic Mean**
0.01	332.60 €
0.05	338.61 €
0.1	347.79 €

Πίνακας 4.11: Αποτελέσματα με βάση το ϵ για: Minas30D, Q-learning.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	320.90 €
0.01	2	356.89 €
0.01	3	320.01 €
0.05	1	326.98 €
0.05	2	363.82 €
0.05	3	325.04 €
0.1	1	335.06 €
0.1	2	375.45 €
0.1	3	332.84 €

Πίνακας 4.12: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas30D, Q-learning.

4.1.4 Q-learning, Minas57D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.60	0.01	0.01	3	351.10 €	390.10 €	432.62 €
0.60	0.01	0.01	1	365.34 €	390.78 €	437.47 €
0.45	0.01	0.01	3	360.46 €	391.48 €	433.99 €
0.90	0.01	0.01	3	358.46 €	391.53 €	439.06 €
0.75	0.01	0.01	3	356.56 €	392.24 €	432.77 €
0.99	0.01	0.01	3	365.15 €	393.61 €	431.78 €
0.90	0.01	0.01	1	355.82 €	394.14 €	438.53 €
0.15	0.01	0.01	3	363.39 €	394.32 €	429.71 €
0.45	0.01	0.01	1	358.26 €	394.53 €	437.68 €
0.30	0.01	0.01	3	361.27 €	394.72 €	431.99 €

Πίνακας 4.13: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas57D, Q-learning.

Reward Function	**Post-Heuristic Mean**
1	560.28 €
2	622.69 €
3	556.30 €

Πίνακας 4.14: Αποτελέσματα με βάση την reward function για: Minas57D, Q-learning.

ϵ	**Post-Heuristic Mean**
0.01	556.30 €
0.05	578.23 €
0.1	604.38 €

Πίνακας 4.15: Αποτελέσματα με βάση το ϵ για: Minas57D, Q-learning.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	537.87 €
0.01	2	596.49 €
0.01	3	534.53 €
0.05	1	559.13 €
0.05	2	619.20 €
0.05	3	556.35 €
0.1	1	583.47 €
0.1	2	651.99 €
0.1	3	577.67 €

Πίνακας 4.16: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas57D, Q-learning.

4.1.5 SARSA, Bahia30D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.75	0.01	0.01	3	267.93 €	278.58 €	297.75 €
0.60	0.01	0.01	1	267.93 €	279.00 €	299.55 €
0.60	0.01	0.01	3	267.66 €	279.56 €	299.88 €
0.45	0.01	0.01	3	269.62 €	279.57 €	298.67 €
0.90	0.01	0.01	1	265.61 €	279.74 €	301.18 €
0.90	0.01	0.01	3	268.37 €	280.12 €	299.12 €
0.75	0.01	0.01	1	269.27 €	281.45 €	299.60 €
0.99	0.01	0.01	3	267.21 €	281.61 €	299.97 €
0.45	0.01	0.01	1	267.21 €	281.65 €	300.33 €
0.99	0.01	0.01	1	267.21 €	281.88 €	302.35 €

Πίνακας 4.17: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Bahia30D, SARSA.

Reward Function	**Post-Heuristic Mean**
1	376.58 €
2	412.35 €
3	374.94 €

Πίνακας 4.18: Αποτελέσματα με βάση την reward function για: Bahia30D, SARSA.

ϵ	**Post-Heuristic Mean**
0.01	373.48 €
0.05	388.21 €
0.1	402.18 €

Πίνακας 4.19: Αποτελέσματα με βάση το ϵ για: Bahia30D, SARSA.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	363.36 €
0.01	2	394.97 €
0.01	3	362.10 €
0.05	1	376.74 €
0.05	2	412.36 €
0.05	3	375.52 €
0.1	1	389.62 €
0.1	2	429.70 €
0.1	3	387.20 €

Πίνακας 4.20: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Bahia30D, SARSA.

4.1.6 SARSA, Minas24D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.30	0.01	0.01	1	225.71 €	230.00 €	241.27 €
0.45	0.01	0.01	3	225.71 €	231.17 €	242.93 €
0.30	0.01	0.01	3	225.71 €	231.55 €	241.36 €
0.45	0.01	0.01	1	225.71 €	231.68 €	244.63 €
0.30	0.01	0.05	3	225.71 €	231.69 €	242.96 €
0.60	0.01	0.01	3	225.71 €	231.87 €	244.41 €
0.30	0.01	0.05	1	225.71 €	232.18 €	244.71 €
0.75	0.01	0.01	3	225.71 €	232.29 €	247.54 €
0.15	0.01	0.05	1	225.71 €	232.62 €	242.60 €
0.60	0.01	0.01	1	225.71 €	232.84 €	246.04 €

Πίνακας 4.21: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas24D, SARSA.

Reward Function	**Post-Heuristic Mean**
1	292.80 €
2	326.88 €
3	291.27 €

Πίνακας 4.22: Αποτελέσματα με βάση την reward function για: Minas24D, SARSA.

ϵ	**Post-Heuristic Mean**
0.01	296.08 €
0.05	303.31 €
0.1	311.56 €

Πίνακας 4.23: Αποτελέσματα με βάση το ϵ για: Minas24D, SARSA.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	286.31 €
0.01	2	316.73 €
0.01	3	285.20 €
0.05	1	292.29 €
0.05	2	326.63 €
0.05	3	291.02 €
0.1	1	299.81 €
0.1	2	337.29 €
0.1	3	297.58 €

Πίνακας 4.24: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas24D, SARSA.

4.1.7 SARSA, Minas30D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.90	0.01	0.01	1	237.21 €	244.37 €	257.42 €
0.99	0.01	0.01	3	237.38 €	244.43 €	258.55 €
0.75	0.01	0.01	3	237.38 €	245.26 €	259.19 €
0.45	0.01	0.01	3	237.21 €	245.68 €	257.54 €
0.60	0.01	0.01	3	237.21 €	245.75 €	258.58 €
0.45	0.01	0.01	1	237.21 €	246.36 €	259.45 €
0.99	0.01	0.01	1	237.20 €	246.85 €	260.01 €
0.30	0.01	0.01	1	237.04 €	247.43 €	259.97 €
0.90	0.01	0.01	3	237.38 €	247.44 €	260.94 €
0.75	0.01	0.01	1	237.38 €	247.89 €	261.82 €

Πίνακας 4.25: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas30D, SARSA.

Reward Function	**Post-Heuristic Mean**
1	336.01 €
2	378.54 €
3	333.82 €

Πίνακας 4.26: Αποτελέσματα με βάση την reward function για: Minas30D, SARSA.

ϵ	**Post-Heuristic Mean**
0.01	337.01 €
0.05	349.22 €
0.1	362.14 €

Πίνακας 4.27: Αποτελέσματα με βάση το ϵ για: Minas30D, SARSA.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	325.02 €
0.01	2	362.30 €
0.01	3	323.70 €
0.05	1	335.70 €
0.05	2	378.61 €
0.05	3	333.37 €
0.1	1	347.32 €
0.1	2	394.72 €
0.1	3	344.40 €

Πίνακας 4.28: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas30D, SARSA.

4.1.8 SARSA, Minas57D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.90	0.01	0.01	3	354.82	389.56	432.54 €
0.45	0.01	0.01	1	341.99	391.18	436.01 €
0.75	0.01	0.01	3	351.38	391.46	436.53 €
0.60	0.01	0.01	3	350.82	391.66	431.83 €
0.60	0.15	0.01	3	359.80	391.75	437.77 €
0.90	0.01	0.01	1	355.52	392.18	436.64 €
0.30	0.01	0.01	1	366.82	392.89	431.81 €
0.15	0.01	0.01	3	368.84	395.10	430.65 €
0.75	0.01	0.01	1	369.21	395.10	434.16 €
0.15	0.01	0.05	3	367.57	395.24	438.14 €

Πίνακας 4.29: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas57D, SARSA.

Reward Function	**Post-Heuristic Mean**
1	582.57 €
2	662.90 €
3	577.87 €

Πίνακας 4.30: Αποτελέσματα με βάση την reward function για: Minas57D, SARSA.

ϵ	**Post-Heuristic Mean**
0.01	569.37 €
0.05	607.63 €
0.1	646.34 €

Πίνακας 4.31: Αποτελέσματα με βάση το ϵ για: Minas57D, SARSA.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	548.75 €
0.01	2	613.86 €
0.01	3	545.51 €
0.05	1	581.99 €
0.05	2	662.95 €
0.05	3	577.96 €
0.1	1	616.98 €
0.1	2	711.90 €
0.1	3	610.14 €

Πίνακας 4.32: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas57D, SARSA.

4.1.9 Παρατηρήσεις αποτελεσμάτων στο πρόβλημα Α

Οι παρατηρήσεις ισχύουν και για τους δύο αλγόριθμους RL, για όλα τα instances.

- Η reward function 3 που προτείνουμε δίνει τον καλύτερο μέσο όρο αποτελεσμάτων. Προσφέρει οριακά καλύτερες λύσεις από την 1 που προτάθηκε στο [23].
- Το ϵ που δίνει τα καλύτερες αποτελέσματα είναι το 0.01.
- Ο συνδυασμός ϵ = 0.01 και reward function 3 είναι ο καλύτερος δυνατός.
- Το γ = 0.01 κυριαρχεί στους πίνακες αποτελεσμάτων.

4.2 Αποτελέσματα στο πρόβλημα B

Σε αυτή την ενότητα παρουσιάζουμε τα αποτελέσματα για το *πρόβλημα B*, όπου η αρχική πόλη - και άρα και η τελική - είναι σταθερή, και επιλέγεται κάθε φορά η πόλη με id = 0 για κάθε σύνολο πόλεων. Επίσης η δεξαμενή καυσίμου είναι γεμάτη όταν ξεκινάει το ταξίδι, καθώς το κόστος ανεφοδιασμού στην πρώτη πόλη θα ήταν σταθερό, εφόσον ο σταθμός εκκίνησης είναι και αυτός σταθερός. Η παρουσίαση των αποτελεσμάτων γίνεται πάλι σε μορφή πινάκων, για κάθε instance.

4.2.1 Q-learning, Bahia30D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.60	0.15	0.05	1	192.81 €	208.96 €	237.55 €
0.45	0.15	0.05	1	193.09 €	209.57 €	234.97 €
0.60	0.15	0.01	1	192.81 €	209.62 €	238.17 €
0.45	0.15	0.05	3	194.40 €	209.70 €	234.30 €
0.75	0.15	0.05	3	193.09 €	209.82 €	237.53 €
0.60	0.15	0.05	3	192.81 €	210.06 €	235.53 €
0.75	0.15	0.05	1	193.09 €	210.16 €	240.07 €
0.75	0.15	0.01	3	193.53 €	210.24 €	239.10 €
0.30	0.15	0.05	1	192.81 €	210.45 €	237.52 €
0.45	0.15	0.01	1	194.85 €	212.13 €	243.76 €

Πίνακας 4.33: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Bahia30D, Q-learning.

Reward Function	**Post-Heuristic Mean**
1	278.41 €
2	325.88 €
3	278.26 €

Πίνακας 4.34: Αποτελέσματα με βάση την reward function για: Bahia30D, Q-learning.

ϵ	**Post-Heuristic Mean**
0.01	292.00 €
0.05	291.92 €
0.1	298.64 €

Πίνακας 4.35: Αποτελέσματα με βάση το ϵ για: Bahia30D, Q-learning.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	275.58 €
0.01	2	324.94 €
0.01	3	275.47 €
0.05	1	276.51 €
0.05	2	322.74 €
0.05	3	276.51 €
0.1	1	283.14 €
0.1	2	329.96 €
0.1	3	282.80 €

Πίνακας 4.36: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Bahia30D, Q-learning.

4.2.2 Q-learning, Minas24D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.30	0.01	0.05	3	149.90 €	173.52 €	179.49 €
0.30	0.01	0.10	1	149.74 €	173.53 €	180.30 €
0.60	0.01	0.05	3	150.51 €	173.77 €	179.75 €
0.45	0.01	0.05	1	148.79 €	174.71 €	180.83 €
0.45	0.01	0.05	3	149.90 €	174.73 €	180.58 €
0.75	0.01	0.05	3	155.17 €	175.11 €	180.52 €
0.60	0.01	0.05	1	158.49 €	175.59 €	181.40 €
0.45	0.01	0.10	3	149.90 €	175.62 €	181.87 €
0.30	0.01	0.10	3	151.17 €	175.63 €	180.23 €
0.60	0.01	0.10	3	152.76 €	176.22 €	182.69 €

Πίνακας 4.37: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas24D, Q-learning.

Reward Function	**Post-Heuristic Mean**
1	212.92 €
2	256.64 €
3	211.30 €

Πίνακας 4.38: Αποτελέσματα με βάση την reward function για: Minas24D, Q-learning.

ϵ	**Post-Heuristic Mean**
0.01	225.25 €
0.05	225.61 €
0.1	230.00 €

Πίνακας 4.39: Αποτελέσματα με βάση το ϵ για: Minas24D, Q-learning.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	212.24 €
0.01	2	253.06 €
0.01	3	210.44 €
0.05	1	211.40 €
0.05	2	255.24 €
0.05	3	210.19 €
0.1	1	215.11 €
0.1	2	261.61 €
0.1	3	213.28 €

Πίνακας 4.40: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas24D, Q-learning.

4.2.3 Q-learning, Minas30D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.45	0.01	0.05	1	163.77 €	179.44 €	186.46 €
0.30	0.01	0.05	1	162.73 €	179.85 €	186.23 €
0.30	0.01	0.05	3	164.90 €	180.28 €	184.96 €
0.45	0.01	0.05	3	163.45 €	183.75 €	190.89 €
0.60	0.01	0.05	3	157.70 €	183.90 €	192.34 €
0.15	0.01	0.05	1	169.89 €	183.98 €	189.79 €
0.30	0.01	0.10	1	165.59 €	184.30 €	192.09 €
0.15	0.01	0.05	3	170.86 €	184.53 €	189.65 €
0.45	0.01	0.01	3	161.75 €	185.98 €	198.49 €
0.30	0.01	0.01	1	171.63 €	186.42 €	194.68 €

Πίνακας 4.41: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas30D, Q-learning.

Reward Function	**Post-Heuristic Mean**
1	247.73 €
2	297.62 €
3	246.56 €

Πίνακας 4.42: Αποτελέσματα με βάση την reward function για: Minas30D, Q-learning.

ϵ	**Post-Heuristic Mean**
0.01	260.04 €
0.05	262.24 €
0.1	269.63 €

Πίνακας 4.43: Αποτελέσματα με βάση το ϵ για: Minas30D, Q-learning.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	244.58 €
0.01	2	291.43 €
0.01	3	244.10 €
0.05	1	246.06 €
0.05	2	295.46 €
0.05	3	245.19 €
0.1	1	252.54 €
0.1	2	305.95 €
0.1	3	250.39 €

Πίνακας 4.44: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas30D, Q-learning.

4.2.4 Q-learning, Minas57D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.60	0.01	0.01	1	273.45	306.23	338.00 €
0.75	0.01	0.01	3	273.66	308.55	338.62 €
0.30	0.01	0.05	3	271.46	308.63	333.43 €
0.45	0.01	0.01	3	273.88	309.00	339.73 €
0.60	0.01	0.01	3	279.11	309.88	334.47 €
0.75	0.01	0.01	1	279.27	310.33	338.21 €
0.99	0.01	0.01	3	276.63	310.68	343.57 €
0.45	0.01	0.05	3	286.92	312.73	336.04 €
0.60	0.01	0.05	3	275.93	312.90	339.54 €
0.99	0.01	0.01	1	274.16	313.14	349.34 €

Πίνακας 4.45: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas57D, Q-learning.

Reward Function	**Post-Heuristic Mean**
1	461.54 €
2	541.44 €
3	458.42 €

Πίνακας 4.46: Αποτελέσματα με βάση την reward function για: Minas57D, Q-learning.

ϵ	**Post-Heuristic Mean**
0.01	468.68 €
0.05	482.66 €
0.1	510.21 €

Πίνακας 4.47: Αποτελέσματα με βάση το ϵ για: Minas57D, Q-learning.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	443.89 €
0.01	2	520.67 €
0.01	3	441.47 €
0.05	1	457.58 €
0.05	2	535.89 €
0.05	3	454.52 €
0.1	1	483.15 €
0.1	2	567.40 €
0.1	3	479.27 €

Πίνακας 4.48: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas57D, Q-learning.

4.2.5 SARSA, Bahia30D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.60	0.01	0.01	3	194.40 €	209.40 €	232.86 €
0.45	0.01	0.05	3	194.85 €	211.21 €	230.11 €
0.60	0.01	0.01	1	193.25 €	211.84 €	235.83 €
0.60	0.01	0.05	3	194.40 €	212.38 €	231.23 €
0.45	0.01	0.05	1	194.40 €	212.50 €	231.40 €
0.99	0.01	0.05	1	194.40 €	212.92 €	240.04 €
0.75	0.01	0.05	3	192.81 €	213.30 €	232.10 €
0.90	0.01	0.01	3	194.85 €	213.84 €	236.65 €
0.45	0.15	0.05	3	196.06 €	214.05 €	238.67 €
0.60	0.15	0.05	3	194.85 €	214.13 €	242.32 €

Πίνακας 4.49: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Bahia30D, SARSA.

Reward Function	**Post-Heuristic Mean**
1	290.59 €
2	340.65 €
3	289.56 €

Πίνακας 4.50: Αποτελέσματα με βάση την reward function για: Bahia30D, SARSA.

ϵ	**Post-Heuristic Mean**
0.01	296.56 €
0.05	305.78 €
0.1	318.47 €

Πίνακας 4.51: Αποτελέσματα με βάση το ϵ για: Bahia30D, SARSA.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	281.53 €
0.01	2	326.46 €
0.01	3	281.70 €
0.05	1	289.42 €
0.05	2	339.85 €
0.05	3	288.06 €
0.1	1	300.83 €
0.1	2	355.64 €
0.1	3	298.92 €

Πίνακας 4.52: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Bahia30D, SARSA.

4.2.6 SARSA, Minas24D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.45	0.01	0.05	1	149.74 €	172.76 €	178.44 €
0.30	0.01	0.05	3	149.74 €	173.82 €	180.05 €
0.60	0.01	0.05	1	149.74 €	174.24 €	180.12 €
0.45	0.01	0.10	3	148.77 €	174.66 €	180.57 €
0.45	0.01	0.05	3	150.51 €	174.81 €	180.54 €
0.60	0.01	0.05	3	156.30 €	174.88 €	179.73 €
0.30	0.01	0.05	1	149.74 €	175.17 €	180.16 €
0.30	0.01	0.10	3	158.94 €	175.20 €	180.80 €
0.60	0.01	0.10	3	159.80 €	175.70 €	180.56 €
0.15	0.01	0.10	3	149.90 €	176.51 €	183.72 €

Πίνακας 4.53: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas24D, SARSA.

Reward Function	**Post-Heuristic Mean**
1	218.47 €
2	260.68 €
3	216.71 €

Πίνακας 4.54: Αποτελέσματα με βάση την reward function για: Minas24D, SARSA.

ϵ	**Post-Heuristic Mean**
0.01	227.77 €
0.05	230.57 €
0.1	237.52 €

Πίνακας 4.55: Αποτελέσματα με βάση το ϵ για: Minas24D, SARSA.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	215.46 €
0.01	2	254.10 €
0.01	3	213.76 €
0.05	1	217.145 €
0.05	2	258.74 €
0.05	3	215.83 €
0.1	1	222.815 €
0.1	2	269.19 €
0.1	3	220.55 €

Πίνακας 4.56: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas24D, SARSA.

4.2.7 SARSA, Mina30D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.15	0.01	0.05	3	171.01 €	181.43 €	187.07 €
0.30	0.01	0.05	3	163.30 €	181.51 €	187.20 €
0.30	0.01	0.05	1	163.93 €	181.51 €	188.06 €
0.45	0.01	0.05	1	162.13 €	181.52 €	188.49 €
0.45	0.01	0.05	3	164.73 €	182.33 €	188.24 €
0.15	0.01	0.05	1	162.94 €	183.90 €	189.22 €
0.45	0.01	0.01	1	163.93 €	184.89 €	197.38 €
0.30	0.01	0.10	1	169.34 €	185.15 €	195.18 €
0.30	0.01	0.10	3	160.68 €	185.40 €	193.10 €
0.45	0.01	0.01	3	166.04 €	185.52 €	196.35 €

Πίνακας 4.57: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas30D, SARSA.

Reward Function	**Post-Heuristic Mean**
1	257.14 €
2	307.19 €
3	255.37 €

Πίνακας 4.58: Αποτελέσματα με βάση την reward function για: Minas30D, SARSA.

ϵ	**Post-Heuristic Mean**
0.01	264.47 €
0.05	271.74 €
0.1	283.49 €

Πίνακας 4.59: Αποτελέσματα με βάση το ϵ για: Minas30D, SARSA.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	250.35 €
0.01	2	293.87 €
0.01	3	249.18 €
0.05	1	255.80 €
0.05	2	305.39 €
0.05	3	254.02 €
0.1	1	265.27 €
0.1	2	322.30 €
0.1	3	262.90 €

Πίνακας 4.60: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas30D, SARSA.

4.2.8 SARSA, Mina57D

a	γ	ϵ	**Reward Function**	**Best Solution**	**Post-Heuristic Mean**	**Pre-Heuristic Mean**
0.60	0.01	0.01	1	279.53 €	305.21 €	335.02 €
0.75	0.01	0.01	3	277.32 €	306.52 €	338.31 €
0.45	0.01	0.01	3	273.39 €	307.53 €	334.39 €
0.60	0.01	0.01	3	265.99 €	308.07 €	334.29 €
0.90	0.01	0.01	3	271.39 €	309.62 €	339.38 €
0.99	0.01	0.01	3	265.09 €	311.16 €	343.39 €
0.45	0.15	0.01	1	279.88 €	312.23 €	344.97 €
0.90	0.01	0.01	1	269.13 €	312.23 €	347.33 €
0.30	0.01	0.05	3	274.49 €	312.26 €	336.74 €
0.45	0.01	0.05	3	275.71 €	312.52 €	340.66 €

Πίνακας 4.61: Καλύτερα αποτελέσματα βάση το post-heuristic mean για: Minas57D, SARSA.

Reward Function	**Post-Heuristic Mean**
1	494.17 €
2	586.57 €
3	490.36 €

Πίνακας 4.62: Αποτελέσματα με βάση την reward function για: Minas57D, SARSA.

ϵ	**Post-Heuristic Mean**
0.01	485.77 €
0.05	522.66 €
0.1	562.68 €

Πίνακας 4.63: Αποτελέσματα με βάση το ϵ για: Minas57D, SARSA.

ϵ	**Reward Function**	**Post-Heuristic Mean**
0.01	1	462.20 €
0.01	2	535.32 €
0.01	3	459.78 €
0.05	1	492.13 €
0.05	2	587.11 €
0.05	3	488.73 €
0.1	1	528.19 €
0.1	2	637.29 €
0.1	3	522.57 €

Πίνακας 4.64: Αποτελέσματα με βάση το ϵ και την reward function συνδυαστικά για: Minas57D, SARSA.

4.2.9 Παρατηρήσεις αποτελεσμάτων στο πρόβλημα B

- Η συνάρτηση ανταμοιβής 3 που προτείνουμε δίνει τον καλύτερο μέσο όρο αποτελεσμάτων. Προσφέρει οριακά καλύτερα αποτελέσματα από την συνάρτηση ανταμοιβής 1 που προτάθηκε στο [23].
- Το ϵ που δίνει τα καλύτερα αποτελέσματα είναι το 0.01 σε όλες τις περιπτώσεις, εκτός από το instance Bahia30D, όταν ο αλγόριθμος RL είναι ο Q-learning. Σε αυτή τη περίπτωση το ϵ = 0.05 προσφέρει οριακά καλύτερες λύσεις κάτα μέσο όρο.

- Οι συνδυασμοί ϵ και reward function που δίνουν τα καλύτερα αποτελέσματα διαφέρουν από περίπτωση σε περίπτωση, με τον συνδυασμό $\epsilon = 0.01$ και reward function 3 να συναντάται πιο συχνά.
- Το $\gamma = 0.01$ κυριαρχεί στους πίνακες αποτελεσμάτων, με εξαίρεση το instance Bahia30D, όταν ο αλγόριθμος RL είναι ο Q-learning, σε αυτή τη περίπτωση το $\gamma = 0.15$ ήταν και στους 10 καλύτερους συνδυασμούς παραμέτρων.

4.3 Αποτελέσματα δεύτερου σταδίου πειραματισμού

Η καλύτερη λύση για κάθε εξεταζόμενη περίπτωση του non-uniform TSPWR παρουσιάζεται στο παράρτημα Α σε μορφή χάρτη. Πρώτα εξετάζουμε το *πρόβλημα Α*:

- Για το instance Bahia30D η καλύτερη λύση που βρέθηκε κάτα το πρώτο στάδιο πειραμάτων ήταν αυτή των 265.61 €, η οποία είχε βρεθεί με αλγόριθμο ενισχυτικής μάθησης τον SARSA και παραμέτρους $a = 0.9$, $\gamma = 0.01$, $\epsilon = 0.01$ και reward function→1. Παρόλο που αυτή η λύση συναντήθηκε πολλές φορές στο δεύτερο στάδιο πειραμάτων, δε βρέθηκε χαμηλότερη.
- Για το instance Minas24D η καλύτερη λύση που βρέθηκε κάτα το πρωταρχικό στάδιο πειραματρισμού ήταν αυτή των 225.71 €, αυτή η λύση βρέθηκε και με τους δύο αλγόριθμους RL με ποίκιλους συνδυασμούς παραμέτρων, ενδεικτικά $a = 0.6$, $\gamma = 0.01$, $\epsilon = 0.01$ και reward function→3 με Q-learning. Στο δεύτερο στάδιο δεν βρέθηκε χαμηλότερη λύση, παρόλο που αυτή η λύση συναντήθηκε πολλές φορές.
- Για το instance Minas30D συμβαίνει το ίδιο. Η καλύτερη λύση που βρέθηκε στο πρώτο στάδιο πειραμάτων, 237.04 €, δε βελτιώθηκε περαιτέρω. Αυτή η λύση βρέθηκε χρησιμοποιώντας και τους δύο αλγόριθμους RL, με διάφορος συνδυασμούς παραμέτρων, όπως $a = 0.75$, $\gamma = 0.01$, $\epsilon = 0.01$ και reward function→3 με Q-learning.
- Για το instance Minas57D η καλύτερη λύση που συναντήθηκε είναι αυτή των 341.99 €, η οποία επιτεύχθει με SARSA και παραμέτρους $a = 0.45$, $\gamma = 0.01$, $\epsilon = 0.01$ και reward function→1. Στις επιμέρους δοκιμές η λύση βελτιώθηκε, χρησιμοποιώντας τον αλγόριθμο Q-learning και παραμέτρους $a = 0.6$, $\gamma = 0.01$, $\epsilon = 0.01$ και reward function→1, πέφτωντας στα 332.35 €.

Για το *πρόβλημα Β*:

- Για το instance Bahia30D, η καλύτερη λύση των 192.81 € βρέθηκε από πολλούς συνδυασμούς παραμέτρων, όπως $a = 0.6$, $\gamma = 0.15$, $\epsilon = 0.01$ και reward function→1 με Q-learning. Αυτή η λύση επίσης δε βελτιώθηκε στις επόμενες δοκιμές.
- Για το instance Minas24D, η καλύτερη λύση των 147.68 € βρέθηκε χρησιμοποιώντας αλγόριθμο Q-learning με τις παραμέτρους $a = 0.99$, $\gamma = 0.15$, $\epsilon = 0.05$ και reward function→1. Σε αυτή τη περίπτωση συναντήθηκε βελτιωμένη λύση κάτα το δεύτερο στάδιο πειραματισμού (146.57 €), με παράμετρους $a = 0.45$, $\gamma = 0.01$, $\epsilon = 0.05$, reward function→1 και Q-learning.
- Για το instance Minas30D, η καλύτερη λύση των 156.88 € βρέθηκε με αλγόριθμο

Q-learning και παραμέτρους a = 0.9, γ = 0.01, ϵ = 0.01 και reward function→3. Αυτή είναι η μοναδική περίπτωση όπου το δεύτερο στάδιο πειραμάτων παρουσιάσε χειρότερη λύση από ότι το πρώτο.

- Για το instance Minas57D η λύση των 260.80 € που βρέθηκε αρχικά χρησιμοποιώντας Q-learning με a = 0.3, γ = 0.01, ϵ = 0.01 και reward function→3, βελτιώθηκε στις μετέπειτα δοκιμές. Η βελτιωμένη λύση των 257.80 € εντοπίστηκε χρησιμοποιώντας Q-learning με a = 0.6, γ = 0.01, ϵ = 0.01 και reward function→3.

4.4 Σύγκριση αποτελεσμάτων με την βιβλιογραφία

Όπως αναφέραμε το *πρόβλημα Β* εξετάστηκε για λόγους σύγκρισης με την εργασία [23], καθώς εκεί εξετάζεται το non-uniform TSPWR με την πόλη εκκίνησης να είναι σταθερή για κάθε instance και τη δεξαμενή καυσίμου γεμάτη στην αρχή του ταξιδιού. Όπως φαίνεται στον Πίνακα 4.65 οι προτεινόμενες λύσεις που προκύπτουν από την παρούσα εργασία είναι οικονομικότερες για κάθε ένα από τα τέσσερα σύνολα πόλεων. Αξιοσημείωτο είναι επίσης πως για κάθε instance ο μέσος όρος λύσεων μας, για τους 10 καλύτερους συνδυασμούς παραμέτρων του, είναι χαμηλότερος από την καλύτερη προτεινόμενη λύση στην εργασία [23]. Οι προτεινόμενες λύσεις παρουσιάζονται οπτικά στα παραρτήματα Α΄ και Β΄, συγκεκριμένα είναι οι χάρτες μετά την εφαρμογή του ευρετικού αλγόριθμου βελτίωσης (Heuristic Improvement).

Instance	**Simiral work**	**Proposed Solution**
Bahia30D	269.48 €	**192.81** €
Minas24D	225.18 €	**146.57** €
Minas30D	258.31 €	**156.88** €
Minas57D	388.59 €	**257.80** €

4.65: Σύγκριση μεταξύ των προτεινόμενων λύσεων της εργασίας [23] και των προτεινόμενων λύσεων της παρούσας εργασίας.

4.5 Σύγκριση αλγορίθμων ενισχυτικής μάθησης της εργασίας με αυτούς της βιβλιογραφίας

Σε αυτό το σημείο οφείλουμε να επισημάνουμε πως οι αλγόριθμοι ενισχυτικής μάθησης που εφάρμοσαμε αποδίδουν καλύτερα από αυτούς στην εργασία [23] ακόμα και χωρίς την βοήθεια του ευρετικού αλγορίθμου (Algorithm 6), αυτό φαίνεται από τις λύσεις που καταγράφονται στις Εικόνες Β΄.1, Β΄.4, Β΄.7, Β΄.10, όπου η προτεινόμενη (pre-heuristic) λύση είναι χαμηλότερη από τη καλύτερη προτεινόμενη στην εργασία [23]. Αυτό οφείλεται στο γεγονός πως στους αλγόριθμους RL που προτείνουμε η μηχανή μάθησης εξετάζει το ενδεχόμενο του ανεφοδιασμού σε κάθε στάση και κρίνει η ίδια άμα ο ανεφοδιασμός είναι συμφέρων ή όχι, ενώ στη παραπάνω εργασία ο agent ανεφοδιάζει κάθε

φορά που η στάθμη του πέφτει κάτω από το 25% στη δεξαμενή καυσίμου. Στην Εικόνα Β΄.2 ωστόσο φαίνεται πως η λύση του αλγόριθμου ενισχυτικής μάθησης περιλαμβάνει και μία τιμωρία (Out of fuel), πράγμα που θα μπορούσε να κάνει την σύγκριση άδικη, καθώς στην εργασία με την οποία συγκρίνουμε τα αποτελέσματα μας η τιμωρία όταν μείνει ο agent από καύσιμα είναι σταθερή και ίση με 200 *R*$ (32.63 €), ενώ στη δική μας η τιμωρία είναι μεταβλητή και ίση με 20 *R*$ × *ποσότητα καυσίμου που απαιτείται ώστε να φτάσει ο agent μέχρι την επόμενη πόλη*. Στη συγκεκριμένη περίπτωση απαιτούνται 0.91 λίτρα καυσίμου για να φτάσει στη πόλη 5 από την πόλη 27, αυτό σημαίνει πως η τιμωρία που δέχτηκε ήταν ίση με 2.97 €, αφαιρώντας αυτό το πόσο και προσθέτοντας 32.63 € προκύπτει λύση με κόστος 251.38 €, άρα τελικά ακόμα και με την τιμωρία που εφαρμόζεται στην εργασία [23] η λύση είναι καλύτερη και χωρίς την βελτίωση μέσω του ευρετικού αλγόριθμου. Το ίδιο ισχύει και για τις λύσεις των αλγορίθμων RL που παρουσιάζονται στις Εικόνες Β΄.5, Β΄.8, Β΄.11, όπου με την τιμωρία της εργασίας [23] οι λύσεις παραμένουν οικονομικότερες. Αυτό γιατί προσέθοντας το επιπλέον κόστος των 32.63 € (χωρίς καν την αφαίρεση της τιμωρίας που εφαρμόζουμε), το συνολικό κόστος της διαδρομής παραμένει χαμηλότερο, το οποίο αποδυκνύει την καλύτερη απόδοση των αλγορίθμων ενισχυτικής μάθησης που παρουσιάστηκαν στην εργασία.

Μη ξεχνάμε επίσης πως οι λύσεις των αλγόριθμων RL που παρουσιάζονται στα παραρτήματα δεν είναι οι καλύτερες δυνατές, αλλά αυτές που οδηγούν σε καλύτερα αποτελέσματα μετά την εφαρμογή του ευρετικού αλγόριθμου, αυτό σημαίνει πως η διαφορά επίδοσης μεταξύ των αλγορίθμων RL της εργασίας και των αλγορίθμων RL της εργασίας [23] είναι ακόμα μεγαλύτερη από αυτή που παρουσιάζεται.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντικές Προεκτάσεις

Στην παρούσα διπλωματική εργασία, μελετήθηκε το πρόβλημα Traveling Salesman Problem with Refueling (TSPWR), το οποίο αποτελεί παραλλαγή του κλασικού προβλήματος του Πλανόδιου Πωλητή (TSP). Στόχος της εργασίας ήταν η επίλυση του προβλήματος με τη χρήση αλγορίθμων ενισχυτικής μάθησης, όπως οι Q-learning και SARSA, σε συνδυασμό με έναν ευρετικό αλγόριθμο βελτίωσης. Δύο διαφορετικές εκδοχές του TSPWR εξετάστηκαν, ενώ οι λύσεις που προέκυψαν από την εφαρμογή αυτών των αλγορίθμων αξιολογήθηκαν και συγκρίθηκαν. Η προσέγγιση που ακολουθήθηκε είχε ως σκοπό την εξεύρεση βέλτιστων λύσεων με έμφαση στη βελτίωση των αποτελεσμάτων μέσω της χρήσης του ευρετικού αλγόριθμου, και τα πειραματικά αποτελέσματα παρουσιάζονται αναλυτικά, αποδεικνύοντας την αποτελεσματικότητα των μεθόδων που χρησιμοποιήθηκαν.

Διάφορα συμπεράσματα προκύπτουν απο την παρουσα εργασία. Αρχικά είναι φανερό πως ο αλγόριθμος ευρετικής βελτίωσης που παρουσιάσαμε (Algorithm 6) βελτιώνει εντυπωσιακά τις λύσεις που προκύπτουν από τους αλγόριθμους ενισχυτικής μάθησης. Αυτό γίνεται ξεκάθαρο στους Πίνακες του Κεφαλαίου 4[1], όπου οι στήλες Post-Heuristic Mean δείχνουν τον μέσο όρο κόστους των λύσεων μετά την ευρετική βελτίωση συγκριτικά με τον μέσο όρο κόστους των λύσεων πριν την ευρετική βελτίωση, στη στήλη Pre-Heuristic Mean. Όλες οι προτεινόμενες λύσεις για κάθε εξεταζόμενη περίπτωση φυσικά είναι λύσεις που προέκυψαν από τον heuristic improvement αλγόριθμο. Αυτή η βελτίωση γίνεται με ελάχιστα παραπάνω iterations συγκριτικά με τα iterations των αλγορίθμων RL. Επίσης αξίζει να σημειωθεί πως οι αλγόριθμοι ενισχυτικής μάθησης που εφαρμόσαμε αποδίδουν καλύτερα από αυτούς στην εργασία [23] ακόμα και χωρίς την βοήθεια του ευρετικού αλγορίθμου (Algorithm 6), όπως εξηγείται αναλυτικά στο Υποκεφάλαιο 4.5. Επιπλέον, στο δεύτερο στάδιο πειραμάτων δεν εντοπίστηκε βελτιωμένη λύση στις περισσότερες περιπτώσεις, ενώ σε αυτές που εντοπίστηκε η βελτίωση ήταν αρκετά μικρή. Συμπεραίνουμε, λοιπόν, ότι οι 500 επιπλέον δοκιμές για τους 10 καλύτερους συνδυασμούς παραμέτρων για κάθε εξεταζόμενη περίπτωση δεν είχαν κάποιο όφελος και μπορούν να θεωρηθούν περιττές. Επιπροσθέτως, αν και για κάθε εξεταζόμενη περίπτωση οι παράμετροι που οδηγούσαν σε καλύτερες λύσεις διέφεραν,

[1] Συγκεκριμένα στους Πίνακες 4.1, 4.5, 4.9, 4.13, 4.17, 4.21, 4.25, 4.33, 4.37, 4.41, 4.49, 4.53, 4.57, 4.61

αυτό που παρατηρήθηκε είναι πως τα $\gamma = 0.01$ και $\epsilon = 0.01$ ήταν αυτά που ξεχώρισαν με τις επιδόσεις τους. Το reward function 3 απέδιδε οριακά καλύτερα από το reward function 1, ενώ και τα δύο απέδιδαν πολύ καλύτερα από το reward function 2 σε όλες τις περιπτώσεις. Τέλος για την παράμετρο a η μόνη τιμή που δεν εμφανίστηκε στους πίνακες με τους 10 καλύτερους συνδυασμούς σε καμία περίπτωση είναι η τιμή $a = 0.01$, ενώ όλες οι άλλες παρατηρήθηκαν, με συχνότερες τις $a = 0.30$, $a = 0.45$, $a = 0.60$, $a = 0.75$. Το βασικότερο συμπέρασμα ωστόσο είναι πως ο συνδυασμός ευρετικών αλγορίθμων και αλγορίθμων ενισχυτικής μάθησης είναι πολλά υποσχόμενος.

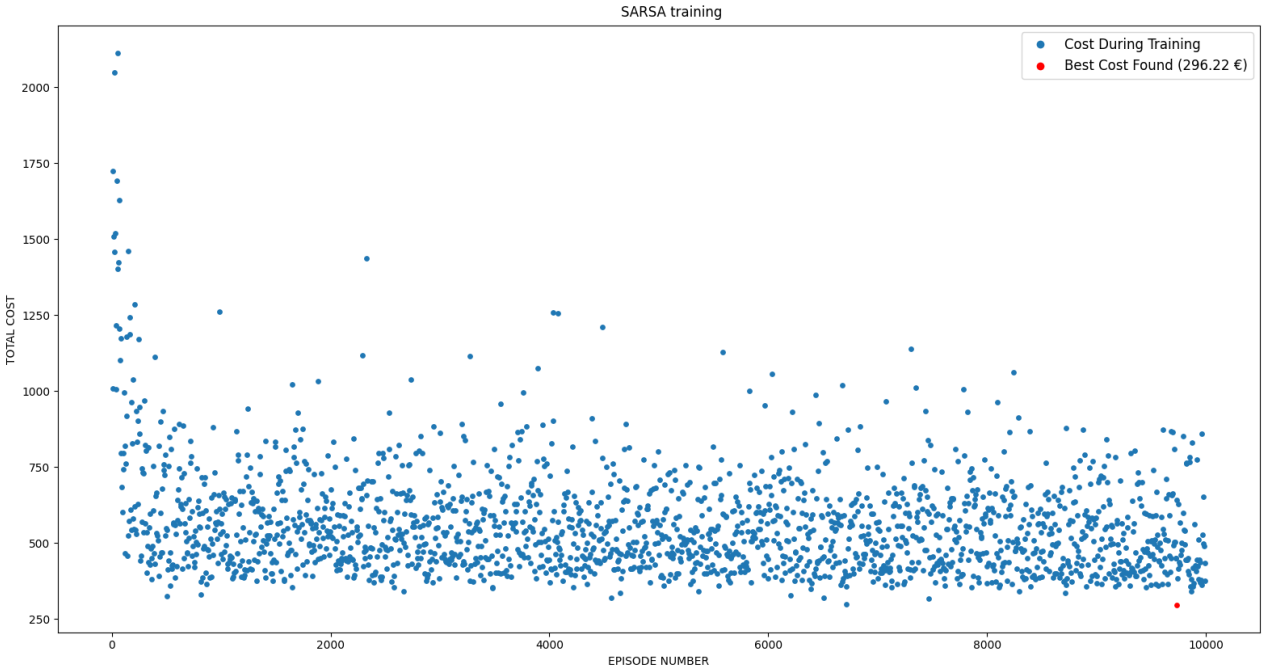
Οσον αφορά μελλοντικές επεκτάσεις, η διαδικασία επίλυσης του προβλήματος TSPWR θα μπορούσε πιθανά να ωφεληθεί περαιτέρω με ποικίλους τρόπους. Στη παρούσα εργασία δεν εφαρμόσαμε στατιστικές μεθόδους (π.χ. RSM, ANOVA, Tukey Test) που θα μπορούσαν να προτείνουν τιμές στις παραμέτρους a και γ που οδηγούν σε χαμηλότερες λύσεις, κάτι τέτοιο μπορεί να δοκιμαστεί σε μελλοντικές έρευνες. Επιπλέον μία ενδιαφέρουσα προσέγγιση θα ήταν η παράμετρος ϵ της μεθόδου ϵ-greedy να μην είναι σταθερή, άλλα να ξεκινάει από μία μεγάλη τιμή και σταδιακά να πέφτει προς το μηδέν, οπότε με αυτόν τον τρόπο ο agent θα εξερευνούσε περισσότερο στην αρχή της εκπαίδευσης και λιγότερο προς το τέλος της. Τέλος, μπορούν μελλοντικά να δοκιμαστούν και άλλοι αλγόριθμου ενισχυτικής μάθησης, όπως ο Double Q-learning, ή ακόμα και να προταθούν νέοι, αποδοτικότεροι ευρετικοί αλγόριθμοι βελτίωσης.

Παράρτημα Α΄

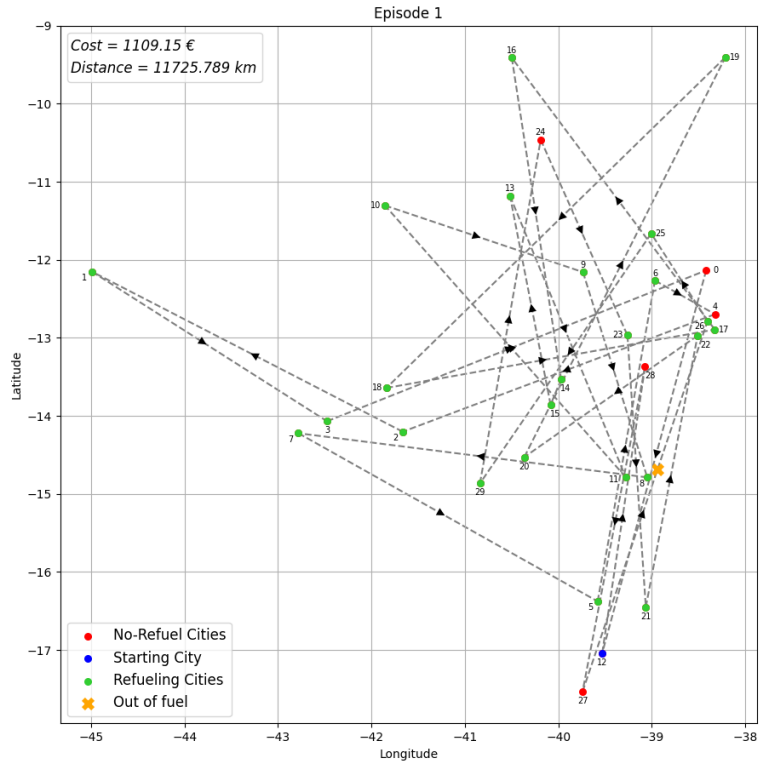
Προτεινόμενες Λύσεις για το Πρόβλημα Α

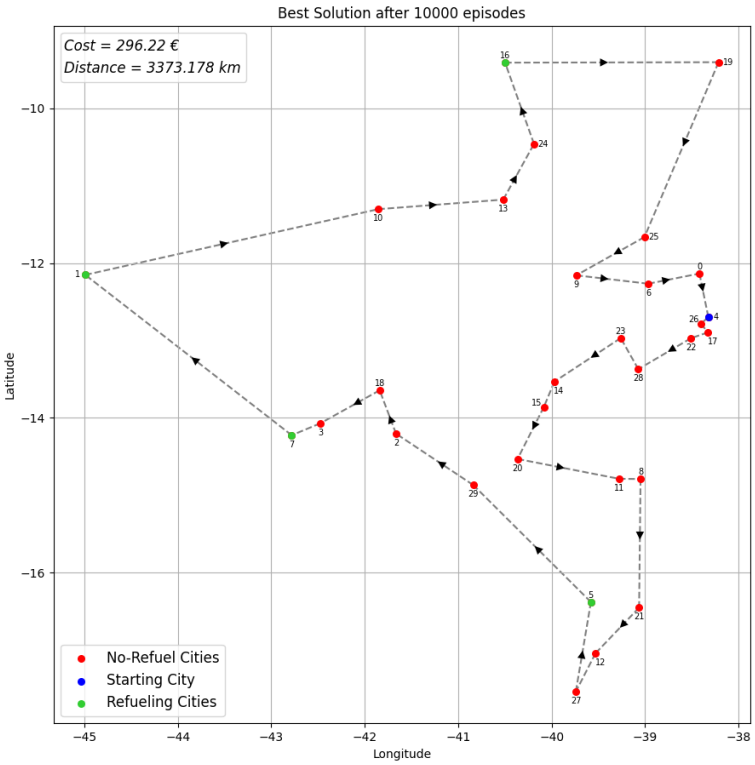
Για κάθε εξεταζόμενη περίπτωση παρουσιάζουμε 3 εικόνες, στη πρώτη φαίνεται το κόστος κάτα την διαδικασία εκμάθησης του αλγόριθμου RL, η δεύτερη αποτελείται από την διαδρομή που ακολουθεί ο agent στο πρώτο επεισόδιο συγκριτικά με την διαδρομή που ακολουθεί στη καλύτερη λύση που βρήκε ο αλγόριθμος ενισχυτικής μάθησης, και τέλος στη τρίτη δείχνω, πάλι σε μορφή χάρτη, την βελτίωση μέσω του heuristic improvement algorithm που παρουσιάσαμε, η οποία είναι η προτεινόμενη λύση.

Bahia30D (265.61€):

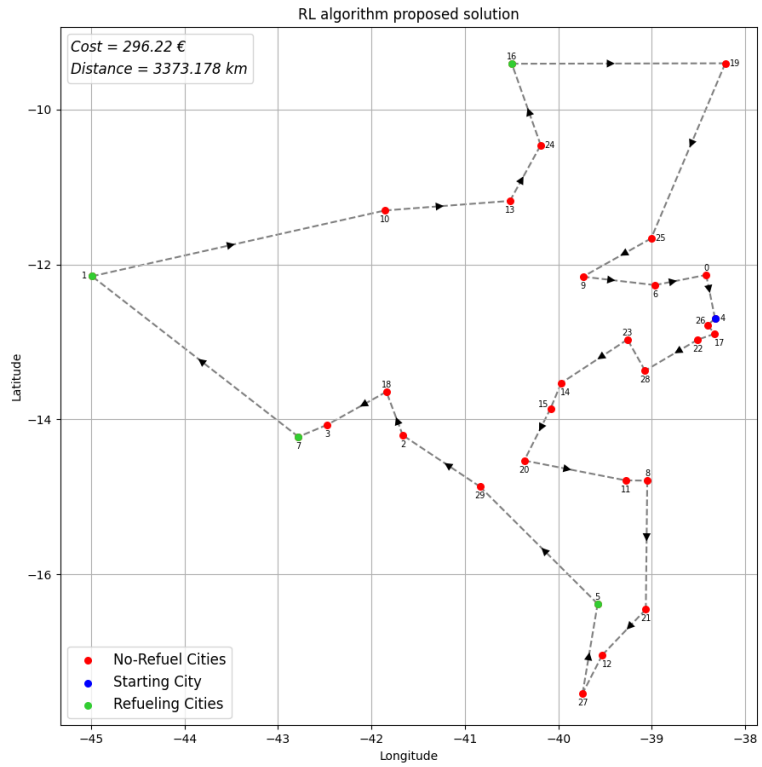


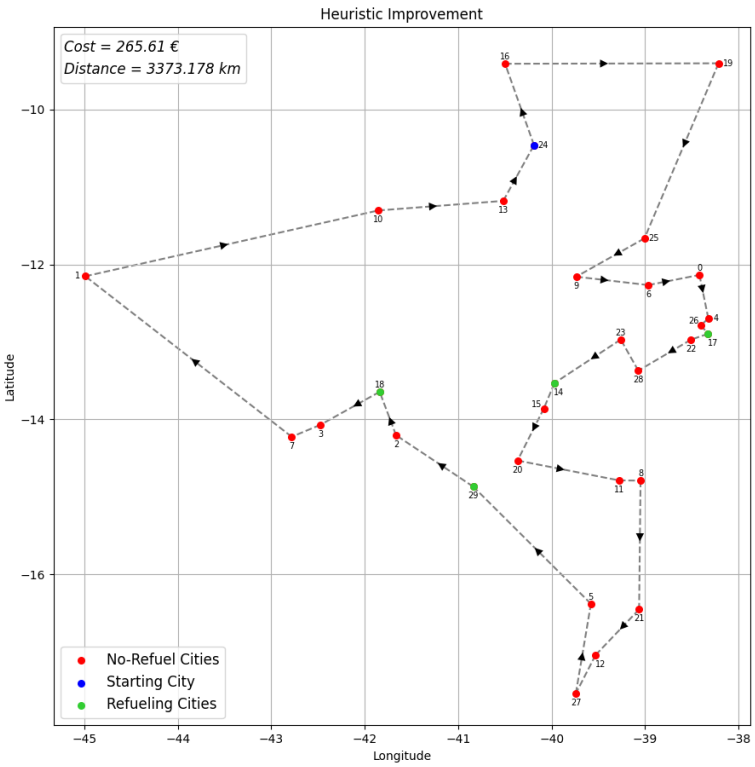
Εικόνα Α΄.1: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα Α, Bahia30D.





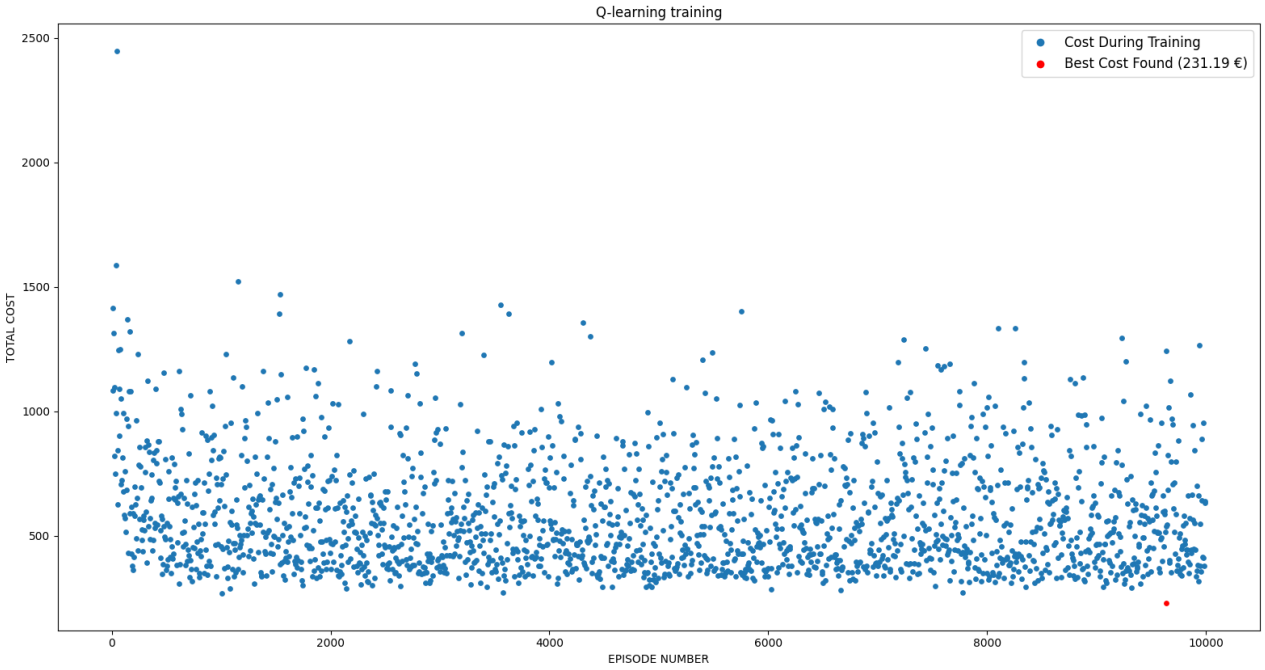
Εικόνα Α΄.2: RL εκπαίδευση για: Πρόβλημα A, Bahia30D



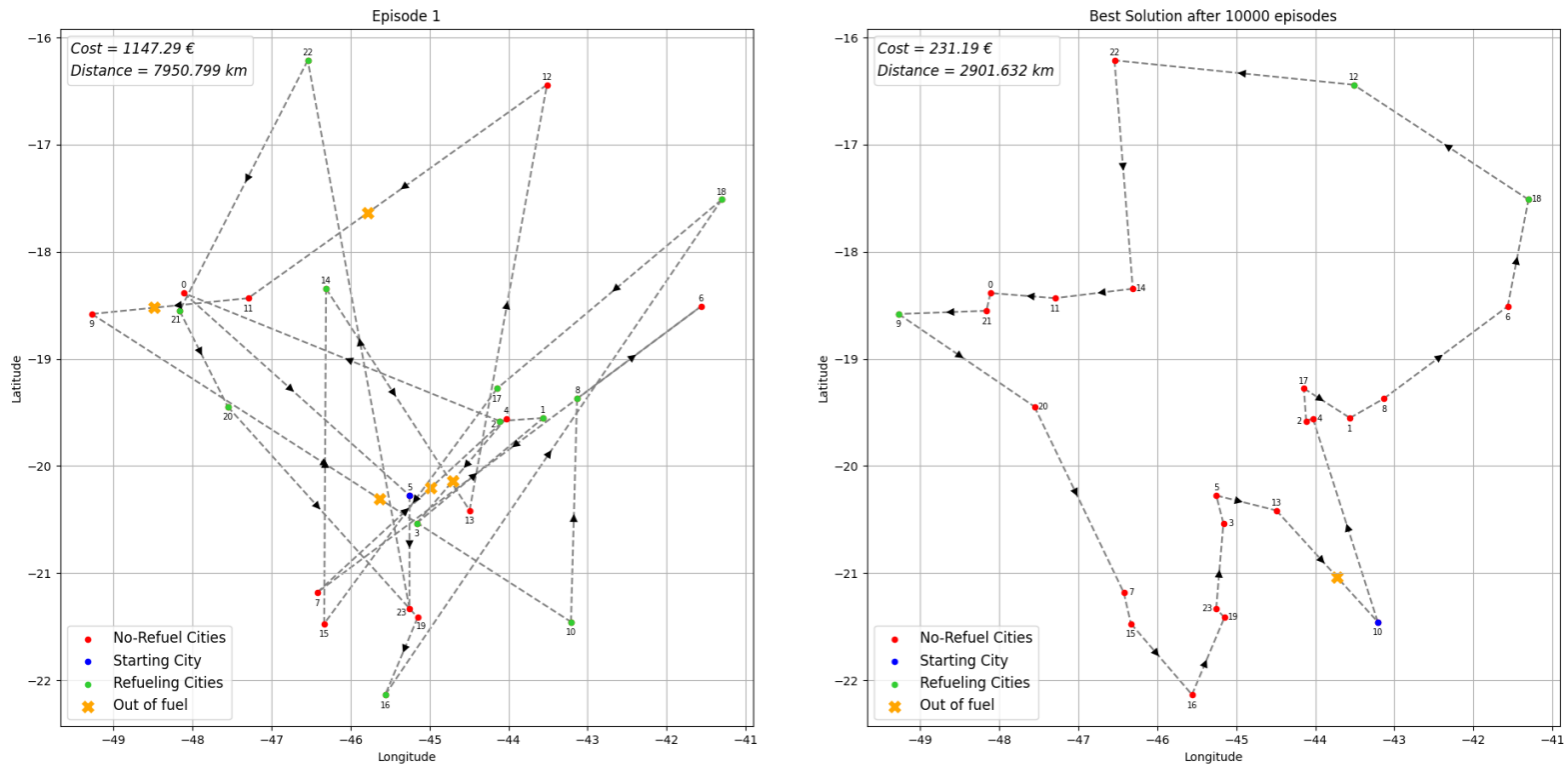


Εικόνα Α΄.3: Η.Ι. που οδήγησε στην καλύτερη λύση για: Πρόβλημα A, Bahia30D.

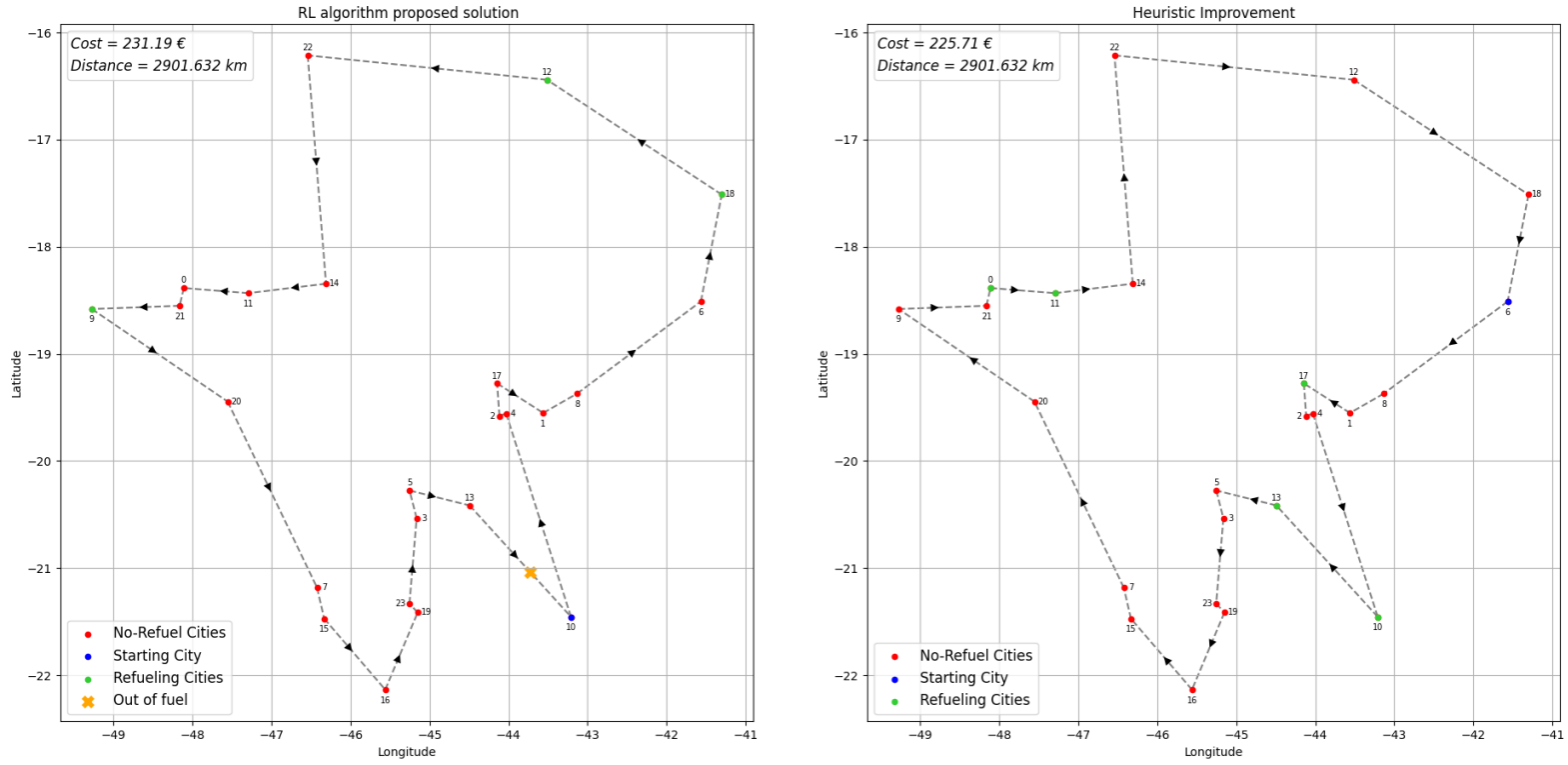
Minas24D (225.71€):



Εικόνα Α΄.4: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα A, Minas24D.

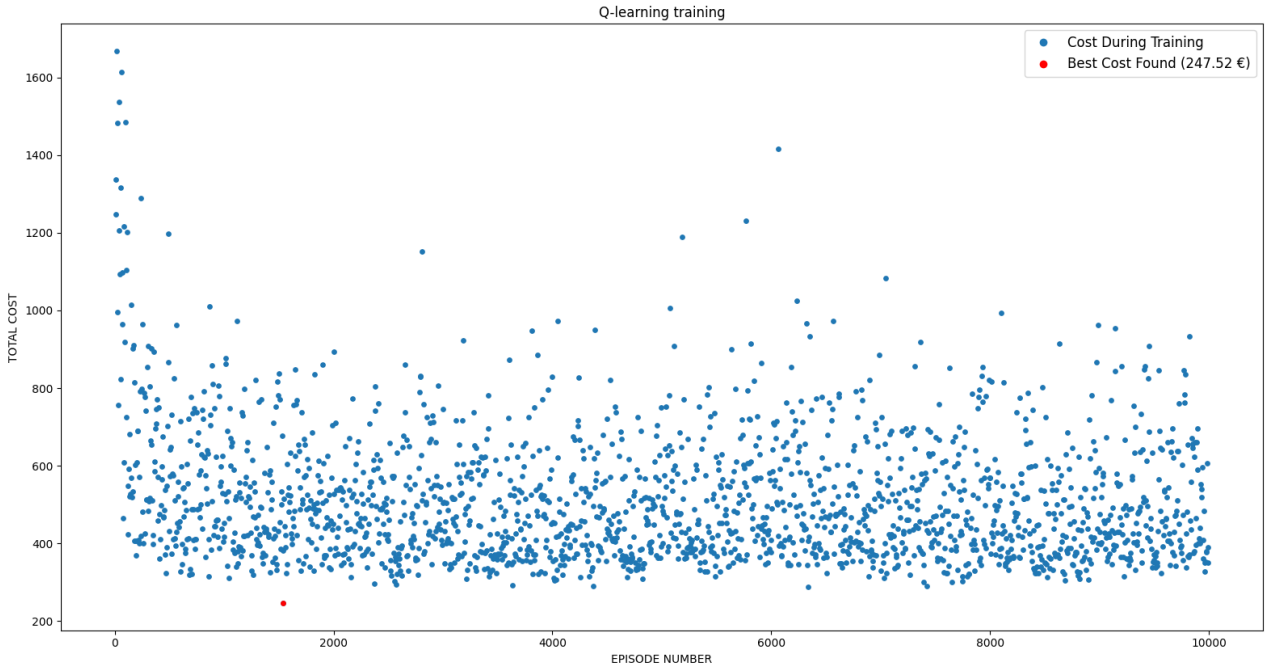


Εικόνα Α΄.5: RL εκπαίδευση για: Πρόβλημα A, Minas24D

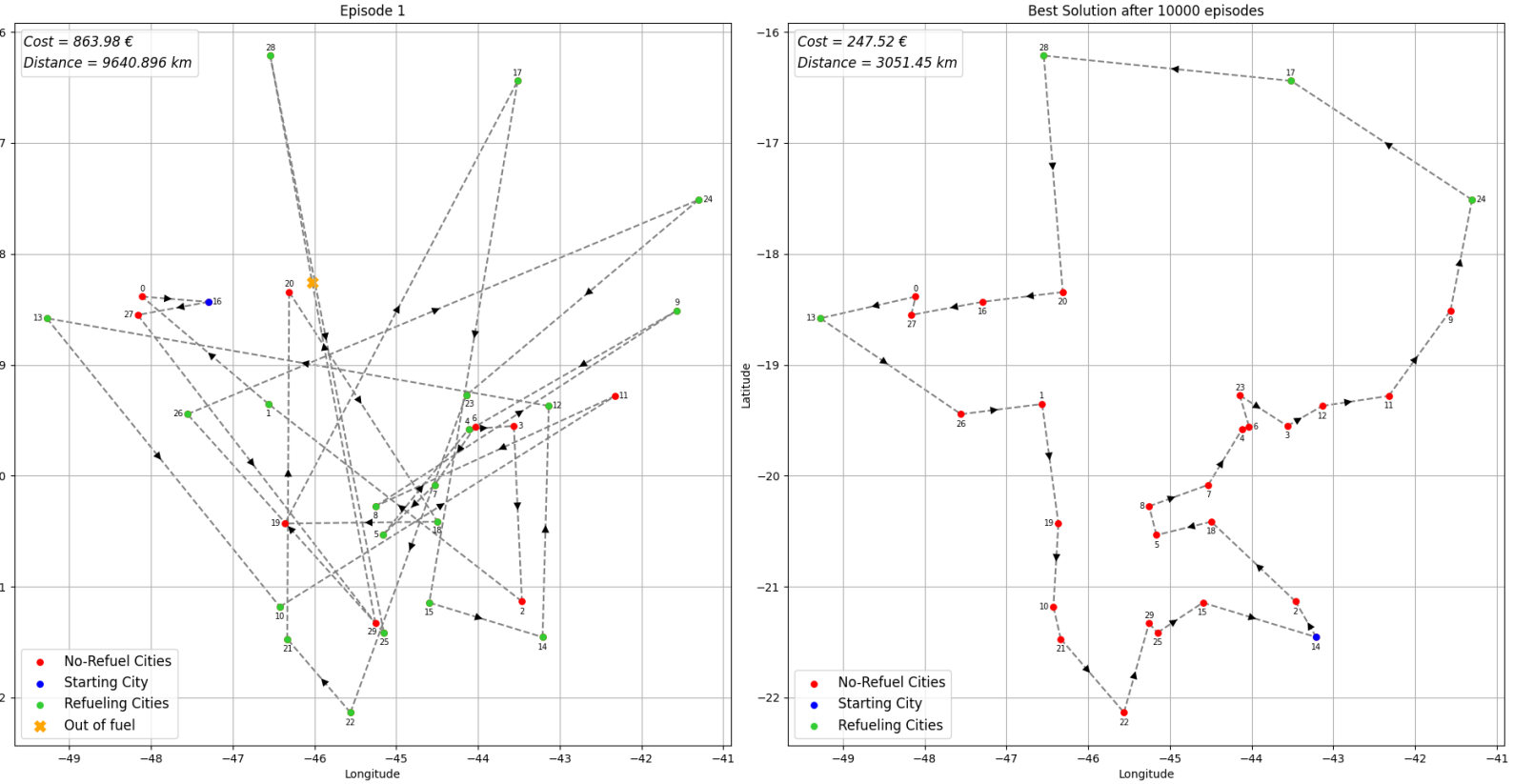


Εικόνα Α΄.6: Η.Ι. που οδήγησε στην καλύτερη λύση για: Πρόβλημα Α, Minas24D.

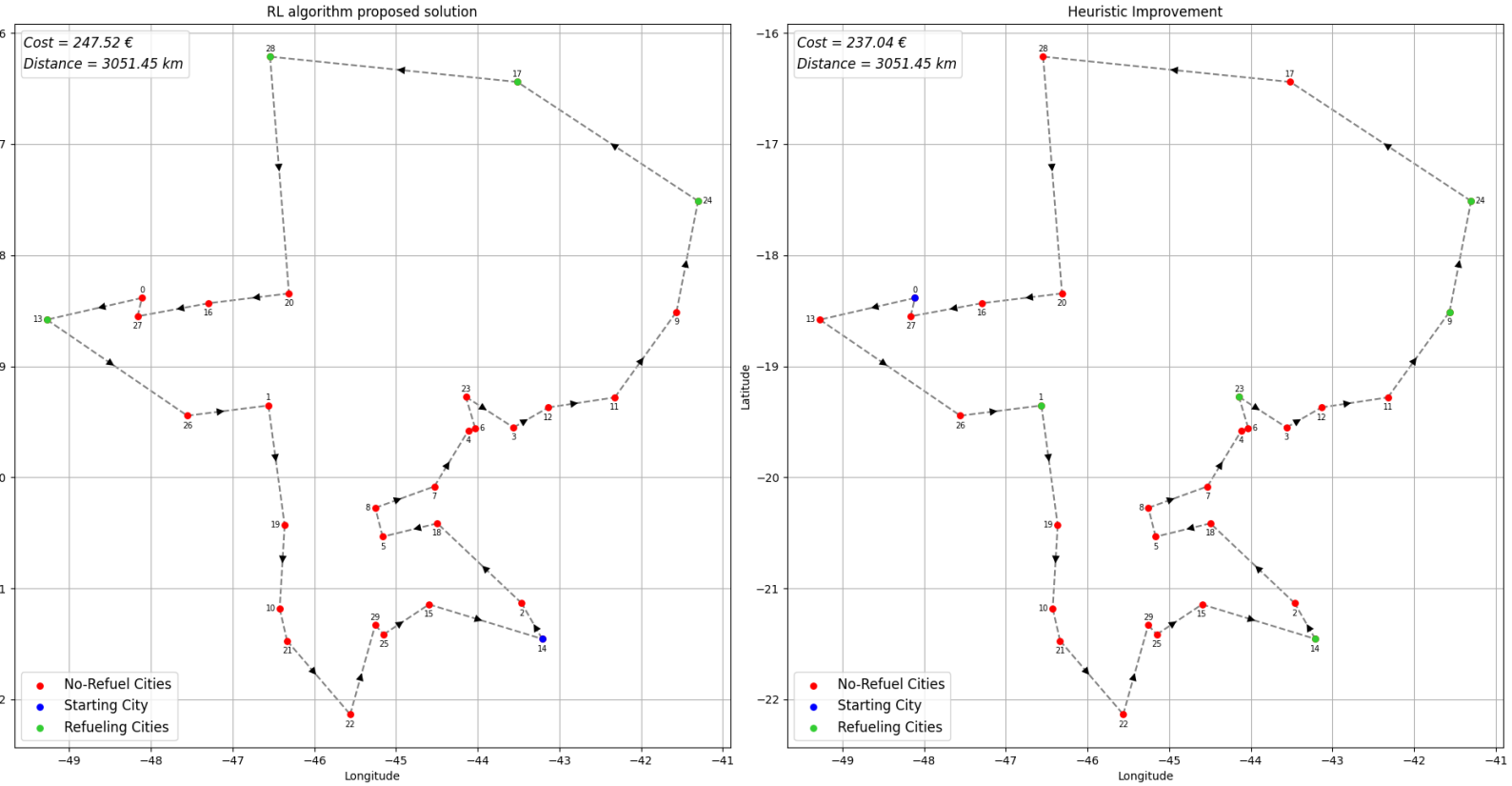
Minas30D (237.04€):



Εικόνα Α΄.7: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα Α, Minas30D.

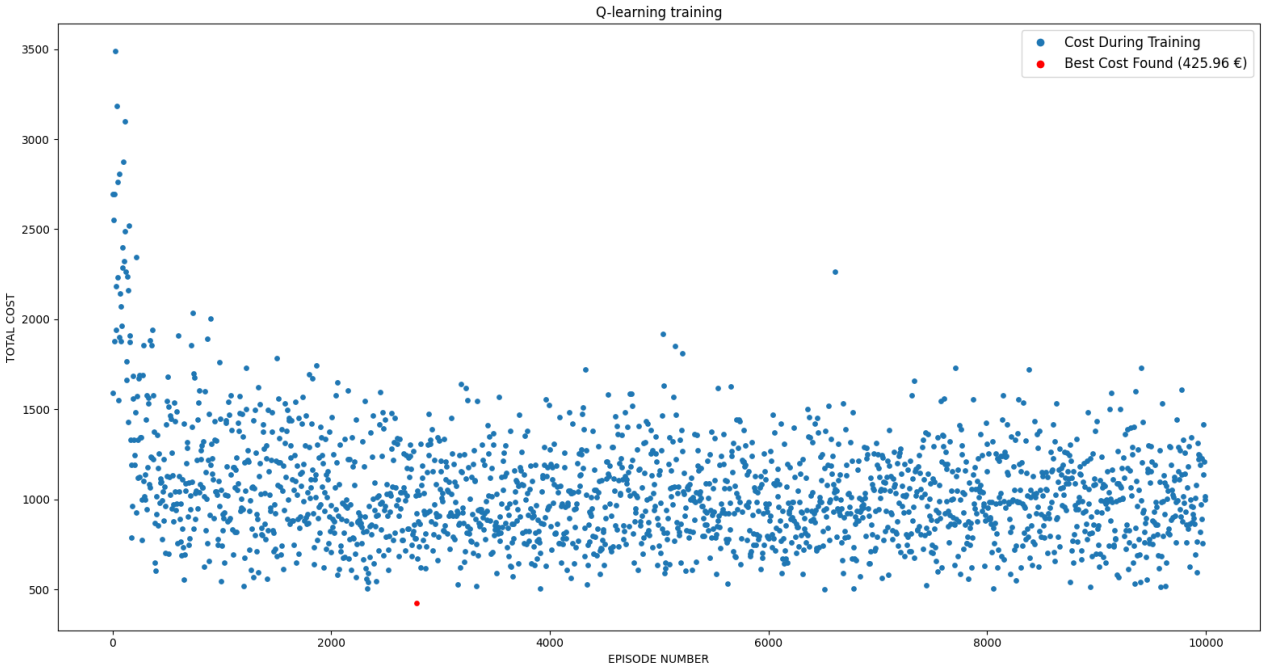


Εικόνα Α΄.8: RL εκπαίδευση για: Πρόβλημα A, Minas30D

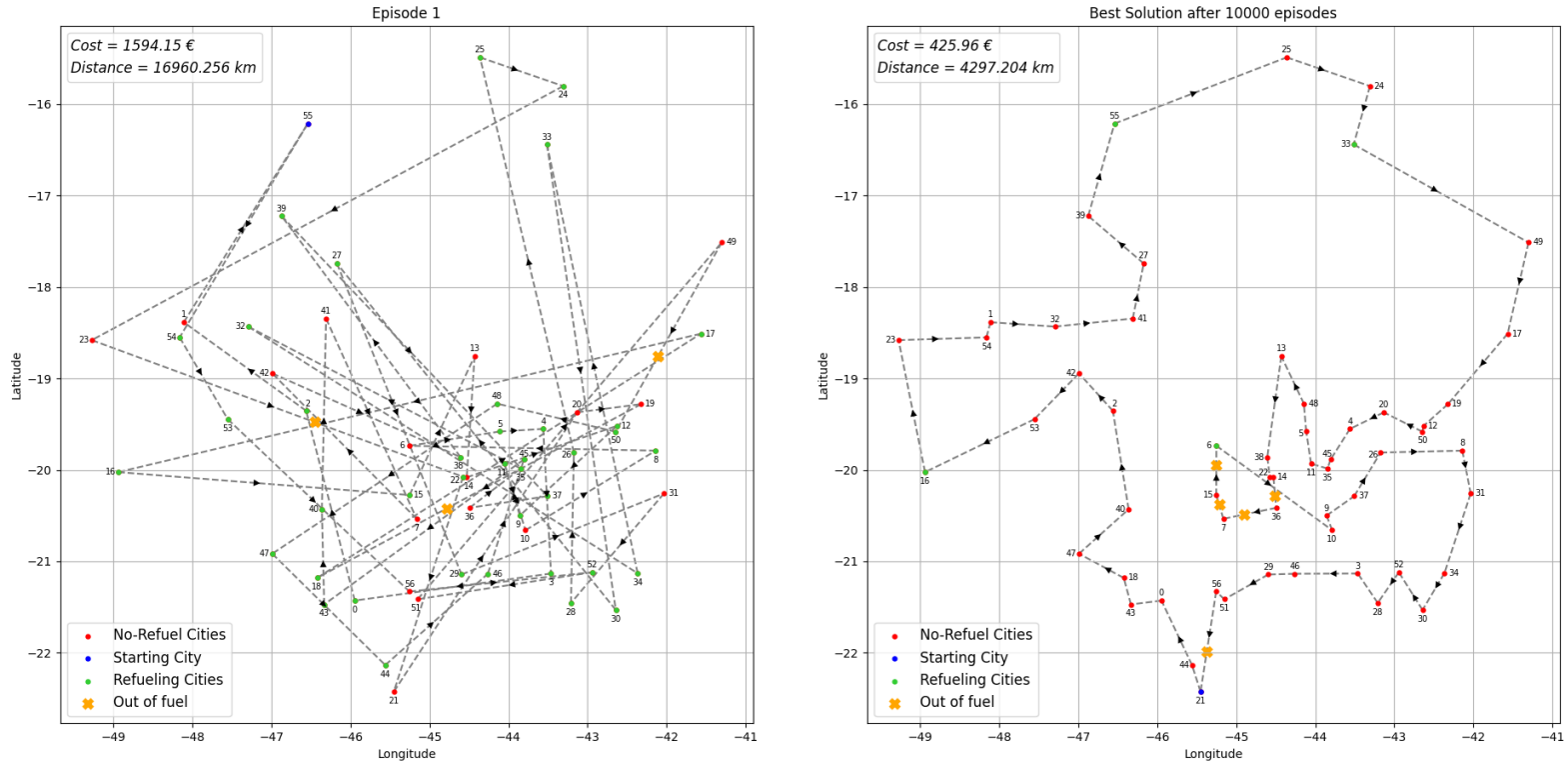


Εικόνα Α΄.9: H.I. που οδήγησε στην καλύτερη λύση για: Πρόβλημα A, Minas30D.

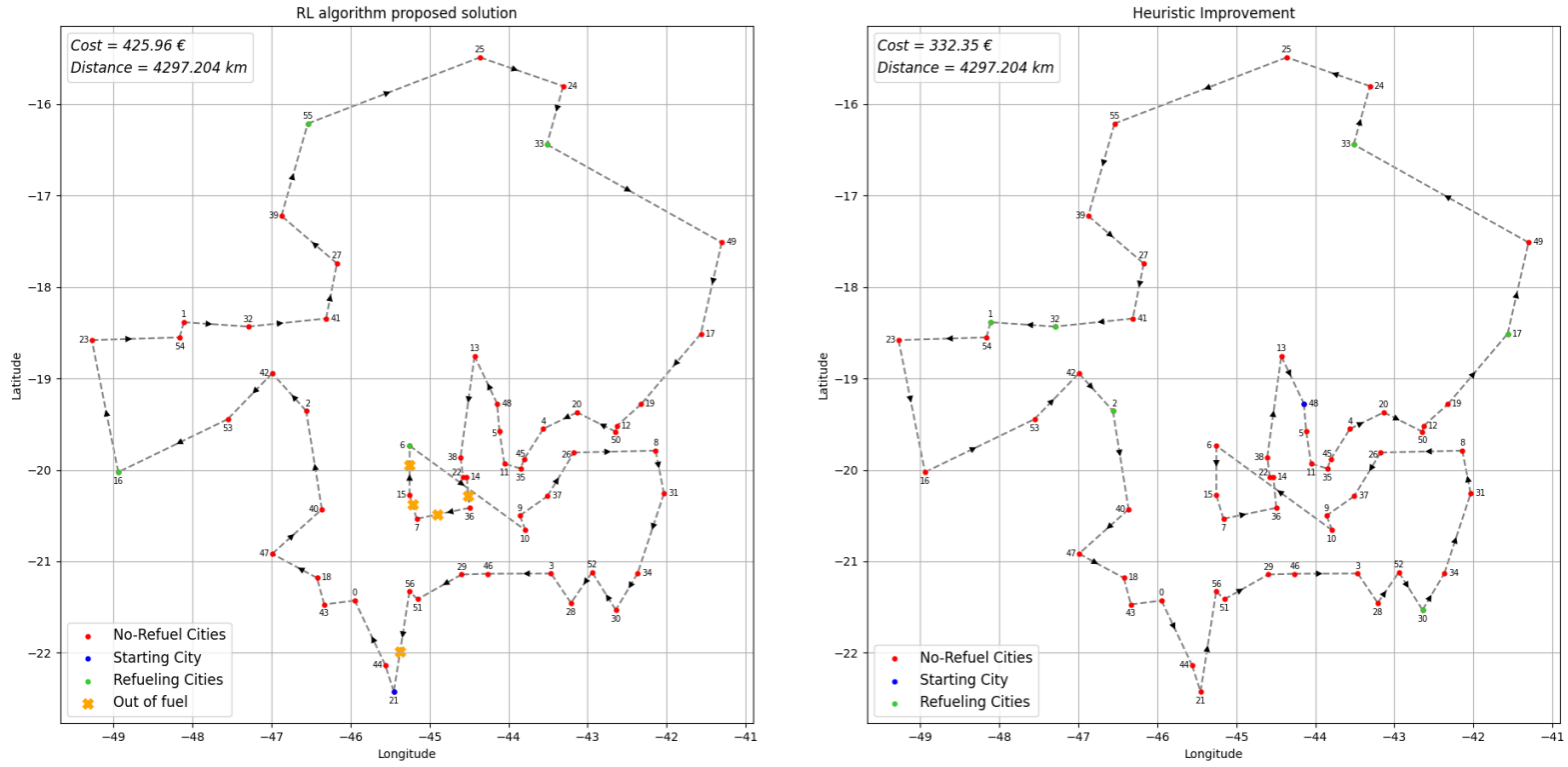
Minas57D (332.35€):



Εικόνα Α΄.10: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα A, Minas57D.



Εικόνα Α΄.11: RL εκπαίδευση για: Πρόβλημα A, Minas57D



Εικόνα Α΄.12: Η.Ι. που οδήγησε στην καλύτερη λύση για: Πρόβλημα Α, Minas57D.

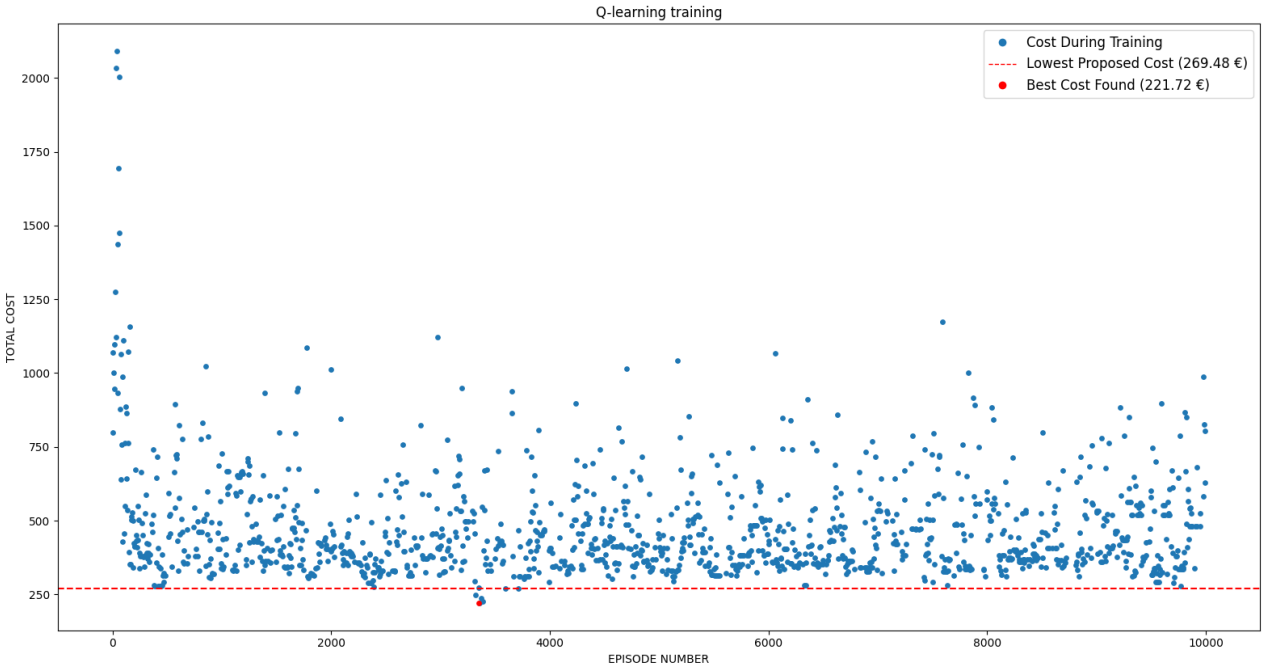
Σημείωση: Όλες οι εικόνες στο παράρτημα προκύπτουν με τις παραμέτρους που οδηγούν στην καλύτερη τελική λύση για κάθε instance, αυτές οι παράμετροι σημειώνονται στο Υποκεφάλαιο 4.3. Επίσης αυτονόητο είναι πως οι λύσεις RL που παρουσιάζονται δεν είναι οι καλύτερες εφικτές Pre-Heuristic λύσεις, αλλά αυτές που οδηγούν σε καλύτερη τελική λύση μετά την εφαρμογή του ευρετικού αλγόριθμου βελτίωσης.

Παράρτημα Β΄

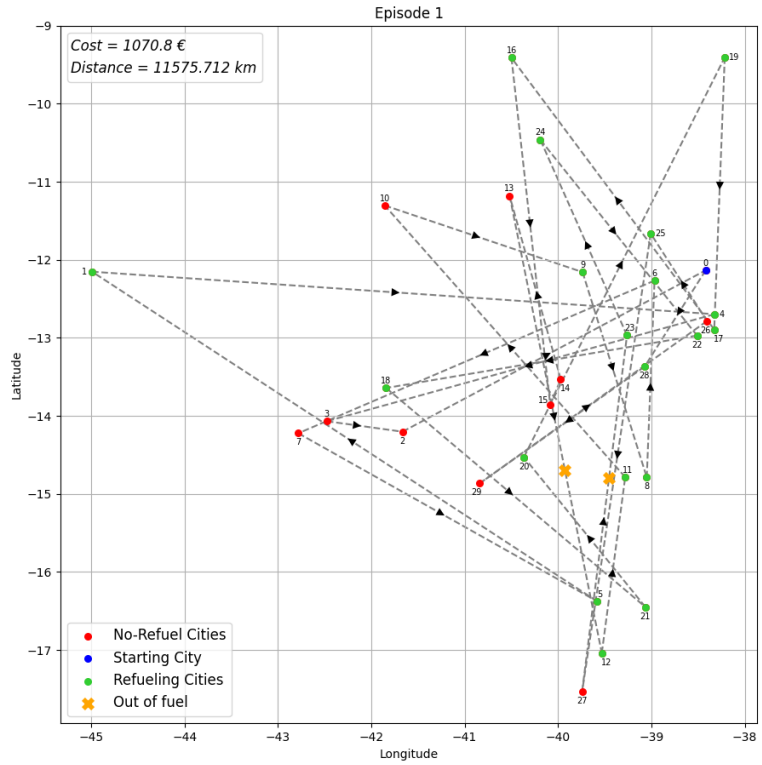
Προτεινόμενες Λύσεις για το Πρόβλημα Β

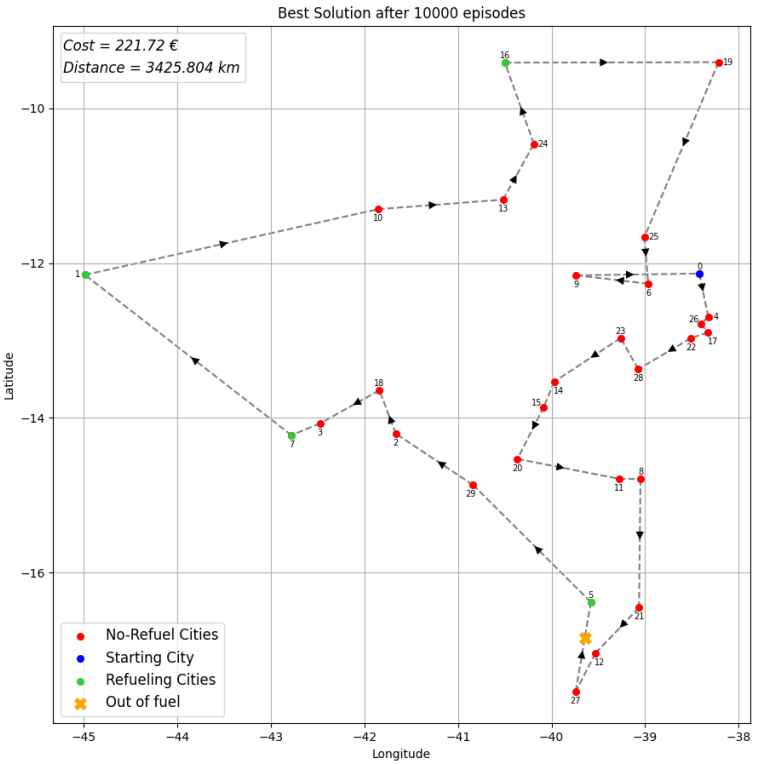
Πλέον στο γράφημα κόστους κάτα τη διαδικασία εκπαίδευσης φαίνεται και το κόστος της προτεινόμενης λύσης της εργασίας [23] με κόκκινη διακεκομένη γραμμή.

Bahia30D (192.81€):

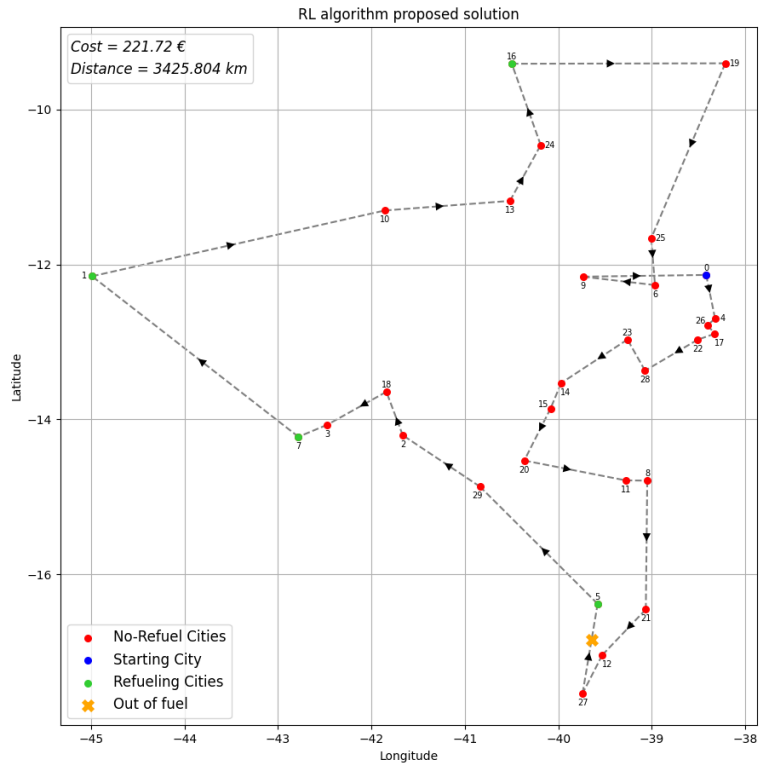


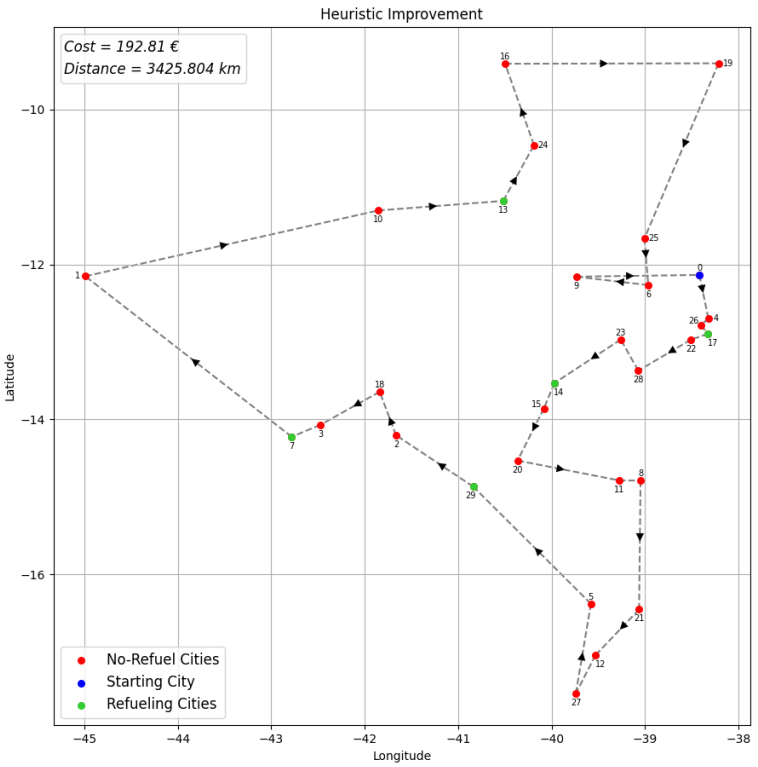
Εικόνα Β΄.1: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα Β, Bahia30D.





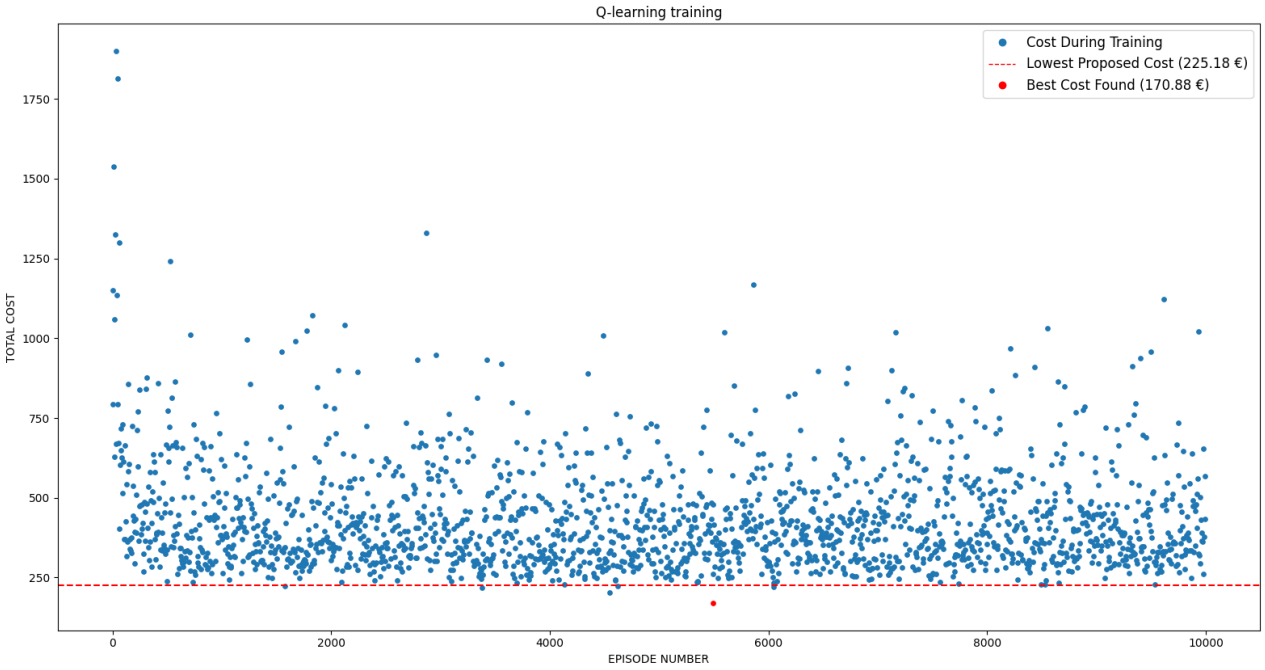
Εικόνα Β΄.2: RL εκπαίδευση για: Πρόβλημα B, Bahia30D



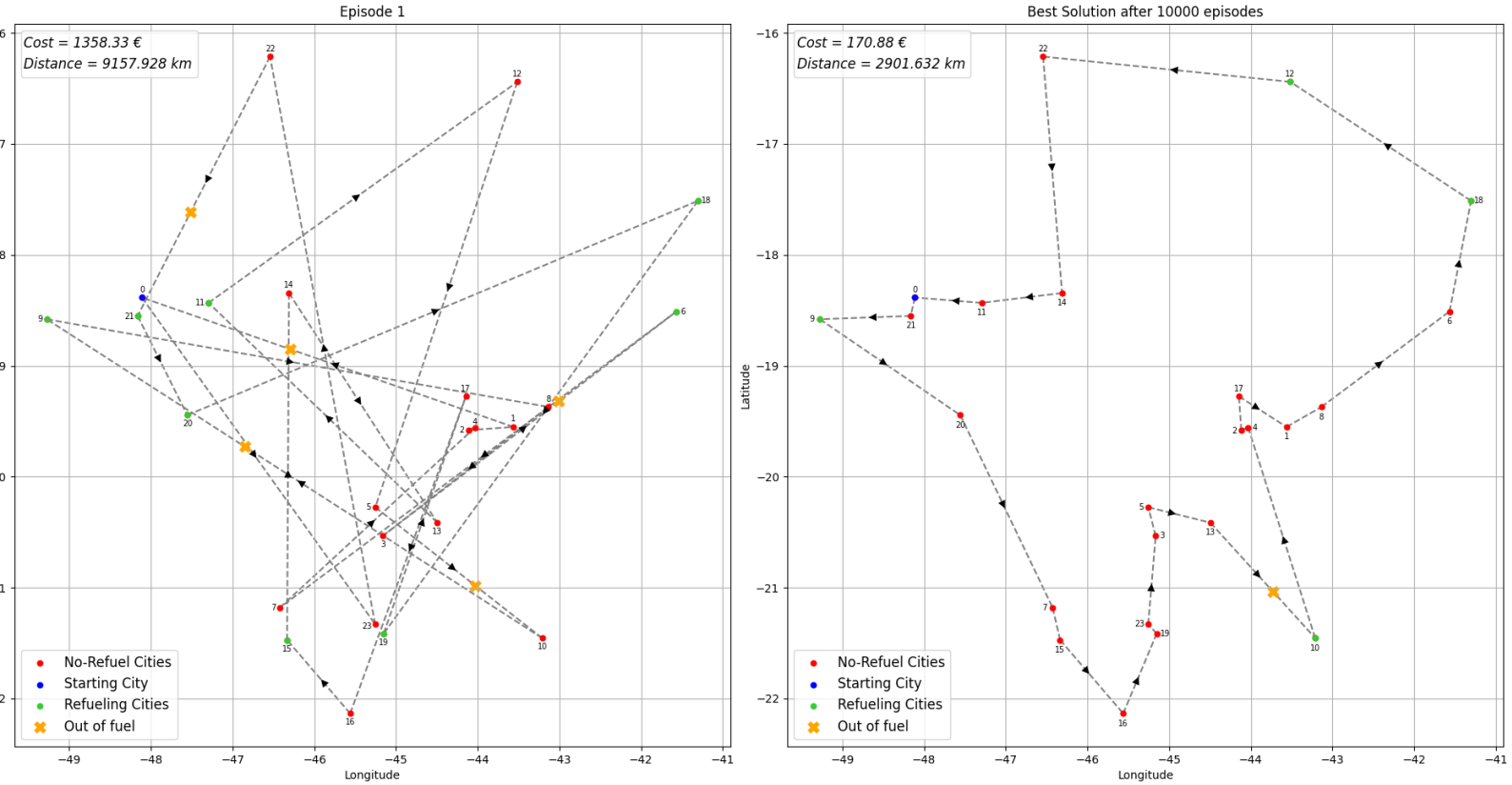


Εικόνα Β΄.3: Η.Ι. που οδήγησε στην καλύτερη λύση για: Πρόβλημα B, Bahia30D.

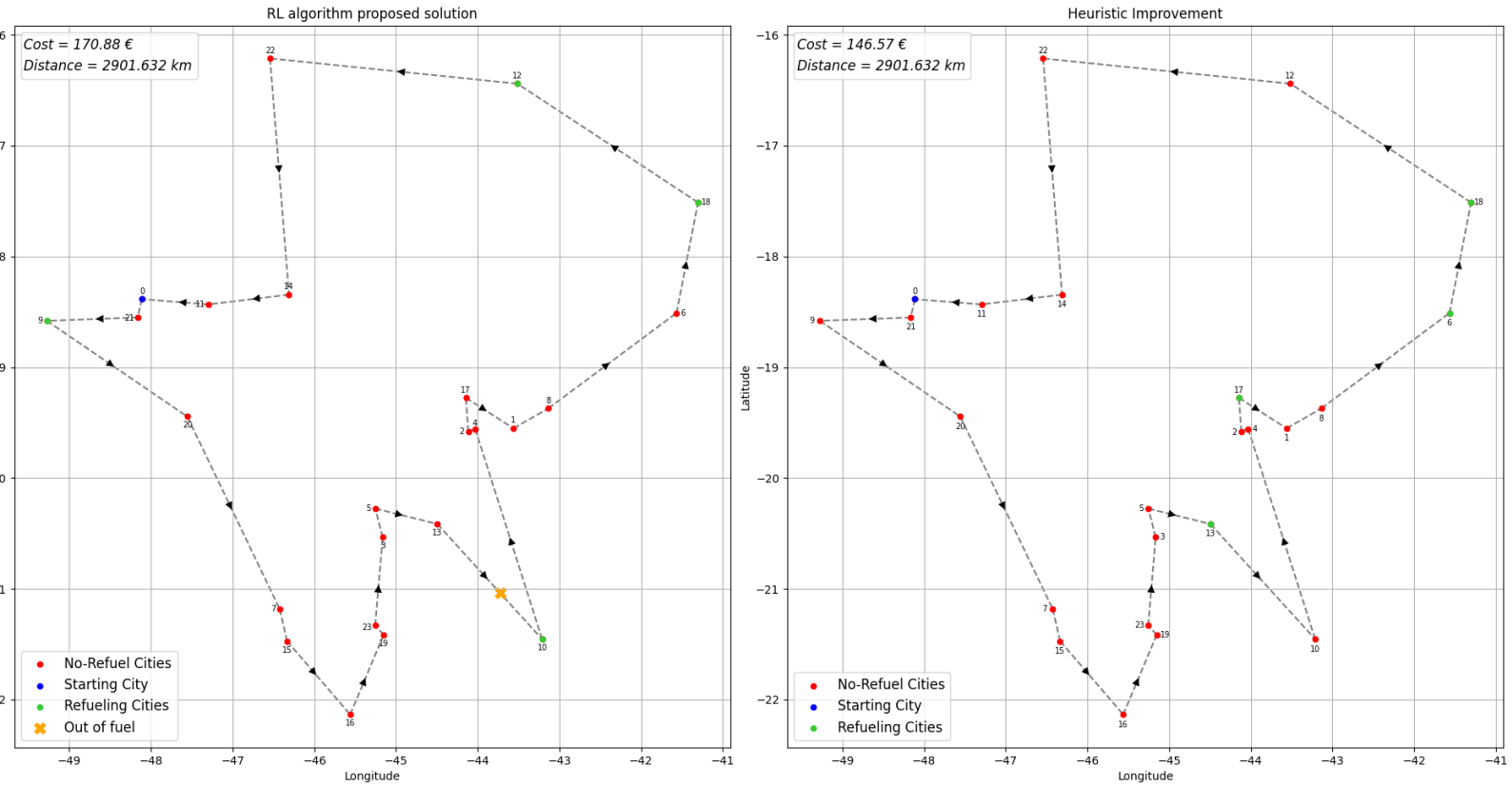
Minas24D (146.57€):



Εικόνα Β΄.4: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα Β, Minas24D.

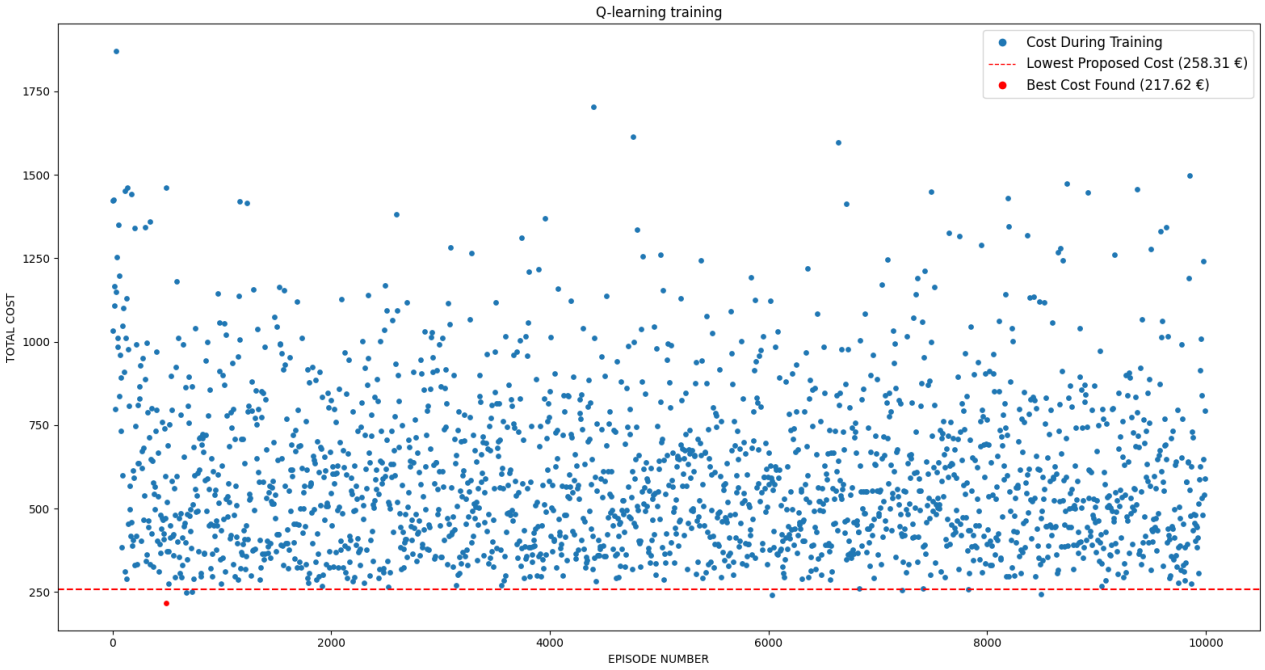


Εικόνα Β΄.5: RL εκπαίδευση για: Πρόβλημα Β, Minas24D

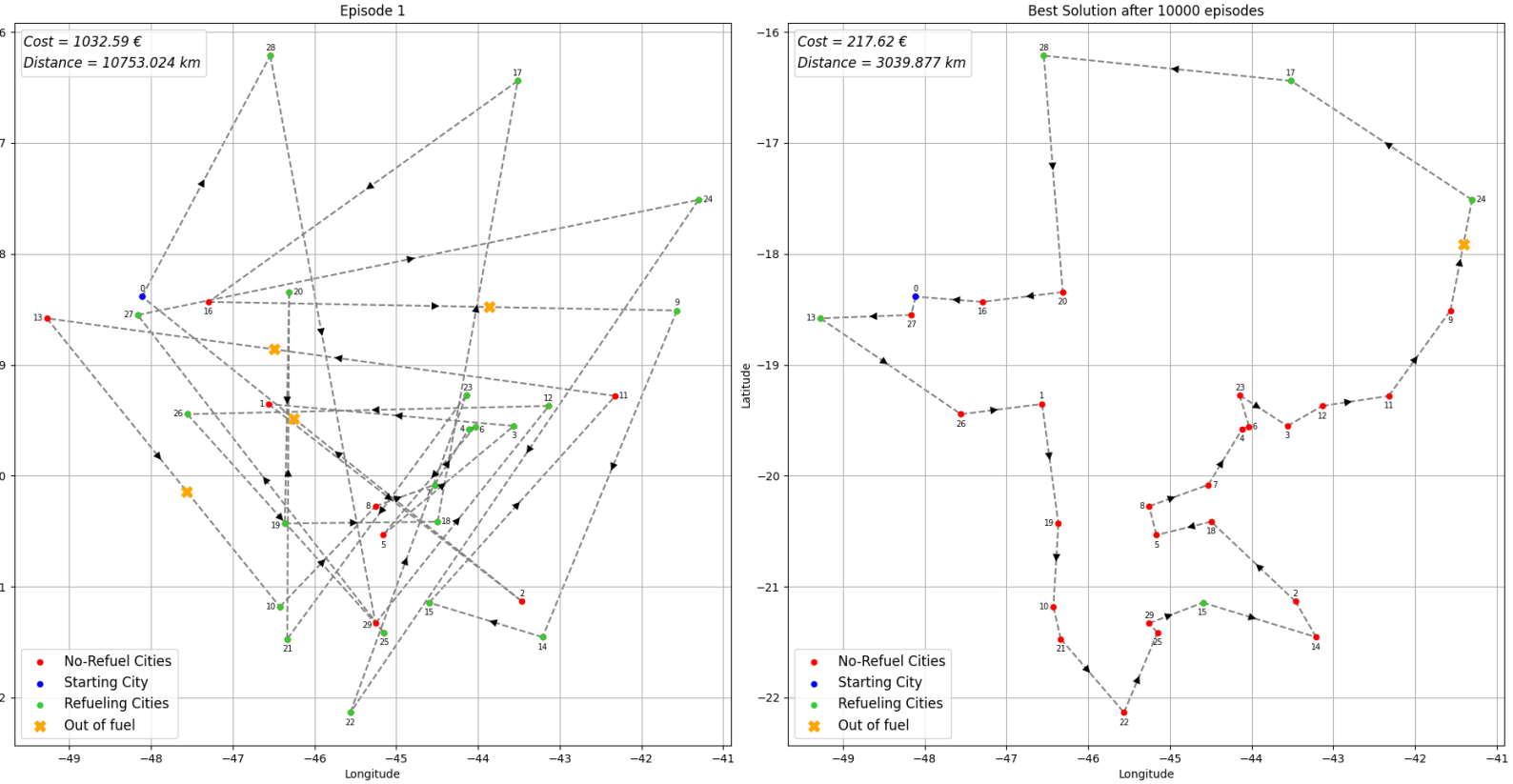


Εικόνα Β΄.6: Η.Ι. που οδήγησε στην καλύτερη λύση για: Πρόβλημα Β, Minas24D.

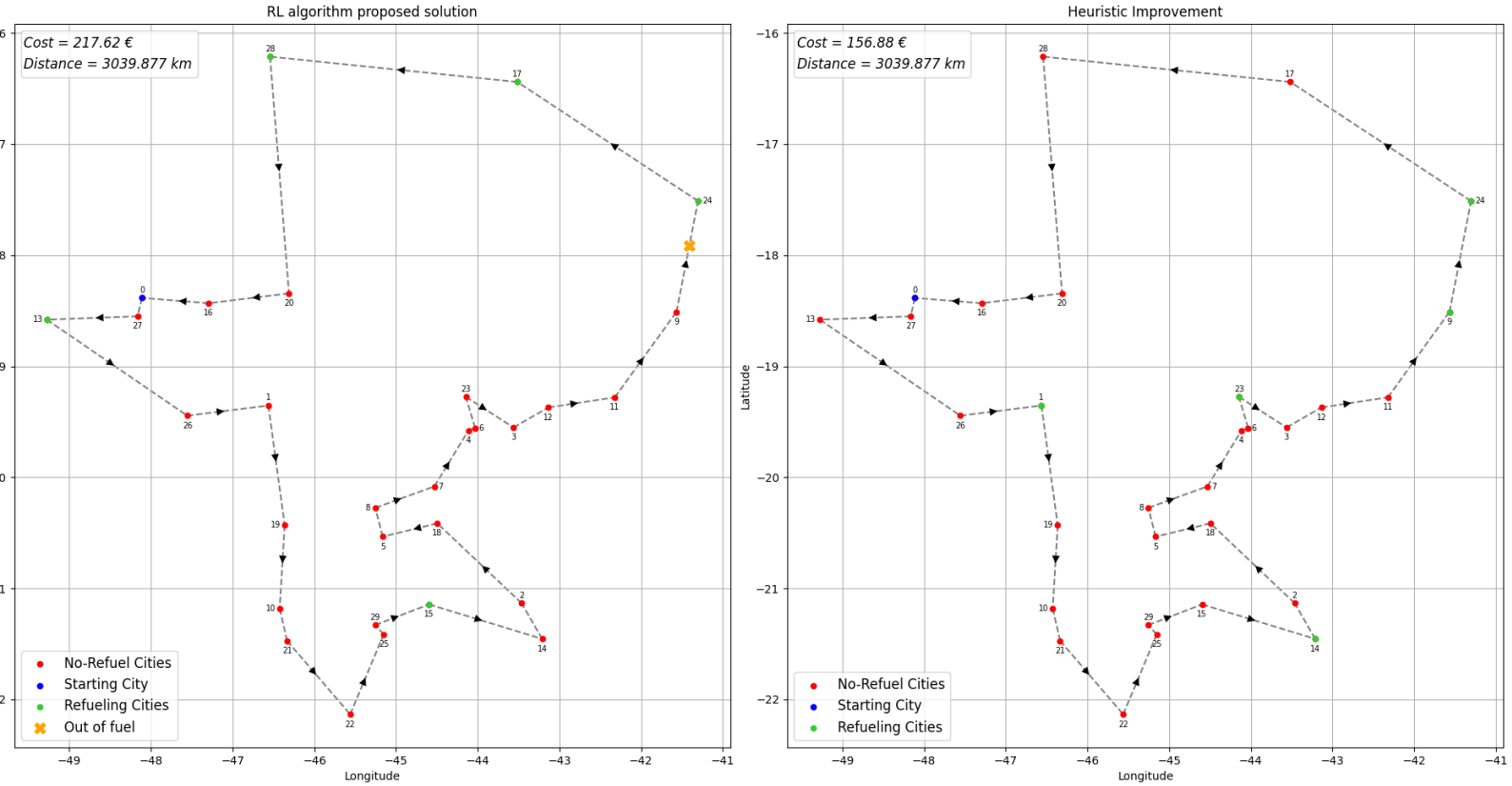
Minas30D (156.88€):



Εικόνα Β΄.7: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα Β, Minas30D.

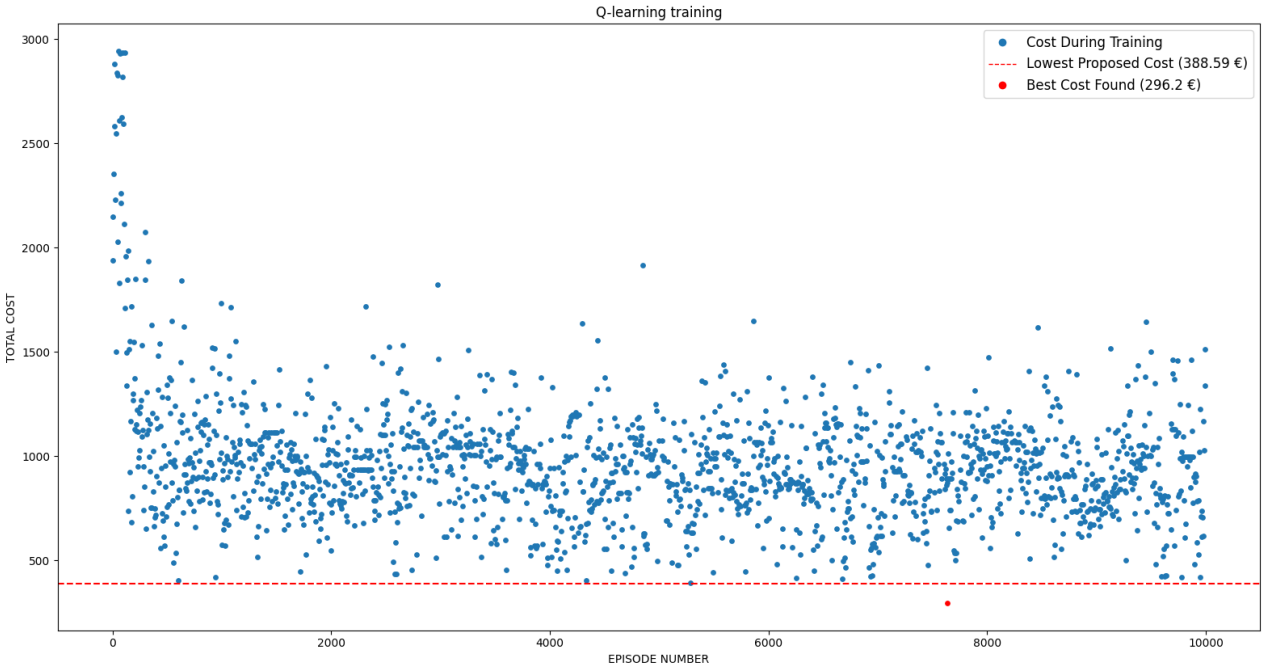


Εικόνα Β΄.8: RL εκπαίδευση για: Πρόβλημα Β, Minas30D

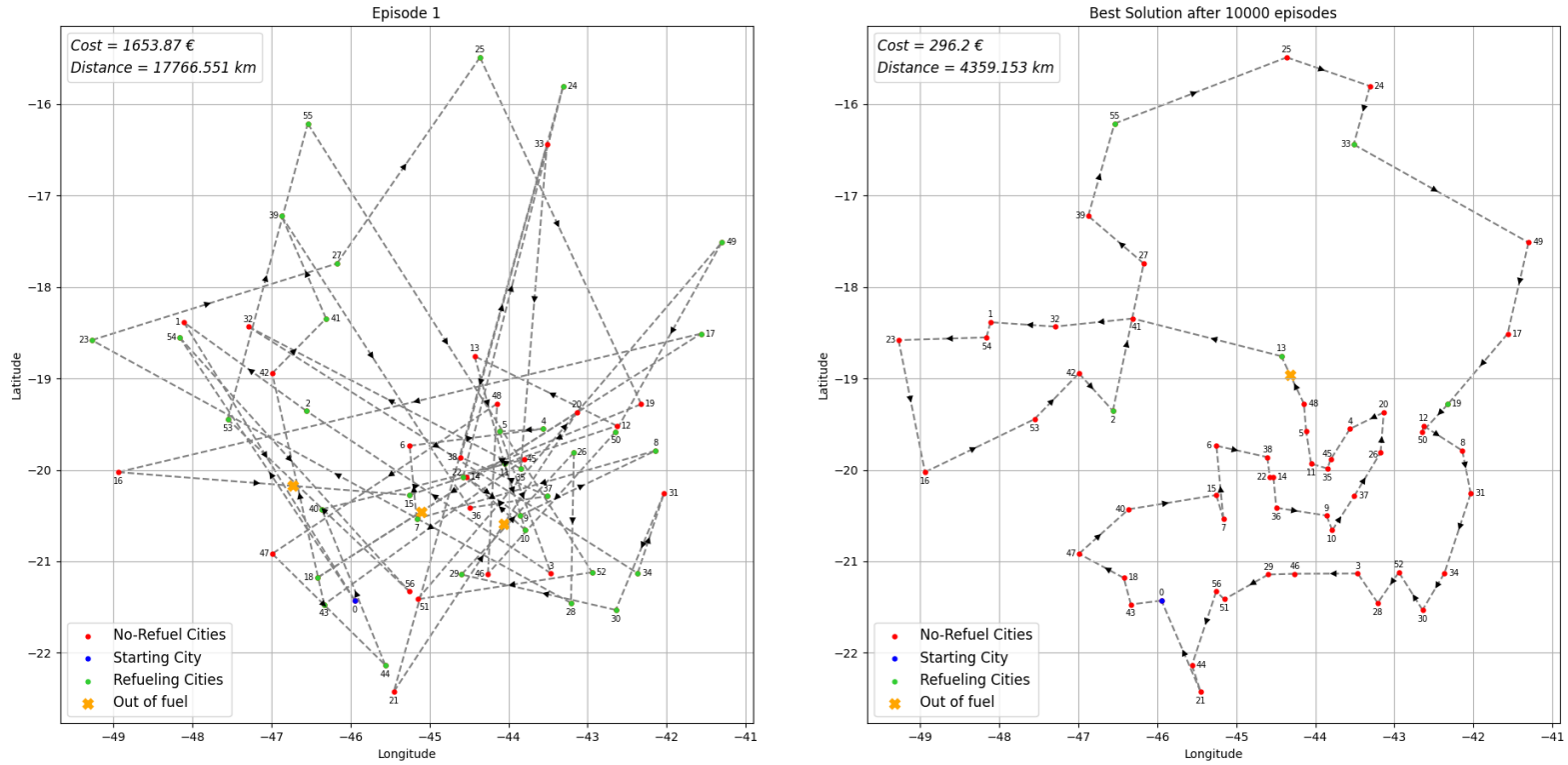


Εικόνα Β΄.9: Η.Ι. που οδήγησε στην καλύτερη λύση για: Πρόβλημα Β, Minas30D.

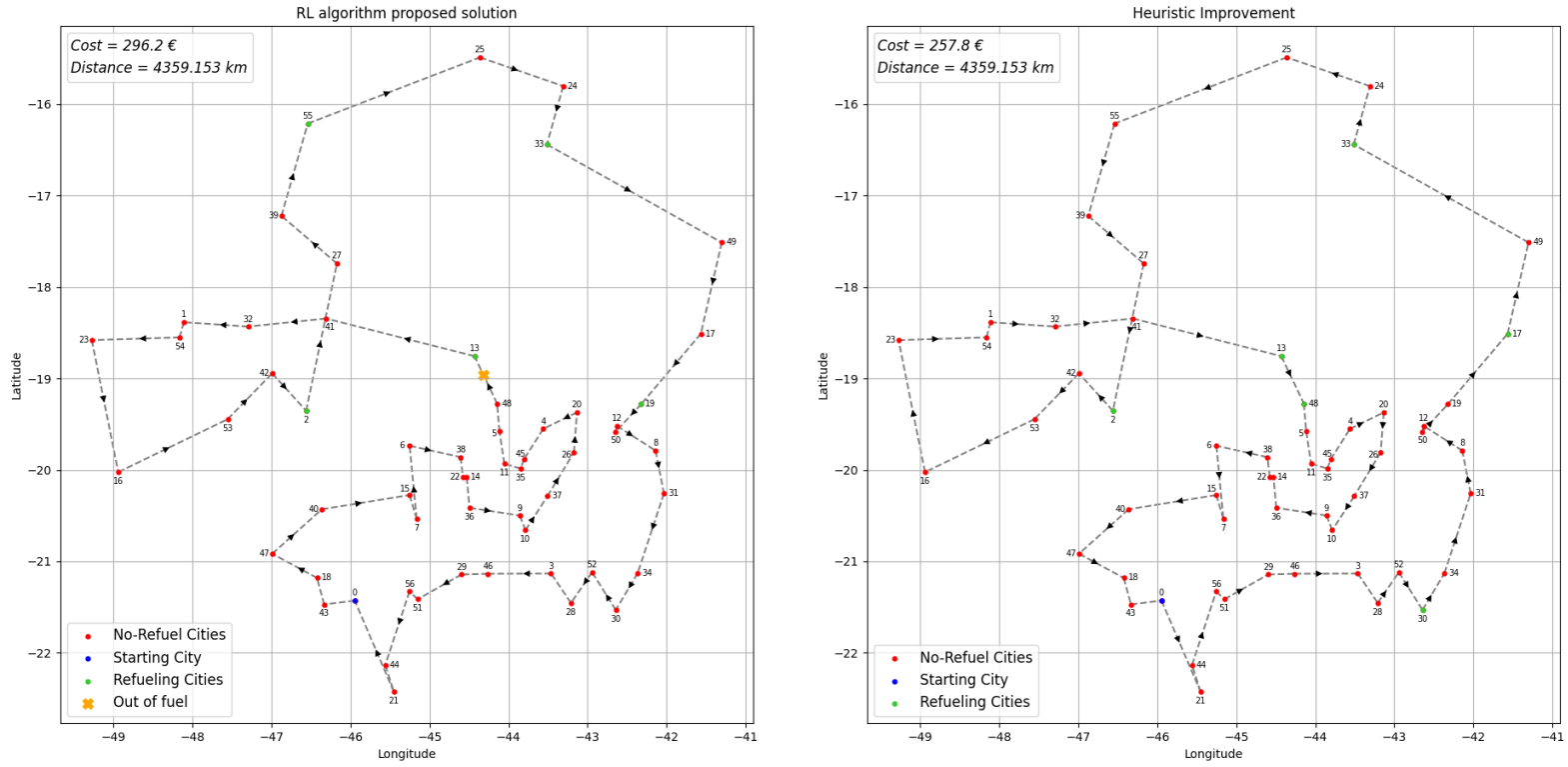
Minas57D (257.80€):



Εικόνα Β΄.10: Κόστος κατά την εκπαίδευση RL για: Πρόβλημα Β, Minas57D.



Εικόνα Β΄.11: RL εκπαίδευση για: Πρόβλημα Β, Minas57D



Εικόνα Β΄.12: Η.Ι. που οδήγησε στην καλύτερη λύση για: Πρόβλημα Β, Minas57D.

Bibliography

[1] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. Wiley, 1999.

[2] J. Zhang, C. Liu, J. Yan, X. Li, H.-L. Zhen, and M. Yuan, "A survey for solving mixed integer programming via machine learning," *Neurocomputing*, vol. 519, pp. 205–217, 2023.

[3] E. Alpaydin, *Machine Learning*. MIT Press, 2021.

[4] G. Wiederhold and J. McCarthy, "Arthur Samuel: Pioneer in machine learning," *IBM Journal of Research and Development*, vol. 36, no. 3, pp. 329 – 331, 1992.

[5] D. Williams, "We've been here before: AI promised humanlike machines in 1958," The Conversation, February 2023. [Online]. Available: https://theconversation.com/weve-been-here-before-ai-promised-humanlike-machines-in-1958-222700

[6] A. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[7] R. Karjian, "History and evolution of machine learning: A timeline," TechTarget, June 2024. [Online]. Available: https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History

[8] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[9] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21 – 27, 1967.

[10] N. J. Nilsson, "Shakey the robot," Stanford Research Institute, Tech. Rep., 1984. [Online]. Available: https://ai.stanford.edu/~nilsson/OnlinePubs-Nils/shakey-the-robot.pdf

[11] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[12] Chess.com, "Deep blue vs. Garry Kasparov: The significance of their chess matches," Chess.com, October 2018. [Online]. Available: https://www.chess.com/article/view/deep-blue-kasparov-chess

[13] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine Learning Techniques for Multimedia*, ser. Cognitive Technologies, M. Cord and P. Cunningham, Eds. Springer, Berlin, Heidelberg, 2008, ch. 2, pp. 21–49.

[14] M. E. Celebi and K. Aydin, *Unsupervised learning algorithms*. Springer, 2016.

[15] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[16] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Science Review*, vol. 5, no. 1, p. 30–43, 2017.

[17] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[18] P. Probst, A.-L. Boulesteix, and B. Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms," *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1–32, 2019.

[19] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[20] E. Barron and H. Ishii, "The Bellman equation for minimizing the maximum cost." *Nonlinear Anal. Theory Methods Applic.*, vol. 13, no. 9, pp. 1067–1090, 1989.

[21] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM computing surveys (CSUR)*, vol. 45, no. 3, pp. 1–33, 2013.

[22] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[23] A. L. Ottoni, E. G. Nepomuceno, M. S. d. Oliveira, and D. C. d. Oliveira, "Reinforcement learning for the traveling salesman problem with refueling," *Complex & Intelligent Systems*, vol. 8, no. 3, pp. 2001–2015, 2022.

[24] N. R. Chopde and M. Nichat, "Landmark based shortest path detection by using a* and haversine formula," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 2, pp. 298–302, 2013.