



Μεθευρετικές μέθοδοι και τεχνικές εμπνευσμένες από τον
κβαντικό υπολογισμό για αλγόριθμους βελτιστοποίησης

Χρήστος Παπαλίτσας

Δήλωση Αυθεντικότητας

Εγώ, ο Χρήστος Παπαλίτσας, δηλώνω ότι η διατριβή με τίτλο "Μεθευρετικές μέθοδοι και τεχνικές εμπνευσμένες από τον κβαντικό υπολογισμό για αλγόριθμους βελτιστοποίησης" και οι εργασίες που παρουσιάζονται σε αυτήν, μου έχουν δημιουργηθεί από εμένα.

Επιβεβαιώνω ότι:

- Αυτή η εργασία έγινε εξ' ολοκλήρου για την εκπόνηση της διδακτορικής έρευνας στο Ιόνιο Πανεπιστήμιο.
- Οποιοδήποτε μέρος αυτής της διατριβής έχει προηγουμένως υποβληθεί για πτυχίο ή οποιοδήποτε άλλο πτυχίο σε αυτό το Πανεπιστήμιο ή σε οποιοδήποτε άλλο ίδρυμα, αυτό έχει δηλωθεί με σαφήνεια.
- Όπου έχω συμβουλευτεί το δημοσιευμένο έργο άλλων, αυτό αποδίδεται πάντα ξεκάθαρα.
- Όπου έχω αναφέρει από το έργο άλλων, η πηγή δίνεται πάντα. Με εξαίρεση τέτοια αποσπάσματα, αυτή η διατριβή είναι εξ' ολοκλήρου δική μου δουλειά.
- Όπου η διατριβή βασίζεται στη δουλειά που έκανα ο ίδιος από κοινού με άλλους, έχω ξεκαθαρίσει ακριβώς τι έγινε από άλλους και τι έχω συνεισφέρει.

Υπογραφή:

Ημερομηνία:

Διδακτορική Διατριβή, Ιόνιο Πανεπιστήμιο

Η παρούσα διδακτορική διατριβή πραγματοποιήθηκε υπό την επίβλεψη του Δρ. Θοδωρή Ανδρόνικου

Ιούλιος 2020

Μεθευρετικές μέθοδοι και τεχνικές εμπνευσμένες από τον χβαντικό
υπολογισμό για αλγόριθμους βελτιστοποίησης

από τον Χρήστο Παπαλίτσα

Μέλη της επταμελούς επιτροπής

Θοδωρής Ανδρόνικος
Αναπληρωτής Καθηγητής
Επιβλέπων
Τμήμα Πληροφορικής
Ιόνιο Πανεπιστήμιο

Παναγιώτης Βλάχος
Καθηγητής
Μέλος Τριμελούς Συμβουλευτικής
Επιτροπής
Τμήμα Πληροφορικής
Ιόνιο Πανεπιστήμιο

Μάρκος Αυλωνίτης
Αναπληρωτής Καθηγητής
Μέλος Τριμελούς Συμβουλευτικής
Επιτροπής
Τμήμα Πληροφορικής
Ιόνιο Πανεπιστήμιο

Μιχαήλ Στεφανιδάκης
Επίκουρος Καθηγητής
Εξεταστής
Τμήμα Πληροφορικής
Ιόνιο Πανεπιστήμιο

Νεκτάριος Κοζύρης
Καθηγητής
Εξεταστής
Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Η/Υ
Εθνικό Μετσόβιο Πολυτεχνείο

Παναγιώτης Τσανάκας
Καθηγητής
Εξεταστής
Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Η/Υ
Εθνικό Μετσόβιο Πολυτεχνείο

Σπυρίδων Σιούτας
Καθηγητής
Εξεταστής
Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Πατρών

Χωρίς την ατέλεια, ούτε εσύ, ούτε εγώ θα υπήρχαμε.

Στίβεν Χόκινγκ



Περιεχόμενα

Περίληψη	13
Abstract	15
Πρόλογος - Ευχαριστίες	17
Δημοσιεύσεις	19
Λέξεις Κλειδιά	21
1 Εισαγωγή	23
1.1 Ο τομέας του υπολογισμού	23
1.1.1 Ιστορική αναδρομή	24
1.1.2 Συμβατικός υπολογισμός	29
1.1.3 Μη συμβατικός υπολογισμός	30
1.2 Ο κλάδος της βελτιστοποίησης	31
1.2.1 Κατηγορίες προβλημάτων βελτιστοποίησης	38
1.3 Μεθευρετικές μέθοδοι	41
1.4 Μεθευρετικοί αλγόριθμοι για βελτιστοποίηση	44
2 Η Μεθοδολογία	51
2.1 Προσεγγιστικές και μεθευρετικές μέθοδοι	51
2.1.1 Εισαγωγή στις μεθευρετικές διαδικασίες	53

2.1.2	Ιστορικά στοιχεία	54
2.1.3	Εφαρμογές	57
2.1.4	Ιδιότητες και κατηγοριοποίηση	58
2.1.5	Κριτήρια σύγκρισης μεθευρετικών	60
2.2	Η Αναζήτηση Μεταβαλλόμενης Γειτονιάς	60
2.2.1	Αναζήτηση Μεταβαλλόμενης Γειτονιάς	61
2.2.2	Μαθηματική Περιγραφή της VNS	62
2.2.3	Εκδοχές της VNS	62
2.2.4	Σχετικά με τη VNS και την GVNS	63
2.2.5	Κβαντικά εμπνευσμένη GVNS	64
2.2.6	Περιγραφή της κβαντικά εμπνευσμένης GVNS (qGVNS)	65
2.3	Η μη συμβατική (κβαντική) μεθευρετική διαδικασία QUBO	66
3	Το Πρόβλημα του Πλανόδιου Πωλητή	67
3.1	Ιστορικά στοιχεία	69
3.2	Μαθηματική Περιγραφή του TSP	72
3.3	Υπολογιστική Πολυπλοκότητα	73
3.4	Κατηγοριοποίηση του TSP	75
3.4.1	Εφαρμογές του TSP και συνδέσεις με άλλα προβλήματα	76
3.4.2	Εφαρμογές του mTSP και διασύνδεσή του με άλλα προβλήματα	77
3.5	Παραλλαγές του Προβλήματος του Πλανόδιου Πωλητή	78
3.6	TSP με χρονικά παράθυρα	79
3.7	Η κλασική μαθηματική περιγραφή του TSPTW	81
3.7.1	Ακριβείς Αλγόριθμοι για το TSPTW	83
3.7.2	Ευρετικές για το TSPTW	83
4	Η κβαντικά εμπνευσμένη μεθευρετική μέθοδος	85
4.1	Οι προτεινόμενες μεθευρετικές υλοποιήσεις	85
4.1.1	Δομές Γειτνίασης	86
4.1.2	Μέθοδοι διαταραχής	86
4.1.3	GVNS σχήματα που προκύπτουν από τις διαφορετικές μεθόδους διαταραχής	89
4.2	Υπολογιστική και συγκριτική μελέτη	90
4.2.1	Υπολογιστικό περιβάλλον και ορισμός πειραματικών παραμέτρων	90
4.2.2	Υπολογιστικά Αποτελέσματα	91
4.3	Στατιστική Ανάλυση των διαφορετικών μεθόδων	108
4.3.1	Στατιστική Ανάλυση στα ασύμμετρα TSP	108
4.3.2	Στατιστική Ανάλυση στα συμμετρικά TSP	110
4.3.3	Στατιστική Ανάλυση των Εθνικών TSP	111
4.4	Σύγκριση της μεθόδου μας με άλλες υλοποιήσεις	112

5	Το πρόβλημα περισυλλογής απορριμάτων	115
5.1	Το πρόβλημα περισυλλογής απορριμάτων ως TSP	115
5.1.1	Η προσέγγισή μας	116
5.1.2	Υπολογιστικά αποτελέσματα	118
5.1.3	Σύγκριση κβαντικά εμπνευσμένης GVNS και του πλησιέστερου γείτονα	118
5.1.4	Υπεροχή κβαντικά εμπνευσμένης έναντι κλασσικής GVNS	122
5.1.5	Γραφική Αναπαράσταση των αποτελεσμάτων	124
5.2	Το πρόβλημα περισυλλογής απορριμάτων ως TSPTW	129
5.2.1	Η προτεινόμενη μέθοδος	131
5.2.2	Διαταραχή, Δομές γειτνίασης	133
5.2.3	Υπολογιστικά αποτελέσματα	133
5.2.4	Η μελέτη περίπτωσης της Κέρκυρας	135
6	Κβαντικός υπολογισμός	139
6.1	Ο Κβαντικός υπολογισμός	139
6.1.1	Τεχνικοί κίνδυνοι	144
6.2	Quantum Annealing - Κβαντική Ανόπτηση	146
6.3	D-WAVE	147
6.4	Βιβλιογραφική Ανασκόπηση	149
7	Επίλογος	153
7.1	Μελλοντική Έρευνα	154
	Βιβλιογραφία-Αναφορές	154
	Γλωσσάρι Ξενικών Όρων	169



Περίληψη

Η εκπόνηση της παρούσας διατριβής αφορά το πεδίο της Βελτιστοποίησης Αλγορίθμων και πιο συγκεκριμένα με το πεδίο της συνδυαστικής βελτιστοποίησης. Στην ουσία πραγματεύεται τη διερεύνηση μεθευρετικών μεθόδων που βασίζονται σε συμβατικές ή μη συμβατικές, εμπνευσμένες από τον χβαντικό υπολογισμό τεχνικές με σκοπό την εφαρμογή τους σε αλγόριθμους βελτιστοποίησης για την επίλυση προβλημάτων από τον πραγματικό κόσμο. Πιο συγκεκριμένα, παρουσιάζουμε την χβαντικά εμπνευσμένη μεθευρετική μέθοδο qGVNS (quantum General Variable Neighbor-hood Search), την οποία χρησιμοποιούμε για να επιλύσουμε το πρόβλημα του πλανόδιου πωλητή (Travelling Salesman Problem, TSP εν συντομία) όπως επίσης και το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα (Travelling Salesman Problem with Time Windows, TSPTW εν συντομία). Ο τελικός σκοπός είναι η επίλυση ρεαλιστικών προβλημάτων από τον πραγματικό κόσμο που έχουν μοντελοποιηθεί σαν προβλήματα βελτιστοποίησης (TSP ή TSPTW) χρησιμοποιώντας τις μη συμβατικές μεθευρετικές μεθόδους που αναπτύξαμε.

Το πρόβλημα του πλανόδιου πωλητή χρησιμοποιείται ως σημείο αναφοράς σε πολλές μεθόδους βελτιστοποίησης και έχει πλήθος εφαρμογών σε πολλές και διαφορετικές περιοχές, όπως στην τεχνητή νοημοσύνη, την βιοπληροφορική, την βέλτιστη αναζήτηση διαδρομών και αλλού. Σε αυτή τη διατριβή διερευνούμε ιδιαίτερα την εκδοχή του TSP προβλήματος, που αποτελείται από κάποιους επιπρόσθετους περιορισμούς, όπως τα χρονικά παράθυρα (time windows). Η εκδοχή αυτή είναι γνωστή ως TSPTW (Travelling Salesman Problem with Time Windows).

Με βάση το πρόβλημα του πλανόδιου πωλητή (TSP) αλλά και το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα (TSPTW) επιλύσαμε το πρόβλημα περισυλλογής απορριμάτων από τον πραγματικό κόσμο, σε δύο διαφορετικές εκδοχές του. Το απλό πρόβλημα της περισυλλογής των απορριμάτων όπως και το πρόβλημα περισυλλογής των απορριμάτων με χρονικά παράθυρα.

Συνεχίζοντας, πραγματοποιήσαμε αναλυτική και διεξοδική συγκριτική μελέτη απόδοσης ανάμεσα στην νέα χβαντικά εμπνευσμένη μέθοδο που προτείνουμε και σε άλλες συμβατικές μεθόδους. Η παρούσα διατριβή ολοκληρώνεται με την παρουσίαση μερικών βασικών σημείων και βιβλιογραφίας σχετική με την χβαντική βελτιστοποίηση και την χβαντική ανόπτηση (quantum annealing).



Abstract

The composition of this dissertation concerns the field of Algorithm Optimization and more specifically the field of Combinatorial Optimization (CO). Fundamentally, it deals with the exploration of Metaheuristic methods based on conventional and unconventional methods, inspired by quantum computational techniques, with the aim of applying them to optimization algorithms to solve real-world problems. Specifically, we present the quantum-inspired qGVNS (quantum General Variable Neighborhood Search) method, which is used to solve the Travelling Salesman Problem (TSP in short) as well as its variant of Travelling Salesman Problem with Time Windows, (TSPTW in short). The ultimate goal is to solve real-world problems that have been modeled as optimization problems (TSP or TSPTW) using the unconventional metaheuristic methods we have developed.

The Travelling Salesman Problem is used as a reference point in many optimization methods and has many applications in many different areas, such as artificial intelligence, bio computing, routing optimization and elsewhere. In this dissertation we particularly explore the version of the TSP problem, which consists of some additional limitations, such as time windows. This version is known as TSPTW (Travelling Salesman Problem with Time Windows). Based on the Travelling Salesman Problem (TSP) but also the problem of the Travelling Salesman Problem with time windows (TSPTW) we solved the problem of garbage collection from the real world, in two different versions. The simple problem of garbage collection as well as the problem of waste collection with time windows.

We proceed to a detailed and thorough comparative and performance analysis between the new quantum-inspired method that we proposed and other conventional methods. This dissertation concludes with a presentation of some key points and bibliography related to quantum optimization and quantum annealing.



Πρόλογος - Ευχαριστίες

Η παρούσα διατριβή με θέμα «Μεθευρετικές μέθοδοι και τεχνικές εμπνευσμένες από τον χβαντικό υπολογισμό για αλγόριθμους βελτιστοποίησης» πραγματοποιήθηκε στο πλαίσιο των διδακτορικών σπουδών μου στο Τμήμα Πληροφορικής του Ιονίου Πανεπιστημίου. Μετά από τόσα χρόνια εκπόνησης της παρούσας διατριβής, νιώθω την υποχρέωση να ευχαριστήσω όλους αυτούς που συνέβαλλαν σε αυτή την προσπάθεια. Να ευχαριστήσω του φίλους, συναδέλφους και καθηγητές που συνεργάστηκα όλο αυτό τον καιρό.

Στην ολοκλήρωση αυτής της προσπάθειας συνέβαλε, πρωτίστως, ο Επιβλέπων Καθηγητής της εργασίας, κος Θοδωρής Ανδρόνικος, στον οποίο θα ήθελα να εκφράσω τις ειλικρινείς και θερμές ευχαριστίες μου για τη συνεχή καθοδήγηση και τις ουσιώδεις συμβουλές του. Η ευγνωμοσύνη μου για τον επιβλέπων είναι μεγάλη γιατί ήταν πάντα κοντά μου από τις πρώτες ερευνητικές προσπάθειες μέχρι και σε προσωπικά ζητήματα που έτυχε να αντιμετωπίσω. Θέλω ξανά να εκφράσω τις θερμές μου ευχαριστίες για την συνεργασία και να τονίσω ότι πάνω απ' όλα ο κος Ανδρόνικος είναι ένας υπέροχος άνθρωπος με "Α" κεφαλαίο. Δεν θα μπορούσα να παραλείψω να δώσω, τις θερμές μου ευχαριστίες προς τον κ. Σπύρο Σιούτα για την πολύτιμη βοήθεια που μου παρείχε κατά την εκπόνηση της διδακτορικής διατριβής, όπως και τον κ. Άγγελο Σιφαλέρα, για τις πολύτιμες γνώσεις που μου έδωσε στο πεδίο της συνδυαστικής βελτιστοποίησης.

Ευχαριστώ θερμά την οικογένειά μου για την αμέριστη υποστήριξη που μου παρείχε σε όλο το διάστημα των σπουδών μου αλλά και για τις αρχές που μου μεταλαμπάδευσε από την στιγμή που γεννήθηκα, μέχρι και σήμερα. Συγκεκριμένα, θέλω να ευχαριστήσω τον πατέρα μου, Αχιλλέα, που μου έμαθε να περιφρονώ αντί να εκδικούμαι και που το παράδειγμα του, έδειξε πως είναι να παλεύεις λυσσαλέα με πάθος και δίχως σταματημό, για να πετύχεις τους στόχους σου. Την μητέρα μου Μαρία, που μου έδωσε απλόχερα όλη της την αγάπη και όλη την σημασία του κόσμου. Νιώθω ευλογημένος, γιατί πάντα έδινε όλο της τον χρόνο για να με ακούει είτε είχα να πω κάτι, είτε όχι. Το παράδειγμα της είναι ένας φάρος στον αξιακό μου κώδικα που μου υπαγορεύει, πως πολλές φορές πρέπει να δίνεις δίχως -απαραίτητα- να παίρνεις πίσω, και την ευχαριστώ πολύ γι' αυτό. Τέλος, ευχαριστώ την αδερφή μου Ανδριάννα που ήταν δίπλα μου. Δεν θα μπορούσα να παραλείψω να ευχαριστήσω την Μαρία για την υπομονή που μου έδειξε. Ένας υπέροχος άνθρωπος με μία υπέροχη αύρα που μου έδινε δύναμη να συνεχίσω ακόμα και όταν οι δυνάμεις

μου οριακά με εγκατέλειπαν. Οι επιρροές της είναι αισθητές σε εμένα, σίγουρα είναι και στην παρούσα διατριβή, και την ευχαριστώ πολύ.

Επίσης, θα ήθελα να ευχαριστήσω όλους τους συνάδελφους ερευνητές που ήμασταν μαζί σε όλη αυτή τη διαδρομή και δίπλα τους έμαθα τόσα σημαντικά πράγματα. Τον Κωνσταντίνο Γιαννάκη, συνεργάτη μα και καλό φίλο πάνω απ' όλα, που με βοήθησε στα πρώτα μου ερευνητικά βήματα και πάντα μου έλυne και την παραμικρή απορία. Ίσως να τον ενόχλησα λίγο παραπάνω απ' το κανονικό, αλλά και πάλι τον ευχαριστώ πολύ! Την Καλλιόπη Κασταμπολίδου για τα εποικοδομητικά της σχόλια και την άψογη συνεργασία που είχαμε, όπου με την ευκαιρία της εύχομαι καλή συνέχεια με την εκπόνηση του διδακτορικού της και να το ολοκληρώσει σύντομα. Είμαι σίγουρος ότι τα ερευνητικά αποτελέσματα που θα παράγει θα είναι εκπληκτικά. Τον φίλο και συνεργάτη, Παναγιώτη Καραχώστα για τις ενδιαφέρουσες συζητήσεις που είχαμε σε όλη την διάρκεια του διδακτορικού.

Ένα μεγάλο ευχαριστώ σε όλο τον περίγυρό μου για το ανθρώπινο ενδιαφέρον που πάντα μου έδειχνε. Πιο συγκεκριμένα, τον φίλο Απόστολο που είναι ένας σπουδαίος άνθρωπος και είναι πάντα δίπλα μου. Τον Μάριο, τον Τάσο, τον Στέφανο και τον Μιχάλη που με ανέχονται τα τελευταία 25 χρόνια. Την Ελένη, τον Μιχάλη Σαββίδη και τον Ιωάννη Τσέπα που έχει μία όμορφη ψυχή και πάντα ήταν εκεί. Τον Λευτέρη Αρβανιτάκη για τις αμέτρητες συζητήσεις που είχαμε, είτε ήταν μέρα είτε νύχτα. Και όλους όσους μου διαφεύγουν και περάσαμε μαζί ευχάριστες στιγμές που μου έδιναν κουράγιο για να συνεχίσω.

Τέλος θα ήθελα να ευχαριστήσω τους συναδέλφους μου από όλες τις εταιρίες που συνεργάστηκα και ανέχτηκαν τον αλλόκοτο χαρακτήρα μου.



Δημοσιεύσεις

Δημοσιεύσεις σε επιστημονικά περιοδικά και συνέδρια σχετικές με την παρούσα διατριβή.

Δημοσιεύσεις σε επιστημονικά περιοδικά:

- **Papalitsas, C.,** Andronikos, T. (2019). Unconventional GVNS for Solving the Garbage Collection Problem with Time Windows. *Technologies*, 7(3), 61.
- **Papalitsas, C.,** Karakostas, P., Andronikos, T. (2019). A Performance Study of the Impact of Different Perturbation Methods on the Efficiency of GVNS for Solving TSP. *Applied System Innovation*, 2(4), 31.
- **Papalitsas, C.,** Karakostas, P., Andronikos, T., Sioutas, S., Giannakis, K. (2018). Combinatorial GVNS (general variable neighborhood search) optimization for dynamic garbage collection. *Algorithms*, 11(4), 38.

Δημοσιεύσεις σε πρακτικά σε διεθνή επιστημονικά συνέδρια με κριτές:

- **Papalitsas, C.,** Andronikos, T., Karakostas, P. (2018, October). Studying the Impact of Perturbation Methods on the Efficiency of GVNS for the ATSP. In *International Conference on Variable Neighborhood Search* (pp. 287-302). Springer, Cham.
- **Papalitsas, C.,** Karakostas, P., Kastampolidou, K. (2017). A quantum inspired GVNS: Some preliminary results. In *GeNeDis 2016* (pp. 281-289). Springer, Cham.
- **Papalitsas, C.,** Giannakis, K., Andronikos, T., Theotokis, D., Sifaleras, A. (2015, July). Initialization methods for the TSP with time windows using variable neighborhood search. In *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)* (pp. 1-6). IEEE

Ανακοινώσεις σε διεθνή επιστημονικά συνέδρια με κριτές:

- **Papalitsas, C.**, Karakostas, P., Giannakis, K., Sifaleras, A., Andronikos, T. (2017, June). Initialization methods for the TSP with Time Windows using qGVNS. In Proceedings of the 6th International Symposium on Operational Research, OR in the Digital Era—ICT Challenges, Thessaloniki, Greece (pp. 8-10).



Λέξεις Κλειδιά - Ορολογία

- **Βελτιστοποίηση (Optimization):**
Κλάδος των εφαρμοσμένων μαθηματικών, ο οποίος αναφέρεται στην αναζήτηση βέλτιστων παραμέτρων ενός - συνήθως περίπλοκου - συστήματος. Προβλήματα βελτιστοποίησης απαντώνται σε πολλά επιστημονικά πεδία όπως, παραδείγματος χάριν, στη φυσική, στη χημεία, στην οικονομία και αλλού. Στα μαθηματικά διατυπώνεται ένα πρόβλημα βελτιστοποίησης σαν πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης μιας συνάρτησης μίας μεταβλητής ή πολλών μεταβλητών.
- **Διακριτή Βελτιστοποίηση (Discrete Optimization):**
Κλάδος της βελτιστοποίησης στα εφαρμοσμένα μαθηματικά και την επιστήμη των υπολογιστών, όπου, σε αντίθεση με τη συνεχή βελτιστοποίηση, οι μεταβλητές που χρησιμοποιούνται στο μαθηματικό πρόγραμμα περιορίζονται να παίρνουν τιμές μόνο από ένα πεπερασμένο ή διακριτό σύνολο τιμών, όπως οι ακέραιοι. Δύο αξιοσημείωτοι κλάδοι της διακριτής βελτιστοποίησης είναι η συνδυαστική βελτιστοποίηση και ο ακέραιος προγραμματισμός.
- **Ευρετικές διαδικασίες (Heuristic Procedures):**
Στην επιστήμη των υπολογιστών, της τεχνητής νοημοσύνης και της μαθηματικής βελτιστοποίησης, μια ευρετική είναι μια τεχνική που σχεδιάστηκε για τη γρηγορότερη επίλυση ενός προβλήματος, όταν οι κλασικές μέθοδοι είναι πολύ αργές, ή για την εξεύρεση μιας προσεγγιστικής λύσης, όταν οι κλασικές μέθοδοι αδυνατούν να βρουν οποιαδήποτε ακριβή λύση. Κατά κάποιον τρόπο, μια ευρετική μπορεί να θεωρηθεί ως συντόμευση.
- **Μεθευρετικές διαδικασίες (Metaheuristic Procedures):**
Στην επιστήμη των υπολογιστών και τη βελτιστοποίηση, μια μεθευρετική είναι μια διαδικασία ή μια ευρετική πολύ υψηλού επιπέδου, σχεδιασμένη για να βρίσκει, να δημιουργεί, ή να επιλέγει μια διαδικασία ή ευρετική χαμηλότερου επιπέδου (αλγόριθμος μερικής αναζήτησης), η οποία μπορεί να παρέχει μια αρκετά καλή λύση σε ένα πρόβλημα βελτιστοποίησης, ιδίως με ελλιπείς ή ατελείς πληροφορίες ή με περιορισμένη υπολογιστική ικανότητα.

- Το πρόβλημα του πλανόδιου πωλητή (**TSP**):
Πρόβλημα της συνδυαστικής βελτιστοποίησης που απαντάει στο παρακάτω ερώτημα: "Λαμβάνοντας υπόψη ένα σύνολο πόλεων και τις αποστάσεις μεταξύ κάθε ζεύγους πόλεων, ποια είναι η συντομότερη διαδρομή που θα πρέπει να κάνει ο πλανόδιος πωλητής που επισκέπτεται κάθε πόλη και επιστρέφει στην πόλη προέλευσης;"
- Αναζήτηση Μεταβαλλόμενης Γειτονιάς (**Variable Neighborhood Search**):
Μεθευρετική μέθοδος που προτάθηκε το 1997 από τους Hansen, Mladenovic για την επίλυση ενός συνόλου προβλημάτων συνδυαστικής και ολικής βελτιστοποίησης, η οποία διερευνά μακρινές γειτονιές της τρέχουσας λύσης και μετακινείται από εκεί σε μία καινούρια γειτονιά, εάν και μόνο εάν έχει επέλθει κάποια βελτίωση.
- Η Κβαντικά Εμπνευσμένη Αναζήτηση Μεταβαλλόμενης Γειτονιάς (**qVNS**):
Μεθευρετική μέθοδος που προτάθηκε από τους Παπαλίτσα, Ανδρόνικο, για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης.
- Αναζήτηση Ταμπού (**Tabu Search**):
Μεθευρετική μέθοδος αναζήτησης που δημιουργήθηκε από τον Fred W. Glover το 1986 και επισημοποιήθηκε το 1989, η οποία χρησιμοποιεί τις μεθόδους τοπικής αναζήτησης που χρησιμοποιούνται στη μαθηματική βελτιστοποίηση. Οι τοπικές αναζητήσεις (στη γειτονιά) παίρνουν μία πιθανή λύση σε ένα πρόβλημα και ελέγχουν όλους τους άμεσους γείτονές της, δηλαδή λύσεις που είναι παρόμοιες εκτός από μία ή δύο μικρές λεπτομέρειες, με την ελπίδα να βρουν μία βελτιωμένη λύση.
- Κβαντικός Υπολογιστής (**Quantum Computer**):
Κβαντικός υπολογιστής ονομάζεται μία υπολογιστική συσκευή που εκμεταλλεύεται χαρακτηριστικές ιδιότητες της κβαντομηχανικής, όπως την αρχή της υπέρθεσης και της διεμπλοκής καταστάσεων, για να φέρει εις πέρας επεξεργασία δεδομένων και εκτέλεση υπολογισμών.
- Κβαντική Ανόπτηση (**Quantum Annealing**):
Η κβαντική ανόπτηση είναι μια μεθευρετική διαδικασία για την εύρεση του ολικού ελάχιστου μιας δεδομένης αντικειμενικής συνάρτησης σε ένα δεδομένο σύνολο υποψηφίων λύσεων, χρησιμοποιώντας μια διαδικασία που χρησιμοποιεί κβαντικές διακυμάνσεις.



1. Εισαγωγή

1.1 Ο τομέας του υπολογισμού

Ένας υπολογιστής μπορεί να περιγραφεί με πολύ απλά ως μια συσκευή που εκτελεί επαναληπτικούς υπολογισμούς αυτόματα. Ο παραπάνω ορισμός δεν μπορεί να περιγράψει με ακρίβεια και σαφήνεια την έννοια του υπολογιστή καθώς τον ορίζει αποκλειστικά και αυστηρά ως μια μαθηματική διαδικασία. Στην πραγματικότητα, ο υπολογισμός βασίζεται σε πολλές δραστηριότητες που δεν θεωρούνται συνήθως μαθηματικές. Το περπάτημα ενός ρομπότ σε ένα δωμάτιο, για παράδειγμα, απαιτεί πολλούς πολύπλοκους, υπολογισμούς. Οι υπολογιστές, επίσης, έχουν αποδειχθεί ικανοί να λύσουν μια πληθώρα προβλημάτων, από το πιο απλό όπως είναι η εξισορρόπηση ενός μπλοκ επιταγών μέχρι και πιο σύνθετα, όπως τα συστήματα καθοδήγησης για ρομπότ που διασχίζουν ένα δωμάτιο [189].

Ως εκ τούτου, πριν από την έλευση της πραγματικής δύναμης της πληροφορικής που μπορούσε να υλοποιηθεί, η άποψη για τον υπολογισμό κατά τα πρώτα χρόνια της μελέτης και έρευνας του έπρεπε να τροποποιηθεί. Οι εφευρέτες που ήθελαν να δημιουργήσουν έναν λειτουργικό υπολογιστή έπρεπε να μάθουν ότι αυτό που εφευρίσκουν δεν ήταν μόνο μία απλή συσκευή. Για παράδειγμα, έπρεπε να μάθουν ότι δεν ήταν απαραίτητο να εφεύρουν έναν νέο υπολογιστή για κάθε νέο υπολογισμό και ότι ένας υπολογιστής θα μπορούσε να σχεδιαστεί για να λύσει πολλά προβλήματα, ακόμη και προβλήματα που δεν είχαν ακόμη φανταστεί όταν δημιουργήθηκε ο υπολογιστής. Με άλλα λόγια, ήταν αναγκαία η εφεύρεση του προγραμματισμού. Έπρεπε να λύσουν όλα τα προβλήματα της ανάπτυξης μιας τέτοιας συσκευής δηλαδή, της υλοποίησης, του σχεδιασμού και της πραγματικής οικοδόμησης του συγκεκριμένου συστήματος. Η ιστορία της επίλυσης αυτών των προβλημάτων είναι η ιστορία του υπολογιστή [49].

Οι πρώτοι υπολογιστές χρησιμοποιήθηκαν κυρίως για αριθμητικούς υπολογισμούς. Ωστόσο, καθώς όλες οι πληροφορίες μπορούν να κωδικοποιηθούν αριθμητικά, οι άνθρωποι σύντομα κατάλαβαν ότι οι υπολογιστές είναι ικανοί για επεξεργασία πληροφοριών γενικού σκοπού. Η ικανότητά τους να χειρίζονται μεγάλες ποσότητες δεδομένων επέκτεινε το εύρος και την ακρίβεια της πρόβλεψης καιρικών συνθηκών. Η ταχύτητά τους, τους επέτρεψε να λαμβάνουν αποφάσεις σχετικά με τη δρομολόγηση τηλεφωνικών συνδέσεων μέσω ενός δικτύου και να ελέγχουν μηχανι-

κά συστήματα όπως αυτοκίνητα, πυρηνικούς αντιδραστήρες και ρομποτικά χειρουργικά εργαλεία. Είναι επίσης αρκετά φθηνά για να ενσωματωθούν στις καθημερινές συσκευές και να καταστήσουν αρκετές συσκευές ως «έξυπνες». Οι υπολογιστές μας επέτρεψαν να θέσουμε και να απαντήσουμε σε ερωτήσεις που δεν μπορούσαν να επιδιωχθούν πριν. Αυτές οι ερωτήσεις μπορεί να έχουν να κάνουν με τις ακολουθίες DNA στα γονίδια, τα πρότυπα δραστηριότητας σε μια καταναλωτική αγορά, ή όλες τις χρήσεις μιας λέξης σε κείμενα που έχουν αποθηκευτεί σε μια βάση δεδομένων. Ολοένα και περισσότερο, οι υπολογιστές μπορούν επίσης να μαθαίνουν και να προσαρμόζονται καθώς λειτουργούν [120].

Οι υπολογιστές έχουν επίσης περιορισμούς, μερικοί από τους οποίους είναι θεωρητικοί. Για παράδειγμα, υπάρχουν μη αποκρίσιμες προτάσεις των οποίων η αλήθεια δεν μπορεί να προσδιοριστεί μέσα σε ένα δεδομένο σύνολο κανόνων, όπως η λογική δομή ενός υπολογιστή. Επειδή δεν μπορεί να υπάρχει καθολική αλγοριθμική μέθοδος για τον εντοπισμό τέτοιων προτάσεων, ένας υπολογιστής που ζήτησε να αποκτήσει την αλήθεια μιας τέτοιας πρότασης θα συνεχίσει (εκτός αν διακοπεί βίαια) επ' αόριστον — μια κατάσταση γνωστή ως «πρόβλημα διακοπής». Άλλοι περιορισμοί αντικατοπτρίζουν την τρέχουσα τεχνολογία. Τα ανθρώπινα μυαλά είναι ειδικευμένα στην αναγνώριση χωρικών μοτίβων - που εύκολα διακρίνουν μεταξύ των ανθρώπινων προσώπων, για παράδειγμα - αλλά αυτό είναι ένα δύσκολο έργο για τους υπολογιστές, που πρέπει να επεξεργάζονται πληροφορίες διαδοχικά, αντί να πιάνουν τις λεπτομέρειες συνολικά με μια ματιά. Μια άλλη προβληματική περιοχή για τους υπολογιστές περιλαμβάνει αλληλεπιδράσεις με τη φυσική γλώσσα. Επειδή τόσες πολλές κοινές γνώσεις και συναφείς πληροφορίες υπάρχουν στην τακτική ανθρώπινη επικοινωνία, οι ερευνητές δεν έχουν ακόμη λύσει το πρόβλημα της παροχής σχετικών πληροφοριών σε προγράμματα γενικής χρήσης της φυσικής γλώσσας [117].

1.1.1 Ιστορική αναδρομή

Η πρωιμότερη γνωστή συσκευή υπολογισμού είναι μάλλον ο άβακας. Χρονολογείται τουλάχιστον στο 1100 π.χ. και εξακολουθεί να χρησιμοποιείται σήμερα. Τώρα, όπως και τότε, συνήθως αποτελείται από ένα ορθογώνιο πλαίσιο με λεπτές παράλληλες ράβδους κρεμασμένες με χάντρες. Πολύ πριν υιοθετηθεί οποιαδήποτε συστηματική σημειογραφία για τη γραφή των αριθμών, ο άβακας ανέθεσε διαφορετικές μονάδες, ή βάρη, σε κάθε ράβδο. Το σχέδιο αυτό επέτρεψε σε ένα ευρύ φάσμα αριθμών να αντιπροσωπεύεται από λίγες μόνο χάντρες και μαζί με την εφεύρεση του μηδενός στην Ινδία, μπορεί να ενέπνευσε την εφεύρεση του ινδουιστικού - αραβικού αριθμητικού συστήματος. Σε κάθε περίπτωση, οι χάντρες του άβακα μπορούν να χειραγωγούνται εύκολα για την εκτέλεση των κοινών αριθμητικών λειτουργιών — πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση — που είναι χρήσιμες για τις εμπορικές συναλλαγές και τα λογιστικά. Ο άβακας είναι μια ψηφιακή συσκευή. Δηλαδή, συμβολίζει διακριτά τις αξίες. Μια χάντρα είναι είτε σε μια προκαθορισμένη θέση είτε σε μια άλλη, που αντιπροσωπεύει κατηγορηματικά, ας πούμε, το ένα ή το μηδέν [92].

Ο υπολογισμός των συσκευών πήρε διαφορετική τροπή όταν ο Τζον Νάπιερ (John Napier), ένας Σκοτσέζος μαθηματικός, δημοσίευσε την ανακάλυψη των λογαρίθμων το 1614. Η πρόσθεση των ψηφίων 2:10 είναι πολύ απλούστερη από το να τα πολλαπλασιάζουν μαζί και ο μετασχηματισμός ενός προβλήματος πολλαπλασιασμού σε ένα πρόβλημα πρόσθεσης είναι ακριβώς αυτό που κάνουν οι λογάριθμοι. Αυτή η απλούστευση είναι δυνατή λόγω της ακόλουθης λογαριθμικής ιδιότητας, όπου ο λογάριθμος του πολλαπλασιασμού δύο αριθμών ισούται με το άθροισμα των λογαρίθμων των αριθμών. Το 1624, συσκευές με 14 σημαντικά ψηφία ήταν διαθέσιμες για λογαρίθμους των αριθμών από το 1 έως το 20.000 και οι επιστήμονες γρήγορα υιοθέτησαν το νέο

εργαλείο εξοικονόμησης εργασίας για κουραστικούς αστρονομικούς υπολογισμούς [87].

Κατά τη δεύτερη δεκαετία του 19ου αιώνα, μία πληθώρα ιδεών που ήταν απαραίτητες για την εφεύρεση του υπολογιστή ήταν στα σκαριά. Πρώτον, εκτιμήθηκαν τα πιθανά οφέλη για την επιστήμη και τη βιομηχανία αν αυτοματοποιήσουν τους υπολογισμούς ρουτίνας. Συγκεκριμένες μέθοδοι για να γίνει ο αυτόματος υπολογισμός πιο πρακτικός, όπως ο πολλαπλασιασμός με την πρόσθεση λογαρίθμων ή με την επανάληψη της πρόσθεσης, είχαν ανακαλυφθεί, και η εμπειρία με τόσο αναλογικές και ψηφιακές συσκευές είχε δείξει μερικά από τα οφέλη της κάθε προσέγγισης. Η συσκευή Ζακάρ είχε δείξει τα οφέλη της κατεύθυνσης μιας συσκευής πολλαπλής χρήσης μέσω κωδικοποιημένων οδηγιών, και είχε δείξει πώς διάτρητες κάρτες θα μπορούσαν να χρησιμοποιηθούν για την τροποποίηση αυτών των οδηγιών γρήγορα και ευέλικτα [91].

Ο Κάρολος Μπάμπατζ (Charles Babbage) ήταν Άγγλος μαθηματικός και εφευρέτης. Αναμόρφωσε το βρετανικό ταχυδρομικό σύστημα και ήταν πρωτοπόρος στους τομείς της επιχειρησιακής έρευνας και της αναλογιστικής επιστήμης. Ως ιδρυτικό μέλος της Βασιλικής Αστρονομικής Εταιρείας, ο Μπάμπατζ είχε δει τη σαφή ανάγκη να σχεδιάσει και να δημιουργήσει μια μηχανική συσκευή που θα μπορούσε να αυτοματοποιήσει κοπιαστικούς αστρονομικούς υπολογισμούς. Ξεκίνησε γράφοντας ένα γράμμα το 1822 στον Σερ Χάμφρι Ντέιβι (Sir Humphry Davy), Πρόεδρο της Βασιλικής εταιρείας, σχετικά με τη δυνατότητα αυτοματοποίησης της κατασκευής μαθηματικών πινάκων — συγκεκριμένα, λογαριθμικών πινάκων για χρήση στην πλοήγηση. Στη συνέχεια έγραψε ένα έγγραφο, σχετικά με τις θεωρητικές αρχές του μηχανήματος για τον υπολογισμό των πινάκων, το οποίο παρουσίασε εκείνο το έτος (Κέρδισε το χρυσό μετάλλιο της Βασιλικής εταιρείας το 1823). Οι πίνακες που χρησιμοποιήθηκαν συχνά περιείχαν σφάλματα, τα οποία θα μπορούσαν να είναι θέμα ζωής και θανάτου για τους ναυτικούς στη θάλασσα και ο Μπάμπατζ ισχυρίστηκε ότι, αυτοματοποιώντας την παραγωγή των πινάκων, θα μπορούσε να διαβεβαιώσει την ακρίβειά τους. Έχοντας αποκτήσει υποστήριξη στην κοινωνία για τη διαφορική μηχανή, όπως την αποκάλεσε, ο Μπάμπατζ μετά στράφηκε στη βρετανική κυβέρνηση για να χρηματοδοτήσει την ανάπτυξη της, αποκτώντας μία από τις πρώτες κυβερνητικές επιχορηγήσεις στον κόσμο για την έρευνα και την τεχνολογική ανάπτυξη [10].

Ο Μπάμπατζ προσέγγισε το έργο πολύ σοβαρά: προσέλαβε έναν μηχανουργό, έστησε ένα πυρίμαχο εργαστήριο, και έχτισε ένα ανθεκτικό περιβάλλον για τον έλεγχο της συσκευής. Μέχρι τότε οι υπολογισμοί σπάνια πραγματοποιήθηκαν με περισσότερα από 6 ψηφία. Ο Μπάμπατζ σχεδίαζε να παράγει 20 ή 30 ψηφία. Η διαφορική μηχανή (Difference engine) ήταν μια ψηφιακή συσκευή: λειτουργούσε με διακριτά ψηφία αντί για συνεχείς ποσότητες, και τα ψηφία ήταν δεκαδικά (0–9), εκπροσωπούμενα από θέσεις σε οδοντωτές ρόδες και όχι τα δυαδικά ψηφία που προτίμησε ο Λάιμπνιτς (Gottfried Wilhelm Leibniz), αλλά δεν χρησιμοποίησε. Η συσκευή ήταν κάτι περισσότερο από μία απλή αριθμομηχανή, ωστόσο. Πήγε πολύ μακρύτερα από τις αριθμομηχανές και με άλλους τρόπους. Όπως και οι σύγχρονοι υπολογιστές, ο μηχανισμός είχε αποθηκευτικό χώρο — δηλαδή έναν τόπο όπου τα δεδομένα μπορούσαν να κρατούνται προσωρινά για μεταγενέστερη επεξεργασία — και σχεδιάστηκε για να σφραγίζουν την παραγωγή σε μαλακό μέταλλο, το οποίο αργότερα θα μπορούσε να χρησιμοποιηθεί για την παραγωγή μιας πινακίδας εκτύπωσης.

Καθώς η τεχνολογία για την υλοποίηση ενός υπολογιστή είχε προετοιμαστεί από τις εταιρείες μηχανημάτων επιχειρήσεων στις αρχές του 20ου αιώνα, τα θεωρητικά θεμέλια τέθηκαν στην ακαδημαϊκή κοινότητα. Κατά τη δεκαετία του 1930, δύο σημαντικά έργα της έρευνας που σχετίζονται με τον υπολογιστή, πραγματοποιήθηκαν στις Ηνωμένες Πολιτείες σε δύο πανεπιστήμια στο Κέμπριτζ της Μασαχουσέτης. Το ένα έργο παρήγαγε το διαφορικό αναλυτή και το άλλο έργο μια σειρά από συσκευές που τελειώνουν με το Μαρκ IV του Χάρβαρντ [29].

Το 1930 ένας μηχανικός ονόματι Βάνεβαρ Μπους (Vannevar Bush) στο Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT) ανέπτυξε τον πρώτο σύγχρονο αναλογικό υπολογιστή. Ο διαφορικός αναλυτής, όπως τον αποκάλεσε, ήταν μια αναλογική αριθμομηχανή που θα μπορούσε να χρησιμοποιηθεί για να λύσει ορισμένες κατηγορίες διαφορικών εξισώσεων, ένα είδος προβλήματος που είναι συνηθισμένο στη φυσική και τις εφαρμογές μηχανικής που είναι συχνά πολύ κουραστικό να λυθεί. Οι μεταβλητές αντιπροσωπεύονται από την κίνηση του άξονα και η πρόσθεση και ο πολλαπλασιασμός πραγματοποιήθηκαν με την προσθήκη των τιμών σε ένα σύνολο ταχυτήτων. Η ενσωμάτωση πραγματοποιήθηκε με τη βοήθεια ενός τροχού που περιστρέφεται σε μια μεταβλητή ακτίνα σε έναν κυκλικό πίνακα. Οι μεμονωμένοι μηχανικοί ενοποιητές στη συνέχεια διασυνδέονται για να λύσουν ένα σύνολο διαφορικών εξισώσεων [143].

Οι υπολογιστές που δημιουργήθηκαν κατά τη διάρκεια του πολέμου δημιουργήθηκαν κάτω από ασυνήθιστους περιορισμούς. Το βρετανικό έργο επικεντρώθηκε σε μεγάλο βαθμό στο σπάσιμο κώδικα, το αμερικανικό έργο στην υπολογιστική τροχιά βλημάτων και υπολογισμών για την ατομική βόμβα. Οι υπολογιστές κατασκευάστηκαν ως συσκευές ειδικού σκοπού, αν και συχνά ενσωμάτωσαν περισσότερες δυνατότητες υπολογιστικής γενικής χρήσης από ό,τι οι προδιαγραφές τους. Οι σωλήνες κενού σε αυτά τα μηχανήματα δεν ήταν εντελώς αξιόπιστοι, αλλά χωρίς κινούμενα μέρη ήταν πιο αξιόπιστοι από τους ηλεκτρομηχανικούς διακόπτες που αντικατέστησαν και ήταν πολύ πιο γρήγοροι. Η αξιοπιστία ήταν ένα ζήτημα, δεδομένου ότι χρησιμοποιούσαν περίπου 1.500 σωλήνες και ο ENIAC 18.000. Αλλά ο ENIAC ήταν, λόγω της ηλεκτρονικής υλοποίησης, 1.000 φορές ταχύτερος. Μια τέτοια ταχύτητα σήμαινε ότι η μηχανή θα μπορούσε να εκτελέσει υπολογισμούς που ήταν πέρα από την ανθρώπινη ικανότητα.

Αν και οι σωλήνες ήταν ένας προάγγελος για την ηλεκτρομηχανική υλοποίηση του Έικεν ή του μηχανολογικού μοντέλου του Μπάμπατζ, η βασική αρχιτεκτονική των μηχανών (δηλαδή, οι λειτουργίες που ήταν σε θέση να εκτελέσουν) δεν ήταν πολύ προχωρημένη πέρα από τον Μπάμπατζ. Στην πραγματικότητα, το αρχικό όνομα για τον ENIAC ήταν ηλεκτρονικός διαφορικός αναλυτής και δημιουργήθηκε για να εκτελέσει υπολογισμούς όπως η μηχανή διαφοράς του Μπάμπατζ.

Μετά τον πόλεμο, οι προσπάθειες επικεντρώθηκαν στην εκπλήρωση της ιδέας μιας υπολογιστικής συσκευής γενικής χρήσης. Το 1945, πριν καν τελειώσει ο ENIAC, ο προγραμματισμός ξεκίνησε από τη σχολή Μουρ για τον διάδοχο του ENIAC, τον ηλεκτρονικό αυτόματο υπολογιστή διακριτής μεταβλητής ή αλλιώς EDVAC. Ο ENIAC παρεμποδίστηκε, όπως έκαναν όλοι οι προηγούμενοι ηλεκτρονικοί υπολογιστές, από την ανάγκη χρήσης ενός σωλήνα κενού για την αποθήκευση κάθε bit (δυαδικού ψηφίου). Ο εφικτός αριθμός σωλήνων κενού σε έναν υπολογιστή θέτει επίσης ένα πρακτικό όριο στην αποθηκευτική ικανότητα — πέρα από ένα ορισμένο σημείο, οι σωλήνες κενού δεσμεύονται να καούν όσο πιο γρήγορα μπορούν να αλλάζουν. Για τον EDVAC, ο Έκερτ (J. Presper Eckert) είχε μια νέα ιδέα για αποθήκευση [50].

Το 1880 οι Γάλλοι φυσικοί Πιέρ και Ζακ Κιουρί (Pier and Zack Curie) ανακάλυψαν ότι η εφαρμογή ηλεκτρικού ρεύματος σε κρύσταλλο χαλαζία θα παρήγαγε μια χαρακτηριστική δόνηση και αντίστροφα. Κατά τη διάρκεια της δεκαετίας του 1930 στα εργαστήρια Bell, ο Γουίλιαμ Σόκλι (William Bradford Shockley), αργότερα εφευρέτης του τρανζίστορ, είχε επιδείξει μια συσκευή — ένα σωλήνα, που ονομάζεται γραμμή καθυστέρησης, που περιέχει νερό και αιθυλική γλυκόλη — για να επιφέρει προβλέψιμη καθυστέρηση στη μετάδοση πληροφοριών. Ο Έκερτ είχε ήδη χτίσει και πειραματιστεί το 1943 με μια τέτοια γραμμή καθυστέρησης (χρησιμοποιώντας υδράργυρο) σε συνδυασμό με την έρευνα ραντάρ, και κάποια στιγμή στο 1944 προώθησε τη νέα ιδέα της τοποθέτησης ενός κρυστάλλου χαλαζία σε κάθε άκρο της γραμμής καθυστέρησης του υδραργύρου, προκειμένου να διατηρηθούν και να τροποποιηθούν τα μοτίβα. Στην πραγματικότητα, εφηύρε μια

νέα συσκευή αποθήκευσης. Ενώ ο ENIAC απαιτούσε ένα σωληνάριο ανά bit, ο EDVAC μπορούσε να χρησιμοποιήσει μια γραμμή καθυστέρησης και 10 σωλήνες κενού για την αποθήκευση 1.000 bits. Πριν από την εφεύρεση της μαγνητικής μνήμης πυρήνα και του τρανζίστορ, η γραμμή καθυστέρησης του υδραργύρου ήταν καθοριστικής σημασίας για την αύξηση της αποθήκευσης και της αξιοπιστίας του υπολογιστή [93].

Ενώ το υψηλό κόστος των πόρων του υπολογιστή έδωσε μια προιμοδότηση σε γρήγορα χειροποίητα προγράμματα γλώσσας - μηχανής, ένα άτομο εργάστηκε ακούραστα για να προωθήσει γλώσσες προγραμματισμού υψηλού επιπέδου και τους συναφείς μεταγλωττιστές τους. Η Γκρέις Μάρεϊ Χόπερ (Grace Murray Hopper) δίδασκε μαθηματικά στο κολέγιο Βάσσαρ, στο Πουκίψι της Νέας Υόρκης (Vassar College, Poughkeepsie, New York), από το 1931 έως το 1943 πριν ενταχθεί στο Ναυτικό των ΗΠΑ. Το 1944, της ανατέθηκε η διαχείριση του προγράμματος υπολογισμού πυρομαχικών στο Πανεπιστήμιο του Χάρβαρντ, όπου προγραμμάτισε το MARK I υπό την κατεύθυνση του Χάουαρντ Έικεν (Howard H. Aiken). Μετά τον Δεύτερο παγκόσμιο πόλεμο δημιούργησε λογισμικό μεταγλωττιστή για τα συστήματα BINAC και UNIVAC. Καθ' όλη τη δεκαετία του 1950 η Χόπερ αγωνίστηκε θερμά για τις υψηλού επιπέδου γλώσσες σε όλες τις Ηνωμένες Πολιτείες, και μέσω των δημόσιων παρουσιών της βοήθησε να αφαιρεθεί την αντίσταση στην ιδέα. Μια τέτοια προτροπή βρήκε ένα δεκτικό ακροατήριο στην IBM, όπου η διοίκηση ήθελε να προσθέσει υπολογιστές στην επιτυχημένη γραμμή των επιχειρηματικών μηχανών της εταιρείας [93].

Στις αρχές της δεκαετίας του 1950 ο Τζον Μπάκους (John Backus) έπεισε τους διευθυντές του στην IBM να τον αφήσουν να δημιουργήσει μια ομάδα για να σχεδιάσουν μια γλώσσα και να γράψουν ένα μεταγλωττιστή για αυτό. Είχε ένα μηχανήμα στο μυαλό του: το IBM 704, το οποίο είχε ενσωματωμένες μαθηματικές πράξεις κινητής υποδιαστολής. Το 704 που χρησιμοποιήθηκε για την αναπαράσταση του κυμαινόμενου σημείου το έκανε ιδιαίτερα χρήσιμο για την επιστημονική εργασία και ο Μπάκους πίστευε ότι μια επιστημονικά προσανατολισμένη γλώσσα προγραμματισμού θα έκανε τη μηχανή ακόμα πιο ελκυστική. Παρ' όλα αυτά, κατάλαβε την αντίσταση σε οτιδήποτε καθυστέρουσε μια μηχανή και ανέλαβε την παραγωγή μιας γλώσσας και ενός μεταγλωττιστή που θα παρήγαγε κώδικα που θα έτρεχε σχεδόν τόσο γρήγορα όσο η γλώσσα της μηχανής και ταυτόχρονα θα έκανε τη διαδικασία σύνταξης του προγράμματος πολύ ευκολότερη [11].

Από το 1954, ο Μπάκους και μια ομάδα προγραμματιστών είχαν σχεδιάσει τη γλώσσα, την οποία αποκαλούσαν FORTRAN (μετάφραση τύπου). Προγράμματα γραμμένα σε FORTRAN φάνινονταν πολύ περισσότερο σαν μαθηματικά από ό,τι οδηγίες μηχανής:

Algorithm 1 παράδειγμα σε FORTRAN 1

```
DO 10, J=1,11;
I = 11 - J;
Y = F(A(I + 1));
IF (400 - Y) 4,8,8;
4 PRINT 5,1;
5 FORMAT (I10, 10H TOO LARGE);
```

Ο μεταγλωττιστής γράφτηκε και η γλώσσα κυκλοφόρησε με ένα επαγγελματικό εγχειρίδιο γραμματοσειρών (ένα πρώτο για τις γλώσσες προγραμματισμού) το 1957 [11]. Η FORTRAN

έκανε ένα άλλο βήμα προς τον πιο προσιτό προγραμματισμό, επιτρέποντας σχόλια στα προγράμματα. Η δυνατότητα εισαγωγής σχολιασμών, που επισημαίνονται για να αγνοηθεί από το πρόγραμμα μετάφρασης αλλά αναγνώσιμη από έναν άνθρωπο, σήμαινε ότι ένα καλά σχολιασμένο πρόγραμμα θα μπορούσε να διαβαστεί με μια ορισμένη έννοια από άτομα χωρίς καθόλου γνώσεις προγραμματισμού. Για πρώτη φορά ένας μη προγραμματιστής θα μπορούσε να πάρει μια ιδέα για το τι έκανε ένα πρόγραμμα – ή τουλάχιστον τι σκόπευε να κάνει – διαβάζοντας (μέρος του) τον κώδικα. Ήταν ένα προφανές αλλά ισχυρό βήμα στο άνοιγμα των υπολογιστών σε ένα ευρύτερο κοινό. Η FORTRAN εξακολούθησε να εξελίσσεται και διατηρεί μια μεγάλη βάση χρηστών στον ακαδημαϊκό χώρο και μεταξύ των επιστημόνων [6].

Σχετικά με την εποχή που ο Μπάκους και η ομάδα του εφεύραν τη FORTRAN, η ομάδα του Χόπερ στην UNIVAC κυκλοφόρησε μια γλώσσα που μοιάζει με το TRAN για τους υπολογιστές UNIVAC. Ήταν βραδύτερο από τη FORTRAN και όχι ιδιαίτερα επιτυχημένο. Μια άλλη γλώσσα που αναπτύχθηκε στο εργαστήριο του Χόπερ την ίδια στιγμή είχε περισσότερη επιρροή, η οποία χρησιμοποιούσε ένα αγγλικό συντακτικό και λεξιλόγιο:

Algorithm 2 παράδειγμα σε FORTRAN 2

```
1 COMPARE PART-NUMBER (A) TO PART-NUMBER (B);
IF GREATER GO TO OPERATION 13;
IF EQUAL GO TO OPERATION 4;
OTHERWISE GO TO OPERATION 2.
```

Στη συνέχεια αναπτύχθηκε από την ομάδα της Χόπερ η COBOL (κοινή επιχειρηματική γλώσσα) το 1959. Η COBOL ήταν ρητά μια γλώσσα επιχειρηματικού προγραμματισμού με ένα πολύ λεπτομερές αγγλικό στυλ. Έγινε κεντρικό σημείο για την ευρεία αποδοχή των υπολογιστών από τις επιχειρήσεις μετά το 1959. Ο Γουίλιαμ Σόκλει (William Bradford Shockley), εφευρέτης του τρανζίστορ, ξεκίνησε τα εργαστήρια Σόκλει το 1955 στην γενέτειρά του. Το 1957 οι οκτώ κορυφαίοι ερευνητές του έφυγαν για να σχηματίσουν την εταιρεία Φερτσαϊλντ (Fairchild). Μαζί με την Χιούλετ-Πάκαρντ (Hewlett Packard), μια άλλη εταιρεία, η Φερτσαϊλντ ήταν ο σπόρος αυτού που θα γινόταν γνωστό ως Σίλικον Βάλει (Silicon Valley). Ιστορικά, η Φέρτσαϊλντ θα αξίζει πάντα να αναγνωριστεί ως μία από τις σημαντικότερες εταιρείες, έχοντας υπηρετήσει ως εκπαιδευτικό έδαφος για τους περισσότερους επιχειρηματίες που πήγαν να ξεκινήσουν τις δικές τους εταιρείες υπολογιστών στη δεκαετία του 1960 και στις αρχές της δεκαετίας του 1970. Από τα μέσα της δεκαετίας του 1960 στην αρχή της δεκαετίας του '70, η επιχείρηση ημιαγωγών και η Texas Instruments ήταν οι κορυφαίοι κατασκευαστές ολοκληρωμένων κυκλωμάτων (ICs) και αύξαναν συνεχώς τον αριθμό των ηλεκτρονικών εξαρτημάτων που είναι ενσωματωμένα σε ένα ενιαίο τσιπ. Καθώς ο αριθμός των συστατικών κλιμακώθηκε σε χιλιάδες, αυτές οι μάρκες άρχισαν να αναφέρονται ως μάρκες ενσωμάτωσης μεγάλης κλίμακας και οι υπολογιστές που τις χρησιμοποιούσαν μερικές φορές ονομάζονται υπολογιστές τέταρτης γενιάς. Η εφεύρεση του μικροεπεξεργαστή ήταν το αποκορύφωμα αυτής της τάσης. Αν και οι υπολογιστές ήταν ακόμα σπάνιοι και συχνά θεωρούνταν απειλή για την απασχόληση, οι αριθμομηχανές ήταν συνηθισμένες και αποδεκτές στα γραφεία. Με την πρόοδο στην τεχνολογία ημιαγωγών, μια αγορά αναδυόταν για εξελιγμένους υπολογιστικούς ηλεκτρονικούς υπολογιστές. Ήταν, στην πραγματικότητα, ένα έργο αριθμομηχανής που μετατράπηκε σε ένα ορόσημο στην ιστορία της τεχνολογίας των υπολογιστών [16].

Το 1969, η Μπούτκομ (Bootcom), μια ιαπωνική εταιρεία, ανέθεσε στην Intel Corporation να

φτιάξει μάρκες για μια σειρά αριθμομηχανών που σκόπευε να πουλήσει η Μπούτκομ. Η Intel ήταν μία από τις διάφορες εταιρείες που προέκυψαν στο Σίλικον Βάλει. Ο Πρόεδρος της Intel, Ρόμπερτ Νόις (Robert Noyce), ενώ ήταν στην Φέρτσαϊλντ, είχε εφεύρει ολοκληρωμένα κυκλώματα, μια διαδικασία στην οποία η καλωδίωση ήταν απευθείας ενσωματωμένη στο πυρίτιο μαζί με τα ηλεκτρονικά εξαρτήματα στο στάδιο της παρασκευής. Η Intel είχε σχεδιάσει να εστιάσει την επιχείρησή της σε τσιπ μνήμης, αλλά το αίτημα της Μπούτκομ για μάρκες για ένα κομπιουτεράκι αποδείχτηκε πολύτιμος αντιπερισπασμός. Ενώ οι εξειδικευμένες μάρκες ήταν αποτελεσματικές στο δεδομένο έργο τους, η μικρή αγορά τις καθιστούσε ακριβές.

Τρεις μηχανικοί της Intel — Φεντερίκο Φάγγκιν (Federico Faggin), Μαρτιάν («Ted») Χοφ (Marcian Hoff) και Σταν Μέιζορ (Stanley Mazor) — εξέτασαν το αίτημα της ιαπωνικής επιχείρησης και πρότειναν έναν πιο ευέλικτο σχεδιασμό. Ο Χοφ είχε εμπειρία με τους μίνι υπολογιστές, που θα μπορούσαν να κάνουν τα πάντα. Επαναστάτησε στην κατασκευή μιας συσκευής ειδικού σκοπού όταν υπήρχε η τεχνολογία για να χτίσει ένα γενικό σκοπό. Η συσκευή γενικού σκοπού που είχε κατά νου, ωστόσο, θα ήταν πολύ σαν ένας υπολογιστής και τότε οι υπολογιστές τρώμαζαν τους ανθρώπους, ενώ οι αριθμομηχανές ήταν ευρέως αποδεκτές. Επιπλέον, υπήρχε μια σαφής και μεγάλη αγορά για αριθμομηχανές και μια περιορισμένη για τους υπολογιστές. Παρ' όλα αυτά, ο Χοφ επικράτησε, και η Intel πρότεινε ένα σχέδιο που ήταν λειτουργικά πολύ παρόμοιο με έναν μικροϋπολογιστή (αν και όχι σε μέγεθος, δύναμη, συσκευές με δυνατότητα προσάρτησης, όπως εκτυπωτές, ή πολλούς άλλους πρακτικούς τρόπους) [51].

Εκτός από την εκτέλεση των λειτουργιών εισόδου/εξόδου που έχουν πραγματοποιήσει οι περισσότερες ICs, ο σχεδιασμός θα αποτελούσε τις οδηγίες για το IC και θα βοηθούσε στον έλεγχο, την αποστολή και τη λήψη σημάτων από άλλες μάρκες και συσκευές. Ένα σύνολο οδηγιών αποθηκεύτηκε στη μνήμη και το τσιπ μπορούσε να τα διαβάσει και να ανταποκριθεί σε αυτά. Έτσι, η συσκευή θα έκανε όχι μόνο ό,τι ήθελε η Μπούτκομ, αλλά θα έκανε πολλά περισσότερα: ήταν η ουσία ενός υπολογιστή γενικού σκοπού. Υπήρχε μικρή προφανής ζήτηση για μια τέτοια συσκευή, αλλά η ομάδα της Intel, κατανοώντας τα μειονεκτήματα των ICs ειδικού σκοπού, αισθάνθηκε ότι ήταν μια οικονομική συσκευή που, κατά κάποιο τρόπο, θα έβρισκε μια αγορά. Στην αρχή η Μπούτκομ δεν ενδιαφερόταν, αλλά η Intel αποφάσισε να προχωρήσει με το σχέδιο ούτως ή άλλως και η ιαπωνική εταιρεία τελικά την αποδέχθηκε. Η Intel ονόμασε το chip 4004, το οποίο αναφερόταν στον αριθμό των χαρακτηριστικών και των τρανζίστορ που είχε. Αυτές περιελάμβαναν μνήμη, είσοδο/έξοδο, έλεγχο και αριθμητικές/λογικές ικανότητες. Ονομάστηκε μικροεπεξεργαστής ή μικροϋπολογιστής. Είναι αυτό το τσιπ που αναφέρεται ως ο εγκέφαλος του προσωπικού επιτραπέζιου υπολογιστή - η κεντρική μονάδα επεξεργασίας, ή CPU [51].

1.1.2 Συμβατικός υπολογισμός

Με την πρώτη ματιά, το απόλυτο όριο υπολογισμού φαίνεται να είναι ένα θέμα μηχανικής. Πόση ενέργεια μπορείς να βάλεις σε ένα τσιπ χωρίς να το λιώσει; Πόσο γρήγορα μπορείς να ανατρέξεις στη μνήμη πυριτίου; Πόσο μεγάλο μπορούμε να κάνουμε τον υπολογιστή για να χωρέσει ακόμα και σε ένα δωμάτιο; Στην πραγματικότητα, ο υπολογισμός είναι πιο αφηρημένος και θεμελιώδης από το να βρούμε τον καλύτερο τρόπο για να οικοδομήσουμε έναν υπολογιστή. Αυτή η συνειδητοποίηση ήρθε στα μέσα της δεκαετίας του 1930, όταν ο μαθηματικός Αλόνζο Τσερτς (Alonzo Church) και ο Άλαν Τούρινγκ (Alan Turing) έδειξαν ότι κάθε υπολογισμός που περιλαμβάνει κομμάτια και byte μπορεί να γίνει σε έναν ιδεατό υπολογιστή γνωστό ως μηχανή Τούρινγκ (Turing machine) [185]. Δείχνοντας ότι όλοι οι κλασικοί υπολογιστές βασίζονται στις ίδιες αρχές, αυτή η ανακάλυψη επέτρεψε στους επιστήμονες και τους μαθηματικούς

να κάνουν βασικές υποθέσεις σχετικά με τον υπολογισμό χωρίς να εγκλωβιστούμε στις λεπτομέρειες της αρχιτεκτονικής του υπολογιστή. Για παράδειγμα, οι θεωρητικοί μπορούν τώρα να ταξινομήσουν υπολογιστικά προβλήματα σε ευρείες κατηγορίες. Τα προβλήματα πολυωνυμικής κλάσης πολυπλοκότητας - P είναι αυτά, σε γενικές γραμμές, που μπορούν να επιλυθούν γρήγορα, όπως είναι η ταξινόμηση σε αλφαβητική σειρά μιας λίστας ονομάτων. Τα προβλήματα μη πολυωνυμικής κλάσης πολυπλοκότητας - NP είναι πολύ δυσκολότερα στην επίλυση, αλλά σχετικά εύκολα να ελεγχθούν μόλις φτάσουμε σε μια εφικτή λύση. Ένα παράδειγμα είναι το πρόβλημα του πλανόδιου πωλητή, βρίσκοντας τη συντομότερη δυνατή διαδρομή μέσα από μια σειρά τοποθεσιών. Όλοι οι γνωστοί αλγόριθμοι για να επιφέρουν ένα αποτέλεσμα χρειάζονται μεγάλη υπολογιστική ισχύ και χρόνο εκτέλεση, και ακόμη και η εκτέλεση προσομοιώσεων σχετικά μικρών μετροπροβλημάτων μπορεί να είναι αδύνατες από οποιοδήποτε κλασικό υπολογιστή. Οι μαθηματικοί έχουν δείξει ότι αν μπορούσαμε να καταλήξουμε με μια γρήγορη και εύκολη συντόμευση για την επίλυση καθενός από τα δυσκολότερα είδη προβλημάτων NP , θα μπορούσαμε να τα επιλύσουμε όλα. Στην πραγματικότητα, εικάζεται ότι, τα προβλήματα κλάσης NP θα μπορούσαν να μετατραπούν σε προβλήματα κλάσης P . Αλλά είναι αβέβαιο αν υπάρχει τέτοια συνεπαγωγή — δηλαδή $P = NP$. Οι επιστήμονες δεν το πιστεύουν, αλλά αποδεικνύεται ότι αυτό είναι ένα από τα μεγάλα αναπάντητα ερωτήματα στα μαθηματικά. Το 1940, ο επιστήμονας των εργαστηρίων Μπελ (Bell Laboratories), Κλοντ Σάνον (Claude Shannon), έδειξε ότι τα κομμάτια (bits) δεν χρησιμοποιούνται μόνο από τους υπολογιστές [172]. Είναι οι βασικές μονάδες περιγραφής των πληροφοριών που ρέουν από ένα αντικείμενο σε ένα άλλο. Υπάρχουν φυσικοί νόμοι που ρυθμίζουν πόσο γρήγορα μπορεί να μετακινηθεί ένα κομμάτι πληροφορίας από το ένα μέρος στο άλλο, πόσες πληροφορίες μπορούν να μεταφερθούν εμπρός και πίσω σε ένα δεδομένο κανάλι επικοινωνιών και πόση ενέργεια χρειάζεται για να διαγραφεί ένα κομμάτι πληροφορίας από τη μνήμη. Όλα τα κλασικά μηχανήματα επεξεργασίας πληροφοριών υπόκεινται σε αυτούς τους νόμους επεξεργασίας της πληροφορίας. Ωστόσο, υπάρχει ένα σημείο πέρα από τον κλασικό υπολογιστή: το κβαντικό. Η πιθανολογική φύση της κβαντικής θεωρίας επιτρέπει σε άτομα και άλλα κβαντικά αντικείμενα να αποθηκεύουν πληροφορίες που δεν περιορίζονται μόνο στο δυαδικό σύστημα (0 ή 1) της θεωρίας πληροφοριών, αλλά μπορεί επίσης να είναι 0 και 1 την ίδια στιγμή. Οι φυσικοί σε όλο τον κόσμο προσπαθούν να κατασκευάσουν υποτυπώδεις κβαντικούς υπολογιστές που εκμεταλλεύονται την παραπάνω ιδιότητα και άλλες κβαντικές επιδράσεις για να εκτελέσουν υπολογισμούς που είναι πρακτικώς αδύνατο για απλούς υπολογιστές, όπως η εύρεση μιας συγκεκριμένης εγγραφής σε μια βάση δεδομένων με πολύ λίγα ερωτήματα. Αλλά οι επιστήμονες ακόμα προσπαθούν να καταλάβουν ποιες κβαντομηχανικές ιδιότητες κάνουν τους κβαντικούς υπολογιστές τόσο ισχυρούς και να κατασκευάσουν κβαντικούς υπολογιστές αρκετά μεγάλους σε ότι αφορά την επεξεργασία της πληροφορίας, ώστε να ξεπεράσουν σε χρησιμότητα τους κλασικούς υπολογιστές που ήδη χρησιμοποιούμε. Μαθαίνοντας την παράξενη λογική του κβαντικού κόσμου και αρχών και χρησιμοποιώντας τα στο πεδίο της Πληροφορικής και της επιστήμης της πληροφορίας, οι επιστήμονες μπαίνουν βαθιά στους νόμους του υπό ατομικού κόσμου. Τσως κάτι τόσο φαινομενικά κοινότοπο όσο η αναζήτηση για υπολογιστική ισχύ μπορεί να οδηγήσει σε μια πρωτόγνωρη κατανόηση του κβαντικού πεδίου των επιστημών.

1.1.3 Μη συμβατικός υπολογισμός

Τα συμβατικά πρότυπα πληροφορικής είναι συνήθως δύο: η συμβολική αναπαράσταση και η προσεγγιστική (ή μηχανιστική). Η πρώτη είναι κυρίαρχη από τη εμφάνιση της τεχνητής νοημοσύνης το καλοκαίρι του 1956. Αυτό το παράδειγμα υποθέτει ότι όλα όσα πρέπει να υπολογιστούν

πρέπει προηγουμένως να αντιπροσωπεύονται από την άποψη των δομών δεδομένων και των επαγωγικών συστημάτων, συμπεριλαμβανομένων των αλγορίθμων και των προγραμμάτων υλοποίησης. Έτσι, όλες οι εντολές πρέπει να είναι ρητές από την αρχή με δηλωτικό τρόπο. Το δεύτερο πρότυπο, περιλαμβάνει το πεδίο των τεχνητών νευρωνικών δικτύων, και άλλες βιολογικά εμπνευσμένες προσεγγίσεις. Στην ουσία, το πρώτο παράδειγμα σχετίζεται με την αρχιτεκτονική γενικού σκοπού φον Νόιμαν (von Neumann), ενώ το δεύτερο σχετίζεται με μηχανές ειδικού σκοπού στις οποίες ο έλεγχος είναι σαφής στο συγκεκριμένο ηλεκτρονικό κύκλωμα που ενσωματώνει κάθε μηχανή επεξεργασίας πληροφοριών. Εκτός από αυτά τα δύο κυρίαρχα πρότυπα, υπάρχουν και άλλες νέες προσεγγίσεις στο έργο του υπολογισμού, όχι τόσο καλά καθιερωμένες, αλλά στο σύνορο των ιδεών της επιστήμης και της μηχανικής. Αυτές οι νέες προσεγγίσεις περιλαμβάνουν κβαντικό υπολογισμό, υπολογισμό μεμβράνης, εξελικτικούς αλγόριθμους και πολλές άλλες εννοιολογικές προτάσεις ακόμα μακριά από εφικτές και ανταγωνιστικές υλοποιήσεις [92].

Αυτό το ειδικό θέμα της φυσικής Πληροφορικής ασχολείται με μερικά από αυτά τα νέα πρότυπα. Τα άρθρα είναι εκτεταμένες εκδόσεις των εγγράφων που επιλέγονται από τη δεύτερη διεθνή διάσκεψη εργασίας για την αλληλεπίδραση μεταξύ φυσικής και τεχνητής νοημοσύνης (IWINAC) που πραγματοποιήθηκε στο Λα μάνγκα Ντελ Μαρ Μενόρ (Ισπανία) κατά τη διάρκεια του 2007. Η προσέγγιση κβαντικής πληροφορικής ενσωματώνει δύο από τα κυριότερα επιστημονικά ευρήματα του περασμένου αιώνα: την θεωρία πληροφοριών και την κβαντική μηχανική. Κάτω από αυτή την ομπρέλα περιλαμβάνονται δύο έγγραφα του Μόνσερατ και του Κάλιξτο (Montserrat and Calixto). Το πρώτο σχετίζεται περισσότερο με την κβαντική επεξεργασία στην νευροεπιστήμη, προσπαθώντας να εξηγήσει σε κάποιο βαθμό, τη συνείδηση, και η δεύτερη εφαρμόζει αυτές τις βασικές αρχές σε παραλληλισμό, διεμπλοκή και κρυπτογραφία. Το πρότυπο μεμβράνης είναι εμπνευσμένο από μηχανισμούς στο κύτταρο, τις χημικές αντιδράσεις με διαφορετικά συστατικά και διαμερίσματα και τους νόμους παραγωγής και τους κανόνες επικοινωνίας μεταξύ των κατακερματισμένων ορίων. Σε αυτή την πτυχή, συμπεριλαμβάνονται τα έγγραφα των Παούν και Μαουρί (Paun and Mauri). Ο πρώτος ασχολείται με τη δύναμη και την αποδοτικότητα των συστημάτων νευρικών δικτύων, ενώ ο δεύτερος σχετίζεται με την εικασία του Παούν στα συστήματα μεμβράνης [155]. Η γενετική προσέγγιση αντιμετωπίζει προβλήματα βελτιστοποίησης με την αντιμετώπιση του χώρου αναζήτησης σύμφωνα με τις στρατηγικές που εμπνέονται από τη βιολογική εξέλιξη.

1.2 Ο κλάδος της βελτιστοποίησης

Η βελτιστοποίηση, επίσης γνωστή ως μαθηματικός προγραμματισμός, αποτελεί συλλογή μαθηματικών αρχών και μεθόδων που χρησιμοποιούνται για την επίλυση ποσοτικών προβλημάτων σε πολλούς κλάδους, συμπεριλαμβανομένης της φυσικής, της βιολογίας, της μηχανικής, των οικονομικών και των επιχειρήσεων. Ο συγκεκριμένος κλάδος έγινε ευρέως γνωστός από τη συνειδητοποίηση ότι τα ποσοτικά προβλήματα σε σαφώς διαφορετικούς κλάδους έχουν κοινά σημαντικά μαθηματικά στοιχεία. Εξαιτίας αυτού του κοινού, πολλά προβλήματα μπορούν να διατυπωθούν και να λυθούν με τη χρήση του ενοποιημένου συνόλου ιδεών και μεθόδων που αποτελούν το πεδίο της βελτιστοποίησης. Ο ιστορικός όρος μαθηματικού προγραμματισμού, σε γενικές γραμμές συνώνυμος με τη βελτιστοποίηση, επινοήθηκε τη δεκαετία του 1940 πριν ο προγραμματισμός εξισωθεί με τον προγραμματισμό του υπολογιστή. Ο μαθηματικός προγραμματισμός περιλαμβάνει τη μελέτη της μαθηματικής δομής των προβλημάτων βελτιστοποίησης, την εφεύρεση των μεθόδων για την επίλυση αυτών των προβλημάτων, τη μελέτη των μαθηματικών ιδιοτήτων αυτών των μεθόδων και την εφαρμογή αυτών των μεθόδων στους υπολογιστές. Οι

ταχύτεροι υπολογιστές έχουν επεκτείνει σημαντικά το μέγεθος και την πολυπλοκότητα των προβλημάτων βελτιστοποίησης που μπορούν να επιλυθούν. Η ανάπτυξη τεχνικών βελτιστοποίησης έχει παράλληλη πρόοδο όχι μόνο στην επιστήμη των υπολογιστών, αλλά και στην επιχειρησιακή έρευνα, την αριθμητική ανάλυση, τη θεωρία παιγνίων, τα μαθηματικά των οικονομικών, τη θεωρία ελέγχου και τη συνδυαστική. Τα προβλήματα βελτιστοποίησης έχουν συνήθως τρία θεμελιώδη στοιχεία.

1. Το πρώτο είναι μια μεμονωμένη αριθμητική ποσότητα ή αντικειμενική συνάρτηση, που πρόκειται να μεγιστοποιηθεί ή να ελαχιστοποιηθεί. Στόχος μπορεί να είναι η μεγιστοποίηση της απόδοσης ενός χαρτοφυλακίου μετοχών, η μείωση του κόστους παραγωγής ή η μεγιστοποίηση των κερδών της εταιρείας, η ώρα άφιξης ενός οχήματος σε συγκεκριμένο προορισμό ή το μερίδιο ψήφου ενός πολιτικού υποψηφίου.
2. Το δεύτερο στοιχείο είναι μια συλλογή μεταβλητών, οι οποίες είναι ποσότητες των οποίων οι τιμές μπορούν να μεταβάλλονται, προκειμένου να βελτιστοποιηθούν οι στόχοι. Στα παραδείγματα περιλαμβάνονται οι ποσότητες αποθέματος που πρόκειται να αγοραστούν ή να πωληθούν, τα ποσά των διαφόρων πόρων που πρέπει να διατεθούν σε διαφορετικές παραγωγικές δραστηριότητες, η διαδρομή που πρέπει να ακολουθήσει ένα όχημα μέσω ενός δικτύου κυκλοφορίας, ή οι πολιτικές που πρέπει να υποστηρίξει ένας υποψήφιος.
3. Το τρίτο στοιχείο ενός προβλήματος βελτιστοποίησης είναι ένα σύνολο περιορισμών, οι οποίοι είναι περιορισμοί στις τιμές που μπορούν να λάβουν οι μεταβλητές. Για παράδειγμα, μια διεργασία παραγωγής δεν μπορεί να απαιτήσει περισσότερους πόρους από όσους είναι διαθέσιμοι, ούτε μπορεί να χρησιμοποιήσει λιγότερους από μηδέν πόρους.

Μέσα σε αυτό το ευρύ πλαίσιο, τα προβλήματα βελτιστοποίησης μπορούν να έχουν διαφορετικές μαθηματικές ιδιότητες. Τα προβλήματα στα οποία οι μεταβλητές είναι συνεχείς ποσότητες (όπως στο παράδειγμα εκχώρησης πόρων) απαιτούν διαφορετική προσέγγιση από τα προβλήματα στα οποία οι μεταβλητές είναι διακριτές ή συνδυαστική ποσότητα (όπως στην επιλογή μιας διαδρομής οχήματος μεταξύ ενός προκαθορισμένου συνόλου δυνατοτήτων). Μια σημαντική κατηγορία βελτιστοποίησης είναι γνωστή ως γραμμικός προγραμματισμός. Η γραμμική ένδειξη υποδεικνύει ότι δεν έχουν αυξηθεί οι μεταβλητές σε ανώτερες δυνάμεις. Για την κλάση αυτή, τα προβλήματα περιλαμβάνουν την ελαχιστοποίηση (ή μεγιστοποίηση) μιας γραμμικής αντικειμενικής συνάρτησης, των οποίων οι μεταβλητές είναι πραγματικοί αριθμοί που ελαχιστοποιούνται για να ικανοποιήσουν ένα σύστημα γραμμικών εξισώσεων και ανισοτήτων.

Μια άλλη σημαντική κατηγορία βελτιστοποίησης είναι γνωστή ως μη γραμμικός προγραμματισμός. Στον μη γραμμικό προγραμματισμό οι μεταβλητές είναι πραγματικοί αριθμοί και ο στόχος ή μερικοί από τους περιορισμούς είναι η βελτιστοποίηση μη γραμμικών συναρτήσεων (ενδεχομένως με τη συμμετοχή τετραγώνων, τετραγωνικών ριζών, τριγωνομετρικών λειτουργιών ή παράγωγα των μεταβλητών). Άλλες σημαντικές κατηγορίες προβλημάτων βελτιστοποίησης περιλαμβάνουν τον στοχαστικό προγραμματισμό, στον οποίο η αντικειμενική συνάρτηση ή οι περιορισμοί εξαρτώνται από τυχαίες μεταβλητές, έτσι ώστε το βέλτιστο βρίσκεται σε ορισμένα σημεία. Η βελτιστοποίηση δικτύων, η οποία περιλαμβάνει τη βελτιστοποίηση κάποιας ιδιότητας μιας ροής μέσω ενός δικτύου, όπως η μεγιστοποίηση της ποσότητας του υλικού που μπορεί να μεταφερθεί μεταξύ δύο δεδομένων θέσεων στο δίκτυο και η συνδυαστική βελτιστοποίηση, στην οποία η λύση πρέπει να βρεθεί μεταξύ ενός πεπερασμένου αλλά πολύ μεγάλου συνόλου πιθανών τιμών, όπως

οι πολλοί πιθανοί τρόποι για να αντιστοιχίσουν 20 μονάδες παραγωγής σε 20 τοποθεσίες.

Αν και χρησιμοποιείται ευρέως τώρα για την επίλυση προβλημάτων καθημερινής λήψης αποφάσεων, ο γραμμικός προγραμματισμός ήταν συγκριτικά άγνωστος πριν από το 1947. Καμία δουλειά δεν είχε γίνει πριν από αυτή την ημερομηνία, παρόλο που ο Γάλλος μαθηματικός Τζόζεφ Φουριέρ (Joseph Fourier) φαινόταν να γνωρίζει τις δυνατότητες του υποκειμένου ήδη από το 1823 [77]. Το 1939 ένας Ρώσος μαθηματικός, ο Λεονίντ Καντόροβιτς (Leonid Kantorovich), δημοσίευσε μια εκτεταμένη μονογραφία, («μαθηματικές μέθοδοι για την οργάνωση και τον προγραμματισμό της παραγωγής»), το οποίο τώρα πιστώνεται με το να είναι η πρώτη πραγματεία που αναγνωρίζει ότι ορισμένες σημαντικές ευρείες τάξεις των προβλημάτων προγραμματισμού είχαν καλά καθορισμένες μαθηματικές δομές. Δυστυχώς, οι προτάσεις του Καντόροβιτς παρέμειναν ως επί το πλείστον άγνωστες τόσο στη Σοβιετική Ένωση όσο και αλλού για σχεδόν δύο δεκαετίες. Εν τω μεταξύ, ο γραμμικός προγραμματισμός είχε αναπτυχθεί σημαντικά στις Ηνωμένες Πολιτείες και τη Δυτική Ευρώπη [95].

Την περίοδο μετά τον Β' Παγκόσμιο πόλεμο, αξιωματούχοι της κυβέρνησης των Ηνωμένων Πολιτειών πίστεψαν ότι ο αποτελεσματικός συντονισμός των ενεργειών και των πόρων ενός ολόκληρου έθνους σε περίπτωση πυρηνικού πολέμου θα απαιτούσε τη χρήση τεχνικών επιστημονικού σχεδιασμού. Η έλευση του υπολογιστή κατέστησε εφικτή μια τέτοια προσέγγιση. Η εντατική εργασία ξεκίνησε το 1947 στην Πολεμική Αεροπορία των ΗΠΑ. Το γραμμικό μοντέλο προγραμματισμού προτάθηκε επειδή ήταν σχετικά απλό από μαθηματική άποψη, και ωστόσο παρείχε ένα αρκετά γενικό και πρακτικό πλαίσιο για την εκπροσώπηση αλληλεξαρτώμενων δραστηριοτήτων που μοιράζονται περιορισμένους πόρους. Στο γραμμικό μοντέλο προγραμματισμού, ο τύπος βλέπει το σύστημα να βελτιστοποιείται καθώς αποτελείται από διάφορες δραστηριότητες που υποτίθεται ότι απαιτούν ροή εισροών (π.χ. εργασία και πρώτες ύλες) και εκροές (π.χ. τελικά αγαθά και υπηρεσίες) διαφόρων τύπων ανάλογα με το επίπεδο της δραστηριότητας.

Τα επίπεδα δραστηριότητας θεωρείται ότι προέρχονται αντιπροσωπευτικά από μη αρνητικούς αριθμούς. Το επαναστατικό χαρακτηριστικό της προσέγγισης έγκειται στην έκφραση του στόχου της διαδικασίας λήψης αποφάσεων όσον αφορά την ελαχιστοποίηση ή μεγιστοποίηση μιας γραμμικής αντικειμενικής συνάρτησης — για παράδειγμα, τη μεγιστοποίηση των πιθανών συσχετίσεων στην περίπτωση της Πολεμικής Αεροπορίας, ή τη μεγιστοποίηση των κερδών στη βιομηχανία.

Πριν από το 1947 όλοι οι πρακτικοί σχεδιασμοί χαρακτηρίζονταν από μια σειρά που επέβαλε κανόνες διαδικασίας και προτεραιότητων. Δεν αναφέρθηκε ποτέ γενικός στόχος, πιθανώς λόγω της αδυναμίας εκτέλεσης των αναγκαίων υπολογισμών για την ελαχιστοποίηση μιας αντικειμενικής συνάρτησης υπό περιορισμούς. Στο 1947 μια μέθοδος (που περιγράφεται στην ενότητα η μέθοδος simplex) εισήχθη που αποδείχθηκε για την αποτελεσματική επίλυση πρακτικών προβλημάτων [40]. Το ενδιαφέρον για τον γραμμικό προγραμματισμό αυξήθηκε ταχέως και κατά το 1951 η χρήση του εξαπλώθηκε στη βιομηχανία. Σήμερα είναι σχεδόν αδύνατο να ονομάσουμε μια βιομηχανία που δεν χρησιμοποιεί μαθηματικό προγραμματισμό σε κάποια μορφή, αν και οι εφαρμογές και ο βαθμός στον οποίο χρησιμοποιείται ποικίλλουν σημαντικά, ακόμη και εντός της ίδιας βιομηχανίας.

Το ενδιαφέρον για γραμμικό προγραμματισμό έχει επίσης επεκταθεί και στα οικονομικά. Το 1937 ο Ούγγρος μαθηματικός Τζον φον Νόιμαν (John von Neumann) ανέλυσε μια σταθερά διευρυνόμενη οικονομία βασισμένη σε εναλλακτικές μεθόδους παραγωγής και σταθερούς τεχνικούς συντελεστές. Όσον αφορά τη μαθηματική ιστορία, η μελέτη των συστημάτων γραμμικής ανισότητας δεν κίνησε σχεδόν κανένα ενδιαφέρον πριν από το 1936. Το 1911 μια κίνηση κορυφής προς κορυφή κατά μήκος των άκρων ενός πολυέδρου προτάθηκε ως ένας τρόπος για να επιλυθεί ένα πρόβλημα που αφορούσε τη βελτιστοποίηση και το 1941, η κίνηση κατά μήκος των άκρων

προτάθηκε για ένα πρόβλημα που αφορά τη μεταφορά. Τα εύσημα για την τοποθέτηση σε πολλά από τα μαθηματικά θεμέλια θα πρέπει να πάνε στον φον Νόιμαν. Το 1928 δημοσίευσε τη διάσημη βίβλο του για τη θεωρία των παιγνίων και το έργο του κορυφώθηκε το 1944 με τη δημοσίευση, σε συνεργασία με τον αυστριακό οικονομολόγο Όσκαρ Μόργκενστερν, της κλασικής θεωρίας παιγνίων και της οικονομικής συμπεριφοράς. Το 1947 ο φον Νόιμαν υποστήριζε την ισοδυναμία των γραμμικών προγραμμάτων και των παιγνίων πινάκων, εισήγαγε τη σημαντική έννοια της δυϊκότητας, και έκανε αρκετές προτάσεις για την αριθμητική λύση του γραμμικού προγραμματισμού και τα προβλήματα του παιχνιδιού. Σοβαρό ενδιαφέρον από άλλους μαθηματικούς ξεκίνησε το 1948 με την αυστηρή ανάπτυξη της δυϊκότητας και συναφή θέματα [115].

Η μέθοδος γενικής μονής ήταν προγραμματισμένη για πρώτη φορά το 1951 για το γραφείο προτύπων SEAC των Ηνωμένων Πολιτειών. Ξεκινώντας από το 1952, η μέθοδος του απλού υπολογιστή προγραμματίστηκε για χρήση σε διάφορους υπολογιστές IBM και αργότερα για άλλες εταιρείες. Ως αποτέλεσμα, οι εμπορικές εφαρμογές των γραμμικών προγραμμάτων στη βιομηχανία και την κυβέρνηση αυξήθηκαν ραγδαία. Νέες υπολογιστικές τεχνικές και παραλλαγές παλαιότερων τεχνικών συνέχισαν να αναπτύσσονται.

Πιο πρόσφατα υπήρξε μεγάλο ενδιαφέρον για την επίλυση μεγάλων γραμμικών προβλημάτων με ειδικές δομές — για παράδειγμα, τα εταιρικά μοντέλα και τα μοντέλα εθνικού σχεδιασμού, είναι δυναμικά και παρουσιάζουν μια ιεραρχική δομή. Εκτιμάται ότι ορισμένες αναπτυσσόμενες χώρες θα έχουν τη δυνατότητα να αυξήσουν το ακαθάριστο εθνικό τους προϊόν (ΑΕΠ) κατά 10 έως 15 τοις εκατό ετησίως, εάν μπορούν να κατασκευαστούν, να βελτιστοποιηθούν και να υλοποιηθούν αναλυτικά μοντέλα ανάπτυξης της οικονομίας.

Ένα απλό πρόβλημα στον γραμμικό προγραμματισμό είναι αυτό στο οποίο είναι απαραίτητο να βρεθεί η μέγιστη (ή ελάχιστη) τιμή μιας απλής συνάρτησης που υπόκειται σε συγκεκριμένους περιορισμούς. Ένα παράδειγμα μπορεί να είναι αυτό ενός εργοστασίου που παράγει δύο εμπορεύματα. Σε κάθε εκτέλεση παραγωγής, το εργοστάσιο παράγει $5x$ του πρώτου τύπου και x^2 του δεύτερου. Εάν το κέρδος στο δεύτερο τύπο είναι διπλάσιο από το πρώτο, τότε το $Qx + 2x^2$ αντιπροσωπεύει το συνολικό κέρδος. Η συνάρτηση $Q12 + 2x^2$ είναι γνωστή ως η αντικειμενική συνάρτηση.

Σαφώς το κέρδος θα είναι υψηλότερο αν το εργοστάσιο αφιερώσει ολόκληρη την παραγωγική του ικανότητα για να κάνει το δεύτερο είδος εμπορεύματος. Σε μια πρακτική κατάσταση, ωστόσο, αυτό μπορεί να μην είναι εφικτό. Ένα σύνολο περιορισμών εισάγεται από παράγοντες όπως η διαθεσιμότητα του χρόνου μηχανής, της εργασίας και των πρώτων υλών. Για παράδειγμα, εάν ο δεύτερος τύπος εμπορεύματος απαιτεί μια πρώτη ύλη που είναι περιορισμένη ώστε να μην μπορούν να γίνουν περισσότερες από πέντε σε οποιαδήποτε παρτίδα, τότε το x^2 πρέπει να είναι μικρότερο ή ίσο με πέντε, δηλαδή, $x^2 \leq 5$. Εάν το πρώτο εμπόρευμα απαιτεί έναν άλλο τύπο υλικού που τον περιορίζει σε οκτώ ανά παρτίδα, τότε $x \leq 8$. Εάν το $Q1$ και το x^2 χρειαστούν ίσο χρόνο για να υλοποιηθούν και ο χρόνος που διατίθεται για το μηχάνημα επιτρέπει το πολύ 10 να γίνει σε μια παρτίδα, τότε το $Q1 + x^2$ πρέπει να είναι μικρότερο ή ίσο με 10. Δηλαδή, $Q1 + x^2 \leq 10$.

Δύο άλλοι περιορισμοί είναι ότι το $Q1$ και το x^2 πρέπει να είναι μεγαλύτερα ή ίσα από το μηδέν, επειδή είναι αδύνατο να γίνει αρνητικός αριθμός. Δηλαδή, $Q1 \geq 0$ και $x^2 \geq 0$. Το πρόβλημα είναι να βρείτε τις τιμές του Q και x^2 για τις οποίες το κέρδος είναι το μέγιστο. Οποιαδήποτε λύση μπορεί να συμβολίζεται με ένα ζεύγος αριθμών (x^5x, x^2) . Για παράδειγμα, εάν $QQ = 3$ και $x^2 = 6$, η λύση είναι (3, 6). Αυτοί οι αριθμοί μπορούν να αντιπροσωπεύονται από σημεία που σχεδιάζονται σε δύο άξονες. Η απόσταση κατά μήκος του οριζόντιου άξονα αντιπροσωπεύει το x και ο κατά μήκος της κατακόρυφης αντιπροσωπεύει x^2 . Λόγω των περιορισμών που αναφέρονται παραπάνω, οι εφικτές λύσεις πρέπει να βρίσκονται σε μια ορισμένη περιοχή του γραφήματος. Για παράδειγμα,

ο περιορισμός $Q10 \geq 0$ σημαίνει ότι τα σημεία που αντιπροσωπεύουν εφικτές λύσεις βρίσκονται στα δεξιά του άξονα x_2 . Ομοίως, ο περιορισμός $X2 \geq 0$ σημαίνει ότι βρίσκονται επίσης πάνω ή πάνω από τον άξονα $X10$. Η εφαρμογή ολόκληρου του συνόλου των περιορισμών δίνει το εφικτό σύνολο λύσεων, το οποίο οριοθετείται από ένα πολύγωνο που σχηματίζεται από την τομή των γραμμών $Q1 = 0, X2 = 0, Q1 = 8, X2 = 5$ και $Q1 + X2 = 10$. Για παράδειγμα, η παραγωγή τριών τεμαχίων εμπορευμάτων $Q4$ και τεσσάρων του $X2$ είναι μια εφικτή λύση, αφού το σημείο $(3, 4)$ βρίσκεται σε αυτή την περιοχή.

Για να βρεθεί η καλύτερη λύση, ωστόσο, η αντικειμενική συνάρτηση $Q2 + 2x_2 = k$ σχεδιάζεται για κάποια τιμή k , έστω $k = 4$. Καθώς η k αυξάνεται, παράγεται μια οικογένεια παράλληλων γραμμών και η γραμμή για το $k = 15$ αγγίζει απλώς τον περιορισμό που έχει οριστεί στο σημείο $(5,5)$. Εάν η k αυξηθεί περαιτέρω, οι τιμές του x και του $X2$ θα βρίσκονται εκτός του συνόλου των εφικτών λύσεων. Έτσι, η καλύτερη λύση είναι αυτή στην οποία γίνονται ίσες ποσότητες κάθε εμπορεύματος. Δεν είναι τυχαίο ότι μια βέλτιστη λύση εμφανίζεται σε μια κορυφή, ή "ακραίο σημείο" της περιοχής. Αυτό θα ισχύει πάντα για γραμμικά προβλήματα, αν και η βέλτιστη λύση μπορεί να μην είναι μοναδική. Έτσι, η λύση αυτών των προβλημάτων μειώνει την εύρεση του ακραίου σημείου (ή σημείων) που αποδίδει τη μεγαλύτερη τιμή για την αντικειμενική συνάρτηση.

Η γραφική μέθοδος της λύσης που περιγράψαμε είναι χρήσιμη μόνο για συστήματα ανισοτήτων που αφορούν δύο μεταβλητές. Στην πράξη, τα προβλήματα συχνά περιλαμβάνουν εκατοντάδες εξισώσεις με χιλιάδες μεταβλητές, οι οποίες μπορούν να οδηγήσουν σε αστρονομικό αριθμό ακραίων σημείων. Το 1947, ο Ντάζιγκ (George Dantzig), ένας μαθηματικός σύμβουλος της Πολεμικής Αεροπορίας των ΗΠΑ, επινόησε τη μέθοδο **simplex** για να περιορίσει τον αριθμό των ακραίων σημείων που πρέπει να εξεταστούν. Η μέθοδος **simplex** είναι ένας από τους πιο χρήσιμους και αποδοτικούς αλγόριθμους που εφευρέθηκε ποτέ και εξακολουθεί να είναι η συνήθης μέθοδος που χρησιμοποιείται σε υπολογιστές για την επίλυση προβλημάτων βελτιστοποίησης. Πρώτον, η μέθοδος υποθέτει ότι είναι γνωστό ένα ακραίο σημείο. (Εάν δεν δοθεί ακραίο σημείο, χρησιμοποιείται μια παραλλαγή της μεθόδου **simplex**, που ονομάζεται φάση I, για να βρεθεί μία ή για να προσδιοριστεί ότι δεν υπάρχουν εφικτές λύσεις.) Στη συνέχεια, χρησιμοποιώντας μια αλγεβρική περιγραφή του προβλήματος, ένα τεστ καθορίζει αν αυτό το ακραίο σημείο είναι βέλτιστο. Εάν η δοκιμή για την ποιότητα δεν περάσει, ένα γειτονικό ακραίο σημείο επιδιώκεται κατά μήκος ενός άκρου προς την κατεύθυνση για την οποία η αξία της αντικειμενικής συνάρτησης αυξάνεται με τον ταχύτερο ρυθμό [40].

Μερικές φορές κάποιος μπορεί να κινηθεί κατά μήκος ενός ακραίου σημείου και να κάνει στην αντικειμενική τιμή συνάρτησης αύξηση χωρίς δέσμευση. Εάν συμβεί αυτό, η διαδικασία τερματίζεται κατά μήκος του ακραίου σημείου του οποίου ο στόχος πηγαίνει στο θετικό άπειρο. Εάν όχι, επιτυγχάνεται ένα νέο ακραίο σημείο που έχει τουλάχιστον την ίδια τιμή αντικειμενικής συνάρτησης με τον προκάτοχό του. Η ακολουθία που περιγράφεται επαναλαμβάνεται. Ο τερματισμός επιτυγχάνεται όταν βρεθεί ένα βέλτιστο άκρο σημείο ή εμφανίζεται η μη οριοθετημένη υπόθεση. Αν και κατ' αρχήν τα απαραίτητα βήματα μπορεί να αυξηθούν εκθετικά με τον αριθμό των ακραίων σημείων, στην πράξη η μέθοδος συνήθως παραπέμπει στη βέλτιστη λύση σε μια σειρά βημάτων που είναι μόνο ένα μικρό πολλαπλάσιο του αριθμού των ακραίων σημείων. Για να απεικονίσει τη μέθοδο **simplex**, το παράδειγμα θα επιλυθεί ξανά. Το πρόβλημα πρέπει πρώτα να τεθεί σε κανονική μορφή μετατρέποντας τις γραμμικές ανισότητες σε εξισώσεις εισάγοντας "σταθερές μεταβλητές" $Q1 \geq 0$ (έτσι ώστε $Q1 + Q1 = 8$), $Q8 \geq 0$ (έτσι ώστε $X2 + Q1 = 5$), $Q5 \geq 0$ (έτσι ώστε $Q1 + X2 + Q5 = 10$) και η μεταβλητή x_0 για την τιμή του αντικειμένου συνάρτησης (έτσι ώστε $Q12 + 2x_2x_0 = 0$). Το πρόβλημα μπορεί στη συνέχεια να αναδιατυπωθεί ως προς την εύρεση μη αρνητικών ποσοτήτων $Q5, \dots, x_5$ και το μεγαλύτερο δυνατό x_0 που ικανοποιεί τις εξισώσεις

που προκύπτουν.

Μια προφανής λύση είναι να οριστούν οι μεταβλητές στόχου $Q1 = X2 = 0$, η οποία αντιστοιχεί στο ακραίο σημείο στην προέλευση. Εάν μία από τις μεταβλητές του στόχου αυξηθεί από το μηδέν, ενώ η άλλη είναι σταθερή στο μηδέν, η αντικειμενική τιμή $x0$ θα αυξηθεί όπως θέλετε (με την επιφύλαξη των σταθερών μεταβλητών που ικανοποιούν τους περιορισμούς της ισότητας). Η μεταβλητή $X2$ παράγει τη μεγαλύτερη αύξηση $x0$ ανά μονάδα αλλαγής, έτσι χρησιμοποιείται πρώτα. Η αύξησή του περιορίζεται από την απαίτηση μη αρνητικότητας στις μεταβλητές. Συγκεκριμένα, εάν το $X2$ αυξηθεί πέραν των 5, το $X5$ γίνεται αρνητικό. Στο $X2 = 5$, αυτή η κατάσταση παράγει μια νέα λύση — $(x0, Q1, X2, Q1, Q1, Q5) = (10, 0, 5, 8, 0, 5)$ — που αντιστοιχεί στο ακραίο σημείο (0,5) στο σχήμα. Το σύστημα των εξισώσεων τίθεται σε μια ισοδύναμη μορφή με την επίλυση για τις μη μηδενικές μεταβλητές $x0, X2, Q5, X5$ σε όρους αυτών των μεταβλητών τώρα στο μηδέν; Δηλαδή, $X5X$ και $X5A$. Έτσι, η νέα αντικειμενική συνάρτηση είναι $Q12x4 = 10$, ενώ οι περιορισμοί είναι $Q1 + Q1 = 8, X2 + Q4 = 5$, και $Q1Q4 + Q1 = 5$. Είναι πλέον προφανές ότι η αύξηση του $X5X$ ενώ κατέχει $X5A$ ίσο με το μηδέν θα επιφέρει περαιτέρω αύξηση στο $x0$. Ο μη αρνητικός περιορισμός στο $Q5$ αποτρέπει το $Q15$ από το να υπερβεί τα πέντε. Η νέα λύση — $(x0, Q1, X2, Q3, Q1, Q5) = (15, 5, 5, 3, 0, 0)$ — αντιστοιχεί στο ακραίο σημείο (5, 5) στο σχήμα. Τέλος, δεδομένου ότι η επίλυση για το $x0$ από την άποψη των μεταβλητών $Q1$ και $X5$ (που είναι επί του παρόντος σε μηδενική τιμή) αποδίδει $x0 = 15Q1Q5$, μπορεί να παρατηρηθεί ότι οποιαδήποτε περαιτέρω αλλαγή σε αυτές τις σταθερές μεταβλητές θα μειώσει την αντικειμενική τιμή. Ως εκ τούτου, υπάρχει μια βέλτιστη λύση στο ακραίο σημείο (5, 5).

Στην πράξη, τα προβλήματα βελτιστοποίησης διαμορφώνονται χρησιμοποιώντας τους πίνακες — ένας συμπαγής συμβολισμός για τον χειρισμό των περιορισμών και τη δοκιμή της αντικειμενικής συνάρτησης αλγεβρικά. Το αρχικό (ή «αρχέγονο») πρόβλημα βελτιστοποίησης πήρε την τυπική σύνθεσή του από τον φον Νόιμαν το 1947. Στο πρωτογενές πρόβλημα, ο στόχος αντικαθίσταται από το προϊόν (px) ενός διανύσματος $x = (x_1, X_2, Q_1, \dots, X_N)T$, τα στοιχεία των οποίων είναι οι μεταβλητές του στόχου και όπου το σύμβολο «μεταφορά» εκθέτη υποδεικνύει ότι το διάνυσμα πρέπει να γράφεται κάθετα, και ένα άλλο διάνυσμα $p = (p_1, p_2, \dots, p_N)$, των οποίων τα συστατικά είναι οι συντελεστές κάθε μιας από τις αντικειμενικές μεταβλητές. Επιπλέον, το σύστημα των περιορισμών ανισότητας αντικαθίσταται από $AX \leq b$, όπου το m από n Πίνακας A αντικαθιστά τους περιορισμούς m στις μεταβλητές n , και $b = (B_1, B_2, B_3, \dots, B_M)T$ είναι ένα διάνυσμα των οποίων τα συστατικά είναι τα όρια ανισότητας.

Ο κλάδος των μαθηματικών που είναι γνωστός ως διαφορικός λογισμός άρχισε με τις προσπάθειες για την απόδειξη αυτής της λύσης, μαζί με την πρόκληση το 1696 από τον Ελβετό μαθηματικό Γιόχαν Μπερνούλι (Johann Bernoulli) να βρει την καμπύλη που ελαχιστοποιεί το χρόνο που παίρνει ένα αντικείμενο για να γλιστρήσει, κάτω από μόνο τη δύναμη της βαρύτητας, μεταξύ δύο μη κατακόρυφων σημείων. Εκτός από τον Γιόχαν Μπερνούλι, ο αδερφός του, Τζέικομπ Μπερνούλι (Jacob Bernoulli), ο Γερμανός Γκότφριντ Βίλχελμ Λάιμπνιτς (Gottfried Wilhelm Leibniz), και ο Άγγλος Ισαάκ Νιούτον (Isaac Newton), όλοι παρήγαγαν σωστές λύσεις. Συγκεκριμένα, η προσέγγιση του Νεύτωνα στη λύση διαδραματίζει θεμελιώδη ρόλο σε πολλούς μη γραμμικούς αλγόριθμους. Άλλες επιρροές στην ανάπτυξη του μη γραμμικού προγραμματισμού, όπως κυρτή ανάλυση, θεωρία δυαδικότητας και θεωρία ελέγχου, αναπτύχθηκαν σε μεγάλο βαθμό μετά από το 1940. Για προβλήματα που περιλαμβάνουν περιορισμούς καθώς και μια αντικειμενική συνάρτηση, οι συνθήκες βελτιστοποίησης που ανακάλυψε ο Αμερικανός μαθηματικός Γουίλιαμ Καρούες (William Carruthers) και άλλοι στα τέλη της δεκαετίας του 1940 έγιναν ουσιαστικό εργαλείο για την αναγνώριση λύσεων και την οδήγηση της συμπεριφοράς αλγορίθμων [196].

Ένας σημαντικός πρώιμος αλγόριθμος για την επίλυση μη γραμμικών προγραμμάτων δόθηκε

από τον βραβευμένο με Νόμπελ Νορβηγό οικονομολόγο Ράγκναρ Φρίς (Ragnar Frisch) στα μέσα της δεκαετίας του 1950. Περιέργως, η προσέγγισή του μπήκε στο περιθώριο για ορισμένες δεκαετίες, επανεμφανιζόμενη ως βιώσιμη και ανταγωνιστική προσέγγιση μόνο τη δεκαετία του 1990. Άλλες σημαντικές αλγοριθμικές προσεγγίσεις περιλαμβάνουν διαδοχικούς τετραγωνικούς προγραμματισμούς, στους οποίους επιλύεται ένα κατά προσέγγιση πρόβλημα με έναν τετραγωνικό στόχο και γραμμικούς περιορισμούς για να επιτευχθεί κάθε βήμα αναζήτησης και τις μεθόδους επιβολής ποινών, συμπεριλαμβανομένης της «μεθόδου των πολλαπλασιαστών», στις οποίες τα σημεία που δεν ικανοποιούν τους περιορισμούς επιβαρύνουν τους όρους ποινής με στόχο την αποθάρρυνση των αλγορίθμων από την επίσκεψή τους [19]. Ο βραβευμένος με Νόμπελ Αμερικανός οικονομολόγος Χάρι Μάρκωβιτς (Harry Markowitz) παρείχε μια ώθηση για μη γραμμική βελτιστοποίηση το 1958, όταν διατύπωσε το πρόβλημα της εύρεσης ενός αποτελεσματικού επενδυτικού χαρτοφυλακίου ως μη γραμμικό πρόβλημα βελτιστοποίησης με μια δευτεροβάθμια αντικειμενική συνάρτηση [195].

Οι τεχνικές μη γραμμικής βελτιστοποίησης χρησιμοποιούνται πλέον ευρέως στη χρηματοδότηση, την οικονομία, την κατασκευή, τον έλεγχο, τη μοντελοποίηση καιρού και όλους τους κλάδους της μηχανικής. Ένα πρόβλημα βελτιστοποίησης είναι μη γραμμικό εάν η αντικειμενική συνάρτηση $f(x)$ ή οποιοσδήποτε από τους περιορισμούς ανισότητας $CI_{(x)} \leq 0, i = 1, 2, \dots, m$, ή περιορισμοί ισότητας $DJ_{(x)} = 0, j = 1, 2, \dots, n$, είναι μη γραμμικές συναρτήσεις του φορέα των μεταβλητών x . Για παράδειγμα, αν το x περιέχει τα στοιχεία Q_1 και Q_2 , τότε η συνάρτηση $3 + 2Q_17Q_2$ είναι γραμμική, ενώ οι συναρτήσεις $(Q_1)3 + 2xQ_1$ και $3Q_1 + 2Q_1Q_2 + Q_2$ είναι μη γραμμικές.

Τα μη γραμμικά προβλήματα προκύπτουν όταν ο στόχος ή οι περιορισμοί δεν μπορούν να εκφραστούν ως γραμμικές συναρτήσεις χωρίς να θυσιάζονται ορισμένα ουσιώδη μη γραμμικά χαρακτηριστικά του πραγματικού παγκόσμιου συστήματος. Για παράδειγμα, η διπλωμένη διάπλαση ενός μορίου πρωτεΐνης πιστεύεται ότι είναι αυτή που ελαχιστοποιεί μια ορισμένη μη γραμμική συνάρτηση των αποστάσεων μεταξύ των πυρήνων των ατόμων που τα απαρτίζουν και αυτές οι αποστάσεις είναι μη γραμμικές συναρτήσεις των θέσεων των Πυρήνων.

Στα χρηματοοικονομικά, ο κίνδυνος που συνδέεται με ένα χαρτοφυλάκιο επενδύσεων, όπως μετράται από τη διακύμανση της απόδοσης του χαρτοφυλακίου, είναι μια μη γραμμική συνάρτηση των ποσών που επενδύονται σε κάθε εγγύηση στο χαρτοφυλάκιο. Στη χημεία, η συγκέντρωση κάθε χημικού στοιχείου σε ένα διάλυμα είναι συχνά μια μη γραμμική συνάρτηση του χρόνου, καθώς οι αντιδράσεις μεταξύ χημικών στοιχείων συνήθως λαμβάνουν χώρα σύμφωνα με εκθετικούς τύπους.

Μη γραμμικά προβλήματα μπορεί να κατηγοριοποιηθούν σύμφωνα με διάφορες ιδιότητες. Υπάρχουν προβλήματα στα οποία ο στόχος και οι περιορισμοί είναι ομαλές συναρτήσεις και υπάρχουν μη ομαλά προβλήματα στα οποία η κλίση ή η αξία μιας συνάρτησης μπορεί να αλλάξει απότομα. Υπάρχουν προβλήματα χωρίς περιορισμό, στα οποία ο στόχος είναι να ελαχιστοποιηθεί (ή να μεγιστοποιηθεί) η αντικειμενική συνάρτηση $f(x)$ χωρίς περιορισμούς στην τιμή του x και υπάρχουν περιορισμένα προβλήματα, στα οποία τα συστατικά του x πρέπει να ικανοποιούν ορισμένα όρια ή άλλα πιο σύνθετα διασυνδέσεις. Στα κυρτά προβλήματα, το γράφημα της αντικειμενικής συνάρτησης και το εφικτό σύνολο είναι κυρτά (όπου ένα κυρτό σύνολο, εάν μια γραμμή που ενώνει δύο σημεία του συνόλου που περιέχεται στο σετ). Μια άλλη ειδική περίπτωση είναι ο τετραγωνικός προγραμματισμός, στον οποίο οι περιορισμοί είναι γραμμικοί, αλλά η αντικειμενική συνάρτηση είναι τετραγωνική. Δηλαδή, περιέχει όρους που είναι πολλαπλάσια του προϊόντος δύο συστατικών του x . (για παράδειγμα, η συνάρτηση $3(Q_1)^2 + 4Q_1Q_2 + 2(Q_2)^2$ είναι μια τετραγωνική συνάρτηση του Q_1 και του Q_2 .)

Ένας άλλος χρήσιμος τρόπος για να ταξινομηθούν μη γραμμικά προβλήματα είναι σύμφωνα

με τον αριθμό των μεταβλητών (δηλαδή, συστατικά του x). Ένα πρόβλημα λέγεται ότι είναι "μεγάλο" αν έχει περισσότερες από χίλιες μεταβλητές, αν και το κατώφλι του συνεχώς αυξάνεται καθώς οι υπολογιστές γίνονται πιο ισχυροί. Μια άλλη χρήσιμη διάκριση είναι μεταξύ των προβλημάτων που είναι υπολογιστικά "δύσκολα" για την αξιολόγηση και εκείνων που είναι σχετικά εύκολα, όπως συμβαίνει με τον γραμμικό προγραμματισμό.

Οι μη γραμμικοί αλγόριθμοι προγραμματισμού συνήθως προχωρούν κάνοντας μια σειρά από διαδικασίες του μεταβλητού διανύσματος x (γνωστές ως επαναλήψεις και διακρίνονται από τις υπερ-γραφές $Q_1, Q_2, Q_1, \dots$) με στόχο να προσδιορίσουν τελικά τη βέλτιστη τιμή του x . Συχνά, δεν είναι πρακτικό να προσδιορίσει την καθολικά βέλτιστη τιμή του x . Σε αυτές τις περιπτώσεις, πρέπει κάποιος να συμβιβαστεί με μια τοπική βέλτιστη — την καλύτερη τιμή σε κάποια περιοχή των εφικτών λύσεων. Κάθε επανάληψη επιλέγεται με βάση τις γνώσεις σχετικά με τον περιορισμό και τις αντικειμενικές συναρτήσεις που συγκεντρώνονται σε προηγούμενες επαναλήψεις.

Οι περισσότεροι μη γραμμικοί αλγόριθμοι προγραμματισμού στοχεύουν σε μια συγκεκριμένη υποκατηγορία προβλημάτων. Για παράδειγμα, ορισμένοι αλγόριθμοι στοχεύουν ειδικά σε μεγάλα, ομαλά προβλήματα χωρίς περιορισμούς, στα οποία ο πίνακας των δεύτερων παραγώγων του $f(x)$ περιέχει λίγες μη μηδενικές καταχωρίσεις και είναι δαπανηρή η αξιολόγησή του, ενώ άλλοι αλγόριθμοι στοχεύουν ειδικά σε κυρτές θέματα προγραμματισμού, και ούτω καθεξής. Το λογισμικό για την επίλυση προβλημάτων βελτιστοποίησης είναι διαθέσιμο τόσο στο εμπόριο όσο και στο δημόσιο τομέα. Εκτός από τα προγράμματα βελτιστοποίησης υπολογιστή, υπάρχουν διαθέσιμες πολλές γλώσσες μοντελοποίησης βελτιστοποίησης που επιτρέπουν στο χρήστη να περιγράψει το πρόβλημα με διαισθητικό τρόπο, τα οποία στη συνέχεια μεταφράζονται αυτόματα στη μαθηματική μορφή που απαιτείται από το λογισμικό βελτιστοποίησης.

1.2.1 Κατηγορίες προβλημάτων βελτιστοποίησης

Τα προβλήματα βελτιστοποίησης μπορούν να ταξινομηθούν ανάλογα με τη δυσκολία που συνδέεται με τον υπολογισμό της βέλτιστης λύσης μεταξύ όλων των διαθέσιμων και εφικτών λύσεων. Μπορούν να κατηγοριοποιηθούν σε δύο περαιτέρω ομάδες με βάση το αν μπορούν να διατυπωθούν χρησιμοποιώντας μεταβλητές που λαμβάνουν τιμές από ένα διακριτό ή συνεχές σύνολο. Ιδανικά, η διακριτή βελτιστοποίηση είναι ένα πρόβλημα βελτιστοποίησης που έχει διακριτή μεταβλητή και σε ένα τέτοιο πρόβλημα, η εστίαση τοποθετείται στην εύρεση ενός αντικειμένου όπως μια μετάλλαξη, ένα γράφημα ή ένας αχέραιος από ένα άπειρο ή πεπερασμένο σύνολο. Τα προβλήματα που σχετίζονται με τις συνεχείς μεταβλητές είναι προβλήματα πολύτροπα και περιορισμένα. Πολλά προβλήματα του πραγματικού κόσμου ασχολούνται με τη μεγιστοποίηση ή την ελαχιστοποίηση κάποιας ποσότητας, προκειμένου να βελτιστοποιηθεί κάποιο αποτέλεσμα. Για να βρούμε τις καλύτερες λύσεις, ο υπολογισμός χρησιμεύει ως το κύριο εργαλείο για πρακτικά προβλήματα. Συνήθως, τα βήματα που εμπλέκονται στην επίλυση προβλημάτων βελτιστοποίησης είναι: σχεδίαση ενός διαγράμματος που απεικονίζει το σενάριο, αντιστοίχιση συμβόλων στο διάγραμμα, διεξαγωγή ανάλυσης του διαγράμματος που σχετίζεται με γνωστές και άγνωστες μεταβλητές και χρήση λογισμικού για την εύρεση ακραίων τιμών.

Η ερευνητική κοινότητα έχει δείξει μεγάλο ενδιαφέρον για την επίλυση απτών προβλημάτων του πραγματικού κόσμου μέσω μεθόδων που εφαρμόζονται σε προβλήματα συνδυαστικής βελτιστοποίησης (CO). Πρόσφατα, πολλοί συγγραφείς προσπαθούν ενεργά να ενισχύσουν τις συμβατικές μεθόδους βελτιστοποίησης εισάγοντας αρχές και τεχνικές που προέρχονται από μη συμβατικές μεθόδους υπολογισμού με την ελπίδα ότι θα αποδειχθούν καλύτερες από τις παραδοσιακές προσεγγίσεις. Για παράδειγμα, οι Σαντίπ κ.α. (Sandip et. al.) [46] πρότειναν αρκετές

νέες τεχνικές, οι οποίες χρησιμοποιούσαν τη βελτιστοποίηση της αποικίας μυρμηγκιών με κβαντική έμπνευση, τη διαφορική εξέλιξη της κβαντικής έμπνευσης και τη βελτιστοποίηση κβαντικών σωματιδίων, αντίστοιχα, για πολυεπίπεδο χρώμα οριακής τιμής εικόνας. Αυτές οι τεχνικές βρίσκουν βέλτιστες οριακές τιμές σε διαφορετικά επίπεδα οριακής τιμής για έγχρωμες εικόνες.

Ένας νέος αλγόριθμος της κοινωνικής εξέλιξης που εμπνέεται από την κβαντική ανάπτυξη (QSE) προτάθηκε από τον υβριδισμό ενός γνωστού αλγορίθμου κοινωνικής εξέλιξης με μια αναδυόμενη κβαντική εμπνευσμένη εξελικτική. Ο προτεινόμενος αλγόριθμος QSE εφαρμόστηκε στο πρόβλημα του σακιδίου 0 – 1 και η απόδοση του αλγορίθμου συγκρίθηκε με διάφορες εξελικτικές και κβαντικές παραλλαγές εξελικτικών αλγοριθμικών παραλλαγών. Οι Γουεϊ Φανγκ κ.α. (Wei Fang et al.) [52] πρότειναν μια αποκεντρωμένη μορφή κβαντικής βελτιστοποίησης των σωματιδίων με έναν κυτταρικό δομημένο πληθυσμό για τη διατήρηση της ποικιλομορφίας του πληθυσμού και την εξισορρόπηση της καθολικής και τοπικής αναζήτησης. Ο Ζενγκ (Zeng) διεξήγαγε μια ενδιαφέρουσα μελέτη εφαρμόζοντας ένα νέο υβρίδιο κβαντικού, εμπνευσμένου εξελικτικού αλγορίθμου σε ένα πρόβλημα προγραμματισμού ροής καταστημάτων. Πρότεινε μια απλή μέθοδο αναπαράστασης για τον προσδιορισμό της ακολουθίας θέσεων εργασίας στο πρόβλημα του προγραμματισμού της ροής παραγωγής, με βάση το εύρος πιθανότητας των qubits. Οι Λου κ.α. (Lu et al.) [119] σχεδίασαν έναν αλγόριθμο αναζήτησης χώρου με κβαντική έμπνευση, προκειμένου να λύσουν αριθμητικά προβλήματα βελτιστοποίησης. Στον αλγόριθμο τους, η εφικτή λύση αποσυντίθεται σε περιφέρειες από την άποψη της κβαντικής εκπροσώπησης. Η αναζήτηση εξελίσσεται από τη μια γενιά στην άλλη, ενώ τα κβαντικά bits εξελίσσονται σταδιακά για να αυξήσουν την πιθανότητα επιλογής περιοχής. Οι Γου κ.α. (Wu et al.) [198] πρότειναν μια νέα προσέγγιση με τη χρήση ενός κβαντικού αλγορίθμου με βάση τις αρχές της θεωρίας παιγνίων. Ειδικότερα, μείωσαν το πρόβλημα που μελετούσαν για την επιλογή στρατηγικών σε εξελικτικά παιχνίδια.

Τα κβαντικά παιχνίδια και οι στρατηγικές τους φαίνονται πολύ υποσχόμενες, προσφέροντας βελτιωμένες δυνατότητες ξεπερνώντας τις κλασικές. Οι λύσεις που βασίζονται στην μεθοδολογία μεταβλητής αναζήτησης γειτονιάς (VNS) έχουν εφαρμοστεί σε προβλήματα σχεδιασμού διαδρομής. Ο Τσε κ.α. (Sze et al.) [181] πρότεινε έναν υβριδικό αλγόριθμο προσαρμογής της γειτονικής μεταβλητής για την επίλυση του προβλήματος δρομολόγησης του οχήματος (VRP). Για την αντιμετώπιση του προβλήματος VRP από τον Ντεφριν (Defryn) [44] αναπτύχθηκε μία μεθυστική VNS διαδικασία δύο επιπέδων. Ο Παπαλίτσας κ.α. (Papalitsas et. al.) [151] πρότεινε μια προσέγγιση της γενικής αναζήτησης μεταβλητής γειτονιάς (GVNS) για το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα και μια κβαντικά εμπνευσμένη GVNS (qGVNS) για την επίλυση του πλανόδιου πωλητή με χρονικά παράθυρα. Επιπλέον, η προτεινόμενη μέθοδος μπορεί επίσης να εφαρμοστεί σε άλλες κατηγορίες προβλημάτων από τον πραγματικό κόσμο, όπως η βελτιστοποίηση για τον εντοπισμό σε νοσοκομεία, έξυπνες πόλεις και έξυπνα συστήματα στάθμευσης.

Συγκεντρωτικά, οι διαφορετικές κατηγορίες βελτιστοποίησης που μπορούν να προκύψουν ανάλογα με τα δεδομένα του προβλήματος παρουσιάζονται ως εξής:

- Συνεχής βελτιστοποίηση έναντι διακριτής βελτιστοποίησης: Ορισμένα μοντέλα έχουν νόημα μόνο εάν οι μεταβλητές λαμβάνουν τιμές από ένα διακριτό σύνολο, συχνά ένα υποσύνολο ακεραίων, ενώ άλλα μοντέλα περιέχουν μεταβλητές που μπορούν να αναλάβουν οποιαδήποτε πραγματική τιμή. Τα μοντέλα με διακριτές μεταβλητές είναι διακριτά προβλήματα βελτιστοποίησης. Μοντέλα με συνεχείς μεταβλητές είναι συνεχή προβλήματα βελτιστοποίησης. Τα συνεχή προβλήματα βελτιστοποίησης τείνουν να λύνονται ευκολότερα από τα διακριτά προβλήματα βελτιστοποίησης. Η ομοιότητα των συναρτήσεων σημαίνει ότι η αντικειμενική συνάρτηση και οι τιμές συνάρτησης περιορισμού σε ένα σημείο x μπορούν να

χρησιμοποιη-θούν για να συμπεράνουν πληροφορίες σχετικά με σημεία σε μια γειτονιά του x . Ωστόσο, οι βελτιώσεις σε αλγόριθμους σε συνδυασμό με την πρόοδο στην επιστήμη της πληροφορικής έχουν αυξήσει δραματικά το μέγεθος και την πολυπλοκότητα των διακριτών προβλημάτων βελτιστοποίησης που μπορούν να επιλυθούν αποτελεσματικά. Οι συνεχείς αλγόριθμοι βελτιστοποίησης είναι σημαντικοί στη διακριτή βελτιστοποίηση, επειδή πολλοί αλγόριθμοι διακριτής βελτιστοποίησης δημιουργούν μια ακολουθία συνεχών δευτερευόντων προβλημάτων.

- Μη περιορισμένη βελτιστοποίηση έναντι περιορισμένης βελτιστοποίησης: Μια άλλη σημαντική διάκριση είναι μεταξύ των προβλημάτων στα οποία δεν υπάρχουν περιορισμοί στις μεταβλητές και τα προβλήματα στα οποία υπάρχουν περιορισμοί στις μεταβλητές. Τα προβλήματα βελτιστοποίησης χωρίς περιορισμό προκύπτουν άμεσα σε πολλές πρακτικές εφαρμογές. Προκύπτουν επίσης από την αναδιατύπωση των περιορισμένων προβλημάτων βελτιστοποίησης, στα οποία οι περιορισμοί αντικαθίστανται από έναν όρο ποινής στην αντικειμενική συνάρτηση. Τα προβλήματα περιορισμένης βελτιστοποίησης προκύπτουν από εφαρμογές στις οποίες υπάρχουν σαφείς περιορισμοί στις μεταβλητές. Οι περιορισμοί στις μεταβλητές μπορεί να ποικίλλουν ευρέως από απλά όρια σε συστήματα εξισώσεων και ανισοτήτων που μοντελοποιούν σύνθετες σχέσεις μεταξύ των μεταβλητών. Τα προβλήματα περιορισμένης βελτιστοποίησης μπορούν να επιλυθούν ανάλογα με τη φύση των περιορισμών (π.χ. γραμμική, μη γραμμική, κυρτή) και την ομαλότητα των συναρτήσεων (π.χ. διαφορικών ή μη διαφορικών).
- Κανένας, ένας ή πολλοί στόχοι: Τα περισσότερα προβλήματα βελτιστοποίησης έχουν μία μόνο αντικειμενική συνάρτηση, ωστόσο, υπάρχουν ενδιαφέρουσες περιπτώσεις όταν τα προβλήματα βελτιστοποίησης δεν έχουν αντικειμενική συνάρτηση ή πολλές αντικειμενικές συναρτήσεις. Τα προβλήματα σκοπιμότητας είναι προβλήματα στα οποία ο στόχος είναι η εύρεση τιμών για τις μεταβλητές που ικανοποιούν τους περιορισμούς ενός μοντέλου χωρίς συγκεκριμένο στόχο για βελτιστοποίηση. Τα προβλήματα συμπληρωματικότητας είναι διάχυτα στη μηχανική και την οικονομία. Ο στόχος είναι να βρεθεί μια λύση που να ικανοποιεί τις συνθήκες συμπληρωματικότητας. Πολλαπλά αντικειμενικά προβλήματα βελτιστοποίησης ανακύπτουν σε πολλούς τομείς, όπως η μηχανική και η οικονομία, όταν πρέπει να ληφθούν βέλτιστες αποφάσεις για την παρουσία συναλλαγών μεταξύ δύο ή περισσότερων αντικρουόμενων στόχων. Για παράδειγμα, η ανάπτυξη ενός νέου στοιχείου μπορεί να περιλαμβάνει την ελαχιστοποίηση του βάρους, ενώ η μεγιστοποίηση της δύναμης ή της επιλογής ενός χαρτοφυλακίου μπορεί να περιλαμβάνει μεγιστοποίηση της αναμενόμενης απόδοσης, ελαχιστοποιώντας παράλληλα τον κίνδυνο. Στην πράξη, τα προβλήματα με πολλούς στόχους συχνά αναδιατυπώνονται ως ενιαία αντικειμενικά προβλήματα είτε με τη διαμόρφωση ενός σταθμισμένου συνδυασμού των διαφόρων στόχων είτε με την αντικατάσταση ορισμένων στόχων από περιορισμούς.
- Ντετερμινιστική βελτιστοποίηση έναντι στοχαστικής βελτιστοποίησης: Στην ντετερμινιστική βελτιστοποίηση, θεωρείται ότι τα δεδομένα για το δεδομένο πρόβλημα είναι γνωστά με ακρίβεια. Ωστόσο, για πολλά πραγματικά προβλήματα, τα δεδομένα δεν μπορούν να γίνουν γνωστά με ακρίβεια για διάφορους λόγους. Ο πρώτος λόγος οφείλεται στο απλό σφάλμα μέτρησης. Ο δεύτερος και πιο θεμελιώδης λόγος είναι ότι ορισμένα δεδομένα αντιπροσωπεύουν πληροφορίες σχετικά με το μέλλον (π.χ. ζήτηση ή τιμή προϊόντος για

μια μελλοντική χρονική περίοδο) και απλά δεν μπορούν να γίνουν γνωστές με βεβαιότητα. Στη βελτιστοποίηση υπό αβεβαιότητα, ή στοχαστική βελτιστοποίηση, η αβεβαιότητα ενσωματώνεται στο μοντέλο. Ισχυρές τεχνικές βελτιστοποίησης μπορούν να χρησιμοποιηθούν όταν οι παράμετροι είναι γνωστές μόνο εντός ορισμένων ορίων. Ο στόχος είναι να βρεθεί μια λύση που να είναι εφικτή για όλα τα δεδομένα και να είναι η βέλτιστη κατά κάποιο τρόπο. Στοχαστικά μοντέλα προγραμματισμού εκμεταλλεύονται το γεγονός ότι οι κατανομές πιθανοτήτων που διέπουν τα δεδομένα είναι γνωστές ή μπορούν να εκτιμηθούν. Ο στόχος είναι να βρεθεί κάποια μεθοδολογία που είναι εφικτή για όλες (ή σχεδόν όλες) τις πιθανές παρουσίες δεδομένων και να βελτιστοποιεί την αναμενόμενη απόδοση του μοντέλου.

1.3 Μεθευρετικές μέθοδοι

Η μεθευρετική βελτιστοποίηση επιλύει προβλήματα βελτιστοποίησης χρησιμοποιώντας μεθευρετικούς αλγόριθμους. Η βελτιστοποίηση είναι ουσιαστικά παντού, από το σχεδιασμό της μηχανικής μέχρι τα οικονομικά και από το σχεδιασμό διακοπών για τη δρομολόγηση του διαδικτύου. Καθώς τα χρήματα, οι πόροι και ο χρόνος είναι πάντα περιορισμένοι, η βέλτιστη χρησιμότητα αυτών των διαθέσιμων πόρων είναι ζωτικής σημασίας. Οι περισσότερες βελτιστοποιήσεις πραγματικού κόσμου είναι εξαιρετικά μη γραμμικές και πολυτροπικές, κάτω από διάφορους σύνθετους περιορισμούς. Οι διαφορετικοί στόχοι συχνά συγκρούονται. Ακόμη και για έναν μόνο στόχο, μερικές φορές, οι βέλτιστες λύσεις μπορεί να μην υπάρχουν καθόλου. Γενικά, η εξεύρεση μιας βέλτιστης λύσης ή ακόμη και μη βέλτιστων λύσεων δεν είναι εύκολη εργασία. Με την απλούστερη έννοια, η βελτιστοποίηση μπορεί να θεωρηθεί ως πρόβλημα ελαχιστοποίησης ή βελτιστοποίησης.

Για παράδειγμα, η συνάρτηση $f(x) = Q_2$ έχει ένα ελάχιστο $f_{(min)} = 0$ στο $x = 0$ σε ολόκληρο τον τομέα $-\infty < x < \infty$. Σε γενικές γραμμές, αν μια συνάρτηση είναι αρκετά απλή, μπορούμε να χρησιμοποιήσουμε την πρώτη παράγωγο $f'(x) = 0$ για να καθορίσουμε τις πιθανές θέσεις και να χρησιμοποιήσουμε τη δεύτερη παράγωγο $f''(x)$ για να επιβεβαιώσουμε αν το διάλυμα είναι μέγιστο ή ελάχιστο. Ωστόσο, για μη γραμμικές, πολύτροπες, πολυμεταβλητές συναρτήσεις, αυτό δεν είναι ένα εύκολο έργο. Επιπλέον, ορισμένες συναρτήσεις μπορεί να έχουν ασυνέχειες και, συνεπώς, οι παράγωγοι πληροφορίες δεν είναι εύκολο να αποκτηθούν. Αυτό μπορεί να προκαλέσει διάφορες προκλήσεις σε πολλές παραδοσιακές μεθόδους αναζήτησης, όπως η αναρρίχηση λόφου.

Γενικά, ένα πρόβλημα βελτιστοποίησης μπορεί να γραφτεί ως η ελαχιστοποίηση του $f1_{(x)}, \dots, fi_{(x)}, \dots, fI_{(x)}$, $x = (Q_1, \dots, Y_b)$, (2) υπό την επιφύλαξη $(x) = 0, (j = 1, 2, \dots, J) GK_{(x)} \leq 0, (k = 1, 2, \dots, K)$, όπου η $f_1, \dots, f_i$ είναι οι στόχοι, ενώ η Q_T και η GK είναι οι περιορισμοί ισότητας και ανισότητας, αντίστοιχα. Στην περίπτωση που $I = 1$, ονομάζεται μοναδιαία βελτιστοποίηση αντικειμενικής συνάρτησης. Όταν $I \geq 2$, γίνεται ένα πολυαντικειμενικό πρόβλημα του οποίου η στρατηγική λύση είναι διαφορετική από εκείνες για έναν ενιαίο στόχο. Σε γενικές γραμμές, όλες οι συναρτήσεις f_i , ταυτόχρονα και η GK είναι μη γραμμική. Στην ειδική περίπτωση, όταν όλες αυτές οι συναρτήσεις είναι γραμμικές, το πρόβλημα βελτιστοποίησης γίνεται ένα γραμμικό πρόβλημα προγραμματισμού το οποίο μπορεί να λυθεί με τη χρήση της μεθόδου **simplex**. Η μεθευρετική βελτιστοποίηση αφορά πιο γενικευμένα, μη γραμμικά προβλήματα βελτιστοποίησης. Αξίζει να επισημανθεί ότι το παραπάνω πρόβλημα ελαχιστοποίησης μπορεί επίσης να διατυπωθεί ως πρόβλημα μεγιστοποίησης εάν η f_i αντικατασταθεί με $-f_i$. Ομοίως, οι περιορισμοί ανισότητας GK μπορούν επίσης να γραφτούν αντίθετα, αν η GK αντικατασταθεί από $-GK$.

Προφανώς, η απλούστερη περίπτωση βελτιστοποίησης είναι η βελτιστοποίηση της συνάρτησης χωρίς περιορισμούς. Για παράδειγμα, οι συναρτήσεις πολλαπλών μεταβλητών χρησιμοποιούνται συχνά για την επικύρωση νέων αλγορίθμων βελτιστοποίησης και ένα καλό παράδειγμα είναι η συνάρτηση του Οκλεί (Ackley), η οποία έχει ένα συνολικό ελάχιστο $f_{min} = 0$ στο $(0, 0)$. Για να επιλυθεί το πρόβλημα βελτιστοποίησης, απαιτούνται αποτελεσματικοί αλγόριθμοι αναζήτησης ή βελτιστοποίησης. Υπάρχουν πολλοί αλγόριθμοι βελτιστοποίησης που μπορούν να ταξινομηθούν με πολλούς τρόπους, ανάλογα με την εξειδίκευση και τα χαρακτηριστικά τους. Εάν το παράγωγο μιας συνάρτησης είναι η εστίαση, η βελτιστοποίηση μπορεί να ταξινομηθεί σε αλγόριθμους που βασίζονται σε διαβαθμίσεις και σε αλγόριθμους χωρίς παράγωγα. Οι αλγόριθμοι χωρίς παράγωγα δεν χρησιμοποιούν καμία παράγωγη πληροφορία, αλλά τις τιμές της ίδιας της συνάρτησης.

Από μια διαφορετική προοπτική, οι αλγόριθμοι βελτιστοποίησης μπορούν να ταξινομηθούν σε τροχιά και με βάση τον πληθυσμό. Ένας αλγόριθμος που βασίζεται σε τροχιά συνήθως χρησιμοποιεί έναν μόνο παράγοντα ή μία λύση κάθε φορά, η οποία θα εντοπίσει μια διαδρομή καθώς συνεχίζονται οι επαναλήψεις. Η αναρρίχηση στο λόφο είναι βασισμένη στην τροχιά, και συνδέει το σημείο εκκίνησης με το τελικό σημείο μέσω ενός κομματιού του μονοπατιού.

Ένα άλλο σημαντικό παράδειγμα είναι η προσομοίωση ανόπτησης, η οποία είναι ένας ευρέως χρησιμοποιούμενος μεθευρετικός αλγόριθμος. Από την άλλη πλευρά, με βάση τον πληθυσμό αλγορίθμων όπως η βελτιστοποίηση του σμήνους σωματιδίων χρησιμοποιούν πολλαπλούς παράγοντες που θα αλληλεπιδρούν και να εντοπίζουν πολλαπλά μονοπάτια. Αλγόριθμοι βελτιστοποίησης μπορεί επίσης να χαρακτηριστούν ως ντετερμινιστικοί ή στοχαστικοί. Εάν ένας αλγόριθμος λειτουργεί με μηχανικό ντετερμινιστικό τρόπο χωρίς τυχαίο χαρακτήρα, ονομάζεται ντετερμινιστικός. Ένας τέτοιος αλγόριθμος θα φτάσει στην ίδια τελική λύση αν αρχίσουμε με το ίδιο αρχικό σημείο. Η αναρρίχηση και η απλή κάθοδος είναι καλά παραδείγματα ντετερμινιστικών αλγορίθμων. Από την άλλη πλευρά, αν υπάρχει κάποια τυχαιότητα στον αλγόριθμο, ο αλγόριθμος θα φτάσει συνήθως σε ένα διαφορετικό σημείο κάθε φορά που εκτελείται, ακόμα και αν χρησιμοποιείται το ίδιο σημείο εκκίνησης κάθε φορά. Οι γενετικοί αλγόριθμοι είναι καλά παραδείγματα στοχαστικών αλγορίθμων.

Η δυνατότητα αναζήτησης μπορεί επίσης να αποτελέσει βάση για την ταξινόμηση αλγορίθμων. Σε αυτή την περίπτωση, οι αλγόριθμοι μπορούν να χωριστούν σε τοπικούς και ολικούς αλγόριθμους αναζήτησης. Οι τοπικοί αλγόριθμοι αναζήτησης συνήθως συγκλίνουν προς μια τοπική βέλτιστη, όχι απαραίτητα (συχνά όχι) στην ολική βέλτιστη και ένας τέτοιος αλγόριθμος είναι συχνά ντετερμινιστικός και δεν έχει καμία ικανότητα διαφυγής από το τοπικό ελάχιστο. Μια απλή αναρρίχηση σε λόφο είναι ένα τέτοιο παράδειγμα. Από την άλλη πλευρά, είναι η ολική βελτιστοποίηση, όπου οι αλγόριθμοι τοπικής αναζήτησης δεν είναι κατάλληλοι και θα πρέπει να χρησιμοποιούνται καθολικοί αλγόριθμοι αναζήτησης.

Οι σύγχρονοι μεθευρετικοί αλγόριθμοι στις περισσότερες περιπτώσεις τείνουν να είναι κατάλληλοι για καθολική βελτιστοποίηση, αν και δεν είναι πάντα επιτυχημένοι ή αποτελεσματικοί. Μια απλή στρατηγική, όπως η αναρρίχηση σε λόφο με τυχαίες επανεκκινήσεις, μπορεί να μετατρέψει έναν τοπικό αλγόριθμο αναζήτησης σε έναν αλγόριθμο με δυνατότητα καθολικής αναζήτησης. Στην ουσία, η τυχαιότητα είναι ένα αποτελεσματικό συστατικό για τους αλγόριθμους καθολικής αναζήτησης. Προφανώς, οι αλγόριθμοι μπορεί να μην ταιριάζουν ακριβώς σε κάθε κατηγορία. Μπορεί να είναι ένας μεικτός τύπος ή υβρίδιο, το οποίο χρησιμοποιεί κάποιο συνδυασμό των ντετερμινιστικών συστατικών με τυχαιότητα, ή συνδυάζει έναν αλγόριθμο με έναν άλλο, ώστε να σχεδιαστούν πιο αποτελεσματικοί αλγόριθμοι.

Οι αλγόριθμοι με στοχαστικά συστατικά αναφέρθηκαν συχνά ως ευρετικοί στο παρελθόν, αν και η πρόσφατη βιβλιογραφία τείνει να αναφέρεται σε αυτούς ως μεθευρετικούς. Η λέξη

"μεθευρετική" επινοήθηκε από τον Φρεντ Γκλόβερ και μπορεί να θεωρηθεί ως μια κύρια στρατηγική που καθοδηγεί και τροποποιεί άλλες ευρετικές για να παράγει λύσεις πέρα από αυτές που συνήθως δημιουργούνται σε μια αναζήτηση για τοπικές βέλτιστες [73].

Επιπλέον, όλοι οι μεθευρετικοί αλγόριθμοι χρησιμοποιούν μια ορισμένη ανταλλαγή τυχαίας επιλογής και τοπικής αναζήτησης. Οι λύσεις ποιότητας σε δύσκολα προβλήματα βελτιστοποίησης μπορούν να βρεθούν σε εύλογο χρονικό διάστημα, αλλά δεν υπάρχει καμία εγγύηση ότι μπορούν να επιτευχθούν οι βέλτιστες λύσεις. Σχεδόν όλοι οι μεθευρετικοί αλγόριθμοι τείνουν να είναι κατάλληλοι για προβλήματα ολικής βελτιστοποίησης. Μια εξαιρετική ανασκόπηση διεξήχθη από τον Μπος (Bosch) [24].

Δύο σημαντικά συστατικά οποιονδήποτε μεθευρετικών αλγορίθμων είναι: η εντατικοποίηση και η διαφοροποίηση, η εκμετάλλευση και η εξερεύνηση. Διαφοροποίηση σημαίνει τη δημιουργία ποικίλων λύσεων, ώστε να εξερευνήσουμε τον χώρο αναζήτησης σε παγκόσμια κλίμακα, ενώ η εντατικοποίηση σημαίνει να εστιάσουμε την αναζήτηση σε μια τοπική περιοχή γνωρίζοντας ότι υπάρχει μια τρέχουσα καλή λύση σε αυτήν την περιοχή. Μια καλή ισορροπία μεταξύ της εντατικοποίησης και της διαφοροποίησης θα πρέπει να βρεθεί κατά την επιλογή των καλύτερων λύσεων για τη βελτίωση του ποσοστού σύγκλισης των αλγορίθμων. Η επιλογή των καλύτερων διασφαλίζει ότι οι λύσεις θα συγκλίνουν στη βέλτιστη, ενώ η διαφοροποίηση μέσω τυχαίας επιλογής επιτρέπει την αναζήτηση στο διάστημα από την τοπική βέλτιστη και, ταυτόχρονα, αυξάνει την ποικιλομορφία των λύσεων. Ένας καλός συνδυασμός αυτών των δύο κύριων συστατικών θα διασφαλίσει συνήθως ότι η ολική βέλτιστη είναι εφικτή.

Πολλές διαδικασίες επίλυσης προβλημάτων τείνουν να είναι ευρετικές σε όλη την ανθρώπινη ιστορία. Ωστόσο, η ευρετική ως επιστημονική μέθοδος βελτιστοποίησης είναι ένα σύγχρονο φαινόμενο. Από τη δεκαετία του 1940 έως το 1960, οι ευρετικές μέθοδοι έχουν χρησιμοποιηθεί σε διάφορες εφαρμογές, αλλά το πρώτο ορόσημο ήρθε με την έλευση των εξελικτικών αλγορίθμων. Το 1963 και στο τεχνικό Πανεπιστήμιο του Βερολίνου, μαθηματικοί ανέπτυξαν εξελικτικές στρατηγικές ενώ ο Φόγκελ (L. J. Fogel et al.) [60] ανέπτυξε εξελικτικό προγραμματισμό το 1966. Οι γενετικοί αλγόριθμοι αναπτύχθηκαν από τον Όλλαντ (J. Holland) [89] κατά τη δεκαετία του 1960 και του 1970, αν και το βιβλίο του για τους γενετικούς αλγόριθμους δημοσιεύθηκε το 1975. Η δεκαετία του 1980 και του 1990 ήταν η πιο συναρπαστική στιγμή για τους μεθευρετικούς αλγόριθμους. Ένα μεγάλο βήμα ήταν η ανάπτυξη προσομοιωμένης ανόπτησης (Simulated Annealing) το 1983, μια τεχνική βελτιστοποίησης και εμπνευσμένη από τη διαδικασία ανόπτησης των μετάλλων. Ένα άλλο σημαντικό βήμα ήταν η ανάπτυξη μεθευρετικών μεθόδων τεχνητών ανοσολογικών συστημάτων από τον Φάρμερ (Farmer et al.) [55] το 1986.

Το 1992, ο Μάρκο Ντορίγκο (Marco Dorigo) [48] ολοκλήρωσε τη διδακτορική του διατριβή σχετικά με τη βελτιστοποίηση και τους φυσικούς αλγόριθμους, στην οποία περιέγραψε την καινοτόμο εργασία του για τη βελτιστοποίηση της αποικίας μυρμηγκιών (beam - ACO). Αυτή η τεχνική αναζήτησης εμπνεύστηκε από τη νοημοσύνη του σμήνους των κοινωνικών μυρμηγκιών χρησιμοποιώντας φερομόνη ως χημικό αγγελιοφόρο.

Στη συνέχεια, το 1992, ο Κόζα (John R. Koza) [109] του Πανεπιστημίου Στάνφορντ δημοσίευσε ένα βιβλίο για τον γενετικό προγραμματισμό, το οποίο έθεσε τα θεμέλια μιας νέας περιοχής μηχανικής μάθησης, με επανάσταση στον προγραμματισμό των υπολογιστών. Ελαφρώς αργότερα το 1995, μια άλλη σημαντική πρόοδος ήταν η ανάπτυξη της βελτιστοποίησης του σμήνους σωματιδίων από τον Κένεντι (Kennedy) [106].

Περίπου το 1996 και αργότερα το 1997, οι Στορν και Πράις (Storn and Price) [179] [180] ανέπτυξαν το διάλυμα εξελικτικού αλγορίθμου, που ονομάζεται διαφορική εξέλιξη. Αυτός ο αλγόριθμος αποδείχθηκε πιο αποτελεσματικός από τους γενετικούς αλγόριθμους σε πολλές εφαρ-

μογές.

Το 2004, ο Νακράνι (S. Nakrani) [135] πρότεινε τον αλγόριθμο μέλισσας και την εφαρμογή του για τη βελτιστοποίηση των κέντρων φιλοξενίας στο διαδίκτυο, το οποίο ακολουθήθηκε από την ανάπτυξη ενός νέου αλγορίθμου μέλισσας από τον Φαμ (D. T. Pham et al.) [157] το 2005 και την αποικία τεχνητών μελισσών (ABC) από τον Καραμπόγκα (D. Karaboga) [96] το 2005.

1.4 Μεθευρετικοί αλγόριθμοι για βελτιστοποίηση

- Προσομοιωμένη ανόπτηση:

Η προσομοιωμένη ανόπτηση βασίζεται στην επεξεργασία μετάλλων ανόπτησης. Σε αντίθεση με άλλες ντετερμινιστικές μεθόδους αναζήτησης, το κύριο πλεονέκτημα της προσομοιωμένης ανόπτησης είναι η ικανότητά της να αποφεύγει να παγιδευτεί σε τοπικά βέλτιστα. Μεταφορικά μιλώντας, αυτό ισοδυναμεί με την απόθεση σφαιρών αναπηδώντας πάνω από ένα τοπίο και καθώς οι μπάλες αναπηδούν και χάνουν την ενέργεια, εγκαθίστανται σε ορισμένα τοπικά ελάχιστα. Εάν οι μπάλες επιτρέπεται να αναπηδούν αρκετό καιρό και να χάσουν ενέργεια αρκετά αργά, μερικές από τις μπάλες θα πέσουν τελικά στις καθολικά χαμηλότερες θέσεις, ως εκ τούτου θα επιτευχθεί το ολικό ελάχιστο. Ουσιαστικά, η προσομοιωμένη ανόπτηση είναι μια αναζήτηση κατά μήκος μιας αλυσίδας Μάρκοβ (Markov Chain), η οποία γίνεται υπό τις κατάλληλες συνθήκες.

Στην προσομοιωμένη ανόπτηση, οι πραγματικές κινήσεις αναζήτησης εντοπίζουν κομμάτια διαδρομής. Με κάθε κίνηση, αξιολογείται μια πιθανότητα αποδοχής, η οποία όχι μόνο αποδέχεται αλλαγές που βελτιώνουν την αντικειμενική συνάρτηση (για ένα πρόβλημα ελαχιστοποίησης, μια χαμηλότερη αντικειμενική τιμή), αλλά διατηρεί επίσης ορισμένες αλλαγές που δεν βελτιώνουν τον στόχο (μια μεγαλύτερη αντικειμενική τιμή). Αυτή η πιθανότητα μετάβασης βασίζεται στην κατανομή Μπόλτςμαν (Boltzmann) στη στατιστική μηχανική. Η αλλαγή στη συνάρτηση στόχου, $\Delta\Phi$, μπορεί να σχετίζεται με το δε στον ακόλουθο τύπο, $\gamma = \delta\phi$, όπου γ είναι μια πραγματική σταθερά (συνήθως, $\gamma = 1$ για απλότητα). Είναι σαφές ότι $p \rightarrow 0$ ως $T \rightarrow 0$. Δηλαδή, το σύστημα γίνεται σχεδόν μια μέθοδος αναρρίχησης σε πολύ χαμηλή θερμοκρασία.

Ο τρόπος για τον έλεγχο των διακυμάνσεων της θερμοκρασίας ουσιαστικά ελέγχει πώς συμπεριφέρεται ο αλγόριθμος και την αποτελεσματικότητά του. Υπάρχουν πολλοί τρόποι για να ελέγξουμε το ρυθμό ψύξης ή τη μείωση της θερμοκρασίας. Δύο συχνά χρησιμοποιούμενα χρονοδιαγράμματα ανόπτησης (ή προγράμματα ψύξης) είναι γραμμικά και γεωμετρικά. Από την άλλη πλευρά, ένα γεωμετρικό πρόγραμμα ψύξης μειώνει ουσιαστικά τη θερμοκρασία με έναν συντελεστή ψύξης $0 < \alpha < 1$, έτσι ώστε το T να αντικατασταθεί από την αT $T(t) = T_0 \alpha t, t = 1, 2, \dots, TF$. Το πλεονέκτημα της δεύτερης μεθόδου είναι ότι $\tau \rightarrow 0$ όταν $t \rightarrow \infty$, και έτσι δεν υπάρχει λόγος να προσδιοριστεί ο μέγιστος αριθμός επαναλήψεων. Για το λόγο αυτό, το γεωμετρικό πρόγραμμα ψύξης χρησιμοποιείται ευρύτερα. Η διαδικασία ψύξης θα πρέπει να είναι αρκετά αργή για να επιτρέψει το σύστημα να σταθεροποιηθεί. Στην πράξη, χρησιμοποιείται συνήθως $\alpha \in [0.7, 0.99]$.

Επιπλέον, για μια δεδομένη θερμοκρασία, απαιτούνται πολλαπλές συγκρίσεις της αντικειμενικής συνάρτησης. Εάν υπάρχουν πολύ λίγες συγκρίσεις, υπάρχει κίνδυνος το σύστημα να μην σταθεροποιηθεί και στη συνέχεια να μην συγκλίνει προς την ολική βέλτιστη. Αντιστρόφως, πάρα πολλές συγκρίσεις είναι χρονοβόρες και το σύστημα συνήθως συγκλίνει πολύ αργά, καθώς ο αριθμός των επαναλήψεων για την επίτευξη σταθερότητας μπορεί να είναι εκθετικός με το μέγεθος του προβλήματος. Ως εκ τούτου, υπάρχει μια λεπτή ισορροπία μεταξύ του αριθμού των συγκρίσεων και της ποιότητας της λύσης. Μπορούμε είτε να εκτελέσουμε πολλές συγκρίσεις σε μερικά επίπεδα θερμοκρασίας ή να κάνουμε μόνο μερικές συγκρίσεις σε πολλά επίπεδα

θερμοκρασί-ας.

Υπάρχουν δύο τρόποι για να ορίσουμε τον αριθμό των επαναλήψεων: σταθερές ή μεταβλητές. Ο πρώτος χρησιμοποιεί ένα σταθερό αριθμό επαναλήψεων σε κάθε θερμοκρασία, ενώ ο δεύτερος σκοπεύει να αυξήσει τον αριθμό των επαναλήψεων σε χαμηλότερες θερμοκρασίες για να διερευνήσει πλήρως τα τοπικά ελάχιστα.

- **Γενετικοί αλγόριθμοι:**

Οι γενετικοί αλγόριθμοι (GAs) είναι πιθανώς οι πιο δημοφιλείς εξελικτικοί αλγόριθμοι με ένα ευρύ φάσμα εφαρμογών. Η συντριπτική πλειονότητα των γνωστών προβλημάτων βελτιστοποίησης έχουν λυθεί με γενετικούς αλγόριθμους. Επιπλέον, οι γενετικοί αλγόριθμοι βασίζονται στον πληθυσμό και πολλοί σύγχρονοι εξελικτικοί αλγόριθμοι βασίζονται άμεσα ή έχουν ισχυρές ομοιότητες με γενετικούς αλγόριθμους. Οι γενετικοί αλγόριθμοι, οι οποίοι αναπτύχθηκαν από τον Χόλαντ (Holland) [89] και τους συνεργάτες του κατά τη δεκαετία του 1960 και του 1970, αποτελούν μοντέλο ή αφαίρεση της βιολογικής εξέλιξης με βάση τη θεωρία του Καρόλου Δαρβίνου για τη φυσική επιλογή. Η Ολλανδία ήταν η πρώτη που χρησιμοποίησε διασταυρωμένη, επανασυνδυασμένη μετάλλαξη και επιλογή στη μελέτη των προσαρμοστικών και τεχνητών συστημάτων. Αυτοί οι γενετικοί φορείς είναι τα βασικά συστατικά των γενετικών αλγορίθμων ως στρατηγική επίλυσης προβλημάτων. Έκτοτε, πολλές παραλλαγές γενετικών αλγορίθμων έχουν αναπτυχθεί και εφαρμοστεί σε ένα ευρύ φάσμα προβλημάτων βελτιστοποίησης, από το χρωματισμό γραφημάτων μέχρι την αναγνώριση μοτίβων, από διακριτά συστήματα (όπως το πρόβλημα του πλανόδιου πωλητή) σε συνεχή συστήματα (π.χ. τον αποτελεσματικό σχεδιασμό της αεροδιαστημικής μηχανικής) και από τις χρηματοπιστωτικές αγορές έως την πολυαντικειμενική μηχανική βελτιστοποίηση.

Η ουσία των γενετικών αλγορίθμων περιλαμβάνει την κωδικοποίηση των λύσεων ως πίνακες **bits** ή χορδές χαρακτήρων (χρωμοσώματα), τη χειραγώγηση αυτών των χορδών από γενετικούς τελεστές και μια επιλογή με βάση την καταλληλότητα τους για να βρουν μια λύση σε ένα δεδομένο πρόβλημα. Αυτό γίνεται συχνά με την ακόλουθη διαδικασία:

1. Ορισμός ενός συστήματος κωδικοποίησης
2. Ορισμός της συνάρτησης καταλληλότητας ή του κριτηρίου επιλογής
3. Δημιουργία πληθυσμού χρωμοσωμάτων
4. Αξιολόγηση της καταλληλότητας κάθε χρωμοσώματος στον πληθυσμό
5. Δημιουργία ενός νέου πληθυσμού με την εκτέλεση αναλογικής επιλογής, διασταύρωσης και μετάλλαξης της καταλληλότητας
6. Αντικατάσταση του παλιού πληθυσμού από τον νέο

Τα βήματα 4), 5) και 6) επαναλαμβάνονται για πολλές γενιές. Στο τέλος, το καλύτερο χρωμόσωμα αποκωδικοποιείται για να αποκτήσει μια λύση στο πρόβλημα. Κάθε επανάληψη, η οποία οδηγεί σε έναν νέο πληθυσμό, ονομάζεται γενιά. Οι συμβολοσειρές χαρακτήρων σταθερού μήκους χρησιμοποιούνται στους περισσότερους γενετικούς αλγόριθμους σε κάθε γενιά, αν και υπάρχει ουσιαστική έρευνα σε συμβολοσειρές μεταβλητού μήκους και δομές κωδικοποίησης. Η κωδικοποίηση της

αντικειμενικής συνάρτησης είναι συνήθως με τη μορφή δυαδικών συστοιχιών ή πινάκων πραγματικών τιμών σε προσαρμοστικούς γενετικούς αλγόριθμους. Ένα σημαντικό ζήτημα είναι η διατύπωση ή η επιλογή μιας κατάλληλης συνάρτησης καταλληλότητας που καθορίζει το κριτήριο επιλογής σε ένα συγκεκριμένο πρόβλημα. Για την ελαχιστοποίηση μιας συνάρτησης με τη χρήση γενετικών αλγορίθμων, ένας απλός τρόπος για να κατασκευάσουμε μια *fitness* συνάρτηση είναι να χρησιμοποιήσουμε την απλούστερη μορφή $F = Ay$ με μια μεγάλη σταθερά (αν και ένα $= 0$ θα κάνει αν η φυσική κατάσταση δεν χρειάζεται να είναι μη αρνητική) και $y = f_{(x)}$. Έτσι, ο στόχος είναι η μεγιστοποίηση της συνάρτησης καταλληλότητας και, στη συνέχεια, η ελαχιστοποίηση της αντικειμενικής συνάρτησης $f_{(x)}$.

Ωστόσο, υπάρχουν πολλοί διαφορετικοί τρόποι για να ορίσουμε μια *fitness* συνάρτηση. Για παράδειγμα, μπορούμε να αναθέσουμε μια ατομική καταλληλότητα σε σχέση με το σύνολο του πληθυσμού (X_I) = $f_{\Sigma N_i} = 1F$, όπου είναι η φαινοτυπική αξία των μεμονωμένων i , και N είναι το μέγεθος των πληθυσμών. Θα πρέπει να βεβαιωθούμε ότι τα χρωμοσώματα με υψηλότερη καταλληλότητα επιλέγονται συχνότερα από εκείνα με χαμηλότερη φυσική κατάσταση. Οι κακές *fitness* συναρτήσεις μπορεί να οδηγήσουν σε κακές λύσεις. Ένα άλλο σημαντικό ζήτημα είναι η επιλογή διαφόρων τιμών παραμέτρων. Ο υπολογισμός της πιθανότητας διασταύρωσης είναι συνήθως πολύ υψηλός. Από την άλλη πλευρά, η πιθανότητα μετάλλαξης PM είναι συνήθως μικρή (συνήθως, στο διάστημα $[0.001, 0.05]$). Αν ο υπολογιστής είναι πολύ μικρής υπολογιστικής δυνατότητας, τότε η διασταύρωση εφαρμόζεται σπάνια, κάτι που δεν είναι επιθυμητό. Εάν η πιθανότητα μετάλλαξης είναι πολύ υψηλή, ο αλγόριθμος θα μπορούσε ακόμα να «μεταπηδήσει» ακόμα και αν η βέλτιστη λύση είναι κοντά. Η επιλογή του ισχυρότερου πραγματοποιείται σύμφωνα με την καταλληλότητα των χρωμοσωμάτων.

Μερικές φορές, προκειμένου να διασφαλίσουμε ότι τα καλύτερα χρωμοσώματα παραμένουν στον πληθυσμό, μεταφέρονται στην επόμενη γενιά χωρίς μεγάλη αλλαγή, η οποία ονομάζεται ελιτισμός. Ένα κατάλληλο κριτήριο για την επιλογή των καλύτερων χρωμοσωμάτων είναι επίσης σημαντικό, επειδή καθορίζει πώς τα χρωμοσώματα με υψηλότερη φυσική κατάσταση διατηρούνται και μεταφέρονται στην επόμενη γενιά. Αυτό γίνεται συχνά σε συνδυασμό με μια ορισμένη μορφή ελιτισμού. Η βασική μορφή είναι η επιλογή του καλύτερου χρωμοσώματος (σε κάθε γενιά) που θα μεταφερθεί στη νέα γενιά χωρίς να τροποποιηθεί από τους γενετικούς φορείς. Αυτό διασφαλίζει ότι μια καλή λύση επιτυγχάνεται ταχύτερα.

Άλλα ζητήματα περιλαμβάνουν πολλαπλές τοποθεσίες για μετάλλαξη και τη χρήση διαφόρων μεγεθών του πληθυσμού. Η μετάλλαξη σε μια μεμονωμένη τοποθεσία δεν είναι πολύ αποτελεσματική, έτσι η μετάλλαξη σε πολλαπλές τοποθεσίες συνήθως αυξάνει την εξέλιξη της αναζήτησης. Από την άλλη πλευρά, πάρα πολλοί μεταλλαγμένοι θα δυσχεράνουν το σύστημα να συγκλίνει ή ακόμη και να οδηγήσει το σύστημα σε λανθασμένες λύσεις. Σε πραγματικά οικολογικά συστήματα, εάν το ποσοστό μετάλλαξης είναι πολύ υψηλό υπό υψηλή πίεση επιλογής, τότε ολόκληρος ο πληθυσμός μπορεί να εξαφανιστεί. Επιπλέον, η επιλογή του δεξιού μεγέθους του πληθυσμού είναι επίσης πολύ σημαντική. Εάν το μέγεθος του πληθυσμού είναι πολύ μικρό, δεν θα υπάρξει αρκετή εξέλιξη, και υπάρχει κίνδυνος να συγκλίνει όλο το πληθυσμό πρόωρα.

Στον πραγματικό κόσμο, η οικολογική θεωρία υποδηλώνει ότι ένα είδος με μικρό πληθυσμό διατρέχει πραγματικό κίνδυνο εξαφάνισης. Σε ένα μικρό πληθυσμό, εάν ένα χρωμόσωμα με μια φυσική κατάσταση σημαντικά μεγαλύτερη από τη φυσική κατάσταση των άλλων χρωμοσωμάτων στον πληθυσμό εμφανίζεται πολύ νωρίς, μπορεί να παράγει αρκετούς απογόνους για να κατακλύσει ολόκληρο τον (μικρό) πληθυσμό. Αυτό θα οδηγήσει τελικά το σύστημα σε μια τοπική βέλτιστη (όχι καθολική βέλτιστη). Από την άλλη πλευρά, εάν ο πληθυσμός είναι πολύ μεγάλος, απαιτούνται περισσότερες αξιολογήσεις της αντικειμενικής συνάρτησης, οι οποίες θα απαιτήσουν εκτεταμένο

χρόνο υπολογισμού.

- Διαφορική εξέλιξη:

Η διαφορική εξέλιξη (DE) αναπτύχθηκε από τους Στορν και Πραις (R. Storn and K. Price) στα ονομαστικά έγγραφά τους το 1996 και το 1997 [179][180]. Είναι ένας διανυσματικός εξελικτικός αλγόριθμος, και μπορεί να θεωρηθεί ως μια περαιτέρω ανάπτυξη των γενετικών αλγορίθμων. Σε αντίθεση με τους γενετικούς αλγόριθμους, η διαφορική εξέλιξη εκτελεί λειτουργίες πάνω από κάθε συστατικό (ή κάθε διάσταση του διαλύματος). Σχεδόν όλα γίνονται από την άποψη των φορέων, και μπορεί να θεωρηθεί ως μια αυτοοργάνωση αναζήτησης, που κατευθύνεται προς το βέλτιστο.

Για ένα πρόβλημα βελτιστοποίησης παραμέτρων διαστάσεων με παραμέτρους $\mathbf{d}$, ένας πληθυσμός των φορέων λύσεων $\mathbf{n}$ $\mathbf{XI}$, ($i = 1, 2, \dots, n$), παράγεται αρχικά. Για κάθε λύση $\mathbf{XI}$ σε οποιαδήποτε γενιά t , χρησιμοποιούμε το συμβατικό $x_{ti} = (x_{t1}, i, x_{t2}, i, \dots, x_{td}, i)$. Αυτό το διάνυσμα μπορεί να θεωρηθεί ως ένα χρωμόσωμα. Η διαφορική εξέλιξη αποτελείται από τρία κύρια βήματα: μετάλλαξη, διασταύρωση και επιλογή. Η μετάλλαξη γίνεται ως εξής. Για κάθε διάνυσμα X_i ανά πάσα στιγμή ή γενιά t , αρχικά επιλέγουμε τυχαία τρεις διακριτούς φορείς X_p , X_q και r , και στη συνέχεια δημιουργούμε ένα λεγόμενο διάνυσμα δότη με το ακόλουθο σχήμα μετάλλαξης $V_T + 1I = X_q + f_{(xqX_q)}$, (3) όπου το $F \in [0, 2]$ είναι μια παράμετρος, που συχνά αναφέρεται ως το διαφορικό βάρος. Το ελάχιστο μέγεθος πληθυσμού είναι $n \geq 4$. Καταρχήν, το $F \in [0, 2]$, αλλά στην πράξη, ένα σύστημα με $F \in [0, 1]$ είναι πιο αποδοτικό και σταθερό. Η διαταραχή $w = F(X_q - X_r)$ στο διάνυσμα X_p χρησιμοποιείται για να δημιουργήσει ένα διάνυσμα δότη V_i , και αυτή η διαταραχή κατευθύνεται και αυτοοργανώνεται. Η διασταύρωση ελέγχεται από την πιθανότητα $CR \in [0, 1]$. Η πραγματική διασταύρωση μπορεί να πραγματοποιηθεί με δύο τρόπους: διωνυμική και εκθετική. Το διωνυμικό σχήμα εκτελεί διασταυρώσεις σε κάθε ένα από τα $\mathbf{d}$ στοιχεία ή μεταβλητές/παραμέτρους. Με τη δημιουργία ενός ομοιόμορφου κατανομημένου αριθμού $RI \in [0, 1]$, το στοιχείο j_{ih} του $\mathbf{XI}$ ορίζεται ως $+1J, i = VT + 1J, II fRI \leq CR, (j = 1, 2, \dots, d)$, διαφορετικά, παραμένει αμετάβλητο. Με αυτόν τον τρόπο, αποφασίζεται τυχαία αν θα αντικαταστήσει ένα συστατικό χρησιμοποιώντας το διάνυσμα δότη ή όχι. Το βασικό σύστημα DE/Rand/1/bin πραγματοποιείται στην εξίσωση, αλλά έχουν διατυπωθεί τουλάχιστον 10 διαφορετικά καθεστώτα.

- Βελτιστοποίηση αποικίας μυρμηγκιών:

Η βελτιστοποίηση της αποικίας των μυρμηγκιών πρώτο ξεκίνησε από τον Ντορίγκο (Marco Dorigo) [48] το 1992 και βασίζεται στη συμπεριφορά των κοινωνικών μυρμηγκιών για την αναζήτηση τροφής. Πολλά έντομα, όπως τα μυρμήγκια, χρησιμοποιούν φερομόνη ως χημικό αγγελιοφόρο. Τα μυρμήγκια είναι κοινωνικά έντομα και ζουν μαζί σε οργανωμένες αποικίες που αποτελούνται από περίπου 2 έως 25.000.000 άτομα. Κατά την αναζήτηση τροφής, ένα σμήνος μυρμηγκιών ή κινητών παραγόντων αλληλεπιδρούν ή επικοινωνούν στο τοπικό τους περιβάλλον.

Κάθε μυρμήγκι βάζει αρώματα χημικών ή φερομόνης για να επικοινωνεί με άλλους. Κάθε μυρμήγκι είναι επίσης σε θέση να ακολουθήσει τη διαδρομή που επισημαίνονται με φερομόνες που έχουν γίνει από άλλα μυρμήγκια. Όταν ένα μυρμήγκι βρει μια πηγή τροφής, θα το σηματοδοτήσει με φερομόνη και επίσης θα σηματοδοτήσει το μονοπάτι προς και από αυτό. Ωστόσο, η συγκέντρωση φερομόνης f αποσπάται ή εξατμίζεται με σταθερό ρυθμό g . Δηλαδή, $f(t) = f_0[g]$, όπου f_0 είναι η αρχική συγκέντρωση σε $t = 0$. Εδώ η εξάτμιση είναι σημαντική, καθώς εξασφαλίζει τη δυνατότητα σύγκλισης και αυτοοργάνωσης. Από την αρχική διαδρομή τυχαίας τροφής,

η συγκέντρωση φερομόνης ποικίλλει και τα μυρμηγκια ακολουθούν τη διαδρομή με υψηλότερη συγκέντρωση φερομόνης. Με τη σειρά της, η φερομόνη ενισχύεται από τον αυξανό-μενο αριθμό των μυρμηγκιών. Καθώς όλο και περισσότερα μυρμηγκια ακολουθούν την ίδια διαδρομή, γίνεται το κύριο μονοπάτι. Έτσι, εμφανίζονται κάποιες αγαπημένες διαδρομές, συχνά οι πιο σύντομες ή πιο αποτελεσματικές. Αυτός είναι πράγματι ένας θετικός μηχανισμός ανάδρασης. Καθώς το σύστημα εξελίσσεται, εμφανίζεται σε μια αυτοοργανωμένη κατάσταση, η οποία είναι η ουσία οποιουδήποτε αλγορίθμου μυρμηγκιών.

- Αλγόριθμοι μέλισσας:

Οι αλγόριθμοι μέλισσας είναι μια άλλη κατηγορία μεθευρετικών αλγορίθμων, εμπνευσμένοι από τη συμπεριφορά των μελισσών με αναζήτηση τροφής. Μερικές παραλλαγές υπάρχουν στη βιβλιογραφία, συμπεριλαμβανομένου του αλγορίθμου μέλισσας, της αποικίας τεχνητών μελισσών, του αλγορίθμου της μέλισσας, του αλγορίθμου του εικονικού μέλισσας και των αλγορίθμων ζευγαρώματος των μελισσών.

Οι μέλισσες ζουν σε μια αποικία και τρέφονται και αποθηκεύουν μέλι στην κατασκευασμένη αποικία τους. Οι μέλισσες μπορούν να επικοινωνούν με φερομόνες. Για παράδειγμα, μια ανήσυχη μέλισσα μπορεί να απελευθερώσει ένα χημικό μήνυμα (φερομόνης) για να τονώσει την αντίδραση επίθεσης σε άλλες μέλισσες. Επιπλέον, όταν οι μέλισσες βρίσκουν μια καλή πηγή τροφής και φέρνουν λίγο νέκταρ πίσω στην κυψέλη, θα κοινοποιήσουν τη θέση της πηγής τροφής εκτελώντας ένα χορό ως σύστημα σηματοδότησης. Τέτοιοι χοροί σηματοδότησης ποικίλλουν από είδη σε είδος, ωστόσο, στοχεύουν στην πρόσληψη περισσότερων μελισσών με τη χρήση κατευθυντήριου χορού με διαφορετική δύναμη, έτσι ώστε να επικοινωνούν η κατεύθυνση και η απόσταση της πηγής τροφής. Για πολλαπλές πηγές τροφίμων, όπως τα έμπλαστρα λουλουδιών, μελέτες δείχνουν ότι μια αποικία μελισσών φαίνεται να είναι σε θέση να διαθέσει τις μέλισσες ανάμεσα σε διαφορετικά έμπλαστρα λουλουδιών, έτσι ώστε να μεγιστοποιήσει τη συνολική πρόσληψη νέκταρ τους. Φαίνεται ότι ο αλγόριθμος μέλισσας διαμορφώθηκε για πρώτη φορά γύρω το 2004 από τον Κρεκ (Craig) στη Τζόρτζια Τεκ σε συνεργασία με τον Νακράνι (Nacarani) στη συνέχεια στο Πανεπιστήμιο της Οξφόρδης ως μέθοδο για την εκχώρηση υπολογιστών μεταξύ διαφορετικών πελατών και διακομιστών φιλοξενίας ιστού.

Αργότερα το 2004 και στις αρχές της 2005, ο Γιανγκ (Yang) στο Πανεπιστήμιο του Cambridge ανέπτυξε έναν αλγόριθμο εικονικής μέλισσας (VBA) για την επίλυση προβλημάτων συνεχούς βελτιστοποίησης. Την ίδια ώρα περίπου, οι Φαμ (Pham et al.) [157] ανέπτυξαν τους αλγόριθμους των μελισσών. Ελαφρώς αργότερα το 2005, ο Αφσαρ (Afsar) [2] και οι συνάδελφοί του παρουσίασαν έναν αλγόριθμο βελτιστοποίησης ζευγαρώματος της μέλισσας, ο οποίος εφαρμόστηκε στη συνέχεια σε μοντελοποίηση και σύμπλεγμα. Περίπου την ίδια στιγμή, ο Καραμπόγα (Karaboga) [96] στην Τουρκία ανέπτυξε έναν αλγόριθμο τεχνητής μελισσοκομίας (ABC) για αριθμητική βελτιστοποίηση της συνάρτησης. Οι αλγόριθμοι μυρμηγκιών και μέλισσας είναι πιο κατάλληλοι για διακριτή και συνδυαστική βελτιστοποίηση και έχουν εφαρμοστεί σε ένα ευρύ φάσμα εφαρμογών.

- Βελτιστοποίηση σμήνους σωματιδίων:

Η βελτιστοποίηση του σμήνους σωματιδίων αναπτύχθηκε από τον Κένεντι (Kennedy) και τον Εμπερχάρτ (Eberhart) το 1995, [106] με βάση τη συμπεριφορά του σμήνους που παρατηρήθηκε στη φύση, όπως τα ψάρια και η εκπαίδευση των πτηνών. Έκτοτε, έχει τραβήξει μεγάλη προσοχή

και τώρα σχηματίζει ένα συναρπαστικό, συνεχώς διευρυνόμενο ερευνητικό θέμα στον τομέα της ευφυΐας του σμήνους. Έχει εφαρμοστεί σχεδόν σε κάθε περιοχή στη βελτιστοποίηση, την υπολογιστική νοημοσύνη και τις εφαρμογές σχεδιασμού/προγραμματισμού. Υπάρχουν τουλάχιστον δύο δεκάδες παραλλαγών, καθώς και υβριδικοί αλγόριθμοι που λαμβάνονται συνδυάζοντας την μέθοδο με άλλους υπάρχοντες αλγόριθμους, οι οποίοι είναι επίσης όλο και πιο δημοφιλείς. Η μέθοδος αναζητά το χώρο μιας αντικειμενικής συνάρτησης προσαρμόζοντας τις τροχιές μεμονωμένων παραγόντων, που ονομάζονται σωματίδια. Κάθε σωματίδιο ανιχνεύει μια γραμμή που μπορεί να μοντελοποιηθεί ως διάνυσμα θέσης που εξαρτάται από το χρόνο. Η κίνηση ενός σωματιδίου αποτελείται από δύο σημαντικά συστατικά: ένα στοχαστικό συστατικό και ένα ντετερμινιστικό συστατικό.

Κάθε σωματίδιο έλκεται προς τη θέση του τρέχοντος καθολικού βέλτιστου g και της δικής του πιο γνωστής τοποθεσίας $x * i$, ενώ ταυτόχρονα παρουσιάζει την τάση να κινείται τυχαία. Όταν ένα σωματίδιο βρίσκει μια τοποθεσία που είναι καλύτερη από τις τοποθεσίες που βρέθηκαν παλαιότερα, ενημερώνει αυτήν τη θέση ως το νέο τρέχον καλύτερο για το σωματίδιο i . Υπάρχει μια τρέχουσα βέλτιστη για όλα τα σωματίδια ανά πάσα στιγμή t σε κάθε επανάληψη. Στόχος είναι να βρεθεί η καθολική βέλτιστη μεταξύ όλων των σημερινών καλύτερων λύσεων μέχρις ότου ο στόχος να μην βελτιώνεται πλέον ή μετά από ορισμένο αριθμό επαναλήψεων. Αν X_i και V_i είναι η θέση και η ταχύτητα, αντίστοιχα, του σωματιδίου i , το νέο διάνυσμα ταχύτητας προσδιορίζεται από τον ακόλουθο τύπο $VT + 1I = VT_i + A1[g * XT_i] + B2[x * iXT_i]$. Η E_1 και η E_2 είναι δύο τυχαία διανύσματα και κάθε εγγραφή παίρνει μια τιμή μεταξύ 0 και 1. Οι παράμετροι α και β είναι οι παράμετροι μάθησης ή οι σταθερές επιτάχυνσης, οι οποίες είναι συνήθως ίσες με, ας πούμε, $\alpha \approx \beta \approx 2$. Οι αρχικές θέσεις όλων των σωματιδίων θα πρέπει να διανέμονται σχετικά ομοιόμορφα, ώστε να μπορούν να δοκιμάσουν τις περισσότερες περιοχές, πράγμα που είναι ιδιαίτερα σημαντικό για τα πολυτροπικά προβλήματα. Η αρχική ταχύτητα ενός σωματιδίου μπορεί να οριστεί σε μηδέν, δηλαδή $VT = 0i = 0$. Στη συνέχεια, η νέα θέση μπορεί να ενημερωθεί από το τυπομοντέλο $+1I = XT_i + VT + 1I$. Αν και η VI μπορεί να λάβει οποιαδήποτε τιμή, συνήθως οριοθετείται σε κάποια περιοχή $[0, V_{max}]$. Υπάρχουν πολλές παραλλαγές που επεκτείνουν τον τυπικό αλγόριθμο και η πιο αισθητή βελτίωση είναι πιθανώς να χρησιμοποιηθεί μια αδρανής συνάρτηση $\theta_{(t)}$ έτσι ώστε το VT_i να αντικαθίσταται από $\theta_{(t)}VT_i$ όπου θ λαμβάνει μια τιμή μεταξύ 0 και 1. Στην απλούστερη περίπτωση, η αδρανής συνάρτηση μπορεί να ληφθεί ως σταθερή, τυπικά $\theta \in [0.5, 0, 9]$. Αυτό ισοδυναμεί με την εισαγωγή μιας εικονικής μάζας για τη σταθεροποίηση της κίνησης των σωματιδίων, και έτσι ο αλγόριθμος αναμένεται να συγκλίνει πιο γρήγορα. Καθώς οι επαναλήψεις προχωρούν, το σύστημα σωματιδίων μπορεί να συγκλίνει προς ένα ολικό βέλτιστο.

- Αναζήτηση Tabu:

Η αναζήτηση Tabu αναπτύχθηκε από τον Γκλόβερ (Fred Glover) στη δεκαετία του 1970, και το βιβλίο του δημοσιεύτηκε πολύ αργότερα το 1997. [72] Η αναζήτηση Tabu χρησιμοποιεί ρητά τη μνήμη και το ιστορικό αναζήτησης, το οποίο είναι ένα σημαντικό στοιχείο της μεθόδου. Καθώς οι περισσότεροι αλγόριθμοι χρησιμοποιούν μόνο τα αποτελέσματα των τελευταίων ή δύο τελευταίων βημάτων, είναι αρχικά δύσκολο να δούμε το πλεονέκτημα της χρήσης του ιστορικού αναζήτησης. Οι λεπτές αποχρώσεις της μνήμης και της ιστορίας θα μπορούσαν να εισαγάγουν πάρα πολλούς βαθμούς ελευθερίας, και μια μαθηματική ανάλυση της συμπεριφοράς του αλγορίθμου γίνεται ανίατη. Ωστόσο, η αναζήτηση Tabu παραμένει ένα από τα πιο επιτυχημένα και ευρέως χρησιμοποιούμενα μεθευρετικά βελτιστοποίησης. Στην ουσία, η αναζήτηση Tabu μπορεί να θεωρηθεί ως μια εντατική τοπική αναζήτηση και η κατάλληλη χρήση του ιστορικού αναζήτησης

αποφεύγει την επανάληψη τοπικών λύσεων με την εγγραφή πρόσφατα δοκιμασμένα λύσεις σε λίστες **Tabu**. Πάνω από ένα μεγάλο αριθμό διαδοχικών προσεγγίσεων, αυτές οι λίστες **Tabu** θα μπορούσαν να γλυτώσουν ένα σημαντικό υπολογιστικού χρόνου, οδηγώντας σε βελτιώσεις στην αποδοτικότητα της αναζήτησης. Για παράδειγμα, μελέτες δείχνουν ότι η χρήση των λιστών **Tabu** με ακέραιο προγραμματισμό μπορεί να εξοικονομήσει υπολογιστική προσπάθεια από τουλάχιστον δύο σειρές μεγέθους για ένα δεδομένο πρόβλημα, σε σύγκριση με τον τυπικό ακέραιο προγραμματισμό. Πολλοί υβριδικοί αλγόριθμοι έχουν αναπτυχθεί συνδυάζοντας την αναζήτηση **Tabu** με άλλες μεθόδους.

- Αναζήτηση αρμονίας:

Είναι ένας σχετικά νέος αλγόριθμος βελτιστοποίησης που αναπτύχθηκε πρώτα από τον Γκιμ (Geem et al.) το 2001 [64]. Η αναζήτηση αρμονίας σχετίζεται με τη διαδικασία αυτοσχεδιασμού ενός μουσικού. Όταν ένας μουσικός αυτοσχεδιάζει, έχει τρεις δυνατές επιλογές:

1. παίζει οποιοδήποτε διάσημο μουσικό κομμάτι ακριβώς από τη μνήμη του
2. παίζει κάτι παρόμοιο με ένα γνωστό κομμάτι (προσαρμόζοντας έτσι ελαφρά το σημείο) ή
3. συνθέτει νέες ή τυχαίες σημειώσεις.

Εάν επισημοποιήσουμε αυτές τις τρεις επιλογές για τη βελτιστοποίηση, έχουμε τρία αντίστοιχα στοιχεία: χρήση της μνήμης αρμονίας, ρύθμιση του βήματος, και τυχαία επιλογή.

- Αλγόριθμος πυγολαμπίδας:

Ο αλγόριθμος **FA** αναπτύχθηκε από την Γιανγκ (Xin-He Yang) και βασίζεται στα μοτίβα που αναβιβάζουν και τη συμπεριφορά των Πυγολαμπίδων. Στην ουσία, η **FA** χρησιμοποιεί τους ακόλουθους τρεις εξιδανικευμένους κανόνες: Οι πυγολαμπίδες είναι αμφιφυλόφιλες έτσι ώστε μια πυγολαμπίδα να έλκεται από άλλες πυγολαμπίδες ανεξάρτητα από το φύλο τους. Η ελκυστικότητα είναι ανάλογη με τη φωτεινότητα και μειώνεται καθώς η απόσταση μεταξύ δύο πυγολαμπίδων αυξάνεται. Έτσι για κάθε δύο πυγολαμπίδες, η φωτεινότερη πυγολαμπίδα θα προσελκύσει την άλλη. Εάν καμία από τις δύο δεν είναι φωτεινότερη, τότε πραγματοποιείται μια τυχαία κίνηση. Η φωτεινότητα μιας πυγολαμπίδας καθορίζεται από το τοπίο της αντικειμενικής συνάρτησης. Καθώς η ελκυστικότητα μιας πυγολαμπίδας είναι ανάλογη με την ένταση του φωτός που βλέπουν οι παρακείμενες πυγολαμπίδες, μπορούμε τώρα να ορίσουμε τη διαχύμανση της ελκυστικότητας $\mathbf{b}$ με την απόσταση $\mathbf{r}$ με $\mathbf{b} = b_0 \exp(-r^2)$, όπου b_0 είναι η ελκυστικότητα σε $\mathbf{r} = 0$. Η κίνηση μιας πυγολαμπίδας, που ελκύεται από μια άλλη πιο ελκυστική (φωτεινότερη) πυγολαμπίδα, προσδιορίζεται από το $\mathbf{X}T + 1I = \mathbf{X}T\mathbf{i} + b_0 \exp(-r_i^2) (\mathbf{X}_j^T \mathbf{X}_i^T)$.



2. Η Μεθοδολογία

2.1 Προσεγγιστικές και μεθευρετικές μέθοδοι

Η συνδυαστική βελτιστοποίηση είναι μια τεχνική η οποία χρησιμοποιείται για να βρίσκει την βέλτιστη, από το σύνολο των λύσεων, λύση. Υπάρχουν διάφοροι τρόποι για να πραγματοποιηθεί η συνδυαστική βελτιστοποίηση. Η φυσική γλώσσα, οι λογικές συνεπαγωγές, οι γράφοι και ο μαθηματικός προγραμματισμός είναι κάποιοι από τους τρόπους, αν και συνήθως γίνεται χρήση των ακέραιων λύσεων.

Για την επίλυση προβλημάτων βελτιστοποίησης, με έμφαση σε NP-complete προβλήματα, μπορούν να χρησιμοποιηθούν τόσο γενικές μέθοδοι, οι οποίες εφαρμόζονται σε οποιοδήποτε πρόβλημα, όσο και ειδικές. Η πηγή των μεθόδων αυτών είναι επιστημονικοί κλάδοι όπως η τεχνητή νοημοσύνη, ο μαθηματικός προγραμματισμός και η υπολογιστική νοημοσύνη. Ως γενικές μέθοδοι μπορούμε να κατατάξουμε τους ακριβείς (exact) αλγόριθμους καθώς και την τοπική αναζήτηση (local search), με τις μεθευρετικές της (metaheuristics) και τις επεκτάσεις. Οι ειδικές μέθοδοι σχεδιάζονται ανάλογα με τη φύση του προβλήματος και εξαρτώνται άμεσα από το ίδιο το πρόβλημα. Ως τέτοιες μπορούμε να κατατάξουμε τις ευρετικές (heuristics) και τους προσεγγιστικούς (approximation) αλγόριθμους.

Ο αλγόριθμος simplex προέρχεται από τον πολύ σημαντικό επιστημονικό κλάδο του γραμμικού προγραμματισμού και εφαρμόζεται κατά κόρον μαζί με τις παραλλαγές του ως αναθεωρημένος αλγόριθμος, simplex δύο φάσεων κτλ. Ο αλγόριθμος simplex αποτελεί ένα μόνο παράδειγμα από την πληθώρα των ακριβών αλγόριθμων του κλάδου του μαθηματικού προγραμματισμού (οι οποίοι αναφέρονται στη βιβλιογραφία και ως αλγόριθμοι επιχειρησιακής έρευνας).

Παρ' όλα αυτά, οι συμβατικές μέθοδοι δεν ενδείκνυνται στη συνδυαστική ή ολική βελτιστοποίηση. Για NP-hard προβλήματα δηλαδή, όπως η δρομολόγηση οχημάτων (vehicle routing), η κοπή-τεμαχισμός (cut), η εγκατάσταση (facility location), ο χρονοπρογραμματισμός (scheduling), η επένδυση και η ανάθεση (assignment). Για αυτά τα προβλήματα, όπου συνήθως ο χώρος αναζήτησής τους είναι μεγάλος και πολύπλοκος, οι συμβατικές μέθοδοι δεν είναι αρκετά αποτελεσματικές, καθώς για τα NP-hard υπολογιστικά προβλήματα δεν είναι δυνατή η εύρεση λύσης σε πολυωνυμικό χρόνο (εκτός εάν $P = NP$).

Οι τεχνικές διακλάδωσης και οριοθέτησης (**branch and bound**) και **cutting plane** από την άλλη προσφέρουν μια αποδοτική λύση για τέτοια προβλήματα και προέρχονται από το επιστημονικό πεδίο του ακέραιου προγραμματισμού. Είναι οι πλέον κατάλληλες για δύσκολα προβλήματα που αφορούν στην πρακτική εφαρμογή της συνδυαστικής βελτιστοποίησης. Η πρώτη τεχνική, διακλάδωση και οριοθέτηση (**branch and bound**), δίνει τη δυνατότητα να αποκόπτονται τμήματα του δέντρου αναζήτησης και ταυτοχρόνως να υπολογίζονται φράγματα για την βέλτιστη τιμή σε κάθε κόμβο. Η δεύτερη τεχνική, **cutting plane**, εντοπίζει με επαναληπτικό τρόπο τις περικοπές (**cuts**), δηλαδή είτε το σύνολο των εφικτών λύσεων είτε την αντικειμενική συνάρτηση μέσω γραμμικών ανισοτήτων.

Στη βιβλιογραφία βεβαίως υπάρχουν και διάφορες ευρετικές μέθοδοι, οι οποίες σε σύντομο χρονικό διάστημα υπολογισμού καταλήγουν σε υποβέλτιστες λύσεις. Οι ευρετικές μέθοδοι αναζήτησης είναι αποτέλεσμα δημιουργικής και απλής διαισθητικής σκέψης του ανθρώπου και χρησιμοποιούνται για τον εντοπισμό καλών λύσεων σε σύντομο χρονικό διάστημα, στην τοπική αναζήτηση σε συγκεκριμένες, περιορισμένες περιοχές. Οι μεθευρετικές μέθοδοι από την άλλη θεωρούνται μέθοδοι υψηλότερου επιπέδου και αναλύονται εκτενέστερα στη συνέχεια. Ουσιαστικά, καθοδηγούν όλη την διαδικασία της αναζήτησης με συστηματικό τρόπο, με τη χρήση ευρετικών μεθόδων. Αυτές οι μέθοδοι, παρόλο που δεν εξασφαλίζουν πάντα την εύρεση της ολικά βέλτιστης λύσης, είναι πολύ αποτελεσματικές σε ποικίλα πρακτικά προβλήματα. Σε αντίθεση με τις κλασικές μεθόδους, οι οποίες είναι πολύ αργές ή αδυνατούν να βρουν λύση με ακρίβεια, η ευρετική είναι τεχνική προσανατολισμένη στο να λύνει περίπλοκα προβλήματα ταχύτατα και να παρουσιάζει προσεγγιστικές λύσεις εκεί που οι κλασικές μέθοδοι αποτυγχάνουν. Αυτά τα αποτελέσματα προκύπτουν με την ενσωμάτωση μεθόδων βελτιστοποίησης στην ευρετική μέθοδο. Οι μέθοδοι βελτιστοποίησης έχουν το πλεονέκτημα της πληρότητας, της ακρίβειας καθώς και της ταχύτητας.

Πρωταρχικό μέλημα ενός ευρετικού αλγορίθμου είναι η εύρεση λύσης μέσα σε ένα σύντομο και λογικό χρονικό πλαίσιο για το εκάστοτε πρόβλημα. Η λύση που θα βρεθεί πιθανώς να μην είναι η βέλτιστη σε σχέση με όλες τις άλλες πιθανές λύσεις που μπορεί να υπάρχουν για το συγκεκριμένο πρόβλημα αλλά μπορεί να προσεγγίζει την βέλτιστη. Αυτό βέβαια δεν την κάνει λιγότερο σημαντική, διότι ο χρόνος εύρεσής της είναι μέσα στα πλαίσια που ορίζει το πρόβλημα. Οι ευρετικές μέθοδοι μπορούν είτε να βρουν το αποτέλεσμα μόνες τους, είτε για μεγαλύτερη αποτελεσματικότητα απέναντι σε προβλήματα όπως η δημιουργία καλών αντικειμενικών τιμών, συνδυαστικά με αλγορίθμους βελτιστοποίησης.

Οι μεθευρετικές μέθοδοι είναι υψηλού επιπέδου ευρετικά πλαίσια, σχεδιασμένα να εντοπίζουν, να δημιουργούν ή να επιλέγουν ένα χαμηλότερο ευρετικό επίπεδο, όπως είναι ένας τοπικός αλγόριθμος αναζήτησης, για την εύρεση λύσης σε ένα πρόβλημα βελτιστοποίησης. Αυτές οι μέθοδοι είναι εξαιρετικά χρήσιμες σε περιπτώσεις μερικής ή ολικής απώλειας πληροφοριών καθώς και σε περιπτώσεις όπου περιορίζεται η υπολογιστική ικανότητα. Η λέξη "μεθευρετική", σύμφωνα με τη βιβλιογραφία, επινοήθηκε και προτάθηκε από τον Γκλόβερ (Glover) [73]. Το σημαντικό πλεονέκτημα που προσφέρουν οι μεθευρετικοί αλγόριθμοι είναι ότι μπορούν να κάνουν υποθέσεις γύρω από το πρόβλημα βελτιστοποίησης το οποίο καλούνται να επιλύσουν και κατά συνέπεια μπορούν να χρησιμοποιηθούν για ένα μεγάλο εύρος προβλημάτων. Ωστόσο, υστερούν σε σχέση με τις ακριβείς μεθόδους, καθώς δεν είναι σίγουρο ότι το αποτέλεσμα που θα επιστρέψουν θα είναι και η καθολικά βέλτιστη λύση για κάθε κατηγορία προβλημάτων.

Για να αποφασίσει κανείς αν θα χρησιμοποιήσει μια ευρετική μέθοδο για να λύσει το εκάστοτε πρόβλημα, μπορεί να συμβουλευτεί τα παρακάτω κριτήρια:

- Βέλτιστη λύση:

Στην περίπτωση που στο πρόβλημα που μας απασχολεί προκύπτει ένα σύνολο από πιθανές λύσεις, είναι απαραίτητο να βρεθεί η βέλτιστη λύση;

- Πληρότητα:

Στην περίπτωση που στο πρόβλημα που μας απασχολεί προκύπτει ένα σύνολο από πιθανές λύσεις, είναι απαραίτητο να βρεθούν οι λύσεις; Μπορούν οι ευρετικοί αλγόριθμοι να τις βρουν όλες;

- Ακρίβεια:

Σε περίπτωση που η γραμμή σφάλματος της λύσης είναι αδικαιολόγητα μεγάλη και ο ευρετικός αλγόριθμος δεν μπορεί να παρέχει ένα διάστημα εμπιστοσύνης για την λύση του προβλήματος.

- Χρόνος εκτέλεσης:

Ποια μέθοδος είναι γνωστή για την επίλυση τέτοιου είδους προβλημάτων; Και κυρίως, ποια από αυτές έχει τον ταχύτερο χρόνο εκτέλεσης;

Σε κάποιες περιπτώσεις δεν είναι σαφές αν η λύση που προέκυψε από τον ευρετικό αλγόριθμο είναι η βέλτιστη, διότι κάποια προβλήματα έχουν αρκετά πιο περίπλοκη δομή. Παραδείγματος χάρη, το πρόβλημα του πλανόδιου πωλητή, όπως περιγράφεται από τον Τζον Μπέντλεϊ (**John Bentley**) [15] "δεδομένης μιας λίστας πόλεων και των αποστάσεων μεταξύ κάθε ζεύγους πόλεων, ποια είναι η συντομότερη δυνατή διαδρομή που επισκέπτεται κάθε πόλη και επιστρέφει στην πόλη προέλευσης;"

Οι μεθευρετικοί αλγόριθμοι συχνά συνδυάζονται με στοχαστικές βελτιστοποιήσεις. Επί της ουσίας, ένα σύνολο τυχαίων μεταβλητών καθορίζει το παραγόμενο αποτέλεσμα. Το πλεονέκτημα των μεθευρετικών αλγορίθμων είναι ότι μπορούν να εντοπίσουν βέλτιστες λύσεις σε ένα πρόβλημα σε ένα μεγάλο σύνολο πιθανών λύσεων, με λιγότερη υπολογιστική προσπάθεια από τις υπόλοιπες μεθόδους, όπως είναι οι απλές ευρετικές, οι επαναληπτικές μέθοδοι και οι ακριβείς αλγόριθμοι. Αυτό φυσικά τις καθιστά εξαιρετικά χρήσιμες για προβλήματα βελτιστοποίησης σε ποικίλες πρακτικές καταστάσεις.

2.1.1 Εισαγωγή στις μεθευρετικές διαδικασίες

Μια μεθευρετική μέθοδος για την επιστήμη των υπολογιστών και τη μαθηματική βελτιστοποίηση, είναι μια, ευρετική πολύ υψηλού επιπέδου, διαδικασία η οποία έχει δημιουργηθεί για να εντοπίζει, να δημιουργεί ή να συνθέτει άλλες ευρετικές διαδικασίες χαμηλότερου επιπέδου - όπως είναι ο αλγόριθμος μερικής αναζήτησης - με σκοπό να προσφέρει μια όσο τον δυνατό βέλτιστη λύση σε προβλήματα βελτιστοποίησης, ιδιαίτερα όταν αυτά είναι ελλιπή ή ατελή ως προς τις πληροφορίες τους ή υπάρχει περιορισμός ως προς την υπολογιστική ικανότητα. [18] Βάσει βιβλιογραφίας, [200] ο πρώτος που εισήγαγε τον όρο "μεθευρετική" (metaheuristics) ήταν ο καθηγητής επιστήμης των υπολογιστών και εφαρμοσμένων μαθηματικών, Φρεντ Γκλόβερ (**Fred Glover**), στο πανεπιστήμιο του Κολοράντο, στην πόλη Κάνσας των ΗΠΑ.

Η καινοτομία των μεθευρετικών διαδικασιών έγκειται στο ότι μπορούν να κάνουν υποθέσεις σχετικά με το πρόβλημα βελτιστοποίησης το οποίο καλούνται να επιλύσουν, οπότε είναι ιδανικές για ένα μεγάλο εύρος προβλημάτων [20]. Εκεί που υστερούν είναι ότι, σε αντίθεση με τους αλγορίθμους βελτιστοποίησης και τις επαναληπτικές μεθόδους, δεν εγγυώνται ότι θα βρεθεί μια καθολικά βέλτιστη λύση για όλα τα είδη προβλημάτων στα οποία θα εφαρμοστούν [20]. Γι' αυτό το λόγο, συχνά πολλές μεθευρετικές μέθοδοι συνδυάζονται με κάποια μορφή στοχαστικής βελτιστοποίησης, ώστε ένα σύνολο τυχαίων μεταβλητών να είναι αυτό από το οποίο θα εξαρτάται η λύση που θα προκύψει [18].

Οι μεθευρετικές διαδικασίες είναι σαφώς πιο γρήγορες και αποτελεσματικές από τους ευρετικούς αλγορίθμους και τις επαναληπτικές μεθόδους, καθώς αναζητούν την βέλτιστη λύση σε

ένα σύνολο εφικτών λύσεων με λιγότερη υπολογιστική προσπάθεια [20]. Αυτό τις καθιστά ιδανικές για προβλήματα βελτιστοποίησης. Οι μεθευρετικές διαδικασίες είναι ένα κεφάλαιο που έχει απασχολήσει πολύ τους ερευνητές διεθνώς και υπάρχει εκτεταμένη δημοσιευμένη έρευνα γύρω από αυτό το θέμα [18] [20] [75] [73] [182].

Οι δημοσιευμένες έρευνες γύρω από τις μεθευρετικές διαδικασίες στηρίζονται κυρίως στο πειραματικό μέρος του προβλήματος, παρουσιάζοντας εμπειρικά αποτελέσματα που προκύπτουν μετά από πειράματα με αυτούς τους αλγόριθμους σε υπολογιστές. Ωστόσο, εκτός από το πειραματικό μέρος, διατίθενται και έρευνες με θεωρητικό υπόβαθρο, οι οποίες κατά κύριο λόγο ασχολούνται με την σύγκριση και τη δυνατότητα εύρεσης του καθολικού βέλτιστου [20]. Το πεδίο χαρακτηρίζεται κατά βάση από έρευνα υψηλής ποιότητας [200], καθώς μια πληθώρα δημοσιεύσεων έχουν αξιώσεις καινοτομίας και πρακτικής αποτελεσματικότητας. Ωστόσο, δεν λείπουν και οι δημοσιεύσεις κατώτερου επιπέδου, τις οποίες χαρακτηρίζουν η ασάφεια, η έλλειψη εννοιολογικής επεξεργασίας, τα ανεπαρκή πειράματα και η άγνοια της υπάρχουσας βιβλιογραφίας γύρω από αυτό το πεδίο.

2.1.2 Ιστορικά στοιχεία

Η βιβλιογραφία έχει να επιδείξει μια μεγάλη συλλογή μεθευρετικών τεχνικών και διαρκώς προκύπτουν νέες ή παραλλαγές των υπάρχουσών τεχνικών. Στον παρακάτω πίνακα αναφέρονται συγκεντρωτικά μερικές από τις πιο σημαντικές τεχνικές του πεδίου αυτού:

Table 2.1: Μεθευρετικές τεχνικές

Πότε	Ποιος	Συνεισφορά
1952	Χέρμπερτ Ρόμπινς-Σάτον Μονρό (Herbert Robbins - Sutton Monro)	Στοχαστικές μέθοδοι βελτιστοποίησης [167]
1954	Νιλς Μπαριτσέλι (Nils Barricelli)	Πρώτες προσομοιώσεις της διαδικασίας εξέλιξης [12]
1963	Λεονάρντ Α. Ραστρίγκιν (Leonard A. Rastrigin)	Τυχαία αναζήτηση [161]
1965	Ι. Ματίας (J. Matias)	Τυχαία βελτιστοποίηση [123]
1965	Τζον Ασγουόρθ Νέλντερ και Ρότζερ Μεντ (John A. Nelder and Roger Mead)	Simplex ευρετική, η οποία αποδείχθηκε ότι συγκλίνει σε μη στάσιμα σημεία σε μερικά προβλήματα [136]
1966	Λόρενς Τζ. Φόγκελ, Άλβιν Τζ. Όουενς και Μάικλ Τζ. Γουόλς (Lawrence J. Fogel, Alvin J. Owens and Michael J. Walsh)	Εξελικτικός προγραμματισμός [60]
1970	Ουίλιαμ Κιθ Χάστιγκς (William Keith Hastings)	Αλγόριθμος Metropolis-Hastings [86]
1970	Ντάνιελ Τζόζεφ Καβίτσιο (Daniel Joseph Cavicchio)	Προσαρμογή των παραμέτρων ελέγχου για έναν βελτιστοποιητή [28]
1970	Μπράιαν Ουίλσον Κέρνιγκαν και Σεν Λιν (Brian Wilson Kernighan and Shen Lin)	Μέθοδος διαμέρισης γράφων, σχετική με την αναζήτηση μεταβλητού βάθους και την αναζήτηση ταμπού βάσει απαγόρευσης [107]
1975	Τζον Χένρι Χόλαντ (John Henry Holland)	Γενετικός αλγόριθμος [89]

Μεθευρετικές τεχνικές – συνέχεια

Πότε	Ποιος	Συνεισφορά
1977	Φρεντ Γκλόβερ (Fred Glover)	Διασκορπισμένη αναζήτηση [70]
1978	Ρόμπερτ Ε. Μέρσερ και Τζέφρι Ρ. Σαμπσόν (Robert E. Mercer and Jeffrey R. Sampson)	Πίνακας για τη ρύθμιση των παραμέτρων ενός βελτιστοποιητή, χρησιμοποιώντας έναν άλλο βελτιστοποιητή [125]
1980	Στέφεν Φ. Σμιθ (Stephen Frederick Smith)	Γενετικός προγραμματισμός [178]
1983	Σκοτ Κερκπάτρικ, Ντάνιελ Τζελάτ και Μάριο Π. Βέτσι (Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi)	Προσομοιωμένη ανόπτηση [108]
1986	Φρεντ Γκλόβερ (Fred Glover)	Αναζήτηση ταμπού, πρώτη αναφορά του όρου «metaheuristic» [71]
1989	Πάμπλο Μοσκάτο (Pablo Moscato)	Μιμητικοί αλγόριθμοι [132]
1992	Μάρκο Ντορίγκο (Marco Dorigo)	Βελτιστοποίηση αποικίας μυρμηγκιών [48]
1995	Ντέιβιντ Η. Γουόλπερτ, Γουίλιαμ Τσαρλς Μακρέντι κ.α. (David H. Wolpert, William G. Macready and others)	Απόδειξη θεωρημάτων «no free lunch» [197]

Ο όρος "μεθευρετικές" ("metaheuristic") αναφέρεται για πρώτη φορά το 1986, όπως φαίνεται και από τον πίνακα 1.1. Η έρευνα, ωστόσο, του κλάδου της βελτιστοποίησης ξεκίνησε αρκετές δεκαετίες νωρίτερα από την πρώτη αναφορά του όρου. Το πρώτο σημαντικό επίτευγμα στον κλάδο της βελτιστοποίησης έκανε την εμφάνισή του περί το 1952, με τη δημοσίευση των Χέρμπερτ Ρόμπινς-Σάτον Μονρό (Herbert Robbins - Sutton Monro) σχετικά με τις στοχαστικές μεθόδους βελτιστοποίησης [167]. Από τότε και έπειτα, έγινε η αρχή για νέες προσεγγίσεις στην επίλυση προβλημάτων βελτιστοποίησης σε συνδυασμό με μεθευρετικές τεχνικές. Οι πρώτες προσομοιώσεις της διαδικασίας εξέλιξης παρουσιάστηκαν από τον Νιλς Ααλ Μπαρισέλι (Nils Aall Barriceli) γύρω στο 1954, όπου προτάθηκε η χρήση τους σε γενικά προβλήματα βελτιστοποίησης [12].

Το 1963 παρουσιάστηκε μια πρώιμη μορφή της τυχαίας αναζήτησης (Random Search - RS) μαζί με την μαθηματική της ανάλυση από τον Λέοναρντ Αντρέιβιτς Ραστρίγκιν (Leonard Andreevich Rastrigin), η οποία προτάθηκε για πρώτη φορά ως μέθοδος επίλυσης προβλημάτων βελτιστοποίησης [161]. Η μέθοδος της τυχαίας βελτιστοποίησης (Random Optimization - RO) προτάθηκε το 1965 από τον Ι. Matias [123]. Η τυχαία αναζήτηση και η τυχαία βελτιστοποίηση μπορούν να χρησιμοποιηθούν σε συναρτήσεις χωρίς αυτές να είναι συνεχείς (continuous) ή διαφορικές (differentiable) κι αυτό διότι ανήκουν στις αριθμητικές μεθόδους βελτιστοποίησης οι οποίες δεν απαιτούν να έχει βελτιστοποιηθεί το πρόβλημα. Είναι γνωστές ως μέθοδοι άμεσης αναζήτησης (direct-search), ελεύθερες παραγώγου (derivative-free) ή ως μέθοδοι μαύρου κουτιού (black-box). Η λογική της τυχαίας αναζήτησης στηρίζεται στη μετακίνηση σε καλύτερες θέσεις στο χώρο αναζήτησης επαναληπτικά. Οι θέσεις έχουν προκύψει έπειτα από δειγματοληψία στην υπερσφαίρα που περιβάλλει την τρέχουσα θέση. Η λογική της τυχαίας βελτιστοποίησης στηρίζεται στην μετακίνηση σε καλύτερες θέσεις στο χώρο αναζήτησης, οι οποίες όμως έχουν προκύψει από μια κανονική κατανομή (normal distribution) γύρω από την τρέχουσα θέση.

Το 1965 οι Τζον Ασγουόρθ Νέλντερ και Ρότζερ Μεντ (John A. Nelder and Roger Mead)

παρουσίασαν για πρώτη φορά τη μέθοδο *simplex* ευρετική Nelder-Mead. Η μέθοδος αυτή χρησιμοποιείται στη μηχανική και τη στατιστική και έχει αποδειχθεί από τον Μάικλ Τζέιμς Ντέιβιντ Πάουελ (Michael James David Powell) ότι συγκλίνει σε μη στάσιμα σημεία σε κάποια προβλήματα [136]. Το 1966 οι Λόρενς Τζ. Φόγκελ, Αλβιν Τζ. Οουενς και Μάικλ Τζ. Γουόλς (Lawrence, J. Fogel, A.J. Owens and M.J. Walsh) παρουσίασαν στην ερευνητική κοινότητα τον εξελικτικό προγραμματισμό, ο οποίος αποτελεί ένα από τα μεγαλύτερα παραδείγματα εξελικτικών αλγορίθμων. Ο εξελικτικός προγραμματισμός έχει πολλές ομοιότητες με τον γενετικό προγραμματισμό. Ωστόσο, η δομή του προγράμματος που πρόκειται να βελτιστοποιηθεί είναι σταθερή, ενώ μπορούν να εξελιχθούν οι αριθμητικές παράμετροί του [60]. Ο Λόρενς Φόγκελ (Lawrence Fogel) είχε κάνει χρήση του εξελικτικού προγραμματισμού ήδη από το 1960 στις ΗΠΑ, με στόχο να δημιουργήσει τεχνητή νοημοσύνη μέσω της προσομοιωμένης εξέλιξης, η οποία θα λειτουργούσε ως διαδικασία μάθησης. Χρησιμοποιήθηκαν μηχανές πεπερασμένης κατάστασης, οι οποίες στη συνέχεια εξελίχθηκαν, ως προγνωστικοί παράγοντες. Στις μέρες μας, ο εξελικτικός προγραμματισμός είναι δύσκολο να διακριθεί από τις εξελικτικές στρατηγικές, καθώς αποτελεί μια μεγάλη υπολογιστική γλώσσα χωρίς σταθερή δομή σε αντίθεση με τις υπόλοιπες γλώσσες.

Ο αλγόριθμος Μετρόπολις-Χάστινγκς (Metropolis-Hastings) χρησιμοποιείται ιδιαίτερα στη στατιστική και στη στατιστική φυσική και παρουσιάστηκε για πρώτη φορά το 1970, από τον Γουίλιαμ Κιθ Χάστινγκς (William Keith Hastings) [86]. Την ίδια περίοδο κάνουν την εμφάνισή τους δύο νέες μέθοδοι, μια προσαρμογή των παραμέτρων ελέγχου για έναν βελτιστοποιητή από τον Ντάνιελ Τζόζεφ Καβίτσιο (Daniel Joseph Cavicchio) [28] και μια μέθοδος διαμέρισης γράφων η οποία σχετίζεται με την αναζήτηση μεταβλητού βάθους και την αναζήτηση ταμπού βάσει απαγόρευσης από τους Μπράιαν Γουίλσον Κέρνιγκαν και Σεν Λιν (Brian Wilson Kernighan and Shen Lin) [107].

Ο Τζον Χένρι Χόλαντ (John Henry Holland) παρουσίασε το 1975 μια ευρετική αναζήτησης, έναν γενετικό αλγόριθμο ο οποίος τείνει να μιμηθεί τη διαδικασία της φυσικής επιλογής. Το 1977 κάνει την εμφάνισή της η διασκορπισμένη αναζήτηση (Scatter Search - SS) από τον Φρεντ Γκλόβερ (Fred Glover) [70]. Η διασκορπισμένη αναζήτηση προτάθηκε ως μέθοδος επίλυσης προβλημάτων αθέρατου προγραμματισμού, συνδυάζοντας κανόνες απόφασης και περιορισμούς. Ουσιαστικά, έχει τη δυνατότητα να συνδυάζει τις λύσεις από προηγούμενα στάδια και να δημιουργεί καινούρια σημεία αναφοράς, με τη βοήθεια επιπλέον βημάτων, αν είναι αναγκαία, για να προκύψουν εφικτές λύσεις. Το 1978, ένας πίνακας ο οποίος ρυθμίζει τις παραμέτρους ενός βελτιστοποιητή με τη βοήθεια ενός άλλου βελτιστοποιητή παρουσιάζεται από τους Ρόμπερτ Ε. Μέρσερ και Τζόφρι Ρ. Σαμπσόν (Robert E. Mercer and Geoffrey R. Sampson) [125].

Ο γενετικός προγραμματισμός έκανε την εμφάνισή του περί το 1980, όταν προτάθηκε από τον Στέφεν Φ. Σμιθ (Stephen F. Smith) [178] και την ίδια δεκαετία, και πιο συγκεκριμένα το 1983, παρουσιάστηκε και η πρόταση για την προσομοιωμένη απόπτηση (Simulated Annealing - SA) από τους Σκοτ Κιρκπατρικ, Γ. Δ. Τζελαρτ και Μάριο Π. Βέτσι (Scott Kirkpatrick, C.D. Gelatt and Mario P. Vecchi) [108]. Ο γενετικός προγραμματισμός για την τεχνητή νοημοσύνη είναι μια μεθοδολογία βασισμένη σε έναν εξελικτικό αλγόριθμο. Η μεθοδολογία αυτή ουσιαστικά εμπνέεται από τη βιολογική εξέλιξη, με τη βοήθεια της οποίας προκύπτουν προγράμματα του Η/Υ τα οποία εκτελούν μια ορισμένη από τον χρήστη εργασία. Η προσομοιωμένη απόπτηση αποτελεί μια γενική, πιθανολογική μεθυστική μέθοδο για το πρόβλημα της ολικής βελτιστοποίησης. Το πρόβλημα αυτό αφορά την τοποθέτηση μιας καλής προσέγγισης στο καθολικό βέλτιστο για να δοσμένη συνάρτηση σε ένα μεγάλο χώρο αναζήτησης. Η προσομοιωμένη απόπτηση χρησιμοποιείται συχνά όταν ο χώρος αναζήτησης είναι διακριτός - για όλες τις διαδρομές που επισκέπτονται ένα σύνολο πόρων παραδείγματος χάριν - και για ορισμένα προβλήματα είναι πιο αποτελεσματική

από την εξαντλητική απαρίθμηση. Αυτό βεβαίως ισχύει με την προϋπόθεση ότι ο στόχος είναι η εύρεση μιας αποδεκτά καλής λύσης σε ένα καθορισμένο χρονικό διάστημα και όχι η εύρεση της καθολικά βέλτιστης λύσης.

Η αναζήτηση ταμπού (tabu search) πρωτοπαρουσιάστηκε το 1986 [71], μαζί με την πρώτη αναφορά του όρου "μεθευρετικός" ("metaheuristic") από τον Φρεντ Γκλόβερ (Fred Glover). Η αναζήτηση ταμπού είναι μια μεθευρετική μέθοδος αναζήτησης, η οποία κάνει χρήση των τοπικών μεθόδων αναζήτησης της μαθηματικής βελτιστοποίησης. Οι τοπικές μέθοδοι αναζήτησης (ή αναζητήσεις γειτονιάς) εξετάζουν μια πιθανή λύση ενός προβλήματος με το να ελέγχουν τους άμεσους γείτονες αυτής. Εξετάζουν τις παρόμοιες λύσεις και οι οποίες έχουν μικροδιαφορές μεταξύ τους με σκοπό να βρουν μια βελτιωμένη λύση. Βεβαίως, αυτό έχει ως αποτέλεσμα τέτοιες αναζητήσεις να κολλάνε σε μεγάλες περιοχές ή σε υποβέλτιστες, όταν υπάρχουν πολλές λύσεις εξίσου ταιριαστές. Η αναζήτηση ταμπού λύνει αυτό το πρόβλημα και ενισχύει την απόδοση των τεχνικών αυτών, κάνοντας χρήση δομών μνήμης. Αυτές οι δομές καταγράφουν τις λύσεις που έχουν εξεταστεί ήδη ή τα σύνολα κανόνων που παρέχονται από τον χρήστη [72]. Μια πιθανή λύση χαρακτηρίζεται ως "ταμπού", ως απαγορευμένη δηλαδή, όταν την έχει επισκεφθεί προηγουμένως η μέθοδος αναζήτησης σε ένα ορισμένο και εύλογο χρονικό διάστημα ή όταν έχει παραβιάσει έναν κανόνα. Ο αλγόριθμος λαμβάνει υπόψιν του τις απαγορευμένες λύσεις και αποφεύγει να τις επαναλάβει.

Οι μιμητικοί αλγόριθμοι (Memetic Algorithms - MA), οι οποίοι αποτελούν έναν πρόσφατα αναπτυσσόμενο κλάδο έρευνας στην εξελικτική υπολογιστική, έκαναν αισθητή την παρουσία τους από το 1989, όπου και προτάθηκαν από τον Πάμπλο Μοσκάτο (Pablo Moscato) [132]. Το 1992 ο Μάρκο Ντορίνγκο (Marco Dorigo) δημοσίευσε τη διδακτορική του διατριβή, η οποία έμελλε να αποτελέσει μια από τις πιο πρόσφατες συνεισφορές στον τομέα των μεθευρετικών, διότι εισήγαγε την ιδέα της βελτιστοποίησης μιας αποικίας μυρμηγκιών (ant colony optimization) [48]. Αυτή η τεχνική βελτιστοποίησης - βελτιστοποίηση μιας αποικίας μυρμηγκιών (ant colony optimization - ACO) - είναι ιδιαίτερα χρήσιμη για την επίλυση υπολογιστικών προβλημάτων, καθώς μπορεί να μειωθεί η δυσεπιλυσιμότητά τους με την εξεύρεση καλών μονοπατιών διαμέσου γραφημάτων. Ουσιαστικά είναι μια πιθανολογική τεχνική η οποία προσφέρει λύσεις σε συγκεκριμένα προβλήματα.

Επίσης σημαντική συνεισφορά στον τομέα αποτέλεσε και η απόδειξη του θεωρήματος "μη δωρεάν γευμάτων" ("no free lunch" theorem) [9] από τους Ντέιβιντ Χ. Γουόλπερτ και Γουίλιαμ Τσαρλς Μακρεντι (David H. Wolpert and William Charles Macready) το 1995 [197]. Το θεώρημα "no free lunch" δηλώνει ότι για ορισμένους τύπους μαθηματικών προβλημάτων, το υπολογιστικό κόστος για την εξεύρεση λύσης είναι το ίδιο για κάθε μέθοδο λύσης κι αυτό προκύπτει εξισώνοντας το υπολογιστικό κόστος με το μέσο όρο του κόστους πάνω από όλα τα προβλήματα της κλάσης. Ως εκ τούτου, με αυτό το θεώρημα το οποίο χρησιμοποιείται κατ' εξοχήν στην υπολογιστική πολυπλοκότητα και τη βελτιστοποίηση, φαίνεται ότι καμία λύση δεν είναι καλύτερη από οποιαδήποτε άλλη.

2.1.3 Εφαρμογές

Οι μεθευρετικές τεχνικές είναι ιδανικές για συνδυαστική βελτιστοποίηση στις περιπτώσεις στις οποίες επιδιώκεται η εύρεση της βέλτιστης λύσης σε έναν διακριτό χώρο αναζήτησης. Για να γίνει πιο κατανοητό αυτό, στο πρόβλημα του πλανόδιου πωλητή, όπου ο χώρος αναζήτησης των υποψήφιων λύσεων αναπτύσσεται ταχύτερα από εκθετικά καθώς το μέγεθος του προβλήματος αυξάνεται, η μέθοδος της εξαντλητικής αναζήτησης προς εύρεση της βέλτιστης λύσης

είναι ανούσια. Σε προβλήματα τα οποία είναι πολυδιάστατα και συνδυαστικά, όπως τα περισσότερα προβλήματα σχεδιασμού στη μηχανική, έχουν τον παράγοντα της διάστασης, ο οποίος καθιστά επίσης ανέφικτη την εξαντλητική αναζήτηση ή τις αναλυτικές μεθόδους. Τέτοια προβλήματα είναι για παράδειγμα, το πρόβλημα της εύρεσης της μορφής και το πρόβλημα εύρεσης της συμπεριφοράς. Στην περίπτωση των συνδυαστικών προβλημάτων, δημοφιλείς μεθευρετικές είναι η προ-σομοιωμένη ανόπτηση [108], οι γενετικοί αλγόριθμοι των Χόλαντ κ.α. (Holland et al.) [89], η αναζήτηση διασποράς [70] και η αναζήτηση ταμπού του Γκλόβερ (Glover) [71].

2.1.4 Ιδιότητες και κατηγοριοποίηση

Οι χαρακτηριστικές ιδιότητες των περισσότερων μεθευρετικών παρουσιάζονται ως εξής:

1. Οι μεθευρετικές είναι στρατηγικές οι οποίες καθοδηγούν τη διαδικασία της αναζήτησης,
2. έχουν ως στόχο την αποτελεσματική διερεύνηση του χώρου αναζήτησης με σκοπό την εύρεση σχεδόν βέλτιστων λύσεων,
3. ποικίλλουν από απλές διαδικασίες τοπικής αναζήτησης μέχρι πολύπλοκες διαδικασίες μάθησης,
4. οι αλγόριθμοι τους είναι προσεγγιστικοί και συνήθως μη-ντετερμινιστικοί,
5. δεν αφορούν κάποιο συγκεκριμένο πρόβλημα,
6. είναι πιθανό να ενσωματώνουν μηχανισμούς διαχείρισης σε περίπτωση παγίδευσης σε συγκεκριμένες περιοχές του χώρου αναζήτησης,
7. έχουν μια περιγραφή αφηρημένου επιπέδου, καθώς οι βασικές τους έννοιες το επιτρέπουν αυτό,
8. επιτρέπουν επίσης μια εύκολη παράλληλη υλοποίηση
9. και τέλος, οι μεθευρετικές χρησιμοποιούν γνώση από άλλους τομείς με τη μορφή των ευρετικών, οι οποίες ελέγχονται από τη στρατηγική του ανώτερου επιπέδου.

Υπάρχει μια πληθώρα μεθευρετικών τεχνικών, καθώς και ένα σύνολο ιδιοτήτων βάσει των οποίων μπορεί να γίνει ένας διαχωρισμός αυτών [20]. Ένας διαχωρισμός μπορεί να γίνει βάσει του τύπου στρατηγικής της αναζήτησης. Ο τύπος στρατηγικής της αναζήτησης είναι μια βελτίωση σε σχέση με τους απλούς αλγόριθμους τοπικής αναζήτησης. Τέτοιου είδους μεθευρετικές είναι η αναζήτηση ταμπού, η προσομοιωμένη ανόπτηση, η αναζήτηση μεταβαλλόμενης γειτονιάς, η επαναλαμβανόμενη τοπική αναζήτηση και η άπληστη τυχαιοποιημένη προσαρμοστική διαδικασία αναζήτησης (Greedy Randomized Adaptive Search Procedure - GRASP) [20]. Ο άλλος διαχωρισμός μπορεί να γίνει βάσει μιας μαθησιακής συνιστώσας για την αναζήτηση. Τέτοιου είδους μεθευρετικές είναι η βελτιστοποίηση μιας αποικίας μυρμηγκιών, η εξελικτική υπολογιστική και οι γενετικοί αλγόριθμοι.

Ακόμη, μπορεί να γίνει διαχωρισμός σύμφωνα με το αν οι αναζητήσεις επιστρέφουν μοναδική λύση ή αν είναι βασισμένες στον πληθυσμό [20] [182]. Οι πρώτες, οι αναζητήσεις που επιστρέφουν μια μοναδική λύση, εστιάζουν σε μία λύση και την τροποποιούν και τη βελτιώνουν μέχρι το επιθυμητό αποτέλεσμα. Ως μεθευρετικές μοναδικής λύσης χαρακτηρίζονται η προσομοιωμένη ανόπτηση, η αναζήτηση μεταβαλλόμενης γειτονιάς, η επαναλαμβανόμενη τοπική αναζήτηση και η καθοδηγούμενη τοπική αναζήτηση [182]. Οι δεύτερες, οι αναζητήσεις που βασίζονται στον πληθυσμό, δεν εστιάζουν σε μία λύση. Αντίθετα, βελτιώνουν πολλαπλές υποψήφιες λύσεις και καθοδηγούν την αναζήτηση χρησιμοποιώντας χαρακτηριστικά του πληθυσμού. Τέτοιες μεθευρετικές είναι η εξελικτική υπολογιστική, οι γενετικοί αλγόριθμοι και η βελτιστοποίηση σμήνους σωματιδίων [182].

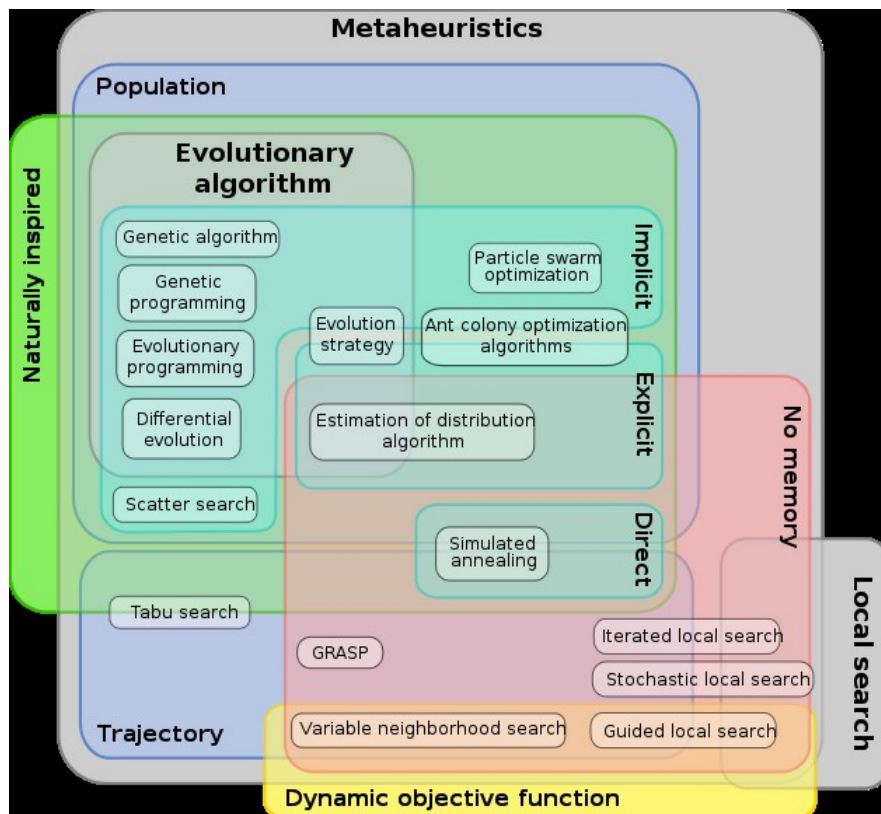
Μια ακόμα κατηγορία μεθευρετικών η οποία προκύπτει είναι αυτή της νοημοσύνης του σμήνους (swarm optimization). Αναφέρεται στην συλλογική συμπεριφορά των αποκεντρωμένων, αυτο-οργανωμένων μέσα σε ένα πληθυσμό ή σμήνος. Παραδείγματα αυτής της κατηγορίας αποτελούν η βελτιστοποίηση αποικίας μυρμηγκιών [48], η βελτιστοποίηση σμήνους σωματιδίων [182] και η τεχνητή αποικία μελισσών [97].

Όλοι οι προαναφερθέντες μεθευρετικοί αλγόριθμοι λειτουργούν με σειριακό τρόπο. Εκτός από αυτές τις περιπτώσεις, υπάρχουν και οι υβριδικοί και οι παράλληλοι μεθευρετικοί αλγόριθμοι [182]. Η υβριδική μεθευρετική μέθοδος είναι ένας συνδυασμός από προσεγγίσεις βελτιστοποίησης, όπως είναι αλγόριθμοι μαθηματικού προγραμματισμού, μηχανικής μάθησης και προγραμματισμού με περιορισμούς. Τα συστατικά μιας υβριδικής μεθευρετικής μπορούν να εκτελούνται ταυτόχρονα και να παρέχουν πληροφορίες το ένα στο άλλο ώστε να καθοδηγείται η αναζήτηση.

Η παράλληλη μεθευρετική μέθοδος κάνει χρήση τεχνικών του παράλληλου προγραμματισμού με σκοπό να εκτελέσει παράλληλα πολλαπλές μεθευρετικές αναζητήσεις. Υπάρχουν διάφορες τέτοιες τεχνικές οι οποίες αλληλεπιδρούν μεταξύ τους, από απλά καταναεμημένα σχήματα σε ταυτόχρονες εκτελέσεις αναζήτησης, με σκοπό να βελτιώσουν τη συνολική λύση.

Η εικόνα 2.1 παρουσιάζει συνοπτικά τις διαφορετικές κατηγοριοποιήσεις των μεθευρετικών μεθόδων [194]

Figure 2.1: Κατηγοριοποιήσεις των μεθευρετικών [194]



2.1.5 Κριτήρια σύγκρισης μεθευρετικών

Σκοπός όλων των μεθευρετικών είναι να διαφεύγουν των τοπικών βέλτιστων (*local optima*), παρόλο που οι διαφορές μεταξύ των διαφόρων μεθευρετικών μεθόδων είναι σημαντικές. Κάθε μέθοδος είναι μοναδική ως προς τη λειτουργικότητά της και αυτό καθιστά την σύγκριση μεταξύ τους πολύ πιο δύσκολη από τις υπόλοιπες αλγοριθμικές συγκρίσεις [174]. Παρόλα αυτά, μπορεί να γίνει σύγκριση των μεθευρετικών, εκτός από την απλότητά τους (*simplicity*), ως προς τον χρόνο εκτέλεσης (*runtime*) και ως προς την ποιότητα της παρεχόμενης λύσης (*solution quality*).

Το πρώτο κριτήριο, ο χρόνος εκτέλεσης, είναι ένας σημαντικός παράγοντας και καθοριστικό στοιχείο μιας μεθευρετικής. Ο γρήγορος χρόνος εκτέλεσης μιας μεθευρετικής είναι τόσο σημαντικός, όσο και η ίδια η βέλτιστη λύση. Ο μόνος λόγος για να διαλέξουμε μια μεθευρετική έναντι μιας ευρετικής ή ενός ακριβή αλγορίθμου είναι η ταχύτητα που προσφέρει μια τέτοια μέθοδος. Βέβαια, συναντά κανείς ένα σημαντικό πρόβλημα, το οποίο έγκειται στο γεγονός ότι οι συγκρίσεις του χρόνου εκτέλεσης είναι από τις πιο δύσκολες για να διεκπεραιωθούν, καθώς ο χρόνος εκτέλεσης κάθε αλγορίθμου διαφέρει. Αυτή η διαφορά προκύπτει συγκεκριμένα όταν οι προς εξέταση αλγόριθμοι έχουν μεταγλωττιστεί σε διαφορετικά μηχανήματα (υπολογιστές), με διαφορετικούς μεταγλωττιστές και σημαίες μεταγλώττισης και πιθανώς σε διαφορετικά "testbeds" (δηλαδή πλατφόρμες πειραματισμού για μεγάλα έργα ανάπτυξης) [174].

Το δεύτερο κριτήριο, η καλή και ποιοτική λύση, αποτελεί επίσης κρίσιμο παράγοντα μιας μεθευρετικής. Οι μεθευρετικές σχεδιάστηκαν εξ' αρχής για να είναι καλύτερες από τις ακριβείς προσεγγίσεις, καθώς δίνουν ποιοτικές λύσεις ως προς τον χρόνο εκτέλεσης. Το νόημα εξάλλου μιας μεθευρετικής είναι ακριβώς αυτό, οι ποιοτικές και αποδεκτές λύσεις βασισμένες σε κάποιον ορισμό ως προς το τι θεωρείται αποδεκτό. Φυσικά, το ποσοστό επιτρεπτής απόκλισης της λύσης που θα προκύψει διαφέρει από πρόβλημα σε πρόβλημα και από εφαρμογή σε εφαρμογή. Ένα υψηλό ποσοστό επιτρεπτής απόκλισης σε ένα πρόβλημα βελτιστοποίησης, το οποίο θα χρησιμοποιηθεί για βραχυπρόθεσμο σχεδιασμό, είναι ικανοποιητικό ενώ ένα χαμηλότερο ποσοστό επιτρεπόμενου σφάλματος μπορεί να είναι περισσότερο ταιριαστό σε προβλήματα μακροπρόθεσμου σχεδιασμού. Ακόμη, όμως, και για ένα πρόβλημα, είναι πιθανό να διαφέρουν αισθητά οι τιμές των ποσοστών των επιτρεπτών σφαλμάτων. Συνεπώς, λόγω αυτού του προβλήματος, είναι προφανές ότι είναι δύσκολο να προσδιοριστεί η ποιότητα της λύσης που θα προκύψει για ένα πρόβλημα συνδυαστικής βελτιστοποίησης, ακόμα και αδύνατο κάποιες φορές. Γι' αυτόν τον λόγο, οι ευρετικές πρέπει να συγκρίνονται μεταξύ τους και όχι μόνο να συγκρίνονται μεμονωμένα ως προς το όριο της ποιότητας της λύσης που προσφέρουν [174].

2.2 Η Αναζήτηση Μεταβαλλόμενης Γειτονιάς

Η πλέον κατάλληλη τεχνική συνδυαστικής και καθολικής βελτιστοποίησης προβλημάτων είναι η αναζήτηση μεταβαλλόμενης γειτονιάς (*Variable Neighborhood Search - VNS*), η οποία αποτελεί μια μεθευρετική προσέγγιση [84] [127]. Η κλασική μεθευρετική VNS εφαρμόζει διαρκώς την αλλαγή γειτονιάς σε κάθεδο σε τοπικά ελάχιστα και σε απόδραση από τις κοιλάδες. Η VNS περιλαμβάνει και διάφορες επεκτάσεις για να ικανοποιηθούν οι ανάγκες μιας πληθώρας προβλημάτων και να βρεθεί με αποτελεσματικότητα η βέλτιστη ή η σχεδόν βέλτιστη λύση. Η κάθε τεχνική VNS περιλαμβάνει τρία στάδια. Το πρώτο στάδιο ονομάζεται *shaking* (φάση της διαφοροποίησης) και είναι η διαδικασία κατά την οποία το πρόβλημα ξεφεύγει από τις τοπικές βέλτιστες λύσεις. Το δεύτερο στάδιο είναι η αλλαγή γειτονιάς. Ουσιαστικά, σε αυτό το στάδιο καθορίζεται η επόμενη δομή γειτονιάς η οποία θα εξερευνηθεί στη συνέχεια. Κατά τη διάρκεια αυτής της φάσης θα εφαρμοστεί και ένα κριτήριο έγκρισης ή απόρριψης στην τελευταία λύση η

οποία θα προκύψει. Το τρίτο στάδιο αποτελεί την φάση της βελτίωσης (εντατικοποίηση). Αυτό το στάδιο περιλαμβάνει την εξερεύνηση των δομών της γειτονιάς με την εφαρμογή διαφόρων τοπικών αναζητήσεων. Για να αλλαχθεί η περιοχή σε αυτό το στάδιο, υπάρχουν κάποια βήματα τα οποία πρέπει να ακολουθηθούν και αυτά είναι:

- το κυκλικό βήμα αλλαγής γειτονιάς, κατά το οποίο η αναζήτηση προχωρά σε επόμενη δομή γειτονιάς της λίστας, είτε υπάρχει βελτίωση σε κάποια γειτονιά είτε όχι,
- αν η τρέχουσα λύση βελτιώνεται σε κάποια γειτονιά, η εξερεύνηση σε αυτή τη γειτονιά θα συνεχιστεί.
- και τέλος, η διαταραχή γειτονιάς, κατά την οποία γίνεται εξερεύνηση κοιλάδων μακριά από μια συγκεκριμένη λύση. Σε αυτό το βήμα, γίνονται δεκτές λύσεις που μπορεί να βελτιώνουν τις λύσεις ή ακόμα και να είναι χειρότερες από την τρέχουσα βέλτιστη.

Μια πιθανή λύση αξιολογείται σύμφωνα με τις αντικειμενικές αξίες του προβλήματος και την βέλτιστη λύση. Η περιοχή αλλάζει σε δύο φάσεις της VNS. Η πρώτη φάση αφορά μια κάθοδο για να βρεθεί μια τοπική βέλτιστη λύση και η δεύτερη την φάση διασταύρωσης για να βγει από την κοιλάδα. Η VNS βρίσκει εφαρμογή σε πολυάριθμα πεδία πλέον. Χρησιμοποιείται ευρέως σε προβλήματα θεωρίας τοποθεσίας, ανάλυσης συμπλεγμάτων, στον προγραμματισμό, στην δρομολόγηση οχημάτων, στον σχεδιασμό δικτύων, στην τεχνητή νοημοσύνη, την μηχανική, τη βιολογία και πλήθος άλλων πεδίων.

2.2.1 Αναζήτηση Μεταβαλλόμενης Γειτονιάς

Η μεθευρετική μέθοδος της αναζήτησης μεταβαλλόμενης γειτονιάς (Variable Neighborhood Search - VNS) είναι ιδιαίτερα χρήσιμη για την επίλυση προβλημάτων συνδυαστικής και ολικής βελτιστοποίησης, καθώς εξετάζει τις μακρινές γειτονιές μιας τρέχουσας λύσης και κινείται από εκεί σε μια νέα και βελτιωμένη λύση. Για να ληφθούν τα τοπικά βέλτιστα (local optima) των προηγούμενων λύσεων από τις γειτονιές γίνεται χρήση της μεθόδου τοπικής αναζήτησης (Local Search - LS) αλλά επαναληπτικά (iterated local search - ILS). Δηλαδή, η αναζήτηση VNS εξερευνά επαναληπτικά τις μεγαλύτερες γειτονιές για ένα τοπικό βέλτιστο, μέχρι να βρεθεί μια βελτιωμένη λύση. Μόλις βρεθεί κάτι τέτοιο, επαναλαμβάνεται η έρευνα στις επεκτεινόμενες γειτονιές. Έτσι, προκύπτουν τρεις βασικές αρχές για αυτή τη στρατηγική:

1. ένα τοπικό ελάχιστο για μία δομή γειτονιάς μπορεί να μην αποτελεί τοπικό ελάχιστο για μια διαφορετική δομή γειτονιάς,
2. ένα ολικό ελάχιστο είναι ένα τοπικό ελάχιστο για όλες τις πιθανές δομές της γειτονιάς, και
3. τα τοπικά ελάχιστα είναι στενά συνδεδεμένα με τα ολικά ελάχιστα για πολλές κλάσεις προβλημάτων.

Η αρχική λύση επιλέγεται από την ευρετική τοπικής αναζήτησης, η οποία εκτελείται ξεκινώντας. Εάν ανακαλύψει μια κάθοδο εντός μιας γειτονιάς προχωρά προς το ελάχιστο της εντός της γειτονιάς. Εάν δεν ανακαλύψει κάποια κάθοδο σταματά την εκτέλεσή της η ευρετική και η τρέχουσα λύση παραμένει και η λύση του προβλήματος, χωρίς να έχει βρεθεί κάποια καλύτερη. Στην αντίθετη περίπτωση επαναλαμβάνεται. Συνήθως η βέλτιστη λύση είναι η υψηλότερη κάθοδος που προκύπτει από την πλήρη εξερεύνηση κάθε γειτονιάς σε κάθε βήμα.

Η μεθευρετική VNS σχεδιάστηκε με σκοπό την εύρεση προσεγγιστικών λύσεων για προβλήματα βελτιστοποίησης, τόσο διακριτά όσο και συνεχή. Στόχος της μεθόδου αυτής είναι η επίλυση προβλημάτων γραμμικού προγραμματισμού, αχέραιου προγραμματισμού, μικτού και αχέραιου προγραμματισμού, μη γραμμικού προγραμματισμού και άλλων προβλημάτων. Προτάθηκε το 1997 από τους Μλαντένοβιτς και Χάνσεν (Mladenović και Hansen) [127].

Η VNS και οι προεκτάσεις της έχουν το πλεονέκτημα ότι είναι απλές μέθοδοι και συχνά λειτουργούν με μερικές ή ακόμη και καθόλου παραμέτρους, αντίθετα με πολλές άλλες μεθόδους. Σαν αποτέλεσμα, προσφέρουν πολύ καλές λύσεις με απλούστερους τρόπους σε σχέση με τις υπόλοιπες μεθόδους και σαν αποτέλεσμα έχει καλύτερη αποδοτικότητα με πιο εξελιγμένες υλοποιήσεις.

2.2.2 Μαθηματική Περιγραφή της VNS

Στην υποενότητα αυτή, αναφέρεται η μαθηματική αναπαράσταση της μεθόδου VNS. Έστω ένα ντετερμινιστικό πρόβλημα βελτιστοποίησης με:

$$\min\{f(x)|x \in X, X \subseteq S\} \quad (2.1)$$

όπου S, X, x και f είναι ο χώρος λύσης, το εφικτό σύνολο, μια εφικτή λύση και μια αντικειμενική συνάρτηση πραγματικών τιμών αντίστοιχα. Ένα πρόβλημα συνδυαστικής βελτιστοποίησης μπορεί να προκύψει εάν το S είναι ένα πεπερασμένο αλλά μεγάλο σύνολο. Αντίστοιχα στην περίπτωση όπου $S = \mathbb{R}^n$, προκύπτει ένα μοντέλο συνεχούς βελτιστοποίησης.

Μια λύση $x^* \in X$ είναι βέλτιστη, εάν ισχύει ότι $f(x^*) \leq f(x)$, με $\forall x \in X$. Για να είναι ακριβής ο παραπάνω αλγόριθμος (2.1) φτάνει να βρεθεί μια βέλτιστη λύση x^* , με μια βέλτιστη επικυρωμένη δομή. Αν δεν μπορεί να επικυρωθεί η δομή, η μέθοδος θα δείξει ότι δεν υπάρχει εφικτή λύση, $X = \emptyset$ ή ότι η λύση δεν μπορεί να διευκρινιστεί. Για τέτοιες μεθόδους και προβλήματα πρέπει να καθορίζεται ο υπολογιστικός χρόνος και να είναι πεπερασμένος και σύντομος. Στην επαναλαμβανόμενη βελτιστοποίηση ο αλγόριθμος θα σταματήσει κάποια δεδομένη στιγμή, καθώς υπάρχει ένας βαθμός ανοχής (tolerance). Η διαδικασία τερματίζει δηλαδή, όταν μια εφικτή λύση x^* έχει εντοπιστεί, τέτοια ώστε $f(x^*) \leq f(x) + \varepsilon, \forall x \in X$ ή $(f(x^*) - f(x))/f(x^*) < \varepsilon, \forall x \in X$.

Σε κάποιες περιπτώσεις, κάποιες ευρετικές μέθοδοι παρουσιάζουν μια βέλτιστη λύση ή μια προσεγγιστική λύση χωρίς να έχουν επικυρώσει εάν είναι όντως βέλτιστη. Μπορεί να έχει δοθεί εσφαλμένη πιστοποίηση στη λύση, εάν ικανοποιεί τα κριτήρια για κάποιο λόγο αν και αυτές οι περιπτώσεις είναι σπάνιες. Στην προσπάθεια να αποφύγουν τον τεράστιο υπολογιστικό χρόνο, οι ευρετικές μέθοδοι έρχονται αντιμέτωπες με το πρόβλημα των τοπικών βέλτιστων. Ένα τοπικό βέλτιστο x_L του προβλήματος είναι τέτοιο ώστε $f(x_L) \leq f(x), \forall x \in N(x_L) \cap X$, όπου $N(x_L)$ δηλώνει μια γειτονιά του τοπικού βέλτιστου x_L .

Μια ευρετική τοπική αναζήτηση ξεκινά με την επιλογή της τρέχουσας λύσης x και καθορίζεται μια διαδρομή καθόδου από το x , μέσα στα πλαίσια της γειτονιάς $N(x)$ με κατεύθυνση προς το ελάχιστο της $f(x)$ μέσα στην $N(x)$ προς την ίδια κατεύθυνση. Εάν δεν υπάρχει κατεύθυνση καθόδου η ευρετική σταματά, διαφορετικά θεωρείται επαναλαμβανόμενη. Κατά κύριο λόγο χρησιμοποιείται ως η καλύτερη βέλτιστη λύση το υψηλότερο σημείο της καθόδου και συμβολίζεται με x' . Σε κάθε βήμα της μεθόδου, εξερευνάται η γειτονιά N_x του x και συνήθως σε αυτό το βήμα χρησιμοποιείται η ευρετική μέθοδος.

2.2.3 Εκδοχές της VNS

Οι βασικές επεκτάσεις και πλαίσια της VNS είναι αρκετά απλά και με ελάχιστες απαιτήσεις, σε σχέση με άλλες μεθόδους και κάποιες φορές δεν απαιτούν παραμέτρους. Συνεπώς, η VNS προσφέρει πολύ καλές λύσεις με απλούς τρόπους συγκριτικά με άλλες μεθόδους και αυτό μπορεί να οδηγήσει σε ακόμα πιο αποτελεσματικές και πολυσύνθετες εφαρμογές. Έχουν προκύψει κατά καιρούς πολλές παραλλαγές της VNS και χρησιμοποιούνται για την επίλυση πιο σύνθετων και δυσεπίλυτων προβλημάτων βελτιστοποίησης.

Η πρώτη προς εξέταση επέκταση της VNS είναι η κατάβαση μεταβαλλόμενης γειτονιάς (Variable Neighborhood Descent - VND). Σε αυτή την εκδοχή, η βασική VNS επεκτείνεται σε ένα σχήμα δύο επιπέδων και είναι ντετερμινιστική μέθοδος ως προς την αλλαγή των γειτονιών[25]. Υποτίθεται ότι η αρχική λύση δίνεται και γίνεται χρήση ελάχιστων γειτονιών κατά την κατάβαση των ευρετικών τοπικής αναζήτησης. Η τελική λύση προκύπτει ως το τοπικό ελάχιστο σε σχέση με όλες τις γειτονιές, κάτι που συνεπάγεται ότι οι πιθανότητες να βρεθεί το ολικό ελάχιστο είναι αυξημένες όταν γίνεται χρήση της VND.

Η δεύτερη εκδοχή είναι η μειωμένη VNS (Reduced VNS - RVNS) και έχει ως σκοπό την αύξηση της αποτελεσματικότητας ή της ταχύτητας. Είναι στοχαστική μέθοδος [83]. Ουσιαστικά, σε αυτήν την παραλλαγή, οι λύσεις διαλέγονται τυχαία από τις γειτονιές του προβλήματος και αντικαθίστανται από άλλες καλύτερες, μόνο αν το αποτέλεσμα της αντικειμενικής συνάρτησης προκύπτει καλύτερο. Το αποτέλεσμα αυτής της μεθόδου είναι ταχύτερο και με καλά - αν και όχι πάντα τα βέλτιστα - αποτελέσματα.

Η τρίτη εκδοχή είναι η γενική εκδοχή της VNS (General VNS - GVNS) και χρησιμοποιεί ως μέθοδο τοπικής αναζήτησης την κατάβαση μεταβαλλόμενης γειτονιάς (VND). Η εκδοχή αυτή είναι ιδιαίτερα αποτελεσματική σε προβλήματα όπως ο πλανόδιος πωλητής, η δρομολόγηση οχημάτων και άλλα παρεμφερή [128].

Η τέταρτη εκδοχή είναι η "πειραγμένη" VNS (skewed VNS - SVNS), στόχος της οποίας είναι εξερεύνηση κοιλάδων πέρα από την κοιλάδα της τρέχουσας λύσης. Όπως φαίνεται από την έρευνα του Χάνσεν (Hansen) [84] τα τοπικά βέλτιστα τείνουν να βρίσκονται στην κορυφή των μεγαλύτερων λόφων, οπότε η εκδοχή αυτή επικεντρώνεται στην τοπολογία των τοπικών βέλτιστων και στους λόφους και τις πεδιάδες των γειτονιών.

2.2.4 Σχετικά με τη VNS και την GVNS

Αξίζει να αναφερθεί κανείς στα ποικίλα χαρακτηριστικά στοιχεία της VNS. Η απλότητα της VNS είναι από τα πιο σημαντικά στοιχεία, καθώς είναι σαφής και καθίσταται με αυτόν τον τρόπο καθολικά εφαρμόσιμη, σε οποιοδήποτε πρόβλημα βελτιστοποίησης και αν χρησιμοποιηθεί.

Η ακρίβεια της VNS είναι άλλο ένα ιδιαίτερο στοιχείο της, καθώς διατυπώνεται με ακριβείς μαθηματικούς ορισμούς και δεν αφήνει περιθώρια λάθους, υπάρχει συνοχή σε όλες τις ευρετικές μεθόδους που χρησιμοποιούνται για την επίλυση προβλημάτων και στις ενέργειες αυτών και φυσικά η αποτελεσματικότητα της VNS είναι ίσως το πιο σημαντικό στοιχείο της. Παρέχει βέλτιστες ή σχεδόν βέλτιστες λύσεις για τα περισσότερα προβλήματα τα οποία καλείται να επιλύσει.

Η απόδοση της VNS με μέτριο υπολογιστικό χρόνο έχει ως αποτέλεσμα βέλτιστες ή κοντά στις βέλτιστες λύσεις, η ανθεκτικότητά της απέναντι σε μια σειρά προβλημάτων δηλώνει συνέπεια ως προς τη λειτουργία της και η εργονομία της εκδηλώνεται καθώς δεν έχει παραμέτρους, οπότε η κατανόηση και η χρήση της καθίστανται εύκολες διαδικασίες.

Η καινοτομία της VNS προκύπτει από τους νέους τύπους εφαρμογών, η γενικότητά της από τα καλά αποτελέσματα που παράγει για μια πληθώρα διαφορετικών προβλημάτων, η διαδραστικότητά της επιτρέπει στον χρήστη να προσθέτει τα δικά του στοιχεία για τη βελτίωση της διαδικασίας και η πολλαπλότητά της έγκειται στο ότι είναι σε θέση να παράγει διάφορες, συγκεκριμένες, σχεδόν βέλτιστες λύσεις από τις οποίες ο χρήστης είναι ελεύθερος να διαλέξει.

Το 2000 ελάχιστες δημοσιεύσεις είχαν πραγματοποιηθεί σχετικά με τη VNS. Σταδιακά, υπήρξε μια αύξηση περί των δώδεκα άρθρων, το 2007 ο αριθμός έφτασε τις 50 δημοσιεύσεις με αποκορύφωμα πλέον να δημοσιεύονται εκατοντάδες άρθρα σχετικά με το θέμα αυτό κάθε χρόνο. Φαίνε-

ται ότι το ενδιαφέρον γύρω από τη VNS αυξάνεται και όλο και περισσότεροι ερευνητές επικεντρώνονται σε αυτό το θέμα, με ραγδαίους ρυθμούς.

2.2.5 Κβαντικά εμπνευσμένη GVNS

Γενικά, ο μη συμβατικός υπολογισμός (unconventional computing) είναι ένα ευρύ φάσμα νέων ή ασυνήθιστων προτεινόμενων υπολογιστικών μοντέλων. Ένα μέρος αυτών των υπολογιστικών μοντέλων είναι ο φυσικός υπολογισμός (natural computing). Μπορούμε να ορίσουμε τον φυσικό υπολογισμό ως τη διαδικασία που μας κάνει να σκεφτούμε «φυσικά» για τις υπολογιστικές διαδικασίες [141]. Ο υπολογισμός που εμπνέεται από τη φύση έχει αναδειχθεί ως ένα αποτελεσματικό παράδειγμα για το σχεδιασμό και την προσομοίωση καινοτόμων υπολογιστικών μοντέλων εμπνευσμένων από φυσικά φαινόμενα για την επίλυση σύνθετων μη γραμμικών και δυναμικών προβλημάτων. Μερικά από τα γνωστά υπολογιστικά συστήματα και οι αλγόριθμοι που εμπνέονται από τη φύση είναι [46]:

1. Εξελικτικοί αλγόριθμοι εμπνευσμένοι από τα βιολογικά συστήματα.
2. Αλγόριθμοι σμήνους πληροφοριών (Swarm Intelligence Algorithms) εμπνευσμένοι από τη συμπεριφορά του σμήνους / της ομάδας των παραγόντων.
3. Κοινωνικοί και πολιτιστικοί αλγόριθμοι εμπνευσμένοι από τις ανθρώπινες αλληλεπιδράσεις και τις πεποιθήσεις στην κοινωνία.
4. Αλγόριθμοι με κβαντική έμπνευση, εμπνευσμένοι από την κβαντική φυσική.

Οι κβαντικά εμπνευσμένες διαδικασίες χρησιμοποιούν τις βασικές αρχές του κβαντικού υπολογισμού. Ο κβαντικός υπολογισμός (Quantum computing), είναι ένα υποσύστημα του φυσικού υπολογισμού (natural computing) και ένα πεδίο που εισήχθη σχετικά πρόσφατα (στη δεκαετία του 1980) από τον Ρίτσαρντ Φέινμαν (Richard Feynman), ο οποίος δήλωσε ότι δεν είναι δυνατή η αποτελεσματική προσομοίωση ενός πραγματικού κβαντικού συστήματος χρησιμοποιώντας έναν κλασσικό υπολογιστή, καθώς θα υπήρχε μια εκθετική επιβράδυνση όταν προσομοιώνουν τις πραγματικές κβαντικές διεργασίες [57, 58]. Ο κβαντικός υπολογισμός, μια γενική έννοια που αναγνωρίζει τη διαδικασία πληροφορικής ως φυσικό φαινόμενο, θα μπορούσε να είναι μια σημαντική προσθήκη στις υφιστάμενες τυποποιημένες διαδικασίες υπολογισμού. Ο κβαντικός υπολογισμός συνδυάζει ορισμούς, φορμαλισμούς και έννοιες από διάφορες φυσικές επιστήμες εκτός από την επιστήμη των υπολογιστών. Περιλαμβάνει κυρίως τα μαθηματικά (π.χ. γραμμική άλγεβρα) και τη φυσική (κβαντική μηχανική). Σε αυτό το μέρος αναφέρονται μόνο τα πιο απαραίτητα από την πλευρά της υπολογιστικής επιστήμης.

Υπάρχουν δύο τύποι κβαντικών καταστάσεων: καθαρές και μίκτες καταστάσεις. Μια καθαρή κατάσταση είναι μια κατάσταση που αντιπροσωπεύεται από ένα μόνο διάνυσμα

Στον φορμαλισμό Dirac, κάθε κατάσταση της μηχανής είναι μια υπέρθεση των καταστάσεων βάσης $|\psi\rangle$.

Κάθε κατάσταση έχει τη μορφή:

Ένας κβαντικός καταχωρητής είναι το κβαντικό ανάλογο ενός κλασσικού μητρώου επεξεργαστή. Μια μαθηματική περιγραφή ενός κβαντικού καταχωρητή επιτυγχάνεται με τη χρήση ταυστικών γινομένων των qubit bra ή ket. Πιο συγκεκριμένα, ένας n -qubit κβαντικός καταχωρητής αντιπροσωπεύεται από το στοιχείο

$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_n\rangle$, για παράδειγμα ένα ket στο ταυστικό γινόμενο στον χώρο του Hilbert είναι:

$H = H_1 \otimes H_2 \otimes \dots \otimes H_n$, όπου n είναι ο αριθμός των qubits.

Η υλοποίηση της μεθόδου μας εμπνέεται από αυτές τις χβαντικές αρχές [81].

2.2.6 Περιγραφή της χβαντικά εμπνευσμένης **GVNS (qGVNS)**

Όπως και η κλασική **GVNS**, η χβαντικά-εμπνευσμένη **GVNS (qGVNS)** αποτελείται από μια τοπική αναζήτηση **VND**, μια διαδικασία διαταραχής και ένα βήμα αλλαγής γειτονιάς. Πιο συγκεκριμένα, στην προτεινόμενη μέθοδο μας, την **Pipe-VND** (εξερεύνηση στην ίδια γειτονιά, ενώ έχουν γίνει επίσης βελτιώσεις). Η **pipe VND** αποτελείται από δύο πολύ γνωστές δομές γειτονιάς, την **relocate** (λύσεις που έχουν ληφθεί από τη μετεγκατάσταση του κόμβου σε περιήγηση) και **2-opt** (λύσεις που λαμβάνονται με διακοπή και διαφορετική επανασύνδεση δύο τόξων περιήγησης). Η κύρια διαφορά μεταξύ **qGVNS** και κλασσικού **GVNS** έχει αναφερθεί στη φάση διαφοροποίησης. Στην παρούσα προσέγγιση, οι διαταραχές έχουν επιτευχθεί υιοθετώντας αρχές χβαντικού υπολογισμού. Σε κάθε κλήση της διαταραχής, ένας χβαντικός καταχωρητής θα δημιουργήσει έναν αριθμό αναγκαίων **qubits** (π.χ. n). Στη συνέχεια, αυτά τα **qubits** θα παράγουν τα αντίστοιχα στοιχεία, ανάλογα με τη διάσταση του προβλήματος. Αυτά τα στοιχεία θα είναι ίσα ή μεγαλύτερα από τον αριθμό των κόμβων της διαδρομής. Στη συνέχεια, ο αλγόριθμος θα επιλέξει σειριακά όλα τα συστατικά που απαιτούνται στην περίπτωση του προβλήματος μας και θα τα βάλει σε $1 \times n$ διάνυσμα. Επιπλέον, κάθε μία από αυτές θα αντιστοιχεί σε κάθε κόμβο της τρέχουσας λύσης. Στη συνέχεια, τα στοιχεία θα χρησιμοποιηθούν ως σημαία σε κάθε κόμβο της τρέχουσας λύσης. Τα στοιχεία μπορούν να είναι $0 \leq C \leq 1$.

Λόγω της αντιστοίχησης μεταξύ των στοιχείων και των κόμβων της διαδρομής, η ταξινόμηση στον πρώτο διάνυσμα θα επηρεάσει τη σειρά του κόμβου στο διάνυσμα λύσης. Έτσι, η διατεταγμένη διαδρομή θα είναι η διαδρομή που θα συμβεί μετά την κίνηση της διαταραχής, οδηγώντας την προσπάθεια εξερεύνησης σε έναν άλλο χώρο αναζήτησης. Σε αυτό το σημείο, πρέπει να αναφερθεί ότι χρησιμοποιείται η ευρετική του πλησιέστερου γείτονα (**Nearest Neighbor**) προκειμένου να παραχθεί μια αρχική εφικτή λύση (ο πρώτος κόμβος ορίζεται ως αποθήκη). Από μια αλγοριθμική προοπτική, το **qGVNS** συνοψίζεται στον επόμενο ψευδοκώδικα.

```

1: Input: an initial solution
2: Output: an optimized solution
3: Initialization of the feasibility distance matrix
4: Begin
5:  $X < -$  Nearest Neighbor heuristic;
6:
7: Repeat (until optimal solution is found or time limit is met)
8:
9:  $X' < -$  Quantum-Perturbation( $X$ )
10:
11:  $X'' < -$  pipeVND( $X'$ )
12:
13: if  $X''$  is better than  $X'$  then
14:    $X < -X''$ 
15:
```

2.3 Η μη συμβατική (χβαντική) μεθευρετική διαδικασία QUBO

Μια σημαντική κατηγορία προβλημάτων βελτιστοποίησης, ιδιαίτερα δεκτική στην χβαντική ανόπτηση του D-Wave, είναι αυτή στην οποία τα προβλήματα μπορούν να εκφραστούν ως τμήματα μη περιορισμένης δυαδικής βελτιστοποίησης (QUBO). Το QUBO αναφέρεται σε μια τεχνική αντιστοίχισης μοτίβων που, μεταξύ άλλων εφαρμογών, μπορεί να χρησιμοποιηθεί στη μηχανική μάθηση και τη βελτιστοποίηση, και η οποία περιλαμβάνει την ελαχιστοποίηση μίας τετραβάθμιας πολυωνυμικής εξίσωσης πάνω από δυαδικές μεταβλητές. Μερικά από τα πιο διάσημα προβλήματα συνδυαστικής βελτιστοποίησης που μπορούν να λυθούν ως QUBO προβλήματα είναι η μέγιστη αποκοπή, ο χρωματισμός του γραφήματος και το πρόβλημα partition. Περισσότερες λεπτομέρειες σχετικά με τη διατύπωση του QUBO και τα συναφή αποτελέσματα (που είναι πέραν του πεδίου εφαρμογής του παρόντος εγγράφου) μπορούν να βρεθούν στο έγγραφο έρευνας του Κοχενμπεργκερ (Kochenberger) [74].

Το QUBO είναι ισοδύναμο με το μοντέλο Ising, ένα γνωστό και εκτενώς μελετημένο μοντέλο στη φυσική, το οποίο εισήχθη στα μέσα της δεκαετίας του 1920 από τον Ερνστ Άισινγκ και τον Βίλχελμ Λεντς (Ernst Ising και Wilhelm Lenz) [139] στον τομέα του σιδηρομαγνητισμού. Η υποκείμενη λογική αρχιτεκτονική αυτού του μοντέλου είναι ότι οι μεταβλητές αναπαριστώνται ως qubits και οι αλληλεπιδράσεις μεταξύ των qubits ορίζονται ως το κόστος που σχετίζεται με κάθε ζεύγος qubits. Συγκεκριμένα, αυτή η αρχιτεκτονική μπορεί να απεικονιστεί ως ένας μη κατευθυνόμενος γράφος με qubits ως κορυφές και ζεύγες ως ακμές μεταξύ τους. Το λογισμικό ανοιχτού κώδικα qbsolv που εισήγαγε η D-Wave στο 2017 στοχεύει στην αντιμετώπιση προβλημάτων QUBO υψηλότερης κλίμακας από τις προηγούμενες απόπειρες, αξιοποιώντας μια πιο σύνθετη δομή γράφων με υψηλότερη συνδεσιμότητα μεταξύ των Qpu, χωρίζοντας τις εισροές σε τμήματα, ώστε να λυθούν ανεξάρτητα. Αυτή η διαδικασία επαναλαμβάνεται χρησιμοποιώντας μια αναζήτηση που βασίζεται στην αναζήτηση Tabu μέχρι να βρεθεί καμία περαιτέρω βελτίωση.

Σύμφωνα με το έργο των Παπαλίτσα κ.α. (Papalitsas et al.) [152], παρόλο που πολλά προβλήματα συνδυαστικής βελτιστοποίησης έχουν εκφραστεί στη διατύπωση του QUBO (ή ισο-δύναμα, στη διαμόρφωση), το συγκεκριμένο πρόβλημα δεν ήταν ένα από αυτά. Προφανώς, ο λόγος είναι ότι το TSPTW επιβάλλει πολλούς πρόσθετους περιορισμούς, επειδή τα παράθυρα του χρόνου των πελατών πρέπει να ικανοποιούνται. Πρόκειται για περιορισμούς ανισότητας που είναι πολύ δύσκολο να αντιμετωπιστούν στο πλαίσιο του QUBO. Υπενθυμίζεται ότι οι έγκυρες εκφράσεις QUBO πρέπει να έχουν τη μορφή: $x^T Q x$, όπου x είναι μια στήλη διανυσμάτων των δυαδικών μεταβλητών αποφάσεων, x^T οι μεταβάσεις του και Q ένας τετραγωνικός πίνακας των σταθερών. Έτσι, από όσο γνωρίζουμε, αυτή είναι η πρώτη φορά που το TSPTW δημιουργείται σε μορφή QUBO.

Αυτό το βήμα είναι μια απαραίτητη προϋπόθεση για να μπορούμε να εκτελέσουμε προβλήματα TSPTW στην τρέχουσα γενιά υπολογιστών D-Wave. Μια μελλοντική κατεύθυνση σε αυτόν τον τομέα θα μπορούσε να είναι η χαρτογράφηση των δεικτών αναφοράς του TSPTW έτσι ώστε να ληφθούν πειραματικά αποτελέσματα.



3. Το Πρόβλημα του Πλανόδιου Πωλητή

Στο κεφάλαιο αυτό παρουσιάζεται ένα ιδιαίτερα σημαντικό και ευρέως μελετημένο πρόβλημα διακριτής βελτιστοποίησης, το οποίο είναι ευρέως γνωστό ως «πρόβλημα του πλανόδιου πωλητή». Το πρόβλημα του πλανόδιου πωλητή (Travelling Salesman Problem - TSP) είναι ίσως το πιο μελετημένο πρόβλημα διακριτής βελτιστοποίησης και ακριβέστερα, συνδυαστικής βελτιστοποίησης. Η δημοτικότητά του οφείλεται στην ευκολία με την οποία μπορεί κανείς να το διατυπώσει, στη δυσκολία επίλυσής του, αλλά και στο τεράστιο εύρος εφαρμογών που διαθέτει. Η ιδιαίτερη σημασία του TSP έγκειται στο γεγονός ότι αποτελεί ένα τυπικό πρόβλημα του είδους του, δηλαδή της συνδυαστικής βελτιστοποίησης.

Η επίλυσή του ορίζεται ως η εύρεση του ελάχιστου κόστους μετακίνησης ώστε ο πωλητής (το πρόσωπο ή η οντότητα η οποία ταξιδεύει κατά μήκος μιας συγκεκριμένης διαδρομής η οποία περιέχει πολλούς κόμβους) να μπορεί να ξεκινήσει από ένα σημείο προέλευσης και να επιστρέψει στην προέλευση αφού περάσει από όλους τους συγκεκριμένους κόμβους της διαδρομής μία φορά.

Ουσιαστικά, γίνεται προσπάθεια για την ελαχιστοποίηση της συνολικής απόστασης, οπότε το πρόβλημα ανήκει στον κλάδο της βελτιστοποίησης. Λόγω της συνδυαστικής φύσης του προβλήματος, δεν μπορεί να γίνει χρήση της μεθόδου του διαφορικού λογισμού, δηλαδή να τεθούν οι παράγωγοι στο μηδέν. Οι επιλογές σε αυτό το πρόβλημα δεν γίνονται από ένα συνεχές σύνολο, αλλά από το σύνολο όλων των πιθανών διαδρομών [114].

Το TSP είναι το πρώτο πρόβλημα που περιγράφηκε στο [63], ένα βιβλίο των Γκάρεϊ και Τζόνσον (Garey και Johnson) το 1979 που αποτελεί το πρώτο σύγγραμμα για τη θεωρία της *NP*-πληρότητας και της υπολογιστικής δυσεπιλυσιμότητας και το οποίο είχε μεγάλη επίδραση στην επιστήμη των υπολογιστών.

Το πρόβλημα του πλανόδιου πωλητή κάνει την ακόλουθη ερώτηση: "δεδομένης της λίστας των πόλεων και των αποστάσεων μεταξύ κάθε ζεύγους πόλεων, ποια είναι η συντομότερη δυνατή διαδρομή που επισκέπτεται κάθε πόλη και επιστρέφει στην πόλη προέλευσης;" Είναι ένα πρόβλημα *NP-hard* στο συνδυαστική βελτιστοποίηση, σημαντική στην έρευνα επιχειρήσεων και τη θεωρητική πληροφορική. Το πρόβλημα του αγοραστή και το πρόβλημα δρομολόγησης του οχήματος είναι και οι δύο γενικεύσεις του προβλήματος. Στη θεωρία της υπολογιστικής πολυπλοκότητας, η έκδοση απόφασης του προβλήματος (όπου, δεδομένης της διάρκειας L , είναι να αποφασιστεί

αν το γράφημα έχει οποιαδήποτε περιοδικότητα μικρότερη από L) ανήκει στην κατηγορία των προβλημάτων NP.

Έτσι, είναι πιθανό ότι η χειρότερη περίπτωση χρόνου λειτουργίας για οποιοδήποτε αλγόριθμο αυξάνεται με τον αριθμό των πόλεων. Το πρόβλημα διαμορφώθηκε για πρώτη φορά το 1930 και είναι ένα από τα πιο εντατικά μελετημένα προβλήματα στη βελτιστοποίηση. Χρησιμοποιείται ως σημείο αναφοράς για πολλές μεθόδους βελτιστοποίησης. Ακόμα και αν το πρόβλημα είναι υπολογιστικά δύσκολο, ένας μεγάλος αριθμός αλγορίθμων είναι γνωστοί, έτσι ώστε ορισμένες περιπτώσεις με δεκάδες χιλιάδες πόλεις να μπορούν να λυθούν πλήρως και ακόμη και προβλήματα με εκατομμύρια πόλεις μπορούν να προσεγγίζονται μέσα σε ένα μικρό κλάσμα του 1%.

Το πρόβλημα έχει πολλές εφαρμογές ακόμα και στην πιο αγνή σύνθεσή του, όπως ο προγραμματισμός και η κατασκευή μικροchips. Ελαφρώς τροποποιημένο, εμφανίζεται ως υπο-πρόβλημα σε πολλούς τομείς, όπως η αλληλουχία του DNA. Σε αυτές τις εφαρμογές, η έννοια πόλη αντιπροσωπεύει, για παράδειγμα, θραύσματα DNA, και η απόσταση αντιπροσωπεύει τον χρόνο μετακίνησης ή το κόστος, ή ένα μέτρο ομοιότητας μεταξύ θραυσμάτων του DNA. Το πρόβλημα εμφανίζεται επίσης στην αστρονομία, καθώς οι αστρονόμοι που παρατηρούν πολλές πηγές θα θέλουν να ελαχιστοποιήσουν το χρόνο που δαπανάται μετακινώντας το τηλεσκόπιο μεταξύ των πηγών. Σε πολλές εφαρμογές, ενδέχεται να επιβληθούν πρόσθετοι περιορισμοί, όπως περιορισμένοι πόροι ή χρονικά παράθυρα.

Το πρόβλημα μπορεί να διαμορφωθεί ως ένα μη κατευθυνόμενο σταθμισμένο γράφημα, έτσι ώστε οι πόλεις να είναι οι κορυφές του γραφήματος, οι διαδρομές να είναι οι άκρες του γραφήματος και η απόσταση του μονοπατιού είναι το βάρος του άκρου. Είναι ένα πρόβλημα ελαχιστοποίησης ξεκινώντας και τελειώνοντας σε μια καθορισμένη κορυφή αφού επισκεφτήκαμε κάθε άλλη κορυφή ακριβώς μία φορά. Συχνά, το μοντέλο είναι ένα πλήρες γράφημα (δηλαδή, κάθε ζεύγος κορυφών συνδέεται με ένα άκρο). Εάν δεν υπάρχει διαδρομή μεταξύ δύο πόλεων, η προσθήκη ενός αυθαίρετα μεγάλης άκρης θα ολοκληρώσει το γράφημα χωρίς να επηρεαστεί η βέλτιστη περιήγηση.

Στο συμμετρικό πρόβλημα, η απόσταση μεταξύ δύο πόλεων είναι η ίδια σε κάθε αντίθετη κατεύθυνση, σχηματίζοντας έναν μη κατευθυνόμενο γράφο. Αυτή η συμμετρία αποτελεί το ήμισυ του αριθμού των πιθανών λύσεων. Στο ασύμμετρο πρόβλημα, οι διαδρομές μπορεί να μην υπάρχουν και στις δύο κατευθύνσεις ή οι αποστάσεις μπορεί να είναι διαφορετικές, σχηματίζοντας έναν κατευθυνόμενο γράφο. Οι κυκλοφοριακές συγκρούσεις και οι αεροπορικοί ναύλοι για πόλεις με διαφορετικά τέλη αναχώρησης και άφιξης είναι παραδείγματα του πώς μπορεί να σπάσει αυτή η συμμετρία.

Συναφή προβλήματα:

- Μια ισοδύναμη διατύπωση από την άποψη της θεωρίας γράφων είναι: δεδομένου ενός πλήρους σταθμισμένου γράφου (όπου οι κορυφές θα αντιπροσωπεύουν τις πόλεις, οι άκρες θα αντιπροσωπεύουν τους δρόμους, και τα βάρη θα είναι το κόστος ή η απόσταση αυτού του δρόμου), να βρεθεί ένας χαμιλτονιανός κύκλος με το ελάχιστο βάρος. Η απαίτηση να επιστρέψει ο πωλητής στην πόλη εκκίνησης δεν αλλάζει την υπολογιστική πολυπλοκότητα του προβλήματος.
- Ένα άλλο σχετικό πρόβλημα είναι το πρόβλημα της κυκλοφοριακής συμφόρησης του πωλητή: να βρεθεί ένας χαμιλτονιανός κύκλος σε ένα σταθμισμένο γράφο με το ελάχιστο βάρος της ζυγότερης άκρης. Για παράδειγμα, αποφεύγοντας στενούς δρόμους με μεγάλα λεωφορεία. Το πρόβλημα είναι σημαντικής πρακτικής σημασίας, εκτός από τις εμφανείς ζώνες μεταφορών και εφοδιαστικής αλυσίδας. Ένα κλασικό παράδειγμα είναι στην παραγω-

γή τυπωμένου κυκλώματος: Προγραμματισμός μιας οδού για το τρυπάνι για να γίνουν τρύπες σε ένα PCB. Σε ρομποτική κατεργασία ή γεωτρήσεις, οι "πόλεις" είναι μέρη για το μηχάνημα.

- Το γενικευμένο πρόβλημα του πλανόδιου πωλητή, ασχολείται με τις "καταστάσεις" που έχουν (μία ή περισσότερες) "πόλεις" και ο πωλητής πρέπει να επισκεφθεί ακριβώς μία "πόλη" από κάθε "κατάσταση". Μια εφαρμογή συναντάται στην παραγγελία μιας λύσης στο πρόβλημα τεμαχισμού του αποθέματος, προκειμένου να ελαχιστοποιηθούν οι αλλαγές. Ένα άλλο αφορά τη γεώτρηση σε παραγωγή ημιαγωγών.
- Το πρόβλημα της διαδοχικής παραγγελίας ασχολείται με το πρόβλημα της επίσκεψης σε ένα σύνολο πόλεων όπου υπάρχουν σχέσεις προτεραιότητας μεταξύ των πόλεων.
- Μια κοινή ερώτηση για συνέντευξη στη Google είναι πώς θα δρομολογηθούν δεδομένα μεταξύ κόμβων επεξεργασίας δεδομένων. Οι διαδρομές ποικίλλουν ανάλογα με το χρόνο για τη μεταφορά των δεδομένων, αλλά οι κόμβοι διαφέρουν επίσης από την υπολογιστική ισχύ και αποθήκευσή τους, επιδεινώνοντας το πρόβλημα της αποστολής δεδομένων.

3.1 Ιστορικά στοιχεία

Η προέλευση του προβλήματος του πλανόδιου πωλητή είναι ασαφής. Το 1832 τυπώθηκε ένα βιβλίο στη Γερμανία [188], ένα εγχειρίδιο για τους πλανόδιους πωλητές το οποίο αναφέρει το πρόβλημα και περιλαμβάνει παράδειγμα με ξεναγήσεις μέσω Γερμανίας και Ελβετίας, αλλά δεν περιέχει μαθηματική μεταχείριση ή επεξεργασία του προβλήματος. Παρόλο που είναι αφιερωμένο, στο μεγαλύτερο μέρος του, σε άλλα θέματα, στο τελευταίο κεφάλαιο φτάνει στην ουσία του TSP προβλήματος: «Με μία κατάλληλη επιλογή και προγραμματισμό της διαδρομής, μπορεί κανείς συχνά να κερδίζει τόσο πολύ χρόνο που πρέπει να κάνουμε κάποιες προτάσεις... Η πιο σημαντική πτυχή είναι να καλύψουμε όσο το δυνατόν περισσότερες τοποθεσίες, δίχως να επισκεφτούμε μία τοποθεσία για δεύτερη φορά...» [114].

Το TSP διατυπώνεται με μαθηματικό τρόπο για πρώτη φορά τον 19ο αιώνα από τον Ιρλανδό μαθηματικό Χάμιλτον και τον Βρετανό μαθηματικό Τόμας Κέρκμαν (Sir William Rowan Hamilton και Thomas Kirkman). Το παιχνίδι του Χάμιλτον ήταν ένα ψυχαγωγικό παζλ βασισμένο στην εξεύρεση ενός χαμιλτωνιακού κύκλου. Ο Χάμιλτον (Sir W.R. Hamilton) χρησιμοποίησε για τη μαθηματική διατύπωση του TSP τη θεωρία των γράφων, εξέχοντα κλάδο των συνδυαστικών μαθηματικών. Με τον όρο «γράφος» εννοείται ένα πεπερασμένο σύνολο κορυφών, κάποια ζεύγη από τις οποίες ενώνονται με ακμές. Ένας κύκλος σε ένα γράφο είναι ένα σύνολο κορυφών του γράφου, τέτοιο ώστε να είναι πιθανό να μετακινούμαστε από κορυφή σε κορυφή, κατά μήκος των ακμών του γράφου, έτσι ώστε όλες οι κορυφές να διαπερνώνται ακριβώς μία φορά και να ολοκληρώνεται εκεί από όπου ξεκίνησε. Εάν ένας κύκλος περιέχει όλες τις κορυφές του γράφου λέγεται Χαμιλτονιανός [114].

Το πρόβλημα του πλανόδιου πωλητή για έναν γράφο με καθορισμένα μήκη ακμών είναι το πρόβλημα της εύρεσης ενός Χαμιλτονιανού κύκλου του μικρότερου μήκους. Αντιστρόφως, το πρόβλημα του να αποφανθούμε σχετικά με το εάν ένας γράφος έχει έναν Χαμιλτονιανό κύκλο, είναι μια ειδική περίπτωση του προβλήματος του πλανόδιου πωλητή: εάν εκχωρήσουμε σε όλες τις ακμές του γράφου μήκος μηδέν (0) και για κάθε ακμή που λείπει, δημιουργήσουμε μια ακμή μήκους ένα (1), τότε έχουμε έναν καινούριο γράφο, ο οποίος έχει Χαμιλτονιανούς κύκλους. Η

επίλυση αυτής της ειδικής περίπτωσης του TSP θα αποδώσει έναν Χαμιλτονιανό κύκλο για τον καινούριο γράφο, ο οποίος θα έχει είτε συνολικό μήκος μηδέν (0), οπότε θα αποτελεί Χαμιλτονιανό κύκλο για τον αρχικό γράφο, είτε θετικό μήκος, πράγμα το οποίο θα αποδεικνύει ότι δεν υπάρχει τέτοιος κύκλος για τον αρχικό γράφο [114].

Πριν από τον Χάμιλτον (Sir W.R. Hamilton), παρόμοια ερωτήματα είχαν απασχολήσει και τους Όιλερ (Euler) το 1759 και Βαντερμοντ (Vandermonde) το 1771, οι οποίοι έκαναν λόγο για το πρόβλημα της διαδρομής του ιππότη (Knight's tour). Η διαδρομή ενός ιππότη είναι ένας Χαμιλτονιανός κύκλος για τον γράφο, οι κορυφές του οποίου είναι τα 64 τετράγωνα μιας σκακιέρας, με δύο κορυφές παρακείμενες εάν και μόνο εάν ένας ιππότης μπορεί να μετακινηθεί με ένα βήμα από το ένα τετράγωνο στο άλλο [114].

Τον 20ο αιώνα, το TSP είναι ήδη γνωστό ανάμεσα στους μαθηματικούς και στατιστικολόγους της εποχής με διαφορετικά, όμως, ονόματα. Η γενική μορφή φαίνεται να έχει μελετηθεί για πρώτη φορά από μαθηματικούς κατά τη διάρκεια της δεκαετίας του 1930 στη Βιέννη και στο Χάρβαρντ, κυρίως από τον Καρλ Μένγκερ (Karl Menger), ο οποίος καθορίζει το πρόβλημα, θεωρώντας τον προφανή αλγόριθμο εξαντλητικής αναζήτησης (brute-force) και ο οποίος παρατήρησε τη μη βελτιστότητα (non-optimality) της ευρετικής μεθόδου του πλησιέστερου γείτονα (nearest neighbor heuristic). Η παραλλαγή του Καρλ Μένγκερ (Karl Menger) για το TSP ονομάζεται «πρόβλημα του αγγελιοφόρου». Στην αγγλική βιβλιογραφία το πρόβλημα αυτό συναντάται ως “messenger problem”, ενώ στη γερμανική ως “Botenproblem”.

Αρχικά τα μαθηματικά του προβλήματος φαίνεται να παρουσιάστηκαν στη δεκαετία του 1930 από τον Μέριλ Φλουυντ (Meryl Floyd) ο οποίος έψαχνε να λύσει το πρόβλημα της δρομολόγησης του σχολικού λεωφορείου. Όσον αφορά την πατρότητα του όρου «πρόβλημα του πλανόδιου πωλητή», αυτή, παρόλο που δεν είναι επιστημονικά αποδεδειγμένο κάτι τέτοιο, θεωρείται ότι ανήκει στον Χάσλερ Γουίτνεϊ (Hassler Whitney) του Πανεπιστημίου του Πρίνστον, ο οποίος απέδωσε στο TSP πρόβλημα την ονομασία του γύρω στα 1931-1932. Σύμφωνα με τους Λόουλερ κ.α. (Lawler et al.) το 1985, στη δεκαετία του 1950 και του 1960, το TSP πρόβλημα έγινε όλο και πιο δημοφιλές στους επιστημονικούς κύκλους της Ευρώπης και των ΗΠΑ [114].

Κατά τη δεκαετία του 1950 και του 1960, το πρόβλημα έγινε όλο και πιο δημοφιλές στους επιστημονικούς κύκλους στην Ευρώπη και τις ΗΠΑ, αφού η RAND Corporation στη Σάντα Μόνικα προσέφερε βραβεία για την επίλυση του προβλήματος. Αξιοσημείωτες συνεισφορές έγιναν από τον Τζωρτζ Ντάζιγκ, τον Ντέλμπερτ Ρέι Φούλκερσον και τον Σέλμερ μ. Τζόνσον (George Dantzig, Delbert Ray Fulkerson και Selmer M. Johnson) από την RAND Corporation, οι οποίοι εξέφρασαν το πρόβλημα ως ακέραιο γραμμικό πρόγραμμα και ανέπτυξαν τη μέθοδο κοπής (cutting plane) για τη λύση του. Έγραψαν ένα έγγραφο για το θέμα όπου με αυτές τις νέες μεθόδους έλυσε ένα παράδειγμα με 49 πόλεις, δημιουργώντας μια διαδρομή και αποδεικνύοντας ότι καμία άλλη διαδρομή δεν θα μπορούσε να είναι συντομότερη.

Ο Ντάζιγκ, ο Φούλκερσον και ο Τζόνσον, ωστόσο, πιθανολογούν ότι δεδομένης μιας σχεδόν βέλτιστης λύσης, είναι πιθανό να βρεθεί η βέλτιστη ή να αποδειχθεί η καλύτερη βελτιστοποίηση προσθέτοντας ένα μικρό αριθμό επιπλέον ανισοτήτων (περικοπές). Χρησιμοποίησαν αυτή την ιδέα για να λύσουν το αρχικό πρόβλημα των 49 πόλεων χρησιμοποιώντας ένα μοντέλο συμβολοσειράς. Βρήκαν ότι χρειαζόνταν μόνο 26 περικοπές για να έρθουν σε μια λύση για το πρόβλημα των 49 πόλεων. Ενώ αυτό το έγγραφο δεν έδινε μια αλγοριθμική προσέγγιση στα προβλήματα, οι ιδέες που βρίσκονταν μέσα του ήταν απαραίτητες για να δημιουργηθούν αργότερα ακριβείς μέθοδοι λύσης, αν και χρειάστηκαν 15 χρόνια για να βρεθεί μια αλγοριθμική προσέγγιση στη δημιουργία αυτών των περικοπών.

Τις επόμενες δεκαετίες, το πρόβλημα μελετήθηκε από πολλούς ερευνητές από τους κλά-

δους των μαθηματικών, της επιστήμης των υπολογιστών, της χημείας, της φυσικής και άλλων επιστημών. Στη δεκαετία του 1960, ωστόσο, δημιουργήθηκε μια νέα προσέγγιση, η οποία αντί να αναζητά βέλτιστες λύσεις, παρήγαγε μια λύση της οποίας το μήκος οριοθετούνταν από ένα πολλαπλάσιο του βέλτιστου μήκους και έτσι δημιουργούνταν χαμηλότερα όρια για το πρόβλημα. Αυτά μπορούν στη συνέχεια να χρησιμοποιηθούν με υποκαταστήματα και δεσμευμένες προσεγγίσεις. Μια μέθοδος για να γίνει αυτό ήταν να δημιουργηθεί ένα ελάχιστο δέντρο που εκτείνεται στο γράφημα και στη συνέχεια διπλασιάζει όλες τις άκρες του. Αυτή η μέθοδος παράγει το δεσμευμένο πλαίσιο ότι το μήκος της βέλτιστης διαδρομής είναι το διπλάσιο του βάρους ενός ελάχιστου μήκους δέντρου.

Ο Χριστοφίδης έκανε μια μεγάλη προσφορά σε αυτή την προσέγγιση, δίνοντας μια προσέγγιση για την οποία γνωρίζουμε το χειρότερο σενάριο. Ο αλγόριθμος Χριστοφίδη που δόθηκε το 1976, στη χειρότερη περίπτωση είναι 1,5 φορές καλύτερος από τη βέλτιστη λύση. Καθώς ο αλγόριθμος ήταν τόσο απλός και γρήγορος, πολλοί ήλπιζαν ότι θα έδινε τη λύση σε μια σχεδόν βέλτιστη μέθοδο λύσης. Αυτή παραμένει η μέθοδος με το καλύτερο και το χειρότερο σενάριο.

Το 1972, ο Ρίτσαρντ Μ. Καρπ (Richard M. Karp) έδειξε ότι το πρόβλημα του Χαμιλτονιανού κύκλου ήταν NP-complete, πράγμα που συνεπάγεται την NP-δυσεπιλυσιμότητα του TSP. Το γεγονός αυτό παρείχε μία μαθηματική εξήγηση για την προφανή υπολογιστική δυσκολία εύρεσης βέλτιστων διαδρομών. Στα τέλη της δεκαετίας του 1970 και του 1980, επιτεύχθηκε μεγάλη πρόοδος, όταν οι Γκρετσελ, Παντμπεργκ, Ρινάλντι κ.α. (Grötschel, Padberg, Rinaldi et al.) κατάφεραν να επιλύσουν με ακρίβεια περιπτώσεις με έως και 2392 πόλεις, χρησιμοποιώντας μεθόδους cutting-planes και branch-and-bound (διακλάδωσης και οριοθέτησης), ενώ στη δεκαετία του 1990, οι Άπλγκεϊτ, Μπίξμπι, Τσβάταλ και Κουκ (Applegate, Bixby, Chvátal και Cook) ανέπτυξαν το πρόγραμμα Κονκόρντ (Concorde), το οποίο χρησιμοποιήθηκε σε πολλές πρόσφατες λύσεις ρεκόρ.

Το 1991, ο Γκέρχαρντ Ράιντσντ (Gerhard Reinelt) δημοσίευσε το TSPLIB, μία συλλογή προβλημάτων αναφοράς μεταβλητής δυσκολίας, η οποία έχει χρησιμοποιηθεί από πολλές ερευνητικές ομάδες για τη σύγκριση των αποτελεσμάτων.

Πολλοί ερευνητές έχουν μελετήσει το πρόβλημα του κλασικού πλανόδιου πωλητή με τα χρονικά παράθυρα (TSPTW). Η σχετική βιβλιογραφία παρέχει ακριβείς αλγόριθμους για την επίλυση του TSPTW. Οι Λανγκεβίν κ.α. (Langevin et al.) [112] εισήγαγαν μια σύνθεση ροής δύο στοιχείων, τα οποία μπορούν να επεκταθούν στο κλασικό πρόβλημα. Οι Ντούμας κ.α. (Dumas et al.) [49] χρησιμοποίησαν μια δυναμική προσέγγιση προγραμματισμού μειώνοντας τις δοκιμές, η οποία βελτίωσε την απόδοση και κλιμακώνεται στο χώρο αναζήτησης, εκ των προτέρων και κατά τη διάρκεια της εκτέλεσής του, καθώς και πριν. Δεδομένου ότι το TSPTW είναι ένα πρόβλημα NP-hard, στην πράξη, είναι απαραίτητο να χρησιμοποιηθεί ένα μεθευρετικό μοντέλο, ικανό να λύσει αποτελεσματικά ρεαλιστικές περιπτώσεις μέσα σε εύλογο χρονικό διάστημα.

Οι Γκεντρώ κ.α. (Gendreau et al.) [65] πρότειναν μια γενικευμένη μεθευρετική, η οποία σταδιακά χτίζει το μονοπάτι για τη φάση της κατασκευής και βελτιώνει τη φάση βελτίωσης. Οι Ουρουτία κ.α. (Urrutia et al.) [38] πρότειναν ένα μεθευρετικό μοντέλο σε δύο στάδια, βασισμένο στο VNS. Στο πρώτο βήμα, μια εφικτή λύση κατασκευάζεται με VNS, όπου η μικτή γραμμική, ακεραία, αντισυμβατική συνάρτηση αντιπροσωπεύεται ως μια μέτρηση εφικτότητας. Στο δεύτερο στάδιο, το μεθευρετικό βελτιώνει την εφικτή λύση χρησιμοποιώντας τη γενική έκδοση του VNS (General VNS-GVNS).

Το 2006, ο Κουκ (Cook) και άλλοι υπολόγισαν μία βέλτιστη διαδρομή σε ένα στιγμιότυπο 85900 πόλεων, δοσμένο από ένα πρόβλημα διάταξης μικροσίπ, το οποίο αποτελεί επί του παρόντος το μεγαλύτερο παράδειγμα TSPLIB που έχει επιλυθεί. Για πολλές άλλες περιπτώσεις με

εκατομμύρια πόλεις, μπορούν να βρεθούν λύσεις, οι οποίες εγγυημένα να είναι μέσα στο 2-3 % της βέλτιστης διαδρομής [164].

Οι Παπαλίτσας κ.α.[149] ανέπτυξαν μια μεθευρετική βάση, βάσει των συμβατικών αρχών για την εξεύρεση εφικτών λύσεων για το TSPTW εντός σύντομου χρονικού διαστήματος. Στη συνέχεια, προτάθηκε μια νέα μη συμβατική μεθευρετική μέθοδος με χβαντική έμπνευση, με βάση την αρχική γενική μεταβλητή περιοχή αναζήτησης (GVNS), για την επίλυση του τυπικού προβλήματος. Αυτό το μεθευρετικό μοντέλο, εμπνευσμένο από την χβαντική πηγή εφαρμόστηκε στη λύση του προβλήματος συλλογής σκουπιδιών που μοντελοποίησε ως παρουσία του προβλήματος. Πρόσφατα, οι Παπαλίτσας κ.α. (Papalitsas et al.) [151] εφάρμοσαν αυτό το μεθευρετικό μοντέλο εμπνευσμένο από τον χβαντικό κόσμο στο πρακτικό πρόβλημα της συλλογής σκουπιδιών με τα χρονικά παράθυρα. Για την πλειονότητα των περιπτώσεων αναφοράς που χρησιμοποιήθηκαν για την αξιολόγηση των προτεινόμενων μεθευρετικών μοντέλων, τα πειραματικά αποτελέσματα ήταν ιδιαίτερα υποσχόμενα.

3.2 Μαθηματική Περιγραφή του TSP

Το πρόβλημα αυτό είναι ένα NP-hard πρόβλημα συνδυαστικής βελτιστοποίησης, με μεγάλη σημασία για τους κλάδους της επιχειρησιακής έρευνας και της θεωρητικής επιστήμης των υπολογιστών. Το πρόβλημα του πλανόδιου πωλητή θέτει την εξής ερώτηση: «Με δεδομένο ένα σύνολο πόλεων και το κόστος της περιήγησης (ή την απόσταση της περιήγησης) μεταξύ κάθε πιθανού ζεύγους αυτών των πόλεων, ποια είναι η συντομότερη δυνατή διαδρομή που επισκέπτεται κάθε πόλη ακριβώς μία φορά και επιστρέφει στην πόλη προέλευσης;». Με άλλα λόγια, το TSP αναφέρεται στην εύρεση του καλύτερου δυνατού τρόπου, με τον οποίο μπορούμε να επισκεφτούμε όλες τις πόλεις και να επιστρέψουμε στο σημείο εκκίνησης, ελαχιστοποιώντας το κόστος (ή την απόσταση) της περιήγησης.

Ο όρος "το πρόβλημα του πλανόδιου πωλητή" πρωτοεμφανίστηκε μεταξύ 1931 και 1932 και το πρόβλημα φαίνεται να περιγράφηκε πρώτη φορά αφαιρετικά σε ένα γερμανικό βιβλίο του 1832 [188].

Η μαθηματική περιγραφή του TSP αποδόθηκε από τους Χάμιλτον και Κίρκμαν (Hamilton και Kirkman) [188]. Δεδομένου ενός γράφου, ένας κύκλος είναι ένα κλειστό μονοπάτι το οποίο ξεκινά και τελειώνει στον ίδιο κόμβο, αφού έχει περάσει από κάθε άλλο κόμβο μια φορά. Ένας χαμιλτονιανός κύκλος είναι ο κύκλος ο οποίος ικανοποιεί τον περιορισμό ο οποίος περιλαμβάνει όλες τις πλευρές του γράφου. Ουσιαστικά, το πρόβλημα του πλανόδιου πωλητή έχει ως σκοπό να υπολογίζει τον πιο οικονομικό τρόπο να γίνει η διαδρομή μέσα από κάθε πόλη και να επιστρέψει πάλι στο αρχικό σημείο.

Στον γράφο 3.1, ένας πωλητής πρέπει να επισκεφθεί όλες τις πόλεις με τα ονόματα A, B, C και D. Το πρόβλημα περιλαμβάνει και τη διαδικασία εύρεσης φθηνών αεροπορικών εισιτηρίων και τον λιγότερο χρόνο ταξιδιού.

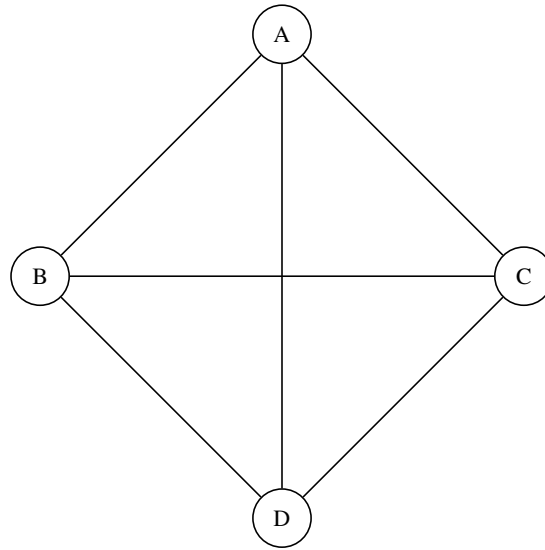


Figure 3.1: Παράδειγμα διαδρομής μεταξύ 4 κόμβων

Το TSP μπορεί να οριστεί σαν ένα πλήρες μη κατευθυνόμενο γράφημα, εάν είναι συμμετρικό ή σαν ένα κατευθυνόμενο γράφημα, εάν είναι ασύμμετρο. Ο μη κατευθυνόμενος γράφος συμβολίζεται ως $G = (V, E)$, ενώ ο κατευθυνόμενος ως $G = (V, A)$. Έστω ότι το n συμβολίζει τον αριθμό των πόλεων-κόμβων, το σύνολο $V = \{1, 2, 3, \dots, n\}$ είναι το σύνολο των πλευρών, το $E = \{(i, j) : i, j \in V, \text{ where } i < j\}$ είναι το σύνολο των κορυφών και $A = \{(i, j) : i, j \in V, \text{ where } i \neq j\}$ είναι το σύνολο των ακμών. Κάθε ακμή σχετίζεται με ένα βάρος c_{ij} , το οποίο αναπαριστά το κόστος (ή την απόσταση) για την μετακίνηση από τον ένα κόμβο i στον κόμβο j . Αν το c_{ij} ισούται με το c_{ji} , το TSP είναι συμμετρικό (sTSP), διαφορετικά είναι ασύμμετρο (aTSP).

Το TSP θεωρείται NP-complete πρόβλημα και δεν υπάρχει γνωστός πολυωνυμικός αλγόριθμος που να μπορεί να βρει τη βέλτιστη λύση ανεξάρτητα από το μέγεθος του εκάστοτε προβλήματος [163].

Ένας πίνακας κόστους $C = [c_{i,j}]$ ορίζεται στο E ή στο A. Ο πίνακας αυτός ικανοποιεί την τριγωνική ανισότητα $c_{i,j} \leq c_{i,k} + c_{k,j}$ κάθε φορά που i, j, k ορίζονται για κάθε πλευρά ή ακμή. Ειδικότερα, αυτή είναι περίπτωση για τα προβλήματα για τα οποία οι κορυφές είναι σημεία $P_i = (X_i, Y_i)$ στο ευκλείδειο επίπεδο και $c_{i,j} = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2}$ είναι η Ευκλείδεια απόσταση. Η τριγωνική ανισότητα επίσης ικανοποιείται εάν το $c_{i,j}$ είναι το μήκος του μικρότερου μονοπατιού από το i στο j στον γράφο [122].

3.3 Υπολογιστική Πολυπλοκότητα

Όσον αφορά την υπολογιστική πολυπλοκότητα του TSP, εάν είναι ο αριθμός των πόλεων που πρόκειται να επισκεφθούμε, ο συνολικός αριθμός των πιθανών διαδρομών που καλύπτουν όλες τις πόλεις μπορεί να εκφραστεί ως ένα σύνολο των εφικτών λύσεων του TSP και ισούται με [91].

Στα τέλη της δεκαετίας του 1960, ήταν καλά αντιληπτό ότι υπήρχε μία σημαντική διαφορά μεταξύ δυσεπίλυτων προβλημάτων, όπως το TSP και εύκολα υπολογίσιμων προβλημάτων, όπως τα προβλήματα εκχώρησης (assignment) και μεταφοράς (transportation). Η διαφορά αυτή έγκειται στο γεγονός ότι, για την πρώτη κατηγορία προβλημάτων οι μόνοι διαθέσιμοι αλγόριθμοι βελτιστοποίησης ήταν απαριθμητικής (enumerative) φύσεως, ενώ για τη δεύτερη κατηγορία προβλημά-

των, υπήρχαν καλοί αλγόριθμοι, με τον όρο «καλός» να έχει δημιουργηθεί από τον Τζ. Ρ. Εντμοντς (J.R. Edmonds) για να περιγράψει μεθόδους επίλυσης, των οποίων ο χρόνος εκτέλεσης αυξάνει το πολύ πολυωνυμικά σε σχέση με το μέγεθος του προβλήματος.

Υπήρχαν εμπειρικά στοιχεία ότι όλες οι απαριθμητικές μέθοδοι που είχαν αναπτυχθεί για το TSP και άλλα δυσεπίλυτα προβλήματα συμπεριφερόντουσαν πολύ χειρότερα και φαινόταν λογικό να εικάζεται ότι προβλήματα όπως το TSP είχαν τέτοια εγγενή πολυπλοκότητα που η απαιτούμενη υπολογιστική προσπάθεια από οποιαδήποτε μέθοδο για την επίλυσή τους θα αυξάνονταν υπερπολυωνυμικά σε σχέση με το μέγεθος του προβλήματος [114]. Η εικασία αυτή είναι ακόμη ανεπιβεβαίωτη, αλλά οι εξελίξεις στη θεωρητική επιστήμη των υπολογιστών γύρω στο 1970 παρέχουν ένα πολύ ισχυρότερο θεμέλιο για τη διατύπωσή της, καθώς και μια πολύ καλύτερη κατανόηση του τρόπου με τον οποίο τα δυσεπίλυτα προβλήματα σχετίζονται μεταξύ τους.

Από τρία θεμελιώδη άρθρα [35] [103] [116] προέκυψε ότι τα προβλήματα για τα οποία υπήρχε υποψία ότι ήταν εγγενώς δύσκολα ήταν όλα υπολογιστικά ισοδύναμα μεταξύ τους, με την έννοια ότι ένας πολυωνυμικός αλγόριθμος για ένα από αυτά θα μπορούσε να χρησιμοποιηθεί για να επιλύσει όλα τα υπόλοιπα επίσης σε πολυωνυμικό χρόνο. Σε γενικές γραμμές, ο τρόπος για να καθιερώσουμε αυτά τα αποτελέσματα είναι να δείξουμε αρχικά ότι όλα αυτά τα προβλήματα επιτρέπουν (πολυωνυμικά οριοθετημένες) διατυπώσεις ως προβλήματα ακέραιου προγραμματισμού κι έπειτα να αποδείξουμε ότι ο ακέραιος προγραμματισμός είναι με τη σειρά του μια ειδική περίπτωση συγκεκριμένων, μεμονωμένων, δυσεπίλυτων προβλημάτων (συμπεριλαμβανομένου του TSP). Αυτά τα προβλήματα ονομάζονται NP προβλήματα. Οι έννοιες της υπολογιστικής πολυπλοκότητας παρήγαγαν πολλή έρευνα κατά τη διάρκεια της δεκαετίας του 1970 και η δραστηριότητα αυτή καταγράφηκε στην εντυπωσιακή έρευνα των Γκάρρεϊ και Τζονσον (Garey και Johnson) το 1979 [63].

Παρόλο που η υπολογιστική δυσκολία του TSP δε συνεπάγεται ότι εκθετικοί χρόνοι εκτέλεσης της χειρότερης περίπτωσης για την επίλυσή του είναι αναπόφευκτοι, εξυπηρετεί να ενισχύσει την πεποίθηση ότι η ύπαρξη πολυωνυμικού αλγορίθμου για το TSP είναι εξαιρετικά απίθανη. Ωστόσο, μέχρι σήμερα δεν έχει επιτευχθεί κάποια απόδειξη της πεποίθησης αυτής, με συνέπεια η εικασία να κερδίζει σε φήμη και σεβασμό χρόνο με το χρόνο. Η πιθανή δυσεπιλυσιμότητα της γενικής εκδοχής του TSP δεν αποκλείει την πιθανότητα να υπάρχουν πολυωνυμικές διαδικασίες επίλυσης για κάποιες ειδικές περιπτώσεις του TSP. Πράγματι, μια από τις πιο εκπληκτικές πτυχές της θεωρίας υπολογιστικής πολυπλοκότητας είναι ότι υπάρχει μια απότομη οριογραμμή που διακρίνει τα εύκολα από τα δύσκολα προβλήματα [63].

Στη μια πλευρά της οριογραμμής βρίσκει κανείς ειδικές περιπτώσεις δύσκολων προβλημάτων, για τα οποία είναι δυνατό να αποδειχθεί ότι είναι NP-hard, όπως το Ευκλείδειο TSP, όπου οι πόλεις δίνονται ως σημεία σε μια δισδιάστατη χαμπύλη (επίπεδη επιφάνεια δίχως πάχος) και οι αποστάσεις τους υπολογίζονται σύμφωνα με την Ευκλείδεια μετρική (συνάρτηση η οποία μετράει τη "συνήθη" (Ευκλείδεια) απόσταση μεταξύ δύο σημείων στον επίπεδο, δισδιάστατο χώρο κάνοντας επανειλημμένη χρήση του Πυθαγόρειου θεωρήματος) [63].

Στην άλλη πλευρά της οριογραμμής βρίσκει κανείς προβλήματα που μοιάζουν αρκετά παρόμοια με κάποιο από τα δυσεπίλυτα προβλήματα, αλλά ωστόσο αποδεικνύονται να είναι επιλύσιμα σε πολυωνυμικό χρόνο, όπως παραδείγματος χάριν, το TSP με έναν άνω τριγωνικό πίνακα απόστασης. Το πρώτο ειδικό TSP η δομή του οποίου οδήγησε σε έναν πολυωνυμικό αλγόριθμο ήταν ένα πρόβλημα ακολουθίας εργασιών (job scheduling) [69]. Τα επόμενα χρόνια, αξιοσημείωτη ήταν η ερευνητική δραστηριότητα στη Σοβιετική Ένωση, η οποία εστίασε στην ταυτοποίηση άλλων εύκολων περιπτώσεων του TSP [63].

3.4 Κατηγοριοποίηση του TSP

Σε γενικές γραμμές, το TSP ταξινομείται ως συμμετρικό (symmetric TSP - sTSP), ως ασύμμετρο (asymmetric TSP - aTSP) και ως πολλαπλό (multi TSP - mTSP) [122]. Η ενότητα αυτή περιγράφει αυτά τα τρία ευρέως μελετημένα TSP:

- **sTSP:** Έστω $V = v_1, v_2 \dots v_n$ είναι ένα σύνολο πόλεων, $A = (r, s) : r, s \in V$ είναι το σύνολο ακμών και $d_{rs} = d_{sr}$ είναι ένα μέτρο κόστους που σχετίζεται με την ακμή $(r, s) \in A$. Το sTSP είναι το πρόβλημα της εύρεσης μιας κλειστής διαδρομής ελάχιστου μήκους, η οποία να επισκέπτεται κάθε πόλη μία ακριβώς φορά. Στην περίπτωση που οι πόλεις $v_i \in V$ δίνονται από τις συντεταγμένες τους (x_i, y_i) και d_{rs} είναι η Ευκλείδεια απόσταση μεταξύ r και s , προκύπτει το λεγόμενο «Ευκλείδειο TSP» (Euclidean TSP).
- **aTSP:** Εάν $d_{rs} \neq d_{sr}$ για τουλάχιστον μία ακμή (r, s) , τότε το TSP γίνεται ασύμμετρο (asymmetric TSP).
- **mTSP:** Το πολλαπλό TSP ορίζει ότι σε ένα δεδομένο σύνολο κόμβων, έστω ότι υπάρχουν πωλητές, οι οποίοι βρίσκονται σε έναν μόνο κόμβο - αποθήκη. Οι υπόλοιποι κόμβοι (πόλεις) που πρόκειται να δεχθούν επίσκεψη είναι ενδιαμέσοι κόμβοι. Το πολλαπλό TSP συνίσταται στην εξεύρεση διαδρομών για όλους τους πωλητές, όλες εκ των οποίων ξεκινούν από τον κόμβο-αποθήκη και καταλήγουν σε αυτόν, έτσι ώστε κάθε ενδιαμέσος κόμβος να δέχεται επίσκεψη ακριβώς μία φορά και το συνολικό κόστος της επίσκεψης σε όλους τους κόμβους να ελαχιστοποιείται. Η μετρική του κόστους μπορεί να ορισθεί σε σχέση με την απόσταση, το χρόνο και άλλα μεγέθη [122]. Κάποιες πιθανές παραλλαγές του mTSP προβλήματος αποτελούν τα εξής:
 - Ένας ή πολλοί κόμβοι-αποθήκες: Στην περίπτωση του ενός κόμβου, όλοι οι πωλητές ολοκληρώνουν τις διαδρομές τους σε ένα και μόνο σημείο, ενώ στην περίπτωση των πολλών κόμβων οι πωλητές μπορούν είτε να επιστρέψουν στον αρχικό τους κόμβο-αποθήκη, είτε να επιστρέψουν σε οποιονδήποτε άλλον κόμβο, αρκεί ο αρχικός αριθμός των πωλητών σε κάθε κόμβο-αποθήκη να παραμένει αμετάβλητος αφού ολοκληρωθεί η διαδικασία της περιήγησης.
 - Αριθμός πωλητών: Ο αριθμός των πωλητών στο πρόβλημα μπορεί να είναι, είτε μία σταθερά, είτε μία φραγμένη μεταβλητή.
 - Κόστος: Όταν ο αριθμός των πωλητών δεν είναι σταθερός, τότε ο κάθε πωλητής έχει συνήθως ένα σταθερό κόστος που σχετίζεται με αυτόν τον συγκεκριμένο πωλητή και το οποίο προκύπτει οποτεδήποτε χρησιμοποιείται ο πωλητής αυτός.
 - Χρονικό πλαίσιο: Στην παραλλαγή αυτή, κάποιοι κόμβοι πρέπει να δεχθούν επίσκεψη σε συγκεκριμένες χρονικές περιόδους που ονομάζονται παράθυρα χρόνου, αποτελώντας έτσι ουσιαστικά επέκταση του πολλαπλού TSP (mTSP). Αυτή η εκδοχή του TSP αναφέρεται ως πολλαπλό πρόβλημα του πλανόδιου πωλητή με καθορισμένο χρονικό πλαίσιο (multiple TSP with time windows - mTSPTW). Μία αρκετά ενδεικτική εφαρμογή του mTSPTW είναι τα προβλήματα προγραμματισμού αεροσκαφών (aircraft scheduling problems).
 - Άλλοι περιορισμοί: Περιορισμοί μπορεί να υπάρξουν, επίσης, σχετικά με τον αριθμό των κόμβων τους οποίους μπορεί να επισκεφθεί κάθε πωλητής, τη μέγιστη ή την ελάχιστη απόσταση την οποία επιτρέπεται να διανύσει ένας πωλητής ή οποιοδήποτε άλλοι περιορισμοί [122].

3.4.1 Εφαρμογές του TSP και συνδέσεις με άλλα προβλήματα

Τόσο η γενική εκδοχή του προβλήματος του πλανόδιου πωλητή, όσο και η παραλλαγή του πολλαπλού TSP έχουν πλήθος εφαρμογών και μεγάλη σχέση με πολλά γνωστά προβλήματα. Η υποενότητα αυτή παρουσιάζει και περιγράφει τις εφαρμογές του TSP, του mTSP, καθώς και τη σχέση τους με άλλα προβλήματα.

Παρά το γεγονός ότι το μοντέλο του πλανόδιου πωλητή ισχύει άμεσα σε μια κατάσταση που μοιάζει να είναι πολύ χρήσιμη και συγκεκριμένα σε αυτή ενός πωλητή, ο οποίος επιθυμεί να ελαχιστοποιήσει την απόσταση του ταξιδιού του, οι περισσότερες από τις εφαρμογές που θα αναφερθούν είναι αρκετά διαφορετικές [63].

Μία άμεση εφαρμογή του TSP είναι το πρόβλημα διάτρησης (drilling) των τυπωμένων πινάκων κυκλώματος (printed circuit boards - PCBs) των Γκρέτσελ κ.α. (Grötschel et al., 1991). Για τη σύνδεση ενός αγωγού που βρίσκεται σε ένα επίπεδο με έναν αγωγό που βρίσκεται σε ένα άλλο επίπεδο, ή για να τοποθετήσουμε τις ακίδες των ολοκληρωμένων κυκλωμάτων, θα πρέπει να δημιουργήσουμε τρύπες στην πλακέτα. Οι οπές μπορεί να έχουν διαφορετικά μεγέθη μεταξύ τους. Εάν θέλουμε να διανοίξουμε 2 οπές διαφορετικής διαμέτρου διαδοχικά, η κεφαλή της μηχανής πρέπει να κινηθεί σε μια εργαλειοθήκη και να αλλάξει τον εξοπλισμό διάτρησης. Το γεγονός αυτό είναι αρκετά χρονοβόρο. Έτσι, καταλήγουμε στο συμπέρασμα ότι είναι προτιμότερο να επιλέξει κανείς κάποια αρχική διάμετρο, να ανοίξει όλες τις οπές που θα έχουν αυτήν τη διάμετρο και έπειτα να αλλάξει τον εξοπλισμό διάτρησης, να ανοίξει πάλι όλες τις οπές της επόμενης ίδιας διαμέτρου, κ.ο.κ. Έτσι, το πρόβλημα της διάτρησης της πλακέτας του κυκλώματος μπορεί να θεωρηθεί ως μία ακολουθία περιπτώσεων TSP, μία για κάθε διάμετρο οπής, όπου οι “πόλεις” είναι η αρχική θέση και το σύνολο όλων των οπών που μπορεί να διανοιχτούν με τον ίδιο εξοπλισμό διάτρησης. Η “απόσταση” μεταξύ 2 πόλεων δίνεται από το χρόνο που χρειάζεται για να κινηθεί η κεφαλή διάτρησης από τη μία θέση στην άλλη. Ο στόχος είναι να ελαχιστοποιηθεί η διάρκεια της περιήγησης για την κεφαλή της μηχανής [122].

Μια άλλη εφαρμογή του TSP είναι και η επιθεώρηση των αεριοστροβίλων (overhauling gas turbine engines), η οποία αναφέρθηκε για πρώτη φορά από τους Πλαντέ κ.α. το 1987 (Plante et al.) και συμβαίνει όταν οι αεροστρόβιλοι ενός αεροσκάφους πρέπει να ελεγχθούν. Προκειμένου να διασφαλισθεί ομοιόμορφη ροή αερίου διαμέσου των στροβίλων υπάρχουν τα λεγόμενα «στόμια-οδηγοί» συναρμολογούμενων πτερυγίων, τα οποία βρίσκονται σε κάθε στάδιο του στροβίλου. Μια τέτοια συναρμολόγηση αποτελείται, βασικά, από ένα αριθμό τέτοιων στομιών που είναι τοποθετημένα στην περιφέρειά της. Όλα αυτά τα πτερύγια έχουν ξεχωριστά χαρακτηριστικά και η σωστή τοποθέτησή τους μπορεί να οδηγήσει σε ουσιαστικά και σημαντικά οφέλη, όπως είναι η μείωση των κραδασμών, η αύξηση της ομοιομορφίας της ροής αερίου, η μείωση της κατανάλωσης καυσίμων, κ.ά. Το πρόβλημα της βέλτιστης δυνατής τοποθέτησης των πτερυγίων μπορεί να μοντελοποιηθεί ως ένα πρόβλημα TSP με μια ειδική αντικειμενική συνάρτηση [122].

Μία ακόμη σημαντική εφαρμογή του TSP είναι η χρυσταλλογραφία (crystallography) ακτίνων X (X-rays), η οποία αναλύει τη δομή των χρυστάλλων. Στην περίπτωση αυτή, χρησιμοποιείται ένα διαθλασίμετρο ακτίνων X για την απόκτηση πληροφοριών σχετικά με τη δομή του χρυσταλλικού υλικού. Για τον σκοπό αυτό, ένας ανιχνευτής μετρά την ένταση των αντανάκλασεων X-ray του χρυστάλλου από διάφορες θέσεις. Παρόλο που η διαδικασία της μέτρησης μπορεί να διεξαχθεί πολύ γρήγορα, υπάρχει ένας σημαντικός φόρτος στο χρόνο τοποθέτησης, αφού για μερικά πειράματα πρέπει να ληφθούν υπόψη μέχρι και εκατοντάδες χιλιάδες θέσεις. Το πρόβλημα δηλαδή, σε γενικές γραμμές, συνίσταται στην εύρεση μιας αλληλουχίας η οποία να ελαχιστοποιεί το συνολικό χρόνο τοποθέτησης. Αυτό οδηγεί σε ένα πρόβλημα πλανόδιου πωλητή [122].

Η καλωδίωση του υπολογιστή αποτελεί επίσης μία εφαρμογή του TSP. Οι Λένστρα και Καν (Lenstra και Kan) το 1974 ανέφεραν μία ειδική περίπτωση συνδεσμολογίας των συσκευών σε μια πλακέτα υπολογιστή. Οι μονάδες που βρίσκονται τοποθετημένες σε μια πλακέτα του υπολογιστή πρέπει να συνδεθούν με ένα δεδομένο υποσύνολο ακίδων. Σε αντίθεση με τη συνήθη περίπτωση όπου είναι επιθυμητή μια σύνδεση τύπου δένδρο Steiner, εδώ δεν πρέπει να συνδέονται περισσότερα από δύο καλώδια με κάθε ακίδα. Ως εκ τούτου, προκύπτει το πρόβλημα της εύρεσης ενός βραχύτατου Χαμιλτονιανού μονοπατιού με απροσδιόριστα σημεία εκκίνησης και τερματισμού.

Μια παρόμοια κατάσταση εμφανίζεται και στη λεγόμενη «καλωδίωση testbus». Για να ελέγξουμε την κατασκευασμένη πλακέτα, θα πρέπει να θεωρήσουμε μια σύνδεση η οποία εισέρχεται στην πλακέτα σε κάποιο συγκεκριμένο σημείο, τρέχει διαμέσου όλων των μονάδων και καταλήγει σε κάποιο συγκεκριμένο σημείο. Για κάθε μονάδα έχουμε, επίσης, ένα συγκεκριμένο σημείο εισόδου και αντίστοιχα, εξόδου, για αυτή τη δοκιμαστική καλωδίωση. Το πρόβλημα αυτό απαιτεί, επίσης, την επίλυση ενός προβλήματος εύρεσης ενός Χαμιλτονιανού μονοπατιού, με τη διαφορά ότι οι αποστάσεις δεν είναι συμμετρικές και ότι το σημείο εκκίνησης και τερματισμού είναι καθορισμένα [122].

3.4.2 Εφαρμογές του mTSP και διασύνδεσή του με άλλα προβλήματα

Η κύρια εφαρμογή του mTSP αφορά πραγματικά σενάρια, καθώς είναι σε θέση να χειριστεί πολλαπλούς πωλητές. Οι καταστάσεις αυτές προκύπτουν κυρίως σε διάφορα προβλήματα δρομολόγησης και χρονοπρογραμματισμού [122]. Στη συνέχεια παρουσιάζονται κάποιες ακόμη εφαρμογές που έχουν αναφερθεί στη βιβλιογραφία:

- Πρόβλημα προγραμματισμού ενός τυπογραφικού πιεστηρίου: Μια από τις σημαντικότερες και κύριες εφαρμογές του mTSP προκύπτει στον προγραμματισμό ενός τυπογραφικού πιεστηρίου για ένα περιοδικό με πολλαπλές εκδόσεις. Εδώ, υπάρχουν πέντε ζεύγη κυλίνδρων μεταξύ των οποίων εναλλάσσεται το χαρτί, ώστε και οι δύο πλευρές μιας σελίδας να εκτυπώνονται ταυτόχρονα. Υπάρχουν τρία είδη μορφών, οι οποίες χρησιμοποιούνται για την εκτύπωση των εκδόσεων, ήτοι μορφή 4-, 6-, ή 8-σελίδων. Το πρόβλημα του προγραμματισμού έγκειται στο να αποφασισθεί ποια μορφή θα χρησιμοποιείται σε κάθε εκτέλεση του πιεστηρίου και το μήκος του κάθε γύρου. Στο λεξιλόγιο του mTSP, τα κόστη αλλαγής της εκάστοτε μορφής που χρησιμοποιείται στο συγκεκριμένο γύρο αντιστοιχούν στα κόστη της διαδρομής μεταξύ των πόλεων.
- Πρόβλημα δρομολόγησης ενός σχολικού λεωφορείου: Το πρόβλημα αυτό διερευνήθηκε ως μια παραλλαγή του mTSP με κάποιους περιορισμούς. Ο στόχος του προγραμματισμού είναι να δημιουργηθεί ένα σχέδιο λειτουργίας των λεωφορείων, έτσι ώστε να ελαχιστοποιηθεί ο αριθμός των διαδρομών, η συνολική απόσταση που διανύεται από όλα τα λεωφορεία να διατηρηθεί στο ελάχιστο, κανένα λεωφορείο να μην είναι υπερφορτωμένο και ο χρόνος που απαιτείται για να διασχίσει οποιοδήποτε λεωφορείο οποιαδήποτε διαδρομή να μην υπερβαίνει μια μέγιστη επιτρεπόμενη πολιτική [122].
- Πρόβλημα προγραμματισμού πληρώματος: Μια εφαρμογή για μεταφορά καταθέσεων μεταξύ διαφορετικών υποκαταστημάτων τραπεζών. Εδώ, οι καταθέσεις πρέπει να γίνονται στα υποκαταστήματα των τραπεζών και να επιστρέφουν στο κεντρικό γραφείο από ένα πλήρωμα αγγελιοφόρων. Το πρόβλημα είναι να προσδιορισθούν οι διαδρομές που έχουν συνολικά ελάχιστο κόστος [122].

- Πρόβλημα προγραμματισμού συνέντευξης: Οι Γκιλμπερτ και Χοφστρα (Gilbert και Hofstra) [68] βρήκαν την εφαρμογή του mTSP στον προγραμματισμό συνεντεύξεων μεταξύ των μεσιτών μιας διαδρομής και των προμηθευτών της τουριστικής βιομηχανίας. Κάθε μεσίτης αντιστοιχεί σε έναν πωλητή ο οποίος πρέπει να επισκεφθεί ένα καθορισμένο σύνολο από τοποθεσίες του προμηθευτή, οι οποίες αντιπροσωπεύονται από ένα σύνολο T πόλεων [122].
- Πρόβλημα σχεδιασμού αποστολής: Το πρόβλημα αυτό συνίσταται στον καθορισμό μιας βέλτιστης διαδρομής για κάθε σώμα στρατιωτών, ώστε να επιτευχθούν οι στόχοι της αποστολής στο ελάχιστο δυνατό χρονικό διάστημα. Ο αρμόδιος για το σχεδιασμό της αποστολής χρησιμοποιεί μια παραλλαγή του mTSP, όπου υπάρχουν n σχεδιαστές, m στόχοι που πρέπει να επισκεφθούν ορισμένοι από τους σχεδιαστές και μια πόλη-βάση, στην οποία πρέπει τελικά να επιστρέψουν όλοι οι σχεδιαστές. Με παρόμοιο τρόπο, τα προβλήματα δρομολόγησης που προκύπτουν στο σχεδιασμό εφαρμογών μη επανδρωμένων εναέριων οχημάτων μπορούν επίσης να μοντελοποιηθούν ως στιγμιότυπα του mTSP [122].
- Σχεδιασμός του παγκόσμιου δορυφορικού συστήματος πλοήγησης που αποτυπώνει δίκτυα: Μια πολύ πρόσφατη και πολύ ενδιαφέρουσα εφαρμογή του mTSP, προκύπτει στο σχεδιασμό του παγκόσμιου δορυφορικού συστήματος πλοήγησης (Global Navigation Satellite System, GNSS) που αποτυπώνει δίκτυα. Ένα GNSS είναι ένα δορυφορικό σύστημα με βάση το διάστημα, το οποίο παρέχει κάλυψη για όλες τις τοποθεσίες σε όλο τον κόσμο και είναι ιδιαίτερα σημαντικό σε πραγματικές εφαρμογές, όπως η έγκαιρη προειδοποίηση και η διαχείριση των καταστροφών, η παρακολούθηση του περιβάλλοντος και της γεωργίας κ.λ.π. Ο στόχος της αποτύπωσης είναι να προσδιορισθούν οι γεωγραφικές θέσεις άγνωστων σημείων πάνω στη γη, αλλά και πάνω από τη γη με τη χρήση δορυφορικού εξοπλισμού. Αυτά τα σημεία, στα οποία έχουν τοποθετηθεί δέκτες, συντονίζονται από μια σειρά συνόδων παρατήρησης. Όταν υπάρχουν πολλαπλοί δέκτες ή πολλαπλές περιόδους λειτουργίας, το πρόβλημα της εύρεσης της καλύτερης σειράς των συνόδων για τους δέκτες μπορεί να διαμορφωθεί ως ένα mTSP.

3.5 Παραλλαγές του Προβλήματος του Πλανόδιου Πωλητή

- Μετρική: Στη μετρική εκδοχή, οι υπεραστικές αποστάσεις ικανοποιούν την ανισότητα τριγώνου. Ένας πολύ φυσικός περιορισμός είναι να απαιτήσει ότι οι αποστάσεις μεταξύ των πόλεων σχηματίζουν ένα μετρικό για να ικανοποιήσει την ανισότητα του τριγώνου που είναι η άμεση σύνδεση από το A στο B και δεν είναι ποτέ μακρύτερα από τη διαδρομή μέσω του ενδιάμεσου C . Η ακμή εκτείνεται στη συνέχεια και δημιουργεί μια μέτρηση στο σύνολο των κορυφών. Όταν οι πόλεις προβάλλονται ως πόντοι στο επίπεδο, πολλές φυσικές λειτουργίες απόστασης είναι μετρήσεις και τόσες πολλές φυσικές παρουσίες ικανοποιούν αυτόν τον περιορισμό. Αυτή η μέτρηση συχνά ονομάζεται μέτρηση απόστασης της πόλης. Στη μέγιστη μέτρηση, η απόσταση μεταξύ δύο σημείων είναι το μέγιστο των απόλυτων τιμών των διαφορών των συντεταγμένων x και y . Οι δύο τελευταίες μετρήσεις εμφανίζονται, για παράδειγμα, στη δρομολόγηση ενός μηχανήματος που τρυπάει ένα δεδομένο σύνολο οπών σε μια εκτυπωμένη πλακέτα κυκλώματος. Το μετρικό του Μανχάταν αντιστοιχεί σε ένα μηχανήμα που ρυθμίζει το πρώτο συντονισμό και μετά το άλλο, έτσι ώστε ο χρόνος να μετακινηθεί σε ένα νέο σημείο είναι το άθροισμα και των δύο κινήσεων. Η μέγιστη μέτρηση αντιστοιχεί σε ένα μηχανήμα που ρυθμίζει ταυτόχρονα και

τις δύο συντεταγμένες, έτσι ώστε ο χρόνος να μετακινηθεί σε ένα νέο σημείο είναι η πιο αργή από τις δύο κινήσεις. Στον ορισμό του, το πρόβλημα δεν επιτρέπει στις πόλεις να τις επισκεπτονται δύο φορές, αλλά πολλές εφαρμογές δεν χρειάζονται αυτόν τον περιορισμό. Σε αυτές τις περιπτώσεις, μια συμμετρική, μη μετρική παρουσία μπορεί να μειωθεί σε ένα μετρικό.

- Ευκλείδεια: Όταν οι αριθμοί εισόδου μπορεί να είναι αυθαίρετοι πραγματικοί αριθμοί, η Ευκλείδεια εκδοχή είναι μια συγκεκριμένη περίπτωση του μετρικού, καθώς οι αποστάσεις σε ένα επίπεδο υπακούουν στην ανισότητα του τριγώνου. Όταν οι αριθμοί εισόδου πρέπει να είναι ακέραιοι, η σύγκριση των μήκους των περιηγήσεων περιλαμβάνει τη σύγκριση αθροισμάτων από τετραγωνικές ρίζες. Με λογικές συντεταγμένες και διακριτική μέτρηση (αποστάσεις στρογγυλοποιημένες μέχρι έναν ακέραιο), το πρόβλημα είναι πλήρες. Με αυθαίρετες πραγματικές συντεταγμένες, η Ευκλείδεια εκδοχή δεν μπορεί να είναι σε τέτοιες τάξεις, καθώς υπάρχουν πολλές δυνατές εισροές. Ωστόσο, είναι πιθανώς η ευκολότερη εκδοχή για την προσέγγιση.

Σε γενικές γραμμές, για οποιοδήποτε $c > 0$, όπου d είναι ο αριθμός των διαστάσεων στο χώρο, υπάρχει ένας αλγόριθμος πολυωνυμικής ώρας που βρίσκει μια διαδρομή μήκους το πολύ $(1 + 1/c)$ φορές τη βέλτιστη για γεωμετρικές εμφανίσεις του προβλήματος. Στην πράξη, εξακολουθούν να χρησιμοποιούνται απλούστερες εκδοχές με ασθενέστερες εγγυήσεις.

3.6 TSP με χρονικά παράθυρα

Το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα (Travelling Salesman Problem with Time Windows - TSPTW) είναι μια παραλλαγή του ευρέως γνωστού προβλήματος του πλανόδιου πωλητή (TSP). Το TSPTW είναι παρόμοιο με το TSP, με εξαίρεση ότι οι πόλεις (ή πελάτες) θα πρέπει να δέχονται επίσκεψη μέσα σε ένα συγκεκριμένο χρονικό διάστημα (ή «παράθυρο»).

Στο έργο του, ο Παπαλίτσας κ.α. (Papalitsas et al.) [151] δίνει τον επίσημο ορισμό του κλασικού TSPTW:

- Ορισμός 1. Αν $G = (N, A)$ όπου $N = 0, 1, \dots, N$ είναι το πεπερασμένο σύνολο κόμβων ή, πιο συχνά αναφέρεται σε αυτό το πλαίσιο ως πελάτες, και $A = N \times N$ είναι το σύνολο των ακμών που συνδέουν τους πελάτες. Για κάθε ζεύγος (u, v) πελατών, υπάρχει μια ακμή στο A . Μια διαδρομή ορίζεται από τη σειρά με την οποία επισκεπτονται τους πελάτες. Μερικές παραδοχές διευκολύνουν τη διατύπωση του TSPTW.
- Ορισμός 2. Αφήνουμε τον πελάτη 0 να δηλώσει την αρχική θέση και υποθέστε ότι κάθε διαδρομή αρχίζει και τελειώνει στην αρχική θέση. Κάθε ένας από τους εναπομείναντες n πελάτες εμφανίζεται ακριβώς μία φορά στην περιοδεία. Δηλώνεται μια ξενάγηση ως ταξινομημένη λίστα $P = (P_0, R_1, \dots, P_N, P_{N+1})$, όπου p_i είναι ο δείκτης του πελάτη στη θέση της i -της διαδρομής. Σύμφωνα με την προηγούμενη υπόθεση, $P_0 = P_{N+1} = 0$.
- Ορισμός 3. Για κάθε ζεύγος (u, v) των πελατών $u, v \in N$, υπάρχει ένα κόστος $c_{u,v}$, για την τροχιά της ακμής (u, v) . Αυτό το κόστος της διαδρομής της ακμής από την αρχή στο v γενικά αποτελείται από οποιοδήποτε χρόνο εξυπηρέτησης στον πελάτη U συν το χρόνο ταξιδιού από τον πελάτη στον πελάτη v .
- Ορισμός 4. Σε κάθε πελάτη $v \in N$, υπάρχει ένα συσχετισμένο χρονικό παράθυρο $[E_v, L_v]$, κατά τη διάρκεια του οποίου πρέπει να επισκεφθεί κανείς τον πελάτη v . Υποθέτουμε ότι επιτρέπεται η αναμονή. Ένα όχημα επιτρέπεται να φθάσει στον πελάτη v πριν από την έναρξη του χρονικού του παραθύρου, E_v , αλλά το όχημα δεν μπορεί να αναχωρήσει από

τον πελάτη v πριν από τον E_v .

Κατά κάποιο τρόπο, η δυσκολία του TSPTW πηγάζει από το γεγονός ότι είναι δύο προβλήματα σε ένα: ένα πρόβλημα πλανόδιου πωλητή και ένα πρόβλημα προγραμματισμού. Το πρόβλημα μόνο είναι ένα από τα πιο διάσημα προβλήματα βελτιστοποίησης NP, ενώ το τμήμα προγραμματισμού, με ημερομηνίες απελευθέρωσης και παράδοσης, προσθέτει σημαντικά εμπόδια στην ήδη υπάρχουσα δυσκολία. Για να επαληθεύσουμε ότι η περιήγηση είναι εφικτή, δηλαδή, ικανοποιεί τα χρονικά παράθυρα, είναι σκόπιμο να εισαχθεί η ώρα άφιξης στον πελάτη και ο χρόνος κατά τον οποίο η υπηρεσία ξεκινά από τον πελάτη, ο οποίος συμβολίζεται με API και dpi , αντίστοιχα. Σε αυτό το σημείο, κάνουμε τη σημαντική παρατήρηση ότι το dpi είναι η ώρα αναχώρησης από τον πελάτη με την περίπτωση μηδενικού χρόνου υπηρεσίας. Η παραδοχή ότι ο χρόνος μηδενικής εξυπηρέτησης χρησιμοποιείται ευρέως στη βιβλιογραφία για την απλοποίηση του προβλήματος [49].

Αυτός ο πρόσθετος χρονικός περιορισμός περιορίζει το δένδρο αναζήτησης, διότι όλες οι TSP λύσεις δεν είναι απαραίτητα και TSPTW λύσεις. Ωστόσο, καθιστά το πρόβλημα ακόμη πιο δύσκολο στην πράξη. Πράγματι, η όμορφη συμμετρία του TSP, όπου οποιαδήποτε μετάθεση των πόλεων αποτελεί μια εφικτή λύση, παραβιάζεται και ακόμα και η αναζήτηση εφικτών λύσεων είναι δύσκολη [169].

Ας υποθέσουμε τώρα ότι μας δίνεται μια αποθήκη (depot), ένα σύνολο από πελάτες, ο χρόνος εξυπηρέτησης (δηλαδή, το χρονικό διάστημα που πρέπει να αναλωθεί στον πελάτη) και ένα χρονικό παράθυρο (δηλαδή, ο χρόνος ετοιμότητας του και η ημερομηνία λήξης του). Το πρόβλημα TSPTW συνίσταται στην εξεύρεση μιας διαδρομής ελαχίστου κόστους, η οποία ξεκινάει και καταλήγει σε μια δεδομένη αποθήκη, αφούτου κάθε πελάτης την έχει επισκεφθεί μόνο μία φορά πριν από την ημερομηνία λήξης της.

Η αποθήκη έχει ιδιαίτερη σημασία για την εκδοχή TSPTW, ενώ κάτι τέτοιο δεν ισχύει για τη γενική εκδοχή του TSP. Ο πλανόδιος πωλητής επιτρέπεται να φθάσει στον πελάτη πριν από το χρόνο ετοιμότητας του, αλλά πρέπει να περιμένει. Προφανώς, υπάρχουν διαδρομές οι οποίες δεν επιτρέπουν στον πλανόδιο πωλητή να τηρεί τις προβλεπόμενες προθεσμίες όλων των πελατών.

Αυτές οι διαδρομές καλούνται ανέφικτες, ενώ όσες διαδρομές επιτρέπουν στον πλανόδιο πωλητή να τηρεί τις προβλεπόμενες προθεσμίες όλων των πελατών λέγονται εφικτές διαδρομές (ή εφικτές λύσεις). Το κόστος μιας διαδρομής είναι η συνολική απόσταση που έχει διανυθεί [128]. Δίνεται γράφημα $G = (V, A)$ όπου $V = [1, 2, \dots, n]$. Έστω ότι το 0 υποδηλώνει μια αποθήκη και $A = (i, j) : i, j \in V \cup 0$ είναι το σύνολο των ακμών μεταξύ των πελατών.

Το κόστος της περιπλάνησης (travelling cost) από το i στο j αντιπροσωπεύεται από το $c_{i,j}$, η οποία περιλαμβάνει τόσο το χρόνο εξυπηρέτησης του πελάτη i , όσο και το χρόνο που χρειάζεται για να μεταβούμε από το i στο j . Κάθε πελάτης i έχει ένα συσχετιζόμενο με αυτόν χρονικό παράθυρο a_i, b_i , όπου τα a_i και b_i αντιπροσωπεύουν το χρόνο ετοιμότητας και την ημερομηνία λήξης, αντίστοιχα. Έτσι, το TSPTW μπορεί να δηλωθεί, μαθηματικά, ως ένα πρόβλημα εξεύρεσης μιας Χαμιλτονιανής διαδρομής, η οποία ξεκινά και καταλήγει στην αποθήκη (depot), ικανοποιώντας όλους τους χρονικούς περιορισμούς και ελαχιστοποιώντας τη συνολικά διανυόμενη απόσταση [128].

Σε πραγματικές εφαρμογές, ο χρόνος που δαπανάται σε έναν πελάτη μπορεί να περιορίζεται ανάλογα με την υπηρεσία. Παραδείγματος χάριν, ίσως να χρειάζεται να περιμένει κανείς μπροστά από το γραφείο του πελάτη. Είθισται να θεωρείται ότι η εξυπηρέτηση αρχίζει και ολοκληρώνεται το συντομότερο δυνατόν. Μερικοί συγγραφείς, όπως παραδείγματος χάριν οι Σ. Ντας κ.α. (S. Dash et al.) [41], εκχωρούν δύο κόστη στις ακμές του γράφου: το κόστος της περιπλάνησης (travel cost) και το χρόνο (travel time) της. Ενώ οι χρόνοι της περιπλάνησης πρέπει να σέβονται

τους περιορισμούς σχετικά με τα χρονικά παράθυρα, η τιμή της αντικειμενικής συνάρτησης είναι το άθροισμα από τα κόστη της περιπλάνησης στις ακμές του γράφου.

Το TSPTW έχει ποικίλες πρακτικές εφαρμογές σε προβλήματα δρομολόγησης (routing) και προγραμματισμού (scheduling). Ειδικότερα, το TSPTW μπορεί να θεωρηθεί ως ένα υποπρόβλημα του προβλήματος δρομολόγησης οχημάτων (Vehicle Routing Problem - VRP) με χρονικά παράθυρα (VRP with Time Windows - VRPTW), λαμβάνοντας υπόψη τη χαλάρωση των περιορισμών δυναμικότητας (capacity constraints).

3.7 Η κλασική μαθηματική περιγραφή του TSPTW

Στην περίπτωση του συμμετρικού TSP ο αριθμός όλων των πιθανών διαδρομών που καλύπτουν τις πόλεις και ανταποκρίνονται σε πιθανές λύσεις δίνεται από το $(n-1)!/2$ (όπου n ο αριθμός των πόλεων). Το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα (Traveling Salesman Problem with Time Windows - TSPTW) είναι μια εξειδικευμένη περίπτωση του TSP προβλήματος, όπου ο πωλητής πρέπει να περάσει από κάθε πόλη μέσα σε ένα συγκεκριμένο, καθορισμένο χρονικό όριο (ή αλλιώς παράθυρο). Αυτή η προϋπόθεση θέτει έναν αυστηρό χρονικό περιορισμό. Σε αυτή την περίπτωση, οι λύσεις του TSP δεν είναι και όλες λύσεις του TSPTW. Κατά συνέπεια, το ζήτημα έχει γίνει ακόμη πιο δύσκολο στην πραγματικότητα. Πράγματι, στο τυπικό TSP μπορεί κανείς να αναγνωρίσει μια κομψή συμμετρία με την έννοια ότι κάθε μεταβολή πόλεων είναι μια βιώσιμη λύση. Αυτή η συμμετρία είναι σπασμένη στο TSPTW και, δυστυχώς, ακόμη και η εύρεση μιας αρχικής αποδεκτής λύσης γίνεται πολύ δύσκολη [170]. Συνήθως, στο TSPTW δίνονται τα ακόλουθα δεδομένα εισόδου:

- i. μια συλλογή πόλεων (αναφέρεται επίσης ως πελάτες),
- ii. ένας συγκεκριμένος πελάτης που ονομάζεται αποθήκη,
- iii. ο χρόνος εξυπηρέτησης που θα δαπανηθεί σε κάθε πελάτη
- iv. το χρονικό παράθυρο κάθε πελάτη (δηλαδή την ώρα έναρξης και την ώρα λήξης).

Στη συνέχεια πρέπει να υπολογιστεί η διαδρομή ελαχίστου κόστους, η οποία ξεκινά και τελειώνει στην αποθήκη, τέτοια ώστε ο πωλητής να έχει επισκεφθεί κάθε πελάτη πριν από την ώρα λήξης του.

Σημειώνεται ότι στην περίπτωση του TSPTW ο τρόπος καθορισμού της αποθήκης είναι κρίσιμος, κάτι που προφανώς δεν ισχύει στο τυπικό TSP. Διευκρινίζεται επίσης ότι ο πωλητής επιτρέπεται να φτάσει σε έναν πελάτη πριν να είναι έτοιμος, με αποτέλεσμα αναπόφευκτη αναμονή. Υπάρχουν, φυσικά, διαδρομές που απαγορεύουν στον πωλητή να συμμορφώνεται με το χρονικό διάστημα κάθε πελάτη. Αυτές οι διαδρομές αναφέρονται ως ανέφικτες και αυτές που επιτρέπουν την πλήρη συμμόρφωση με το χρονικό διάστημα κάθε πελάτη αναφέρονται ως εφικτές διαδρομές. Το πλήρες εύρος που καλύπτεται είναι το κόστος της διαδρομής.

Σε αυτό το σημείο δίνεται ο επίσημος ορισμός του κλασικού TSPTW. Στηρίζεται στο [142], το οποίο είναι καθιερωμένο άρθρο στη σχετική βιβλιογραφία.

Ορισμός 1. Έστω ότι $G = (N, A)$ είναι ένας κατευθυνόμενος γράφος, όπου $N = \{0, 1, \dots, n\}$ είναι το πεπερασμένο σύνολο των κόμβων, ή όπως αναφέρονται πιο συχνά, το σύνολο των πελατών και $A = N \times N$ είναι το σύνολο των ακμών τα οποία ενώνουν τους πελάτες. Για κάθε ζεύγος πελατών (u, v) υπάρχει μια ακμή στο A . Μια διαδρομή ορίζεται από τη σειρά με την οποία ο πωλητής θα επισκεφθεί τους πελάτες.

Μερικές υποθέσεις διευκολύνουν τη διατύπωση του TSPTW.

Ορισμός 2. Έστω ότι ο πελάτης 0 είναι η αποθήκη και υποθέτουμε ότι κάθε διαδρομή ξεκινά και τελειώνει στην αποθήκη. Κάθε ένας από τους υπόλοιπους n πελάτες εμφανίζονται ακριβώς μία φορά στη διαδρομή. Η διαδρομή ορίζεται ως μια ταξινομημένη λίστα $\mathcal{P} = (p_0, p_1, \dots, p_n, p_{n+1})$, όπου p_i είναι ο δείκτης του πελάτη στην $i^{\text{η}}$ θέση της διαδρομής. Σύμφωνα με την προηγούμενη υπόθεση $p_0 = p_{n+1} = 0$.

Ορισμός 3. Για κάθε ζεύγος (u, v) πελατών $u, v \in N$, υπάρχει ένα κόστος $c_{u,v}$ για να διασχίσει ο πωλητής την ακμή (u, v) . Αυτό το κόστος μετακίνησης στην ακμή από το u στο v γενικά αποτελείται από τον χρόνο εξυπηρέτησης του πελάτη u συν το κόστος ταξιδιού από τον πελάτη u στον πελάτη v .

Ορισμός 4. Για κάθε πελάτη $v \in N$ υπάρχει ένα ορισμένο χρονικό παράθυρο $[e_v, l_v]$, κατά το οποίο ο πελάτης v πρέπει να δεχθεί επίσκεψη από τον πωλητή. Υποθέτουμε ότι επιτρέπεται να περιμένει. Ένα όχημα μπορεί να φτάσει στον πελάτη v πριν από την έναρξη του χρονικού παραθύρου του, e_v , αλλά το όχημα δεν μπορεί να αναχωρήσει από τον πελάτη v πριν e_v . Μια διαδρομή είναι εφικτή εάν ικανοποιεί τα χρονικά παράθυρα κάθε πελάτη.

Στη βιβλιογραφία εξετάζονται συνήθως δύο βασικές αντικειμενικές συναρτήσεις TSPTW:

- ελαχιστοποίηση του αθροίσματος του κόστους μετακίνησης μεταξύ των ακμών σε μια διαδρομή και
- ελαχιστοποίηση του χρόνου επιστροφής στην αποθήκη.

Κατά κάποιο τρόπο, η δυσκολία του TSPTW πηγάζει από το γεγονός ότι είναι δύο προβλήματα σε ένα, ένα πρόβλημα πλανόδιου πωλητή (TSP) και ένα πρόβλημα προγραμματισμού (scheduling problem). Το TSP από μόνο του είναι ένα από τα πιο διάσημα NP-hard προβλήματα βελτιστοποίησης, ενώ το τμήμα προγραμματισμού, με προθεσμίες κυκλοφορίας και λήξης, προσθέτει σημαντικά στην ήδη υπάρχουσα δυσκολία. Για να επαληθευτεί ότι η περιήγηση είναι εφικτή, δηλαδή ικανοποιεί τα χρονικά διαστήματα, είναι σκόπιμο να εισαχθεί η ώρα άφιξης στον $i^{\text{ο}}$ πελάτη και την ώρα έναρξης στον $i^{\text{ο}}$ πελάτη, που υποδηλώνεται από A_{p_i} και D_{p_i} , αντίστοιχα.

Σε αυτό το σημείο γίνεται μια σημαντική παρατήρηση ότι το D_{p_i} είναι η ώρα αναχώρησης από τον $i^{\text{ο}}$ πελάτη στην περίπτωση μηδενικού χρόνου υπηρεσίας. Σημειώνεται ωστόσο ότι η υπόθεση μηδενικού χρόνου υπηρεσίας δεν σημαίνει απαραίτητα ότι $D_{p_i} = A_{p_i}$, λόγω της πιθανότητας αναμονής εάν $A_{p_i} < e_{p_i}$. Η υπόθεση μηδενικού χρόνου υπηρεσίας χρησιμοποιείται ευρέως στη βιβλιογραφία για να απλοποιηθεί το πρόβλημα και αυτό ακολουθείται και σε αυτή την υπόθεση.

Χρησιμοποιώντας μια προσέγγιση μεθόδου ποινής, κάποιος μπορεί να αποσυνθέσει εν μέρει αυτά τα δύο συστατικά και να πραγματοποιήσει μια ευρετική αναζήτηση. Χαλαρώνοντας τους χρονικούς περιορισμούς θεωρείται ότι προκύπτουν ανέφικτες λύσεις

$$\{D_{p_i} \leq l_{p_i} : i = 1, \dots, n\} \quad (3.1)$$

μέσα στην αντικειμενική συνάρτηση με μια συνάρτηση ποινής της μορφής:

$$p(\mathcal{S}) = \sum_{i=1}^n \max [(0, D_{p_i} - l_{p_i})]^s \quad (3.2)$$

για κάποια $s > 0$.

Αποδείχτηκε στο [13] ότι για έναν πολύ μεγάλο μη αρνητικό πολλαπλασιαστή ποινών, λ , οι λειτουργίες ποινής διατηρούν ισχυρή δυαδικότητα μεταξύ της χαλάρωσης και της αρχικής διατύπωσης.

Εάν το $(p_0, p_1, \dots, p_n, p_{n+1})$ είναι μια εφικτή περιήγηση και με δεδομένες τις προηγούμενες παραδοχές, η κλασική διαμόρφωση του TSPTW μπορεί να συνοψιστεί από τους τύπους (3.3) - (3.6) που δίνεται παρακάτω (δείτε επίσης [142]).

$$\min \sum_{i=1}^{n+1} c_{p_{i-1}, p_i} \quad (3.3)$$

$$D_{p_0} = 0. \quad (3.4)$$

$$A_{p_i} = D_{p_{i-1}} + c_{p_{i-1}, p_i} \quad (1 \leq i \leq n+1). \quad (3.5)$$

$$D_{p_i} = \max\{A_{p_i}, e_{p_i}\} \quad (1 \leq i \leq n). \quad (3.6)$$

3.7.1 Ακριβείς Αλγόριθμοι για το TSPTW

Η διεθνής βιβλιογραφία περιέχει κάποιους ακριβείς αλγόριθμους για το TSPTW. Οι Λάνγκεβιν κ.α. (Langevin et al.) [112] εισήγαγαν μια διατύπωση ροής δύο στοιχείων, η οποία μπορεί να επεκταθεί στο κλασικό “makespan” πρόβλημα. Με τον όρο “makespan” εννοούμε το μέγιστο χρόνο επεξεργασίας και ολοκλήρωσης, ήτοι όταν όλες οι θέσεις εργασίας έχουν ολοκληρώσει την επεξεργασία. Οι Ντούμας κ.α. (Dumas et al.) [49] χρησιμοποιούν μια προσέγγιση δυναμικού προγραμματισμού με δοκιμές εξάλειψης, οι οποίες βελτιώνουν την απόδοση και μειώνουν το χώρο αναζήτησης, τόσο εκ των προτέρων, όσο και κατά τη διάρκεια της εκτέλεσής της. Οι Πεσάντ κ.α. (Pesant et al.) προτείνουν έναν αλγόριθμο διακλάδωσης και οριοθέτησης (branch and bound), χρησιμοποιώντας ένα μοντέλο προγραμματισμού με περιορισμούς. Ομοίως, οι Φοκάτσι κ.α. (Focacci et al.) [59] προτείνουν μια υβριδική προσέγγιση, η οποία βασίζεται στον προγραμματισμό με περιορισμούς και στις τεχνικές βελτιστοποίησης [38].

3.7.2 Ευρετικές για το TSPTW

Το TSPTW είναι NP-hard πρόβλημα, δεδομένου ότι είναι μια ειδική περίπτωση του περίφημου TSP, το οποίο είναι NP-hard. Έτσι, υπάρχει ανάγκη για μια ευρετική, η οποία να είναι σε θέση να λύσει αποτελεσματικά ρεαλιστικές περιπτώσεις σε εύλογο χρονικό διάστημα. Σε αυτή την κατεύθυνση, έχουν ήδη γίνει μερικά βήματα.

Οι Κάρλτον και Μπαρνς (Carlton και Barnes) [27] χρησιμοποιούν μια ευρετική αναζήτηση ταμπού με στατική συνάρτηση ποινής, χρησιμοποιώντας ανέφικτες λύσεις στην αναζήτηση. Οι Γκέντρου κ.α. (Gendreau et al.) [65] προτείνουν μια ευρετική εισαγωγή (insertion heuristic) βασισμένη στην ευρετική GENIUS, η οποία χτίζει σταδιακά τη διαδρομή στη φάση κατασκευής και τη βελτιώνει σε μια φάση βελτίωσης (βασισμένη σε διαδοχική αφαίρεση και επανένταξη των κόμβων). Ο Κάλβο (Calvo) [26] λύνει ένα πρόβλημα ανάθεσης με μια ad hoc αντικειμενική συνάρτηση και χτίζει μια εφικτή διαδρομή, συγχωνεύοντας όσες τέτοιες διαδρομές έχουν βρεθεί σε μια κύρια

διαδρομή έπειτα, εφαρμόζεται μια 3-opt διαδικασία τοπικής αναζήτησης για τη βελτίωση της αρχικής εφικτής λύσης.

Οι Όλμαν και Τόμας (Ohlmann και Thomas) [142] χρησιμοποιούν μια παραλλαγή της προσομοιωμένης απόπτησης, που ονομάζεται συμπίεσμένη απόπτηση, η οποία χαλαρώνει τους περιορισμούς των χρονικών παραθύρων, ενσωματώνοντας μια μέθοδο μεταβλητής ποινής εντός μιας στοχαστικής διαδικασίας αναζήτησης. Το 2010, προτείνονται δύο νέες ευρετικές, από τους Μπλουμ κ.α. (Blum et al.) [20] και από τους Ουρούτια κ.α. (Urrutia et al.) [38], ενώ το 2012, στο [128] δημοσιεύεται μια ενδιαφέρουσα σύγκριση για τα αποτελέσματα των παραπάνω ευρετικών, καθώς και μια νέα ευρετική για το TSPTW. Οι τρεις ευρετικές που προαναφέρθηκαν μπορούν να θεωρηθούν επί του παρόντος οι σημαντικότερες ευρετικές για το TSPTW πρόβλημα.

Οι Μπλουμ (Blum et al.) [20] προτείνουν μια υβριδική μέθοδο, η οποία συνδυάζει τη βελτιστοποίηση αποικίας μυρμηγκιών (ant colony optimization - ACO) με την ακτινική αναζήτηση (Beam Search - BS). Οι αλγόριθμοι αυτοί ονομάζονται Beam-ACO αλγόριθμοι και στηρίζονται σε μεγάλο βαθμό σε ακριβείς και υπολογιστικά φθηνές πληροφορίες οριοθέτησης για τη διαφοροποίηση μεταξύ των επιμέρους λύσεων. Οι Ουρούτια κ.α. (Urrutia et al.) [38] πρότειναν μια ευρετική δύο σταδίων, η οποία βασίζεται στην αναζήτηση μεταβαλλόμενης γειτονιάς (Variable Neighborhood Search – VNS). Στο πρώτο στάδιο, κατασκευάζεται μια εφικτή λύση με τη χρήση της VNS, όπου η μεικτή, γραμμική, αθέρα, αντικειμενική συνάρτηση αναπαρίσταται ως ένα μέτρο ανεφικτότητας (infeasibility). Στο δεύτερο στάδιο η ευρετική βελτιώνει την εφικτή λύση με μια ευρετική της γενικής εκδοχής της VNS (General VNS - GVNS).



4. Η κβαντικά εμπνευσμένη μεθευρετική μέθοδος

4.1 Οι προτεινόμενες μεθευρετικές υλοποιήσεις

Σε μια προσπάθεια να σχηματίσουμε μια ολοκληρωμένη άποψη σχετικά με αυτόν τον πιθανό αντίκτυπο των μεθόδων διαταραχής των μεθευρετικών διαδικασιών στην συνολική ποιότητα της λύσης που αυτές παρέχουν, προχωρήσαμε σε περαιτέρω ανάλυση στις λύσεις από διαφορετικές υλοποιήσεις για ασύμμετρα συμμετρικά και παγκόσμια TSP μετροπροβλήματα από την TSPLib. Για να εξεταστεί αυτός ο πιθανός αντίκτυπος των διαφόρων στρατηγικών διαταραχής, οι τρεις μέθοδοι διαταραχής εξετάστηκαν στο ίδιο βήμα βελτίωσης. Επιπλέον, τα σχήματα GVNS που προέκυψαν εκτελέστηκαν με δύο διαφορετικές στρατηγικές αναζήτησης για την βελτιστοποίηση της λύσης. Με βάση την πρώτη βελτίωση που θα επιτευχθεί (first improvement search strategy) και με βάση την καλύτερη βελτίωση που θα επιτευχθεί σε έναν κύκλο (best improvement search strategy). Επίσης, δύο διαφορετικά χρονικά όρια χρησιμοποιήθηκαν ως τα κύρια κριτήρια διακοπής κατά την εκτέλεση των πειραμάτων: τα 60 δευτερόλεπτα και τα 120 δευτερόλεπτα. Τα πειραματικά αποτελέσματα αναλύθηκαν στατιστικά για να διαπιστωθεί εάν η χρήση διαφορετικών μεθόδων διαταραχής επηρεάζει την απόδοση του αλγορίθμου GVNS. Τα ευρήματά μας δείχνουν ότι η χρήση διαφορετικών στρατηγικών διαταραχής επηρεάζει σαφώς την ποιότητα της λύσης στις περιπτώσεις μη συμμετρικών TSP στιγμιοτύπων, ενώ δεν παρατηρήθηκαν σημαντικές διαφορές για την περίπτωση συμμετρικών στιγμιοτύπων (sTSP), με εξαίρεση τα πειράματα που πραγματοποιήθηκαν με τη χρήση της στρατηγικής της καλύτερης βελτίωσης (Best Improvement) και το χρονικό όριο εκτέλεσης των 120 δευτερολέπτων. Επιπλέον, για να εξεταστεί η αποτελεσματικότητα των διαμορφωμένων μεθόδων, πραγματοποιείται σύγκριση μεταξύ των αποτελεσμάτων που έχουν ληφθεί και άλλων πρόσφατων προσεγγιστικών λύσεων για το TSP από την βιβλιογραφία. Όπως μπορεί να επιβεβαιωθεί από τα πειραματικά μας αποτελέσματα, τα προτεινόμενα σχήματα GVNS παράγουν καλύτερες λύσεις από τις άλλες μεθευρετικές. Οι προτεινόμενες μέθοδοι GVNS χρησιμοποιούν το σχήμα pipe-VND, που σημαίνει ότι η αναζήτηση πραγματοποιείται στην ίδια γειτονιά όπου πραγματοποιείται η βελτίωση, όπως και η φάση βελτίωσής τους.

4.1.1 Δομές Γειτνίασης

Τρεις τοπικοί τελεστές αναζήτησης εξετάζονται για την εξερεύνηση διαφορετικών λύσεων:

- **1-0 Relocate.** Αυτή η κίνηση αφαιρεί τον κόμβο από την τρέχουσα θέση του στη διαδρομή και το εισάγει ξανά μετά σε έναν άλλο επιλεγμένο κόμβο.
- **2-Opt.** Η κίνηση 2-Opt σπάει δύο τόξα στην τρέχουσα λύση και τα επανασυνδέει με διαφορετικό τρόπο.
- **1-1 Exchange.** Αυτή η κίνηση ανταλλάσσει δύο κόμβους στην τρέχουσα διαδρομή.

Και οι τρεις δομές γειτνίασης ενσωματώνονται σε ένα σχήμα pipe-VND, όπως απεικονίζεται στον Αλγόριθμο 3, όπου το $l_{max} = 3$ δηλώνει τον αριθμό των δομών γειτονιάς.

Algorithm 3 pipe-VND.

```

1: procedure PVND( $N, l_{max}$ )
2:    $l = 1$ 
3:   while  $l \leq l_{max}$  do
4:     select case( $l$ )
5:       case(1) :  $S' \leftarrow$  1-0 Relocate( $S$ )
6:       case(2) :  $S' \leftarrow$  2-Opt( $S$ )
7:       case(3) :  $S' \leftarrow$  1-1 Exchange( $S$ )
8:     end select
9:     if  $f(S') < f(S)$  then
10:       $S \leftarrow S'$ 
11:     else
12:       $l = l + 1$ 
13:     end if
14:   end while
15:   return  $S$ 
16: end procedure
    =0

```

4.1.2 Μέθοδοι διαταραχής

Για να αποφευχθεί η παγίδευση σε τοπικά ελάχιστα, εξετάζονται τρεις διαφορετικές διαδικασίες διαταραχής. Αυτές οι μέθοδοι διαταραχής είναι οι ακόλουθες:

Διαταραχή_1. Αυτή η μέθοδος διαταραχής της λύσης επιλέγει τυχαία μία από τις προκαθορισμένες δομές γειτνίασης και εφαρμόζει την εφαρμόζει k φορές ($1 < k < k_{max}$, όπου το k_{max} είναι ο μέγιστος αριθμός των επαναλήψεων της μεθόδου διαταραχής) στην τρέχουσα λύση του αλγορίθμου. Η συγκεκριμένη μέθοδος συνοψίζεται στον αλγόριθμο 4.

Algorithm 4 Διαταραχή_1.

```

1: procedure SHAKE_1( $S, k, l_{max}$ )
2:    $l = \text{random\_integer}(1, l_{max})$ 
3:   for  $i \leftarrow 1, k$  do
4:     select case( $l$ )
5:       case(1)
6:          $S' \leftarrow \text{1-0 Relocate}(S)$ 
7:       case(2)
8:          $S' \leftarrow \text{2-Opt}(S)$ 
9:       case(3)
10:         $S' \leftarrow \text{1-1 Exchange}(S)$ 
11:     end select
12:   end for
13:   return  $S'$ 
14: end procedure

```

Διαταραχή_2 [148]. Η επιστημονική κοινότητα φαίνεται να τείνει να στρέφεται σε νέες μη συμβατικές υπολογιστικές μεθόδους. Συνολικά, ο μη συμβατικός υπολογισμός είναι ένα ευρύ φάσμα από προτεινόμενα νέα ή ασυνήθιστα υπολογιστικά μοντέλα. Μέρος αυτών των υπολογιστικών μοντέλων είναι ο φυσικός υπολογισμός (**Natural Computing**). Ο φυσικός υπολογισμός εμπνέεται από μοντέλα και φαινόμενα από την φύση και τον πραγματικό κόσμο. Αυτά εφαρμόζονται σε αλγόριθμους βελτιστοποίησης προσπαθώντας να οδηγήσουν την διαδικασία σε εύρεση καλύτερης λύσης [141]. Ο υπολογισμός που εμπνέεται από τη φύση έχει αναδειχθεί ως ένα αποτελεσματικό παράδειγμα για το σχεδιασμό και τη προσομοίωση καινοτόμων υπολογιστικών μοντέλων εμπνευσμένων από φυσικά φαινόμενα για την επίλυση σύνθετων μη γραμμικών, δυναμικών ή και ειδικών προβλημάτων. Μερικά από τα γνωστά υπολογιστικά συστήματα και αλγόριθμους εμπνευσμένους από τη φύση είναι [46]:

- i. Εξελικτικοί, βιολογικά εμπνευσμένοι αλγόριθμοι.
- ii. Αλγόριθμοι νοημοσύνης Σμήνους εμπνευσμένοι από τη συμπεριφορά του συνόλου του σμήνους / πρακτόρων.
- iii. Κοινωνικοί και πολιτιστικοί αλγόριθμοι εμπνευσμένοι από τις αλληλεπιδράσεις και τις πεποιθήσεις της κοινωνίας.
- iv. Εμπνευσμένοι από την χβαντική φυσική, δηλαδή χβαντικά εμπνευσμένοι αλγόριθμοι.

Ο νέο προτεινόμενος αλγόριθμος από την μεριά μας, που περιγράφεται αναλυτικά στην δημοσίευση [148] στηρίζεται στην τελευταία κατηγορία της λίστας και είναι εμπνευσμένος από τις χβαντικές αρχές.

Οι χβαντικά εμπνευσμένες μέθοδοι μιμούνται τις βασικές αρχές του χβαντικού υπολογισμού. Η χβαντική υπολογιστική, είναι ένα υποσύνολο του φυσικού υπολογισμού και είναι ένα πεδίο που παρουσιάστηκε πρόσφατα από τον Feynman (1980). Ο Feynman συνειδητοποίησε ότι μια

αποτελεσματική προσομοίωση ενός πραγματικού κβαντικού συστήματος χρησιμοποιώντας έναν τυπικό υπολογιστή δεν είναι δυνατή επειδή η προσομοίωση των πραγματικών κβαντικών διεργασιών θα επιβραδυνόταν εκθετικά [57, 58].

Ο κβαντικός υπολογισμός είναι μια σημαντική προσθήκη στα υπάρχοντα πρότυπα μοντέλα υπολογισμού. Στην ουσία είναι μια γενική ιδέα που θεωρεί την υπολογιστική διαδικασία ως κβαντικό φαινόμενο. Ο κβαντικός υπολογισμός συνδυάζει, εκτός από την επιστήμη των υπολογιστών, τους ορισμούς, τα θεωρητικά μαθηματικά και τη φυσική. Τα μαθηματικά, όπως η γραμμική άλγεβρα, και η φυσική, όπως η κβαντική μηχανική, εμπλέκονται κυρίως.

Το *qubit* είναι το κβαντικό ανάλογο του κλασσικού *bit*. Παρομοίως, ο κβαντικός καταχωρητής, που είναι μια συλλογή από *qubits*, είναι το κβαντικό ανάλογο του κλασσικού καταχωρητή. Σε κάθε κλήση αυτής της μεθόδου διαταραχής, ένας προσομοιωμένος *n-qubit* κβαντικός καταχωρητής δημιουργεί ένα κανονικοποιημένο μιγαδικό *n*-διαστάσεων μοναδιαίο διάνυσμα. Σε αυτό το πλαίσιο, με τον όρο 'κανονικοποιημένο' εννοούμε ότι εάν $(z_1, \dots, z_n)$ είναι το μιγαδικό διάνυσμα, τότε $|z_1|^2 + \dots + |z_n|^2 = 1$. Η διάσταση *n* του μιγαδικού διανύσματος είναι ίση ή μεγαλύτερη με τη συνολική διάσταση του προβλήματος. Το μιγαδικό *n*-διαστάσεων διάνυσμα μετατρέπεται σε ένα πραγματικό *n*-διαστάσεων διάνυσμα, τα στοιχεία του οποίου είναι πραγματικοί αριθμοί μεταξύ $[0, 1]$. Αν z_i και r_i είναι τα *i*-ιστά στοιχεία του μιγαδικού και πραγματικού διανύσματος αντίστοιχα, τότε $r_i = |z_i|^2$, για παράδειγμα r_i είναι ίσο με το συντελεστή του τετραγώνου του z_i . Επίσης, καθένα από τα επιλεγμένα στοιχεία του πραγματικού διανύσματος αντιστοιχούν σε έναν κόμβο από τους συνολικούς κόμβους της τρέχουσας λύσης. Για κάθε κόμβο της τρέχουσας λύσης τα παραπάνω στοιχεία του διανύσματος, χρησιμοποιούνται για να 'μαρκάρουν' τον κόμβο. Η ταξινόμηση του πρώτου διανύσματος επηρεάζει τη σειρά του διανύσματος λύσης λόγω της αντιστοιχίας μεταξύ στοιχείων και κόμβων σε μια διαδρομή και έτσι οδηγεί την προσπάθεια εξερεύνησης του αλγορίθμου σε άλλο σημείο του χώρου αναζήτησης. Ο ψευδοκώδικας της συγκεκριμένης -κβαντικά εμπνευσμένης διαδικασίας διαταραχής δίνεται από τον αλγόριθμο 5.

Algorithm 5 Διαταραχή_2.

- 1: **procedure** SHAKE_2(S, n)
 - 2: $NQubits \leftarrow \text{QuantumRegister}(n)$
 - 3: Compute the components based on the qubits.
 - 4: Save the *n* components in the vector *QCompVector*.
 - 5: Matching each element in the *QCompVector* with a node in *S*.
 - 6: Descending sorting on *QCompVector* produces *S'*.
 - 7: Recalculate the cost of the new *S'*.
 - 8: **return** *S'*
 - 9: **end procedure**
-

Διαταραχή_3. Αυτή η μέθοδος διαταραχής είναι μια μέθοδος ανακατεύθυνσης, όπου σε κάθε επανάληψη οι πελάτες τοποθετούνται σε τυχαία σειρά. Η μέθοδος εμφανίζεται στον Αλγόριθμο 6.

Algorithm 6 Διαταραχή_3.

```

1: procedure SHAKE_3( $S$ )
2:    $S' \leftarrow \text{Shuffle}(S)$ 
3:   return  $S'$ 
4: end procedure

```

4.1.3 GVNS σχήματα που προκύπτουν από τις διαφορετικές μεθόδους διαταραχής

Για κάθε μέθοδο διαταραχής σχηματίζεται και ένα αντίστοιχο σχήμα Γενικής Αναζήτησης Μεταβαλλόμενης Γειτονιάς - GVNS. Συγκεκριμένα, η GVNS_1 περιέχει την Διαταραχή_1 ως μέθοδο διαταραχής, η GVNS_2 χρησιμοποιεί την Διαταραχή_2 για τη διαφοροποίηση των λύσεων και η GVNS_3 υιοθετεί τη μέθοδο διαταραχής Διαταραχή_3. Η αρχική λύση παράγεται από την ευρετική του Πλησιέστερου Γείτονα (Nearest Neighbor - NN) σε όλα τα σχήματα της GVNS. Ο ψευδοκώδικας για τις τρεις προσεγγίσεις GVNS δίνεται στους Αλγόριθμους 7–9, αντίστοιχα.

Algorithm 7 GVNS_1.

```

1: procedure GVNS_1( $S, k_{max}, max\_time, l_{max}$ )
2:   while  $time \leq max\_time$  do
3:     for  $k \leftarrow 1, k_{max}$  do
4:        $S^* = \text{Shake\_1}(S, k, l_{max})$ 
5:        $S' = pVND(S^*)$ 
6:       if  $f(S') < f(S)$  then
7:          $S \leftarrow S'$ 
8:       end if
9:     end for
10:  end while
11:  return  $S$ 
12: end procedure

```

Algorithm 8 GVNS_2.

```

1: procedure GVNS_2( $S, n, max\_time$ )
2:   while  $time \leq max\_time$  do
3:      $S^* = \text{Shake\_2}(S, n)$ 
4:      $S' = pVND(S^*)$ 
5:     if  $f(S') < f(S)$  then
6:        $S \leftarrow S'$ 
7:     end if
8:   end while
9:   return  $S$ 
10: end procedure

```

Algorithm 9 GVNS_3.

```

1: procedure GVNS_3( $S, max\_time$ )
2:   while  $time \leq max\_time$  do
3:      $S^* = \text{Shake\_3}(S)$ 
4:      $S' = pVND(S^*)$ 
5:     if  $f(S') < f(S)$  then
6:        $S \leftarrow S'$ 
7:     end if
8:   end while
9:   return  $S$ 
10: end procedure

```

Στο σημείο αυτό θα πρέπει να τονιστεί ότι και στα τρία GVNS σχήματα χρησιμοποιούνται και οι 2 διαφορετικές στρατηγικές αναζήτησης. Η αναζήτηση δηλαδή, πραγματοποιείται και κατά την εύρεση της πρώτης βελτιωμένης λύσης (**First Improvement**) και κατά την εύρεση της καλύτερης βελτιωμένης λύσης σε έναν πλήρη κύκλο αναζήτησης (**Best Improvement**).

4.2 Υπολογιστική και συγκριτική μελέτη

4.2.1 Υπολογιστικό περιβάλλον και ορισμός πειραματικών παραμέτρων

Οι προαναφερθείσες μέθοδοι υλοποιήθηκαν στη γλώσσα προγραμματισμού Fortran και εκτελέστηκαν σε υπολογιστή με Windows 64-bit σε επεξεργαστή Intel Core i7-6700 στα 2,6 GHz με 16 GB RAM. Η μεταγλώττιση του κώδικα έγινε χρησιμοποιώντας τον επεξεργαστή Intel Fortran 64 XE με την επιλογή βελτιστοποίησης / O3. Το μέγιστο χρονικό όριο εκτέλεσης των πειραμάτων ορίστηκε σε $max_time = 60$ δευτερόλεπτα και $max_time = 120$ δευτερόλεπτα και ο μέγιστος αριθμός των επαναλήψεων της διαταραχής για την μέθοδο διαταραχής 1 ορίστηκε μετά

από πειραματικό έλεγχο σε $k_{max} = 12$.

4.2.2 Υπολογιστικά Αποτελέσματα

Αυτή η ενότητα παρουσιάζει τα υπολογιστικά αποτελέσματα των διαφορετικών στρατηγικών διαταραχής για κάθε κλάση μετροπροβλημάτων. Τα GVNS σχήματα με τις διαφορετικές μεθόδους διαταραχής το καθένα, δοκιμάζονται στα μετροπροβλήματα της TSPLIB βιβλιοθήκης. Το TSP είναι ένα από τα πιο διάσημα προβλήματα συνδυαστικής βελτιστοποίησης και είναι NP-hard υπολογιστικής πολυπλοκότητας. Η επίλυση του TSP σημαίνει την εύρεση της συντομότερης διαδρομής, έτσι ώστε ο πωλητής να ξεκινά από έναν συγκεκριμένο κόμβο και να επιστρέφει σε αυτόν τον κόμβο αφού περάσει μία φορά από όλους τους άλλους κόμβους. Όλα τα πειράματα εκτελέστηκαν 5 φορές για κάθε μετροπρόβλημα (benchmark) και υπολογίσαμε στα τελικά αποτελέσματα την μέση τιμή από τις συνολικές εκτελέσεις.

Οι πίνακες 4.1 και 4.2 περιέχουν τα υπολογιστικά αποτελέσματα για το ασύμμετρο TSP. Πιο συγκεκριμένα, περιλαμβάνουν το όνομα του μετροπροβλήματος, την βέλτιστη τιμή για το κάθε μετροπρόβλημα, τα κόστη των τριών διαφορετικών GVNS σχημάτων (GVNS_1, GVNS_2 and GVNS_3) και οι αντίστοιχες διαφορές τους από την βέλτιστη τιμή. Τα αποτελέσματα που απεικονίζονται στον Πίνακα 4.1 ανακτήθηκαν χρησιμοποιώντας την στρατηγική αναζήτησης πρώτης βελτίωσης και χρόνο εκτέλεσης του κάθε πειράματος στο 1 λεπτό, ενώ τα αποτελέσματα στον Πίνακα 4.2 πραγματοποιήθηκαν χρησιμοποιώντας την στρατηγική αναζήτησης καλύτερης βελτίωσης και χρόνο εκτέλεσης του κάθε πειράματος επίσης 1 λεπτό. Όπως αναφέρθηκε και προηγουμένως, το υπολογιζόμενο κόστος του κάθε GVNS σχήματος είναι η μέση τιμή από 5 εκτελέσεις του αλγόριθμου για το ίδιο μετροπρόβλημα στον ίδιο χρόνο. Επίσης η αναφερόμενη διαφορά από τη βέλτιστη λύση υπολογίζεται ως εξής: Δεδομένου του αποτελέσματος x , η διαφορά του από την βέλτιστη τιμή OV δίνεται από τον τύπο $\frac{100 \times (x - OV)}{OV}$.

Table 4.1: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης πρώτης βελτίωσης και χρονικό όριο εκτέλεσης 1 λεπτό[147].

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
<i>br17.atsp</i>	39	39	39	39	0.00	0.00	0.00
<i>ft53.atsp</i>	6905	7189	7328	7737	4.11	6.13	12.05
<i>ft70.atsp</i>	38673	39782	40691	40537	2.87	5.22	4.82
<i>ftv33.atsp</i>	1286	1318	1339	1450	2.49	4.12	12.75
<i>ftv35.atsp</i>	1473	1484	1499	1596	0.75	1.77	8.35
<i>ftv38.atsp</i>	1530	1546	1585	1579	1.05	3.59	3.20
<i>ftv44.atsp</i>	1613	1651	1760	1797	2.36	9.11	11.41
<i>ftv47.atsp</i>	1778	1821	1992	2101	2.42	12.04	18.17
<i>ftv55.atsp</i>	1608	1666	1985	1912	3.61	23.45	18.91
<i>ftv64.atsp</i>	1839	1961	2382	2395	6.63	29.53	30.23
<i>ftv70.atsp</i>	1950	2136	2557	2484	9.54	31.13	27.38
<i>ftv170.atsp</i>	2755	3487	3923	3923	26.57	42.40	42.40
<i>kro124p.atsp</i>	36230	39024	43187	40259	7.71	19.20	11.12
<i>p43.atsp</i>	5620	5620	5623	5658	0.00	0.05	0.68
<i>rbg323.atsp</i>	1326	1516	1563	1626	14.32	17.87	22.62
<i>rbg358.atsp</i>	1163	1347	1437	1404	15.82	23.55	20.72
<i>rbg403.atsp</i>	2465	2535	2587	2565	9.78	4.42	11.76
<i>rbg443.atsp</i>	2720	2814	2859	2814	3.46	5.11	3.46
<i>ry48p.atsp</i>	14422	14549	14901	14738	0.88	3.32	2.19
Μέση Τιμή	6599.74	6920.26	7328.37	7190.21	12.33	12.64	24.10

Τα αποτελέσματα στον Πίνακα 4.1 υποδεικνύουν ένα συγκεκριμένο πρότυπο, δηλαδή ότι και οι δυο GVNS_1 and GVNS_2 παρουσιάζουν καλύτερες επιδόσεις σε σχέση με την GVNS_3 στις περισσότερες περιπτώσεις. Ανακαλούμε ότι η GVNS_3 είναι μια τυχαία στρατηγική διαταραχής. Για παράδειγμα, σκεφτείτε το μετροπρόβλημα *ftv47*; Μπορούμε να δούμε ότι το κόστος της GVNS_1 είναι 1821, το κόστος της GVNS_2 είναι 1992 και της GVNS_3 είναι 2101. Η GVNS_1 και η GVNS_2 ξεπερνούν την GVNS_3 και είναι και οι 2 παρόμοια κοντά στην βέλτιστη τιμή (1778).

Table 4.2: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης καλύτερης βελτίωσης και χρονικό όριο εκτέλεσης 1 λεπτό [147].

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
<i>br17.atsp</i>	39	39	39	39	0.00	0.00	0.00
<i>ft53.atsp</i>	6905	7043	7135	7674	2.00	3.33	11.14
<i>ft70.atsp</i>	38673	39507	40206	40539	2.16	3.96	4.83
<i>ftv33.atsp</i>	1286	1289	1286	1379	0.23	0.00	7.23
<i>ftv35.atsp</i>	1473	1476	1473	1533	0.20	0.00	4.07
<i>ftv38.atsp</i>	1530	1538	1541	1599	0.52	0.72	4.51
<i>ftv44.atsp</i>	1613	1632	1644	1728	1.18	1.92	7.13
<i>ftv47.atsp</i>	1778	1792	1816	1940	0.79	2.14	9.11
<i>ftv55.atsp</i>	1608	1642	1665	2012	2.11	3.54	25.12
<i>ftv64.atsp</i>	1839	1908	1986	2193	3.75	7.99	19.25
<i>ftv70.atsp</i>	1950	2110	2157	2346	8.21	10.62	20.31
<i>ftv170.atsp</i>	2755	3341	3852	3923	21.27	39.82	42.40
<i>kro124p.atsp</i>	36230	36501	37076	38195	0.75	2.34	5.42
<i>p43.atsp</i>	5620	5620	5620	5627	0.00	0.00	0.12
<i>rbg323.atsp</i>	1326	1486	1539	1633	12.06	16.06	23.15
<i>rbg358.atsp</i>	1163	1307	1409	1437	12.38	21.15	23.55
<i>rbg403.atsp</i>	2465	2510	2547	2554	11.76	11.76	11.76
<i>rbg443.atsp</i>	2720	2765	2824	2844	1.65	3.16	4.56
<i>ry48p.atsp</i>	14422	14480	14498	14659	0.40	0.12	1.64
Μέση Τιμή	6599.74	6736.11	6858.58	7044.95	15.88	17.69	22.28

Επιπλέον, τα παρεχόμενα αποτελέσματα στον Πίνακα 4.2 οδηγούν στην ίδια διαπίστωση ότι τόσο η GVNS_1 και η GVNS_2 παράγουν καλύτερα αποτελέσματα σε σχέση με την GVNS_3 στις περισσότερες των περιπτώσεων. Ο Πίνακας 4.3 παρουσιάζει τα υπολογιστικά αποτελέσματα των GVNS σχημάτων για εκτελέσεις των δύο λεπτών για το κάθε πείραμα και με την στρατηγική αναζήτησης πρώτης βελτίωσης. Τα αποτελέσματα του Πίνακα 4.3 τονίζουν ότι η GVNS_1 ξεπερνά την GVNS_2 και την GVNS_3 στις περισσότερες περιπτώσεις. Πάραυτα η μόνη διαφορά σε σχέση με τα αποτελέσματα των Πινάκων 4.1 και 4.2 είναι ότι στην προκειμένη η συμπεριφορά της GVNS_2 είναι πιο κοντά σε αυτήν της GVNS_3 προσέγγισης της λύσης. Ο Πίνακας 4.4 παρουσιάζει τα υπολογιστικά αποτελέσματα που επιτεύχθηκαν από τα GVNS σχήματα με 2 λεπτά χρονικό όριο στην εκτέλεση των πειραμάτων και σαν στρατηγική αναζήτησης αυτήν της καλύτερης βελτίωσης. Τα αποτελέσματα του Πίνακα 4.4 σε συνδυασμό με τα συμπεράσματα που προκύπτουν από τους Πίνακες 4.1 και 4.2 είναι ότι και οι δύο οι GVNS, GVNS_1 and GVNS_2 ξεπερνούν σε απόδοση την GVNS_3 στις περισσότερες περιπτώσεις.

Table 4.3: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης πρώτης βελτίωσης και χρονικό όριο εκτέλεσης τα 2 λεπτά [147].

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
<i>br17.atsp</i>	39	39	39	39	0.00	0.00	0.00
<i>ft53.atsp</i>	6905	7024	7498	7752	1.72	8.59	12.27
<i>ft70.atsp</i>	38673	39615	40827	40505	2.44	5.57	4.74
<i>ftv33.atsp</i>	1286	1330	1370	1454	3.42	6.53	13.06
<i>ftv35.atsp</i>	1473	1482	1519	1604	0.61	3.12	8.89
<i>ftv38.atsp</i>	1530	1547	1618	1576	1.11	5.75	3.01
<i>ftv44.atsp</i>	1613	1628	1839	1812	0.93	14.01	12.34
<i>ftv47.atsp</i>	1778	1787	2020	2097	0.51	13.61	17.94
<i>ftv55.atsp</i>	1608	1668	2012	1912	3.73	25.12	18.91
<i>ftv64.atsp</i>	1839	1951	2484	2476	6.09	35.07	34.64
<i>ftv70.atsp</i>	1950	2165	2571	2484	11.03	31.85	27.38
<i>ftv170.atsp</i>	2755	3412	3923	3923	23.85	42.40	42.40
<i>kro124p.atsp</i>	36230	39344	44243	40849	8.60	22.12	12.75
<i>p43.atsp</i>	5620	5620	5628	5657	0.00	0.14	0.66
<i>rbg323.atsp</i>	1326	1499	1576	1586	13.04	18.85	19.60
<i>rbg358.atsp</i>	1163	1329	1410	1406	14.27	21.23	20.89
<i>rbg403.atsp</i>	2465	2509	2586	2547	2.27	4.10	11.76
<i>rbg443.atsp</i>	2720	2808	2849	2811	3.24	4.74	3.35
<i>ry48p.atsp</i>	14422	14475	14936	14708	0.37	3.56	1.98
Μέση Τιμή	6599.74	6906.95	7418.32	7220.95	5.29	13.96	24.78

Table 4.4: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης καλύτερης βελτίωσης και χρονικό όριο εκτέλεσης τα 2 λεπτά [147].

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
<i>br17.atsp</i>	39	39	39	39	0.00	0.00	0.00
<i>ft53.atsp</i>	6905	7043	7207	7773	2.00	4.37	12.57
<i>ft70.atsp</i>	38673	39358	40230	40588	1.77	4.03	4.95
<i>ftv33.atsp</i>	1286	1286	1290	1370	0.00	0.31	6.53
<i>ftv35.atsp</i>	1473	1474	1475	1509	0.07	0.14	2.44
<i>ftv38.atsp</i>	1530	1538	1555	1599	0.52	1.63	4.51
<i>ftv44.atsp</i>	1613	1636	1664	1731	1.43	3.16	7.32
<i>ftv47.atsp</i>	1778	1787	1837	1903	0.51	3.32	7.03
<i>ftv55.atsp</i>	1608	1640	1686	2012	1.99	4.85	25.12
<i>ftv64.atsp</i>	1839	1914	2032	2217	4.08	10.49	20.55
<i>ftv70.atsp</i>	1950	2038	2189	2342	4.51	12.26	20.10
<i>ftv170.atsp</i>	2755	3351	3918	3923	21.63	42.21	42.40
<i>kro124p.atsp</i>	36230	36379	37378	37915	0.41	3.17	4.65
<i>p43.atsp</i>	5620	5620	5620	5625	0.00	0.00	0.09
<i>rbg323.atsp</i>	1326	1473	1531	1610	11.08	15.46	21.41
<i>rbg358.atsp</i>	1163	1292	1405	1435	11.09	20.80	23.38
<i>rbg403.atsp</i>	2465	2498	2547	2553	1.30	3.25	11.76
<i>rbg443.atsp</i>	2720	2771	2822	2842	1.88	3.75	4.49
<i>ry48p.atsp</i>	14422	14468	14464	14678	0.32	0.29	1.78
Μέση Τιμή	6599.74	6716.05	6888.89	7034.95	3.28	7.05	22.16

Οι Πίνακες 4.5–4.8 περιέχουν τα αντίστοιχα υπολογιστικά αποτελέσματα για τα μετροπροβλήματα του συμμετρικού TSP. Ειδικότερα, περιέχουν το όνομα του μετροπροβλήματος, την βέλτιστη τιμή (zOpt), το αντίστοιχο κόστος για καθεμία από τις τρεις διαφορετικές GVNS παραλλαγές (GVNS_1, GVNS_2 and GVNS_3) και τις αντίστοιχες διαφορές των GVNS σχημάτων από τη βέλτιστη τιμή. Ο πίνακας 4.5 απεικονίζει την GVNS χρησιμοποιώντας την στρατηγική αναζήτησης πρώτης βελτίωσης και για χρόνο εκτέλεσης ενός λεπτού. Ο Πίνακας 4.6 δείχνει την GVNS χρησιμοποιώντας την στρατηγική αναζήτησης καλύτερης βελτίωσης και με χρόνο εκτέλεσης του κάθε πειράματος το 1 λεπτό. Τα πειράματα των Πινάκων 4.7 και 4.8 εκτελέστηκαν με χρονικό περιορισμό τα 2 λεπτά σε πρώτη και καλύτερης βελτίωσης στρατηγική αναζήτησης αντίστοιχα. Τα παρακάτω υπολογιστικά αποτελέσματα δείχνουν ότι η GVNS_3 παράγει καλύτερες λύσεις σε σχέση με τις άλλες δύο παραλλαγές της GVNS, όπως επίσης και ότι οι GVNS_1 and GVNS_2 αλγόριθμοι δεν έχουν αξιοσημείωτες διαφορές μεταξύ τους.

Table 4.5: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης πρώτης βελτίωσης και χρονικό όριο εκτέλεσης 1 λεπτό

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
a280.tsp	2579	2683	2739	2745	4.03	6.20	6.44
att48.tsp	10628	10628	10628	10635	0.00	0.00	0.07
bayg29.tsp	1610	1610	1610	1610	0.00	0.00	0.00
bays29.tsp	2020	2020	2020	2020	0.00	0.00	0.00
bier127.tsp	118282	118636	120066	119966	0.30	1.51	1.42

Συνεχίζεται στην επόμενη σελίδα

Table 4.5 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
kroA100.tsp	21282	21296	21375	21398	0.07	0.44	0.55
burma14.tsp	3323	3323	3323	3323	0.00	0.00	0.00
ch130.tsp	6110	6156	6239	6235	0.75	2.11	2.05
ch150.tsp	6528	6583	6720	6723	0.84	2.94	2.99
d493.tsp	35002	36928	37307	37166	5.50	6.59	6.18
kroB100.tsp	22141	22187	22339	22362	0.21	0.89	1.00
kroC100.tsp	20749	20759	20864	20834	0.05	0.55	0.41
kroD100.tsp	21294	21370	21573	21667	0.36	1.31	1.75
kroE100.tsp	22068	22114	22291	22360	0.21	1.01	1.32
kroA150.tsp	26524	26792	27247	27206	1.01	2.73	2.57
kroB150.tsp	26130	26382	26680	26767	0.96	2.10	2.44
kroA200.tsp	29368	29753	30420	30392	1.31	3.58	3.49
kroB200.tsp	29437	30164	30727	30711	2.47	4.38	4.33
d198.tsp	15780	15908	16116	16147	0.81	2.13	2.33
brg180.tsp	1950	1960	2024	2040	0.51	3.79	4.62
berlin52.tsp	7542	7542	7542	7542	0.00	0.00	0.00
dantzig42.tsp	699	699	699	699	0.00	0.00	0.00
eil51.tsp	426	426	426	428	0.00	0.00	0.47
eil76.tsp	538	539	544	545	0.19	1.12	1.30
eil101.tsp	629	630	645	647	0.16	2.54	2.86
fri26.tsp	937	937	937	937	0.00	0.00	0.00
gil262.tsp	2378	2460	2509	2509	3.45	5.51	5.51
gr17.tsp	2085	2085	2085	2085	0.00	0.00	0.00
gr21.tsp	2707	2707	2707	2707	0.00	0.00	0.00
gr24.tsp	1272	1272	1272	1272	0.00	0.00	0.00
gr48.tsp	5046	5046	5046	5048	0.00	0.00	0.04
gr96.tsp	55209	55285	55635	55713	0.14	0.77	0.91
gr120.tsp	6942	6979	7085	7103	0.53	2.06	2.32
gr137.tsp	69853	70207	71158	71330	0.51	1.87	2.11
gr202.tsp	40160	41232	41752	41850	2.67	3.96	4.21
gr229.tsp	134602	137642	139570	140144	2.26	3.69	4.12
gr431.tsp	171414	179950	182365	182884	4.98	6.39	6.69
hk48.tsp	11461	11461	11461	11470	0.00	0.00	0.08
lin105.tsp	14379	14386	14433	14458	0.05	0.38	0.55
lin318.tsp	42029	43641	44175	44183	3.84	5.11	5.13
pcb442.tsp	50778	53301	54176	54486	4.97	6.69	7.30
pr76.tsp	108159	108168	108411	108621	0.01	0.23	0.43
pr107.tsp	44303	44428	44695	44708	0.28	0.88	0.91
pr124.tsp	59030	59045	59169	59222	0.03	0.24	0.33
pr136.tsp	96772	97875	99118	99300	1.14	2.42	2.61
pr144.tsp	58537	58538	58629	58627	0.00	0.16	0.15
pr152.tsp	73682	74016	74379	74299	0.45	0.95	0.84
pr226.tsp	80369	80605	81007	81267	0.29	0.79	1.12
pr264.tsp	49135	50237	50883	50847	2.24	3.56	3.48
pr299.tsp	48191	50331	50917	50883	4.44	5.66	5.59
pr439.tsp	107217	112633	114121	114191	5.05	6.44	6.50
rat99.tsp	1211	1215	1240	1243	0.33	2.39	2.64
rat195.tsp	2323	2371	2456	2453	2.07	5.73	5.60
rd100.tsp	7910	7925	8011	8051	0.19	1.28	1.78

Συνεχίζεται στην επόμενη σελίδα

Table 4.5 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
rd400.tsp	15281	15951	16281	16269	4.38	6.54	6.47
si175.tsp	21407	21430	21482	21486	0.11	0.35	0.37
st70.tsp	675	677	675	677	0.30	0.00	0.30
swiss42.tsp	1273	1273	1273	1273	0.00	0.00	0.00
ts225.tsp	126643	126858	128223	128359	0.17	1.25	1.35
tsp225.tsp	3916	4014	4116	4122	2.50	5.11	5.26
u159.tsp	42080	42556	43373	43374	1.13	3.07	3.08
ulysses16.tsp	6859	6859	6859	6859	0.00	0.00	0.00
ulysses22.tsp	7013	7013	7013	7013	0.00	0.00	0.00
ali535.tsp	202339	218486	221688	217544	7.98	9.56	7.51
att532.tsp	27686	29351	29747	28818	6.01	7.44	4.09
brazil58.tsp	25395	33181	25395	25396	30.66	0.00	0.00
brg180.tsp	1950	1959	2040	2158	0.46	4.62	10.67
d657.tsp	48912	52126	53015	51059	6.57	8.39	4.39
d1291.tsp	50801	59103	60214	55243	16.34	18.53	8.74
d1655.tsp	62128	73791	74028	73982	18.77	19.15	19.08
d2103.tsp	80450	86653	86653	86653	7.71	7.71	7.71
dsj1000.tsp	18659688	24056781	24631467	20034159	20.28	23.16	0.17
fl417.tsp	11861	12366	12232	12227	4.26	3.13	3.09
fl1400.tsp	20127	27242	27447	25980	35.35	36.37	29.08
fl1577.tsp	22249	27941	27996	27996	25.58	25.83	25.83
fl3795.tsp	28772	35262	35285	35285	22.56	22.64	22.64
fnl4461.tsp	182566	229963	229963	229963	25.96	25.96	25.96
gr666.tsp	294358	317446	324339	309556	7.84	10.19	5.16
nrv1379.tsp	56638	67679	68964	61769	19.49	21.76	9.06
p654.tsp	34643	36502	36558	35569	5.37	5.53	2.67
pa561.tsp	2763	2928	17798	3003	5.97	544.15	8.69
pcb1173.tsp	56892	70520	71978	61273	23.95	26.52	7.70
pcb3038.tsp	137694	175926	176310	176310	27.77	28.04	28.04
pr1002.tsp	259045	323543	331103	277196	24.90	27.82	7.01
pr2392.tsp	378032	460547	461170	461170	21.83	21.99	21.99
rat575.tsp	6773	7179	18258	7153	5.99	169.57	5.61
rat783.tsp	8806	9634	14701	9341	9.40	66.94	6.08
rl1304.tsp	252948	330540	335779	277603	30.68	32.75	9.75
rl1323.tsp	270199	331586	332103	293133	22.72	22.91	8.49
rl1889.tsp	316536	388695	389270	389270	22.80	22.98	22.98
rl5915.tsp	565530	695602	695602	695602	23.00	23.00	23.00
rl5934.tsp	556045	672412	672412	672412	20.93	20.93	20.93
si535.tsp	48450	48697	48848	48807	0.51	0.82	0.74
si1032.tsp	92650	92883	94571	92909	0.25	2.07	0.28
u574.tsp	36905	40206	40020	39488	8.94	8.44	7.00
u724.tsp	41910	45583	45988	44646	8.76	9.73	6.53
u1060.tsp	224094	297757	308980	242181	32.87	37.88	8.07
u1432.tsp	152970	185839	188807	166714	21.49	23.43	8.98
u1817.tsp	57201	71999	72030	72030	25.87	25.92	25.92
u2152.tsp	64253	78870	79260	79260	22.75	23.36	23.36
u2319.tsp	234256	275453	278765	278765	17.59	19.00	19.00
vm1084.tsp	239297	295088	301477	258248	23.31	25.98	7.92

Συνεχίζεται στην επόμενη σελίδα

Table 4.5 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
vm1748.tsp	336556	406536	408102	408102	20.79	21.26	21.26
Μέση Τιμή	266956.86	317607.30	324186.64	276033.63	7.31	15.39	6.03

Table 4.6: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης καλύτερης βελτίωσης και χρονικό όριο εκτέλεσης 1 λεπτό

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
a280.tsp	2579	2632	2738	2734	2.06	6.17	6.01
att48.tsp	10628	10628	10628	10631	0.00	0.00	0.03
bayg29.tsp	1610	1610	1610	1610	0.00	0.00	0.00
bays29.tsp	2020	2020	2020	2020	0.00	0.00	0.00
bier127.tsp	118282	118411	119593	119962	0.11	1.11	1.42
kroA100.tsp	21282	21282	21332	21413	0.00	0.23	0.62
burma14.tsp	3323	3323	3323	3323	0.00	0.00	0.00
ch130.tsp	6110	6147	6219	6237	0.61	1.78	2.08
ch150.tsp	6528	6571	6704	6725	0.66	2.70	3.02
d493.tsp	35002	36559	37182	37119	4.45	6.23	6.05
kroB100.tsp	22141	22162	22282	22362	0.09	0.64	1.00
kroC100.tsp	20749	20749	20837	20880	0.00	0.42	0.63
kroD100.tsp	21294	21346	21507	21581	0.24	1.00	1.35
kroE100.tsp	22068	22129	22241	22264	0.28	0.78	0.89
kroA150.tsp	26524	26696	27169	27220	0.65	2.43	2.62
kroB150.tsp	26130	26233	26631	26696	0.39	1.92	2.17
kroA200.tsp	29368	29549	30416	30381	0.62	3.57	3.45
kroB200.tsp	29437	29888	30618	30688	1.53	4.01	4.25
d198.tsp	15780	15845	16077	16090	0.41	1.88	1.96
brg180.tsp	1950	1963	2024	2035	0.67	3.79	4.36
berlin52.tsp	7542	7542	7542	7563	0.00	0.00	0.28
dantzig42.tsp	699	699	699	699	0.00	0.00	0.00
eil51.tsp	426	426	426	428	0.00	0.00	0.47
eil76.tsp	538	538	544	547	0.00	1.12	1.67
eil101.tsp	629	630	645	648	0.16	2.54	3.02
fri26.tsp	937	937	937	937	0.00	0.00	0.00
gil262.tsp	2378	2437	2515	2518	2.48	5.76	5.89
gr17.tsp	2085	2085	2085	2085	0.00	0.00	0.00
gr21.tsp	2707	2707	2707	2707	0.00	0.00	0.00
gr24.tsp	1272	1272	1272	1272	0.00	0.00	0.00
gr48.tsp	5046	5046	5046	5049	0.00	0.00	0.06
gr96.tsp	55209	55293	55521	55582	0.15	0.57	0.68
gr120.tsp	6942	6977	7085	7120	0.50	2.06	2.56
gr137.tsp	69853	69948	70964	71412	0.14	1.59	2.23
gr202.tsp	40160	41079	41693	41720	2.29	3.82	3.88
gr229.tsp	134602	136416	139729	140377	1.35	3.81	4.29
gr431.tsp	171414	177946	181693	182205	3.81	6.00	6.30
hk48.tsp	11461	11461	11461	11471	0.00	0.00	0.09
lin105.tsp	14379	14382	14396	14437	0.02	0.12	0.40
lin318.tsp	42029	43094	44070	44309	2.53	4.86	5.42
pcb442.tsp	50778	52584	54456	54581	3.56	7.24	7.49

Συνεχίζεται στην επόμενη σελίδα

Table 4.6 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
pr76.tsp	108159	108159	108278	108513	0.00	0.11	0.33
pr107.tsp	44303	44396	44539	44607	0.21	0.53	0.69
pr124.tsp	59030	59030	59058	59081	0.00	0.05	0.09
pr136.tsp	96772	97262	98966	98879	0.51	2.27	2.18
pr144.tsp	58537	58537	58561	58561	0.00	0.04	0.04
pr152.tsp	73682	73781	74027	73966	0.13	0.47	0.39
pr226.tsp	80369	80462	80861	80834	0.12	0.61	0.58
pr264.tsp	49135	49670	50905	50886	1.09	3.60	3.56
pr299.tsp	48191	49245	50614	50646	2.19	5.03	5.09
pr439.tsp	107217	111621	113347	113038	4.11	5.72	5.43
rat99.tsp	1211	1213	1234	1240	0.17	1.90	2.39
rat195.tsp	2323	2356	2451	2457	1.42	5.51	5.77
rd100.tsp	7910	7927	7963	8022	0.21	0.67	1.42
rd400.tsp	15281	15802	16312	16311	3.41	6.75	6.74
si175.tsp	21407	21420	21472	21476	0.06	0.30	0.32
st70.tsp	675	676	675	676	0.15	0.00	0.15
swiss42.tsp	1273	1273	1273	1273	0.00	0.00	0.00
ts225.tsp	126643	126721	127690	127849	0.06	0.83	0.95
tsp225.tsp	3916	3987	4115	4119	1.81	5.08	5.18
u159.tsp	42080	42329	42969	43024	0.59	2.11	2.24
ulysses16.tsp	6859	6859	6859	6859	0.00	0.00	0.00
ulysses22.tsp	7013	7013	7013	7013	0.00	0.00	0.00
ali535.tsp	202339	213387	218429	218205	5.46	7.95	7.84
att532.tsp	27686	28764	29614	29525	3.89	6.96	6.64
brazil58.tsp	25395	33181	25395	25412	30.66	0.00	0.07
brg180.tsp	1950	1963	2019	2198	0.67	3.54	12.72
d657.tsp	48912	51497	52986	51934	5.29	8.33	6.18
d1291.tsp	50801	54927	57302	55431	8.12	12.80	9.11
d1655.tsp	62128	67236	70399	67683	8.22	13.31	8.94
d2103.tsp	80450	83240	86653	83486	3.47	7.71	3.77
dsj1000.tsp	18659688	20201248	20449409	20063920	8.26	9.59	7.53
fl417.tsp	11861	12023	12119	12161	1.37	2.18	2.53
fl1400.tsp	20127	21244	21198	21166	5.55	5.32	5.16
fl1577.tsp	22249	23721	24355	23736	6.62	9.47	6.68
fl3795.tsp	28772	34663	33535	35214	20.47	16.55	22.39
fnl4461.tsp	182566	217998	204703	199441	19.41	12.13	9.24
gr666.tsp	294358	313338	321656	315223	6.45	9.27	7.09
nwr1379.tsp	56638	60516	62459	60983	6.85	10.28	7.67
p654.tsp	34643	36083	35544	36741	4.16	2.60	6.06
pa561.tsp	2763	2893	2896	3058	4.71	4.81	10.68
pcb1173.tsp	56892	61883	63335	62161	8.77	11.32	9.26
pcb3038.tsp	137694	153475	156692	149788	11.46	13.80	8.78
pr1002.tsp	259045	278408	285203	279922	7.47	10.10	8.06
pr2392.tsp	378032	410784	430379	408360	8.66	13.85	8.02
rat575.tsp	6773	7195	7090	7224	6.23	4.68	6.66
rat783.tsp	8806	9373	9391	9391	6.44	6.64	6.64
rl1304.tsp	252948	282487	282839	274566	11.68	11.82	8.55
rl1323.tsp	270199	293350	300601	285668	8.57	11.25	5.73
rl1889.tsp	316536	344218	356697	342893	8.75	12.69	8.33

Συνεχίζεται στην επόμενη σελίδα

Table 4.6 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
rl5915.tsp	565530	680825	695602	695602	20.39	23.00	23.00
rl5934.tsp	556045	664895	672412	661012	19.58	20.93	18.88
si535.tsp	48450	48622	48783	48847	0.36	0.69	0.82
si1032.tsp	92650	92918	93397	92908	0.29	0.81	0.28
u574.tsp	36905	40374	40022	39792	9.40	8.45	7.82
u724.tsp	41910	44662	45765	45273	6.57	9.20	8.02
u1060.tsp	224094	242630	246175	243291	8.27	9.85	8.57
u1432.tsp	152970	165304	170578	165833	8.06	11.51	8.41
u1817.tsp	57201	62782	66243	62050	9.76	15.81	8.48
u2152.tsp	64253	70205	74581	70787	9.26	16.07	10.17
u2319.tsp	234256	243928	249738	245475	4.13	6.61	4.79
vm1084.tsp	239297	256431	262955	255369	7.16	9.89	6.72
vm1748.tsp	336556	362026	373926	360551	7.57	11.10	7.13
Μέση τιμή	253944.13	274974.75	278630.04	273507.26	13.05	4.88	4.40

Table 4.7: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης πρώτης βελτίωσης και χρονικό όριο εκτέλεσης τα 2 λεπτά

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
a280.tsp	2579	2672	2728	2706	3.61	5.78	4.92
att48.tsp	10628	10628	10628	10724	0.00	0.00	0.90
bayg29.tsp	1610	1610	1610	1615	0.00	0.00	0.31
bays29.tsp	2020	2020	2020	2028	0.00	0.00	0.40
bier127.tsp	118282	118518	119737	119860	0.20	1.23	1.33
kroA100.tsp	21282	21283	21337	21699	0.00	0.26	1.96
burma14.tsp	3323	3323	3323	3323	0.00	0.00	0.00
ch130.tsp	6110	6157	6218	6289	0.77	1.77	2.93
ch150.tsp	6528	6583	6680	6649	0.84	2.33	1.85
d493.tsp	35002	36659	37040	36557	4.73	5.82	4.44
kroB100.tsp	22141	22168	22317	22557	0.12	0.79	1.88
kroC100.tsp	20749	20757	20806	21321	0.04	0.27	2.76
kroD100.tsp	21294	21329	21567	21904	0.16	1.28	2.86
kroE100.tsp	22068	22144	22250	22407	0.34	0.82	1.54
kroA150.tsp	26524	26630	27050	27738	0.40	1.98	4.58
kroB150.tsp	26130	26232	26701	26742	0.39	2.19	2.34
kroA200.tsp	29368	29570	30177	29718	0.69	2.75	1.19
kroB200.tsp	29437	29801	30459	30205	1.24	3.47	2.61
d198.tsp	15780	15871	16077	15964	0.58	1.88	1.17
brg180.tsp	1950	1956	2026	2153	0.31	3.90	10.41
berlin52.tsp	7542	7542	7542	7591	0.00	0.00	0.65
dantzig42.tsp	699	699	699	706	0.00	0.00	1.00
eil51.tsp	426	426	426	430	0.00	0.00	0.94
eil76.tsp	538	538	543	544	0.00	0.93	1.12
eil101.tsp	629	629	642	649	0.00	2.07	3.18
fri26.tsp	937	937	937	948	0.00	0.00	1.17
gil262.tsp	2378	2444	2505	2542	2.78	5.34	6.90
gr17.tsp	2085	2085	2085	2085	0.00	0.00	0.00
gr21.tsp	2707	2707	2707	2707	0.00	0.00	0.00

Συνεχίζεται στην επόμενη σελίδα

Table 4.7 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
gr24.tsp	1272	1272	1272	1272	0.00	0.00	0.00
gr48.tsp	5046	5046	5046	5063	0.00	0.00	0.34
gr96.tsp	55209	55247	55549	56109	0.07	0.62	1.63
gr120.tsp	6942	6960	7063	7103	0.26	1.74	2.32
gr137.tsp	69853	70005	71002	71579	0.22	1.64	2.47
gr202.tsp	40160	40973	41598	41672	2.02	3.58	3.76
gr229.tsp	134602	136884	139544	138929	1.70	3.67	3.21
gr431.tsp	171414	178310	181839	180800	4.02	6.08	5.48
hk48.tsp	11461	11461	11461	11664	0.00	0.00	1.77
lin105.tsp	14379	14394	14413	14672	0.10	0.24	2.04
lin318.tsp	42029	43463	44016	44024	3.41	4.73	4.75
pcb442.tsp	50778	52867	53957	52362	4.11	6.26	3.12
pr76.tsp	108159	108159	108373	109446	0.00	0.20	1.19
pr107.tsp	44303	44400	44687	44769	0.22	0.87	1.05
pr124.tsp	59030	59033	59108	59444	0.01	0.13	0.70
pr136.tsp	96772	97158	98762	101877	0.40	2.06	5.28
pr144.tsp	58537	58537	58605	60768	0.00	0.12	3.81
pr152.tsp	73682	73801	74185	75515	0.16	0.68	2.49
pr226.tsp	80369	80533	80840	81472	0.20	0.59	1.37
pr264.tsp	49135	49633	50391	50951	1.01	2.56	3.70
pr299.tsp	48191	49595	50642	50023	2.91	5.09	3.80
pr439.tsp	107217	111921	113644	113269	4.39	5.99	5.64
rat99.tsp	1211	1212	1236	1266	0.08	2.06	4.54
rat195.tsp	2323	2365	2448	2404	1.81	5.38	3.49
rd100.tsp	7910	7925	7998	8122	0.19	1.11	2.68
rd400.tsp	15281	15847	16203	15936	3.70	6.03	4.29
si175.tsp	21407	21428	21475	21514	0.10	0.32	0.50
st70.tsp	675	676	675	691	0.15	0.00	2.37
swiss42.tsp	1273	1273	1273	1273	0.00	0.00	0.00
ts225.tsp	126643	126815	128183	127374	0.14	1.22	0.58
tsp225.tsp	3916	3996	4101	4093	2.04	4.72	4.52
u159.tsp	42080	42341	43196	44203	0.62	2.65	5.05
ulysses16.tsp	6859	6859	6859	6860	0.00	0.00	0.01
ulysses22.tsp	7013	7013	7013	7013	0.00	0.00	0.00
ali535.tsp	202339	217245	219075	216606	7.37	8.27	7.05
att532.tsp	27686	29167	29615	28916	5.35	6.97	4.44
brazil58.tsp	25395	36518	25395	25395	43.80	0.00	0.00
brg180.tsp	1950	1956	2023	2165	0.31	3.74	11.03
d657.tsp	48912	51769	52769	50997	5.84	7.89	4.26
d1291.tsp	50801	56007	60214	55060	10.25	18.53	8.38
d1655.tsp	62128	73181	74028	66384	17.79	19.15	6.85
d2103.tsp	80450	86582	86653	83454	7.62	7.71	3.73
dsj1000.tsp	18659688	22538673	24631467	19905772	20.79	32.00	6.68
fl417.tsp	11861	12118	12233	12190	2.17	3.14	2.77
fl1400.tsp	20127	27026	27447	21220	34.28	36.37	5.43
fl1577.tsp	22249	27461	27996	23500	23.43	25.83	5.62
fl3795.tsp	28772	35731	35285	35285	24.19	22.64	22.64
fnl4461.tsp	182566	229761	229963	229963	25.85	25.96	25.96
gr666.tsp	294358	316182	321701	308908	7.41	9.29	4.94

Συνεχίζεται στην επόμενη σελίδα

Table 4.7 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
nrv1379.tsp	56638	67094	68964	60193	18.46	21.76	6.28
p654.tsp	34643	36022	36019	35708	3.98	3.97	3.07
pa561.tsp	2763	2905	15680	3008	5.14	467.50	8.87
pcb1173.tsp	56892	65023	70731	61249	14.29	24.33	7.66
pcb3038.tsp	137694	175799	176310	176310	27.67	28.04	28.04
pr1002.tsp	259045	279694	296142	275098	7.97	14.32	6.20
pr2392.tsp	378032	459687	461170	461170	21.60	21.99	21.99
rat575.tsp	6773	7136	20069	7158	5.36	196.31	5.68
rat783.tsp	8806	9501	19613	9352	7.89	122.72	6.20
rl1304.tsp	252948	322802	335779	274942	27.62	32.75	8.70
rl1323.tsp	270199	316844	332103	291235	17.26	22.91	7.79
rl1889.tsp	316536	388400	389270	389270	22.70	22.98	22.98
rl5915.tsp	565530	695466	695602	695602	22.98	23.00	23.00
rl5934.tsp	556045	672290	672412	672412	20.91	20.93	20.93
si535.tsp	48450	48648	48803	48765	0.41	0.73	0.65
si1032.tsp	92650	92864	93285	92925	0.23	0.69	0.30
u574.tsp	36905	40248	39803	39467	9.06	7.85	6.94
u724.tsp	41910	44972	45492	44598	7.31	8.55	6.41
u1060.tsp	224094	286667	251451	240193	27.92	12.21	7.18
u1432.tsp	152970	181206	188807	164045	18.46	23.43	7.24
u1817.tsp	57201	71024	72030	63539	24.17	25.92	11.08
u2152.tsp	64253	78581	79260	79217	22.30	23.36	23.29
u2319.tsp	234256	274542	278765	266890	17.20	19.00	13.93
vm1084.tsp	239297	265725	279047	255832	11.04	16.61	6.91
vm1748.tsp	336556	407007	408102	361567	20.93	21.26	7.43
Μέση Τιμή	253944.13	301717.44	322971.52	273781.10	14.50	14.81	5.26

Σε μερικές περιπτώσεις, η GVNS_3 στο χρονικό όριο του ενός λεπτού έδωσε καλύτερα αποτελέσματα σε σχέση με τις εκτελέσεις σε 2 λεπτά για τα συμμετρικά στιγμιότυπα. Αυτό μπορεί να συνέβη εξ αιτίας του εντελώς τυχαίου τρόπου διαταραχής όπως αυτό προκύπτει χρησιμοποιώντας τον τελεστή ανακατέματος. Ακριβέστερα, και εκτελώντας περισσότερες φορές τα πειράματα για την GVNS_3, ο τελεστής ανακατέματος εκτελείτε επίσης περισσότερες φορές και συνεπώς μπορεί να οδηγήσει την αναζήτηση σε περιοχές με όχι και τόσο καλές λύσεις. Έτσι, η αναζήτηση μπορεί να παγιδευτεί σε τοπικά βέλτιστα χαμηλής ποιότητας.

Table 4.8: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης καλύτερης βελτίωσης και χρονικό όριο εκτέλεσης τα 2 λεπτά

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1	GAP_2	GAP_3
a280.tsp	2579	2630	2725	2706	1.98	5.66	4.92
att48.tsp	10628	10628	10628	10820	0.00	0.00	1.81
bayg29.tsp	1610	1610	1610	1613	0.00	0.00	0.19
bays29.tsp	2020	2020	2020	2029	0.00	0.00	0.45
bier127.tsp	118282	118421	119583	119527	0.12	1.10	1.05
kroA100.tsp	21282	21282	21312	21631	0.00	0.14	1.64
burma14.tsp	3323	3323	3323	3323	0.00	0.00	0.00
ch130.tsp	6110	6137	6208	6337	0.44	1.60	3.72
ch150.tsp	6528	6564	6680	6749	0.55	2.33	3.39

Συνεχίζεται στην επόμενη σελίδα

Table 4.8 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
d493.tsp	35002	36232	37168	36822	3.51	6.19	5.20
kroB100.tsp	22141	22168	22249	22517	0.12	0.49	1.70
kroC100.tsp	20749	20749	20804	21402	0.00	0.27	3.15
kroD100.tsp	21294	21317	21461	21935	0.11	0.78	3.01
kroE100.tsp	22068	22122	22201	22450	0.24	0.60	1.73
kroA150.tsp	26524	26656	27119	27400	0.50	2.24	3.30
kroB150.tsp	26130	26232	26605	26816	0.39	1.82	2.63
kroA200.tsp	29368	29494	30276	30135	0.43	3.09	2.61
kroB200.tsp	29437	29732	30481	30843	1.00	3.55	4.78
d198.tsp	15780	15832	16036	16036	0.33	1.62	1.62
brg180.tsp	1950	1955	2014	2166	0.26	3.28	11.08
berlin52.tsp	7542	7542	7542	7737	0.00	0.00	2.59
dantzig42.tsp	699	699	699	703	0.00	0.00	0.57
eil51.tsp	426	426	426	432	0.00	0.00	1.41
eil76.tsp	538	538	542	543	0.00	0.74	0.93
eil101.tsp	629	630	644	648	0.16	2.38	3.02
fri26.tsp	937	937	937	940	0.00	0.00	0.32
gil262.tsp	2378	2432	2505	2497	2.27	5.34	5.00
gr17.tsp	2085	2085	2085	2086	0.00	0.00	0.05
gr21.tsp	2707	2707	2707	2707	0.00	0.00	0.00
gr24.tsp	1272	1272	1272	1273	0.00	0.00	0.08
gr48.tsp	5046	5046	5046	5089	0.00	0.00	0.85
gr96.tsp	55209	55259	55431	56269	0.09	0.40	1.92
gr120.tsp	6942	6975	7072	7130	0.48	1.87	2.71
gr137.tsp	69853	69869	70885	71215	0.02	1.48	1.95
gr202.tsp	40160	40860	41588	41762	1.74	3.56	3.99
gr229.tsp	134602	136101	139315	138770	1.11	3.50	3.10
gr431.tsp	171414	176629	181296	178732	3.04	5.76	4.27
hk48.tsp	11461	11461	11461	11519	0.00	0.00	0.51
lin105.tsp	14379	14379	14398	14505	0.00	0.13	0.88
lin318.tsp	42029	42989	43941	43768	2.28	4.55	4.14
pcb442.tsp	50778	52381	54108	52870	3.16	6.56	4.12
pr76.tsp	108159	108159	108227	109317	0.00	0.06	1.07
pr107.tsp	44303	44384	44473	44502	0.18	0.38	0.45
pr124.tsp	59030	59030	59039	59460	0.00	0.02	0.73
pr136.tsp	96772	97202	98613	100072	0.44	1.90	3.41
pr144.tsp	58537	58537	58544	60558	0.00	0.01	3.45
pr152.tsp	73682	73808	73884	74209	0.17	0.27	0.72
pr226.tsp	80369	80411	80677	81071	0.05	0.38	0.87
pr264.tsp	49135	49324	50715	52468	0.38	3.22	6.78
pr299.tsp	48191	48906	50363	51424	1.48	4.51	6.71
pr439.tsp	107217	110910	112735	114367	3.44	5.15	6.67
rat99.tsp	1211	1211	1232	1251	0.00	1.73	3.30
rat195.tsp	2323	2349	2448	2395	1.12	5.38	3.10
rd100.tsp	7910	7912	7943	8190	0.03	0.42	3.54
rd400.tsp	15281	15684	16272	16102	2.64	6.49	5.37
si175.tsp	21407	21422	21463	21510	0.07	0.26	0.48
st70.tsp	675	676	675	690	0.15	0.00	2.22
swiss42.tsp	1273	1273	1273	1273	0.00	0.00	0.00

Συνεχίζεται στην επόμενη σελίδα

Table 4.8 – Συνεχίζεται από την προηγούμενη σελίδα

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
ts225.tsp	126643	126654	127458	128716	0.01	0.64	1.64
tsp225.tsp	3916	3976	4107	4044	1.53	4.88	3.27
u159.tsp	42080	42282	42941	44205	0.48	2.05	5.05
ulysses16.tsp	6859	6859	6859	6860	0.00	0.00	0.01
ulysses22.tsp	7013	7013	7013	7041	0.00	0.00	0.40
ali535.tsp	202339	211954	217611	217607	4.75	7.55	7.55
att532.tsp	27686	28736	29545	29247	3.79	6.71	5.64
brazil58.tsp	25395	36518	25395	25395	43.80	0.00	0.00
brg180.tsp	1950	1964	2012	2178	0.72	3.18	11.69
d657.tsp	48912	51393	52852	51827	5.07	8.06	5.96
d1291.tsp	50801	54485	57344	55645	7.25	12.88	9.54
d1655.tsp	62128	67025	70119	67295	7.88	12.86	8.32
d2103.tsp	80450	83138	86653	83452	3.34	7.71	3.73
dsj1000.tsp	18659688	20039179	20433714	20142317	7.39	9.51	7.95
fl417.tsp	11861	11987	12111	12128	1.06	2.11	2.25
fl1400.tsp	20127	21201	21064	21204	5.34	4.66	5.35
fl1577.tsp	22249	23558	24252	23746	5.88	9.00	6.73
fl3795.tsp	28772	34390	32152	30584	19.53	11.75	6.30
fnl4461.tsp	182566	199718	204450	196648	9.39	11.99	7.71
gr666.tsp	294358	312215	319791	314260	6.07	8.64	6.76
nrv1379.tsp	56638	60460	62354	60951	6.75	10.09	7.62
p654.tsp	34643	37951	35465	36421	9.55	2.37	5.13
pa561.tsp	2763	2880	2886	3067	4.23	4.45	11.00
pcb1173.tsp	56892	61479	63387	61469	8.06	11.42	8.05
pcb3038.tsp	137694	151130	155733	149676	9.76	13.10	8.70
pr1002.tsp	259045	275588	284850	279390	6.39	9.96	7.85
pr2392.tsp	378032	410246	428532	408360	8.52	13.36	8.02
rat575.tsp	6773	7133	7153	7245	5.32	5.61	6.97
rat783.tsp	8806	9344	9250	9390	6.11	5.04	6.63
rl1304.tsp	252948	278657	280329	275084	10.16	10.82	8.75
rl1323.tsp	270199	291236	298519	285864	7.79	10.48	5.80
rl1889.tsp	316536	343698	353960	341771	8.58	11.82	7.97
rl5915.tsp	565530	677844	654919	633460	19.86	15.81	12.01
rl5934.tsp	556045	661867	644987	604596	19.03	16.00	8.73
si535.tsp	48450	48588	48769	48812	0.28	0.66	0.75
si1032.tsp	92650	92889	93382	92896	0.26	0.79	0.27
u574.tsp	36905	41429	39874	39865	12.26	8.04	8.02
u724.tsp	41910	44377	45604	44785	5.89	8.81	6.86
u1060.tsp	224094	241290	245211	243817	7.67	9.42	8.80
u1432.tsp	152970	164667	170476	165505	7.65	11.44	8.19
u1817.tsp	57201	62417	65840	61861	9.12	15.10	8.15
u2152.tsp	64253	69701	74338	70656	8.48	15.70	9.97
u2319.tsp	234256	243207	249416	245493	3.82	6.47	4.80
vm1084.tsp	239297	254315	262020	256838	6.28	9.50	7.33
vm1748.tsp	336556	359808	373774	356880	6.91	11.06	6.04
Μέση Τιμή	253944.1262	272756.94	277586.89	273027.78	3.55	4.49	4.24

Οι Πίνακες 4.9–4.12 περιέχουν τα συγκεντρωτικά πειραματικά αποτελέσματα για τα Εθνικά μετροπροβλήματα του TSP (National TSP). Αυτοί επίσης περιλαμβάνουν το όνομα του μετροπρο-

βλήματος, την βέλτιστη τιμή (zOpt), το κόστος για καθεμία από τις GVNS υλοποιήσεις (GVNS_1, GVNS_2 and GVNS_3) και τις διαφορές της καθεμιάς GVNS υλοποίησης από την βέλτιστη τιμή. Ο Πίνακας 4.9 παρουσιάζει τα GVNS σχήματα με την χρήση της στρατηγικής αναζήτησης πρώτης βελτίωσης και για χρόνο εκτέλεσης του κάθε πειράματος το ένα λεπτό. Ο Πίνακας 4.10 παραθέτει τις GVNS υλοποιήσεις με στρατηγική αναζήτησης την καλύτερη βέλτιστη και χρόνο εκτέλεσης του κάθε πειράματος το 1 λεπτό. Οι Πίνακες 4.11 and 4.12 παρέχουν τα αποτελέσματα που ανακτήθηκαν για πειράματα που εκτελέστηκαν για 2 λεπτά το καθένα και για πρώτη και καλύτερη βελτίωση σαν στρατηγική αναζήτησης. Μια αξιοσημείωτη παρατήρηση είναι ότι γενικά δεν υπάρχουν σημαντικές διαφορές μεταξύ διαφορετικών μεθόδων. Ωστόσο, παρατηρούμε ότι στην πρώτη βελτίωση τόσο για 1 όσο και για 2 λεπτά η GVNS_3 ξεπερνά τις GVNS_1 και GVNS_2. Αντίθετα, στην καλύτερη βελτίωση και οι τρεις μέθοδοι αποδίδουν καλύτερα γενικά από ό,τι στην πρώτη βελτίωση. Ωστόσο, δεν υπάρχει σημαντική διαφορά μεταξύ τους.

Table 4.9: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης πρώτης βελτίωσης και χρονικό όριο εκτέλεσης 1 λεπτό.

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
ar9152.tsp	837479	1648442	1648596	1648596	96.83	96.85	96.85
gr9882.tsp	300899	388910	388944	388944	29.25	29.26	29.26
eg7146.tsp	172387	220232	220315	220315	27.75	27.80	27.80
fi10639.tsp	520527	649604	649604	649604	24.80	24.80	24.80
ho14473.tsp	177105	484571	484812	484812	173.61	173.74	173.74
ei8246.tsp	206171	258851	258889	258889	25.55	25.57	25.57
ja9847.tsp	491924	612157	612304	612304	24.44	24.47	24.47
kz9976.tsp	1061882	1358247	1358247	1358247	27.91	27.91	27.91
lu980.tsp	11340	18708	23688	12236	64.97	108.89	7.90
mo14185.tsp	427377	529729	529729	529729	23.95	23.95	23.95
nu3496.tsp	96132	220359	221920	221920	129.23	130.85	130.85
mu1979.tsp	86891	119104	120908	120908	37.07	39.15	39.15
qa194.tsp	9352	9501	9727	9706	1.59	4.01	3.79
rw1621.tsp	26051	53343	58148	58052	104.76	123.21	122.84
tz6117.tsp	394718	500936	501184	501184	26.91	26.97	26.97
uy734.tsp	79114	84131	86022	83282	6.34	8.73	5.27
wi29.tsp	27603	27603	27603	27681	0.00	0.00	0.28
ym7663.tsp	238314	308477	308747	308747	29.44	29.55	29.55
zi929.tsp	95345	101541	112775	100572	6.50	18.28	5.48
ca4663.tsp	1290319	1646429	1646889	1646889	27.60	27.63	27.63
Μέση Τιμή	327546.50	462043.75	463452.55	462130.85	44.43	48.58	42.70

Table 4.10: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης καλύτερης βελτίωσης και χρονικό όριο εκτέλεσης 1 λεπτό

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
ar9152.tsp	837479	1317886	1648596	1648596	57.36	96.85	96.85
gr9882.tsp	300899	368968	388944	388944	22.62	29.26	29.26
eg7146.tsp	172387	206756	220315	220315	19.94	27.80	27.80
fi10639.tsp	520527	625547	649604	649604	20.18	24.80	24.80
ho14473.tsp	177105	362827	484812	459459	104.87	173.74	159.43
ei8246.tsp	206171	246249	258889	258889	19.44	25.57	25.57
ja9847.tsp	491924	595913	612304	612304	21.14	24.47	24.47
kz9976.tsp	1061882	1296253	1358247	1358247	22.07	27.91	27.91
lu980.tsp	11340	12052	12388	12530	6.28	9.24	10.49
mo14185.tsp	427377	514995	529729	529729	20.50	23.95	23.95
nu3496.tsp	96132	108839	108639	108685	13.22	13.01	13.06
mu1979.tsp	86891	93453	95413	93433	7.55	9.81	7.53
qa194.tsp	9352	9468	9717	9791	1.24	3.90	4.69
rw1621.tsp	26051	28360	28866	29127	8.86	10.81	11.81
tz6117.tsp	394718	478074	483082	445901	21.12	22.39	12.97
uy734.tsp	79114	83595	86201	84628	5.66	8.96	6.97
wi29.tsp	27603	27603	27603	27603	0.00	0.00	0.00
ym7663.tsp	238314	291939	308747	308747	22.50	29.55	29.55
zi929.tsp	95345	100461	103557	100474	5.37	8.61	5.38
ca4663.tsp	1290319	1532650	1463565	1428402	18.78	13.43	10.70
Μέση Τιμή	327546.50	415094.40	443960.90	438770.40	20.93	29.20	27.66

Table 4.11: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης πρώτης βελτίωσης και χρονικό όριο εκτέλεσης τα 2 λεπτά.

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
ar9152.tsp	837479	1648304	1648596	1648596	96.82	96.85	96.85
gr9882.tsp	300899	388916	388944	388944	29.25	29.26	29.26
eg7146.tsp	172387	220290	220315	220315	27.79	27.80	27.80
fi10639.tsp	520527	649604	649604	649604	24.80	24.80	24.80
ho14473.tsp	177105	484793	484812	484812	173.73	173.74	173.74
ei8246.tsp	206171	258867	258889	258889	25.56	25.57	25.57
ja9847.tsp	491924	612304	612304	612304	24.47	24.47	24.47
kz9976.tsp	1061882	1358246	1358247	1358247	27.91	27.91	27.91
lu980.tsp	11340	20909	23688	14382	84.38	108.89	26.83
mo14185.tsp	427377	529699	529729	529729	23.94	23.95	23.95
nu3496.tsp	96132	221466	221920	221920	130.38	130.85	130.85
mu1979.tsp	86891	119586	120908	120908	37.63	39.15	39.15
qa194.tsp	9352	9543	9811	9731	2.04	4.91	4.05
rw1621.tsp	26051	56293	58148	58148	116.09	123.21	123.21
tz6117.tsp	394718	501137	501184	501184	26.96	26.97	26.97
uy734.tsp	79114	85036	99005	83383	7.49	25.14	5.40
wi29.tsp	27603	27603	27603	27701	0.00	0.00	0.36
ym7663.tsp	238314	308713	308747	308747	29.54	29.55	29.55
zi929.tsp	95345	104572	113927	101795	9.68	19.49	6.76
ca4663.tsp	1290319	1645854	1646889	1646889	27.55	27.63	27.63
Μέση Τιμή	327546.50	462586.75	464163.50	462311.40	46.30	49.51	43.76

Table 4.12: Τα αποτελέσματα που παρουσιάζονται εδώ χρησιμοποίησαν τη στρατηγική αναζήτησης καλύτερης βελτίωσης και χρονικό όριο εκτέλεσης τα 2 λεπτά

Στιγμιότυπο	zOpt	GVNS_1	GVNS_2	GVNS_3	GAP_1 (%)	GAP_2 (%)	GAP_3 (%)
ar9152.tsp	837479	1372411	1648596	1648596	63.87	96.85	96.85
gr9882.tsp	300899	377898	388944	388944	25.59	29.26	29.26
eg7146.tsp	172387	209515	220315	220315	21.54	27.80	27.80
fi10639.tsp	520527	635913	649604	649604	22.17	24.80	24.80
ho14473.tsp	177105	449416	484812	474609	153.76	173.74	167.98
ei8246.tsp	206171	248441	258889	258889	20.50	25.57	25.57
ja9847.tsp	491924	602674	612304	612304	22.51	24.47	24.47
kz9976.tsp	1061882	1327647	1358247	1358247	25.03	27.91	27.91
lu980.tsp	11340	12151	12458	12559	7.15	9.86	10.75
mo14185.tsp	427377	527335	529729	529729	23.39	23.95	23.95
nu3496.tsp	96132	115162	108803	109804	19.80	13.18	14.22
mu1979.tsp	86891	94072	95874	94358	8.26	10.34	8.59
qa194.tsp	9352	9525	9757	9759	1.85	4.33	4.35
rw1621.tsp	26051	28784	29118	29131	10.49	11.77	11.82
tz6117.tsp	394718	475863	501184	501184	20.56	26.97	26.97
uy734.tsp	79114	84585	86358	84686	6.92	9.16	7.04
wi29.tsp	27603	27603	27603	27612	0.00	0.00	0.03
ym7663.tsp	238314	295414	308747	308747	23.96	29.55	29.55
zi929.tsp	95345	101161	104327	100881	6.10	9.42	5.81
ca4663.tsp	1290319	1573320	1646889	1470076	21.93	27.63	13.93
Μέση Τιμή	327546.50	428444.50	454127.90	444501.70	25.27	30.33	29.08

4.3 Στατιστική Ανάλυση των διαφορετικών μεθόδων

Αυτή η ενότητα παρουσιάζει τις στατιστικές δοκιμές που πραγματοποιήθηκαν στα υπολογιστικά αποτελέσματα, για την αξιολόγηση της απόδοσης των τριών διαφορετικών μεθόδων GVNS. Διαφορετικές στατιστικές δοκιμές εφαρμόζονται σε διαφορετικές δομές δεδομένων. Συγκεκριμένα, οι μέθοδοι στατιστικής ανάλυσης μπορούν να χωριστούν σε παραμετρικές και μη παραμετρικές δοκιμές. Η πρώτη κατηγορία εξετάζει κανονικές μεταβλητές, ενώ οι άλλες μέθοδοι αφορούν μη κανονικές μεταβλητές[34].

Αρχικά, η εφαρμογή ενός τεστ κανονικότητας έδειξε ότι τα αριθμητικά δεδομένα δεν ακολουθούν την κανονική κατανομή. Κατά συνέπεια, εφαρμόσαμε το τεστ **Kruskal-Wallis** για τον έλεγχο της ύπαρξης στατιστικά σημαντικής διαφοράς μεταξύ των μεθόδων. Σε αυτό το τεστ, η λήψη *p-value* μικρότερη από 0,05 σημαίνει ότι υπάρχει στατιστικά σημαντική διαφορά. Στο παρόν σημείο, πρέπει να αναφερθεί ότι η στατιστική ανάλυση πραγματοποιήθηκε σε μέσες τιμές για την εξάλειψη πιθανής ακραίας απόκλισης στις μέσες τιμές με βάση ακραίες τιμές. Επίσης, η σχετική με την ανάλυση aTSP προέρχεται από την προηγούμενη εργασία μας[147] και παρουσιάζεται εδώ για τη δημιουργία μιας πιο ολοκληρωμένης εικόνας.

4.3.1 Στατιστική Ανάλυση στα ασύμμετρα TSP

Στον Πίνακα 4.13 μπορούμε να δούμε ότι για όλες τις περιπτώσεις *p-value* η τιμή είναι μικρότερη από 0,05, πράγμα που σημαίνει ότι υπάρχει μια στατιστικά σημαντική διαφορά μεταξύ των τριών μεθόδων σε όλες τις περιπτώσεις. Για περαιτέρω εξέταση, πραγματοποιήθηκαν δοκιμές **Wilcoxon** κατά ζεύγη.

Table 4.13: Kruskal–Wallis τεστ κατηγοριοποίησης αθροίσματος

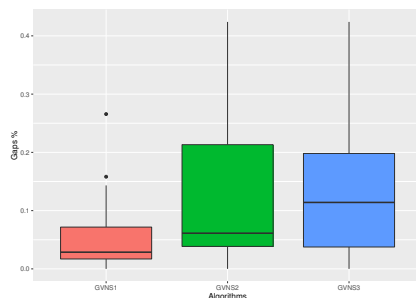
	X^2	df	<i>p</i> -Value
FI_1min	6.8689	2	0.0322
FI_2mins	9.0314	2	0.0109
BI_1min	9.2739	2	0.0097
BI_2mins	9.6658	2	0.008

Σύμφωνα με τις δοκιμές κατά ζεύγη που συνοψίζονται στον Πίνακα 4.14, είναι σαφές ότι το GVNS_1 έχει σημαντικές διαφορές με τα άλλα δύο σχήματα σε όλες τις περιπτώσεις. Τόσο το GVNS_2 όσο και το GVNS_3 αποδίδουν ισοδύναμα χρησιμοποιώντας τη στρατηγική αναζήτησης πρώτης βελτίωσης ανεξάρτητα από το χρονικό όριο εκτέλεσης, ενώ χρησιμοποιώντας τη στρατηγική καλύτερης βελτίωσης έχουν σημαντικές διαφορές που έχουν αναφερθεί και με τα δύο χρονικά όρια[147].

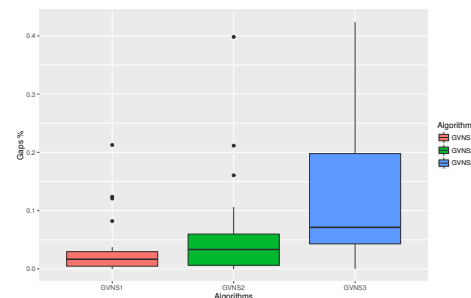
Table 4.14: KPairwise συγκρίσεις χρησιμοποιώντας τη Wilcoxon κατηγοριοποίηση δοκιμής

FI_1min		
	GVNS1	GVNS2
GVNS2	0.00064	
GVNS3	0.00064	0.6701
FI_2mins		
	GVNS1	GVNS2
GVNS2	0.00064	
GVNS3	0.00064	0.4488
BI_1min		
	GVNS1	GVNS2
GVNS2	0.00109	
GVNS3	0.00064	0.00064
BI_2mins		
	GVNS1	GVNS2
GVNS2	0.00109	
GVNS3	0.00064	0.00064

Σε σχέση με αυτήν τη στατιστική ανάλυση Kruskal – Wallis, έχουμε τέσσερα γραφήματα που απεικονίζονται στα σχήματα 4.1a, 4.1b, 4.2a και 4.2b. Κάθε ένα απεικονίζει είτε κατά Πρώτη Βελτίωση είτε κατά Καλύτερη Βελτίωση για ένα λεπτό, καθώς και για δύο λεπτά.

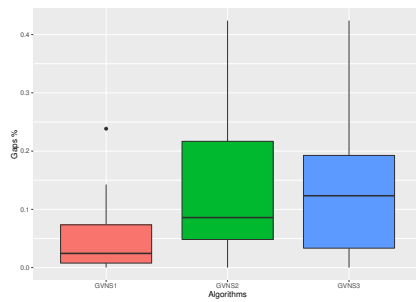


(a) Statistical test box plots for aTSP 1min FI

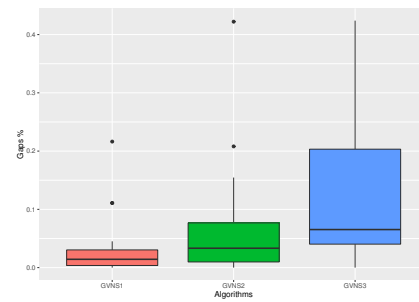


(b) Statistical test box plots for aTSP 1min BI

Figure 4.1: Στατιστικό test για το aTSP (1/2).



(a) Statistical test box plots for aTSP 2mins FI



(b) Statistical test box plots for aTSP 2mins BI

Figure 4.2: Στατιστικό τεστ για το aTSP (2/2).

4.3.2 Στατιστική Ανάλυση στα συμμετρικά TSP

Σε αυτήν την υποενότητα παρουσιάζεται η στατιστική ανάλυση των αποτελεσμάτων που προέκυψε από τα τρία σχήματα GVNS για τα συμμετρικά στιγμιότυπα (sTSP). Σύμφωνα με τις τιμές στον Πίνακα 4.15 μπορούμε να δούμε ότι μόνο για τη στρατηγική αναζήτησης καλύτερης βελτίωσης εντός χρονικού ορίου εκτέλεσης 2 λεπτών, υπάρχουν στατιστικά σημαντικές διαφορές.

Table 4.15: Kruskal–Wallis τεστ κατηγοριοποίησης αθροίσματος

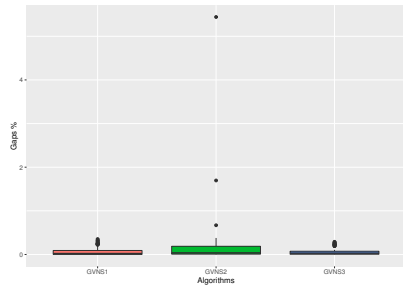
	X^2	df	p-Value
FI_1min	2.4392	2	0.2954
FI_2min	4.5181	2	0.1045
BI_1min	5.5397	2	0.06267
BI_2min	11.677	2	0.002913

Πιο συγκεκριμένα, οι τιμές στον Πίνακα 4.16 επισημαίνουν ότι υπάρχει διαφορά μεταξύ του GVNS_1 και των άλλων δύο αλγορίθμων GVNS. Συγκεκριμένα, με βάση τις ακόλουθες γραφικές παραστάσεις, το GVNS_1 είναι ελαφρώς καλύτερο από τα GVNS_2 και GVNS_3, τα οποία αποδίδουν σχεδόν ισοδύναμα.

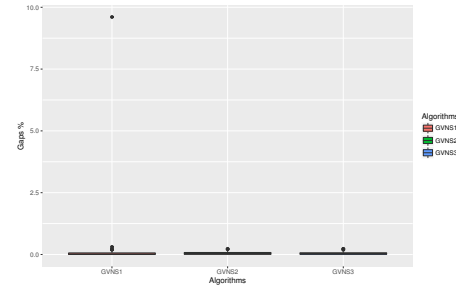
Table 4.16: KPairwise συγκρίσεις χρησιμοποιώντας τη Wilcoxon κατηγοριοποίηση δοκιμής

BI_2min		
	GVNS1	GVNS2
GVNS2	0.0000000990	
GVNS3	0.0000002300	0.6800000000

Στη συνέχεια αυτής της στατιστικής ανάλυσης Kruskal – Wallis, έχουμε τέσσερα γραφήματα που απεικονίζονται στα σχήματα 4.3a, 4.3b, 4.4a και 4.4b. Κάθε ένα απεικονίζει είτε εκτελέσεις κατά στρατηγική αναζήτησης πρώτης βελτίωσης είτε κατά καλύτερης βελτίωσης για ένα λεπτό, καθώς και για δύο λεπτά.

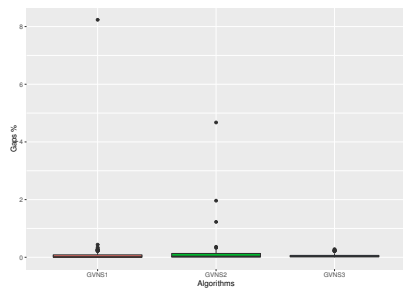


(a) Statistical test box plots for sTSP 1min FI

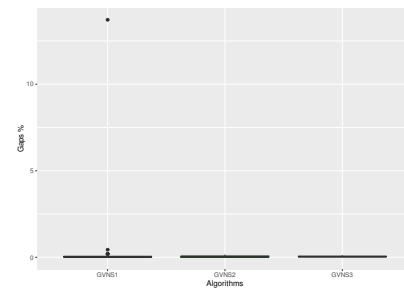


(b) Statistical test box plots for sTSP 1min BI

Figure 4.3: Στατιστικό τεστ για το sTSP (1/2).



(a) Statistical test box plots for sTSP 2min FI



(b) Statistical test box plots for sTSP 2min BI

Figure 4.4: Στατιστικό τεστ για το sTSP (2/2).

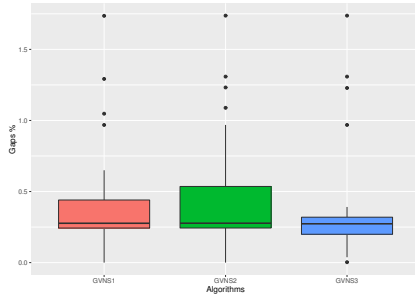
4.3.3 Στατιστική Ανάλυση των Εθνικών TSP

Στην περίπτωση των εθνικών TSP (nTSP) και με βάση τις τιμές που δίνονται στον Πίνακα 4.17 μπορούμε να συμπεράνουμε ότι δεν υπάρχει σημαντική στατιστική διαφορά μεταξύ των τριών μεθόδων.

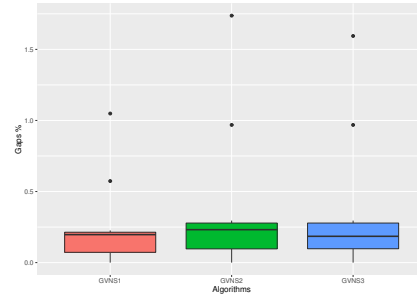
Table 4.17: Kruskal–Wallis τεστ κατηγοριοποίησης αθροίσματος

	X^2	df	p -Value
FI_1min	0.22253	2	0.8947
FI_2min	0.18068	2	0.9136
BI_1min	1.825	2	0.4015
BI_2min	1.9646	2	0.3744

Ως αποτέλεσμα αυτής της στατιστικής ανάλυσης Kruskal – Wallis, έχουμε τις εξής τέσσερις γραφικές παραστάσεις που απεικονίζονται στα σχήματα 4.5a, 4.5b, 4.6a και 4.6b. Κάθε ένα απεικονίζει το στατιστικό τεστ μεταξύ των τριών μεθόδων είτε κατά στρατηγική αναζήτησης πρώτης βελτίωσης είτε κατά καλύτερης βελτίωσης για ένα λεπτό, καθώς και για δύο λεπτά. Με τον έλεγχο της μέσης τιμής κάθε μεθόδου, είναι σαφές ότι οι τρεις αλγόριθμοι αποδίδουν ισοδύναμα.

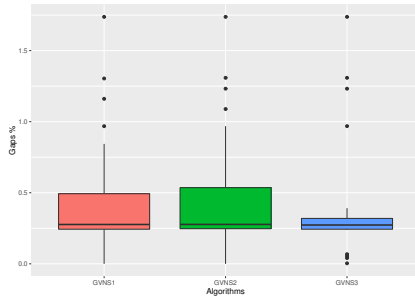


(a) Statistical test box plots for nTSP 1min FI

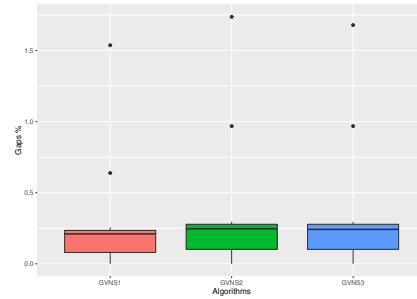


(b) Statistical test box plots for nTSP 1min BI

Figure 4.5: Στατιστικό τεστ για το nTSP (1/2).



(a) Statistical test box plots for nTSP 2min FI



(b) Statistical test box plots for nTSP 2min BI

Figure 4.6: Στατιστικό τεστ για το nTSP (2/2).

4.4 Σύγκριση της μεθόδου μας με άλλες υλοποιήσεις

Στην πρόσφατη δουλειά τους, οι Halim et Al. έχουν παρουσιάσει μια εκτενή ανάλυση σχετικά με την απόδοση διαφορετικών ευρετικών και μεθευρετικών αλγορίθμων σε ορισμένες περιπτώσεις της βιβλιοθήκης μετροπροβλημάτων, TSPLib [82]. Επιπλέον, Hore et al. [90] πρότειναν έναν βελτιωμένο υβριδικό αλγόριθμο VNS για την επίλυση στιγμιοτύπων TSP. Σε αυτήν την ενότητα, πραγματοποιείται σύγκριση μεταξύ των προτεινόμενων συστημάτων GVNS (GVNS_1 και GVNS_2) και αυτών των αλγορίθμων που παρουσιάζονται στις παρακάτω εργασίες [82, 90].

Ο πίνακας 4.18 δείχνει τη σύγκριση μεταξύ των GVNS_1 και GVNS_2 μας, εντός της στρατηγικής αναζήτησης Βέλτιστης Βελτίωσης και χρονικού ορίου 120 δευτερολέπτων, με όλες τις προσεγγιστικές προσεγγίσεις λύσεων που παρουσιάζονται στο έργο των Halim et al. [82]. Τα αποτελέσματα δείχνουν ότι οι μέθοδοι μας παράγουν καλύτερα αποτελέσματα από τα προαναφερθέντα μεθευρετικά στοιχεία, εκτός από την περίπτωση του rat195.tsp όπου το GA και το TS αποδίδουν καλύτερα από το GVNS_2.

Table 4.18: Συγκρίσεις μεταξύ των GVNS_1 και GVNS_2 με καλύτερη βελτίωση και δυο λεπτά με την πρόσφατη εργασία των Halim et al.

Στιγμιότυπο	OV	GVNS_1	GVNS_2	GA	SA	TS	ACO	TPO
eil51.tsp	426	426	426	454.1	439.13	439.1	467.46	437.26
berlin52.tsp	7542	7542	7542	7946.4	7960.67	7740.1	7922.32	7705.8
st70.tsp	675	676	675	700.72	696.33	690.27	756.55	697.12
kroA100.tsp	21282	21282	21312	22726.2	22277.5	22521.64	22941.68	22463.6
ch130.tsp	6110	6137	6208	6610.8	6558.7	6717.06	6913.99	6515.28
rat195.tsp	2323	2349	2448	2414.52	2537.99	2373.94	2465.11	2573.47
a280.tsp	2579	2630	2725	2789.83	2830.18	2800.79	2867.85	2790.54
rd400.tsp	15281	15684	16272	16567.29	16816.65	20723.56	19259.06	18190.84
pcb442.tsp	50778	52381	54108	55718.9	57421.04	83123.01	63436.7	60750.43

Ο πίνακας 4.19 δείχνει τη σύγκριση μεταξύ των GVNS_1 και GVNS_2 μας, εντός της στρατηγικής αναζήτησης Βέλτιστης Βελτίωσης και χρονικού ορίου 120 δευτερολέπτων με τα αποτελέσματα που αποκτήθηκαν από μια υβριδική υλοποίηση της VNS στη δεύτερη αναφερόμενη εργασία [90]. Παρατηρείται σαφώς ότι οι μέθοδοι μας ξεπερνούν την υβριδική VNS. Πιο συγκεκριμένα, η GVNS_1 ξεπερνά τον υβριδικό αλγόριθμο VNS σε όλες τις δοκιμασμένες περιπτώσεις, ενώ η GVNS_2 παράγει καλύτερα αποτελέσματα από αυτά που επιτεύχθηκαν από τους Hore και λοιποί σε επτά από τα 10 στιγμιότυπα των προβλημάτων που εξετάζουμε.

Table 4.19: Συγκρίσεις μεταξύ των GVNS_1 και GVNS_2 με καλύτερη βελτίωση και δυο λεπτά με την πρόσφατη δουλεία με βάση την VNS των Hore et al.

Στιγμιότυπο	OV	GVNS_1	GVNS_2	Μέση Τιμή	Μέση Τιμή Χρόνου
eil51.tsp	426	426	426	428.98	454.1
berlin52.tsp	7542	7542	7542	7544.36	7946.4
st70.tsp	675	676	675	677.11	700.72
kroA100.tsp	21282	21282	21312	21695.79	22726.2
ch130.tsp	6110	6137	6208	6153.72	6610.8
rat195.tsp	2323	2349	2448	2453.81	1382.34
rd400.tsp	15281	15684	16272	16250.21	1953.49
pcb1173.tsp	56892	61479	63387	63435.95	9531.54
pcb442.tsp	50778	52381	54108	50800.24	2183.27

Ο Πίνακας 4.20 δείχνει τις συντομογραφίες σχετικά με τους μεθευρετικούς αλγόριθμους που παρουσιάζονται στην εργασία [82].

Table 4.20: Συντομογραφίες.

GA	Γενετικός Αλγόριθμος
SA	Προσομοιωμένη Ανόπτηση
TS	Αναζήτηση Ταμπού
ACO	Βελτιστοποίηση Αποικιών
TPO	Βελτιστοποίηση Φυσιολογίας Δέντρων

Μια διεξοδική και περιεκτική ανάλυση απόδοσης σχετικά με την αποτελεσματικότητα τριών αλγορίθμων **GVNS** παρουσιάστηκε στα παραπάνω κεφάλαια. Η κύρια διαφορά έγκειται στη στρατηγική διαταραχής που χρησιμοποιείται. Η συγκριτική ανάλυση απόδοσης περιλαμβάνει ασύμμετρα και συμμετρικά στιγμιότυπα του **TSP** που επιλύονται χρησιμοποιώντας την **GVNS**. Τα γνωστά μετροπροβλήματα του **TSP** πάρθηκαν από την **TSPLIB** και χρησιμοποιούνται για εκτεταμένες πειραματικές δοκιμές. Πιστεύουμε ότι τα πειραματικά αποτελέσματα είναι αρκετά πειστικά, καθώς επιβεβαιώνουν ότι για τα ασύμμετρα στιγμιότυπα του **TSP** η **GVNS_1** ξεπερνά τις άλλες δύο μεθόδους και η **GVNS_2** παρέχει με συνέπεια καλύτερες λύσεις σε όλες τις περιπτώσεις σε σύγκριση με την **GVNS_3**. Ταυτόχρονα, η στρατηγική διαταραχής δεν φαίνεται να επηρεάζει σημαντικά τις λύσεις των συμμετρικών στιγμιότυπων του **TSP**. Αξίζει επίσης να τονιστεί ότι παρόλο που το στάδιο βελτίωσης των **GVNS** αλγορίθμων έχει περιορισμένη δυνατότητα, το παρόν κεφάλαιο δείχνει ότι οι δοκιμασμένες προσεγγίσεις για τη λύση, μπορεί να είναι αρκετά ελπιδοφόρες. Αυτό δικαιολογείται από τις παραγόμενες λύσεις, οι οποίες είναι σημαντικά καλύτερες από άλλες μεθευρετικές προσεγγίσεις από την πρόσφατη βιβλιογραφία.



5. Το πρόβλημα περισυλλογής απορριμάτων

5.1 Το πρόβλημα περισυλλογής απορριμάτων ως TSP

Μια διαδικασία με ουσιαστική σημασία για τον χειρισμό του καθημερινού προγραμματισμού της κυκλοφορίας μιας πόλης είναι η δρομολόγηση των απορριματοφόρων από τις αποθήκες τους σε κάθε κάδο απορριμμάτων μέσω των προγραμματισμένων διαδρομών τους και πίσω στις αποθήκες τους. Οι βέλτιστες διαδρομές αντιστοιχούν στον ελάχιστο απαιτούμενο χρόνο μεταφοράς και στην ελάχιστη απόσταση. Η εύρεση βέλτιστων διαδρομών αποδεικνύεται συνήθως χρονοβόρα, ειδικά στην περίπτωση μητροπολιτικών πόλεων με πολύ πυκνά οδικά δίκτυα. Ωστόσο, εκμεταλλευόμενοι τις πρόσφατες εξελίξεις από τον τομέα της συνδυαστικής βελτιστοποίησης και πιο συγκεκριμένα στον χώρο των μεθευρετικών διαδικασιών, είναι δυνατόν να επιτευχθούν αποτελεσματικές, σχεδόν βέλτιστες λύσεις σε σύντομο χρονικό διάστημα για πολλές πρακτικές περιπτώσεις. Από την ερευνητική μας προσπάθεια, εκμεταλλευόμαστε το πλεονέκτημα που μας παρέχει η βελτίωση της απόδοσης μέσω της νέας μεθευρετικής διαδικασίας που προτείνουμε και βασίζεται στην αναζήτηση μεταβλητής γειτονιάς (VNS). Η νέα GVNS που αναπτύξαμε -και έχουμε αναφερθεί εκτενώς σε αυτήν και στις παραλλαγές της σε προηγούμενες ενότητες της παρούσας διατριβής, σε συνδυασμό με τον ελάχιστο απαιτούμενο υπολογιστικό χρόνο, μπορεί να προσφέρει σημαντικά βελτιωμένες λύσεις για τέτοιου είδους προβλήματα από τον πραγματικό κόσμο. Η ενσωμάτωση της βελτιωμένης διαδικασίας VNS στο σύστημα γεωπληροφόρησης GIS θα μειώσει το χρόνο απόκρισης του συστήματος και θα παρέχει σχεδόν βέλτιστες λύσεις. Η τεχνολογία γεωπληροφόρησης (GIS) ενσωματώνει κοινές λειτουργίες βάσης δεδομένων, όπως ερωτήματα και στατιστική ανάλυση με τα μοναδικά οφέλη οπτικοποίησης και γεωγραφικής ανάλυσης που προσφέρονται από τους χάρτες [61, 133].

Μεταξύ άλλων, ένα σύστημα γεωπληροφόρησης (GIS) διευκολύνει τη μοντελοποίηση χωρικών δικτύων (π.χ. οδικά δίκτυα) προσφέροντας αλγόριθμους για την αναζήτηση και την ανάλυσή τους. Τα χωρικά δίκτυα μοντελοποιούνται με γραφήματα. Στην περίπτωση οδικών δικτύων, τα τόξα του γραφήματος αντιστοιχούν σε τμήματα οδών, ενώ οι κόμβοι αντιστοιχούν σε διασταυρώσεις τμήματος δρόμου. Κάθε τόξο έχει ένα βάρος που σχετίζεται με αυτό, που αντιπροσωπεύει το κόστος της διέλευσής του.

Τα συστήματα γεωπληροφόρησης παρέχουν συνήθως έναν αριθμό εργαλείων για την ανάλυση χωρικών δικτύων. Γενικά προσφέρουν εργαλεία για την εύρεση της συντομότερης ή ελάχιστης διαδρομής μέσω ενός δικτύου και ευρετικές διαδικασίες για την εύρεση της πιο αποτελεσματικής διαδρομής σε μια σειρά τοποθεσιών. Ένα τέτοιο πρόβλημα συνήθως διαμορφώνεται και μοντελοποιείται ως παράδειγμα του προβλήματος του πλανόδιου πωλητή (TSP). Συνεχίζοντας, η υλοποίησή μας λύνει αποτελεσματικά το πρόβλημα του πλανόδιου πωλητή (TSP), βρίσκοντας σχεδόν βέλτιστες λύσεις για μια σειρά μικρών, μεσαίων ή μεγαλύτερων προβλημάτων αναφοράς. Ο υπολογισμός του πίνακα των αποστάσεων μπορεί να χρησιμοποιηθεί για τον υπολογισμό των αποστάσεων μεταξύ ζευγών κόμβων που αντιπροσωπεύουν προέλευση και προορισμούς, ενώ οι συναρτήσεις κατανομής θέσης καθορίζουν τις τοποθεσίες. Αυτές οι δυνατότητες του συστήματος γεωπληροφόρησης για ανάλυση των χωρικών δικτύων του επιτρέπουν να χρησιμοποιηθεί ως σύστημα υποστήριξης αποφάσεων για τη δρομολόγηση των οχημάτων [36, 104].

Η δρομολόγηση ενός απορριμματοφόρου από την αποθήκη του σε κάθε κάδο απορριμμάτων και προς τα πίσω διαμορφώνεται με την εύρεση διαδρομών για το πρόβλημα του πλανόδιου πωλητή. Το σύστημα γεωπληροφόρησης θα χρησιμοποιηθεί για την εύρεση των βέλτιστων διαδρομών που αντιστοιχούν στον ελάχιστο απαιτούμενο χρόνο μεταφοράς. Το σύστημα γεωπληροφόρησης μπορεί επίσης να παρουσιάσει στον οδηγό οδηγίες που αντιστοιχούν στις διαδρομές που δημιουργούνται. Αυτές οι οδηγίες θα μεταδοθούν στο απορριμματοφόρο. Σε ένα σύστημα πραγματικού χρόνου (real time system), η απόδοση του χρόνου δρομολόγησης είναι ζωτικής σημασίας. Οι μεθευρετικές διαδικασίες, όπως η εφαρμογή μας που παρουσιάζεται στην παρούσα διατριβή, μπορεί να το εγγυηθεί.

5.1.1 Η προσέγγισή μας

Σε αυτή την εργασία, περιγράφουμε ένα σύστημα που προσφέρει λύση στο πρόβλημα της διαχείρισης των δρομολογίων των απορριμματοφόρων. Βασίζεται σε μία κβαντικά εμπνευσμένη μεθευρετική που στην ουσία επιλύει το Πρόβλημα του Πλανόδιου Πωλητή (TSP) και ενσωματώνεται στις τεχνολογίες των συστημάτων γεωπληροφορικής και γεωπληροφόρησης (GIS / GPS). Η προσέγγισή μας είναι μια ολοκληρωμένη λύση διαχείρισης αποβλήτων. Βάσει των λειτουργικών απαιτήσεων και ορισμένων περιπτωσιολογικών μελετών, [8] τα στοιχεία της εφαρμογής μας σχεδιάζονται και αναλύονται σε υποσυστήματα και μικρότερες λειτουργικές μονάδες. Οι λειτουργίες και οι σχέσεις μεταξύ υποσυστημάτων καθορίζονται για κάθε υποσύστημα όπως φαίνεται από την παρακάτω λίστα:

- Αισθητήρες κάδου: εξειδικευμένος εξοπλισμός που εκτιμά το επίπεδο πληρότητας του κάδου απορριμμάτων και συλλέγει, αποθηκεύει και μεταδίδει τα δεδομένα των κάδων. Αυτό θα βοηθήσει το κύριο σύστημά μας να λάβει υπόψη συγκεκριμένους κάδους και να αποφύγει να συμπεριλάβει στον υπολογισμό της τελικής διαδρομής του άδειους κάδους.
- Συλλογή δεδομένων από κάδους: Για την συλλογή αυτή υπάρχει μια μονάδα επεξεργασίας που επικοινωνεί με τους αισθητήρες των κάδων και ανακτά τις συλλεγόμενες πληροφορίες στο κεντρικό σύστημα. Μπορεί να εγκατασταθεί σε οχήματα που περνούν και αποτελείται από τρία εξαρτήματα:
 - i. από την διαδικασία της επικοινωνίας, που υλοποιεί την επικοινωνία με τον κάδο.
 - ii. από την διαδικασία αποθήκευσης, που αποθηκεύει προσωρινά τα δεδομένα έως ότου μεταφερθούν στο κεντρικό σύστημα.
 - iii. από τη διαδικασία μεταφοράς, που εφαρμόζει τη μεταφορά δεδομένων στο κεντρικό σύστημα.

Όλες αυτές οι λειτουργίες επιτυγχάνονται μέσω του δικτύου κινητής τηλεφωνίας (GSM).

- Εφαρμογή πλοήγησης βασισμένη σε σύστημα γεωπληροφόρησης (GPS) ενσωματωμένη στα φορτηγά απορριμματοφόρα: Πρόκειται για μια κλασική εφαρμογή πλοήγησης μέσω GPS, η οποία θα παρέχει την καθοδήγηση για την πλοήγηση στον οδηγό του απορριμματοφόρου και θα δίνει τις απαραίτητες οδηγίες σχετικά με το ποιοι κάδοι πρέπει να συλλέγονται.
- Κεντρικό σύστημα: Είναι το back-end σύστημα της εφαρμογής μας. Το κύριο μέρος του είναι η αποθήκευση των δεδομένων, τα δεδομένα που συλλέγονται από τους κάδους, τα δεδομένα των οχημάτων και όλα τα απαραίτητα δεδομένα για τον υπολογισμό των πιο αποδοτικών διαδρομών. Επιπλέον, η αποθήκευση των δεδομένων διατηρεί τυχόν πληροφορίες που ανακτώνται από τα άλλα υποσυστήματα, ιδιαίτερα από το υποσύστημα ανάκτησης δεδομένων (data mining sub-system) και το υποσύστημα της βελτιστοποίησης των διαδρομών (routing optimization system). Όλες οι δημιουργούμενες χωρικές πληροφορίες για την τρέχουσα διαδρομή θα αποθηκευτούν τοπικά σε μια χωρική βάση δεδομένων.
- Υποδομή χαρτών: Ένα API τεχνολογίας RESTful βασισμένο σε χάρτες που θα παρέχουν όλες τις απαιτούμενες λειτουργίες για τη δημιουργία εφαρμογών εμπλουτισμένου ιστού βάσει γεωγραφικών και περιγραφικών δεδομένων.
- Υποσύστημα Data Mining: Χρησιμοποιείται κυρίως για την εκτίμηση της πληρότητας των κάδων όταν δεν έχουμε ενημερωμένες τις διαθέσιμες πληροφορίες.
- Δομή της βελτιστοποίησης δρομολόγησης: Η κύρια συμβολή μας βασίζεται κυρίως στην εφαρμογή αποτελεσματικών αλγορίθμων δρομολόγησης για αυτήν την εφαρμογή. Προτείνουμε αυτή τη νέα χβαντικά εμπνευσμένη μεθυστική διαδικασία που παρέχει βέλτιστες ή σχεδόν βέλτιστες λύσεις σε σύντομο χρονικό διάστημα για το πρόβλημα της δρομολόγησης των απορριμματοφόρων και κατ'επέκταση του συνολικού προβλήματος της συγκομιδής των απορριμμάτων. Η υλοποίηση της λειτουργικότητας δρομολόγησης δίνει ένα συνολικό συγκριτικό πλεονέκτημα σε σύγκριση με τις άλλες υλοποιήσεις για δύο βασικούς λόγους
 - i. Μπορούμε να υπολογίσουμε αποτελεσματικές λύσεις σε σύντομο χρονικό διάστημα. Αυτό καθιστά ολόκληρη την εφαρμογή αποτελεσματική, επειδή μπορούμε να υπολογίσουμε και να ξανά υπολογίσουμε σε πραγματικό χρόνο και πολλές φορές για την ίδια διαδρομή και να τροφοδοτήσουμε τα αποτελέσματα στην εφαρμογή.
 - ii. Ο αλγόριθμος δρομολόγησης ξεπερνά σε αποτελέσματα τις κλασικές μεθόδους που χρησιμοποιούνται για την εύρεση διαδρομών σε συστήματα γεωπληροφόρησης (GPS) όπως ο αλγόριθμος του κοντινότερου γείτονα (Nearest Neighbor, NN) και οι λύσεις για κάθε κατηγορία προβλημάτων είναι σχεδόν βέλτιστες.

Μια λειτουργία μεγάλης σημασίας για τον χειρισμό καθημερινών περιστατικών στην πραγματική ζωή μιας πόλης, είναι η δρομολόγηση των απορριμματοφόρων από τον σταθμό εκκίνησης προς τους κάδους απορριμμάτων και η επιστροφή τους πίσω στον σταθμό με αποτελεσματικό τρόπο, προκειμένου να μειωθεί το κόστος του καυσίμου αλλά και ο χρόνος περιήγησης του απορριμματοφόρου.

Ένα σύστημα που εκτελείται σε πραγματικό χρόνο όπως το δικό μας θα πρέπει να είναι σε θέση να δίνει άμεσες απαντήσεις σε τέτοια ερωτήματα, διότι σε αυτές τις περιπτώσεις ο χρόνος απόκρισης είναι ζωτικής σημασίας. Χρησιμοποιώντας αποτελεσματικές δομές τοπικής αναζήτησης και την νέα χβαντικά εμπνευσμένη GVNS που προτείνουμε για την επίλυση τέτοιων προβλημάτων δρομολόγησης, ο μεθυστικός αλγόριθμος μπορεί να προσφέρει καλύτερα αποτελέσματα.

Οι περισσότερες από τις εφαρμογές συστημάτων γεωπληροφόρησης (GIS / GPS) χρησιμοποιούν είτε τον πλησιέστερο γείτονα (NN) είτε ορισμένες από τις παραλλαγές του για τον υπολογισμό των διαδρομών. Ο αλγόριθμος του πλησιέστερου γείτονα είναι μια ευρέως γνωστή και ευρέως χρησιμοποιούμενη ευρετική κατασκευή στα προβλήματα σχεδιασμού δικτύων. Αρχικά στον πλησιέστερο γείτονα, ένας τυχαίος κόμβος εισάγεται στη διαδρομή ως ο αρχικός κόμβος και στη συνέχεια, επαναληπτικά ο πλησιέστερος προς τον τελευταίο κόμβο που προστέθηκε, επιλέγεται για εισαγωγή στη διαδρομή. Η διαδικασία τερματίζεται όταν έχουν προστεθεί όλοι οι κόμβοι στη διαδρομή. Επομένως, λόγω της δημοτικότητας του πλησιέστερου γείτονα, όταν δοκιμάζουμε την απόδοση του αλγορίθμου μας, συγκρίνουμε τα αποτελέσματά του τόσο με τη βέλτιστη λύση (του κάθε μετροπροβλήματος) όσο και με τις λύσεις που επιτεύχθηκαν από τον πλησιέστερο γείτονα. Αυτή η σύγκριση δείχνει ότι η χβαντικά εμπνευσμένη GVNS παράγει αποτελέσματα που σε όλες τις περιπτώσεις είναι σχεδόν βέλτιστα και σημαντικά καλύτερα από τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor), ο οποίος χρησιμοποιείται ευρέως από τα περισσότερα λογισμικά γεωπληροφόρησης. Ως αποτέλεσμα αυτών των καλύτερων λύσεων δρομολόγησης, είναι η συνολική ποσότητα του καυσίμου που καταναλώνεται από τα απορριμματοφόρα να μειώνεται δραστικά. Έτσι, η μέθοδος μας είναι φιλική προς το περιβάλλον, καθώς η κατανάλωση καυσίμου έχει άμεσο αντίκτυπο στο περιβάλλον.

5.1.2 Υπολογιστικά αποτελέσματα

Αυτή η ενότητα είναι παρουσιάζει τα πειραματικά αποτελέσματα που παρουσιάζουν τα πλεονεκτήματα της νέας, χβαντικά εμπνευσμένης έκδοσης της GVNS. Τα πειράματα υλοποιήθηκαν στην γλώσσα προγραμματισμού Fortran 90 και εκτελέστηκαν σε φορητό υπολογιστή με επεξεργαστή Intel Core i7-6700HQ 2.59GHz και με μνήμη 16 GB. Τρέχουμε κάθε μετροπρόβλημα τουλάχιστον πέντε φορές και έχουμε όριο εκτέλεσης του αλγορίθμου τα 60 δευτερόλεπτα ανά στιγμιότυπο. Επιλέξαμε αυτό το κατώφλι για δύο λόγους. Πρώτον, αρκεί να υπολογιστεί μια βέλτιστη ή σχεδόν βέλτιστη λύση και, δεύτερον, είναι αρκετά σύντομη ώστε να επιτρέπονται οι επαναλαμβανόμενες εκτελέσεις. Στη συνέχεια, διατηρήσαμε την καλύτερη λύση που βρήκαμε και υπολογίσαμε τον μέσο όρο όλων των διαδρομών προκειμένου να αποκομίσουμε το τελικό κόστος. Οι αλγόριθμοι δοκιμάστηκαν σε 48 μετροπροβλήματα από την βιβλιοθήκη στιγμιότυπων αναφοράς TSPLIB. Η TSPLIB είναι μια βιβλιοθήκη που περιέχει μια συλλογή στιγμιότυπων αναφοράς για το TSP. Αυτά, χαρακτηρίζονται από την ποικιλομορφία τους και την παραλλαγή τους σε σχέση με τη διάσταση του προβλήματος. Ακριβώς για αυτές τις ιδιότητες χρησιμοποιούνται ευρέως από τους ερευνητές για τη σύγκριση των αποτελεσμάτων[165].

5.1.3 Σύγκριση χβαντικά εμπνευσμένης GVNS και του πλησιέστερου γείτονα

Στην παρούσα ερευνητική εργασία, προτείνουμε μια νέα έκδοση GVNS που βασίζεται σε αρχές και τεχνικές χβαντικών υπολογιστών και την εφαρμόζουμε σε μια εφαρμογή συλλογής απορριμμάτων που είναι ένα πραγματικό πρόβλημα που βασίζεται στα συστήματα γεωπληροφόρησης (GPS). Παρακάτω παρουσιάζουμε τα υπολογιστικά αποτελέσματα της προσέγγισής μας. Όπως έχει ήδη αναφερθεί, ο αλγόριθμος του πλησιέστερου γείτονα και οι παραλλαγές του χρησιμοποιούνται ευρέως σε εφαρμογές γεωπληροφόρησης για την κατασκευή της συνολικής διαδρομής του υποκείμενου γραφήματος. Συγκεκριμένα, συγκρίνουμε τη μέθοδο μας με την ευρετική του πλησιέστερου γείτονα (NN) και δύο από τις πιο γνωστές παραλλαγές της: τον αλγόριθμο του επαναλαμβανόμενου πλησιέστερου γείτονα (repeated Nearest Neighbor) και τον αλγόριθμο του βελτιωμένου κοντινότερου γείτονα (improved Nearest Neighbor). Αυτή η ενότητα παρουσιάζει τη

συγκριτική ανάλυση μεταξύ των διαφορετικών παραλλαγών του κοντινότερου γείτονα, της νέας GVNS και της βέλτιστης τιμής, που δείχνει τα αποτελέσματα σε μια σειρά αριθμών και πινάκων. Είναι σημαντικό να επισημάνουμε ότι επιλέξαμε να εξετάσουμε τον πλησιέστερο γείτονα λόγω της ευρείας χρήσης του σε εφαρμογές γεωπληροφορικής.

Table 5.1: Τα υπολογιστικά αποτελέσματα δείχνουν ότι η νέα GVNS ξεπερνά τον αλγόριθμο του πλησιέστερου γείτονα σε όλα τα στιγμιότυπα του TSP.

Στιγμιότυπο	NN	Gap NN vs OV	GVNS BI	Gap GVNS BI vs. OV	GVNS FI	Gap GVNS FI vs. OV	OV
a280	3157	-0.2241	2779	-0.0775	2766	-0.0725	2579
att48	12861	-0.2101	10645	-0.0016	10654	-0.0024	10628
bayg29	2005	-0.2453	1610	0.0000	1610	0.0000	1610
bays29	2258	-0.1178	2020	0.0000	2020	0.0000	2020
bier127	135737	-0.1475	121393	-0.0263	121551	-0.0276	118282
kroA100	27807	-0.3065	21664	-0.0179	21774	-0.0231	21282
burma14	4501	-0.3544	3454	-0.0394	3454	-0.0394	3323
ch130	7579	-0.2404	6342	-0.0380	6373	-0.0430	6110
ch150	8191	-0.2547	6849	-0.0492	6871	-0.0525	6528
d493	41666	-0.1903	37715	-0.0775	37882	-0.0823	35002
kroB100	29158	-0.3169	22514	-0.0168	22786	-0.0291	22141
kroC100	26227	-0.2640	21148	-0.0192	21245	-0.0239	20749
kroD100	26947	-0.2654	21768	-0.0223	21916	-0.0292	21294
kroE100	27460	-0.2443	22512	-0.0201	22709	-0.0290	22068
kroA150	33633	-0.2680	27641	-0.0421	27794	-0.0479	26524
kroB150	34499	-0.3202	27032	-0.0345	27274	-0.0438	26130
kroA200	35859	-0.2210	30900	-0.0522	31179	-0.0617	29368
kroB200	36980	-0.2562	31119	-0.0571	31387	-0.0662	29437
d198	18240	-0.1558	16196	-0.0264	16260	-0.0304	15780
brg180	69550	-34.6666	2026	-0.0390	2038	-0.0451	1950
berlin52	8980	-0.1906	7547	-0.0007	7590	-0.0064	7542
dantzig42	956	-0.3676	701	-0.0029	701	-0.0029	699
eil101	803	-0.2766	647	-0.0286	649	-0.0318	629
eil51	511	-0.1939	428	0.0000	429	-0.0023	428
eil76	642	-0.1933	548	-0.0186	549	-0.0204	538
frn26	1112	-0.1867	937	0.0000	937	0.0000	937
gil262	3208	-0.3490	2571	-0.0812	2558	-0.0757	2378
gr17	2187	-0.0489	2085	0.0000	2085	0.0000	2085
gr21	3333	-0.2312	2707	0.0000	2707	0.0000	2707
gr24	1553	-0.2209	1272	0.0000	1272	0.0000	1272
gr48	6098	-0.2084	5048	-0.0004	5054	-0.0016	5046
gr96	75065	-0.3596	56084	-0.0158	56133	-0.0167	55209
gr120	9351	-0.3470	7197	-0.0367	7199	-0.0370	6942
gr137	98720	-0.4132	72381	-0.0362	72536	-0.0384	69853
gr202	47080	-0.1723	42419	-0.0563	42287	-0.0530	40160
gr229	169715	-0.2608	141387	-0.0504	142175	-0.0563	134602
gr431	221402	-0.2916	184140	-0.0742	184993	-0.0792	171414
hk48	13181	-0.1500	11498	-0.0032	11531	-0.0061	11461
lin105	20356	-0.4156	14613	-0.0163	14607	-0.0159	14379
lin318	54019	-0.2852	45018	-0.0711	45179	-0.0749	42029
pcb442	61979	-0.2205	55416	-0.0913	55804	-0.0990	50778
pr76	153462	-0.4188	109103	-0.0087	109421	-0.0117	108159
pr107	46680	-0.0536	45183	-0.0199	45464	-0.0262	44303
pr124	69297	-0.1739	59705	-0.0114	59742	-0.0121	59030
pr136	120769	-0.2479	99622	-0.0295	100718	-0.0408	96772
pr144	61652	-0.0532	58776	-0.0041	58991	-0.0078	58537
pr152	85699	-0.1630	74703	-0.0139	74943	-0.0171	73682
pr226	94683	-0.1781	81379	-0.0126	81781	-0.0176	80369

Ο πίνακας 5.1 περιέχει τα συγκεντρωτικά πειραματικά αποτελέσματα. Συγκεκριμένα, περιέχει το όνομα του στιγμιότυπου, το κόστος του πλησιέστερου γείτονα (NN), τον μέσο όρο των τιμών για την GVNS με στρατηγική αναζήτησης την πρώτη βελτίωση (FI) για κάθε πρόβλημα, την διαφορά μεταξύ της GVNS που χρησιμοποιεί την στρατηγική FI έναντι της βέλτιστης τιμής (OV), το μέσο όρο των τιμών για την GVNS που χρησιμοποιεί την στρατηγική αναζήτησης καλύτερης βελτίωσης (BI) για κάθε πρόβλημα, τη διαφορά μεταξύ της GVNS που χρησιμοποιεί την στρατηγική BI από την βέλτιστη τιμή (OV) και τη βέλτιστη τιμή (OV). Δεδομένου του αποτελέσματος x , το κενό του από τη βέλτιστη τιμή OV υπολογίζεται από τον τύπο $\frac{OV-x}{OV}$. Η διαφορά από τη βέλτιστη, χρησιμοποιείται ευρέως στον τομέα της βελτιστοποίησης για να μετρηθεί πόσο κοντά στο βέλτιστο είναι μια συγκεκριμένη λύση. Τα δεδομένα δείχνουν ότι οι διαφορετι-

κές παραλλαγές της νέας GVNS

- είναι σταθερά πολύ κοντά στη βέλτιστη τιμή και με στρατηγική αναζήτησης πρώτης βελτίωσης και με καλύτερης βελτίωσης, και
- ξεπερνάει τον πλησιέστερο γείτονα σε όλες τις περιπτώσεις.

Οι σκιασμένες γραμμές στον Πίνακα 5.1 υποδεικνύουν την ανωτερότητα της GVNS για ορισμένες περιπτώσεις των στιγμιotypών αναφοράς. Συγκεκριμένα επισημαίνουμε αυτές τις περιπτώσεις όπου η GVNS έχει 2% ή λιγότερο διαφορά από τη βέλτιστη τιμή. Για παράδειγμα, η διαφορά του πλησιέστερου γείτονα από τη βέλτιστη τιμή για το *att48* είναι -0,2101, ενώ η στρατηγική αναζήτησης της GVNS είναι η καλύτερη βελτίωση (BI), είναι -0,0016 και η GVNS χρησιμοποιώντας την στρατηγική πρώτης βελτίωσης (FI) είναι -0,0024. Επιπλέον, η GVNS επιτυγχάνει τη βέλτιστη τιμή (η διαφορά είναι 0,0000 από την βέλτιστη τιμή) για *bayg29*, *bays29*, *eil51*, *fri26*, *gr17*, *gr21*, και *gr24*. Μια άλλη χαρακτηριστική περίπτωση είναι το σημείο αναφοράς *lin105* όπου το ο πλησιέστερος γείτονας (NN) έχει διαφορά -0.4156 από το βέλτιστο ενώ η GVNS με στρατηγική καλύτερης βελτίωσης επιτυγχάνει -0.0163 και -0.0159 για την στρατηγική της πρώτης βελτίωσης.

Εκτός από την προηγούμενη σειρά πειραμάτων, πραγματοποιήσαμε επιπλέον δοκιμές χρησιμοποιώντας τις δύο πιο δημοφιλείς παραλλαγές της ευρετικής του Πλησιέστερου Γείτονα. Αυτές είναι ο επαναλαμβανόμενος πλησιέστερος γείτονας (*Repeat NN*) και ο βελτιωμένος πλησιέστερος γείτονας (*Improved NN*). Ο βελτιωμένος πλησιέστερος γείτονας είναι μια παραλλαγή του κλασικού πλησιέστερου γείτονα, στην οποία το ζεύγος εκκίνησης της διαδρομής είναι η μικρότερη ακμή στον πίνακα των αποστάσεων [158]. Οι υπόλοιποι κόμβοι προστίθενται στη διαδρομή με τρόπο όμοιο με την απλή μέθοδο του πλησιέστερου γείτονα. Ο αλγόριθμος του επαναλαμβανόμενου πλησιέστερου γείτονα είναι μια άλλη τροποποίηση στην οποία ο αλγόριθμος του πλησιέστερου γείτονα εφαρμόζεται για κάθε κόμβο. Τέλος, η διαδρομή με το ελάχιστο συνολικό κόστος επιλέγεται ως η καλύτερη.

Συγκρίναμε το GVNS με το βελτιωμένο πλησιέστερο γείτονα και τον επαναλαμβανόμενο πλησιέστερο γείτονα και τα πειραματικά αποτελέσματα περιέχονται στους πίνακες 5.2 και 5.3. Αυτοί οι πίνακες δείχνουν το όνομα συγκριτικής αξιολόγησης, το κόστος περιήγησης του βελτιωμένου πλησιέστερου γείτονα, το κόστος περιήγησης του επαναλαμβανόμενου πλησιέστερου γείτονα, την υλοποίηση της GVNS χρησιμοποιώντας σαν στρατηγική αναζήτησης την καλύτερη βελτίωση (BI), την GVNS χρησιμοποιώντας σαν στρατηγική αναζήτησης την πρώτη βελτίωση (FI) και τη βέλτιστη τιμή (OV). Για συνέπεια, εκτελέσαμε τα ίδια πειράματα όπως και πριν, αλλά αντί για τον απλό πλησιέστερο γείτονα, χρησιμοποιήσαμε τον βελτιωμένο πλησιέστερο γείτονα και τον επαναλαμβανόμενο πλησιέστερο γείτονα και συγκρίναμε τα αποτελέσματα με τις διαφορετικές χβαντικά εμπνευσμένες παραλλαγές της GVNS που έχουμε υλοποιήσει και με τη βέλτιστη τιμή. Ο βελτιωμένος και ο επαναλαμβανόμενος πλησιέστερος γείτονας επιτυγχάνουν καλύτερα αποτελέσματα σε σύγκριση με τον απλό πλησιέστερο γείτονα για κάθε μετροπρόβλημα. Ωστόσο, το γεγονός παραμένει ότι η GVNS και στις δύο περιπτώσεις στρατηγικών αναζήτησης εξακολουθούν να ξεπερνούν αυτές τις βελτιωμένες μεθόδους που βασίζονται στον πλησιέστερο γείτονα.

Table 5.2: Τα πειραματικά αποτελέσματα δείχνουν ότι η GVNS υπερτερεί του βελτιωμένου και του επαναλαμβανόμενου πλησιέστερου γείτονα σε στιγμιότυπα του TSP (1/2).

Στιγμιότυπο	Improved NN	Repeated NN	GVNS BI	GVNS FI	OV
a280	3171	3008	2779	2766	2579
att48	13447	12012	10645	10654	10628
bayg29	1938	1935	1610	1610	1610
bays29	2307	2134	2020	2020	2020
bier127	148330	133953	121393	121551	118282
kroA100	28244	24698	21664	21774	21282
burma14	4470	3822	3454	3454	3323
ch130	7342	7129	6342	6373	6110
ch150	7699	7113	6849	6871	6528
d493	41858	40189	37715	37882	35002
kroB100	28525	25884	22514	22786	22141
kroC100	25511	23660	21148	21245	20749
kroD100	29202	24852	21768	21916	21294
kroE100	28125	24782	22512	22709	22068
kroA150	32019	31479	27641	27794	26524
kroB150	37113	31611	27032	27274	26130
kroA200	36825	34543	30900	31179	29368
kroB200	38844	35389	31119	31387	29437
d198	18485	17620	16196	16260	15780
brg180	98460	59550	2026	2038	1950
berlin52	9156	8181	7547	7590	7542
dantzig42	957	864	701	701	699
eil101	823	746	647	649	629
eil51	555	482	428	429	428

Οι πίνακες 5.2 και 5.3 επιβεβαιώνουν ότι η GVNS τόσο χρησιμοποιώντας την πρώτη ή την καλύτερη βελτίωση σαν στρατηγική αναζήτησης ξεπερνά και τον βελτιωμένο και τον επαναλαμβανόμενο πλησιέστερο γείτονα. Ας εξετάσουμε για παράδειγμα το στιγμιότυπο αναφοράς *eil101*, το οποίο αποτελείται από 101 κόμβους. Τα υπολογιστικά αποτελέσματα δείχνουν ότι ο βελτιωμένος πλησιέστερος γείτονας έχει κόστος διαδρομής 823, ο επαναλαμβανόμενος έχει κόστος διαδρομής 746, ή GVNS με στρατηγική αναζήτησης την πρώτη βελτίωση έχει 649 και η GVNS με στρατηγική αναζήτησης την καλύτερη βελτίωση έχει 647, ενώ η βέλτιστη τιμή για τη της διαδρομή του συγκεκριμένου στιγμιότυπου είναι 629. Είναι σαφές ότι η χβαντικά εμπνευσμένη GVNS είναι οριακά κοντά στη βέλτιστη τιμή (OV) και η διαφορά μεταξύ των δύο εκδοχών της και των δύο παραλλαγών του πλησιέστερου γείτονα είναι σχετικά μεγάλη. Άλλα παραδείγματα σημείων αναφοράς που επισημαίνουν την ανωτερότητα της χβαντικά εμπνευσμένης GVNS είναι τα *bayg29*, *d198* και *ch130*.

Table 5.3: Τα πειραματικά αποτελέσματα δείχνουν ότι η GVNS υπερτερεί του βελτιωμένου και του επαναλαμβανόμενου πλησιέστερου γείτονα σε στιγμιότυπα του TSP (2/2)

Στιγμιότυπο	Improved NN	Repeated NN	GVNS BI	GVNS FI	OV
eil76	614	608	548	549	538
fri26	982	965	937	937	937
gil262	3027	2823	2571	2558	2378
gr17	2187	2178	2085	2085	2085
gr21	3265	3003	2707	2707	2707
gr24	1769	1553	1272	1272	1272
gr48	6287	5840	5048	5054	5046
gr96	70060	65416	56084	56133	55209
gr120	8791	8438	7197	7199	6942
gr137	93749	84376	72381	72536	69853
gr202	47460	45030	42419	42287	40160
gr229	167062	157745	141387	142175	134602
gr431	218383	197405	184140	184993	171414
hk48	13052	12137	11498	11531	11461
lin105	20359	16935	14613	14607	14379
lin318	54031	49201	45018	45179	42029
pcb442	64623	58950	55416	55804	50778
pr76	154708	130921	109103	109421	108159
pr107	46765	46680	45183	45464	44303
pr124	69154	67055	59705	59742	59030
pr136	127551	114553	99622	100718	96772
pr144	61904	60964	58776	58991	58537
pr152	85349	79564	74703	74943	73682
pr226	95573	92552	81379	81781	80369

Ο πίνακας 5.3 περιέχει υπολογιστικά δεδομένα που επιβεβαιώνουν ότι η GVNS έχει καλύτερη απόδοση από τον βελτιωμένο και τον επαναλαμβανόμενο πλησιέστερο γείτονα. Τα χαρακτηριστικά παραδείγματα είναι τα *gr229*, *lin105* και *pr136*, τα οποία αποτελούν στιγμιότυπα αναφοράς των 229, 105 και 136 κόμβων αντίστοιχα. Μπορούμε επομένως να συμπεράνουμε ότι το GVNS είναι πάντα κοντά στη βέλτιστη τιμή και σαφώς ξεπερνά τις παραλλαγές του πλησιέστερου γείτονα για όλες τις κλάσεις των προβλημάτων.

5.1.4 Υπεροχή χβαντικά εμπνευσμένης έναντι κλασσικής GVNS

Τα δεδομένα στον Πίνακα 5.4 αποκαλύπτουν ότι η νέα χβαντικά εμπνευσμένη GVNS είναι πράγματι βελτιωμένη σε σχέση με την κλασσική έκδοσή της. Από τα δεδομένα μπορεί κανείς να καταλήξει στο συμπέρασμα ότι σε όλες τις περιπτώσεις αυτή η νέα έκδοση της GVNS είναι τουλάχιστον τόσο καλή όσο η κλασσική GVNS, και σε πολλές περιπτώσεις υπερτερεί των κλασσι-κών GVNS. Αυτό είναι σύμφωνο με το έργο στο [148], το οποίο, μέσω μιας συγκριτικής ανάλυσης, έδειξε επίσης ότι η χβαντικά εμπνευσμένη GVNS επιτυγχάνει καλύτερα αποτελέσματα από τη συμβατική GVNS.

Table 5.4: Σύγκριση μεταξύ της νέας GVNS και της συμβατικής GVNS [148].

Στιγμιότυπο	Av. Value (nGVNS)	Best (GVNS)	Στιγμιότυπο	Av. Value (nGVNS)	Best (GVNS)
bayg29	1610	1653	br17	39	39
bays29	2020	2069	ftv33	1357.8	1489
fri26	937	969	ftv35	1544.6	1791
gr17	2085	2085	ftv38	1616.6	1778
gr24	1272	1278	ftv44	1763.8	2014
ulysses16	6859	6859	p43	5554	5629
ulysses22	7013	7013	ry48p	14698.2	15134
gr48	5049.4	5325			
hk48	11508.6	11884			

Ο πίνακας 5.5 περιέχει τα πειραματικά αποτελέσματα της χβαντικά εμπνευσμένης GVNS σε ορισμένα ασύμμετρα σημεία αναφοράς TSP.

Table 5.5: Η GVNS για ασύμμετρα TSP.

Στιγμιότυπο	OV	GVNS FI	GVNS BI
br17	39	39	39
ft53	6905	7328	7135
ft70	38673	40691	40206
ftv33	1286	1339	1286
ftv35	1473	1499	1473
ftv38	1530	1585	1541
ftv44	1613	1760	1644
ftv47	1778	1992	1816
ftv55	1608	1985	1665
ftv64	1839	2382	1986
ftv70	1950	2557	2157
ftv170	2755	3923	3852
kro124p	36230	43187	37076
p43	5620	5623	5620
rbg323	1326	2755	2755
rbg358	1163	2755	2755
rbg403	2465	2755	2755
rbg443	2720	2821	2806
ry48p	14422	14699	14439

Table 5.6: Η GVNS για εθνικά TSP.

Στιγμιότυπο	OV	GVNS FI	GVNS BI
ar9152	837479	1648596	1648596
gr9882	300899	388944	388944
eg7146	172387	220315	220315
fi10639	520527	649604	649604
ho14473	177105	484812	484812
ei8246	206171	258889	258889
ja9847	491924	612304	612304
kz9976	1061882	1358247	1358247
lu980	11340	23688	12388
mo14185	427377	529729	529729
nu3496	96132	221920	108639
mu1979	86891	120908	95413
qa194	9352	9727	9717
rw1621	26051	58148	28866
tz6117	394718	501184	483082
uy734	79114	86022	86201
wi29	27603	27603	27603
ym7663	238314	308747	308747
zi929	95345	112775	103557
ca4663	1290319	1646889	1463565
it16862	557315	706069	706069

Για να δοκιμάσουμε περαιτέρω τη εφικτότητα της μεθόδου μας, εφαρμόσαμε το νέο χβαντικά εμπνευσμένο GVNS σχήμα σε ορισμένα εθνικά TSP (nTSPs) προκειμένου να δοκιμάσουμε τη συμπεριφορά σε πολύ μεγαλύτερες περιπτώσεις. Ο παραπάνω πίνακας 5.6 περιέχει τα υπολογιστικά αποτελέσματα του νέου GVNS σε nTSPs.

Στόχος μας σε αυτό το άρθρο ήταν η εφαρμογή μιας μεθόδου βασισμένης σε γνωστούς μεθευρετικούς που θα ήταν αρκετά αποτελεσματικοί για την επίλυση πραγματικών προβλημάτων, όπως η συλλογή απορριμμάτων που μελετήθηκε εδώ. Θέλαμε να αναπτύξουμε έναν αλγόριθμο που θα μπορούσε να βρει σχεδόν βέλτιστες λύσεις σε σχετικά σύντομο χρονικό διάστημα. Για να επιτύχουμε τους στόχους μας, επιλέξαμε να εφαρμόσουμε αυτήν τη νέα έκδοση του GVNS, καθώς περιμέναμε να ξεπεράσουμε τα συμβατικά GVNS. Αυτό επιβεβαιώθηκε πράγματι πειραματικά, όπως δείχνουν τα αποτελέσματα στον Πίνακα 5.4. Το μεγαλύτερο μέρος των πειραμάτων μας αποσκοπούσε στον προσδιορισμό του τρόπου τιμής του GVNS έναντι καθιερωμένων μεθόδων που εφαρμόζονται ευρέως για το GPS, όπως ο πλησιέστερος γείτονας και οι σημαντικότερες παραλλαγές του. Οι πίνακες 5.1, 5.2 και 5.3 περιέχουν πειραματικά στοιχεία που υποδηλώνουν ότι το GVNS ξεπερνάει τον πλησιέστερο γείτονα, τον επαναλαμβανόμενο πλησιέστερο γείτονα και το βελτιωμένο ευρετικό κοντινότερο γείτονα σε όλες τις περιπτώσεις.

5.1.5 Γραφική Αναπαράσταση των αποτελεσμάτων

Σε αυτήν την ενότητα, παρουσιάζουμε δύο διαφορετικά αριθμητικά σύνολα με τρία διαφορετικά γραφήματα ράβδων για κάθε σύνολο. Κάθε σχήμα αφορά ένα διαφορετικό υποσύνολο των

συνολικών πειραμάτων, ταξινομημένο σύμφωνα με τη βέλτιστη τιμή. Στο πρώτο σετ (Figures 5.1, 5.2 και 5.3), κάθε πρόβλημα αναφοράς αντιπροσωπεύεται με 4 διαφορετικές γραμμές, μία για κάθε μέθοδο. Συγκεκριμένα, το ένα αντιπροσωπεύει τον αλγόριθμο Πλησιέστερου Γείτονα, το ένα τη βέλτιστη τιμή και άλλα δύο απεικονίζουν την GVNS (καλύτερη και πρώτη βελτίωση). Μπορεί εύκολα να διαπιστωθεί ότι οι αλγόριθμοι μας μας επιτυγχάνουν αποτελέσματα πολύ κοντά στο βέλτιστο, σε αντίθεση με τον αλγόριθμο του πλησιέστερου γείτονα που δεν παρέχει αποτελεσματικές λύσεις.

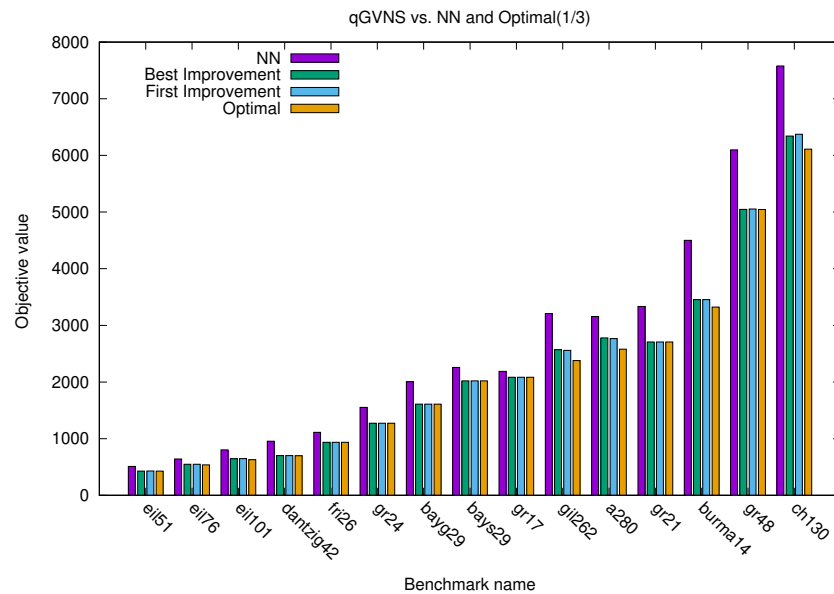


Figure 5.1: Σύγκριση μεταξύ των GVNS του πλησιέστερου γείτονα και της βέλτιστης τιμής (1/3).

Ένα αξιοσημείωτο παράδειγμα του Σχήματος 5.1 είναι το *ch130*. Μπορούμε να δούμε ότι η GVNS (και οι δύο παραλλαγές) είναι πολύ πιο κοντά στη βέλτιστη τιμή, σε σύγκριση με τον πλησιέστερο γείτονα.

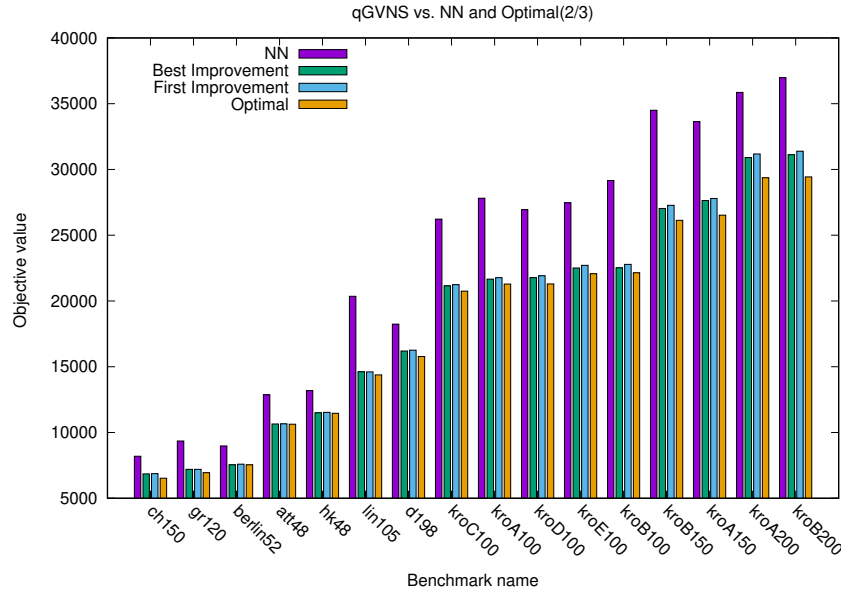


Figure 5.2: Σύγκριση μεταξύ των GVNS του πλησιέστερου γείτονα και της βέλτιστης τιμής (2/3).

Είναι σαφές από το Σχήμα 5.2 ότι στο *kroC100*, *KroA100*, *KroD100*, *KroE100*, *KroB100*, *KroB150*, *KroA150*, *KroA200* και *KroB200* η διαφορά μεταξύ της μεθόδου του πλησιέστερου γείτονα και των δύο παραλλαγών της GVNS είναι πολύ μεγάλη. Μπορούμε επίσης να παρατηρήσουμε ορισμένες περιπτώσεις στο Σχήμα 5.2, όπου η GVNS επιτυγχάνει τη βέλτιστη τιμή. Αυτά είναι *berlin52*, *hk48* και *att48*.

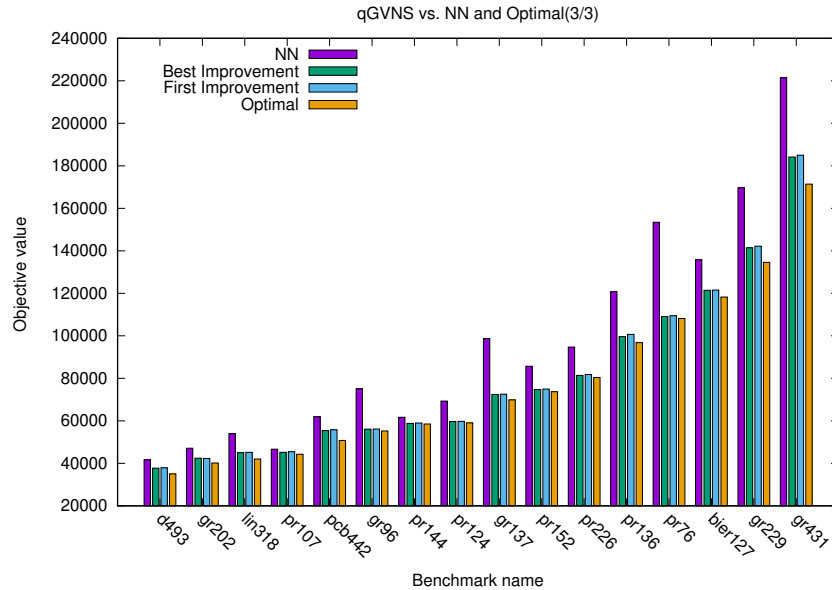


Figure 5.3: Σύγκριση μεταξύ των GVNS του πλησιέστερου γείτονα και της βέλτιστης τιμής (3/3).

Παρατηρούμε από ό,τι στα περισσότερα στιγμιότυπα αναφοράς στο Σχήμα 5.3 η GVNS

χρησιμοποιώντας την πρώτη και την καλύτερη βελτίωση πλησιάζει τη βέλτιστη τιμή, όταν, ταυτόχρονα, η ράβδος του πλησιέστερου γείτονα φαίνεται να απέχει πολύ από τη βέλτιστη τιμή. Τέτοια παραδείγματα είναι: *lin318*, *pr124*, *gr137pr76*, *bier127*.

Στο επόμενο σύνολο αριθμών (δηλ. Σχήματα 5.4, 5.5 και 5.6) κάθε σημείο αναφοράς αντιπροσωπεύεται με 5 διαφορετικές μπάρες. ένα για τον αλγόριθμο Βελτιωμένου πλησιέστερου γείτονα, ένα για τον αλγόριθμο επαναλαμβανόμενου πλησιέστερου γείτονα, ένα για τη βέλτιστη τιμή και ένα άλλο για τις δύο παραλλαγές των αλγορίθμων που έχουμε αναπτύξει (κατά καλύτερη και πρώτη βελτίωση). Μια προσεκτική εξέταση αποκαλύπτει ότι οι GVNS υλοποιήσεις βρίσκονται πάλι πολύ κοντά στις βέλτιστες τιμές, ενώ οι Βελτιωμένοι και Επαναλαμβανόμενοι Κοντινότεροι Γείτονες δεν παρέχουν εξίσου καλές λύσεις.

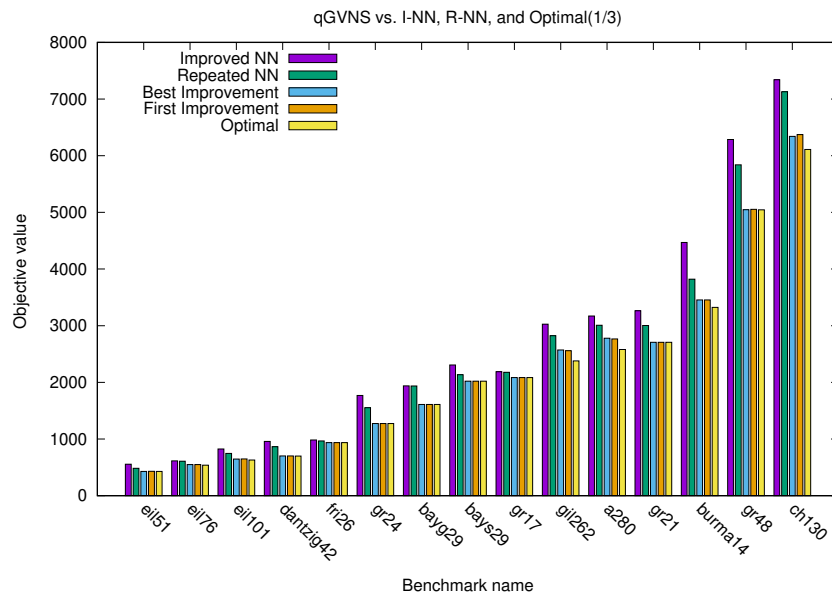


Figure 5.4: Σύγκριση μεταξύ των GVNS των παραλλαγών του πλησιέστερου γείτονα και της βέλτιστης τιμής (1/3).

Κοιτάζοντας το Σχήμα 5.4, συμπεραίνουμε ότι όταν πρόκειται για μεσαία στιγμιότυπα συγκριτικής αξιολόγησης, η GVNS προσεγγίζει σημαντικά τη βέλτιστη τιμή, σε αντίθεση με και τις δύο παραλλαγές του πλησιέστερου γείτονα που απέχουν πολύ από το βέλτιστο. Μπορούμε να το παρατηρήσουμε ιδιαίτερα στα *kroB100*, *lin105*, *kroB150* και *kroB200*.

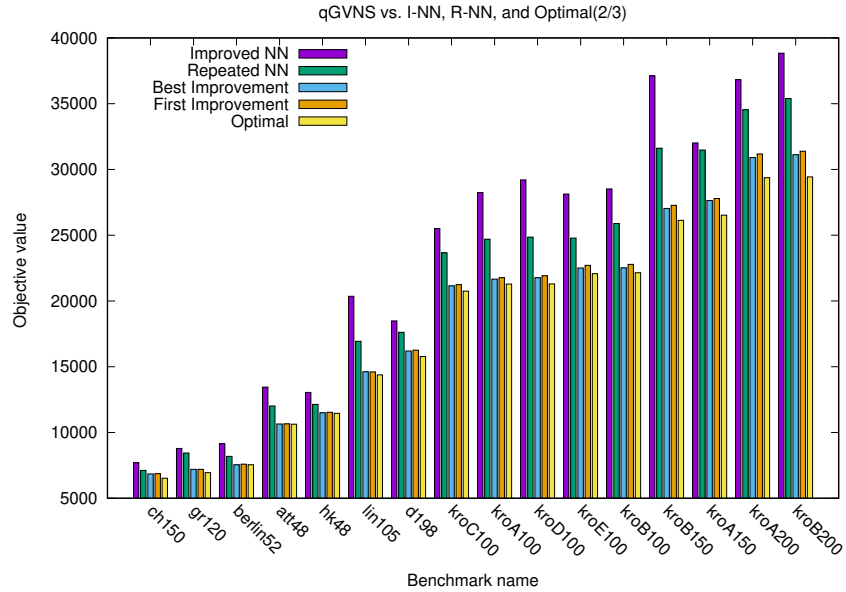


Figure 5.5: Σύγκριση μεταξύ των GVNS των παραλλαγών του πλησιέστερου γείτονα και της βέλτιστης τιμής(2/3).

Γενικά, τα σχήματα 5.4, 5.5 και 5.6 δείχνουν ότι ο βελτιωμένος πλησιέστερος γείτονας φαίνεται να είναι μια καλύτερη μέθοδος από τον επαναλαμβανόμενο πλησιέστερο γείτονα. Ωστό-σο, σε όλες τις περιπτώσεις και οι δύο παραλλαγές της GVNS ξεπερνούν τις δύο παραλλαγές του πλησιέστερου γείτονα.

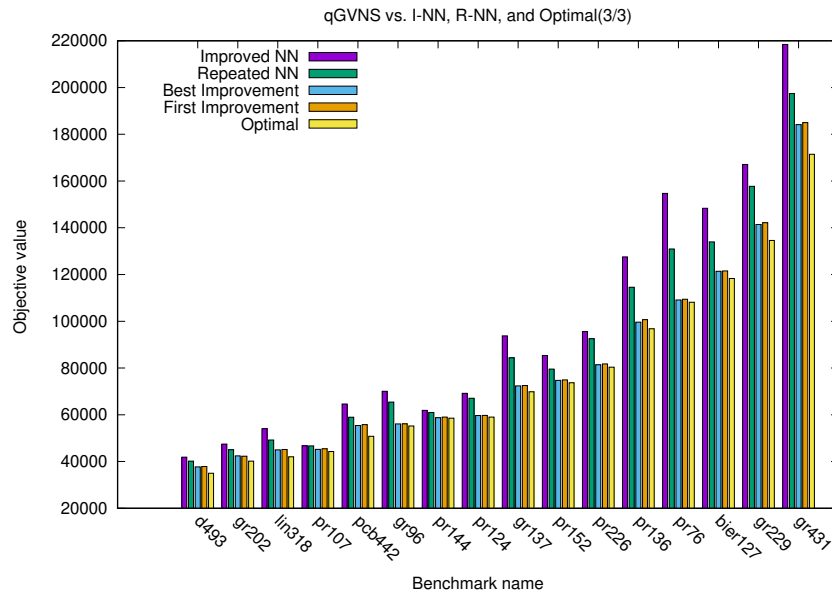


Figure 5.6: Σύγκριση μεταξύ των GVNS των παραλλαγών του πλησιέστερου γείτονα και της βέλτιστης τιμής (3/3).

Το σχήμα 5.6 δείχνει για άλλη μια φορά την ανωτερότητα της GVNS. Ας εξετάσουμε το στιγμιότυπο αναφοράς *pr76*, όπου η GVNS αγγίζει το βέλτιστο σημείο, ενώ τόσο ο επαναλαμβανόμενος όσο και ο βελτιωμένος πλησιέστερος γείτονας έχουν πολύ χαμηλότερη απόδοση. Επιπλέον, στα σχήματα 5.4, 5.5 και 5.6 μπορούμε να δούμε για ένα διαφορετικό σύνολο προβλημάτων που η GVNS ξεπερνά και πάλι τις παραλλαγές του πλησιέστερου γείτονα, πλησιάζοντας σημαντικά στη βέλτιστη τιμή σε κάθε περίπτωση.

Συνοψίζοντας, τα γραφικά αποτελέσματα μας επιτρέπουν να συμπεράνουμε ότι η GVNS με την καλύτερη και την πρώτη βελτίωση παράγει αποτελέσματα που είναι αρκετά κοντά στο βέλτιστο για τα περισσότερα μετροπρόβλήματα, ενώ ο αλγόριθμος του πλησιέστερου γείτονα και οι παραλλαγές του, απέχουν πολύ από το να είναι κοντά (για κάθε περίπτωση). Προηγουμένως, παρείχαμε τα αναλυτικά αποτελέσματα σε μορφή πίνακα. Η γραφική αναπαράσταση αυτών των αποτελεσμάτων καθιστά εύκολο να δούμε την αποτελεσματικότητα του προτεινόμενου αλγορίθμου, καθώς μπορεί κανείς να δει αμέσως ότι η απόκλιση από τη βέλτιστη είναι σχεδόν αμελητέα.

Σε αυτό το κεφάλαιο μελετήσαμε μια εφαρμογή δρομολόγησης απορριμματοφόρων χρησιμοποιώντας μια νέα μεθυστική διαδικασία βασισμένη σε μια νέα έκδοση της GVNS. Διαμορφώσαμε το υποκείμενο πρόβλημα ως στιγμιότυπο του προβλήματος του πλανόδιου πωλητή (TSP) και συνεχίσαμε να το επιλύουμε χρησιμοποιώντας την GVNS. Ο αλγόριθμος GVNS μας, διαφέρει από τις συμβατικές προσεγγίσεις λόγω της έμπνευσής του που προέρχεται από αρχές κβαντικής πληροφορικής. Η μελέτη μας επικεντρώθηκε σε γρήγορες και αποτελεσματικές μεταβάσεις σε διαφορετικές περιοχές του χώρου αναζήτησης. Αυτό μας επέτρεψε να βρούμε αποτελεσματικές διαδρομές σε σύντομο χρονικό διάστημα. Για να αξιολογήσουμε την αποτελεσματικότητα της προσέγγισής μας, πραγματοποιήσαμε εκτεταμένες πειραματικές δοκιμές χρησιμοποιώντας γνωστά προβλήματα αναφοράς από την TSPLIB. Τα αποτελέσματα ήταν αρκετά ενθαρρυντικά, καθώς επιβεβαίωσαν ότι τα νέα προτεινόμενα σχήματα της GVNS ξεπερνούν τις μεθόδους που χρησιμοποιούνται ευρέως στην πράξη, όπως το πλησιέστερο γείτονα, τον επαναλαμβανόμενος πλησιέστερο γείτονα και τον βελτιωμένο πλησιέστερο γείτονα.

5.2 Το πρόβλημα περισυλλογής απορριμμάτων ως TSPTW

Ένα σύστημα γεωπληροφόρησης (GIS) προσφέρει συνήθως διάφορα εργαλεία για την ανάλυση χωρικών δικτύων. Παρέχει κυρίως εργαλεία για την εύρεση της ελάχιστης ή μικρότερης διαδρομής μέσω ενός δικτύου, καθώς και ευρετικές διαδικασίες για την εύρεση των πιο αποτελεσματικών διαδρομών σε διάφορες τοποθεσίες. Τέτοια προβλήματα συνήθως διαμορφώνονται ως παρουσίες TSP. Αυτή η εφαρμογή επιλύει το πρόβλημα TSP με αποτελεσματικό τρόπο και βρίσκει σχεδόν τέλειες λύσεις για διάφορες περιπτώσεις αναφοράς. Οι υπολογισμοί μήτρα απόστασης επιτρέπουν τον υπολογισμό των αποστάσεων μεταξύ κόμβων που αντιπροσωπεύουν προορισμούς και προελεύσεις, ενώ οι συναρτήσεις κατανομής-θέσης εντοπίζουν τοποθεσίες τοποθεσίας, ενώ ταυτόχρονα εκχωρούν ζήτηση σε ιστότοπους. Αυτές οι δυνατότητες GIS για την ανάλυση χωρικών δικτύων διευκολύνουν τη χρήση τους ως συστήματα υποστήριξης αποφάσεων για τη δρομολόγηση οχημάτων [105]. Το επίκεντρο αυτής της εργασίας είναι η εύρεση σχεδόν βέλτιστων λύσεων για το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα (TSPTW). Συγκεκριμένα, θα αντιμετωπίσουμε το ζήτημα της δρομολόγησης απορριμμάτων οχημάτων με χρονικά παράθυρα για τη συλλογή στερεών αποβλήτων. Αυτή η εργασία είναι εμπνευσμένη από την προηγούμενη προσπάθειά μας να αντιμετωπίσουμε το γενικό σενάριο συλλογής απορριμμάτων, μοντελοποιώντας το ως πρόβλημα πλανόδιου πωλητή [151]. Καθώς δεν αναμένεται αλγόριθμος πολυωνύμου χρόνου για την εξεύρεση λύσεων σε οποιαδήποτε μορφή του TSP, οι Κέναν κ.α. (Keenan et al.)

[105] υποστηρίζουν ότι μία από τις πιο δημοφιλείς προσεγγίσεις είναι οι ευρετικές προσεγγίσεις. Υπάρχουν διαφορές μεταξύ ευρετικών από άποψη απόδοσης, δυσκολίας υλοποίησης και χρόνου υπολογισμού.

Για να δρομολογήσετε ένα απορριματοφόρο από την αποθήκη προς όλους τους κάδους απορριμμάτων και να το επαναδρομολογήσετε πίσω, είναι απαραίτητο να βρείτε τις αντίστοιχες διαδρομές του υποκείμενου TSP. Το GIS θα διαδραματίσει κρίσιμο ρόλο στην εύρεση σχεδόν βέλτιστων διαδρομών που προσεγγίζουν τον ελάχιστο χρόνο που απαιτείται για τη μεταφορά. Το GIS θα μπορούσε επίσης να κατευθύνει το πρόγραμμα οδήγησης στις παραγόμενες διαδρομές. Η μετάδοση των οδηγιών θα γίνει στο σκουπίδια. Το πώς επιτυγχάνεται η δρομολόγηση σε σχέση με τους χρονικούς περιορισμούς είναι πολύ σημαντικό σε αυτό το σύστημα σε πραγματικό χρόνο, και αυτό θα μπορούσε να διασφαλιστεί από τη μεταεπισκόπηση. Σε αυτό το άρθρο, δίνουμε μια περιγραφή ενός συστήματος που επιλύει το πρόβλημα διαχείρισης δρομολόγησης απορριμμάτων. Το βασικό συστατικό αυτής της λύσης είναι το κβαντικό εμπνευσμένο μεθευρετικό που εφαρμόζεται στο πρόβλημα του πλανόδιου πωλητή (TSP) και είναι ενσωματωμένο στο GPS / GIS. Η προσέγγιση που χρησιμοποιείται είναι μια λύση βασισμένη στην ολοκληρωμένη διαχείριση προβλημάτων. Λαμβάνοντας υπόψη τις λειτουργικές απαιτήσεις και τις σχετικές μελέτες (όπως ο Ασημα-κόπουλος κ.ά). Για κάθε υποσύστημα, καθορίζονται οι λειτουργίες του και οι σχέσεις του με τα άλλα υποσυστήματα.

Οι Αισθητήρες Κάδου είναι ο εξοπλισμός που παρέχει εκτίμηση του επιπέδου πλήρωσης απορριμμάτων. Αυτό είναι κρίσιμο για τη συλλογή, αποθήκευση και μετάδοση δεδομένων σχετικά με κάθε κάδο. Δεύτερον, η συλλογή δεδομένων από κάδους ορίζεται ως μονάδα επικοινωνίας με αισθητήρες κάδου. Η κύρια ιδέα είναι να παραδοθούν οι πληροφορίες που έχουν συλλεχθεί από τους αισθητήρες στο κεντρικό σύστημα. Υπάρχουν τρία εξαρτήματα που μπορούν να εγκατασταθούν σε οχήματα. Το πρώτο συστατικό είναι ο υπεύθυνος επικοινωνίας για την επικοινωνία με τον κάδο, το δεύτερο είναι η διαδικασία αποθήκευσης που αποθηκεύει δεδομένα έως ότου μεταφερθούν στο κεντρικό σύστημα και το τρίτο είναι η διαδικασία μεταφοράς που ξεδιπλώνει δεδομένα από το κεντρικό σύστημα.

Το παγκόσμιο σύστημα δικτύου κινητής επικοινωνίας θα χρησιμοποιηθεί για την εφαρμογή όλων αυτών των λειτουργιών: εφαρμογές πλοήγησης GPS ενσωματωμένες σε φορτηγά, ένα κεντρικό σύστημα, μια υποδομή χάρτη, ένα υποσύστημα εξόρυξης δεδομένων και μια δομή βελτιστοποίησης δρομολόγησης. Το κεντρικό σύστημα είναι το σύστημα υποστήριξης της περιπτώσιολογικής μας μελέτης. Όλες οι χωρικές πληροφορίες που δημιουργούνται για τη διαδρομή θα αποθηκευτούν εκεί. Επίσης, η αποθήκευση δεδομένων συνεπάγεται πληροφορίες που λαμβάνονται / ανακτώνται από άλλα υποσυστήματα. Ένα υποσύστημα εξόρυξης δεδομένων χρησιμοποιείται για την εκτίμηση της πληρότητας του κάδου όταν υπάρχει έλλειψη ενημερωμένων πληροφοριών. Αυτό το έργο υποστηρίζει μια δομή βελτιστοποίησης δρομολόγησης ως σημαντικό συστατικό που μπορεί να προσφέρει βέλτιστες ή σχεδόν βέλτιστες λύσεις σε σύντομο χρονικό διάστημα.

Μέσω της εφαρμογής δυνατοτήτων δρομολόγησης, παρατηρείται ένα συγκριτικό πλεονέκτημα σε σχέση με άλλες παρόμοιες προσπάθειες. Δεδομένου ότι αυτό το έργο χρησιμοποιεί ένα σύστημα σε πραγματικό χρόνο, θα πρέπει να έχει την ικανότητα να παρέχει άμεσες απαντήσεις σε τέτοια προβλήματα, δεδομένου ότι η περίοδος απόκρισης είναι αρκετά σημαντική. Μέσω της ανάπτυξης τοπικών δομών αναζήτησης καθώς και του GVNS για το πρόβλημα, αναμένονται καλύτερα αποτελέσματα.

Σε αυτό το σενάριο που αφορά τα χρονικά διαστήματα, κάθε κάδος σκουπιδιών πρέπει να είναι επισκέψιμο μέσα σε ένα συγκεκριμένο χρονικό πλαίσιο. Η πειραματική σύγκριση επιβεβαιώνει την υπόθεση ότι το GVNS οδηγεί σε αρκετά ελπιδοφόρα αποτελέσματα. Λόγω αυτών των

βελτιωμένων λύσεων δρομολόγησης, χρησιμοποιείται χαμηλότερη ποσότητα καυσίμου. Κατά συνέπεια, η προτεινόμενη μέθοδος είναι επίσης περιβαλλοντική.

5.2.1 Η προτεινόμενη μέθοδος

Το πρόβλημα της περισυλλογής απορριμμάτων γίνεται πιο περίπλοκο όταν επιτρέπεται στα απορριμματοφόρα να έχουν πρόσβαση σε συγκεκριμένους κάδους μόνο για συγκεκριμένα χρονικά διαστήματα κατά τη διάρκεια της ημέρας. Ένα τυπικό παράδειγμα αυτής της κατάστασης συμβαίνει στην Κέρκυρα που βρίσκεται στην Ελλάδα. Αυτή η δυσκολία προκαλείται από την τουριστική περίοδο και τους περιοδικούς κανονισμούς κυκλοφορίας που επιβάλλονται σε ορισμένους δρόμους για συγκεκριμένες ώρες κατά τη διάρκεια της ημέρας. Επιπλέον, η παλιά πόλη της Κέρκυρας είναι ένα μνημείο της ΟΥΝΕΣΚΟ που χαρακτηρίζεται από μεγάλη ιδιαιτερότητα όσον αφορά τον χωροταξικό σχεδιασμό και την προσβασιμότητα. Μοντελοποιώντας πιστά αυτήν την κατάσταση, μετατρέπει το κλασικό πρόβλημα περισυλλογής απορριμμάτων σε πρόβλημα περισυλλογής απορριμμάτων με τα χρονικά παράθυρα. Αυτό το σενάριο, από αλγοριθμική άποψη, μοντελοποιείται επαρκώς ως το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα (Traveling Salesman Problem with Time Windows, TSPTW).

Στο υπόλοιπο κείμενο περιγράφουμε την προσέγγιση που υιοθετήσαμε για την αντιμετώπιση του προβλήματος, χρησιμοποιώντας τη μεθευρετική Αναζήτηση Μεταβλητής Γειτονιάς (VNS). Για μια ολοκληρωμένη αξιολόγηση παρουσιάζουμε τόσο μια κλασική προσέγγιση όσο και μια κβαντικά εμπνευσμένη διαδικασία για την επίλυση αυτού του προβλήματος.

Αυτή η εργασία μελετά την υλοποίηση μιας μετευρετικής GVNS για το TSPTW που εμπνέεται από τη μέθοδο που περιγράφεται στο [38]. Εκτός από την προαναφερθείσα συμβατική προσέγγιση, προτείνουμε ένα μη συμβατικό σχήμα GVNS που επιλύει το TSP με Time Windows χρησιμοποιώντας αρχές κβαντικής έμπνευσης για τη λειτουργία διαταραχής. Για αυτές τις δύο ξεχωριστές υλοποιήσεις ακολουθήσαμε επίσης τα [128] και [148]. Σε αυτήν την παράγραφο θα συζητηθεί η διαδικασία σχεδιασμού και εφαρμογής. Η επισκόπηση που εξηγείται εδώ χρησιμοποιεί την τεχνική που περιγράφεται στο [38] που σχεδιάστηκε για το TSP με χρονικά παράθυρα.

Algorithm 10 Ο ψευδοκώδικας της μεθόδου της μας [150]

```

1: procedure V(N)S for finding initial solution
2:   Input: an unfeasible solution X
3:   Output a feasible solution X
4:   Initialization of feasibility distance matrix
5:   while the solution is not feasible do
6:     X < Construct an initial solution()
7:     level < 1
8:     X < LocalSearch(X)
9:     while S do solution is not feasible and level <= 8
10:      X' < Perturbation(X,level)
11:      X'' < LocalSearch (X')
12:      if X then'' is better than X then
13:        X < X''
14:        level < 1
15:      else
16:        level < level + 1
17:      end if
18:    end while
19:  end while
20: end procedure

```

Η μέθοδος στο Σχήμα 10 είναι η κύρια και εστιάζει στην εύρεση μιας εφικτής λύσης χρησιμοποιώντας την GVNS μέθοδο. Πρώτα απ' όλα, στη γραμμή 1 δημιουργούμε έναν πίνακα που δείχνει (πριν από την έναρξη του αλγορίθμου) ποιοι κόμβοι είναι προσβάσιμοι ο ένας από τον άλλον και ποιοι όχι, λαμβάνοντας υπόψη τους περιορισμούς που επιβάλλονται από τα χρονικά παράθυρα και ιδιαίτερα τον χρόνο λήξης του κόμβου που επισκέπτεται. Ο βασικός αλγόριθμος χρησιμοποιεί αυτήν τη μέθοδο για να αποκλείσει τη δυνατότητα ελέγχου διαδρομών που δεν ικανοποιούν τους χρονικούς περιορισμούς. Η μέθοδος από τη γραμμή 2 έως το τέλος επαναλαμβάνεται έως ότου βρεθεί μια εφικτή λύση. Στην τρίτη σειρά παράγεται μια τυχαία ή ταξινομημένη λύση. Το επίπεδο(μέτρο) της διαταραχής έχει οριστεί σε 1 στη γραμμή 4. Το μέτρο αυτό ελέγχει τον αριθμό των επαναλήψεων της λειτουργίας διαταραχής. Η λειτουργία διαταραχής είναι υπεύθυνη για τη διαφυγή της τρέχουσας λύσης από τοπικά ελάχιστα που πιθανόν κατά την διάρκεια της αναζήτησης να εγκλωβιστεί, και την αποδοχή της παραχθείσας λύσης. Το υψηλότερο επίπεδο του μέτρου της διαταραχής είναι 8 επειδή έχει πειραματικά προσδιοριστεί ότι αυτή είναι η καλύτερη τιμή [38]. Η πρώτη τοπική αναζήτηση πραγματοποιείται πριν ξεκινήσει η εκτέλεση του βρόχου σε περίπτωση που το πρόβλημα μπορεί να επιλυθεί αμέσως. Ο κύριος βρόχος εκτελείται ενώ το μεταβλητό μέτρο της διαταραχής είναι μικρότερο από 8 και δεν έχει βρεθεί αποδεκτή λύση. Μέσα στο βρόχο, μια διαταραγμένη λύση προσδιορίζεται με βάση την τιμή του επιπέδου της διαταραχής. Σε αυτό το σημείο, εάν η νέα λύση που κατασκευάζεται μετά από μια τοπική αναζήτηση, είναι καλύτερη από την παρούσα, την ορίζουμε ως η παρούσα προτιμώμενη λύση. Εάν ανακαλυφθεί μια νέα καλύτερη λύση, θα ορίσουμε το επίπεδο σε 1. Εάν η λύση μετά την τοπική αναζήτηση δεν είναι καλύτερη από την παρούσα, τότε αυξάνουμε την τιμή της παραμέτρου του μέτρου της διαταραχής και προχωράμε στον επόμενο κύκλο. Τονίζουμε ότι η καλύτερη λύση είναι αυτή με τον μικρότερο αριθμό παραβιασμένων κόμβων. Εάν δύο λύσεις έχουν τον ίδιο αριθμό παραβιασμένων κόμβων,

διατηρούμε την καλύτερη λύση, αυτή με τη χαμηλότερη αντικειμενική τιμή. Υπενθυμίζουμε ότι ένας κόμβος παραβιάζεται εάν έχει επισκεφθεί μετά την ώρα λήξης του χρονικού παραθύρου του κόμβου. Οι μεμονωμένες υπορουτίνες της GVNS περιγράφονται παρακάτω.

Το σχήμα 5.7 απεικονίζει ένα απλό εικονογραφημένο παράδειγμα ενός προβλήματος του πλανόδιου πωλητή με χρονικά παράθυρα (TSPTW) που δείχνει με συγκεκριμένες τιμές δεδομένων την απλή μοντελοποίηση του προβλήματος.

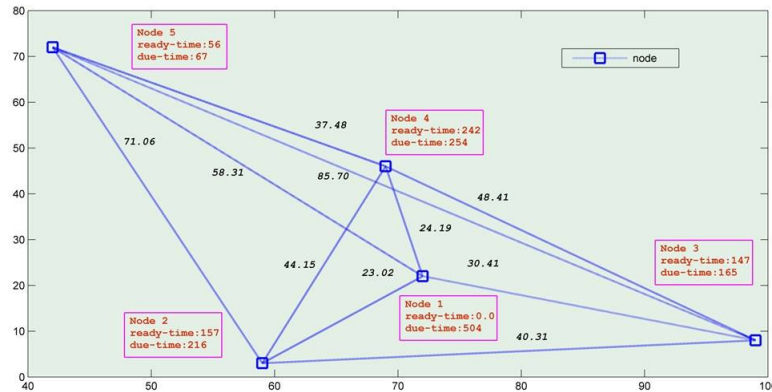


Figure 5.7: Μια απλή γραφική αναπαράσταση του προβλήματος του πλανόδιου πωλητή με χρονικά παράθυρα με 5 κόμβους [150].

5.2.2 Διαταραχή, Δομές γειτνίασης

Η τοπική αναζήτηση πραγματοποιείται χρησιμοποιώντας μια μέθοδο 1-shift ως δομή γειτονιάς. Αυτή η προσέγγιση αποσκοπεί στη βελτίωση της τρέχουσας λύσης μετακινώντας μερικούς κόμβους μαζί με άλλους στη σειρά. Εάν η τρέχουσα λύση δεν είναι εφικτή, εμφανίζονται δύο τύποι κόμβων: οι παραβιασμένοι κόμβοι (αυτοί που επισκέφτηκαν μετά την προθεσμία) και οι μη παραβιασμένοι κόμβοι (αυτοί οι κόμβοι που επισκέφτηκαν πριν από την προθεσμία). Κατά συνέπεια, η κύρια γειτονιά μπορεί να χωριστεί σε δύο νέες υπο-γειτονιές. Στην πρώτη γειτονιά, οι κινήσεις αποτελούνται από παραβιασμένους κόμβους και στη δεύτερη γειτονιά από μη παραβιασμένους κόμβους. Η ύπαρξη δύο γειτονιών συνεπάγεται ότι πρέπει να δοθεί προσοχή στη σειρά της εξερεύνησης τους. Η βέλτιστη σειρά έρευνας γειτονιάς δίνεται παρακάτω [38]:

- Μετακινήστε προς τα πίσω τους παραβιασμένους κόμβους προς τα πίσω,
- Μετακινήστε τους κόμβους που δεν έχουν παραβιαστεί,
- Μετακινήστε τους μη παραβιασμένους κόμβους προς τα πίσω και
- Μετακίνηση προς τα εμπρός των παραβιασμένων κόμβων.

Αυτή η σειρά αποδίδει καλύτερα από άλλες, όπου ο όρος "καλύτερο" ερμηνεύεται ως ο συνδυασμός γειτονιών που παράγει μια εφικτή λύση πιο γρήγορα. Η διαδικασία λειτουργίας της διαταραχής χρησιμοποιείται ως τοπική διαδικασία μιας συνάρτησης αποφυγής της παγίδευσης σε τοπικό ελάχιστο. Στην εκτέλεσή μας, ο αλγόριθμος πραγματοποιεί τυχαίες μετατοπίσεις κόμβων (1-shift) σε μερικούς τυχαία επιλεγμένους κόμβους στην παρούσα λύση χωρίς να λαμβάνει υπόψη οποιαδήποτε συνθήκη ελέγχου για να προχωρήσει σε μια καλύτερη λύση [150].

5.2.3 Υπολογιστικά αποτελέσματα

Η υλοποίησή μας πραγματοποιήθηκε στην γλώσσα προγραμματισμού JAVA. Ένας σημαντικός λόγος για την επιλογή της JAVA, είναι ότι από όσο γνωρίζουμε δεν υπάρχει αντικειμενοσ-

τρεφής εφαρμογή για αυτό το συγκεκριμένο πρόβλημα βελτιστοποίησης. Επιπλέον, η JAVA είναι μια πολύ γνωστή γλώσσα, επομένως ο αλγόριθμος μας θα είναι προσβάσιμος σε ένα ευρύ κοινό.

Δοκιμάσαμε και τις τρεις διαφορετικές παραλλαγές, συμπεριλαμβανομένων συμμετρικών στιγμοτύπων, με διαφορετικά προβλήματα αναφοράς που προτείνονται από τους [26] και [142]. Το όνομα κάθε προβλήματος αναφοράς έχει την ακόλουθη δομή. γράμμα *n* (δηλαδή, αριθμός των κόμβων), ακολουθούμενο από πραγματικό αριθμό πελατών. Στη συνέχεια, το γράμμα *w* (δηλαδή, μέγιστο πλάτος παραθύρου) ακολουθούμενο από την υψηλότερη τιμή μήκους παραθύρου. Το όνομα αρχείου *n20w100* κωδικοποιεί πρόβλημα με 20 πελάτες και μέγιστο μήκος παραθύρου 100.

Σε αυτήν την ενότητα παρουσιάζουμε την εκτενή συγκριτική μελέτη τριών διαφορετικών παραλλαγών της μεθευρετικής **GVNS**. Η πρώτη ξεκινά με μια τυχαία λύση, ενώ το κύριο σώμα αυτής της διαδικασίας προσπαθεί να βελτιώσει τη λύση μας μέσω τεσσάρων διαφορετικών παραλλαγών ευρετικής *1-shift*. Η δεύτερη υλοποίηση ξεκινά με μια ταξινομημένη λύση ενώ χρησιμοποιεί την ίδια ευρετική *1-shift* (για άλλη μια φορά παράγοντας τέσσερις διαφορετικές κινήσεις χρησιμοποιώντας την ευρετική *1-shift*). Το τρίτο ξεκινά με μια χβαντική εμπνευσμένη μέθοδο στην αρχική είσοδο, ενώ η φάση βελτίωσης εξακολουθεί να βασίζεται στην ίδια *1-shift*. Η μέθοδος διαταραχής εμπνέεται από τον χβαντικό υπολογισμό.

Παρακάτω, παρουσιάζουμε τα υπολογιστικά αποτελέσματα για τις τρεις ξεχωριστές παραλλαγές του σχήματος **GVNS** σε ορισμένα γνωστά σημεία αναφοράς. Συγκεκριμένα, τα υπολογιστικά αποτελέσματα παρουσιάζονται σε δύο πίνακες, τον πίνακα 5.7 και τον πίνακα 5.8. Μια συγκριτική μελέτη της παραλλαγής **RSC GVNS** και του **QISC** παρουσιάζεται στον Πίνακα 5.7 και μια παρόμοια σύγκριση μεταξύ του συμβατικού **GVNS SSC** και του **QISC** δίνεται στον Πίνακα 5.8. Τα αποτελέσματα **QISC** και στους δύο πίνακες Ο πίνακας 5.7 και ο πίνακας 5.8 εμφανίζονται με έντονη γραφή, καθώς θέλαμε να τονίσουμε την ανωτερότητα της χβαντικής εμπνευσμένης παραλλαγής για συγκεκριμένες περιπτώσεις των σημείων αναφοράς. Για το σκοπό αυτό, επισημαίνουμε στον Πίνακα 5.7 ότι σε πέντε στα δέκα συνολικά σημεία αναφοράς το **QISC** παράγει καλύτερο αποτέλεσμα από το **RSC**, ενώ στον Πίνακα 5.8 το **QISC** ξεπερνά το **SSC** σε εννέα στα δέκα στα συνολικά σημεία αναφοράς.

Table 5.7: Υπολογιστικά Αποτελέσματα- RSC vs. QISC υπολογιστική μελέτη.

Όνομα	N	Max WL	Best Sol.	Προτάθηκε από	RSC	dif. from best sol.	QISC	dif. from best sol.
n20w120	20	120	265.6	Ohlmann and Thomas (2007)	357.49	34%	354.03	33%
n20w140	20	140	232.8	Ohlmann and Thomas (2007)	337.05	44%	306.88	32%
n20w160	20	160	218.2	Ohlmann and Thomas (2007)	331.00	51%	390.72	79%
n20w180	20	180	236.6	Ohlmann and Thomas (2007)	336.58	42%	365.64	54%
n20w200	20	200	241.0	Ohlmann and Thomas (2007)	365.87	51%	365.36	51%
n40w120	40	120	360.0	Calvo (2000)	521.18	44%	551.84	53%
n40w140	40	140	348.4	Calvo (2000)	493.35	41%	518.36	49%
n40w160	40	160	326.8	Ohlmann and Thomas (2007)	482.20	47%	552.24	69%
n40w180	40	180	326.8	Calvo (2000)	532.49	62%	519.42	59%
n40w200	40	200	313.8	Ohlmann and Thomas (2007)	493.65	57%	468.01	49%

Table 5.8: Υπολογιστική Αποτελέσματα - SSC vs. QISC υπολογιστική μελέτη.

Όνομα	N	Max WL	Best Sol.	Προτάθηκε από	SSC	dif. from best sol.	QISC	dif. from best sol.
n20w120	20	120	265.6	Ohlmann and Thomas (2007)	371.78	39%	354.03	33%
n20w140	20	140	232.8	Ohlmann and Thomas (2007)	370.61	59%	306.88	32%
n20w160	20	160	218.2	Ohlmann and Thomas (2007)	398.08	82%	390.72	79%
n20w180	20	180	236.6	Ohlmann and Thomas (2007)	418.13	76%	365.64	54%
n20w200	20	200	241.0	Ohlmann and Thomas (2007)	414.70	72%	365.36	51%
n40w120	40	120	360.0	Calvo (2000)	549.97	52%	551.84	53%
n40w140	40	140	348.4	Calvo (2000)	563.83	61%	518.36	49%
n40w160	40	160	326.8	Ohlmann and Thomas (2007)	553.81	69%	552.24	69%
n40w180	40	180	326.8	Calvo (2000)	561.09	71%	519.42	59%
n40w200	40	200	313.8	Ohlmann and Thomas (2007)	558.65	78%	468.01	49%

Table 5.9: Συντομογραφίες για τον πίνακα II

Συντομογραφία (στα αγγλικά)	Ολογράφως (στα ελληνικά)
N	ο αριθμός των κόμβων
Max WL	το μέγιστο μέγεθος του χρονικού παραθύρου
Best Sol	η βέλτιστη τιμή
RSC	το κόστος της τυχαίας λύσης
RST	ο χρόνος της τυχαίας λύσης
dif. from best sol.	διαφορά από την βέλτιστη τιμή
SSC	το κόστος της ταξινομημένης λύσης
SST	ο χρόνος της ταξινομημένης λύσης
QISC	το κόστος της χβαντικά εμπνευσμένης λύσης

5.2.4 Η μελέτη περίπτωσης της Κέρκυρας

Μια πρακτική μελέτη περίπτωσης όπου μπορεί να εφαρμοστεί το πρόβλημά της περισυλλογής απορριμμάτων με χρονικά παράθυρα, είναι η μελέτη περίπτωσης της πόλης της Κέρκυρας. Η Κέρκυρα είναι ιδιαίτερα γεμάτη κατά τη θερινή περίοδο λόγω της εξαιρετικά υψηλής τουριστικής δραστηριότητας. Κατά συνέπεια, πολλοί δρόμοι που πρόκειται να χρησιμοποιηθούν από οχήματα είναι κλειστοί και επιτρέπεται μόνο στους πεζούς να περάσουν για ένα συγκεκριμένο χρονικό διάστημα. Έτσι, αυτοί οι δρόμοι έχουν έτοιμο και κατάλληλο χρόνο για να έχουν πρόσβαση τα απορριμματοφόρα. Αυτό το πρόβλημα μπορεί να μοντελοποιηθεί ως πρόβλημα περισυλλογής απορριμμάτων με χρονικά παράθυρα και μπορεί να αντιμετωπιστεί με την επίλυση του προβλήματος του πλανόδιου πωλητή με χρονικά παράθυρα.

Κατασκευάσαμε το σύνολο δεδομένων, το οποίο χρησιμεύει ως στιγμιότυπο αναφοράς πραγματικού προβλήματος, με πληροφορίες διαδρομής από την Κέρκυρα και με πληροφορίες δρομολόγησης των απορριμμάτων από το σταθμό ανακύκλωσης σε κάθε θέση κάδου και πίσω στο σταθμό ανακύκλωσης (αποθήκη), χρησιμοποιώντας το API κατεύθυνσης της Google. Το API οδηγίων [47] είναι μια υπηρεσία που υπολογίζει τις αποστάσεις μεταξύ τοποθεσιών χρησιμοποιώντας ένα αίτημα HTTP. Αυτό το API παρέχει πολλές επιλογές χρήσης στον χρήστη. Το On μπορεί να ζητήσει οδηγίες για διάφορους τρόπους μεταφοράς, όπως διέλευση, οδήγηση, περπάτημα ή ποδηλασία. Επιπλέον, μπορεί να επιστρέψει πολλαπλές κατευθύνσεις χρησιμοποιώντας μια σειρά σημείων. Μπορεί να καθορίσει αφετηρίες, προορισμούς και σημεία ως συμβολοσειρές κειμένου ή ως συντεταγμένες γεωγραφικού πλάτους/μήκους ή ως αναγνωριστικά θέσης.

Έχουμε χρησιμοποιήσει όλες τις παραπάνω πληροφορίες για να φτιάξουμε την περιήγηση ενός απορριμματοφόρων που έχει ως αποθήκη το κέντρο ανακύκλωσης της πόλης της Κέρκυρας και μια περιήγηση με κόμβους σε μερικά από τα πιο σημαντικά μέρη όπου βρίσκονται οι κάδοι ανακύκλωσης. Για να ολοκληρώσουμε το παράδειγμά μας, έχουμε προσθέσει χρονικούς περιορισμούς σχετικά με τον χρόνο επίσκεψης κάθε κόμβου, όπως απαιτείται, προκειμένου να προσομοιώσουμε πλήρως αυτό το πρόβλημα. Έχουμε αντιμετωπίσει αυτό το νέο πραγματικό πρόβλημα χρησιμοποιώντας την χβαντική εμπνευσμένη έκδοση της GVNS, καθώς φαίνεται να παρέχει καλύτερες λύσεις από τις άλλες δύο συμβατικές παραλλαγές.

Το σημείο αναφοράς μας αποτελείται από 21 κόμβους, μια αποθήκη (όπου βρίσκεται το κέντρο ανακύκλωσης) και 20 κόμβους όπου τοποθετούνται οι κάδοι ανακύκλωσης. Οι συντεταγμένες κάθε κόμβου παρατίθενται στον Πίνακα 5.10.

Table 5.10

Κόμβος	Γεωγραφικό μήκος	Γεωγραφικό πλάτος	Σχόλιο
1	39.6422466	19.8223686	η αφετηρία
2	39.6404733	19.8229525	ένας κόμβος/κάδος
3	39.6413691	19.8306238	ένας κόμβος/κάδος
4	39.6407922	19.8370158	ένας κόμβος/κάδος
5	39.6374761	19.8386927	ένας κόμβος/κάδος
6	39.6329728	19.8526526	ένας κόμβος/κάδος
7	39.6308208	19.8660169	ένας κόμβος/κάδος
8	39.6246663	19.8778979	ένας κόμβος/κάδος
9	39.6234445	19.8863486	ένας κόμβος/κάδος
10	39.6247393	19.8899176	ένας κόμβος/κάδος
11	39.6284544	19.8880917	ένας κόμβος/κάδος
12	39.6297152	19.8887043	ένας κόμβος/κάδος
13	39.6257976	19.9083606	ένας κόμβος/κάδος
14	39.625156	19.9085661	ένας κόμβος/κάδος
15	39.6243003	19.9219708	ένας κόμβος/κάδος
16	39.6256647	19.9108012	ένας κόμβος/κάδος
17	39.6262482	19.9179632	ένας κόμβος/κάδος
18	39.6251191	19.9187309	ένας κόμβος/κάδος
19	39.6254352	19.9195927	ένας κόμβος/κάδος
20	39.6246342	19.9219791	ένας κόμβος/κάδος
21	39.6243003	19.9219708	ένας κόμβος/κάδος

Εκτός από τον παραπάνω Πίνακα 5.10, παρουσιάζουμε την ακόλουθη γραφική απεικόνιση στο Σχήμα 5.8, που εξάγεται από χάρτες, δείχνοντας πού βρίσκονται οι αποθήκες και οι άλλοι κόμβοι, ώστε να επεξηγηθεί καλύτερα το συγκεκριμένο πρόβλημα.

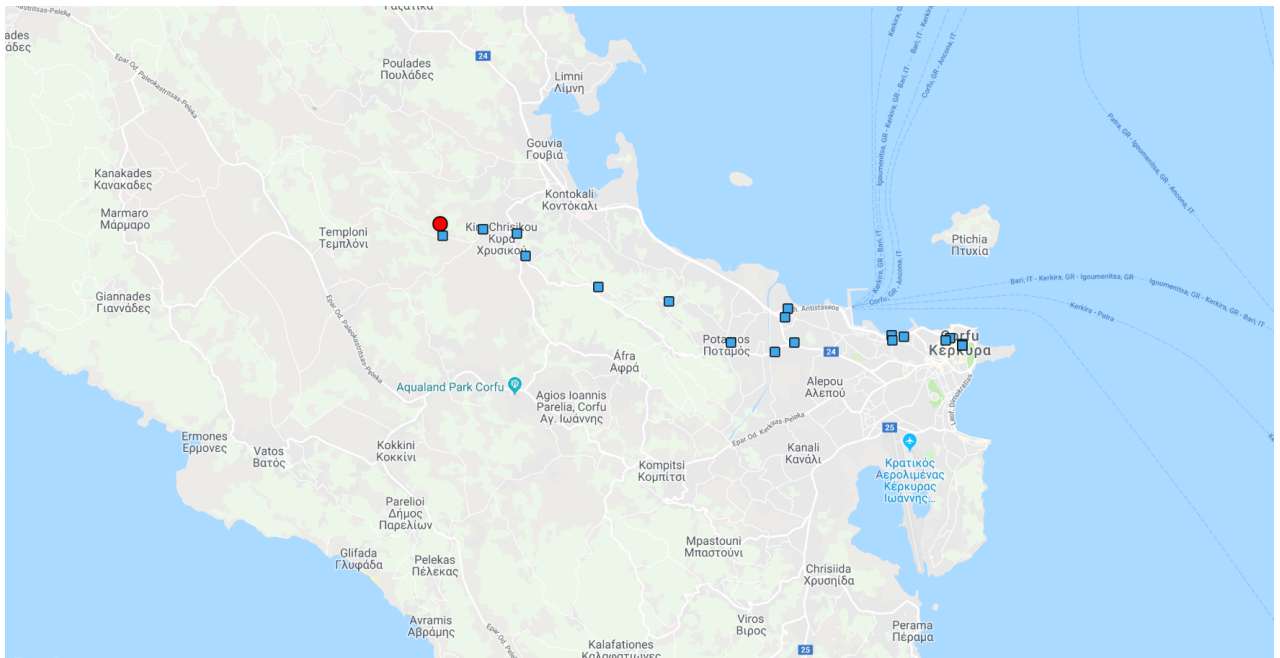


Figure 5.8: Η μελέτη περίπτωσης ανακύκλωσης απορριμμάτων της Κέρκυρας.

Χρησιμοποιώντας αυτό το στιγμιότυπο αναφοράς που δημιουργήσαμε με πραγματικά δεδομένα για την εφαρμογή του χβαντικά εμπνευσμένου αλγόριθμου που υλοποιήσαμε, βρήκαμε μια εφικτή λύση για αυτό το στιγμιότυπο αναφοράς μας. Η σειρά των κόμβων μετά την εκτέλεση είναι η ακόλουθη: [0,6,2,11,19,3,4,8, 5,14,15,10,9,7,20,13,18,16,12,17,1], και το προκύπτον κόστος της παραγόμενης λύσης είναι 319,0. Τα παραπάνω αποτελέσματα είναι αρκετά ελπιδοφόρα και μας ενθαρρύνουν να εφαρμόσουμε την έρευνά μας για εναλλακτικά μοντέλα βελτιστοποίησης και σε άλλα πραγματικά προβλήματα.

Στο συγκεκριμένο κεφάλαιο μελετήσαμε το πρόβλημα της περισυλλογής απορριμμάτων με επιπλέον περιορισμούς στο χρόνο που κάθε κόμβος είναι διαθέσιμος για επίσκεψη από το απορριματοφόρο. Η προτεινομένη μέθοδος είναι μια μεθυστική που βασίζεται σε μια νέα παραλλαγή της *GVNS* που εμπνέεται από τον χβαντικό υπολογισμό. Το πρόβλημα διαμορφώθηκε σαν το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα και επιλύθηκε χρησιμοποιώντας μια παραλλαγή της γενικής αναζήτησης μεταβλητής γειτονιάς. Λόγω της χρησιμοποίησης των μεθόδων χβαντικού υπολογισμού, η μεθυστική μας διαφέρει από τις συμβατικές προσεγγίσεις. Προκειμένου να διεξαχθεί μια αξιολόγηση της αποτελεσματικότητας της προσέγγισης, εκτελέστηκαν εκτεταμένες πειραματικές δοκιμές χρησιμοποιώντας γνωστά μετροπροβλήματα αλλά και δημιουργώντας τα δικά μας με δεδομένα από τον πραγματικό κόσμο.



6. Κβαντικός υπολογισμός

6.1 Ο Κβαντικός υπολογισμός

Η κβαντική πληροφορική είναι ο τομέας που επικεντρώνεται στην ανάπτυξη της τεχνολογίας των υπολογιστών με βάση τις αρχές της κβαντικής θεωρίας, η οποία εξηγεί τη φύση και τη συμπεριφορά της ενέργειας και της ύλης στο κβαντικό (ατομικό και υποατομικό) επίπεδο. Η ανάπτυξη ενός κβαντικού υπολογιστή, αν είναι πρακτικό και εφικτό, θα σηματοδοτήσει ένα άλμα προς τα εμπρός στην υπολογιστική ικανότητα, πολύ μεγαλύτερη από εκείνη ενός συμβατικού υπολογιστή, με πολύ μεγάλη υπολογιστική υπεροχή. Ο κβαντικός υπολογιστής, που ακολουθεί τους νόμους της κβαντικής φυσικής, θα εξασφαλίζει τεράστια επεξεργαστική ισχύ μέσω της ικανότητας του να είναι σε πολλαπλές καταστάσεις ταυτόχρονα, που ως αποτέλεσμα θα έχει να εκτελεί εργασίες χρησιμοποιώντας όλες τις πιθανές καταστάσεις ταυτόχρονα. Τα μεγαλύτερα και σημαντικότερα κέντρα ερευνών στον κβαντικό υπολογισμό περιλαμβάνουν το MIT, την IBM, το Πανεπιστήμιο της Οξφόρδης και το εθνικό εργαστήριο του Λος Άλαμος. Τα ουσιαστικά στοιχεία της κβαντικής πληροφορικής προέρχονταν από τον Πολ Μπένιοφ (Paul Beniof) [14], που εργαζόταν στα εθνικά εργαστήρια Argonne, το 1981. Είχε μια θεωρία για έναν κλασικό υπολογιστή που λειτουργούσε με κβαντικές μηχανικές αρχές. Αλλά είναι γενικά αποδεκτό ότι ο Ντβιντ Ντοϊτς (David Deutsch) του Πανεπιστημίου της Οξφόρδης παρείχε την κρίσιμη ώθηση για την έρευνα της κβαντικής πληροφορικής. Το 1984, βρισκόταν σε ένα συνέδριο θεωρίας υπολογισμού και άρχισε να αναρωτιέται για τη δυνατότητα σχεδιασμού ενός υπολογιστή που βασιζόταν αποκλειστικά σε κβαντικούς κανόνες, και στη συνέχεια δημοσίευσε την επαναστατική του εργασία λίγους μήνες αργότερα [45]. Με αυτό, ξεκίνησε η ενασχόληση της επιστημονικής κοινότητας για την εκμετάλλευση των ιδεών του. Η ανάπτυξη της κβαντικής θεωρίας ξεκίνησε το 1900 με την παρουσίαση του Μαξ Πλανκ (Max Planck), στην οποία εισήγαγε την ιδέα ότι η ενέργεια υπάρχει σε μεμονωμένες μονάδες (την οποία αποκαλούσε "QUANTA"). Τα βασικά στοιχεία της κβαντικής θεωρίας:

- i. Η ενέργεια, όπως και η ύλη, αποτελείται από διακριτές μονάδες, και όχι αποκλειστικά ως συνεχές κύμα.
- ii. Στοιχειώδη σωματίδια τόσο ενέργειας όσο και ύλης, ανάλογα με τις συνθήκες, μπορεί να

συμπεριφέρονται είτε σαν σωματίδια είτε με κύματα.

- iii. Η κίνηση των στοιχειωδών σωματιδίων είναι εγγενώς τυχαία και, επομένως, απρόβλεπτη.
- iv. Η ταυτόχρονη μέτρηση δύο συμπληρωματικών αξιών, όπως η θέση και η ορμή ενός στοιχειώδους σωματιδίου, είναι αναπόφευκτα ελαττωματική. Όσο ακριβέστερα μετριέται μία τιμή, τόσο πιο ελαττωματική θα είναι η μέτρηση της άλλης τιμής.

Ο Νιλς Μπορ (Niels Bohr) πρότεινε την ερμηνεία της κβαντικής θεωρίας της Κοπεγχάγης (Αρχή της Συμπληρωματικότητας), που στην ουσία ισχυρίζεται ότι σύμφωνα με το οποίο κάθε συστατικό στη φύση έχει δύο διακριτούς και αμοιβαία αποκλειόμενους χαρακτήρες, έναν σωματιδιακό και έναν κυματικό, και το ποιος από τους δύο χαρακτήρες εμφανίζεται εξαρτάται κάθε φορά από το πώς ο παρατηρητής εκτελεί τη μέτρησή του. Πιο συγκεκριμένα, ισχυρίζεται ότι ένα σωματίδιο είναι οτιδήποτε μετριέται (για παράδειγμα, ένα κύμα ή ένα σωματίδιο) αλλά ότι δεν μπορεί να θεωρηθεί ότι έχει συγκεκριμένες ιδιότητες, μέχρι να μετρηθεί. Εν ολίγοις, ο Bohr έλεγε ότι δεν υπάρχει αντικειμενική πραγματικότητα. Αυτό μεταφράζεται σε μια αρχή που ονομάζεται υπέρθεση που ισχυρίζεται ότι, ενώ δεν γνωρίζουμε ποια είναι η κατάσταση κάθε αντικειμένου, είναι στην πραγματικότητα σε όλες τις πιθανές καταστάσεις ταυτόχρονα, εφόσον δεν ψάχνουμε να ελέγξουμε.

Για να απεικονίσουμε αυτή τη θεωρία, μπορούμε να χρησιμοποιήσουμε τη διάσημη αναλογία της γάτας του Σρόντινγκεν (Schrodinger). Πρώτα, έχουμε μια ζωντανή γάτα και την τοποθετούμε σε ένα παχύ κουτί. Σε αυτό το στάδιο, δεν υπάρχει αμφιβολία ότι η γάτα είναι ζωντανή. Στη συνέχεια, πετάμε ένα φιαλίδιο κυανίου και σφραγίζουμε το κουτί. Δεν ξέρουμε αν η γάτα είναι ζωντανή ή αν έχει σπάσει την κάψουλα με το κυάνιο και πεθαίνει. Δεδομένου ότι δεν γνωρίζουμε, η γάτα είναι και νεκρή και ζωντανή, σύμφωνα με τον κβαντικό νόμο. Μόνο όταν σπάσουμε το κουτί και δούμε σε τι κατάσταση βρίσκεται η γάτα, η υπέρθεση χάνεται, και η γάτα πρέπει να είναι είτε ζωντανή είτε νεκρή [130].

Η δεύτερη ερμηνεία της κβαντικής θεωρίας είναι το πολυσύμπαν ή η θεωρία πολλών κόσμων. Θεωρεί ότι μόλις υπάρχει δυνατότητα για οποιοδήποτε αντικείμενο να είναι σε οποιαδήποτε κατάσταση, το σύμπαν αυτού του αντικειμένου μεταγγίζει σε μια σειρά παράλληλων κόσμων ίσων με τον αριθμό των πιθανών κρατών στα οποία μπορεί να υπάρχει το αντικείμενο, με κάθε σύμπαν να περιέχει μία μοναδική ενιαία πιθανή κατάσταση αυτού του αντικειμένου. Επιπλέον, υπάρχει ένας μηχανισμός αλληλεπίδρασης μεταξύ αυτών των κόσμων που επιτρέπει με κάποιο τρόπο σε όλες τις πιθανές καταστάσεις να είναι προσβάσιμες με κάποιο τρόπο και για να επηρεαστούν όλες οι πιθανές καταστάσεις με κάποιο τρόπο. Ο Στέφεν Χόκινγκ (Stephen Hawking) και ο Ρίτσαρντ Φέινμαν (Richard Feynman) είναι μεταξύ των επιστημόνων που εξέφρασαν την προτίμησή τους για τη θεωρία του παράλληλου σύμπαντος (πολλών κόσμων). Η αρχή ότι, κατά κάποιο τρόπο, ένα σωματίδιο μπορεί να υπάρχει σε πολλές καταστάσεις ταυτόχρονα, δημιουργεί βαθιές επιπτώσεις στον τρόπο που αντιλαμβανόμαστε την σύγχρονη πληροφορική.

Η κλασική υπολογιστική βασίζεται, στο υπέρτατο επίπεδο της, στις αρχές που εκφράζονται από την άλγεβρα Μπουλ, η οποία λειτουργεί με τις αρχές των λογικών πυλών. Τα δεδομένα πρέπει να υποβληθούν σε επεξεργασία σε μια αποκλειστική δυαδική κατάσταση σε οποιοδήποτε χρονικό σημείο-δηλαδή, είτε 0 (off/FALSE) είτε 1 (on/TRUE). Αυτές οι τιμές είναι δυαδικά ψηφία ή bits. Τα εκατομμύρια τρανζίστορ και πυκνωτές στην καρδιά των υπολογιστών μπορούν να είναι μόνο σε μια κατάσταση σε οποιοδήποτε σημείο. Ενώ ο χρόνος που χρειάζεται κάθε τρανζίστορ ή πυκνωτής είτε σε 0 είτε σε 1 πριν από την εναλλαγή των καταστάσεων είναι πλέον μετρήσιμος σε δισεκατομμυριοστά του δευτερολέπτου, υπάρχει ακόμα ένα όριο για το πόσο γρήγορα μπορούν να γίνουν αυτές οι συσκευές για να αλλάξουν κατάσταση. Καθώς προοδεύουμε σε μικρότερα και ταχύτερα κυκλώματα, αρχίζουμε να φτάνουμε στα φυσικά όρια των υλικών και στο κατώφλι για

την εφαρμογή των κλασικών νόμων της φυσικής (ο νόμος του Μουρ).

Σε έναν κβαντικό υπολογιστή, μπορεί να χρησιμοποιηθεί ένας αριθμός σωματιδίων όπως τα ηλεκτρόνια ή τα φωτόνια (στην πράξη, έχει επίσης επιτευχθεί με ιόντα), είτε με την φόρτιση ή την πόλωση που ενεργούν ως αναπαράσταση του 0 ή/και του 1. Κάθε ένα από αυτά τα σωματίδια είναι γνωστό ως κβαντικό **bit**, ή **qubit**, η φύση και η συμπεριφορά αυτών των σωματιδίων αποτελούν τη βάση της κβαντικής πληροφορικής. Οι δύο πιο συναφείς πτυχές της κβαντικής φυσικής είναι οι αρχές της υπέρθεσης και της εμπλοκής. Η περιστροφή του ηλεκτρονίου μπορεί να είναι είτε σε ευθυγράμμιση με το πεδίο, το οποίο είναι γνωστό ως κατάσταση περιστροφής, είτε απέναντι από το πεδίο, το οποίο είναι γνωστό ως κατάσταση περιστροφής-κάτω. Η αλλαγή της περιστροφής του ηλεκτρονίου από τη μια κατάσταση στην άλλη επιτυγχάνεται με τη χρήση ενός παλμού ενέργειας, όπως από ένα λέιζερ.

Αλλά αν χρησιμοποιήσουμε μόνο μισή μονάδα ενέργειας λέιζερ και απομονώσουμε εντελώς το σωματίδιο από όλες τις εξωτερικές επιρροές; Σύμφωνα με τον κβαντικό νόμο, το σωματίδιο εισέρχεται στη συνέχεια σε μια υπέρθεση, στο οποίο συμπεριφέρεται σαν να ήταν και στις δύο καταστάσεις ταυτόχρονα. Κάθε **qubit** που χρησιμοποιείται θα μπορούσε να πάρει μια υπέρθεση του 0 και 1. Έτσι, ο αριθμός των υπολογισμών που θα μπορούσε να αναλάβει ένας κβαντικός υπολογιστής είναι 2^n , όπου n είναι ο αριθμός των **qubits** που χρησιμοποιούνται. Ένας κβαντικός υπολογιστής αποτελούμενος από 500 **qubits** θα είχε τη δυνατότητα να κάνει 2^{500} υπολογισμούς σε ένα μόνο βήμα [126].

Τα σωματίδια διεμπλοκής (όπως τα φωτόνια, τα ηλεκτρόνια ή τα **qubits**) που έχουν αλληλεπιδράσει σε κάποιο σημείο διατηρούν έναν τύπο σύνδεσης και μπορούν να εμπλακούν μεταξύ τους σε ζεύγη, σε μια διεργασία γνωστή ως διεμπλοκή. Γνωρίζοντας την κατάσταση περιστροφής ενός εμπλεκόμενου σωματιδίου, επιτρέπει σε κάποιον να γνωρίζει ότι η περιστροφή είναι προς την αντίθετη κατεύθυνση. Η κατάσταση περιστροφής του μετρούμενου σωματιδίου αποφασίζεται κατά τη στιγμή της μέτρησης και κοινοποιείται στο σχετιζόμενο σωματίδιο, το οποίο ταυτόχρονα αναλαμβάνει την αντίθετη κατεύθυνση περιστροφής με εκείνη του μετρούμενου σωματιδίου. Το φαινόμενο της κβαντικής διεμπλοκής, είναι ένα πραγματικό φαινόμενο που ο Αϊνστάιν το αποκάλεσε «στοιχειωμένη δράση σε απόσταση» (*spooky action at a distance*), ο μηχανισμός του οποίου δεν μπορεί, μέχρι στιγμής, να εξηγηθεί από οποιαδήποτε θεωρία-αλλά πρέπει να ληφθεί ως δεδομένη. Η κβαντική εμπλοκή επιτρέπει τα **qubits** που διαχωρίζονται από απίστευτες αποστάσεις να αλληλεπιδρούν μεταξύ τους αχαριαία (δεν περιορίζονται στην ταχύτητα του φωτός).

Μαζί, η κβαντική υπέρθεση και η κβαντική διεμπλοκή δημιουργούν μια εξαιρετικά ενισχυμένη υπολογιστική ισχύ. Όταν ένας καταχωρητής 2-bit σε ένα συνηθισμένο υπολογιστή μπορεί να αποθηκεύσει μόνο μία από τις τέσσερις δυαδικές διαμορφώσεις (00, 01, 10, ή 11) σε οποιαδήποτε δεδομένη στιγμή, ένας καταχωρητής 2-qubit σε έναν κβαντικό υπολογιστή μπορεί να αποθηκεύσει και τους τέσσερις αριθμούς ταυτόχρονα, επειδή κάθε **qubit** αντιπροσωπεύει δύο τιμές. Εάν προστεθούν περισσότερα **qubits**, η αυξημένη δυναμικότητα επεκτείνεται εκθετικά.

Ίσως ακόμα πιο ενδιαφέρουσα από την απόλυτη δύναμη της κβαντικής πληροφορικής είναι η ικανότητα που προσφέρει για να δημιουργηθούν προγράμματα με έναν εντελώς νέο τρόπο. Για παράδειγμα, ένας κβαντικός υπολογιστής θα μπορούσε να ενσωματώσει μια ακολουθία προγραμματισμού που θα ήταν κατά μήκος των γραμμών του "Πάρτε όλες τις υπερθέσεις όλων των προτέρων υπολογισμών" -κάτι που δεν έχει νόημα με έναν κλασικό υπολογιστή- που θα επέτρεπε εξαιρετικά γρήγορους τρόπους επίλυσης ορισμένων μαθηματικών προβλημάτων.

Έχουν γίνει δύο αξιοσημείωτες επιτυχίες μέχρι στιγμής με τον κβαντικό προγραμματισμό. Το πρώτο συνέβη το 1994 από τον Πίτερ Σορ (Peter Shor), που ανέπτυξε έναν κβαντικό αλγόριθμο ο οποίος θα μπορούσε να λειτουργήσει αποτελεσματικά στην παραγοντοποίηση μεγάλων

πρώτων αριθμών. Επικεντρώνεται σε ένα σύστημα που χρησιμοποιεί τη θεωρία αριθμών για να υπολογίσει την περιοδικότητα μιας μεγάλης σειράς ακολουθίας [173]. Το άλλο μεγάλο επίτευγμα συνέβη με τα εργαστήρια **Lov Grover** του εργαστηρίου **Bell** το 1996, με έναν πολύ γρήγορο αλγόριθμο που αποδεικνύεται ότι είναι ο ταχύτερος δυνατός για αναζήτηση σε μη δομημένες βάσεις δεδομένων.

Ο αλγόριθμος είναι τόσο αποτελεσματικός που απαιτεί μόνο, κατά μέσο όρο, περίπου N τετράγωνη ρίζα (όπου N είναι ο συνολικός αριθμός των στοιχείων) αναζητήσεις για να βρεθεί το επιθυμητό αποτέλεσμα, σε αντίθεση με μια αναζήτηση στην κλασική πληροφορική, η οποία κατά μέσο όρο χρειάζεται $N/2$ αναζητήσεις [78].

Τα παραπάνω ακούγονται πολλά υποσχόμενα, αλλά υπάρχουν ακόμη τεράστια εμπόδια που πρέπει να ξεπεραστούν. Μερικά από τα προβλήματα με την κβαντική υπολογιστική είναι τα εξής:

- i. Παρεμβολές-κατά τη διάρκεια της φάσης υπολογισμού ενός κβαντικού υπολογισμού, η παραμικρή διαταραχή σε ένα κβαντικό σύστημα προκαλεί την κατάρρευση του κβαντικού υπολογισμού, μια διαδικασία γνωστή ως αποσυνεκτικότητα (*decoherence*). Ένας κβαντικός υπολογιστής πρέπει να απομονώνεται εντελώς από όλες τις εξωτερικές παρεμβολές κατά τη διάρκεια της φάσης υπολογισμού. Κάποια επιτυχία έχει επιτευχθεί με τη χρήση των **qubits** σε έντονα μαγνητικά πεδία, με τη χρήση ιόντων.
- ii. Διόρθωση σφαλμάτων -επειδή η απομόνωση ενός κβαντικού συστήματος έχει αποδειχθεί τόσο δύσκολη, έχουν αναπτυχθεί συστήματα διόρθωσης σφαλμάτων για κβαντικούς υπολογισμούς. Τα **qubits** δεν είναι ψηφιακά **bit** δεδομένων, επομένως δεν μπορούν να χρησιμοποιήσουν συμβατική (και πολύ αποτελεσματική) διόρθωση σφαλμάτων, όπως η χρήση τριπλής πλεονάζουσας μεθόδου. Δεδομένης της φύσης της κβαντικής πληροφορικής, η διόρθωση σφαλμάτων είναι εξαιρετικά κρίσιμη-ακόμη και ένα μεμονωμένο σφάλμα σε έναν υπολογισμό μπορεί να προκαλέσει κατάρρευση της εγκυρότητας ολόκληρου του υπολογισμού. Έχει σημειωθεί σημαντική πρόοδος σε αυτόν τον τομέα, με έναν αλγόριθμο διόρθωσης σφαλμάτων που αναπτύχθηκε και χρησιμοποιεί 9 **qubits**. Πιο πρόσφατα, υπήρξε μια ανακάλυψη από την **IBM** που χρησιμοποιεί ένα σύνολο 5 **qubits**.
- iii. Τήρηση εξόδου-Η ανάκτηση δεδομένων εξόδου μετά από έναν κβαντικό υπολογισμό είναι ο κίνδυνος να διαφθείρει τα δεδομένα. Σε ένα παράδειγμα κβαντικού υπολογιστή με 500 **qubits**, έχουμε 1 στις 2^{500} πιθανότητες να παρατηρήσουμε τη σωστή έξοδο αν ποσοτικοποιήσουμε την έξοδο. Έτσι, αυτό που χρειάζεται είναι μια μέθοδος για να διασφαλιστεί ότι, μόλις πραγματοποιηθούν όλοι οι υπολογισμοί και πραγματοποιηθεί η πράξη παρατήρησης, η παρατηρηθείσα αξία θα αντιστοιχεί στη σωστή απάντηση. Πώς μπορεί να γίνει αυτό; Έχει επιτευχθεί από τον **Grover** με τον αλγόριθμο αναζήτησης βάσης δεδομένων του, που βασίζεται στο ειδικό σχήμα "κυματομορφής" της χαμπύλης πιθανότητας που είναι εγγενής στους κβαντικούς υπολογιστές, που εξασφαλίζει, μόλις γίνουν όλοι οι υπολογισμοί, η πράξη μέτρησης θα δει την κβαντική κατάσταση να αποκρυπτογραφήσει τη σωστή απάντηση. Αν και υπάρχουν πολλά προβλήματα που πρέπει να ξεπεραστούν, τα επιτεύγματα τα τελευταία 15 χρόνια, και ειδικά τα τελευταία 3, έχουν κάνει κάποια μορφή πρακτικής κβαντικής πληροφορικής όχι ανέφικτη, αλλά υπάρχει μεγάλη συζήτηση για το αν αυτό είναι λιγότερο από μια δεκαετία μακριά ή εκατό χρόνια είναι στο μέλλον. Ωστόσο, το δυναμικό που προσφέρει αυτή η τεχνολογία προσελκύει τεράστιο ενδιαφέρον τόσο από την κυβέρνηση όσο και από τον ιδιωτικό τομέα. Οι στρατιωτικές εφαρμογές περιλαμβάνουν την ικανότητα να σπάνε κλειδιά κρυπτογραφήσεων μέσω των μη στρατιωτικών ερευνών, ενώ οι μη στρατιωτικές εφαρμογές κυμαίνονται από μοντελοποίηση **DNA** έως πολύπλοκη επιστημονική ανάλυση υλικών. Αυτό το δυναμικό είναι αυτό που σπάει γρήγορα τα εμπόδια

σε αυτή την τεχνολογία, αλλά αν όλα τα εμπόδια μπορούν να σπάσουν, και τότε, είναι ένα πολύ ανοικτό ερώτημα.

Η κβαντική μηχανική, το υποπεδίο της φυσικής που περιγράφει τη συμπεριφορά των πολύ μικρών σωματιδίων, παρέχει τη βάση για ένα νέο πρότυπο πληροφορικής. Η κβαντική υπολογιστική προτάθηκε για πρώτη φορά στη δεκαετία του 1980 ως ένας τρόπος για να βελτιωθεί η υπολογιστική μοντελοποίηση της συμπεριφοράς των πολύ μικρών ("κβαντικών") φυσικών συστημάτων. Το ενδιαφέρον για το πεδίο αυξήθηκε τη δεκαετία του 1990 με την εισαγωγή του αλγορίθμου του Shor, ο οποίος, αν υλοποιηθεί σε κβαντικό υπολογιστή, θα επιταχύνει εκθετικά μια σημαντική κατηγορία κρυπτοανάλυσης και ενδέχεται να απειλήσει ορισμένες από τις κρυπτογραφικές μεθόδους που χρησιμοποιούνται για προστασία της κυβέρνησης και των πολιτικών επικοινωνιών και των αποθηκευμένων δεδομένων. Στην πραγματικότητα, οι κβαντικές υπολογιστές είναι το μόνο γνωστό μοντέλο για την πληροφορική που θα μπορούσε να προσφέρει εκθετική ταχύτητα στους σημερινούς υπολογιστές.

Ενώ τα αποτελέσματα αυτά ήταν πολύ συναρπαστικά τη δεκαετία του 1990, υπήρχε μόνο θεωρητικό ενδιαφέρον: κανείς δεν γνώριζε μια μέθοδο για την κατασκευή ενός υπολογιστή από τα κβαντικά συστήματα. Σήμερα, σχεδόν 25 χρόνια αργότερα, η πρόοδος στη δημιουργία και τον έλεγχο των στοιχείων της κβαντικής πληροφορίας, ή "qubits", έχει προχωρήσει στο σημείο ότι μια σειρά από ερευνητικές ομάδες έχουν επιδείξει μικρές αποδείξεις κβαντικών υπολογιστών. Αυτό το έργο έχει αναζωογονήσει το πεδίο και οδήγησε σε σημαντικές επενδύσεις του ιδιωτικού τομέα.

Ένας κλασικός υπολογιστής χρησιμοποιεί bits για να αναπαριστά τις τιμές στις οποίες λειτουργεί. Ένας κβαντικός υπολογιστής χρησιμοποιεί κβαντικά κομμάτια ή qubit. Ένα bit μπορεί να είναι είτε 0 είτε 1, ενώ ένα qubit μπορεί να αναπαριστά τις τιμές 0 ή 1 ή κάποιο συνδυασμό και των δύο ταυτόχρονα (γνωστή ως "υπέρθεση"). Ενώ η κατάσταση ενός κλασικού υπολογιστή προσδιορίζεται από τις δυαδικές τιμές μιας συλλογής bits, σε οποιοδήποτε μεμονωμένο σημείο του χρόνου η κατάσταση ενός κβαντικού υπολογιστή με τον ίδιο αριθμό κβαντικών bits μπορεί να εκτείνεται σε όλες τις πιθανές καταστάσεις του αντίστοιχου κλασικού υπολογιστή, και έτσι λειτουργεί σε ένα εκθετικά μεγαλύτερο πρόβλημα χώρο. Ωστόσο, η ικανότητα να κάνετε χρήση αυτού του χώρου απαιτεί ότι όλα τα qubits είναι εγγενώς αλληλένδετα ("μπλεγμένα"), καλά απομονωμένα από το εξωτερικό περιβάλλον, και ελέγχονται με ακρίβεια.

Πολλές καινοτομίες τα τελευταία 25 χρόνια επέτρεψαν στους ερευνητές να οικοδομήσουν σωματιδιακά συστήματα που αρχίζουν να παρέχουν την απαιτούμενη απομόνωση και έλεγχο για την κβαντική Υπολογιστική. Το 2018, δύο τεχνολογίες χρησιμοποιούνται στους περισσότερους κβαντικούς υπολογιστές (παγιδευμένα ιόντα και τεχνητά «άτομα» που παράγονται από υπεραγώγιμα κυκλώματα), αλλά επί του παρόντος διερευνώνται πολλές διαφορετικές τεχνολογίες για τη βασική φυσική εφαρμογή των qubits, ή "σωματιδιακή του qubit." Δεδομένης της ταχείας προόδου στον τομέα και των μεγάλων βελτιώσεων που εξακολουθούν να απαιτούνται, είναι πολύ νωρίς για να «στοιχηματίσει» σε μία πλήρως εμπορική τεχνολογία για την κβαντική πληροφορική.

Ακόμη και αν κάποιος είναι σε θέση να παράγει πολύ υψηλής ποιότητας qubits, δημιουργώντας και κάνοντας χρήση αυτών των κβαντικών υπολογιστών, προκύπτει μια νέα σειρά προκλήσεων. Χρησιμοποιούν ένα διαφορετικό σύνολο λειτουργιών από αυτά των κλασικών υπολογιστών, απαιτώντας νέους αλγόριθμους, λογισμικό, τεχνολογίες ελέγχου και τεχνικές άντλησης ενέργειας του υλικού.

Η κβαντική πληροφορική είναι μια επαναστατική έννοια στην επεξεργασία ψηφιακών δεδομένων με βάση τις θεμελιώδεις αρχές με τις οποίες λειτουργεί η φύση, δηλαδή την κβαντική μηχανική. Με προόδους στη φυσική στις αρχές του 20ου αιώνα, οι μέθοδοι παρατήρησης των υλικών έφ-

τασαν στο επίπεδο στο οποίο ορισμένα κβαντικά φαινόμενα έγιναν ανιχνεύσιμα. Το καλύτερο παράδειγμα αυτού είναι ένα τυπικό τρανζίστορ που υπάρχει στο σύγχρονο υπολογιστή ή συσκευή. Λειτουργεί κατευθύνοντας μεγάλα φορτία ηλεκτρικού ρεύματος με τη χρήση υλικών και κβαντικών αρχών (δομή ζώνης, μεταφρασμένες καταστάσεις κ.λπ.). Παράγουν ασυνήθιστη συμπεριφορά για φυσικά υλικά που βρέθηκαν – μια ικανότητα να ελέγχετε με ακρίβεια το ρεύμα με το ρεύμα ή το ρεύμα μέσω του φωτός ή του φωτός μέσω του ρεύματος. Πρόσφατα, η σμίκρυνση των τρανζίστορ, των ανιχνευτών και, ταυτόχρονα, η πρόσβαση σε εξαιρετικά χαμηλές θερμοκρασίες (όπως οι -273 κελσίου) έχουν ανοίξει τη δυνατότητα να λειτουργούν, να παρατηρούν και, το σημαντικότερο, να απομονώνουν μεμονωμένα σωματίδια με τις εγγενώς κβαντικές τους καταστάσεις.

Αυτές οι ιδιότητες είναι ένας λόγος που οι επιτραπέζιοι ή φορητοί υπολογιστές που χρησιμοποιούμε έχουν μικρή ομοιότητα με κβαντικούς υπολογιστές. Παρόλο που η θεωρία δημιουργήθηκε πριν από 100 χρόνια, αυτές οι ιδιότητες έχουν παρατηρηθεί και επαληθευτεί μόνο πρόσφατα σε ένα ευρύ φάσμα σωματιδίων και συστημάτων. Ενώ τα φαινόμενα δεν μπορούν να παρατηρηθούν στην καθημερινότητά μας, η κβαντική μηχανική έχει ήδη οδηγήσει σε δραματικές προόδους σε λίγους συγκεκριμένους τομείς της ιατρικής (π.χ. μαγνητική τομογραφία, χειρουργική επέμβαση λέιζερ) με δυνατότητα να φέρει επανάσταση στην ιατρική έρευνα και την κλινική φροντίδα. Επειδή οι σύγχρονοι υπολογιστές έχουν σε μεγάλο βαθμό κορεσμένη υπολογιστική ισχύ και δεν αναπτύσσεται πλέον εκθετικά, όπως συνέβη τον τελευταίο αιώνα, η κβαντική Υπολογιστική, αν πραγματοποιηθεί, είναι η πιο ελπιδοφόρα τεχνολογία για μεγάλες προόδους σε διεργασίες που βρίσκονται σήμερα πέρα από την υπάρχουσα υπολογιστική ισχύ.

6.1.1 Τεχνικοί κίνδυνοι

- Τα qubits δεν μπορούν εγγενώς να απορρίψουν τον θόρυβο: Μία από τις σημαντικότερες διαφορές μεταξύ ενός κλασικού υπολογιστή και ενός κβαντικού υπολογιστή είναι ο τρόπος με τον οποίο χειρίζεται μικρές ανεπιθύμητες παραλλαγές ή θόρυβο στο σύστημα. Δεδομένου ότι ένα κλασικό bit είναι είτε ένα είτε μηδέν, ακόμη και αν η τιμή είναι ελαφρώς απενεργοποιημένη (κάποιο θόρυβο στο σύστημα) είναι εύκολο για τις λειτουργίες σε αυτό το σήμα για να αφαιρέσετε αυτό το θόρυβο. Στην πραγματικότητα, οι σημερινές κλασικές πύλες, οι οποίες λειτουργούν σε bits και χρησιμοποιούνται για τη δημιουργία υπολογιστών, έχουν πολύ μεγάλα περιθώρια θορύβου, μπορούν να απορρίψουν μεγάλες διακυμάνσεις στις εισροές τους και να παράγουν.
- Προσθήκη χωρίς καθαρές εξόδους χωρίς θόρυβο: Επειδή ένα qubit μπορεί να είναι οποιοσδήποτε συνδυασμός του ενός και του μηδενός, τα qubit και οι κβαντικές πύλες δεν μπορούν να απορρίψουν εύκολα μικρά σφάλματα (θόρυβος) που εμφανίζονται στα φυσικά κυκλώματα. Ως αποτέλεσμα, μικρά λάθη στη δημιουργία των επιθυμητών κβαντικών λειτουργιών, ή τυχόν αδέσποτων σημάτων στο φυσικό σύστημα, μπορούν τελικά να οδηγήσουν σε λανθασμένες εκροές που εμφανίζονται στον υπολογισμό. Έτσι, μία από τις σημαντικότερες παραμέτρους σχεδιασμού για τα συστήματα που λειτουργούν σε φυσικά qubits είναι το ποσοστό σφάλματος τους. Τα χαμηλά ποσοστά σφάλματος ήταν δύσκολο να επιτευχθούν. Ακόμη και στα μέσα του 2018, τα ποσοστά σφάλματος για λειτουργίες 2-qubit σε συστήματα με 5 ή περισσότερα qubit είναι περισσότερα από λίγα τοις εκατό. Τα περισσότερα ποσοστά σφαλμάτων έχουν καταδειχθεί σε μικρότερα συστήματα και αυτή η βελτιωμένη λειτουργία πιστότητας πρέπει να μετακινηθεί σε μεγαλύτερα συστήματα qubit για να είναι επιτυχής η κβαντική Υπολογιστική. Αν και οι φυσικές λειτουργίες qubit είναι ευαίσθητες

στο θόρυβο, είναι δυνατή η εκτέλεση ενός αλγορίθμου διόρθωσης κβαντικού σφάλματος (QEC) σε έναν φυσικό κβαντικό υπολογιστή, ώστε να προσομοιώνει έναν κβαντικό υπολογιστή χωρίς θόρυβο ή "πλήρως διορθωμένο σφάλμα". Χωρίς το QEC, είναι απίθανο ένα πολύπλοκο κβαντικό πρόγραμμα, όπως αυτό που υλοποιεί τον αλγόριθμο του Shor, να τρέξει ποτέ σωστά σε έναν κβαντικό υπολογιστή. Ωστόσο, η QEC παρουσιάζει σημαντικά έξοδα όσον αφορά τόσο τον αριθμό των σωματικών Κουκ που απαιτούνται για την εξομείωση ενός πιο ισχυρού και σταθερού Kbit, όσο και τον αριθμό των αρχέγονων qubits που πρέπει να εκτελεστούν σε φυσικές qubits. Ενώ τα QECs θα είναι απαραίτητα για τη δημιουργία κβαντικών υπολογιστών χωρίς σφάλματα στο μέλλον, είναι πολύ εντατικές για χρήση πόρων για να χρησιμοποιηθούν βραχυπρόθεσμα: οι κβαντικές υπολογιστές στο εγγύς διάστημα είναι πιθανό να έχουν σφάλματα. Αυτή η κατηγορία μηχανών αναφέρεται ως θορυβώδεις υπολογιστές μεσαίας κλίμακας (NISQ). Οι μεγάλες εισροές δεδομένων δεν μπορούν να φορτωθούν σε ένα κβαντικό υπολογιστή αποτελεσματικά. Ενώ ένας κβαντικός υπολογιστής μπορεί να χρησιμοποιήσει ένα μικρό αριθμό qubits για να αντιπροσωπεύει μια εκθετικά μεγαλύτερη ποσότητα δεδομένων, δεν υπάρχει αυτή τη στιγμή μια μέθοδος για να μετατρέψει γρήγορα μια μεγάλη ποσότητα των κλασικών δεδομένων σε μία κβαντική κατάσταση (αυτό δεν ισχύει αν τα δεδομένα μπορούν να δημιουργηθούν αλγοριθμικά). Για προβλήματα που απαιτούν μεγάλες εισροές, το χρονικό διάστημα που απαιτείται για τη δημιουργία της εισαγωγής κβαντικής κατάστασης θα κυριαρχούσε συνήθως στο χρόνο υπολογισμού και θα μείωνε σημαντικά το κβαντικό πλεονέκτημα.

- Το σχέδιο κβαντικού αλγορίθμου είναι προκλητικό: Η μέτρηση της κατάστασης ενός κβαντικού υπολογιστή "συμπύσσει" τη μεγάλη κβαντική κατάσταση σε ένα ενιαίο κλασικό αποτέλεσμα. Αυτό σημαίνει ότι μπορεί κανείς να εξαγάγει μόνο την ίδια ποσότητα δεδομένων από έναν κβαντικό υπολογιστή που θα μπορούσε κάποιος από έναν κλασικό υπολογιστή ίδιου μεγέθους. Για να αποκομίσουν τα οφέλη ενός κβαντικού υπολογιστή, οι κβαντικοί αλγόριθμοι πρέπει να αξιοποιούν μοναδικά κβαντικά χαρακτηριστικά, όπως παρεμβολές και εμπλοκή, για να φθάσουν στο τελικό κλασικό αποτέλεσμα. Έτσι, η επίτευξη κβαντικού ταχύτατο απαιτεί εντελώς νέα είδη αρχών σχεδιασμού αλγορίθμου και πολύ έξυπνου σχεδιασμού αλγορίθμων. Η ανάπτυξη του κβαντικού αλγορίθμου είναι μια κρίσιμη πτυχή του πεδίου.
- Οι κβαντικοί υπολογιστές θα χρειαστούν μια νέα στοίβα λογισμικού: Όπως και με όλους τους υπολογιστές, η δημιουργία μιας χρήσιμης συσκευής είναι πολύ πιο περίπλοκη από την απλή δημιουργία του υλικού — απαιτούνται εργαλεία για τη δημιουργία και τον εντοπισμό σφαλμάτων στο λογισμικό για τους κβαντικούς υπολογιστές. Δεδομένου ότι τα κβαντικά προγράμματα είναι διαφορετικά από τα προγράμματα για τους κλασικούς υπολογιστές, η έρευνα και η ανάπτυξη είναι απαραίτητη για την περαιτέρω ανάπτυξη της στοίβας εργαλείων λογισμικού. Επειδή αυτά τα εργαλεία λογισμικού οδηγούν το υλικό, η σύγχρονη ανάπτυξη της αλυσίδας εργαλείων υλικού και λογισμικού θα συντομεύσει τον χρόνο ανάπτυξης για έναν χρήσιμο κβαντικό υπολογιστή. Στην πραγματικότητα, η χρήση πρόωρων εργαλείων για την ολοκλήρωση του σχεδιασμού από άκρο σε άκρο (σχεδιασμός εφαρμογής στα τελικά αποτελέσματα) βοηθά στην αποσαφήνιση κρυφών ζητημάτων και οδηγεί σε σχέδια με την καλύτερη ευκαιρία για συνολική επιτυχία, μια προσέγγιση που χρησιμοποιείται στον κλασικό σχεδιασμό του υπολογιστή.
- Η ενδιάμεση κατάσταση ενός κβαντικού υπολογιστή δεν μπορεί να μετρηθεί απευθείας: Οι μέθοδοι για τον εντοπισμό σφαλμάτων κβαντικού υλικού και λογισμικού είναι κρίσιμης σημασίας. Οι τρέχουσες μέθοδοι εντοπισμού σφαλμάτων για κλασικούς υπολογιστές

βασίζονται στη μνήμη και στην ανάγνωση των ενδιάμεσων κρατών μηχανών. Ούτε σε κβαντικό υπολογιστή. Μια κβαντική κατάσταση δεν μπορεί απλά να αντιγραφεί (ανά το λεγόμενο θεώρημα μη κλωνοποίησης) για μεταγενέστερη εξέταση, και οποιαδήποτε μέτρηση μιας κβαντικής κατάστασης την συμπτύσσει σε ένα σύνολο κλασικών κομματιών, φέρνοντας τον υπολογισμό σε αδιέξοδο. Οι νέες προσεγγίσεις στον εντοπισμό σφαλμάτων είναι απαραίτητες για την ανάπτυξη κβαντικών υπολογιστών μεγάλης κλίμακας.

Αν και η σκοπιμότητα ενός κβαντικού υπολογιστή μεγάλης κλίμακας δεν είναι ακόμα βέβαιη, τα οφέλη της προσπάθειας για την ανάπτυξη ενός πρακτικού κβαντικού υπολογιστή είναι πιθανό να είναι μεγάλα, και μπορούν να συνεχίσουν να διοχετεύονται σε άλλες πιο βραχυπρόθεσμες εφαρμογές της κβαντικής τεχνολογίας πληροφοριών, όπως η ανίχνευση με βάση το qubit. Εκτός από τα πνευματικά και δυνητικά κοινωνικά οφέλη της κβαντικής πληροφορικής, ο τομέας αυτός έχει επιπτώσεις στην εθνική ασφάλεια. Κάθε οντότητα στην κατοχή ενός μεγάλου, πρακτικού κβαντικού υπολογιστή θα μπορούσε να σπάσει τα σημερινά ασύμμετρα κρυπτοσυστήματα που θα είναι ένα σημαντικό πλεονέκτημα για την ανάκτηση ευαίσθητων δεδομένων.

Η συνειδητοποίηση αυτού του κινδύνου έχει δρομολογήσει προσπάθειες για τη δημιουργία και την ανάπτυξη κρυπτογραφικών συστημάτων που είναι ανθεκτικά στην κβαντική κρυπτανάλυση, για τα οποία υπάρχουν αρκετοί υποψήφιοι που πιστεύεται ότι είναι ασφαλείς στην κβαντική ασφάλεια. Ωστόσο, ενώ η ανάπτυξη της μετα-κβαντικής κρυπτογραφίας σε κυβερνητικά και πολιτικά συστήματα μπορεί να προστατεύσει τις επακόλουθες επικοινωνίες, δεν θα αφαιρέσει τον κίνδυνο ασφάλειας για τα προ-κβαντικά κρυπτογραφημένα δεδομένα που έχουν ήδη αναχαιτιστεί, αν και το μέγεθος αυτού του κινδύνου μειώνεται καθώς η ώρα άφιξης ενός κβαντικού υπολογιστή ικανό να αναπτύξει τον αλγόριθμο του Shor αυξάνεται και τα δεδομένα καθίστανται λιγότερο συναφή. Επιπλέον, οι νέοι κβαντικοί αλγόριθμοι ή υλοποιήσεις θα μπορούσαν να οδηγήσουν σε νέες κβαντικές κρυπτογραφικές τεχνικές. Όπως και με την κυβερνοασφάλεια γενικότερα, η μετακβαντική ανθεκτικότητα θα απαιτήσει συνεχιζόμενη έρευνα ασφάλειας.

Αλλά τα θέματα εθνικής ασφάλειας ξεπερνούν την κρυπτογράφηση. Το μεγαλύτερο στρατηγικό ερώτημα αφορά τη μελλοντική οικονομική και τεχνολογική ηγεσία. Ιστορικά, η κλασική πληροφορική είχε μεγάλη επίπτωση στον μετασχηματισμό των κοινωνιών. Αντίστοιχη επίπτωση αναμένεται να έχει και η έλευση του κβαντικού υπολογισμού. Ενώ η δυνατότητα εφαρμογής κβαντικών αλγορίθμων σε βιομηχανικές και ερευνητικές εφαρμογές έχει αρχίσει μόνο να διερευνάται, είναι σαφές ότι η κβαντική Υπολογιστική έχει τη δυνατότητα να υπερβεί τα τρέχοντα υπολογιστικά όρια. Η δυνατότητα βελτίωσης της αποδοτικότητας σε πολλούς τομείς υπολογισμού επιβεβαιώνει ότι η υποστήριξη μιας ισχυρής ερευνητικής κοινότητας του στις Ηνωμένες Πολιτείες (αλλά και σε άλλες χώρες με αντίστοιχα ερευνητικά προγράμματα) έχει στρατηγική αξία.

6.2 Quantum Annealing - Κβαντική Ανόπτηση

Η Κβαντική Ανόπτηση (Quantum Annealing - QA) είναι μια μεθυστική μέθοδος για την εξεύρεση του ολικού ελάχιστου μιας δεδομένης αντικειμενικής συνάρτησης σε ένα δεδομένο σύνολο υποψηφίων λύσεων, με μια διαδικασία που χρησιμοποιεί κβαντικές διακυμάνσεις. Η Κβαντική Ανόπτηση χρησιμοποιείται κυρίως για προβλήματα όπου ο χώρος αναζήτησης είναι διακριτός με πολλά τοπικά ελάχιστα, όπως η εύρεση της κατάστασης εδάφους ενός γυαλιού περιστροφής (spin glass). Διατυπώθηκε με τη σημερινή του μορφή από τον Καντογουάκι (Kadowaki) και τον Νισιμόρι (Nishimori) [94].

Η Κβαντική Ανόπτηση ξεκινά από μια κβαντομηχανική υπέρθεση όλων των πιθανών καταστάσεων με ίσα βάρη. Στη συνέχεια, το σύστημα εξελίσσεται ακολουθώντας την εξίσωση του χρό-

νου που εξαρτάται από το χρόνο, μια φυσική κβαντομηχανική εξέλιξη των σωματικών συστημάτων. Τα πλάτη όλων των υποψηφίων καταστάσεων συνεχίζουν να αλλάζουν, συνειδητοποιώντας έναν κβαντικό παραλληλισμό, σύμφωνα με την εξαρτώμενη από το χρόνο δύναμη του εγκάρσιου πεδίου, η οποία προκαλεί κβαντική διεμπλοκή μεταξύ των καταστάσεων. Εάν ο ρυθμός μεταβολής του εγκάρσιου πεδίου είναι αρκετά αργός, το σύστημα παραμένει κοντά στην κατάσταση του εδάφους της στιγμιαίας χαμιλτονιανής. Εάν επιταχυνθεί το ποσοστό μεταβολής του εγκάρσιου πεδίου, το σύστημα μπορεί να εγκαταλείψει προσωρινά την κατάσταση του εδάφους, αλλά και να προκαλέσει μεγαλύτερη πιθανότητα σύναψης στην κατάσταση του εδάφους του τελικού προβλήματος, δηλαδή τον δια-καταστασιακό κβαντικό υπολογισμό.

Το εγκάρσιο πεδίο είναι τελικά απενεργοποιημένο και το σύστημα αναμένεται να έχει φθάσει στην κατάσταση εδάφους του μοντέλου κλασικής βελτίωσης που αντιστοιχεί στη λύση στο αρχικό πρόβλημα βελτιστοποίησης. Μια πειραματική επίδειξη της επιτυχίας της κβαντικής ανόπτησης αναφέρθηκε αμέσως μετά την αρχική θεωρητική πρόταση. Μια εισαγωγή στα προβλήματα συνδυαστικής βελτιστοποίησης (NP-hard), η γενική δομή των αλγορίθμων Κβαντικής Ανόπτησης και δύο παραδείγματα αυτού του είδους αλγορίθμων για την επίλυση παρουσιών του Max-SAT και ελάχιστα πολλαπλά προβλήματα μαζί με μία επισκόπηση των συστημάτων Κβαντικής Ανόπτησης που κατασκευάζονται από τα συστήματα D-Wave παρουσιάζονται σε μελέτες.

Η Κβαντική Ανόπτηση (quantum annealing) μπορεί να συγκριθεί με την Προσομοιωμένη Ανόπτηση (simulated annealing), της οποίας η παράμετρος "θερμοκρασία" παίζει παρόμοιο ρόλο με την ισχύ του πεδίου διοχέτευσης της διασφάλισης ποιότητας. Στην Προσομοιωμένη Ανόπτηση, η θερμοκρασία καθορίζει την πιθανότητα μετακίνησης σε κατάσταση υψηλότερης "ενέργειας" από μία τρέχουσα κατάσταση. Στην Κβαντική Ανόπτηση, η δύναμη του εγκάρσιου πεδίου καθορίζει την κβαντομηχανική πιθανότητα να αλλάξει τα πλάτη όλων των καταστάσεων παράλληλα. Αναλυτικά και αριθμητικά στοιχεία δείχνουν ότι η Κβαντική Ανόπτηση ξεπερνά την Προσομοιωμένη Ανόπτηση υπό ορισμένες συνθήκες.

Το πεδίο διοχέτευσης είναι βασικά ένας όρος κινητικής ενέργειας που δεν μετακινείται με το κλασικό δυναμικό ενεργειακό μέρος του αρχικού γυαλιού. Η όλη διαδικασία μπορεί να προσομοιωθεί σε έναν υπολογιστή χρησιμοποιώντας ένα κβαντικό Monte Carlo αλγόριθμο (ή κάποια άλλη στοχαστική τεχνική), και έτσι να αποκτήσει ένα ευρετικό αλγόριθμο για την εύρεση της κατάστασης εδάφους του.

Έχει αποδειχθεί πειραματικά, καθώς και θεωρητικά, ότι η Κβαντική Ανόπτηση μπορεί πράγματι να ξεπεράσει την θερμική ανόπτηση (Προσομοίωση Ανόπτησης) σε ορισμένες περιπτώσεις, ειδικά όταν το δυνητικό ενεργειακό (κόστος) τοπίο αποτελείται από πολύ υψηλά αλλά λεπτά εμπόδια γύρω από ρηχά τοπικά ελάχιστα. Εικάζεται ότι σε έναν κβαντικό υπολογιστή, τέτοιες προσομοιώσεις θα ήταν πολύ πιο αποτελεσματικές και ακριβείς από αυτές που έγιναν σε έναν κλασικό υπολογιστή, επειδή μπορεί να εκτελέσει την σήραγγα απευθείας, αντί να χρειάζεται να την προσθέσει με το χέρι. Επιπλέον, μπορεί να είναι σε θέση να το κάνει αυτό χωρίς τους στενούς ελέγχους σφαλμάτων που απαιτούνται για την αξιοποίηση της κβαντικής διεμπλοκής που χρησιμοποιείται σε πιο παραδοσιακούς κβαντικούς αλγόριθμους [94].

6.3 D-WAVE

Η D-Wave Systems, Inc. είναι μια канаδική εταιρεία κβαντικής πληροφορικής, με έδρα το Μπέρναμπι. Η D-Wave ήταν η πρώτη εταιρεία του κόσμου που πουλούσε υπολογιστές για να εκμεταλλευτεί τις κβαντικές επιδράσεις στη λειτουργία τους. Οι πρώτοι πελάτες της D-Wave περιλαμβάνουν την Λόκχιντ Μάρτιν (Lockheed Martin), το Πανεπιστήμιο της Νότιας Καλιφόρνιας,

τη Google, την NASA και το εθνικό ερευνητικό εργαστήριο του Λος Άλαμος στην Καλιφόρνια.

Το 2015, ο 2X κβαντικός υπολογιστής της D-Wave με περισσότερα από 1000 qubits εγκαταστάθηκε στο εργαστήριο κβαντικής νοημοσύνης στο ερευνητικό κέντρο Έιμς της NASA. Στη συνέχεια έχουν αποστείλει συστήματα με 2048 qubits. Η D-Wave ανακοίνωσε ένα 5000 qubit σύστημα διαθέσιμο, χρησιμοποιώντας το νέο τσιπ που βασίστηκε στην αρχιτεκτονική του Πήγασου με 15 συνδέσεις ανά qubit.

Τον Ιούνιο του 2014, η D-Wave ανακοίνωσε ένα νέο οικοσύστημα κβαντικών εφαρμογών με την υπολογιστική χρηματοοικονομική εταιρεία ITB και την ομάδα έρευνας καρκίνου DNA για να επικεντρωθεί στην επίλυση προβλημάτων του πραγματικού κόσμου με κβαντικό υλικό. Στις 11 Μαΐου 2011, τα συστήματα D-Wave ανακοίνωσαν τον D-Wave One, που περιγράφεται ως "ο πρώτος εμπορικά διαθέσιμος κβαντικός υπολογιστής", που λειτουργεί σε ένα chipset 128-qubit χρησιμοποιώντας την μέθοδο της Κβαντικής Ανόπτησης, μια γενική μέθοδος για την εξεύρεση του ολικού ελάχιστου μιας συνάρτησης με μια διαδικασία που χρησιμοποιεί κβαντικές διακυμάνσεις. Το πρωτότυπο ήταν ένας επεξεργαστής Κβαντικής Ανόπτησης 16-qubit, που παρουσιάστηκε στις 13 Φεβρουαρίου 2007, στο μουσείο ιστορίας των υπολογιστών στο Mountain View της Καλιφόρνιας. Ο D-Wave απέδειξε αυτό που ισχυρίστηκε ότι είναι ένας επεξεργαστής Κβαντικής Ανόπτησης 28-qubit στις 12 Νοεμβρίου 2007. Το τσιπ κατασκευάστηκε στο εργαστήριο μικροσυσκευών της NASA στην Πασαντίνα της Καλιφόρνιας.

Τον Μάιο του 2013, μια συνεργασία μεταξύ της NASA, της Google και της διαστημικής ερευνητικής Ένωσης Πανεπιστημίων (USRA) ξεκίνησε μια ερευνητική προσπάθεια για την κβαντική τεχνητή νοημοσύνη στο εργαστήριο με βάση το σύστημα D-Wave 2 που θα χρησιμοποιηθεί για να γίνει έρευνα για τη μηχανική μάθηση, μεταξύ άλλων τομέων της μελέτης. Στις 20 Αυγούστου 2015, τα συστήματα D-Wave ανακοίνωσαν τη γενική διαθεσιμότητα του συστήματος D-Wave 2X, ενός κβαντικού υπολογιστή με παραπάνω από 1000 qubit. Ακολούθησε ανακοίνωση στις 28 Σεπτεμβρίου 2015, ότι είχε εγκατασταθεί στο εργαστήριο κβαντικής νοημοσύνης στο ερευνητικό κέντρο Έιμς της NASA. Τον Ιανουάριο του 2017, η D-Wave κυκλοφόρησε το D-Wave 2000 qubit και ένα αποθετήριο ανοιχτού κώδικα που περιείχε εργαλεία λογισμικού για κβαντικές διαδικασίες. Περιέχει το προγραμματιστικό περιβάλλον Qbsolv, το οποίο είναι ένα κομμάτι λογισμικού ανοιχτού κώδικα που λύνει τα προβλήματα του QUBO στους κβαντικούς επεξεργαστές της εταιρείας και στις κλασικές αρχιτεκτονικές του υλικού.

Ο D-Wave λειτουργούσε από διάφορες τοποθεσίες στο Βανκούβερ, τη βρετανική Κολούμπια, και εργαστηριακούς χώρους στο UBC πριν μετακομίσει στην τρέχουσα θέση του στο γειτονικό προάστιο Μπέρναμπι. Η D-Wave έχει επίσης γραφεία στο Πάλο Άλτο και τη Βιέννη. Ο πρώτος επεξεργαστής D-Wave που παράγεται για εμπορικούς σκοπούς ήταν ένα προγραμματιζόμενο, υπεραγωγίμο ολοκληρωμένο κύκλωμα με έως 128 κυκλώματα ροής. Ο επεξεργαστής 128-qubit αντικαταστάθηκε από έναν επεξεργαστή 512-qubit το 2013. Ο επεξεργαστής έχει σχεδιαστεί για την εφαρμογή μιας Κβαντικής Ανόπτησης ειδικού σκοπού, σε αντίθεση με τη λειτουργία του ως παγκοσμίου μοντέλου κβαντικού υπολογιστή.

Οι βασικές ιδέες για την προσέγγιση D-Wave προέκυψαν από πειραματικά αποτελέσματα στη φυσική της συμπυκνωμένης ύλης, και ειδικότερα τις εργασίες για την Κβαντική Ανόπτηση σε μαγνήτες που πραγματοποίησε ο Αμπιλ (Abile), ο Ρόσενμπάουμ (Rosenbaum) και οι συνεργάτες του, οι οποίοι είχαν ελέγξει τα πλεονεκτήματα, που προτάθηκαν από τον Μπίκας (Bikas). Αυτές οι ιδέες αργότερα επαναδιατυπώθηκαν στη γλώσσα του κβαντικού υπολογισμού από τους φυσικούς του MIT, Φαρχί, Λόιντ, Ορλάντο και Καμίνσκι (Farhi, Lloyd, Orlando, Kaminski) των οποίων οι δημοσιεύσεις [3, 54] το 2000 και το 2004 παρείχαν τόσο ένα θεωρητικό μοντέλο κβαντικού υπολογισμού που ταίριαζε με το προηγούμενο έργο στον κβαντικό μαγνητισμό, καθώς και μια

συγκεκριμένη ενεργοποίηση της ιδέας αυτής με την υπεραγώγιμη ροή των κομματιών. Για να κατανοήσουμε την προέλευση του μεγάλου μέρους της αντιπαράθεσης γύρω από την προσέγγιση **D-Wave**, είναι σημαντικό να σημειωθεί ότι η προέλευση της προσέγγισης **D-Wave** στον κβαντικό υπολογισμό δεν προέκυψε από το συμβατικό πεδίο κβαντικής πληροφορικής, αλλά από την πειραματική φυσική της συμπυκνωμένη ύλης.

Στις 11 Μαΐου 2011, τα συστήματα **D-Wave** ανακοίνωσαν το **D-Wave One**, ένα ολοκληρωμένο κβαντικό σύστημα υπολογιστή που λειτουργεί σε έναν επεξεργαστή **128-qubit**. Ο επεξεργαστής που χρησιμοποιείται στο **D-Wave One**, με κωδικό όνομα "Ρένιερ" εκτελεί μια μεμονωμένη μαθηματική λειτουργία, διακριτή βελτιστοποίηση. Ο Ρένιερ χρησιμοποιεί Κβαντική Ανόπτηση για να λύσει προβλήματα βελτιστοποίησης. Το **D-Wave One** ισχυρίστηκε ότι ήταν το πρώτο εμπορικά διαθέσιμο κβαντικό σύστημα υπολογιστών στον κόσμο. Η τιμή του αναφέρεται σε περίπου **10000000** δολάρια. Μια ερευνητική ομάδα με επικεφαλής τον Τροιερ (Troyer) διαπίστωσε ότι, ενώ υπάρχουν ενδείξεις Κβαντικής Ανόπτησης στο **D-Wave One**, δεν είδαν καμία αύξηση ταχύτητας σε σύγκριση με τους κλασσικούς υπολογιστές. Εφάρμοσαν ένα βελτιστοποιημένο κλασικό αλγόριθμο για να λύσει το ίδιο συγκεκριμένο πρόβλημα με το **D-Wave One**.

Στις 20 Αυγούστου 2015, η **D-Wave** κυκλοφόρησε γενική διαθεσιμότητα του υπολογιστή τους **D-Wave 2X**, με **1000 qubits** αν και, λόγω των μαγνητικών μετατοπίσεων και της παραγωγής μεταβλητότητας που είναι εγγενείς στην κατασκευή κυκλωμάτων υπεραγωγών, λιγότερα από **1152 qubits** είναι λειτουργικά και διαθέσιμα προς χρήση. Ο ακριβής αριθμός των **qubits** που προέκυψαν θα ποικίλλει ανάλογα με κάθε συγκεκριμένο μεταποιητή που κατασκευάζεται. Αυτό συνοδεύτηκε από μια έκθεση σύγκρισης ταχυτήτων με υψηλής ταχύτητας μονού νήματος **CPU**. Σε αντίθεση με τις προηγούμενες αναφορές, αυτό ανέφερε ρητώς ότι το ζήτημα της κβαντικής ταχύτητας δεν ήταν κάτι που προσπαθούσαν να αντιμετωπίσουν, και επικεντρώθηκε σε σταθερά κέρδη απόδοσης παράγοντα πάνω από το κλασικό υλικό. Για προβλήματα γενικής χρήσης, αναφέρθηκε μια επιτάχυνση του **15x**, αλλά αξίζει να σημειωθεί ότι αυτοί οι κλασσικοί αλγόριθμοι επωφελούνται αποτελεσματικά από την παραλληλοποίηση, -έτσι ώστε ο υπολογιστής να αποδίδει επί ίσοις όροις με, ίσως, **30** πυρήνες μονού νήματος υψηλής ποιότητας. Ο επεξεργαστής **D-Wave 2X** βασίζεται σε ένα **chip 2048-qubit** με το μισό από τα **qubits** να είναι απενεργοποιημένα.

6.4 Βιβλιογραφική Ανασκόπηση

Η μέθοδος της Κβαντικής ανόπτησης (**Quantum Annealing**) απέδειξε την ικανότητά της να λύνει ένα ευρύ φάσμα προβλημάτων συνδυαστικής βελτιστοποίησης, όχι μόνο στην πληροφορική, αλλά και σε άλλους τομείς, όπως η κβαντική χημεία, η βιοπληροφορική και η δρομολόγηση οχημάτων. Όλα αυτά τα προβλήματα στοχεύουν στην ελαχιστοποίηση μιας συνάρτησης κόστους, η οποία μπορεί να θεωρηθεί «σωματιδιακά» ως η εξεύρεση της κατάστασης του εδάφους (**ground state**) ενός τυπικού χαμιλτονιανού αλγορίθμου. Ωστόσο, πρόκειται για μία πολύ δύσκολη διαδικασία, να υπολογιστεί ένα ολικό ελάχιστο σε προβλήματα όπου υπάρχουν πολλαπλά τοπικά ελάχιστα.

Η πιθανή υπεροχή του κβαντικού υπολογισμού θα μπορούσε να μεταφραστεί είτε σε μια καλύτερη λύση (δηλ. πιο κοντά στη βέλτιστη) ή φτάνοντας σε μια λύση ταχύτερα ή δημιουργώντας ένα ευρύ σύνολο λύσεων (για την πολυπαραμετρική περίπτωση). Ορισμένες γνωστές περιπτώσεις όπου οι μέθοδοι αυτές λειτουργούν καλά είναι ο προγραμματισμός καταστημάτων εργασίας, η μηχανική μάθηση, η κατάτμηση γραφημάτων, η δρομολόγηση οχημάτων και ο προγραμματισμός, τα νευρωνικά δίκτυα, όπου οι παράμετροι του προβλήματος μπορούν να εκφραστούν ως μεταβλητές δυαδικών τιμών. Οι τεχνικές Κβαντικής Ανόπτησης χρησιμοποιούνται επίσης για

την αντιμετώπιση προβλημάτων πολλαπλής αντικειμενικής βελτιστοποίησης.

Από την άποψη της κλασικής προσέγγισης, οι Μπόρος κ.α. (Boros et al.) παρουσίασαν ένα σύνολο τοπικών σημείων αναζήτησης για προβλήματα δυαδικής βελτιστοποίησης (QUBO), παρέχοντας ενδεικτικά αποτελέσματα προσομοίωσης σε διάφορες δοκιμές αναφοράς [23]. Η μείωση του μεγάλου μεγέθους πινάκων στο QUBO μοντέλο ήταν το κύριο θέμα στο έργο από τον Λούις και τον Γκλόβερ (Lewis and Glover) [117]. Ο Γκλόβερ έδειξε σε μια βήμα προς βήμα διαδικασία πώς μπορεί κανείς να μεταφράσει ένα πρόβλημα με συγκεκριμένα χαρακτηριστικά στο QUBO μοντέλο [74].

Οι αλγόριθμοι κβαντικής βελτιστοποίησης (quantum approximate optimization algorithm, QAOA), με βάση τις αρχές του κβαντικού υπολογισμού, είναι μια οικογένεια αλγορίθμων που μπορούν να χρησιμοποιηθούν για την επίλυση προβλημάτων QUBO. Ο Χόι (Choi) έδειξε ότι τα προβλήματα με το QUBO μπορούν να λυθούν χρησιμοποιώντας έναν κβαντικό υπολογιστή που χρησιμοποιεί ένα χαμιλτονιανό αλγόριθμο [31]. Αυτό επιτεύχθηκε με τη μείωση, μέσω δευτερεύουσας ενσωμάτωσης, του υποκείμενου γραφήματος στο κβαντικό γράφημα υλικού. Οι Παγκάνο κ.α. (Pagano et al.) κατασκεύασαν ένα μηχανισμό που εφαρμόζει ένα ρηχό βάθος QAOA με την εκτίμηση της ενέργειας της κατάστασης του εδάφους του μοντέλου με τη χρήση ενός αναλογικού κβαντικού εξομοιωτή [144], ενώ ο Φαρχί και ο Χάρροου (Farhi, Harrow) προσπάθησαν να δείξουν τα πλεονεκτήματα των QAOA σε σύγκριση με τις κλασικές προσεγγίσεις [53]. Τα περιορισμένα προβλήματα βελτιστοποίησης που χρησιμοποιήθηκαν για την κατάργηση των μεθόδων κβαντικού πολλαπλού υπολογισμού συζητήθηκαν πρόσφατα από τον Ρέμπεντροστ κ.α. (Rebentrost et al.) [162]. Παρόμοια προσέγγιση με εκείνη που αναλύθηκε λήφθηκε από τους Βισκοσίλ και Ντιτζεβ (Vysokil and Djidjev) σχετικά με τον τρόπο εφαρμογής περιορισμών στα συστήματα QUBO. Ειδικότερα, για να αποφευχθεί η χρήση μεγάλων συντελεστών (συνεπώς, περισσότερων qubits) που προκύπτουν από τη χρήση των τεταρτηδικών περιορισμών (quadratic constraints), πρότειναν μία νέα συνδυαστική σχεδίαση που αφορά την επίλυση προβλημάτων μεικτού ακέραιου γραμμικού προγραμματισμού για να χωρέσει η εφαρμογή των επιθυμητών περιορισμών [190].

Λόγω των φυσικών περιορισμών και των επιπέδων θορύβου, δεν είναι δυνατή η εκμετάλλευση μερικών qubits και έτσι απενεργοποιούνται. Ως εκ τούτου, το υποκείμενο γράφημα είναι οριακά ελλιπές. Το διάγραμμα αρχιτεκτονικής επόμενης γενιάς της D-Wave, που ονομάζεται Πήγασος, υποτίθεται ότι θα προσφέρει μεγαλύτερη ευελιξία και εκφραστικότητα σε προηγούμενες τοπολογίες, όπως πιο αποτελεσματική ενσωμάτωση βελτιωμένων χρόνων εκτέλεσης, ενισχυμένες ενεργειακές κλίμακες, καλύτερες του χειρισμού των σφαλμάτων κ.λπ. Ομοίως, ο Ντατάνι (Dattani) παράθεσε κάποιες διαφορές μεταξύ των δύο τελευταίων αρχιτεκτονικών της Κβαντικής Ανόπτησης, δηλαδή των διαγραμμάτων της Χίμαιρας και του Πήγασου. Το D-Wave 2, το 2X και το 2000q όλα χρησιμοποίησαν το γράφημα της Χίμαιρας, το οποίο αποτελούνταν από μια μονάδα επεξεργασίας K4. Κάθε γενιά αυτού του γραφήματος έχει εξελιχθεί αξιοποιώντας όλο και περισσότερα qubits [43].

Από την άλλη πλευρά, η αρχιτεκτονική του Πήγασου άλλαξε εντελώς τη ρύθμιση προσθέτοντας πιο σύνθετη συνδεσιμότητα (κάθε qubit ή κορυφή συνδυάζεται με 15 άλλα). Αυτή η βελτιωμένη συνδεσιμότητα επιτρέπει την καλύτερη αξιοποίηση, οπότε λιγότερες κορυφές είναι ικανές για ευρύτερους υπολογισμούς [42]. Ο Ουσιζίμα-Μεσιγούα κ.α. (Ushijima-Mwesigwa et al.) απέδειξαν τον μηχανισμό διαμέρισης του γραφήματος των υπολογιστών D-Wave χρησιμοποιώντας τα εργαλεία Κβαντικής Ανόπτησης στο D-Wave 2X [186]. Μια άλλη επαναληπτική εκδοχή της Κβαντικής Ανόπτησης ευρετικής για προβλήματα QUBO με βάση την αναζήτηση Tabu παρουσιάστηκε από τον Ρόζενμπεργκ κ.α. (Rosenberg et al.) [168].

Ο Λούκας (Lucas) παρουσίασε την προσαρμογή των Άϊσινγκ (Ising) μοντελοποιήσεων και

φορμαλισμών για μια πληθώρα προβλημάτων βελτιστοποίησης κλάσης NP, με έμφαση στη χρήση όσο το δυνατόν λιγότερων qubits [120]. Οι Μάρτονακ κ.α. (martoňák et al.) εισήγαγαν ένα διαφορετικό σύστημα Κβαντικής Ανόπτησης με βάση τις ολοκληρωμένες διεργασίες του Monte Carlo για την αντιμετώπιση της συμμετρικής έκδοσης του προβλήματος του πλανόδιου πωλητή [121]. Η προσέγγισή τους βασίζεται σε μια μάλλον περιορισμένη αναπαράσταση που μοιάζει με τον τρόπο και συγκρίνεται με το πρότυπο Προσομοιωμένης Ανόπτησης ευρετικής σε διάφορες δοκιμές αναφοράς, αποδεικνύοντας την ανωτερότητά του. Τέτοια έργα επισημαίνουν το γεγονός ότι οι κβαντικές τεχνικές έχουν ήδη αποδειχθεί χρήσιμες όταν πρόκειται για συγκεκριμένες παραλλαγές του προβλήματος. Μια άλλη εργασία για την Κβαντική Ανόπτηση και το πρόβλημα του πλανόδιου πωλητή παρουσιάστηκε από τον Γουόρεν (Warren). Ο Γουόρεν μελέτησε μικρές κλίμακας περιπτώσεις του προβλήματος του πλανόδιου πωλητή, αναδεικνύοντας πώς ένα μηχάνημα D-Wave που χρησιμοποιούσε Κβαντική Ανόπτηση θα μπορούσε να λύσει αυτά τα προβλήματα [192].



7. Επίλογος

Η παραπάνω διατριβή πραγματοποιήθηκε στο πλαίσιο των διδακτορικών σπουδών μου στο Τμήμα Πληροφορικής του Ιονίου Πανεπιστημίου και πραγματεύεται τις μεθευρετικές μεθόδους βελτιστοποίησης εμπνευσμένες από τον χβαντικό υπολογισμό. Όπως είναι γνωστό στον κλάδο της συνδυαστικής βελτιστοποίησης, σε ότι έχει να κάνει με τους προσεγγιστικούς αλγόριθμους και πιο συγκεκριμένα με τις μεθευρετικές, εκτελούνται σε σύντομο χρόνο παρέχοντας μία αρκετά 'καλή' λύση ποιοτικά και ποσοτικά. Το γεγονός αυτό μας επιτρέπει να τις χρησιμοποιήσουμε σε συστήματα πραγματικού χρόνου (*real time*) που επιλύουν προβλήματα από τον πραγματικό κόσμο.

Στα πλαίσια αυτής της προσπάθειας, πραγματοποιήθηκε αναλυτική παρουσίαση της μεθόδου μας μαζί με άλλες συμβατικές μεθόδους βελτιστοποίησης. Ειδικότερα πραγματοποιήθηκαν και οι αντίστοιχες προγραμματιστικές υλοποιήσεις. Με βάση αυτές τις υλοποιήσεις, πραγματοποιήθηκαν εξονυχιστικά πειράματα με σκοπό να γίνει εκτενής υπολογιστική και συγκριτική μελέτη για να αξιολογηθεί η απόδοση του νέου προτεινόμενου μη συμβατικού αλγόριθμου μας με ήδη γνωστές μεθευρετικές παραλλαγές αλλά και με μεθευρετικές διαδικασίες που προτάθηκαν στην πρόσφατη βιβλιογραφία (2019) με τις καλύτερες μέχρι τότε επιδόσεις τις οποίες και ξεπέρασε ο δικός μας μεθευρετικός αλγόριθμος. Συνεπώς, η απόδοση όπως και η ανάλυση της υπολογιστικής μελέτης ήταν μέρος της έρευνας που πραγματοποιήθηκε. Συμπληρωματικά με την υπολογιστική μελέτη για την συμπεριφορά του αλγορίθμου μας έναντι των άλλων, για τα γνωστά μετροπροβλήματα της βιβλιοθήκης TSPLIB, πραγματοποιήθηκαν και αναλυτικά στατιστικά τεστ τα οποία επιβεβαιώνουν τα υπολογιστικά αποτελέσματα της αναλυτικής υπολογιστικής μελέτης.

Έπειτα, οι παραπάνω υλοποιήσεις χρησιμοποιήθηκαν για να επιλύσουν το πρόβλημα του πλανόδιου πωλητή (*Travelling Salesman Problem*) και το πρόβλημα του πλανόδιου πωλητή με χρονικά παράθυρα (*Travelling Salesman Problem with Time Windows*). Στη συνέχεια εφαρμόσαμε τις παραπάνω μεθευρετικές μεθόδους στο πρόβλημα της περισυλλογής απορριμάτων. Το πρόβλημα της περισυλλογής των απορριμάτων αποκτά ιδιαίτερη σημασία ειδικά στις μέρες μας αφού είναι μια καθιερωμένη διαδικασία για κάθε δήμο. Βελτιστοποιώντας την διαδικασία της δρομολόγησης των απορριμματοφόρων, μειώνεται δραστικά η κατανάλωση σε καύσιμα, αφενός αυτό καθιστά τη διαδικασία φιλική προς το περιβάλλον αφού μειώνονται οι εκπομπές των ρύπων αφετέρου μειώνε-

ται δραστικά το κόστος της διαδικασίας της συγκομιδής των απορριμάτων, με θετικές επιπτώσεις συνολικά για τον οργανισμό. Τα αποτελέσματα ήταν ενθαρρυντικά αφού η μέθοδος μας παρουσίασε καλή απόδοση παρέχοντας γρήγορα ποιοτικές λύσεις. Σαν μελλοντική έρευνα θα προσπαθήσουμε να υλοποιήσουμε αλγόριθμους κβαντικής ανόπτησης, που να μπορούν να τρέξουν σε πραγματικό κβαντικό υπολογιστή, για τα συγκεκριμένα προβλήματα από τον πραγματικό κόσμο που μοντελοποιούνται σαν προβλήματα συνδυαστικής βελτιστοποίησης.

Μερικά συμπεράσματα που προέκυψαν από την έρευνα που διενεργήθηκε:

- Οι μεθευρετικές μπορούν να παρέχουν μια 'καλή' ποιοτικά και ποσοτικά λύση
- Διαφορετικά σχήματα μεθευρετικών εμπνευσμένα από διαφορετικές διαδικασίες διαφέρουν στην ποιότητα των λύσεων
- Εξ αιτίας της ικανότητας αυτών των μεθόδων να βρίσκουν σε σύντομο χρονικό διάστημα ποιοτικές λύσεις, μπορούν αυτές να χρησιμοποιηθούν για την επίλυση περίπλοκων προβλημάτων από τον πραγματικό κόσμο
- Η κβαντικά εμπνευσμένη αναζήτηση μεταβλητής γειτονιάς, ξεπερνά σε απόδοση υλοποιήσεις από την πρόσφατη βιβλιογραφία που έδιναν τις μέχρι τώρα καλύτερες λύσεις
- Ένα πεδίο για μελλοντική έρευνα θα μπορούσε να είναι ο συνδυασμός συμβατικών και μη συμβατικών διαδικασιών για την επίλυση τέτοιων προβλημάτων
- Η γνώση και η εξονυχιστική ανάλυση τέτοιων συμβατικών και μη μεθευρετικών που θα εκτελεστούν όμως σε συμβατικό (κλασσικό) υπολογιστή δίνουν κάποια απαραίτητα εφόδια και γνώσεις στον ερευνητή για να προχωρήσει σε έρευνα και υλοποίηση των αμιγώς μη συμβατικών μεθόδων βελτιστοποίησης, όπως για παράδειγμα της κβαντικής ανόπτησης (quantum annealing)

7.1 Μελλοντική Έρευνα

Όπως φαίνεται και από το τελευταίο (έκτο) κεφάλαιο της διατριβής, στο τέλος της ερευνητικής μας προσπάθειας, μελετήσαμε μερικές εισαγωγικές έννοιες του κβαντικού υπολογισμού και κάναμε μια εκτενή βιβλιογραφική ανασκόπηση όλου του πρόσφατου συγγραφικού έργου που αφορά την κβαντική ανόπτηση. Ήδη εργαζόμαστε και ερευνούμε μεθόδους και διαδικασίες μη συμβατικού υπολογισμού και πιο συγκεκριμένα καθολικές μεθόδου κβαντικής βελτιστοποίησης, όπως η κβαντική ανόπτηση με σκοπό να υλοποιήσουμε αυτές τις διαδικασίες και να τις εκτελέσουμε σε έναν πραγματικό κβαντικό υπολογιστή. Στον κβαντικό υπολογιστή θα προσπαθήσουμε να εκτελέσουμε προγράμματα που θα επιλύουν επίσης προβλήματα από τον πραγματικό κόσμο. Η ερευνητική μας προσπάθεια βρίσκεται σε εξέλιξη και είναι σε καλό δρόμο για την ολοκλήρωσή της.

Τέλος, μέρος της μελλοντικής μας ερευνητικής προσπάθειας θα αφορά και την εφαρμογή αυτών των πρωτότυπων αλγορίθμων σε προβλήματα από τον πραγματικό κόσμο που έχουν περιβαλλοντολογικό αντίκτυπο, για παράδειγμα την βελτιστοποίηση του προβλήματος της δρομολόγησης αλλά και ανεφοδιασμού ηλεκτροκίνητων οχημάτων.



Βιβλιογραφία-Αναφορές

- [1] Scott Aaronson. “Guest column: NP-complete problems and physical reality”. In: *ACM Sigact News* 36.1 (2005), pages 30–52.
- [2] Bekir Afşar et al. “Self-adaptive and adaptive parameter control in improved artificial bee colony algorithm”. In: *Informatica* 28.3 (2017), pages 415–438 (cited on page 48).
- [3] Dorit. Aharonov et al. “Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation”. In: *SIAM Review* 50.4 (2008), pages 755–787. eprint: <https://doi.org/10.1137/080734479>. URL: <https://doi.org/10.1137/080734479> (cited on page 148).
- [4] Valia Allori. “Decoherence and the classical limit of quantum mechanics”. In: (2002).
- [5] Mohammad HS Amin and Miles FH Steining. *Adiabatic quantum computation with superconducting qubits*. US Patent 7,135,701. Nov. 2006.
- [6] Norbert Ascheuer, Matteo Fischetti, and Martin Grötschel. “Solving the asymmetric travelling salesman problem with time windows by branch-and-cut”. In: *Mathematical Programming* 90.3 (2001), pages 475–506 (cited on page 28).
- [7] George Asimakopoulos et al. “Architecture and Implementation Issues, Towards a Dynamic Waste Collection Management System”. In: *Proceedings of the 24th International Conference on World Wide Web*. ACM. 2015, pages 1383–1388.
- [8] George Asimakopoulos et al. “Architecture and Implementation Issues, Towards a Dynamic Waste Collection Management System”. In: *Proceedings of the 24th International Conference on World Wide Web*. WWW ’15 Companion. New York, NY, USA: ACM, 2015, pages 1383–1388 (cited on page 116).
- [9] Anne Auger and Olivier Teytaud. “Continuous lunches are free plus the design of optimal optimization algorithms”. In: *Algorithmica* 57.1 (2010), pages 121–146 (cited on page 57).
- [10] Charles Babbage. *Science and reform: selected works of Charles Babbage*. Cambridge University Press, 1989 (cited on page 25).

- [11] John Backus. “The history of Fortran I, II, and III”. In: *ACM Sigplan Notices* 13.8 (1978), pages 165–180 (cited on page 27).
- [12] Nils Aall Barricelli et al. “Esempi numerici di processi di evoluzione”. In: *Methodos* 6.21-22 (1954), pages 45–68 (cited on pages 54, 55).
- [13] Atidel Ben Hadj-Alouane and James C Bean. “A genetic algorithm for the multiple-choice integer program”. In: *Operations research* 45.1 (1997), pages 92–101 (cited on page 83).
- [14] Paul Benioff. “The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines”. In: *Journal of statistical physics* 22.5 (1980), pages 563–591 (cited on page 139).
- [15] Jon Louis Bentley. “Fast algorithms for geometric traveling salesman problems”. In: *ORSA Journal on computing* 4.4 (1992), pages 387–411 (cited on page 53).
- [16] Leslie R Berlin. “Robert Noyce and Fairchild Semiconductor, 1957-1968”. In: *The Business History Review* (2001), pages 63–101 (cited on page 28).
- [17] Raquel Bernardino and Ana Paias. “Solving the family traveling salesman problem”. In: *European Journal of Operational Research* 267.2 (2018), pages 453–466.
- [18] Leonora Bianchi et al. “A survey on metaheuristics for stochastic combinatorial optimization”. In: *Natural Computing* 8.2 (2009), pages 239–287 (cited on pages 53, 54).
- [19] Olav Bjerkholt. “Ragnar Frisch and the Foundation of the Econometric Society and”. In: *Econometrica* (1998), pages 26–57 (cited on page 37).
- [20] Christian Blum and Andrea Roli. “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”. In: *ACM computing surveys (CSUR)* 35.3 (2003), pages 268–308 (cited on pages 53, 54, 58, 84).
- [21] Sergio Boixo et al. “Quantum annealing with more than one hundred qubits”. In: *arXiv preprint arXiv:1304.4595* (2013).
- [22] J Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013.
- [23] Endre Boros, Peter L Hammer, and Gabriel Tavares. “Local search heuristics for quadratic unconstrained binary optimization (QUBO)”. In: *Journal of Heuristics* 13.2 (2007), pages 99–132 (cited on page 150).
- [24] R Bosch and M Trick. *Search Methodologies: Introductory tutorials in optimisation and decision support techniques, chapter Integer Programming*. 2005 (cited on page 43).
- [25] Jack Brimberg et al. “Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem”. In: *Operations research* 48.3 (2000), pages 444–460 (cited on page 63).
- [26] Roberto Wolfler Calvo. “A new heuristic for the traveling salesman problem with time windows”. In: *Transportation Science* 34.1 (2000), pages 113–124 (cited on pages 83, 134).
- [27] William B Carlton and J Wesley Barnes. “Solving the traveling-salesman problem with time windows using tabu search”. In: *IIE transactions* 28.8 (1996), pages 617–629 (cited on page 83).
- [28] Daniel Joseph Cavicchio. *Adaptive search using simulated evolution*. Technical report. 1970 (cited on pages 54, 56).

- [29] Chi-Bin Cheng and Chun-Pin Mao. “A modified ant colony system for solving the travelling salesman problem with time windows”. In: *Mathematical and Computer Modelling* 46.9-10 (2007), pages 1225–1235 (cited on page 25).
- [30] Hua-Pei Chiang et al. “A quantum-inspired Tabu search algorithm for solving combinatorial optimization problems”. In: *Soft Computing* 18.9 (2014), pages 1771–1781.
- [31] Vicky Choi. “Minor-embedding in adiabatic quantum computation: I. The parameter setting problem”. In: *Quantum Information Processing* 7.5 (2008), pages 193–209 (cited on page 150).
- [32] Vašek Chvátal and W Cook. *The traveling salesman problem*. 1985.
- [33] Florina M Ciorba, Theodore Andronikos, and George Papakonstantinou. “Adaptive cyclic scheduling of nested loops”. In: *7th Hellenic European Research on Computer Mathematics and its Applications (HERCMA'05)* (2005).
- [34] Marie Coffin and Matthew J. Saltzman. “Statistical Analysis of Computational Tests of Algorithms and Heuristics”. In: *INFORMS Journal on Computing* 12.1 (2000), pages 24–44. DOI: 10.1287/ijoc.12.1.24.11899 (cited on page 108).
- [35] Stephen A Cook. “The complexity of theorem-proving procedures”. In: *Proceedings of the third annual ACM symposium on Theory of computing*. 1971, pages 151–158 (cited on page 74).
- [36] Martin D Crossland, Bayard E Wynne, and William C Perkins. “Spatial decision support systems: An overview of technology and a test of efficacy”. In: *Decision support systems* 14.3 (1995), pages 219–235 (cited on page 116).
- [37] AV Abs da Cruz, Marley Maria Bernardes Rebuzzi Vellasco, and Marco Aurélio Cavalcanti Pacheco. “Quantum-inspired evolutionary algorithm for numerical optimization”. In: *Hybrid evolutionary algorithms*. Springer, 2007, pages 19–37.
- [38] Rodrigo Ferreira Da Silva and Sebastián Urrutia. “A General VNS heuristic for the traveling salesman problem with time windows”. In: *Discrete Optimization* 7.4 (2010), pages 203–211 (cited on pages 71, 83, 84, 131–133).
- [39] Edward D Dahl. “Programming with d-wave: Map coloring problem”. In: *D-Wave Official Whitepaper* (2013).
- [40] George B Dantzig. “Origins of the simplex method”. In: *A history of scientific computing*. 1990, pages 141–151 (cited on pages 33, 35).
- [41] Sanjeeb Dash et al. “A time bucket formulation for the traveling salesman problem with time windows”. In: *INFORMS Journal on Computing* 24.1 (2012), pages 132–147 (cited on page 80).
- [42] Nike Dattani and Nick Chancellor. “Embedding quadratization gadgets on Chimera and Pegasus graphs”. In: *arXiv preprint arXiv:1901.07676* (2019) (cited on page 150).
- [43] Nike Dattani, Szilard Szalay, and Nick Chancellor. “Pegasus: The second connectivity graph for large-scale quantum annealing hardware”. In: *arXiv preprint arXiv:1901.07636* (2019) (cited on page 150).
- [44] Christof Defryn and Kenneth Sörensen. “A fast two-level variable neighborhood search for the clustered vehicle routing problem”. In: *Computers & Operations Research* 83 (2017), pages 78–94 (cited on page 39).

- [45] David Deutsch. “Quantum theory, the Church–Turing principle and the universal quantum computer”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pages 97–117 (cited on page 139).
- [46] Sandip Dey, Siddhartha Bhattacharyya, and Ujjwal Maulik. “New quantum inspired meta-heuristic techniques for multi-level colour image thresholding”. In: *Applied Soft Computing* 46 (2016), pages 677–702. ISSN: 1568-4946. DOI: <http://dx.doi.org/10.1016/j.asoc.2015.09.042>. URL: <http://www.sciencedirect.com/science/article/pii/S1568494615006146> (cited on pages 38, 64, 87).
- [47] *Directions API* (cited on page 135).
- [48] Marco Dorigo. “Optimization, learning and natural algorithms”. In: *PhD Thesis, Politecnico di Milano* (1992) (cited on pages 43, 47, 55, 57, 59).
- [49] Yvan Dumas et al. “An optimal algorithm for the traveling salesman problem with time windows”. In: *Operations research* 43.2 (1995), pages 367–371 (cited on pages 23, 71, 80, 83).
- [50] Jr John Presper Eckert and John W Mauchly. *Electronic numerical integrator and computer*. US Patent 3,120,606. Feb. 1964 (cited on page 26).
- [51] Federico Faggin et al. “The History of the 4004”. In: *Ieee Micro* 16.6 (1996), pages 10–20 (cited on page 29).
- [52] Wei Fang et al. “A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population”. In: *Information Sciences* 330 (2016), pages 19–48. DOI: 10.1016/j.ins.2015.09.055 (cited on page 39).
- [53] Edward Farhi and Aram W Harrow. “Quantum supremacy through the quantum approximate optimization algorithm”. In: *arXiv preprint arXiv:1602.07674* (2016) (cited on page 150).
- [54] Edward Farhi et al. “Quantum computation by adiabatic evolution”. In: *arXiv preprint quant-ph/0001106* (2000) (cited on page 148).
- [55] J Doyne Farmer, Norman H Packard, and Alan S Perelson. “The immune system, adaptation, and machine learning”. In: *Physica D: Nonlinear Phenomena* 22.1-3 (1986), pages 187–204 (cited on page 43).
- [56] Dominique Feillet, Pierre Dejax, and Michel Gendreau. “Traveling salesman problems with profits”. In: *Transportation science* 39.2 (2005), pages 188–205.
- [57] Richard P. Feynman. “Simulating physics with computers”. In: *International Journal of Theoretical Physics Int J Theor Phys* 21.6-7 (1982), pages 467–488. DOI: 10.1007/bf02650179 (cited on pages 64, 88).
- [58] Richard Phillips Feynman, JG Hey, and Robin W Allen. “Feynman lectures on computation”. In: (1998) (cited on pages 64, 88).
- [59] Filippo Focacci, Andrea Lodi, and Michela Milano. “A hybrid exact algorithm for the TSPTW”. In: *INFORMS Journal on Computing* 14.4 (2002), pages 403–417 (cited on page 83).
- [60] Lawrence J Fogel, Alvin J Owens, and Michael J Walsh. “Artificial intelligence through simulated evolution”. In: (1966) (cited on pages 43, 54, 56).

- [61] Carl Franklin. “An Introduction to Geographic Information Systems: Linking Maps to Databases”. In: *Database* 15.2 (Apr. 1992), pages 12–21. ISSN: 0162-4105 (cited on page 115).
- [62] Gustavo Erick Anaya Fuentes et al. “Solution to travelling salesman problem by clusters and a modified multi-restart iterated local search metaheuristic”. In: *PloS one* 13.8 (2018), e0201868.
- [63] MR Gary and DS Johnson. “Computers and Intractability: A Guide to the Theory of NPCompleteness WH Freeman and Co”. In: *New York* (1979) (cited on pages 67, 74, 76).
- [64] Zong Woo Geem, Joong Hoon Kim, and Gobichettipalayam Vasudevan Loganathan. “A new heuristic optimization algorithm: harmony search”. In: *simulation* 76.2 (2001), pages 60–68 (cited on page 50).
- [65] Michel Gendreau et al. “A generalized insertion heuristic for the traveling salesman problem with time windows”. In: *Operations Research* 46.3 (1998), pages 330–335 (cited on pages 71, 83).
- [66] K. Giannakis, C. Papalitsas, and T. Andronikos. “Quantum automata for infinite periodic words”. In: *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*. 2015, pages 1–6.
- [67] Konstantinos Giannakis et al. “Particular biomolecular processes as computing paradigms”. In: *GeNeDis 2018*. Springer, 2020, pages 225–238.
- [68] Kenneth C Gilbert and Ruth B Hofstra. “A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem”. In: *Decision Sciences* 23.1 (1992), pages 250–259 (cited on page 78).
- [69] Paul C Gilmore and Ralph E Gomory. “Sequencing a one state-variable machine: A solvable case of the traveling salesman problem”. In: *Operations research* 12.5 (1964), pages 655–679 (cited on page 74).
- [70] Fred Glover. “Heuristics for integer programming using surrogate constraints”. In: *Decision sciences* 8.1 (1977), pages 156–166 (cited on pages 55, 56, 58).
- [71] Fred Glover. “Future paths for integer programming and links to artificial intelligence”. In: *Computers operations research* 13.5 (1986), pages 533–549 (cited on pages 55, 57, 58).
- [72] Fred Glover. “Tabu search—part I”. In: *ORSA Journal on computing* 1.3 (1989), pages 190–206 (cited on pages 49, 57).
- [73] Fred W Glover and Gary A Kochenberger. *Handbook of metaheuristics*. Volume 57. Springer Science & Business Media, 2006 (cited on pages 43, 52, 54).
- [74] Fred Glover, Gary Kochenberger, and Yu Du. “A tutorial on formulating and using qubo models”. In: *arXiv preprint arXiv:1811.11538* (2018) (cited on pages 66, 150).
- [75] David E Goldberg. “Genetic algorithms in search”. In: *Optimization, and Machine Learning* (1989) (cited on page 54).
- [76] AT DATA STUDENT BASED ON GPA. “Analysis and implementation of algorithm clustering affinity propagation and k-means at data student based on GPA and duration of bachelor-thesis completion”. In: *Journal of Theoretical and Applied Information Technology* 35.1 (2012).

- [77] I Grattan-Guinness. “Joseph Fourier’s anticipation of linear programming”. In: *Journal of the Operational Research Society* 21.3 (1970), pages 361–364 (cited on page 33).
- [78] Lov K Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pages 212–219 (cited on page 142).
- [79] Gregory Gutin and Abraham P Punnen. *The traveling salesman problem and its variations*. Volume 12. Springer Science & Business Media, 2006.
- [80] Paul Isaac Hagouel and Ioannis G Karafyllidis. “Quantum computers: Registers, gates and algorithms”. In: *2012 28th International Conference on Microelectronics Proceedings*. IEEE. 2012, pages 15–21.
- [81] Paul Isaac Hagouel and Ioannis G. Karafyllidis. “Quantum computers: Registers, gates and algorithms”. In: *2012 28th International Conference on Microelectronics Proceedings*. Institute of Electrical and Electronics Engineers (IEEE), May 2012. DOI: 10.1109/miel.2012.6222789. URL: <http://dx.doi.org/10.1109/MIEL.2012.6222789> (cited on page 65).
- [82] A Hanif Halim and I Ismail. “Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem”. In: *Archives of Computational Methods in Engineering* 26.2 (2019), pages 367–380 (cited on pages 112, 113).
- [83] Pierre Hansen and Nenad Mladenovic. *A tutorial on variable neighborhood search*. Groupe d’études et de recherche en analyse des décisions, HEC Montréal, 2003 (cited on page 63).
- [84] Pierre Hansen and Nenad Mladenović. “Variable neighborhood search: Principles and applications”. In: *European journal of operational research* 130.3 (2001), pages 449–467 (cited on pages 60, 63).
- [85] Pierre Hansen et al. “Variable neighborhood search: basics and variants”. In: *EURO Journal on Computational Optimization* 5.3 (2017), pages 423–454.
- [86] W Keith Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: (1970) (cited on pages 54, 56).
- [87] Julian Havil. *John Napier: life, logarithms, and legacy*. Princeton University Press, 2014 (cited on page 25).
- [88] Anthony Hey. *Feynman and computation*. CRC Press, 2018.
- [89] JH Holland. “adaptation in natural and artificial systems, university of michigan press,” in: *University of Michigan press* 100 (1975) (cited on pages 43, 45, 54, 58).
- [90] Samrat Hore, Aditya Chatterjee, and Anup Dewanji. “Improving variable neighborhood search to solve the traveling salesman problem”. In: *Applied Soft Computing* 68 (2018), pages 83–91 (cited on pages 112, 113).
- [91] Sandra Huber and Martin Josef Geiger. “Order matters—A Variable Neighborhood Search for the Swap-Body Vehicle Routing Problem”. In: *European Journal of Operational Research* 263.2 (2017), pages 419–445 (cited on page 25).
- [92] Georges Ifrah et al. *The universal history of computing: From the abacus to quantum computing*. John Wiley & Sons, Inc., 2000 (cited on pages 24, 31).

- [93] Jeff Jones and Andrew Adamatzky. “Computation of the travelling salesman problem by a shrinking blob”. In: *Natural Computing* 13.1 (2014), pages 1–16 (cited on page 27).
- [94] Tadashi Kadowaki and Hidetoshi Nishimori. “Quantum annealing in the transverse Ising model”. In: *Physical Review E* 58.5 (1998), page 5355 (cited on pages 146, 147).
- [95] Leonid V Kantorovich. “Mathematical methods of organizing and planning production”. In: *Management science* 6.4 (1960), pages 366–422 (cited on page 33).
- [96] Dervis Karaboga. *An idea based on honey bee swarm for numerical optimization*. Technical report. Technical report-tr06, Erciyes university, engineering faculty, computer ..., 2005 (cited on pages 44, 48).
- [97] Dervis Karaboga. “Artificial bee colony algorithm”. In: *scholarpedia* 5.3 (2010), page 6915 (cited on page 59).
- [98] P. Karakostas, A. Sifaleras, and C.M. Georgiadis. “Adaptive GVNS Heuristics for Solving the Pollution Location Inventory Routing Problem”. In: *Learning and Intelligent Optimization. LION 2019. Lecture Notes in Computer Science*. Edited by N. Matsatsinis, Y. Marinakis, and P. Pardalos. Volume 11968. Springer, Cham, 2020, pages 157–170.
- [99] P. Karakostas et al. “Optimization of CAR T-cell therapies Supply Chains”. In: *Computers & Chemical Engineering* 139 (2020). DOI: 10.1016/j.compchemeng.2020.106913.
- [100] Panagiotis Karakostas, Angelo Sifaleras, and Michael C Georgiadis. “Basic VNS algorithms for solving the pollution location inventory routing problem”. In: *International Conference on Variable Neighborhood Search*. Springer. 2018, pages 64–76.
- [101] Panagiotis Karakostas, Angelo Sifaleras, and Michael C Georgiadis. “A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing”. In: *Computers & Chemical Engineering* 126 (2019), pages 263–279.
- [102] Panagiotis Karakostas, Angelo Sifaleras, and Michael C Georgiadis. “Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem”. In: *Expert Systems with Applications* (2020), page 113444.
- [103] Richard M Karp. “Reducibility among combinatorial problems”. In: *Complexity of computer computations*. Springer, 1972, pages 85–103 (cited on page 74).
- [104] Peter B Keenan. “Spatial decision support systems for vehicle routing”. In: *Decision Support Systems* 22.1 (1998), pages 65–71 (cited on page 116).
- [105] Peter Bernard Keenan and Piotr Jankowski. “Spatial Decision Support Systems: Three decades on”. In: *Decision Support Systems* 116 (2019), pages 64–76. ISSN: 0167-9236. DOI: <https://doi.org/10.1016/j.dss.2018.10.010>. URL: <http://www.sciencedirect.com/science/article/pii/S0167923618301672> (cited on pages 129, 130).
- [106] James Kennedy and Russell Eberhart. “Particle swarm optimization”. In: *Proceedings of ICNN’95-International Conference on Neural Networks*. Volume 4. IEEE. 1995, pages 1942–1948 (cited on pages 43, 48).
- [107] Brian W Kernighan and Shen Lin. “An efficient heuristic procedure for partitioning graphs”. In: *The Bell system technical journal* 49.2 (1970), pages 291–307 (cited on pages 54, 56).

- [108] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. “Optimization by simulated annealing”. In: *science* 220.4598 (1983), pages 671–680 (cited on pages 55, 56, 58).
- [109] John R Koza and John R Koza. *Genetic programming: on the programming of computers by means of natural selection*. Volume 1. MIT press, 1992 (cited on page 43).
- [110] Archana Kurian, J Mathew, and PA Kumar. “Study on Computational Complexity Classes”. In: *International Journal of Engineering Technology, Management and Applied Sciences* 3.3 (2015), pages 219–225.
- [111] Thaddeus D Ladd et al. “Quantum computers”. In: *Nature* 464.7285 (2010), page 45.
- [112] André Langevin et al. “A two-commodity flow formulation for the traveling salesman and the makespan problems with time windows”. In: *Networks* 23.7 (1993), pages 631–640 (cited on pages 71, 83).
- [113] Trevor Lanting et al. “Entanglement in a quantum annealing processor”. In: *Physical Review X* 4.2 (2014), page 021041.
- [114] Eugene L Lawler. “The traveling salesman problem: a guided tour of combinatorial optimization”. In: *Wiley-Interscience Series in Discrete Mathematics* (1985) (cited on pages 67, 69, 70, 74).
- [115] Robert J Leonard. “From parlor games to social science: von Neumann, Morgenstern, and the creation of game theory 1928-1944”. In: *Journal of economic literature* 33.2 (1995), pages 730–761 (cited on page 34).
- [116] Leonid Anatolevich Levin. “Universal sequential search problems”. In: *Problemy peredachi informatsii* 9.3 (1973), pages 115–116 (cited on page 74).
- [117] Mark Lewis and Fred Glover. “Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis”. In: *Networks* 70.2 (2017), pages 79–97 (cited on pages 24, 150).
- [118] Gengcheng Liu, Shiji Song, and Cheng Wu. “Two techniques to improve the NEH algorithm for flow-shop scheduling problems”. In: *International Conference on Intelligent Computing*. Springer. 2011, pages 41–48.
- [119] Tzyy-Chyang Lu and Jyh-Ching Juang. “Quantum-inspired space search algorithm (QSSA) for global numerical optimization”. In: *Applied Mathematics and Computation* 218.6 (2011), pages 2516–2532. DOI: 10.1016/j.amc.2011.07.067 (cited on page 39).
- [120] Andrew Lucas. “Ising formulations of many NP problems”. In: *Frontiers in Physics* 2 (2014), page 5 (cited on pages 24, 151).
- [121] Roman Martoňák, Giuseppe E Santoro, and Erio Tosatti. “Quantum annealing of the traveling-salesman problem”. In: *Physical Review E* 70.5 (2004), page 057701 (cited on page 151).
- [122] Rajesh Matai, Surya Prakash Singh, and Murari Lal Mittal. “Traveling salesman problem: an overview of applications, formulations, and solution approaches”. In: *Traveling salesman problem, theory and applications* 1 (2010) (cited on pages 73, 75–78).
- [123] J Matyas. “Random optimization”. In: *Automation and Remote control* 26.2 (1965), pages 246–253 (cited on pages 54, 55).
- [124] Catherine C McGeoch. “Adiabatic quantum computation and quantum annealing: Theory and practice”. In: *Synthesis Lectures on Quantum Computing* 5.2 (2014), pages 1–93.

- [125] Robert E Mercer and JR Sampson. “Adaptive search using a reproductive meta-plan”. In: *Kybernetes* (1978) (cited on pages 55, 56).
- [126] N David Mermin. *Quantum computer science: an introduction*. Cambridge University Press, 2007 (cited on page 141).
- [127] Nenad Mladenović and Pierre Hansen. “Variable neighborhood search”. In: *Computers & operations research* 24.11 (1997), pages 1097–1100 (cited on pages 60, 61).
- [128] Nenad Mladenović, Raca Todosijević, and Dragan Urošević. “An efficient GVNS for solving traveling salesman problem with time windows”. In: *Electronic Notes in Discrete Mathematics* 39 (2012), pages 83–90 (cited on pages 63, 80, 84, 131).
- [129] Nenad Mladenović, Raca Todosijević, and Dragan Urošević. “Less is more: basic variable neighborhood search for minimum differential dispersion problem”. In: *Information Sciences* 326 (2016), pages 160–171.
- [130] Christopher Monroe et al. “A “Schrödinger cat” superposition state of an atom”. In: *Science* 272.5265 (1996), pages 1131–1136 (cited on page 140).
- [131] Chiung Moon et al. “An efficient genetic algorithm for the traveling salesman problem with precedence constraints”. In: *European Journal of Operational Research* 140.3 (2002), pages 606–617.
- [132] Pablo Moscato et al. “On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms”. In: *Caltech concurrent computation program, C3P Report* 826 (1989), page 1989 (cited on pages 55, 57).
- [133] Jean-Claude Muller. “Guest editorial latest developments in GIS/LIS”. In: *International Journal of Geographical Information Science* 7.4 (1993), pages 293–303 (cited on page 115).
- [134] Mariana Nagy and Dana Negru. “Using clustering software for exploring spatial and temporal patterns in noncommunicable diseases”. In: *European Scientific Journal* 10.33 (2014), page 3747.
- [135] Sunil Nakrani and Craig Tovey. “On honey bees and dynamic server allocation in internet hosting centers”. In: *Adaptive Behavior* 12.3-4 (2004), pages 223–240 (cited on page 44).
- [136] John A Nelder and Roger Mead. “A simplex method for function minimization”. In: *The computer journal* 7.4 (1965), pages 308–313 (cited on pages 54, 56).
- [137] Florian Neukart et al. “Traffic flow optimization using a quantum annealer”. In: *Frontiers in ICT* 4 (2017), page 29.
- [138] Hartmut Neven et al. “QBoost: Large Scale Classifier Training with Adiabatic Quantum Optimization.” In: *ACML*. 2012, pages 333–348.
- [139] Gordon F Newell and Elliott W Montroll. “On the theory of the Ising model of ferromagnetism”. In: *Reviews of Modern Physics* 25.2 (1953), page 353 (cited on page 66).
- [140] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information (10th Anniv. Version)*. 2010.
- [141] De Castro Leandro Nunes. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. Chapman Hall/CRC, 2006 (cited on pages 64, 87).

- [142] Jeffrey W Ohlmann and Barrett W Thomas. “A compressed-annealing heuristic for the traveling salesman problem with time windows”. In: *INFORMS Journal on Computing* 19.1 (2007), pages 80–90 (cited on pages 81, 83, 84, 134).
- [143] Larry Owens. “Vannevar Bush and the differential analyzer: the text and context of an early computer”. In: *Technology and culture* 27.1 (1986), pages 63–95 (cited on page 26).
- [144] G Pagano et al. “Quantum Approximate Optimization with a Trapped-Ion Quantum Simulator”. In: *arXiv preprint arXiv:1906.02700* (2019) (cited on page 150).
- [145] Christos Papalitsas and Theodore Andronikos. “Unconventional GVNS for Solving the Garbage Collection Problem with Time Windows”. In: *Technologies* 7.3 (2019), page 61.
- [146] Christos Papalitsas, Theodore Andronikos, and Panayiotis Karakostas. “Studying the Impact of Perturbation Methods on the Efficiency of GVNS for the ATSP”. In: *International Conference on Variable Neighborhood Search*. Springer. 2018, pages 287–302.
- [147] Christos Papalitsas, Panayiotis Karakostas, and Theodore Andronikos. “Studying the impact of perturbation methods on the efficiency of GVNS for the ATSP”. In: *LNCS Proc. of the 6th International Conference on Variable Neighborhood Search (ICVNS 2018)* (cited on pages 92–95, 108).
- [148] Christos Papalitsas, Panayiotis Karakostas, and Kalliopi Kastampolidou. “A Quantum Inspired GVNS: Some Preliminary Results”. In: *GeNeDis 2016*. Edited by Panayiotis Vlamos. Cham: Springer International Publishing, 2017, pages 281–289. ISBN: 978-3-319-56246-9 (cited on pages 87, 122, 123, 131).
- [149] Christos Papalitsas, Panayiotis Karakostas, and Kalliopi Kastampolidou. “A quantum inspired GVNS: some preliminary results”. In: *GeNeDis 2016*. Springer, 2017, pages 281–289 (cited on page 72).
- [150] Christos Papalitsas et al. “Initialization methods for the TSP with time windows using variable neighborhood search”. In: *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE. 2015, pages 1–6 (cited on pages 132, 133).
- [151] Christos Papalitsas et al. “Combinatorial GVNS (general variable neighborhood search) optimization for dynamic garbage collection”. In: *Algorithms* 11.4 (2018), page 38 (cited on pages 39, 72, 79, 129).
- [152] Christos Papalitsas et al. “A qubo model for the traveling salesman problem with time windows”. In: *Algorithms* 12.11 (2019), page 224 (cited on page 66).
- [153] C Papalitsas et al. “Initialization methods for the TSP with Time Windows using qGVNS”. In: *Proceedings of the 6th International Symposium on Operational Research, OR in the Digital Era—ICT Challenges, Thessaloniki, Greece*. 2017, pages 8–10.
- [154] Panos M Pardalos and Gregory P Rodgers. “Computational aspects of a branch and bound algorithm for quadratic zero-one programming”. In: *Computing* 45.2 (1990), pages 131–144.
- [155] Gheorghe Paun. *Membrane computing: an introduction*. Springer Science & Business Media, 2012 (cited on page 31).
- [156] Alejandro Perdomo-Ortiz et al. “A quantum annealing approach for fault detection and diagnosis of graph-based systems”. In: *The European Physical Journal Special Topics* 224.1 (2015), pages 131–148.

- [157] DT Pham et al. “The bees algorithm”. In: *Technical Note, Manufacturing Engineering Centre, Cardiff University, UK* (2005) (cited on pages 44, 48).
- [158] Fernando Guilherme Silvano Lobo Pimentel. “Double-ended nearest and loneliest neighbour—a nearest neighbour heuristic variation for the travelling salesman problem”. In: *Revista de Ciências da Computação* 6.6 (2016) (cited on page 120).
- [159] Jarryd J Pla et al. “A single-atom electron spin qubit in silicon”. In: *Nature* 489.7417 (2012), page 541.
- [160] Michaël Defoin Platel, Stefan Schliebs, and Nikola Kasabov. “Quantum-inspired evolutionary algorithm: a multimodel EDA”. In: *IEEE Transactions on Evolutionary Computation* 13.6 (2008), pages 1218–1232.
- [161] LA Rastrigin. “The convergence of the random search method in the extremal control of a many parameter system”. In: *Automaton & Remote Control* 24 (1963), pages 1337–1342 (cited on pages 54, 55).
- [162] Patrick Rebentrost et al. “Quantum gradient descent and Newton’s method for constrained polynomial optimization”. In: *New Journal of Physics* (2019) (cited on page 150).
- [163] César Rego et al. “Traveling salesman problem heuristics: Leading methods, implementations and latest advances”. In: *European Journal of Operational Research* 211.3 (2011), pages 427–441. ISSN: 0377-2217. DOI: <http://dx.doi.org/10.1016/j.ejor.2010.09.010>. URL: <http://www.sciencedirect.com/science/article/pii/S0377221710006065> (cited on page 73).
- [164] César Rego et al. “Traveling salesman problem heuristics: Leading methods, implementations and latest advances”. In: *European Journal of Operational Research* 211.3 (2011), pages 427–441 (cited on page 72).
- [165] Gerhard Reinelt. *TSPLIB*. <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>. [Online; accessed 25-September-2017]. 2013 (cited on page 118).
- [166] Weiya Ren, Guohui Li, and Dan Tu. “Graph clustering by congruency approximation”. In: *IET Computer Vision* 9.6 (2015), pages 841–849.
- [167] Herbert Robbins and Sutton Monro. “A stochastic approximation method”. In: *The annals of mathematical statistics* (1951), pages 400–407 (cited on pages 54, 55).
- [168] Gili Rosenberg et al. “Building an iterative heuristic solver for a quantum annealer”. In: *Computational Optimization and Applications* 65.3 (2016), pages 845–869 (cited on page 150).
- [169] MW Salvesbergh. “Local search in routing problems with time windows”. In: *Annals of Operations Research* 4.1 (1985), pages 285–305 (cited on page 80).
- [170] Martin WP Savelsbergh. “Local search in routing problems with time windows”. In: *Annals of Operations research* 4.1 (1985), pages 285–305 (cited on page 81).
- [171] Juan Carlos Seck Tuoh Mora. “Improving the Multi-Restart Local Search Algorithm by Permutation Matrices and Sorted Completion Times for the Flow Shop Scheduling Problem”. In: (2014).
- [172] Claude E Shannon. “A symbolic analysis of relay and switching circuits”. In: *Electrical Engineering* 57.12 (1938), pages 713–723 (cited on page 30).

- [173] Peter W Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM review* 41.2 (1999), pages 303–332 (cited on page 142).
- [174] John Silberholz and Bruce Golden. “Comparison of metaheuristics”. In: *Handbook of metaheuristics*. Springer, 2010, pages 625–640 (cited on page 60).
- [175] Alexandros Singh et al. “Methods and patterns for user-friendly quantum programming”. In: *GeNeDis 2016*. Springer, 2017, pages 201–210.
- [176] Ekta Singhal, Shalu Singh, and Aneesh Dayma. “An Improved Heuristic for Permutation Flow Shop Scheduling”. In: *NEH ALGORITHM. International Journal of Computational Engineering Research* 2.6 (2012), pages 95–100.
- [177] R Sivaraj, T Ravichandran, and R Devi Priya. “Solving travelling salesman problem using clustering genetic algorithm”. In: *International Journal on Computer Science and Engineering (IJCSE)* 4.07 (2012), pages 1310–1317.
- [178] Stephen Frederick Smith. “A learning system based on genetic adaptive algorithms”. In: (1980) (cited on pages 55, 56).
- [179] Rainer Storn and Kenneth Price. “Minimizing the real functions of the ICEC’96 contest by differential evolution”. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. IEEE. 1996, pages 842–844 (cited on pages 43, 47).
- [180] Rainer Storn and Kenneth Price. “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces”. In: *Journal of global optimization* 11.4 (1997), pages 341–359 (cited on pages 43, 47).
- [181] Jeeu Fong Sze, Said Salhi, and Niaz Wassan. “A hybridisation of adaptive variable neighbourhood search and large neighbourhood search: Application to the vehicle routing problem”. In: *Expert Systems with Applications* 65 (2016), pages 383–397 (cited on page 39).
- [182] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. Volume 74. John Wiley & Sons, 2009 (cited on pages 54, 58, 59).
- [183] Hichem Talbi, Amer Draa, and Mohamed Batouche. “A new quantum-inspired genetic algorithm for solving the travelling salesman problem”. In: *2004 IEEE International Conference on Industrial Technology, 2004. IEEE ICIT’04*. Volume 3. IEEE. 2004, pages 1192–1197.
- [184] Aman Toor. “An advanced clustering algorithm (ACA) for clustering large data set to achieve high dimensionality”. In: *Global Journal of Computer Science and Technology* (2014).
- [185] Alan Mathison Turing. “Systems of logic based on ordinals”. In: *Proceedings of the London Mathematical Society, Series 2* 45 (1939), pages 161–228 (cited on page 29).
- [186] Hayato Ushijima-Mwesigwa, Christian FA Negre, and Susan M Mniszewski. “Graph partitioning using quantum annealing on the D-Wave system”. In: *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*. ACM. 2017, pages 22–29 (cited on page 150).
- [187] Marjolein Veenstra et al. “The pickup and delivery traveling salesman problem with handling costs”. In: *European Journal of Operational Research* 257.1 (2017), pages 118–132.
- [188] Bernhard Friedrich Voigt. “Der Handlungsreisende, wie er sein soll und was er zu thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiss zu sein”. In: *Commis-Voageur, Ilmenau* (1831) (cited on pages 69, 72).

- [189] Tomas Vyskocil and Hristo Djidjev. “Embedding Equality Constraints of Optimization Problems into a Quantum Annealer”. In: *Algorithms* 12.4 (2019), page 77 (cited on page 23).
- [190] Tomas Vyskocil and Hristo Djidjev. “Embedding Equality Constraints of Optimization Problems into a Quantum Annealer”. In: *Algorithms* 12.4 (2019). ISSN: 1999-4893. DOI: 10.3390/a12040077. URL: <https://www.mdpi.com/1999-4893/12/4/77> (cited on page 150).
- [191] Tomáš Vyskočil, Scott Pakin, and Hristo N Djidjev. “Embedding Inequality Constraints for Quantum Annealing Optimization”. In: *International Workshop on Quantum Technology and Optimization Problems*. Springer. 2019, pages 11–22.
- [192] Richard H Warren. “Adapting the traveling salesman problem to an adiabatic quantum computer”. In: *Quantum information processing* 12.4 (2013), pages 1781–1785 (cited on page 151).
- [193] Richard H Warren. “Small traveling salesman problems”. In: *Journal of Advances in Applied Mathematics* 2.2 (2017).
- [194] Wikipedia. accessed:2020 (cited on page 59).
- [195] Jarrod W Wilcox. “Harry Markowitz and the discretionary wealth hypothesis”. In: *The Journal of Portfolio Management* 29.3 (2003), pages 58–65 (cited on page 37).
- [196] Jan C Willems. “1696: The birth of optimal control”. In: *Proceedings of 35th IEEE Conference on Decision and Control*. Volume 2. IEEE. 1996, pages 1586–1587 (cited on page 36).
- [197] David H Wolpert, William G Macready, et al. *No free lunch theorems for search*. Technical report. Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995 (cited on pages 55, 57).
- [198] Qiuyi Wu et al. “A novel quantum-inspired immune clonal algorithm with the evolutionary game approach”. In: *Progress in Natural Science* 19.10 (2009), pages 1341–1347 (cited on page 39).
- [199] Yan WU and Jian-jun BAO. “New mixed quantum-inspired evolutionary algorithm for TSP [J]”. In: *Journal of Computer Applications* 10 (2006).
- [200] Xin-She Yang. “Metaheuristic optimization”. In: *Scholarpedia* 6.8 (2011), page 11472 (cited on pages 53, 54).
- [201] Tianmin Zheng and Mitsuo Yamashiro. “A novel hybrid quantum-inspired evolutionary algorithm for permutation flow-shop scheduling”. In: *Journal of Statistics and Management Systems* 12.6 (2009), pages 1165–1182.



Γλωσσάρι Ξενικών Όρων

Table 7.1: Γλωσσάρι Ξενικών όρων

English:	Greek:
Metaheuristic	Μεθευρετική
Heuristic	Ευρετική
Variable Neighborhood Search	Αναζήτηση Μεταβλητής Γειτονιάς
General Variable Neighborhood Search	Γενική Αναζήτηση Μεταβλητής Γειτονιάς
Quantum Inspired Optimization Algorithms	Κβαντικά Εμπνευσμένοι Αλγόριθμοι Βελτιστοποίησης
Quantum Computing	Κβαντικός Υπολογισμός
Quantum Computer	Κβαντικός Υπολογιστής
Quantum Annealing	Κβαντική Ανόπτηση
Travelling Salesman Problem	Το Πρόβλημα του Πλανόδιου Πωλητή
Travelling Salesman Problem with Time Windows	Το Πρόβλημα του Πλανόδιου Πωλητή με Χρονικά Παράθυρα
Nearest Neighborhood	η μέθοδος αναζήτησης του Πλησιέστερου Γείτονα
Geographical Information Systems	Συστήματα Γεωπληροφόρησης
Optimal Value (OV)	Βέλτιστη τιμή
Superposition	Υπέρθωση
Entanglement	Διεμπλοκή