

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
Σχολή Μηχανικών Παραγωγής και Διοίκησης

**Χρήση Εξελικτικών Αλγορίθμων και  
Αλγορίθμων Εμπνευσμένων από τη Φύση  
για την Επίλυση Προβλημάτων  
Δρομολόγησης Οχημάτων με Κέρδος**

**Μεταπτυχιακή Διατριβή**

**Δήμητρα Τραχανατζή**

Επιβλέπων  
Ιωάννης Μαρινάκης

**Χανιά, Ιούλιος 2017**

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
Σχολή Μηχανικών Παραγωγής και Διοίκησης

**Χρήση Εξελικτικών Αλγορίθμων και  
Αλγορίθμων Εμπνευσμένων από τη Φύση  
για την Επίλυση Προβλημάτων  
Δρομολόγησης Οχημάτων με Κέρδος**

**Μεταπτυχιακή Διατριβή**

**Δήμητρα Τραχανατζή**

Εξεταστική Επιτροπή  
Ιωάννης Μαρινάκης, Επίκουρος Καθηγητής (Επιβλέπων)  
Νικόλαος Ματσατσίνης, Καθηγητής  
Στέλιος Τσαφάρáκης, Επίκουρος Καθηγητής

**Χανιά, Ιούνιος 2017**

**TECHNICAL UNIVERSITY OF CRETE**  
**School of Production Engineering and Management**

**Evolutionary and Nature Inspired  
Algorithms for Solving Vehicle Routing  
Problems with Profits**

**Master Thesis**

**Dimitra Trachanatzi**

Supervisor  
Yannis Marinakis

**Chania, July 2017**

*Στον αδερφό μου  
Γιώργο*

## Περίληψη

Η συγκεκριμένη μεταπτυχιακή διατριβή προτείνει δύο υβριδικούς αλγορίθμους για την επίλυση προβλημάτων δρομολόγησης οχημάτων με κέρδος. Συγκεκριμένα, προτείνεται το μοντέλο Distance Related, που αφορά στη μετατροπή της κωδικοποίηση των λύσεων, από διακριτή αναπαράσταση κόμβων, σε συνεχή τιμές (βάση της ευκλείδειας απόστασης μεταξύ διαδοχικών κόμβων) και αντίστροφα. Το μοντέλο αυτό ενσωματώνεται στον αλγόριθμο τεχνητής αποικίας μελισσών (ABC) και στον αλγόριθμο της διαφορικής εξέλιξης (DE) και παρουσιάζονται οι αλγόριθμοι DRABC και DRDE. Τα προβλήματα που επιλύουν οι προτεινόμενοι αλγόριθμοι είναι: το πρόβλημα προσανατολισμού ομάδας (TOP), το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (CTOP) και το πρόβλημα δρομολόγησης οχημάτων με συλλογή βραβείου (PCVRP). Τα πρόβληματα αυτά χρησιμοποιούνται για την μοντελοποίηση του προβλήματος σχεδιασμού τουριστικών διαδρομών (TTDP). Ακόμα περιλαμβάνεται μία πρωτόπορα προσέγγιση του προβλήματος TTDP, αφού εξετάζεται ο σχεδιασμός τουριστικών διαδρομών για άτομα μίας ομάδας με διαφορετικές προτιμήσεις. Η επέκταση αυτή αντιμετωπίζεται με τη χρήση του παιγνίου: N ατόμων Μάχη των Φύλων (N-person BOS), το οποίο παρέχει μία συνολική βαθμολόγηση των κόμβων μιας τουριστικής διαδρομής, βάση των προτιμήσεων όλων των μελών της ομάδας.

### Λέξεις κλειδιά

Αλγόριθμος Τεχνητής Αποικίας Μελισσών, Αλγόριθμος Διαφορικής Εξέλιξης, N ατόμων Μάχη των Φύλων, Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών

## **Abstract**

This master thesis proposes two hybrid algorithms to solve vehicle routing problems with profits. In particular, the Distance Related model is proposed, which involves converting solution coding, from discrete node representation into continuous values, based on the Euclidean distance between successive nodes, and vice versa. This model is integrated into the artificial bee colony algorithm (ABC) and the differential evolution algorithm (DE), such the DRABC and DRDE algorithms are presented. The problems solved by the proposed algorithms are: the team orienteering problem (TOP), the capacitated team orienteering problem (CTOP) and the prize-collecting vehicle routing problem (PCVRP). These problems are used to model the tourist trip design problem (TTDP). It also includes a pioneering approach of the TTDP problem, considering tourist trips design for people in a group with different preferences. To solve the later problem's extension, the N-person Battle of the Sexes game is used, which provides an overall scoring of the nodes of a tour based on the preferences of all group's members.

## **Keywords**

Artificial Bee Colony, Differential Evolution, N-person Battle Of Sexes, Tourist Trip Design Problem

# Περιεχόμενα

## Ευχαριστίες

<b>1</b>	<b>Εισαγωγή</b>	<b>1</b>
1.1	Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών . . . . .	1
1.2	Προβλήματα Δρομολόγησης Οχημάτων με Κέρδος . . . . .	3
1.3	Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών Πολλών Ατόμων . . . . .	4
1.4	Σύνοψη Προτεινόμενου Μοντέλου . . . . .	5
<b>2</b>	<b>Προβλήματα Δρομολόγησης Οχημάτων με Κέρδος</b>	<b>6</b>
2.1	Το πρόβλημα προσανατολισμού ομάδας (Team orienteering problem) . . . . .	6
2.1.1	Βιβλιογραφική ανασκόπηση στους τρόπους επίλυσης του TOP . . . . .	6
2.1.2	Μαθηματική διατύπωση του TOP . . . . .	10
2.2	Το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (Capacitated team orienteering problem) . . . . .	11
2.2.1	Βιβλιογραφική ανασκόπηση στους τρόπους επίλυσης του CTOP . . . . .	12
2.2.2	Μαθηματική διατύπωση του CTOP . . . . .	13
2.3	Πρόβλημα Δρομολόγησης Οχημάτων με Συλλογή Βραβείου (Prize-collecting vehicle routing problem) . . . . .	15
2.3.1	Βιβλιογραφική ανασκόπηση στους τρόπους επίλυσης του PCVRP . . . . .	16
2.3.2	Μαθηματική διατύπωση του PCVRP . . . . .	17

<b>3</b>	<b>Αλγόριθμοι Εμπνευσμένοι από τη Φύση</b>	<b>20</b>
3.1	Αλγόριθμος τεχνητής αποικίας μελισσών (Artificial bee colony algorithm) . . . . .	20
3.1.1	Η δομή του ABC αλγόριθμου . . . . .	20
3.1.2	Βιβλιογραφική ανασκόπηση πάνω στην επίλυση VRP με χρήση του αλγορίθμου ABC . . . . .	22
3.2	Αλγόριθμος της διαφορικής εξέλιξης (Differential evolution algorithm) . . . . .	24
3.2.1	Η δομή του DE αλγόριθμου . . . . .	24
3.2.2	Βιβλιογραφική ανασκόπηση πάνω στην επίλυση VRP με χρήση του αλγορίθμου DE . . . . .	26
<b>4</b>	<b>Κωδικοποίηση Distance Related σε Μεθευρετικούς Αλγορίθμους</b>	<b>28</b>
4.1	Το μοντέλο του Distance Related . . . . .	28
4.1.1	Αλγόριθμος Distance Related Artificial Bee Colony . . . . .	30
4.1.2	Αλγόριθμος Distance Related Differential Evolution . . . . .	40
<b>5</b>	<b>Η N-ατόμων Μάχη των Φύλων στην Επίλυση του Προβλήματος Σχεδιασμού Τουριστικών Διαδρομών</b>	<b>45</b>
5.1	Εισαγωγή στη Θεωρία Παιγνίων . . . . .	45
5.2	Η Μάχη των Φύλων (Battle Of Sexes) . . . . .	46
5.3	Η N-ατόμων Μάχη των Φύλων (N-person Battle Of Sexes) . . . . .	47
5.4	Εφαρμογή του N-person BOS στο Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών Πολλών Ατόμων . . . . .	49
5.4.1	Παράδειγμα εφαρμογής του N-person BOS στο Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών Πολλών Ατόμων . . . . .	51
<b>6</b>	<b>Πειραματικά Αποτελέσματα</b>	<b>53</b>
6.1	Παραδείγματα Αναφοράς . . . . .	53
6.2	Πειραματικά Αποτελέσματα . . . . .	55
6.2.1	Πειραματικά Αποτελέσματα στο Πρόβλημα Προσανατολισμού Ομάδας . . . . .	56



6.2.2	Πειραματικά Αποτελέσματα στο Πρόβλημα Προσανατολισμού Ομάδας με Χωρητικότητα . . . . .	64
6.2.3	Πρόβλημα Δρομολόγησης Οχημάτων με Συλλογή Κέρδους	68
<b>7</b>	<b>Συμπεράσματα</b>	<b>72</b>
	<b>Βιβλιογραφία</b>	<b>75</b>

## **Κατάλογος Σχημάτων**

1.1	Διάγραμμα ροής επίλυσης NTTDP . . . . .	5
4.1	Διάγραμμα ροής του αλγόριθμου DRABC . . . . .	31
4.2	Διάγραμμα ροής του αλγόριθμου DRDE . . . . .	41

## Κατάλογος Πινάκων

5.1 Η Μάχη των Φύλων (Πίνακας Κέρδους) . . . . .	46
5.2 Προτίμηση $N = 4, P = 10$ . . . . .	52
5.3 Απόφαση $N = 4, P = 10$ . . . . .	52
5.4 <i>Payoff</i> , $N = 4, P = 10$ . . . . .	52
5.5 <i>Payoff</i> για κάθε <i>POI</i> . . . . .	52
6.1 Παραδείγματα αναφοράς για την επίλυση του TOP . . . . .	54
6.2 Παραδείγματα αναφοράς για την επίλυση του CTOP . . . . .	55
6.3 Πειραματικά Αποτελέσματα στο $Set_{\{1\}}$ . . . . .	57
6.4 Πειραματικά Αποτελέσματα στο $Set_{\{2\}}$ . . . . .	58
6.5 Πειραματικά Αποτελέσματα στο $Set_{\{3\}}$ . . . . .	59
6.6 Πειραματικά Αποτελέσματα στο $Set_{\{4\}}$ . . . . .	60
6.7 Πειραματικά Αποτελέσματα στο $Set_{\{5\}}$ . . . . .	61
6.8 Πειραματικά Αποτελέσματα στο $Set_{\{6\}}$ . . . . .	62
6.9 Πειραματικά Αποτελέσματα στο $Set_{\{7\}}$ . . . . .	63
6.10 Πειραματικά Αποτελέσματα για $Q_{max}, T_{max} = 50$ . . . . .	65
6.11 Πειραματικά Αποτελέσματα για $Q_{max}, T_{max} = 75$ . . . . .	66
6.12 Πειραματικά Αποτελέσματα για $Q_{max}, T_{max} = 100$ . . . . .	67
6.13 Πειραματικά Αποτελέσματα για $Q_{max}, T_{max} = 50$ . . . . .	69
6.14 Πειραματικά Αποτελέσματα για $Q_{max}, T_{max} = 75$ . . . . .	70
6.15 Πειραματικά Αποτελέσματα για $Q_{max}, T_{max} = 100$ . . . . .	71

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους όσους με στήριξαν κατά την περίοδο των σπουδών μου, σε προπτυχιακό και μεταπτυχιακό επίπεδο, την οικογένεια, τους φίλους και τους καθηγητές μου.

Ευχαριστώ ιδιαίτερα τον καθηγητή μου Ιωάννη Μαρινάκη ο οποίος υπήρξε για μένα πάνω από όλα μέντορας κατά την διάρκεια των ακαδημαϊκών μου σπουδών και στάθηκε δίπλα μου με υπομονή και κατανόηση. Τον ευχαριστώ επίσης, για τις συμβουλές του και για τον δρόμο που μου άνοιξε σε ερευνητικό επίπεδο. Θέλω επίσης να ευχαριστήσω θερμά την Δρ. Μαγδαληνή Μαρινάκη πού ήταν πάντα δίπλα μου, πρόθυμη να με συμβουλέψει και να με διορθώσει. Ευχαριστώ επίσης, τον σύντροφο μου Μανούσο, που απλόχερα μου προσέφερε, τόσο πολύτιμη ακαδημαϊκή συνεργασία, όσο και προσωπική στήριξη, μέσα από τα οποία, βελτιώθηκα και έφτασα ως εδώ. Ακόμα, ευχαριστώ όλους μου τους φίλους, οι οποίοι, ο καθένας με τον δικό του ξεχωριστό τρόπο, αποτελούν αναπόσπαστο κομμάτι της ζωής μου. Τέλος, θέλω να ευχαριστήσω την οικογένεια μου, τους γονείς μου Μανώλη και Μαρία και τον αδερφό μου, Γιώργο, που όλα αυτά τα χρόνια με φροντίζουν, με νοιάζονται, με αγαπάνε και που μου προσφέρουν την αμέριστη συμπαράσταση τους.

# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών

Η παρούσα μεταπτυχιακή εργασία, προσεγγίζει με αλγοριθμικό τρόπο την ανάπτυξη ενός ολοκληρωμένου συστήματος υποστήριξης αποφάσεων, το οποίο απευθύνεται στον σχεδιασμό τουριστικών διαδρομών. Για να γίνει κατανοητός ο τρόπος προσέγγισης και συνδυασμού των διαφορετικών προβλημάτων και αλγορίθμων, πρέπει κανείς να αντιληφθεί το βασικό πρόβλημα των τουριστικών διαδρομών. Πολλοί τουρίστες επισκέπτονται μια περιοχή ή μια πόλη, για μία ή περισσότερες ημέρες, όμως δεν είναι πάντα δυνατή η επίσκεψη σε κάθε τουριστικό αξιοθέατο ή σε περιοχή πολιτιστικής κληρονομιάς κατά τη διάρκεια μιας τόσο περιορισμένης περιόδου. Είναι λοιπόν, αναγκαία η επιλογή κάποιων σημείων ενδιαφέροντος (Point of Interest (POIs)), και η οποία συνήθως βασίζεται σε πληροφορίες που βρίσκονται σε ιστότοπους, άρθρα, περιοδικά ή οδηγούς από εξειδικευμένα βιβλιοπωλεία ή βιβλιοθήκες. Αφού ο τουριστικός επισκέπτης ξεχωρίσει τα σημεία που θέλει να επισκεφθεί, πρέπει να αποφασίσει και την σειρά με την οποία θα περιηγηθεί σε αυτά, έχοντας κατά νου τις ώρες λειτουργίας, τις αποστάσεις μεταξύ τους και τον διαθέσιμο χρόνο του. Με άλλα λόγια, προσπαθεί να κατασκευάσει μια εφικτή διαδρομή, αλλά η διαδικασία αυτή δεν είναι απλή. Ένα άτομο δυσκολεύεται στον χειρισμό του μεγάλου όγκου πληροφορίας που σχετίζεται με ένα μεγάλο αριθμό πιθανών σημείων ενδιαφέροντος. Συνήθως οι επισκέπτες θα είναι ευχαριστημένοι εάν σχεδιάσουν ένα κάπως ελκυστικό και εφικτό χρονοδιάγραμμα, αλλά αγνοούν μία καλύτερη προσέγγιση του. Μερικοί ταξιδιωτικοί οδηγοί αναγνωρίζουν αυτό το πρόβλημα και προτείνουν γενικές περιηγήσεις μιας πόλης ή μιας περιοχής. Φυσικά, αυτές οι περιηγήσεις είναι κατασκευασμένες έτσι ώστε να ικανοποιούν τα συμφέροντα της πλειοψηφίας και όχι τις προτιμήσεις του συγκεκριμένου ατόμου.

Σύμφωνα με την βιβλιογραφία, υπάρχουν μοντέλα που προσφέρουν υποστήριξη σε τουριστικούς επισκέπτες, όπως, οι κινητοί τουριστικοί οδηγοί (MTG). Με βάση ένα προφίλ ενδιαφέροντος και ενημερωμένες πληροφορίες ταξιδιού, ένα MTG μπορεί να προτείνει μια (σχεδόν) βέλτιστη και εφικτή επιλογή αξιοθέατων και μια διαδρομή μεταξύ τους [1]. Οι γενικές περιηγήσεις επισκεπτών δεν λαμβάνουν υπόψη το περιβάλλον του χρήστη, όπως η θέση εκκίνησης και τερματισμού, ο διαθέσιμος χρόνος, η τρέχουσα ώρα, ο καιρός κλπ. Η προσθήκη της κατανόησης του περιβάλλοντος και της τοποθεσίας αποτελούν πρακτικές προκλήσεις για την ανάπτυξη έξυπνων εφαρμογών και έτσι υποστηρίζεται η ιδέα της δημιουργίας εξατομικευμένων περιηγήσεων αντί των γενικών. Ορισμένες εφαρμογές ιστού και κινητής τηλεφωνίας έχουν ενσωματώσει πρόσφατα τις συστάσεις τουριστικών διαδρομών στο πλαίσιο της βασικής τους λειτουργικότητας [2, 3, 4]. Στην πραγματικότητα, δημιουργούν εξατομικευμένες διαδρομές λαμβάνοντας υπόψη αρκετές παραμέτρους καθορισμένες από το χρήστη (ημέρες επίσκεψης, προτιμήσεις σε κατηγορίες POI, θέση έναρξης / συχνότητα και ταχύτητα επίσκεψης). Ακόμα, οι συνιστώμενες περιηγήσεις απεικονίζονται στους χάρτες Planner Trip City [5, 2, 3] επιτρέποντας στους χρήστες να περιηγούνται σε επιλεγμένα POI. Από τη βιβλιογραφία προκύπτει το συμπέρασμα ότι ο εξατομικευμένος ηλεκτρονικός τουριστικός οδηγός μελετάται εκτενώς και διατυπώνεται με διαφορετικούς τρόπους, ενώ το πρόβλημα που σχετίζεται με την δημιουργία τουριστικών διαδρομών αναφέρεται ως Tourist Trip Design Problem (TTDP). Στόχος του προβλήματος TTDP είναι η δημιουργία εφικτών διαδρομών μεταξύ των σημείων ενδιαφέροντος και η μεγιστοποίηση της ικανοποίησης του χρήστη. Η ικανοποίηση του χρήστη, είναι υποκειμενική και κυρίως βασίζεται στο κέρδος που θα αποκομίσει από την επίσκεψη του στα σημεία των ορισμένων διαδρομών. Τα βασικά στοιχεία που απαιτούνται για την δημιουργία εξατομικευμένων διαδρομών και την επίλυση του TTDP είναι:

1. Ένα σύνολο υποψηφίων σημείων ενδιαφέροντος (POI), που το καθένα που συνδέεται με συγκεκριμένα χαρακτηριστικά (π.χ. τύπος, τοποθεσία).
2. Ο χρόνος διαδρομής μεταξύ των POI.
3. Το πόσο σημαντικό είναι το κάθε POI, σε σχέση με τις προτιμήσεις του χρήστη.
4. Ο αριθμός των διαδρομών που πρέπει να δημιουργηθούν, ανάλογα με την περίοδο παραμονής του χρήστη στον τουριστικό προορισμό.
5. Η προβλεπόμενη διάρκεια επίσκεψης ενός χρήστη σε ένα POI που προκύπτει από τη μέση διάρκεια και το πιθανό ενδιαφέρον του χρήστη για το συγκεκριμένο POI.

6. Η συνολική διάρκεια ημερήσιας διαδρομής (δηλαδή ο συνολικός χρόνος που αφιερώνεται στη μετακίνηση από ένα ΡΟΙ σε ένα άλλο) που επιθυμεί ο χρήστης να δαπανήσει.

## 1.2 Προβλήματα Δρομολόγησης Οχημάτων με Κέρδος

Το πρόβλημα TTDP, σύμφωνα με την βιβλιογραφία, επιλύεται αλγοριθμικά ως ένα πρόβλημα δρομολόγησης οχημάτων με κέρδος (Vehicle Routing Problems with Profits (VRPPs)). Το βασικό χαρακτηριστικό της κατηγορίας των προβλημάτων δρομολόγησης οχημάτων με κέρδος, σε αντίθεση με το κλασσικό πρόβλημα δρομολόγησης οχημάτων (VRP) [6], είναι ότι το σύνολο των πελατών που πρέπει να εξυπηρετηθούν, δεν είναι ορισμένο. Συνεπώς, πρέπει να ληφθούν δύο διαφορετικές αποφάσεις: (i) ποιοι πελάτες θα εξυπηρετήσουν και (ii) πως θα ομαδοποιηθούν και δρομολογηθούν οι πελάτες σε διαφορετικές διαδρομές. Το κέρδος που αναφέρεται στα προβλήματα αυτά, έχει να κάνει με το πόσο πολύτιμη ή σημαντική είναι η εξυπηρέτηση κάποιου πελάτη. Η αντικειμενική συνάρτηση των VRPPs ποικίλει, μπορεί να αναφέρεται είτε στη μεγιστοποίηση του κέρδους από την εξυπηρέτηση των πελατών, είτε στη διαφορά μεταξύ κέρδους και κόστους συνολικής διανυσθείσας απόστασης. Οι ομοιότητες με το κλασσικό πρόβλημα συνοψίζονται στα ότι οι διαδρομές ξεκινάνε και τερματίζουν σε συγκεκριμένους κόμβους και ότι η εφικτότητα κάθε διαδρομής ελέγχεται από συγκεκριμένους περιορισμούς. Τα επικρατέστερα προβλήματα σε αυτήν την κατηγορία είναι: το πρόβλημα προσανατολισμού ομάδας (Team Orienteering Problem), το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (Capacitated Team Orienteering Problem) και το πρόβλημα δρομολόγησης οχημάτων με συλλογή βραβείου (Prize-collecting vehicle routing problem). Τα τρία αυτά βασικά προβλήματα περιγράφονται στο επόμενο κεφάλαιο (βλ. Κεφάλαιο 2). Συσχετίζοντας, ενδεικτικά, τα προβλήματα δρομολόγησης οχημάτων με κέρδος με το TTDP, οι πελάτες των προβλημάτων αναπαριστούν τους κόμβους, δηλαδή τα σημεία ενδιαφέροντος (POIs), ο χρόνος διαδρομής μεταξύ των κόμβων αναφέρεται στην ευκλείδεια απόσταση τους, το κέρδος κάθε κόμβου συμβαδίζει με την προτίμηση του χρήστη ως προς το σημείο και κάθε διαδρομή αντιστοιχεί σε ημερήσια επίσκεψη. Ακόμα, στα προβλήματα που αναφέρεται η χωρητικότητα ως περιορισμός της συνολικής ζήτησης, συσχετίζεται η ζήτηση κάθε κόμβου με τον χρόνο επίσκεψης σε αυτόν.

Για την επίλυση των τριών προβλημάτων δρομολόγησης οχημάτων με κέρδος, που αναφέρθηκαν, κατασκευάστηκαν δύο υβριδικοί αλγόριθμοι, βασισμένοι στον αλγόριθμο τεχνητής αποικίας μελισσών (Artificial bee colony algorithm

(ABC)) και στον αλγόριθμο διαφορικής εξέλιξης (Differential evolution algorithm (DE)). Στο Κεφάλαιο 3, αναλύεται το κλασσικό μοντέλο των δύο αυτών αλγορίθμων και παρατίθεται μια σύντομη βιβλιογραφική έρευνα πάνω στην επίλυση προβλημάτων δρομολόγησης από αυτούς. Ένα από τα καινοτόμα σημεία αυτής της εργασίας αναλύεται στο Κεφάλαιο 4, πρόκειται για το βασικό χαρακτηριστικό των υβριδικών αλγορίθμων που προτείνονται, του Distance Related Artificial Bee Colony και του Distance Related Differential Evolution. Το μοντέλο που προτείνεται αναφέρεται ως DR και αποτελεί έναν νέο τρόπο κωδικοποίησης λύσεων των προβλημάτων συνδυαστικής βελτιστοποίησης (όπως των VRPPs), ώστε να επιτρέπεται η επίλυση τους από αλγορίθμους σχεδιασμένους για βελτιστοποίηση προβλημάτων συνεχών τιμών.

### **1.3 Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών Πολλών Ατόμων**

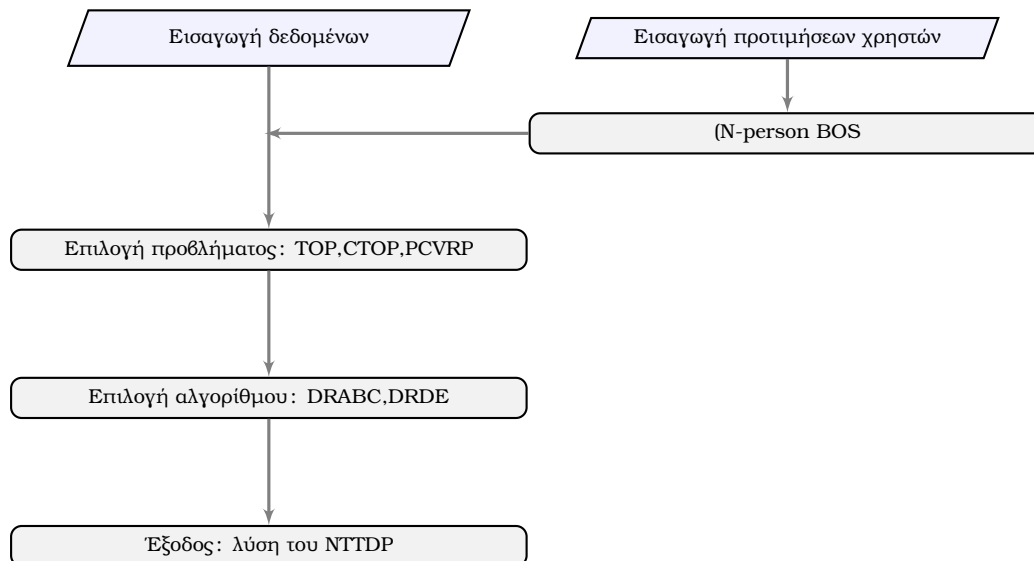
Όπως έχει αναφερθεί, έχει εξεταστεί εκτενώς η δημιουργία εξατομικευμένων ηλεκτρονικών ταξιδιωτικών οδηγών, για την σύσταση διαδρομής σε σημεία ενδιαφέροντος βάση των προτιμήσεων ενός χρήστη (επιλύοντας το πρόβλημα TTDP). Η παρούσα μεταπτυχιακή εργασία, εξετάζει την επέκταση των συστημάτων αυτών σε παραπάνω από ένα άτομα, με διαφορετικές προτιμήσεις. Ας υποθέσουμε μια ομάδα φίλων, οι οποίοι σκοπεύουν να προγραμματίσουν ένα κοινό ταξίδι σε μια συγκεκριμένη περιοχή και πρέπει να επιλέξουν τα σημεία ενδιαφέροντος που θα επισκεφθούν, ενώ έχουν διαφορετικές προτιμήσεις, με σκοπό να μεγιστοποιήσουν την ικανοποίησή τους. Σε περίπτωση που τουλάχιστον ένα από τα μέλη της ομάδας έχει αντικρουόμενες προτιμήσεις από τους υπόλοιπους, τότε, το κλασσικό TTDP δεν μπορεί να εφαρμοστεί, αφού απαιτεί κοινές προτιμήσεις από όλα τα μέλη της ομάδας. Για να αντιμετωπιστεί αυτή η κατάσταση, παρουσιάζεται ένας συνδυασμός του προβλήματος TTDP με ένα παίγνιο από τη Θεωρία Παιγνίων. Το αναφερόμενο παίγνιο είναι η μάχη των φύλων Battle Of Sexes (BOS) και παρουσιάζεται στο Κεφάλαιο 5. Επειδή, όμως, η κλασσική έκδοση του παιγνίου περιλαμβάνει μόνο δυο άτομα, στο μοντέλο που παρουσιάζεται χρησιμοποιούμε την επέκταση του παιγνίου για περισσότερα από δύο άτομα, το N-person Battle Of Sexes (N-person BOS). Η συνεισφορά του παιγνίου στο πρόβλημα σχεδιασμού τουριστικών διαδρομών είναι υψίστης σημασίας, αφού μέσα από αυτό γίνεται μία βαθμολόγηση όλων των κόμβων (προσδιορισμός κέρδους ανά κόμβο), συνδυάζοντας τις προτιμήσεις όλων των χρηστών μίας ομάδας, με γρήγορο και αποδοτικό τρόπο.



### 1.4 Σύνοψη Προτεινόμενου Μοντέλου

Το συνολικό αλγοριθμικό πλαίσιο που παρουσιάζεται, στοχεύει στην επίλυση του προβλήματος σχεδιασμού τουριστικών διαδρομών πολλών ατόμων, N-Tourist Trip Design Problem (NTTDP). Η αλγοριθμική διαδικασία βασίζεται στο προτεινόμενο μοντέλο DR, όπως αυτό εφαρμόζεται σε αλγορίθμους εμπνευσμένους από τη φύση (DRABC,DRDE) και η μοντελοποίηση του προβλήματος σχεδιασμού τουριστικών διαδρομών αντιστοιχείται σε προβλήματα δρομολόγησης οχημάτων με κέρδος (TOP,CTOP,PCVRP). Τα δεδομένα κέρδους που απαιτούνται, διαμορφώνονται από το εκτεταμένο παίγνιο (N-person BOS σύμφωνα με τις διαφορετικές προτιμήσεις μίας ομάδας χρηστών. Τα πειραματικά αποτελέσματα του μοντέλου αναφέρονται στο Κεφάλαιο 6 και η διαδικασία συνοψίζεται στο Σχήμα 1.1.

Σχήμα 1.1: Διάγραμμα ροής επίλυσης NTTDP



## Κεφάλαιο 2

# Προβλήματα Δρομολόγησης Οχημάτων με Κέρδος

### 2.1 Το πρόβλημα προσανατολισμού ομάδας (Team orienteering problem)

Το πρόβλημα προσανατολισμού ομάδας (Team Orienteering Problem (TOP)) είναι μια επέκταση του προβλήματος του προσανατολισμού (OP) με περισσότερες από μία διαδρομές και παρουσιάστηκε για πρώτη φορά από τους Butt και Cavalier το 1994 [7], ως το MTMCP (Multiple Tour Maximum Collection Problem). Το 1996, οι Chao et al. [8] καθιέρωσαν το πρόβλημα ως μέρος της κατηγορίας των προβλημάτων δρομολόγησης οχημάτων με κέρδη (VRPPs). Το πρόβλημα συνίσταται από ένα σύνολο κόμβων με συγκεκριμένο χρόνο μετακίνησης μεταξύ τους και σε μια βαθμολογία που συνδέεται με κάθε έναν. Ο στόχος είναι να επιλεγεί ένα σετ κόμβων και να διαταχθούν σε έναν συγκεκριμένο αριθμό υπο-διαδρομών. Οι υπο-διαδρομές θα πρέπει να ικανοποιούν τον περιορισμό της συνολικής απόστασης ταξιδιού, μεγιστοποιώντας τη συνολική βαθμολογία που συγκεντρώνει κάθε υπο-διαδρομή, το άθροισμα της βαθμολογίας των κόμβων που εντάσσονται σε αυτή. Το πρόβλημα του προσανατολισμού έχει αποδειχθεί ως ένα πρόβλημα  $NP - hard$  από τους Golden et al. [9] το 1987 και έτσι το TOP πέφτει, επίσης, στην ίδια κατηγορία.

#### 2.1.1 Βιβλιογραφική ανασκόπηση στους τρόπους επίλυσης του TOP

Η πιο πρόσφατη βιβλιογραφική ανασκόπηση πάνω στα προβλήματα προσανατολισμού (Orienteering Problem) και προσανατολισμού ομάδας (Team Orienteering Problem), όπως και των διαφορετικών τους προσεγγίσεων, έγινε από

τους Gunawan et al. [10], το 2016. Στο έργο τους συμπεριλαμβάνουν και επεκτείνουν την αντίστοιχη ανασκόπηση των: Vansteenwegen et al. [11], Gavalas et al. [12] και Archetti et al.[6]. Συνοψίζοντας και επεκτείνοντας τις παραπάνω ανασκοπήσεις, ακολουθεί μια σύντομη βιβλιογραφική αναφορά των δημοσιευμένων εργασιών πάνω στην επίλυση του προσανατολισμού ομάδας. Αρχικά, οι Butt και Cavalier [7] το 1994, παρουσίασαν έναν άπληστο ευρετικό αλγόριθμο, οποίος αναθέτει του κόμβους σε μία διαδρομή λαμβάνοντας υπόψιν κάποια βάρη ανά ζευγάρι κόμβων. Στη συνέχεια, οι Chao et al. [13] πρότειναν ένα ευρετικό αλγόριθμο βάση του οποίου οι υπο-διαδρομές κάθε λύσης δημιουργούνται ακολουθώντας τον κανόνα της οικονομικότερης εισαγωγής και βελτιώνονται με την χρήση επιπλέον στρατηγικών τοπικής αναζήτησης (1-point movement, 2-point exchange, 2-opt). Ο πρώτος ακριβής αλγόριθμος για την επίλυση του ομαδικού προσανατολισμού οχημάτων προτάθηκε από τους Butt και Ryan [14] το 1999. Συγκεκριμένα, πρότειναν έναν μοντέλο μεικτού ακέραιου προγραμματισμού σε συνδυασμό με έναν απλό κατασκευαστικό αλγόριθμο. Χρησιμοποίησαν την αναλογία κέρδους προς απόστασης, η οποία βελτιωνόταν από μια μέθοδο δημιουργίας στηλών. Από τότε και έπειτα, σύμφωνα με όσα γνωρίζουμε, δεν έχει δημοσιευθεί αντίστοιχη εργασία μέχρι το 2005, όταν οι Tang και Miller-Hooks [42] πρότειναν ένα μοντέλο μεικτού ακέραιου προγραμματισμού. Συνδύασαν το μοντέλο τους με έναν αλγόριθμο αναζήτησης Tabu με προσαρμοστική μνήμη, ο οποίος επέτρεπε την εναλλαγή της γειτονίας αναζήτησης από μικρή σε μεγάλη. Μετά από δύο χρόνια, οι Boussier et al. [16] πρότειναν έναν αλγόριθμο διακλάδωσης και τιμολόγησης, που συμπεριλαμβάνει δυναμικό προγραμματισμό και διαδικασίες επιτάχυνσής πάνω στη δημιουργία στηλών. Ο αλγόριθμός τους αποδείχθηκε πιο αποτελεσματικός σε σύγκριση με τον αρχικό που πρότειναν οι Butt και Ryan [14]. Παράλληλα, οι Archetti et al. [36] ανέπτυξαν δύο παραλλαγές του γενικευμένου αλγόριθμου αναζήτησης Tabu, η δεύτερη εκ των οποίων επιτρέπει την διαμόρφωση μη εφικτών διαδρομών με τιμωρία. Οι αλγόριθμοι αυτοί συμβολίζονται ως TSF και TSU αντίστοιχα. Στη συνέχεια οι Ke et al. [18] χρησιμοποίησαν μια προσέγγιση του αλγορίθμου αποικίας μυρμηγκιών (Ant Colony Optimization, ACO), σύμφωνα με τον οποίο κάθε μυρμηγκί κατασκευάζει μια υποψήφια λύση, βάση τεσσάρων διαφορετικών μεθόδων. Συγκεκριμένα, πρότειναν μια ακολουθητική μέθοδο (ASe) που βασίζεται στην διαμόρφωση ολόκληρων υπο-διαδρομών, η οποία είναι και η πιο αποτελεσματική. Ακόμα πρότειναν την τυχαία-ταυτόχρονη μέθοδο (ARC), κατά την οποία οι κόμβοι εισάγονται σε τυχαίες υπο-διαδρομές στον ίδιο χρόνο. Η επόμενη διαδικασία είναι η ντετερμινιστική-ταυτόχρονη μέθοδος (ADC), στην οποία εφαρμόζεται μία σταθερή ακολουθία υπο-διαδρομών. Η τέταρτη διαδικασία είναι η ταυτόχρονη μέθοδος (ASi) που γεμίζει όλες τις διαδρομές μέχρι το όριο τους εισάγοντας κόμβους σε κάθε επανάληψη.

Το 2009, οι Vansteenwegen et al. [19] εφάρμοσαν μία δομή καθοδηγούμενης τοπικής αναζήτησης (GLS) και αργότερα μια δομή έμπειρης μεταβλητής γειτονιάς αναζήτησης (SVNS)[20]. Αρχικά, οι δύο αυτοί αλγόριθμοι εφαρμόζουν μια άπληστη κατασκευαστική φάση, στη συνέχεια διαδικασίες τοπικής αναζήτησης και τέλος διαδικασίες εντατικοποίησης και διαφοροποίησης που αποσκοπούν στην αύξηση του κέρδους και στη μείωση του χρόνου ταξιδιού μιας διαδρομής. Ο αλγόριθμος SVNS είναι πιο γρήγορος και ποιο αποδοτικός συγκριτικά με τον GLS. Συνεχίζοντας την ανασκόπηση, οι Souffriaux et al. [21] παρουσίασαν δύο παραλλαγές μιας διαδικασίας άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (GRASP) μαζί με την διαδικασία αποκατάστασης διαδρομής (PR). Η αργή έκδοση (SPR) είχε εξαιρετικά αποτελέσματα και η γρήγορη έκδοση (FPR) είχε ικανοποιητικά. Σε αυτόν τον αλγόριθμο GRASP, εκτελούνται τέσσερις διαδικασίες διαδοχικά για ένα σταθερό αριθμό επαναλήψεων, συμπεριλαμβανομένης μιας διαδικασίας κατασκευής, ενός αριθμού τοπικών διαδικασιών αναζήτησης και μιας διαδικασίας επανασύνδεσης διαδρομής που παρέχει ένα σύνολο μη εφικτών λύσεων με την προσθήκη μακράς μνήμης. Η διαδικασία PR δημιουργεί μη εφικτές λύσεις και τις μετατρέπει μέσω τοπικής αναζήτησης σε εφικτές. Ακολουθεί ακόμα ένας ακριβής αλγόριθμος για την επίλυση του TOP, ο οποίος προτάθηκε από τους Roggi et al. [22] το 2010. Πρόκειται για έναν αλγόριθμο διακλάδωσης και διχοτόμησης και τιμολόγησης, κατά τον οποίο εφαρμόζονται δυο τεχνικές διακλάδωσης, η Min Cut inequalities και η Triangle Clique cuts. Ακόμα μέσα στο 2010, οι Bouly et al. [23] πρότειναν μία προσέγγιση μιμητικού αλγορίθμου. Ανέπτυξαν έναν υβριδικό γενετικό αλγόριθμο που χρησιμοποιεί μια διαδικασία που ονομάζεται Optimal Split για να αξιολογήσει τα χρωμοσώματα και τοπικές τεχνικές αναζήτησης για τη διαδικασία της μετάλλαξης, ενώ ο αλγόριθμος δίνει νέες βελτιωμένες λύσεις. Επιπλέον, το επόμενο έτος παρουσιάστηκε από τους Muthuswamy και Lam [24] ένας διακριτός αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (DSPSO), ο οποίος περιλαμβάνει τη μειωμένη μεταβλητή γειτονιά αναζήτησης (RVNS) και τη μέθοδο 2-Opt στη διαδικασία ενημέρωσης σωματιδίων. Οι Dang et al. [25] πρότειναν, το 2011, έναν αποτελεσματικό αλγόριθμο βελτιστοποίησης σωματιδίων βασισμένο σε μιμητική μέθοδο (PSOMA). Το PSOMA περιλαμβάνει βέλτιστες τεχνικές διαίρεσης διαδρομών, γενετικές μεθόδους crossover και, επίσης, την εντατική χρήση τοπικών τεχνικών αναζήτησης. Τα πειράματα δείχνουν ότι το PSOMA πέτυχε την καλύτερη γνωστή απόκλιση από το βέλτιστο της τάξης του 0,016%.

Επιπλέον, το 2013, οι Dang et al. [26] παρουσίασε έναν αλγόριθμο διακλάδωσης και διχοτόμησης για την επίλυση του TOP και πρότεινε αρκετές τεχνικές διχοτόμησης που ενισχύουν την κλασική γραμμική διατύπωση της συμμετρι-

κής κοπής, γενικευμένους περιορισμούς, όρια στα κέρδος / αριθμός πελατών, περικοπές και άλλα. Ο αλγόριθμος είναι αποτελεσματικός και κλείνει 29 ανοικτά παραδείγματα. Μέχρι στιγμής, τα διαθέσιμα παραδείγματα συγκριτικής αξιολόγησης που χρησιμοποιήθηκαν στη βιβλιογραφία προτάθηκαν από τους Chao et al.[8]. Το 2013, οι Dang et al. [27] εισήγαγαν ένα νέο σύνολο μεγαλύτερων παραδειγμάτων για το TOP με εκ νέου υπολογισμό του διαθέσιμου χρόνου ανά όχημα. Επίσης, οι Dang et al. [27] με βάση την προηγούμενη τους έρευνα, ανέπτυξαν έναν αλγόριθμο εμπνευσμένο από τον PSO(PSOia) βασισμένο σε ένα μοντέλο γραφικών διαστημάτων, με αποτέλεσμα να ανιχνεύσουν όλες τις γνωστές βέλτιστες λύσεις και επιπλέον να βελτιώσουν μία από αυτές. Επίσης, το 2013, ο Lin [28] περιγράφει έναν αλγόριθμο προσομοιωμένης ανόπτησης πολλαπλών εκκινήσεων MSA, συνδυάζοντας μια μεθευρετική διαδικασία, βασισμένη σε προσομοιωμένη ανόπτηση και σε μια στρατηγική αναρρίχησης λόφων πολλαπλών εκκινήσεων. Ο αλγόριθμος MSA πέτυχε τις καλύτερες λύσεις στο 0,85 % των προβλημάτων αναφοράς. Την ίδια στιγμή, οι Kim et al. [29] πρότειναν μια ενισχυμένη μέθοδο αναζήτησης γειτονιάς με τρεις αλγορίθμους βελτίωσης: τοπική αναζήτηση, βελτίωση μετατόπισης και εισαγωγής και βελτίωση αντικατάστασης. Ο αλγόριθμός τους ανιχνεύει τις γνωστές βέλτιστες λύσεις για τα 386 από τα 387 προβλήματα αναφοράς και σε ένα πέτυχε απόκλιση μίας μονάδας. Μια άλλη προσέγγιση γενετικού αλγορίθμου υλοποιήθηκε από τον Ferreira [30], ως μέρος ενός αλγοριθμικού πλαισίου που ονομάζεται Γενετικός Αλγόριθμος Προβλήματος Προσανατολισμού Ομάδας (GATOP), το οποίο εγκρίθηκε από το Πορτογαλικό Ίδρυμα Επιστήμης και Τεχνολογίας. Οι Keshtkaran et al. [31] πρότειναν μια μέθοδο διακλάδωσης και τιμολόγησης (BP) και μια μέθοδο διακλάδωσης και διχοτόμησης και τιμολόγησης (BCP). Ο αλγόριθμος έκλεισε περίπου 5% των προηγουμένως ανοιχτών περιπτώσεων. Το 2015, ένας μιμητικός αλγόριθμος κατά Pareto (PMA), προτάθηκε από τους Ke et al. [32] και παρουσιάστηκαν δύο νέες τεχνικές, η *mimic* που μιμείται μια κατεστημένη λύση και η *swallow* που επιδιορθώνει μη εφικτές διαδρομές.

Ένας νέος ακριβής αλγόριθμος προτάθηκε, το 2016, από τον El-Hajj et al. [33], που δίνει έμφαση στις διαδικασίες κοπής-επιπέδου για τη μείωση του χώρου αναζήτησης. Οι αποδοτικές περικοπές που πρότειναν περιλαμβάνουν χαρακτηριστικά όπως σπάσιμο συμμετρίας, άσχετες περικοπές στοιχείων, όρια για τα κέρδη και τον αριθμό των πελατών, περικοπές υποχρεωτικών πελατών και άλλες. Ο αλγόριθμος του επιπέδου κοπής είναι παραγωγικός, αφού έκλεισε 12 προηγουμένως ανεπίλυτες περιπτώσεις και ξεπέρασε όλες τις προηγούμενες ακριβείς προσεγγίσεις, εκτός από τον βελτιωμένο αλγόριθμο BR-2R του [31] που έχει χαρακτηριστεί ως συμπληρωματικός του. Επίσης, οι Zettam και Elbenani [34] πρότειναν ένα νέο τυχαίοποιημένο ευρετικό μοντέλο ικανό να

δημιουργήσει αρχικό πληθυσμό για μεθευρετικούς αλγορίθμους που βασίζονται σε πληθυσμό, δημιουργώντας τυχαία έναν αριθμό μεταβολών. Τέλος, οι Karouni και Elbenani [35] πρότειναν μια προσέγγιση λύσης βασισμένη στον αλγόριθμο επιλογής κλώνων.

### 2.1.2 Μαθηματική διατύπωση του TOP

Το πρόβλημα Team Orienteering Team (TOP) μπορεί να περιγραφεί χρησιμοποιώντας ένα πλήρες γράφημα  $G = (V, A)$ , όπου  $V = \{1, \dots, N\}$  είναι το σύνολο κόμβων και  $A = \{(i, j) | i, j \in V\}$  είναι το σύνολο των τόξων. Κάθε κόμβος  $i$  που περιλαμβάνεται στο  $V$  σει σχετίζεται με μία βαθμολογία που στο εξής θα αναφέρεται ως σκορ  $s_i$ , επιπλέον ο απαιτούμενος χρόνος ταξιδιού μεταξύ κόμβων  $t_{ij}$  δίνεται για κάθε ζεύγος  $(i, j)$ . Ο κόμβος έναρξης και ο τελικός κόμβος έχουν σκορ ίσο με μηδέν. Για κάθε λύση (διαδρομή) πρέπει να δημιουργηθεί ένας περιορισμένος αριθμός υπο-διαδρομών  $M$ . Κάθε υπο-διαδρομή είναι εφικτή σε σχέση με ένα προκαθορισμένο όριο ταξιδιού  $T_{max}$ . Κάθε υπο-διαδρομή πρέπει να ξεκινά από τον αρχικό κόμβο 1 και να καταλήγει στον τελικό κόμβο  $N$ , κάθε κόμβος είναι επισκέψιμος το πολύ μια φορά και ανήκει σε ακριβώς μία διαδρομή. Ο στόχος είναι να προσδιοριστεί το σύνολο των υπο-διαδρομών  $M$  που μεγιστοποιούν το συνολικό *Score* που λαμβάνεται από τους κόμβους που ανήκουν σε αυτή. Χρησιμοποιώντας το ακόλουθο σύνολο μεταβλητών απόφασης το πρόβλημα μπορεί να διατυπωθεί ως ένα ακέραιο πρόβλημα [18]:

- $y_{id} = 1$  Αν ο κόμβος  $i$  ( $i = 1, \dots, N$ ) περιλαμβάνεται στην υπο-διαδρομή  $d$  ( $d = 1, \dots, M$ ),  $y_{id} = 0$  διαφορετικά.
- $x_{ijd} = 1$  Αν η υπο-διαδρομή  $d$  ( $d = 1, \dots, M$ ) περιλαμβάνει το τόξο  $i, j$  ( $i, j = 1, \dots, N$ ), σε διαφορετική περίπτωση  $x_{ijd} = 0$ . Η συμμετρία του προβλήματος το εξασφαλίζει ότι  $t_{ij} = t_{ji}$ , άρα, ορίζονται μόνο τα  $x_{ijd}$  για  $i < j$ .

Η μαθηματική διατύπωση του TOP είναι:

$$\text{Maximize Score} : \sum_{i=2}^{N-1} \sum_{d=1}^M s_i y_{id} \quad (2.1)$$

$$\sum_{j=2}^N \sum_{d=1}^M x_{1jd} = \sum_{i=1}^{N-1} \sum_{d=1}^M x_{iNd} = M \quad (2.2)$$

$$\sum_{i < j} x_{ijd} + \sum_{i > j} x_{jid} = 2y_{jd}, \forall j = 2, \dots, N-1; \forall d = 1, \dots, M \quad (2.3)$$

$$\sum_{d=1}^m y_{id} \leq 1, \forall i = 2, \dots, N-1 \quad (2.4)$$

$$\sum_{i=1}^{N-1} \sum_{j>i} t_{ij} x_{ijd} \leq T_{max}, \forall d = 1, \dots, M \quad (2.5)$$

$$\sum_{(i,j) \in S, i < j} x_{ijd} \leq |S| - 1, \forall S \subset V \setminus \{1, N\}; 2 \leq |S| \leq N-2; \forall d = 1, \dots, M \quad (2.6)$$

$$x_{ijd} \in \{0, 1\}, 1 \leq i < j \leq N; d = 1, \dots, M \quad (2.7)$$

$$y_{1d} = y_{Nd} = 1, y_{jd} \in \{0, 1\}; \forall i = 2, \dots, N-1; \forall d = 1, \dots, M \quad (2.8)$$

Η αντικειμενική συνάρτηση (2.1) εξασφαλίζει τη μεγιστοποίηση του αθροιστικού σκορ που συλλέγεται από τους κόμβους σε κάθε υπο-διαδρομή. Ο περιορισμός (2.2) διασφαλίζει ότι κάθε υπο-διαδρομή αρχίζει από τον πρώτο κόμβο στο σύνολο  $V$  και τερματίζει στον κόμβο  $N$ . Καθώς δημιουργείται η διαδρομή του οχήματος, η συνδεσιμότητα για κάθε κόμβο που περιλαμβάνεται προστατεύεται από τον περιορισμό (2.3). Ο περιορισμός (2.4) διασφαλίζει ότι κάθε κόμβος είναι επισκέψιμος το πολύ μία φορά και ο περιορισμός (2.5) καθορίζει ο συνολικός χρόνος κάθε υπο-διαδρομής είναι μέσα στο προκαθορισμένο όριο. Ο περιορισμός (2.6) απαγορεύει τις δευτερεύουσες διαδρομές. Τέλος, οι περιορισμοί (2.7), (2.8) καθορίζουν τις μεταβλητές του προβλήματος.

## 2.2 Το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (Capacitated team orienteering problem)

Μια παραλλαγή του προβλήματος προσανατολισμού ομάδας TOP που αναφέρθηκε στο Κεφάλαιο 2.1, είναι το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (Capacitated Team Orienteering Problem, CTOP). Τα δύο προβλήματα έχουν ίδια αντικειμενική συνάρτηση, μεγιστοποίησης κέρδους, όμως το CTOP περιλαμβάνει έναν επιπλέον περιορισμό χωρητικότητας ανά υπο-διαδρομή. Άρα μία επιπλέον ιδιότητα ανά κόμβο, την ζήτηση του. Θεωρητικά

το άθροισμα της ζήτησης των κόμβων ανά υπο-διαδρομή δεν θα πρέπει να ξεπερνάει ένα ανώτατο όριο. Στη συνέχεια ακολουθεί μία βιβλιογραφική ανασκόπηση πάνω στο πρόβλημα προσανατολισμού ομάδας με χωρητικότητα και στους μέχρι πρότεινως διαφορετικούς τρόπους επίλυσης τους.

### 2.2.1 Βιβλιογραφική ανασκόπηση στους τρόπους επίλυσης του CTOP

Το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα προτάθηκε αρχικά το 2009 από τους Archetti et al. [36], όμως παρά την σημασία του προβλήματος σε εφαρμογές της πραγματικής ζωής, λίγοι ερευνητές έχουν ασχοληθεί με την εύρεση τεχνικών για την επίλυση του. Η Archetti και η ομάδα της πρότειναν τρεις ευρετικούς αλγόριθμους: τον αλγόριθμό της μεταβλητής γειτονιάς αναζήτησης (VNS) και δύο παραλλαγές του αλγορίθμου αναζήτησης Tabu, τους TabuFeasible και Tabu Admissible. Οι προτεινόμενες στρατηγικές τοπικής αναζήτησης βασίζονται στην ομαδοποίηση όλων των κόμβων σε δρομολόγια και στη συνέχεια την επιλογή των πιο κερδοφόρων για να σχηματίσουν μια λύση. Επιπλέον, διερευνούν όχι μόνο εφικτές περιοχές του χώρου λύσης (TabuFeasible), αλλά και περιοχές στις οποίες εμφανίζεται μη εφικτότητα στις διαδρομές που περιλαμβάνονται στην αποδεκτή λύση (Tabu Admissible). Για να εξεταστεί η ποιότητα των αλγορίθμων, δημιουργήθηκαν ακόμα και καινούργια παραδείγματα αναφοράς με βάση τις ανάγκες του προβλήματος προσανατολισμού διαδρομής, τα οποία είναι διαθέσιμα στον ιστότοπο: <http://tarantilis.dmst.aueb.gr/docs/>. Τα παραδείγματα αναφοράς επιλύθηκαν κατά το βέλτιστο από ακριβή αλγόριθμο διακλάδωσης και τιμολόγησης και τα αποτελέσματα των αλγορίθμων Tabu και VNS συγκριτικά με αυτές τις τιμές ήταν αρκετά ικανοποιητικά.

Έπειτα το 2013, οι Archetti et al. [37] παρουσίασαν έναν νέο αλγόριθμο διακλάδωσης και τιμολόγησης, χρησιμοποιώντας μία ευρετική μέθοδο με βάση την στήλη, σε κάθε κόμβο του δέντρου διακλάδωσης ώστε να ληφθούν οι αρχικές τιμές ορίων. Ο αλγόριθμος αυτός αποδείχθηκε πιο αποτελεσματικός από τον αντίστοιχο προηγούμενο [36], αφού παράγει βέλτιστες τιμές σε αρκετά προτέρως άλυτα παραδείγματα αναφοράς. Την ίδια χρονιά, οι Tarantilis et al. [38] μελετήσαν το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα προτείνοντας ένα ιεραρχικό πλαίσιο αναζήτησης δύο επιπέδων, μια μέθοδο Filter-and-Fan. Συνολικά, η προτεινόμενη μέθοδος αποτελείται από: έναν ερευνητικό άπληστο παράλληλο αλγόριθμο εισαγωγής για να παράγει μια αρχική εφικτή λύση, μία αναζήτηση με βάση τον αλγόριθμο Tabu, για να εντοπίσει μια τοπική βέλτιστη λύση και μια νέα αναζήτηση Filter-and-Fan για να εξερευνήσει μεγαλύτερες γειτονιές και να δημιουργήσει πολλές τροχιές αναζήτησης, ώστε



να ξεπεραστεί η τοπική βέλτιστη συμπεριφορά. Ο προτεινόμενος αλγόριθμος δοκιμάστηκε στα υπάρχον προβλήματα αναφοράς [36] και σε ένα νέο σύνολο μεγάλης κλίμακας, [Http : //tarantilis.dmst.aueb.gr/docs/](http://tarantilis.dmst.aueb.gr/docs/). Τα πειραματικά αποτελέσματα έδειξαν την αποδοτικότητα και αποτελεσματικότητα της μεθόδου Bi-level Filter-and-Fan, αφού βελτιώνει 18 από τα παραδείγματα αναφοράς. Ακόμα, οι Luo et al. [39] πρότειναν έναν ευρετικό αλγόριθμο για το CTOP που εφαρμόζει δύο διαφορετικούς κανόνες προτεραιότητας για την εισαγωγή ενός κόμβου σε μια διαδρομή. Εφαρμόζοντας μια τοπική διαδικασία αναζήτησης και σύμφωνα με τον κανόνα προτεραιότητας που χρησιμοποιείται, ο αλγόριθμος εισάγει επανειλημμένα τον μη επισκεπτόμενο κόμβο με την υψηλότερη προτεραιότητα στην διαδρομή. Εάν η λύση δεν βελτιωθεί για συγκεκριμένο αριθμό επαναλήψεων, η λύση διαταράσσεται και ο κανόνας προτεραιότητας αλλάζει. Τα πειραματικά αποτελέσματα έδειξαν ότι ο αλγόριθμος είναι πιο αποτελεσματικός τόσο στην ποιότητα των λύσεων όσο και στον χρόνο εκτέλεσης από άλλους δημοσιευμένους πάνω στην επίλυση του CTOP. Η πιο πρόσφατη δημοσίευση πάνω στην επίλυση του προβλήματος προσανατολισμού ομάδας με χωρητικότητα, είναι αυτή των Ben-Said et al. [40], το 2016. Ο αλγόριθμος πρότειναν συμβολίζεται ως AIDCH και αποτελείται από μια προσαρμοστική διαδικασία κατασκευής με βάση τον αλγόριθμο εισαγωγής και από ένα προσαρμοστικό μηχανισμό διαταραχής, που ελέγχει τον αριθμό των πελατών που έχουν αφαιρεθεί ανάλογα με την αποτελεσματικότητα της αναζήτησης. Τα πειραματικά αποτελέσματα που πραγματοποιήθηκαν [36] δείχνουν την αποτελεσματικότητα του αλγόριθμου, με σχετικό σφάλμα 0,008%.

### 2.2.2 Μαθηματική διατύπωση του CTOP

Το πρόβλημα Capacitated Team Orienteering Team (CTOP) μπορεί να περιγραφεί χρησιμοποιώντας ένα πλήρες γράφημα  $G = (V, A)$ , όπου  $V = \{1, \dots, N\}$  είναι το σύνολο κόμβων και  $A = \{(i, j) | i, j \in V\}$  είναι το σύνολο των τόξων. Κάθε κόμβος  $i$  που περιλαμβάνεται στο  $V$  σεται σχετίζεται με μία βαθμολογία που στο εξής θα αναφέρεται ως κέρδος  $p_i$  και με μία τιμή ζήτησης  $D_i$ , επιπλέον ο απαιτούμενος χρόνος ταξιδιού μεταξύ κόμβων  $t_{ij}$  δίνεται για κάθε ζεύγος  $(i, j)$ . Ο κόμβος έναρξης έχει κέρδος ίσο με μηδέν. Για κάθε λύση (διαδρομή) πρέπει να δημιουργηθεί ένας περιορισμένος αριθμός υπο-διαδρομών  $M$ . Κάθε υπο-διαδρομή είναι εφικτή σε σχέση με ένα προκαθορισμένο όριο ταξιδιού  $T_{max}$  και με ένα προκαθορισμένο όριο χωρητικότητας  $Q_{max}$ . Κάθε υπο-διαδρομή πρέπει να ξεκινά από τον αρχικό κόμβο 1 και να καταλήγει σε αυτόν, κάθε κόμβος είναι επισκέψιμος το πολύ μια φορά και ανήκει σε ακριβώς μία διαδρομή. Ο στόχος είναι να προσδιοριστεί το σύνολο των υπο-διαδρομών  $M$  που

μεγιστοποιούν το συνολικό *Profit*, κέρδος που λαμβάνεται από τους κόμβους που ανήκουν σε αυτή. Χρησιμοποιώντας το ακόλουθο σύνολο μεταβλητών απόφασης το πρόβλημα μπορεί να διατυπωθεί ως ένα ακέραιο πρόβλημα [36]:

- $y_{id} = 1$  Αν ο κόμβος  $i$  ( $i = 1, \dots, N$ ) περιλαμβάνεται στην υπο-διαδρομή  $d$  ( $d = 1, \dots, M$ ),  $y_{id} = 0$  διαφορετικά.
- $x_{ijd} = 1$  Αν η υπο-διαδρομή  $d$  ( $d = 1, \dots, M$ ) περιλαμβάνει το τόξο  $i, j$  ( $i, j = 1, \dots, N$ ), σε διαφορετική περίπτωση  $x_{ijd} = 0$ . Η συμμετρία του προβλήματος το εξασφαλίζει ότι  $t_{ij} = t_{ji}$ , άρα, ορίζονται μόνο τα  $x_{ijd}$  για  $i < j$ .

Η μαθηματική διατύπωση του CTOP είναι:

$$\text{Maximize Profit: } \sum_{i=2}^N \sum_{d=1}^M p_i y_{id} \quad (2.9)$$

$$\sum_{j=2}^N \sum_{d=1}^M x_{1jd} = \sum_{i=1}^N \sum_{d=1}^M x_{i1d} = M \quad (2.10)$$

$$\sum_{d=1}^M y_{id} \leq 1, \forall i = 2, \dots, N \quad (2.11)$$

$$\sum_{j \in N} x_{ijd} + \sum_{j \in N} x_{jid} = y_{jd}, \forall j = 2, \dots, N; \forall d = 1, \dots, M \quad (2.12)$$

$$\sum_{i \in N} \sum_{j \in N} t_{ij} x_{ijd} \leq T_{max}, \forall d = 1, \dots, M \quad (2.13)$$

$$\sum_{i \in N} D_i x_{ijd} \leq Q_{max}, \forall d = 1, \dots, M \quad (2.14)$$

$$\sum_{i \in S} \sum_{ij \in S} x_{ijd} \leq |S| - 1, \forall S \subseteq V, |S| \geq 2; \forall d = 1, \dots, M \quad (2.15)$$

$$x_{ijd}, y_{id} \in \{0, 1\}, 1 \leq i < j \leq N; d = 1, \dots, M \quad (2.16)$$

Η αντικειμενική συνάρτηση (2.9) μεγιστοποιεί το συνολικό κέρδος από όλες τις υπο-διαδρομές. Ο περιορισμός (2.10) επιβάλλει ότι οι υπο-διαδρομές αρχίζουν και τερματίζουν στον κόμβο 1. Ο περιορισμός (2.11) εξασφαλίζει ότι κάθε κόμβος θα επισκέπτεται το πολύ μια φορά. Επίσης, ο περιορισμός (4.1) διασφαλίζει τη δυνατότητα σύνδεσης ενός κόμβου με την διαδρομή. Οι περιορισμοί (2.13) και (2.14) επιβάλλουν διάρκεια ταξιδιού και τον περιορισμό χωρητικότητας για κάθε υπο-διαδρομή. Ο περιορισμός (2.15) απαγορεύει τις δευτερεύουσες διαδρομές. Τέλος, ο (2.16) καθορίζει τις μεταβλητές του προβλήματος.

### **2.3 Πρόβλημα Δρομολόγησης Οχημάτων με Συλλογή Βραβείου (Prize-collecting vehicle routing problem)**

Το επόμενο πρόβλημα που εξετάσει η παρούσα εργασία είναι το πρόβλημα δρομολόγησης οχημάτων με συλλογή βραβείου (Prize-collecting Vehicle Routing Problem (PCVRP)). Πρόκειται για μία παραλλαγή του κλασσικού προβλήματος δρομολόγησης οχημάτων (Vehicle routing problem (VRP)), τα κοινά σημεία των προβλημάτων είναι τα εξής:

- όλα τα οχήματα πρέπει να φύγουν και να επιστρέψουν στην ενιαία αποθήκη,
- ο στόλος αποτελείται από πανομοιότυπα οχήματα με την ίδια χωρητικότητα,
- κάθε κόμβος είναι επισκέψιμος μόνο μία φορά και από ένα όχημα
- και η συνολική ζήτηση πελατών που επισκέπτονται ένα όχημα δεν μπορεί να υπερβαίνει τη χωρητικότητα του οχήματος

Ωστόσο, το PCVRP έχει τα ιδιαίτερα χαρακτηριστικά του σε σύγκριση με το κλασσικό VRP. Αυτά τα χαρακτηριστικά συνοψίζονται ως εξής:

- ο αριθμός των διαθέσιμων οχημάτων είναι περιορισμένος, δηλαδή τα διαθέσιμα οχήματα είναι ανεπαρκή για να ικανοποιήσουν τις απαιτήσεις όλων των πελατών
- κάθε φορά που ένας κόμβος επισκέπτεται συλλέγεται ένα βραβείο (ή κέρδος) από αυτόν, κάτι που μπορεί να συμβάλει στη μείωση της αξίας της αντικειμενικής συνάρτησης

- δεν απαιτείται υποχρεωτικά η εξυπηρέτηση κάθε κόμβου

Στόχος είναι να καθοριστεί ένα σύνολο διαδρομών ώστε να ελαχιστοποιηθεί το συνολικό κόστος μεταφοράς και ταυτόχρονα να μεγιστοποιηθεί το βραβείο που συλλέγεται από όλα τα οχήματα. Το συνολικό κόστος μεταφοράς αποτελείται από τη συνολική απόσταση των οχημάτων και τον αριθμό των οχημάτων που χρησιμοποιούνται.

### 2.3.1 Βιβλιογραφική ανασκόπηση στους τρόπους επίλυσης του PCVRP

Το πρόβλημα δρομολόγησης οχημάτων με συλλογή βραβείου αποτελεί την επέκταση του πλανόδιου πωλητή με συλλογή βραβείου PCTSP, το οποίο προτάθηκε από τους Balas και Martin [41], ως μοντέλο για τον προγραμματισμό της καθημερινής λειτουργίας μιας μηχανής χαλύβδινης έλασης. Το PCVRP έχει ένα ισχυρό υπόβαθρο πρακτικής στη βιομηχανία και οι πρώτοι που το μοντελοποίησαν ήταν οι Tang και Wang [42], ενώ πρότειναν έναν επαναληπτικό αλγόριθμο τοπικής αναζήτησης (ILS) βασισμένο σε πολύ μεγάλης κλίμακας γειτονία (VLSN) με χρήση κυκλικής μεταφοράς. Για να ελεγχθεί η αποτελεσματικότητα του αλγορίθμου χρησιμοποιήθηκαν τα παραδείγματα αναφοράς του κλασικού προβλήματος δρομολόγησης οχημάτων με χωρητικότητα (CVRP), τα οποία είναι διαθέσιμα στον ιστότοπο: <http://www.branchandcut.org/VRP/data>. Ωστόσο, για να γίνει αντιστοίχιση με τα δεδομένα του PCVRP, έπρεπε να μειωθεί ο αριθμός των διαθέσιμων οχημάτων για κάθε περίπτωση κατά 1 ή 2, να ορισθεί ένα σταθερό κόστος κάθε οχήματος σε 1000 (το οποίο είναι πολύ μεγάλο σε σχέση με τις αποστάσεις μεταξύ κόμβων) και τέλος να δοθεί μία τιμή βραβείου σε κάθε κόμβο, η οποία παράγεται τυχαία στο [0, 100]. Τα αποτελέσματα του αλγορίθμου συγκρίθηκαν με τα αποτελέσματα δύο παραλλαγών του σε ίδιους χρόνους λειτουργίας. Στο ίδιο πλαίσιο, οι Zhang et al. [58], το 2009, χρησιμοποίησαν το PCVRP για να καθορίσουν τον προγραμματισμό μίας παρτίδας θερμής έλασης. Ο αλγόριθμος που πρότειναν για την επίλυση του είναι ένας αλγόριθμος βελτιστοποίηση σμήνους σωματιδίων (Particle Swarm Optimization (PSO)) με δύο χαρακτηριστικά. Συγκεκριμένα, κάνει χρήση μιας συνάρτησης χαρτογράφησης αριθμών, για τη μετατροπή των συνεχών τιμών των θέσεων των σωματιδίων σε διακριτές τιμές, σε όρους της σειράς, των ταινιών χάλυβα και ενσωματώνει μια αναζήτηση Tabu σε κάθε επανάληψη. Τα πειραματικά αποτελέσματα υποδηλώνουν ότι ο προτεινόμενος αλγόριθμος PSO ξεπέρασε τις επιδόσεις του παραδοσιακού. Πρόσφατα, οι Li και Tian [45] το 2016, παρουσίασαν έναν δύο επιπέδων προσαρμοστικό αλγόριθμο μεταβλητής γειτονιάς αναζήτησης (TL SAVNS). Αυτός ο αλγόριθμος έχει δύο χαρακτηριστικά

γνωρίσματα: το πρώτο είναι μια δομή VNS δύο επιπέδων και το δεύτερο είναι ένας αυτο-προσαρμοστικός μηχανισμός επιλογής γειτονιάς. Στη δομή VNS δύο επιπέδων το πρώτο επίπεδο αντιστοιχεί στην επιλογή των κόμβων που θα επισκέπτονται και το δεύτερο επίπεδο αντιστοιχεί στην δρομολόγηση των οχημάτων. Τα παραδείγματα που χρησιμοποιήσαν ήταν αυτά του (CVRP) και τα βραβεία ανα κόμβο παράχθηκαν τυχαία στο  $[1, 100]$ . Για να συγκρίνουν τα αποτελέσματά τους με άλλους αλγόριθμους, ανα-δημιούργησαν τους αλγόριθμους των (Vidal et al. [44], οι οποίοι είχαν παρουσιαστεί για ένα παρόμοιο πρόβλημα, το πρόβλημα της κερδοφόρα δρομολόγησης με χωρητικότητα (Capacitated Profitable Tour Problem (CPTP)). Ο αλγόριθμος τους αποδείχθηκε αρκετά ανταγωνιστικός.

### 2.3.2 Μαθηματική διατύπωση του PCVRP

Το πρόβλημα Prize-collecting Vehicle Routing Problem (PCVRP) μπορεί να περιγραφεί χρησιμοποιώντας ένα πλήρες γράφημα  $G = (V, A)$ , όπου  $V = \{1, \dots, N\}$  είναι το σύνολο κόμβων και  $A = \{(i, j) | i, j \in V\}$  είναι το σύνολο των τόξων. Κάθε κόμβος  $i$  που περιλαμβάνεται στο  $V$  σει σχετίζεται με μία βαθμολογία που στο εξής θα αναφέρεται ως βραβείο  $p_i$  και με μία τιμή ζήτησης  $D_i$ , επιπλέον ο απαιτούμενος χρόνος ταξιδιού μεταξύ κόμβων  $t_{ij}$  δίνεται για κάθε ζεύγος  $(i, j)$ . Ο κόμβος έναρξης έχει κέρδος ίσο με μηδέν. Για κάθε λύση (διαδρομή) πρέπει να δημιουργηθεί ένας περιορισμένος αριθμός υπο-διαδρομών  $M$ . Κάθε υπο-διαδρομή είναι εφικτή σε σχέση με ένα προκαθορισμένο όριο χωρητικότητας  $Q_{max}$ . Κάθε υπο-διαδρομή πρέπει να ξεκινά από τον αρχικό κόμβο 1 και να καταλήγει σε αυτόν, κάθε κόμβος είναι επισκέψιμος το πολύ μια φορά και ανήκει σε ακριβώς μία διαδρομή. Ο στόχος είναι να προσδιοριστεί το σύνολο των υπο-διαδρομών  $M$  που ελαχιστοποιούν την διαφορά συνολικού κόστους και συνολικού βραβείου, όπου στο κόστος περιλαμβάνεται ο συνολικός χρόνος ταξιδιού μεταξύ των κόμβων και ένα σταθερό κόστος  $G$  ανά υπο-διαδρομή. Ακόμα, λαμβάνουμε υπόψη την παράμετρο ολοκλήρωσης  $a$ , που προκύπτει από τη διαίρεση του προκαθορισμένου ποσού της ζήτησης από το σύνολο της ζήτησης όλων των κόμβων. Χρησιμοποιώντας το ακόλουθο σύνολο μεταβλητών απόφασης το πρόβλημα μπορεί να διατυπωθεί ως ένα ακέραιο πρόβλημα [42]:

- $y_{id} = 1$  Αν ο κόμβος  $i$  ( $i = 1, \dots, N$ ) περιλαμβάνεται στην υπο-διαδρομή  $d$  ( $d = 1, \dots, M$ ),  $y_{id} = 0$  διαφορετικά.
- $x_{ijd} = 1$  Αν η υπο-διαδρομή  $d$  ( $d = 1, \dots, M$ ) περιλαμβάνει το τόξο  $i, j$  ( $i, j = 1, \dots, N$ ), σε διαφορετική περίπτωση  $x_{ijd} = 0$ . Η συμμετρία του

προβλήματος το εξασφαλίζει ότι  $t_{ij} = t_{ji}$ , άρα, ορίζονται μόνο τα  $x_{ijd}$  για  $i < j$ .

Η μαθηματική διατύπωση του PCVRP είναι:

$$\text{Minimize : } \sum_{d \in M} \sum_{i \in N} \sum_{j \in N} t_{ij} x_{ijd} + G * \sum_{d=1}^M \sum_{i=2}^N x_{1id} - \sum_{i=2}^N \sum_{d=1}^M p_i y_{id} \quad (2.17)$$

$$\sum_{d=1}^M y_{id} \leq 1, \forall i = 2, \dots, N \quad (2.18)$$

$$\sum_{d=1}^M x_{i1d} \leq M, \forall i = 2, \dots, N \quad (2.19)$$

$$\sum_{d=1}^M x_{i1d} = \sum_{d=1}^M x_{1id}, \forall i = 2, \dots, N \quad (2.20)$$

$$\sum_{i \in N} D_i x_{ijd} \leq Q_{max}, \forall d = 1, \dots, M \quad (2.21)$$

$$\sum_{j=2}^N x_{ijd} = \sum_{i=1}^N x_{jid} = y_{id}, \forall i = 2, \dots, N; \forall d = 1, \dots, M \quad (2.22)$$

$$\sum_{(i,j) \in S} x_{ijd} \leq |S| - 1, \forall S \subset V \setminus \{1, N\}; 2 \leq |S| \leq N - 1; \forall d = 1, \dots, M \quad (2.23)$$

$$\frac{\sum_{d=1}^M \sum_{i=1}^N y_{id} D_i}{\sum_{i=1}^N D_i} \geq a \quad (2.24)$$

$$x_{ijd}, y_{id} \in \{0, 1\}, 1 \leq i < j \leq N; d = 1, \dots, M \quad (2.25)$$

Η αντικειμενική συνάρτηση (2.17) ελαχιστοποιεί το συνολικό κόστος από όλες τις υπο-διαδρομές, συνυπολογίζοντας το σταθερό κόστος και αφαιρώντας το συνολικό βραβείο (κέρδος) που έχει συλλεχθεί. Ο περιορισμός (2.18) εξασφαλίζει ότι κάθε κόμβος θα επισκέπτεται το πολύ μια φορά. Ο περιορισμός

(2.19) διασφαλίζει ότι ο αριθμός των οχημάτων που χρησιμοποιούνται είναι εφικτός. Ο περιορισμός (2.20) επιβάλλει ότι οι υπο-διαδρομές αρχίζουν και τερματίζουν στον κόμβο 1. Ο περιορισμός (2.21) διασφαλίζει τον περιορισμό χωρητικότητας για κάθε υπο-διαδρομή. Ακόμα, ο περιορισμός (2.22) διασφαλίζει τη δυνατότητα σύνδεσης ενός κόμβου με την διαδρομή. Ο περιορισμός (2.23) απαγορεύει τις δευτερεύουσες διαδρομές. Ο περιορισμός (2.24) είναι ένας περιορισμός ολοκλήρωσης για πραγματικά προβλήματα, και εξασφαλίζει ότι η ζήτηση των επισκεπτόμενων πελάτων δεν πρέπει να είναι μικρότερη από ένα προκαθορισμένο ποσό. Τέλος, ο (2.25) καθορίζει τις μεταβλητές του προβλήματος.

## Κεφάλαιο 3

# Αλγόριθμοι Εμπνευσμένοι από τη Φύση

### 3.1 Αλγόριθμος τεχνητής αποικίας μελισσών (Artificial bee colony algorithm)

Οι μέλισσες εκτελούν πολλά έξυπνα καθήκοντα ώστε να εξυπηρετήσουν τις ανάγκες της αποικίας και η αναζήτηση τροφής είναι ένα από τα πιο σημαντικά. Ο αλγόριθμος της τεχνητής αποικίας μελισσών (Artificial Bee Colony (**ABC**)) βασίζεται στην διαδικασία που ακολουθούν οι μέλισσες στη φύση, κατά την προσπάθειά τους να βρουν τροφή. Η αναζήτηση τροφής πραγματοποιείται με τη χρήση της συλλογής πληροφοριών που βασίζονται στην επικοινωνία και αλληλεπίδραση των μελισσών.

#### 3.1.1 Η δομή του ABC αλγόριθμου

Ο αλγόριθμος της τεχνητής αποικίας μελισσών (ABC) εμπίπτει στην κλάση των εξελικτικών αλγορίθμων, που μιμούνται την έξυπνη συμπεριφορά των μελισσών σε μια αποικία στην προσπάθειά τους να βρουν πηγές τροφής με νέκταρ. Η προσομοίωση της συμπεριφοράς των μελισσών, σε μια τεχνητή αποικία χωρίζει τις μέλισσες σε τρεις ομάδες: τις εργάτριες (employed), τις θεατές (onlookers) και τις ανιχνεύτριες (scout bees), ενώ η κάθε ομάδα έχει διαφορετικό ρόλο. Αρχίζοντας τη διαδικασία, οι εργάτριες μέλισσες εγκαταλείπουν την κυψέλη για να αναζητήσουν τυχαίες πηγές τροφής και να μετρήσουν την ποσότητα του νέκταρ μέσα τους. Στη συνέχεια, επιστρέφουν στην κυψέλη και μέσω της κίνησης τους, χορός, επικοινωνούν με τις υπόλοιπες μέλισσες, διαβιβάζοντας τους τις πληροφορίες σχετικά με την ποιότητα των πηγών τροφίμων (ποσότητα νέκταρ). Μετά την επικοινωνία, κάθε εργάτρια επιστρέφει στην πηγή τροφής που έχει



η ίδια προηγουμένως επισκεφθεί και ψάχνει για μία νέα πηγή κοντά στην αρχική. Οι μέλισσες αξιολογούν την ποσότητα του νέκταρ και αποθηκεύουν στη μνήμη τους τη θέση της πλουσιότερης πηγής τροφής. Οι θεατές μέλισσες πιθανολογικά και με βάση τις πληροφορίες για το νέκταρ των πηγών, αποφασίζουν ποια πηγή τροφής πρέπει να επισκεφθούν και να εκμεταλλευτούν. Ως εκ τούτου, οι θεατές βρίσκουν νέα πηγή τροφής στη γειτονιά των επιλεγμένων από τις εργάτριες και διατηρούν στη μνήμη τους την πλουσιότερη πηγή τροφής. Όταν μια πηγή τροφής εξαντλείται, τότε, η αντίστοιχη εργάτρια μέλισσα μετατρέπεται σε ανιχνεύτρια και κατευθύνεται προς τυχαία πηγή τροφής. Τα παραπάνω βήματα επαναλαμβάνονται, έως ότου οι μέλισσες επιστρέψουν στην κυψέλη και αυτή η κυκλική διαδικασία είναι το πλαίσιο του αλγορίθμου τεχνητής αποικίας μελισσών.

Αναλύοντας τον αλγόριθμο, οι πηγές τροφής αντιστοιχούν στις εφικτές λύσεις του προβλήματος βελτιστοποίησης και η ποσότητα νέκταρ που περιέχουν, αντιπροσωπεύει την ποιότητα της λύσης (την τιμή της αντικειμενικής συνάρτησης). Συνήθως ο αριθμός των εργατριών μελισσών είναι ίσος με των θεατών. Συνολικά, υπάρχουν τρεις παράμετροι που ελέγχουν τον αλγόριθμο, ο αριθμός των μελισσών  $SN$  (εργάτριες ή θεατές), ο μέγιστος αριθμός επαναλήψεων  $MC$  και το  $limit$ . Το  $limit$  είναι ένας αριθμός κατωφλίου που υποδηλώνει τον μέγιστο αριθμό επαναλήψεων κατά τις οποίες η λύση δεν βελτιώνεται και έτσι θα πρέπει να εγκαταλειφθεί και να ακολουθηθεί η διαδικασία των ανιχνευτριών. Οι εξισώσεις του αλγορίθμου ABC παρουσιάζονται στη συνέχεια :

$$P_i = \frac{F_i}{F(SN)} = \frac{F_i}{\sum_{k=1}^{SN} F(k)} \quad (3.1)$$

$$v_{ij} = z_{ij} + \phi_{ij}(z_{ij} - z_{kj}) \quad (3.2)$$

$$z_{ij} = z_{min,j} + rand[0, 1](z_{max,j} - z_{min,j}) \quad (3.3)$$

Σύμφωνα με την εξίσωση (3.1) υπολογίζεται η πιθανότητα  $P_i$  για την επιλογή μιας πηγής τροφής από τις θεατές μέλισσες, όπου  $F_i$  είναι η τιμή αντικειμενικής συνάρτησης της λύσης  $i$ . Οι θεατές και οι εργάτριες μέλισσες κινούνται σε μια νέα λύση  $v_{ij}$ , που παράγεται από την εξίσωση (3.2), βάση της αρχικής λύσης  $z_{ij}$  και μίας διαφορετικής  $z_{kj}$ . Κάθε λύση αναπαρήσταται από ένα  $D$ -σδιάστατο διάνυσμα, όπου  $j \in \{1 \cdots D\}$ ,  $k \in \{1 \cdots SN\}$  και όπου  $\phi_{ij}$  ένας τυχαίος αριθμός μεταξύ  $[-1, 1]$ . Η εξίσωση (3.3) ελέγχει τη συμπεριφορά των ανιχνευτριών, όπου  $z_{ij}$  είναι η λύση που πρέπει να εγκαταλειφθεί και  $z_{min,j}$ ,  $z_{max,j}$  είναι αντιστοίχα τα κατώτατα και ανώτατα όρια.

### 3.1.2 Βιβλιογραφική ανασκόπηση πάνω στην επίλυση VRP με χρήση του αλγορίθμου ABC

Ο αλγόριθμος της τεχνητής αποικίας μελισσών προτάθηκε αρχικά από τον Karaboga [46, 47, 48] το 2005, για προβλήματα βελτιστοποίησης χωρίς περιορισμούς. Οι Szeto και Ho [49] πρότειναν μια βελτιωμένη έκδοση της τεχνητής αποικίας των μελισσών για το πρόβλημα δρομολόγησης οχημάτων με χωρητικότητα (CVRP). Χρησιμοποίησαν τεχνικές γειτονικής αναζήτησης για να βρουν νέες πηγές τροφής κοντά στις αρχικές, όπως, την τυχαία ανταλλαγή, την παρεμβολή, αντιστροφή και άλλα. Αυτή η έκδοση του ABC διαφέρει από την πρωτότυπη όσον αφορά την συνθήκη σύμφωνα με την οποία μια αρχική λύση αντικαθίσταται από μία καινούργια. Το ίδιο πρόβλημα επιλύθηκε επίσης από τον Brajevic [50] το 2011, ο οποίος κωδικοποίησε τη λύση του προβλήματος σε δισδιάστατο πίνακα και εφάρμοσε παρόμοιες τεχνικές. Οι Ji και Wu [51] πρότειναν ένα αναθεωρημένο πλαίσιο για τον αλγόριθμο ABC για την επίλυση του προβλήματος δρομολόγησης οχημάτων με χωρητικότητα και χρονικά εξαρτώμενους χρόνους διαδρομής. Συγκεκριμένα πρότειναν, αντίθετα με την αρχική έκδοση του ABC, η καλύτερη λύση που θα βρεθεί από τις θεατές μέλισσες να αντικαταστήσει την πηγή τροφής που δεν έχει βελτιωθεί για τον μεγαλύτερο αριθμό επαναλήψεων και οι ανιχνεύτριες μέλισσες δεν θα είναι τυχαία παραγόμενες λύσεις, αλλά λύσεις που λαμβάνονται από τεχνικές γειτονικής αναζήτησης. Μια εναλλακτική στρατηγική επιλογής έχει προταθεί από τους Shi et al. [52] στον αλγόριθμο ABC-T, για την επίλυση του προβλήματος δρομολόγησης οχημάτων με χρονικά παράθυρα (VRPTW), πρόκειται για την επιλογή από μία ομάδα λύσεων (Tournament Selection). Ένα άλλο περίπλοκο πρόβλημα, το πρόβλημα τρισδιάστατης χωρητικότητας και φόρτωσης οχήματος (3L-CVRP) έχει επιλυθεί από έναν υβριδικό ABC αλγόριθμο με δύο ευρετικές μεθόδους από τους Wu et al. [53] το 2013. Στην έρευνα τους πρότειναν νέο τρόπο αναπαράστασης της λύσης και μια διαδικασία αποκωδικοποίησης βασισμένη σε λειτουργίες μονάδων σε πραγματικούς αριθμούς, μαζί με τις τοπικές διαδικασίες αναζήτησης 2-opt και 3-opt. Αργότερα, εφαρμόστηκε η ίδια ιδέα για το αντίστοιχο δισδιάστατο πρόβλημα (2L-CVRP) [54]. Οι Yao et al. [55] πρότειναν έναν βελτιωμένο αλγόριθμο τεχνητής αποικίας μελισσών για την επίλυση του περιοδικού προβλήματος δρομολόγησης (PVRP). Κατά τον οποίο εφαρμόζονται τροποποιημένοι κανόνες μεταφοράς, πολυδιάστατες ευρετικές πληροφορίες και τοπική βελτιστοποίηση βάσει στρατηγικής σάρωσης. Επιπλέον, το 2014, οι Gomez και Salhi [56] πρότειναν μια νέα παραλλαγή, το ABCLS. Ο αλγόριθμος διατηρεί τις βασικές αρχές του ABC αν και οι εξερευνητριες και οι ανιχνεύτριες μέλισσες επικεντρώνονται στη διαφοροποίηση της λύσης, ενώ οι θεατές στην εντατικοποίηση. Το 2014, οι Nahum et al. [57]

παρουσίασαν έναν αλγόριθμο βελτιστοποίησης βασισμένο στον ABC, ο οποίος υποδηλώνεται ως VE-ABC και περιλαμβάνει μια τεχνική αξιολόγησης για την επίλυση του VRPTW ως πρόβλημα πολλαπλών αντικειμενικών συναρτήσεων. Η έρευνα των Zhang et al. [58] επικεντρώθηκε στο περιβαλλοντικό πρόβλημα δρομολόγησης οχημάτων (EVRP) και σε έναν υβριδικό αλγόριθμο ABC για να το λύσει. Προκειμένου να επιταχυνθεί η σύγκλιση, συμπεριλήφθηκε ένας νέος μηχανισμός εκκίνησης και ενσωμάτωσης. Στο πλαίσιο του ABC, οι Iqbal et al. [59] πρότειναν, το 2015, το Bee\_VRPSTW, μια προσέγγιση για την επίλυση του προβλήματος δρομολόγησης οχημάτων με πολλές αντικειμενικές συναρτήσεις και χαλαρά χρονικά παράθυρα (VRPSTW). Στη συνέχεια, οι Zhang και Lee [60] παρουσίασαν τον αλγόριθμο RABC, έναν βελτιωμένο αλγόριθμο ABC με κατευθυνόμενη διεύθυνση και εισήγαγαν επίσης τον τελεστή διασταύρωσης και χαρτογράφησης διαδρομής (BMX).

Επιπλέον, πρόσφατες δημοσιεύσεις από το 2016, δείχνουν ότι η εφαρμογή του αλγορίθμου ABC σε προβλήματα δρομολόγησης οχημάτων παραμένει δημοφιλής. Οι Yin και Chuang [61] πρότειναν έναν υβριδικό συνδυασμό του προγραμματισμού προσαρμοστικής μνήμης (AMP) και του ABC, ο οποίος χαρακτηρίζεται ως αλγόριθμος AMABC, για την επίλυση του οικολογικού προβλήματος δρομολόγησης οχημάτων με cross-docking. Επίσης, οι Alzaqebah et al. [62] εστίασαν την τροποποίησή τους στις ανιχνεύτριες μέλισσες δίνοντάς τους τη μνήμη των εγκαταλελειμμένων λύσεων και την ικανότητα να τις αντικαταστήσουν με μη τυχαίες λύσεις. Προκειμένου να αντιμετωπιστεί ένα πρόβλημα πραγματικής περίπτωσης, ως VRPTW, οι Yu et al. [63] πρότειναν έναν βελτιωμένο αλγόριθμο ABC (IABC), ο οποίος εμπνεύστηκε από τους γενετικούς αλγόριθμους (GA) και περιλαμβάνει τη φάση διασταύρωσης και μετάλλαξης μετά τη λειτουργία των μελισσών. Ένα διαφορετικό πρόβλημα από το προηγούμενο, που εξετάστηκε στη βιβλιογραφία, το συνδυασμένο πρόβλημα δρομολόγησης του ανθρώπινου δυναμικού (CMVRP) με ομάδες με πολλαπλές δεξιότητες, έχει επιλυθεί από τους Seidgar et al. [64]. Πρότειναν τη χρήση των GA και ABC με ρύθμιση παραμέτρων σύμφωνα με τη μέθοδο Taguchi. Στην έρευνα που παρουσιάστηκε από τους Mao et al. [65] επιλύθηκε το αθέβαιο πρόβλημα χρόνου δρομολόγησης με χαλαρά χρονικά παράθυρα (UTDVRPTW) από μια νέα παραλλαγή η οποία χρησιμοποιεί ένα μηχανισμό για την παραγωγή αρχικών εφικτών λύσεων, έναν τροποποιημένο τελεστή εισαγωγής και έναν τελεστή αντίστροφης ανταλλαγής. Επιπλέον, οι Tuntitippawan και Asawarungsaengkul [66] παρουσίασαν έναν αλγόριθμο ABC σε συνδυασμό με μια  $\lambda$  - εναλλακτική τοπική αναζήτηση, για να λύσουν το πρόβλημα δρομολόγησης οχημάτων με παραλαβή και χρονικά παράθυρα (VRPBTW). Σχετικά με την έρευνά μας, υπάρχει μια δημοσίευση στη βιβλιογραφία σχετικά με το πρόβλημα προσα-

νατολισμού ομάδας (TOP) με χρονικά παράθυρα (TOPTW) που επιλύθηκε με τροποποιημένο ευρετικό ABC αλγόριθμο, από την Cura [67]. Ο προτεινόμενος αλγόριθμος εισάγει την ιδέα της αποδοχής μιας νέας πηγής τροφής με μικρή πιθανότητα, ακόμη και αν είναι χειρότερη από την αρχική και την ιδέα της ενεργοποίησης των ανιχνευτριών μελισσών σε τυχαία επιλεγμένες επαναλήψεις με πολύ μικρή πιθανότητα.

### 3.2 Αλγόριθμος της διαφορικής εξέλιξης (Differential evolution algorithm)

Ο αλγόριθμος της διαφορικής εξέλιξης (Differential evolution algorithm, (DE)) είναι ένας στοχαστικός αλγόριθμος, βασιζόμενος στον πληθυσμό, που προτάθηκε από τους Storn και Price [68], το 1995. Ο αλγόριθμος σχεδιάστηκε αρχικά για βελτιστοποίηση προβλημάτων συνεχών τιμών. Πρόκειται για έναν εξελικτικό αλγόριθμο για αυτό και περιέχει διαδικασίες μετάλλαξης (mutation), διασταύρωσης (crossover) και επιλογής (selection), ανάμεσα στα μέλη ενός πληθυσμού, για ένα αριθμό επαναλήψεων. Η μέθοδος του DE εστιάζει στην απόσταση μεταξύ των μελών του πληθυσμού και στις διαφορετικές κατευθύνσεις που μπορεί να κινηθεί κάποιο μέλος του πληθυσμού.

#### 3.2.1 Η δομή του DE αλγόριθμου

Η κύρια ιδέα του αλγόριθμου της διαφορικής εξέλιξης είναι η διατάραξη ενός πληθυσμού διανυσμάτων μέσω διανυσματικών διαφορών και ανασυνδυασμών, για έναν αριθμό επαναλήψεων (γενιών),  $NG$ . Τα διανύσματα (άτομα) του πληθυσμού αποτελούν πραγματικές κωδικοποιημένες λύσεις του προβλήματος. Ένα από τα πλεονεκτήματα του αλγορίθμου είναι ο μικρός αριθμός παραμέτρων ελέγχου. Καθοριστικό ρόλο διαδραματίζουν: το μέγεθος του πληθυσμού ( $NP$ ), ο ρυθμός μετάλλαξης ( $F$ ) και ο ρυθμός διασταύρωσης ( $Cr$ ). Αρχικά, δημιουργείται ένας τυχαία διαταραγμένος πληθυσμός ατόμων  $NP$ . Κάθε μέλος του πληθυσμού είναι ένα  $D$ -διαστασιακό διάνυσμα  $x_{ij}$ , όπου  $i \in \{1, \dots, NP\}$  και  $j \in \{1, \dots, D\}$ . Η πρώτη εξελικτική διαδικασία που λαμβάνει χώρα σε κάθε γενιά είναι η μετάλλαξη, έτσι ώστε, να παραχθεί ένα δοκιμαστικό διάνυσμα για κάθε μέλος του πληθυσμού. Κατά τη διάρκεια της μετάλλαξης ενός διανύσματος  $x_{ij}(t)$ , στην επανάληψη  $t$ , επιλέγονται τρία τυχαία διανύσματα από τον πληθυσμό, για την ακρίβεια, ένα διάνυσμα βάσης ( $i_1 \neq i$ ) και δύο άλλα ( $i \neq i_1 \neq i_2 \neq i_3$ ). Τα διανύσματα αυτά λαμβάνουν μέρος στην εξίσωση (3.4). Η διαφορά  $x_{i_2}$  και  $x_{i_3}$  ενισχύεται από το ποσοστό μετάλλαξης  $F$ , το οποίο είναι μια

πραγματική τιμή μεταξύ 0 και 2. Η κλιμακωτή διαφορά προστίθεται στο διάνυσμα βάσης για να σχηματιστεί το μεταλλαγμένο διάνυσμα  $v_{ij}(t+1)$  (mutant vector), για κάθε διάνυσμα στόχου του πληθυσμού,  $x_{ij}(t)$  (target vector).

$$v_{ij}(t+1) = x_{i1j}(t) + F * (x_{i2j}(t) - x_{i3j}(t)) \quad (3.4)$$

Μετά την ολοκλήρωση της διαδικασίας της μετάλλαξης εφαρμόζεται ένας τελεστής διασταύρωσης. Συγκεκριμένα, χρησιμοποιείται ο **δυωνυμικός τελεστής διασταύρωσης** (binomial crossover) ή **ομοιόμορφος τελεστής διασταύρωσης** (uniform crossover). Σύμφωνα με αυτόν τον τελεστή διασταύρωσης, παράγεται το δοκιμαστικό διάνυσμα  $u_{ij}(t+1)$  (trial vector), ανασυνδιάζοντας το διάνυσμα στόχου και το μεταλλαγμένο, που του αντιστοιχεί. Η εξίσωση (3.5), ορίζει ποιά στοιχεία του δοκιμαστικού διανύσματος θα κληροδοτηθούν από το διάνυσμα στόχου και από τα μεταλλαγμένο. Καθοριστικό ρόλο στη διαδικασία έχει η παράμετρος  $C_r$ , η οποία έχει πραγματική τιμή από το 0 μέχρι το 1. Έτσι για κάθε στοιχείο  $j$ , παράγεται ένας τυχαίος αριθμός  $\phi$  και ένας τυχαίος δείκτης  $j_{rand}$ , όπου  $\phi, j_{rand} \in [0, 1]$ . Εάν ο τυχαίος αριθμός  $\phi$  είναι μικρότερος ή ίσος από την τιμή της παραμέτρου  $C_r$ , ή αν ο δείκτης της παραμέτρου ισούται με  $j_{rand}$ , το δοκιμαστικό διάνυσμα κληρονομεί το αντίστοιχο στοιχείο από το μεταλλαγμένο διάνυσμα και διαφορετικά από το διάνυσμα στόχου. Το  $j_{rand}$  εξασφαλίζει ότι τουλάχιστον ένα στοιχείο θα προωθηθεί στο δοκιμαστικό από το μεταλλαγμένο διάνυσμα.

$$u_{ij}(t+1) = \begin{cases} v_{ij}(t+1), & \text{αν } \phi \leq C_r \text{ ορ } j = j_{rand} \\ x_{ij}(t), & \text{διαφορετικά} \end{cases} \quad (3.5)$$

Οι δύο παραπάνω διαδικασίες, η μετάλλαξη και η διασταύρωση, αυξάνουν την ποικιλομορφία του πληθυσμού και επομένως διεξάγουν τη φάση της εξερεύνησης κατά την αναζήτηση. Προκειμένου να συμπεριληφθεί μια φάση εντατικοποίησης της αναζήτησης και να διατηρηθεί το μέγεθος του πληθυσμού, εκτελείται και μια διαδικασία επιλογής. Έτσι, μετά από τη διαδικασία διασταύρωσης, πρέπει να διατηρηθεί στον πληθυσμό ή το διάνυσμα στόχου ή το δοκιμαστικό διάνυσμα και το άλλο να απορριφθεί. Η διαφορική εξέλιξη χρησιμοποιεί μια άπληστη τεχνική σε αυτή τη φάση και το διάνυσμα που αντιστοιχεί στην υψηλότερη τιμή της αντικειμενικής συνάρτησης θα συμπεριληφθεί στον πληθυσμό της επόμενης γενιάς, όπως φαίνεται στην εξίσωση (3.6).

$$x_{ij}(t+1) = \begin{cases} u_{ij}(t+1), & \text{αν } f(u_{ij}(t+1)) \leq f(x_{ij}(t)) \\ x_{ij}(t), & \text{διαφορετικά} \end{cases} \quad (3.6)$$

Έχουν αναπτυχθεί αρκετές παραλλαγές του βασικού αλγόριθμου διαφορικής εξέλιξης με την πάροδο των ετών, οι οποίες περιέχουν διαφορετικούς συνδυασμούς από τις βασικές διαδικασίες που παρουσιάστηκαν παραπάνω. Οι διαφορετικές παραλλαγές αναπαρίστανται από μια γενική συμβολοσειρά  $DE/x/y/z$ , όπου το  $x$  αναφέρεται στον τρόπο με τον οποίο επιλέγεται ένα διάνυσμα-στόχος, το  $y$  είναι ο αριθμός των διανυσμάτων διαφοράς που χρησιμοποιούνται και το  $z$  αναφέρεται τον τρόπο διασταύρωσης. Ο συνδυασμός που περιγράψαμε αναφέρεται ως  $DE/rand/1/bin$ , επομένως το διάνυσμα στόχος επιλέγεται τυχαία, χρησιμοποιείται ένα ζεύγους διανυσμάτων στην εξίσωση μετάλλαξης και εφαρμόζεται η διωνυμική διασταύρωση.

### 3.2.2 Βιβλιογραφική ανασκόπηση πάνω στην επίλυση VRP με χρήση του αλγορίθμου DE

Μετά από δεκατέσσερα χρόνια από την παρουσίαση του αλγορίθμου της διαφορικής εξέλιξης, το 2009, ακολούθησε δημοσίευση πάνω στη χρήση του για την επίλυση προβλήματος δρομολόγησης οχημάτων, από τους Erbao και Mingyong [69]. Συγκεκριμένα σχεδίασαν έναν υβριδικό αλγόριθμο, στοχαστικής προσομοίωσης και διαφορικής εξέλιξης για την επίλυση του προβλήματος δρομολόγησης οχημάτων με ασαφή ζήτηση (VRPFD). Οι ίδιοι τον επόμενο χρόνο, χρησιμοποίησαν τον ίδιο αλγόριθμο για την επίλυση του ανοιχτού προβλήματος δρομολόγησης οχημάτων με ασαφή ζήτηση [70]. Ενώ, στη συνέχεια, [71], πρότειναν έναν βελτιωμένο αλγόριθμο διαφορικής εξέλιξης (IDE) για το πρόβλημα της δρομολόγησης με ταυτόχρονες παραλαβές και παραδόσεις και χρονικά παράθυρα (VRP-SPDTW), ενώ πρότειναν μια δεκαδική κωδικοποίηση για την κατασκευή του αρχικού πληθυσμού. Το 2013, οι Hou, [72] παρουσίασαν έναν διακριτό αλγόριθμο διαφορικής εξέλιξης, τον DDE, για το στοχαστικό πρόβλημα της δρομολόγησης με ταυτόχρονες παραλαβές και παραδόσεις. Οι Liu et al. [73], το 2012, πρότειναν έναν βελτιωμένο αλγόριθμο μιμητικού DE με γενικευμένη αντικειμενική συνάρτηση (MDEGF) για το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα, σε συνδυασμό με τρία είδη τοπικής αναζήτησης. Αργότερα, οι Kucukoglu και Ozturk [74] χρησιμοποίησαν αλγόριθμο DE στην έρευνα τους πάνω στο πρόβλημα δρομολόγησης οχημάτων με backhauls και χρονικά παράθυρα (VRPBW) σε πρακτική εφαρμογή. Το 2014, οι Berhan et al. [75], επέλυσαν το στοχαστικό πρόβλημα δρομολόγησης οχημάτων με ταυτόχρονες παραλαβές και παραδόσεις, εφαρμοσμένο σε πραγματικές συνθήκες, σύμφωνα με ένα μοντέλο βασισμένο στον αλγόριθμο DE. Έπειτα, το 2015, οι Dechamprai et al. [76] ασχολήθηκαν με την εκτεταμένη έκδοση του VRP με παράδοση και παραλαβή, κατασκευάζοντας έναν ευρετικό

αλγόριθμο βασισμένο στον (DE). Ακόμα, οι Marinakis et al.[77], παρουσίασαν μία μιμητική παραλλαγή του αλγόριθμου της διαφορικής εξέλιξης και τον εφάρμοσαν στην επίλυση του προβλήματος δρομολόγησης οχημάτων με στοχαστική ζήτηση (SVRP). Το 2016, οι Psychas et al. [78], πρότειναν τρεις παράλληλους, μη κυριαρχούμενους αλγόριθμους διαφορικής εξέλιξης πολλαπλών εκκινήσεων (PMS-NSDEs), για τη λύση τεσσάρων προβλημάτων δρομολόγησης οχημάτων με κατανάλωση καυσίμου (MRFCVRPs). Τέλος, η πιο πρόσφατη δημοσίευση είναι των Viktorin et al. [79], στην οποία χρησιμοποιείται μία νέα πολύ-χαστική παραλλαγή του αλγορίθμου της διαφορικής εξέλιξης, για την επίλυση ενός μοντέλου προβλήματος δρομολόγησης οχημάτων με κέρδη.

## Κεφάλαιο 4

# Κωδικοποίηση Distance Related σε Μεθευρετικούς Αλγορίθμους

### 4.1 Το μοντέλο του Distance Related

Σε αυτό το κεφάλαιο, περιγράφεται η ανάγκη που μας οδήγησε στην δημιουργία του μοντέλου Distance Related (**DR**). Το προτεινόμενο μοντέλο αποτελεί μια μέθοδο προσαρμογής των μεθευρετικών αλγορίθμων βελτιστοποίησης στην επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης και πιο συγκεκριμένα στο πρόβλημα δρομολόγησης οχημάτων.

Ένας μεγάλος αριθμός μεθευρετικών αλγορίθμων έχει αναπτυχθεί για την επίλυση προβλημάτων βελτιστοποίησης, οι οποίοι στην πλειοψηφία τους έχουν προταθεί για την επίλυση προβλημάτων στο συνεχές χώρο λύσεων. Για παράδειγμα, οι εξελικτικοί αλγόριθμοι και πολλοί αλγόριθμοι εμπνευσμένοι από την φύση, χρησιμοποιούν, στις κλασσικές εκδοχές τους, εξισώσεις συνεχών τιμών. Ενώ, τα προβλήματα που εξετάζουμε απαιτούν διακριτές τιμές, και κατά συνέπεια διακριτά κωδικοποιημένα διανύσματα λύσεων, στα οποία οι εξισώσεις αυτές δεν μπορούν να εφαρμοστούν με άμεσο τρόπο. Το μοντέλο Distance Related (DR) που προτείνουμε, βασίζεται κυρίως σε ένα καινοτόμο τρόπο κωδικοποίησης του διανύσματος λύσης, σε προβλήματα δρομολόγησης οχημάτων καθώς και σε έναν απλό ευρετικό αλγόριθμο για την αποκωδικοποίηση του.

Στη βιβλιογραφία προτείνονται κάποιοι τρόποι αντιμετώπισης αυτού του προβλήματος, είτε με υβριδοποίηση των κλασσικών αλγορίθμων, είτε με εναλλακτική κωδικοποίηση του διανύσματος λύσης. Ενδεικτικά αναφέρουμε την πρόταση των Snyder και Daskin [80], οι οποία χρησιμοποίησαν την κωδικοποίηση των τυχαίων κλειδιών [81] για την επίλυση του προβλήματος του γενικευμένου πλανόδιου πωλητή με χρήση γενετικού αλγορίθμου. Ένα παράδειγμα με



σύνολο κόμβων  $V = 1, \dots, 20$ , με υποσύνολα  $V = 1, \dots, 5$ ,  $V = 6, \dots, 10$ ,  $V = 11, \dots, 15$ ,  $V = 16, \dots, 20$ . Έστω η κωδικοποίηση της λύσης ως: [4,32 307 1,80 3,76]. Το ακέραιο μέρος της τιμής αντιπροσωπεύει ποιο ψηφίο θα επιλεγεί από το κάθε υποσύνολο βάση της θέσης του. Τα δεκαδικά ψηφία διαβάζονται κατά αύξοντα τρόπο και υποδηλώνουν το υποσύνολο από το οποίο θα επιλεγεί ο κόμβος. Έτσι, η αποκωδικοποιημένη λύση είναι η: [8  $\rightarrow$  4  $\rightarrow$  18  $\rightarrow$  11].

Το μοντέλο κωδικοποίησης DR που παρουσιάζεται στη παρούσα μεταπτυχιακή διατριβή, βασίζεται **στις ευκλείδειες αποστάσεις μεταξύ των κόμβων**. Στόχος, ήταν να δοθεί μία πραγματική διάσταση στον τρόπο κωδικοποίησης του διανύσματος λύσης, να είναι δηλαδή άρρηκτα συνδεδεμένος με τα δεδομένα του κάθε προβλήματος. Το μοντέλο Distance Related, είναι καινοτόμο, αφού σύμφωνα με την βιβλιογραφική έρευνα που έχει διεξαχθεί, δεν έχει προταθεί κάτι αντίστοιχο σε αυτό. Τα βασικά πλεονεκτήματα της κωδικοποίησης DR είναι:

- ότι μπορεί να εφαρμοστεί μέσα στο πλαίσιο κάθε μεθευρετικού αλγορίθμου που απαιτεί πράξεις μεταξύ συνεχών τιμών,
- ότι μπορεί να χρησιμοποιηθεί σε κάθε τύπο προβλήματος δρομολόγησης οχημάτων,
- ότι τα δεδομένα που απαιτούνται για την κωδικοποίηση είναι αμετάβλητα και δεν επηρεάζονται από την διαδικασία αναζήτησης,
- ότι απαιτεί χαμηλής πολυπλοκότητας αποκωδικοποίηση,
- και ότι μπορεί να προσαρμοστεί και σε άλλα προβλήματα, όπως στα προβλήματα προγραμματισμού εργασιών, αντικαθιστώντας τις τιμές απόστασης, με χρόνους εργασίας.

Ας δώσουμε ένα γενικό παράδειγμα: Έστω μία εφικτή λύση του προβλήματος προσανατολισμού ομάδας, με 21 κόμβους και 2 υπο-διαδρομές:

[1 $\rightarrow$ 11 $\rightarrow$ 10 $\rightarrow$ 2 $\rightarrow$ 7 $\rightarrow$ 21|1 $\rightarrow$ 13 $\rightarrow$ 9 $\rightarrow$ 14 $\rightarrow$ 21]

Η κωδικοποιημένη λύση θα αποτελείται από την ευκλείδεια απόσταση κάθε κόμβου από τον προηγούμενο του:

[- 1.88 1.43 3.77 3.27 2.86 **1.55** 1.23 4.24 4.51 1.71]

Όπου, 1.88 μονάδες είναι η απόσταση του κόμβου 11 από τον κόμβο 1, 1.43 μονάδες είναι η απόσταση του 10 από τον 11 κόμβο και ούτω καθεξής. Παρατηρούμε ότι στο κωδικοποιημένο διάνυσμα λύσης, οι υπο-διαδρομές μπορούν

εύκολα να διακριθούν μέσα στη λύση αφού η απόσταση μεταξύ του πρώτου και του τελευταίου κόμβου έχει γνωστή και σταθερή τιμή σε όλη την διάρκεια μιας αλγοριθμικής μεθόδου. Είναι γεγονός ότι οι τιμές του διανύσματος λύσης θα αλλάζουν σύμφωνα με την εξίσωση του εκάστοτε αλγορίθμου, όπως είναι η εξίσωση κίνησης του ABC (εξ. 3.2) ή η διαδικασία μετάλλαξης του DE (εξ. 3.4). Έτσι, το νέο διαταραγμένο διάνυσμα λύσης θα μπορούσε να είναι το: [ - 2.20 1.36 0.86 2.43 1.58 1.17 0.86 1.16 0.48 0.66]. Περνώντας στη διαδικασία της αποκωδικοποίησης, χρησιμοποιείται ένας απλός ευρετικός αλγόριθμος, που ανακατασκευάζει το διάνυσμα λύσης βηματικά. Η λειτουργία της αποκωδικοποίησης βασίζεται στην αντιστοίχιση των νέων τιμών του διανύσματος σε πραγματικές τιμές απόστασης μεταξύ κόμβων. Αναλυτικότερα, κάθε λύση ξεκινάει από τον κόμβο 1, μετά βάση της τιμής στην 2η θέση του διανύσματος, 2.20, βρίσκονται οι κόμβοι που έχουν απόσταση από τον κόμβο 1, πιο κοντά στην τιμή αυτή (2.20). Έστω ότι οι ταξινομημένοι κόμβοι είναι οι: 14, 11, 6, 7, 10 με τις αντίστοιχες αποστάσεις 2.46, 1.87, 2.91, 1.31, 3.23. Έτσι, επιλέγεται ο κόμβος 14 και αν δεν έχει χρησιμοποιηθεί και δεν παραβιάζει τους περιορισμούς, εντάσσεται στην διαδρομή: [1 14]. Στην συνέχεια, ο αλγόριθμος εστιάζει στην 3η θέση του διανύσματος, στην τιμή 1.36, και αναζητούνται οι κόμβοι που απέχουν από τον 14 μονάδες απόστασης κοντά στη τιμή αυτή (1.36). Έστω ότι τώρα οι ταξινομημένοι κόμβοι είναι οι: 13, 21, 11, 10, 12 με τις αντίστοιχες αποστάσεις 1.34, 1.72, 2.14, 2.60, 2.72. Έτσι, επιλέγεται ο κόμβος 13 και αν δεν έχει χρησιμοποιηθεί και δεν παραβιάζει τους περιορισμούς, εντάσσεται στην διαδρομή: [1 14 13]. Η διαδικασία αυτή συνεχίζεται μέχρι να αποκωδικοποιηθεί όλο το διάνυσμα λύσης, ενώ αν παραβιαστεί κάποιος περιορισμός του προβλήματος, δημιουργείται νέα διαδρομή.

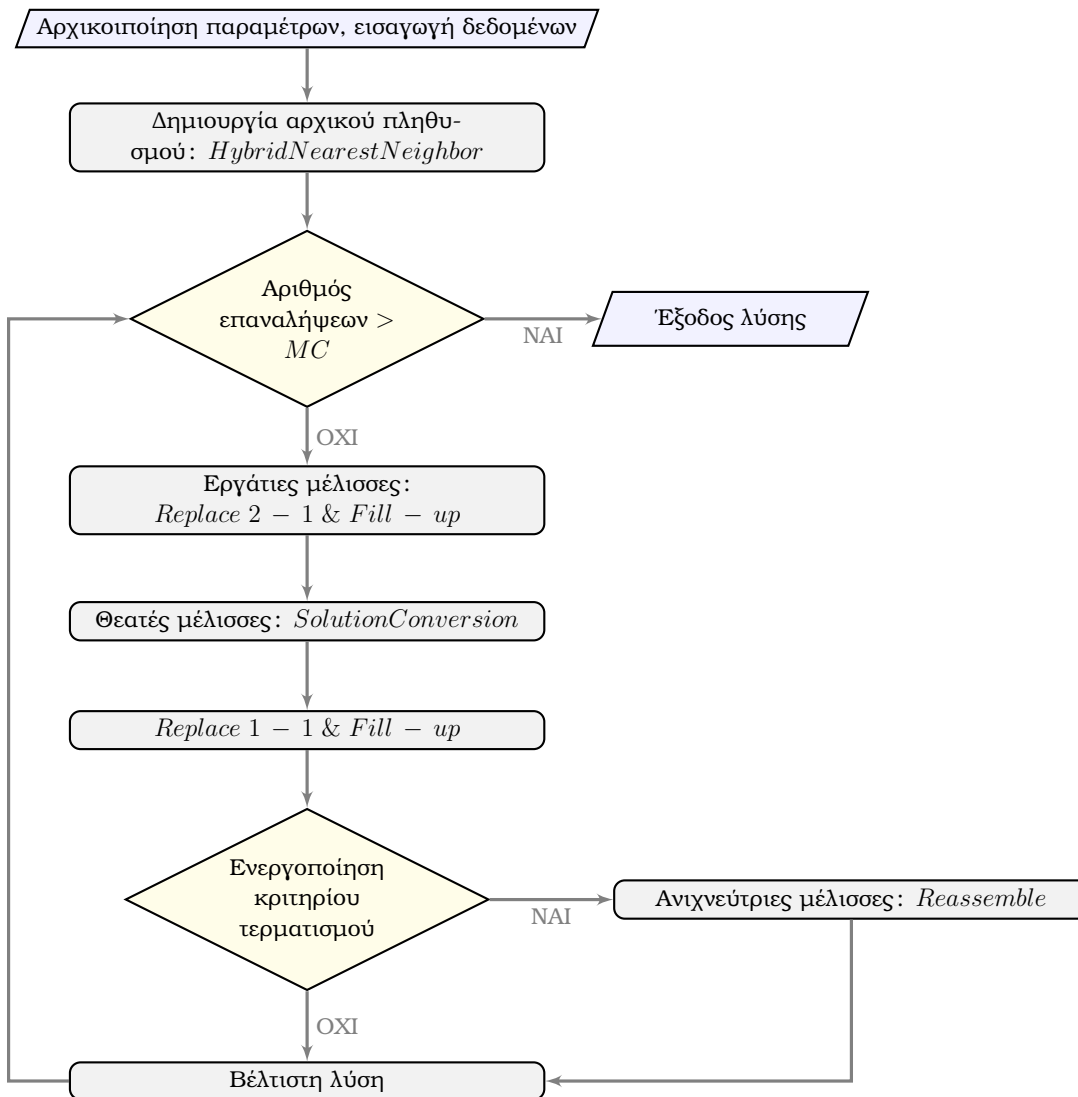
Στη συνέχεια του κεφαλαίου παρουσιάζεται η ενσωμάτωση του μοντέλου Distance Related στον αλγόριθμο της τεχνητής αποικίας μελισσών, που θα συμβολίζεται ως DRABC και στον αλγόριθμο της διαφορικής εξέλιξης, που θα συμβολίζεται ως DRDE.

#### 4.1.1 Αλγόριθμος Distance Related Artificial Bee Colony

Ο πρώτος αλγόριθμος που προτείνεται είναι ένας υβριδικός αλγόριθμος της τεχνητής αποικίας μελισσών που ενσωματώνει το μοντέλο Distance Related, οποίος αναφέρεται ως DRABC. Το αλγοριθμικό πλαίσιο που παρουσιάζεται περιλαμβάνει: έναν ευρετικό αλγόριθμο, έναν υβριδικό πλησιέστερο γείτονα (*HybridNearestNeighbor*) για την δημιουργία του αρχικού πληθυσμού, μία διαδικασία τοπικής αναζήτησης που αντιστοιχεί στις εργάτριες μέλισσες (*Replace2 - 1*), έναν ευρετικό αλγόριθμο αποκωδικοποίησης που αντιστοιχεί

στις θεατές μέλισσες (*SolutionConversion*), μία διαδικασία τοπικής αναζήτησης που αντιστοιχεί στις ανιχνεύτριες (*Reassemble*), καθώς και δύο τεχνικές τοπικής αναζήτησης (*Fill - up*, *Replace1 - 1*) για να ενισχύσουν τις βασικές αυτές διαδικασίες. Η αλγοριθμική διαδικασία του DRABC περιγράφεται αναλυτικά στη συνέχεια και παρουσιάζεται στο παρακάτω διάγραμμα ροής, Σχήμα 4.1.

Σχήμα 4.1: Διάγραμμα ροής του αλγόριθμου DRABC



Σε αυτό το σημείο είναι σημαντικό να αναφερθεί ότι ο αλγόριθμος DRABC έχει

προσαρμοστεί και εκτελεστεί πάνω στην επίλυση των τριών προβλημάτων δρομολόγησης οχημάτων με κέρδος, τα οποία παρουσιάστηκαν στο Κεφάλαιο 2. Η ανάλυση του αλγορίθμου DRABC (βλ. Αλγόριθμος 1) που ακολουθεί βασίζεται στη επίλυση του προβλήματος προσανατολισμού ομάδας Team Orienteering Problem (TOP).

### Αναπαράσταση λύσης και δημιουργία αρχικού πληθυσμού

Όπως έχει αναφερθεί, για την επίλυση του προβλήματος προσανατολισμού ομάδας απαιτούνται: το σετ κόμβων  $V = \{1, \dots, N\}$ , το σκορ  $s_i$  κάθε κόμβου  $i$ , ο αριθμός των υπο-διαδρομών  $M$  και το προκαθορισμένο όριο ταξιδιού  $T_{max}$ . Τα δεδομένα αυτά εισάγονται στον αλγόριθμο και υπολογίζεται η ευκλείδεια απόσταση  $t_{ij}$  μεταξύ όλων των κόμβων. Μία εφικτή λύση αποτελείται από  $M$  υπο-διαδρομές, οι οποίες ξεκινάνε από τον κόμβο 1 και τερματίζουν στον κόμβο  $N$ , έτσι η λύση αναπαρίσταται από ένα διάνυσμα ακέραιων τιμών, συμβολίζοντας την αλληλουχία των κόμβων στη διαδρομή. Το μέγιστο μήκος του διανύσματος είναι  $N + (M - 1) * 2$ . Ακόμα, ορίζονται το μέγεθος του πληθυσμού  $SN$  και ο μέγιστος αριθμός επαναλήψεων του αλγορίθμου  $MC$ . Μετά την αρχικοποίηση, παράγεται ο αρχικός πληθυσμός λύσεων, σύμφωνα με τον υβριδικό ευρετικό αλγόριθμο του πλησιέστερου γείτονα (*HybridNearestNeighbor*), Αλγόριθμος 2. Ένα παράδειγμα λύσης για  $N = 21$  κόμβους και  $M = 3$  υπο-διαδρομές, θα είναι ένα διάνυσμα της μορφής: [1,5,3,21,1,18,20,11,21,1,4,12,9,15,21].

Για να φτάσουμε σε αυτή την ολοκληρωμένη διαδρομή ο απλός κατασκευαστικός αλγόριθμος που παρουσιάζεται, αρχικοποιεί την λύση με τον κόμβο 1 και στην συνέχεια προσθέτει κόμβους στην διαδρομή. Ο κλασικός πλησιέστερος γείτονας ορίζει ότι ο υποψήφιος κάθε φορά κόμβος για να εισαχθεί στην διαδρομή, είναι ο κοντινότερος σε μονάδες απόστασης από τον τελευταίο σε αυτή. Όμως ο ευρετικός που παρουσιάζεται προκειμένου να ανταποκριθεί στις απαιτήσεις της αντικειμενικής συνάρτησης του προβλήματος TOP, επιλέγει τον υποψήφιο κόμβο βάση μίας πιθανότητας που ελέγχεται από μία γεννήτρια τυχαίου αριθμού. Έτσι, εάν ο αριθμός είναι μεγαλύτερος από την τιμή 0,7 επιλέγεται ο κοντινότερος κόμβος που δεν έχει χρησιμοποιηθεί (όπως και στον κλασικό), ενώ σε αντίθετη περίπτωση, επιλέγεται ο κόμβος με το μεγαλύτερο σκορ, που δεν έχει χρησιμοποιηθεί. Είναι φανερό ότι στη δεύτερη περίπτωση ενισχύουμε την ποιότητα της λύσης. Για κάθε υποψήφιο κόμβο ελέγχονται οι περιορισμοί του προβλήματος, δηλαδή η συνολική απόσταση της διαδρομής να μην ξεπερνάει την τιμή του  $T_{max}$ . Όταν ο υποψήφιος κόμβος παραβιάζει τον περιορισμό, ελέγχονται και οι υπόλοιποι κόμβοι που δεν έχουν χρησιμοποιηθεί, όταν κανένας κόμβος δεν μπορεί να εισαχθεί στην λύση, τότε η υπο-διαδρομή

**Αλγόριθμος 1** DRABC

---

```

1: Αρχικοποίηση παραμέτρων, εισαγωγή δεδομένων
2: Δημιουργία αρχικών λύσεων  $z_{ij}$ ,  $i=1, \dots, SN$ ,  $\theta=1, \dots, N + (M - 1) * 2$  //
   βλ. Αλγόριθμος 2
3: Εκτίμηση  $F_i$  // βλ. Εξίσωση 2.1
4:  $l := 0$ ,  $lm := 0$ 
5: while  $l \leq MC$ 
6: Αντιστοίχιση κάθε εργάτριας μέλισσας σε λύση,  $z_{ij}$ 
7: for  $i = 1 : SN$ 
8: Χρήση Replace 2 – 1 για κάθε  $z_{ij}$  // βλ. Αλγόριθμος 3
9: Χρήση Fill – up για  $z_{ij} \rightarrow z'_{ij}$  // βλ. Αλγόριθμος 4
10: if  $F_{z'_{ij}} > F_{z_{ij}}$ 
11:  $z_{ij} \leftarrow z'_{ij}$ 
12: else
13:  $lm_i := lm_i + 1$ 
14: end if
15: end for
16: for  $i = 1 : SN$ 
17: Για κάθε θεατή μέλισσα  $i$  επιλογή λύσεων βάση του κανόνα της ρουλέτας
   // βλ. Εξίσωση 3.1
18: Χρήση SolutionConversion για  $v_{ij}$  // βλ. Αλγόριθμος 5
19: Χρήση Replace 1 – 1 για  $v_{ij} \rightarrow v'_{ij}$  // βλ. Αλγόριθμος 6
20: Χρήση Fill – up για  $v'_{ij}$  // βλ. Αλγόριθμος 4
21: Έυρεση  $v'_{ij} : \max F_{v'_{ij}} \forall v'_{ij}$ 
22: if  $F_{v'_{ij}} > F_{v_{ij}}$ 
23:  $v_{ij} \leftarrow v'_{ij}$ 
24: else
25:  $lm_i := lm_i + 1$ 
26: end if
27: if  $lm_i == limit$ 
28: Αντικατάσταση λύσης  $i$  από  $z_{ij}$  // βλ. Αλγόριθμος 6
29:  $lm_i := 0$ 
30: end if
31: end for
32:  $best_{z_{ij}} \leftarrow i : \max F_i$ 
33: end while

```

---

τερματίζει και ξεκινάει η επόμενη, η διαδικασία συνεχίζεται μέχρι να δημιουργηθεί ο απαιτούμενος αριθμός υπο-διαδρομών.

**Αλγόριθμος 2** Δημιουργία αρχικού πληθυσμού: *HybridNearestNeighbor*


---

```

1: Εισαγωγή:  $V, M, T_{max}, s_i, t_{ij}$ 
2:  $V' := V$ 
3: Αφαιρούμε απο το σετ  $V'$  τους κόμβους 1,  $N$ 
4: Αρχικοποίηση διαδρομής:  $z_{ij} = [1]$ 
5:  $Total = 0$ 
6: while  $d < M$ 
7:   for  $\forall n \in V'$ 
8:     if  $Total + t_{z_{i,j-1},n} + t_{n,N} \leq T_{max}$ 
9:        $Total \leftarrow Total + t_{z_{i,j-1},n}$ 
10:    Ενημέρωση  $Score, Total$  υπο-διαδρομής
11:     $z_{ij} \leftarrow n$ , αφαίρεση  $n$  από  $V'$ 
12:     $j = j + 1$ 
13:   end if
14:    $d = d + 1$ 
15: end for
16: end while
17: Έξοδος:  $z_{ij}$ 

```

---

**Στάδιο των εργατριών μελισσών**

Έπειτα από την δημιουργία του αρχικού πληθυσμού λύσεων, πραγματοποιείται η διαδικασία που αντιστοιχεί στις εργατρίες μέλισσες. Κάθε εργάτρια μέλισσα αποδίδεται σε μια αρχική λύση για να την εκμεταλλευτεί και να αναζητεί άλλες λύσεις στην γειτονική περιοχή τους. Έτσι, πραγματοποιείται η διαδικασία που υποδηλώνεται ως *Replace2 - 1* (βλ. Αλγόριθμος 3) και στόχος είναι η εντατικοποίηση της αναζήτησης. Στο πλαίσιο της διαδικασίας αυτής, οι κόμβοι που έχουν ήδη συμπεριληφθεί σε μια λύση αντικαθίσταται από άλλους που δεν έχουν χρησιμοποιηθεί και βελτιώνουν την ποιότητα της λύσης. Συγκεκριμένα, σε πρώτη φάση, ταξινομούνται οι μη χρησιμοποιημένοι κόμβοι κατά φθίνουσα σειρά, σύμφωνα με το σκορ τους ( $s_i$ ). Στη συνέχεια, για κάθε υπο-διαδρομή στη λύση, εντοπίζονται οι δύο κόμβοι με το μικρότερο σκορ και το αθροιστικό σκορ τους συγκρίνεται διαδοχικά με το σκορ των κόμβων που δεν έχουν χρησιμοποιηθεί. Σε περίπτωση που κάποιος μη χρησιμοποιημένος κόμβος έχει μεγαλύτερο σκορ από το αθροιστικό τους, οι δύο κόμβοι αφαιρούνται από το διάλυμα λύσης. Ενώ, ο κόμβος αυτός θα εισαχθεί στην εξεταζόμενη υπο-διαδρομή, στη βέλτιστη θέση, σύμφωνα με την διαδικασία της πλησιέστερης εισαγωγής (*Nearest Insertion*). Για να εντοπιστεί η βέλτιστη θέση, υπολογίζεται το κόστος όλων των δυνατών θέσεων, δηλαδή των θέσεων μεταξύ όλων των

διαδοχικών κόμβων και επιλέγετε η οικονομικότερη ενδιάμεση τοποθέτηση.

---

**Αλγόριθμος 3** Εργάτιες μέλισσες: *Replace 2 – 1*


---

```

1: Εισαγωγή:  $V, M, T_{max}, s_i, t_{ij}, z_{ij}$ 
2:  $NU := \{V\} - \{z_{ij}\}$ 
3: Ταξινόμηση κόμβων στο  $NU$  βάση του  $s_i$ 
4: for  $\forall n \in NU$ 
5:   for  $z_{id} \in z_{ij} \mid z_{id}$ : υπο-διαδρομή,  $d = 1, \dots, M$ 
6:     Εντοπισμός  $k1, k2$  κόμβων:  $\min s_i$ 
7:     if  $s_n > s_{k1} + s_{k2}$ 
8:        $z'_{id} \leftarrow z_{id}$ 
9:       Αφαίρεση  $k1, k2$  από  $z'_{id}$ 
10:      Εύρεση βέλτιστου  $j$  για εισαγωγή  $n$  στο  $z'_{id}$ 
11:       $z'_{ij} \leftarrow node$ 
12:      if  $Total(z'_{id}) \leq T_{max}$ 
13:         $z_{id} \leftarrow z'_{id}$ 
14:      Ενημέρωση  $Score, Total$  υπο-διαδρομής
15:    break
16:  end if
17: end if
18: end for
19: end for
20: Έξοδος:  $z_{ij}$ 

```

---

Παρόλα αυτά, η λύση έχει ακόμα περιθώρια βελτίωσης, αφού είναι πιθανό, μετά την αντικατάσταση δυο κόμβων από έναν, να υπάρχει χώρος (σε όρους απόστασης) για την εισαγωγή επιπλέον κόμβων. Έτσι εφαρμόζεται μία απλή ευρετική διαδικασία, που υποδηλώνεται ως *Fill – up* (βλ. Αλγόριθμος 4). Πρόκειται για μία τεχνική συμπλήρωσης του διανύσματος λύσης, αφού η διαδικασία προσθέτει κόμβους, για όσο όμως, η συνολική διανυθείσα απόσταση της κάθε υπο-διαδρομής δεν υπερβαίνει το όριο του  $T_{max}$ . Τα βασικά χαρακτηριστικά της είναι ότι οι επιπλέον κόμβοι που προστίθενται δεν υπάρχουν ήδη στο διάνυσμα, διασφαλίζοντας έτσι την εφικτότητα της λύσης και ότι οι κόμβοι που διαδοχικά είναι υποψήφιοι για εισαγωγή, εξετάζονται με προτεραιότητα βάση του σκορ τους, ενισχύοντας έτσι την ποιότητα της λύσης.

**Αλγόριθμος 4** *Fill – up*


---

```

1: Εισαγωγή:  $V, M, T_{max}, s_i, t_{ij}, z_{ij}$ 
2:  $NU := \{V\} - \{z_{ij}\}$ 
3: Ταξινόμηση κόμβων στο  $NU$  βάση του  $s_i$ 
4: for  $\forall n \in NU$ 
5:   for  $\forall j \in z_{ij}$ 
6:      $Savings_n = t_{j-1,n} + t_{n,j} - t_{j-1,j}$ 
7:   end for
8:   Εύρεση  $j$ :  $\min Savings_n$ 
9:   Εύρεση  $d$ :  $j \in z_{id} \subset z_{ij}$ 
10:  if  $Total(z_{id}) + Savings_{node} \leq T_{max}$ 
11:     $z_{ij} \leftarrow node$  | μεταξύ των θέσεων  $(j - 1, j)$ 
12:  Ενημέρωση  $Score, Total$  υπο-διαδρομής
13:  end if
14: end for
15: Έξοδος:  $z_{ij}$ 

```

---

**Στάδιο των θεατών μελισσών**

Το πιο κρίσιμο μέρος του προτεινόμενου αλγορίθμου DRABC είναι το στάδιο των θεατών μελισσών, κατά το οποίο χρησιμοποιείται μια διαδικασία για να πραγματοποιηθεί η εκμετάλλευση επιλεγμένων λύσεων. Αρχικά, κατά την προσπάθεια εντοπισμού γειτονικών λύσεων, οι θεατές μέλισσες επιλέγουν μία από τις ήδη υπάρχουσες λύσεις βάση πιθανότητας και συναρτήσεως της αντίστοιχης τιμής της αντικειμενικής συνάρτησης, (βλ. Εξίσωση 3.1). Στη συνέχεια, για κάθε επιλεγμένη λύση, ακολουθεί η διαδικασία *SolutionConversion* (βλ. Αλγόριθμος 5). Στη διαδικασία αυτή, ενσωματώνεται η πρωτότυπη μέθοδος κωδικοποίησης της λύσης DR, όπως και η τεχνική αποκωδικοποίησης του διανύσματος λύσης. Αναλύοντας την διαδικασία *SolutionConversion*, πρώτο ρόλο έχει η μετατροπή του διανύσματος λύσης, από αναπαράσταση κόμβων σε αναπαράσταση με συνεχείς τιμές, σύμφωνα με τις δοθέντες αποστάσεις μεταξύ των κόμβων. Όπως έχει ήδη αναφερθεί η τεχνική αυτή αποτελεί το βασικό στοιχείο του μοντέλου DR. Έτσι κάθε τιμή του τροποποιημένου διανύσματος αφορά στις μονάδες απόστασης μεταξύ διαδοχικών κόμβων, σύμφωνα με την αλληλουχία κόμβων που ορίζει η αντίστοιχη λύση. Στη συνέχεια, εφαρμόζεται η εξίσωση κίνησης (βλ. Εξίσωση 3.2) και δημιουργείται ένα νέο διάνυσμα λύσης με συνεχείς τιμές. Σε αυτό το σημείο, λαμβάνει χώρα μία απλή ευρετική μέθοδος για την μετατροπή του διανύσματος λύσης, από συνεχείς τιμές, ξανά σε αναπαράσταση κόμβων. Η λύση μετατρέπεται βηματικά, αφού ξεκινώ-



ντας από τον κόμβο 1, προστίθενται κόμβοι των οποίων η απόσταση από κάθε προηγούμενο προσεγγίζει τις αντίστοιχες συνεχείς τιμές του νέου διανύσματος. Κατά αυτό τον τρόπο δημιουργούνται εφικτές υπο-διαδρομές, σύμφωνα με τα δεδομένα του προβλήματος, που ξεκινάν από τον κόμβο 1 και τερματίζουν στον κόμβο  $N$ , χωρίς να παραβιάζεται η συνολική απόσταση  $T_{max}$ . Για να διευκολυνθεί η ολοκλήρωση της υπο-διαδρομής χωρίς να παραβιάζεται ο περιορισμός, χρησιμοποιήσουμε μια νέα τιμή μικρότερη από την τιμή  $T_{max}$ , την  $T_{low}$  που υπολογίζεται με την ακόλουθη εξίσωση:

$$T_{low} = a_{low} * (T_{max} - \frac{\sum_{k=1}^{N-1} t_{k,1}}{N-1}), \quad a_{low} \in (0, 1) \quad (4.1)$$

Ουσιαστικά, η τιμή  $T_{low}$  είναι το αναμορφωμένο άνω όριο του περιορισμού απόστασης και περιλαμβάνει την μέση απόσταση όλων των κόμβων από τον τελευταίο ( $N$ ), διευκολύνοντας έτσι την ολοκλήρωση κάθε υπο-διαδρομής. Αν κατά τη διάρκεια τοποθέτησης των κόμβων, η συνολική διανυθίσα απόσταση ξεπεράσει την τιμή  $T_{low}$ , εξετάζεται ο τερματισμός της διαδρομής με τον τελευταίο κόμβο χωρίς την προσθήκη άλλου. Εάν το κατασκευασμένο μέρος της λύσης είναι εφικτό και δεν έχουν ολοκληρωθεί όλες οι υπο-διαδρομές  $M$ , ξεκινάει η κατασκευή της διαδοχικής υπο-διαδρομής. Διαφορετικά, ακολουθούνται δύο εναλλακτικές στρατηγικές, πρώτα εξετάζεται η ολοκλήρωση της υπο-διαδρομής, χωρίς την προσθήκη επιπλέον κόμβου πέραν του  $N$ . Αν πάλι η περίπτωση αυτή δεν είναι εφικτή, ο τελευταίος κόμβος πριν από το  $N$  πρέπει να αντικατασταθεί, από έναν αχρησιμοποίητο κόμβο που βρίσκεται πλησιέστερα σε αυτόν. Η στρατηγική αυτή αποτελεί έναν πλησιέστερο γείτονα με φορά από τον κόμβο τερματισμού προς τον αρχικό κόμβο 1, ακολουθώντας διαδοχικές αντικαταστάσεις κόμβων και ελέγχοντας την εφικτότητα της υπο-διαδρομής. Με την ολοκλήρωση της περιγραφόμενης διαδικασίας, καταλήγουμε σε μία εφικτή λύση, η οποία βελτιώνεται περαιτέρω με διαδικασίες τοπικής αναζήτησης. Συγκεκριμένα, χρησιμοποιείται η διαδικασία *Replace1-1* (βλ. Αλγόριθμος 6), παρόμοια με την *Replace2-1*, με την διαφορά ότι αντικαθίσταται μόνο ένας κόμβος αντί για δύο. Στη συνέχεια, χρησιμοποιείται η τεχνική *Fill-up* και τέλος, η λύση αντιστοιχείται σε μία θεατή μέλισσα.

### Στάδιο των ανιχνεύτριων μελισσών

Στον παρουσιαζόμενο αλγόριθμο, όπως και στον κλασικό αλγόριθμο ABC, οι εργάτριες και οι θεατές μέλισσες αντιπροσωπεύουν την εκμετάλλευση των λύσεων, ενώ την διαδικασία εξερεύνησης των λύσεων αναλαμβάνουν οι ανιχνεύτριες μέλισσες. Συνήθως, όταν μια λύση δεν έχει βελτιωθεί για ένα προκαθορισμένο

**Αλγόριθμος 5** *SolutionConversion*


---

```

1:
2: Εισαγωγή:  $V, M, T_{max}, s_i, t_{ij}, z_{ij}, T_{low}$ 
3:  $ztmp_{ij} \leftarrow t_{z_{i,j-1}, z_{i,j}}$ 
4:  $vtmp_{ij} \leftarrow ztmp_{ij} + rand[-1, 1](ztmp_{ij} - ztmp_{kj}), k \neq i$ 
5:  $NU := \{V\} - \{1\} - \{N\}$ 
6: while  $d < M$ 
7:  $v_{i,1} \leftarrow 1$ 
8: for  $j = 2 \rightarrow length(vtmp_{ij})$ 
9: Εύρεση  $n \in NU : \mu n \mid t_{n, v_{i,j-1}} - vtmp_{i,j} \mid, \forall n$ 
10: if  $Total(v_{id}) + t_{n, v_{i,j-1}} \leq T_{low}$ 
11:  $v_{i,j} \leftarrow n$ 
12: Αφαίρεση  $n$  από  $NU$ 
13: Ενημέρωση  $Score, Total$  υπο-διαδρομής,  $v_{id}$ 
14: else
15: if  $Total(v_{id}) + t_{n, v_{i,j-1}} + t_{n, N} \leq T_{max}$ 
16:  $v_{i,j} \leftarrow n \quad v_{i,j+1} \leftarrow N$ 
17: Αφαίρεση  $n$  από  $NU$ 
18: else
19:  $v_{i,j} \leftarrow N$ 
20: if  $Total(v_{id}) + t_{v_{i,j-1}, N} > T_{max}$ 
21: repeat
22: Εύρεση  $n' \neq n$ 
23:  $v_{i,j-1} \leftarrow n'$ 
24: Ενημέρωση  $Total$  υπο-διαδρομής,  $v_{id}$ 
25: until  $Total(v_{id}) \leq T_{max}$ 
26: end if
27: end if
28: Ενημέρωση  $Score, Total$  υπο-διαδρομής,  $v_{id}$ 
29:  $d \leftarrow d + 1$ 
30: end if
31: end for
32: end while
33: Έξοδος:  $v_{ij}$ 

```

---

αριθμό επαναλήψεων (*limit*) οι μέλισσες αντικαθιστούν τη λύση αυτή με μία τυχαία, ώστε να επιτευχθεί βαθιά εξερεύνηση. Η ποιότητα αυτής της νέας τυχαίας λύσης είναι αμφιλεγόμενη, αφού ο αλγόριθμος μπορεί να ξεπεράσει ένα τοπικό βέλτιστο ή μπορεί να οδηγηθεί σε χειρότερης ποιότητας λύση. Για

**Αλγόριθμος 6** Εργάτιες μέλισσες: *Replace 1 – 1*


---

```

1: Εισαγωγή:  $V, M, T_{max}, s_i, t_{ij}, v_{ij}$ 
2:  $NU := \{V\} - \{v_{ij}\}$ 
3: Ταξινόμηση κόμβων στο  $NU$  βάση του  $s_i$ 
4: for  $\forall n \in NU$ 
5:   for  $\forall j \in v_{ij}$ 
6:     if  $s_n > s_{v_{ij}}$ 
7:        $v'_{ij} \leftarrow n$ 
8:       Έρευνα  $d : j \in v_{id} \subset v_{ij}$ 
9:       if  $Total(v'_{id}) \leq T_{max}$ 
10:         $v_{id} \leftarrow v'_{id}$ 
11:      Ενημέρωση  $Score, Total$  υπο-διαδρομής
12:    break
13:  end if
14: end if
15: end for
16: end for
17: Έξοδος:  $v_{ij}$ 

```

---

να αποφευχθεί αυτό, ο αλγόριθμος DRABC στο στάδιο των ανιχνεύτριων μελισσών, χρησιμοποιεί μια ευρετική διαδικασία που δηλώνεται ως *Reassemble* (βλ. Αλγόριθμος 7, για να αλλαχθεί η αλληλουχία των κόμβων στο διάγραμμα λύσης. Η μέθοδος αποτελείται από δύο κύριες φάσεις. Στη πρώτη φάση, εφαρμόζεται η διαδικασία πλησιέστερης εισαγωγής *NearestInsertion*, με τη διαφορά ότι το άνω όριο του περιορισμού της απόστασης αλλάζει σε  $T_{limit}$ . Όπου,  $T_{limit} = a_{limit} * T_{max}$ , και  $a_{limit} > 1$ . Έτσι, η τιμή  $T_{limit}$  εκφράζεται ως πολλαπλάσιο της τιμής του  $T_{max}$  και οι λύσεις καθίστανται ανέφικτες. Στη συνέχεια, στη δεύτερη φάση, η λύση πρέπει να ξαναγίνει εφικτή, με μία αντίστροφη διαδικασία της πλησιέστερης εισαγωγής. Κάθε υπο-διαδρομή εξετάζεται διαδοχικά και μέχρι η συνολική απόσταση της καθεμίας να είναι εφικτή, σύμφωνα με τον περιορισμό. Η μέθοδος αναζητά επαναληπτικά για κάθε υπο-διαδρομή, τον κόμβο που όταν αφαιρεθεί από αυτή, θα έχει ως αποτέλεσμα την μέγιστη μείωση της συνολικής απόστασης. Μετά από το συνδυασμό αυτών των δύο διαδικασιών, η νέα λύση που προκύπτει είναι διαφορετική από την αρχική της και καλής ποιότητας.

**Αλγόριθμος 7** *Reassemble*


---

```

1: Εισαγωγή:  $V, M, T_{max}, s_i, t_{ij}, v_{ij}, T_{limit}$ 
2:  $NU := \{V\} - \{v_{ij}\}$ 
3: Ταξινόμηση κόμβων στο  $NU$  βάση του  $s_i$ 
4: for  $\forall n \in NU$ 
5: for  $\forall j \in v_{ij}$ 
6:  $Savings_n = t_{j-1,n} + t_{n,j} - t_{j-1,j}$ 
7: end for
8: Έρευση  $j$ : μιν  $Savings_n$ 
9: Έρευση  $d$ :  $j \in v_{id} \subset v_{ij}$ 
10: if  $Total(v_{id}) + Savings_n \leq T_{limit}$ 
11:  $Total(v_{id}) \leftarrow Total(v_{id}) + Savings_n$ 
12:  $Score(v_{id}) \leftarrow Score(v_{id}) + s_n$ 
13:  $v_{i,j} \leftarrow n$  | μεταξύ των θέσεων  $(j - 1, j)$ 
14: end if
15: end for
16: for  $\forall d \in M$ 
17: while  $Total(v_{id}) > T_{max}$ 
18: for  $\forall j \in v_{id}$ 
19:  $a_j = t_{j-1,j+1} - t_{j-1,j} - t_{j,j+1}$ 
20: end for
21: Έρευση  $j$ : μιν  $a_j$ 
22: Αφαίρεση  $j$  φρομ  $v_{id}$ 
23: Ενημέρωση  $Score, Total$  υπο-διαδρομής
24: end while
25: end for
26:  $z_{ij} \leftarrow v_{ij}$ 
27: Έξοδος:  $z_{ij}$ 

```

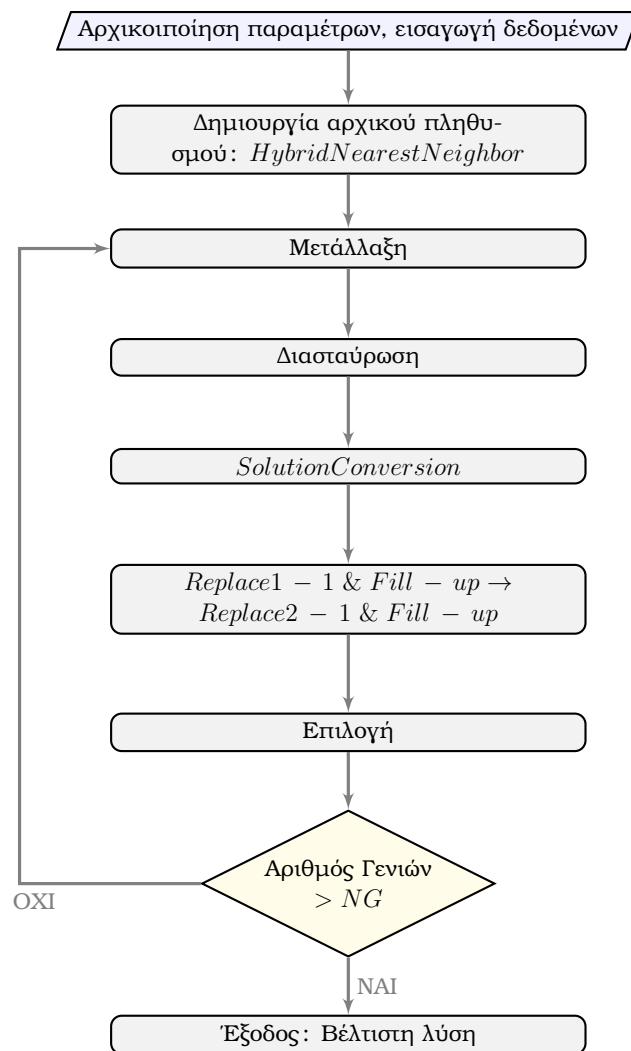
---

**4.1.2 Αλγόριθμος Distance Related Differential Evolution**

Ο δεύτερος αλγόριθμος που προτείνεται είναι ο υβριδικός αλγόριθμος της διαφορικής εξέλιξης, και αυτός έχει εφαρμοστεί μέσα στο πλαίσιο του μοντέλου Distance Related, ο οποίος θα αναφέρεται ως DRDE. Η αλγοριθμική προσέγγιση που παρουσιάζεται αποτελείται από τις διαδικασίες: του υβριδικού πλησιέστερου γείτονα, *HybridNearestNeighbor* για την δημιουργία του αρχικού πληθυσμού, από τον ευρετικό αλγόριθμο αποκωδικοποίησης, *SolutionConversion* και από τις διαδικασίες τοπικής αναζήτησης: *Replace2 - 1*, *Replace1 - 1* και *Fill - up*. Η αλγοριθμική διαδικασία του DRDR περιγράφεται αναλυτικά στη

συνέχεια (βλ. Αλγόριθμος 8) και παρουσιάζεται στο παρακάτω διάγραμμα ροής, Σχήμα 4.2. Η περιγραφή του αλγορίθμου συνδυάζεται με την επίλυση του προβλήματος προσανατολισμού ομάδας με χωρητικότητα Capacitated Team Orienteering Problem (CTOP).

Σχήμα 4.2: Διάγραμμα ροής του αλγορίθμου DRDE



### Αναπαράσταση λύσης και δημιουργία αρχικού πληθυσμού

Αντίστοιχα με το πρόβλημα προσανατολισμού ομάδας, για την επίλυση του προβλήματος προσανατολισμού ομάδας με χωρητικότητα, CTOP, απαιτούνται:

**Αλγόριθμος 8** DRDE

---

```

1: Αρχικοποίηση παραμέτρων, εισαγωγή δεδομένων
2: Δημιουργία αρχικών λύσεων  $x_{ij}$ ,  $t=1, \dots, NP$ ,  $\theta=1, \dots, N + (M - 1) * 2$  //
   βλ. Αλγόριθμος 2
3: Εκτίμηση  $F_i$  // βλ. Εξίσωση 2.9
4: for  $l = 1 : NG$ 
5: for  $i = 1 : NP$ 
6: Εφαρμογή κωδικοποίησης DR στο διάνυσμα  $x_{ij}$ 
7: Έυρεση:  $x_{i1j} \neq x_{i2j} \neq x_{i3j} \neq x_{ij}$ 
8: Εφαρμογή Εξίσωσης μετάλλαξης 3.4 για κατασκευή  $v_{ij}$ 
9: Εφαρμογή Εξίσωσης διασταύρωσης 3.5 για κατασκευή  $u_{ij}$ 
10: Χρήση SolutionConversion για αποκωδικοποίηση  $u_{ij}$  // βλ. Αλγόριθμος
    5
11: Χρήση Replace 1 - 1 και Fill - up για  $u_{ij}$  // βλ. Αλγόριθμος 6 και 4
12: Χρήση Replace 2 - 1 και Fill - up για  $u_{ij}$  // βλ. Αλγόριθμος 3 και 4
13: Διαδικασία επιλογής // βλ. Εξίσωση 3.6
14: if  $F_{u_{ij}} > F_{x_{ij}}$ 
15:  $x_{ij} \leftarrow u_{ij}$ 
16: end if
17: end for
18:  $best_{x_{ij}} \leftarrow i : max F_i$ 
19: end for

```

---

το σειτ κόμβων  $V = \{1, \dots, N\}$ , το κέρδος  $p_i$  κάθε κόμβου  $i$ , ο αριθμός των υπο-διαδρομών  $M$ , το προκαθορισμένο όριο ταξιδιού  $T_{max}$  και η ευκλείδεια απόσταση  $t_{ij}$  μεταξύ όλων των κόμβων. Σε αυτό το πρόβλημα, απαιτείται επιπλέον και το προκαθορισμένο όριο χωρητικότητας  $Q_{max}$  και η τιμή ζήτησης  $D_i$  κάθε κόμβου  $i$ . Μία εφικτή λύση του CTOP αποτελείται από  $M$  υπο-διαδρομές, οι οποίες ξεκινάνε από τον κόμβο 1 και τερματίζουν σε αυτόν, έτσι η λύση αναπαρίσταται από ένα διάνυσμα ακέραιων τιμών, συμβολίζοντας την αλληλουχία των κόμβων στη διαδρομή. Το μέγιστο μήκος του διανύσματος είναι  $N + (M - 1) * 2$ . Για την λειτουργία του αλγορίθμου, ορίζονται το μέγεθος του πληθυσμού  $NP$  και ο μέγιστος αριθμός επαναλήψεων (γενιών) του αλγορίθμου  $NG$ . Μετά την αρχικοποίηση, παράγεται ο αρχικός πληθυσμός λύσεων, σύμφωνα με τον υβριδικό ευρετικό αλγόριθμο του πλησιέστερου γείτονα (*HybridNearestNeighbor*). Ο αλγόριθμος για την κατασκευή των αρχικών λύσεων, εκτελείται όπως έχει αναφερθεί στην περίπτωση του TOP, με την διαφορά ότι ο κόμβος  $N$  δεν χρησιμοποιείται σαν κόμβος τερματισμού και ότι στον περιορισμό της εφικτότητας προστίθεται και η συνολική χωρητικότητα

τα, έτσι ο περιορισμός τροποποιείται ως εξής (βλ. Αλγόριθμος 2, Γραμμή 9):  
 $if : Total + t_{z_{i,j-1},n} + t_{n,1} \leq T_{max} \&\& Demand + D_n \leq Q_{max}.$

### Μετάλλαξη και Διασταύρωση του Πληθυσμού

Στη συνέχεια εφαρμόζουμε τις διαδικασίες μετάλλαξης και διασταύρωσης στον πληθυσμό. Βασική προϋπόθεση για την εφαρμογή της εξίσωσης μετάλλαξης (βλ. Εξίσωση 3.4), είναι η ύπαρξη διανύσματος συνεχών τιμών και σε αυτό το σημείο εφαρμόζεται το μοντέλο Distance Related. Συνεπώς, η κωδικοποίηση των διανυσμάτων λύσης αλλάζει, από αναπαράσταση κόμβων σε συνεχείς τιμές, οι οποίες αντιπροσωπεύουν τις ευκλείδειες αποστάσεις μεταξύ διαδοχικών κόμβων. Με το πέρας της διαδικασίας, για κάθε λύση του πληθυσμού, δηλαδή για κάθε διάνυσμα στόχου  $x_{ij}$ , έχει σχηματιστεί ένα μεταλλαγμένο διάνυσμα  $v_{ij}$ . Ακολουθεί η διαδικασία διασταύρωσης, όπως ορίζεται από την Εξίσωση (3.5). Συγκεκριμένα εφαρμόζεται ο διωνυμικός τελεστής διασταύρωσης και για κάθε διάνυσμα στόχου παράγεται δοκιμαστικό διάνυσμα  $u_{ij}$ . Το δοκιμαστικό διάνυσμα αποτελεί ένα συνδυασμό των τιμών του στόχου και του μεταλλαγμένου διανύσματος. Από αυτό το σημείο και έπειτα, η κωδικοποίηση της λύσης πρέπει να αλλάξει ξανά, από συνεχείς τιμές σε αναπαράσταση κόμβου, ώστε να είναι δυνατός ο υπολογισμός της τιμής της αντικειμενικής συνάρτησης, που απαιτείται για την επόμενη βασική διαδικασία, της επιλογής. Έτσι, χρησιμοποιείται ο ευρετικός αλγόριθμος αποκωδικοποίησης της λύσης *SolutionConversion* που χρησιμοποιήθηκε και στην περίπτωση του TOP (βλ. Αλγόριθμος 5). Όμως, κάθε φορά που γίνεται έλεγχος της εφικτότητας που αφορά στη συνολική απόσταση, ελέγχεται και η συνολική χωρητικότητα, ενώ ο κόμβος  $N$  δεν επηρεάζει την δομή της λύσης, αφού ο κόμβος 1 είναι κόμβος έναρξης και τερματισμού, όπως αναφέρεται και παραπάνω.

### Διαδικασίες Βελτίωσης του Πληθυσμού και Επιλογή

Μετά την ολοκλήρωση της επαναφοράς των διανυσμάτων λύσης σε αναπαράσταση κόμβων, έχει δημιουργηθεί ένας νέος πληθυσμός με εφικτές λύσεις. Για να ενισχυθεί η αποτελεσματικότητα της μεθόδου DRDE εφαρμόζονται στο πληθυσμό αυτόν, οι διαδικασίες τοπικής αναζήτησης που έχουν αναλυθεί στο Κεφάλαιο 4.1.1. Πρόκειται για τις: *Replace2-1*, *Replace1-1* και *Fill-up* (βλ Αλγόριθμος 6,3 και 4). Είναι πλέον σαφές, ότι σε όλους τους απλούς αυτούς αλγόριθμους, η συνθήκη εφικτότητας είναι τροποποιημένη, ώστε να συνυπολογίζεται και η συνολική χωρητικότητα. Οι διαδικασίες αυτές χρησιμοποιήθηκαν με τον ίδιο τρόπο, όπως στην περίπτωση του TOP, αφού έχει ίδια αντικειμε-

νική συνάρτηση με το CTOP, δηλαδή την μεγιστοποίηση του κέρδους. Για να ολοκληρωθεί μία γενιά, απομένει η διαδικασία της επιλογής (βλ. Εξίσωση 3.6), πλέον οι νέες ενισχυμένες λύσεις, συγκρίνονται σε ποιότητα με τις αρχικές (διανύσματα στόχου) που τους αντιστοιχούν, και επικρατεί η λύση με το υψηλότερο κέρδος. Έτσι, το μέγεθος του πληθυσμού διατηρείται σταθερό από γενιά σε γενιά, ενώ η ποιότητα του πληθυσμού βελτιώνεται.



## **Κεφάλαιο 5**

# **Η N-ατόμων Μάχη των Φύλων στην Επίλυση του Προβλήματος Σχεδιασμού Τουριστικών Διαδρομών**

### **5.1 Εισαγωγή στη Θεωρία Παιγνίων**

Η θεωρία παιγνίων παρέχει τη μαθηματική βάση για την ανάλυση διαδραστικών διαδικασιών λήψης αποφάσεων ανάμεσα σε ορθολογικά άτομα τα οποία είναι αμοιβαία αλληλεξαρτώμενα και μάλιστα με αντικρουόμενα συμφέροντα. Ένα παίγνιο περιλαμβάνει τρία σύνολα: α) το σύνολο των παικτών, β) το σύνολο των δράσεων, γ) το σύνολο των κερδών. Οι παίκτες είναι αυτοί που λαμβάνουν τις αποφάσεις στο εκάστοτε υπό μελέτη πρόβλημα. Οι παίκτες μπορεί να είναι ένα πρόσωπο, μία οργάνωση, ένα κράτος, ή ένας συνασπισμός. Η θεωρία παιγνίων μπορεί να εφαρμοστεί σε διάφορα προβλήματα πολιτικής, ψυχολογικής, κοινωνικής και οικονομικής μορφής. Γενικά, όταν μελετάμε ένα παίγνιο, υποθέτουμε ότι κάθε παίκτης δεν επιλέγει απαραίτητα μία μόνο από τις διαθέσιμες ενέργειές του, αλλά είναι ελεύθερος να επιλέξει οποιαδήποτε κατανομή πιθανότητας στο σύνολο των ενεργειών του. Μία τέτοια κατανομή πιθανότητας ονομάζεται στρατηγική. Η ειδική περίπτωση στρατηγικής που θέτει πιθανότητα 1 σε μία μόνο ενέργεια ονομάζεται καθαρή στρατηγική. Η σπουδαιότερη και περισσότερο αναγνωρισμένη έννοια λύσης στην θεωρία παιγνίων είναι η ισορροπία Nash [82]. Πρόκειται για έναν συνδυασμό στρατηγικών, μία για κάθε παίκτη, όπου δεν υπάρχει παίκτης που να μπορεί να αυξήσει την αναμενόμενη ωφέλεια του αν αλλάξει την στρατηγική του. Με άλλα λόγια σε μία ισορροπία Nash [83] κάθε παίκτης επιλέγει την στρατηγική που μεγιστοποιεί την αναμενόμενη ωφέλεια του, δοθέντος των στρατηγικών των άλλων παικτών. Αυτή η σημαντική έννοια ισορροπίας προτάθηκε από τον Nash [83] το 1951, ο οποίος

μάλιστα απέδειξε ότι κάθε παίγνιο (με πεπερασμένο πλήθος παικτών) έχει μία τέτοια ισορροπία. Ένα από τα πιο διάσημα προβλήματα της θεωρίας παιγνίων που χρησιμοποιήθηκε στην συγκεκριμένη διατριβή είναι το παίγνιο της Μάχης των Φύλων (Battle Of Sexes) όπου αναλύεται εκτενέστερα παρακάτω.

## 5.2 Η Μάχη των Φύλων (Battle Of Sexes)

Ένα παιχνίδι Battle Of Sexes (BOS) δύο ατόμων παρουσιάζεται συνήθως με την παρακάτω ιστορία: Ένας άντρας και η σύζυγός του θέλουν να περάσουν ένα βράδυ μαζί, ώστε να προσπαθήσουν να επιλέξουν μια ψυχαγωγία κοινού ενδιαφέροντος. Ο σύζυγος (Παίκτης 1) προτιμά ένα ποδοσφαιρικό παιχνίδι, ενώ η σύζυγος (Παίκτης 2) προτιμά μια παράσταση μπαλέτου. Ωστόσο, και οι δύο προτιμούν να πάνε μαζί. Το παιχνίδι BOS δύο ατόμων είναι ένα παιχνίδι συντονισμού με αντικρουόμενες προτιμήσεις. Υπάρχουν δύο επιλογές για τους παίκτες ενώ οι προτιμώμενες επιλογές των δύο παικτών συγκρούονται μεταξύ τους. Ανάλογα με τον συνδυασμό των αποφάσεων των δύο παικτών, οι παίκτες λαμβάνουν αποδόσεις σύμφωνα με τον πίνακα κέρδους που φαίνεται στον Πίνακα 5.1. Οι στήλες του πίνακα αντιστοιχούν στις αποφάσεις του παίκτη 2 και οι σειρές αντιστοιχούν στην απόφαση του παίκτη 1. Η πρώτη είσοδος στην παρένθεση κάθε κελιού αντιπροσωπεύει την απόδοση του παίκτη 1 και η δεύτερη καταχώρηση αντιπροσωπεύει την αντίστοιχη πληρωμή του παίκτη 2.

Πίνακας 5.1: Η Μάχη των Φύλων (Πίνακας Κέρδους)

		Παίκτης 2 (προτίμηση Μπαλέτο)	
		Ποδόσφαιρο	Μπαλέτο
Παίκτης 1 (προτίμηση Ποδόσφαιρο)	Ποδόσφαιρο	2,1	0,0
	Μπαλέτο	0,0	1,2

Με βάση τον πίνακα αποδοχών που παρουσιάζεται στον Πίνακα 5.1, οι κύριες ιδιότητες ενός παιχνιδιού BOS δύο ατόμων συνοψίζονται ως εξής:

- Ένας παίκτης έχει δύο επιλογές (ποδόσφαιρο ή μπαλέτο), η οποία είναι δυαδική επιλογή κατά την έννοια του Schelling [84].
- Οι προτιμήσεις των δύο παικτών είναι αντίθετες.

- Ένας παίκτης θα λάβει υψηλότερο κέρδος όταν και οι δύο παίκτες έχουν την ίδια επιλογή από ό, τι στην περίπτωση διαφορετικών επιλογών.
- Ένας παίκτης θα λάβει υψηλότερο κέρδος εάν η κοινή επιλογή είναι αυτό που του αρέσει, παρά εάν η επιλογή είναι αυτό που δεν του αρέσει.

### 5.3 Η Ν-ατόμων Μάχη των Φύλων (N-person Battle Of Sexes)

Ένα παιχνίδι N-person BOS [85] έχει παρόμοιες ιδιότητες με ένα BOS δύο ατόμων με τη μόνη διαφορά ότι ο αριθμός των παικτών είναι μεγαλύτερος. Ωστόσο, πρέπει να διατηρήσει τη μεγάλη σύγκρουση κάθε παίκτη μεταξύ της προτίμησής του και της βούλησης να ακολουθήσουν όσο το δυνατόν περισσότερους άλλους παίκτες με κοινή επιλογή. Υπάρχουν, ωστόσο, ορισμένες δυσκολίες στην επέκταση των κύριων ιδιοτήτων των παιχνιδιών BOS 2 ατόμων, απευθείας σε περιπτώσεις Ν-ατόμων χωρίς ορισμένες τροποποιήσεις. Πριν ορίσει το N-person BOS, πρέπει να εισαχθούν ορισμένες επεξηγήσεις. Ο συνολικός αριθμός των παικτών είναι  $n$ , ο αριθμός των παικτών που τους αρέσει το ποδόσφαιρο είναι  $F$ , οπότε υπάρχουν παίκτες  $n - F$  που τους αρέσει το μπαλέτο. Ο αριθμός των συνεργαζόμενων παικτών (κοινή επιλογή ανεξαρτήτως προτίμησης)  $y$ , και ο αριθμός των παικτών που δεν συνεργάζονται είναι  $n - y$ . Ο αριθμός των παικτών που επιλέγουν το ποδόσφαιρο είναι  $x$ , οπότε ο αριθμός των παικτών που επιλέγουν μπαλέτο είναι  $n-x$ .

Στο N-person BOS, όταν ο αριθμός των παικτών μεγαλώνει, ο αριθμός των επιλογών (ή των αποφάσεων) αυξάνεται εκθετικά. Σε ένα παιχνίδι n-person BOS, το  $F$  δηλώνει τον αριθμό των παικτών που τους αρέσει το ποδόσφαιρο, σαφώς  $1 \leq F \leq n - 1$  που δηλώνει ότι για κάθε παίκτη υπάρχει τουλάχιστον ένας με αντίθετη προτίμηση. Υπάρχουν  $F + 1$  πιθανές καταστάσεις (ή συνδυασμών αποφάσεων) που αντανακλώνται από τον αριθμό των συνεργατών, ο οποίος μπορεί να είναι  $0, 1, 2, \dots, F$ . Στον αντίποδα, υπάρχουν  $n-F$  παίκτες με  $n-F+1$  πιθανές καταστάσεις που αντανακλώνται από τον αριθμό των συνεργατών. Ως εκ τούτου, σε παιχνίδια n-person BOS υπάρχουν συνολικά οι παρακάτω πιθανοί συνδυασμοί:

$$\sum_{F=1}^{n-1} (F + 1)(n - F + 1) = \frac{(n^3 + 6n^2 - n - 6)}{6} = \frac{(n + 6)(n^2 - 1)}{6} \quad (5.1)$$

Έτσι, για την μοντελοποίηση του n-person BOS θα χρησιμοποιηθούν συναρ-

τήσεις για τον υπολογισμό του κέρδους κάθε παίκτη αντί των πινάκων πληρωμής. Ενώ στο n-person Prisoners Dilemma οι συναρτήσεις κέρδους (payoff functions) διαμορφώνονται ως συνάρτηση του αριθμού των συνεργατών δηλαδή βασίζονται σε μια συνάρτηση συνεργασίας και μια συνάρτηση αποστασίας στο n-person BOS η κατάσταση είναι πιο περίπλοκη. Χρησιμοποιώντας τον αριθμό των συνεργατών ως τη μόνη μεταβλητή στη μοντελοποίηση παιχνιδιών n-person BOS, θα υπήρχαν μόνο  $n + 1$  διαφορετικές καταστάσεις που θα έπρεπε να αναλυθούν οι οποίες είναι πολύ λιγότερες από τον συνολικό αριθμό περιπτώσεων που δίνονται στην Εξίσωση 5.1.

Επομένως, είναι λογικότερο να μοντελοποιούνται οι συναρτήσεις κέρδους με βάση τον αριθμό των παικτών που έχουν την ίδια επιλογή και όχι βάση του αριθμού των συνεργατών του παιχνιδιού. Ήδη κατηγοριοποιήσαμε τους παίκτες σε δύο ομάδες. Όλοι οι παίκτες στην μια ομάδα επιλέγουν το ποδόσφαιρο και όλοι οι παίκτες στην άλλη ομάδα επιλέγουν το μπαλέτο. Σε κάθε ομάδα, υπάρχουν δύο τύποι παικτών, εκείνοι που τους αρέσει η επιλογή τους, και εκείνοι που δεν τους αρέσει. Ως εκ τούτου, υπάρχουν τέσσερα είδη παικτών: οι παίκτες που τους αρέσει το ποδόσφαιρο και επιλέγουν το ποδόσφαιρο, οι παίκτες που τους αρέσει το μπαλέτο και επιλέγουν το ποδόσφαιρο, οι παίκτες που τους αρέσει το ποδόσφαιρο και επιλέγουν μπαλέτο, καθώς και οι παίκτες που τους αρέσει το μπαλέτο και επιλέγουν το μπαλέτο. Θα χρησιμοποιήσουμε αντίστοιχα Ff, Bf, Fb, Bb για να αντιπροσωπεύσουμε τα τέσσερα είδη παικτών. Τα πρώτα κεφαλαία γράμματα δείχνουν τις προτιμήσεις των παικτών, ενώ δεύτερα μικρά γράμματα αντιπροσωπεύουν τις αποφάσεις των παικτών. Το F ή το f δείχνει το ποδόσφαιρο ενώ το B ή το b αντιπροσωπεύει το μπαλέτο.

Για τους παίκτες που επιλέγουν ποδόσφαιρο έχουν οριστεί δύο συναρτήσεις. Εάν σε έναν παίκτη αρέσει στο ποδόσφαιρο, τότε το κέρδος του καθορίζεται από την συνάρτηση  $L(x)$  (5.2), διαφορετικά  $DL(x)$  (5.3), όπου το  $x$  είναι ο αριθμός των παικτών που επιλέγουν το ποδόσφαιρο. Και οι δύο συναρτήσεις  $L(x)$  και  $DL(x)$  αυξάνονται γραμμικά σε σχέση με τον αριθμό  $x$ . Ανάλογη ανάλυση ισχύει για την ομάδα που επιλέγει μπαλέτο. Ως εκ τούτου, έχουμε τέσσερις συναρτήσεις κέρδους.

Οι γραμμικές συναρτήσεις κέρδους μπορούν να μοντελοποιηθούν ως εξής:

$$L(x) = ax + b \tag{5.2}$$

$$DL(x) = cx + d, x \geq 1 \tag{5.3}$$

$$L'(x) = a'(n - x) + b' \quad (5.4)$$

$$DL'(x) = c'(n - x) + d' \quad (5.5)$$

Όπου  $x$  είναι ο αριθμός των παικτών που επιλέγουν το ποδόσφαιρο.  $L$  και  $DL$  είναι οι απολαβές των παικτών που επιλέγουν ποδόσφαιρο και  $L'$  (Εξίσωση: 5.4) και  $DL'$  (Εξίσωση: 5.5) είναι οι απολαβές εκείνων που επιλέγουν μπαλέτο. Για λόγους απλότητας, θεωρούμε  $a = a', b = b', c = c'$  και  $d = d'$ . Σύμφωνα με τις ιδιότητες του *BOS* [85] θα πρέπει να πληρούνται οι ακόλουθες προϋποθέσεις:

$$a, c > 0 \quad (5.6)$$

$$d < b \quad (5.7)$$

$$cn + d < an + b \quad (5.8)$$

$$cn + d > a + b. \quad (5.9)$$

#### 5.4 Εφαρμογή του N-person BOS στο Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών Πολλών Ατόμων

Όπως έχει αναφερθεί στο Κεφάλαιο 1, μέσα στα πλαίσια της παρούσας μεταπτυχιακής εργασίας, στόχος είναι επίλυση του προβλήματος σχεδιασμού τουριστικών διαδρομών για πολλά άτομα και λαμβάνοντας υπόψιν τις διαφορετικές προτιμήσεις τους. Πρέπει να σημειωθεί ότι αντίστοιχη προσέγγιση δεν έχει εξεταστεί ξανά, σύμφωνα με όσα γνωρίζουμε. Η ιδιαιτερότητα της προσέγγισης έγκειται στο ότι τα άτομα μίας ομάδας που θέλουν να επιλέξουν μια κοινή διαδρομή μέσα από έναν μεγάλο αριθμό σημείων ενδιαφέροντος, έχοντας διαφορετικές προτιμήσεις, είναι δύσκολο να διαχειριστούν ταυτόχρονα ένα μεγάλο πλήθος πιθανών επιλογών και να είναι παράλληλα όλοι ευχαριστημένοι. Σε αυτό το σημείο, το N-person Battle of Sexes χρησιμοποιείται ώστε να γίνει μια γρήγορη και αποτελεσματική βαθμολόγηση των κόμβων, διευκολύνοντας τη συνεννόηση και τον συντονισμό των μελών μίας ομάδας.

Για παράδειγμα, μία ομάδα ατόμων επισκέπτεται μία τουριστική περιοχή και πρέπει να επιλέξει κάποια σημεία ενδιαφέροντος, όμως η επιλογή ενός σημείου μπορεί να μην ικανοποιεί ένα ή περισσότερα άτομα, ενώ η κοινή επιθυμία τους είναι να παραμείνουν όλοι μαζί κατά την διάρκεια της διαδρομής. Η προσέγγιση του N-person BOS είναι αποτελεσματική γιατί πρόκειται για ένα

παίγνιο συντονισμού ανάμεσα σε πολλά άτομα και απαιτεί μία απλή δυαδική απόφαση από τον καθένα. Αναλυτικότερα, για να επιλυθεί το παίγνιο και να βαθμολογηθούν οι κόμβοι (τα σημεία ενδιαφέροντος) απαιτείται ένα διάνυσμα για κάθε μέλος της ομάδας, που θα περιέχει αποφάσεις  $[0, 1]$ , για κάθε πιθανό κόμβο. Έστω,  $N$  ο αριθμός των μελών της ομάδας και  $P$  ο συνολικός αριθμός των σημείων ενδιαφέροντος, για κάθε άτομο  $i = 1, \dots, N$ , δημιουργείται ένα διάνυσμα μεγέθους  $(1, P)$ . Κάθε στοιχείο  $j = 1, \dots, P$  του διανύσματος έχει την τιμή 1, εάν στο άτομο  $i$  αρέσει το σημείο  $j$  και την τιμή 0, διαφορετικά. Συνολικά, λοιπόν διαμορφώνεται ένας πίνακας μεγέθους  $(N, P)$ , με τις προτιμήσεις όλων των μελών για κάθε σημείο. Ο πίνακας προτιμήσεων είναι το πιο σημαντικό στοιχείο της μεθόδου, αφού απλοποιεί την διαδικασία απόφασης κάθε ατόμου, για παράδειγμα δεν απαιτείται ούτε η βαθμολόγηση κάθε σημείου, ούτε η συνολική ταξινόμηση τους, που αποτελούν πιο πολύπλοκες και χρονοβόρες διαδικασίες.

Συνεχίζοντας, η μοντελοποίηση του N-person BOS απαιτεί και την τελική απόφαση των μελών της ομάδας για κάθε σημείο, δηλαδή, πέρα από την προτίμησή τους, το αν θα το επισκεφθούν τελικά ή όχι. Αλγοριθμικά, αυτό σχετίζεται με έναν δεύτερο πίνακα μεγέθους  $(N, P)$ , με δυαδικές τιμές, τον πίνακα απόφασης. Στο σημείο αυτό, μπορούν να εφαρμοστούν οι Εξισώσεις (5.2-5.5) για όλα τα άτομα  $N$ , όπου  $x$  ο αριθμός των ατόμων που επιλέγουν να επισκεφθούν το σημείο  $j$ , για κάθε  $j \in 1, \dots, P$ . Έτσι δημιουργείται ένας αντίστοιχος πίνακας κέρδους, *Payoff*. Κάθε στήλη  $j$  του πίνακα *Payoff* δείχνει το κέρδος που θα απολάβουν τα μέλη της ομάδας, σχετικά με την προτίμησή και την απόφασή τους ως προς τον αντίστοιχο σημείο  $j$ . Συνεπώς, μπορεί να θεωρηθεί ότι το άθροισμα κάθε στήλης, δείχνει το κέρδος όλης της ομάδας, σε σχέση με το εκάστοτε σημείο ενδιαφέροντος. Με αυτή την απλή λογική, πραγματοποιείται γρήγορα μία συνολική βαθμολόγηση των σημείων. Έπειτα, το διάνυσμα κέρδους ανά σημείο, που θα προκύψει από την παραπάνω διαδικασία, θα χρησιμοποιηθεί ως είσοδο για την επίλυση του προβλήματος NTTDP, σε οποιαδήποτε μοντελοποίηση. Ανάλογα με τα δεδομένα κάθε προβλήματος (βλ. Κεφάλαιο 2), θα αποτελέσει για κάθε σημείο: το  $s_i$  για το πρόβλημα προσανατολισμού ομάδας (TOP) ή το  $p_i$  για το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (CTOP) και για το πρόβλημα δρομολόγησης οχημάτων με συλλογή βραβείου (PCVRP). Τέλος, στο συνολικό μοντέλο που προτείνεται για την επίλυση του προβλήματος NTTDP (βλ. Σχήμα 1.1), μετά την εισαγωγή των δεδομένων των σημείων ενδιαφέροντος και την βαθμολόγηση τους μέσα από το N-person BOS, επιλέγεται το επιθυμητό πρόβλημα δρομολόγησης οχημάτων με κέρδος, ανάμεσα από τα TOP, CTOP και PCVRP, ανάλογα με τις απαιτήσεις των χρηστών, και ένας από τους προτεινόμενους αλγόριθμους: DRABC και

DRDE, για την επίλυση του.

#### **5.4.1 Παράδειγμα εφαρμογής του N-person BOS στο Πρόβλημα Σχεδιασμού Τουριστικών Διαδρομών Πολλών Ατόμων**

Σε ένα υποθετικό παράδειγμα, μία παρέα τεσσάρων ατόμων επισκέπτεται το Παρίσι. Τα τυχαία άτομα είναι, η Δήμητρα, ο Μανούσος, ο Γιώργος και η Αναστασία. Τα άτομα αυτά έχουν διαφορετικές προτιμήσεις μεταξύ τους και καλούνται να επιλέξουν ανάμεσα στα κορυφαία δέκα σημεία ενδιαφέροντος της πόλης και να τα επισκεφτούν μέσα σε ένα εφικτό χρονικό πλαίσιο. Οι κορυφαίοι δέκα προορισμοί (POIs) είναι:

1. ο πύργος του Άιφελ,
2. το μουσείο του Λούβρο,
3. το μουσείο της Ορανζερί,
4. η Παναγία των Παρισίων,
5. οι κήποι του Λουξεμβούργου,
6. η όπερα,
7. το Moulin Rouge,
8. η γκαλερί Λαφαγιέτ,
9. το γήπεδο Παρκ ντε Πρενς,
10. και ο λόφος της Μονμάρτης.

Σε αυτό το σημείο, θα εφαρμοστούν οι εξισώσεις του N-person BOS, ώστε να γίνει μια συνολική βαθμολόγηση των δέκα σημείων από τα τέσσερα άτομα με διαφορετικές προτιμήσεις και στη συνέχεια να επιλυθεί ένα από τα προβλήματα δρομολόγησης οχημάτων με κέρδος. Η πρώτη είσοδος που απαιτείται από τους χρήστες είναι η προτίμηση τους για κάθε σημείο ενδιαφέροντος, 1, αν τους αρέσει και 0, διαφορετικά. Στη συνέχεια, πρέπει να δηλώσουν την απόφασή τους, 1, αν θα επισκέπτονταν τελικά το σημείο και 0, διαφορετικά (κάποιο άτομο μπορεί να επιλέξει την επίσκεψη σε ένα σημείο που δεν του αρέσει έτσι ώστε να ικανοποιήσει τα υπόλοιπα μέλη της τουριστικής ομάδας). Έτσι, δημιουργούνται οι δύο Πίνακες 5.2 και 5.3, προτίμησης και απόφασης, αντίστοιχα, μεγέθους (4,10).

Πίνακας 5.2: Προτίμηση  $N = 4, P = 10$

	$POI_1$	$POI_2$	$POI_3$	$POI_4$	$POI_5$	$POI_6$	$POI_7$	$POI_8$	$POI_9$	$POI_{10}$
Δήμητρα	1	1	1	0	1	0	1	1	0	0
Μανούσος	1	0	1	1	0	1	0	1	1	1
Γιώργος	1	0	0	1	1	1	0	1	1	0
Αναστασία	1	0	0	1	0	1	0	1	0	1

Πίνακας 5.3: Απόφαση  $N = 4, P = 10$

	$POI_1$	$POI_2$	$POI_3$	$POI_4$	$POI_5$	$POI_6$	$POI_7$	$POI_8$	$POI_9$	$POI_{10}$
Δήμητρα	1	1	1	0	1	0	1	1	1	0
Μανούσος	1	0	1	0	0	1	0	1	0	1
Γιώργος	1	1	0	1	1	0	1	1	0	0
Αναστασία	1	0	1	0	1	1	0	1	1	1

Παρατηρώντας τους παραπάνω πίνακες, βλέπουμε για παράδειγμα ότι ο Μανούσος και ο Γιώργος, θα ήθελαν να επισκεφθούν το γήπεδο Παρκ ντε Πρενς ( $POI_9$ ), ενώ οι γυναίκες της παρέας δεν θα επιθυμούσαν, επιλέγουν τελικά να μην το επισκεφτούν. Άλλο παράδειγμα είναι το μουσείο του Λούβρου ( $POI_2$ ), το οποίο θέλει να επισκεφτεί μόνο η Δήμητρα, επιλέγει εν τέλη και ο Γιώργος να την ακολουθήσει. Για τιμές των μεταβλητών στις Εξισώσεις (5.2-5.5),  $a = 2, b = 1, c = 1, d = 0.5$ , υπολογίζεται για κάθε κόμβο η τιμή του κέρδους, δηλαδή η ικανοποίηση που θα απολάβει κάθε άτομο της ομάδας απο αυτόν και σχηματίζεται ο Πίνακας 5.4, *Payoff*. Τέλος, αθροιστικά προκύπτει το κέρδος του κάθε κόμβου, όπως φαίνεται στον Πίνακα 5.5.

Πίνακας 5.4: *Payoff*,  $N = 4, P = 10$

	$POI_1$	$POI_2$	$POI_3$	$POI_4$	$POI_5$	$POI_6$	$POI_7$	$POI_8$	$POI_9$	$POI_{10}$
Δήμητρα	9	5	7	7	7	5	5	9	2,5	5
Μανούσος	9	5	7	3,5	3	5	5	9	2,5	5
Γιώργος	9	2,5	3	3	7	2,5	2,5	9	2,5	5
Αναστασία	9	5	3,5	3,5	3,5	5	5	9	2,5	5

Πίνακας 5.5: *Payoff* για κάθε  $POI$

	$POI_1$	$POI_2$	$POI_3$	$POI_4$	$POI_5$	$POI_6$	$POI_7$	$POI_8$	$POI_9$	$POI_{10}$
$Payoff_{POI}$	36	17,5	20,5	17	20,5	17,5	17,5	36	10	20



# Κεφάλαιο 6

## Πειραματικά Αποτελέσματα

### 6.1 Παραδείγματα Αναφοράς

Τα πειραματικά που έγιναν στους προτεινόμενους αλγορίθμους, διεξάχθηκαν σε επεξεργαστή Intel Core i5 στα M430@2,27GHz και 4GB RAM, σε λειτουργικό σύστημα Windows 10 Pro 64-bit. Οι αλγοριθμικές διαδικασίες έχουν αναπτυχθεί σε περιβάλλον Matlab. Για να εξεταστεί η αποτελεσματικότητα των αλγορίθμων DRABC και DRDE, τα αποτελέσματα τους στην επίλυση των προβλημάτων TOP, CTOP και PCVRP συγκρίθηκαν με τα παγκοσμίως γνωστά βέλτιστα αποτελέσματα στα παραδείγματα αναφοράς, από την σχετική βιβλιογραφία. Για όλα τα παραδείγματα αναφοράς που παρουσιάζονται, έχουν πραγματοποιηθεί τρεις εκτελέσεις κάθε αλγορίθμου.

#### Πρόβλημα προσανατολισμού ομάδας

Τα παραδείγματά αναφοράς που δίνονται στη βιβλιογραφία για το πρόβλημα προσανατολισμού ομάδας, Team Orienteering Problem, παρέχονται από τους Chao et al. [8]. Τα παραδείγματα χωρίζονται σε επτά διαφορετικά σύνολα με κοινά χαρακτηριστικά και στο σύνολο τους είναι 353. Στον Πίνακα 6.1 παρουσιάζονται τα χαρακτηριστικά των συνόλων, δηλαδή το πλήθος των παραδειγμάτων που περιέχουν, τον συνολικό αριθμό των κόμβων που περιέχουν ( $N$ ) και τον αριθμό υπο-διαδρομών που απαιτούν ( $M$ ).

Πίνακας 6.1: Παραδείγματα αναφοράς για την επίλυση του TOP

Παραδείγματα των Chao et al. [8]	Πλήθος παραδειγμάτων	Πλήθος κόμβων ( $N$ )	Πλήθος υπο-διαδρομών ( $M$ )
$Set_{\{1\}}$	48	32	2,3,4
$Set_{\{2\}}$	33	21	2,3,4
$Set_{\{3\}}$	60	33	2,3,4
$Set_{\{4\}}$	56	100	2,3,4
$Set_{\{5\}}$	74	66	2,3,4
$Set_{\{6\}}$	24	64	2,3,4
$Set_{\{7\}}$	58	102	2,3,4

### Πρόβλημα προσανατολισμού ομάδας με χωρητικότητα

Τα παραδείγματά αναφοράς που δίνονται στη βιβλιογραφία για το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα, Capacitated Team Orienteering Problem, παρέχονται από τους Archetti et al. [36]. Πρόκειται για 90 παραδείγματα, τα οποία έχουν βασιστεί στα παραδείγματα των Christofides et al. [86] για την επίλυση του κλασσικού προβλήματος δρομολόγησης οχημάτων με χωρητικότητα. Ο αριθμός των πελατών στα παραδείγματα ποικίλλει από 50 έως 199. Για κάθε παράδειγμα έχουν δημιουργηθεί 9 διαφορετικοί συνδυασμοί στις τιμές του αριθμού των υπο-διαδρομών ( $M$ ), του ορίου χωρητικότητας ( $Q_{max}$ ) καθώς και του ορίου της συνολικής διανυθείσας απόστασης ( $T_{max}$ ). Ο παρακάτω Πίνακας 6.2 συνοψίζει τα χαρακτηριστικά των παραδειγμάτων που χρησιμοποιήθηκαν.

### Πρόβλημα δρομολόγησης οχημάτων με συλλογή βραβείου

Σύμφωνα με την βιβλιογραφία, δεν έχουν προταθεί παραδείγματα αναφοράς για την επίλυση του προβλήματος δρομολόγησης οχημάτων με συλλογή βραβείου, Prize-Collecting Vehicle Routing Problem. Κοινή πρακτική, είναι η χρήση των παραδειγμάτων των Christofides et al. [86] για την επίλυση του κλασσικού προβλήματος δρομολόγησης οχημάτων με χωρητικότητα. Με την επιπλέον προσθήκη ενός διανύσματος  $p$  με τυχαίες τιμές βραβείου στο διάστημα  $[0, 100]$  και μη τυποποιημένη μείωση του αριθμού των οχημάτων, όπως έχει ήδη αναφερθεί στο Κεφάλαιο 2.3.1. Για να γίνει μια αντικειμενική σύγκριση των δύο προτεινόμενων αλγορίθμων στην επίλυση του PCVRP, στην παρούσα

εργασία προτείνεται η χρήση των παραδειγμάτων αναφοράς των Archetti et al. [36].

Πίνακας 6.2: Παραδείγματα αναφοράς για την επίλυση του CTOP

Παραδείγματα των Archetti et al. [36]	Πλήθος κόμβων ( $N$ )	$Q_{max}, T_{max}$	Πλήθος υπο-διαδρομών ( $M$ )
#3	101	50,75,100	2,3,4
#6	51	50,75,100	2,3,4
#7	76	50,75,100	2,3,4
#8	101	50,75,100	2,3,4
#9	151	50,75,100	2,3,4
#10	200	50,75,100	2,3,4
#13	121	50,75,100	2,3,4
#14	101	50,75,100	2,3,4
#15	151	50,75,100	2,3,4
#16	200	50,75,100	2,3,4

## 6.2 Πειραματικά Αποτελέσματα

Στους Πίνακες από τον (6.3) μέχρι τον (6.12), που ακολουθούν, παρουσιάζονται τα πειραματικά αποτελέσματα των προτεινόμενων αλγορίθμων για κάθε πρόβλημα που επιλύθηκε, καθώς και η σύγκριση τους με τις βέλτιστες τιμές από τη βιβλιογραφία. Η πρώτη και έβδομη στήλη κάθε πίνακα, αναφέρουν το όνομα κάθε παραδείγματος, ενώ η δεύτερη και η όγδοη στήλη τις αντίστοιχες βέλτιστες δημοσιευμένες τιμές, που συμβολίζονται ως  $Best$ . Οι στήλες 3,4,9 και 10 περιέχουν τα αποτελέσματα του προτεινόμενου αλγορίθμου DRABC, όπου  $z_{best}$  η καλύτερη τιμή της αντικειμενικής συνάρτησης και  $rpe$  η απόκλιση από της από την βέλτιστη. Η τιμή  $rpe$  υπολογίζεται από την Εξίσωση (6.1). Αντίστοιχα, τα αποτελέσματα του αλγορίθμου DRDE παρουσιάζονται στις στήλες 5,6,11 και 12.

$$rpe = \frac{Best - z_{best}}{Best} * 100 \quad (6.1)$$

### 6.2.1 Πειραματικά Αποτελέσματα στο Πρόβλημα Προσανατολισμού Ομάδας

Σε αυτό το σημείο παρουσιάζονται τα πειραματικά αποτελέσματα πάνω στην επίλυση των παραδειγμάτων που αφορούν στο πρόβλημα προσανατολισμού ομάδας, Team Orienteering Problem. Όπως δείχνουν οι πίνακες 6.3, 6.4, 6.5, οι προτεινόμενοι αλγόριθμοι DRABC και DRDE επιτυγχάνουν τις βέλτιστες μέχρι τώρα γνωστές λύσεις, σε όλα τα παραδείγματα αναφοράς που περιέχονται στα σύνολα  $Set_{\{1\}}$ ,  $Set_{\{2\}}$  και  $Set_{\{3\}}$ . Πρόκειται για παραδείγματα με πλήθος κόμβων 32, 21 και 33. Στα παραδείγματα αυτά οι αλγόριθμοι αποδείχτηκαν εξαιρετικά αποτελεσματικοί. Εξίσου αποτελεσματικοί, αποδείχτηκαν και στα σύνολα  $Set_{\{5\}}$  και  $Set_{\{6\}}$ , που περιέχουν παραδείγματα με 66 και 64 κόμβους αντίστοιχα, αφού επιτεύχθηκαν βέλτιστες τιμές της αντικειμενικής συνάρτησης (βλ. Πίνακες 6.7 και 6.8).

Στο σύνολο  $Set_{\{4\}}$  (βλ. Πίνακα 6.6), που περιλαμβάνει 100 κόμβους και υπάρχει αυξημένη πολυπλοκότητα, οι προτεινόμενοι αλγόριθμοι έδωσαν καλές λύσεις, άλλες βέλτιστες και άλλες με μικρή απόκλιση από την εκάστοτε βέλτιστη τιμή. Συγκεκριμένα, ο υβριδικός αλγόριθμος της τεχνητής αποικίας μελισσών, DRABC, στα παραδείγματά με 2 υπο-διαδρομές, παρουσίασε μέσο όρο σχετικών αποκλίσεων 1,01%, σε αυτά με 3 υπο-διαδρομές 1,18% και σε αυτά με τις 4 ήταν 1,37%. Ενώ, η μεγαλύτερη σχετική απόκλιση που εμφάνισε σε όλο το σύνολο ήταν 3,29%. Αναφορικά με το ίδιο σύνολο, ο υβριδικός αλγόριθμος της διαφορικής εξέλιξης, DRDE, απέδωσε λίγο καλύτερα, αφού εμφάνισε πιο ποιοτικές λύσεις και επιτεύχθηκαν περισσότερες βέλτιστες τιμές. Αναλυτικότερα, στο σύνολο  $Set_{\{4\}}$ , στα παραδείγματα που απαιτούσαν 2 υπο-διαδρομές, ο αλγόριθμος παρουσίασε μέσο όρο σχετικών αποκλίσεων 0,87%, σε αυτά με 3 υπο-διαδρομές 1,04% και σε αυτά με τις 4 ήταν 1,18%.

Αντίστοιχα, στο σύνολο  $Set_{\{7\}}$  (βλ. Πίνακα 6.9), οι δύο αλγόριθμοι έδωσαν βέλτιστες και σχεδόν βέλτιστες, ενώ αποδείχθηκαν και αρκετά ανταγωνιστικοί μεταξύ τους. Στο σύνολο αυτό περιέχονται 102 κόμβοι και στα παραδείγματα που απαιτούσαν 2 υπο-διαδρομές, ο αλγόριθμος DRABC παρουσίασε μέσο όρο σχετικών αποκλίσεων 0,70%, ενώ ο αλγόριθμος DRDE 0,89%. Στα παραδείγματα με 3 υπο-διαδρομές οι αντίστοιχες τιμές ήταν 0,52% και 0,73%, ενώ σε αυτά με τις 4 υπο-διαδρομές ήταν 0,35% και 0,61%. Τέλος, πρέπει να αναφερθεί ότι στο σύνολο αυτό, ο αλγόριθμος DRABC σημείωσε μεγαλύτερη σχετική απόκλιση με τιμή 0,35%, ενώ ο DRDE 0,61%. Συνολικά, φαίνεται ότι στο σύνολο  $Set_{\{7\}}$ , ο αλγόριθμος DRABC υπήρξε περισσότερο αποτελεσματικός από τον DRDE.

Πίνακας 6.3: Πειραματικά Αποτελέσματα στο  $Set_{\{1\}}$

Παράδειγμα	DRABC		DRDE		Παράδειγμα		DRABC		DRDE		
	$Best$	$z_{best}$	$rpe$	$z_{best}$	$Best$	$rpe$	$z_{best}$	$rpe$	$z_{best}$	$rpe$	
$p1.2.b$	15	15	0,00%	15	0,00%	$p1.3.j$	115	115	0,00%	115	0,00%
$p1.2.c$	20	20	0,00%	20	0,00%	$p1.3.k$	135	135	0,00%	135	0,00%
$p1.2.d$	30	30	0,00%	30	0,00%	$p1.3.l$	155	155	0,00%	155	0,00%
$p1.2.e$	45	45	0,00%	45	0,00%	$p1.3.m$	175	175	0,00%	175	0,00%
$p1.2.f$	80	80	0,00%	80	0,00%	$p1.3.n$	190	190	0,00%	190	0,00%
$p1.2.g$	90	90	0,00%	90	0,00%	$p1.3.o$	205	205	0,00%	205	0,00%
$p1.2.h$	110	110	0,00%	110	0,00%	$p1.3.p$	220	220	0,00%	220	0,00%
$p1.2.i$	135	135	0,00%	135	0,00%	$p1.3.q$	230	230	0,00%	230	0,00%
$p1.2.j$	155	155	0,00%	155	0,00%	$p1.3.r$	250	250	0,00%	250	0,00%
$p1.2.k$	175	175	0,00%	175	0,00%						
$p1.2.l$	195	195	0,00%	195	0,00%	$p1.4.d$	15	15	0,00%	15	0,00%
$p1.2.m$	215	215	0,00%	215	0,00%	$p1.4.e$	15	15	0,00%	15	0,00%
$p1.2.n$	235	235	0,00%	235	0,00%	$p1.4.f$	25	25	0,00%	25	0,00%
$p1.2.o$	240	240	0,00%	240	0,00%	$p1.4.g$	35	35	0,00%	35	0,00%
$p1.2.p$	250	250	0,00%	250	0,00%	$p1.4.h$	45	45	0,00%	45	0,00%
$p1.2.q$	265	265	0,00%	265	0,00%	$p1.4.i$	60	60	0,00%	60	0,00%
$p1.2.r$	280	280	0,00%	280	0,00%	$p1.4.j$	75	75	0,00%	75	0,00%
					$p1.4.k$	100	100	0,00%	100	0,00%	
$p1.3.c$	15	15	0,00%	15	0,00%	$p1.4.l$	120	120	0,00%	120	0,00%
$p1.3.d$	15	15	0,00%	15	0,00%	$p1.4.m$	130	130	0,00%	130	0,00%
$p1.3.e$	30	30	0,00%	30	0,00%	$p1.4.n$	155	155	0,00%	155	0,00%
$p1.3.f$	40	40	0,00%	40	0,00%	$p1.4.o$	165	165	0,00%	165	0,00%
$p1.3.g$	50	50	0,00%	50	0,00%	$p1.4.p$	175	175	0,00%	175	0,00%
$p1.3.h$	70	70	0,00%	70	0,00%	$p1.4.q$	190	190	0,00%	190	0,00%
$p1.3.i$	105	105	0,00%	105	0,00%	$p1.4.r$	210	210	0,00%	210	0,00%

Πίνακας 6.4: Πειραματικά Αποτελέσματα στο  $Set_{\{2\}}$

Παράδειγμα	DRABC		DRDE		Παράδειγμα		DRABC		DRDE	
	$Best$	$z_{best}$	$rpe$	$rpe$	$Best$	$z_{best}$	$rpe$	$rpe$	$z_{best}$	$rpe$
$p2.2.a$	90	90	0,00%	0,00%	$p2.3.g$	145	145	0,00%	145	0,00%
$p2.2.b$	120	120	0,00%	0,00%	$p2.3.h$	165	165	0,00%	165	0,00%
$p2.2.c$	140	140	0,00%	0,00%	$p2.3.i$	200	200	0,00%	200	0,00%
$p2.2.d$	160	160	0,00%	0,00%	$p2.3.j$	200	200	0,00%	200	0,00%
$p2.2.e$	190	190	0,00%	0,00%	$p2.3.k$	200	200	0,00%	200	0,00%
$p2.2.f$	200	200	0,00%	0,00%						
$p2.2.g$	200	200	0,00%	0,00%	$p2.4.a$	10	10	0,00%	10	0,00%
$p2.2.h$	230	230	0,00%	0,00%	$p2.4.b$	70	70	0,00%	70	0,00%
$p2.2.i$	230	230	0,00%	0,00%	$p2.4.c$	70	70	0,00%	70	0,00%
$p2.2.j$	260	260	0,00%	0,00%	$p2.4.d$	70	70	0,00%	70	0,00%
$p2.2.k$	275	275	0,00%	0,00%	$p2.4.e$	70	70	0,00%	70	0,00%
					$p2.4.f$	105	105	0,00%	105	0,00%
$p2.3.a$	70	70	0,00%	0,00%	$p2.4.g$	105	105	0,00%	105	0,00%
$p2.3.b$	70	70	0,00%	0,00%	$p2.4.h$	120	120	0,00%	120	0,00%
$p2.3.c$	105	105	0,00%	0,00%	$p2.4.i$	120	120	0,00%	120	0,00%
$p2.3.d$	105	105	0,00%	0,00%	$p2.4.j$	120	120	0,00%	120	0,00%
$p2.3.e$	120	120	0,00%	0,00%	$p2.4.k$	180	180	0,00%	180	0,00%

Πίνακας 6.5: Πειραματικά Αποτελέσματα στο  $Set_{\{3\}}$

Παράδειγμα	DRABC		DRDE		Παράδειγμα		DRABC		DRDE		
	$Best$	$r_{pe}$	$z_{best}$	$r_{pe}$	$Best$	$r_{pe}$	$z_{best}$	$r_{pe}$	$z_{best}$	$r_{pe}$	
$p3.2.a$	90	0,00%	90	0,00%	$p3.3.k$	440	0,00%	440	0,00%	440	0,00%
$p3.2.b$	150	0,00%	150	0,00%	$p3.3.l$	480	0,00%	480	0,00%	480	0,00%
$p3.2.c$	180	0,00%	180	0,00%	$p3.3.m$	520	0,00%	520	0,00%	520	0,00%
$p3.2.d$	220	0,00%	220	0,00%	$p3.3.n$	570	0,00%	570	0,00%	570	0,00%
$p3.2.e$	260	0,00%	260	0,00%	$p3.3.o$	590	0,00%	590	0,00%	590	0,00%
$p3.2.f$	300	0,00%	300	0,00%	$p3.3.p$	640	0,00%	640	0,00%	640	0,00%
$p3.2.g$	360	0,00%	360	0,00%	$p3.3.q$	680	0,00%	680	0,00%	680	0,00%
$p3.2.h$	410	0,00%	410	0,00%	$p3.3.r$	710	0,00%	710	0,00%	710	0,00%
$p3.2.i$	460	0,00%	460	0,00%	$p3.3.s$	720	0,00%	720	0,00%	720	0,00%
$p3.2.j$	510	0,00%	510	0,00%	$p3.3.t$	760	0,00%	760	0,00%	760	0,00%
$p3.2.k$	550	0,00%	550	0,00%							
$p3.2.l$	590	0,00%	590	0,00%	$p3.4.a$	20	0,00%	20	0,00%	20	0,00%
$p3.2.m$	620	0,00%	620	0,00%	$p3.4.b$	30	0,00%	30	0,00%	30	0,00%
$p3.2.n$	660	0,00%	660	0,00%	$p3.4.c$	90	0,00%	90	0,00%	90	0,00%
$p3.2.o$	690	0,00%	690	0,00%	$p3.4.d$	100	0,00%	100	0,00%	100	0,00%
$p3.2.p$	720	0,00%	720	0,00%	$p3.4.e$	140	0,00%	140	0,00%	140	0,00%
$p3.2.q$	760	0,00%	760	0,00%	$p3.4.f$	190	0,00%	190	0,00%	190	0,00%
$p3.2.r$	790	0,00%	790	0,00%	$p3.4.g$	220	0,00%	220	0,00%	220	0,00%
$p3.2.s$	800	0,00%	800	0,00%	$p3.4.h$	240	0,00%	240	0,00%	240	0,00%
$p3.2.t$	800	0,00%	800	0,00%	$p3.4.i$	270	0,00%	270	0,00%	270	0,00%
					$p3.4.j$	310	0,00%	310	0,00%	310	0,00%
$p3.3.a$	30	0,00%	30	0,00%	$p3.4.k$	350	0,00%	350	0,00%	350	0,00%
$p3.3.b$	90	0,00%	90	0,00%	$p3.4.l$	380	0,00%	380	0,00%	380	0,00%
$p3.3.c$	120	0,00%	120	0,00%	$p3.4.m$	390	0,00%	390	0,00%	390	0,00%
$p3.3.d$	170	0,00%	170	0,00%	$p3.4.n$	440	0,00%	440	0,00%	440	0,00%
$p3.3.e$	200	0,00%	200	0,00%	$p3.4.o$	500	0,00%	500	0,00%	500	0,00%
$p3.3.f$	230	0,00%	230	0,00%	$p3.4.p$	560	0,00%	560	0,00%	560	0,00%
$p3.3.g$	270	0,00%	270	0,00%	$p3.4.q$	560	0,00%	560	0,00%	560	0,00%
$p3.3.h$	300	0,00%	300	0,00%	$p3.4.r$	600	0,00%	600	0,00%	600	0,00%
$p3.3.i$	330	0,00%	330	0,00%	$p3.4.s$	670	0,00%	670	0,00%	670	0,00%
$p3.3.j$	380	0,00%	380	0,00%	$p3.4.t$	670	0,00%	670	0,00%	670	0,00%

Πίνακας 6.6: Πειραματικά Αποτελέσματα στο  $Set_{\{4\}}$

Παράδειγμα	DRABC		DRDE		Παράδειγμα		DRABC		DRDE	
	$z_{best}$	$r_{pe}$	$z_{best}$	$r_{pe}$	Best	$z_{best}$	$r_{pe}$	$z_{best}$	$r_{pe}$	
<i>p4.2.a</i>	206	0,00%	206	0,00%	861	851	1,16%	855	0,70%	
<i>p4.2.b</i>	341	0,00%	341	0,00%	919	898	2,29%	898	2,29%	
<i>p4.2.c</i>	452	0,00%	452	0,00%	979	966	1,33%	960	1,94%	
<i>p4.2.d</i>	531	0,00%	531	0,00%	1063	1039	2,26%	1039	2,26%	
<i>p4.2.e</i>	618	0,00%	618	0,00%	1121	1103	1,61%	1108	1,16%	
<i>p4.2.f</i>	687	1,89%	675	1,75%	1172	1150	1,88%	1151	1,79%	
<i>p4.2.g</i>	757	1,32%	749	1,06%	1222	1207	1,23%	1207	1,23%	
<i>p4.2.h</i>	835	2,28%	819	1,92%	1253	1237	1,28%	1239	1,12%	
<i>p4.2.i</i>	918	0,00%	918	0,00%	1273	1262	0,86%	1267	0,47%	
<i>p4.2.j</i>	965	1,87%	946	1,97%	1295	1277	1,39%	1279	1,24%	
<i>p4.2.k</i>	1022	1,66%	1019	0,29%	1305	1296	0,69%	1296	0,69%	
<i>p4.2.l</i>	1074	2,33%	1051	2,14%						
<i>p4.2.m</i>	1132	2,39%	1106	2,30%	38	38	0,00%	38	0,00%	
<i>p4.2.n</i>	1174	1,62%	1155	1,62%	183	183	0,00%	183	0,00%	
<i>p4.2.o</i>	1218	1,56%	1198	1,64%	324	324	0,00%	324	0,00%	
<i>p4.2.p</i>	1242	0,24%	1239	0,24%	461	453	1,74%	461	0,00%	
<i>p4.2.q</i>	1268	1,26%	1253	1,18%	571	569	0,35%	569	0,35%	
<i>p4.2.r</i>	1292	1,01%	1277	1,16%	657	657	0,00%	657	0,00%	
<i>p4.2.s</i>	1304	0,77%	1301	0,23%	732	727	0,68%	731	0,14%	
<i>p4.2.t</i>	1306	0,00%	1306	0,00%	821	818	0,37%	819	0,24%	
<i>p4.3.b</i>	38	0,00%	38	0,00%	880	871	1,02%	871	1,02%	
<i>p4.3.c</i>	193	0,00%	193	0,00%	919	906	1,41%	906	1,41%	
<i>p4.3.d</i>	335	0,00%	335	0,00%	977	956	2,15%	956	2,15%	
<i>p4.3.e</i>	468	0,64%	468	0,00%	1061	1036	2,36%	1030	2,92%	
<i>p4.3.f</i>	579	1,38%	579	0,00%	1124	1091	2,94%	1091	2,94%	
<i>p4.3.g</i>	653	0,92%	646	1,07%	1161	1125	3,10%	1124	3,19%	
<i>p4.3.h</i>	729	1,37%	713	2,19%	1216	1176	3,29%	1195	1,73%	
<i>p4.3.i</i>	809	2,22%	796	1,61%	1260	1233	2,14%	1233	2,14%	
					1285	1262	1,79%	1262	1,79%	



Πίνακας 6.7: Πειραματικά Αποτελέσματα στο  $Set_{\{5\}}$

Παράδειγμα	DRABC		DRDE		Παράδειγμα		DRABC		DRDE		
	$Best$	$r_{pe}$	$z_{best}$	$r_{pe}$	$Best$	$r_{pe}$	$z_{best}$	$r_{pe}$	$z_{best}$	$r_{pe}$	
$p5.2.b$	20	0,00%	20	0,00%	$p5.3.n$	755	0,00%	755	0,00%	755	0,00%
$p5.2.c$	50	0,00%	50	0,00%	$p5.3.o$	870	0,00%	870	0,00%	870	0,00%
$p5.2.d$	80	0,00%	80	0,00%	$p5.3.p$	990	0,00%	990	0,00%	990	0,00%
$p5.2.e$	180	0,00%	180	0,00%	$p5.3.q$	1070	0,00%	1070	0,00%	1070	0,00%
$p5.2.f$	240	0,00%	240	0,00%	$p5.3.r$	1125	0,00%	1125	0,00%	1125	0,00%
$p5.2.g$	320	0,00%	320	0,00%	$p5.3.s$	1190	0,00%	1190	0,00%	1190	0,00%
$p5.2.h$	410	0,00%	410	0,00%	$p5.3.t$	1260	0,00%	1260	0,00%	1260	0,00%
$p5.2.i$	480	0,00%	480	0,00%	$p5.3.u$	1345	0,00%	1345	0,00%	1345	0,00%
$p5.2.j$	580	0,00%	580	0,00%	$p5.3.v$	1425	0,00%	1425	0,00%	1425	0,00%
$p5.2.k$	670	0,00%	670	0,00%	$p5.3.w$	1485	0,00%	1485	0,00%	1485	0,00%
$p5.2.l$	800	0,00%	800	0,00%	$p5.3.x$	1555	0,00%	1555	0,00%	1555	0,00%
$p5.2.m$	860	0,00%	860	0,00%	$p5.3.y$	1595	0,00%	1595	0,00%	1595	0,00%
$p5.2.n$	925	0,00%	925	0,00%	$p5.3.z$	1635	0,00%	1635	0,00%	1635	0,00%
$p5.2.o$	1020	0,00%	1020	0,00%							
$p5.2.p$	1150	0,00%	1150	0,00%	$p5.4.c$	20	0,00%	20	0,00%	20	0,00%
$p5.2.q$	1195	0,00%	1195	0,00%	$p5.4.d$	20	0,00%	20	0,00%	20	0,00%
$p5.2.r$	1260	0,00%	1260	0,00%	$p5.4.e$	20	0,00%	20	0,00%	20	0,00%
$p5.2.s$	1340	0,00%	1340	0,00%	$p5.4.f$	80	0,00%	80	0,00%	80	0,00%
$p5.2.t$	1400	0,00%	1400	0,00%	$p5.4.g$	140	0,00%	140	0,00%	140	0,00%
$p5.2.u$	1460	0,00%	1460	0,00%	$p5.4.h$	140	0,00%	140	0,00%	140	0,00%
$p5.2.v$	1505	0,00%	1505	0,00%	$p5.4.i$	240	0,00%	240	0,00%	240	0,00%
$p5.2.w$	1565	0,00%	1565	0,00%	$p5.4.j$	340	0,00%	340	0,00%	340	0,00%
$p5.2.x$	1610	0,00%	1610	0,00%	$p5.4.k$	340	0,00%	340	0,00%	340	0,00%
$p5.2.y$	1645	0,00%	1645	0,00%	$p5.4.l$	430	0,00%	430	0,00%	430	0,00%
$p5.2.z$	1680	0,00%	1680	0,00%	$p5.4.m$	555	0,00%	555	0,00%	555	0,00%
					$p5.4.n$	620	0,00%	620	0,00%	620	0,00%
$p5.3.b$	15	0,00%	15	0,00%	$p5.4.o$	690	0,00%	690	0,00%	690	0,00%
$p5.3.c$	20	0,00%	20	0,00%	$p5.4.p$	765	0,00%	765	0,00%	765	0,00%
$p5.3.d$	60	0,00%	60	0,00%	$p5.4.q$	860	0,00%	860	0,00%	860	0,00%
$p5.3.e$	95	0,00%	95	0,00%	$p5.4.r$	960	0,00%	960	0,00%	960	0,00%
$p5.3.f$	110	0,00%	110	0,00%	$p5.4.s$	1030	0,00%	1030	0,00%	1030	0,00%
$p5.3.g$	185	0,00%	185	0,00%	$p5.4.t$	1160	0,00%	1160	0,00%	1160	0,00%
$p5.3.h$	260	0,00%	260	0,00%	$p5.4.u$	1300	0,00%	1300	0,00%	1300	0,00%
$p5.3.i$	335	0,00%	335	0,00%	$p5.4.v$	1320	0,00%	1320	0,00%	1320	0,00%
$p5.3.j$	470	0,00%	470	0,00%	$p5.4.w$	1390	0,00%	1390	0,00%	1390	0,00%
$p5.3.k$	495	0,00%	495	0,00%	$p5.4.x$	1450	0,00%	1450	0,00%	1450	0,00%
$p5.3.l$	595	0,00%	595	0,00%	$p5.4.y$	1520	0,00%	1520	0,00%	1520	0,00%
$p5.3.m$	650	0,00%	650	0,00%	$p5.4.z$	1620	0,00%	1620	0,00%	1620	0,00%

Πίνακας 6.8: Πειραματικά Αποτελέσματα στο  $Set_{\{6\}}$

Παράδειγμα	DRABC		DRDE		Παράδειγμα		DRABC		DRDE	
	$Best$	$z_{best}$	$rpe$	$z_{best}$	$Best$	$z_{best}$	$rpe$	$z_{best}$	$rpe$	$z_{best}$
<i>p6.2.d</i>	192	192	0,00%	192	444	444	0,00%	444	0,00%	444
<i>p6.2.e</i>	360	360	0,00%	360	642	642	0,00%	642	0,00%	642
<i>p6.2.f</i>	588	588	0,00%	588	828	828	0,00%	828	0,00%	828
<i>p6.2.g</i>	660	660	0,00%	660	894	894	0,00%	894	0,00%	894
<i>p6.2.h</i>	780	780	0,00%	780	1002	1002	0,00%	1002	0,00%	1002
<i>p6.2.i</i>	888	888	0,00%	888	1080	1080	0,00%	1080	0,00%	1080
<i>p6.2.j</i>	948	948	0,00%	948	1170	1170	0,00%	1170	0,00%	1170
<i>p6.2.k</i>	1032	1032	0,00%	1032						
<i>p6.2.l</i>	1116	1116	0,00%	1116	366	366	0,00%	366	0,00%	366
<i>p6.2.m</i>	1188	1188	0,00%	1188	528	528	0,00%	528	0,00%	528
<i>p6.2.n</i>	1260	1260	0,00%	1260	696	696	0,00%	696	0,00%	696
					<i>p6.4.m</i>	912	912	912	0,00%	912
<i>p6.3.g</i>	282	282	0,00%	282	1068	1068	0,00%	1068	0,00%	1068

Πίνακας 6.9: Πειραματικά Αποτελέσματα στο  $Set_{\{7\}}$

Παράδειγμα	DRABC		DRDE		Παράδειγμα		DRABC		DRDE	
	$Best$	$z_{best}$	$rpe$	$z_{best}$	$Best$	$rpe$	$z_{best}$	$rpe$	$z_{best}$	$rpe$
<i>p7.2.a</i>	30	30	0,00%	30	0,00%	633	633	0,00%	632	0,16%
<i>p7.2.b</i>	64	64	0,00%	64	0,00%	<i>p7.3.k</i>	684	0,29%	679	0,73%
<i>p7.2.c</i>	101	101	0,00%	101	0,00%	<i>p7.3.l</i>	762	1,18%	755	0,92%
<i>p7.2.d</i>	190	190	0,00%	190	0,00%	<i>p7.3.m</i>	820	0,73%	814	0,73%
<i>p7.2.e</i>	290	289	0,34%	290	0,00%	<i>p7.3.n</i>	874	0,34%	865	1,03%
<i>p7.2.f</i>	387	384	0,78%	387	0,00%	<i>p7.3.o</i>	929	0,86%	917	1,29%
<i>p7.2.g</i>	459	459	0,00%	459	0,00%	<i>p7.3.p</i>	987	1,32%	966	2,13%
<i>p7.2.h</i>	521	521	0,00%	521	0,00%	<i>p7.3.q</i>	1026	1,75%	1008	1,75%
<i>p7.2.i</i>	580	579	0,17%	573	1,21%	<i>p7.3.r</i>	1081	1,02%	1061	1,85%
<i>p7.2.j</i>	646	641	0,77%	634	1,86%	<i>p7.3.s</i>	1120	1,52%	1103	1,52%
<i>p7.2.k</i>	705	700	0,71%	685	2,84%	<i>p7.3.t</i>				
<i>p7.2.l</i>	767	753	1,83%	758	1,17%	<i>p7.4.b</i>	30	0,00%	30	0,00%
<i>p7.2.m</i>	827	821	0,73%	821	0,73%	<i>p7.4.c</i>	46	0,00%	46	0,00%
<i>p7.2.n</i>	888	877	1,24%	878	1,13%	<i>p7.4.d</i>	79	0,00%	79	0,00%
<i>p7.2.o</i>	945	934	1,16%	933	1,27%	<i>p7.4.e</i>	123	0,00%	123	0,00%
<i>p7.2.p</i>	1002	977	2,50%	977	2,50%	<i>p7.4.f</i>	164	0,00%	164	0,00%
<i>p7.2.q</i>	1044	1033	1,05%	1031	1,25%	<i>p7.4.g</i>	217	0,00%	217	0,00%
<i>p7.2.r</i>	1094	1075	1,74%	1067	2,47%	<i>p7.4.h</i>	285	0,00%	285	0,00%
<i>p7.2.s</i>	1136	1136	0,00%	1134	0,18%	<i>p7.4.i</i>	366	0,55%	359	1,91%
<i>p7.2.t</i>	1179	1168	0,93%	1165	1,19%	<i>p7.4.j</i>	462	0,65%	459	0,65%
<i>p7.3.b</i>	46	46	0,00%	46	0,00%	<i>p7.4.k</i>	520	0,38%	511	1,73%
<i>p7.3.c</i>	79	79	0,00%	79	0,00%	<i>p7.4.l</i>	590	0,34%	588	0,34%
<i>p7.3.d</i>	117	117	0,00%	117	0,00%	<i>p7.4.m</i>	646	0,00%	644	0,31%
<i>p7.3.e</i>	175	175	0,00%	175	0,00%	<i>p7.4.n</i>	730	0,55%	721	1,23%
<i>p7.3.f</i>	247	247	0,00%	247	0,00%	<i>p7.4.o</i>	781	0,00%	778	0,38%
<i>p7.3.g</i>	344	344	0,00%	344	0,00%	<i>p7.4.p</i>	846	0,47%	838	0,95%
<i>p7.3.h</i>	425	425	0,00%	425	0,00%	<i>p7.4.q</i>	909	0,99%	899	1,10%
<i>p7.3.i</i>	487	484	0,62%	480	1,44%	<i>p7.4.r</i>	970	0,72%	966	0,41%
<i>p7.3.j</i>	564	563	0,18%	562	0,35%	<i>p7.4.s</i>	1022	1,17%	1005	1,66%
						<i>p7.4.t</i>	1077	0,74%	1067	0,93%

### 6.2.2 Πειραματικά Αποτελέσματα στο Πρόβλημα Προσανατολισμού Ομάδας με Χωρητικότητα

Ακολουθούν τα πειραματικά αποτελέσματα για το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα, Capacitated Team Orienteering Problem. Στον Πίνακα 6.10, παρουσιάζονται τα αποτελέσματα των παραδειγμάτων αναφοράς για μέγιστη χωρητικότητα και μήκος διαδρομής, ίσες με 50 μονάδες. Οι δύο προτεινόμενοι αλγόριθμοι DRABC και DRDE, σε αυτά τα παραδείγματα έχουν εντοπίσει αρκετές από τις βέλτιστες τιμές της βιβλιογραφίας. Συγκεκριμένα, ο αλγόριθμος DRABC εντόπισε βέλτιστες τιμές στα 18 από τα 30 παραδείγματα, ενώ παρουσίασε μέση σχετική απόκλιση από τις βέλτιστες τιμές στο σύνολο των παραδειγμάτων, 0,50%. Πρέπει να αναφερθεί ότι οι τιμές της σχετικής απόκλισης δεν ξεπέρασαν το 2%, με την πλειοψηφία τους να βρίσκεται μεταξύ 0% και 1%. Εξαιρέση αποτελεί ένα συγκεκριμένο παράδειγμα, στο οποίο και οι δύο αλγόριθμοι παρουσίασαν την συνολικά μέγιστη απόκλιση, 4,56%. Στο συγκεκριμένο παράδειγμα (#14 για  $M = 4$ ), οι προτεινόμενοι αλγόριθμοι δεν κατάφεραν να ξεπεράσουν την τοπικά μέγιστη τιμή και να φτάσουν σε μία ολικά βέλτιστη. Προσπερνώντας αυτό το παράδειγμα, παρατηρούμε ότι ο αλγόριθμος DRDE, έδωσε καλύτερα αποτελέσματα, αφού έδωσε βέλτιστες τιμές σε 22 από τα 30 παραδείγματα ενώ σε 6 από αυτά η τυπική απόκλιση δεν ξεπέρασε το 0,81%.

Στον Πίνακα 6.11 παρουσιάζονται τα αποτελέσματα των αλγορίθμων για οριακές τιμές  $Q_{max}$  και  $T_{max}$  ίσες με 75 μονάδες. Σε αυτό το σύνολο παραδειγμάτων ο αλγόριθμος DRDE απέδωσε λίγο καλύτερα από τον DRABC, αφού εντόπισε 13 στις 30 βέλτιστες λύσεις, έναντι των 7 στις 30. Όμως και οι δύο αλγόριθμοι παρουσίασαν τυπικές αποκλίσεις κάτω από 2%, με την πλειοψηφία τους να βρίσκεται μεταξύ 0% και 1%. Τέλος, οι αλγόριθμοι παρουσίασαν μέση σχετική απόκλιση στο σύνολο των 30 παραδειγμάτων, 0,70% ο DRABC και 0,55% ο DRDE, κάτι που αποδεικνύει ότι είναι ανταγωνιστικοί μεταξύ τους.

Τα αντίστοιχα αποτελέσματα των παραδειγμάτων για οριακές τιμές στις 100 μονάδες, παρουσιάζει ο Πίνακας 6.12. Οι δύο προτεινόμενοι αλγόριθμοι, DRABC και DRDE, έδωσαν ικανοποιητικές λύσεις, αφού εντόπισαν λύσεις με μέση τιμή σχετικής απόκλισης 1,29% και 1,05%, αντίστοιχα, ενώ οι μέγιστες αποκλίσεις που σημειώθηκαν ήταν 2,86% και 1,93%. Ακόμα και σε αυτά τα παραδείγματα, φαίνεται η υπεροχή του αλγορίθμου DRDE, αφού σε όλα τα 30 παραδείγματα έχει παρουσιάσει σχετική απόκλιση κάτω από 2%.

Πίνακας 6.10: Πειραματικά Αποτελέσματα για  $Q_{max}, T_{max} = 50$

Παράδειγμα, $Q_{max}, T_{max} = 50$			<i>DRABC</i>		<i>DRDE</i>	
#	<i>M</i>	<i>Best</i>	$z_{best}$	<i>rpe</i>	$z_{best}$	<i>rpe</i>
3	2	133	133	0,00%	133	0,00%
6	2	121	121	0,00%	121	0,00%
7	2	126	126	0,00%	126	0,00%
8	2	133	133	0,00%	133	0,00%
9	2	137	137	0,00%	137	0,00%
10	2	134	134	0,00%	134	0,00%
13	2	134	134	0,00%	134	0,00%
14	2	124	123	0,81%	123	0,81%
15	2	134	133	0,75%	133	0,75%
16	2	137	137	0,00%	137	0,00%
3	3	198	197	0,51%	198	0,00%
6	3	177	177	0,00%	177	0,00%
7	3	187	187	0,00%	187	0,00%
8	3	198	196	1,01%	198	0,00%
9	3	201	199	1,00%	201	0,00%
10	3	200	198	1,00%	200	0,00%
13	3	193	193	0,00%	193	0,00%
14	3	184	180	2,17%	180	2,17%
15	3	200	199	0,50%	199	0,50%
16	3	203	203	0,00%	203	0,00%
3	4	260	260	0,00%	260	0,00%
6	4	222	222	0,00%	222	0,00%
7	4	240	240	0,00%	240	0,00%
8	4	260	260	0,00%	260	0,00%
9	4	262	260	0,76%	260	0,76%
10	4	265	263	0,75%	263	0,75%
13	4	243	243	0,00%	243	0,00%
14	4	241	230	4,56%	230	4,56%
15	4	266	263	1,13%	264	0,75%
16	4	269	269	0,00%	269	0,00%

Πίνακας 6.11: Πειραματικά Αποτελέσματα για  $Q_{max}, T_{max} = 75$

Παράδειγμα, $Q_{max}, T_{max} = 75$			<i>DRABC</i>		<i>DRDE</i>	
#	<i>M</i>	<i>Best</i>	$z_{best}$	<i>rpe</i>	$z_{best}$	<i>rpe</i>
3	2	208	208	0,00%	208	0,00%
6	2	183	182	0,55%	183	0,00%
7	2	193	193	0,00%	193	0,00%
8	2	208	208	0,00%	208	0,00%
9	2	210	209	0,48%	209	0,48%
10	2	208	208	0,00%	208	0,00%
13	2	193	192	0,52%	193	0,00%
14	2	190	190	0,00%	190	0,00%
15	2	211	210	0,47%	211	0,00%
16	2	212	212	0,00%	212	0,00%
3	3	307	305	0,65%	307	0,00%
6	3	269	267	0,74%	269	0,00%
7	3	287	284	1,05%	287	0,00%
8	3	307	306	0,33%	306	0,33%
9	3	312	306	1,92%	306	1,92%
10	3	311	305	1,93%	305	1,93%
13	3	265	263	0,75%	263	0,75%
14	3	279	279	0,00%	279	0,00%
15	3	315	312	0,95%	313	0,63%
16	3	317	315	0,63%	315	0,63%
3	4	403	398	1,24%	401	0,50%
6	4	349	348	0,29%	348	0,29%
7	4	378	371	1,85%	373	1,32%
8	4	403	401	0,50%	399	0,99%
9	4	408	402	1,47%	402	1,47%
10	4	410	403	1,71%	403	1,71%
13	4	323	321	0,62%	321	0,62%
14	4	366	364	0,55%	360	1,64%
15	4	414	410	0,97%	412	0,48%
16	4	420	416	0,95%	416	0,95%

Πίνακας 6.12: Πειραματικά Αποτελέσματα για  $Q_{max}, T_{max} = 100$

Παράδειγμα, $Q_{max}, T_{max} = 100$			<i>DRABC</i>		<i>DRDE</i>	
#	<i>M</i>	<i>Best</i>	$z_{best}$	<i>rpe</i>	$z_{best}$	<i>rpe</i>
3	2	277	275	0,72%	277	0,00%
6	2	252	251	0,40%	251	0,40%
7	2	266	266	0,00%	266	0,00%
8	2	277	275	0,72%	275	0,72%
9	2	279	275	1,43%	277	0,72%
10	2	282	277	1,77%	277	1,77%
13	2	253	252	0,40%	251	0,79%
14	2	271	270	0,37%	270	0,37%
15	2	282	282	0,00%	282	0,00%
16	2	285	284	0,35%	284	0,35%
3	3	408	407	0,25%	406	0,49%
6	3	369	363	1,63%	365	1,08%
7	3	397	390	1,76%	391	1,51%
8	3	408	404	0,98%	405	0,74%
9	3	414	405	2,17%	406	1,93%
10	3	417	409	1,92%	411	1,44%
13	3	344	342	0,58%	341	0,87%
14	3	399	391	2,01%	394	1,25%
15	3	417	412	1,20%	414	0,72%
16	3	422	417	1,18%	415	1,66%
3	4	531	521	1,88%	523	1,51%
6	4	482	471	2,28%	473	1,87%
7	4	521	510	2,11%	512	1,73%
8	4	531	519	2,26%	523	1,51%
9	4	545	531	2,57%	536	1,65%
10	4	552	538	2,54%	543	1,63%
13	4	419	414	1,19%	413	1,43%
14	4	525	510	2,86%	515	1,90%
15	4	549	543	1,09%	543	1,09%
16	4	554	551	0,54%	553	0,18%

### 6.2.3 Πρόβλημα Δρομολόγησης Οχημάτων με Συλλογή Κέρδους

Για το πρόβλημα δρομολόγησης οχημάτων με συλλογή κέρδους, Prize Collection Vehicle Routing Problem, όπως αναφέρθηκε σε παραπάνω σημείο, δεν υπάρχουν συγκεκριμένα παραδείγματα αναφοράς ειδικά διαμορφωμένα για αυτό το πρόβλημα. Έτσι, τα πειραματικά αποτελέσματα, διεξήχθησαν στα ίδια παραδείγματα αναφοράς με το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα και τα αποτελέσματα των δύο προτεινόμενων αλγορίθμων συγκρίνονται μεταξύ τους βάση της σχετικής απόκλισης της τιμής της αντικειμενικής συνάρτησης. Συνοψίζοντας τους παρακάτω πίνακες των αποτελεσμάτων (βλ. Πίνακας 6.13, 6.14 και 6.15) παρατηρούμε ότι ο αλγόριθμος DRDE υπερέχει του DRABC. Συγκεκριμένα, ο αλγόριθμος DRDE έχει παρουσιάσει χαμηλότερη τιμή της αντικειμενικής σε 20,18 και 17 από κάθε σύνολο προβλημάτων για  $Q_{max}$  και  $T_{max}$  ίσα με 50,75 και 100, αντίστοιχα.



Πίνακας 6.13: Πειραματικά Αποτελέσματα για  $Q_{max}, T_{max} = 50$

Παράδειγμα, $Q_{max}, T_{max} = 50$		<i>DRABC</i>	<i>DRDE</i>	<i>pd</i>
#	<i>M</i>	$z_{best}$	$z_{best}$	
3	2	195,41	196,1	-0,35%
6	2	197,27	196,61	0,33%
7	2	195,08	195,22	-0,07%
8	2	196,34	194,66	0,86%
9	2	194,75	195,19	-0,23%
10	2	194,75	194,12	0,32%
13	2	195,22	194,15	0,55%
14	2	197,84	197,11	0,37%
15	2	195,98	195,26	0,37%
16	2	194,41	194,73	-0,16%
3	3	295,40	295,04	0,12%
6	3	296,83	297,25	-0,14%
7	3	294,80	294,25	0,19%
8	3	296,33	294,46	0,63%
9	3	293,68	294,87	-0,41%
10	3	292,27	294,08	-0,62%
13	3	295,98	293,99	0,67%
14	3	299,36	297	0,79%
15	3	295,16	294,29	0,29%
16	3	293,15	294,63	-0,50%
3	4	396,92	393,21	0,93%
6	4	396,81	397,08	-0,07%
7	4	394,47	393,02	0,37%
8	4	397,69	394,65	0,76%
9	4	396,99	393,08	0,98%
10	4	392,80	394,22	-0,36%
13	4	391,89	391,66	0,06%
14	4	398,12	396,72	0,35%
15	4	396,03	394,43	0,40%
16	4	395,41	392,99	0,61%

Πίνακας 6.14: Πειραματικά Αποτελέσματα για  $Q_{max}, T_{max} = 75$

Παράδειγμα, $Q_{max}, T_{max} = 75$		<i>DRABC</i>	<i>DRDE</i>	<i>pd</i>
#	<i>M</i>	$z_{best}$	$z_{best}$	
3	2	193,55	194,05	-0,26%
6	2	195,00	194,71	0,15%
7	2	192,37	192,72	-0,18%
8	2	192,73	194,02	-0,67%
9	2	194,67	193,94	0,37%
10	2	192,68	189,78	1,51%
13	2	197,00	191,3	2,89%
14	2	197,35	195,59	0,89%
15	2	191,67	193,12	-0,76%
16	2	192,24	193,29	-0,55%
3	3	292,10	292,66	-0,19%
6	3	295,26	293,9	0,46%
7	3	292,69	292,84	-0,05%
8	3	293,12	293,29	-0,06%
9	3	291,31	293,81	-0,86%
10	3	291,78	286,8	1,71%
13	3	298,91	289,23	3,24%
14	3	298,38	293,62	1,60%
15	3	290,63	291,3	-0,23%
16	3	291,34	291,29	0,02%
3	4	395,16	392,55	0,66%
6	4	397,50	394,5	0,75%
7	4	394,34	392,54	0,46%
8	4	396,53	390,59	1,50%
9	4	392,69	392,22	0,12%
10	4	392,76	387,67	1,30%
13	4	398,98	388,58	2,61%
14	4	399,70	392,18	1,88%
15	4	390,51	391,74	-0,31%
16	4	388,34	391,72	-0,87%

Πίνακας 6.15: Πειραματικά Αποτελέσματα για  $Q_{max}, T_{max} = 100$

Παράδειγμα, $Q_{max}, T_{max} = 100$		<i>DRABC</i>	<i>DRDE</i>	
#	<i>M</i>	$z_{best}$	$z_{best}$	<i>pd</i>
3	2	189,71	190,31	-0,32%
6	2	195,14	193,71	0,74%
7	2	190,03	191,96	-1,01%
8	2	189,46	190,7	-0,65%
9	2	191,50	192,13	-0,33%
10	2	191,24	191,33	-0,05%
13	2	195,47	188,38	1,61%
14	2	195,47	192,37	1,61%
15	2	189,97	189,89	0,04%
16	2	188,29	189,22	-0,49%
3	3	292,51	290,65	0,64%
6	3	294,23	293,58	0,22%
7	3	289,63	291,41	-0,61%
8	3	292,72	291,35	0,47%
9	3	292,47	292,84	-0,13%
10	3	287,64	288,16	-0,18%
13	3	290,66	283,67	2,46%
14	3	295,29	291,57	1,28%
15	3	290,54	291,22	-0,23%
16	3	287,07	288,5	-0,50%
3	4	394,59	385,37	2,39%
6	4	397,24	396,57	0,17%
7	4	392,66	392,21	0,11%
8	4	392,42	386,52	1,53%
9	4	389,87	390,12	-0,06%
10	4	386,07	382,61	0,90%
13	4	391,37	382,94	2,20%
14	4	398,29	391,18	1,82%
15	4	385,67	385,42	0,06%
16	4	385,47	386,97	-0,39%

## Κεφάλαιο 7

### Συμπεράσματα

Στόχος της παρούσας μεταπτυχιακής εργασίας ήταν η επίλυση ενός προβλήματος που δεν έχει μελετηθεί μέχρι τώρα, σύμφωνα με την βιβλιογραφία. Πρόκειται για το πρόβλημα σχεδιασμού τουριστικών διαδρομών πολλών ατόμων, N-Tourist Trip Design Problem. Η βασική ιδέα στην οποία επικεντρώνεται το πρόβλημα είναι ο συντονισμός πολλών ατόμων με διαφορετικές προτιμήσεις, με στόχο τη μεγιστοποίηση της ικανοποίησής τους, κατά την επίσκεψή τους σε κάποια σημεία ενδιαφέροντος, POIs, στο πλαίσιο μίας κοινής τουριστικής διαδρομής. Τα προβλήματα που εμφανίζονται κατά την επίλυση του είναι: (i) η επιλογή των σημείων που θα ικανοποιήσουν όλες τις διαφορετικές προτιμήσεις των ατόμων, (ii) η επιλογή ενός μοντέλου για την αναπαράσταση του προβλήματος και (iii) η ανάπτυξη ενός κατάλληλου αλγοριθμικού πλαισίου για την επίλυση του προβλήματος.

Για να αντιμετωπιστεί το πρόβλημα της επιλογής των κατάλληλων σημείων, και κατά επέκταση, η αξιολόγηση τους από τα άτομα που επιθυμούν μια κοινή διαδρομή, έχοντας όμως διαφορετικές προτιμήσεις σε αυτά, χρησιμοποιήθηκε ένα παίγνιο από την Θεωρία Παιγνίων. Πρόκειται για το παίγνιο συντονισμού της Μάχης των Φύλων. Το παίγνιο αυτό στην κλασική του εκδοχή, επικεντρώνεται σε δυο μόνο άτομα με αντικρουόμενες προτιμήσεις. Έτσι, ήταν απαραίτητη η χρήση του αντιστοίχου εκτεταμένου παιγνίου για πολλά άτομα, το N-person Battle Of Sexes. Από το παίγνιο αυτό και μέσω κατάλληλων συναρτήσεων, βασισμένες σε αποκλειστικά δυαδικές αποφάσεις από τα άτομα, αναφορικά με την προτίμησή τους για κάθε σημείο, δημιουργείται μία βαθμολόγηση όλων των πιθανών σημείων προς επίσκεψη. Η βαθμολόγηση αυτή, όχι μόνο συνοπολογίζει τις διαφορετικές προτιμήσεις και επιλογές των ατόμων, αλλά αποτελεί και βασική συνιστώσα στην επίλυση του προβλήματος. Το προτεινόμενο μοντέλο βασίζεται στη βαθμολόγηση αυτή, σε κατάλληλο πρόβλημα

για να το αναπαραστήσει και σε έναν αποτελεσματικό αλγόριθμο επίλυσης.

Το πρόβλημα του σχεδιασμού τουριστικών διαδρομών επιλύεται με μοντελοποίηση προβλημάτων δρομολόγησης οχημάτων με κέρδος. Συγκεκριμένα, τα προβλήματα που αποτέλεσαν τη βάση του προτεινόμενου μοντέλου ήταν: το πρόβλημα προσανατολισμού ομάδας (Team Orienteering Problem), το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (Capacitated Team Orienteering Problem) και το πρόβλημα δρομολόγησης οχημάτων με συλλογή βραβείου (Prize-collecting Vehicle Routing Problem). Κάθε πρόβλημα βασίζεται στην επιλογή κόμβων ανάλογα με το κέρδος τους (την ικανοποίηση του χρήστη) και στη δημιουργία κατάλληλων διαδρομών, υπόψιν κάποιων περιορισμών.

Για την επίλυση των παραπάνω προβλημάτων έγινε χρήση δύο αλγορίθμων εμπνευσμένων από τη φύση, του αλγόριθμου της τεχνητής αποικίας μελισσών (Artificial Bee Colony) και του αλγορίθμου της διαφορικής εξέλιξης (Differential Evolution). Οι παραπάνω αλγόριθμοι έχουν προταθεί για τη βελτιστοποίηση προβλημάτων συνεχών τιμών, ενώ σε αντίθεση τα προβλήματα που επιλύονται, στην παρούσα διατριβή, αναφέρονται σε διακριτές τιμές, λόγω της αναπαράστασης των κόμβων σε μία διαδρομή. Σε αυτό το σημείο, εμφανίζεται και άλλη μία πρωτοτυπία της παρούσας διατριβής, το μοντέλο Distance Related, (DR). Πρόκειται για ένα μοντέλο υβριδοποίησης των αλγορίθμων, σύμφωνα με το οποίο οι λύσεις κωδικοποιούνται από διακριτές σε συνεχείς τιμές και αποκωδικοποιούνται με ένα απλό ευρετικό αλγοριθμικό πλαίσιο. Η μετατροπή των τιμών βασίζεται στην ευκλείδεια απόσταση μεταξύ διαδοχικών κόμβων σε μία διαδρομή και σαν διαδικασία δεν έχει προταθεί ξανά, σύμφωνα με τη βιβλιογραφία. Έτσι, αναπτύχθηκαν δυο υβριδικοί αλγόριθμοι, ο Distance Related Artificial Bee Colony, (DRABC) και ο Distance Related Differential Evolution, (DRDE).

Οι δύο προτεινόμενοι αλγόριθμοι χρησιμοποιήθηκαν για την επίλυση των τριών προβλημάτων, TOP, CTOP και PCVRP. Για να ελεγχθεί η απόδοση και η αποτελεσματικότητα των αλγορίθμων διεξήχθησαν πειράματα, πάνω στην επίλυση κατάλληλων παραδειγμάτων αναφοράς της βιβλιογραφίας. Σε ότι αφορά την επίλυση του TOP, οι αλγόριθμοι DRABC και DRDE εντόπισαν βέλτιστες λύσεις στο 78,5% και 78,8% των παραδειγμάτων, αντίστοιχα. Τα αποτελέσματα των αλγορίθμων στην επίλυση του CTOP ήταν αρκετά ικανοποιητικά, αφού εμφάνισαν βέλτιστες τιμές και τιμές με σχετικές αποκλίσεις από τις βέλτιστες κοντά στο 1%. Ακόμα, παρατηρώντας τα αποτελέσματα φαίνεται ότι ο αλγόριθμος DRDE υπερέρχει κατά λίγο του DRABC σε ότι αφορά την ποιότητα των λύσεων. Για το πρόβλημα PCVRP, επειδή δεν έχουν δημοσιευτεί συγκεκριμένα παραδείγματα αναφοράς, χρησιμοποιήθηκαν αυτά του CTOP και έγινε σύγκριση των δύο αλγορίθμων πάνω σε αυτά. Τα αποτελέσματα έδειξαν ότι στην επίλυση

και αυτού του προβλήματος υπερέχει ο αλγόριθμος DRDE. Καταλήγοντας είναι προφανές ότι οι αλγόριθμοι είναι σε θέση να επιτύχουν βέλτιστες ή σχεδόν βέλτιστες λύσεις στα εξεταζόμενα προβλήματα.

## Βιβλιογραφία

- [1] P. Vansteenwegen, D. Van Oudheusden, The mobile tourist guide: an OR opportunity. *OR insight*, 20(3)(2007), pp.21-27.
- [2] D. Gavalas, M. Kenteris, C. Konstantopoulos, G. Pantziou, Web application for recommending personalised mobile tourist routes. *IET software*, 6(4)(2012), pp.313-322.
- [3] <http://www.mtrip.com/>.
- [4] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The orienteering problem: A survey. *European Journal of Operational Research*, 209(1)(2011), pp.1-10.
- [5] M. D. Dunlop, P. Ptaskinski, A. Morrison, S. McCallum, C. Risbey, F. Stewart, Design and development of Taeneb city guide-from paper maps and guidebooks to electronic guides, (2004). [https://personal.cis.strath.ac.uk/mark.dunlop/research/publications/04dunlop\\_enter.pdf](https://personal.cis.strath.ac.uk/mark.dunlop/research/publications/04dunlop_enter.pdf)
- [6] C. Archetti, M.G. Speranza, D. Vigo, Vehicle routing problems with profits, in: P. Toth, D. Vigo (Eds.), *Vehicle routing: Problems, methods, and applications*. MOS-SIAM Series on Optimization, Philadelphia: SIAM, 2014, pp. 273-298.
- [7] S. Butt, T. Cavalier, A heuristic for the multiple tour maximum collection problem, *Computers and Operations Research*, 21(1994), pp. 101-111.
- [8] I.M. Chao, B.L. Golden, E.A. Wasil, The team orienteering problem, *European Journal of Operational Research*, 88(3)(1996), pp. 464-474.
- [9] B.L. Golden, L. Levy, R. Vohra, The orienteering problem, *Naval research logistics*, 34(3)(1987), pp. 307-318.

- [10] A. Gunawan, H.C. Lau, P. Vansteenwegen, Orienteering Problem: A survey of recent variants, solution approaches and applications, *European Journal of Operational Research*, 255(2)(2016), pp. 315-332.
- [11] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The orienteering problem: A survey, *European Journal of Operational Research*, 209(1)(2011), pp. 1-10.
- [12] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, A survey on algorithmic approaches for solving tourist trip design problems, *Journal of Heuristics*, 20(3)(2014), pp. 291-328.
- [13] I.M. Chao, B.L. Golden, E.A. Wasil, A fast and effective heuristic for the orienteering problem, *European Journal of Operational Research*, 88(3)(1996), pp. 475-489.
- [14] S. Butt, D. Ryan, An optimal solution procedure for the multiple tour maximum collection problem using column generation, *Computers and Operations Research*, 26(1999), pp. 427-441.
- [15] H. Tang, E. Miller-Hooks, A tabu search heuristic for the team orienteering problem, *Computer and Operations Research*, 32(2005), pp. 1379-1407.
- [16] S. Boussier, D. Feillet, M. Gendreau, An exact algorithm for the team orienteering problem, *4OR*, 5(2007), pp. 211-230.
- [17] C. Archetti, A. Hertz, M. Speranza, Metaheuristics for the team orienteering problem, *Journal of Heuristics*, 13(2007), pp. 49-76.
- [18] L. Ke, C. Archetti, Z. Feng, Ants can solve the team orienteering problem, *Computers and Industrial Engineering*, 54(2008), pp. 648-665.
- [19] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, D. Van Oudheusden, A guided local search metaheuristic for the team orienteering problem, *European Journal of Operational Research*, 196(1)(2009), pp. 118-127.
- [20] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, D. Van Oudheusden, Metaheuristics for tourist trip planning, in: M. Geiger, W. Habenicht, M. Sevaux, K. Sørensen (Eds.), *Metaheuristics in the Service Industry*, *Lecture Notes in Economics and Mathematical Systems*, 624, Springer-Verlag, 2009, pp. 15-31.



- [21] W. Souffriau, P. Vansteenwegen, G.V. Berghe, D. Van Oudheusden, A path relinking approach for the team orienteering problem, *Computers and Operations Research*, 37(11)(2010), pp. 1853-1859.
- [22] M. Poggi, H. Viana, E. Uchoa, The team orienteering problem: Formulations and branch-cut and price, *OASICS-OpenAccess Series in Informatics*, 14, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.
- [23] H. Bouly, D.C. Dang, A. Moukrim, A memetic algorithm for the team orienteering problem, *4OR*, 8(1)(2010), pp. 49-70.
- [24] S. Muthuswamy, S.S. Lam, Discrete particle swarm optimization for the team orienteering problem, *Memetic Computing*, 3(4)(2011), pp. 287-303.
- [25] D.C. Dang, R.N. Guibadj, A. Moukrim, A pso-based memetic algorithm for the team orienteering problem, *European Conference on the Applications of Evolutionary Computation*, Springer Berlin Heidelberg, (2011), pp. 471-480.
- [26] D.C. Dang, R. El-Hajj, A. Moukrim, A branch-and-cut algorithm for solving the team orienteering problem, *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Springer Berlin Heidelberg, (2013), pp. 332-339.
- [27] D.C. Dang, R.N. Guibadj, A. Moukrim, An effective PSO-inspired algorithm for the team orienteering problem, *European Journal of Operational Research*, 229(2)(2013), pp. 332-344.
- [28] S.W. Lin, Solving the team orienteering problem using effective multi-start simulated annealing, *Applied Soft Computing*, 13(2)(2013), pp. 1064-1073.
- [29] B.I. Kim, H. Li, A.L. Johnson, An augmented large neighbourhood search method for solving the team orienteering problem, *Expert Systems with Applications*, 40(8)(2013), pp. 3065-3072.
- [30] J. Ferreira, A. Quintas, J.A. Oliveira, Solving the team orienteering problem: developing a solution tool using a genetic algorithm approach, in: P. Kromer, M. Koppen, G. Schaefer (Eds.), *Soft computing in industrial applications, Advances in Intelligent Systems and Computing*, 223, Springer, 2014, pp. 365-375.

- [31] M. Keshtkaran, K. Ziarati, A. Bettinelli, D. Vigo, Enhanced exact solution methods for the team orienteering problem, *International Journal of Production Research*, 54(2)(2016), pp. 591-601.
- [32] L. Ke, L. Zhai, J. Li, F.T. Chan, Pareto mimic algorithm: An approach to the team orienteering problem, *Omega*, 61(2016), pp. 155-166.
- [33] R. El-Hajj, D.C. Dang, A. Moukrim, Solving the team orienteering problem with cutting planes, *Computers and Operations Research*, 74(2016), pp. 21-30.
- [34] M. Zettam, B. Elbenani, A novel randomized heuristic for the team orienteering problem, *Logistics Operations Management (GOL)*, 3rd International Conference, IEEE, 2016.
- [35] B. Karoum, B. Elbenani, Clonal selection algorithm for the team orienteering problem, *Intelligent Systems: Theories and Applications (SITA)*, 11th International Conference, IEEE, 2016.
- [36] C. Archetti, D. Feillet, A. Hertz, M. G. Speranza, The capacitated team orienteering and profitable tour problems, *Journal of the Operational Research Society*, 60(6)(2009), pp. 831-842.
- [37] C. Archetti, N. Bianchessi, M. G. Speranza, Optimal solutions for routing problems with profits. *Discrete Applied Mathematics*, 161(4)(2013), pp. 547-557.
- [38] C. D. Tarantilis, F. Stavropoulou, P. P. Repoussis, The capacitated team orienteering problem: a bi-level filter-and-fan method. *European Journal of Operational Research*, 224(1)(2013), pp.65-78.
- [39] Z. Luo, B. Cheang, A. Lim, W. Zhu, An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem. *European Journal of Operational Research*, 229(3)(2013), pp.673-682.
- [40] A. Ben-Said, R. El-Hajj, A. Moukrim, An adaptive heuristic for the Capacitated Team Orienteering Problem. *IFAC-PapersOnLine*, 49(12)(2016), pp.1662-1666.
- [41] E. Balas and C.H. Martin, *ROLL-A-ROUND: Software package for scheduling the rounds of a rolling mill*, Copyright Balas and Martin Associates, 104 Maple Heights Road, Pittsburgh, USA, (1985).

- [42] L. Tang, X. Wang, Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem. *The International Journal of Advanced Manufacturing Technology*, 29(11)(2006), pp. 1246-1258.
- [43] T. Zhang, W. A. Chaovalitwongse, Y. J. Zhang, P. M. Pardalos, The hot-rolling batch scheduling method based on the prize collecting vehicle routing problem. *Journal of Industrial and Management Optimization*, 5(4)(2009), pp. 749-765.
- [44] T. Vidal, N. Maculan, L. S. Ochi, P. H. Vaz Penna, Large neighborhoods with implicit customer selection for vehicle routing problems with profits. *Transportation Science*, 50(2)(2015), pp. 720-734.
- [45] K. Li, H. Tian, A two-level self-adaptive variable neighborhood search algorithm for the prize-collecting vehicle routing problem. *Applied Soft Computing*, 43(2016), pp. 469-479.
- [46] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 200(2005).
- [47] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39 (2007), pp. 459-471.
- [48] D. Karaboga, B. Basturk, On the performance of artificial bee colony (abc) algorithm. *Applied Soft Computing*, 8 (2008), pp. 687-697.
- [49] W.Y. Szeto, S.C. Ho, An artificial bee colony algorithm for the capacitated vehicle routing problem, *European Journal of Operational Research*, 215(1)(2011), pp. 126-135.
- [50] I. Brajevic, Artificial bee colony algorithm for the capacitated vehicle routing problem, in: *Proceedings of the European computing conference*, (2011), pp. 239-244.
- [51] P. Ji, Y. Wu, An improved artificial bee colony algorithm for the capacitated vehicle routing problem with time-dependent travel times, in: *Tenth International Symposium on Operations Research and its Applications*, 2011, pp. 75-82.

- [52] Y.J. Shi, F.W. Meng, G. J. Shen, A modified artificial bee colony algorithm for vehicle routing problems with time windows, *Information Technology Journal*, 11(10)(2012), p. 1490.
- [53] B. Wu, J.G. Lin, M. Dong, Artificial bee colony algorithm for three-dimensional loading capacitated vehicle routing problem, in *Proceedings of 20th International Conference on Industrial Engineering and Engineering Management*, Springer Berlin Heidelberg, 2013, pp. 815-825.
- [54] B. Wu, H. Cai, Z. Cui, Artificial bee colony algorithm for two-dimensional loading capacitated vehicle routing problem, *Management Science and Engineering (ICMSE)*, International Conference, IEEE, 2013, pp. 406-412.
- [55] B. Yao, P. Hu, M. Zhang, S. Wang, Artificial bee colony algorithm with scanning strategy for the periodic vehicle routing problem, *Simulation*, 89(6)(2013), pp. 762-770.
- [56] A. Gomez, S. Salhi, Solving capacitated vehicle routing problem by artificial bee colony algorithm, *Computational Intelligence in Production and Logistics Systems (CIPLS)*, IEEE Symposium, IEEE, 2014, pp. 48-52.
- [57] O.E. Nahum, Y. Hadas, U. Spiegel, Multi-objective vehicle routing problems with time windows: a vector evaluated artificial bee colony approach, *International Journal of Computer and Information Technology* 3(1)(2014), pp. 41-47.
- [58] S. Zhang, C.K.M. Lee, K.L. Choy, W. Ho, W.H. Ip, Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem, *Transportation Research Part D: Transport and Environment*, 31(2014), pp. 85-99.
- [59] S. Iqbal, M. Kaykobad, M.S. Rahman, Solving the multi-objective vehicle routing problem with soft time windows with the help of bees, *Swarm and Evolutionary Computation*, 24(2015), pp. 50-64.
- [60] S.Z. Zhang, C.K.M. Lee, . An improved artificial bee colony algorithm for the capacitated vehicle routing problem, in: *Systems, Man, and Cybernetics (SMC)*, 2015 IEEE International Conference, IEEE, 2015, pp. 2124-2128.

- [61] P.Y. Yin, Y.L. Chuang, Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking, *Applied Mathematical Modelling*, 40(21)(2016), pp. 9302-9315.
- [62] M. Alzaqebah, S. Abdullah, S. Jawarneh, Modified artificial bee colony for the vehicle routing problems with time windows, *SpringerPlus*, 5(1)(2016), pp. 1298.
- [63] S. Yu, C. Tai, Y. Liu, L. Gao, An improved artificial bee colony algorithm for vehicle routing problem with time windows: A real case in Dalian, *Advances in Mechanical Engineering*, 8(8)(2016), pp. 1-9.
- [64] H. Seidgar, M. Kiani, H. Fazlollahtabar, Genetic and artificial bee colony algorithms for scheduling of multi-skilled manpower in combined manpower-vehicle routing problem, *Production and Manufacturing Research*, 4(1)(2016), pp. 133-151.
- [65] S. Mao, M. Zheng, X. Zhao, W. Xie, Z. Wang, The uncertain time dependent vehicle routing problem with soft time windows, in *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on IEEE, 2016*, pp. 38-45.
- [66] N. Tuntitippawan, K. Asawarungsaengkul, An artificial bee colony algorithm with local search for vehicle routing problem with backhauls and time windows, *KKU Engineering Journal*, 43(2016), pp. 404-408.
- [67] T. Cura, An artificial bee colony algorithm approach for the team orienteering problem with time windows, *Computers and Industrial Engineering*, 74(2014), pp. 270-290.
- [68] R. Storn, K. Price, Differential Evolution - A simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, 11(4)(1995), pp. 41-359.
- [69] C. Erbao, L. Mingyong, A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands. *Journal of computational and applied mathematics*, 231(1)(2009), pp.302-310.
- [70] C. Erbao, L. Mingyong, The open vehicle routing problem with fuzzy demands, *Expert Systems with Applications*, 37(2010), pp.2405-2411.
- [71] L. Mingyong, C. Erbao, An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries

and time windows, *Engineering Applications of Artificial Intelligence*, 23(2)(2010), pp.188-195.

- [72] L. Hou, H. Zhou, J. Zhao, A novel discrete differential evolution algorithm for stochastic VRPSPD, *Journal of Computational Information Systems*, 6(8)(2010), pp.2483-2491.
- [73] W. Liu, X. Wang, X. Li, Memetic differential evolution for vehicle routing problem with time windows, *Advances in Swarm Intelligence*, (2012), pp.358-365.
- [74] I. Kucukoglu, N. Ozturk, A differential evolution approach for the vehicle routing problem with backhauls and time windows, *Journal Of Advanced Transportation*, 48(8)(2014), pp.942-956.
- [75] E. Berhan, P. Kromer, D. Kitaw, A. Abraham, V. Snasel, Solving stochastic vehicle routing problem with real simultaneous pickup and delivery using differential evolution. In *Innovations in Bio-inspired Computing and Applications*, Springer International Publishing, (2014), pp. 187-200.
- [76] D. Dechampai, L. Tanwanichkul, K. Sethanan, R. Pitakaso, A differential evolution algorithm for the capacitated VRP with flexibility of mixing pickup and delivery services and the maximum duration of a route in poultry industry. *Journal of Intelligent Manufacturing*, (2015), pp.1-20.
- [77] Y. Marinakis, M. Marinaki, P. Spanou, A memetic differential evolution algorithm for the vehicle routing problem with stochastic demands, In *Adaptation and hybridization in computational intelligence*, Springer International Publishin, (2015), pp. 185-204.
- [78] I. D. Psychas, M. Marinaki, Y. Marinakis, A. Migdalas, Non-dominated sorting differential evolution algorithm for the minimization of route based fuel consumption multiobjective vehicle routing problems, *Energy Systems*, (2016) pp.1-30.
- [79] A. Viktorin, D. Hrabec, M. Pluhacek, Multi-chaotic differential evolution for vehicle routing problem with profits. In *Proceedings-30th European Conference on Modelling and Simulation*, European Council for Modelling and Simulation, (2016).

- [80] L. V. Snyder, M. S. Daskin, A random-key genetic algorithm for the generalized traveling salesman problem, *European J. of Operational Research*, 174(1)(2006), pp. 38-53.
- [81] J. C. Bean, Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6(2)(1994), pp.154-160.
- [82] J.F. Nash, Equilibrium Points in n-Person Games. *Proceedings of the National Academy of Sciences*, 36(1950), pp.48-49.
- [83] J.F. Nash, Non-Cooperative Games. *Annals of Mathematics*, 54(1951), pp.286-295.
- [84] T. Schelling, Hockey helmets, concealed weapons, and daylight saving: A study of binary choices with externalities. *Journal of Conflict resolution*, 17(3)(1973), pp.381-428.
- [85] J. Zhao, MN. Szilagy, F. Szidarovszky, An n-person battle of sexes game. *Physica A: Statistical Mechanics and its Applications*, 387(14)(2008), pp.3669-3677.
- [86] N. Christofides, S. Eilon, An algorithm for one vehicle-dispatching problem. *Oper Res Q* 20(3)(1969), pp.309-318.