

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Ευφυούς Εφαρμογής Διατροφικής
Βελτιστοποίησης με Χρήση Εξελκτικών Αλγορίθμων
Πολλαπλών Στόχων»



Του φοιτητή
Γεώργιου Χαριτάκη
Αρ. Μητρώου: 2020188

Επιβλέπων
Αδαμίδης Παναγιώτης
Καθηγητής

Θεσσαλονίκη, Ιανουάριος 2026

Τίτλος Δ.Ε.: Ανάπτυξη Ευφυούς Εφαρμογής Διατροφικής Βελτιστοποίησης με Χρήση Εξελικτικών Αλγορίθμων Πολλαπλών Στόχων

Κωδικός Δ.Ε.: 25167

Όνοματεπώνυμο φοιτητή: Χαριτάκης Γεώργιος

Όνοματεπώνυμο εισηγητή: Αδαμίδης Παναγιώτης

Ημερομηνία ανάληψης Δ.Ε.: 11/03/2025

Ημερομηνία περάτωσης Δ.Ε.: 13/01/2026

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

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

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

«Στους ανθρώπους που με στηρίζουν»

Πρόλογος

Η επιλογή του θέματος της παρούσας διπλωματικής εργασίας προέκυψε από το ιδιαίτερο ενδιαφέρον για τη σύζευξη της τεχνητής νοημοσύνης με εφαρμογές που έχουν άμεσο αντίκτυπο στην καθημερινότητα. Η διατροφή αποτελεί πεδίο στο οποίο η τεχνολογία μπορεί να προσφέρει ουσιαστικές λύσεις, ιδιαίτερα σε ζητήματα εξατομίκευσης και καθοδήγησης, όπου η παραδοσιακή προσέγγιση συχνά παρουσιάζει περιορισμούς. Η ανάπτυξη μιας ολοκληρωμένης εφαρμογής Android, ικανής να δημιουργεί προσαρμοσμένα διατροφικά πλάνα με τη βοήθεια εξελικτικών αλγορίθμων πολλαπλών στόχων, αποτέλεσε πρόκληση με σημαντική ερευνητική αλλά και πρακτική αξία. Η διαδικασία υλοποίησης της εργασίας συνέβαλε ουσιαστικά στην εμβάθυνση σε τεχνολογίες κινητών συσκευών, σε σύγχρονες μεθόδους βελτιστοποίησης και στην αξιοποίηση επίσημων διατροφικών βάσεων δεδομένων. Παράλληλα, ανέδειξε τις δυσκολίες της διατροφικής μοντελοποίησης και την ανάγκη για συστήματα που συνδυάζουν θεωρητική τεκμηρίωση και λειτουργική αποτελεσματικότητα. Το αποτέλεσμα αποτελεί ένα πλήρες σύστημα που επιτρέπει τη διερεύνηση, αξιολόγηση και πρακτική εφαρμογή των αρχών της εξατομικευμένης διατροφής, προσφέροντας πολύτιμη εμπειρία και γνώσεις για μελλοντικές επεκτάσεις και ερευνητικές δραστηριότητες.

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει τη δυνατότητα αξιοποίησης εξελικτικών αλγορίθμων πολλαπλών στόχων για τη δημιουργία εξατομικευμένων διατροφικών πλάνων και υλοποιεί μια ολοκληρωμένη εφαρμογή Android που ενσωματώνει αυτή τη μεθοδολογία. Το σύστημα συλλέγει και επεξεργάζεται δεδομένα χρηστών, όπως προσωπικά χαρακτηριστικά, διατροφικούς στόχους, προτιμήσεις και περιορισμούς, ενώ αντλεί αναλυτικά θρεπτικά στοιχεία από το USDA FoodData Central API, το οποίο αξιοποιείται για τον σχηματισμό ενός εκτεταμένου συνόλου τροφίμων με πραγματικές διατροφικές τιμές. Στη συνέχεια εφαρμόζεται ένας εξελικτικός αλγόριθμος πολλαπλών στόχων βασισμένος στη βιβλιοθήκη Jenetics, ο οποίος αναζητά συνδυασμούς τροφών που ικανοποιούν τα θερμιδικά, μακροθρεπτικά και μικροθρεπτικά όρια του χρήστη, λαμβάνοντας παράλληλα υπόψη διατροφικούς περιορισμούς, αλλεργίες και τύπους γευμάτων. Η υλοποίηση περιλαμβάνει μηχανισμούς ανάκτησης, αποθήκευσης και ταξινόμησης τροφών, τη χρήση του Paging 3 για τη σταδιακή φόρτωση μεγάλου όγκου δεδομένων, καθώς και δυνατότητες όπως παραγωγή πλάνου, προβολή αναλυτικών στοιχείων και παρακολούθηση βάρους. Μέσα από μια σειρά δοκιμών αξιολογήθηκε η αξιοπιστία της εφαρμογής, η απόδοση του εξελικτικού αλγορίθμου και η ποιότητα των παραγόμενων διατροφικών πλάνων. Τα αποτελέσματα έδειξαν ότι ο αλγόριθμος καταφέρνει να προσεγγίσει αποτελεσματικά τους διατροφικούς στόχους, επιτυγχάνοντας ισορροπία μακροθρεπτικών και μικροθρεπτικών στοιχείων, ενώ η διαδικασία ρύθμισης των παραμέτρων ανέδειξε τη σημασία κατάλληλων επιλογών πληθυσμού, τελεστών και γενεών για τη σταθερότητα των λύσεων. Η εργασία καταλήγει ότι η προσέγγιση αυτή είναι ικανή να παράγει ρεαλιστικά και προσαρμοσμένα προγράμματα διατροφής, αναδεικνύοντας τις δυνατότητες περαιτέρω επέκτασης σε πιο σύνθετα μοντέλα, επιπλέον APIs και βελτιωμένες τεχνικές προσαρμογής στις ανάγκες των χρηστών.

Development of an Intelligent Nutritional Optimization Application Using Multi-Objective Evolutionary Algorithms

Georgios Charitakis

Abstract

This thesis investigates the use of multi-objective evolutionary algorithms for generating personalized dietary plans and presents a fully developed Android application that implements this methodology. The system collects and processes user data, including personal characteristics, nutritional goals, preferences and restrictions, while retrieving detailed nutrient information from the USDA FoodData Central API, which is used to construct an extensive set of foods with realistic nutritional profiles. A multi-objective evolutionary algorithm, implemented using the Jenetics library, is then applied to identify food combinations that satisfy the user's caloric, macronutrient, and selected micronutrient targets, while simultaneously accounting for dietary restrictions, allergies and meal types. The implementation includes mechanisms for retrieving, storing and classifying foods, the use of Paging 3 for incrementally loading large datasets, and features such as plan generation, detailed nutritional views, and weight-tracking capabilities. A series of tests evaluated the reliability of the application, the performance of the evolutionary algorithm, and the quality of the generated dietary plans. The results demonstrate that the algorithm effectively approximates the nutritional objectives, achieving balanced macronutrient and micronutrient distributions, while the parameter tuning process highlighted the importance of selecting appropriate population sizes, operators, and generation counts for stable solutions. The thesis concludes that this approach can produce realistic and personalized dietary programs and identifies opportunities for future extensions involving more advanced models, additional APIs, and improved adaptation mechanisms tailored to user needs.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Αδαμίδα Παναγιώτη, για την καθοδήγηση και τη στήριξή του κατά την εκπόνηση της παρούσας εργασίας. Ευχαριστώ επίσης την οικογένειά μου και τους φίλους μου, για την αδιάκοπη ενθάρρυνση και τις χρήσιμες συζητήσεις που συνέβαλαν στην ολοκλήρωση αυτού του έργου.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xii
Κατάλογος Πινάκων.....	xiii
Συνομογραφίες.....	xiv
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Εισαγωγή.....	1
1.2 Σκοπός και στόχοι της εργασίας.....	2
1.3 Κίνητρο και σημασία της διπλωματικής εργασίας.....	2
1.4 Αντικείμενο και περιορισμοί.....	3
1.5 Δομή της εργασίας	3
1.6 Επίλογος.....	4
Κεφάλαιο 2ο: Περιγραφή προβλήματος και πεδίο εφαρμογής.....	5
2.1 Εισαγωγή.....	5
2.2 Βασικές αρχές διατροφικού προγραμματισμού	5
2.3 Μακροθρεπτικά και μικροθρεπτικά συστατικά	7
2.3.1 Μακροθρεπτικά συστατικά	7
2.3.2 Μικροθρεπτικά συστατικά	7
2.4 Εξατομικευμένη διατροφή και υπολογισμός ενεργειακών αναγκών	8
2.5 Υφιστάμενα συστήματα και εφαρμογές διατροφής	9
2.5.1 Εμπορικές εφαρμογές ευεξίας.....	9
2.5.2 Ερευνητικά και ακαδημαϊκά συστήματα.....	10
2.5.3 Συγκριτική αξιολόγηση	10
2.6 Αξιολόγηση διατροφικών APIs.....	10
2.6.1 USDA FoodData Central.....	10
2.6.2 Edamam API.....	11
2.6.3 Spoonacular API.....	12
2.6.4 Nutritionix API.....	12
2.6.5 Σύγκριση και τεκμηρίωση της επιλογής USDA.....	12

2.7	Επίλογος.....	13
Κεφάλαιο 3ο: Εξελικτικοί αλγόριθμοι.....		14
3.1	Εισαγωγή.....	14
3.2	Εισαγωγή στους εξελικτικούς αλγόριθμους.....	14
3.3	Κύρια στοιχεία ΕΑ.....	15
3.3.1	Πληθυσμός (Population).....	16
3.3.2	Επιλογή (Selection).....	16
3.3.3	Διασταύρωση (Crossover).....	16
3.3.4	Μετάλλαξη (Mutation).....	17
3.3.5	Συνάρτηση καταλληλότητας (Fitness function).....	17
3.3.6	Ισορροπία εξερεύνησης-εκμετάλλευσης και ελιτισμός.....	17
3.3.7	Κωδικοποίηση και σχεδιαστικές επιλογές.....	17
3.4	Θεωρητικό πλαίσιο εξελικτικών αλγορίθμων πολλαπλών στόχων.....	18
3.4.1	Κυριαρχία Pareto και σύνολα μη κυριαρχούμενων λύσεων.....	18
3.4.2	Διατήρηση ποικιλίας και crowding distance.....	19
3.4.3	Δημοφιλείς εξελικτικοί αλγόριθμοι πολλαπλών στόχων.....	19
3.4.4	Μετρικές αξιολόγησης μετώπων Pareto.....	21
3.4.5	Σύγκλιση, υπολογιστικός χρόνος και χειρισμός περιορισμών.....	22
3.4.6	Σύνδεση με το υπό μελέτη πρόβλημα.....	23
3.5	Εφαρμογές ΕΑ στη διατροφή και προηγούμενες μελέτες.....	23
3.6	Βιβλιοθήκες υλοποίησης ΕΑ.....	25
3.7	Βιβλιοθήκη Jenetics.....	26
3.8	Επίλογος.....	27
Κεφάλαιο 4ο: Ανάλυση και σχεδίαση της εφαρμογής.....		28
4.1	Εισαγωγή.....	28
4.2	Γενική αρχιτεκτονική της εφαρμογής.....	28
4.3	Χρησιμοποιούμενες τεχνολογίες.....	31
4.4	Απαιτήσεις χρήστη.....	33
4.5	Σχεδίαση βάσης δεδομένων και REST API.....	35
4.6	Επίλογος.....	37
Κεφάλαιο 5ο: Υλοποίηση εφαρμογής.....		38
5.1	Εισαγωγή.....	38
5.2	Αυθεντικοποίηση χρήστη.....	38
5.3	Αρχικοποίηση προφίλ χρήστη.....	41
5.3.1	Εκχώρηση βασικών στοιχείων.....	41

5.3.2	Ρύθμιση θερμίδων και διατροφικών προτιμήσεων.....	43
5.3.3	Αποθήκευση δεδομένων και δημιουργία χρήστη.....	44
5.4	Παραγωγή διατροφικού πλάνου.....	44
5.4.1	Ανάκτηση δεδομένων από το API και διαχείριση του όγκου τους.....	46
5.4.2	Λογική ταξινόμησης τροφίμων.....	48
5.4.3	Υλοποίηση εξελικτικού αλγόριθμου.....	48
5.4.4	Αποθήκευση και συγχρονισμός διατροφικού πλάνου.....	52
5.5	Αρχική σελίδα της εφαρμογής.....	53
5.6	Επισκόπηση και επεξεργασία διατροφικού πλάνου.....	56
5.7	Οθόνη παρακολούθησης και καταγραφής βάρους.....	61
5.8	Οθόνη προφίλ χρήστη.....	63
5.9	Επίλογος.....	67
Κεφάλαιο 6ο:	Δοκιμές, αποτελέσματα και αξιολόγηση.....	69
6.1	Εισαγωγή.....	69
6.2	Δοκιμές λειτουργικότητας και αξιοπιστίας της εφαρμογής.....	69
6.3	Σενάρια οριακών περιπτώσεων και μηχανισμοί αποτυχίας.....	71
6.4	Δοκιμές λειτουργικότητας και αξιολόγηση εξελικτικού αλγορίθμου.....	72
6.4.1	Εξέλιξη του εξελικτικού αλγορίθμου από την αρχική στην τελική μορφή του.....	72
6.4.2	Ρύθμιση των παραμέτρων και αξιολόγηση του εξελικτικού αλγορίθμου.....	74
6.5	Περιορισμοί της υλοποίησης.....	81
6.6	Επίλογος.....	82
Κεφάλαιο 7ο:	Συμπεράσματα και μελλοντικές επεκτάσεις.....	83
7.1	Εισαγωγή.....	83
7.2	Συνοπτική παρουσίαση συνεισφοράς της εργασίας.....	83
7.3	Βασικά ερευνητικά ευρήματα.....	84
7.4	Προτάσεις βελτίωσης της εφαρμογής και του αλγορίθμου.....	85
7.5	Επίλογος.....	86
BIBΛΙΟΓΡΑΦΙΑ.....		87
ΠΑΡΑΡΤΗΜΑ Α : USDA API /foods/search σώμα αιτήματος.....		93
ΠΑΡΑΡΤΗΜΑ Β: USDA API /foods/search σώμα απόκρισης.....		93
Παράρτημα Γ: Η συνάρτηση buildCodec.....		94

Κατάλογος Σχημάτων

Σχήμα 4.1: Η δομή του core package.....	30
Σχήμα 4.2: Υψηλού επιπέδου ροή εκτέλεσης του εξελικτικού αλγορίθμου πολλαπλών στόχων στο πλαίσιο της εφαρμογής.....	34
Σχήμα 5.1: Η οθόνη καλωσορίσματος	39
Σχήμα 5.2: Η οθόνη σύνδεσης	40
Σχήμα 5.3: Ο Credentials Manager στην οθόνη σύνδεσης	40
Σχήμα 5.4: Η οθόνη επαναφοράς κωδικού.....	41
Σχήμα 5.5: Η οθόνη δημιουργίας λογαριασμού	41
Σχήμα 5.6: Η πρώτη οθόνη της ροής αρχικοποίησης του προφίλ του χρήστη	42
Σχήμα 5.7: Ο Photo Picker	42
Σχήμα 5.8: Το πρώτο μέρος της δεύτερης οθόνης της ροής αρχικοποίησης προφίλ χρήστη.....	43
Σχήμα 5.9: Το δεύτερο μέρος της δεύτερης οθόνης της ροής αρχικοποίησης προφίλ χρήστη	43
Σχήμα 5.10: Η οθόνη επιλογής πλήθους τροφών.....	45
Σχήμα 5.11: Η οθόνη αναμονής παραγωγής του διατροφικού πλάνου.....	45
Σχήμα 5.12: Το πρώτο μέρος της αρχικής σελίδας με εστίαση στην κορυφή της	55
Σχήμα 5.13: Το δεύτερο μέρος της αρχικής σελίδας με εστίαση στο κάτω μέρος της	55
Σχήμα 5.14: Η οθόνη επισκόπησης του διατροφικού πλάνου.....	56
Σχήμα 5.15: Η οθόνη επισκόπησης του διατροφικού πλάνου με εστίαση στη λίστα τροφών μιας κατηγορίας γεύματος και στην κατηγορία unknown	56
Σχήμα 5.16: Η οθόνη αναλυτικής διατροφικής επισκόπησης των θρεπτικών στοιχείων του διατροφικού πλάνου	57
Σχήμα 5.17: Η οθόνη επεξεργασίας τροφίμου	59
Σχήμα 5.18: Η οθόνη αναζήτησης τροφίμου	59
Σχήμα 5.19: Η οθόνη αναζήτησης τροφίμου με επιλεγμένη τη ροή «My foods».....	60
Σχήμα 5.20: Η οθόνη δημιουργίας νέου τροφίμου με εστίαση στα υποχρεωτικά πεδία.....	61
Σχήμα 5.21: Η οθόνη δημιουργίας νέου τροφίμου με εστίαση στα μικροθρεπτικά στοιχεία	61
Σχήμα 5.22: Η οθόνη παρακολούθησης και καταγραφής βάρους	62
Σχήμα 5.23: Η οθόνη προφίλ χρήστη.....	63
Σχήμα 5.24: Η κορυφή της οθόνης επεξεργασίας στοιχείων προφίλ	64
Σχήμα 5.25: Η μέση της οθόνης επεξεργασίας στοιχείων προφίλ	64
Σχήμα 5.26: Το κάτω μέρος της οθόνης επεξεργασίας στοιχείων προφίλ	65
Σχήμα 5.27: Η οθόνη προφίλ χρήστη με επεκταμένη την ενότητα «BASIC CREDENTIALS».....	66
Σχήμα 5.28: Η οθόνη προφίλ χρήστη με επεκταμένη την ενότητα «OTHER OPTIONS».....	66
Σχήμα 5.29: Η οθόνη ρύθμισης της ώρας των ειδοποιήσεων	67
Σχήμα 5.30: Ο time picker dialog του Android στην εφαρμογή	67
Σχήμα 6.1: Trade-off macroScore-microScore στον τελικό πληθυσμό	77
Σχήμα 6.2: Trade-off θερμιδικής απόκλισης και macroScore.....	78
Σχήμα 6.3: Heatmap ικανοποίησης θρεπτικών στόχων	79
Σχήμα 6.4: Κατανομή τελικών τιμών αντικειμενικών συναρτήσεων	79
Σχήμα 6.5: Σύγκλιση microScore ανά γενιά	80
Σχήμα 6.6: Σύγκλιση θερμιδικής απόκλισης ανά γενιά	80

Κατάλογος Πινάκων

Πίνακας 2.1: Συντελεστές PAL.....	6
Πίνακας 3.1: Σύγκριση δημοφιλών εξελικτικών αλγορίθμων πολλαπλών στόχων	21

Συντομογραφίες

ΔΕ	Διπλωματική Εργασία
EA	Εξελικτικός/οί Αλγόριθμος/οι
AI	Artificial Intelligence
API	Application Programming Interface
BMR	Basal Metabolic Rate
DE	Differential Evolution
FAO	Food and Agriculture Organization
FDC	FoodData Central
FNDDS	Food and Nutrient Database for Dietary Studies
GA	Genetic Algorithm
GD	Generational Distance
GP	Genetic Programming
HV	Hypervolume
IDE	Integrated Development Environment
IGD	Inverted Generational Distance
IOM	Institute of Medicine
JSON	JavaScript Object Notation
JSONL	JavaScript Object Notation Lines
JVM	Java Virtual Machine
MaOEA	Many-Objective Evolutionary Algorithm
MOEA	Multi-Objective Evolutionary Algorithm
MVI	Model-View-Intent
NHANES	National Health and Nutrition Examination Survey
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PAL	Physical Activity Level
REST	Representational State Transfer
SBX	Simulated Binary Crossover
SR	Standard Reference
TDEE	Total Daily Energy Expenditure
TEF	Thermic Effect of Food

UI/UX	User Interface/User Experience
UNU	United Nations University
URL	Uniform Resource Locator
USDA	United States Department of Agriculture
WHO	World Health Organization
XML	Extensible Markup Language

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η διατροφή αποτελεί έναν από τους σημαντικότερους παράγοντες που επηρεάζουν την υγεία, την ποιότητα ζωής και τη μακροζωία του ανθρώπου. Τις τελευταίες δεκαετίες, η παγκόσμια κοινότητα αντιμετωπίζει μια συνεχώς αυξανόμενη πρόκληση που σχετίζεται με την κακή διατροφή, είτε αυτή αφορά την υπερκατανάλωση θερμίδων είτε την ελλιπή πρόσληψη απαραίτητων θρεπτικών συστατικών. Η παχυσαρκία έχει χαρακτηριστεί από τον Παγκόσμιο Οργανισμό Υγείας ως «πανδημία», καθώς συνδέεται άμεσα με την εμφάνιση χρόνιων νοσημάτων όπως ο σακχαρώδης διαβήτης τύπου 2, τα καρδιαγγειακά προβλήματα και ορισμένες μορφές καρκίνου [1]. Παράλληλα, η έλλειψη βασικών μικροθρεπτικών συστατικών, γνωστή και ως «κρυφή πείνα», επηρεάζει εκατομμύρια ανθρώπους, περιορίζοντας την ικανότητά τους να διατηρήσουν καλή υγεία και παραγωγικότητα [2].

Οι παραδοσιακές διατροφικές οδηγίες που εκδίδονται από κυβερνήσεις ή διεθνείς οργανισμούς, αν και προσφέρουν ένα γενικό πλαίσιο υγιεινής διατροφής, συχνά δεν λαμβάνουν υπόψιν τους την ποικιλομορφία του πληθυσμού. Παράγοντες όπως, η ηλικία, το φύλο, το επίπεδο φυσικής δραστηριότητας αλλά και η παρουσία αλλεργιών, ασθενειών ή άλλων διατροφικών προτιμήσεων, καθιστούν απαραίτητη μια πιο εξατομικευμένη προσέγγιση [3]. Η έννοια της εξατομικευμένης διατροφής έχει αναδειχθεί τα τελευταία χρόνια, ως απάντηση σε αυτή την ανάγκη, προτείνοντας ότι κάθε άτομο μπορεί να επωφεληθεί από διατροφικές παρεμβάσεις σχεδιασμένες ειδικά για αυτό, καθώς πρόσφατες έρευνες δείχνουν ότι μια τέτοιου είδους διατροφή, οδηγεί σε καλύτερα αποτελέσματα όσον αφορά την απώλεια βάρους, την ισορροπία των μακροθρεπτικών και μικροθρεπτικών στοιχείων και τη μακροχρόνια τήρηση ενός υγιεινού τρόπου ζωής [4].

Η ανάπτυξη εξατομικευμένων διατροφικών προγραμμάτων ωστόσο, είναι μια σύνθετη διαδικασία. Ο σωστός υπολογισμός της απαραίτητης, για κάθε άτομο, ενεργειακής πρόσληψης και η εξισορρόπηση των πρωτεϊνών, των υδατανθράκων και των λιπών, σε συνδυασμό με τις ανάγκες σε βιταμίνες, μέταλλα και ιχνοστοιχεία, αποτελεί πολυδιάστατο πρόβλημα. Στην εξίσωση αυτού του προβλήματος, προστίθενται αρκετές φορές και άλλες απαιτήσεις, όπως η γευστική ποικιλία, η οικονομική προσιτότητα ή ο περιορισμός συγκεκριμένων κατηγοριών τροφών [5]. Η χειροκίνητη δημιουργία ενός τέτοιου πλάνου, ακόμη και από έμπειρους διατροφολόγους, είναι χρονοβόρα και πολλές φορές δεν μπορεί να εξετάσει εξαντλητικά όλες τις πιθανές επιλογές.

Για να αντιμετωπιστεί αυτή η πολυπλοκότητα, η επιστημονική κοινότητα έχει στραφεί στη χρήση υπολογιστικών μεθόδων βελτιστοποίησης. Ήδη από το 1945 με τον Stigler και το κλασικό «Diet Problem», στο οποίο χρησιμοποίησε γραμμικό προγραμματισμό για να προσδιορίσει τη φθηνότερη διαίτα που πληροί τις διατροφικές ανάγκες, η διατροφική βελτιστοποίηση έχει εξελιχθεί σημαντικά [6]. Σήμερα χρησιμοποιούνται πιο προηγμένες τεχνικές, όπως η στοχαστική βελτιστοποίηση, η ασαφής λογική και οι μεταερευτικές μέθοδοι. Ανάμεσά τους, οι εξελικτικοί αλγόριθμοι (EA) έχουν αποδειχθεί ιδιαίτερα αποτελεσματικοί, καθώς μιμούνται μηχανισμούς της φυσικής εξέλιξης για να αναζητήσουν βέλτιστες λύσεις σε προβλήματα με πολλαπλά και συχνά αντικρουόμενα κριτήρια [7]. Εφαρμογές τέτοιων αλγορίθμων έχουν ήδη παρουσιαστεί στη βιβλιογραφία για τον υπολογισμό υποθερμιδικών πλάνων [8], την ανάπτυξη προγραμμάτων χαμηλού κόστους [9], αλλά και για την ενσωμάτωση παραμέτρων όπως η προτίμηση του χρήστη ή ο χρόνος προετοιμασίας [10]. Η χρήση αλγορίθμων πολλαπλών στόχων έχει καταστεί κεντρική στις πιο πρόσφατες μελέτες. Μέθοδοι όπως η NSGA-II, επιτρέπουν την εύρεση συνόλων βέλτιστων λύσεων, οι οποίες προσφέρουν στον χρήστη εναλλακτικές

διατροφικές επιλογές με διαφορετικά πλεονεκτήματα [11]. Ωστόσο, περισσότερες λεπτομέρειες σχετικά με τους ΕΑ και τις υποκατηγορίες τους, θα δοθούν σε επόμενα κεφάλαια. Παράλληλα, η αξιοποίηση βάσεων δεδομένων όπως το USDA FoodData Central επιτρέπει τον άμεσο συνδυασμό πραγματικών τροφών με ακριβή θρεπτικά προφίλ, προσδίδοντας ρεαλισμό στις προτάσεις [12].

1.2 Σκοπός και στόχοι της εργασίας

Σκοπός της παρούσας διπλωματικής εργασίας είναι η διερεύνηση της δυνατότητας αξιοποίησης σύγχρονων τεχνικών τεχνητής νοημοσύνης (AI) και ειδικότερα εξελικτικών αλγορίθμων πολλαπλών στόχων (MOEA), για τη δημιουργία ρεαλιστικών και εξατομικευμένων διατροφικών προγραμμάτων. Η εργασία επιδιώκει να γεφυρώσει τη διατροφική επιστήμη με τις υπολογιστικές μεθόδους βελτιστοποίησης, μετασχηματίζοντας τις φυσιολογικές ανάγκες, τις προτιμήσεις και τους στόχους ενός ατόμου σε υπολογίσιμα μεγέθη που επιτρέπουν την παραγωγή προσαρμοσμένων λύσεων.

Βασικός στόχος είναι η ανάπτυξη ενός ολοκληρωμένου μεθοδολογικού πλαισίου, που περιγράφει το πρόβλημα του σχεδιασμού ημερήσιου διατροφικού πλάνου ως πρόβλημα βελτιστοποίησης πολλαπλών στόχων. Στο πλαίσιο αυτό επιχειρείται η διατύπωση και ανάλυση των ενεργειακών απαιτήσεων, των μακροθρεπτικών και των επιλεγμένων μικροθρεπτικών συστατικών, καθώς και των παραμέτρων που καθορίζουν τις διατροφικές προτιμήσεις ή τους περιορισμούς του χρήστη. Παράλληλα, εξετάζονται οι κατάλληλες συναρτήσεις καταλληλότητας και οι μετρικές απόδοσης που επιτρέπουν την αξιολόγηση των παραγόμενων λύσεων ως προς τη συνέπεια και τη συμμόρφωσή τους με τους διατροφικούς στόχους. Επιμέρους στόχος είναι η υλοποίηση ενός λειτουργικού πρωτοτύπου που ενσωματώνει την προτεινόμενη μεθοδολογία και επιτρέπει την πειραματική επαλήθευση της αποτελεσματικότητάς της σε πραγματικά δεδομένα.

Η Android εφαρμογή που αναπτύχθηκε στο πλαίσιο της εργασίας λειτουργεί ως μέσο υλοποίησης και αξιολόγησης των παραπάνω στόχων. Μέσα από αυτήν εφαρμόζεται η μεθοδολογία, συλλέγονται τα αποτελέσματα και εξάγονται τα τελικά συμπεράσματα. Η αξιολόγηση της εφαρμογής πραγματοποιείται τόσο ως προς τη συμπεριφορά του αλγορίθμου όσο και ως προς τη λειτουργικότητα του συστήματος, μέσα από ελεγχόμενες δοκιμές σεναρίων χρήσης, διαχείριση οριακών περιπτώσεων και μηχανισμών αποτυχίας, καθώς και την ορθότητα της παραγωγής και παρουσίασης των διατροφικών πλάνων. Παράλληλα, η απόδοση της μεθόδου εξετάζεται με βάση τη σύγκλιση των παραγόμενων λύσεων προς τους διατροφικούς στόχους, την ισορροπία μακροθρεπτικών και επιλεγμένων μικροθρεπτικών συστατικών και τη σταθερότητα της εξελικτικής διαδικασίας. Τα πειραματικά αποτελέσματα δείχνουν ότι η προσέγγιση είναι λειτουργική και ικανή να παράγει διατροφικά πλάνα που ανταποκρίνονται στις απαιτήσεις του χρήστη, ενισχύοντας τη δυνατότητα πρακτικής αξιοποίησης της μεθόδου σε πλατφόρμες κινητών συσκευών. Επομένως, η συνεισφορά της παρούσας εργασίας δεν περιορίζεται στην ανάπτυξη ενός εργαλείου, αλλά επεκτείνεται στην τεκμηριωμένη απόδειξη ότι οι MOEA μπορούν να αποτελέσουν ένα αποτελεσματικό μέσο για τη μετάφραση των αρχών της διατροφικής επιστήμης σε εφαρμόσιμες και εξατομικευμένες προτάσεις.

1.3 Κίνητρο και σημασία της διπλωματικής εργασίας

Η επιλογή του θέματος βασίστηκε σε συνδυασμό προσωπικών και ακαδημαϊκών κινήτρων. Σε προσωπικό επίπεδο, η εργασία αντανάκλα το ενδιαφέρον για την τεχνητή νοημοσύνη, τις εξελικτικές μεθόδους, την επιστήμη της διατροφής και την ανάπτυξη εφαρμογών Android. Η σύγκλιση αυτών των πεδίων προσφέρει μια μοναδική ευκαιρία σύνθεσης γνώσεων και πρακτικών δεξιοτήτων.

Σε ακαδημαϊκό και κοινωνικό επίπεδο, η σημασία της εργασίας έγκειται στην παροχή άμεσης και οικονομικά προσιτής πρόσβασης σε έναν διατροφικό οδηγό, χωρίς την ανάγκη συνεχούς παρέμβασης διαιτολόγου και την αντίστοιχη οικονομική επιβάρυνση, γεγονός που δεν υποκαθιστά ούτε απαξιώνει τον ρόλο του επαγγελματία, αλλά προσφέρει στους χρήστες τη δυνατότητα να ακολουθήσουν πιο εύκολα, μέσα από το κινητό τους τηλέφωνο, έναν πιο υγιεινό τρόπο ζωής. Παράλληλα, η αξιοποίηση της τεχνολογίας λειτουργεί ως κίνητρο συνέπειας και τήρησης των διατροφικών συστάσεων, συμβάλλοντας έτσι στη διάδοση και ενίσχυση υγιεινών συνηθειών στην καθημερινότητα. Η δυνατότητα ανάπτυξης εργαλείων που διευκολύνουν την πρόσβαση σε εξατομικευμένες διατροφικές συστάσεις μπορεί να υποστηρίξει και την πρόληψη και να μειώσει το κόστος περίθαλψης. Παράλληλα, η εργασία ενισχύει την ερευνητική δραστηριότητα στον τομέα των συστημάτων υποστήριξης αποφάσεων και επιβεβαιώνει τον ρόλο της τεχνητής νοημοσύνης στην καθημερινή ζωή.

1.4 Αντικείμενο και περιορισμοί

Το αντικείμενο της εργασίας είναι η ανάπτυξη μιας εφαρμογής Android που δημιουργεί προγράμματα διατροφής αξιοποιώντας στοχαστικές τεχνικές βελτιστοποίησης και δεδομένα από το USDA FoodData Central API. Με τον όρο API (Application Programming Interface) εννοείται ένα σύνολο από κανόνες, συναρτήσεις και πρωτόκολλα που επιτρέπουν την επικοινωνία και ανταλλαγή δεδομένων μεταξύ διαφορετικών λογισμικών ή συστημάτων. Μέσω ενός API, μια εφαρμογή μπορεί να ζητά ή να αποστέλλει πληροφορίες από άλλη υπηρεσία, χωρίς να απαιτείται πρόσβαση στον εσωτερικό της κώδικα. Η εφαρμογή δίνει επιπλέον τη δυνατότητα στους χρήστες να καταχωρούν προσωπικά στοιχεία, στόχους, προτιμήσεις και περιορισμούς, ώστε το παραγόμενο πρόγραμμα να είναι όσο το δυνατόν πιο προσαρμοσμένο στις ανάγκες τους. Επίσης παρέχεται η δυνατότητα καταγραφής και παρακολούθησης του βάρους των χρηστών, καθώς και ένα εύχρηστο περιβάλλον, ώστε να διαχειρίζονται τη διατροφή τους και το προφίλ τους.

Ωστόσο, υπάρχουν συγκεκριμένοι περιορισμοί που πρέπει να ληφθούν υπόψιν:

- Η εφαρμογή αναπτύχθηκε αποκλειστικά για Android συσκευές.
- Χρησιμοποιήθηκε αποκλειστικά το USDA API λόγω της δωρεάν πρόσβασης και της ποιότητας των δεδομένων που παρέχει.
- Τα παραγόμενα πλάνα βασίζονται αποκλειστικά στην επίτευξη των διατροφικών στόχων του χρήστη και όχι στη γαστρονομική συμβατότητα των τροφών.
- Η διαδικασία ταξινόμησης τροφών σε κατηγορίες (vegan, vegetarian, keto friendly, dairy free) και η αναγνώριση αλλεργιογόνων βασίζονται σε λέξεις κλειδιά του τίτλου, κάτι που μπορεί να οδηγήσει σε σφάλματα.
- Η ταξινόμηση των τροφών σε γεύματα (πρωινό, μεσημεριανό, δείπνο, σνακ, επιδόρπιο) έγινε με βάση συνήθεις διατροφικές πρακτικές και όχι απόλυτα αντικειμενικά κριτήρια.

1.5 Δομή της εργασίας

Η εργασία αποτελείται από επτά κεφάλαια. Μετά το παρόν πρώτο κεφάλαιο, στο δεύτερο πραγματοποιείται η περιγραφή του προβλήματος της εξατομικευμένης διατροφής και καθορίζεται το πεδίο εφαρμογής του, μέσα από την ανάλυση βασικών εννοιών διατροφικού προγραμματισμού και την παρουσίαση των διαθέσιμων τεχνολογικών λύσεων και πηγών δεδομένων. Στο τρίτο κεφάλαιο αναλύονται οι εξελικτικοί αλγόριθμοι, οι βασικές αρχές τους και οι εφαρμογές τους στον τομέα της διατροφής, με έμφαση στους αλγορίθμους πολλαπλών στόχων. Το τέταρτο κεφάλαιο επικεντρώνεται

στην ανάλυση και τον σχεδιασμό της εφαρμογής, ενώ το πέμπτο περιγράφει αναλυτικά την υλοποίησή της. Στο έκτο κεφάλαιο παρουσιάζονται τα αποτελέσματα και η αξιολόγηση τόσο της εφαρμογής όσο και της προτεινόμενης μεθοδολογίας, ενώ στο έβδομο κεφάλαιο συνοψίζονται τα συμπεράσματα της εργασίας και προτείνονται κατευθύνσεις για μελλοντική επέκταση και βελτίωση.

1.6 Επίλογος

Η παρούσα διπλωματική εργασία πραγματεύεται τον σχεδιασμό και την υλοποίηση μιας εφαρμογής Android για τη δημιουργία εξατομικευμένων διατροφικών πλάνων, αξιοποιώντας εξελκτικούς αλγορίθμους και διατροφικά δεδομένα από αξιόπιστες εξωτερικές πηγές. Η προτεινόμενη προσέγγιση στοχεύει στην προσαρμογή των παραγόμενων πλάνων στις ανάγκες και τις προτιμήσεις του χρήστη, λαμβάνοντας υπόψη διατροφικούς στόχους, περιορισμούς και εξατομικευμένα χαρακτηριστικά. Τα πρώτα αποτελέσματα καταδεικνύουν ότι το σύστημα μπορεί να παράγει διατροφικές προτάσεις που ανταποκρίνονται με ικανοποιητικό τρόπο στις απαιτήσεις του χρήστη. Στο επόμενο κεφάλαιο παρουσιάζεται το απαραίτητο θεωρητικό υπόβαθρο σε θέματα διατροφής, καθώς και τα APIs που χρησιμοποιούνται για την απόκτηση και αξιοποίηση των σχετικών δεδομένων.

Κεφάλαιο 2ο: Περιγραφή προβλήματος και πεδίο εφαρμογής

2.1 Εισαγωγή

Το παρόν κεφάλαιο εισάγει τις βασικές έννοιες που απαιτούνται για την κατανόηση και τη σχεδίαση συστημάτων διατροφικού προγραμματισμού. Αρχικά, παρουσιάζονται οι θεμελιώδεις αρχές ενεργειακής ισορροπίας και ο ρόλος των θρεπτικών συστατικών στην κάλυψη φυσιολογικών αναγκών. Ιδιαίτερη έμφαση δίνεται στον υπολογισμό του βασικού μεταβολικού ρυθμού (Basal/Resting Metabolic Rate - BMR) και της ημερήσιας ενεργειακής δαπάνης, καθώς και στη χρήση τυποποιημένων εξισώσεων πρόβλεψης, με κεντρική θέση στη σύγχρονη βιβλιογραφία για τον τύπο Mifflin-St Jeor [13]. Η προσαρμογή στο επίπεδο φυσικής δραστηριότητας εισάγεται μέσω κατηγοριών Physical Activity Level (PAL), όπως καθορίζονται στα πρότυπα των Dietary Reference του Institute of Medicine (IOM) και σε διεθνείς τεχνικές εκθέσεις, οι οποίες παρέχουν εύρη τιμών για καθιστική, χαμηλά ενεργή, ενεργή και πολύ ενεργή καθημερινότητα [14], [15].

Στη συνέχεια, εξετάζονται τα μακροθρεπτικά (πρωτεΐνες, υδατάνθρακες, λιπαρά) και τα μικροθρεπτικά συστατικά (π.χ. ασβέστιο, σίδηρος, κάλιο, νάτριο/αλάτι, βιταμίνη C, φυτικές ίνες, σάκχαρα), τα οποία αποτελούν κρίσιμες παραμέτρους τόσο για τη διαμόρφωση στόχων όσο και για τη λειτουργία αλγοριθμικών μεθόδων βελτιστοποίησης. Η ενότητα για την εξατομικευμένη διατροφή συσχετίζει τις φυσιολογικές απαιτήσεις με τα ατομικά γνωρίσματα και εισάγει την ανάγκη μοντελοποίησης περιορισμών και προτιμήσεων, που αργότερα αξιοποιούνται στον σχεδιασμό των αλγορίθμων.

Τέλος, παρέχεται μια συνοπτική αποτύπωση των υφιστάμενων συστημάτων και APIs που υποστηρίζουν την ανάκτηση διατροφικών δεδομένων (π.χ. USDA FoodData Central, Edamam, Spoonacular, Nutritionix), με έμφαση σε ζητήματα οικονομικού κόστους, πολιτικών caching και ποιότητας των δεδομένων. Το caching, αναφέρεται στη διαδικασία προσωρινής αποθήκευσης των δεδομένων που ανακτώνται από ένα API σε τοπικό ή ενδιάμεσο επίπεδο, ώστε να μειώνεται η ανάγκη για επαναλαμβανόμενα αιτήματα προς τον διακομιστή και να επιτυγχάνεται ταχύτερη απόκριση και εξοικονόμηση πόρων ή η δυνατότητα ανάγνωσης δεδομένων όταν δεν υπάρχει σύνδεση στο διαδίκτυο. Το USDA FoodData Central αναδεικνύεται ως ενοποιημένη υποδομή με πέντε τύπους δεδομένων, κατάλληλη για εφαρμογές που απαιτούν πλήρη και διαλειτουργική πληροφόρηση για προφίλ τροφίμων [16].

2.2 Βασικές αρχές διατροφικού προγραμματισμού

Ο διατροφικός προγραμματισμός βασίζεται στην αρχή της ενεργειακής ισορροπίας, σύμφωνα με την οποία το σωματικό βάρος παραμένει σταθερό όταν η ενεργειακή πρόσληψη από τις τροφές ισούται με την ενεργειακή δαπάνη του οργανισμού. Η ενεργειακή δαπάνη περιλαμβάνει τρεις κύριες συνιστώσες: τον βασικό μεταβολικό ρυθμό (BMR), την ενεργειακή δαπάνη φυσικής δραστηριότητας και τη θερμική επίδραση της τροφής (Thermic Effect of Food - TEF) [17]. Ο BMR εκφράζει την ενέργεια που απαιτείται για τη διατήρηση των βασικών λειτουργιών του σώματος σε κατάσταση ηρεμίας, όπως η αναπνοή, η κυκλοφορία του αίματος και η ρύθμιση της θερμοκρασίας. Αποτελεί συνήθως το 60-75% της ημερήσιας ενεργειακής δαπάνης, ενώ το υπόλοιπο ποσοστό κατανέμεται ανάλογα με το επίπεδο δραστηριότητας και τη διατροφή [18]. Για την εκτίμηση του BMR έχουν αναπτυχθεί διάφορες εξισώσεις πρόβλεψης. Μεταξύ αυτών, η Mifflin-St Jeor θεωρείται σήμερα η πιο αξιόπιστη για ενήλικες υγιή άτομα, καθώς στηρίχθηκε σε εκτεταμένο δείγμα 498 ατόμων και έδειξε μικρότερη απόκλιση από

Κεφάλαιο 2

τις μετρούμενες τιμές ενεργειακής δαπάνης συγκριτικά με παλαιότερους τύπους, όπως ο Harris-Benedict [13]. Η εξίσωση, είναι διαφορετική για τα δύο φύλα και για τους άνδρες εκφράζεται ως εξής:

$$BMR = 10 \times \text{βάρος (kg)} + 6.25 \times \text{ύψος (cm)} - 5 \times \text{ηλικία} + 5 \quad (2.1)$$

ενώ για τις γυναίκες:

$$BMR = 10 \times \text{βάρος (kg)} + 6.25 \times \text{ύψος (cm)} - 5 \times \text{ηλικία} - 161 \quad (2.2)$$

Ο βασικός μεταβολισμός προσαρμόζεται στη συνέχεια σύμφωνα με το επίπεδο φυσικής δραστηριότητας μέσω του παράγοντα δραστηριότητας PAL. Οι τιμές που χρησιμοποιούνται διεθνώς και ενσωματώνονται σε διατροφικά μοντέλα, συμπεριλαμβανομένης της παρούσας εφαρμογής, προέρχονται από τις οδηγίες του IOM και των Food and Agriculture Organization (FAO), World Health Organization (WHO) και United Nations University (UNU). Η επιτροπή αυτή αποτελεί διεθνές επιστημονικό όργανο που εκδίδει τεχνικές αναφορές για τις ανθρώπινες ενεργειακές απαιτήσεις, καθορίζοντας τα παγκόσμια πρότυπα υπολογισμού του BMR και του PAL [14], [15]. Οι τιμές αυτές κυμαίνονται από 1.2 για καθιστική ζωή έως 1.9 για πολύ υψηλή δραστηριότητα, όπως παρουσιάζεται στον Πίνακα 2.1.

Πίνακας 2.1: Συντελεστές PAL

Επίπεδο δραστηριότητας	Περιγραφή	Συντελεστής PAL
Καθιστικός (Sedentary)	Ελάχιστη φυσική δραστηριότητα, καθιστική εργασία	1,2
Ελαφρώς δραστήριος (Light)	Μέτρια φυσική δραστηριότητα 1-3 φορές/εβδομάδα	1,375
Μέτρια δραστήριος (Moderate)	Τακτική άσκηση 3-5 φορές/εβδομάδα	1,55
Ενεργός (Active)	Έντονη φυσική δραστηριότητα 6-7 φορές/εβδομάδα	1,725
Πολύ ενεργός (Athlete)	Επαγγελματίας αθλητής ή καθημερινή έντονη άσκηση	1,9

Η συνολική ημερήσια ενεργειακή δαπάνη (Total Daily Energy Expenditure - TDEE) υπολογίζεται πολλαπλασιάζοντας τον BMR με τον κατάλληλο συντελεστή δραστηριότητας και προσθέτοντας ή αφαιρώντας θερμίδες ανάλογα με τον επιδιωκόμενο στόχο (απώλεια, διατήρηση ή αύξηση βάρους). Η διαφορά 500kcal/ημέρα αντιστοιχεί κατά προσέγγιση σε μεταβολή 0,5kg σωματικού λίπους ανά εβδομάδα, γεγονός που τη καθιστά πρακτική τιμή για εφαρμογές διατροφικού προγραμματισμού [19].

Η μετατροπή της ποσότητας τροφής σε θερμιδικό ισοδύναμο βασίζεται στα ενεργειακά ισοδύναμα Atwater, τα οποία χρησιμοποιούνται διεθνώς για την εκτίμηση των θερμίδων που αποδίδουν τα μακροθρεπτικά συστατικά. Σύμφωνα με αυτό το σύστημα, κάθε γραμμάριο πρωτεΐνης αποδίδει κατά μέσο όρο 4 kcal, κάθε γραμμάριο υδατανθράκων 4 kcal, και κάθε γραμμάριο λιπαρών 9 kcal [20]. Τα ισοδύναμα αυτά αποτελούν τον ακρογωνιαίο λίθο του υπολογισμού της ενεργειακής πρόσληψης και έχουν ενσωματωθεί σε διεθνείς βάσεις δεδομένων τροφίμων, όπως το USDA FoodData Central [16]. Η ορθή εφαρμογή των παραπάνω αρχών επιτρέπει τη σχεδίαση διατροφικών προγραμμάτων που

εξισορροπούν την ενεργειακή πρόσληψη με τις ανάγκες του οργανισμού, θέτοντας τη βάση για την ανάπτυξη εξελικτικών μοντέλων βελτιστοποίησης που χρησιμοποιούνται στην παρούσα εργασία.

2.3 Μακροθρεπτικά και μικροθρεπτικά συστατικά

Οι τροφές που αποτελούν τη διατροφή ενός ατόμου καθορίζονται από τη σύνθεση των μακροθρεπτικών και μικροθρεπτικών τους συστατικών. Τα μακροθρεπτικά στοιχεία αποτελούν τις κύριες πηγές ενέργειας του οργανισμού, ενώ τα μικροθρεπτικά, όπως οι βιταμίνες, τα μέταλλα και τα ιχνοστοιχεία, είναι απαραίτητα για τη ρύθμιση των μεταβολικών διεργασιών, την ομαλή λειτουργία του ανοσοποιητικού συστήματος και τη διατήρηση της ομοιόστασης του οργανισμού [21]. Η σωστή αναλογία και ισορροπία ανάμεσά τους είναι καθοριστική για την κάλυψη των ενεργειακών και φυσιολογικών αναγκών του ανθρώπινου σώματος. Στις επόμενες υποενότητες εξετάζονται χωριστά τα μακροθρεπτικά και τα μικροθρεπτικά συστατικά, ώστε να αναδειχθεί ο ιδιαίτερος ρόλος που διαδραματίζει κάθε κατηγορία στη συνολική θρεπτική επάρκεια.

2.3.1 Μακροθρεπτικά συστατικά

Οι πρωτεΐνες, αποτελούν βασικό δομικό υλικό για τα κύτταρα, τους ιστούς και τα ένζυμα του οργανισμού. Κάθε γραμμάριο πρωτεΐνης αποδίδει περίπου 4 kcal, ενώ η επαρκής πρόσληψή τους είναι κρίσιμη για τη διατήρηση της μυϊκής μάζας, ειδικά κατά τη διάρκεια προγραμμάτων απώλειας βάρους ή έντονης σωματικής δραστηριότητας. Οι ημερήσιες ανάγκες σε πρωτεΐνη υπολογίζονται περίπου στα 0.8-1.0 g/kg σωματικού βάρους για τον γενικό πληθυσμό, αυξανόμενες έως και 1.6-2.0 g/kg σωματικού βάρους σε αθλητές ή άτομα με αυξημένες απαιτήσεις [22]. Οι υδατάνθρακες, αποτελούν την κύρια πηγή ενέργειας για τον εγκέφαλο και τους μύς και παρέχουν επίσης 4 kcal ανά γραμμάριο. Διακρίνονται σε απλούς (μονοσακχαρίτες και δισακχαρίτες) και σύνθετους (πολυσακχαρίτες), με τους δεύτερους να θεωρούνται πιο ωφέλιμους λόγω της σταδιακής απορρόφησης και της επίδρασής τους στη γλυκαιμική ισορροπία [23]. Τα λιπαρά είναι απαραίτητα για την απορρόφηση λιποδιαλυτών βιταμινών (A, D, E, K), τη δομή των κυτταρικών μεμβρανών και τη ρύθμιση ορμονικών λειτουργιών. Κάθε γραμμάριο λίπους αποδίδει 9 kcal, δηλαδή περισσότερο από το διπλάσιο των πρωτεϊνών ή των υδατανθράκων. Σημαντική είναι και η αναλογία των διαφορετικών τύπων λιπαρών. Τα πολυακόρεστα και μονοακόρεστα λιπαρά (όπως τα ω-3 και ω-6) προάγουν την καρδιαγγειακή υγεία, ενώ τα κορεσμένα και trans λιπαρά συνδέονται με αυξημένο καρδιαγγειακό κίνδυνο [24].

Η κατανομή των μακροθρεπτικών στοιχείων στη συνολική ημερήσια πρόσληψη ενέργειας προτείνεται από τον WHO και το IOM ως εξής: 45-65% της ενέργειας από υδατάνθρακες, 20-35% από λίπη και 10-35% από πρωτεΐνες [25]. Οι αναλογίες αυτές λαμβάνονται υπόψη στον διατροφικό προγραμματισμό, καθώς επηρεάζουν τόσο το θερμιδικό ισοζύγιο όσο και την ποιότητα της διατροφής.

2.3.2 Μικροθρεπτικά συστατικά

Σε αντίθεση με τα μακροθρεπτικά, τα μικροθρεπτικά συστατικά δεν παρέχουν ενέργεια, αλλά είναι απαραίτητα σε μικρές ποσότητες για την ομαλή λειτουργία του οργανισμού. Ελλείψεις ή υπερβολές μπορούν να προκαλέσουν σοβαρές επιπτώσεις στην υγεία [26]. Μεταξύ των μικροθρεπτικών που αξιολογούνται στην παρούσα εργασία συγκαταλέγονται τα εξής:

- **Σίδηρος (Fe):** απαραίτητος για τη σύνθεση της αιμοσφαιρίνης και τη μεταφορά οξυγόνου. Η ανεπάρκειά του οδηγεί σε σιδηροπενική αναιμία, ιδιαίτερα συχνή σε γυναίκες αναπαραγωγικής ηλικίας [27].

- **Ασβέστιο (Ca):** βασικό συστατικό των οστών και των δοντιών. Συμμετέχει επίσης στη μυϊκή συστολή και την ενδοκυττάρια σηματοδότηση. Η ανεπαρκής πρόσληψη αυξάνει τον κίνδυνο οστεοπόρωσης [28].
- **Κάλιο (K):** συμβάλλει στη ρύθμιση της αρτηριακής πίεσης και της νευρομυϊκής λειτουργίας. Η επαρκής πρόσληψη καλίου συνδέεται με μειωμένο καρδιαγγειακό κίνδυνο [29].
- **Νάτριο (Na) - Αλάτι:** αν και απαραίτητο για την ισορροπία των υγρών και τη νευρική λειτουργία, η υπερβολική πρόσληψη νατρίου σχετίζεται με υπέρταση. Ο WHO συνιστά περιορισμό κάτω από 2g νατρίου ημερησίως (ισοδύναμο με 5g αλατιού) [30].
- **Βιταμίνη C:** ισχυρό αντιοξειδωτικό, απαραίτητο για τη σύνθεση κολλαγόνου και την απορρόφηση σιδήρου φυτικής προέλευσης [31].
- **Φυτικές ίνες:** αν και τεχνικά δεν αποτελούν βιταμίνες ή μέταλλα, εντάσσονται συχνά στις μικροθρεπτικές παραμέτρους λόγω της προστατευτικής τους δράσης. Προάγουν την καλή λειτουργία του εντέρου και συνδέονται με μειωμένο κίνδυνο εμφάνισης διαβήτη τύπου 2 και καρδιαγγειακών νοσημάτων [32].
- **Σάκχαρα:** παρότι αποτελούν μορφή υδατανθράκων, η υπερβολική πρόσληψή τους σχετίζεται με αυξημένο κίνδυνο παχυσαρκίας και τερηδόνας. Ο WHO συνιστά η πρόσληψη ελεύθερων σακχάρων να μην υπερβαίνει το 10% της ημερήσιας ενέργειας [33].

Η επάρκεια και η ισορροπία όλων των παραπάνω μικροθρεπτικών συστατικών συνιστούν δείκτη ποιοτικής διατροφής. Οι πληροφορίες αυτές αξιοποιούνται στην εφαρμογή για τη διαμόρφωση στόχων βελτιστοποίησης, ώστε τα παραγόμενα διατροφικά πλάνα να προσεγγίζουν τις διεθνείς συστάσεις τόσο σε επίπεδο μακροθρεπτικών όσο και μικροθρεπτικών στοιχείων. Η επιλογή των συγκεκριμένων μικροθρεπτικών συστατικών (σίδηρος, ασβέστιο, κάλιο, νάτριο, βιταμίνη C, φυτικές ίνες και σάκχαρα) έγινε βάσει τριών βασικών κριτηρίων. Πρώτον, αποτελούν θεμελιώδη θρεπτικά στοιχεία που σχετίζονται άμεσα με κρίσιμες φυσιολογικές λειτουργίες και τη διατήρηση της υγείας, όπως η αιματοποίηση, η οστική πυκνότητα, η ρύθμιση της πίεσης και η αντιοξειδωτική άμυνα. Δεύτερον, πρόκειται για παραμέτρους που παρακολουθούνται συστηματικά από διεθνείς οργανισμούς, όπως ο WHO και ο FAO, καθιστώντας εφικτή την τεκμηριωμένη αξιολόγηση της πρόσληψής τους βάσει διεθνών προτύπων [25], [26], [30], [33]. Τρίτον, τα συγκεκριμένα μικροθρεπτικά διαθέτουν υψηλή διαθεσιμότητα δεδομένων στις επίσημες διατροφικές βάσεις, ιδίως στο USDA FoodData Central, το οποίο παρέχει πλήρη και αξιόπιστα προφίλ για αυτά τα συστατικά. Η επιλογή τους, επομένως, επιτρέπει στην εφαρμογή να ισορροπεί μεταξύ επιστημονικής εγκυρότητας και πρακτικής υλοποιησιμότητας.

2.4 Εξατομικευμένη διατροφή και υπολογισμός ενεργειακών αναγκών

Η έννοια της εξατομικευμένης διατροφής, όπως περιληπτικά αναφέρθηκε στο Κεφάλαιο 1, βασίζεται στην αναγνώριση ότι κάθε άτομο ανταποκρίνεται διαφορετικά στα ίδια διατροφικά ερεθίσματα, λόγω διαφορών στο μεταβολισμό, στη γενετική σύσταση, στις ορμονικές λειτουργίες και στο μικροβίωμα του εντέρου [22]. Οι παράγοντες αυτοί, σε συνδυασμό με το περιβάλλον και τον τρόπο ζωής, καθορίζουν τη διατροφική απόκριση και τις πραγματικές ανάγκες σε ενέργεια και θρεπτικά συστατικά. Η παραδοσιακή προσέγγιση της ενιαίας διατροφικής σύστασης, αν και χρήσιμη για τη χάραξη δημόσιας πολιτικής, συχνά αποτυγχάνει να λάβει υπόψη αυτή τη βιολογική ποικιλομορφία, οδηγώντας σε συστάσεις που δεν ανταποκρίνονται πλήρως στις ανάγκες όλων των ατόμων. Η εξατομίκευση των διατροφικών παρεμβάσεων προκύπτει μέσα από τη συστηματική συλλογή και ανάλυση πληροφοριών που περιλαμβάνουν φυσιολογικά χαρακτηριστικά, δείκτες σύστασης σώματος και ενεργειακών αναγκών, καθώς και δεδομένα που σχετίζονται με τις διατροφικές συνήθειες, τις προτιμήσεις και τις ανοχές σε τρόφιμα. Στόχος είναι η ανάπτυξη προγραμμάτων που όχι μόνο επιτυγχάνουν ενεργειακή ισορροπία, αλλά ταυτόχρονα προάγουν τη βέλτιστη πρόσληψη μακροθρεπτικών και μικροθρεπτικών στοιχείων σύμφωνα με τα διεθνή πρότυπα του WHO και του USDA [16], [25]. Η διασύνδεση αυτών

των στοιχείων επιτρέπει την προσαρμογή των συστάσεων στο προφίλ του κάθε ατόμου, ενισχύοντας την αποτελεσματικότητα των παρεμβάσεων.

Η πρόοδος στην επεξεργασία δεδομένων και στις τεχνολογίες τεχνητής νοημοσύνης έχει συμβάλει καθοριστικά στην πραγμάτωση αυτής της εξατομίκευσης. Σύγχρονες ερευνητικές κατευθύνσεις ενσωματώνουν υπολογιστικά μοντέλα στη διαδικασία διατροφικού σχεδιασμού, επιτρέποντας τον συνδυασμό των φυσιολογικών χαρακτηριστικών και των στόχων του κάθε ατόμου με πραγματικά δεδομένα τροφίμων, ώστε οι διατροφικές προτάσεις να προσαρμόζονται δυναμικά στις ανάγκες του [34]. Έτσι, η εξατομικευμένη διατροφή μετατρέπεται σταδιακά από θεωρητική έννοια σε εφαρμόσιμη πρακτική, ικανή να υποστηρίξει μια πιο βιώσιμη και ρεαλιστική προσέγγιση στη διαχείριση της υγείας και του σωματικού βάρους. Παράλληλα, έχει αποδειχθεί ότι τα προγράμματα που λαμβάνουν υπόψη τα ιδιαίτερα χαρακτηριστικά του κάθε ατόμου οδηγούν σε υψηλότερα ποσοστά συμμόρφωσης και σε πιο σταθερή υιοθέτηση υγιεινών διατροφικών προτύπων [35].

2.5 Υφιστάμενα συστήματα και εφαρμογές διατροφής

Η πρακτική εφαρμογή της εξατομικευμένης διατροφής περιλαμβάνει τη μετατροπή των αρχών της διατροφικής επιστήμης σε εργαλεία καθημερινής καθοδήγησης και αυτοδιαχείρισης. Στο πλαίσιο αυτό, έχουν αναπτυχθεί ψηφιακά εργαλεία και πληροφοριακά συστήματα που αξιοποιούν δεδομένα χρηστών για τη δημιουργία προσαρμοσμένων πλάνων, εφαρμόζοντας αλγοριθμικές μεθόδους που μετατρέπουν τις φυσιολογικές και διατροφικές παραμέτρους σε υπολογιστικά μοντέλα. Αν και κάποιες από τις τεχνικές πτυχές των διαδικασιών αυτών θα παρουσιαστούν αναλυτικότερα στο επόμενο κεφάλαιο, αξίζει να επισημανθεί ότι τα συστήματα αυτά στηρίζονται σε τεκμηριωμένες διατροφικές εξισώσεις και διεθνείς βάσεις δεδομένων, εξασφαλίζοντας ακρίβεια και αξιοπιστία. Με τον τρόπο αυτό, η τεχνολογία λειτουργεί ως σύνδεσμος ανάμεσα στη θεωρία της εξατομίκευσης και την πρακτική της εφαρμογή, προσφέροντας μια δυναμική, προσαρμόσιμη εμπειρία για τον χρήστη [36]. Παράλληλα, η ψηφιακή εξατομίκευση επιτρέπει τη συνεχή παρακολούθηση και αναθεώρηση των διατροφικών στόχων, δημιουργώντας έναν κύκλο ανατροφοδότησης που ενισχύει τη μακροπρόθεσμη αποτελεσματικότητα των προτάσεων. Ο συνδυασμός επιστημονικής τεκμηρίωσης και προσαρμοστικότητας των ψηφιακών συστημάτων συνιστά τον πυρήνα των σύγχρονων προσεγγίσεων στη διατροφική βελτιστοποίηση, οι οποίες διακρίνονται σε δύο κύριες κατηγορίες: τις εμπορικές εφαρμογές ευεξίας και τα ερευνητικά ή ακαδημαϊκά συστήματα βελτιστοποίησης διατροφής.

2.5.1 Εμπορικές εφαρμογές ευεξίας

Οι πιο διαδεδομένες εφαρμογές ευεξίας, όπως το MyFitnessPal, το Lifesum, το Yazio και το EatThisMuch, έχουν συμβάλει σημαντικά στη διάδοση της αυτο-παρακολούθησης (self-monitoring) της διατροφής. Η λειτουργία τους βασίζεται συνήθως σε μεγάλα διατροφικά ημερολόγια, βάσεις δεδομένων τροφίμων και αυτοματοποιημένες εκτιμήσεις θερμίδων, οι οποίες υπολογίζονται από καταχωρήσεις του χρήστη. Παρέχουν επίσης στατιστικά γραφήματα, ειδοποιήσεις και δυνατότητα στόχων για απώλεια ή διατήρηση βάρους. Ωστόσο, τα περισσότερα από αυτά τα συστήματα δεν πραγματοποιούν βελτιστοποίηση της διατροφής με βάση θρεπτικά κριτήρια, αλλά λειτουργούν περιγραφικά. Πιο συγκεκριμένα, καταγράφουν και αναλύουν τις επιλογές του χρήστη, χωρίς να παράγουν αυτόματα εξατομικευμένα πλάνα. Επιπλέον, οι εμπορικές εφαρμογές χρησιμοποιούν κλειστές βάσεις δεδομένων, συχνά χωρίς δυνατότητα επέκτασης ή διασύνδεσης με επίσημες πηγές, όπως το USDA API. Σε αρκετές περιπτώσεις, η πρόσβαση σε πλήρη δεδομένα ή διατροφικά προγράμματα παρέχεται μόνο μέσω συνδρομητικών πακέτων, περιορίζοντας τη χρήση για ερευνητικούς ή αναπτυξιακούς σκοπούς [37].

2.5.2 Ερευνητικά και ακαδημαϊκά συστήματα

Αντίθετα, τα ερευνητικά συστήματα εστιάζουν στην υπολογιστική βελτιστοποίηση της διατροφής και τη μοντελοποίηση διατροφικών προβλημάτων μέσω τεχνικών AI, μεταξύ των οποίων βρίσκονται και οι ΕΑ. Στην εργασία της Koroušić Seljak [3], αναπτύχθηκε ένα υπολογιστικό σύστημα σχεδιασμού μενού, το οποίο χρησιμοποιεί γραμμικό προγραμματισμό για τη δημιουργία ισορροπημένων πλάνων βασισμένων σε διατροφικές οδηγίες. Παρόμοιες προσεγγίσεις με χρήση γενετικών αλγορίθμων προτάθηκαν από τους Kołodziejczyk και Przybyłek [4], οι οποίοι αυτοματοποίησαν τη διαδικασία επιλογής τροφίμων με βάση θερμοειδικούς και μακροθρεπτικούς περιορισμούς. Εξέλιξη αυτών των προσεγγίσεων αποτέλεσαν τα συστήματα βελτιστοποίησης πολλαπλών στόχων, όπως των Türkmenoğlu et al. [7], που ενσωμάτωσαν παραμέτρους κόστους, χρόνου προετοιμασίας και διατροφικής ποιότητας, αξιοποιώντας Many-Objective Evolutionary Algorithms (MaOEA). Αντίστοιχα, οι Jia et al. [11] πρότειναν ένα σύστημα αυτόματης δημιουργίας πλάνων διατροφής με χρήση γενετικών αλγορίθμων (GA), ενσωματώνοντας δεδομένα χρηστών και API τροφίμων. Οι προσεγγίσεις αυτές απέδειξαν ότι οι αλγοριθμικές μέθοδοι μπορούν να επιτύχουν ρεαλιστικές και προσαρμόσιμες λύσεις, με υψηλό βαθμό εξατομίκευσης και αποδοτικότητας.

2.5.3 Συγκριτική αξιολόγηση

Η κύρια διαφορά μεταξύ εμπορικών και ερευνητικών εφαρμογών έγκειται στη φιλοσοφία χρήσης. Οι πρώτες στοχεύουν στην καθημερινή αυτο-παρακολούθηση και την απλοποιημένη καθοδήγηση, ενώ οι δεύτερες εστιάζουν στην υπολογιστική βελτιστοποίηση και στην επίτευξη ισορροπίας μεταξύ θερμίδων, μακροθρεπτικών και μικροθρεπτικών στοιχείων, καθώς και άλλων παραμέτρων. Στο πλαίσιο της παρούσας εργασίας, η ανάπτυξη της εφαρμογής επιχειρεί να συνδυάσει τα πλεονεκτήματα και των δύο προσεγγίσεων, επιτρέποντας τη δημιουργία μιας καινοτόμου πλατφόρμας εξατομικευμένου διατροφικού προγραμματισμού.

2.6 Αξιολόγηση διατροφικών APIs

Η ανάπτυξη εφαρμογών διατροφικού προγραμματισμού προϋποθέτει, όπως προαναφέρθηκε, την πρόσβαση σε αξιόπιστα δεδομένα τροφίμων, τα οποία περιλαμβάνουν λεπτομερείς πληροφορίες για τα μακροθρεπτικά και μικροθρεπτικά συστατικά. Για τον σκοπό αυτό, έχουν δημιουργηθεί διεθνώς διάφορα διατροφικά APIs, τα οποία επιτρέπουν την αυτόματη ανάκτηση τέτοιων δεδομένων από βάσεις πληροφοριών τροφίμων. Στην ενότητα αυτή πραγματοποιείται μια συγκριτική ανάλυση τεσσάρων ευρέως χρησιμοποιούμενων APIs, των USDA FoodData Central, Edamam, Spoonacular και Nutritionix, με βάση τα κριτήρια που αναφέρθηκαν στην εισαγωγική ενότητα του κεφαλαίου.

2.6.1 USDA FoodData Central

Το USDA FoodData Central (FDC) αποτελεί το επίσημο API του U.S. Department of Agriculture και χρησιμοποιείται εκτενώς στην επιστημονική έρευνα και στις διατροφικές εφαρμογές [38]. Πρόκειται για μια πλήρως ανοιχτή και δωρεάν πλατφόρμα, η οποία παρέχει πρόσβαση σε μια ενιαία βάση δεδομένων που συνδυάζει πέντε διαφορετικές κατηγορίες πληροφοριών: Foundation Foods, Food and Nutrient Database for Dietary Studies (FNDDS), Standard Reference (SR), Branded Foods και Experimental Foods [39]. Η πρόσβαση στα δεδομένα επιτρέπεται χωρίς κόστος, γεγονός που το καθιστά ιδιαίτερα ελκυστικό για ακαδημαϊκές και ερευνητικές χρήσεις.

Κάθε μία από αυτές τις κατηγορίες εξυπηρετεί διαφορετικό σκοπό μέσα στο ευρύτερο πλαίσιο της διατροφικής τεκμηρίωσης. Η κατηγορία Foundation Foods περιλαμβάνει βασικά τρόφιμα, ακατέργαστα ή ελάχιστα επεξεργασμένα, για τα οποία παρέχονται εκτενή στοιχεία σύνθεσης, μέθοδοι

ανάλυσης, πληροφορίες για τη γεωγραφική προέλευση και μεταδεδομένα σχετικά με την πηγή των δειγμάτων [40]. Η κατηγορία SR Legacy περιλαμβάνει την ιστορική βάση δεδομένων του USDA, η οποία συγκεντρώνει τιμές θρεπτικών συστατικών από εργαστηριακές αναλύσεις, βιβλιογραφικές αναφορές και υπολογιστικά μοντέλα. Αν και δεν ενημερώνεται πλέον, παραμένει σημείο αναφοράς για σύγκριση [39]. Η βάση FNDDS, περιλαμβάνει δεδομένα που χρησιμοποιούνται σε διατροφικές έρευνες του προγράμματος “What We Eat in America”, το οποίο αποτελεί μέρος του National Health and Nutrition Examination Survey (NHANES), συνδέοντας τις καταγραφές τροφίμων με τις αντίστοιχες θρεπτικές τιμές [41]. Η κατηγορία Branded Foods περιλαμβάνει δεδομένα από εμπορικά προϊόντα και τις ετικέτες τους, όπως αυτά υποβάλλονται από τους ίδιους τους κατασκευαστές ή από δημόσιες βάσεις εμπορικών προϊόντων. Τέλος, η κατηγορία Experimental Foods συγκεντρώνει τρόφιμα που έχουν μελετηθεί σε ερευνητικά ή πειραματικά πλαίσια, παρέχοντας συμπληρωματικά δεδομένα για ειδικές συνθήκες καλλιέργειας, μεταποίησης ή εμπλουτισμού [39].

Ένα από τα βασικά πλεονεκτήματα του FDC είναι η τεκμηριωμένη επιστημονική ακρίβεια των δεδομένων που παρέχει [16]. Κάθε καταχώρηση συνοδεύεται από μεθοδολογία ανάλυσης, πηγές και τιμές αβεβαιότητας, κάτι που προσφέρει διαφάνεια και αξιοπιστία. Παράλληλα, η τεκμηρίωση του API είναι εκτενής και ακολουθεί τη λογική RESTful αρχιτεκτονικής [38], επιτρέποντας εύκολη ενσωμάτωση σε εφαρμογές. Ένα RESTful API (Representational State Transfer API) είναι ένας τύπος διαδικτυακού API που ακολουθεί τις αρχές της αρχιτεκτονικής REST, δηλαδή της «Μεταφοράς Αναπαραστάσιμης Κατάστασης». Βασίζεται σε τυποποιημένες HTTP μεθόδους, όπως GET, POST, PUT και DELETE και ανταλλάσσει δεδομένα συνήθως σε μορφή JSON ή XML. Η αρχιτεκτονική REST είναι ελαφριά, ευέλικτη και ευρέως χρησιμοποιούμενη, καθώς επιτρέπει τη γρήγορη και αποδοτική επικοινωνία μεταξύ εφαρμογών που βρίσκονται σε διαφορετικά περιβάλλοντα. Επιπλέον, το USDA επιτρέπει την προσωρινή αποθήκευση (caching) των δεδομένων [39]. Αυτό μειώνει το φορτίο των δικτυακών κλήσεων και καθιστά τη λειτουργία των εφαρμογών πιο αποδοτική.

Ωστόσο, ένα μειονέκτημα του FDC είναι ότι η πλειονότητα των τροφίμων που περιλαμβάνει αφορά προϊόντα της αμερικανικής αγοράς, με αποτέλεσμα να υπάρχει περιορισμένη κάλυψη για διεθνή προϊόντα ή παραδοσιακές συνταγές άλλων χωρών. Επιπλέον, το API δεν υποστηρίζει εγγενώς φίλτρα για κατηγορίες όπως vegan, gluten-free ή keto, κάτι που πρέπει να επιλυθεί σε επίπεδο εφαρμογής, εάν αυτός ο διαχωρισμός είναι επιθυμητός, μέσω λέξεων-κλειδιών ή πρόσθετης ταξινόμησης.

2.6.2 Edamam API

Το Edamam Nutrition API είναι μια εμπορική πλατφόρμα που προσφέρει δεδομένα διατροφικής ανάλυσης για τρόφιμα και συνταγές [42]. Παρέχει λεπτομερείς πληροφορίες για μακροθρεπτικά και μικροθρεπτικά στοιχεία, ενώ υποστηρίζει και ανάλυση φυσικής γλώσσας, επιτρέποντας την αυτόματη αναγνώριση συστατικών μέσα από περιγραφές συνταγών. Η δομή του είναι επίσης RESTful, με καλά οργανωμένα endpoints, κάτι που το καθιστά ιδιαίτερα φιλικό προς προγραμματιστές [43].

Ωστόσο, το Edamam διατίθεται με διαφορετικά εμπορικά πακέτα, όπου οι δυνατότητες και οι περιορισμοί εξαρτώνται από το κόστος [43]. Το βασικό, δωρεάν πλάνο προσφέρει περιορισμένο αριθμό αιτημάτων ανά λεπτό και δεν επιτρέπει την αποθήκευση των δεδομένων. Η δυνατότητα caching ενεργοποιείται μόνο στα επαγγελματικά πακέτα, τα οποία απαιτούν συνδρομή. Αυτοί οι περιορισμοί καθιστούν το Edamam λιγότερο πρακτικό για εφαρμογές που χρειάζονται συνεχή πρόσβαση σε διατροφικές πληροφορίες χωρίς οικονομική επιβάρυνση. Παρόλα αυτά, το API παραμένει ιδιαίτερα δημοφιλές λόγω της καλής τεκμηρίωσης, της εύκολης ενσωμάτωσης και της αξιοπιστίας των υπολογισμών του για διατροφική ανάλυση.

2.6.3 Spoonacular API

Το Spoonacular API αποτελεί μια ευέλικτη πλατφόρμα που συνδυάζει δεδομένα τροφίμων, συνταγών και διατροφικής ανάλυσης [44]. Εκτός από τη δυνατότητα ανάκτησης πληροφοριών για μακροθρεπτικά και μικροθρεπτικά στοιχεία, υποστηρίζει λειτουργίες όπως αναζήτηση συνταγών, δημιουργία μενού και προτάσεις γευμάτων. Αυτή η πολυλειτουργικότητα το καθιστά ιδιαίτερα ελκυστικό για εφαρμογές που στοχεύουν στη συνολική εμπειρία διατροφής και μαγειρικής.

Παρά τα πλεονεκτήματά του, το Spoonacular διαθέτει επίσης περιορισμούς. Το δωρεάν πλάνο χρήσης επιτρέπει μόνο έναν μικρό αριθμό ημερήσιων αιτημάτων και δεν προσφέρει πλήρη πρόσβαση στις δυνατότητες caching [45]. Η πλήρης λειτουργικότητα απαιτεί μετάβαση σε συνδρομητικό πακέτο, το οποίο μπορεί να αυξήσει σημαντικά το κόστος για μια ανεξάρτητη ή εκπαιδευτική εφαρμογή. Επιπλέον, σε ορισμένες περιπτώσεις, τα δεδομένα θρεπτικών στοιχείων δεν προέρχονται από εργαστηριακές μετρήσεις αλλά βασίζονται σε εκτιμήσεις ή μέσους όρους, μειώνοντας την ακρίβεια των αποτελεσμάτων [44]. Παρ' όλα αυτά, το API παραμένει αξιόπιστο εργαλείο για εφαρμογές που εστιάζουν περισσότερο στη γαστρονομική διάσταση της διατροφής.

2.6.4 Nutritionix API

Το Nutritionix API είναι μια υπηρεσία που παρέχει πρόσβαση σε μια εκτενή βάση δεδομένων τροφίμων και πιάτων, η οποία εν μέρει βασίζεται σε στοιχεία που υποβάλλονται από τους ίδιους τους χρήστες [46]. Αυτό το μοντέλο συλλογικής συνεισφοράς (crowdsourced), έχει ως αποτέλεσμα την ύπαρξη μιας πολύ μεγάλης ποικιλίας δεδομένων, συμπεριλαμβανομένων προϊόντων γνωστών εμπορικών σημάτων, καθώς και γευμάτων από εστιατόρια. Το API προσφέρει επιπλέον εργαλεία για την καταγραφή της διατροφής, τη δημιουργία ημερολογίων γευμάτων και την ενσωμάτωση σε εφαρμογές φυσικής κατάστασης.

Ωστόσο, το μοντέλο συλλογής δεδομένων του Nutritionix ενέχει κινδύνους ασυνέπειας και ανακρίβειας, καθώς η ποιότητα των πληροφοριών εξαρτάται από τις καταχωρήσεις των χρηστών και όχι αποκλειστικά από επίσημες πηγές. Αν και το API διαθέτει δωρεάν επίπεδο πρόσβασης, αυτό περιορίζεται σε βασική λειτουργικότητα, ενώ οι πιο προχωρημένες δυνατότητες, όπως η αποθήκευση και η ανάλυση μεγάλου όγκου δεδομένων, παρέχονται μόνο μέσω πληρωμένων πακέτων [47]. Εξαιτίας αυτών των παραγόντων, το Nutritionix είναι πιο κατάλληλο για εφαρμογές που επικεντρώνονται στην καταγραφή και ανάλυση της καθημερινής πρόσληψης τροφής, παρά για εκείνες που απαιτούν αυστηρή βελτιστοποίηση και ακρίβεια στα θρεπτικά δεδομένα.

2.6.5 Σύγκριση και τεκμηρίωση της επιλογής USDA

Η συγκριτική αξιολόγηση των τεσσάρων APIs ανέδειξε σημαντικές διαφορές ως προς το κόστος, τις πολιτικές caching, την πληρότητα των δεδομένων και την αξιοπιστία των πληροφοριών. Από την ανάλυση προκύπτει ότι το USDA FoodData Central υπερτερεί συνολικά έναντι των Edamam, Spoonacular και Nutritionix για τις ανάγκες της εφαρμογής που αναπτύσσεται στο πλαίσιο της παρούσας ΔΕ, καθώς συνδυάζει την επιστημονική εγκυρότητα με τη δωρεάν, ανοιχτή και ευέλικτη πρόσβαση. Το Edamam παρέχει αναλυτικά δεδομένα και είναι εύχρηστο στην ενσωμάτωση, ωστόσο το δωρεάν πλάνο του έχει σημαντικούς περιορισμούς στις κλήσεις και στην αποθήκευση δεδομένων, κάτι που αυξάνει το λειτουργικό κόστος [43]. Το Spoonacular προσφέρει πλούσιες δυνατότητες αλλά στηρίζεται σε εκτιμήσεις και επιβάλλει περιορισμούς ανάλογα με το συνδρομητικό πακέτο [45], ενώ το Nutritionix παρέχει μεγάλο όγκο δεδομένων, αλλά με μειωμένη ακρίβεια λόγω της χρήσης crowdsourced πληροφοριών [46]. Αντίθετα, το USDA FoodData Central παρέχει δωρεάν, πλήρη και

αξιόπιστα δεδομένα, επιτρέπει την προσωρινή αποθήκευση, και διαθέτει υψηλό επίπεδο τεκμηρίωσης και διαφάνειας [42], [39]. Παρότι επικεντρώνεται κυρίως σε προϊόντα της αμερικανικής αγοράς, η ποιότητα και η επιστημονική εγκυρότητα των δεδομένων του το καθιστούν την πλέον κατάλληλη επιλογή για την ανάπτυξη εφαρμογών διατροφικού προγραμματισμού που απαιτούν ακρίβεια, επεκτασιμότητα και σταθερότητα. Για τους λόγους αυτούς, το USDA API επιλέχθηκε ως κύρια πηγή δεδομένων για την εφαρμογή που αναπτύχθηκε στο πλαίσιο της παρούσας εργασίας, εξασφαλίζοντας τη βέλτιστη ισορροπία μεταξύ αξιοπιστίας, χρηστικότητας και κόστους.

2.7 Επίλογος

Το παρόν κεφάλαιο θέτει τις βάσεις για την κατανόηση του προβλήματος που πραγματεύεται η ΔΕ, οριοθετώντας το πεδίο εφαρμογής της εξατομικευμένης διατροφής σε ένα ρεαλιστικό και υλοποιήσιμο πλαίσιο. Μέσα από τον συνδυασμό διατροφικών αρχών, φυσιολογικών παραμέτρων και διαθέσιμων πηγών δεδομένων, διαμορφώνεται ένα συνεκτικό υπόβαθρο στο οποίο οι ανάγκες και οι στόχοι του χρήστη μπορούν να εκφραστούν με σαφή και μετρήσιμο τρόπο. Το κεφάλαιο λειτουργεί ως ένα εννοιολογικό σημείο αναφοράς που καθορίζει ποιες πτυχές της διατροφικής επιστήμης λαμβάνονται υπόψη και με ποιον τρόπο μπορούν να αξιοποιηθούν σε ένα υπολογιστικό σύστημα.

Με τον τρόπο αυτό, προσδιορίζεται το πλαίσιο μέσα στο οποίο η δημιουργία διατροφικών πλάνων αντιμετωπίζεται ως πρόβλημα σύνθετης λήψης αποφάσεων, όπου συνυπάρχουν πολλαπλές απαιτήσεις, περιορισμοί και πρακτικές παράμετροι εφαρμογής. Η ανάγκη ταυτόχρονης ικανοποίησης ενεργειακών στόχων, θρεπτικής επάρκειας και εξατομικευμένων προτιμήσεων αναδεικνύει τον πολυδιάστατο χαρακτήρα του προβλήματος. Στο επόμενο κεφάλαιο παρουσιάζονται οι εξελικτικοί αλγόριθμοι και οι αρχές της βελτιστοποίησης πολλαπλών στόχων, οι οποίοι παρέχουν το κατάλληλο θεωρητικό και αλγοριθμικό πλαίσιο για την αποτελεσματική αντιμετώπιση του προβλήματος όπως αυτό ορίστηκε.

Κεφάλαιο 3ο: Εξελικτικοί αλγόριθμοι

3.1 Εισαγωγή

Όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο, η ανάπτυξη εξατομικευμένων διατροφικών προγραμμάτων απαιτεί την επίλυση ενός σύνθετου προβλήματος βελτιστοποίησης, στο οποίο πρέπει να ληφθούν υπόψη πολλαπλά και συχνά αντικρουόμενα κριτήρια, όπως η ενεργειακή ισορροπία, τα μακροθρεπτικά και μικροθρεπτικά όρια, αλλά και οι προσωπικές προτιμήσεις του χρήστη. Οι κλασικές μεθοδολογίες, όπως ο γραμμικός ή μη γραμμικός προγραμματισμός, δυσκολεύονται να αποδώσουν ικανοποιητικές λύσεις όταν το πρόβλημα αποκτά στοχαστικό ή πολυδιάστατο χαρακτήρα. Η ανάγκη για πιο ευέλικτες προσεγγίσεις οδήγησε στη χρήση μεταερευτικών μεθόδων και, ειδικότερα, των Εξελικτικών Αλγόριθμων.

Οι ΕΑ αντλούν έμπνευση από τους μηχανισμούς της φυσικής εξέλιξης [48] και έχουν αποδειχθεί ιδιαίτερα αποτελεσματικοί στην αναζήτηση βέλτιστων λύσεων σε προβλήματα με πολύπλοκο χώρο αναζήτησης και πολλαπλούς στόχους. Στο πεδίο της διατροφής, η εφαρμογή τους έχει επιτρέψει τη δημιουργία προγραμμάτων που δεν περιορίζονται σε μία μοναδική βέλτιστη λύση, αλλά προσφέρουν σύνολα εναλλακτικών πλάνων, καθένα από τα οποία ικανοποιεί διαφορετικούς διατροφικούς στόχους ή περιορισμούς. Έτσι, επιτυγχάνεται μια πιο ρεαλιστική και εξατομικευμένη προσέγγιση, όπως απαιτεί η σύγχρονη αντίληψη της διατροφικής επιστήμης.

Στο παρόν κεφάλαιο εξετάζονται οι θεωρητικές βάσεις των εξελικτικών αλγορίθμων, με έμφαση τόσο στα βασικά δομικά τους στοιχεία όσο και στη λειτουργία τους σε προβλήματα βελτιστοποίησης πολλαπλών στόχων. Αρχικά παρουσιάζονται οι θεμελιώδεις μηχανισμοί λειτουργίας των εξελικτικών αλγορίθμων, όπως ο πληθυσμός, η επιλογή, η διασταύρωση, η μετάλλαξη και η συνάρτηση καταλληλότητας. Στη συνέχεια, αναπτύσσεται εκτενώς το θεωρητικό πλαίσιο των εξελικτικών αλγορίθμων πολλαπλών στόχων, περιλαμβάνοντας έννοιες όπως η κυριαρχία Pareto, η διατήρηση ποικιλίας, οι μηχανισμοί σύγκλισης και οι μετρικές αξιολόγησης μετώπων Pareto. Παράλληλα, παρουσιάζονται και συγκρίνονται χαρακτηριστικοί αλγόριθμοι της κατηγορίας αυτής, με στόχο την κατανόηση των σχεδιαστικών επιλογών που επηρεάζουν τη συμπεριφορά και την απόδοσή τους. Τέλος, γίνεται αναφορά σε εφαρμογές των εξελικτικών αλγορίθμων στον τομέα της διατροφής και παρουσιάζεται η βιβλιοθήκη Jenetics, η οποία αξιοποιήθηκε για την υλοποίηση του αλγορίθμου της παρούσας διπλωματικής εργασίας. Με τον τρόπο αυτό, το κεφάλαιο θέτει το ολοκληρωμένο θεωρητικό και τεχνολογικό υπόβαθρο που απαιτείται για την κατανόηση της μεθοδολογίας και της υλοποίησης που ακολουθούν στα επόμενα κεφάλαια.

3.2 Εισαγωγή στους εξελικτικούς αλγόριθμους

Οι Εξελικτικοί Αλγόριθμοι (ΕΑ) αποτελούν μία κατηγορία στοχαστικών μεθόδων βελτιστοποίησης που βασίζονται στις αρχές της φυσικής επιλογής και της γενετικής εξέλιξης, όπως περιγράφηκαν από τον Charles Darwin. Η βασική τους αρχή είναι ότι οι υποψήφιας λύσεις ενός προβλήματος μπορούν να θεωρηθούν ως άτομα ενός πληθυσμού, τα οποία εξελίσσονται μέσα από επαναληπτικές διαδικασίες επιλογής, διασταύρωσης και μετάλλαξης, με σκοπό τη διατήρηση και ενίσχυση των πιο κατάλληλων λύσεων [48]. Κάθε υποψήφια λύση αξιολογείται μέσω μιας συνάρτησης καταλληλότητας (fitness function), η οποία εκτιμά τον βαθμό στον οποίο ικανοποιεί τα κριτήρια του προβλήματος [49]. Η λειτουργία των ΕΑ ακολουθεί τη φιλοσοφία της φυσικής προσαρμογής, δηλαδή οι λύσεις που παρουσιάζουν υψηλότερη καταλληλότητα έχουν αυξημένη πιθανότητα να επιλεγούν για αναπαραγωγή, μεταφέροντας τα χαρακτηριστικά τους στις επόμενες γενιές, ενώ οι λιγότερο επιτυχημένες

απορρίπτονται. Με αυτόν τον τρόπο, ο πληθυσμός συγκλίνει σταδιακά προς ένα σύνολο λύσεων που πλησιάζουν το βέλτιστο. Παράλληλα, οι τελεστές μετάλλαξης συμβάλλουν στη διατήρηση της ποικιλίας και στην αποφυγή στασιμότητας κατά την εξελικτική διαδικασία [50], [51]. Περαιτέρω λεπτομέρειες σχετικά με τους τεχνικούς όρους των ΕΑ θα δοθούν στη συνέχεια.

Οι πρώτες μορφές ΕΑ εμφανίστηκαν τη δεκαετία του 1950, με τις Εξελικτικές Στρατηγικές (Evolution Strategies) των Rechenberg και Schwefel, τον GA του Holland και τον Γενετικό Προγραμματισμό (Genetic Programming - GP) του Koza [48], [52]. Κοινό τους στοιχείο είναι η χρήση ενός πληθυσμού υποψήφιων λύσεων, που αναπαράγονται με στοχαστικούς τελεστές, διατηρώντας ποικιλία και βελτιώνοντας σταδιακά την ποιότητα των λύσεων μέσω επαναληπτικών κύκλων αξιολόγησης και επιλογής. Οι αλγόριθμοι αυτοί αποτέλεσαν τη βάση για τις σύγχρονες προσεγγίσεις που χρησιμοποιούνται σήμερα σε πληθώρα επιστημονικών πεδίων, από τη μηχανική και τη ρομποτική έως τη βιοπληροφορική και τη διατροφική βελτιστοποίηση [45].

Στη σύγχρονη βιβλιογραφία, οι ΜΟΕΑ έχουν αποκτήσει σημαντική θέση, καθώς αντιμετωπίζουν προβλήματα με περισσότερες από μία αντικειμενικές συναρτήσεις. Με τον όρο αντικειμενική συνάρτηση (objective function) εννοούμε τη μαθηματική έκφραση που αποτυπώνει το ζητούμενο κριτήριο ή στόχο προς βελτιστοποίηση, για παράδειγμα, την ελαχιστοποίηση του κόστους, τη μέγιστη κάλυψη θερμιδικών αναγκών ή τη βελτιστοποίηση της θρεπτικής ισορροπίας. Σε προβλήματα με πολλούς στόχους, κάθε αντικειμενική συνάρτηση εκφράζει ένα διαφορετικό κριτήριο, το οποίο μπορεί να συγκρούεται ή να συνεργάζεται με τα υπόλοιπα. Σε αντίθεση με τις μονοκριτηριακές προσεγγίσεις, οι ΜΟΕΑ επιδιώκουν την παραγωγή ενός συνόλου βέλτιστων λύσεων, γνωστού ως Pareto front, το οποίο εκφράζει διαφορετικούς συμβιβασμούς μεταξύ των στόχων και θα εξηγηθεί στη συνέχεια. Οι γνωστοί αλγόριθμοι NSGA-II και NSGA-III έχουν καθιερωθεί ως πρότυπα στον τομέα αυτό, επιτυγχάνοντας υψηλή σταθερότητα και ποικιλία αποτελεσμάτων [53], [54].

Πρόσφατα, η εξέλιξη των μεθόδων βελτιστοποίησης πολλαπλών στόχων οδήγησε στην ανάπτυξη των MaOEA, οι οποίοι επεκτείνουν τη λογική των ΜΟΕΑ σε προβλήματα με περισσότερους από τέσσερις ή πέντε στόχους. Η βασική τους διαφορά έγκειται στο γεγονός ότι οι ΜΟΕΑ εστιάζουν κυρίως στη βελτιστοποίηση λίγων, συγκρουόμενων αντικειμενικών συναρτήσεων, ενώ οι MaOEA σχεδιάζονται για να διαχειρίζονται έναν πολύ μεγαλύτερο αριθμό στόχων, αντιμετωπίζοντας έτσι ζητήματα κλιμάκωσης και αυξημένης πολυπλοκότητας. Οι MaOEA αξιοποιούν τεχνικές αναφοράς σημείων και προσαρμοστικού περιορισμού για να διαχειριστούν τον αυξημένο αριθμό μη κυριαρχούμενων λύσεων, επιτυγχάνοντας πιο ρεαλιστικές και ισορροπημένες λύσεις σε εφαρμογές με πολυδιάστατα κριτήρια [55].

3.3 Κύρια στοιχεία ΕΑ

Όπως προαναφέρθηκε, οι Εξελικτικοί Αλγόριθμοι αποτελούν προσεγγίσεις που βασίζονται σε αρχές εξέλιξης και προσαρμογής. Για να κατανοηθεί πληρέστερα ο τρόπος λειτουργίας τους, είναι απαραίτητο να παρουσιαστούν τα κύρια συστατικά τους και ο ρόλος που επιτελεί καθένα από αυτά κατά τη διάρκεια της εξελικτικής διαδικασίας. Οι ΕΑ υλοποιούν μια πληθυσμιακή διαδικασία αναζήτησης, στην οποία πολλές υποψήφιες λύσεις εξελίσσονται ταυτόχρονα. Κάθε λύση κωδικοποιείται σε χρωμόσωμα (π.χ. δυαδικό διάνυσμα, ακέραιοι, πραγματικοί), και ένα σύνολο τέτοιων χρωμοσωμάτων συγκροτεί τον πληθυσμό της τρέχουσας γενιάς. Η εξέλιξη προκύπτει από επαναλαμβανόμενους κύκλους αξιολόγησης-επιλογής-αναπαραγωγής-μετάλλαξης, μέχρι να ικανοποιηθεί κάποιο κριτήριο τερματισμού (π.χ. μέγιστος αριθμός γενεών ή σύγκλιση της επίδοσης) [48], [49], [50]. Η πληθυσμιακή αναζήτηση

επιτρέπει ταυτόχρονη εξερεύνηση πολλών περιοχών του χώρου λύσεων και περιορίζει τον εγκλωβισμό σε τοπικά άκρα, σε αντίθεση με ντετερμινιστικές μεθόδους [50], [51], [55].

3.3.1 Πληθυσμός (Population)

Ο πληθυσμός είναι η κατάσταση του αλγόριθμου σε κάθε γενιά. Το μέγεθος πληθυσμού ελέγχει το εύρος εξερεύνησης. Μεγάλοι πληθυσμοί αυξάνουν την ποικιλία αλλά έχουν μεγαλύτερο υπολογιστικό κόστος, ενώ μικρότεροι πληθυσμοί είναι ταχύτεροι αλλά κινδυνεύουν από πρόωρη σύγκλιση [49], [50]. Η επιλογή αναπαράγει υποψήφιες λύσεις με πιθανότητα ανάλογη της ποιότητάς τους, αλλά η διατήρηση ποικιλίας (diversity) στον πληθυσμό είναι κρίσιμη για αποφυγή στασιμότητας. Τεχνικές όπως το crowding και η ελίτ επιλογή (elitism) βοηθούν να διατηρηθούν καλές λύσεις ενώ παράλληλα εξερευνώνται νέες [53], [55].

3.3.2 Επιλογή (Selection)

Ο μηχανισμός επιλογής καθορίζει ποιες λύσεις θα «αναπαραχθούν», με άλλα λόγια επιλέγονται τα χρωμοσώματα που θα συμμετάσχουν στη δημιουργία του επόμενου πληθυσμού. Κλασικές τεχνικές είναι η fitness-proportionate, η tournament και οι rank-based μέθοδοι. Στη fitness-proportionate, η πιθανότητα επιλογής είναι ανάλογη της τιμής fitness, ωστόσο μπορεί να «υποφέρει» όταν οι διαφορές fitness είναι πολύ μεγάλες. Συνήθης τρόπος υλοποίησης αυτής της ιδέας είναι η roulette wheel selection. Η tournament, επιλέγει το καλύτερο από τυχαίο υποσύνολο λύσεων και παρέχει σταθερό έλεγχο της «πιστικότητας» της επιλογής μέσω του μεγέθους του τουρνουά. Οι κατατακτικές μέθοδοι μειώνουν την εξάρτηση από την απόλυτη κλίμακα του fitness και ενισχύουν τη σταθερότητα [49], [50].

Στους MOEA, η ποιότητα δεν είναι μονοδιάστατη. Χρησιμοποιείται η έννοια της μη-κυριαρχίας (Pareto dominance) για να κατασκευαστούν μέτωπα λύσεων (fronts). Ο NSGA-II συνδυάζει μη-κυριαρχούμενη ταξινόμηση (non-dominated sorting) και απόσταση συμφορήσης (crowding distance) ώστε να ισορροπεί η αριστεία και η ποικιλία κατά την επιλογή [53]. Για προβλήματα με πολλούς στόχους, ο NSGA-III εισάγει σημεία αναφοράς (reference points) ώστε να διατηρεί καλή διασπορά στο υψηλών διαστάσεων μέτωπο [54]. Αναλυτικότερη περιγραφή των επιμέρους μηχανισμών των MOEA θα παρουσιαστεί στη συνέχεια του κεφαλαίου, καθώς στο παρόν σημείο η εκτενής ανάλυσή τους θα απέκλινε από τον στόχο της υποενότητας.

3.3.3 Διασταύρωση (Crossover)

Η διασταύρωση παράγει απογόνους συνδυάζοντας «γονεϊκή» πληροφορία. Σε δυαδική κωδικοποίηση, συνήθεις τελεστές είναι οι single point, two point και uniform crossover. Στο single-point crossover, επιλέγεται ένα τυχαίο σημείο στο χρωμόσωμα και ανταλλάσσονται τα τμήματα μετά από αυτό μεταξύ των δύο γονέων, δημιουργώντας δύο νέους απογόνους. Στο two-point crossover, επιλέγονται δύο σημεία τομής και ανταλλάσσεται το ενδιάμεσο τμήμα, διατηρώντας έτσι περισσότερη ποικιλία στη δομή των απογόνων. Τέλος, στο uniform crossover, κάθε γονίδιο του χρωμοσώματος έχει ανεξάρτητη πιθανότητα να προέλθει από τον έναν ή τον άλλον γονέα, ενισχύοντας τη στοχαστικότητα της αναπαραγωγής [49], [50].

Σε πραγματικές μεταβλητές, ο Simulated Binary Crossover (SBX) μιμείται τη συμπεριφορά της two-point διασταύρωσης στον συνεχή χώρο. Ο SBX χρησιμοποιεί έναν παράγοντα διανομής (distribution index), γνωστό και ως παράμετρο έντασης (intensity parameter), ο οποίος καθορίζει πόσο κοντά στους γονείς θα βρίσκονται οι απόγονοι. Υψηλές τιμές οδηγούν σε απογόνους παρόμοιους με τους γονείς (ήπιος ανασυνδυασμός), ενώ χαμηλές τιμές επιτρέπουν μεγαλύτερες αποκλίσεις και επομένως

αυξημένη ποικιλία [49], [50], [53]. Η διασταύρωση προάγει την εκμετάλλευση (exploitation) συνδυάζοντας επιτυχημένα δομικά στοιχεία, αλλά χωρίς επαρκή τυχαιότητα μπορεί να οδηγήσει σε σύγκλιση γύρω από υποβέλτιστες περιοχές [49], [50].

3.3.4 Μετάλλαξη (Mutation)

Η μετάλλαξη εισάγει μικρές τυχαίες τροποποιήσεις, διατηρώντας την εξερεύνηση (exploration). Στη δυαδική κωδικοποίηση, η bit-flip αντιστρέφει τυχαία bits με κάποια δεδομένη πιθανότητα. Στις πραγματικές μεταβλητές χρησιμοποιούνται γκαουσιανή ή πολυωνυμική μετάλλαξη, που προσαρμόζουν τις τιμές με ελεγχόμενο πλάτος διασποράς [49], [50], [53]. Ο ρυθμός μετάλλαξης πρέπει να είναι αρκετά μικρός για να μην καταστρέφει συστηματικά καλά μοτίβα, αλλά αρκετά μεγάλος ώστε να αποτρέπει την ομογενοποίηση του πληθυσμού [49], [50].

3.3.5 Συνάρτηση καταλληλότητας (Fitness function)

Η fitness function αποτυπώνει πόσο καλή είναι μια λύση σε σχέση με τους στόχους. Στα προβλήματα ενός στόχου είναι συνήθως άμεση (π.χ. ελαχιστοποίηση ή μεγιστοποίηση κόστους). Στα προβλήματα πολλαπλών στόχων, η αξιολόγηση βασίζεται σε μη-κυριαρχία και δείκτες ποικιλίας (π.χ. crowding distance στον NSGA-II) αντί για μια μεμονωμένη αριθμητική τιμή [53], [55]. Οι περιορισμοί (constraints) μπορούν να ενσωματωθούν με διάφορους τρόπους:

- Ποινές (penalties): προσθήκη όρου που επιβαρύνει το fitness ανάλογα με την παραβίαση.
- Επισκευές (repairs): τροποποίηση λύσεων ώστε να καταστούν εφικτές.
- Κανόνες εφικτότητας στην επιλογή: προτιμώνται εφικτές λύσεις. Μεταξύ δύο ανεφάρμοστων, προτιμάται εκείνη με μικρότερη παραβίαση [49], [50], [55].

3.3.6 Ισορροπία εξερεύνησης-εκμετάλλευσης και ελιτισμός

Κεντρική αρχή των EA είναι η ισορροπία ανάμεσα στην αναζήτηση νέων περιοχών (εξερεύνηση) και στη βελτίωση υποσχόμενων περιοχών (εκμετάλλευση). Ο ελιτισμός διασφαλίζει ότι οι καλύτερες λύσεις μιας γενιάς μεταφέρονται άθικτες στην επόμενη, επιταχύνοντας τη σύγκλιση χωρίς να καταρρέει η ποικιλία αν συνδυάζεται με κατάλληλους τελεστές και παραμέτρους [49], [53]. Στους MOEA, ο ελιτισμός υλοποιείται φυσικά μέσω της αρχαιοθέτησης των μη-κυριαρχούμενων λύσεων [53], [55]. Παρόλα αυτά, η υπερβολική χρήση του μπορεί να οδηγήσει σε πρόωρη σύγκλιση και απώλεια ποικιλίας στον πληθυσμό, καθώς οι ίδιες λύσεις αναπαράγονται συνεχώς εις βάρος της εξερεύνησης νέων περιοχών [49], [50]. Για την αποφυγή αυτού του φαινομένου, συχνά χρησιμοποιούνται μηχανισμοί διατήρησης ποικιλίας, όπως το crowding distance στους NSGA-II ή τεχνικές ελεγχόμενης επιλογής (controlled selection), που συνδυάζουν ελιτισμό με στοχαστικά στοιχεία ώστε να εξισορροπείται η εξερεύνηση και η εκμετάλλευση [53], [55].

3.3.7 Κωδικοποίηση και σχεδιαστικές επιλογές

Η κωδικοποίηση (binary/integer/real/permutation) πρέπει να αντανακλά τα δομικά χαρακτηριστικά του προβλήματος, ώστε οι τελεστές να παράγουν έγκυρους και χρήσιμους απογόνους. Για συνεχείς ποσότητες, όπως μεγέθη μερίδων, προτιμώνται πραγματικές αναπαραστάσεις με SBX/πολυωνυμική μετάλλαξη, για επιλογή από καταλόγους, για παράδειγμα σύνολα τροφίμων, χρησιμοποιούνται ακέραιες ή δυαδικές αναπαραστάσεις ή και περμουτασιακές μορφές όταν παίζει ρόλο η σειρά [49], [50], [53]. Η σωστή κωδικοποίηση συχνά μειώνει την ανάγκη για περίπλοκες ποινές, διότι ενσωματώνει δομικούς περιορισμούς εξαρχής [50].

3.4 Θεωρητικό πλαίσιο εξελικτικών αλγορίθμων πολλαπλών στόχων

Όπως προέκυψε από την ανάλυση του προβλήματος του διατροφικού προγραμματισμού στα προηγούμενα κεφάλαια, ο σχεδιασμός ενός εξατομικευμένου διατροφικού πλάνου δεν αποσκοπεί στη βελτιστοποίηση ενός μοναδικού μεγέθους, αλλά στην ταυτόχρονη ικανοποίηση πολλαπλών στόχων, οι οποίοι συχνά βρίσκονται μεταξύ τους σε σύγκρουση. Η επίτευξη του επιθυμητού θερμιδικού στόχου, η ισορροπία μακροθρεπτικών και μικροθρεπτικών συστατικών, καθώς και η συμμόρφωση με διατροφικούς περιορισμούς ή προτιμήσεις, συνιστούν ανεξάρτητες απαιτήσεις που δεν μπορούν να εκφραστούν επαρκώς μέσω μιας ενιαίας αντικειμενικής συνάρτησης. Η βελτίωση μιας εκ των παραπάνω παραμέτρων ενδέχεται να επιφέρει επιδείνωση σε κάποια άλλη, γεγονός που καθιστά το πρόβλημα εγγενώς πρόβλημα πολλαπλών στόχων. Δεδομένου ότι η παρούσα εργασία βασίζεται στη χρήση εξελικτικών αλγορίθμων πολλαπλών στόχων για τη μοντελοποίηση και επίλυση του προβλήματος, κρίνεται αναγκαία η πληρέστερη θεωρητική ανάλυση του πλαισίου αυτού του είδους της βελτιστοποίησης και των βασικών εννοιών που το διέπουν.

3.4.1 Κυριαρχία Pareto και σύνολα μη κυριαρχούμενων λύσεων

Στα προβλήματα βελτιστοποίησης πολλαπλών στόχων, η έννοια της βέλτιστης λύσης διαφοροποιείται ουσιαστικά σε σχέση με τα προβλήματα ενός μόνο στόχου. Η ταυτόχρονη ύπαρξη περισσότερων του ενός στόχων, καθιστά αδύνατη την ύπαρξη μιας λύσης που να είναι καλύτερη από όλες τις υπόλοιπες ως προς κάθε στόχο. Για τον λόγο αυτό, η αξιολόγηση και η σύγκριση των λύσεων βασίζεται στην έννοια της κυριαρχίας Pareto, η οποία αποτελεί θεμελιώδη αρχή στη θεωρία της βελτιστοποίησης πολλαπλών στόχων [56]. Σύμφωνα με τον ορισμό της κυριαρχίας Pareto, μία λύση A θεωρείται ότι κυριαρχεί μία άλλη λύση B όταν η A δεν είναι χειρότερη από τη B σε όλους τους στόχους και ταυτόχρονα είναι αυστηρά καλύτερη σε τουλάχιστον έναν από αυτούς. Εάν δεν υφίσταται καμία άλλη λύση που να κυριαρχεί μια δεδομένη λύση, τότε η λύση αυτή χαρακτηρίζεται ως μη κυριαρχούμενη. Το σύνολο όλων των μη κυριαρχούμενων λύσεων ενός προβλήματος σχηματίζει το λεγόμενο σύνολο Pareto, ενώ η απεικόνισή του στον χώρο των στόχων ονομάζεται μέτωπο Pareto [57].

Το μέτωπο Pareto αποτυπώνει τους δυνατούς συμβιβασμούς μεταξύ των στόχων και προσφέρει μια σφαιρική εικόνα των εναλλακτικών λύσεων που μπορεί να παραχθούν από έναν αλγόριθμο πολλαπλών στόχων. Κάθε σημείο του μετώπου αντιπροσωπεύει μια λύση για την οποία δεν είναι δυνατή η περαιτέρω βελτίωση ενός στόχου χωρίς ταυτόχρονη επιδείνωση κάποιου άλλου. Στο πλαίσιο εφαρμογών όπως ο διατροφικός σχεδιασμός, η ιδιότητα αυτή είναι ιδιαίτερα σημαντική, καθώς επιτρέπει την παραγωγή διαφορετικών αποδεκτών λύσεων που ικανοποιούν τις ίδιες βασικές απαιτήσεις, αλλά δίνουν έμφαση σε διαφορετικές προτεραιότητες, όπως η αυστηρότερη συμμόρφωση σε θρεπτικούς στόχους ή η μεγαλύτερη ευελιξία στις επιλογές τροφίμων.

Σε ορισμένες περιπτώσεις, η αυστηρή εφαρμογή της κυριαρχίας Pareto μπορεί να οδηγήσει στη δημιουργία πολύ μεγάλων συνόλων μη κυριαρχούμενων λύσεων, ιδίως όταν ο αριθμός των στόχων αυξάνεται. Για την αντιμετώπιση αυτού του φαινομένου έχουν προταθεί παραλλαγές της έννοιας της κυριαρχίας, όπως η «ε-κυριαρχία», η οποία εισάγει ένα κατώφλι ανοχής στις διαφορές μεταξύ των στόχων. Με τον τρόπο αυτό επιτυγχάνεται ομαδοποίηση παρόμοιων λύσεων και περιορισμός του πλήθους των μη κυριαρχούμενων λύσεων, χωρίς να αλλοιώνεται ουσιαστικά η δομή των συμβιβασμών που περιγράφει το μέτωπο Pareto [58].

Η έννοια της κυριαρχίας Pareto και ο σχηματισμός συνόλων μη κυριαρχούμενων λύσεων αποτελούν τη θεωρητική βάση πάνω στην οποία στηρίζονται οι εξελικτικοί αλγόριθμοι πολλαπλών στόχων. Οι αλγόριθμοι αυτοί αξιοποιούν την κατάταξη των λύσεων σε μέτωπα Pareto προκειμένου να

καθοδηγήσουν τη διαδικασία επιλογής και εξέλιξης του πληθυσμού, διατηρώντας παράλληλα ένα σύνολο εναλλακτικών λύσεων που προσεγγίζουν αποτελεσματικά το μέτωπο Pareto.

3.4.2 Διατήρηση ποικιλίας και crowding distance

Η προσέγγιση του μετώπου Pareto αποτελεί βασικό στόχο των εξελικτικών αλγορίθμων πολλαπλών στόχων, ωστόσο εξίσου σημαντική είναι και η διατήρηση επαρκούς ποικιλίας στο σύνολο των παραγόμενων λύσεων. Η αποκλειστική εστίαση στη σύγκλιση μπορεί να οδηγήσει σε πρόωρη συγκέντρωση του πληθυσμού σε περιορισμένες περιοχές του χώρου λύσεων, με αποτέλεσμα την απώλεια εναλλακτικών συμβιβασμών και την ελλιπή αναπαράσταση του μετώπου Pareto. Για τον λόγο αυτόν, η διατήρηση ποικιλίας θεωρείται κρίσιμη συνιστώσα της απόδοσης ενός αλγορίθμου πολλαπλών στόχων και αποτελεί αντικείμενο εκτενούς μελέτης στη σχετική βιβλιογραφία [56], [57].

Η έννοια της ποικιλίας αναφέρεται στον βαθμό κατά τον οποίο οι λύσεις ενός πληθυσμού κατανέμονται ομοιόμορφα κατά μήκος του μετώπου Pareto και καλύπτουν διαφορετικές περιοχές του χώρου στόχων. Ένας πληθυσμός με υψηλή ποικιλία προσφέρει πληρέστερη εικόνα των δυνατών συμβιβασμών, επιτρέποντας στον τελικό χρήστη ή στον σχεδιαστή του συστήματος να επιλέξει λύσεις που ανταποκρίνονται καλύτερα σε διαφορετικές προτεραιότητες. Αντίθετα, η απώλεια ποικιλίας περιορίζει τις διαθέσιμες επιλογές και μειώνει τη χρησιμότητα του παραγόμενου συνόλου λύσεων, ακόμη και αν αυτές βρίσκονται κοντά στο μέτωπο Pareto. Για την ενίσχυση της ποικιλίας έχουν προταθεί διάφοροι μηχανισμοί, οι οποίοι λειτουργούν συμπληρωματικά προς την κατάταξη των λύσεων βάσει της κυριαρχίας Pareto. Ένας από τους πλέον διαδεδομένους και ευρέως χρησιμοποιούμενους μηχανισμούς είναι το μέτρο crowding distance, το οποίο εισήχθη στο πλαίσιο του αλγορίθμου NSGA-II.

Το crowding distance παρέχει μια εκτίμηση της σχετικής απομόνωσης μιας λύσης σε σχέση με τις γειτονικές της λύσεις στον χώρο των στόχων, υπολογίζοντας την απόσταση μεταξύ διαδοχικών λύσεων για κάθε στόχο [58]. Η βασική ιδέα του crowding distance είναι ότι οι λύσεις που βρίσκονται σε λιγότερο πυκνές περιοχές του μετώπου Pareto θεωρούνται πιο επιθυμητές, καθώς συμβάλλουν στη διατήρηση της ομοιόμορφης κατανομής του πληθυσμού. Κατά τη διαδικασία επιλογής, όταν δύο λύσεις ανήκουν στο ίδιο μέτωπο Pareto και επομένως δεν μπορούν να διακριθούν βάσει της κυριαρχίας, προτιμάται εκείνη με τη μεγαλύτερη τιμή crowding distance. Με τον τρόπο αυτό, ο αλγόριθμος αποφεύγει τη συσσώρευση λύσεων σε συγκεκριμένες περιοχές και ενισχύει την εξερεύνηση εναλλακτικών συμβιβασμών.

Η διατήρηση ποικιλίας μέσω του crowding distance επιτρέπει στους εξελικτικούς αλγορίθμους πολλαπλών στόχων να επιτυγχάνουν ισορροπία μεταξύ σύγκλισης και εξερεύνησης, χωρίς να απαιτείται η εισαγωγή πρόσθετων παραμέτρων ή πολύπλοκων μηχανισμών ελέγχου. Η ιδιότητα αυτή καθιστά το crowding distance ιδιαίτερα ελκυστικό σε εφαρμογές όπου η ποιότητα του τελικού συνόλου λύσεων δεν εξαρτάται μόνο από την εγγύτητά του στο μέτωπο Pareto, αλλά και από την ποικιλία και την αντιπροσωπευτικότητά του.

3.4.3 Δημοφιλείς εξελικτικοί αλγόριθμοι πολλαπλών στόχων

Η ανάπτυξη εξελικτικών αλγορίθμων πολλαπλών στόχων έχει οδηγήσει στη δημιουργία πληθώρας μεθόδων που διαφέρουν ως προς τον τρόπο με τον οποίο προσεγγίζουν το μέτωπο Pareto, διατηρούν την ποικιλία του πληθυσμού και κλιμακώνονται ως προς τον αριθμό των στόχων. Παρότι όλοι οι αλγόριθμοι αυτής της κατηγορίας βασίζονται στις ίδιες θεμελιώδεις έννοιες, όπως η κυριαρχία Pareto και η πληθυσμιακή αναζήτηση, οι σχεδιαστικές τους επιλογές επηρεάζουν σημαντικά τη συμπεριφορά και την απόδοσή τους σε διαφορετικά προβλήματα [56], [57].

Ο NSGA-II αποτελεί έναν από τους πιο διαδεδομένους και ευρέως χρησιμοποιούμενους αλγορίθμους πολλαπλών στόχων, καθώς συνδυάζει αποτελεσματικά σύγκλιση προς το μέτωπο Pareto και διατήρηση ποικιλίας με σχετικά χαμηλή υπολογιστική πολυπλοκότητα. Κεντρικό στοιχείο της λειτουργίας του είναι η μη-κυριαρχούμενη ταξινόμηση, μέσω της οποίας οι λύσεις ενός πληθυσμού κατατάσσονται σε διαδοχικά μέτωπα Pareto με βάση τις σχέσεις κυριαρχίας. Το πρώτο μέτωπο περιλαμβάνει όλες τις μη κυριαρχούμενες λύσεις, ενώ τα επόμενα σχηματίζονται αφαιρώντας διαδοχικά τα προηγούμενα και επαναξιολογώντας τις σχέσεις κυριαρχίας. Η διαδικασία αυτή επιτρέπει τη σαφή ιεράρχηση των λύσεων χωρίς τη συγχώνευση των στόχων σε μία ενιαία συνάρτηση, διατηρώντας έτσι την πληροφορία των επιμέρους συμβιβασμών. Παράλληλα, ο NSGA-II ενσωματώνει μηχανισμό ελιτισμού, εξασφαλίζοντας ότι οι καλύτερες λύσεις κάθε γενιάς διατηρούνται και μεταφέρονται στον επόμενο πληθυσμό. Η επιλογή μεταξύ λύσεων που ανήκουν στο ίδιο μέτωπο Pareto πραγματοποιείται με τη χρήση του crowding distance, το οποίο προάγει τη διατήρηση της ποικιλίας ευνοώντας λύσεις που βρίσκονται σε λιγότερο πυκνοκατοικημένες περιοχές του χώρου στόχων. Ο συνδυασμός μη-κυριαρχούμενης ταξινόμησης, ελιτισμού και μηχανισμού διατήρησης ποικιλίας καθιστά τον NSGA-II ιδιαίτερα ισορροπημένο και ανθεκτικό σε πρακτικά προβλήματα, γεγονός που έχει οδηγήσει στην εκτεταμένη χρήση του σε πληθώρα εφαρμογών και αποτελεί βασικό λόγο της επιλογής του σε συστήματα που απαιτούν παραγωγή αξιόπιστων και εναλλακτικών λύσεων [53].

Παρά την επιτυχία του NSGA-II, η απόδοσή του μειώνεται όταν ο αριθμός των στόχων αυξάνεται σημαντικά, καθώς το ποσοστό των μη κυριαρχούμενων λύσεων τείνει να μεγαλώνει. Για την αντιμετώπιση αυτού του ζητήματος προτάθηκε ο NSGA-III, ο οποίος επεκτείνει τη βασική φιλοσοφία του NSGA-II εισάγοντας την έννοια των σημείων αναφοράς στον χώρο των στόχων. Τα σημεία αυτά χρησιμοποιούνται για την καθοδήγηση της επιλογής και την ενίσχυση της ομοιόμορφης κάλυψης του μετώπου Pareto, καθιστώντας τον NSGA-III καταλληλότερο για προβλήματα με μεγάλο αριθμό στόχων [54].

Ο SPEA2 αποτελεί μια εναλλακτική προσέγγιση, η οποία βασίζεται στη διατήρηση ενός εξωτερικού αρχείου μη κυριαρχούμενων λύσεων. Η αξιολόγηση των λύσεων πραγματοποιείται μέσω μιας συνάρτησης ισχύος (strength), η οποία λαμβάνει υπόψη τόσο την κυριαρχία όσο και την πυκνότητα των λύσεων στον χώρο των στόχων. Με τον τρόπο αυτό, ο SPEA2 επιδιώκει την ταυτόχρονη σύγκλιση προς το μέτωπο Pareto και τη διατήρηση της ποικιλίας, χωρίς να βασίζεται αποκλειστικά σε μηχανισμούς όπως το crowding distance [59].

Μια διαφορετική φιλοσοφία υιοθετεί ο MOEA/D, ο οποίος προσεγγίζει το πρόβλημα πολλαπλών στόχων μέσω αποσύνθεσης σε ένα σύνολο επιμέρους προβλημάτων ενός στόχου. Κάθε υποπρόβλημα αντιστοιχεί σε διαφορετικό συνδυασμό βαρών και επιλύεται ταυτόχρονα με τα υπόλοιπα, επιτρέποντας την ανταλλαγή πληροφορίας μεταξύ γειτονικών υποπροβλημάτων. Η προσέγγιση αυτή έχει αποδειχθεί ιδιαίτερα αποτελεσματική σε προβλήματα με μεγάλο αριθμό στόχων, καθώς μειώνει τη δυσκολία της άμεσης διαχείρισης της κυριαρχίας Pareto σε υψηλές διαστάσεις [60].

Για να γίνουν περισσότερο κατανοητές οι βασικές διαφορές μεταξύ των παραπάνω προσεγγίσεων, κρίνεται χρήσιμη μια συνοπτική συγκριτική επισκόπηση των βασικών χαρακτηριστικών τους, προκειμένου να αναδειχθούν οι κύριες διαφορές και τα πλεονεκτήματα κάθε προσέγγισης. Ο πίνακας που ακολουθεί (Πίνακας 3.1), συνοψίζει τα σημαντικότερα σημεία σύγκρισης μεταξύ των NSGA-II, NSGA-III, SPEA2 και MOEA/D, εστιάζοντας σε θεωρητικά και σχεδιαστικά χαρακτηριστικά που επηρεάζουν την καταλληλότητά τους για διαφορετικές κατηγορίες προβλημάτων.

3.4.4 Μετρικές αξιολόγησης μετώπων Pareto

Η αξιολόγηση της απόδοσης ενός εξελκτικού αλγορίθμου πολλαπλών στόχων δεν μπορεί να βασιστεί σε μία μόνο τιμή καταλληλότητας, καθώς το αποτέλεσμα της διαδικασίας βελτιστοποίησης είναι ένα σύνολο λύσεων και όχι μία μοναδική βέλτιστη λύση. Για τον λόγο αυτό, έχουν αναπτυχθεί ειδικές μετρικές αξιολόγησης που επιτρέπουν τη σύγκριση διαφορετικών συνόλων μη κυριαρχούμενων λύσεων ως προς την ποιότητά τους, λαμβάνοντας υπόψη τόσο τη σύγκλιση προς το μέτωπο Pareto όσο και την κατανομή των λύσεων στον χώρο των στόχων [56], [57].

Πίνακας 3.1: Σύγκριση δημοφιλών εξελκτικών αλγορίθμων πολλαπλών στόχων

Αλγόριθμος	Τρόπος κατάταξης λύσεων	Διατήρηση ποικιλίας	Κλιμάκωση ως προς τον αριθμό στόχων	Βασική φιλοσοφία
NSGA-II	Μη-κυριαρχούμενη ταξινόμηση σε μέτωπα Pareto	Crowding distance	Περιορισμένη αποδοτικότητα σε μεγάλο αριθμό στόχων	Ισορροπία σύγκλισης και ποικιλίας με ελιτισμό
NSGA-III	Μη-κυριαρχούμενη ταξινόμηση με σημεία αναφοράς	Καθοδήγηση μέσω σημείων αναφοράς	Κατάλληλος για προβλήματα με πολλούς στόχους	Στοχευμένη κάλυψη του μετώπου Pareto
SPEA2	Αξιολόγηση μέσω strength και εξωτερικού αρχείου	Εκτίμηση πυκνότητας λύσεων	Μέτρια κλιμάκωση	Έμφαση στη διατήρηση αρχείου μη κυριαρχούμενων λύσεων
MOEA/D	Αποσύνθεση σε επιμέρους προβλήματα ενός στόχου	Συνεργασία γειτονικών υποπροβλημάτων	Υψηλή αποδοτικότητα σε πολλούς στόχους	Μετατροπή του προβλήματος σε σύνολο απλούστερων υποπροβλημάτων

Μία από τις πλέον διαδεδομένες μετρικές είναι το Hypervolume (HV), το οποίο μετρά τον όγκο του χώρου στόχων που κυριαρχείται από το παραγόμενο σύνολο λύσεων και ορίζεται σε σχέση με ένα προκαθορισμένο σημείο αναφοράς. Το Hypervolume ενσωματώνει ταυτόχρονα πληροφορία σύγκλισης και ποικιλίας, καθώς αυξάνεται όταν οι λύσεις προσεγγίζουν το μέτωπο Pareto και καλύπτουν μεγαλύτερο εύρος του χώρου στόχων. Η ιδιότητά του αυτή το καθιστά ιδιαίτερα ελκυστικό για τη συνολική αξιολόγηση αλγορίθμων πολλαπλών στόχων, αν και ο υπολογισμός του μπορεί να είναι υπολογιστικά απαιτητικός καθώς αυξάνεται ο αριθμός των στόχων [61].

Η Generational Distance (GD) αποτελεί μετρική που εστιάζει κυρίως στη σύγκλιση του παραγόμενου συνόλου λύσεων προς το πραγματικό μέτωπο Pareto. Υπολογίζεται ως η μέση απόσταση των λύσεων του αλγορίθμου από ένα γνωστό ή προσεγγιστικό μέτωπο αναφοράς και παρέχει ένδειξη του πόσο κοντά βρίσκονται οι λύσεις στο ιδανικό σύνολο. Μικρότερες τιμές GD υποδηλώνουν καλύτερη

σύγκλιση, ωστόσο η μετρική αυτή δεν λαμβάνει άμεσα υπόψη την κατανομή ή την ποικιλία των λύσεων κατά μήκος του μετώπου [62].

Για την αντιμετώπιση του περιορισμού αυτού έχει προταθεί η Inverted Generational Distance (IGD), η οποία αντιστρέφει τον ρόλο των συνόλων που χρησιμοποιούνται στον υπολογισμό της απόστασης. Στην περίπτωση αυτή, υπολογίζεται η μέση απόσταση των σημείων του μετώπου αναφοράς από το παραγόμενο σύνολο λύσεων. Με τον τρόπο αυτό, η IGD αξιολογεί τόσο τη σύγκλιση όσο και την πληρότητα κάλυψης του μετώπου Pareto, καθώς υψηλές τιμές μπορεί να προκύψουν είτε λόγω κακής σύγκλισης είτε λόγω ανεπαρκούς κατανομής λύσεων [62], [63].

Οι παραπάνω μετρικές χρησιμοποιούνται ευρέως στη βιβλιογραφία για τη σύγκριση και την αξιολόγηση εξελικτικών αλγορίθμων πολλαπλών στόχων, καθώς παρέχουν συμπληρωματικές οπτικές της ποιότητας των παραγόμενων λύσεων. Η επιλογή της κατάλληλης μετρικής εξαρτάται από το ζητούμενο της αξιολόγησης, όπως η έμφαση στη σύγκλιση, στην ποικιλία ή στη συνολική αναπαράσταση του μετώπου Pareto, και αποτελεί σημαντικό στοιχείο στον σχεδιασμό και την ανάλυση πειραματικών αποτελεσμάτων.

3.4.5 Σύγκλιση, υπολογιστικός χρόνος και χειρισμός περιορισμών

Η απόδοση ενός εξελικτικού αλγορίθμου πολλαπλών στόχων αξιολογείται όχι μόνο από την ποιότητα του τελικού μετώπου Pareto, αλλά και από τη δυναμική της εξέλιξης κατά τη διάρκεια των γενεών, τον απαιτούμενο υπολογιστικό χρόνο και τον τρόπο με τον οποίο αντιμετωπίζονται οι περιορισμοί του προβλήματος. Τα στοιχεία αυτά είναι στενά αλληλένδετα και επηρεάζουν καθοριστικά τη χρησιμότητα ενός αλγορίθμου σε πρακτικές εφαρμογές, όπου οι υπολογιστικοί πόροι και οι χρονικοί περιορισμοί αποτελούν συχνά κρίσιμους παράγοντες [56], [57].

Η σύγκλιση αναφέρεται στον βαθμό και τον ρυθμό με τον οποίο ο πληθυσμός λύσεων προσεγγίζει το μέτωπο Pareto κατά τη διάρκεια της εξελικτικής διαδικασίας. Ένας αλγόριθμος θεωρείται ότι παρουσιάζει καλή συμπεριφορά σύγκλισης όταν οι λύσεις βελτιώνονται σταδιακά ως προς τους στόχους και σταθεροποιούνται σε ένα σύνολο μη κυριαρχούμενων λύσεων μετά από έναν εύλογο αριθμό γενεών. Η πρόωρη σύγκλιση, δηλαδή η ταχεία συγκέντρωση του πληθυσμού σε περιορισμένες περιοχές του χώρου λύσεων, μπορεί να οδηγήσει σε απώλεια εναλλακτικών συμβιβασμών και υποβάθμιση της ποιότητας του τελικού μετώπου Pareto. Για τον λόγο αυτό, η σύγκλιση εξετάζεται πάντοτε σε συνδυασμό με τη διατήρηση ποικιλίας, όπως αναλύθηκε στην προηγούμενη υποενότητα [56].

Ο υπολογιστικός χρόνος ενός αλγορίθμου πολλαπλών στόχων επηρεάζεται από διάφορους παράγοντες, όπως το μέγεθος του πληθυσμού, ο αριθμός των στόχων, η πολυπλοκότητα της συνάρτησης αξιολόγησης και ο μηχανισμός κατάταξης των λύσεων. Μέθοδοι που βασίζονται στη μη-κυριαρχούμενη ταξινόμηση ενδέχεται να παρουσιάζουν αυξημένο υπολογιστικό κόστος, ιδιαίτερα σε προβλήματα με μεγάλο πληθυσμό ή αυξημένο αριθμό στόχων. Ως αποτέλεσμα, η επιλογή παραμέτρων και αλγοριθμικών τεχνικών αποτελεί συμβιβασμό μεταξύ ακρίβειας, ποιότητας λύσεων και αποδεκτού χρόνου εκτέλεσης, ειδικά σε περιβάλλοντα όπου απαιτείται επαναλαμβανόμενη ή διαδραστική παραγωγή αποτελεσμάτων [57], [58].

Ένα επιπλέον κρίσιμο ζήτημα στα προβλήματα πολλαπλών στόχων είναι ο χειρισμός περιορισμών, καθώς οι περισσότερες πραγματικές εφαρμογές περιλαμβάνουν περιορισμούς που ορίζουν ποια λύση θεωρείται αποδεκτή. Οι περιορισμοί μπορεί να αφορούν ανώτερα ή κατώτερα όρια στόχων, λογικούς κανόνες ή συνδυασμούς χαρακτηριστικών που δεν επιτρέπονται. Στο πλαίσιο των εξελικτικών αλγορίθμων, έχουν προταθεί διάφορες στρατηγικές χειρισμού περιορισμών, όπως η ενσωμάτωσή τους στη συνάρτηση καταλληλότητας μέσω ποινών, η προτεραιοποίηση εφικτών λύσεων έναντι μη εφικτών

ή η διόρθωση λύσεων που παραβιάζουν περιορισμούς. Η επιλογή της κατάλληλης στρατηγικής επηρεάζει άμεσα τόσο τη σύγκλιση όσο και την ποιότητα των παραγόμενων λύσεων, καθώς μια υπερβολικά αυστηρή ή υπερβολικά ελαστική αντιμετώπιση των περιορισμών μπορεί να οδηγήσει σε ανεπιθύμητα αποτελέσματα [56], [58].

3.4.6 Σύνδεση με το υπό μελέτη πρόβλημα

Οι έννοιες και οι τεχνικές που παρουσιάστηκαν στις προηγούμενες υποενότητες συνθέτουν το θεωρητικό πλαίσιο πάνω στο οποίο βασίζεται η προσέγγιση της παρούσας εργασίας για τον σχεδιασμό εξατομικευμένων διατροφικών πλάνων. Η ταυτόχρονη ύπαρξη πολλαπλών διατροφικών στόχων, όπως η προσέγγιση θερμιδικών απαιτήσεων, η ισορροπία μακροθρεπτικών και μικροθρεπτικών στοιχείων και η συμμόρφωση με διατροφικούς περιορισμούς, καθιστά το πρόβλημα ιδιαίτερα κατάλληλο για τη χρήση εξελκτικών αλγορίθμων πολλαπλών στόχων. Η έννοια της κυριαρχίας Pareto, η διατήρηση ποικιλίας μέσω μηχανισμών όπως το crowding distance, καθώς και οι στρατηγικές σύγκλισης και χειρισμού περιορισμών, επιτρέπουν την παραγωγή συνόλων λύσεων που αντανάκλουν διαφορετικούς, αλλά εξίσου αποδεκτούς, συμβιβασμούς μεταξύ των στόχων.

Με βάση τα παραπάνω, στην παρούσα εργασία επιλέχθηκε ο NSGA-II και η επιλογή αυτή αιτιολογείται από τον ισορροπημένο συνδυασμό απλότητας, αποδεκτού υπολογιστικού κόστους και ικανότητας παραγωγής ποικίλων και σταθερών λύσεων, χαρακτηριστικά που είναι ιδιαίτερα σημαντικά σε εφαρμογές με πρακτικούς περιορισμούς και ανάγκη για επαναλαμβανόμενη εκτέλεση. Η θεωρητική ανάλυση των αλγορίθμων πολλαπλών στόχων που προηγήθηκε θέτει έτσι τις βάσεις για την κατανόηση των σχεδιαστικών επιλογών που υιοθετούνται στα επόμενα κεφάλαια. Πιο συγκεκριμένα, η τεχνική υλοποίηση του αλγορίθμου περιγράφεται αναλυτικά στην υποενότητα 5.4.3, ενώ η αξιολόγηση των αποτελεσμάτων παρατίθεται στο Κεφάλαιο 6.

3.5 Εφαρμογές ΕΑ στη διατροφή και προηγούμενες μελέτες

Έχοντας αναφέρει τα βασικά συστατικά των ΕΑ, είναι καλό σε αυτό το σημείο να αναφερθούν ορισμένες εφαρμογές και μελέτες αυτών στον τομέα της διατροφής, που είναι και το πλαίσιο μέσα στο οποίο αναπτύχθηκε η παρούσα ΔΕ. Η εφαρμογή των ΕΑ στη διατροφή έχει αναδειχθεί ως ένα από τα πιο ενδιαφέροντα πεδία βελτιστοποίησης πολλαπλών στόχων, συνδυάζοντας επιστημονικές αρχές της υπολογιστικής νοημοσύνης με τις πρακτικές ανάγκες του διατροφικού σχεδιασμού. Στη βιβλιογραφία συναντάται πληθώρα προσεγγίσεων που αξιοποιούν ΕΑ για την αυτόματη ή ημιαυτόματη δημιουργία διατροφικών πλάνων, τη βελτιστοποίηση της θρεπτικής ισορροπίας, την εξατομίκευση με βάση προτιμήσεις χρηστών, καθώς και τη διαχείριση περιορισμών όπως κόστος ή θερμιδικός στόχος. Οι ερευνητικές αυτές προσπάθειες αποδεικνύουν την ευελιξία των ΕΑ, οι οποίοι μπορούν να προσαρμοστούν σε διαφορετικά επίπεδα πολυπλοκότητας και να παράγουν λύσεις εφαρμόσιμες σε πραγματικά περιβάλλοντα.

Μια από τις πρώτες κατευθύνσεις εστιάζει στην αυτόματη δημιουργία ημερήσιων/εβδομαδιαίων μενού με GA. Ενδεικτικά, οι Kołodziejczyk και Przybyłek ανέπτυξαν σύστημα αυτόματου σχεδιασμού μενού όπου κάθε υποψήφια λύση κωδικοποιεί ένα πλάνο γευμάτων και αξιολογείται με βάση θερμιδικούς/θρεπτικούς στόχους και ποικιλία. Η χρήση GA επιτρέπει την αναζήτηση σε μεγάλα σύνολα τροφίμων και τη σταδιακή βελτίωση των συνδυασμών χωρίς να απαιτείται αυστηρή αναλυτική μορφή του προβλήματος [26]. Το μοντέλο τους καταδεικνύει ότι η κατάλληλη κωδικοποίηση και ο χειρισμός περιορισμών (π.χ. με ποινές ανάλογες της υπέρβασης/υστέρησης) οδηγούν σε πλάνα που είναι ταυτόχρονα εφικτά και πρακτικά.

Αργότερα, μελέτες στράφηκαν σε προσωποποιημένη παραγωγή πλάνων με ενσωμάτωση προτιμήσεων χρηστών και διατροφικών περιορισμών. Οι Jia et al. αξιοποίησαν GA για να παραγάγουν εξατομικευμένα πλάνα που λαμβάνουν υπόψη προτιμήσεις, αλλεργίες και θρεπτικούς στόχους, επιτυγχάνοντας ισορροπία ανάμεσα στην κάλυψη αναγκών και τη ρεαλιστική εφαρμογή των προτάσεων [27]. Η εργασία τους δείχνει πώς τα βάρη στους στόχους και οι παράμετροι των τελεστών (π.χ. SBX/πολυωνυμική μετάλλαξη για πραγματικές ποσότητες) επηρεάζουν την ποιότητα και την ποικιλία των λύσεων, υπογραμμίζοντας τη σημασία της ρύθμισης του αλγορίθμου στο προφίλ του χρήστη.

Με τη μετάβαση σε MOEA, οι ερευνητές αξιοποίησαν αλγορίθμους τύπου NSGA για να παραγάγουν όχι μία, αλλά ένα σύνολο μη-κυριαρχούμενων πλάνων (Pareto front), προσφέροντας διαφορετικούς συμβιβασμούς μεταξύ στόχων (π.χ. πληρότητα μακρο/μικροθρεπτικών, θερμιδικό ισοζύγιο, κόστος, ποικιλία). Οι Türkmenoğlu et al. διατύπωσαν το Many-Objective πρόβλημα διατροφής και υλοποίησαν βελτιστοποίηση με NSGA-III, δείχνοντας ότι η χρήση σημείων αναφοράς και μηχανισμών διατήρησης ποικιλίας βοηθά στη διασπορά λύσεων σε χώρους με πολλούς στόχους, οδηγώντας σε πιο ρεαλιστικά και ισορροπημένα πλάνα [28]. Η συνεισφορά αυτή είναι ιδιαίτερα σημαντική όταν οι απαιτήσεις ξεπερνούν τον κλασικό τριπλό στόχο θερμίδων-μακροθρεπτικών-κόστους και επεκτείνονται σε περισσότερα μικροθρεπτικά ή σε λειτουργικούς περιορισμούς.

Παράλληλα, έχουν προταθεί υβριδικές αρχιτεκτονικές που συνδυάζουν διαφορετικές μεταερευτικές για να βελτιώσουν ταυτόχρονα την ποιότητα και την ποικιλία των λύσεων. Οι Xavier et al. αξιοποίησαν ένα island-based υβριδικό σχήμα που συνδυάζει GA με Διαφορική Εξέλιξη (DE) για υποθερμιδικά πλάνα, αναφέροντας βελτίωση στην κάλυψη στόχων και στη διαφοροποίηση λύσεων μεταξύ «νησιών» (υποπληθυσμών) που εξελίσσονται μερικώς ανεξάρτητα και ανταλλάσσουν λύσεις περιοδικά [29]. Η island-based λογική είναι ελκυστική για μεγάλα διατροφικά σύνολα (π.χ. βάσεις τύπου USDA), όπου το κόστος αξιολόγησης είναι σημαντικό και η διατήρηση ποικιλίας κρίσιμη.

Σε επίπεδο μεθοδολογίας βελτιστοποίησης, η βιβλιογραφία αναδεικνύει ότι ο σχεδιασμός της συνάρτησης καταλληλότητας και ο χειρισμός των περιορισμών είναι καθοριστικοί για την επιτυχία. Η χρήση ποινών αναλογικών με το μέγεθος της παραβίασης, η ιεράρχηση στόχων (π.χ. προτεραιότητα στην εφικτότητα έναντι της βελτιστοποίησης εντός του εφικτού χώρου), η αξιοποίηση της μη-κυριαρχίας και μέτρων ποικιλίας (π.χ. crowding distance), βρέθηκαν να βελτιώνουν τη σταθερότητα και τη χρησιμότητα των αποτελεσμάτων σε πραγματικές συνθήκες [53], [55].

Επιπλέον, μια διαρκής πρόκληση που αναδεικνύεται είναι η ποιότητα και η προτυποποίηση των δεδομένων τροφίμων. Τα έργα που χρησιμοποιούν μεγάλα δημόσια σύνολα, όπως βάσεις τύπου USDA, επισημαίνουν ότι η εναρμόνιση θρεπτικών πεδίων, οι ελλείψεις ή οι αποκλίσεις στις μερίδες και οι ασυνέπειες στη διαθεσιμότητα μικροθρεπτικών επηρεάζουν άμεσα το fitness και την αξιοπιστία της βελτιστοποίησης [26], [27], [28]. Οι μελέτες αντιμετωπίζουν το ζήτημα με προεπεξεργασία δεδομένων, ορισμό εύρους αποδεκτών τιμών και όπου χρειάζεται, δημιουργία συνθετικών γευμάτων από βασικά συστατικά, ώστε να μειωθούν artifacts, δηλαδή λανθασμένα ή παραμορφωμένα στοιχεία που προκύπτουν από αδυναμίες της βάσης.

Τέλος, αρκετά έργα θίγουν ζητήματα χρηστικότητας και υιοθέτησης. Ακόμη και βέλτιστα διατροφικά πλάνα αποτυγχάνουν αν δεν ληφθούν υπόψη προτιμήσεις, ανεκτικότητα σε αλλεργίες, πολιτισμικές συνήθειες, ευκολία παρασκευής και ποικιλία. Στο πλαίσιο αυτό, τα εξελικτικά συστήματα που δημιουργούν σύνολα λύσεων αντί για μία μοναδική πρόταση δίνουν στον χρήστη ή στον επαγγελματία υγείας τη δυνατότητα επιλογής με βάση προσωπικά κριτήρια, βελτιώνοντας την προσωποποίηση και

την συμμόρφωση [27], [28], [29]. Η δυνατότητα μετα-επεξεργασίας (π.χ. αντικατάσταση τροφίμων ισοδύναμης θρεπτικής αξίας, προσαρμογή μερίδων) ενισχύει περαιτέρω την εφαρμοσιμότητα.

3.6 Βιβλιοθήκες υλοποίησης ΕΑ

Η μετάβαση από τη θεωρητική κατανόηση των Εξελικτικών Αλγορίθμων (ΕΑ) στην πρακτική τους υλοποίηση προϋποθέτει την επιλογή κατάλληλων εργαλείων λογισμικού που να υποστηρίζουν αποτελεσματικά τα επιμέρους στάδια της εξελικτικής διαδικασίας. Τα συστατικά που αναλύθηκαν στις προηγούμενες υποενότητες, υλοποιούνται στην πράξη μέσω εξειδικευμένων βιβλιοθηκών προγραμματισμού που παρέχουν έτοιμες δομές, τελεστές και μηχανισμούς εξέλιξης. Η επιλογή της κατάλληλης βιβλιοθήκης δεν καθορίζει μόνο τη λειτουργικότητα του αλγορίθμου, αλλά επηρεάζει άμεσα την αποδοτικότητα, τη συμβατότητα και την ευκολία συντήρησης του συνολικού συστήματος.

Στη διεθνή βιβλιογραφία και κοινότητα ανάπτυξης λογισμικού έχουν καθιερωθεί ορισμένες βιβλιοθήκες που χρησιμοποιούνται ευρέως για την υλοποίηση ΕΑ. Μεταξύ αυτών, οι πιο γνωστές είναι η Jenetics, που είναι γραμμένη σε Java, η DEAP (Distributed Evolutionary Algorithms in Python), που είναι μία από τις πιο διαδεδομένες βιβλιοθήκες Python, καθώς και άλλες βιβλιοθήκες όπως η ECJ (Evolutionary Computation in Java), η PyGAD και η Inspyred, οι οποίες καλύπτουν διαφορετικές ανάγκες και επίπεδα πολυπλοκότητας [49], [53], [64].

Η DEAP, συγκεκριμένα, έχει καθιερωθεί ως σημείο αναφοράς στον χώρο των ΕΑ λόγω της εκτεταμένης τεκμηρίωσης, της ευκολίας σύνθεσης πολύπλοκων γενετικών σχημάτων και της μεγάλης κοινότητας υποστήριξης [65]. Η αρχιτεκτονική της βασίζεται σε Python και παρέχει έτοιμες δομές για γενετικούς αλγόριθμους, εξελικτικές στρατηγικές και βελτιστοποίηση πολλαπλών στόχων, υποστηρίζοντας παράλληλη εκτέλεση και δυναμική παραμετροποίηση. Ωστόσο, η υιοθέτηση της DEAP σε εφαρμογές Android θα απαιτούσε την ανάπτυξη ενός ξεχωριστού backend που θα εκτελούσε τον αλγόριθμο εξ αποστάσεως (server-side), με το κινητό να λειτουργεί μόνο ως διεπαφή χρήστη. Η λύση αυτή, αν και τεχνικά πιο ισχυρή, θα απαιτούσε πρόσθετη υποδομή, διαχείριση endpoints και επικοινωνία μέσω RESTful APIs, αποκλίνοντας από τον βασικό σκοπό της εργασίας.

Για τον λόγο αυτό επιλέχθηκε η Jenetics, η οποία είναι πλήρως γραμμένη σε Java και επομένως απόλυτα συμβατή με την Kotlin, τη γλώσσα στην οποία έχει αναπτυχθεί η παρούσα Android εφαρμογή. Η συμβατότητα αυτή εξασφαλίζει άμεση ενσωμάτωση του αλγορίθμου στο ίδιο περιβάλλον εκτέλεσης της εφαρμογής, χωρίς την ανάγκη διαμεσολάβησης από backend ή εξωτερικές υπηρεσίες. Με αυτόν τον τρόπο, ο εξελικτικός αλγόριθμος εκτελείται απευθείας στη συσκευή του χρήστη, περιορίζοντας καθυστερήσεις, εξαρτήσεις από το διαδίκτυο και θέματα απορρήτου. Η λύση αυτή ευθυγραμμίζεται με τον στόχο της διπλωματικής, που επικεντρώνεται στην τοπική υλοποίηση της βελτιστοποίησης και όχι στην ανάπτυξη κατανεμημένης αρχιτεκτονικής.

Πέρα από τη συμβατότητα με την Kotlin, η Jenetics προσφέρει και ορισμένα πρόσθετα πλεονεκτήματα που την καθιστούν πρακτική επιλογή για την παρούσα εργασία. Διαθέτει αντικειμενοστραφή σχεδίαση, καθαρή ιεραρχία κλάσεων για τα βασικά στοιχεία των ΕΑ (Genotype, Phenotype, Gene, Engine, Selector, Alterer), καθώς και δυνατότητα χρήσης πολυ-νηματικής εκτέλεσης (multi-threading), η οποία αξιοποιεί πλήρως τους επεξεργαστικούς πόρους της συσκευής χωρίς την ανάγκη εξωτερικού framework [64]. Επίσης, υποστηρίζει γενετικούς και αλγορίθμους πολλαπλών στόχων, όπως ο NSGA-II, επιτρέποντας μελλοντική επέκταση ή τροποποίηση της εφαρμογής χωρίς αλλαγή της υποκείμενης βιβλιοθήκης.

3.7 Βιβλιοθήκη Jenetics

Αφού παρουσιάστηκαν κάποιες από τις πολυχρησιμοποιούμενες βιβλιοθήκες για την ανάπτυξη ΕΑ και αιτιολογήθηκε η επιλογή της Jenetics για την παρούσα ΔΕ, κρίνεται πρόπον σε αυτό το σημείο, να δοθούν περισσότερες λεπτομέρειες σχετικά με αυτή. Στην παρούσα ενότητα εξετάζονται πιο αναλυτικά οι δυνατότητες και η αρχιτεκτονική της βιβλιοθήκης, καθώς και οι μηχανισμοί που διευκολύνουν την ενσωμάτωση και παραμετροποίηση των εξελικτικών διαδικασιών.

Η Jenetics είναι μια αντικειμενοστραφής βιβλιοθήκη ανοιχτού κώδικα, η οποία υλοποιεί με σαφή και επεκτάσιμο τρόπο τις αρχές των ΕΑ. Η αρχιτεκτονική της βασίζεται σε ένα ιεραρχικό μοντέλο κλάσεων, το οποίο αντικατοπτρίζει πιστά τις έννοιες που αναλύθηκαν θεωρητικά στις υποενότητες 3.1 και 3.2. Ο πυρήνας της βιβλιοθήκης συγκροτείται γύρω από τις κλάσεις Genotype, Phenotype, Gene, Chromosome, Engine, Selector και Alterer [66].

Η κλάση Gene αποτελεί τη θεμελιώδη μονάδα πληροφορίας στον αλγόριθμο και αντιστοιχεί σε μια μεταβλητή ή χαρακτηριστικό της λύσης. Τα γονίδια οργανώνονται σε δομές τύπου Chromosome, όπου κάθε χρωμόσωμα εκφράζει ένα υποσύνολο της συνολικής λύσης, ενώ το Genotype αναπαριστά το πλήρες γενετικό προφίλ ενός ατόμου, δηλαδή τον συνδυασμό όλων των χρωμοσωμάτων που το απαρτίζουν. Το Phenotype συνδέει το γενετικό υλικό με το περιβάλλον αξιολόγησης, ενσωματώνοντας τη συνάρτηση καταλληλότητας η οποία εκτιμά την ποιότητα κάθε λύσης. Η διάκριση αυτή ανάμεσα σε Genotype και Phenotype επιτρέπει την ευέλικτη σύνδεση διαφορετικών σχημάτων κωδικοποίησης με διαφορετικούς τρόπους αξιολόγησης, κάτι που ενισχύει τη γενικότητα της βιβλιοθήκης [66].

Η βιβλιοθήκη επίσης, παρέχει έτοιμες κλάσεις χρωμοσωμάτων για διαφορετικούς τύπους προβλημάτων, όπως IntegerChromosome, DoubleChromosome, BitChromosome και CharacterChromosome. Κάθε τύπος αντιστοιχεί σε συγκεκριμένη μορφή κωδικοποίησης, ακέραιες, πραγματικές, δυαδικές ή συμβολοσειρές χαρακτήρων. Αντίστοιχα, τα Gene αντικατοπτρίζουν τον τύπο τιμών που μπορούν να λάβουν οι μεταβλητές (IntegerGene, DoubleGene, BitGene, CharacterGene) [66]. Με τον τρόπο αυτό, η Jenetics υποστηρίζει εξ αρχής τόσο δυαδικές όσο και συνεχείς αναπαραστάσεις, καθώς και μεικτές μορφές όταν απαιτείται συνδυασμός τους στο ίδιο Genotype.

Επιπροσθέτως, παρέχεται ένας εύχρηστος μηχανισμός ορισμού της συνάρτησης καταλληλότητας μέσω του Codec, ο οποίος συνδέει το Genotype με την πραγματική μορφή των δεδομένων που αξιολογούνται. Ο Codec καθορίζει πώς μεταφράζεται ένα γενετικό άτομο σε φαινότυπο και αντίστροφα, επιτρέποντας τη δημιουργία διαφορετικών μοντέλων αξιολόγησης χωρίς αλλαγή της βασικής δομής του αλγορίθμου. Η συνάρτηση καταλληλότητας καθορίζεται από τον χρήστη και δέχεται ως είσοδο ένα άτομο (Phenotype), επιστρέφοντας την τιμή αξιολόγησής του. Ο μηχανισμός αυτός επιτρέπει τον πλήρη διαχωρισμό του γενετικού χώρου από τον χώρο αξιολόγησης, κάτι που θεωρείται βέλτιστη πρακτική στη σχεδίαση Εξελικτικών Αλγορίθμων [66].

Η κλάση Engine αποτελεί το κεντρικό στοιχείο της εκτέλεσης του αλγορίθμου και είναι υπεύθυνη για τη ροή της εξέλιξης. Μέσω του Engine Builder, ο χρήστης μπορεί να ορίσει πλήρως τις παραμέτρους λειτουργίας του αλγορίθμου, όπως το μέγεθος του πληθυσμού (populationSize), το ποσοστό απογόνων ανά γενιά (offspringFraction), τη στρατηγική βελτιστοποίησης, τα κριτήρια τερματισμού και το είδος των τελεστών που θα χρησιμοποιηθούν. Ο προγραμματιστικός σχεδιασμός του Engine ακολουθεί το πρότυπο του builder pattern, επιτρέποντας τη σταδιακή και ευανάγνωστη παραμετροποίηση του αλγορίθμου [66].

Η Jenetics περιλαμβάνει πληθώρα μηχανισμών επιλογής, όπως RouletteWheelSelector, TournamentSelector, LinearRankSelector, BoltzmannSelector και ExponentialRankSelector, μεταξύ

άλλων. Ο προγραμματιστής μπορεί να επιλέξει ποιον Selector θα χρησιμοποιήσει απλώς περνώντας τον αντίστοιχο τύπο ως παράμετρο στο Engine, ενώ υποστηρίζεται και ο συνδυασμός διαφορετικών selectors ή η δημιουργία προσαρμοσμένων [66]. Η ευελιξία αυτή επιτρέπει την προσαρμογή του μηχανισμού επιλογής στη φύση κάθε προβλήματος, είτε αυτό απαιτεί υψηλή τυχαιότητα είτε μεγαλύτερη πίεση επιλογής (selection pressure) προς καλύτερες λύσεις.

Υποστηρίζεται ακόμα, πληθώρα τελεστών εξέλιξης (Alterers), οι οποίοι καθορίζουν πώς προκύπτουν νέοι απόγονοι από τους υπάρχοντες γονείς. Οι συνηθέστεροι τελεστές περιλαμβάνουν τη διασταύρωση (Crossover) και τη μετάλλαξη (Mutator), αλλά η Jenetics προσφέρει και τη δυνατότητα δημιουργίας PartialAlterers, δηλαδή τελεστών που εφαρμόζονται μόνο σε συγκεκριμένα τμήματα του πληθυσμού ή σε ορισμένες γενιές. Οι PartialAlterers δίνουν τη δυνατότητα πιο ελεγχόμενης εξελικτικής διαδικασίας, επιτρέποντας τη διαφοροποίηση της συμπεριφοράς του αλγορίθμου ανάλογα με το στάδιο στο οποίο βρίσκεται η εξέλιξη [66]. Έτσι, ο χρήστης μπορεί να αυξήσει ή να μειώσει τη συχνότητα εφαρμογής συγκεκριμένων τελεστών, επιτυγχάνοντας καλύτερη ισορροπία ανάμεσα στην εξερεύνηση και την εκμετάλλευση του χώρου λύσεων.

Η κλάση EvolutionResult καταγράφει την κατάσταση της εξέλιξης σε κάθε γενιά, αποθηκεύοντας πληροφορίες για τον πληθυσμό, τη μέση και τη βέλτιστη τιμή fitness, καθώς και στατιστικά στοιχεία για τη διαδικασία. Οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν για παρακολούθηση της προόδου ή για μεταγενέστερη ανάλυση των αποτελεσμάτων. Η Jenetics υποστηρίζει επίσης εκτέλεση σε πολλαπλά νήματα μέσω του Stream API της Java, αξιοποιώντας πλήρως τις δυνατότητες των πολυπύρηνων επεξεργαστών και βελτιώνοντας τον χρόνο εκτέλεσης χωρίς πρόσθετο προγραμματιστικό φόρτο [66].

Η τεκμηρίωση της Jenetics είναι διαθέσιμη μέσω του επίσημου ιστότοπου και του GitHub repository της [66], όπου περιγράφονται αναλυτικά όλες οι κλάσεις, οι παράμετροι και τα υποστηριζόμενα interfaces. Παρέχονται επίσης πρακτικά παραδείγματα κώδικα και οδηγίες για την παραμετροποίηση διαφορετικών τύπων εξελικτικών σεναρίων, γεγονός που καθιστά τη βιβλιοθήκη ιδιαίτερα κατάλληλη τόσο για ερευνητικούς όσο και για εφαρμοσμένους σκοπούς.

3.8 Επίλογος

Το παρόν κεφάλαιο θέτει το θεωρητικό υπόβαθρο πάνω στο οποίο βασίζεται η αλγοριθμική προσέγγιση της παρούσας διπλωματικής εργασίας, εισάγοντας τις βασικές αρχές και τις σύγχρονες εξελίξεις των εξελικτικών αλγορίθμων, με ιδιαίτερη έμφαση στους αλγορίθμους πολλαπλών στόχων. Μέσα από τη θεωρητική ανάλυση που προηγήθηκε, αναδεικνύεται ο ρόλος των εξελικτικών μεθόδων ως κατάλληλων εργαλείων για την αντιμετώπιση σύνθετων προβλημάτων βελτιστοποίησης με αντικρουόμενες απαιτήσεις, όπως ο εξατομικευμένος διατροφικός σχεδιασμός. Το κεφάλαιο αυτό λειτουργεί ως γέφυρα μεταξύ της εννοιολογικής κατανόησης των αλγορίθμων και της πρακτικής αξιοποίησής τους, προετοιμάζοντας το έδαφος για τη μετάβαση από τη θεωρία στη συστηματική σχεδίαση ενός λειτουργικού συστήματος. Στο επόμενο κεφάλαιο παρουσιάζεται η ανάλυση και η σχεδίαση της εφαρμογής, περιλαμβάνοντας τόσο τις δομικές επιλογές που απορρέουν από τη θεωρητική προσέγγιση των εξελικτικών αλγορίθμων όσο και τις αρχιτεκτονικές αποφάσεις που αφορούν τη συνολική οργάνωση της Android εφαρμογής.

Κεφάλαιο 4ο: Ανάλυση και σχεδίαση της εφαρμογής

4.1 Εισαγωγή

Η ανάπτυξη μιας εφαρμογής που αξιοποιεί εξελκτικούς αλγορίθμους για την παραγωγή εξατομικευμένων διατροφικών πλάνων απαιτεί μια στιβαρή και επεκτάσιμη αρχιτεκτονική, ικανή να διαχειρίζεται πολύπλοκες διεργασίες, να εξασφαλίζει την ορθή ροή δεδομένων και να υποστηρίζει μελλοντικές βελτιώσεις. Έχοντας παρουσιαστεί στο προηγούμενο κεφάλαιο το θεωρητικό υπόβαθρο και η λειτουργία των Εξελκτικών Αλγορίθμων, το παρόν κεφάλαιο επικεντρώνεται στην ανάλυση και σχεδίαση της Android εφαρμογής που τους ενσωματώνει. Στην εφαρμογή χρησιμοποιείται η αρχιτεκτονική προσέγγιση της Clean Architecture [67], η οποία δίνει έμφαση στη δομική οργάνωση του κώδικα και στον διαχωρισμό των επιπέδων λειτουργικότητας, εξασφαλίζοντας χαμηλό coupling και υψηλή επαναχρησιμοποίηση κώδικα. Ο όρος coupling αναφέρεται στον βαθμό αλληλεξάρτησης μεταξύ διαφορετικών τμημάτων του λογισμικού. Όσο μικρότερο είναι το coupling, τόσο πιο ανεξάρτητα μπορούν να τροποποιούνται και να επαναχρησιμοποιούνται τα επιμέρους μέρη χωρίς να επηρεάζεται το σύνολο του συστήματος.

Το κεφάλαιο αυτό παρουσιάζει λεπτομερώς τη συνολική αρχιτεκτονική του συστήματος, τις τεχνολογίες που χρησιμοποιήθηκαν, όπως Kotlin, Jetpack Compose, Koin, Ktor, Firebase και Room, καθώς και τον τρόπο με τον οποίο συνδυάζονται για την επίτευξη ενός ευέλικτου, ασφαλούς και λειτουργικού περιβάλλοντος.

4.2 Γενική αρχιτεκτονική της εφαρμογής

Όπως ήδη επισημάνθηκε στην εισαγωγή του παρόντος κεφαλαίου, η αρχιτεκτονική της εφαρμογής στηρίζεται στις αρχές της Clean Architecture, η οποία προτείνεται από τον Robert C. Martin [67]. Το πρότυπο αυτό στοχεύει στη δημιουργία λογισμικού που είναι εύκολα συντηρήσιμο, επεκτάσιμο και ανεξάρτητο από συγκεκριμένα τεχνολογικά πλαίσια ή βιβλιοθήκες. Στον πυρήνα της φιλοσοφίας του βρίσκεται ο διαχωρισμός της εφαρμογής σε διακριτά επίπεδα ευθύνης (layers), όπου κάθε επίπεδο έχει σαφή ρόλο και δεν γνωρίζει λεπτομέρειες υλοποίησης των υπολοίπων. Η επικοινωνία μεταξύ των επιπέδων δεν γίνεται απευθείας, αλλά μέσω αφηρημένων διεπαφών (abstractions), δηλαδή μέσω interfaces που περιγράφουν τη συμπεριφορά χωρίς να καθορίζουν τον τρόπο με τον οποίο αυτή υλοποιείται. Με τον τρόπο αυτόν, η εξάρτηση κατευθύνεται από τα εξωτερικά προς τα εσωτερικά επίπεδα, διασφαλίζοντας ότι η επιχειρησιακή λογική παραμένει ανεξάρτητη από τις τεχνολογικές λεπτομέρειες, όπως το περιβάλλον ανάπτυξης, η βάση δεδομένων ή η διεπαφή χρήστη.

Η συγκεκριμένη αρχιτεκτονική οργανώνεται γύρω από τέσσερα βασικά επίπεδα: Entities, Use Cases, Interface Adapters και Frameworks & Drivers. Τα Entities εκφράζουν τις θεμελιώδεις επιχειρησιακές οντότητες και τους κανόνες του συστήματος, ενώ τα Use Cases υλοποιούν τη λειτουργική λογική που ενορχηστρώνει τη συμπεριφορά των entities. Τα Interface Adapters αποτελούν το ενδιάμεσο επίπεδο που μετατρέπει τα δεδομένα από μορφές που κατανοούν τα εσωτερικά επίπεδα σε μορφές κατάλληλες για τα εξωτερικά, ενώ το εξωτερικό επίπεδο Frameworks & Drivers περιλαμβάνει τα τεχνολογικά στοιχεία που σχετίζονται με βάσεις δεδομένων, βιβλιοθήκες, APIs και το UI. Το σύνολο αυτών των επιπέδων σχηματίζει μια ακτινωτή δομή, στην οποία οι εξαρτήσεις δείχνουν πάντα προς το κέντρο, εκεί όπου κατοικεί η καθαρή επιχειρησιακή λογική [67].

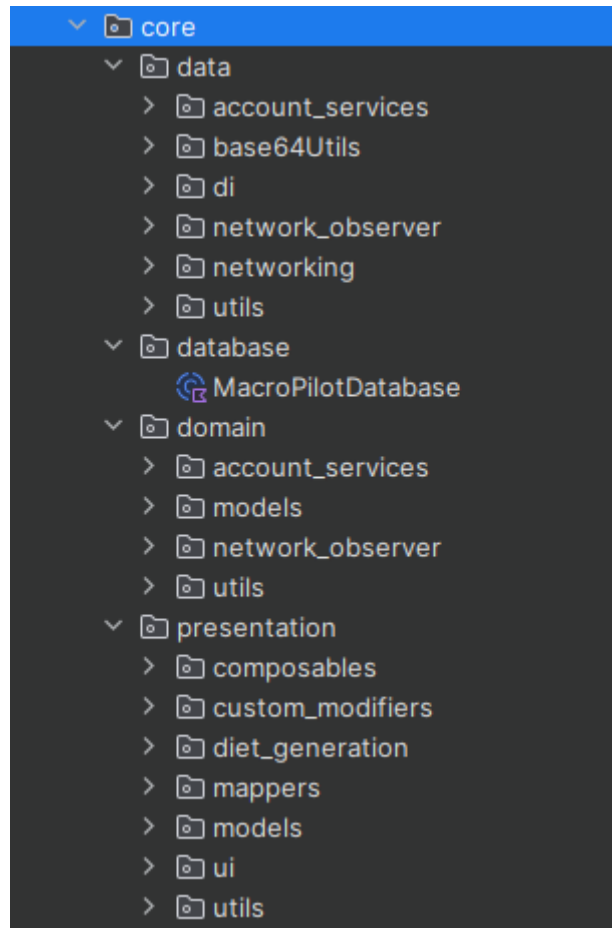
Στην πράξη, ωστόσο, η αυστηρή εφαρμογή αυτής της τετραπλής διάρθρωσης συχνά απλοποιείται ή προσαρμόζεται, ανάλογα με το πλαίσιο ανάπτυξης. Στην περίπτωση της παρούσας εφαρμογής, η Clean

Architecture υλοποιήθηκε με μια τριμερή προσαρμογή, που αποτελεί ευρέως αποδεκτό πρότυπο στον χώρο της ανάπτυξης Android εφαρμογών [68]. Έτσι, η αρχιτεκτονική διαρθρώνεται στα εξής επίπεδα: το domain layer, το data layer και το presentation layer. Το domain layer αντιστοιχεί στα Entities και Use Cases του αρχικού μοντέλου, περιλαμβάνοντας τους κανόνες της επιχειρησιακής λογικής και τις αφαιρέσεις (interfaces) των υπολοίπων επιπέδων. Το data layer ενσωματώνει τις υλοποιήσεις αυτών των αφαιρέσεων και αναλαμβάνει τη διαχείριση των δεδομένων, είτε αυτά προέρχονται από απομακρυσμένες πηγές, όπως APIs, είτε από τοπικές βάσεις δεδομένων. Τέλος, το presentation layer αφορά τη διεπαφή χρήστη και την παρουσίαση των δεδομένων, περιλαμβάνοντας ό,τι σχετίζεται με την απεικόνιση, την πλοήγηση και την αλληλεπίδραση με τον χρήστη. Η προσαρμογή αυτή της Clean Architecture στις ανάγκες της εφαρμογής συνοδεύτηκε από μια συνειδητή επιλογή, την παράλειψη των Use Cases. Ο λόγος αυτής της απόφασης είναι η επιδίωξη απλότητας και καθαρότητας στη δομή, δεδομένου ότι η αρχιτεκτονική MVI (Model-View-Intent), που χρησιμοποιείται στο presentation layer, και θα εξηγηθεί στη συνέχεια, καλύπτει μεγάλο μέρος της λειτουργικότητας που συνήθως ενσωματώνεται στα Use Cases. Η προσθήκη τους θα οδηγούσε σε περιττή πολυπλοκότητα, χωρίς να προσφέρει ουσιαστικά οφέλη στη συγκεκριμένη κλίμακα του έργου. Έτσι, διατηρείται η αρχή του διαχωρισμού ευθυνών, αλλά με μια πιο ευέλικτη και πρακτική προσέγγιση, προσαρμοσμένη στις πραγματικές ανάγκες της εφαρμογής.

Σχετικά με τη γενική αρχιτεκτονική που εφαρμόστηκε, είναι απαραίτητο να αναφερθεί, ότι η εφαρμογή είναι οργανωμένη σε επιμέρους λειτουργικές ενότητες (features), καθεμία εκ των οποίων αποτελεί ένα αυτόνομο πακέτο με τη δική του υλοποίηση των τριών βασικών επιπέδων. Κάθε layer αποτελεί ένα ξεχωριστό πακέτο μέσα στο feature package και κάθε layer χωρίζεται σε επιμέρους πακέτα, ανάλογα με τον τύπο των αρχείων που περιλαμβάνει. Παράλληλα, υπάρχει και ένα κοινό πακέτο, το λεγόμενο core package, το οποίο φιλοξενεί στοιχεία που χρησιμοποιούνται καθολικά από όλα τα features και θα χρησιμοποιηθεί ως παράδειγμα για την κατανόηση της δομής όλων των feature πακέτων του έργου. Το συγκεκριμένο πακέτο είναι οργανωμένο με την ίδια λογική των τριών layers, καθώς και με ένα επιπλέον πακέτο που σχετίζεται με την τοπική βάση δεδομένων και συγκεκριμένα περιλαμβάνει το αρχείο με τον κώδικα αρχικοποίησης της Room Database. Η δομή του πακέτου αυτού παρουσιάζεται στο Σχήμα 4.1.

Στο domain layer του core πακέτου ορίζονται οι αφαιρέσεις, δηλαδή τα interfaces και οι γενικοί τύποι που χρησιμοποιούνται από όλα τα features, επιτρέποντας τον καθαρό διαχωρισμό της επιχειρησιακής λογικής από τις εξαρτήσεις. Αντίστοιχα, στο data layer περιλαμβάνονται οι υλοποιήσεις αυτών των αφαιρέσεων, οι οποίες αφορούν λειτουργίες που μπορεί να χρησιμοποιούνται από διαφορετικά υποσυστήματα της εφαρμογής. Χαρακτηριστικό παράδειγμα αποτελεί η παρακολούθηση της συνδεσιμότητας της συσκευής στο διαδίκτυο, η οποία υλοποιείται μέσω interface που εκθέτει μια λογική μεταβλητή σχετική με τη διαθεσιμότητα δικτύου και αντίστοιχης κλάσης που την εφαρμόζει. Με αυτόν τον τρόπο, λειτουργίες όπως ο έλεγχος σύνδεσης στο διαδίκτυο είναι διαθέσιμες για κάθε feature χωρίς επανάληψη κώδικα.

Κάθε layer μπορεί να περιλαμβάνει ένα πακέτο, που ονομάζεται «utils» και περιλαμβάνει βοηθητικές συναρτήσεις (utilities), όπως μεταξύ άλλων, γενικοί τύποι αποτελεσμάτων (Result classes), οι οποίοι χρησιμοποιούνται εκτενώς για την τυποποιημένη διαχείριση επιτυχών και αποτυχημένων ενεργειών σε repositories και data sources. Η ύπαρξή τους διευκολύνει την ομοιόμορφη αντιμετώπιση σφαλμάτων, καθώς επιτρέπει τον διαχωρισμό των καταστάσεων επιτυχίας και αποτυχίας με τρόπο ασφαλή και εύκολα ανιχνεύσιμο.



Σχήμα 4.1: Η δομή του core package

Το data layer του core package περιλαμβάνει επίσης υπο-πακέτα για τη διαχείριση της αποθήκευσης δεδομένων και του δικτύου. Στο τμήμα που αφορά το δικτυακό επίπεδο για παράδειγμα, περιλαμβάνονται οι ρυθμίσεις για την αρχικοποίηση των συναρτήσεων της βιβλιοθήκης, που χρησιμοποιείται για τις κλήσεις των endpoints του API, καθώς και βοηθητικές επεκτάσεις για την ασφαλή εκτέλεση και διαχείριση αιτημάτων. Στο συγκεκριμένο layer, όπως και στο data layer κάθε feature, περιλαμβάνεται και το πακέτο «di», που περιλαμβάνει το αρχείο που καθορίζει την παροχή και διαχείριση των εξαρτήσεων στο εκάστοτε feature. Τεχνικές λεπτομέρειες σχετικά με τις προαναφερθείσες έννοιες θα παρουσιαστούν σε επόμενη υποενότητα, όπου θα αναλυθούν διεξοδικά οι τεχνολογίες που χρησιμοποιήθηκαν.

Το presentation layer κάθε feature είναι υπεύθυνο για τη διεπαφή με τον χρήστη και υλοποιείται με το αρχιτεκτονικό πρότυπο MVI. Η MVI αρχιτεκτονική αποτελεί ένα μοτίβο διαχείρισης κατάστασης (state management pattern) που στοχεύει στη δημιουργία σταθερών, προβλέψιμων και αντιδραστικών διεπαφών χρήστη. Σε αυτό το πρότυπο, το Model εκφράζει την τρέχουσα κατάσταση της διεπαφής, το View απεικονίζει οπτικά αυτή την κατάσταση και τα Intents (ή Actions/Events) αντιπροσωπεύουν τις ενέργειες του χρήστη ή τα σήματα του συστήματος που επιφέρουν αλλαγές στο Model. Το ViewModel λειτουργεί ως ενδιάμεσος μηχανισμός που λαμβάνει τα Intents, ενημερώνει την κατάσταση (State) και αποστέλλει τα αποτελέσματα πίσω στο View. Με αυτόν τον τρόπο, επιτυγχάνεται μια μονοκατευθυντική ροή δεδομένων, όπου κάθε αλλαγή στη διεπαφή προκύπτει αποκλειστικά από την ενημερωμένη κατάσταση του συστήματος, αποτρέποντας ασυνεπείς ή μη προβλέψιμες συμπεριφορές.

Αξίζει να σημειωθεί ότι συχνά τίθεται εσφαλμένα το ερώτημα «MVI ή Clean Architecture;». Στην πραγματικότητα, τα δύο αυτά πρότυπα δεν αποτελούν εναλλακτικές, αλλά λειτουργούν σε διαφορετικά επίπεδα και μπορούν και συνιστάται να χρησιμοποιούνται ταυτόχρονα. Η Clean Architecture αφορά τη γενική δομή του έργου και τον διαχωρισμό της επιχειρησιακής λογικής από τα τεχνολογικά πλαίσια, ενώ η MVI αφορά αποκλειστικά το presentation layer και τον τρόπο με τον οποίο το UI αντιδρά στις μεταβολές των δεδομένων. Συνεπώς, η συνδυαστική τους χρήση, όπως εφαρμόζεται και στην παρούσα ΔΕ, προσφέρει ένα σαφές, επεκτάσιμο και αποδοτικό πλαίσιο ανάπτυξης εφαρμογών Android.

4.3 Χρησιμοποιούμενες τεχνολογίες

Η επιτυχής υλοποίηση μιας εφαρμογής εξαρτάται σε μεγάλο βαθμό από την ορθή επιλογή των εργαλείων και των τεχνολογιών που τη στηρίζουν. Η επιλογή των τεχνολογιών στην παρούσα εργασία έγινε με γνώμονα την αξιοπιστία, τη σταθερότητα, την αποδοτικότητα και τη δυνατότητα μελλοντικής επεκτασιμότητας, ώστε να εξυπηρετείται η αρχιτεκτονική της εφαρμογής και να διευκολύνεται η διαδικασία ανάπτυξης.

Η ανάπτυξη πραγματοποιήθηκε στο Android Studio, το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) της Google για Android εφαρμογές. Το Android Studio βασίζεται στο IntelliJ IDEA και προσφέρει ένα σύνολο εργαλείων που επιτρέπουν την ολοκληρωμένη διαχείριση του κύκλου ζωής μιας εφαρμογής, από τη συγγραφή του κώδικα έως τη δοκιμή και τη διάθεσή της στο Play Store. Παρέχει υποστήριξη για τεχνολογίες όπως Jetpack Compose, Gradle Build System, εργαλεία debugging και προεπισκόπησης (previews), καθώς και εξομοιωτές για την αξιολόγηση της εφαρμογής σε διαφορετικές συσκευές. Η επιλογή του έγινε τόσο για την ευκολία ενσωμάτωσης των βιβλιοθηκών που χρησιμοποιήθηκαν όσο και για τη σταθερότητα και την ευρεία υποστήριξή του στη διεθνή κοινότητα προγραμματιστών [69].

Για τη διαχείριση του πηγαιού κώδικα και τη συνεργασία με αποθετήρια Git, χρησιμοποιήθηκε η εφαρμογή Fork, ένα γραφικό περιβάλλον για Git operations. Το Fork προσφέρει λειτουργίες όπως commit, branching, merging, conflict resolution και visualization του ιστορικού των αλλαγών, καθιστώντας τη διαδικασία διαχείρισης του έργου πιο άμεση και οργανωμένη. Η χρήση του συνέβαλε στην αποτελεσματική οπτικοποίηση της εξέλιξης του project, στη διατήρηση καθαρής ροής κώδικα και στην εύκολη επαναφορά σε προηγούμενες εκδόσεις, όταν αυτό ήταν απαραίτητο [70].

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Kotlin, μια σύγχρονη, στατικά τυποποιημένη (statically typed) γλώσσα, πλήρως συμβατή με το Java Virtual Machine (JVM). Η Kotlin έχει αναγνωριστεί επίσημα από τη Google ως η κύρια γλώσσα για ανάπτυξη Android εφαρμογών [71]. Η επιλογή της οφείλεται στα πλεονεκτήματά της σε σχέση με τη Java, όπως η αυξημένη ασφάλεια τύπων (null safety), η συνοπτικότητα, η υποστήριξη για coroutines που επιτρέπουν την αποδοτική διαχείριση ασύγχρονων διεργασιών, καθώς και η φυσική ενσωμάτωσή της με τις πιο σύγχρονες βιβλιοθήκες Android.

Για τη δημιουργία του γραφικού περιβάλλοντος χρησιμοποιήθηκε η Jetpack Compose, το σύγχρονο declarative UI framework της Google. Η Compose επιτρέπει την ανάπτυξη διεπαφών χρήστη με τη χρήση συναρτήσεων (composables), οι οποίες περιγράφουν το επιθυμητό UI χωρίς την ανάγκη για XML αρχεία, όπως συνέβαινε στις παλαιότερες προσεγγίσεις. Με αυτόν τον τρόπο, ο σχεδιασμός του UI γίνεται πιο αντιδραστικός και επεκτάσιμος, ενώ η κατάσταση της διεπαφής (UI state) συνδέεται άμεσα με τη λογική του προγράμματος, γεγονός που ευθυγραμμίζεται απόλυτα με την αρχιτεκτονική MVI που εφαρμόστηκε [72].

Η διαχείριση των εξαρτήσεων στην εφαρμογή πραγματοποιήθηκε μέσω της τεχνικής του Dependency Injection (DI) και της βιβλιοθήκης Koin. Το Dependency Injection είναι ένα πρότυπο σχεδίασης, που

στοχεύει στη δημιουργία λογισμικού με χαμηλό coupling και υψηλή επαναχρησιμοποίηση κώδικα. Ουσιαστικά, πρόκειται για έναν μηχανισμό μέσω του οποίου οι εξαρτήσεις των κλάσεων, δηλαδή τα αντικείμενα που χρειάζονται για να λειτουργήσουν, δεν δημιουργούνται εσωτερικά από τις ίδιες, αλλά παρέχονται εξωτερικά από ένα ειδικό σύστημα διαχείρισης [73], [74]. Με αυτόν τον τρόπο, κάθε κλάση δεν γνωρίζει πώς κατασκευάζονται τα αντικείμενα που χρησιμοποιεί, παρά μόνο τι είδους αντικείμενα χρειάζεται. Σε μικρές εφαρμογές, είναι εφικτό να γίνει χειροκίνητη διαχείριση εξαρτήσεων (manual dependency injection), δηλαδή η δημιουργία και παροχή αντικειμένων μέσω constructors ή factories χωρίς τη χρήση βιβλιοθήκης. Ωστόσο, στις σύγχρονες Android εφαρμογές, όπου υπάρχουν πολυάριθμες εξαρτήσεις, αυτή η μέθοδος οδηγεί σε πολύπλοκη και δύσκολα διαχειρίσιμη δομή, με επαναλαμβανόμενο κώδικα και αυξημένο κίνδυνο σφαλμάτων. Για τον λόγο αυτό, χρησιμοποιούνται βιβλιοθήκες DI που αυτοματοποιούν τη διαδικασία παροχής εξαρτήσεων και επιτρέπουν πιο καθαρή διαχείριση του κύκλου ζωής των αντικειμένων [74].

Η Koin επιλέχθηκε ως βιβλιοθήκη Dependency Injection για το έργο αυτό, λόγω της απλότητάς της, της εγγενούς υποστήριξής της στην Kotlin και της καταλληλότητάς της για μελλοντική Kotlin Multiplatform μετατροπή της εφαρμογής. Η Koin, σε αντίθεση με πιο πολύπλοκες λύσεις όπως το Dagger ή το Hilt, δεν απαιτεί code generation ή χρήση annotation processors, αλλά βασίζεται σε καθαρά Kotlin DSL συντακτικό για τη δημιουργία modules. Έτσι, επιτυγχάνεται ευκολότερη κατανόηση και μικρότερο κόστος συντήρησης. Επιπλέον, η δομή των modules στο Koin ευθυγραμμίζεται πλήρως με την Clean Architecture, καθώς κάθε feature διαθέτει το δικό του DI module, ενώ στο core package υπάρχει το app module, το οποίο παρέχει κοινές εξαρτήσεις, όπως το HttpClient, τη βάση δεδομένων και τα global ViewModels. Η Koin διαχειρίζεται αυτόματα τον κύκλο ζωής των αντικειμένων (singleton, factory, scoped) και παρέχει εύκολη ενσωμάτωση με το Android Lifecycle, εξασφαλίζοντας ότι κάθε εξάρτηση είναι διαθέσιμη μόνο στο πλαίσιο που χρειάζεται [75].

Για την επικοινωνία με το USDA FoodData Central API χρησιμοποιήθηκε η βιβλιοθήκη Ktor, μια μοντέρνα, ασύγχρονη πλατφόρμα δικτύωσης της JetBrains, γραμμένη εξ ολοκλήρου σε Kotlin. Παρότι η Retrofit αποτελεί την πλέον διαδεδομένη βιβλιοθήκη για HTTP επικοινωνία στο Android [76], η Ktor επιλέχθηκε για τους ίδιους λόγους που επιλέχθηκε και η Koin. Είναι μοντέρνα, απλή στη χρήση, πλήρως γραμμένη σε Kotlin και εύκολα επεκτάσιμη σε περιβάλλοντα Kotlin Multiplatform. Επιπλέον, παρέχει ενσωματωμένη υποστήριξη για coroutines, προσφέροντας ασύγχρονη επικοινωνία με βελτιστοποιημένη διαχείριση νημάτων [75], [77].

Στην εφαρμογή, η Ktor χρησιμοποιήθηκε για τη δημιουργία και διαχείριση του HTTP client, μέσω μιας custom υλοποίησης που επιτρέπει ασφαλείς και τυποποιημένες κλήσεις προς τα APIs. Συγκεκριμένα, έχει αναπτυχθεί ένα σύνολο από util functions που επεκτείνουν τη λειτουργικότητα του HttpClient (extension functions), προσθέτοντας μηχανισμούς όπως safeCall, responseToResult και constructRoute. Οι συναρτήσεις αυτές εξασφαλίζουν ότι κάθε αίτημα εκτελείται εντός ενός ασφαλούς περιβάλλοντος, διαχειρίζονται εξαιρέσεις (π.χ. UnresolvedAddressException, SerializationException), μετατρέπουν τις αποκρίσεις σε unified result types (Result.Success ή Result.Error), και κατασκευάζουν σωστά τις διαδρομές (routes) με βάση το BuildConfig.BASE_URL. Αυτό το modular και ασφαλές δίκτυο υποσυστήματος διευκολύνει τη συντήρηση, μειώνει την πιθανότητα σφαλμάτων και επιτρέπει τη συνεπή διαχείριση όλων των δικτυακών λειτουργιών.

Η πλατφόρμα Firebase της Google χρησιμοποιήθηκε για την παροχή cloud υπηρεσιών και backend λειτουργιών. Η Firebase αποτελεί ένα ολοκληρωμένο οικοσύστημα εργαλείων cloud, σχεδιασμένο για την υποστήριξη κινητών και web εφαρμογών, παρέχοντας υπηρεσίες όπως authentication, real-time databases, analytics, storage, push notifications και machine learning APIs [78]. Στην παρούσα εργασία αξιοποιήθηκαν δύο βασικές υπηρεσίες, το Firebase Authentication και το Firebase Firestore. Το Firebase Authentication επιτρέπει την ταυτοποίηση χρηστών μέσω email ή μέσω OAuth παρόχων, όπως η Google, εξασφαλίζοντας ασφαλή σύνδεση χωρίς την ανάγκη χειροκίνητης υλοποίησης μηχανισμών κρυπτογράφησης ή αποθήκευσης κωδικών. Ενσωματώνεται άμεσα με τα Android APIs και παρέχει έτοιμα σενάρια για επαναφορά κωδικών ή αυτόματη αποσύνδεση χρηστών. Από την άλλη

πλευρά, το Firebase Firestore είναι μια cloud-based NoSQL βάση δεδομένων, που αποθηκεύει τα δεδομένα σε έγγραφα (documents) και συλλογές (collections), προσφέροντας real-time συγχρονισμό μεταξύ εφαρμογής και cloud [78]. Στην εφαρμογή, χρησιμοποιείται για την αποθήκευση πληροφοριών χρήστη, ρυθμίσεων, και διατροφικού πλάνου, εξασφαλίζοντας αδιάλειπτη πρόσβαση και συνέπεια δεδομένων.

Για την τοπική αποθήκευση δεδομένων χρησιμοποιήθηκε η Room, η οποία αποτελεί τμήμα του Android Jetpack, του επίσημου οικοσυστήματος βιβλιοθηκών που αναπτύσσει η Google για τη διευκόλυνση της ανάπτυξης Android εφαρμογών. Το Android Jetpack περιλαμβάνει εργαλεία και βιβλιοθήκες που βοηθούν τους προγραμματιστές να εφαρμόζουν σύγχρονες αρχιτεκτονικές πρακτικές, να μειώνουν το boilerplate code και να εξασφαλίζουν τη σταθερότητα της εφαρμογής. Η Room λειτουργεί ως αντικειμενοστραφής διεπαφή προς τη SQLite, παρέχοντας abstraction layer που επιτρέπει τη διαχείριση δεδομένων μέσω κλάσεων και αντικειμένων αντί για απευθείας SQL ερωτήματα [79]. Υποστηρίζει annotations όπως @Entity, @Dao και @Database, καθιστώντας την πιο ευανάγνωστη και λιγότερο επιρρεπή σε σφάλματα. Στην παρούσα εφαρμογή, η Room χρησιμοποιείται για την προσωρινή αποθήκευση των δεδομένων που ανακτώνται από το API ή τη Firebase, παρέχοντας δυνατότητα πρόσβασης στα δεδομένα ακόμη και χωρίς ενεργή σύνδεση στο διαδίκτυο.

4.4 Απαιτήσεις χρήστη

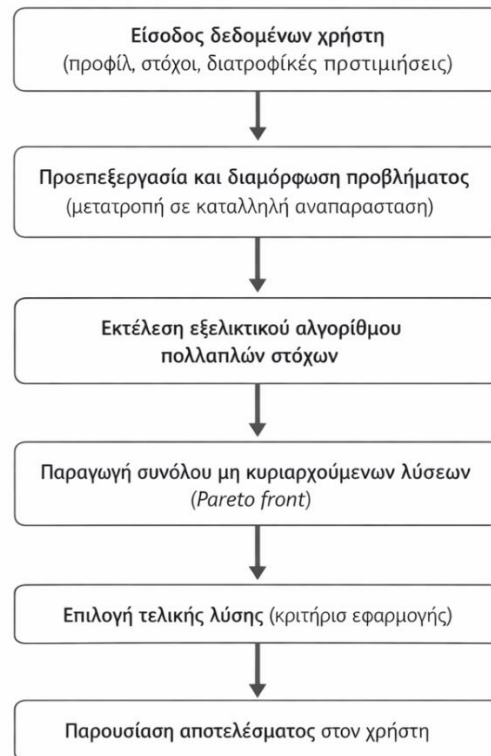
Η επιτυχία μιας εφαρμογής εξαρτάται σε μεγάλο βαθμό από τον βαθμό στον οποίο ανταποκρίνεται στις πραγματικές ανάγκες και προσδοκίες των χρηστών της. Στην παρούσα εργασία, η ανάπτυξη της εφαρμογής βασίστηκε στην αρχή του user-centered design, σύμφωνα με την οποία ο χρήστης αποτελεί το κεντρικό σημείο αναφοράς καθ' όλη τη διάρκεια του σχεδιασμού και της υλοποίησης. Η ανάλυση των απαιτήσεων χρήστη αποσκοπεί στον καθορισμό των βασικών λειτουργιών και χαρακτηριστικών που πρέπει να υποστηρίζει η εφαρμογή, ώστε να παρέχει μια ολοκληρωμένη, εξατομικευμένη και ευχάριστη εμπειρία. Οι απαιτήσεις που περιγράφονται αφορούν τόσο τις λειτουργικές ανάγκες, δηλαδή το σύνολο των ενεργειών που επιθυμεί να εκτελέσει ο χρήστης, όσο και τις μη λειτουργικές ανάγκες, που σχετίζονται με την ποιότητα και τη συμπεριφορά της εφαρμογής.

Αρχικά, ο χρήστης πρέπει να μπορεί να πραγματοποιεί εγγραφή και σύνδεση στο σύστημα με ασφαλή και απλό τρόπο. Η διαδικασία αυτή υλοποιείται μέσω του Firebase Authentication, επιτρέποντας την είσοδο με email και κωδικό ή μέσω του λογαριασμού Google. Μετά τη σύνδεση, ο χρήστης πρέπει να έχει τη δυνατότητα δημιουργίας και διαχείρισης προφίλ, εισάγοντας προσωπικά δεδομένα όπως όνομα, ηλικία, βάρος, ύψος, φύλο και διατροφικό στόχο (απώλεια, διατήρηση ή αύξηση βάρους). Παράλληλα, θα πρέπει να μπορεί να καθορίζει διατροφικές προτιμήσεις και περιορισμούς, όπως αν ακολουθεί χορτοφαγική ή κετογονική διατροφή, αν αποφεύγει τα γαλακτοκομικά ή αν έχει συγκεκριμένες αλλεργίες, ώστε το προτεινόμενο πλάνο να προσαρμόζεται στις εξατομικευμένες ανάγκες του. Όλα αυτά τα χαρακτηριστικά θα μπορούν να τροποποιούνται μέσα από μια ξεχωριστή ενότητα μέσα στην εφαρμογή που θα αφορά το προφίλ του χρήστη.

Μία από τις σημαντικότερες απαιτήσεις του χρήστη είναι η δυνατότητα παραγωγής εξατομικευμένου διατροφικού πλάνου. Η εφαρμογή πρέπει να επιτρέπει στον χρήστη να δημιουργεί αυτόματα ένα πρόγραμμα γευμάτων, βασισμένο στα στοιχεία και στους στόχους που έχει καταχωρήσει. Το πλάνο αυτό παράγεται μέσω του εξελικτικού αλγορίθμου που υλοποιήθηκε με τη βιβλιοθήκη Jenetics, λαμβάνοντας υπόψη θερμίδες, μακροθρεπτικά και μικροθρεπτικά συστατικά, καθώς και περιορισμούς ή προτιμήσεις. Ο χρήστης πρέπει επίσης να έχει τη δυνατότητα προβολής του παραγόμενου πλάνου, επανεκτέλεσης του αλγορίθμου, εφόσον επιθυμεί να δημιουργήσει ένα νέο πλάνο, καθώς και να επεξεργάζεται χειροκίνητα το υπάρχον, προσθέτοντας ή αφαιρώντας τρόφιμα, τροποποιώντας ποσότητες ή εισάγοντας δικές του τροφές. Έτσι, διασφαλίζεται ο συνδυασμός αυτοματοποίησης και

Κεφάλαιο 4

ελέγχου από τον ίδιο τον χρήστη. Το ακόλουθο διάγραμμα που ακολουθεί στο Σχήμα 4.2, παρουσιάζει τη ροή εκτέλεσης του συστήματος σε υψηλό επίπεδο, από τη συλλογή των δεδομένων εισόδου έως την παραγωγή και παρουσίαση της τελικού πλάνου.



Σχήμα 4.2: Υψηλού επιπέδου ροή εκτέλεσης του εξελικτικού αλγορίθμου πολλαπλών στόχων στο πλαίσιο της εφαρμογής.

Όπως φαίνεται στο σχήμα, ο εξελικτικός αλγόριθμος αποτελεί κεντρικό στάδιο της ροής, λειτουργώντας ως μηχανισμός παραγωγής εναλλακτικών λύσεων με βάση τα δεδομένα και τις προτιμήσεις του χρήστη. Η αναλυτική περιγραφή της υλοποίησης του αλγορίθμου και των επιμέρους βημάτων του παρουσιάζεται στο Κεφάλαιο 5.

Επιπλέον, ο χρήστης αναμένεται να έχει ανάγκη από εργαλεία παρακολούθησης προόδου. Η εφαρμογή πρέπει να επιτρέπει την καταγραφή και απεικόνιση της μεταβολής του βάρους μέσα στον χρόνο, παρέχοντας τόσο αριθμητικά δεδομένα όσο και γραφικές απεικονίσεις. Η δυνατότητα αυτή ενισχύει την παρακολούθηση των αποτελεσμάτων, λειτουργεί παρακινητικά και επιτρέπει την αξιολόγηση της αποτελεσματικότητας του διατροφικού προγράμματος. Επιπλέον, η εφαρμογή πρέπει να παρέχει ειδοποιήσεις και υπενθυμίσεις, για παράδειγμα πριν από τα γεύματα, συμβάλλοντας στη διατήρηση συνέπειας και στην καθοδήγηση του χρήστη στην καθημερινή του ρουτίνα.

Εκτός από τις λειτουργικές απαιτήσεις, υπάρχουν και μη λειτουργικές απαιτήσεις που σχετίζονται με την ποιότητα και τη συμπεριφορά του συστήματος. Η εφαρμογή πρέπει να είναι εύχρηστη και διαισθητική, ώστε ακόμη και χρήστες χωρίς τεχνικές γνώσεις να μπορούν να την αξιοποιήσουν αποτελεσματικά. Πρέπει να χαρακτηρίζεται από σταθερότητα και ταχύτητα, εξασφαλίζοντας ομαλή

εμπειρία χωρίς καθυστερήσεις, ακόμη και σε συσκευές με περιορισμένους πόρους. Η ασφάλεια των δεδομένων του χρήστη αποτελεί επίσης κρίσιμο παράγοντα. Για τον λόγο αυτό, η αποθήκευση και η διαχείριση των προσωπικών πληροφοριών γίνεται με τη χρήση ασφαλών μηχανισμών ταυτοποίησης και πιστοποίησης. Παράλληλα, η εφαρμογή οφείλει να λειτουργεί αξιόπιστα σε περιβάλλοντα με μεταβαλλόμενη συνδεσιμότητα, επιτρέποντας τη χρήση βασικών λειτουργιών ακόμη και χωρίς πρόσβαση στο διαδίκτυο, μέσω τοπικής αποθήκευσης δεδομένων στη βάση Room. Τέλος, πρέπει να εξασφαλίζεται η συμβατότητα με σύγχρονες εκδόσεις του Android λειτουργικού, με ελάχιστη υποστήριξη από την έκδοση 10 και άνω, ώστε να καλύπτεται το μεγαλύτερο ποσοστό των χρηστών. Ο τρόπος με τον οποίο ικανοποιούνται οι ανάγκες αυτές, καθώς και η ανάλυση της λειτουργίας κάθε επιμέρους οθόνης και μηχανισμού της εφαρμογής, θα παρουσιαστούν αναλυτικά στο επόμενο κεφάλαιο.

4.5 Σχεδίαση βάσης δεδομένων και REST API

Οι NoSQL βάσεις δεδομένων, όπως η Firebase Firestore, ανήκουν σε μια κατηγορία συστημάτων αποθήκευσης δεδομένων που διαφοροποιούνται από τις παραδοσιακές σχεσιακές βάσεις SQL ως προς τη δομή, την ευελιξία και τον τρόπο κλιμάκωσης. Σε αντίθεση με τις σχεσιακές βάσεις, όπου τα δεδομένα οργανώνονται σε πίνακες και συσχετίζονται μεταξύ τους μέσω ξένων κλειδίων, οι NoSQL βάσεις αποθηκεύουν τα δεδομένα σε μορφή εγγράφων (documents) ή αντικειμένων (objects) μέσα σε συλλογές (collections). Κάθε document περιλαμβάνει ένα σύνολο πεδίων και τιμών, τα οποία μπορούν να διαφέρουν μεταξύ των εγγράφων της ίδιας συλλογής, παρέχοντας έτσι αυξημένη ευελιξία στη διαχείριση δομημένων ή ημι-δομημένων δεδομένων [80]. Η αρχιτεκτονική αυτή επιτρέπει τη δυναμική προσαρμογή του σχήματος, την υψηλή απόδοση σε αναγνώσεις και εγγραφές και την εύκολη κλιμάκωση σε εφαρμογές που απαιτούν αποθήκευση μεγάλου όγκου δεδομένων με γρήγορη προσπέλαση [81].

Στην παρούσα εργασία, η σχεδίαση της βάσης δεδομένων στηρίχθηκε στη Firebase Firestore, με κύρια συλλογή (collection) το users, η οποία περιλαμβάνει όλα τα έγγραφα που αντιστοιχούν στους εγγεγραμμένους χρήστες της εφαρμογής. Κάθε νέος χρήστης που δημιουργεί λογαριασμό μέσω του Firebase Authentication λαμβάνει αυτόματα ένα μοναδικό αναγνωριστικό χρήστη (UID), το οποίο χρησιμοποιείται ως όνομα εγγράφου (document ID) μέσα στο collection users. Κάθε document περιλαμβάνει βασικά πεδία που περιγράφουν τον χρήστη, όπως το αναγνωριστικό, τη διεύθυνση email, το όνομα, το φύλο, την ηλικία, το ύψος, το βάρος, το επίπεδο φυσικής δραστηριότητας, τον στόχο (απώλεια, διατήρηση ή αύξηση βάρους), καθώς και πληροφορίες για αλλεργίες και διατροφικές προτιμήσεις. Περιλαμβάνονται επίσης τα είδη γευμάτων που επιτρέπονται, οι θερμίδες-στόχος και η επιθυμητή κατανομή των μακροθρεπτικών και μικροθρεπτικών συστατικών. Παρόλο που η Firestore είναι schema-less, η εφαρμογή ακολουθεί σταθερή δομή εγγράφων για κάθε χρήστη, ώστε να διασφαλίζεται συνέπεια και απλοποιημένη αναζήτηση δεδομένων.

Κάθε έγγραφο χρήστη μπορεί να διαθέτει τρεις υποσυλλογές (sub-collections), τις customFoods, diets και weightProgress, οι οποίες δημιουργούνται δυναμικά μόνο όταν απαιτείται. Η υποσυλλογή customFoods περιλαμβάνει τα φαγητά που έχει δημιουργήσει ο ίδιος ο χρήστης. Κάθε document σε αυτή τη συλλογή περιέχει μια λίστα από αντικείμενα που αναπαριστούν τα τρόφιμα, με πεδία όπως μοναδικό αναγνωριστικό, περιγραφή, τύπο γεύματος και λίστα θρεπτικών συστατικών κάθε ένα από τα οποία αποθηκεύεται σε μορφή συμβολοσειράς με μορφή JSON. Η υποσυλλογή diets δημιουργείται αυτόματα όταν παραχθεί το πρώτο διατροφικό πλάνο μέσω του εξελικτικού αλγορίθμου. Περιλαμβάνει ένα document με όνομα dietPlan, στο οποίο αποθηκεύεται η τρέχουσα δίαιτα του χρήστη. Το πεδίο foods περιλαμβάνει τη λίστα των φαγητών του πλάνου, όπου κάθε στοιχείο της λίστας περιέχει

πληροφορίες για το φαγητό, την ποσότητα και τη θρεπτική του αξία, έχει δηλαδή ίδια δομή με τα τρόφιμα που αποθηκεύονται στο customFoods έγγραφο. Η υποσυλλογή weightProgress δημιουργείται μόλις ο χρήστης καταχωρήσει για πρώτη φορά το βάρος του. Περιλαμβάνει πολλαπλά documents, καθένα από τα οποία έχει ως όνομα την ημερομηνία της καταχώρησης και ένα μόνο πεδίο weight. Εάν ο χρήστης ενημερώσει το βάρος του πολλές φορές την ίδια ημέρα, η εφαρμογή ενημερώνει το υπάρχον document αντί να δημιουργήσει νέο, εξασφαλίζοντας έτσι τη βέλτιστη αξιοποίηση της μνήμης και τη σταθερότητα της δομής.

Η επιλογή αυτής της αρχιτεκτονικής έχει σημαντικά πλεονεκτήματα. Με τη χρήση υποσυλλογών, κάθε χρήστης διατηρεί πλήρως απομονωμένα δεδομένα, γεγονός που απλοποιεί τη διαχείριση και ενισχύει την ασφάλεια. Επιπλέον, η Firestore διαχειρίζεται αυτόματα τη συσχέτιση των δεδομένων, με αποτέλεσμα η διαγραφή ενός χρήστη να συνεπάγεται και τη διαγραφή όλων των σχετικών υποσυλλογών του, χωρίς την ανάγκη ξεχωριστής διαγραφής για κάθε κατηγορία δεδομένων. Αντίθετα, αν οι ίδιες πληροφορίες ήταν αποθηκευμένες σε ανεξάρτητα collections στο ίδιο επίπεδο, η διαδικασία καθαρισμού των δεδομένων θα ήταν πολύ πιο σύνθετη και επιρρεπής σε σφάλματα.

Πέρα από την Firebase βάση δεδομένων, η εφαρμογή αξιοποιεί ένα εξωτερικό REST API, συγκεκριμένα το USDA FoodData Central API, για την ανάκτηση πληροφοριών σχετικά με τρόφιμα και τα διατροφικά τους στοιχεία. Από τα διαθέσιμα endpoints, χρησιμοποιείται το `/foods/search`, το οποίο υποστηρίζει τόσο τη μέθοδο GET όσο και τη POST. Η κύρια διαφορά μεταξύ των δύο είναι ότι το GET χρησιμοποιείται για απλές αναζητήσεις, μεταφέροντας τα ερωτήματα ως παραμέτρους URL, ενώ το POST επιτρέπει πιο σύνθετες αναζητήσεις, στέλλοντας ένα αντικείμενο αναζήτησης (request body) σε μορφή JSON, που περιλαμβάνει περισσότερες επιλογές φιλτραρίσματος. Στην εφαρμογή επιλέχθηκε η μέθοδος POST, καθώς παρέχει μεγαλύτερη ευελιξία στον καθορισμό παραμέτρων και αποφεύγει περιορισμούς μήκους URL που ισχύουν για τα GET αιτήματα. Στο συγκεκριμένο endpoint, χρησιμοποιείται η λογική της σελιδοποίησης των δεδομένων. Η σελιδοποίηση (pagination) είναι μια τεχνική που επιτρέπει τη διάσπαση μεγάλων συνόλων δεδομένων σε επιμέρους «σελίδες», οι οποίες ανακτώνται σταδιακά μέσω παραμέτρων. Με αυτόν τον τρόπο μειώνεται ο φόρτος του διακομιστή και βελτιστοποιείται η απόδοση της εφαρμογής, καθώς τα δεδομένα μεταφέρονται σταδιακά και όχι όλα μαζί, επιτρέποντας παράλληλα την υλοποίηση δυνατοτήτων όπως το infinite scrolling ή η προ-φόρτωση νέων δεδομένων μόνο όταν χρειάζεται.

Σύμφωνα με το σχήμα αιτήματος που ορίζεται στο επίσημο API documentation του USDA [82], [83], όπως φαίνεται στο Παράρτημα Α, ένα αίτημα προς το `/foods/search` endpoint μπορεί να περιλαμβάνει πεδία όπως `query`, `dataType`, `pageNumber`, `pageSize`, `sortBy`, `brandOwner`, `ingredients` και άλλα. Στο πλαίσιο της παρούσας εργασίας, χρησιμοποιούνται μόνο τα `query`, `dataType`, `pageNumber` και `pageSize`, καθώς η εφαρμογή εστιάζει στην ανάκτηση γενικών δεδομένων τροφίμων και όχι εμπορικών προϊόντων ή συστατικών. Το πεδίο `query` καθορίζει τη λέξη-κλειδί αναζήτησης, ενώ το `dataType` περιορίζεται στις τιμές «Survey (FNDDS)» και «Foundation», ώστε να ανακτώνται αποκλειστικά τρόφιμα με πλήρη και αξιόπιστα δεδομένα θρεπτικής ανάλυσης. Το `pageNumber` και το `pageSize` καθορίζουν τη σελιδοποίηση των αποτελεσμάτων, με την εφαρμογή να ανακτά έως 200 τρόφιμα ανά σελίδα για λόγους απόδοσης.

Το σχήμα απόκρισης, όπως φαίνεται στο Παράρτημα Β, περιλαμβάνει πλήθος πληροφοριών, όπως `totalHits`, `currentPage`, `pageList` και `foods` [82], [83]. Από αυτά, η εφαρμογή χρησιμοποιεί κυρίως τη λίστα `foods`, η οποία περιέχει αντικείμενα με πεδία όπως `fdcId`, `description`, `dataType` και `foodNutrients`. Από το `foodNutrients`, αξιοποιούνται τα θρεπτικά δεδομένα (π.χ. πρωτεΐνες, υδατάνθρακες, λιπαρά,

βιταμίνες και μέταλλα), τα οποία μετατρέπονται σε εσωτερικά αντικείμενα της εφαρμογής και χρησιμοποιούνται τόσο για τη δημιουργία των διατροφικών πλάνων όσο και για την προβολή της διατροφικής αξίας κάθε φαγητού.

4.6 Επίλογος

Το παρόν κεφάλαιο παρουσιάζει τις βασικές αρχιτεκτονικές και σχεδιαστικές επιλογές που διέπουν την υλοποίηση της εφαρμογής, εστιάζοντας στη δομημένη οργάνωση των επιμέρους συνιστωσών και στην καθαρή κατανομή των ρόλων και των ευθυνών τους. Η επιλογή συγκεκριμένων αρχιτεκτονικών προτύπων και τεχνολογιών στοχεύει στη διασφάλιση επεκτασιμότητας, συντηρησιμότητας και σαφήνειας στον σχεδιασμό, δημιουργώντας ένα σταθερό υπόβαθρο για την ανάπτυξη και τη μελλοντική εξέλιξη του συστήματος. Στο πλαίσιο αυτό, η ενσωμάτωση της αλγοριθμικής λογικής πραγματοποιείται με τρόπο που ευθυγραμμίζεται με τη συνολική αρχιτεκτονική φιλοσοφία, χωρίς να επηρεάζει την ανεξαρτησία των υπολοίπων επιπέδων. Στο επόμενο κεφάλαιο παρουσιάζεται η συνολική υλοποίηση της εφαρμογής, στο πλαίσιο της οποίας αναλύεται και η υλοποίηση του εξελικτικού αλγορίθμου που αξιοποιείται για την παραγωγή των διατροφικών πλάνων.

Κεφάλαιο 5ο: Υλοποίηση εφαρμογής

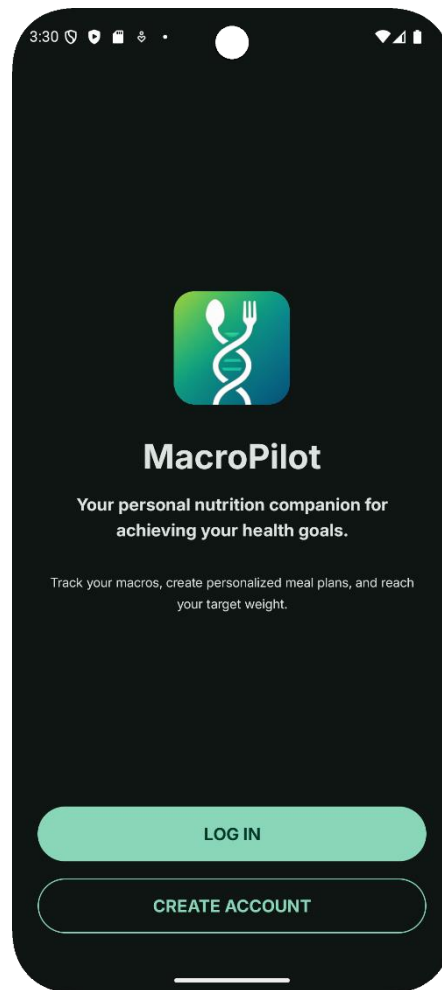
5.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζεται η υλοποίηση της εφαρμογής που αναπτύχθηκε στο πλαίσιο της διπλωματικής εργασίας. Αφού στο προηγούμενο κεφάλαιο αναλύθηκε η αρχιτεκτονική, οι τεχνολογίες και οι λειτουργικές προδιαγραφές του συστήματος, εδώ περιγράφεται αναλυτικά η πρακτική εφαρμογή τους στο τελικό προϊόν. Η ενότητα εστιάζει στη ροή λειτουργίας της εφαρμογής, που ονομάστηκε «MacroPilot», από την πρώτη εκτέλεση έως την πλήρη αλληλεπίδραση του χρήστη με τα επιμέρους χαρακτηριστικά της. Εξετάζονται οι μηχανισμοί αυθεντικοποίησης και δημιουργίας λογαριασμού, η διαδικασία αρχικοποίησης και ρύθμισης του προφίλ, η παραγωγή του διατροφικού πλάνου μέσω εξελικτικού αλγορίθμου, καθώς και οι λειτουργίες διαχείρισης και επεξεργασίας των παραγόμενων δεδομένων. Παράλληλα αναλύεται η συνεργασία των επιμέρους υποσυστημάτων, όπως η επικοινωνία με το USDA API, η χρήση του Firebase Authentication και του Firestore Database, καθώς και η αξιοποίηση της τοπικής βάσης Room για αποθήκευση εκτός σύνδεσης. Ιδιαίτερη έμφαση δίνεται στη λειτουργική συνοχή μεταξύ των οθονών και στη λογική που εκτελείται στο παρασκήνιο κατά την εκτέλεση των ενεργειών του χρήστη, ώστε να αποτυπωθεί πλήρως η διαδικασία μετατροπής του θεωρητικού σχεδίου σε λειτουργική εφαρμογή.

5.2 Αυθεντικοποίηση χρήστη

Η αυθεντικοποίηση του χρήστη αποτελεί το πρώτο στάδιο αλληλεπίδρασης με την εφαρμογή και διασφαλίζει την ασφαλή πρόσβαση στα προσωπικά δεδομένα και τις εξατομικευμένες λειτουργίες της. Με την πρώτη εκτέλεση της εφαρμογής εμφανίζεται η οθόνη καλωσορίσματος, η οποία φαίνεται στο Σχήμα 5.1 και παρουσιάζει το λογότυπο και μια σύντομη περιγραφή των δυνατοτήτων της εφαρμογής, προσφέροντας δύο επιλογές: «Login» για σύνδεση σε υπάρχοντα λογαριασμό και «Create Account» για τη δημιουργία νέου. Η επιλογή κάποιας από αυτές οδηγεί τον χρήστη στις αντίστοιχες οθόνες σύνδεσης ή εγγραφής.

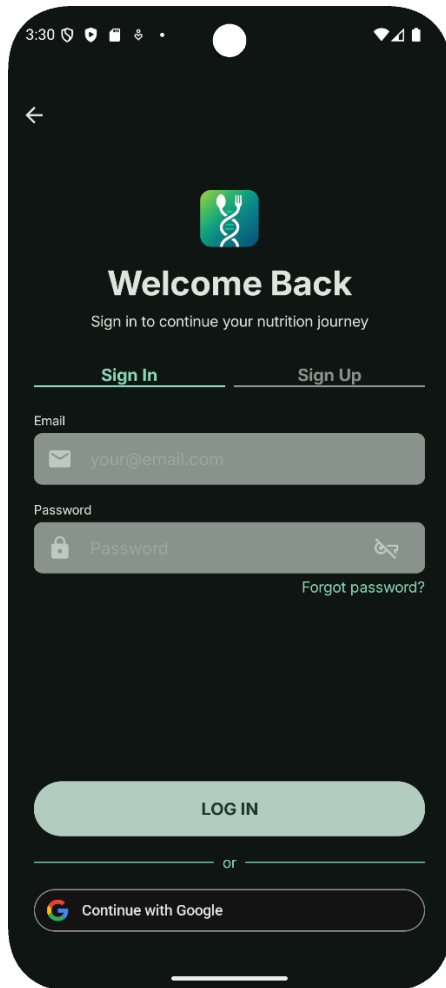
Στην οθόνη σύνδεσης (Σχήμα 5.2) υπάρχει επάνω γραμμή εργαλείων με δυνατότητα επιστροφής στο προηγούμενο βήμα. Ακολουθεί το λογότυπο, ένα συνοδευτικό μήνυμα και οι καρτέλες «Sign In» και «Sign Up», οι οποίες επιτρέπουν την εναλλαγή μεταξύ των δύο λειτουργιών χωρίς ανάγκη επιστροφής στην αρχική οθόνη. Ανάλογα με την επιλεγμένη λειτουργία, η αντίστοιχη επιλογή εμφανίζεται με πιο έντονο χρώμα για οπτική καθοδήγηση του χρήστη. Κατά τη φόρτωση της οθόνης, εάν στη συσκευή υπάρχει αποθηκευμένος λογαριασμός Google, ενεργοποιείται αυτόματα ο Credentials Manager (Σχήμα 5.3), ένα εργαλείο του Android που διαχειρίζεται τους αποθηκευμένους λογαριασμούς και κωδικούς πρόσβασης, επιτρέποντας την αυτόματη εισαγωγή τους. Ο χρήστης μπορεί να επιλέξει έναν από τους διαθέσιμους λογαριασμούς, είτε Google είτε κάποιον λογαριασμό που δημιούργησε χειροκίνητα στην εφαρμογή και αποθήκευσε τα στοιχεία του, ώστε η διαδικασία σύνδεσης να ολοκληρωθεί άμεσα χωρίς πληκτρολόγηση. Σε περίπτωση που επιλεγεί λογαριασμός Google, η αυθεντικοποίηση πραγματοποιείται μέσω της μεθόδου `signInWithCredential(AuthCredential)`, που παρέχεται από τη βιβλιοθήκη του Firebase Authentication, όπου το αντικείμενο `AuthCredential` δημιουργείται με την κλήση `GoogleAuthProvider.getCredential(String, String)`. Αν ο χρήστης επιλέξει αποθηκευμένα στοιχεία της εφαρμογής, εκτελείται η ίδια διαδικασία που ακολουθείται στο χειροκίνητο login.



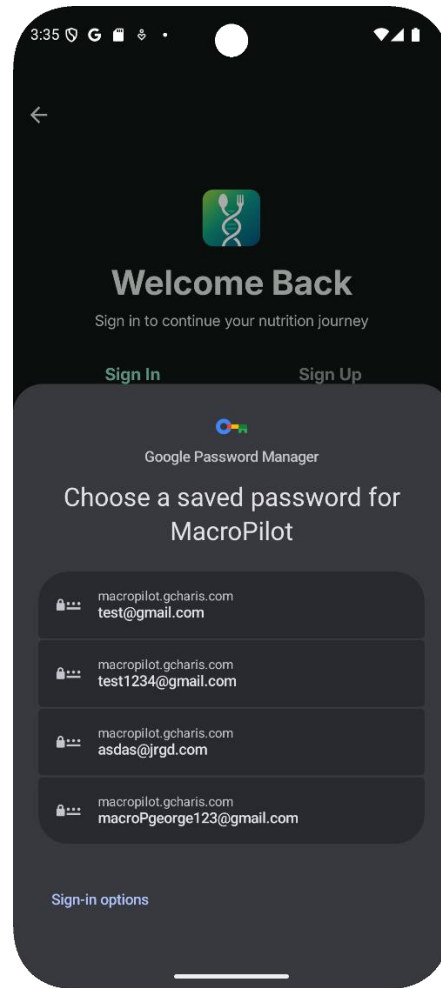
Σχήμα 5.1: Η οθόνη καλωσορίσματος

Ο χρήστης έχει τη δυνατότητα να απορρίψει τον Credentials Manager, αγγίζοντας εκτός του παραθύρου του. Στην περίπτωση αυτή μπορεί να προχωρήσει σε χειροκίνητη εισαγωγή των στοιχείων του στα πεδία Email και Password. Το κουμπί «Login» παραμένει ανενεργό μέχρι να πληρούνται συγκεκριμένοι κανόνες εγκυρότητας: το email πρέπει να ακολουθεί τη σωστή μορφή, ενώ ο κωδικός να έχει τουλάχιστον εννέα χαρακτήρες, έναν αριθμό, ένα κεφαλαίο και ένα πεζό γράμμα. Σε περίπτωση λανθασμένης εισαγωγής εμφανίζεται μήνυμα σφάλματος πάνω από το αντίστοιχο πεδίο. Στο πεδίο του κωδικού υπάρχει επιλογή προβολής ή απόκρυψης του περιεχομένου, ενώ στο πεδίο του email εμφανίζεται ένδειξη επιτυχίας όταν το format είναι σωστό. Η επιτυχής συμπλήρωση ενεργοποιεί το κουμπί «Login», το οποίο αλλάζει χρωματικά για να υποδείξει ότι είναι διαθέσιμο. Με το πάτημά του, εκτελείται η συνάρτηση `firebase.auth.signInWithEmailAndPassword(String email, String password)` που πραγματοποιεί τον έλεγχο των διαπιστευτηρίων, ενώ όσο εκτελείται αυτή η συνάρτηση εμφανίζεται ένας κυκλικός δείκτης προόδου, ώστε να σηματοδοτήσει στο χρήστη την κατάσταση loading.

Κάτω από το κύριο κουμπί σύνδεσης υπάρχει η επιλογή Continue with Google, η οποία επαναφέρει τον Credentials Manager και χρησιμοποιείται για αυθεντικοποίηση μέσω Google. Σε περίπτωση που δεν υπάρχει συνδεδεμένος λογαριασμός Google στη συσκευή, το κουμπί αυτό οδηγεί σε μια ροή που επιτρέπει τον χρήστη να συνδέσει τέτοιου είδους λογαριασμό. Η ίδια συνάρτηση `signInWithCredential(AuthCredential)`, καλύπτει τόσο τη σύνδεση όσο και την εγγραφή, καθώς η συνάρτηση δημιουργεί αυτόματα λογαριασμό αν δεν υπάρχει. Επιπλέον, υπάρχει η επιλογή «Forgot your password», η οποία οδηγεί στην οθόνη επαναφοράς κωδικού (Σχήμα 5.4). Εκεί ο χρήστης εισάγει



Σχήμα 5.2: Η οθόνη σύνδεσης



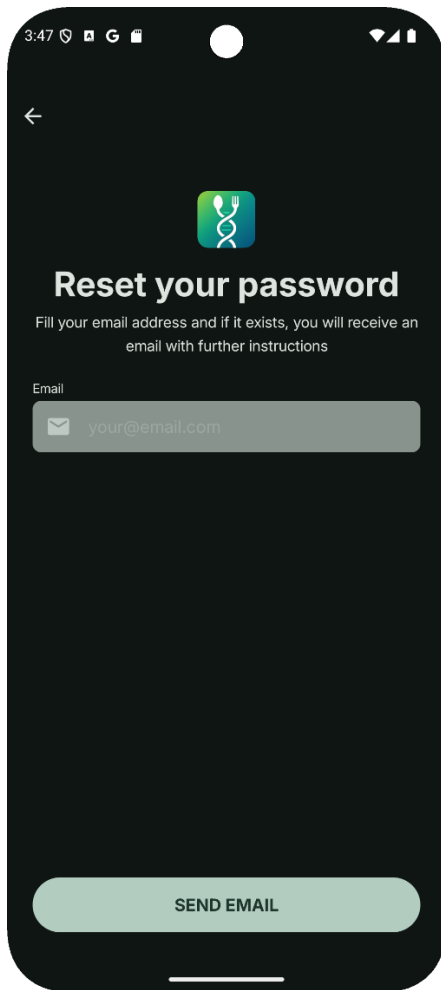
Σχήμα 5.3: Ο Credentials Manager στην οθόνη σύνδεσης

το email του και εφόσον αντιστοιχεί σε υπάρχον λογαριασμό, αποστέλλεται αυτόματα μήνυμα επαναφοράς μέσω της μεθόδου `Firestore.auth.sendPasswordResetEmail(String email)`. Η αποστολή και η διαδικασία επαναφοράς διαχειρίζονται εξ ολοκλήρου από το Firebase Authentication.

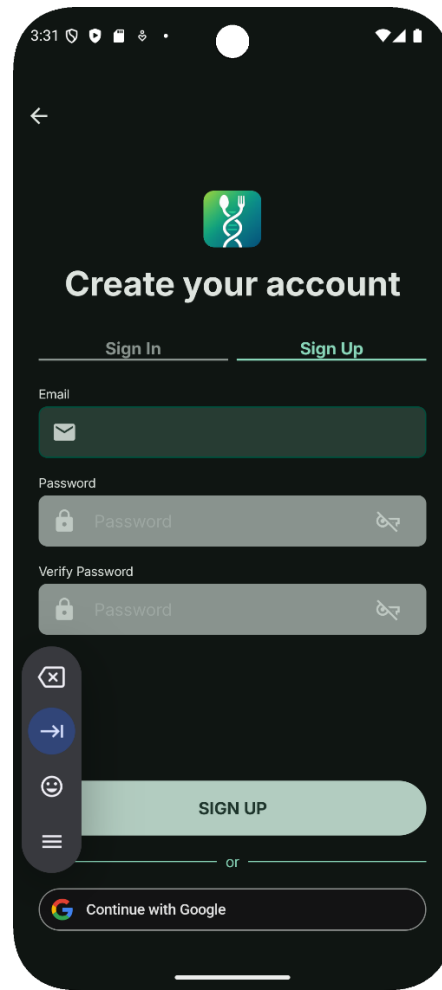
Η οθόνη εγγραφής, που απεικονίζεται στο Σχήμα 5.5, περιλαμβάνει πεδία για email, κωδικό και επαλήθευση κωδικού, καθώς και τα κουμπιά Sign Up και Continue with Google, όπως αναφέρθηκε προηγουμένως. Το κουμπί εγγραφής ενεργοποιείται μόνο όταν τα δεδομένα είναι έγκυρα και οι δύο κωδικοί ταυτίζονται. Η συνάρτηση που εκτελείται κατά την εγγραφή του χρήστη είναι η `Firestore.auth.createUserWithEmailAndPassword(String email, String password)`, η οποία δημιουργεί έναν καινούριο χρήστη στο Firebase Authentication της εφαρμογής. Και σε αυτή την περίπτωση εμφανίζεται η κατάσταση loading, μέσω ενός κυκλικού δείκτη προόδου, κατά τη διάρκεια εκτέλεσης της συνάρτησης. Μετά από επιτυχή εγγραφή, ο χρήστης ερωτάται αν επιθυμεί να αποθηκεύσει τα στοιχεία του στον Credentials Manager, εφόσον και πάλι υπάρχει συσχετιζόμενος λογαριασμός Google, και στη συνέχεια μεταφέρεται στην οθόνη διαμόρφωσης προφίλ.

Για όλες τις παραπάνω λειτουργίες απαιτείται ενεργή σύνδεση στο διαδίκτυο. Αν επιχειρηθεί ενέργεια χωρίς σύνδεση, εμφανίζεται ενημερωτικό παράθυρο (modal) που ειδοποιεί τον χρήστη, ενώ σε περιπτώσεις σφαλμάτων από τον διακομιστή ή την Firebase εμφανίζονται κατάλληλα μηνύματα (toasts) με την αιτία του προβλήματος, όπως για παράδειγμα εάν τα στοιχεία του που καταχώρησε ο χρήστης κατά την χειροκίνητη σύνδεση, δεν αντιστοιχούν σε λογαριασμό που υπάρχει στο Firebase

Authentication. Εάν η διαδικασία ολοκληρωθεί επιτυχώς, ο χρήστης μεταφέρεται είτε στην αρχική σελίδα της εφαρμογής (στην περίπτωση login) είτε στη διαδικασία αρχικοποίησης προφίλ (στην περίπτωση registration).



Σχήμα 5.4: Η οθόνη επαναφοράς κωδικού



Σχήμα 5.5: Η οθόνη δημιουργίας λογαριασμού

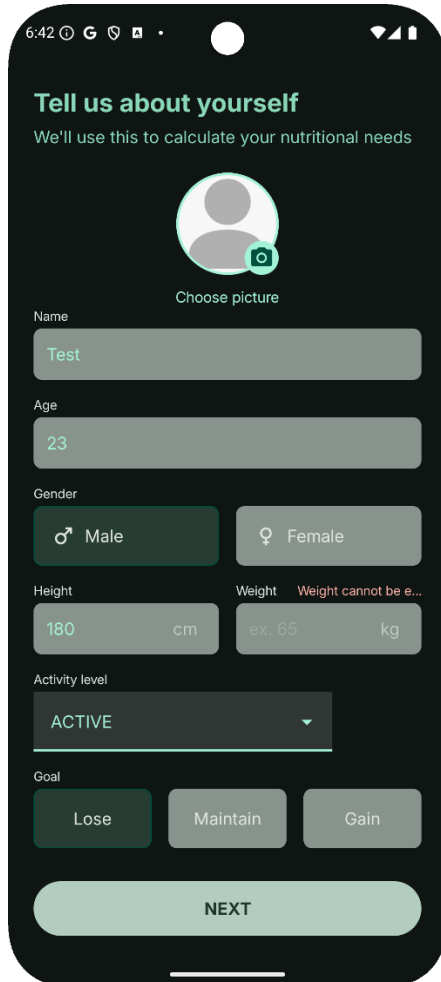
5.3 Αρχικοποίηση προφίλ χρήστη

Μετά την επιτυχή εγγραφή του χρήστη στο Firebase Authentication, η εφαρμογή τον μεταφέρει στη ροή αρχικοποίησης προφίλ, μέσω της οποίας συλλέγονται όλα τα απαραίτητα στοιχεία για τον υπολογισμό των διατροφικών αναγκών και τη μετέπειτα εκτέλεση του εξελικτικού αλγορίθμου. Η διαδικασία αυτή είναι υποχρεωτική, καθώς τα στοιχεία που καταχωρούνται χρησιμοποιούνται τόσο για τον υπολογισμό των θερμιδικών και μακροθρεπτικών στόχων, όσο και για τη δημιουργία του διατροφικού πλάνου. Αν ο χρήστης διακόψει την εφαρμογή πριν ολοκληρωθεί η αρχικοποίηση, κατά την επόμενη εκτέλεση οδηγείται ξανά σε αυτή τη ροή και όχι στην αρχική σελίδα.

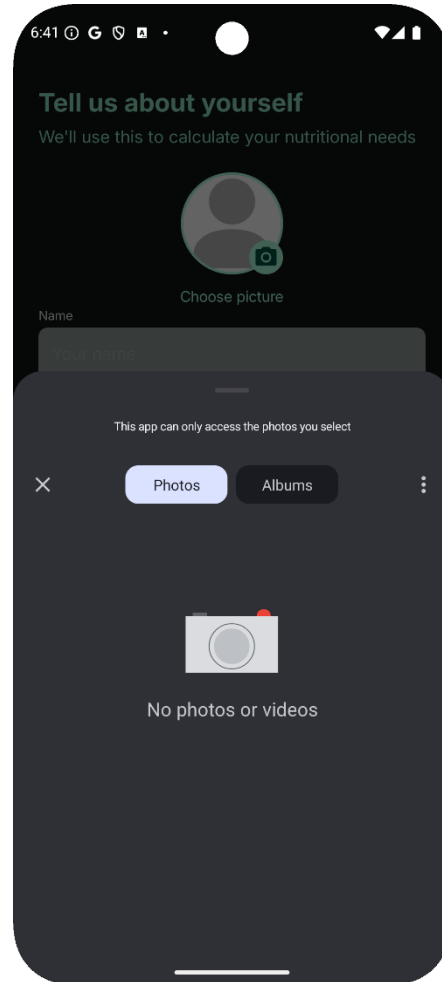
5.3.1 Εκχώρηση βασικών στοιχείων

Η πρώτη οθόνη της ροής αφορά την εισαγωγή βασικών προσωπικών δεδομένων (Σχήμα 5.6). Στο επάνω μέρος υπάρχει η επιλογή φωτογραφίας προφίλ, η οποία είναι προαιρετική. Με την επιλογή «Choose picture» ενεργοποιείται ο Photo Picker, ο οποίος φαίνεται στο Σχήμα 5.7, ένας μηχανισμός του Android που επιτρέπει στον χρήστη να επιλέξει συγκεκριμένες φωτογραφίες από τη συλλογή του, χωρίς να

παρέχεται πλήρης πρόσβαση στα αρχεία της συσκευής. Στη συνέχεια, ο χρήστης συμπληρώνει τα πεδία ονόματος χρήστη, ηλικίας, ύψους και βάρους, ενώ πρέπει να επιλέξει το φύλο του (Male/Female), το επίπεδο φυσικής δραστηριότητας (Activity level) και τον στόχο του (Lose, Maintain ή Gain). Οι επιλογές φυσικής δραστηριότητας αντιστοιχούν στις κατηγορίες SEDENTARY, LIGHT, MODERATE, ACTIVE και ATHLETE, με προεπιλογή την SEDENTARY, όπως περιγράφηκε στο Κεφάλαιο 2 κατά την ανάλυση των επιπέδων PAL.



Σχήμα 5.6: Η πρώτη οθόνη της ροής αρχικοποίησης του προφίλ του χρήστη

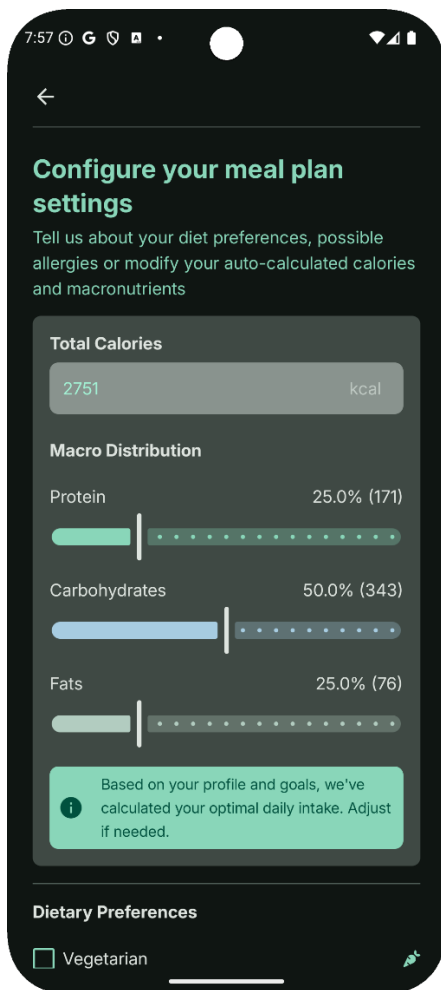


Σχήμα 5.7: Ο Photo Picker

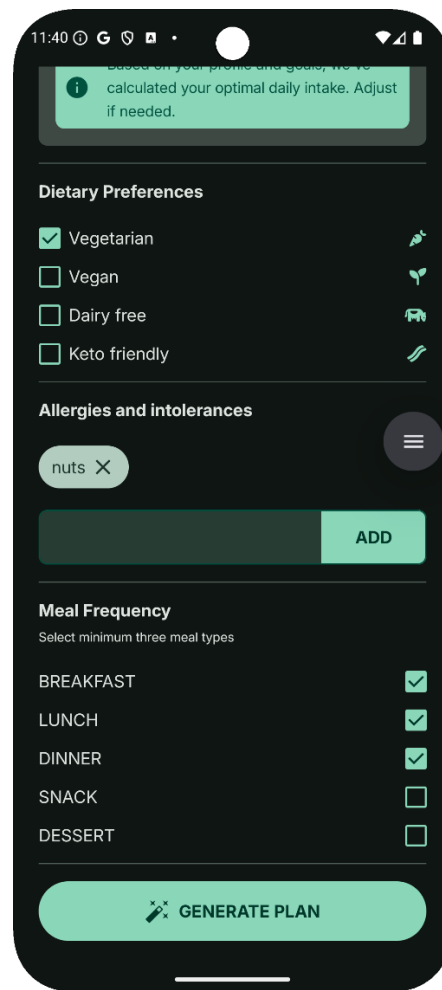
Το κουμπί «NEXT», το οποίο επιτρέπει τη μετάβαση στο επόμενο στάδιο, παραμένει ανενεργό έως ότου όλα τα πεδία, πλην της εικόνας προφίλ, συμπληρωθούν σωστά. Ειδικότερα, η ηλικία πρέπει να κυμαίνεται μεταξύ 8 και 100 ετών, το ύψος μεταξύ 100 και 250 εκατοστών και το βάρος μεταξύ 30 και 250 κιλών. Αν κάποιο πεδίο παραβιάζει τους περιορισμούς αυτούς ή μένει κενό, εμφανίζεται μήνυμα σφάλματος πάνω από το αντίστοιχο πεδίο, ενημερώνοντας τον χρήστη για το πρόβλημα. Με την επιλογή του «NEXT» εκτελείται η συνάρτηση που υπολογίζει τις ημερήσιες θερμίδες με βάση τον τύπο Mifflin-St Jeor, ο οποίος αναλύθηκε στο θεωρητικό υπόβαθρο. Ο υπολογισμός λαμβάνει υπόψη την ηλικία, το φύλο, το βάρος, το ύψος και το επίπεδο δραστηριότητας του χρήστη, ώστε να προκύψει το σύνολο θερμίδων που απαιτείται για την επίτευξη του επιλεγμένου στόχου. Τα δεδομένα αυτά αποθηκεύονται προσωρινά στην τοπική βάση δεδομένων, δημιουργώντας ένα αρχικό αντικείμενο χρήστη.

5.3.2 Ρύθμιση θερμίδων και διατροφικών προτιμήσεων

Μετά την ολοκλήρωση της πρώτης οθόνης, ο χρήστης οδηγείται στη δεύτερη, όπου πραγματοποιείται η διαμόρφωση των διατροφικών ρυθμίσεων (Σχήμα 5.8 και Σχήμα 5.9). Στην κορυφή της οθόνης υπάρχει toolbar με κουμπί επιστροφής, ενώ ακολουθεί μια καρτέλα που παρουσιάζει τις υπολογισμένες θερμίδες και τη μακροθρεπτική κατανομή, όπως προέκυψαν από τον προηγούμενο υπολογισμό. Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τόσο τον αριθμό θερμίδων, πληκτρολογώντας νέα τιμή στο πεδίο Total Calories, η οποία πρέπει να είναι μεγαλύτερη από 400 kcal, όσο και τα ποσοστά των μακροθρεπτικών συστατικών μέσω συρόμενων δεικτών (sliders). Το άθροισμα των ποσοστών δεν μπορεί να υπερβαίνει το 100%. Αν ξεπεραστεί αυτό το όριο, το κείμενο πάνω από τη μπάρα του αντίστοιχου συστατικού αποκτά κόκκινο χρώμα ως οπτική ένδειξη σφάλματος. Για παράδειγμα, μια έγκυρη κατανομή μπορεί να είναι 35% πρωτεΐνη, 50% υδατάνθρακες και 15% λιπαρά.



Σχήμα 5.8: Το πρώτο μέρος της δεύτερης οθόνης της ροής αρχικοποίησης προφίλ χρήστη



Σχήμα 5.9: Το δεύτερο μέρος της δεύτερης οθόνης της ροής αρχικοποίησης προφίλ χρήστη

Στη συνέχεια, ο χρήστης καλείται να επιλέξει τις διατροφικές του προτιμήσεις μέσα από τις διαθέσιμες επιλογές Vegetarian, Vegan, Dairy Free και Keto Friendly. Οι επιλογές αυτές είναι αμοιβαία αποκλειόμενες, αλλά η εφαρμογή επιτρέπει και την απουσία επιλογής, οπότε το διατροφικό πλάνο θα περιλαμβάνει όλα τα είδη τροφών.

Ακολουθεί η ενότητα «Allergies and Intolerances», στην οποία ο χρήστης μπορεί να δηλώσει τροφές που του προκαλούν αλλεργία ή δυσανεξία. Πληκτρολογώντας την τροφή και πατώντας «ADD», δημιουργείται ένα στοιχείο (chip) που εμφανίζεται πάνω από το πεδίο, με δυνατότητα αφαίρεσης. Αν προστεθούν πολλές τροφές, εμφανίζονται διαδοχικά και αναδιπλώνονται σε επόμενη γραμμή αν δεν χωρούν στο πλάτος της οθόνης. Αν δεν υπάρχει αλλεργία ή δυσανεξία, η ενότητα μπορεί να παραλειφθεί. Η επόμενη ενότητα αφορά τη συχνότητα γευμάτων. Ο χρήστης επιλέγει ποια γεύματα επιθυμεί να περιλαμβάνονται στο ημερήσιο πλάνο ανάμεσα σε Breakfast, Lunch, Dinner, Snack και Dessert.

5.3.3 Αποθήκευση δεδομένων και δημιουργία χρήστη

Όταν όλες οι τιμές είναι έγκυρες και δεν υπάρχουν σφάλματα, ενεργοποιείται το κουμπί «GENERATE PLAN», το οποίο ολοκληρώνει τη διαδικασία αρχικοποίησης. Με την επιλογή του, τα δεδομένα του χρήστη ενημερώνονται στην τοπική βάση δεδομένων και παράλληλα αποστέλλονται στο Firebase Firestore, όπου δημιουργείται ένα νέο έγγραφο χρήστη. Σε αυτό το σημείο είναι σημαντικό να γίνει διάκριση μεταξύ Firebase Authentication και Firebase Firestore. Το πρώτο διαχειρίζεται την ταυτοποίηση και εξουσιοδότηση των χρηστών, αποθηκεύοντας στοιχεία όπως το email και το hashed password, χωρίς να περιλαμβάνει επιπλέον πληροφορίες. Το δεύτερο αποτελεί βάση δεδομένων cloud, στην οποία αποθηκεύονται τα προφίλ χρηστών, οι διατροφικές τους ρυθμίσεις, οι θερμιδικοί στόχοι και οι προτιμήσεις, επιτρέποντας τη μελλοντική ανάκτηση και συγχρονισμό με τη συσκευή.

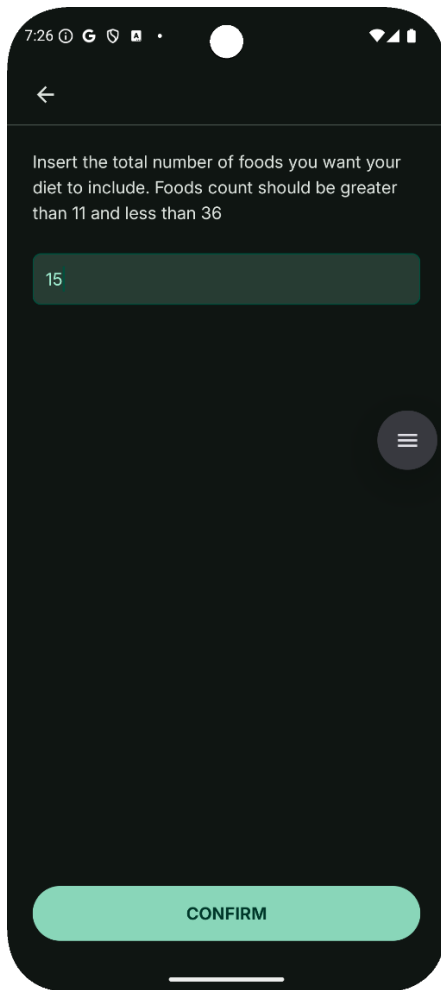
Για την εκτέλεση της συγκεκριμένης λειτουργίας απαιτείται ενεργή σύνδεση στο διαδίκτυο. Αν δεν υπάρχει, εμφανίζεται ενημερωτικό toast που ειδοποιεί για την απουσία δικτύου, ενώ σε περίπτωση αποτυχίας επικοινωνίας με το Firestore εμφανίζεται μήνυμα σφάλματος πάλι με τη μορφή toast. Αν η δημιουργία χρήστη στο Firestore αποτύχει, η αρχικοποίηση θεωρείται μη ολοκληρωμένη και ο χρήστης θα χρειαστεί να επαναλάβει τη διαδικασία κατά την επόμενη εκτέλεση της εφαρμογής, σε περίπτωση που κλείσει την εφαρμογή. Αν από την άλλη η λειτουργία ολοκληρωθεί επιτυχώς, ο χρήστης ανακατευθύνεται στην οθόνη παραγωγής διατροφικού πλάνου μέσω του εξελικτικού αλγορίθμου, η οποία αναλύεται στην επόμενη υποενότητα.

5.4 Παραγωγή διατροφικού πλάνου

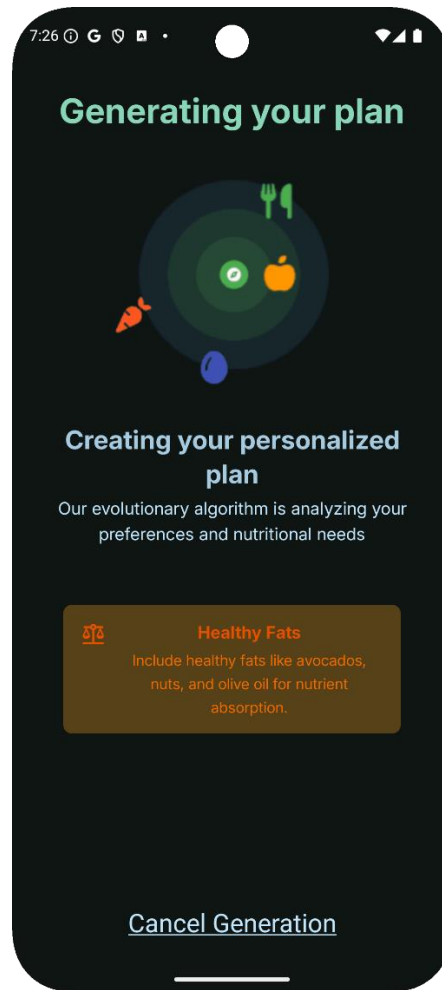
Η ροή παραγωγής διατροφικού πλάνου ξεκινά από την οθόνη επιλογής πλήθους τροφών (Σχήμα 5.10). Στην κορυφή της σελίδας υπάρχει πεδίο εισαγωγής όπου ο χρήστης πληκτρολογεί τον συνολικό αριθμό τροφών που επιθυμεί να περιλαμβάνει το πλάνο, ενώ πάνω από αυτό υπάρχει και μήνυμα προτροπής το οποίο υποδεικνύει την πληροφορία που πρέπει να εισαχθεί στο πεδίο αυτό, καθώς επίσης πληροφορεί τον χρήστη ότι δεκτές γίνονται μόνο τιμές μεγαλύτερες από 11 και μικρότερες από 36. Όταν το πεδίο είναι κενό ή εκτός των επιτρεπόμενων ορίων, εμφανίζεται αντίστοιχο σφάλμα πάνω από το πεδίο. Στο κάτω μέρος της οθόνης, υπάρχει το κουμπί «CONFIRM», το οποίο είναι αρχικά ανενεργό. Με την καταχώριση έγκυρης τιμής, το κουμπί ενεργοποιείται και η επιλογή του οδηγεί στην επόμενη οθόνη.

Η δεύτερη οθόνη λειτουργεί ως οθόνη αναμονής (loading screen), όπως παρουσιάζεται και στο Σχήμα 5.11. Για όσο αυτή η οθόνη υπάρχει στο προσκήνιο, εκτελούνται όλες οι απαραίτητες ενέργειες για την παραγωγή του διατροφικού πλάνου, δηλαδή, με την ακόλουθη σειρά, ανάκτηση των δεδομένων από το USDA API, επεξεργασία των δεδομένων πριν εκχωρηθούν ως είσοδος στον εξελικτικό αλγόριθμο, ανάκτηση δεδομένων χρήστη, εκτέλεση του προαναφερθέντος αλγορίθμου και αποθήκευση του παραγόμενου διατροφικού πλάνου στη Firestore. Η διαδικασία αυτή θα εξηγηθεί εκτενέστερα στις υποενότητες της παρούσας ενότητας. Στο πάνω τμήμα της εν λόγω οθόνης, προβάλλεται κινούμενο

γραφικό φόρτωσης που αντικαθιστά τους παραδοσιακούς κυκλικούς δείκτες φόρτωσης, ενώ κάτω από αυτό εμφανίζονται ένα κείμενο που αναφέρει τον λόγο αναμονής, καθώς και διατροφικές συμβουλές οι οποίες εναλλάσσονται περιοδικά ανά περίπου δέκα δευτερόλεπτα. Οι συμβουλές προέρχονται από ένα στατικό αρχείο JSON που περιλαμβάνεται στον φάκελο «assets» της εφαρμογής, ο οποίος παράγεται αυτόματα με τη δημιουργία ενός έργου στο Android Studio, αποτελεί υποφάκελο του φακέλου πόρων (res) και προορίζεται για αυτόν τον σκοπό, ώστε η παρουσίασή τους να είναι άμεση και ανεξάρτητη από σύνδεση δικτύου. Στο κάτω μέρος της οθόνης υπάρχει η επιλογή ακύρωσης της διαδικασίας. Η ενεργοποίησή της εμφανίζει μήνυμα επιβεβαίωσης και αν ο χρήστης επιβεβαιώσει, η παραγωγή τερματίζεται και πραγματοποιείται επιστροφή στην προηγούμενη οθόνη. Στην πρώτη εκτέλεση γίνεται μετάβαση στην αρχική σελίδα της εφαρμογής, διότι δεν επιθυμείται επιστροφή στη ροή αρχικοποίησης προφίλ.



Σχήμα 5.10: Η οθόνη επιλογής πλήθους τροφών



Σχήμα 5.11: Η οθόνη αναμονής παραγωγής του διατροφικού πλάνου

Καθ' όλη τη διάρκεια της εκτέλεσης ελέγχεται η διαθεσιμότητα δικτύου και η ομαλή ολοκλήρωση των επιμέρους βημάτων. Σε περίπτωση απώλειας σύνδεσης στο διαδίκτυο προβάλλεται ειδικό μήνυμα που ενημερώνει για την ανάγκη ενεργής πρόσβασης στο διαδίκτυο, ενώ για οποιοδήποτε άλλο σφάλμα εμφανίζεται ενημερωτικό παράθυρο με ένα γενικευμένο μήνυμα σφάλματος παραγωγής του διατροφικού πλάνου. Η σύνδεση στο διαδίκτυο χρειάζεται μόνο για την ανάκτηση των δεδομένων τροφών από το API. Η αποθήκευση του διατροφικού πλάνου χειρίζεται με μηχανισμούς συγχρονισμού

που θα αναλυθούν και αυτοί στη συνέχεια. Όταν η παραγωγή ολοκληρωθεί επιτυχώς, παρουσιάζεται ένα modal επιβεβαίωσης που ενημερώνει ότι το διατροφικό πλάνο δημιουργήθηκε και αποθηκεύτηκε. Το πάτημα του αντίστοιχου κουμπιού που υπάρχει σε αυτό το modal κλείνει το μήνυμα και οδηγεί τον χρήστη στην επόμενη σελίδα.

5.4.1 Ανάκτηση δεδομένων από το API και διαχείριση του όγκου τους

Με την εμφάνιση της οθόνης αναμονής ξεκινάει στο παρασκήνιο να εκτελείται και όλη η λογική για την παραγωγή του διατροφικού πλάνου, με τη σειρά που αναφέρθηκε πρωτίστως. Για την εκτέλεση του εξελικτικού αλγόριθμου είναι απαραίτητος ένας εύλογος όγκος δεδομένων, κάτι το οποίο έχει τη δυνατότητα να προσφέρει το USDA FoodData Central. Ένας μεγάλος όγκος δεδομένων απαιτεί ωστόσο και σωστή διαχείριση. Για τον λόγο αυτόν, εκτός από την ανάκτηση των διατροφικών δεδομένων από την απομακρυσμένη βάση δεδομένων, γίνεται και αποθήκευσή τους τοπικά, μέσω της βιβλιοθήκης Paging 3 του Android [84]. Στόχος είναι η σταδιακή και ανθεκτική σε σφάλματα, φόρτωση ενός μεγάλου καταλόγου τροφίμων, η αποθήκευσή τους στη Room, και η παροχή ροών δεδομένων προς το UI και τον εξελικτικό αλγόριθμο, αξιοποιώντας πλήρως τις δυνατότητες μνήμης και δικτύου.

Η αλληλεπίδραση με το USDA πραγματοποιείται μέσω μιας συνάρτησης που υλοποιήθηκε, η οποία ανακτά δεδομένα από το API για μια σελίδα. Για κάθε σελίδα αναζήτησης αποστέλλεται αίτημα με μέγεθος σελίδας ίσο με 200, ενώ το φίλτρο dataType περιορίζει την αναζήτηση στα σύνολα «Survey (FNDDS)» και «Foundation», όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο. Τα διαπιστευτήρια παρέχονται ως query parameter (api key). Η κλήση επιστρέφει μια λίστα από αντικείμενα, καθένα από τα οποία αντιπροσωπεύει ένα τρόφιμο, και περιέχει αναγνωριστικό (FDC ID), περιγραφή και λίστα θρεπτικών με τους αντίστοιχους κωδικούς/τιμές, όπως δημοσιεύονται από το USDA.

Η βιβλιοθήκη Paging 3 στο Android παρέχει έναν τυποποιημένο μηχανισμό σταδιακής φόρτωσης, caching και παρουσίασης μεγάλων συλλογών δεδομένων. Ο κεντρικός μηχανισμός της βασίζεται στον Pager, ο οποίος εκθέτει ροές PagingData προς το UI, και στον Remote Mediator, ο οποίος γεφυρώνει το τοπικό cache με την απομακρυσμένη πηγή αποφασίζοντας πότε θα γίνει δικτυακή φόρτωση και πότε η πλοήγηση μπορεί να εξυπηρετηθεί από την τοπική βάση. Η προσέγγιση αυτή μειώνει δραστικά τη μνήμη που απαιτείται, καθώς το UI παραλαμβάνει τμηματικά μόνο τις εγγραφές που χρειάζεται, ενώ το υπόβαθρο αναλαμβάνει να συμπληρώνει τη βάση με νέες σελίδες. Παράλληλα, ο Remote Mediator χειρίζεται σενάρια χωρίς σύνδεση, επαναπροσπάθειες και τερματισμό της σελιδοποίησης όταν δεν υπάρχουν άλλα δεδομένα, καθιστώντας τη λύση κατάλληλη για πολύ μεγάλους καταλόγους αντικειμένων [84].

Στην υλοποίηση της εφαρμογής, ο FoodRemoteMediator είναι υπεύθυνος να φέρει σταδιακά στην τοπική βάση το σύνολο τροφίμων. Στο αρχείο αυτό, η μεταβλητή targetCount ορίζει τον στόχο των 5.000 εγγραφών/τροφίμων. Στην αρχή κάθε φόρτωσης, γίνεται έλεγχος του τρέχοντος πλήθους εγγραφών στη Room και αν έχει ήδη επιτευχθεί το targetCount, ο mediator δηλώνει μέσω της Boolean μεταβλητής endOfPaginationReached ότι δεν απαιτείται άλλη δικτυακή ανάκτηση. Διαφορετικά, υπολογίζει την επόμενη σελίδα προς λήψη διαιρώντας το τρέχον πλήθος στοιχείων με το 200, που είναι το πλήθος των στοιχείων της κάθε σελίδας, και αυξάνοντας το αποτέλεσμα από αυτή τη διαίρεση κατά μια μονάδα και ζητά τα αντίστοιχα αποτελέσματα καλώντας ξανά το API. Για κάθε επιστρεφόμενο τρόφιμο, φιλτράρονται οι θρεπτικές τιμές, κρατώντας μόνο όσες αντιστοιχούν στο σύνολο κωδικών NUTRIENT_CODES και έχουν ορισμένη τιμή. Το NUTRIENT_CODES είναι μια λίστα που περιλαμβάνει τους κωδικούς των μακροθρεπτικών και μικροθρεπτικών στοιχείων από το USDA API, που επιθυμούνται να χρησιμοποιηθούν στην εφαρμογή. Η τελική αναπαράσταση είναι ένα FoodEntity,

μια δομή δηλαδή που αποθηκεύεται στη Room και περιλαμβάνει το FDC ID, περιγραφή, mealType και ένα String, σε μορφή JSON, των θρεπτικών στοιχείων. Το JSON παράγεται μέσω του NutrientMapConverter, μιας κλάσης που περιλαμβάνει συναρτήσεις μετατροπής ενός αντικείμενου σε String με JSON μορφή και το αντίστροφο. Η παρτίδα των νέων εγγραφών εισάγεται εντός συναλλαγής Room, η οποία εκτελεί όλες τις εντολές μέσα της ως μία ατομική συναλλαγή, ώστε είτε να ολοκληρωθούν όλες με επιτυχία είτε καμία. Στο τέλος, ο mediator υπολογίζει αν έληξε πρακτικά η σελιδοποίηση, όταν δηλαδή δεν παρεμβλήθηκαν νέες σειρές και το συνολικό πλήθος έχει φτάσει τον στόχο, οπότε σηματοδοτεί ότι δεν απαιτείται άλλη δικτυακή κλήση. Όλη η παραπάνω ροή βρίσκεται συγκεντρωμένη στην συνάρτηση «load» του FoodRemoteMediator, όπου επίσης περιλαμβάνεται ενιαία διαχείριση σφαλμάτων και μετατροπή τους σε MediatorResult για τον χειρισμό τους στην ανώτερη στοίβα (repository, viewModel), που θα χρησιμοποιήσει αυτή τη συνάρτηση.

Ένα σημείο της ίδιας ροής που δεν εξηγήθηκε σκοπίμως προηγουμένως κατά την αναφορά των πεδίων του FoodEntity είναι το πεδίο «mealType». Καθώς τα δεδομένα του USDA δεν περιλαμβάνουν κάποιο πεδίο που να αντιστοιχεί ένα τρόφιμο σε μια ή περισσότερες κατηγορίες πρωινού, μεσημεριανού, δείπνου, σνακ ή επιδόρπιου, αναπτύχθηκε μια λογική ταξινόμησης των τροφών αυτών στις προαναφερθείσες κατηγορίες. Πριν την εκχώρηση του κάθε FoodEntity στην τοπική βάση δεδομένων, μέσω του FoodRemoteMediator, εκτελείται η εν λόγω λογική ταξινόμησης και για κάθε τρόφιμο αποδίδεται μια κατηγορία. Η ανάλυση της ταξινόμησης έπεται, καθώς κρίνεται σημαντικό να αφιερωθεί μια αποκλειστική υποενότητα σε αυτή.

Οι παραπάνω λειτουργικότητες συνδυάζονται σε ένα repository, το οποίο είναι και αυτό που χρησιμοποιείται από τα viewModels για την αξιοποίηση όλων αυτών των δυνατοτήτων. Το repository περιλαμβάνει τρεις συναρτήσεις, σχετικές με αυτή τη ροή. Η πρώτη συνάρτηση, που ονομάζεται «getFoodsFromPager», αφορά την ανάκτηση των δεδομένων ανά σελίδα, που χρησιμοποιείται σε οθόνη που θα εξηγηθεί μεταγενέστερα και περιλαμβάνει μια λίστα με όλες τις διαθέσιμες τροφές. Στην ουσία η συνάρτηση αυτή αξιοποιεί ένα αντικείμενο Pager, το οποίο ενεργοποιεί τη load μόνο όταν το UI ζητήσει δεδομένα, για παράδειγμα όταν ο χρήστης κάνει scroll στη λίστα τροφίμων και φροντίζει να φέρνει μόνο τις απαραίτητες σελίδες για ομαλή προβολή.

Η δεύτερη συνάρτηση χρησιμοποιείται πριν από την εκτέλεση του MOEA και εκκινεί τη διαδικασία εκτέλεσης της συνάρτησης «load» χειροκίνητα και προκαταβολικά. Επειδή στην οθόνη εκτέλεσης του EA δεν υπάρχει κάποιο UI trigger, όπως το scroll, η συνάρτηση δημιουργεί τεχνητά ένα PagingState και καλεί επαναληπτικά τη load, ώστε να «αναγκάσει» τη φόρτωση σελίδων από το USDA API μέχρι να συμπληρωθούν τουλάχιστον 5000 εγγραφές στη βάση ή να μην υπάρχουν άλλες σελίδες προς φόρτωση ή να παρουσιαστεί σφάλμα.

Η τρίτη συνάρτηση του repository, χρησιμοποιείται μετά την ολοκλήρωση της διαδικασίας φόρτωσης των δεδομένων και αποτελεί τη συνάρτηση που χρησιμοποιείται στην εκτέλεση του εξελικτικού αλγορίθμου. Σκοπός της είναι να ανακτήσει από τη βάση όλα τα τρόφιμα που έχουν αποθηκευτεί τοπικά και να τα μετατρέψει σε αντικείμενα του domain model, κατάλληλα για χρήση από τον MOEA. Κατά τη φάση αυτή εφαρμόζονται επιπλέον φίλτρα, ώστε το σύνολο των τροφίμων που θα δοθεί ως είσοδος στον αλγόριθμο να είναι απολύτως συμβατό με τις επιλογές και τους περιορισμούς του χρήστη. Συγκεκριμένα, η συνάρτηση διατηρεί μόνο εκείνα τα τρόφιμα των οποίων ο τύπος γεύματος (mealType) αντιστοιχεί σε αυτούς που έχει επιλέξει ο χρήστης κατά το στάδιο της αρχικοποίησης προφίλ, ενώ παράλληλα αποκλείει τρόφιμα που περιέχουν συστατικά στα οποία ο χρήστης έχει δηλώσει αλλεργία ή δυσανεξία.

5.4.2 Λογική ταξινόμησης τροφίμων

Όπως αναφέρθηκε στην προηγούμενη υποενότητα, κατά την αποθήκευση κάθε τροφίμου στη βάση δεδομένων μέσω του FoodRemoteMediator, αποδίδεται σε αυτό ένας τύπος γεύματος (mealType), ώστε να μπορεί στη συνέχεια να χρησιμοποιηθεί κατάλληλα τόσο από τον χρήστη όσο και από τον εξελικτικό αλγόριθμο. Η λογική της ταξινόμησης αυτής υλοποιείται στην κλάση MealClassifierImpl, η οποία έχει ως στόχο να αποδώσει σε κάθε τρόφιμο μια από τις κατηγορίες «breakfast», «lunch», «dinner», «snack» ή «dessert». Υπάρχει ωστόσο και η κατηγορία «unknown», η οποία χρησιμοποιείται για τροφές που δεν ήταν δυνατό να ταξινομηθούν. Οι τροφές αυτές λαμβάνονται υπόψη στην παραγωγή του διατροφικού πλάνου και αυτό σημαίνει ότι τα τρόφιμα που ανήκουν σε αυτή την κατηγορία, μπορεί ο χρήστης να τα εντάξει συνδυαστικά σε οποιοδήποτε από τα υπόλοιπα γεύματα της ημέρας.

Για την αρχικοποίηση του ταξινομητή χρησιμοποιήθηκε ένα αρχείο με όνομα «seed_foods.csv», το οποίο περιλαμβάνει μια σειρά από τρόφιμα που προήλθαν από το USDA API και ταξινομήθηκαν χειροκίνητα. Συγκεκριμένα, κατά τη φάση ανάπτυξης παρατηρήθηκαν κάποια από τα αποτελέσματα του API και επιλέχθηκαν αντιπροσωπευτικά παραδείγματα τροφίμων, στα οποία αποδόθηκε η επιθυμητή κατηγορία γεύματος, προκειμένου να αποτελέσουν τη βάση της λογικής. Κατά την εκκίνηση της εφαρμογής ελέγχεται εάν υπάρχουν ήδη δεδομένα στη βάση. Εάν όχι, τότε εκτελείται μια συνάρτηση, ώστε να ταξινομήσει το αρχικό σύνολο δεδομένων, σύμφωνα με το «seed_foods.csv». Έτσι εξασφαλίζεται ότι κάποια βασικά δεδομένα θα ταξινομηθούν σίγουρα στις επιθυμητές κατηγορίες, χωρίς η λογική ταξινόμησή τους να βασίζεται στη λογική που θα εξηγηθεί παρακάτω, κατά την οποία μπορεί να προκύψουν σφάλματα.

Ο μηχανισμός του MealClassifierImpl βασίζεται κυρίως στη εύρεση λέξεων-κλειδιών που προέρχονται από την περιγραφή του τροφίμου, με προ-ορισμένες λίστες λέξεων για κάθε τύπο γεύματος. Για παράδειγμα, η λίστα breakfastWords περιλαμβάνει λέξεις όπως «egg», «oatmeal», «toast», «pancake», «cereal» ή «coffee», οι οποίες συνήθως συναντώνται σε τρόφιμα που καταναλώνονται κατά το πρωινό. Αντίστοιχα, η λίστα lunchWords περιλαμβάνει λέξεις όπως «sandwich», «salad», «burger», «soup», ενώ η dinnerWords περιλαμβάνει όρους όπως «steak», «pasta», «rice», «chicken» ή «fish». Για τα snacks, υπάρχουν λέξεις όπως «bar», «chips», «nuts», «yogurt», και για τα desserts, λέξεις όπως «cake», «cookie», «ice cream» ή «pie». Κατά την εκτέλεση, η περιγραφή του τροφίμου μετατρέπεται σε πεζά γράμματα και ελέγχεται εάν περιέχει κάποια από τις λέξεις των παραπάνω λιστών. Αν βρεθεί αντιστοιχία, το τρόφιμο ταξινομείται στην αντίστοιχη κατηγορία.

Η λειτουργία αυτή επεκτείνεται ώστε να λαμβάνει υπόψη και τις διατροφικές προτιμήσεις που έχει δηλώσει ο χρήστης κατά τη διαδικασία της αρχικοποίησης προφίλ. Για παράδειγμα, αν ο χρήστης έχει επιλέξει διατροφή τύπου vegan, τότε κατά την ταξινόμηση αποφεύγονται τρόφιμα που περιέχουν λέξεις όπως «egg», «cheese», «milk», «beef» ή «chicken». Αντίστοιχα, για διατροφή vegetarian εξαιρούνται τρόφιμα που περιλαμβάνουν όρους σχετικούς με το κρέας, ενώ για keto δίνονται προτεραιότητες σε τρόφιμα υψηλής περιεκτικότητας σε πρωτεΐνες και λιπαρά, αποφεύγοντας εκείνα με αυξημένους υδατάνθρακες, όπως «bread», «pasta» ή «rice». Οι κανόνες αυτοί υλοποιούνται με βάση προκαθορισμένα σύνολα λέξεων-κλειδιών, τα οποία έχουν ενσωματωθεί στην κλάση του ταξινομητή.

5.4.3 Υλοποίηση εξελικτικού αλγόριθμου

Αφού ολοκληρωθεί η διαδικασία ανάκτησης και ταξινόμησης των τροφίμων, ο EA αναλαμβάνει να παράξει το τελικό διατροφικό πλάνο, με βάση τα εξατομικευμένα δεδομένα του χρήστη. Ο αλγόριθμος έχει υλοποιηθεί στην κλάση MealPlanGeneratorImpl, χρησιμοποιώντας ως μέθοδο βελτιστοποίησης τον αλγόριθμο NSGA-II. Ο αλγόριθμος αυτός επιλέχθηκε, διότι επιτρέπει τη βελτιστοποίηση

πολλαπλών, ανεξάρτητων και ενίοτε αντικρουόμενων στόχων, όπως η επίτευξη της ημερήσιας θερμιδικής πρόσληψης, η ισορροπία μακροθρεπτικών συστατικών και η κάλυψη των μικροθρεπτικών αναγκών.

Η διαδικασία εκτέλεσης του EA ξεκινά με την ανάκτηση των τροφίμων από τη βάση μέσω της συνάρτησης «getAllFoodsFromDb», όπως εξηγήθηκε στην προηγούμενη υποενότητα. Στη λίστα αυτή προστίθενται και τα τρόφιμα που έχει δημιουργήσει ο χρήστης και αφαιρούνται τα ανεπιθύμητα δεδομένα, στα οποία οι πρωτεΐνες, οι υδατάνθρακες και τα λιπαρά είναι ίσα με 0 γραμμάρια. Το σύνολο αυτό αποτελεί τη δεξαμενή από την οποία ο αλγόριθμος θα αντλήσει υποψήφια τρόφιμα για τη σύνθεση του διατροφικού πλάνου. Παράλληλα, αξιοποιούνται τα δεδομένα που έχουν ήδη υπολογιστεί κατά τη διαδικασία δημιουργίας προφίλ, όπως οι ημερήσιες θερμίδες-στόχος και οι στόχοι μακροθρεπτικών και μικροθρεπτικών συστατικών. Υπάρχει επίσης και η μεταβλητή preferredFoodsCount, η οποία έχει αποθηκευμένο τον αριθμό τροφών που επέλεξε ο χρήστης στην πρώτη οθόνη της ροής παραγωγής διατροφικού πλάνου.

Με βάση τις επιλογές του χρήστη, ο αλγόριθμος δημιουργεί δεξαμενές τύπων γεύματος (slot pools) μέσω της συνάρτησης «buildSlotPools». Για κάθε τύπο γεύματος που έχει επιλέξει ο χρήστης, κατασκευάζεται μια βασική λίστα τροφίμων, η οποία περιλαμβάνει μόνο τα τρόφιμα που έχουν ταξινομηθεί ως κατάλληλα για τον συγκεκριμένο τύπο, όπως τρόφιμα πρωινού στη λίστα του breakfast. Παράλληλα, δημιουργείται και μια επεκταμένη εκδοχή της λίστας αυτής, που περιλαμβάνει όλα τα τρόφιμα της βασικής λίστας μαζί με όσα έχουν χαρακτηριστεί ως «unknown». Με αυτόν τον τρόπο, ο αλγόριθμος διατηρεί μια ελάχιστη εγγύηση ποικιλίας, εξασφαλίζοντας ότι ακόμα και αν κάποιες κατηγορίες έχουν περιορισμένο αριθμό τροφίμων, θα υπάρχει πάντα η δυνατότητα επιλογής υποκατάστατων από την επεκταμένη λίστα.

Αφού δημιουργηθούν οι δεξαμενές, η συνάρτηση «buildCodec» ορίζει τον τρόπο με τον οποίο η πληροφορία ενός διατροφικού πλάνου αναπαρίσταται γενετικά, δηλαδή πώς μετατρέπεται ένα συγκεκριμένο πλάνο σε γονότυπο (genotype) και αντιστρόφως. Η κωδικοποίηση (codec) που χρησιμοποιείται έχει σχεδιαστεί με τέτοιο τρόπο ώστε να αντικατοπτρίζει ποια τρόφιμα επιλέγονται και σε ποιες ποσότητες. Ο γονότυπος αποτελείται από δύο μέρη, ένα σταθερό και ένα ευέλικτο. Το σταθερό μέρος εγγυάται ότι για κάθε τύπο γεύματος που έχει επιλέξει ο χρήστης θα υπάρχει τουλάχιστον ένα τρόφιμο. Για κάθε τέτοιο τύπο δημιουργούνται δύο χρωμοσώματα. Το πρώτο καθορίζει ποιο τρόφιμο επιλέγεται, δηλαδή ποιος δείκτης του πίνακα τροφίμων αντιστοιχεί σε αυτό, ενώ το δεύτερο καθορίζει την ποσότητά του, η οποία εκφράζεται μέσω ενός ακεραίου αλληλόμορφου (allele) με τιμή από 1 έως 40. Το ευέλικτο μέρος του γονότυπου επιτρέπει στον αλγόριθμο να συμπληρώνει τα υπόλοιπα τρόφιμα που απαιτούνται για να φτάσει το preferredFoodsCount, επιλέγοντας ελεύθερα από τη συνολική ένωση όλων των επεκταμένων δεξαμενών (extended slot pools). Για το σύνολο των «ελεύθερων» τροφίμων δημιουργούνται δύο χρωμοσώματα. Το πρώτο κωδικοποιεί τους δείκτες όλων των ευέλικτων τροφίμων, είναι τύπου IntegerChromosome και μήκους flexibleCount, όπου flexibleCount

$$flexibleCount = preferredFoodsCount - selectedTypes.size \quad (5.1)$$

, ενώ το δεύτερο κωδικοποιεί τις αντίστοιχες ποσότητές τους και είναι επίσης IntegerChromosome μήκους flexibleCount. Κατά αυτόν τον τρόπο, το ευέλικτο τμήμα αναπαρίσταται με δύο χρωμοσώματα πολλαπλών γονιδίων, αντί για πολλαπλά ζεύγη μονογονιδιακών χρωμοσωμάτων. Συνολικά, η αναπαράσταση του γονότυπου αποτελείται από μια ακολουθία χρωμοσωμάτων της μορφής [indexBreakfast, portionBreakfast, indexLunch, portionLunch, indexDinner, portionDinner, flexibleIndices, flexiblePortions], όπου flexibleIndices και flexiblePortions, τα χρωμοσώματα μήκους flexibleCount, που περιεγράφηκαν προηγουμένως. Η αναπαράσταση αυτή επιλέχθηκε γιατί προσφέρει

διακριτό, εύχρηστο και σταθερής δομής γονότυπο, αποτρέποντας πιθανές παραμορφώσεις της δομής κατά τη διάρκεια των γενετικών τελεστών.

Η διαδικασία της αποκωδικοποίησης ακολουθεί την διάταξη με την οποία χτίστηκε ο γονότυπος και προχωρά σειριακά πάνω στα χρωμοσώματα με έναν μετρητή θέσης (`chromosomeIdx`). Για κάθε επιλεγμένο τύπο γεύματος διαβάζεται πρώτα το χρωμόσωμα του δείκτη τροφίμου και στη συνέχεια το αντίστοιχο χρωμόσωμα της ποσότητας. Ένας δείκτης (`foodIdx`) επιλέγεται από τη βασική δεξαμενή του συγκεκριμένου τύπου, με εναλλακτική πηγή την επεκταμένη λίστα όταν η βασική είναι κενή, ενώ η ποσότητα προκύπτει από το ακέραιο αλληλόμορφο αφού διαιρεθεί με τη σταθερά `PORTION_SCALE`. Στην παρούσα υλοποίηση το `PORTION_SCALE` είναι 20, άρα οι ακέραιες τιμές 1-40 μετατρέπονται σε πραγματικές 0,05-2,00. Επειδή τα τρόφιμα στο USDA API αφορούν ποσότητες των 100 γραμμαρίων, αυτό σημαίνει ότι η παραγόμενη διατροφή μπορεί να περιλαμβάνει ποσότητες τροφίμων από 5 έως 200 γραμμάρια. Η επιλογή της αναπαράστασης των ποσοτήτων με ακεραίους και η μετέπειτα διαίρεση τους κατά την αποκωδικοποίηση έγινε, ώστε όλα τα χρωμοσώματα να είναι `IntegerChromosome` και να αποφεύγονται μεικτοί τύποι, διευκολύνοντας τόσο τον στοχευμένο δρομολογισμό των γενετικών τελεστών με `PartialAlterer` όσο και την αποδοτική εκτέλεση.

Μετά το σταθερό τμήμα, ο decoder χειρίζεται το ευέλικτο υπόλοιπο ελέγχοντας το `flexibleCount` Εφόσον είναι θετικό, ανακτά τα δύο χρωμοσώματα του ευέλικτου τμήματος, δηλαδή το χρωμόσωμα των δεικτών τροφίμων και το χρωμόσωμα των ποσοτήτων. Στη συνέχεια, διατρέχει παράλληλα τα γονίδια και των δύο χρωμοσωμάτων, εξάγοντας για κάθε θέση $i \in [0, \text{flexibleCount})$ τον δείκτη του τροφίμου από το πρώτο χρωμόσωμα και την αντίστοιχη ποσότητα από το δεύτερο. Η επιλογή των τροφίμων γίνεται από τη μεγάλη ενωτική λίστα `allExtendedFoods`. Για κάθε τέτοια επιλογή, ο τύπος γεύματος που αποθηκεύεται στην επιλογή προέρχεται από το ίδιο το τρόφιμο, από τη μεταβλητή `mealType`. Το αποτέλεσμα της αποκωδικοποίησης είναι μια λίστα από αντικείμενα `FoodSelection(food, portion, type)`, η οποία τροφοδοτεί άμεσα τη συνάρτηση καταλληλότητας. Η συνάρτηση `buildCodec`, παρουσιάζεται στο Παράρτημα Γ.

Η συνάρτηση καταλληλότητας (`fitness function`) είναι η καρδιά του αλγορίθμου και αξιολογεί κάθε υποψήφιο πλάνο με βάση τρεις διαφορετικούς στόχους. Οι στόχοι αυτοί είναι:

- η απόκλιση των συνολικών θερμίδων από τον ημερήσιο στόχο του χρήστη,
- η ισορροπία των μακροθρεπτικών συστατικών σε σχέση με τους στόχους του χρήστη, και
- η επάρκεια των μικροθρεπτικών συστατικών.

Κατά την αξιολόγηση, ο γονότυπος μετατρέπεται εκ νέου σε φαινότυπο (`phenotype`), δηλαδή σε συγκεκριμένη λίστα από τρόφιμα και ποσότητες. Για κάθε τρόφιμο, τα θρεπτικά του στοιχεία πολλαπλασιάζονται με τον συντελεστή της ποσότητάς του και όλα αθροίζονται ώστε να προκύψουν τα συνολικά θρεπτικά μεγέθη του πλάνου. Οι θερμίδες υπολογίζονται με βάση τον κλασικό τύπο μετατροπής των μακροθρεπτικών, δηλαδή 4 kcal ανά γραμμάριο πρωτεϊνών και υδατανθράκων, 9 kcal για λιπαρά. Η τιμή του θερμιδικού στόχου συγκρίνεται με το αποτέλεσμα και η απόκλιση κανονικοποιείται ως ποσοστό επί του στόχου.

Για τα μακροθρεπτικά συστατικά υπολογίζεται ο λόγος πραγματικής προς επιθυμητή τιμή για κάθε ένα από αυτά και εφαρμόζεται μια κλιμακωτή ποινή (`penalty`), ανάλογα με το αποτέλεσμα του λόγου. Αν ο λόγος είναι μεγαλύτερος του 1,2 ή μικρότερος του 0,8, εφαρμόζεται μεγαλύτερη ποινή, ενώ εάν είναι εντός αυτών των ορίων εφαρμόζεται πολύ μικρότερη. Οι τιμές 0,8 και 1,2 επιλέχθηκαν, καθώς είναι επιθυμητό τα γραμμάρια του κάθε μακροθρεπτικού συστατικού να μην είναι κάτω από το 80% του στόχου που τέθηκε στην εφαρμογή για το εκάστοτε στοιχείο, ούτε περισσότερα από το 120% αυτού. Έτσι, προκύπτει ένας πίνακας, που περιλαμβάνει τις ποινές για κάθε μακροθρεπτικό συστατικό και η

τελική συνολική ποινή υπολογίζεται από τον μέσο όρο των στοιχείων αυτού του πίνακα. Η ίδια ακριβώς λογική εφαρμόζεται και για τα μικροθρεπτικά συστατικά. Το τελικό διάνυσμα καταλληλότητας είναι τριδιάστατο, αποτελείται από την θερμιδική παρέκκλιση, τον μέσο όρο των ποινών για τα μακροθρεπτικά συστατικά και τον αντίστοιχο μέσο όρο για τα μικροθρεπτικά συστατικά και το οποίο ο αλγόριθμος επιδιώκει να ελαχιστοποιήσει. Μαθηματικά η συνάρτηση καταλληλότητας μπορεί να εκφραστεί ως:

$$F(P) = \left[\begin{array}{c} \frac{|N_{kcal}(P) - T_{kcal}|}{T_{kcal}} \\ \frac{1}{3} \sum_{m \in \{protein, carb, fat\}} penalty(r_m) \\ \frac{1}{M} \sum_{\mu=1}^M |r_{\mu} - 1| \end{array} \right] \quad (5.2)$$

Στην παραπάνω εξίσωση, η συνάρτηση καταλληλότητας $F(P)$ αποτελείται από τρεις συνιστώσες $f_1(P)$, $f_2(P)$ και $f_3(P)$, καθεμία από τις οποίες αξιολογεί διαφορετική πτυχή της ποιότητας του παραγόμενου διατροφικού πλάνου. Ο πρώτος όρος $f_1(P)$, αφορά τον υπολογισμό της ποσοστιαίας απόκλισης των συνολικών θερμίδων του πλάνου από τον θερμιδικό στόχο του χρήστη. Στον τύπο αυτόν, N_{kcal} είναι το άθροισμα των θερμίδων που προκύπτει από τα τρόφιμα του πλάνου και T_{kcal} ο ημερήσιος θερμιδικός στόχος του χρήστη. Η χρήση του απόλυτου λόγου καθιστά τη συνάρτηση ανεξάρτητη από το πρόσημο της απόκλισης, επιτρέποντας στον αλγόριθμο να αντιμετωπίζει εξίσου τόσο τα ελλείμματα όσο και τα πλεονάσματα θερμίδων.

Ο δεύτερος όρος $f_2(P)$ αξιολογεί τον βαθμό επίτευξης των στόχων για τα μακροθρεπτικά συστατικά. Για κάθε μακροθρεπτικό στοιχείο m , υπολογίζεται ο λόγος:

$$r_m = \frac{N_m(P)}{T_m} \quad (5.3)$$

όπου $N_m(P)$ είναι η πραγματική ποσότητα του μακροθρεπτικού στοιχείου m στο πλάνο και T_m η αντίστοιχη επιθυμητή τιμή-στόχος. Για κάθε r_m υπολογίζεται η ποινή, όπως περιεγράφηκε προηγουμένως, η οποία μαθηματικά εκφράζεται:

$$penalty(r_m) = \begin{cases} 0, & 0,8 \leq r_m \leq 1,2 \\ |r_m - 1| & \text{αλλιώς} \end{cases} \quad (5.4)$$

Ο τρίτος όρος $f_3(P)$ εξετάζει την επάρκεια των μικροθρεπτικών στοιχείων, όπως βιταμίνες και μέταλλα. Για κάθε μικροθρεπτικό στοιχείο μ , υπολογίζεται η σχετική απόκλιση από τον στόχο:

$$r_{\mu} = \frac{N_{\mu}(P)}{T_{\mu}} \quad (5.5)$$

, ενώ M είναι το πλήθος των μικροθρεπτικών στοιχείων που εξετάζονται.

Ο μηχανισμός του EA διαμορφώνεται μέσω του αντικειμένου Engine της βιβλιοθήκης Jenetics. Στην παρούσα υλοποίηση, το Engine έχει ρυθμιστεί με βάση τις παραμέτρους που ορίζονται στο MoeaTuning, μια δομή η οποία περιλαμβάνει τις μεταβλητές οι οποίες έχουν αποθηκευμένες τις τιμές, που καθορίζουν το μέγεθος του πληθυσμού, τον αριθμό γενεών, το ποσοστό απογόνων, καθώς και τα ποσοστά crossover και mutation για κάθε τύπο τελεστή. Πιο συγκεκριμένα, οι τιμές που επιλέχθηκαν είναι οι εξής: μέγεθος πληθυσμού 400 άτομα, 500 γενεές, ποσοστό απογόνων 0,6, πιθανότητα index uniform crossover 0,5, πιθανότητα portion uniform crossover 0,6, πιθανότητα mutation 0,1 και πιθανότητα Gaussian mutation 0,25. Οι συγκεκριμένες τιμές προέκυψαν έπειτα από συστηματική ανάλυση της συμπεριφοράς του αλγορίθμου, η οποία παρουσιάζεται αναλυτικά στην ενότητα 6.4.2 του Κεφαλαίου 6. Για τους τελεστές μετάλλαξης χρησιμοποιούνται PartialAlterers, ώστε οι πρώτοι να εφαρμόζονται μόνο στα χρωμοσώματα που έχουν νόημα. Για τα χρωμοσώματα που αφορούν τους

δείκτες των τροφίμων, χρησιμοποιείται UniformCrossover και Mutator, ενώ για εκείνα που αφορούν τις ποσότητες, χρησιμοποιείται ξανά UniformCrossover, σε συνδυασμό όμως με GaussianMutator. Η επιλογή των PartialAlters έγινε για να διασφαλιστεί ότι οι τελεστές θα εφαρμόζονται επιλεκτικά, αποτρέποντας ανεπιθύμητες μεταλλάξεις, όπως την εναλλαγή ενός δείκτη με μια ποσότητα.

Οι παραπάνω τελεστές διαμορφώνουν τη συμπεριφορά του αλγορίθμου ως εξής:

- Ο UniformCrossover ανταλλάσσει γονίδια μεταξύ δύο γονέων με μια συγκεκριμένη πιθανότητα ανά θέση, δημιουργώντας νέους απογόνους που συνδυάζουν επιλογές τροφίμων και ποσοτήτων από διαφορετικά άτομα.
- Ο Mutator στα χρωμοσώματα των τροφίμων αντικαθιστά ένα τυχαίο γονίδιο με ένα νέο εντός των επιτρεπόμενων ορίων, προσθέτοντας ποικιλία και αποτρέποντας τη στασιμότητα.
- Αντίστοιχα, ο GaussianMutator στα χρωμοσώματα των ποσοτήτων προσθέτει μικρές τυχαίες αποκλίσεις σε σχέση με τη μέση τιμή, μιμούμενος έναν «τοπικό θόρυβο» που επιτρέπει μικρές προσαρμογές στις ποσότητες χωρίς να διαταράσσει τη συνολική δομή του πλάνου.

Όπως προαναφέρθηκε, για τη διαδικασία επιλογής και εξέλιξης του πληθυσμού υιοθετείται ο αλγόριθμος NSGA-II, του οποίου η θεωρητική λειτουργία και τα βασικά χαρακτηριστικά έχουν παρουσιαστεί στο Κεφάλαιο 3. Μετά την ολοκλήρωση του καθορισμένου αριθμού γενεών, ο αλγόριθμος διαθέτει ένα σύνολο λύσεων που βρίσκονται στην επιφάνεια Pareto, δηλαδή δεν μπορεί να βελτιωθεί καμία από αυτές σε έναν στόχο χωρίς επιδείνωση κάποιου άλλου. Για να επιλεγεί η τελική λύση, πραγματοποιείται κανονικοποίηση των τιμών καταλληλότητας. Η διαδικασία αυτή υπολογίζει, για κάθε στόχο, τις ελάχιστες και μέγιστες τιμές στο Pareto front και επαναχарτογραφεί κάθε λύση στην κλίμακα $[0, 1]$ σύμφωνα με τη σχέση:

$$(\text{τιμή} - \text{ελάχιστο}) / (\text{μέγιστο} - \text{ελάχιστο}) \quad (5.6)$$

Κατόπιν υπολογίζεται η ευκλείδεια απόσταση κάθε λύσης από το σημείο $(0, 0, 0)$ στον κανονικοποιημένο χώρο. Η λύση με τη μικρότερη απόσταση θεωρείται η πιο ισορροπημένη, καθώς επιτυγχάνει ταυτόχρονα καλή επίδοση σε όλους τους στόχους, χωρίς να υπερέχει υπερβολικά σε έναν εις βάρος των υπολοίπων.

Το τελικό αποτέλεσμα της διαδικασίας είναι ένα List<Pair<Food, Double>>, δηλαδή μια λίστα από ζεύγη που αντιστοιχούν σε κάθε επιλεγμένο τρόφιμο και την αντίστοιχη ποσότητά του. Η λίστα αυτή αποτελεί το βέλτιστο διατροφικό πλάνο που προτείνει ο αλγόριθμος στον χρήστη. Κάθε εκτέλεση του EA μπορεί να οδηγήσει σε διαφορετικό, αλλά ισοδύναμα βέλτιστο αποτέλεσμα, εξασφαλίζοντας ποικιλία και εξατομίκευση στις προτάσεις διατροφής.

5.4.4 Αποθήκευση και συγχρονισμός διατροφικού πλάνου

Μετά την επιτυχή παραγωγή του διατροφικού πλάνου από τον εξελικτικό αλγόριθμο, το πλάνο αποθηκεύεται τόσο τοπικά στη βάση δεδομένων Room όσο και απομακρυσμένα στη Firebase Firestore. Η διαδικασία αυτή υλοποιείται μέσω μιας συνάρτησης, που ονομάζεται «uploadDietLocallyAndInFirestore» και είναι υπεύθυνη για την ολοκληρωμένη αποθήκευση και τον συγχρονισμό των δεδομένων του διατροφικού πλάνου. Η βασική επιδίωξη της υλοποίησης είναι να εξασφαλιστεί ότι η αποθήκευση μπορεί να πραγματοποιηθεί ακόμη και χωρίς ενεργή σύνδεση στο διαδίκτυο, καθώς είναι θεμιτό η αποθήκευση του διατροφικού πλάνου να υποστηρίζει αυτή τη λειτουργικότητα.

Για την επίτευξη αυτής της απαίτησης αξιοποιείται ο WorkManager του Android, ο οποίος επιτρέπει τον προγραμματισμό και την εκτέλεση ασύγχρονων και επαναπρογραμματιζόμενων εργασιών στο παρασκήνιο, ακόμα και μετά από επανεκκίνηση της συσκευής ή προσωρινή απώλεια σύνδεσης δικτύου

[85]. Ο WorkManager παρέχει ένα ενοποιημένο API για τη διαχείριση διεργασιών που απαιτούν εγγυημένη εκτέλεση, εφαρμόζοντας στρατηγικές βηματικής επανάληψης (backoff policies), καθώς και περιορισμούς εκτέλεσης (constraints), όπως η απαίτηση ύπαρξης δικτύου ή φόρτισης της συσκευής. Η επιλογή της συγκεκριμένης βιβλιοθήκης έγινε καθώς συνιστάται για σενάρια επαναληπτικού συγχρονισμού δεδομένων μεταξύ τοπικής και απομακρυσμένης αποθήκευσης.

Η συνάρτηση `uploadDietLocallyAndInFirestore` αρχικά αποθηκεύει το διατροφικό πλάνο στη βάση Room, διασφαλίζοντας ότι υπάρχει πάντα τοπικό αντίγραφο του πλάνου, ακόμη και αν η απομακρυσμένη αποθήκευση αποτύχει προσωρινά. Στη συνέχεια, δημιουργεί ένα αντικείμενο, το οποίο αποτελεί την αναπαράσταση του πλάνου σε μορφή κατάλληλη για αποθήκευση στη Firestore. Αν ανιχνευθεί ότι υπάρχει διαθέσιμη σύνδεση στο διαδίκτυο, το πλάνο ανεβαίνει άμεσα στη Firestore. Αντίθετα, αν η συσκευή βρίσκεται εκτός σύνδεσης, η συνάρτηση καταχωρεί προσωρινά το πλάνο σε έναν πίνακα της Room και προγραμματίζει τη μελλοντική αποστολή του μόλις το δίκτυο επανέλθει. Το πλάνο αποθηκεύεται ως αντικείμενο τύπου `DietSyncEntity`, το οποίο περιέχει το ίδιο το διατροφικό πλάνο, καθώς και το `dietId` που αντιστοιχεί στο `id` του τρέχοντα χρήστη. Αυτή η αντιστοίχιση `dietId-user id`, γίνεται καθώς υπάρχει πιθανότητα ο χρήστης να αποσυνδεθεί από τον λογαριασμό του και να συνδεθεί με άλλα διαπιστευτήρια, χωρίς το διατροφικό του πλάνο να έχει αποθηκευτεί στη Firestore. Σε περίπτωση που δεν αποθηκευόταν το `id` του χρήστη και συνέβαινε ο συγχρονισμός, η δίαιτα του πρώτου χρήστη, που αποσυνδέθηκε, θα αποθηκευόταν ως δίαιτα του τρέχοντος συνδεδεμένου χρήστη, κάτι που προφανώς δεν είναι επιθυμητό και θα παραβίαζε την ιδιωτικότητα των χρηστών.

Ο προγραμματισμός του ανεβάσματος στην εφαρμογή, πραγματοποιείται μέσω της κλάσης `SyncDietSchedulerImpl`, η οποία αποτελεί τον ενδιάμεσο μηχανισμό συγχρονισμού. Η συνάρτηση «`scheduleUploadDietWorker`» που περιλαμβάνεται σε αυτή την κλάση δημιουργεί ένα `OneTimeWorkRequest` για την εργασία αποστολής του πλάνου. Το αίτημα αυτό ρυθμίζεται να εκτελεστεί μόνο όταν υπάρχει ενεργή σύνδεση στο δίκτυο, θέτοντας ως `constraint` την παράμετρο `NetworkType.CONNECTED`. Παράλληλα, ορίζεται πολιτική εκθετικής καθυστέρησης (backoff policy) ώστε, σε περίπτωση αποτυχίας, η εργασία να επαναληφθεί αυτόματα μετά από σύντομο χρονικό διάστημα. Το αίτημα στη συνέχεια προωθείται προς εκτέλεση μέσω του `WorkManager.enqueue(workRequest)`, το οποίο αναλαμβάνει τη διαχείριση και την επανεκτέλεση της εργασίας μέχρι την επιτυχή ολοκλήρωσή της.

Η πραγματική διαδικασία αποστολής του διατροφικού πλάνου στη Firestore εκτελείται από την κλάση `UploadDietWorker`, η οποία επεκτείνει την `CoroutineWorker` του Android. Κατά την εκτέλεση της μεθόδου «`doWork`», αναζητείται το εκκρεμές διατροφικό πλάνο που αντιστοιχεί στον χρήστη. Αν βρεθεί, μετατρέπεται σε μορφή `DietForFirestore` μέσω της συνάρτησης «`toDietForFirestore`» και αποστέλλεται στην Firestore. Σε περίπτωση επιτυχίας, διαγράφεται τοπικά η εκκρεμής εγγραφή, αφού πλέον η αποθήκευση στην απομακρυσμένη βάση ολοκληρώθηκε. Αν η αποστολή αποτύχει, ο `WorkManager` επαναπρογραμματίζει αυτόματα τη διαδικασία βάσει της πολιτικής backoff, εξασφαλίζοντας ότι η συγχρονισμένη αποθήκευση θα πραγματοποιηθεί μόλις επανέλθει η συνδεσιμότητα.

5.5 Αρχική σελίδα της εφαρμογής

Έχοντας ολοκληρώσει την παρουσίαση της διαδικασίας παραγωγής και αποθήκευσης του διατροφικού πλάνου στην ενότητα 5.4, η παρούσα ενότητα επικεντρώνεται στην οθόνη μέσω της οποίας ο χρήστης ξεκινά να αλληλεπιδρά με τα αποτελέσματα αυτής της διαδικασίας, δηλαδή το αρχικό περιβάλλον της εφαρμογής. Η αρχική σελίδα της εφαρμογής (Σχήμα 5.12 και Σχήμα 5.13) λειτουργεί ως πίνακας

ελέγχου (dashboard), συγκεντρώνοντας συνοπτικές πληροφορίες για το διατροφικό πλάνο, το βάρος του χρήστη και τις βασικές ενέργειες που μπορεί να εκτελέσει. Στόχος της οθόνης αυτής δεν είναι η προσθήκη νέας λειτουργικότητας, αλλά η παροχή μιας άμεσης, κατανοητής και αισθητικά ενοποιημένης επισκόπησης κάποιων βασικών λειτουργιών.

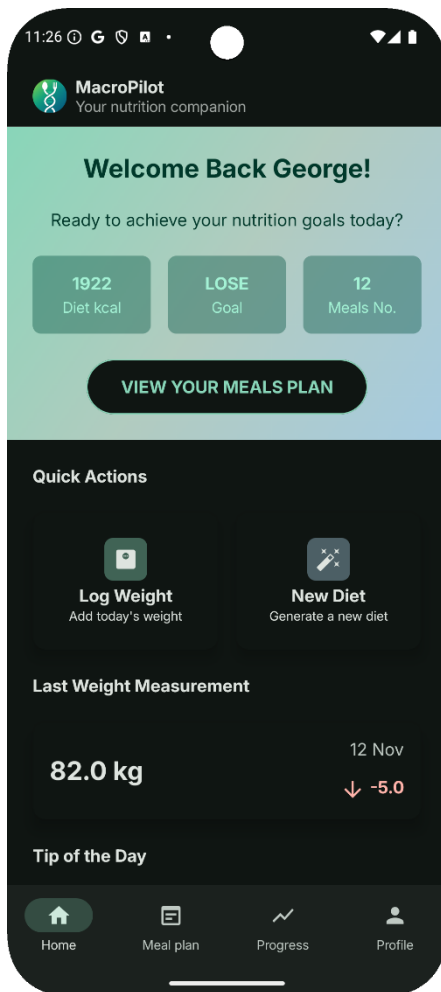
Στο άνω τμήμα της οθόνης βρίσκεται μια γραμμή εργαλείων, η οποία περιλαμβάνει το λογότυπο και το όνομα της εφαρμογής. Αμέσως κάτω από αυτήν ακολουθεί η καρτέλα του διατροφικού πλάνου, η οποία αποτελεί το κύριο σημείο πληροφόρησης του χρήστη. Στην καρτέλα αυτή εμφανίζεται αρχικά ένα μήνυμα καλωσορίσματος και κάτω από αυτό τρία πλαίσια πληροφοριών. Το πρώτο εμφανίζει τις συνολικές θερμίδες του διατροφικού πλάνου που έχει παραχθεί από τον εξελικτικό αλγόριθμο, το δεύτερο εμφανίζει τον στόχο του χρήστη (Lose, Maintain, Gain), ενώ το τρίτο δείχνει το πλήθος των τροφών που περιλαμβάνει το πλάνο. Εάν ο χρήστης κατά την πρώτη του είσοδο ακυρώσει τη διαδικασία παραγωγής πλάνου, τα συγκεκριμένα πεδία δεν έχουν διαθέσιμα δεδομένα και απεικονίζονται με «NA», υποδεικνύοντας την έλλειψη διαθέσιμης πληροφορίας.

Κάτω από τα πλαίσια αυτά βρίσκεται ένα κουμπί που οδηγεί τον χρήστη στην οθόνη παρουσίασης και επεξεργασίας του διατροφικού πλάνου, δίνοντάς του τη δυνατότητα να δει αναλυτικά τα τρόφιμα που περιλαμβάνονται ή να πραγματοποιήσει αλλαγές. Ακολουθεί η ενότητα «Quick Actions», η οποία περιλαμβάνει δύο κουμπιά γρήγορης πρόσβασης. Το πρώτο ανακατευθύνει τον χρήστη στην οθόνη καταγραφής βάρους, ενώ το δεύτερο επιτρέπει την παραγωγή νέου διατροφικού πλάνου, επανεκκινώντας τη διαδικασία εκτέλεσης του εξελικτικού αλγορίθμου.

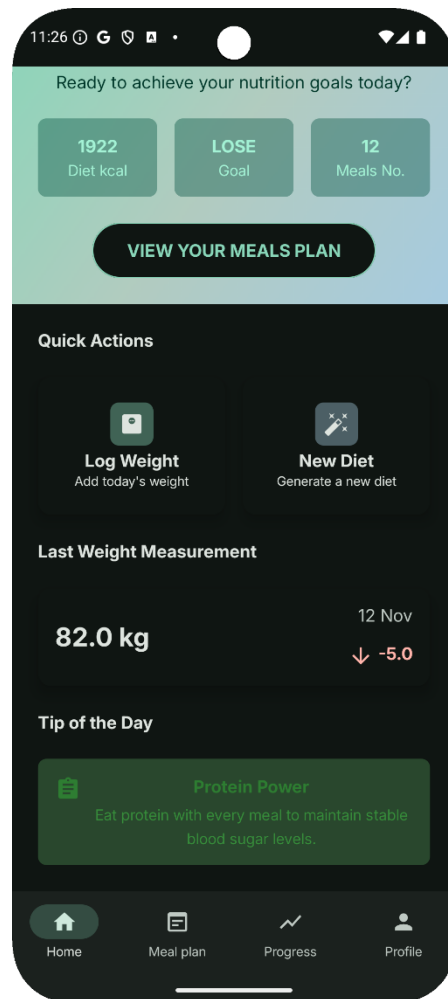
Στο επόμενο τμήμα της οθόνης εμφανίζεται το πλαίσιο πληροφοριών βάρους, το οποίο παρουσιάζει την τελευταία καταχώρηση βάρους του χρήστη, την ημερομηνία της καταγραφής, καθώς και ένα γραφικό που δείχνει αν το βάρος έχει αυξηθεί ή μειωθεί σε σχέση με την προηγούμενη μέτρηση. Αν δεν υπάρχει διαθέσιμη μέτρηση εντός των τελευταίων τριάντα ημερών, οι τιμές εμφανίζονται ως «NA», επιλογή η οποία θα αιτιολογηθεί λεπτομερώς στην ενότητα που αφορά την παρακολούθηση βάρους.

Στο κάτω μέρος της οθόνης εμφανίζεται ένα διατροφικό tip, το οποίο λειτουργεί με την ίδια λογική που χρησιμοποιείται και στην οθόνη αναμονής κατά την παραγωγή του διατροφικού πλάνου, χωρίς όμως εναλλαγή ή animation. Τέλος, η σελίδα περιλαμβάνει στο κάτω μέρος ένα πλαίσιο πλοήγησης (navigation bar), το οποίο παρέχει άμεση πρόσβαση στην αρχική σελίδα, στην οθόνη διατροφικού πλάνου, στην οθόνη καταγραφής βάρους και στο προφίλ. Παρότι όλο το κυρίως περιεχόμενο της αρχικής σελίδας, όπως και των υπόλοιπων οθονών που μπορεί να ανακατευθυνθεί ο χρήστης, παρουσιάζεται μέσα σε μια κατακόρυφα μετακινούμενη διάταξη, το navigation bar παραμένει μόνιμα προσκολλημένο στο κάτω μέρος της οθόνης, χωρίς να επηρεάζεται από την κύλιση.

Κατά την πρώτη εκκίνηση της εφαρμογής και εμφάνισης της αρχικής σελίδας, εμφανίζεται ένα μήνυμα που ζητά από τον χρήστη να επιτρέψει την αποστολή ειδοποιήσεων. Αν ο χρήστης συναινέσει, η εφαρμογή αποκτά άδεια να αποστέλλει ειδοποιήσεις για τα προγραμματισμένα γεύματα (breakfast,



Σχήμα 5.12: Το πρώτο μέρος της αρχικής σελίδας με εστίαση στην κορυφή της



Σχήμα 5.13: Το δεύτερο μέρος της αρχικής σελίδας με εστίαση στο κάτω μέρος της

lunch, dinner κλπ.), σύμφωνα με τις ώρες που έχει ορίσει στο προφίλ του. Αν αρνηθεί, οι ειδοποιήσεις απενεργοποιούνται και η επιλογή μπορεί να αλλάξει μόνο μέσα από τις ρυθμίσεις της συσκευής.

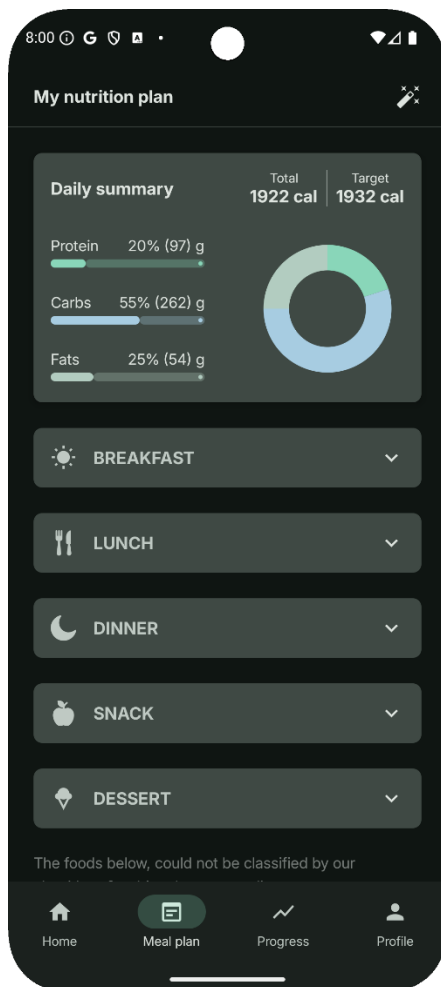
Όσον αφορά τη λογική λειτουργίας της οθόνης, τα δεδομένα του homepage ανακτώνται μέσω του HomepageViewModel, το οποίο υλοποιεί τη λογική φόρτωσης και ενημέρωσης της κατάστασης της σελίδας. Για την ανάκτηση των δεδομένων του διατροφικού πλάνου και προκειμένου να ολοκληρωθεί η εξήγηση της αρχιτεκτονικής συγχρονισμού που περιγράφηκε προηγουμένως, αξιοποιούνται οι ίδιες δομές WorkManager που χρησιμοποιήθηκαν και κατά την αποθήκευση. Συγκεκριμένα, στο στάδιο της αρχικοποίησης του HomepageViewModel, προγραμματίζεται η περιοδική εκτέλεση του FetchDietWorker μέσω της κλάσης SyncDietSchedulerImpl, καλώντας τη αντίστοιχη συνάρτηση. Με αυτόν τον τρόπο, ορίζεται περιοδικός συγχρονισμός ανά τριάντα λεπτά, ώστε η εφαρμογή να ανανεώνει τα δεδομένα του διατροφικού πλάνου από τη Firestore αυτόματα και τα δεδομένα να είναι πάντα συγχρονισμένα. Όπως και ο UploadDietWorker, έτσι και ο FetchDietWorker, επεκτείνει την CoroutineWorker και στην υλοποίηση της συνάρτησης «doWork» εκτελεί μια άλλη συνάρτηση με όνομα «fetchDiets» του DietGenerationRepository, η οποία επικοινωνεί με τη Firestore και ενημερώνει τα τοπικά δεδομένα της Room με το πιο πρόσφατο πλάνο.

Πέραν του περιοδικού συγχρονισμού, κατά την αρχικοποίηση του HomepageViewModel εκτελείται επίσης ένας άμεσος συγχρονισμός για την ανάκτηση των δεδομένων. Αυτό επιτυγχάνεται με τη διαδοχική κλήση των συναρτήσεων «syncPendingDiet» και «fetchDiets» του

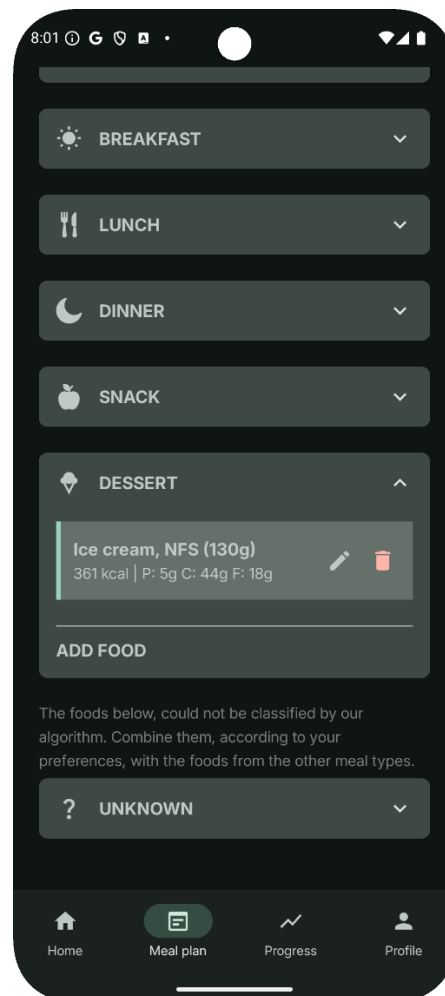
DietGenerationRepositoryImpl. Η syncPendingDiet αποστέλλει στη Firestore τυχόν διατροφικά πλάνα που είχαν αποθηκευτεί τοπικά σε εκκρεμότητα, λόγω προηγούμενης απουσίας σύνδεσης,, ενώ η fetchDiets ανακτά το πιο πρόσφατο πλάνο του χρήστη από την απομακρυσμένη βάση και το ενημερώνει στη Room. Παρόλο που η διαδικασία συγχρονισμού έχει αυτοματοποιηθεί μέσω του UploadDietWorker, η άμεση εκτέλεση της syncPendingDiet λειτουργεί ως μηχανισμός ασφαλείας, διασφαλίζοντας την ακεραιότητα των δεδομένων ακόμα και σε περιπτώσεις όπου ο προγραμματισμένος worker δεν έχει ακόμη εκτελεστεί ή έχει αναβληθεί λόγω περιορισμών συστήματος ή δικτύου.

5.6 Επισκόπηση και επεξεργασία διατροφικού πλάνου

Η οθόνη επισκόπησης και επεξεργασίας του διατροφικού πλάνου, που φαίνεται στα σχήματα 5.14 και 5.15, αποτελεί το κεντρικό σημείο στο οποίο ο χρήστης μπορεί να δει αναλυτικά το πλάνο που παρήχθη από τον εξελικτικό αλγόριθμο, να τροποποιήσει τις ποσότητες των τροφών, να προσθέσει ή να αφαιρέσει τρόφιμα και, γενικότερα, να διαμορφώσει τη διατροφή του σύμφωνα με τις προσωπικές του προτιμήσεις.



Σχήμα 5.14: Η οθόνη επισκόπησης του διατροφικού πλάνου

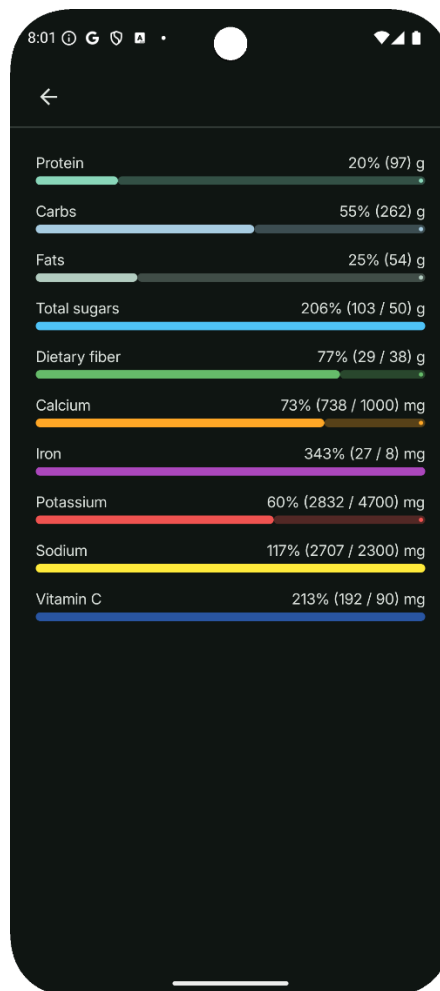


Σχήμα 5.15: Η οθόνη επισκόπησης του διατροφικού πλάνου με εστίαση στη λίστα τροφών μιας κατηγορίας γεύματος και στην κατηγορία unknown

Στο επάνω μέρος της οθόνης υπάρχει μία γραμμή εργαλείων, που υποδεικνύει το τμήμα της εφαρμογής στο οποίο βρίσκεται ο χρήστης, μέσω της ένδειξης «My nutrition plan», και αντιστοιχεί ταυτόχρονα στο ενεργό στοιχείο του navigation bar. Δεξιά του τίτλου υπάρχει ένα εικονίδιο επεξεργασίας, το

πάτημα του οποίου ενεργοποιεί ένα αναδυόμενο παράθυρο επιβεβαίωσης. Εφόσον ο χρήστης επιβεβαιώσει την ενέργεια, ξεκινά η διαδικασία δημιουργίας ενός νέου διατροφικού πλάνου, η οποία επαναλαμβάνει τη ροή παραγωγής που περιεγράφηκε στην ενότητα 5.4.

Αμέσως κάτω από τη γραμμή εργαλείων βρίσκεται μία κάρτα επισκόπησης θερμίδων και μακροθρεπτικών στοιχείων, η οποία συγκρίνει τα συνολικά διατροφικά δεδομένα του πλάνου με τους στόχους του χρήστη. Στην κάρτα αυτή εμφανίζονται αφενός οι θερμίδες του παραγόμενου πλάνου και αφετέρου ο θερμιδικός στόχος, ενώ ταυτόχρονα παρουσιάζονται τα ποσοστά πρωτεϊνών, υδατανθράκων και λιπαρών τόσο σε γραμμική μορφή όσο και σε κυκλικό διάγραμμα. Η γραμμική απεικόνιση δείχνει τα αντίστοιχα ποσοστά και γραμμάρια για κάθε μακροθρεπτικό συστατικό, ενώ το κυκλικό γράφημα παρέχει μία συνοπτική οπτικοποίηση της κατανομής τους. Αν κάποια από τις τιμές αποκλίνει από τον στόχο, ακόμα και ελάχιστα, το αντίστοιχο κείμενο εμφανίζεται με κόκκινο χρώμα, ώστε να γίνεται άμεσα αντιληπτή η απόκλιση. Με το πάτημα της κάρτας, ο χρήστης μεταβαίνει σε ξεχωριστή οθόνη αναλυτικής διατροφικής επισκόπησης, όπου μπορεί να δει λεπτομερώς τα ποσοστά τόσο των μακροθρεπτικών όσο και των μικροθρεπτικών στοιχείων του πλάνου σε σχέση με τους στόχους του (Σχήμα 5.16).



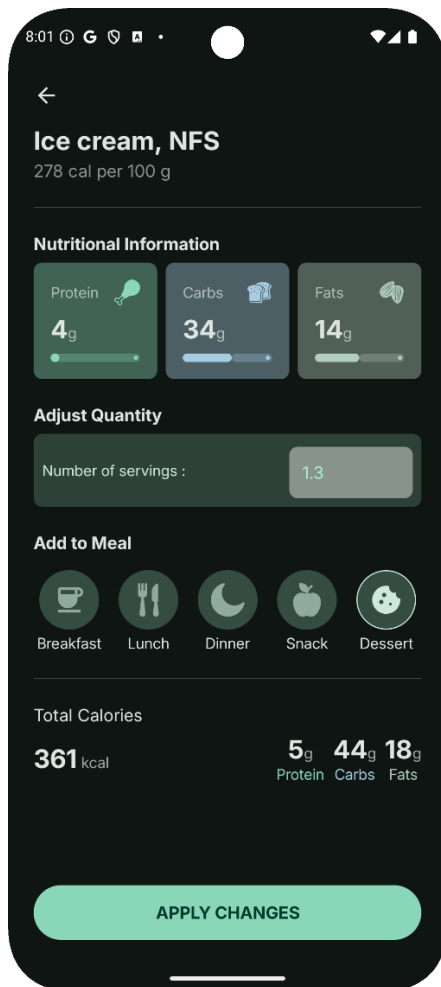
Σχήμα 5.16: Η οθόνη αναλυτικής διατροφικής επισκόπησης των θρεπτικών στοιχείων του διατροφικού πλάνου. Το επόμενο τμήμα της οθόνης αποτελείται από μια σειρά γραφικών στοιχείων διεπαφής (UI components), καθένα από τα οποία αντιπροσωπεύει μία κατηγορία γεύματος. Οι διαθέσιμες κατηγορίες είναι εκείνες που επέλεξε ο χρήστης κατά τη διαδικασία αρχικοποίησης προφίλ (π.χ. Breakfast, Lunch, Dinner, Snack, Dessert), ενώ προστίθεται και μία επιπλέον κατηγορία με την ονομασία UNKNOWN.

Πάνω από την τελευταία κατηγορία εμφανίζεται ένα ενημερωτικό μήνυμα που εξηγεί ότι τα τρόφιμα που εντάσσονται σε αυτή δεν ταξινομήθηκαν από τον αλγόριθμο και μπορούν να ενταχθούν από τον χρήστη σε οποιοδήποτε γεύμα επιθυμεί. Κάθε γραφικό στοιχείο κατηγορίας μπορεί να αναπτυχθεί ή να συμπυκνωθεί. Όταν ο χρήστης πατήσει πάνω του, εφόσον το στοιχείο είναι συμπυκνωμένο, η κατηγορία επεκτείνεται παρουσιάζοντας όλα τα τρόφιμα που περιλαμβάνει το διατροφικό πλάνο και ανήκουν σε αυτή. Για κάθε τρόφιμο εμφανίζονται η ονομασία του, η ποσότητα που πρέπει να καταναλωθεί σε γραμμάρια, οι θερμίδες, καθώς και τα μακροθρεπτικά συστατικά που αντιστοιχούν στη συγκεκριμένη ποσότητα. Κάθε γραμμή τροφίμου συνοδεύεται από δύο εικονίδια, ένα για επεξεργασία και ένα για διαγραφή.

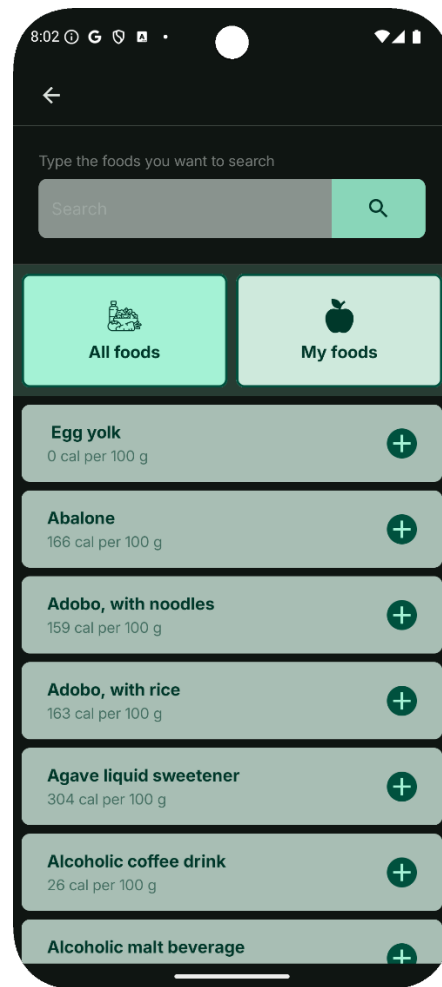
Πατώντας το κουμπί επεξεργασίας, ο χρήστης μεταφέρεται στην οθόνη επεξεργασίας τροφίμου (Σχήμα 5.17), όπου μπορεί να δει και να τροποποιήσει τα χαρακτηριστικά του επιλεγμένου προϊόντος. Στο επάνω μέρος της οθόνης υπάρχει μια γραμμή εργαλείων με κουμπί επιστροφής στην προηγούμενη σελίδα, ενώ ακολουθεί ο τίτλος του τροφίμου και η θερμιδική του αξία ανά 100 γραμμάρια. Κάτω από αυτά εμφανίζονται σε μορφή καρτών οι ποσότητες των μακροθρεπτικών στοιχείων ανά 100 γραμμάρια. Στη συνέχεια, ο χρήστης έχει τη δυνατότητα να καθορίσει τη ποσότητα κατανάλωσης μέσω ενός πεδίου εισαγωγής, όπου δηλώνει τον αριθμό των μερίδων που επιθυμεί να καταναλώσει. Μία μερίδα αντιστοιχεί σε 100 γραμμάρια προϊόντος, οπότε, για παράδειγμα, η τιμή «2» σημαίνει κατανάλωση 200 γραμμαρίων.

Αμέσως πιο κάτω, εμφανίζεται μια σειρά με όλες τις κατηγορίες γευμάτων. Αν το τρόφιμο έχει ταξινομηθεί σε κάποια από αυτές, η αντίστοιχη επιλογή είναι προεπιλεγμένη. Αν ανήκει στην κατηγορία UNKNOWN, προεπιλεγμένη είναι η κατηγορία Breakfast. Η αλλαγή κατηγορίας τροφίμου αφορά την κατηγορία του μέσα στο διατροφικό πλάνο και όχι την κατηγορία που έχει αποθηκευτεί για το ίδιο το τρόφιμο. Στο κάτω μέρος της οθόνης εμφανίζονται οι συνολικές θερμίδες και τα μακροθρεπτικά συστατικά που αντιστοιχούν στην επιλεγμένη ποσότητα, τα οποία ενημερώνονται δυναμικά καθώς ο χρήστης μεταβάλλει την ποσότητα. Η διαδικασία ολοκληρώνεται με το πάτημα του κουμπιού «APPLY CHANGES», το οποίο αποθηκεύει τις τροποποιήσεις στη Room και προγραμματίζει τη μελλοντική ενημέρωση της Firestore μέσω WorkManager, σε περίπτωση που δεν υπάρχει διαθέσιμη σύνδεση στο διαδίκτυο.

Επιστρέφοντας στην οθόνη επισκόπησης, η επιλογή διαγραφής τροφίμου αφαιρεί το επιλεγμένο προϊόν από το διατροφικό πλάνο και ενημερώνει άμεσα τα συνολικά στοιχεία θερμίδων και μακροθρεπτικών. Κάτω από τη λίστα των τροφίμων κάθε κατηγορίας εμφανίζεται η επιλογή «ADD FOOD», η οποία οδηγεί τον χρήστη στην οθόνη αναζήτησης τροφίμου (Σχήμα 5.18). Η οθόνη αναζήτησης περιλαμβάνει ένα πεδίο εισαγωγής (search bar) στο επάνω μέρος, μέσω του οποίου ο χρήστης μπορεί να πληκτρολογήσει το όνομα ενός τροφίμου. Από προεπιλογή, εμφανίζεται η πλήρης λίστα όλων των διαθέσιμων τροφίμων, η οποία αντλείται μέσω της συνάρτησης «getFoodsFromPager», που εξηγήθηκε στην προηγούμενη ενότητα. Όταν ο χρήστης αρχίζει να πληκτρολογεί, εκτελείται κλήση στο USDA FoodData Central API, παρόμοια με εκείνη που χρησιμοποιείται κατά την παραγωγή του διατροφικού πλάνου, αλλά με πρόσθετη παράμετρο «query», που αντιστοιχεί στο περιεχόμενο του πεδίου αναζήτησης. Τα αποτελέσματα της κλήσης αντικαθιστούν προσωρινά την αρχική λίστα, παρουσιάζοντας μόνο τα τρόφιμα που ταιριάζουν με τον όρο αναζήτησης. Αν ο χρήστης διαγράψει το περιεχόμενο του πεδίου και το αφήσει κενό, η λίστα επανέρχεται στα αρχικά δεδομένα.



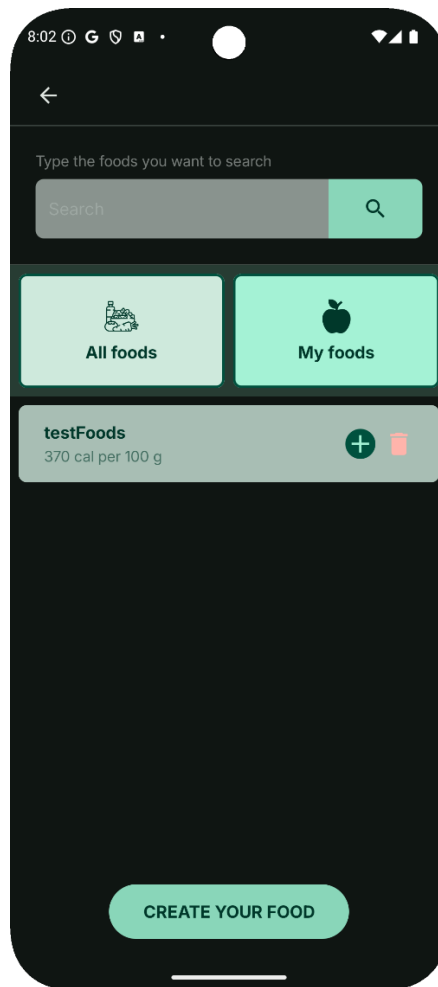
Σχήμα 5.17: Η οθόνη επεξεργασίας τροφίμου



Σχήμα 5.18: Η οθόνη αναζήτησης τροφίμου

Πατώντας σε ένα τρόφιμο από τη λίστα, ο χρήστης μεταφέρεται στην οθόνη επεξεργασίας τροφίμου, όπου μπορεί να καθορίσει την ποσότητα και το γεύμα στο οποίο θα ενταχθεί το επιλεγμένο τρόφιμο, πριν το προσθέσει οριστικά στο διατροφικό του πλάνο. Ανάμεσα στο πεδίο αναζήτησης και τη λίστα τροφίμων υπάρχουν δύο καρτέλες, η «All foods» και η «My foods». Η πρώτη είναι ενεργή από προεπιλογή και περιλαμβάνει όλα τα δεδομένα που εξηγήθηκαν παραπάνω, ενώ η δεύτερη περιλαμβάνει αποκλειστικά τα τρόφιμα που έχουν δημιουργηθεί από τον ίδιο τον χρήστη. Στην επιλογή «My foods», κάθε στοιχείο της λίστας συνοδεύεται από κουμπί προσθήκης και κουμπί διαγραφής, προσφέροντας πλήρη έλεγχο στη διαχείριση των προσωπικών τροφών, όπως παρουσιάζεται στο Σχήμα 5.19.

Στο κάτω μέρος της ίδιας οθόνης υπάρχει το κουμπί «CREATE YOUR FOOD», το οποίο οδηγεί στην οθόνη δημιουργίας νέου τροφίμου (Σχήμα 5.20 και Σχήμα 5.21). Εκεί, ο χρήστης καλείται να συμπληρώσει υποχρεωτικά τα πεδία ονόματος τροφίμου, θερμιδών ανά 100 γραμμάρια, ποσότητας πρωτεϊνών, υδατανθράκων και λιπαρών για την αντίστοιχη ποσότητα και να επιλέξει κατηγορία γεύματος. Προαιρετικά μπορεί να συμπληρώσει επιπλέον τα μικροθρεπτικά στοιχεία, όπως ίνες, σάκχαρα, ασβέστιο, σίδηρο, κάλιο, νάτριο και βιταμίνη C, τα οποία εμφανίζονται με το πάτημα της επιλογής «Show». Το κουμπί «CREATE YOUR FOOD» στο κάτω μέρος παραμένει απενεργοποιημένο έως ότου συμπληρωθούν σωστά όλα τα υποχρεωτικά πεδία. Με την επιβεβαίωση, το νέο τρόφιμο αποθηκεύεται στη Firestore, καθιστώντας το διαθέσιμο τόσο στην επιλογή My foods όσο και στη διαδικασία δημιουργίας επόμενων διατροφικών πλάνων. Για την ολοκλήρωση αυτής της διαδικασίας



Σχήμα 5.19: Η οθόνη αναζήτησης τροφίμου με επιλεγμένη τη ροή «My foods» απαιτείται ενεργή σύνδεση στο διαδίκτυο, ωστόσο η εφαρμογή χειρίζεται τυχόν σφάλματα ή απουσία σύνδεσης με τον ίδιο αξιόπιστο τρόπο που εφαρμόζεται και στις υπόλοιπες ροές της εφαρμογής.

Σχήμα 5.20: Η οθόνη δημιουργίας νέου τροφίμου με εστίαση στα υποχρεωτικά πεδία

Σχήμα 5.21: Η οθόνη δημιουργίας νέου τροφίμου με εστίαση στα μικροθρεπτικά στοιχεία

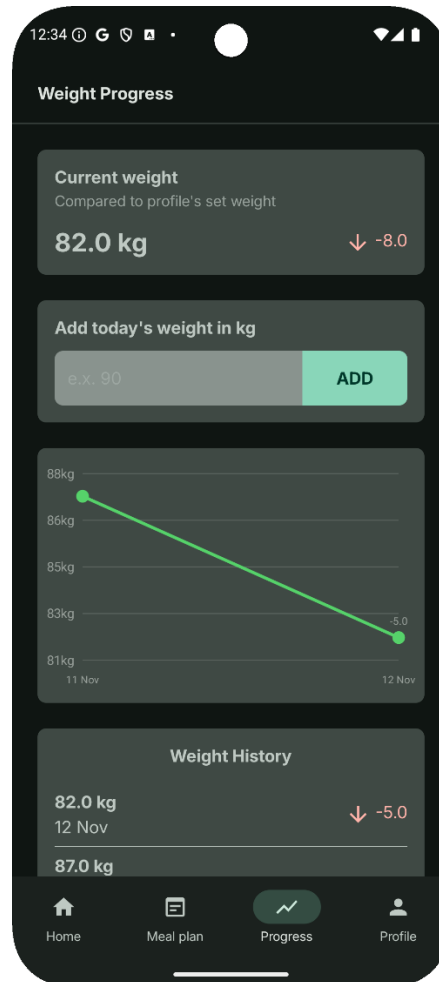
5.7 Οθόνη παρακολούθησης και καταγραφής βάρους

Η οθόνη παρακολούθησης και καταγραφής βάρους (Σχήμα 5.22) αποτελεί ένα σημαντικό μέρος της εφαρμογής, καθώς επιτρέπει στον χρήστη να καταγράφει το βάρος του, να παρακολουθεί την πορεία του μέσα στον χρόνο και να αξιολογεί την πρόδό του σε σχέση με τον στόχο που είχε ορίσει κατά την αρχικοποίηση του προφίλ του. Η οθόνη αυτή είναι προσβάσιμη είτε από την αρχική σελίδα (homepage) μέσω της επιλογής “Quick Actions”, είτε απευθείας μέσω του navigation bar, καθώς αποτελεί μία από τις τέσσερις βασικές ενότητες της εφαρμογής.

Η λογική λειτουργίας της οθόνης στηρίζεται στην ανάκτηση και στο φιλτράρισμα των δεδομένων βάρους από τη Firebase Firestore. Κατά την είσοδο του χρήστη στην οθόνη, εκτελείται συνάρτηση που ανακτά όλα τα διαθέσιμα δεδομένα βάρους του χρήστη και στη συνέχεια φιλτράρει τα αποτελέσματα, διατηρώντας μόνο τις τελευταίες τριάντα εγγραφές. Εάν ο χρήστης δεν έχει πραγματοποιήσει καταγραφή βάρους για περισσότερες από τριάντα ημέρες, η οθόνη εμφανίζει κατάλληλα μηνύματα στα γραφικά στοιχεία, υποδεικνύοντας ότι δεν υπάρχουν διαθέσιμα δεδομένα.

Στο επάνω μέρος της οθόνης βρίσκεται μια γραμμή εργαλείων με τον τίτλο «Weight Progress», που υποδεικνύει το ενεργό τμήμα στο οποίο βρίσκεται ο χρήστης. Ακριβώς κάτω από αυτήν, εμφανίζεται ένα γραφικό στοιχείο το οποίο παρουσιάζει το τρέχον βάρος του χρήστη, όπως αυτό έχει καταχωρηθεί κατά την πιο πρόσφατη εισαγωγή δεδομένων. Δίπλα στην ένδειξη του τρέχοντος βάρους προβάλλεται

μια δεύτερη πληροφορία, η οποία δείχνει τη μεταβολή του βάρους σε σχέση με το αρχικό βάρος που είχε δηλωθεί κατά τη δημιουργία του προφίλ. Το στοιχείο αυτό επιτρέπει στον χρήστη να έχει μια συνοπτική εικόνα της συνολικής του προόδου, υποδεικνύοντας εάν έχει επιτευχθεί απώλεια ή αύξηση βάρους. Σε περιπτώσεις απώλειας, η διαφορά εμφανίζεται με κόκκινη ένδειξη και βέλος προς τα κάτω, ενώ αντίθετα, στην περίπτωση αύξησης, η ένδειξη λαμβάνει πράσινο χρώμα και κατεύθυνση προς τα πάνω.



Σχήμα 5.22: Η οθόνη παρακολούθησης και καταγραφής βάρους

Ακολουθεί η ενότητα καταχώρησης βάρους, η οποία περιλαμβάνει ένα πεδίο εισαγωγής και ένα κουμπί “ADD”. Ο χρήστης μπορεί να πληκτρολογήσει το βάρος του σε κιλά και, πατώντας το κουμπί, να το προσθέσει στη βάση δεδομένων. Εάν πραγματοποιηθούν πολλαπλές καταχωρήσεις την ίδια ημέρα, η εφαρμογή δεν δημιουργεί νέα εγγραφή, αλλά ενημερώνει την ήδη υπάρχουσα, εξασφαλίζοντας ότι κάθε ημερομηνία αντιστοιχεί σε μία μοναδική μέτρηση, όπως είχε παρουσιαστεί και στην υποενότητα 4.5. Η λειτουργία αυτή συμβάλλει στη διατήρηση της συνέπειας και καθαρότητας των δεδομένων, αποτρέποντας περιττές εγγραφές. Επίσης, για τη λειτουργία αυτή είναι απαραίτητη η σύνδεση στο διαδίκτυο. Σε αντίθετη περίπτωση εμφανίζονται τα κλασικά για την εφαρμογή μηνύματα σφάλματος.

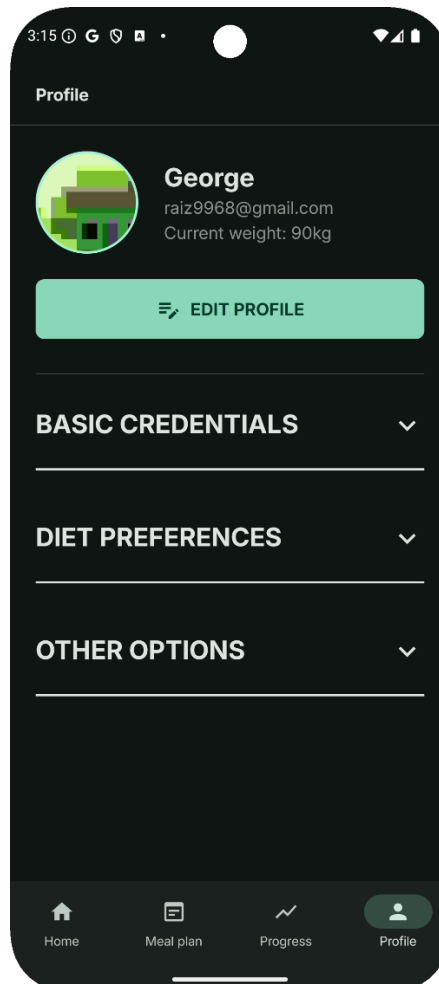
Κάτω από το πεδίο εισαγωγής βρίσκεται ένα διάγραμμα εξέλιξης βάρους, το οποίο απεικονίζει γραφικά την πορεία του βάρους του χρήστη σύμφωνα με τις τελευταίες επτά καταχωρήσεις ή εφόσον υπάρχουν λιγότερες καταχωρήσεις, όσες είναι διαθέσιμες. Στον κατακόρυφο άξονα του διαγράμματος παρουσιάζονται τα κιλά, ενώ στον οριζόντιο οι ημερομηνίες των καταγραφών. Το διάγραμμα αυτό προσφέρει μια άμεση οπτικοποίηση της πορείας του χρήστη, επιτρέποντάς του να εντοπίζει εύκολα

τάσεις, όπως σταδιακή απώλεια ή αύξηση βάρους. Η γραφική απεικόνιση υπολογίζεται δυναμικά κάθε φορά που προστίθεται νέα μέτρηση, ώστε η απεικόνιση να παραμένει πάντοτε ενημερωμένη.

Αμέσως μετά, ακολουθεί η ενότητα «Weight History», η οποία παρουσιάζει ένα εκτενέστερο ιστορικό ζυγίσεων σε μορφή λίστας. Η λίστα περιλαμβάνει τις τελευταίες τριάντα καταχωρήσεις του χρήστη και για κάθε εγγραφή εμφανίζει την ημερομηνία καταχώρησης, το αντίστοιχο βάρος, καθώς και ένα γραφικό στοιχείο ένδειξης της μεταβολής από την προηγούμενη μέτρηση. Ένα βέλος προς τα κάτω συνοδεύει με κόκκινη ένδειξη τις περιπτώσεις απώλειας βάρους, ενώ ένα βέλος προς τα πάνω με πράσινη ένδειξη δείχνει αύξηση. Η απουσία μεταβολής υποδεικνύεται με κόκκινο χρώμα, ενώ για την πρώτη καταγραφή δεν παρουσιάζεται αυτό το γραφικό στοιχείο του βέλους.

5.8 Οθόνη προφίλ χρήστη

Η οθόνη προφίλ (Σχήμα 5.23) επιτρέπει στον χρήστη να διαχειρίζεται τα προσωπικά του στοιχεία, να τροποποιεί τις διατροφικές του προτιμήσεις και να εκτελεί βασικές ενέργειες διαχείρισης λογαριασμού, όπως την αποσύνδεση ή τη διαγραφή του προφίλ του. Τα δεδομένα του χρήστη ανακτώνται αυτόματα μέσω της κατάλληλης συνάρτησης από τη βάση δεδομένων, διασφαλίζοντας ότι όλες οι πληροφορίες που προβάλλονται αντικατοπτρίζουν την τρέχουσα κατάσταση του λογαριασμού του.

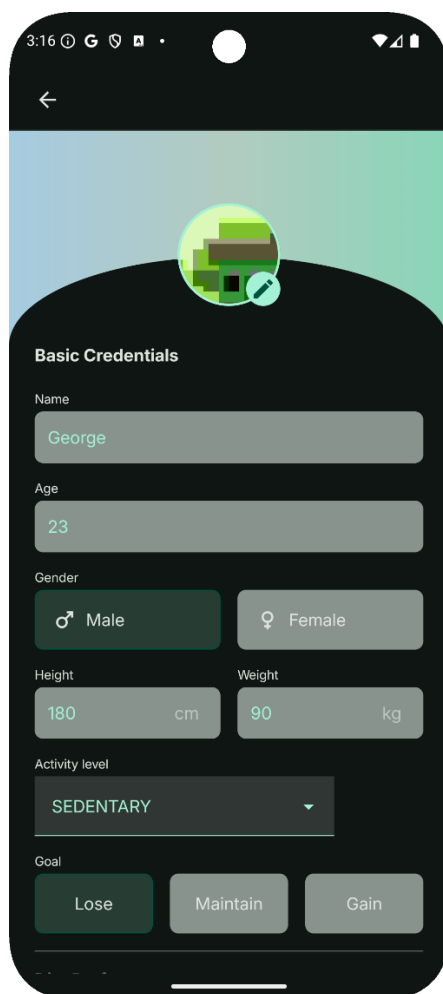


Σχήμα 5.23: Η οθόνη προφίλ χρήστη

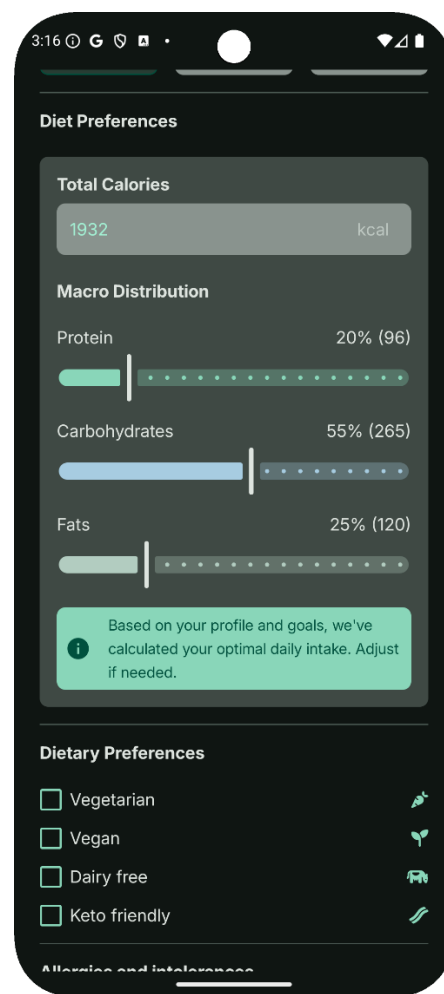
Στην κορυφή της οθόνης βρίσκεται η γραμμή εργαλείων με την ένδειξη «Profile», που υποδεικνύει την τρέχουσα τοποθεσία του χρήστη μέσα στην εφαρμογή, σε συνέπεια με το επιλεγμένο εικονίδιο του

navigation bar. Ακριβώς από κάτω εμφανίζεται η εικόνα προφίλ του χρήστη, συνοδευόμενη από το όνομα, τη διεύθυνση ηλεκτρονικού ταχυδρομείου και το τρέχον βάρος που είχε καταχωρηθεί κατά τη διαδικασία αρχικοποίησης του προφίλ.

Κάτω από τα βασικά στοιχεία του προφίλ υπάρχει το κουμπί «EDIT PROFILE», μέσω του οποίου ο χρήστης ανακατευθύνεται στην οθόνη επεξεργασίας προφίλ (Σχήμα 5.24, Σχήμα 5.25 και Σχήμα 5.26). Η συγκεκριμένη οθόνη συγκεντρώνει τα στοιχεία που είχαν καταχωρηθεί αρχικά στις δύο οθόνες αρχικοποίησης του προφίλ σε μία ενιαία και ανανεωμένη φόρμα. Όλα τα πεδία εμφανίζονται προσυμπληρωμένα σύμφωνα με τις προηγούμενες καταχωρήσεις του χρήστη και μπορούν να τροποποιηθούν. Στο κάτω μέρος της φόρμας υπάρχει το κουμπί «SAVE», το οποίο αποθηκεύει τις αλλαγές και ενημερώνει τα δεδομένα τόσο στη Firestore όσο και στη Room. Μετά την επιτυχή ολοκλήρωση της διαδικασίας, ο χρήστης ανακατευθύνεται αυτόματα στην αρχική οθόνη του προφίλ, ενώ εμφανίζεται ένα μήνυμα ειδοποίησης, συγκεκριμένα ένα toast, που επιβεβαιώνει την επιτυχή αποθήκευση των αλλαγών.



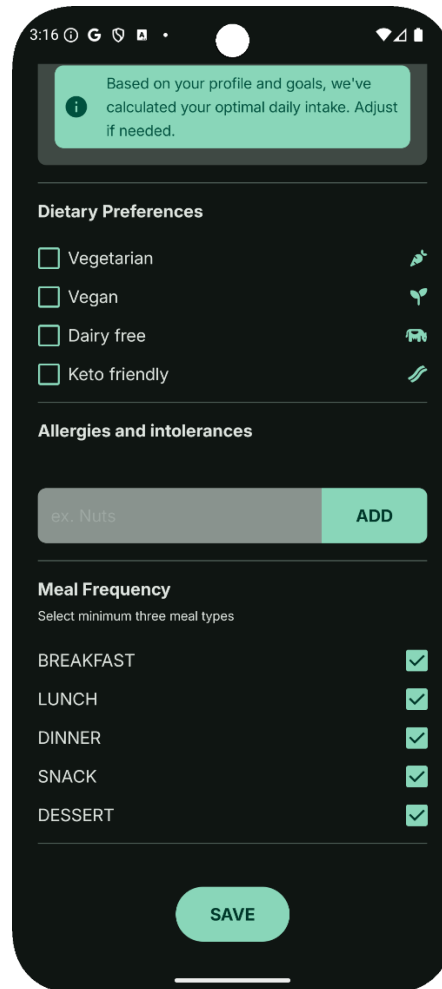
Σχήμα 5.24: Η κορυφή της οθόνης επεξεργασίας στοιχείων προφίλ



Σχήμα 5.25: Η μέση της οθόνης επεξεργασίας στοιχείων προφίλ

Αμέσως κάτω από το κουμπί επεξεργασίας εμφανίζονται τρεις επεκτάσιμες ενότητες, που οργανώνουν τις πληροφορίες του χρήστη σε θεματικές κατηγορίες: «BASIC CREDENTIALS», «DIET PREFERENCES» και «OTHER OPTIONS». Η ενότητα «BASIC CREDENTIALS» (Σχήμα 5.27), περιλαμβάνει τις βασικές παραμέτρους του προφίλ, όπως η ηλικία, το φύλο, το ύψος, το επίπεδο φυσικής δραστηριότητας και ο στόχος του χρήστη. Τα στοιχεία αυτά προβάλλονται σε μορφή λίστας

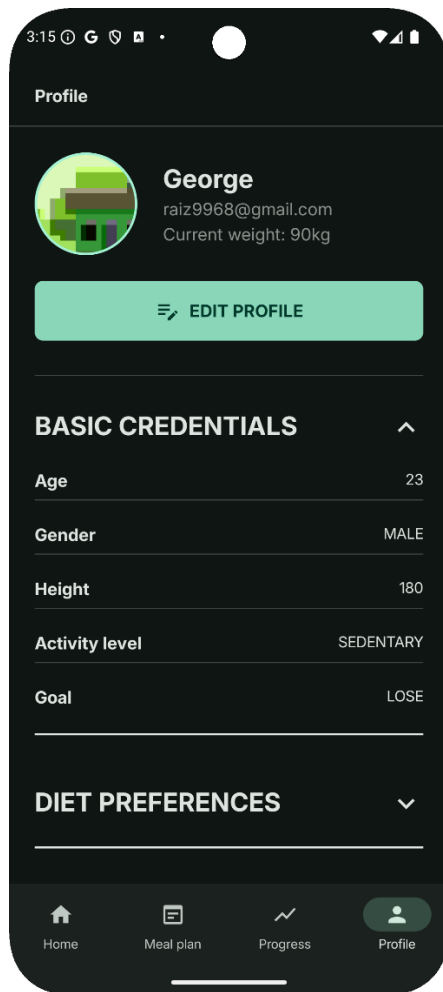
και αντιστοιχούν στις τιμές που έχει επιλέξει ο ίδιος ο χρήστης κατά την αρχικοποίηση ή μεταγενέστερη επεξεργασία του προφίλ του.



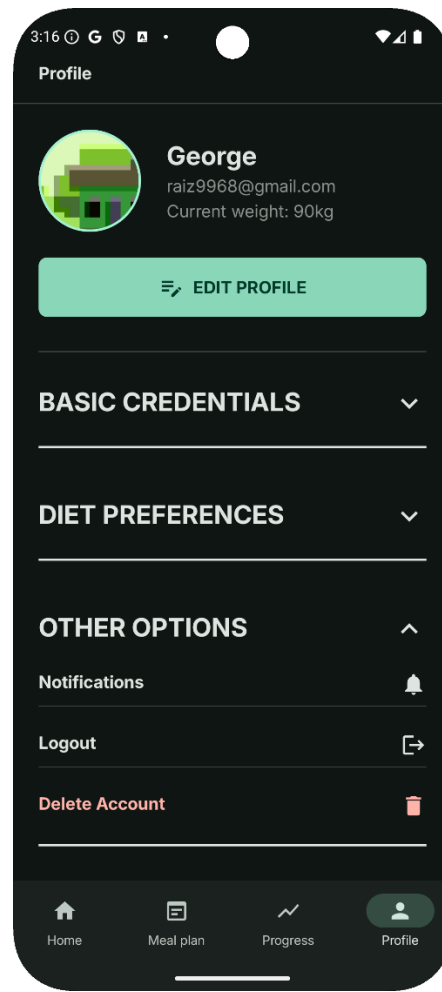
Σχήμα 5.26: Το κάτω μέρος της οθόνης επεξεργασίας στοιχείων προφίλ

Η ενότητα «DIET PREFERENCES» παρουσιάζει συγκεντρωτικά όλα τα στοιχεία που αφορούν τις διατροφικές προτιμήσεις και τους στόχους του χρήστη. Συγκεκριμένα, περιλαμβάνει την ημερήσια θερμιδική πρόσληψη-στόχο, την κατανομή των μακροθρεπτικών συστατικών (πρωτεΐνες, υδατάνθρακες, λιπαρά) εκφρασμένη σε ποσοστά, καθώς και τις επιλεγμένες διατροφικές κατηγορίες (π.χ. Vegan, Vegetarian, Keto, Dairy free). Σε περίπτωση που ο χρήστης δεν έχει δηλώσει κάποια ιδιαίτερη προτίμηση, εμφανίζεται η ένδειξη «NORMAL». Επίσης, προβάλλονται τυχόν αλλεργίες ή δυσανεξίες που έχει δηλώσει. Εάν δεν υπάρχουν, εμφανίζεται η ένδειξη «NONE». Τέλος, παρουσιάζεται η συχνότητα γευμάτων, δηλαδή οι τύποι γευμάτων που έχει επιλέξει ο χρήστης να περιλαμβάνονται καθημερινά (π.χ. Breakfast, Lunch, Dinner, Snack, Dessert).

Η τρίτη ενότητα, «OTHER OPTIONS» (Σχήμα 5.28), περιλαμβάνει επιλογές που σχετίζονται με τη διαχείριση του λογαριασμού και τις ειδοποιήσεις. Οι διαθέσιμες επιλογές είναι: «Notifications», «Logout» και «Delete Account». Η επιλογή «Notifications» οδηγεί τον χρήστη σε ξεχωριστή οθόνη ρύθμισης ειδοποιήσεων, στην οποία μπορεί να προσαρμόσει τις ώρες αποστολής ειδοποιήσεων για κάθε τύπο γεύματος (Σχήμα 5.29). Η οθόνη αυτή περιλαμβάνει μια γραμμή εργαλείων με κουμπί επιστροφής και ένδειξη τίτλου, καθώς και αναλυτικές οδηγίες για τη χρήση της λειτουργίας. Για κάθε τύπο γεύματος



Σχήμα 5.27: Η οθόνη προφίλ χρήστη με επεκταμένη την ενότητα «BASIC CREDENTIALS»



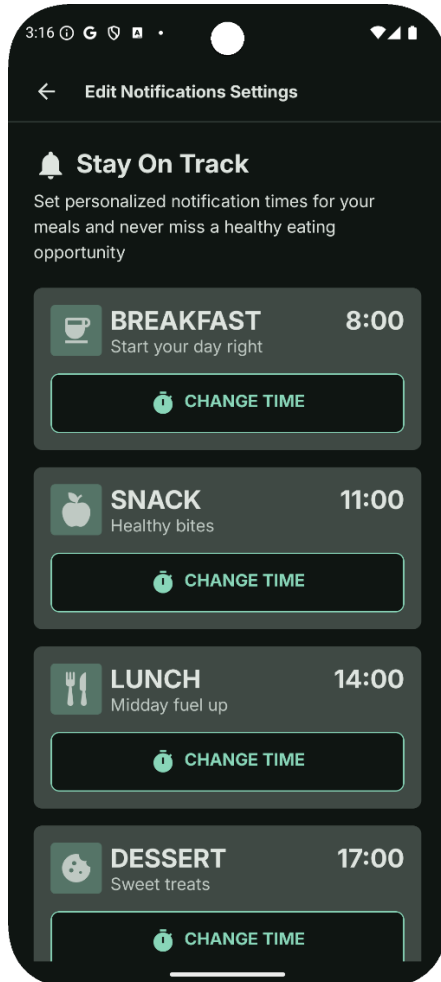
Σχήμα 5.28: Η οθόνη προφίλ χρήστη με επεκταμένη την ενότητα «OTHER OPTIONS»

εμφανίζεται ξεχωριστή κάρτα, η οποία περιλαμβάνει το όνομα του γεύματος, την τρέχουσα ώρα ειδοποίησης και το κουμπί «CHANGE TIME», μέσω του οποίου ο χρήστης μπορεί να τροποποιήσει την ώρα. Εάν ο χρήστης δεν έχει χορηγήσει δικαίωμα αποστολής ειδοποιήσεων στην εφαρμογή κατά την πρώτη εκτέλεση της εφαρμογής (στο homepage), το αντίστοιχο στοιχείο UI, αιτήματος άδειας εμφανίζεται και εδώ. Σε περίπτωση που ο χρήστης δεν αποδεχθεί την άδεια, οι λειτουργικότητες ρύθμισης ειδοποιήσεων δεν εμφανίζονται. Αντίθετα, εμφανίζεται μήνυμα προτροπής προς τον χρήστη, να δώσει άδεια χειροκίνητα μέσα από τις ρυθμίσεις του κινητού του.

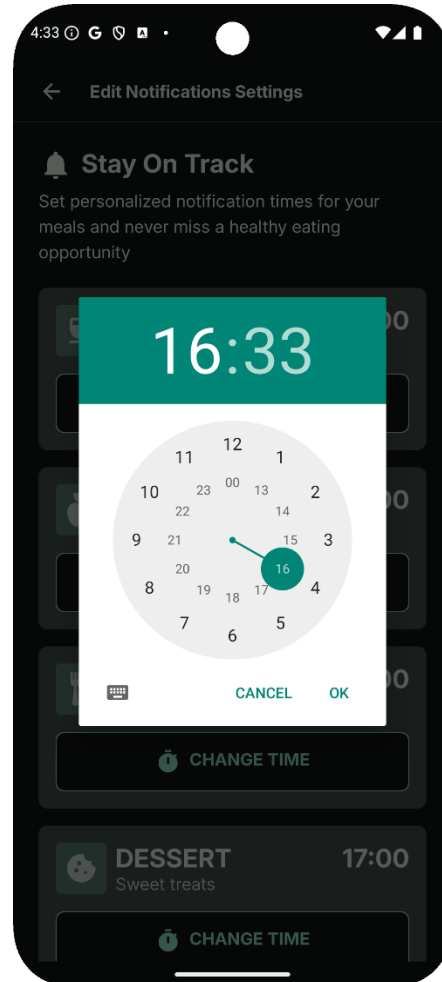
Η επιλογή αλλαγής ώρας ανοίγει το native time picker dialog του Android, το οποίο παρουσιάζεται με τη μορφή γραφικού ρολογιού, επιτρέποντας στον χρήστη να επιλέξει την επιθυμητή ώρα με απλό και οικείο τρόπο (Σχήμα 5.30). Για κάθε γεύμα υπάρχουν προεπιλεγμένες ώρες που έχουν καθοριστεί με βάση συνήθεις διατροφικές συνήθειες, οι οποίες μπορούν να τροποποιηθούν ανάλογα με τις ανάγκες του χρήστη. Οι ώρες αυτές αποθηκεύονται μόνο τοπικά, χωρίς συγχρονισμό με τη Firestore, επομένως σε περίπτωση απεγκατάστασης της εφαρμογής οι ρυθμίσεις αυτές χάνονται.

Η λειτουργία των ειδοποιήσεων βασίζεται στους μηχανισμούς AlarmManager και BroadcastReceiver του Android. Ο «AlarmManager» είναι ένα σύστημα χρονοπρογραμματισμού, που επιτρέπει σε μια εφαρμογή να εκτελεί ενέργειες σε καθορισμένο χρόνο, ακόμα κι αν η εφαρμογή δεν εκτελείται ενεργά στο προσκήνιο [86]. Μέσω αυτού, προγραμματίζονται οι ειδοποιήσεις για τα επιλεγμένα γεύματα, ώστε να ενεργοποιούνται τη συγκεκριμένη ώρα που έχει ορίσει ο χρήστης. Ο «BroadcastReceiver» από την

άλλη πλευρά, είναι ένας μηχανισμός που επιτρέπει στην εφαρμογή να λαμβάνει και να ανταποκρίνεται σε μηνύματα (intents) του συστήματος ή άλλων εφαρμογών [87]. Στην προκειμένη περίπτωση, ο BroadcastReceiver ενεργοποιείται από το AlarmManager τη στιγμή που έχει προγραμματιστεί η ειδοποίηση, δημιουργώντας και εμφανίζοντας την αντίστοιχη ειδοποίηση στο κινητό τηλέφωνο του χρήστη. Οι δύο αυτές τεχνολογίες λειτουργούν συμπληρωματικά, διασφαλίζοντας ότι οι ειδοποιήσεις θα αποστέλλονται ακόμα και αν η εφαρμογή βρίσκεται σε κατάσταση αδράνειας ή οθόνης κλειδώματος.



Σχήμα 5.29: Η οθόνη ρύθμισης της ώρας των ειδοποιήσεων



Σχήμα 5.30: Ο time picker dialog του Android στην εφαρμογή

Οι άλλες δύο επιλογές της ενότητας «OTHER OPTIONS» αφορούν τη διαχείριση λογαριασμού. Η επιλογή «Logout» αποσυνδέει τον χρήστη από την εφαρμογή, διαγράφοντας από τη Room όλα τα δεδομένα που σχετίζονται με τον λογαριασμό του, εκτός από τις ώρες ειδοποιήσεων και τυχόν δίαιτες που βρίσκονται σε εκκρεμότητα συγχρονισμού. Η επιλογή «Delete Account» επιτρέπει την πλήρη και οριστική διαγραφή του λογαριασμού του χρήστη τόσο από τη Firestore όσο και από τη συσκευή. Και στις δύο περιπτώσεις, πριν από την εκτέλεση της ενέργειας, εμφανίζεται αναδυόμενο μήνυμα επιβεβαίωσης που ζητά από τον χρήστη να επιβεβαιώσει την πρόθεσή του, προκειμένου να αποφευχθεί τυχόν ακούσια απώλεια δεδομένων.

5.9 Επίλογος

Το παρόν κεφάλαιο αποτυπώνει τη μετάβαση από τον σχεδιασμό στην υλοποίηση, παρουσιάζοντας την πρακτική ενσωμάτωση των επιμέρους συνιστωσών της εφαρμογής σε ένα λειτουργικό σύνολο. Μέσα

Κεφάλαιο 5

από τις υλοποιητικές επιλογές που αναπτύχθηκαν, καθίσταται σαφές πώς οι αρχιτεκτονικές αποφάσεις και οι αλγοριθμικές αρχές μεταφράζονται σε συγκεκριμένες προγραμματιστικές δομές και μηχανισμούς. Το κεφάλαιο αυτό αναδεικνύει τον τρόπο με τον οποίο η εφαρμογή υποστηρίζει την παραγωγή διατροφικών πλάνων, λαμβάνοντας υπόψη τις απαιτήσεις της βελτιστοποίησης πολλαπλών στόχων, χωρίς να αποσπάται από τον συνολικό σχεδιασμό του συστήματος. Στο επόμενο κεφάλαιο ακολουθεί η πειραματική αξιολόγηση της εφαρμογής και του εξελικτικού αλγορίθμου, όπου εξετάζεται η συμπεριφορά και η αποτελεσματικότητα της προτεινόμενης προσέγγισης μέσα από δοκιμές και αποτελέσματα.

Κεφάλαιο 6ο: Δοκιμές, αποτελέσματα και αξιολόγηση

6.1 Εισαγωγή

Η αξιολόγηση μιας εφαρμογής που συνδυάζει μηχανισμούς ανάκτησης δεδομένων, τοπικής αποθήκευσης, αλγοριθμικής βελτιστοποίησης και δυναμικής διαμόρφωσης περιεχομένου απαιτεί μια πολυεπίπεδη διαδικασία ελέγχου, η οποία δεν περιορίζεται στη βασική λειτουργικότητα του γραφικού περιβάλλοντος. Στο πλαίσιο της παρούσας ΔΕ, η διαδικασία δοκιμών είχε διττό χαρακτήρα. Αφενός επικεντρώθηκε στη διασφάλιση της αξιοπιστίας και συνέπειας της Android εφαρμογής σε πραγματικές συνθήκες χρήσης, αφετέρου περιέλαβε την πειραματική αξιολόγηση των παραμέτρων και της απόδοσης του εξελικτικού αλγορίθμου που αποτελεί τον πυρήνα της παραγωγής του διατροφικού πλάνου. Οι δύο αυτές πτυχές, αν και διακριτές ως προς τη μεθοδολογία σύγκρισής τους, συνδέονται άμεσα, καθώς η σταθερότητα της εφαρμογής αποτελεί προϋπόθεση για την ορθή λειτουργία του αλγοριθμικού μοντέλου και την ομαλή εμπειρία του χρήστη και το αντίστροφο.

Η αξιολόγηση της εφαρμογής πραγματοποιήθηκε μέσα από συστηματικές δοκιμές που βασίστηκαν σε πραγματικά σενάρια χρήσης, μεταβλητές συνθήκες δικτύου και στοχευμένες προσπάθειες πρόκλησης αστοχιών. Παρά το γεγονός ότι δεν εφαρμόστηκαν αυτοματοποιημένα unit ή UI tests, οι δοκιμές κάλυψαν ολόκληρο το φάσμα της λειτουργικότητας, από την πλοήγηση μεταξύ οθονών και την ορθότητα της απεικόνισης του UI, έως τον χειρισμό σφαλμάτων, τις διαδικασίες συγχρονισμού, τη συμπεριφορά της εφαρμογής σε ακραίες εισροές και τη διαχείριση ταυτοχρονισμού σε περιβάλλον πολλαπλών coroutines. Αντίστοιχα, για τον εξελικτικό αλγόριθμο πραγματοποιήθηκε εκτενής διερεύνηση των παραμέτρων του και αξιολόγηση της απόδοσής του με βάση μετρικές που αφορούν τόσο την θερμιδική κατανομή όσο και την πληρότητα των θρεπτικών στόχων.

Στόχος του παρόντος κεφαλαίου είναι να παρουσιάσει με τεκμηριωμένο τρόπο τα αποτελέσματα των παραπάνω διαδικασιών, αναδεικνύοντας τόσο τα ισχυρά σημεία της υλοποίησης όσο και τις προκλήσεις που αναδείχθηκαν κατά τη διάρκεια της ανάπτυξης. Αρχικά, περιγράφονται οι δοκιμές λειτουργικότητας της εφαρμογής και στη συνέχεια εξετάζονται τα σενάρια οριακών περιπτώσεων και οι μηχανισμοί αντιμετώπισης σφαλμάτων. Ακολουθεί η αξιολόγηση της απόδοσης του εξελικτικού αλγορίθμου, όπου παρουσιάζονται συγκριτικά αποτελέσματα διαφορετικών διαμορφώσεων και παράγονται συμπεράσματα για την αποτελεσματικότητα της τελικής επιλογής παραμέτρων. Τέλος, συνοψίζονται οι περιορισμοί της υλοποίησης, οι οποίοι συνιστούν χρήσιμες ενδείξεις για μελλοντική βελτίωση και επέκταση της εφαρμογής.

6.2 Δοκιμές λειτουργικότητας και αξιοπιστίας της εφαρμογής

Η αξιολόγηση της λειτουργικότητας και της αξιοπιστίας της εφαρμογής πραγματοποιήθηκε μέσα από εκτεταμένο manual testing, το οποίο συνοδεύει συχνά την ανάπτυξη πρωτοτύπων ή συστημάτων με έντονα αλληλεπιδραστικό χαρακτήρα. Η διαδικασία αυτή βασίστηκε στη συστηματική αναπαραγωγή ρεαλιστικών σεναρίων χρήσης, στη διερεύνηση πιθανών αστοχιών και στη διαρκή παρατήρηση της συμπεριφοράς του UI και της υποκείμενης λογικής.

Κεντρικό στοιχείο της διαδικασίας ήταν η επαλήθευση της ορθής απεικόνισης του UI σε διαφορετικές συσκευές, ώστε να εξασφαλίζεται η προσαρμοστικότητα της εφαρμογής σε ποικιλία αναλύσεων οθόνης. Η αξιολόγηση επικεντρώθηκε τόσο στην αισθητική συνέπεια όσο και στην ορθή αντανάκλαση της λογικής της εφαρμογής στις UI καταστάσεις. Για παράδειγμα, ελέγχθηκε ότι αλλαγές σε εσωτερικά δεδομένα οδηγούσαν σε αντίστοιχες ενημερώσεις των ορατών στοιχείων, ότι τα μηνύματα και τα

χρώματα ενημέρωσης εμφανίζονταν στον σωστό χρόνο και ότι τα δεδομένα που λαμβάνονταν από απομακρυσμένες πηγές απεικονίζονταν με τη σωστή μορφή, όπως συμβαίνει με τιμές ημερομηνιών, ποσοότητες ή μακροθρεπτικά στοιχεία. Αντίστοιχα, έγινε συστηματικός έλεγχος του μηχανισμού πλοήγησης, με έμφαση στη διαχείριση του back navigation. Εξετάστηκε ότι η επιστροφή του χρήστη δεν οδηγεί σε ανεπιθύμητες ή λανθασμένες οθόνες, και ότι σε συγκεκριμένα σημεία, όπως κατά την αρχική παραγωγή του διατροφικού πλάνου, η επιστροφή διακόπτει ορθά τη ροή αντί να επαναφέρει τον χρήστη σε μη έγκυρες καταστάσεις.

Η αξιοπιστία του συστήματος εξετάστηκε και μέσα από τη στοχευμένη προσομοίωση αποτυχιών στο Firebase Firestore και στο USDA API. Με τεχνητή παραγωγή σφαλμάτων, ελέγχθηκε εάν οι συναρτήσεις που επικοινωνούν με απομακρυσμένους πόρους χειρίζονται κατάλληλα τις εξαιρέσεις και εάν το UI αποτυπώνει με σαφήνεια τις αντίστοιχες καταστάσεις, για παράδειγμα μέσω αναδυόμενων μηνυμάτων ή ειδικών ενδείξεων σφάλματος. Επιπλέον, δοκιμάστηκαν εκτενώς τα σενάρια μη ύπαρξης σύνδεσης στο διαδίκτυο, τόσο σε ροές που την απαιτούν για την επικοινωνία με το API όσο και σε αντίθετες περιπτώσεις, όπως η αποθήκευση διατροφικών δεδομένων με μηχανισμούς συγχρονισμού.

Κομβικό εργαλείο στην παρακολούθηση της συμπεριφοράς της εφαρμογής αποτέλεσαν κάποια εργαλεία του Android Studio, το Logcat, το Debugger και το App Inspection. Το Logcat χρησιμοποιήθηκε για την καταγραφή και ανάγνωση μηνυμάτων εκτέλεσης, επιτρέποντας την ταυτοποίηση της ακριβούς προέλευσης σφαλμάτων και εξαιρέσεων. Ο Debugger διευκόλυνε την εκτέλεση του κώδικα βήμα προς βήμα, προσφέροντας τη δυνατότητα δυναμικής παρακολούθησης των μεταβλητών και της ροής ελέγχου. Το App Inspection χρησιμοποιήθηκε για την εξέταση της τοπικής βάσης δεδομένων και των pending εργασιών συγχρονισμού, επιβεβαιώνοντας ότι οι εγγραφές προστίθενται σωστά στη Room και ότι η επαναφορά της σύνδεσης στον χρήστη οδηγεί στον αναμενόμενο συγχρονισμό με τη Firestore.

Παράλληλα, ιδιαίτερη έμφαση δόθηκε στη διερεύνηση της συμπεριφοράς του συστήματος ως προς τη διαχείριση concurrency, δεδομένου ότι η εφαρμογή εκτελεί ταυτόχρονα πολλαπλές διεργασίες, όπως δικτυακά αιτήματα, αλληλεπιδράσεις με τη βάση δεδομένων και ενημερώσεις του UI. Στο πλαίσιο αυτό εξετάστηκε η ορθή χρήση των Kotlin coroutines, οι οποίες αποτελούν έναν μηχανισμό ασύγχρονου προγραμματισμού που επιτρέπει τη δέσμευση ελαφρών «νημάτων» εκτέλεσης, προσφέροντας παράλληλη επεξεργασία χωρίς το κόστος των κλασικών thread [88]. Η χρήση τους καθιστά δυνατή την ταυτόχρονη εκτέλεση εργασιών με ελεγχόμενο τρόπο, περιορίζοντας τον κίνδυνο καθυστερήσεων ή αποκρίσεων που μπλοκάρουν την εμπειρία χρήστη. Συναφώς, αξιολογήθηκε και η χρήση των Dispatchers, οι οποίοι καθορίζουν το κατάλληλο execution context για κάθε coroutine, σύμφωνα με τις συστάσεις της επίσημης τεκμηρίωσης της Kotlin [89]. Ο «Dispatchers.Default» επιλέχθηκε για την εκτέλεση του εξελικτικού αλγορίθμου, καθώς αποτελεί το προτεινόμενο πλαίσιο για εργασίες που είναι εντατικές σε επεξεργαστική ισχύ. Για λειτουργίες που αφορούν τη Room και άλλες διεργασίες εισόδου/εξόδου χρησιμοποιήθηκε ο «Dispatchers.IO», ενώ οι ενημερώσεις της διεπαφής χρήστη εκτελέστηκαν αποκλειστικά μέσω του «Dispatchers.Main», ώστε οι σχετικές αλλαγές να πραγματοποιούνται στο κύριο νήμα της εφαρμογής. Η ορθή χρήση των παραπάνω μηχανισμών επιβεβαιώθηκε δοκιμαστικά, διασφαλίζοντας την αποδοτική κατανομή φόρτου και την απρόσκοπτη λειτουργία του συστήματος.

Τέλος, σε όλη τη διάρκεια της ανάπτυξης εντοπίστηκαν και διορθώθηκαν πολυάριθμες ανεπιθύμητες συμπεριφορές, οι οποίες αφορούσαν τόσο την οπτική απεικόνιση όσο και την επιχειρησιακή λογική. Η σταδιακή επίλυσή τους, σε συνδυασμό με τη συνεχή παρατήρηση μέσω των εργαλείων ανάπτυξης,

συνέβαλε στη σταθεροποίηση της εφαρμογής και στη διασφάλιση ότι η τελική εμπειρία χρήσης ανταποκρίνεται στις απαιτήσεις της μελέτης.

6.3 Σενάρια οριακών περιπτώσεων και μηχανισμοί αποτυχίας

Η εξέταση οριακών περιπτώσεων αποτελεί κρίσιμο στοιχείο για την αξιολόγηση της ανθεκτικότητας μιας εφαρμογής, καθώς αναδεικνύει συμπεριφορές που ενδέχεται να εμφανιστούν μόνο υπό ακραίες ή μη συνηθισμένες συνθήκες. Στην παρούσα εργασία, τα σενάρια αυτά διερευνήθηκαν συστηματικά τόσο σε επίπεδο UI και αλληλεπίδρασης του χρήστη με την εφαρμογή, όσο και σε επίπεδο υποδομής, όπου εμπλέκονται δικτυακές κλήσεις, διαδικασίες συγχρονισμού και διαχείριση δεδομένων.

Ένα από τα σημαντικότερα σύνολα δοκιμών αφορούσε τη συμπεριφορά της εφαρμογής υπό ασταθή ή μηδενική σύνδεση στο διαδίκτυο. Σενάρια χαμηλής ταχύτητας, πλήρους απώλειας σύνδεσης ή απότομης διακοπής κατά τη διάρκεια αιτήματος προς το USDA API χρησιμοποιήθηκαν για να επιβεβαιωθεί ότι η εφαρμογή αντιδρά με προβλέψιμο και ενημερωτικό τρόπο, χωρίς να καταρρέει ή να παραμένει σε ανεπιθύμητες καταστάσεις. Σε ροές όπου η ύπαρξη δικτύου είναι αναγκαία, όπως η ανάκτηση τροφίμων από το USDA API, η εφαρμογή εμφανίζει κατάλληλα μηνύματα σφάλματος και επιτρέπει στον χρήστη να επαναλάβει τη διαδικασία όταν η σύνδεση αποκατασταθεί. Αντιθέτως, σε ροές που υποστηρίζουν offline-first λειτουργία, όπως η αποθήκευση ή η επεξεργασία διατροφών, η εφαρμογή αξιοποιεί τον μηχανισμό συγχρονισμού που περιεγράφηκε στο προηγούμενο κεφάλαιο, επιτρέποντας την τοπική καταγραφή ενεργειών και τον αυτόματο συγχρονισμό τους όταν το δίκτυο γίνει και πάλι διαθέσιμο.

Η συμπεριφορά του συστήματος σε συνθήκες «offline» χρήσης διερευνήθηκε διεξοδικά μέσω των εργαλείων του Android Studio, ιδίως του App Inspection, όπου παρατηρήθηκε η δημιουργία και αποθήκευση εγγραφών «pending sync» στη Room. Ελέγχθηκε ότι οι εγγραφές αυτές παραμένουν απομονωμένες ανά χρήστη, ακόμη και μετά από αποσύνδεση και σύνδεση με διαφορετικό λογαριασμό, ώστε να αποφευχθεί ο συγχρονισμός δεδομένων μεταξύ χρηστών. Επιβεβαιώθηκε επίσης ότι η επαναφορά της σύνδεσης ενεργοποιεί τον αντίστοιχο WorkManager, ο οποίος μεταφέρει τις εκκρεμείς εγγραφές στη Firestore με ασφάλεια.

Σημαντικό μέρος των οριακών δοκιμών αφορούσε και τη συμπεριφορά της εφαρμογής υπό αυξημένο φόρτο εργασίας ή σε συσκευές με περιορισμένη επεξεργαστική ισχύ. Δοκιμάστηκαν επίσης καταστάσεις όπου ο χρήστης επιχειρεί να εκτελέσει γρήγορα διαδοχικές ενέργειες ή να εκτελέσει κάποια λειτουργία όσο η οθόνη στην οποία προσπαθεί να ενεργήσει, βρίσκεται σε κατάσταση φόρτωσης των δεδομένων. Σε όλες τις περιπτώσεις ελέγχθηκε ότι η εφαρμογή αποτρέπει την εκτέλεση αλληλεπικαλυπτόμενων ενεργειών, εμφανίζει κατάλληλες ενδείξεις φόρτωσης και συνεχίζει να λειτουργεί ομαλά χωρίς να προκαλείται απώλεια δεδομένων ή ασταθής συμπεριφορά.

Εξετάστηκαν επίσης σενάρια ακραίων ή λανθασμένων εισροών, όπως η προσπάθεια εισαγωγής μη αριθμητικών χαρακτήρων σε αριθμητικά πεδία. Στόχος ήταν να διασφαλιστεί ότι η εφαρμογή αναγνωρίζει έγκαιρα τις μη έγκυρες τιμές, αποκλείει την πρόοδο της ροής όταν αυτό κρίνεται απαραίτητο και ενημερώνει τον χρήστη με σαφή τρόπο.

Σημαντική πτυχή των δοκιμών αποτέλεσε και η αξιολόγηση του Paging 3 σε πραγματικές συνθήκες περιήγησης. Δεδομένου ότι η εφαρμογή ανακτά μεγάλο όγκο δεδομένων από το USDA API, ελέγχθηκε η ομαλότητα της κύλισης σε μεγάλες λίστες και η σταδιακή φόρτωση των δεδομένων. Τέλος, διερευνήθηκε και η συμπεριφορά του συστήματος σε σχέση με περιορισμούς που τίθενται από τον χρήστη, όπως η δήλωση αλλεργιών ή η ενεργοποίηση συγκεκριμένων διατροφικών προτιμήσεων. Σε περιπτώσεις όπου οι περιορισμοί ήταν ιδιαίτερα αυστηροί, για παράδειγμα όταν συνδυάζονταν πολλές

αλλεργίες, ελέγχθηκε ότι η εφαρμογή χειρίζεται ορθά τη δραστική μείωση του διαθέσιμου συνόλου τροφών, χωρίς να οδηγεί τον εξελικτικό αλγόριθμο σε ανεπιθύμητες καταστάσεις. Ωστόσο, λόγω των προβλημάτων που εισάγουν οι λέξεις-κλειδιά στις οποίες βασίζεται αυτή η λογική, υπήρξαν και περιπτώσεις που το παραγόμενο διατροφικό πλάνο περιλάμβανε τροφές που δεν θα έπρεπε.

6.4 Δοκιμές λειτουργικότητας και αξιολόγηση εξελικτικού αλγορίθμου

Παρά το γεγονός ότι οι υλοποιήσεις του EA και της Android εφαρμογής συνυπάρχουν στο ίδιο έργο και σχετίζονται άμεσα μεταξύ τους, κρίθηκε αναγκαίο, οι δοκιμές λειτουργικότητας και η αξιολόγηση του πρώτου, να αναλυθούν σε ξεχωριστή υποενότητα, καθώς ακολουθήθηκε μια διαφορετική διαδικασία ελέγχου. Προτού οριστικοποιηθεί η υλοποίηση του EA, που παρουσιάστηκε στο προηγούμενο κεφάλαιο, έγιναν αρκετές αναθεωρήσεις. Η αρχική υλοποίηση του εξελικτικού αλγορίθμου εμφάνισε σύντομα ενδείξεις ανεπαρκούς συμπεριφοράς. Παρά την αύξηση του αριθμού των γενεών και την τροποποίηση των παραμέτρων των γενετικών τελεστών, τα παραγόμενα πλάνα διατροφής παρουσίαζαν σημαντικές αποκλίσεις ως προς τους στόχους θερμίδων, μακροθρεπτικών και, κυρίως, μικροθρεπτικών συστατικών. Η έλλειψη βελτίωσης οδήγησε σε μια συστηματική επανεξέταση της κωδικοποίησης των λύσεων, των τελεστών αναπαραγωγής και μετάλλαξης, καθώς και της χρήσης του NSGA-II στο πλαίσιο ενός προβλήματος πολλαπλών στόχων.

6.4.1 Εξέλιξη του εξελικτικού αλγορίθμου από την αρχική στην τελική μορφή του

Στην αρχική εκδοχή, η κωδικοποίηση του γονότυπου βασιζόταν σε μια επαναλαμβανόμενη δομή πολλαπλών ζευγών χρωμοσωμάτων ανά τύπο γεύματος. Συγκεκριμένα, για κάθε τύπο γεύματος που επέλεγε ο χρήστης (BREAKFAST, LUNCH, DINNER κλπ.), δημιουργούνταν μια λίστα κατάλληλων τροφίμων, και στη συνέχεια ο γονότυπος κωδικοποιούσε μέχρι τέσσερα πιθανά τρόφιμα ανά τύπο γεύματος (slots). Για κάθε ένα από αυτά τα τέσσερα πιθανά τρόφιμα δημιουργούνταν ένα ζεύγος χρωμοσωμάτων που κωδικοποιούσε τον δείκτη του τροφίμου και την ποσότητά του. Η χρήση της μορφής «IntegerChromosome.of(min, max)» για τη δημιουργία των χρωμοσωμάτων χωρίς ορισμό μήκους, συνεπάγεται ότι κάθε χρωμόσωμα αποτελείται από ένα και μόνο γονίδιο. Ο μηχανισμός αυτός, σε συνδυασμό με την επανάληψη που εκτελούνταν τέσσερις φορές, προκειμένου να επιτευχθεί η παραπάνω λειτουργικότητα και να παρέχεται στον αλγόριθμο η ευελιξία να επιλέξει από κανένα έως τέσσερα τρόφιμα για κάθε τύπο γεύματος, ανάλογα με τις ανάγκες της βελτιστοποίησης, παρήγαγε έναν ιδιαίτερα μεγάλο αλλά αραιό χώρο αναζήτησης. Συγκεκριμένα, για κάθε τύπο γεύματος δημιουργούνταν τέσσερα ζεύγη χρωμοσωμάτων, συνολικά δηλαδή οκτώ μονογονιδιακά χρωμοσώματα ανά τύπο γεύματος. Επομένως, για ένα διατροφικό πλάνο με τρία γεύματα, ο γονότυπος αποτελούνταν από 24 μονογονιδιακά χρωμοσώματα. Η αναπαράσταση αυτή δεν αξιοποιούσε καμία δομική πληροφορία για το πρόβλημα, όπως το γεγονός ότι κάθε γεύμα τείνει να περιλαμβάνει έναν μικρό αριθμό βασικών τροφών, ενώ η δυνατότητα επιλογής τεσσάρων τροφίμων ανά γεύμα οδηγούσε συχνά σε αραιές λύσεις όπου πολλά από τα slots παρέμειναν κενά, με δείκτη -1, που υποδήλωνε την απουσία τροφίμου στη συγκεκριμένη θέση ή μηδενική ποσότητα. Ως αποτέλεσμα, ο εκτεταμένος χώρος αποφάσεων καθιστούσε δύσκολη την αποτελεσματική εξερεύνηση συνδυασμών που ικανοποιούν ταυτόχρονα τους διατροφικούς στόχους.

Οι αδυναμίες της αρχικής προσέγγισης αναδείχθηκαν ακόμη περισσότερο κατά την εξέταση των γενετικών τελεστών. Στην αρχική υλοποίηση χρησιμοποιήθηκαν ο SinglePointCrossover και ο SwapMutator σε έναν γονότυπο αποτελούμενο αποκλειστικά από χρωμοσώματα μήκους ενός γονιδίου. Ο SinglePointCrossover, απαιτεί χρωμοσώματα πολλαπλών γονιδίων ώστε η αναδιάταξη τμημάτων τους να έχει ουσιαστική επίδραση στην κληρονομικότητα. Όταν εφαρμοστεί σε χρωμοσώματα μήκους

ενός γονιδίου, ο μηχανισμός αυτός εκφυλίζεται σε απλή ανταλλαγή ολόκληρων χρωμοσωμάτων μεταξύ των γονέων, χωρίς δυνατότητα ουσιαστικού ανακατέματος γενετικής πληροφορίας εντός του χρωμοσώματος. Η θεμελιώδης ιδέα του crossover, η οποία είναι η δημιουργία νέων συνδυασμών γονιδίων από υπάρχουσες καλές λύσεις, χάνεται εντελώς. Κατ' αναλογία, ο SwapMutator, ο οποίος έχει νόημα μόνο σε χρωμοσώματα με πολλαπλές θέσεις, δεν μπορεί να επέμβει σε χρωμοσώματα μήκους ενός, αφού δεν υπάρχουν δύο γονίδια προς ανταλλαγή. Σύμφωνα με την τεκμηρίωση της βιβλιοθήκης Jenetics, ο SwapMutator ανταλλάσσει τις θέσεις δύο γονιδίων εντός του ίδιου χρωμοσώματος, μια λειτουργία που καθίσταται αδύνατη όταν το χρωμόσωμα περιέχει μόνο μία θέση. Ως συνέπεια, η αρχική υλοποίηση στερούνταν πρακτικά οποιουδήποτε μηχανισμού μετάλλαξης ικανού να τροποποιήσει τις τιμές των γονιδίων, κάτι που είναι απαραίτητο για την αποφυγή πρόωρης σύγκλισης και για τη σταδιακή βελτίωση των λύσεων γύρω από τοπικά βέλτιστα.

Επιπλέον, η εξέταση της συνολικής διαδικασίας επιλογής ανέδειξε μια ουσιώδη ασυμβατότητα στην αρχική εκδοχή. Παρά τη χρήση του NSGA-II, ο οποίος λειτουργεί με μη κυριαρχούμενη ταξινόμηση και crowding distance ώστε να διατηρεί σύνολα λύσεων στο μέτωπο Pareto, η τελική επιλογή της καλύτερης λύσης γινόταν με βάση μόνο το πρώτο objective, τη θερμидική απόκλιση. Συγκεκριμένα, η αρχική υλοποίηση χρησιμοποιούσε τη μέθοδο «minWithOrNull» με κριτήριο σύγκρισης το «fitness().data()[0]», επιλέγοντας δηλαδή τη λύση με τη μικρότερη απόκλιση θερμίδων, ανεξάρτητα από την επίδοσή της στα άλλα objectives. Η πρακτική αυτή εξουδετέρωνε ουσιαστικά τον multi-objective χαρακτήρα του αλγορίθμου, καθώς οδηγούσε στην απόρριψη λύσεων με εξαιρετική επίδοση στα μακροθρεπτικά ή μικροθρεπτικά συστατικά, οι οποίες υστερούσαν οριακά μόνο ως προς τις θερμίδες.

Τα παραπάνω προβλήματα εντοπίστηκαν σταδιακά. Αρχικά, η πρακτική αξιολόγηση των παραγόμενων πλάνων ανέδειξε σημαντικές αποκλίσεις από τους διατροφικούς στόχους, ιδιαίτερα στα μικροθρεπτικά συστατικά, παρά τις επανειλημμένες προσπάθειες βελτίωσης μέσω της ρύθμισης των παραμέτρων των γενετικών τελεστών. Στη συνέχεια, η επανεξέταση της θεωρίας των MOEA και η προσεκτική ανάλυση του κώδικα οδήγησαν στη διαπίστωση ότι η επιλογή μιας μόνο λύσης βάσει ενός μόνο objective ακυρώνει την ουσία του NSGA-II.

Έτσι, στην τελική μορφή υλοποίησης του EA, που παρουσιάστηκε στο Κεφάλαιο 5, επανασχεδιάστηκε η δομή του γονότυπου, με στόχο την εισαγωγή στρατηγικού περιορισμού και σαφέστερης οργάνωσης. Η νέα αυτή κωδικοποίηση περιορίζει τον χώρο αναζήτησης σε δομές πλάνων με πιο ρεαλιστικό χαρακτήρα, επιτρέποντας στον αλγόριθμο να επικεντρωθεί στην προσαρμογή των ποσοτήτων και του συνδυασμού τροφών γύρω από έναν σταθερό σκελετό, αντί να αναζητά λύσεις σε έναν υπερβολικά υψηλών διαστάσεων και αποδιοργανωμένο χώρο. Σχετικά με τη χρήση των γενετικών τελεστών, η διαφοροποίηση εφαρμογής αυτών στην τελική μορφή, μέσω των PartialAlters αντικατοπτρίζει τη διαφορετική φύση των αποφάσεων που αντιπροσωπεύουν τα δύο είδη γονιδίων, δηλαδή index τροφής και ποσότητα. Όσον αφορά τα προβλήματα που εισήγαγε στην αρχική έκδοση του EA η συνολική διαδικασία επιλογής, η τελική έκδοση τα επιλύει με τη διαδικασία προσδιορισμού των μη κυριαρχούμενων λύσεων του τελικού πληθυσμού και τον υπολογισμό της μικρότερης ευκλείδειας απόστασης κάθε λύσης από το ιδεατό σημείο μηδέν στον κανονικοποιημένο διανυσματικό χώρο των στόχων.

Στο σημείο αυτό είναι καλό να γίνουν κάποιες διευκρινήσεις σχετικά με τις τελικές επιλογές των γενετικών τελεστών και του selector. Η επιλογή να μην χρησιμοποιηθούν κάποιοι από τους crossover τελεστές της αρχικής υλοποίησης ή άλλοι διαδοσμένοι τελεστές τέτοιου είδους, όπως ο TwoPointCrossover ή ο MultiPointCrossover, βασίστηκε στη δομή του προβλήματος. Οι τελεστές

πολλαπλών σημείων τομής προϋποθέτουν ότι η θέση των γονιδίων εντός του χρωμοσώματος έχει σημασία και ότι υπάρχει κάποια συνάφεια μεταξύ γειτονικών γονιδίων. Ομοίως, απορρίφθηκε ο SwapMutator και δεν εξετάστηκαν τελεστές μετάλλαξης όπως ο MeanAlterer, ο οποίος αντικαθιστά ένα γονίδιο με τη μέση τιμή των γειτονικών του, καθώς η έννοια της γειννίας δεν έχει εφαρμογή στο συγκεκριμένο πρόβλημα. Ο LinearScaleAlterer, που κλιμακώνει τις τιμές των γονιδίων με βάση κάποιον παράγοντα, απορρίφθηκε επίσης, διότι η κλιμάκωση όλων των ποσοτήτων με τον ίδιο συντελεστή δεν αντιστοιχεί σε λογική διατροφική παρέμβαση.

Αναφορικά με την επιλογή του NSGA-II έναντι άλλων μηχανισμών επιλογής, η απόφαση αυτή βασίστηκε στον multi-objective χαρακτήρα του προβλήματος και στην επιθυμία εξέτασης μιας σύγχρονης προσέγγισης. Ο TournamentSelector και ο RouletteWheelSelector είναι μηχανισμοί σχεδιασμένοι για προβλήματα ενός στόχου, όπου υπάρχει μία μόνο τιμή καταλληλότητας ανά άτομο. Η εφαρμογή τους σε προβλήματα πολλαπλών στόχων θα απαιτούσε την αυθαίρετη συνένωση όλων αυτών των στόχων σε μία ενιαία μετρική, είτε μέσω σταθμισμένου αθροίσματος είτε μέσω κάποιας άλλης συναρτησιακής σύνθεσης. Μια τέτοια προσέγγιση θα εισήγε διάφορα προβλήματα, όπως το ότι θα οδηγούσε σε απώλεια πληροφορίας σχετικά με τη σχέση μεταξύ των στόχων και θα εμπόδιζε τον εντοπισμό διαφορετικών συμβιβασμών στο χώρο λύσεων. Ο NSGA-II, αντιθέτως, διατηρεί ένα σύνολο λύσεων που αντιπροσωπεύουν διαφορετικές ισορροπίες μεταξύ των στόχων, επιτρέποντας την τελική επιλογή με βάση γεωμετρικά κριτήρια στον χώρο των objectives. Επιπλέον, το ενδιαφέρον για τη μελέτη και εφαρμογή μιας multi-objective μεθοδολογίας, σε αντιδιαστολή με τις κλασικότερες προσεγγίσεις, συνέβαλε στην επιλογή του NSGA-II ως του πλέον κατάλληλου μηχανισμού για την παρούσα εργασία.

6.4.2 Ρύθμιση των παραμέτρων και αξιολόγηση του εξελικτικού αλγορίθμου

Αφού παρουσιάστηκε η εξέλιξη της υλοποίησης του εξελικτικού αλγορίθμου, από την αρχική του μορφή μέχρι την τελική αρχιτεκτονική που περιγράφεται στο Κεφάλαιο 5, το ενδιαφέρον σε αυτό το σημείο του κειμένου, μετατοπίζεται στη ρύθμιση των παραμέτρων και στον τρόπο με τον οποίο αξιολογήθηκαν συστηματικά οι διαφορετικές παραμετροποιήσεις. Η αξιολόγηση της απόδοσης ενός EA είναι πολύ πιο εύκολα αντιληπτή όταν αποτυπώνεται σε διαγράμματα σύγκλισης, διαγράμματα Pareto και συγκεντρωτικές απεικονίσεις των στόχων. Ωστόσο, στο περιβάλλον Android και ειδικότερα μέσα από το Android Studio, δεν είναι πρακτικό να υλοποιηθούν απευθείας scripts αντίστοιχα με αυτά της Python για την παραγωγή τέτοιων διαγραμμάτων. Για τον λόγο αυτό, επιλέχθηκε μια διαφορετική προσέγγιση, όπου ο αλγόριθμος καταγράφει τις απαραίτητες μετρήσεις σε αρχείο καταγραφής και η ανάλυση, μαζί με την παραγωγή των διαγραμμάτων, μεταφέρεται σε εξωτερικό script Python.

Η παραμετροποίηση του εξελικτικού αλγορίθμου επικεντρώθηκε σε ένα σύνολο βασικών παραμέτρων, οι οποίες, επηρεάζουν άμεσα τη συμπεριφορά της αναζήτησης και την ποιότητα των τελικών λύσεων. Συγκεκριμένα, εξετάστηκαν:

- το μέγεθος του πληθυσμού,
- ο μέγιστος αριθμός γενεών,
- το ποσοστό απογόνων σε κάθε γενιά,
- η πιθανότητα crossover ως προς τους δείκτες επιλογής τροφίμων,
- η πιθανότητα crossover ως προς τις ποσότητες,
- η πιθανότητα μετάλλαξης και
- η πιθανότητα εφαρμογής της γκαουσιανής μετάλλαξης.

Κάθε διαφορετικός συνδυασμός τιμών των παραπάνω παραμέτρων ορίζεται ως ένα ξεχωριστό πείραμα. Για κάθε τέτοιο πείραμα, ο αλγόριθμος εκτελέστηκε επαναληπτικά 50 φορές με τις ίδιες ακριβώς τιμές παραμέτρων. Καθώς οι EA είναι στοχαστικοί, μια μεμονωμένη εκτέλεση μπορεί να είναι ιδιαίτερα

ευνοϊκή ή δυσμενής λόγω τυχαίων επιλογών. Με τις πολλαπλές επαναλήψεις επιδιώκεται να εξαλειφθεί όσο γίνεται η επίδραση της τυχειότητας και να προκύψουν δείκτες απόδοσης που να αντικατοπτρίζουν τη μέση συμπεριφορά του συστήματος για τη συγκεκριμένη παραμετροποίηση. Συνολικά δοκιμάστηκαν 27 διαφορετικά σύνολα τιμών παραμέτρων, επομένως πραγματοποιήθηκαν 27 ανεξάρτητα πειράματα, καθένα από τα οποία περιλάμβανε 50 εκτελέσεις της συνάρτησης «generatePlan». Για κάθε πείραμα αποδίδεται ένας μοναδικός κωδικός `experimentId`, ενώ κάθε επιμέρους εκτέλεση αντιστοιχεί σε ένα μοναδικό `runId`. Οι δύο αυτοί αναγνωριστικοί κωδικοί ακολουθούν το αντικείμενο από τη στιγμή της εκκίνησης του αλγορίθμου μέχρι την καταγραφή των αποτελεσμάτων σε ένα αρχείο με `logs`, επιτρέποντας εκ των υστέρων ομαδοποίηση και φιλτράρισμα ανά πείραμα και ανά εκτέλεση. Σε κάθε πείραμα προφανώς, δεν άλλαζαν όλες οι παράμετροι, αλλά οι περισσότερες έμεναν σταθερές, προκειμένου να γίνει αντιληπτή η επίδραση της αλλαγής της κάθε παραμέτρου στα αποτελέσματα. Είναι προφανές από τα παραπάνω ότι η αξιολόγηση στην παρούσα εργασία βασίζεται κυρίως στην ανάλυση των αντικειμενικών συναρτήσεων και στη γραφική διερεύνηση των συμβιβασμών μεταξύ τους, και όχι στη χρήση καθιερωμένων μετρικών αξιολόγησης μετώπων Pareto, όπως αυτές παρουσιάστηκαν στο Κεφάλαιο 3. Η επιλογή αυτή σχετίζεται με το γεγονός ότι η παρούσα εργασία εστιάζει στην πρακτική αξιολόγηση της συμπεριφοράς του αλγορίθμου στο πλαίσιο μιας εφαρμογής, και όχι στη συγκριτική αποτίμηση πολλαπλών αλγορίθμων βελτιστοποίησης.

6.4.2.1 Υλοποίηση των μηχανισμών αξιολόγησης

Για να υποστηριχθεί ο παραπάνω μηχανισμός εκτέλεσης πειραμάτων, υλοποιήθηκε μια εξειδικευμένη κλάση καταγραφής, η `JsonIEARecorder`. Η κλάση αυτή είναι υπεύθυνη για τη μετατροπή των μετρήσεων του εξελικτικού αλγορίθμου σε εγγραφές μορφής JSON και την αποθήκευσή τους σε αρχείο τύπου JSON Lines (JSONL), ένα format στο οποίο κάθε γραμμή είναι ένα αυτόνομο JSON αντικείμενο. Το αρχείο καταγραφής ονομάζεται «`ea_runs.jsonl`» και αποθηκεύεται στον φάκελο εσωτερικής μνήμης της εφαρμογής (`ea_logs/ea_runs.jsonl`).

Η κλάση αρχικοποιεί ένα αντικείμενο `Json` της βιβλιοθήκης «`kotlinx.serialization`» με ενεργοποιημένες τις προεπιλεγμένες τιμές (`encodeDefaults = true`) και χωρίς `pretty-print`, ώστε κάθε εγγραφή να είναι σε μία μόνο γραμμή. Παράλληλα, χρησιμοποιεί ένα αντικείμενο `Mutex` για να διασφαλιστεί ότι η πρόσβαση στο αρχείο γίνεται σειριακά και χωρίς `race conditions`, καθώς και τα 50 πειράματα εκτελούνται παράλληλα σε ξεχωριστά `threads`, για λόγους ταχύτητας. Ο όρος `race conditions` σημαίνει ότι δύο ή περισσότερα `threads` προσπαθούν να προσπελάσουν ή να αλλάξουν ταυτόχρονα το ίδιο δεδομένο, οδηγώντας σε απρόβλεπτη ή λανθασμένη συμπεριφορά.

Στον πυρήνα της `JsonIEARecorder` βρίσκεται μια βοηθητική, ιδιωτική συνάρτηση «`append(obj: JsonObject)`», η οποία αναλαμβάνει να προσθέσει στο αρχείο καταγραφής κάθε νέα εγγραφή που παράγεται κατά την εκτέλεση του εξελικτικού αλγορίθμου. Η συνάρτηση αυτή δρα ως ενδιάμεσο στρώμα μεταξύ των μηχανισμών συλλογής δεδομένων και της φυσικής αποθήκευσης στο αρχείο και λειτουργεί με τρόπο που εξασφαλίζει την ασφάλεια και σειριακή εγγραφή των δεδομένων, ακόμη και όταν ο αλγόριθμος εκτελείται σε περιβάλλον πολλαπλών `threads`, κάνοντας χρήση του αντικειμένου `Mutex`, αναφέρθηκε προηγουμένως. Με την κατάλληλη χρήση συγχρονισμού αποτρέπεται οποιαδήποτε πιθανότητα ασυνεπούς καταγραφής, ενώ η εγγραφή κάθε αντικειμένου σε ανεξάρτητη γραμμή διατηρεί το αρχείο συμβατό με το format JSON Lines.

Η λειτουργία της καταγραφής σε επίπεδο γενιάς υλοποιείται μέσω της συνάρτησης «`logGeneration`», η οποία δημιουργεί μια εγγραφή που περιλαμβάνει όλες τις κρίσιμες πληροφορίες για τη συγκεκριμένη γενιά και τις αντίστοιχες τιμές των τριών στόχων. Η εγγραφή αυτή περιέχει τα αναγνωριστικά της

εκτέλεσης και του πειράματος, τον αριθμό της γενιάς και τις τιμές των αντικειμενικών συναρτήσεων που αποτυπώνουν τη θερμιδική απόκλιση, το σκορ μακροθρεπτικών και το σκορ μικροθρεπτικών. Η κλήση της συνάρτησης πραγματοποιείται σε κάθε κύκλο εξέλιξης, αμέσως μετά την αξιολόγηση του πληθυσμού της αντίστοιχης γενιάς, επιτρέποντας έτσι την ανακατασκευή της πορείας σύγκλισης του αλγορίθμου κατά την ανάλυση των αποτελεσμάτων. Η συνεχής αυτή ροή δεδομένων προς το αρχείο καταγραφής δημιουργεί μία λεπτομερή απεικόνιση της δυναμικής συμπεριφοράς του ΕΑ και παρέχει τη βάση για την παραγωγή διαγραμμάτων σύγκλισης.

Αντίστοιχα, η συνάρτηση «logFinalPopulation» καταγράφει τον τελικό πληθυσμό κάθε εκτέλεσης, αποθηκεύοντας για κάθε άτομο τις τιμές των τριών στόχων, όπως αυτές έχουν διαμορφωθεί με το πέρας της τελευταίας γενιάς. Με τον τρόπο αυτό συγκεντρώνεται ένα πλήρες σύνολο δεδομένων που αποτυπώνει τη δομή του τελικού μετώπου για κάθε εκτέλεση, επιτρέποντας αργότερα τη συγκριτική μελέτη της ποιότητας και της ποικιλίας των λύσεων που παρήγαγε το εκάστοτε σύνολο παραμέτρων. Τα δεδομένα αυτά είναι ουσιώδη όταν ζητείται η αποτίμηση της συνολικής απόδοσης ενός ΕΑ σε επίπεδο πληθυσμού και όχι μόνο της καλύτερης λύσης.

Η καταγραφή γίνεται ακόμη πιο στοχευμένη μέσω της συνάρτησης «logFinalBest», η οποία επικεντρώνεται αποκλειστικά στην τελική λύση που επιλέγει ο αλγόριθμος για να επιστρέψει στον χρήστη. Πέρα από τις τιμές των στόχων, η εγγραφή αυτή περιλαμβάνει και ένα map «ratios», δηλαδή τους λόγους πραγματικής πρόσληψης προς τον αντίστοιχο στόχο για μια σειρά διατροφικών στοιχείων. Το συγκεκριμένο πεδίο προσθέτει σημαντική πληροφορία, καθώς επιτρέπει την πλήρη αποτίμηση του κατά πόσο το τελικό πλάνο πλησιάζει ή αποκλίνει από τους διαιτητικούς στόχους του χρήστη. Παράλληλα, παρέχει και τη βάση για προχωρημένες αναλύσεις, όπως τη δημιουργία θερμικών χαρτών για επιλεγμένα μικροθρεπτικά συστατικά, που εκφράζουν με μεγαλύτερη λεπτομέρεια την ποιότητα των λύσεων που παράγει ο αλγόριθμος.

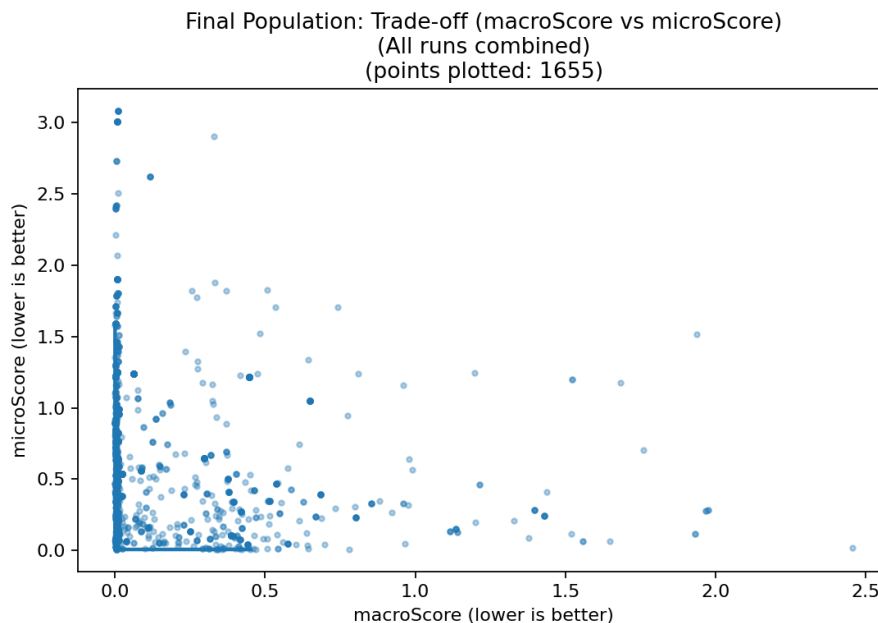
Στην υλοποίηση του εξελικτικού αλγορίθμου, ο JsonIARRecorder αξιοποιείται σε τρία χαρακτηριστικά σημεία. Πρώτον, κατά την εξέλιξη των γενιών, όπου η logGeneration καταγράφει συστηματικά την πρόοδο του αλγορίθμου. Δεύτερον, μετά την ολοκλήρωση της διαδικασίας εξέλιξης, όπου η logFinalPopulation αποθηκεύει τον τελικό πληθυσμό. Τρίτον, στο στάδιο επιλογής της τελικής λύσης, όπου η logFinalBest καταγράφει το πλάνο που επιλέχθηκε ως αντιπροσωπευτικό αποτέλεσμα της συγκεκριμένης εκτέλεσης. Με τη σειρά τους, οι επαναλαμβανόμενες εκτελέσεις των 27 διαφορετικών πειραμάτων και των 50 εκτελέσεων ανά πείραμα συσσωρεύουν όλες αυτές τις εγγραφές σε ένα ενιαίο αρχείο, το οποίο λειτουργεί ως πλήρης αποθήκη δεδομένων για την τελική ανάλυση και την παραγωγή των διαγραμμάτων.

6.4.2.2 Ανάλυση των πειραματικών αποτελεσμάτων και ερμηνεία των διαγραμμάτων

Αφού παρουσιάστηκε η διαδικασία συλλογής των δεδομένων αξιολόγησης μέσω του μηχανισμού καταγραφής της εφαρμογής και αναλύθηκε ο τρόπος με τον οποίο το αρχείο «ea_runs.jsonl» δημιουργείται και οργανώνεται, στην παρούσα υποενότητα εξετάζονται αναλυτικά τα διαγράμματα που παράγονται από το εξωτερικό Python script. Για κάθε ένα από τα 27 πειράματα, μετά την ολοκλήρωση των 50 εκτελέσεων του αλγορίθμου και τη δημιουργία του αντίστοιχου αρχείου «ea_runs.jsonl», το script εκτελούνταν προκειμένου να παραχθούν έξι διαγράμματα που συνοψίζουν την απόδοση της συγκεκριμένης παραμετροποίησης. Στη συνέχεια, τα διαγράμματα αυτά συγκρίνονταν με τα διαγράμματα του μέχρι εκείνη τη στιγμή καλύτερου πειράματος, έτσι ώστε η αξιολόγηση να γίνεται σταδιακά, με βάση ένα ολοένα και πιο βελτιωμένο σημείο αναφοράς. Η ανάλυση των παραγόμενων διαγραμμάτων θα πραγματοποιηθεί παρακάτω, ενώ είναι απαραίτητο να τονιστεί ότι το κάθε σχήμα που σχετίζεται με την περιγραφή του κάθε διαγράμματος αντιστοιχεί στο διάγραμμα που προέκυψε από τον

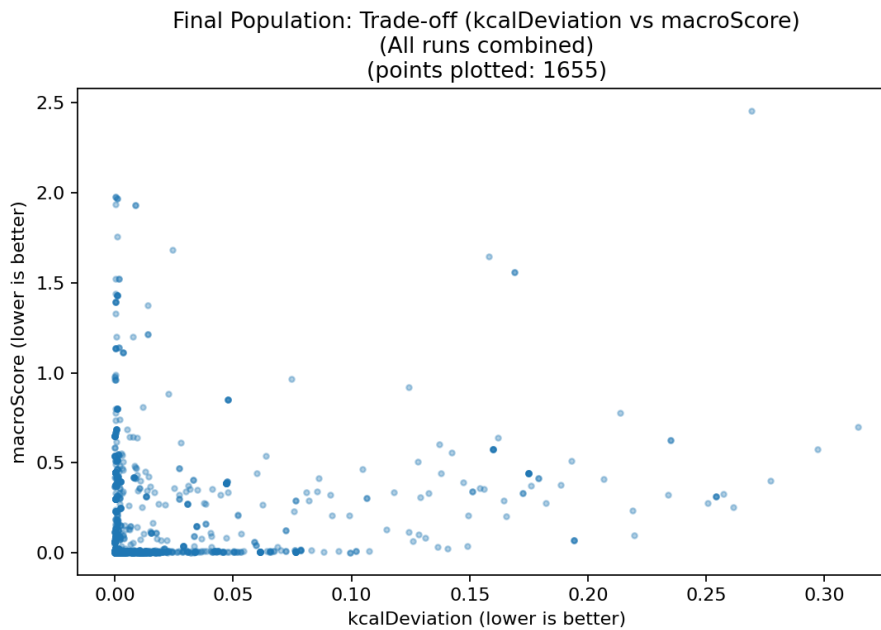
καλύτερο συνδυασμό παραμέτρων, δηλαδή αυτού που χρησιμοποιήθηκε στην εφαρμογή και παρουσιάστηκε στο προηγούμενο κεφάλαιο.

Το πρώτο διάγραμμα (Σχήμα 6.1), απεικονίζει το σύνολο των τελικών λύσεων από όλες τις εκτελέσεις ενός παραμετρικού σετ, προβάλλοντας τον συμβιβασμό μεταξύ των δύο στόχων που αφορούν τα μακροθρεπτικά και τα μικροθρεπτικά συστατικά. Κάθε σημείο αντιστοιχεί σε ένα άτομο του τελικού πληθυσμού, με τον οριζόντιο άξονα να εκφράζει το macroScore και τον κατακόρυφο το microScore, όπου σε αμφοτέρους τους στόχους, χαμηλότερες τιμές θεωρούνται καλύτερες. Η συγκέντρωση των περισσότερων σημείων κοντά στο κάτω αριστερό άκρο αποτελεί ένδειξη ότι η παραμετροποίηση οδηγεί σε λύσεις που επιτυγχάνουν ταυτόχρονα μικρές αποκλίσεις και στις δύο κατηγορίες θρεπτικών συστατικών. Αντιθέτως, η ύπαρξη πολλών σημείων διάσπαρτων προς τα δεξιά ή προς τα επάνω υποδηλώνει μειωμένη ικανότητα του αλγορίθμου να ισορροπήσει τους δύο αυτούς στόχους. Κατά τη σύγκριση δύο τέτοιων διαγραμμάτων, το καλύτερο θεωρείται εκείνο όπου τα σημεία βρίσκονται όσο το δυνατόν πιο κοντά στο κάτω αριστερό τμήμα και σχηματίζουν μια πυκνή περιοχή υψηλής ποιότητας. Ένα κακό διάγραμμα παρουσιάζει μεγάλη διασπορά, με συχνή εμφάνιση λύσεων που υστερούν έντονα είτε στα μακροθρεπτικά είτε στα μικροθρεπτικά.



Σχήμα 6.1: Trade-off macroScore-microScore στον τελικό πληθυσμό

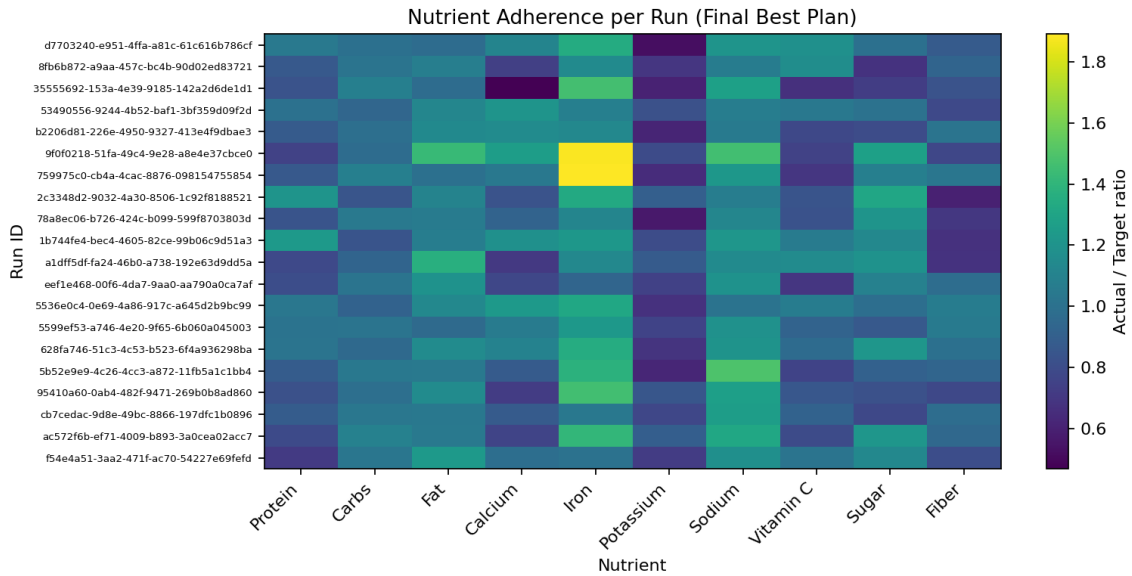
Το δεύτερο διάγραμμα (Σχήμα 6.2), εξετάζει τη σχέση μεταξύ της θερμιδικής απόκλισης και του macroScore. Και σε αυτή την περίπτωση, κάθε σημείο αντιστοιχεί σε μία λύση του τελικού πληθυσμού από όλες τις εκτελέσεις, επιτρέποντας έτσι τη συνολική απεικόνιση της επίδοσης μιας παραμετροποίησης. Η ευθυγράμμιση των περισσότερων σημείων κοντά στον άξονα των μικρών αποκλίσεων υποδηλώνει ότι ο αλγόριθμος καταφέρνει να παράγει πλάνα διατροφής με θερμιδικό ισοζύγιο κοντά στον στόχο. Όταν το μακροθρεπτικό σκορ παραμένει επίσης χαμηλό, τότε η παραμετροποίηση θεωρείται αποτελεσματική ως προς τον συνδυασμό αυτών των δύο διαστάσεων. Αντιθέτως, διάγραμμα όπου τα σημεία απλώνονται προς τα δεξιά ή προς τα επάνω, δείχνει ότι η παραμετροποίηση παράγει συχνότερα λύσεις που αποκλίνουν σημαντικά είτε στις θερμίδες είτε στα μακροθρεπτικά. Κατά τη σύγκριση δύο τέτοιων διαγραμμάτων, προτιμάται εκείνο στο οποίο το σύνολο των σημείων καταλαμβάνει μικρότερο εύρος τιμών και συγκεντρώνεται κοντά στις πιο επιθυμητές περιοχές του επιπέδου, δηλαδή στις χαμηλές αποκλίσεις και στα μικρά μακροθρεπτικά σφάλματα.



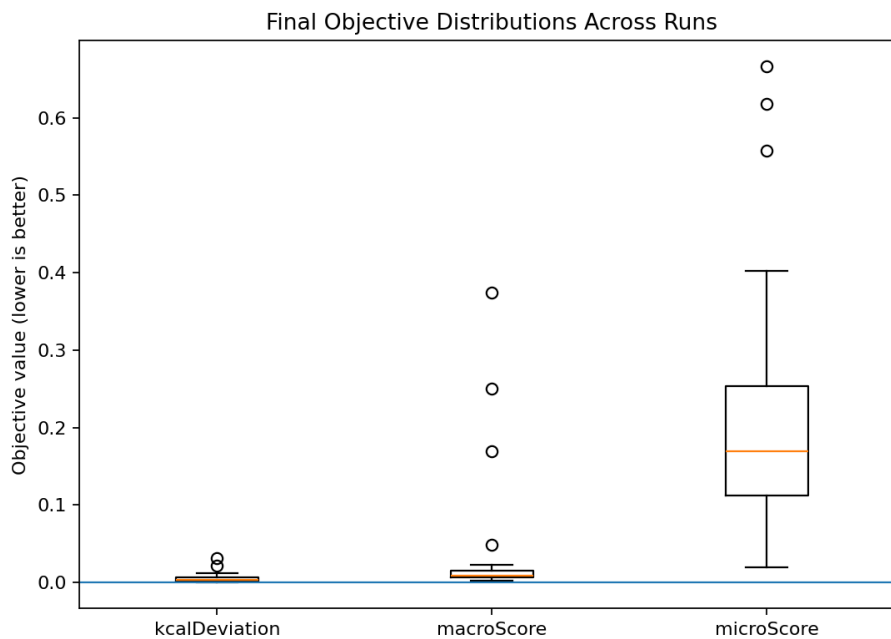
Σχήμα 6.2: Trade-off θερμδικής απόκλισης και macroScore

Το τρίτο διάγραμμα (Σχήμα 6.3) παρουσιάζει με μορφή θερμικού χάρτη (heatmap) τους λόγους πραγματικής προς επιθυμητή πρόσληψη για μια σειρά βασικών θρεπτικών συστατικών. Κάθε γραμμή αντιπροσωπεύει μια εκτέλεση και κάθε στήλη ένα συγκεκριμένο θρεπτικό συστατικό, ενώ η χρωματική απόχρωση υποδηλώνει πόσο κοντά βρίσκεται η υλοποιημένη πρόσληψη στον προτεινόμενο στόχο. Οι τιμές γύρω από το 1.0 υποδηλώνουν σχεδόν πλήρη ευθυγράμμιση με τον στόχο και όσο πιο μικρή είναι η απόκλιση τόσο καλύτερη θεωρείται η ποιότητα της λύσης. Υπερβάσεις ή σημαντικές ελλείψεις εμφανίζονται με χρωματικές εντάσεις στα άκρα του εύρους και υποδεικνύουν ότι το τελικό πλάνο δεν πέτυχε ισορροπία στο συγκεκριμένο συστατικό. Κατά τη σύγκριση παραμετροποιήσεων, ένα καλύτερο heatmap είναι εκείνο όπου κυριαρχούν ενδιάμεσες και ήπιες αποχρώσεις, που βρίσκονται κοντά στο επιθυμητό επίπεδο. Εάν ένα heatmap παρουσιάζει έντονες αποκλίσεις κατά μήκος συγκεκριμένων στηλών, τότε η παραμετροποίηση εμφανίζει συστημική αδυναμία σε ορισμένες θρεπτικές απαιτήσεις. Αντιθέτως, η ομοιομορφία των χρωμάτων και η εγγύτητα προς τους στόχους υποδηλώνουν σταθερή απόδοση και συνεπή συμπεριφορά του αλγορίθμου.

Στο τέταρτο διάγραμμα (Σχήμα 6.4) απεικονίζονται, με τη μορφή boxplots, οι τελικές τιμές των τριών αντικειμενικών συναρτήσεων που επιλέχθηκαν από τις επιμέρους εκτελέσεις μιας παραμετροποίησης. Κάθε boxplot αποθηκεύει στατιστική πληροφορία σχετικά με τη διάμεσο, το ενδοτεταρτημοριακό εύρος και την ύπαρξη πιθανών ακραίων τιμών. Η σύγκριση των boxplots επιτρέπει την εύκολη αποτίμηση της σταθερότητας της παραμετροποίησης. Μικρότερα κουτιά και χαμηλές τιμές διάμεσου αποτελούν δείκτη αυξημένης αξιοπιστίας, καθώς ο αλγόριθμος φαίνεται να παράγει συνεπείς και υψηλής ποιότητας λύσεις στις περισσότερες εκτελέσεις. Αντιθέτως, ιδιαίτερα εκτεταμένα κουτιά, μεγάλοι αριθμοί ακραίων τιμών ή εμφανείς μετατοπίσεις της διάμεσου προς υψηλότερες τιμές υποδηλώνουν χαμηλότερη σταθερότητα ή μειωμένη ποιότητα. Όταν συγκρίνονται δύο παραμετροποιήσεις, καλύτερη θεωρείται εκείνη που εμφανίζει μικρότερη διασπορά και χαμηλότερες κεντρικές τιμές σε όσο το δυνατόν περισσότερους στόχους.

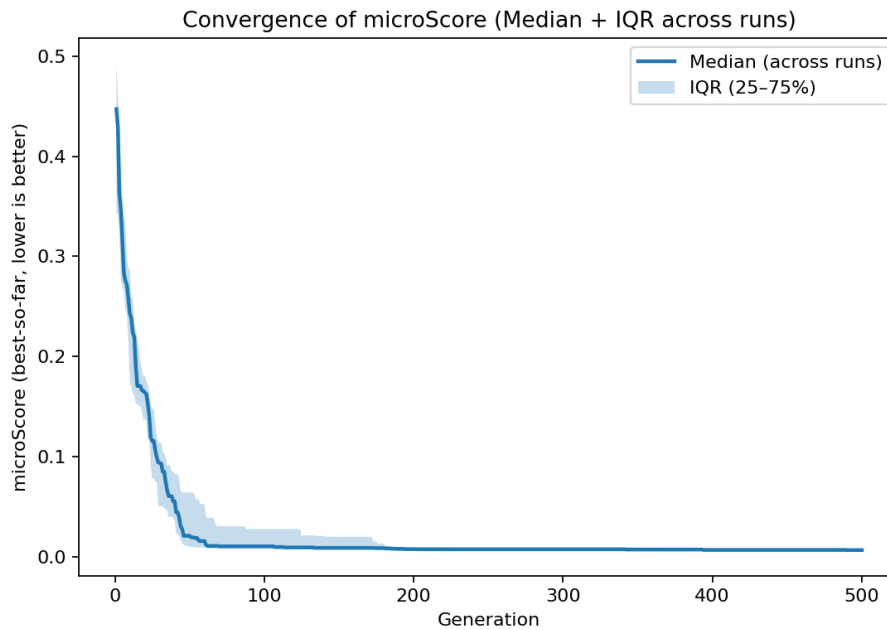


Σχήμα 6.3: Heatmap ικανοποίησης θρεπτικών στόχων



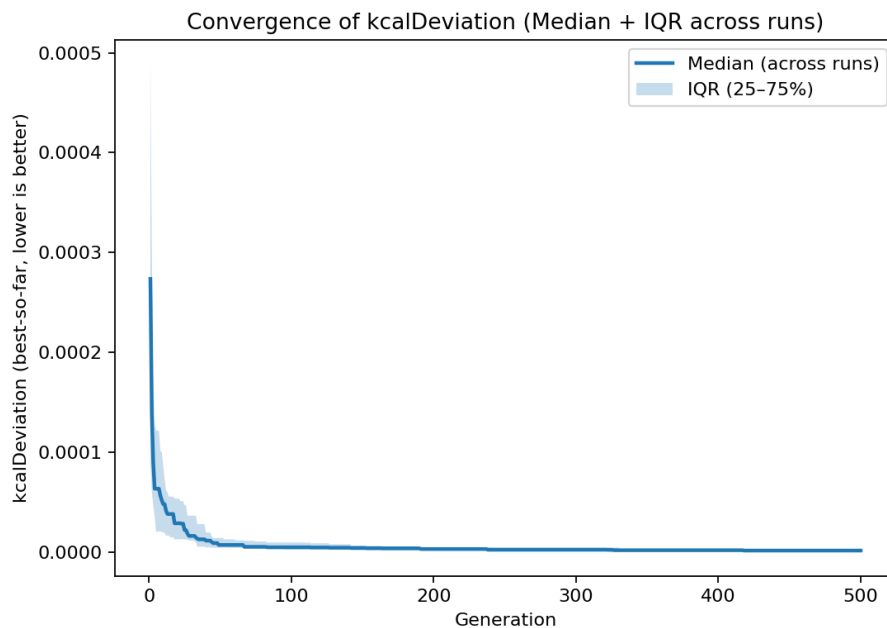
Σχήμα 6.4: Κατανομή τελικών τιμών αντικειμενικών συναρτήσεων

Το πέμπτο διάγραμμα (Σχήμα 6.5) απεικονίζει τη σύγκλιση του microScore κατά τη διάρκεια των γενιών, προβάλλοντας τη διάμεσο επίδοσης των εκτελέσεων και το ενδοτεταρτημοριακό εύρος τους, ώστε να αποτυπωθεί η συνολική τάση του αλγορίθμου. Η πορεία των καμπυλών προς χαμηλές τιμές υποδηλώνει αποτελεσματική βελτίωση των μικροθρεπτικών αποκλίσεων, ενώ η σταδιακή μείωση του εύρους αναδεικνύει την εξομάλυνση της συμπεριφοράς του πληθυσμού όσο η εξέλιξη προχωρά. Όταν συγκρίνονται δύο διαγράμματα σύγκλισης, καλύτερο θεωρείται εκείνο στο οποίο η διάμεσος αποκλίνει ταχύτερα προς μικρές τιμές και το ενδοτεταρτημοριακό εύρος συρρικνώνεται χωρίς σημαντικές διακυμάνσεις. Εάν η σύγκλιση είναι αργή ή το εύρος παραμένει μεγάλο, τότε η παραμετροποίηση εμφανίζει μειωμένη ικανότητα καθοδήγησης του πληθυσμού προς αξιόπιστες και σταθερές λύσεις.



Σχήμα 6.5: Σύγκλιση microScore ανά γενιά

Το έκτο και τελευταίο διάγραμμα (Σχήμα 6.6) παρουσιάζει αντίστοιχη οπτική σύγκλισης, αυτή τη φορά για τη θερμидική απόκλιση. Η χρήση της διάμεσου και του ενδοτεταρτημοριακού εύρους αποτυπώνει την ταχύτητα και τη σταθερότητα με την οποία ο αλγόριθμος επιτυγχάνει θερμидική ισορροπία. Ένα αποτελεσματικό διάγραμμα είναι εκείνο που παρουσιάζει ταχεία κάθοδο της καμπύλης σε τιμές κοντά στο μηδέν και συνάμα μειούμενο εύρος μεταβλητότητας ανάμεσα στις εκτελέσεις. Εάν παρατηρηθούν έντονες ταλαντώσεις, μεγάλες αποκλίσεις ή παρατεταμένη στασιμότητα, τότε η παραμετροποίηση κρίνεται λιγότερο αποτελεσματική σε αυτό το αντικειμενικό κριτήριο.



Σχήμα 6.6: Σύγκλιση θερμидικής απόκλισης ανά γενιά

Η συνολική διαδικασία ανάλυσης των διαγραμμάτων οδήγησε τελικά στη διαμόρφωση του συνόλου παραμέτρων που παρουσιάστηκε στο Κεφάλαιο 5, καθώς επέτρεψε την εμπειρική αξιολόγηση της συμπεριφοράς του αλγορίθμου σε πολλαπλές πειραματικές συνθήκες. Από τα αποτελέσματα προέκυψε ότι ορισμένες τιμές, και ιδίως το μέγεθος του πληθυσμού, διαφοροποιήθηκαν σημαντικά από τις αρχικές εκτιμήσεις. Η συγκεκριμένη επιλογή αιτιολογείται από το γεγονός ότι ο NSGA-II απαιτεί πληθυσμούς αυξημένου μεγέθους προκειμένου να αποτυπώσει επαρκώς τη μορφή του μετώπου Pareto σε πολυδιάστατα προβλήματα. Στο πλαίσιο προβλημάτων πολλαπλών στόχων με τρεις στόχους, το μέτωπο αποτελείται συνήθως από μεγάλο πλήθος αμοιβαία μη κυριαρχούμενων λύσεων, οι οποίες κατανέμονται σε διαφορετικές περιοχές του χώρου των objectives. Μικρότεροι πληθυσμοί δεν διαθέτουν την απαιτούμενη ποικιλία για να αναπαραστήσουν αυτό το εύρος λύσεων, οδηγώντας σε απώλεια τμημάτων του μετώπου και σε μειωμένη ποιότητα συμβιβασμών. Επιπλέον, ο μηχανισμός crowding distance του NSGA-II λειτουργεί αποτελεσματικότερα όταν η εκτίμηση της τοπικής πυκνότητας βασίζεται σε επαρκή αριθμό σημείων, γεγονός που ενισχύει τη σταθερότητα και την ομοιόμορφη διασπορά των λύσεων στο μέτωπο.

Τα πειράματα επιβεβαίωσαν αυτές τις θεωρητικές παρατηρήσεις. Πληθυσμοί της τάξης των 400 ατόμων παρήγαγαν πιο πλήρη και συνεκτική απεικόνιση του μετώπου Pareto, επιτρέποντας στον αλγόριθμο να διατηρήσει μεγαλύτερη ποικιλία λύσεων και να αποφεύγει τη σύγκλιση σε περιορισμένες περιοχές. Αντίστοιχα, ο αριθμός των γενεών ορίστηκε στις 500, καθώς προσέφερε επαρκή χρόνο σύγκλισης χωρίς να επιβαρύνει υπερβολικά την εκτέλεση της εφαρμογής σε πραγματική συσκευή. Δοκιμές με υψηλότερες τιμές δεν απέφεραν δραματική βελτίωση στις τελικές λύσεις, ενώ αύξαναν δυσανάλογα τον χρόνο αναμονής του χρήστη. Συνολικά, η επιλογή των τελικών παραμέτρων προέκυψε από τον συνδυασμό θεωρητικών ιδιοτήτων του NSGA-II, πειραματικών αποτελεσμάτων και πρακτικών περιορισμών της περιβάλλοντος εκτέλεσης, εξασφαλίζοντας μια ισορροπία ανάμεσα στην ποιότητα των παραγόμενων πλάνων και την αποδοτικότητα της εφαρμογής.

6.5 Περιορισμοί της υλοποίησης

Παρά την επιτυχία της εφαρμογής στην παραγωγή εξατομικευμένων διατροφικών πλάνων και την ομαλή λειτουργία όλων των κύριων ροών, η υλοποίηση παρουσιάζει ορισμένους πρακτικούς και τεχνικούς περιορισμούς που συνδέονται τόσο με τις επιλογές σχεδιασμού όσο και με τις ιδιαιτερότητες των εργαλείων και των δεδομένων που χρησιμοποιήθηκαν. Οι περιορισμοί αυτοί δεν αναιρούν τη λειτουργικότητα της εφαρμογής, αλλά καθορίζουν το πλαίσιο μέσα στο οποίο πρέπει να ερμηνεύονται τα αποτελέσματα και υποδεικνύουν ενδεχόμενες κατευθύνσεις βελτίωσης.

Αρχικά, παρότι η προσέγγιση της ταξινόμησης, που παρουσιάζεται στην υποενότητα 5.4.2, είναι λειτουργική και πρακτική, παρουσιάζει ορισμένους περιορισμούς που πρέπει να ληφθούν υπόψη. Η ταξινόμηση βασίζεται σε συσχετίσεις λέξεων-κλειδιών, οι οποίες αποτυπώνουν συνήθεις διατροφικές προτιμήσεις και όχι αυστηρούς κανόνες. Αυτό σημαίνει ότι, για παράδειγμα, ένα τρόφιμο με περιγραφή «steak sandwich» μπορεί να χαρακτηριστεί ως μεσημεριανό ή βραδινό, αν και στην πράξη θα μπορούσε να καταναλωθεί σε οποιοδήποτε γεύμα. Επιπλέον, στις περιπτώσεις των ειδικών διατροφικών προτύπων, όπως το vegan ή το keto, η προσέγγιση με βάση τις λέξεις-κλειδιά δεν μπορεί να εγγυηθεί απόλυτη ακρίβεια, καθώς ενδέχεται ένα τρόφιμο να μην περιλαμβάνει στη λεκτική του περιγραφή κάποια από τις απαγορευμένες ή επιθυμητές λέξεις, με αποτέλεσμα να ταξινομηθεί λανθασμένα. Συνεπώς, αν και ο μηχανισμός προσφέρει ένα σχετικά αξιόπιστο και αποδοτικό σημείο εκκίνησης, η χρήση του θα πρέπει να συνοδεύεται από επίγνωση ότι πρόκειται για μια ημι-ευριστική προσέγγιση που βασίζεται στη γλωσσική περιγραφή των τροφίμων και όχι σε απόλυτα διατροφικά δεδομένα. Παρά τα παραπάνω μειονεκτήματα, η μέθοδος αυτή κρίνεται αποτελεσματική για τις ανάγκες της εφαρμογής.

Σε επίπεδο αλγοριθμικής υποδομής, ο εξελικτικός αλγόριθμος βασίζεται αποκλειστικά σε διατροφικούς στόχους, δηλαδή θερμίδες, μακροθρεπτικά και επιλεγμένα μικροθρεπτικά, χωρίς να ενσωματώνει παραμέτρους σχετικές με γευστική συμβατότητα, πρακτικότητα ή οικονομικό κόστος. Η έλλειψη τέτοιων ποιοτικών κριτηρίων μπορεί να οδηγήσει σε πλάνα που είναι μεν διατροφικά ισορροπημένα, αλλά όχι απαραίτητα ελκυστικά ή λειτουργικά για τον χρήστη. Επιπλέον, η ποιότητα του παραγόμενου πλάνου εξαρτάται από το μέγεθος και τη σύνθεση του food pool. Για χρήστες με πολλές διατροφικές προτιμήσεις ή αλλεργίες, το διαθέσιμο σύνολο επιλογών μειώνεται αισθητά, περιορίζοντας την ποικιλία των λύσεων και επηρεάζοντας τη σταθερότητα της βελτιστοποίησης.

Σε σχέση με το περιβάλλον εκτέλεσης, η γενετική διαδικασία πραγματοποιείται τοπικά στη συσκευή, χωρίς χρήση εξωτερικών υπολογιστικών πόρων. Αν και αυτό εξασφαλίζει απόλυτη αυτονομία και άμεση απόκριση, μειώνει την απόδοση σε παλαιότερες ή χαμηλών προδιαγραφών συσκευές, όπου παρατηρούνται αυξημένοι χρόνοι εκτέλεσης. Παράλληλα, η εφαρμογή δεν περιλαμβάνει μηχανισμούς αυτοματοποιημένων δοκιμών. Η αξιολόγηση και βελτίωση των λειτουργιών βασίστηκε αποκλειστικά σε χειροκίνητο έλεγχο. Η απουσία unit tests ή UI tests περιορίζει την ικανότητα συστηματικής ανίχνευσης παλαιών σφαλμάτων μετά από μελλοντικές αλλαγές ή επεκτάσεις.

Τέλος, η επιλογή αποκλειστικής χρήσης του USDA API συνιστά οικονομικό και πρακτικό περιορισμό. Αν και το USDA προσφέρει πλήρη και δωρεάν πρόσβαση, δεν παρέχει έτοιμες συνταγές, γεγονός που θα μπορούσε να προσφέρει πιο ολοκληρωμένες και ρεαλιστικές διατροφικές επιλογές. Αντίθετα, πλατφόρμες όπως το Edamam, που παρέχουν τέτοιες δυνατότητες, απαιτούν συνδρομή για χρήση σε βάθος. Η παρούσα υλοποίηση, επομένως, επικεντρώνεται αποκλειστικά σε τρόφιμα και όχι σε πλήρη γεύματα, θέτοντας ένα φυσικό όριο στην εμπειρία του χρήστη.

6.6 Επίλογος

Το παρόν κεφάλαιο εστιάζει στην αξιολόγηση της προτεινόμενης εφαρμογής και της αλγοριθμικής προσέγγισης που υιοθετήθηκε, μέσα από τη συστηματική διερεύνηση της συμπεριφοράς τους σε διαφορετικά σενάρια χρήσης. Η ανάλυση των αποτελεσμάτων αναδεικνύει τον βαθμό στον οποίο οι σχεδιαστικές επιλογές και η υλοποίηση ανταποκρίνονται στους στόχους της εργασίας, παρέχοντας τεκμηριωμένη εικόνα για την αποτελεσματικότητα και τη λειτουργικότητα του συστήματος. Το κεφάλαιο αυτό λειτουργεί ως σημείο αναφοράς για την αποτίμηση των περιορισμών και των δυνατοτήτων της προτεινόμενης λύσης, θέτοντας τις βάσεις για τη συνολική αποτίμηση της εργασίας. Στο επόμενο και τελευταίο κεφάλαιο ακολουθεί η εξαγωγή συμπερασμάτων και η συζήτηση πιθανών κατευθύνσεων για μελλοντική επέκταση και βελτίωση της εφαρμογής.

Κεφάλαιο 7ο: Συμπεράσματα και μελλοντικές επεκτάσεις

7.1 Εισαγωγή

Το παρόν κεφάλαιο ολοκληρώνει τη δομή της εργασίας παρουσιάζοντας τα συνολικά συμπεράσματα που προέκυψαν από την ανάπτυξη και αξιολόγηση της εφαρμογής, καθώς και τις προοπτικές μελλοντικής της εξέλιξης. Έπειτα από τη θεωρητική ανάλυση των διατροφικών αρχών, την παρουσίαση των εξελικτικών αλγορίθμων και τη λεπτομερή περιγραφή της υλοποίησης και των δοκιμών, καθίσταται πλέον εφικτό να αποτιμηθεί η συνολική συνεισφορά της εργασίας και να αναδειχθούν τα σημεία στα οποία μπορεί να βασιστεί μια περαιτέρω ερευνητική ή τεχνολογική επέκταση.

Η ανάπτυξη της εφαρμογής ανέδειξε πρακτικές προκλήσεις που σχετίζονται με τη διαχείριση μεγάλου όγκου δεδομένων, την ταξινόμηση των τροφών και την πλήρη ευθυγράμμιση των παραγόμενων πλάνων με τα πρότυπα της διατροφικής επιστήμης. Οι παρατηρήσεις αυτές παρέχουν μια σαφή βάση για τη διατύπωση προτάσεων βελτίωσης, τόσο σε επίπεδο αλγοριθμικής προσέγγισης όσο και σε επίπεδο εμπειρίας χρήστη και συνολικής αρχιτεκτονικής. Επιπλέον, σύγχρονες τεχνολογικές εξελίξεις, όπως τα συστήματα μηχανικής μάθησης, οι multi-platform τεχνολογίες ανάπτυξης και η ενσωμάτωση επιπρόσθετων πηγών δεδομένων, προσφέρουν νέες δυνατότητες για την ενίσχυση της λειτουργικότητας και της ακρίβειας του συστήματος.

7.2 Συνοπτική παρουσίαση συνεισφοράς της εργασίας

Η παρούσα εργασία συνέβαλε στη διερεύνηση και υλοποίηση ενός ολοκληρωμένου συστήματος εξατομικευμένου διατροφικού προγραμματισμού, το οποίο αξιοποιεί εξελικτικούς αλγορίθμους πολλαπλών στόχων για την παραγωγή ρεαλιστικών και προσαρμοσμένων πλάνων διατροφής. Η συνεισφορά της εντοπίζεται σε πολλαπλά επίπεδα, καθώς συνδυάζει τεκμηριωμένες διατροφικές αρχές με στοχαστικές μεθόδους βελτιστοποίησης και ένα σύγχρονο περιβάλλον εφαρμογής για κινητές συσκευές. Η ανάπτυξη της μεθοδολογίας βασίστηκε στη μετατροπή των ενεργειακών και θρεπτικών απαιτήσεων του χρήστη σε υπολογιστικά μεγέθη, τα οποία ενσωματώθηκαν στη συνάρτηση καταλληλότητας του εξελικτικού αλγορίθμου, επιτρέποντας την ταυτόχρονη αξιολόγηση της θερμιδικής ακρίβειας, της ισορροπίας των μακροθρεπτικών και της προσέγγισης επιλεγμένων μικροθρεπτικών συστατικών.

Επιπλέον, η εφαρμογή αξιοποίησε πραγματικά δεδομένα τροφίμων μέσω του USDA FoodData Central API, γεγονός που προσέδωσε ρεαλισμό και υψηλή αξιοπιστία στις παραγόμενες λύσεις. Η ενσωμάτωση τεχνικών όπως το Paging 3, ο μηχανισμός Remote Mediator και η τοπική αποθήκευση στη Room Database επέτρεψαν τη διαχείριση μεγάλου όγκου δεδομένων με τρόπο αποδοτικό και ανθεκτικό σε περιορισμούς συνδεσιμότητας. Παράλληλα, ο ταξινομητικός μηχανισμός των τροφών προσαρμόστηκε στις διατροφικές προτιμήσεις και στους περιορισμούς του χρήστη, επιτρέποντας την αυτοματοποίηση της διαδικασίας επιλογής τροφών συμβατών με το εκάστοτε προφίλ.

Σε επίπεδο πειραματικής αξιολόγησης, η εργασία ανέδειξε τη δυνατότητα βελτίωσης της ποιότητας των παραγόμενων λύσεων μέσω συστηματικής διερεύνησης των παραμέτρων του αλγορίθμου και προσεκτικής ανάλυσης της συμπεριφοράς του κατά τη διάρκεια των γενεών. Η υλοποίηση και καταγραφή πολλαπλών δοκιμών κατέδειξε ότι ο αλγόριθμος μπορεί να προσεγγίσει αποτελεσματικά τις διατροφικές απαιτήσεις του χρήστη και να συγκλίνει σε λύσεις που συνδυάζουν ακρίβεια και ποικιλία.

Πέρα από τη μεθοδολογική και τεχνική της διάσταση, η εργασία διαθέτει και ουσιαστική συνεισφορά προς τον τελικό χρήστη, ο οποίος αποτελεί τον κύριο αποδέκτη του συστήματος. Η εφαρμογή προσφέρει έναν άμεσο και πρακτικό τρόπο παρακολούθησης της διατροφής, επιτρέποντας στον χρήστη να λάβει εξατομικευμένες προτάσεις που βασίζονται σε επιστημονικά τεκμηριωμένες αρχές και πραγματικά δεδομένα τροφίμων. Η δυνατότητα αυστηρής προσαρμογής του πλάνου στις προσωπικές ανάγκες, τις προτιμήσεις, τους στόχους και τους περιορισμούς του κάθε ατόμου, καθιστά το σύστημα προσβάσιμο σε ευρύ φάσμα χρηστών, από άτομα που επιδιώκουν απώλεια ή διατήρηση βάρους μέχρι όσους ακολουθούν συγκεκριμένα διατροφικά πρότυπα.

Τέλος, η εργασία προσφέρει ένα λειτουργικό παράδειγμα εφαρμογής όπου η διατροφική επιστήμη, η τεχνητή νοημοσύνη και η ανάπτυξη λογισμικού για κινητές συσκευές συνδυάζονται σε ένα συνεκτικό σύστημα. Η συνεισφορά αυτή αποδεικνύει ότι οι εξελικτικές μέθοδοι μπορούν να αποτελέσουν ένα ισχυρό εργαλείο για την υποστήριξη της εξατομικευμένης διατροφής, ανοίγοντας τον δρόμο για περαιτέρω έρευνα και ανάπτυξη σε ένα πεδίο με αυξημένο επιστημονικό και κοινωνικό ενδιαφέρον.

7.3 Βασικά ερευνητικά ευρήματα

Πέρα από τη συνοπτική αποτύπωση της συνεισφοράς της εργασίας, προκύπτουν ορισμένα γενικότερα συμπεράσματα που αφορούν τη συμπεριφορά και την καταλληλότητα της εξελικτικής προσέγγισης στο συγκεκριμένο πρόβλημα. Η εφαρμογή εξελικτικών αλγορίθμων πολλαπλών στόχων στο πρόβλημα του εξατομικευμένου διατροφικού σχεδιασμού ανέδειξε ορισμένα χαρακτηριστικά και συμπεράσματα που υπερβαίνουν τη συγκεκριμένη υλοποίηση και αφορούν γενικότερα τη φύση του προβλήματος. Η ταυτόχρονη βελτιστοποίηση αντικρουόμενων διατροφικών στόχων, σε συνδυασμό με αυστηρούς και ετερογενείς περιορισμούς, επιβεβαιώνει ότι η αναζήτηση μίας μοναδικής βέλτιστης λύσης είναι λιγότερο ουσιαστική από την παραγωγή ενός συνόλου εναλλακτικών λύσεων που εκφράζουν διαφορετικούς, αλλά αποδεκτούς, συμβιβασμούς. Στο πλαίσιο αυτό, οι εξελικτικοί αλγόριθμοι πολλαπλών στόχων αποδείχθηκαν ιδιαίτερα κατάλληλοι, καθώς επέτρεψαν τη φυσική ενσωμάτωση της πολυπλοκότητας του προβλήματος χωρίς την ανάγκη τεχνητής συγχώνευσης των στόχων.

Ένα από τα βασικά ευρήματα αφορά τη διαφορετική επίδραση που έχουν οι επιμέρους περιορισμοί στη συμπεριφορά της αναζήτησης. Περιορισμοί που σχετίζονται με αποκλεισμούς τροφίμων, όπως αλλεργίες ή αυστηρές διατροφικές προτιμήσεις, περιορίζουν σημαντικά τον διαθέσιμο χώρο λύσεων και επηρεάζουν άμεσα την ποικιλία των παραγόμενων λύσεων. Σε αυτές τις περιπτώσεις, το μέτωπο Pareto τείνει να παρουσιάζει μικρότερη έκταση και αυξημένη συγκέντρωση λύσεων, γεγονός που υποδεικνύει μειωμένη ελευθερία συμβιβασμών. Αντίθετα, περιορισμοί που αφορούν ποσοστιαία όρια μακροθρεπτικών ή θερμοδικές αποκλίσεις φαίνεται να επηρεάζουν κυρίως τη θέση του μετώπου Pareto στον χώρο των στόχων, χωρίς να περιορίζουν στον ίδιο βαθμό τη δομική του ποικιλία.

Η ανάλυση της συμπεριφοράς των αντικειμενικών συναρτήσεων ανέδειξε επίσης διαφοροποίηση ως προς τη δυσκολία βελτιστοποίησης των επιμέρους στόχων. Η επίτευξη ικανοποιητικής συμμόρφωσης ως προς τα μακροθρεπτικά συστατικά εμφανίστηκε συγκριτικά πιο σταθερή και προβλέψιμη, καθώς βασίζεται σε λιγότερες και πιο ομαλές μεταβλητές. Αντίθετα, η βελτιστοποίηση της μικροθρεπτικής επάρκειας αποδείχθηκε πιο απαιτητική, λόγω της μεγαλύτερης διακριτότητας των θρεπτικών στοιχείων, της άνισης κατανομής τους στα τρόφιμα και της αυξημένης ευαισθησίας τους σε μικρές αλλαγές στη σύνθεση του γεύματος. Το γεγονός αυτό υποδηλώνει ότι στόχοι που σχετίζονται με μικροθρεπτικά στοιχεία εισάγουν αυξημένη πολυπλοκότητα στο πρόβλημα και επηρεάζουν δυσμενέστερα τη σύγκλιση του αλγορίθμου.

Σε επίπεδο δυναμικής σύγκλισης, παρατηρείται ότι η συμπεριφορά του αλγορίθμου επηρεάζεται έντονα από τον τρόπο με τον οποίο αλληλεπιδρούν οι στόχοι μεταξύ τους. Σε περιπτώσεις όπου οι στόχοι είναι έντονα αντικρουόμενοι, η εξέλιξη του πληθυσμού τείνει να σταθεροποιείται σε ένα σύνολο λύσεων που αντανακλά ρεαλιστικούς συμβιβασμούς, χωρίς να παρατηρείται μονομερής βελτιστοποίηση κάποιου στόχου εις βάρος των υπολοίπων. Η ιδιότητα αυτή είναι ιδιαίτερα σημαντική για εφαρμογές διατροφικού σχεδιασμού, όπου η «τέλεια» συμμόρφωση σε έναν στόχο συχνά οδηγεί σε μη ρεαλιστικές ή πρακτικά ανεφάρμοστες λύσεις.

Συνολικά, τα ευρήματα της παρούσας εργασίας υποδεικνύουν ότι η χρήση εξελικτικών αλγορίθμων πολλαπλών στόχων στον διατροφικό σχεδιασμό δεν αποτελεί απλώς μια εναλλακτική υπολογιστική προσέγγιση, αλλά μια μεθοδολογία που ευθυγραμμίζεται εγγενώς με τη φύση του προβλήματος. Η δυνατότητα διαχείρισης πολλαπλών στόχων και περιορισμών, η παραγωγή εναλλακτικών λύσεων και η υποστήριξη συνειδητών συμβιβασμών καθιστούν τους αλγορίθμους αυτούς ιδιαίτερα κατάλληλους για εφαρμογές που στοχεύουν στην εξατομίκευση και στη λήψη αποφάσεων υπό αντικρουόμενες απαιτήσεις.

7.4 Προτάσεις βελτίωσης της εφαρμογής και του αλγορίθμου

Τα ευρήματα που προκύπτουν από την παρούσα ανάλυση αναδεικνύουν, πέρα από τα συμπεράσματα που μπορούν να εξαχθούν, πεδία και κατευθύνσεις στα οποία η προτεινόμενη προσέγγιση μπορεί να επεκταθεί ή να βελτιωθεί περαιτέρω. Μία σημαντική κατεύθυνση μελλοντικής βελτίωσης αφορά το ζήτημα της ταξινόμησης των τροφίμων, το οποίο δεν αποτελεί απλώς τεχνικό στάδιο προεπεξεργασίας, αλλά εγγενές πρόβλημα μοντελοποίησης με άμεσο αντίκτυπο στη διαδικασία βελτιστοποίησης. Κατηγορίες όπως vegan, vegetarian, keto ή διατροφικές δυσανεξίες δεν ορίζονται πάντοτε με αυστηρά και μονοσήμαντα κριτήρια, καθώς συχνά εξαρτώνται από το συμφραζόμενο, τη σύνθεση του τροφίμου και υποκειμενικές ή πολιτισμικές παραμέτρους. Η ασαφής ή εσφαλμένη απόδοση τέτοιων χαρακτηριστικών μπορεί να περιορίσει ή να αλλοιώσει τον χώρο λύσεων, επηρεάζοντας τόσο τη σύγκλιση όσο και τη διαφορετικότητα των παραγόμενων διατροφικών πλάνων. Στο πλαίσιο αυτό, η αξιοποίηση τεχνικών μηχανικής μάθησης, όπως μοντέλων ταξινόμησης βασισμένων σε language embeddings [90] ή fine-tuned neural networks [91], θα μπορούσε να συμβάλει στην πιο αξιόπιστη και συνεπή αναπαράσταση των διατροφικών κατηγοριών, λαμβάνοντας υπόψη το σημασιολογικό περιεχόμενο και όχι μόνο επιφανειακά χαρακτηριστικά των τροφίμων.

Αντίστοιχα, η ενσωμάτωση συστημάτων συστάσεων βασισμένων σε μηχανική μάθηση θα μπορούσε να ενισχύσει την ποιότητα των προτάσεων, καθώς ο εξελικτικός αλγόριθμος θα μπορούσε να ενημερώνεται από μοντέλα που μαθαίνουν δυναμικά τις προτιμήσεις του χρήστη βάσει της αλληλεπίδρασης με την εφαρμογή. Με αυτόν τον τρόπο, το διατροφικό πλάνο δεν θα περιοριζόταν μόνο στην επίτευξη θεραπευτικών στόχων, αλλά θα λάμβανε υπόψη παραμέτρους όπως βαθμολογίες ικανοποίησης, συχνότητα επιλογής συγκεκριμένων τροφών ή αποφυγή ανθυγιεινών μοτίβων. Παράλληλα, η χρήση πολλαπλών πηγών δεδομένων, όπως το Edamam API ή άλλες εξειδικευμένες βάσεις τροφίμων, μπορεί να συμβάλει στη βελτίωση του εύρους και της ποικιλίας των διαθέσιμων τροφών, εξασφαλίζοντας μεγαλύτερη διαθεσιμότητα δεδομένων και πιο ρεαλιστικές προτάσεις.

Τέλος, σε επίπεδο αρχιτεκτονικής, η ανάπτυξη ενός αποκλειστικού backend με RESTful APIs θα παρείχε μεγαλύτερη ευελιξία στη διαχείριση των δεδομένων, θα βελτίωνε τον έλεγχο της ροής πληροφορίας και θα επέτρεπε την εφαρμογή πρόσθετων μηχανισμών ελέγχου, caching, ασφάλειας και analytics. Η σταδιακή μετάβαση σε Kotlin Multiplatform με Compose Multiplatform θα καθιστούσε εφικτή την επέκταση της εφαρμογής σε iOS και web περιβάλλοντα, με ελάχιστες αλλαγές στον κώδικα,

προσφέροντας ενιαία βάση κώδικα και βελτιωμένη διαλειτουργικότητα. Με αυτόν τον τρόπο, η εφαρμογή θα μπορούσε να καλύψει μεγαλύτερο εύρος χρηστών και να διατηρηθεί πιο εύκολα σε βάθος χρόνου, μειώνοντας τον χρόνο και το κόστος συντήρησης.

7.5 Επίλογος

Το παρόν κεφάλαιο εστιάζει στη σύνθεση και ερμηνεία των συμπερασμάτων που προκύπτουν από την παρούσα εργασία, μετατοπίζοντας το βάρος από την περιγραφή της υλοποίησης και των αποτελεσμάτων στη συνολική αποτίμηση της προσέγγισης. Μέσα από την ανάδειξη των βασικών ερευνητικών ευρημάτων και τη συζήτηση πιθανών κατευθύνσεων βελτίωσης, καθίσταται σαφές ότι ο διατροφικός σχεδιασμός συνιστά ένα σύνθετο πρόβλημα, στο οποίο η επιλογή μοντελοποιητικών και αλγοριθμικών παραδοχών επηρεάζει καθοριστικά τη συμπεριφορά και την αποτελεσματικότητα της λύσης. Το κεφάλαιο αυτό ολοκληρώνει τη διπλωματική εργασία, προσφέροντας μια συνολική θεώρηση των αποτελεσμάτων και θέτοντας το πλαίσιο για περαιτέρω μελέτη και εξέλιξη της προτεινόμενης προσέγγισης. Συνολικά, η παρούσα διπλωματική εργασία καθιστά σαφές ότι ο συνδυασμός εξελικτικών αλγορίθμων πολλαπλών στόχων και σύγχρονων τεχνολογιών λογισμικού μπορεί να υποστηρίξει την ανάπτυξη εφαρμογών με στόχο τον αποτελεσματικό σχεδιασμό εξατομικευμένων διατροφικών προγραμμάτων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] C. R. Xavier, J. G. R. Silva, G. R. Duarte, I. A. Carvalho, V. F. Vieira and L. Goliatt, “An island-based hybrid evolutionary algorithm for caloric-restricted diets,” *Evolutionary Intelligence*, vol. 16, pp. 553-564, 2023.
- [2] G. Masset, P. Monsivais, M. Maillot, N. Darmon, and A. Drewnowski, “Diet optimization methods can help translate dietary guidelines into a cancer prevention food plan,” *The Journal of Nutrition*, vol. 139, no. 8, pp. 1541-1548, Aug. 2009.
- [3] B. Koroušić Seljak, “Computer-based dietary menu planning,” *Journal of Food Composition and Analysis*, vol. 22, no. 5, pp. 414-420, Aug. 2009.
- [4] J. Kołodziejczyk and Ł. Przybyłek, “Automatic dietary menu planning based on evolutionary algorithm,” in *Proc. Conference on Computer Science*, 2012. [Online]. Available: <https://doi.org/10.13140/RG.2.1.5183.8881>.
- [5] L. K. Gautam and V. S. Gulhane, “Optimization methods and its application in nutritional diet,” *Proc. Int. Conf. Recent Trends in Applied Science, Engineering and Technology (RTASET)*, Jan. 2021.
- [6] G. J. Stigler, “The cost of subsistence,” *Journal of Farm Economics*, vol. 27, no. 2, pp. 303-314, May 1945.
- [7] C. Türkmenoğlu, A. Ş. E. Uyar, and B. Kiraz, “Recommending healthy meal plans by optimising nature-inspired many-objective diet problem,” *Health Informatics Journal*, vol. 27, no. 1, pp. 1-10, Jan. 2021.
- [8] T. Hernández, A. Saavedra, M. S. Pomares, and A. Rojas, “Goal programming approaches for healthy diet planning,” *Applied Soft Computing*, vol. 78, pp. 595-606, May 2019.
- [9] T. Amin, J. Mulligan-Gow, and H. Zhang, “Integer programming models for healthy diet planning,” *Journal of Health Informatics*, vol. 24, no. 3, pp. 215-228, 2018.
- [10] D. Marrero, F. Fernandez, R. Lopez, and M. Nunez, “Multi-objective evolutionary algorithms for diet formulation,” *Expert Systems with Applications*, vol. 100, pp. 30-45, Jun. 2018.
- [11] N. Jia, J. Chen, R. Wang, and M. Li, “Towards automatically generating meal plan based on genetic algorithm,” *Soft Computing*, vol. 28, pp. 6893-6908, Jan. 2024.
- [12] U.S. Department of Agriculture, “FoodData Central,” Agricultural Research Service, Washington, DC. [Online]. Available: <https://fdc.nal.usda.gov>.
- [13] M. D. Mifflin, S. T. St Jeor, L. A. Hill, B. J. Scott, S. A. Daugherty, and Y. O. Koh, “A new predictive equation for resting energy expenditure in healthy individuals,” *American Journal of Clinical Nutrition*, vol. 51, no. 2, pp. 241-247, Feb. 1990.
- [14] Institute of Medicine (IOM), *Dietary Reference Intakes for Energy, Carbohydrate, Fiber, Fat, Fatty Acids, Cholesterol, Protein, and Amino Acids*. Washington, DC: National Academies Press, 2002/2005.
- [15] FAO/WHO/UNU, *Human Energy Requirements: Report of a Joint Expert Consultation*. Rome: FAO, 2004.
- [16] N. K. Fukagawa, “USDA’s FoodData Central: what is it and why is it needed today?,” *American Journal of Clinical Nutrition*, 2022.

- [17] M. Westerterp, "Control of energy expenditure in humans," *European Journal of Clinical Nutrition*, vol. 71, no. 3, pp. 340-344, 2017.
- [18] J. Speakman and C. Selman, "Physical activity and resting metabolic rate," *Proceedings of the Nutrition Society*, vol. 62, no. 3, pp. 621-634, 2003.
- [19] National Institutes of Health (NIH), "Clinical guidelines on the identification, evaluation, and treatment of overweight and obesity in adults," *Obesity Research*, vol. 6, no. 2, pp. 51S-209S, 1998.
- [20] W. O. Atwater and F. G. Benedict, *Experiments on the metabolism of matter and energy in the human body*, U.S. Department of Agriculture, Office of Experiment Stations, Bulletin No. 69, 1899.
- [21] M. A. Graber, *Human Nutrition: From the Gastrointestinal Tract to the Cells*, 2nd ed. Elsevier, 2019.
- [22] Institute of Medicine (IOM), *Dietary Reference Intakes for Energy, Macronutrients, and Amino Acids*. Washington, DC: National Academies Press, 2005.
- [23] H. Drewnowski, "Carbohydrate quality and dietary patterns," *Nutrients*, vol. 14, no. 3, pp. 451-460, 2022.
- [24] A. Keys, "Serum cholesterol response to dietary fat," *Lancet*, vol. 273, no. 7073, pp. 787-791, 1957.
- [25] World Health Organization (WHO), *Diet, Nutrition and the Prevention of Chronic Diseases*, Technical Report Series No. 916, Geneva, 2003.
- [26] B. A. Bowman and R. M. Russell, *Present Knowledge in Nutrition*, 11th ed. Academic Press, 2020.
- [27] World Health Organization (WHO), *Nutritional anaemias: tools for effective prevention and control*, Geneva, 2017.
- [28] National Institutes of Health (NIH), "Calcium: Fact Sheet for Health Professionals," Office of Dietary Supplements, 2022. [Online]. Available: <https://ods.od.nih.gov/factsheets/Calcium-HealthProfessional>.
- [29] D. E. Bazzano, "Dietary potassium intake and cardiovascular health," *Journal of the American College of Cardiology*, vol. 75, no. 22, pp. 2732-2740, 2020.
- [30] World Health Organization (WHO), *Guideline: Sodium intake for adults and children*, Geneva, 2012.
- [31] S. Carr and B. Frei, "Toward a new recommended dietary allowance for vitamin C based on antioxidant and health effects in humans," *American Journal of Clinical Nutrition*, vol. 69, no. 6, pp. 1086-1107, 1999.
- [32] A. Threapleton et al., "Dietary fibre intake and risk of cardiovascular disease," *BMJ*, vol. 347, pp. f6879-f6887, 2013.
- [33] World Health Organization (WHO), *Guideline: Sugars intake for adults and children*, Geneva, 2015.
- [34] N. Jia, J. Chen, R. Wang, and M. Li, "Towards automatically generating meal plan based on genetic algorithm," *Soft Computing*, vol. 28, pp. 6893-6908, 2024.
- [35] L. Ordovas et al., "Personalized nutrition and health," *BMJ*, vol. 361, pp. k2173, 2018.

- [36] M. Kalatzis, A. R. Cappello, and G. De Luca, "Digital health and the future of personalized nutrition," *Frontiers in Nutrition*, vol. 8, no. 722, pp. 1-10, 2021.
- [37] M. Kamel Boulos and A. Al-Shorbaji, "On the Internet of Things, smart cities and the WHO Healthy Cities," *Int. J. Health Geogr.*, vol. 13, no. 10, pp. 1-13, 2014.
- [38] U.S. Department of Agriculture, "FoodData Central API Documentation," USDA Agricultural Research Service, 2024. [Online]. Available: <https://fdc.nal.usda.gov/api-key-signup.html>.
- [39] U.S. Department of Agriculture, "FoodData Central - About, Terms and Conditions," USDA Agricultural Research Service, 2023. [Online]. Available: <https://fdc.nal.usda.gov/about-us.html>.
- [40] United States Department of Agriculture, *Foundation Foods Documentation*, 2024. [Online]. Available: https://fdc.nal.usda.gov/Foundation_Foods_Documentation.html.
- [41] United States Department of Agriculture, Agricultural Research Service (ARS), *Food and Nutrient Database for Dietary Studies (FNDDS): Overview*, 2024. [Online]. Available: <https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/food-surveys-research-group/docs/fndds/>.
- [42] Edamam Inc., "Edamam Food and Grocery Database API Documentation," 2024. [Online]. Available: <https://developer.edamam.com/food-database-api>.
- [43] Edamam Inc., "Pricing and Terms of Service," 2024. [Online]. Available: <https://developer.edamam.com/plans>.
- [44] Spoonacular API, "API Documentation and Usage Guidelines," 2024. [Online]. Available: <https://spoonacular.com/food-api>.
- [45] Spoonacular API, "Pricing and Terms of Use," 2024. [Online]. Available: <https://spoonacular.com/api-pricing>.
- [46] Nutritionix API, "Developer Documentation," 2024. [Online]. Available: <https://developer.nutritionix.com/>.
- [47] Nutritionix API, "Terms of Use and Licensing," 2024. [Online]. Available: <https://www.nutritionix.com/business/api>.
- [48] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [49] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [50] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1996.
- [51] T. Bäck, H.P. Schwefel, "Evolutionary Computation: An Overview," *Proc. IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 20-25, 1997.
- [52] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.
- [53] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, Apr. 2002.

- [54] K. Deb, H. Jain, “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: NSGA-III,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577-601, 2014.
- [55] C. A. Coello Coello, “Evolutionary Multi-Objective Optimization: A Historical View,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28-36, 2006.
- [56] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: John Wiley & Sons, 2001.
- [57] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York, NY: Springer, 2007.
- [58] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA: Springer, 1999.
- [59] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength Pareto evolutionary algorithm,” Technical Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2001.
- [60] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, Dec. 2007.
- [61] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
- [62] D. A. Van Veldhuizen and G. B. Lamont, “Multiobjective evolutionary algorithm research: A history and analysis,” Technical Report TR-98-03, Dept. of Electrical and Computer Engineering, Air Force Institute of Technology, 1998.
- [63] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, Apr. 2003.
- [64] F. Wilhelmstötter, *Jenetics - Java Genetic Algorithm Library*, version 7.0, GitHub repository, 2024.
- [65] F.A. Fortin, F.M. De Rainville, M.A. Gardner, M. Parizeau, C. Gagné, “DEAP: Evolutionary Algorithms Made Easy,” *Journal of Machine Learning Research*, vol. 13, pp. 2171-2175, 2012.
- [66] F. Wilhelmstötter, “Jenetics: Library User’s Manual 8.2.0,” 2024. [Online]. Available: <https://jenetics.io/manual/manual-8.2.0.pdf>.
- [67] R. C. Martin, *Clean Architecture: A Craftsman’s Guide to Software Structure and Design*, 1η έκδ., Boston, MA, USA: Prentice Hall, 2017.
- [68] Google Developers, “Guide to app architecture,” *Android Developers*, 2024. [Online]. Available: <https://developer.android.com/topic/architecture>.
- [69] Google, “Meet Android Studio,” *Android Developers*, 2025. [Online]. Available: <https://developer.android.com/studio/intro>.
- [70] Fork, “Fork - a fast and friendly Git client for Mac and Windows,” 2025. [Online]. Available: <https://git-fork.com/>.
- [71] Google, “Develop Android apps with Kotlin,” *Android Developers*, 2025. [Online]. Available: <https://developer.android.com/kotlin>.

- [72] Google, “Jetpack Compose Overview,” Android Developers, 2025. [Online]. Available: <https://developer.android.com/compose>.
- [73] M. Fowler, “Inversion of Control Containers and the Dependency Injection Pattern,” 2004. [Online]. Available: <https://martinfowler.com/articles/injection.html>.
- [74] Google, “Dependency injection in Android,” Android Developers, 2025. [Online]. Available: <https://developer.android.com/training/dependency-injection>.
- [75] Koin Developers, “Koin - Dependency Injection for Kotlin,” 2025. [Online]. Available: <https://insert-koin.io/>.
- [76] Square, “Retrofit - Introduction,” 2025. [Online]. Available: <https://square.github.io/retrofit/>.
- [77] JetBrains, “Creating and configuring a client,” Ktor Docs, 2025. [Online]. Available: <https://ktor.io/docs/client-create-and-configure.html>.
- [78] Google, “Firebase Documentation,” 2025. [Online]. Available: <https://firebase.google.com/docs>.
- [79] Google, “Room Persistence Library,” Android Developers, 2025. [Online]. Available: <https://developer.android.com/training/data-storage/room>.
- [80] Google Cloud, “Firestore Data Model,” Google Cloud Documentation, 2024. [Online]. Available: <https://cloud.google.com/firestore/docs/data-model>.
- [81] Firebase, “Cloud Firestore Overview,” Google, 2024. [Online]. Available: <https://firebase.google.com/docs/firestore>.
- [82] U.S. Department of Agriculture, “FoodData Central API - SwaggerHub Specification,” *SwaggerHub*, Version 1.0.1, 2024. [Online]. Available: <https://app.swaggerhub.com/apis/fdcnal/food-data-central-api/1.0.1>.
- [83] USDA, “FoodData Central API Documentation,” U.S. Department of Agriculture, 2024. [Online]. Available: <https://fdc.nal.usda.gov/api-guide.html>.
- [84] Android Developers, “Paging 3 Overview,” 2025. [Online]. Available: <https://developer.android.com/topic/libraries/architecture/paging/v3-overview>.
- [85] Android Developers, “WorkManager overview,” Android Jetpack Documentation, 2025. [Online]. Available: <https://developer.android.com/topic/libraries/architecture/workmanager>.
- [86] Android Developers, “AlarmManager”, 2024. [Online]. Available: <https://developer.android.com/reference/android/app/AlarmManager>.
- [87] Android Developers, “BroadcastReceiver”, 2024. [Online]. Available: <https://developer.android.com/reference/android/content/BroadcastReceiver>.
- [88] Kotlin Foundation, Kotlin Coroutines - Asynchronous Programming Documentation, JetBrains, 2024. [Online]. Available: <https://kotlinlang.org/docs/coroutines-overview.html>.
- [89] Kotlin Foundation, Dispatchers and Execution Context, JetBrains, 2024. [Online]. Available: <https://kotlinlang.org/docs/coroutine-context-and-dispatchers.html>.
- [90] T. Mikolov, K. Chen, G. Corrado and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” arXiv preprint arXiv:1301.3781, 2013. [Online]. Available: <https://arxiv.org/abs/1301.3781>.

[91] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv preprint arXiv:1810.04805, 2018. [Online]. Available: <https://arxiv.org/abs/1810.04805>.

ΠΑΡΑΡΤΗΜΑ Α : USDA API /foods/search σώμα αιτήματος

```
{
  "query": "Cheddar cheese",
  "dataType": [
    "Foundation",
    "SR Legacy"
  ],
  "pageSize": 25,
  "pageNumber": 2,
  "sortBy": "dataType.keyword",
  "sortOrder": "asc",
  "brandOwner": "Kar Nut Products Company",
  "tradeChannel": [
    "CHILD_NUTRITION_FOOD_PROGRAMS",
    "GROCERY"
  ],
  "startDate": "2021-01-01",
  "endDate": "2021-12-30"
}
```

ΠΑΡΑΡΤΗΜΑ Β: USDA API /foods/search σώμα απόκρισης

```
{
  "foodSearchCriteria": {
    "query": "Cheddar cheese",
    "dataType": [
      "Foundation",
      "SR Legacy"
    ],
    "pageSize": 25,
    "pageNumber": 2,
    "sortBy": "dataType.keyword",
    "sortOrder": "asc",
    "brandOwner": "Kar Nut Products Company",
    "tradeChannel": [
      "CHILD_NUTRITION_FOOD_PROGRAMS",
      "GROCERY"
    ],
    "startDate": "2021-01-01",
    "endDate": "2021-12-30"
  },
  "totalHits": 1034,
  "currentPage": 0,
  "totalPages": 0,
  "foods": [
    {
      "fdcId": 45001529,
      "dataType": "Branded",
      "description": "BROCCOLI",
      "foodCode": "string",
      "foodNutrients": [
        {
```

```

    "number": 303,
    "name": "Iron, Fe",
    "amount": 0.53,
    "unitName": "mg",
    "derivationCode": "LCCD",
    "derivationDescription": "Calculated from a daily value percentage per serving size measure"
  }
],
"publicationDate": "4/1/2019",
"scientificName": "string",
"brandOwner": "Supervalu, Inc.",
"gtinUpc": "041303020937",
"ingredients": "string",
"ndbNumber": 0,
"additionalDescriptions": "Coon; sharp cheese; Tillamook; Hoop; Pioneer; New York; Wisconsin; Longhorn",
"allHighlightFields": "string",
"score": 0
}
]
}

```

Παράρτημα Γ: Η συνάρτηση buildCodec

```

private fun buildCodec(
    slotPools: List<SlotPool>,
    preferredFoodsCount: Int,
    selectedTypes: List<MealType>
): Codec<List<FoodSelection>, IntegerGene> {
    val allExtendedFoods = slotPools.flatMap { it.extended }.distinct()
    val chromosomes = buildList<IntegerChromosome> {
        slotPools.forEach { sp ->
            val sourceSize = (sp.primary.ifEmpty { sp.extended }).size
            add(IntegerChromosome.of(0, sourceSize - 1, 1))
            add(IntegerChromosome.of(1, 40, 1))
        }
    }
    val flexibleCount = preferredFoodsCount - selectedTypes.size
    if (flexibleCount > 0) {
        val poolSize = allExtendedFoods.size
        add(IntegerChromosome.of(0, poolSize - 1, flexibleCount))
        add(IntegerChromosome.of(1, 40, flexibleCount))
    }
}

```

```

}
val genotype: Genotype<IntegerGene> = Genotype.of(chromosomes)
return Codec.of(genotype) { gt ->
  val selections = mutableListOf<FoodSelection>()
  var chromosomeIdx = 0
  slotPools.forEach { sp ->
    val foodIdxChromosome = gt[chromosomeIdx++] as IntegerChromosome
    val portionChromosome = gt[chromosomeIdx++] as IntegerChromosome
    val foodIdx = foodIdxChromosome.get(0).allele()
    val portion = portionChromosome.get(0).allele() / PORTION_SCALE.toDouble()
    val source = sp.primary.isEmpty { sp.extended }
    val food = source[foodIdx]
    selections += FoodSelection(food, portion, sp.type)
  }
  val flexibleCount = preferredFoodsCount - selectedTypes.size
  if (flexibleCount > 0) {
    val foodIdxChromosome = gt[chromosomeIdx] as IntegerChromosome
    val portionChromosome = gt[chromosomeIdx + 1] as IntegerChromosome
    for (i in 0 until flexibleCount) {
      val foodIdx = foodIdxChromosome.get(i).allele()
      val portion = portionChromosome.get(i).allele() / PORTION_SCALE.toDouble()
      val food = allExtendedFoods[foodIdx]
      selections += FoodSelection(food, portion, food.mealType)
    }
  }
  selections
}
}

```