



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ
ΠΟΛΥΠΡΙΣΜΑΤΙΚΟΥ ΕΡΓΑΛΕΙΟΥ ΛΗΨΗΣ
ΑΠΟΦΑΣΕΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Σταματόπουλου Ευστάθιου

Επιβλέπων: Δημήτριος Ασκούνης

Καθηγητής Ε.Μ.Π.



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΠΟΛΥΠΡΙΣΜΑΤΙΚΟΥ ΕΡΓΑΛΕΙΟΥ ΛΗΨΗΣ ΑΠΟΦΑΣΕΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Σταματόπουλου Ευστάθιου

Επιβλέπων: Δημήτριος Ασκούνης

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 02/09/2021

.....

Δημήτριος Ασκούνης

Καθηγητής ΕΜΠ

.....

Ιωάννης Ψαρράς

Καθηγητής ΕΜΠ

.....

Χρυσόστομος Δούκας

Καθηγητής ΕΜΠ

Αθήνα, Σεπτέμβριος, 2021

.....
Ευστάθιος Σταματόπουλος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ευστάθιος Σταματόπουλος 2021

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στις μέρες μας, πιο πολύ από ποτέ, που το ανθρώπινο δυναμικό, οι οικονομικοί, ενεργειακοί, καθώς και άλλοι πόροι βρίσκονται σε ανεπάρκεια, καλούμαστε να διαχειριστούμε την κατανομή αυτών κατά τον βέλτιστο τρόπο τόσο για το κοινωνικό όσο και για το επιχειρηματικό και το ερευνητικό όφελος. Η επιστήμη, λοιπόν, η οποία ασχολείται με τη λήψη αποφάσεων για την αποδοτική κατανομή πόρων που είναι διαθέσιμοι σε περιορισμένες ποσότητες ονομάζεται Επιχειρησιακή Έρευνα. Γνωρίζοντας σημαντική εξέλιξη από τα μέσα του 20^{ου} αιώνα μέχρι και σήμερα, η εν λόγω επιστήμη καταφέρνει να αναπαραστήσει τις έννοιες αποδοτική κατανομή και ανεπάρκεια σε ένα καλώς ορισμένο μαθηματικό μοντέλο, αλλά και να διατυπώσει τις υπολογιστικές μεθόδους επίλυσης αυτών των μοντέλων.

Η υποστήριξη των μαθηματικών μοντέλων λήψης αποφάσεων από σύγχρονα υπολογιστικά συστήματα αποτελεί πλέον επιτακτική ανάγκη λόγω του μεγάλου όγκου δεδομένων που είναι προς ανάλυση σε συνδυασμό με την ταχύτητα υπολογισμού των συστημάτων αυτών. Στο πλαίσιο αυτό, η παρούσα διπλωματική εργασία επιχειρεί αρχικά να αναλύσει ορισμένες βασικές μεθόδους της επιχειρησιακής έρευνας καθώς και τις ήδη υπάρχουσες τεχνολογικές λύσεις που σχετίζονται με αυτές. Στη συνέχεια, στόχο έχει τον σχεδιασμό και την υλοποίηση ενός εργαλείου λήψης αποφάσεων το οποίο θα κάνει χρήση των κατάλληλων αλγορίθμων και θα μπορεί να παράγει αποτελέσματα, αξιοποιώντας δεδομένα από διαφορετικές πηγές.

Στο παρόν σύγγραμμα παρουσιάζονται τα εργαλεία λογισμικού που χρησιμοποιήθηκαν για την κατασκευή μιας διαδικτυακής εφαρμογής στο φιλικό περιβάλλον της οποίας ένας χρήστης θα μπορεί να επιλύει προβλήματα δέντρων αποφάσεων, γραμμικού προγραμματισμού και δυναμικού προγραμματισμού. Στη συνέχεια, αναλύονται οι σχεδιαστικές επιλογές και υλοποιήσεις που πραγματοποιήθηκαν για την κατασκευή της διαδρομής των δεδομένων από τη συλλογή τους από τον χρήστη μέχρι την απεικόνιση των αποτελεσμάτων. Ακολουθεί η περιγραφή και επεξήγηση των βασικών οθονών και των λειτουργιών που επιτελεί το εργαλείο λήψης αποφάσεων στην τελική του μορφή, συνοδεία των σεναρίων χρήσης.

Τέλος, αναλύονται τα αποτελέσματα και τα συμπεράσματα τα οποία εξάχθηκαν από την εκπόνηση της διπλωματικής εργασίας, τα πλεονεκτήματα και μειονεκτήματα του εργαλείου καθώς και προτάσεις για τη μελλοντική του ανάπτυξη.

Λέξεις κλειδιά: Επιχειρησιακή έρευνα, λήψη αποφάσεων, δέντρα αποφάσεων, γραμμικός προγραμματισμός, δυναμικός προγραμματισμός

Abstract

Nowadays, it is well understood that human, fiscal, energy and other resources are being depleted. As a result, optimal distribution of the available resources is of utmost importance, in order to promote financial growth, research potential and social welfare. Operational research is the scientific discipline that focuses on decision making for efficient distribution of limited resources. Having undergone significant advances since the mid-twentieth century, this field represents the concept of efficient distribution and deficiency within the confines of a well-defined mathematical model, as well as describes computational methods of solving these models.

Supporting decision making mathematical models via contemporary computational systems is crucial in the era of big data analysis and high speed processing. The aim of the following thesis is, firstly, the analysis of basic operational research methods and its associated, currently existing, technological solutions. Secondly, this thesis attempts to design and implement a decision making tool which will be using appropriate algorithms incorporating data from various sources in order to produce relevant results.

This manuscript presents the software tools used to construct a user-friendly network application, via which the user will be able to solve decision tree, linear programming and dynamic programming problems. It further describes the designing choices and implementations that took place to construct the data pathway from collection by the user to the visualization of results. These are followed by the outline and explanation of the basic screens and functions performed by the decision making tool, in its final form, along with the scenarios of use. Finally, this work summarizes the results and conclusions extracted by the carryout of this thesis, the benefits and drawbacks of the tool in discussion and potential suggestions for its future development.

Keywords: Operations research, decision making, decision trees, linear programming, dynamic programming

Πρόλογος

Η παρούσα Διπλωματική Εργασία εκπονήθηκε στο πλαίσιο ολοκλήρωσης του προπτυχιακού κύκλου σπουδών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Η εργασία ανατέθηκε από το Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης και έχει ως θέμα την ανάπτυξη ενός πολυκριματικού εργαλείου λήψης αποφάσεων το οποίο θα επιλύει προβλήματα συστημάτων αποφάσεων.

Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής η οποία θα συλλέγει τα δεδομένα ενός προβλήματος αποφάσεων από τον χρήστη, θα εφαρμόζει τις κατάλληλες μεθόδους σε αυτά για την επίλυσή του και, τέλος, θα παρουσιάζει ολοκληρωμένη τη λύση. Απώτερος σκοπός, η ύπαρξη ενός, ανοιχτού κώδικα, εργαλείου λήψης αποφάσεων το οποίο θα καλύπτει έναν σημαντικό αριθμό μεθόδων της επιχειρησιακής έρευνας το οποίο θα μπορεί να χρησιμοποιηθεί σε διάφορους τομείς όπου απαιτείται η λήψη αποφάσεων.

Πίνακας Περιεχομένων

1	ΕΙΣΑΓΩΓΗ	19
1.1	ΑΝΤΙΚΕΙΜΕΝΟ ΚΑΙ ΣΚΟΠΟΣ	19
1.2	ΦΑΣΕΙΣ ΥΛΟΠΟΙΗΣΗΣ	21
1.3	ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ	23
2	ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	25
2.1	ΣΥΣΤΗΜΑΤΑ ΑΠΟΦΑΣΕΩΝ	25
2.2	ΔΕΝΤΡΑ ΑΠΟΦΑΣΕΩΝ	26
2.3	ΓΡΑΜΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	28
2.3.1	Γραφική Επίλυση	29
2.3.2	Μέθοδος Simplex	29
2.4	ΔΥΝΑΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	33
2.4.1	Εύρεση Βέλτιστης Διαδρομής	33
2.4.2	Προγραμματισμός Επενδύσεων	34
2.4.3	Προγραμματισμός Παραγωγής και Αποθήκευσης	36
3	ΣΥΓΧΡΟΝΕΣ ΣΧΕΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ	39
3.1	ΣΥΓΧΡΟΝΕΣ ΕΦΑΡΜΟΓΕΣ	39
3.1.1	Lucidchart – Decision Tree Maker	39
3.1.2	WolframAlpha - Linear Programming Solver	39
3.1.3	Visual LinProg	40
3.1.4	Online Calculator	40
3.1.5	DynaProg	40
3.2	ΣΥΜΒΟΛΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	41
4	ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	43
4.1	ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ VISUAL STUDIO CODE	43
4.2	Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PYTHON	43
4.3	DJANGO WEB FRAMEWORK	44
4.4	ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ POSTGRESQL	46
4.5	ΟΙ ΓΛΩΣΣΕΣ ΙΣΤΟΥ HTML, CSS ΚΑΙ JAVASCRIPT	47
4.6	Η ΒΙΒΛΙΟΘΗΚΗ ΟΠΤΙΚΟΠΟΙΗΣΗΣ VISJS	48
4.7	Η ΒΙΒΛΙΟΘΗΚΗ ΟΠΤΙΚΟΠΟΙΗΣΗΣ CHART.JS	48
4.8	ΤΟ ΠΡΟΤΥΠΟ BLACK DASHBOARD	48
5	ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ	51
5.1	ΕΙΣΑΓΩΓΗ	51
5.2	ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΩΝ ΤΗΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ	51
5.3	ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΒΑΣΙΚΗΣ ΕΦΑΡΜΟΓΗΣ	53
5.3.1	Μοντέλα Δεδομένων	53
5.3.2	Λειτουργικότητα	54
5.3.3	Διασύνδεση Λειτουργικότητας με τον χρήστη	55
5.4	ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΕΩΝ	55
5.4.1	Συλλογή, Επεξεργασία και Αποθήκευση Δεδομένων	56
5.4.2	Λειτουργικότητα της Εφαρμογής	59
5.4.3	Διασύνδεση Λειτουργικότητας με τον χρήστη	61

5.5	ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΓΡΑΜΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	62
5.5.1	<i>Συλλογή, Επεξεργασία και Αποθήκευση Δεδομένων.....</i>	63
5.5.2	<i>Λειτουργικότητα της Εφαρμογής</i>	65
5.5.3	<i>Διασύνδεση Λειτουργικότητας με τον χρήστη</i>	67
5.6	ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΔΥΝΑΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	69
5.6.1	<i>Συλλογή, Επεξεργασία και Αποθήκευση Δεδομένων.....</i>	69
5.6.2	<i>Λειτουργικότητα της Εφαρμογής</i>	76
5.6.3	<i>Διασύνδεση Λειτουργικότητας με τον χρήστη</i>	78
6	ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ	81
6.1	ΒΑΣΙΚΗ ΕΦΑΡΜΟΓΗ.....	81
6.1.1	<i>Αρχική Οθόνη</i>	81
6.1.2	<i>Οθόνη Πληροφοριών</i>	82
6.1.3	<i>Οθόνη Οδηγιών</i>	83
6.2	ΕΦΑΡΜΟΓΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΕΩΝ.....	84
6.2.1	<i>Αρχική Οθόνη</i>	84
6.2.2	<i>Οθόνη Δημιουργίας Δέντρου Αποφάσεων</i>	85
6.2.3	<i>Οθόνη Ανεβάσματος Αρχείου</i>	94
6.2.4	<i>Οθόνη Επεξεργασίας Δέντρων Αποφάσεων</i>	98
6.2.5	<i>Οθόνη Αποτελεσμάτων</i>	100
6.3	ΕΦΑΡΜΟΓΗ ΓΡΑΜΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	102
6.3.1	<i>Αρχική Οθόνη</i>	102
6.3.2	<i>Οθόνη Δημιουργία Προβλήματος Γραμμικού Προγραμματισμού</i>	103
6.3.3	<i>Οθόνη Ανεβάσματος Αρχείου</i>	109
6.3.4	<i>Οθόνη Επεξεργασίας Δεδομένων Γραμμικού Προγραμματισμού</i>	110
6.3.5	<i>Οθόνη Αποτελεσμάτων</i>	111
6.4	ΕΦΑΡΜΟΓΗ ΔΥΝΑΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	115
6.4.1	<i>Αρχική Οθόνη</i>	115
6.4.2	<i>Οθόνη Δημιουργίας Προβλήματος Δυναμικού Προγραμματισμού</i>	116
6.4.3	<i>Οθόνη Ανεβάσματος Αρχείου</i>	126
6.4.4	<i>Οθόνη Επεξεργασίας Δεδομένων Δυναμικού Προγραμματισμού</i>	128
6.4.5	<i>Οθόνη Αποτελεσμάτων</i>	129
7	ΣΥΜΠΕΡΑΣΜΑΤΑ, ΠΕΡΙΟΡΙΣΜΟΙ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΜΕΛΛΟΝΤΙΚΗΣ ΕΞΕΛΙΞΗΣ	135
7.1	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	135
7.2	ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ ΥΛΟΠΟΙΗΣΗΣ	137
7.3	ΠΕΡΙΟΡΙΣΜΟΙ	138
7.3.1	<i>Περιορισμοί Συνολική Εφαρμογής.....</i>	138
7.3.2	<i>Περιορισμοί Εφαρμογής Δέντρων Αποφάσεων</i>	138
7.3.3	<i>Περιορισμοί Εφαρμογής Γραμμικού Προγραμματισμού</i>	139
7.3.4	<i>Περιορισμοί Εφαρμογής Δυναμικού Προγραμματισμού</i>	139
7.4	ΠΡΟΤΑΣΕΙΣ ΕΞΕΛΙΞΗΣ.....	139
7.4.1	<i>Προτάσεις σχετικά με το περιεχόμενο</i>	140
7.4.2	<i>Προτάσεις σχετικά με τη λειτουργικότητα.....</i>	140
8	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	143

Πίνακας Εκθεμάτων

Εικόνα 1- 1 ^η Φάση Υλοποίησης.....	22
Εικόνα 2- 2 ^η Φάση Υλοποίησης.....	22
Εικόνα 3- 3 ^η Φάση Υλοποίησης.....	23
Εικόνα 4- Λειτουργία ενός συστήματος.....	25
Εικόνα 5- Παράδειγμα δέντρου αποφάσεων	27
Εικόνα 6- Πίνακας Simplex	30
Εικόνα 7- Πίνακας επίλυσης προβλήματος προγραμματισμού επενδύσεων	35
Εικόνα 8- Τρόπος διαχείρισης αιτημάτων στο Django.....	45
Εικόνα 9- Αλγόριθμος επίλυσης δέντρων αποφάσεων	58
Εικόνα 10- Αλγόριθμος simplex	64
Εικόνα 11- Αλγόριθμος εύρεσης βέλτιστης διαδρομής.....	71
Εικόνα 12- Αλγόριθμος επίλυσης προβλημάτων προγραμματισμού επενδύσεων	73
Εικόνα 13- Αλγόριθμος επίλυσης προβλημάτων προγραμματισμού παραγωγής και αποθήκευσης	75
Εικόνα 14- Αρχική οθόνη συνολικής εφαρμογής.....	82
Εικόνα 15- Οθόνη πληροφοριών.....	83
Εικόνα 16- Οθόνη οδηγιών	84
Εικόνα 17- Αρχική οθόνη εφαρμογής δέντρων αποφάσεων.....	85
Εικόνα 18- Οθόνη δημιουργίας δέντρου αποφάσεων	86
Εικόνα 19- Επιλογές επεξεργασίας δέντρου αποφάσεων	87
Εικόνα 20- Προσθήκη κόμβου.....	88
Εικόνα 21- Προσθήκη κόμβου φύλλο	88
Εικόνα 22- Προσθήκη κόμβου ευκαιρίας.....	89
Εικόνα 23- Προσθήκη κόμβου απόφασης	89
Εικόνα 24- Προσθήκη κόμβου μελέτης.....	89
Εικόνα 25- Χάραξη ακμής.....	91
Εικόνα 26- Προσθήκη απλής ακμής.....	92
Εικόνα 27- Προσθήκη ακμής αβέβαιου γεγονότος.....	93
Εικόνα 28- Παράδειγμα δέντρου αποφάσεων χωρίς μελέτη	93
Εικόνα 29- Εφαρμογή δέντρων αποφάσεων: Οθόνη ανεβάσματος αρχείου	94
Εικόνα 30- Εφαρμογή δέντρων αποφάσεων: Επιλογή αρχείου	95
Εικόνα 31- Εφαρμογή δέντρων αποφάσεων: Σφάλμα τύπου αρχείου εισόδου	96
Εικόνα 32- Σφάλμα περιεχομένων αρχείου εισόδου	97
Εικόνα 33- Εφαρμογή δέντρων αποφάσεων: Φόρμα ανεβάσματος αρχείου συμπληρωμένη	97
Εικόνα 34- Εφαρμογή δέντρων αποφάσεων: Οθόνη επεξεργασίας	98
Εικόνα 35- Προσθήκη ακμής μελέτης	99
Εικόνα 36- Παράδειγμα δέντρων αποφάσεων με μελέτη	100
Εικόνα 37- Εφαρμογή δέντρων αποφάσεων: Οθόνη αποτελεσμάτων – Παράδειγμα χωρίς μελέτη.....	101
Εικόνα 38- Εφαρμογή δέντρων αποφάσεων: Οθόνη αποτελεσμάτων – Παράδειγμα με μελέτη.....	101

Εικόνα 39 - Εφαρμογή γραμμικού προγραμματισμού: Αρχική οθόνη	102
Εικόνα 40- Εφαρμογή γραμμικού προγραμματισμού: Δημιουργία προβλήματος	103
Εικόνα 41- Εφαρμογή γραμμικού προγραμματισμού: Επιλογή κριτηρίου επίλυσης	104
Εικόνα 42- Εφαρμογή γραμμικού προγραμματισμού: Συμπλήρωση δεδομένων αντικειμενικής συνάρτησης	104
Εικόνα 43- Εφαρμογή γραμμικού προγραμματισμού: Κατασκευή περιορισμών	106
Εικόνα 44- Εφαρμογή γραμμικού προγραμματισμού: Κατασκευή δραστηριοτήτων	106
Εικόνα 45- Παράδειγμα προβλήματος δύο μεταβλητών	107
Εικόνα 46- Εφαρμογή γραμμικού προγραμματισμού: Λανθασμένη συμπλήρωση και εκτέλεση	108
Εικόνα 47- Εφαρμογή γραμμικού προγραμματισμού: Σφλαμα μη υποστηριζόμενου πλήθους μεταβλητών απόφασης.....	108
Εικόνα 48- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη ανεβάσματος αρχείου	109
Εικόνα 49- Εφαρμογή γραμμικού προγραμματισμού: Φόρμα ανεβάσματος αρχείου συμπληρωμένη.....	110
Εικόνα 50- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη επεξεργασίας	111
Εικόνα 51- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη αποτελεσμάτων – γραφική επίλυση.....	112
Εικόνα 52- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη αποτελεσμάτων –απεικόνιση σημείων	113
Εικόνα 53- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη αποτελεσμάτων –μέθοδος simplex.....	114
Εικόνα 54- Εφαρμογή δυναμικού προγραμματισμού: Αρχική οθόνη	115
Εικόνα 55- Εφαρμογή δυναμικού προγραμματισμού: Δημιουργία προβλήματος	116
Εικόνα 56- Εφαρμογή δυναμικού προγραμματισμού: Επιλογή τύπου προβλήματος	117
Εικόνα 57- Εφαρμογή δυναμικού προγραμματισμού: Δημιουργία προβλήματος βέλτιστης διαδρομής	118
Εικόνα 58- Εφαρμογή δυναμικού προγραμματισμού: Προσθήκη κόμβου	119
Εικόνα 59- Εφαρμογή δυναμικού προγραμματισμού: Προσθήκη ακμής	119
Εικόνα 60- - Εφαρμογή δυναμικού προγραμματισμού: Παράδειγμα βέλτιστης διαδρομής	120
Εικόνα 61- Εφαρμογή δυναμικού προγραμματισμού: Δημιουργία προβλήματος προγραμματισμού επενδύσεων	121
Εικόνα 62- Συμπλήρωση δεδομένων προβλήματος προγραμματισμού επενδύσεων.....	122
Εικόνα 63- Εφαρμογή δυναμικού προγραμματισμού: Σφάλμα λανθασμένης συμπλήρωσης	122
Εικόνα 64- Παράδειγμα προβλήματος προγραμματισμού επενδύσεων	123
Εικόνα 65- Εφαρμογή δυναμικού προγραμματισμού: Δημιουργία προβλήματος προγραμματισμού παραγωγής και αποθήκευσης	124
Εικόνα 66- Συμπλήρωση δεδομένων προβλήματος προγραμματισμού παραγωγής και αποθήκευσης	125
Εικόνα 67- Παράδειγμα προβλήματος προγραμματισμού παραγωγής και αποθήκευσης	126
Εικόνα 68- Εφαρμογή δυναμικού προγραμματισμού: Οθόνη ανεβάσματος αρχείου	127

Εικόνα 69- Εφαρμογή δυναμικού προγραμματισμού: Οθόνη ανεβάσματος αρχείου – Επιλογή τύπου προβλήματος.....	128
Εικόνα 70- Εφαρμογή δυναμικού προγραμματισμού: Οθόνη επεξεργασίας	129
Εικόνα 71- Εφαρμογή δυναμικού προγραμματισμού: Οθόνη αποτελεσμάτων – παράδειγμα βέλτιστης διαδρομής.....	130
Εικόνα 72- - Εφαρμογή δυναμικού προγραμματισμού: οθόνη αποτελεσμάτων – παράδειγμα προγραμματισμού επενδύσεων.....	131
Εικόνα 73- - Εφαρμογή δυναμικού προγραμματισμού: Οθόνη αποτελεσμάτων - παράδειγμα προγραμματισμού παραγωγής και αποθήκευσης.....	132

1 Εισαγωγή

1.1 Αντικείμενο και Σκοπός

Η επιχειρησιακή έρευνα αποτελεί μια επιστημονική προσέγγιση στην ανάλυση προβλημάτων και στη λήψη αποφάσεων [1]. Ως επιστήμη στοχεύει στην παροχή ορθολογικών βάσεων για τη λήψη αποφάσεων. Αυτό επιτυγχάνεται αναζητώντας την πλήρη κατανόηση και τη δόμηση σύνθετων καταστάσεων ώστε να επιτευχθεί η πρόβλεψη της λειτουργίας του εκάστοτε συστήματος και κατ' επέκταση τη βελτιστοποίηση αυτού. Τα παραπάνω επιτυγχάνονται με τη χρήση αναλυτικών και αριθμητικών τεχνικών για την ανάπτυξη μαθηματικών και υπολογιστικών μοντέλων από οργανωτικά συστήματα που αποτελούνται συνήθως από ανθρώπους, μηχανές και διαδικασίες [2].

Μέχρι και σήμερα είναι αμφιλεγόμενο και αμφίβολο το που οφείλει τις ρίζες της η επιχειρησιακή έρευνα, στη σύγχρονή της μορφή. Ωστόσο η επικρατέστερη άποψη είναι ότι αυτή αναπτύχθηκε κατά τη διάρκεια του Δεύτερου Παγκοσμίου Πολέμου από τον Βρετανικό Στρατό. Σε αρχικό επίπεδο χρησιμοποιήθηκε ως σύμβουλος για την αποδοτικότερη και αποτελεσματικότερη διαχείριση του εξοπλισμού και του προσωπικού. Η εξέλιξη όμως, καθώς και η επιβεβαίωση της σημαντικότητας και της αποτελεσματικότητας της επιχειρησιακής έρευνας οφείλεται στη συνεργασία της Βασιλικής Αεροπορίας με το Τμήμα Επιχειρησιακής Έρευνας του Υπουργείου Αεροπορίας για την κατάστρωση αμυντικών και επιθετικών στρατηγικών. Σε αυτό το σημείο αξίζει να αναφερθεί πως στο Τμήμα Επιχειρησιακής Έρευνας σημαντικό ρόλο κατείχε ο, κατά πολλούς, πατέρας της επιχειρησιακής έρευνας, P. M. S. Blackett [3]. Με την ολοκλήρωση του Δεύτερου Παγκοσμίου Πολέμου, η εμφανής προοπτική της επιχειρησιακής έρευνας σε συνδυασμό με την ανάπτυξη των ηλεκτρονικών υπολογιστών έκανε πολλούς επιστήμονες να ασχοληθούν εντατικότερα με το αντικείμενό της [4]. Με αυτό τον τρόπο αναπτύχθηκε σε επιστημονικό, ερευνητικό αλλά και επιχειρηματικό επίπεδο διότι πολλές θεωρίες της επιχειρησιακής έρευνας βρήκαν εφαρμογή και έλυσαν πολλά προβλήματα της εποχής και συνεχίζουν να είναι συμβατές και να αναπτύσσονται μέχρι και σήμερα.

Αν και η επιχειρησιακή έρευνα αποτελεί μια ξεχωριστή επιστήμη από μόνη της, έχει θεωρηθεί αναπόσπαστο κομμάτι της Βιομηχανικής Μηχανικής. Αυτό δεν προκαλεί έκπληξη όταν αναλογιστεί κανείς ότι και τα δύο αντικείμενα μοιράζονται πολλούς κοινούς στόχους, τεχνικές και σενάρια εφαρμογής. Αν και ο ρυθμός ανάπτυξης των μεθόδων της επιχειρησιακής έρευνας είχε επιβραδυνθεί, σήμερα αποτελεί ένα καλά ανεπτυγμένο πεδίο με μια εξελιγμένη σειρά τεχνικών που χρησιμοποιούνται τακτικά για την επίλυση και τη βελτιστοποίηση προβλημάτων σε ένα ευρύ φάσμα τομέων εφαρμογής [5]. Επίσης, μέθοδοι αυτής διδάσκονται στα πανεπιστήμια διότι οι

αλγόριθμοι και ο τρόπος προσέγγισής τους αποτελούν βασικοί για την επίλυση διάφορων τύπων προβλημάτων. Συγκεκριμένα, στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών (ΣΗΜΜΥ) του Εθνικού Μετσόβιου Πολυτεχνείου (ΕΜΠ), η επιστήμη της επιχειρησιακής έρευνας διδάσκεται στο μάθημα Συστήματα Αποφάσεων [6] της ροής Διοίκησης και Απόφασης.

Στο πλαίσιο του παραπάνω μαθήματος, η παρούσα διπλωματική εργασία επιχειρεί την ανάπτυξη μιας διαδικτυακής εφαρμογής η οποία θα υλοποιεί τους αλγόριθμους και τις μεθόδους που διδάσκονται σε αυτό. Ειδικότερα, αυτή απαρτίζεται από τρεις εφαρμογές την Εφαρμογή Δέντρων Αποφάσεων, την Εφαρμογή Γραμμικού Προγραμματισμού και την Εφαρμογή Δυναμικού Προγραμματισμού. Τα Δέντρα Αποφάσεων, ο Γραμμικός Προγραμματισμός και ο Δυναμικός Προγραμματισμός αποτελούν τις μεθόδους της επιχειρησιακής έρευνας στις οποίες εστιάζει το μάθημα της σχολής και των οποίων την υλοποίηση αναλαμβάνει η εν λόγω εφαρμογή. Στη συνέχεια δίνονται ορισμένες πληροφορίες για την κάθε περίπτωση.

- **Εφαρμογή Δέντρων Αποφάσεων:** Αποτελεί μια εφαρμογή όπου ο χρήστης θα μπορεί, παρέχοντας με διάφορους τρόπους τα δεδομένα του, όχι μόνο να λύνει σύνθετα προβλήματα Δέντρων Αποφάσεων, αλλά και να βλέπει τον τρόπο με τον οποίο προέκυψε η λύση του προβλήματος. Αξίζει να σημειωθεί πως η εφαρμογή θα παρέχει στον χρήστη ένα εύχρηστο εργαλείο σχεδιασμού ενός δέντρου αποφάσεων.
- **Εφαρμογή Γραμμικού Προγραμματισμού:** Αποτελεί μια εφαρμογή όπου θα επιλύει προβλήματα Γραμμικού Προγραμματισμού. Συγκεκριμένα, ο χρήστης θα μπορεί είτε να παρέχει τα δεδομένα του μέσω αρχείου είτε να κατασκευάζει το πρόβλημά του πάνω στην εφαρμογή. Στη συνέχεια, ανάλογα με την περίπτωση του προβλήματος θα του παρουσιάζεται ο τρόπος λύσης, είτε γραφικά, είτε υπολογιστικά.
- **Εφαρμογή Δυναμικού Προγραμματισμού:** Αποτελεί μια εφαρμογή που θα μπορεί να επιλύσει τριών ειδών προβλήματα Δυναμικού Προγραμματισμού: βέλτιστη διαδρομή, προγραμματισμό επενδύσεων και προγραμματισμό παραγωγής και αποθήκευσης. Ο χρήστης θα μπορεί λοιπόν να δημιουργήσει ένα από τα παραπάνω προβλήματα. Έπειτα η εφαρμογή αναλαμβάνει την επίλυση αυτού και την παρουσίαση τόσο της λύσης, όσο και των επαναλήψεων που οδήγησαν σε αυτή.

Τα παραπάνω εργαλεία θα συγκεντρώνονται σε μια σύγχρονη και φιλική στη χρήση εφαρμογή, στόχος της οποίας είναι να χρησιμοποιηθεί για εκπαιδευτικό σκοπό, ως συνοδευτικό εργαλείο εκπαίδευσης του μαθήματος Συστήματα Αποφάσεων. Σημαντικό χαρακτηριστικό της είναι, επίσης, ότι θα σχεδιαστεί και θα υλοποιηθεί με τέτοιο τρόπο ώστε να είναι εύκολα επεκτάσιμη, ώστε να μπορούν στο μέλλον να προστεθούν επιπλέον εργαλεία που θα υλοποιούν άλλες μεθόδους της επιχειρησιακής έρευνας, εφόσον αυτό κριθεί αναγκαίο. Τέλος, αξίζει να σημειωθεί πως η εφαρμογή

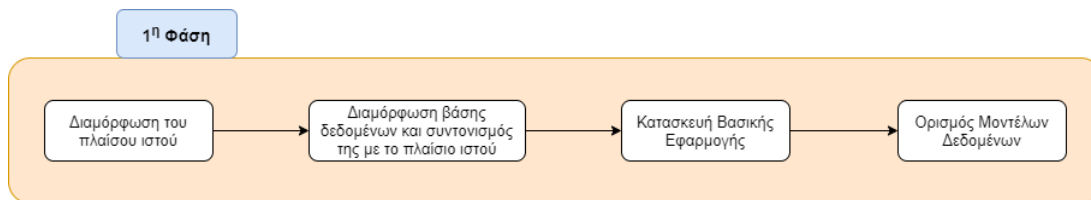
δεν θα εστιάζει στον τύπο των αριθμητικών δεδομένων που παρέχει ο χρήστης ή την προέλευση αυτών, αλλά θα επικεντρώνεται στην επίλυση του εκάστοτε προβλήματος, με μόνη προϋπόθεση την ορθότητα και την πληρότητα των αριθμητικών τιμών τους.

1.2 Φάσεις Υλοποίησης

Πριν ξεκινήσει η υλοποίηση, συζητήθηκαν σε θεωρητικό επίπεδο τα περιεχόμενα και τα σενάρια χρήσης της εφαρμογής που είναι προς κατασκευή. Αφού αναλύθηκαν τα περιεχόμενα του μαθήματος Συστημάτων Αποφάσεων της ΣΗΜΜΥ του ΕΜΠ, επιλέχθηκαν οι μέθοδοι της επιχειρησιακής έρευνας που θα υλοποιήσει η εφαρμογή. Η επιλογή βασίστηκε τόσο στο κατά πόσο εκτεταμένα αναλύονται οι θεωρίες τους στο μάθημα, αλλά και στην επάρκεια των δεδομένων εισόδου ώστε να ελεγχθεί η αξιοπιστία της υλοποίησης. Λαμβάνοντας τα παραπάνω υπόψιν, αποφασίστηκε η εφαρμογή να εστιάζει στην επίλυση προβλημάτων Δέντρων Αποφάσεων, Γραμμικού Προγραμματισμού και Δυναμικού Προγραμματισμού. Αφού αποφασίστηκε το αντικείμενο της εφαρμογής, μελετήθηκαν σχολαστικά οι αλγόριθμοι που αυτοί θα υλοποιεί. Το στάδιο της μελέτης είναι κι ένα από τα θεμελιώδη καθώς παρείχε την απαραίτητη γνώση ώστε να κατασκευαστούν στην συνέχεια ορθά οι αλγόριθμοι υπολογισμού των μεθόδων. Τέλος πραγματοποιήθηκε η επιλογή του λογισμικού και των εργαλείων που θα οδηγούσαν στην κατασκευή ενός αποδοτικού και επεκτάσιμου εργαλείου λήψης αποφάσεων.

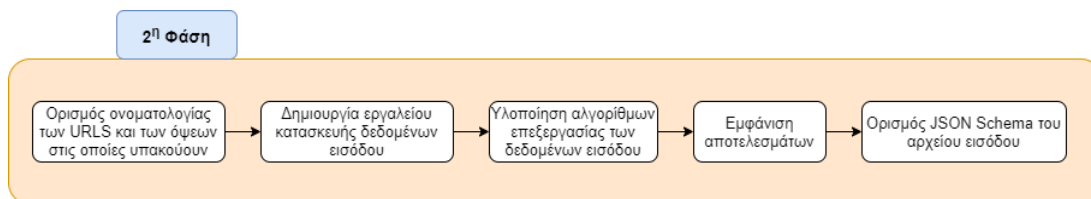
Η εκπόνηση της διπλωματικής εργασίας πραγματοποιήθηκε μεταξύ Οκτωβρίου 2020 και Αυγούστου 2021 και οι φάσεις υλοποίησης παρουσιάζονται παρακάτω.

1^η Φάση Υλοποίησης: Όπως φαίνεται και στην *Εικόνα 1* η πρώτη φάση υλοποίησης εκκινεί με την διαμόρφωση και την κατάλληλη ρύθμιση του έργου στο πλαίσιο ιστού, ώστε να μπορούν με ευκολία να οριστούν και να αναπτυχθούν οι εφαρμογές που απαρτίζουν τη συνολική διαδικτυακή εφαρμογή. Επόμενο βήμα είναι η κατασκευή της βάσης δεδομένων και η διασύνδεσή της με τη διαδικτυακή εφαρμογή. Αφού ολοκληρωθεί με επιτυχία η σύνδεση και η επικοινωνία του εξυπηρετητή με τη βάση δεδομένων, κατασκευάζεται η βασική εφαρμογή μέσω της οποίας θα πραγματοποιείται η πλοήγηση του χρήστη στην εφαρμογή. Τελευταίο στάδιο της πρώτης φάσης είναι ο ορισμός των μοντέλων δεδομένων, τα οποία αποτελούν και τους πίνακες στη βάση δεδομένων.



Εικόνα 1- 1^η Φάση Υλοποίησης

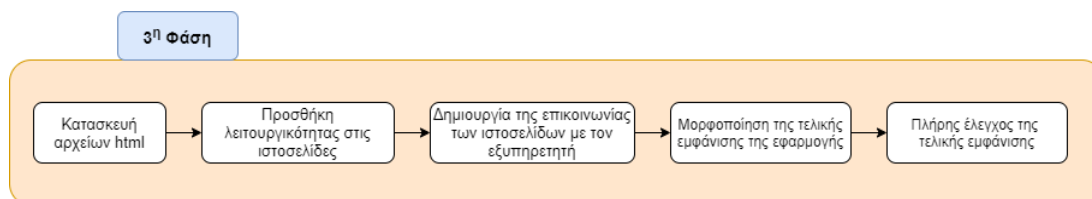
2^η Φάση Υλοποίησης: Σε αυτή τη φάση την υλοποίησης κατασκευάζονται κατά κύριο λόγο οι λειτουργικότητες του εξυπηρετητή. Δηλαδή, ορίζεται και υλοποιείται η διαδικασία συλλογής, επεξεργασίας και αποθήκευσης των δεδομένων. Η διαδικασία που αναλύεται σε αυτή την παράγραφο και απεικονίζεται στην *Εικόνα 2*, επαναλαμβάνεται για κάθε μια από τις εφαρμογές Δέντρων Αποφάσεων, Γραμμικού Προγραμματισμού και Δυναμικού Προγραμματισμού. Αρχικά, ορίζονται οι όψεις – συναρτήσεις που αποτελούν τη λειτουργικότητα του εξυπηρετητή – και οι Ένιαίοι Εντοπιστές Πόρων (URLs) που αποτελούν τις διευθύνσεις στις οποίες θα αναζητεί ο εξυπηρετητής να αποδώσει την αντίστοιχη λειτουργικότητα της όψης. Στη συνέχεια κατασκευάζεται το εργαλείο πάνω στο οποίο ο χρήστης θα μπορεί να συμπληρώσει τα δεδομένα του. Εκτός από αυτή του τη χρήση είναι και απαραίτητο για τη συνέχεια της υλοποίησης ώστε να δημιουργηθεί η διαδικασία συλλογής και επεξεργασίας αυτών. Επόμενο βήμα είναι η κατασκευή των αλγορίθμων επεξεργασίας των δεδομένων εισόδου. Οι αλγόριθμοι αυτοί, ανάλογα με την εφαρμογή, επιλύουν και το αντίστοιχο πρόβλημα επιχειρησιακής έρευνας. Έπειτα ακολουθεί η αποθήκευση των αποτελεσμάτων ώστε να είναι διαθέσιμα προς εμφάνιση. Σε αυτό το σημείο έχει ολοκληρωθεί η υλοποίηση ενός κύκλου διεργασιών που οδηγεί στη συλλογή, επεξεργασία και αποθήκευση των δεδομένων στον εξυπηρετητή. Ωστόσο, επειδή είναι επιθυμητό ο χρήστης να έχει και την επιλογή να παρέχει τα δεδομένα του μέσω αρχείου, ορίζεται το σχήμα JSON στο οποίο θα πρέπει να υπακούουν τα δεδομένα του ώστε να μην προκύπτουν προβλήματα κατά την επεξεργασία.



Εικόνα 2- 2^η Φάση Υλοποίησης

3^η Φάση Υλοποίησης: Η φάση αυτή επικεντρώνεται στη διεπαφή του χρήστη με την εφαρμογή και την προσφορά μιας διαδραστικής και φιλικής προς αυτόν εφαρμογής. Η 3^η Φάση Υλοποίησης επαναλαμβάνεται για κάθε μέθοδο και κατ' επέκταση εφαρμογή που δημιουργήθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας. Εκτός από την ανάλυση των σταδίων της φάσης που ακολουθεί, στην *Εικόνα 3* γίνεται

μια συνοπτική παρουσίαση αυτών. Πρώτο βήμα είναι η κατασκευή των αντικειμένων της κάθε σελίδας που θα έχει πρόσβαση ο χρήστης. Έπειτα προστέθηκε λειτουργικότητα σε κάθε μια από αυτές, ώστε ο χρήστης να μπορεί να πραγματοποιήσει τις ενέργειες που του επιτρέπονται και επιθυμεί. Στη συνέχεια ελέγχεται η διασύνδεση των σελίδων με την εκάστοτε όψη στον εξυπηρετητή. Ένα από τα σημαντικότερα βήματα της φάσης αυτή είναι η μορφοποίηση της τελικής εμφάνισης των σελίδων της εφαρμογής. Σε αυτό το βήμα, δίνεται μεγάλη έμφαση στην ωραιότητα και στη φιλικότητα του κάθε στοιχείου της σελίδας ώστε να προκύψει εν τέλη ένα καλαίσθητο και ευχάριστο περιβάλλον εργασίας για τον χρήστη. Τέλος, αφού ολοκληρώθηκε το αισθητικό κομμάτι της εφαρμογής, πραγματοποιήθηκε λεπτομερής έλεγχος για την ύπαρξη συνέπειας και αντιστοιχίας μεταξύ των εφαρμογών που δημιουργήθηκαν.



Εικόνα 3- 3^η Φάση Υλοποίησης

Αφού ολοκληρώθηκαν οι υλοποιήσεις των εφαρμογών πραγματοποιήθηκαν δοκιμές για την επιβεβαίωση της συνολικά ορθής λειτουργίας της εφαρμογής και τη διόρθωση τυχόν λαθών. Για αυτό το λόγο, με τη χρήση δεδομένων εισόδου από τις ασκήσεις του μαθήματος Συστημάτων Αποφάσεων [7] [8] [9] [10] επιλύθηκαν πραγματικά προβλήματα επιχειρησιακής έρευνας και διορθώθηκαν τα όποια λάθη που προέκυψαν. Μετά και την ολοκλήρωση των δοκιμών και των τελευταίων διορθώσεων η εφαρμογή είναι έτοιμη προς χρήση.

1.3 Οργάνωση του Τόμου

Το σύγγραμμα χωρίζεται σε επτά κεφάλαια:

- Στο πρώτο κεφάλαιο παρουσιάζονται το αντικείμενο, ο σκοπός, αλλά και οι φάσεις υλοποίησης της εργασίας.
- Στο δεύτερο κεφάλαιο γίνεται αναφορά και συνοπτική ανάλυση στη Θεωρία των Συστημάτων Αποφάσεων προβλήματα της οποίας καλείται να επιλύσει η εφαρμογή. Στη συνέχεια περιγράφονται οι μέθοδοι Δέντρων Αποφάσεων, Γραμμικού Προγραμματισμού και Δυναμικού Προγραμματισμού καθώς αυτές οι μέθοδοι υλοποιούνται στην εργασία.

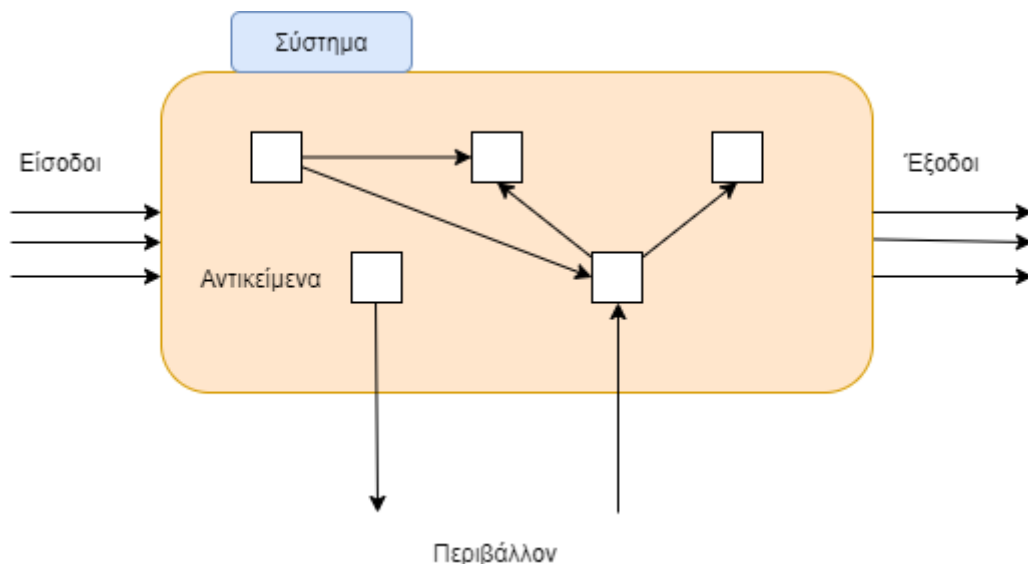
- Στο τρίτο κεφάλαιο παρουσιάζονται υπάρχουσες τεχνολογίες και εφαρμογές Συστημάτων Αποφάσεων. Στη συνέχεια εξηγείται η συμβολή της εφαρμογής της διπλωματικής εργασίας στην εξέλιξη στην επίλυση προβλημάτων της Θεωρίας των Συστημάτων Αποφάσεων.
- Στο τέταρτο κεφάλαιο γίνεται αναφορά στα εργαλεία ανάπτυξης λογισμικού που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής της εργασίας,
- Στο πέμπτο κεφάλαιο περιγράφονται αναλυτικά οι σχεδιαστικές επιλογές και οι υλοποιήσεις που έγιναν για τη δημιουργία των επιμέρους εφαρμογών της εργασίας. Για κάθε μια εξηγείται ο τρόπος συλλογής, επεξεργασίας και αποθήκευσης των δεδομένων, η λειτουργικότητάς της και οι ενιαίοι εντοπιστές πόρων που τις καθιστούν προσβάσιμες από τον χρήστη. Επιπλέον παρουσιάζονται οι αλγόριθμοι επίλυσης που δημιουργήθηκαν στο πλαίσιο των εφαρμογών.
- Στο έκτο κεφάλαιο παρουσιάζονται παραδείγματα χρήσης των εφαρμογών και οι λειτουργικότητες τους αναλυτικά.
- Στο έβδομο κεφάλαιο γίνεται αναφορά στα βασικά συμπεράσματα που προέκυψαν για το πολυπρισματικό εργαλείο λήψης αποφάσεων που κατασκευάστηκε. Επιπλέον πραγματοποιούνται προτάσεις εξέλιξης της εφαρμογής. Συγκεκριμένα προτείνονται επεκτάσεις λειτουργικότητας και επιπλέον μέθοδοι που θα μπορούσαν να προστεθούν ώστε να επεκτείνουν το εύρος προβλημάτων Συστημάτων Αποφάσεων που επιλύει.

2 Θεωρητικό Υπόβαθρο

2.1 Συστήματα Αποφάσεων

Τα συστήματα αποφάσεων αποτελούν ένα μεγάλο τμήμα της επιστήμης της επιχειρησιακής έρευνας. Ουσιαστικά αναφέρονται σε ένα σύνολο ποσοτικών μεθόδων για την ανάλυση αποφάσεων χρησιμοποιώντας την αναμενόμενη χρησιμότητα ως κριτήριο για τον προσδιορισμό της προτιμώμενης εναλλακτικής λύσης [11]. Οι μέθοδοι αυτοί δημιουργήθηκαν και συνεχίζουν να εξελίσσονται από ομάδες επιστημόνων αποτελούμενες από διαφόρων τύπων ειδικοτήτων. Αυτές εφαρμόζονται σε προβλήματα ελέγχου οργανωμένων συστημάτων ώστε να παρέχουν λύσεις που εξυπηρετούν κατά τον καλύτερο τρόπο τους σκοπούς που έχουν τεθεί [1].

Τα κύρια γνωρίσματα της Επιστήμης των Αποφάσεων είναι η συστημική προσέγγιση των προβλημάτων τα οποία καλούνται να επιλύσουν και χρήση μοντέλων για την προσέγγιση των αντικειμένων από τα οποία απαρτίζονται. Συγκεκριμένα, συστημική προσέγγιση είναι η απομόνωση ενός συνόλου στοιχείων που αλληλεπιδρούν μεταξύ τους και είναι οριοθετημένα από άλλα στοιχεία εκτός του συστήματος [1]. Στην *Εικόνα 4* φαίνεται και σχηματικά ο τρόπος με τον οποίο λειτουργεί ένα σύστημα.



Εικόνα 4- Λειτουργία ενός συστήματος

Κάποιες από τις βασικές μεθόδους των συστημάτων αποφάσεων είναι τα Δέντρα Αποφάσεων, ο Γραμμικός Προγραμματισμός και ο Δυναμικός Προγραμματισμός τις

οποίες και αυτή η διπλωματική εργασία έχει ως στόχο να τις υλοποιήσει και να τις συγκεντρώσει σε μια διαδικτυακή εφαρμογή για την επίλυση σχετικών προβλημάτων. Στη συνέχεια ακολουθεί μια ανάλυση των μεθόδων αυτών.

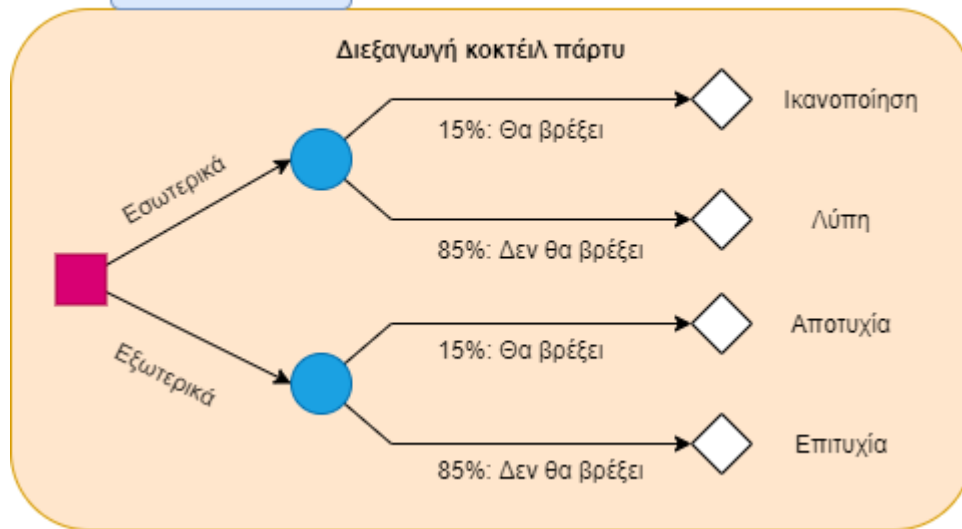
2.2 Δέντρα Αποφάσεων

Τα δέντρα αποφάσεων είναι μια μέθοδος για ορθολογική λήψη αποφάσεων σε συνθήκες αβέβαιου μέλλοντος. Ένα πρόβλημα δέντρων αποφάσεων ορίζεται από τον αποφασίζοντα και τους στόχους τους οποίους αυτός θέτει. Για να φτάσει ο αποφασίζων στον στόχο του θα πρέπει να πάρει ένα σύνολο αποφάσεων, συνυπολογίζοντας τις εναλλακτικές δράσεις που είναι διαθέσιμες συνυπολογίζοντας τις καταστάσεις τις φύσεις και πως αυτές επηρεάζουν το κέρδος ή το κόστος κάθε δράσης [12]. Για τον ορθό ορισμό ενός τέτοιου προβλήματος είναι απαραίτητη η πλήρωση των βασικών παραμέτρων αυτού, οι οποίες είναι οι εξής:

- Στόχοι του αποφασίζοντα
- Τεχνικά Δεδομένα
- Ανάλυση κέρδους/κόστους ανά δράση
- Καταστάσεις της φύσης
- Πληροφορίες για τις καταστάσεις

Στην πράξη, λοιπόν, ένα δέντρο αποφάσεων είναι ένα κατευθυνόμενο δέντρο που αποτυπώνει τα πιθανά αποτελέσματα μια σειράς επιλογών. Ένα δέντρο αποφάσεων εκκινεί με έναν μόνο κόμβο, ο οποίος διακλαδίζεται σε πιθανά αποτελέσματα. Κάθε ένα από αυτά τα αποτελέσματα οδηγεί σε πρόσθετους κόμβους, οι οποίοι διακλαδίζονται σε άλλες δυνατότητες. Τυπικά, ένα τέτοιο δέντρο, εκκινεί από έναν και μόνο κόμβο ο οποίος διακλαδίζεται στα πιθανά αποτελέσματα. Καθένα από αυτά τα αποτελέσματα οδηγούν σε περαιτέρω κόμβους που αποτελούν είτε νέα αποτελέσματα είτε πιθανότητες. Όλη αυτή η διαδικασία οδηγεί σε ένα σχήμα που μοιάζει με δέντρο. Υπάρχουν τριών ειδών κόμβοι: οι κόμβοι απόφασης, οι κόμβοι ευκαιρίας και οι τελικοί κόμβοι. Ο κόμβος απόφασης αναπαριστάται με τετράγωνο σχήμα και τα παιδιά του αποτελούν τις δυνατές επιλογές απόφασης. Ο κόμβος ευκαιρίας αναπαριστάται με κυκλικό σχήμα και τα παιδιά του είναι εκβάσεις οι οποίες έχουν κάποια πιθανότητα να επαληθευτούν. Ο τερματικός κόμβος δεν έχει κάποιο συγκεκριμένο σχήμα, αρκεί να διαφέρει από τα υπόλοιπα είδη κόμβων και δηλώνει το τελικό αποτέλεσμα ενός μονοπατιού απόφασης [13]. Στην *Εικόνα 5* φαίνεται ένα απλό παράδειγμα [12] δέντρου αποφάσεων.

Δέντρο Αποφάσεων



Εικόνα 5- Παράδειγμα δέντρου αποφάσεων

Σε αντίθεση με το παραπάνω παράδειγμα, για την επίλυση ενός δέντρου θα πρέπει να έχουμε αριθμητικά αποτελέσματα των εκβάσεων. Στόχος της λύσης είναι σε κάθε κόμβο απόφασης να επιλεχθεί η επιλογή, βάσει κάποιου επιθυμητού κριτηρίου, που επιφέρει το μέγιστο κέρδος και σε κάθε κόμβο ευκαιρίας να υπολογιστεί, με βάση τις πιθανότητες των παιδιών της και του επιθυμητού κριτηρίου, το σχετικό κέρδος των κόμβων παιδιών της.

Στην προηγούμενη παράγραφο αναφέρθηκε ο όρος επιθυμητό κριτήριο. Τα κριτήρια τα οποία υποστηρίζει η εφαρμογή που κατασκευάστηκε είναι ονομαστικά τα εξής: MAXIMIN, MAXIMAX, Ισοδύναμης Βεβαιότητας. Στη συνέχεια παρουσιάζεται η επίπτωση που έχουν τα παραπάνω κριτήρια στους κόμβους αποφάσεων ενός δέντρου.

Κριτήριο MAXIMAX: Το κριτήριο MAXIMAX είναι μια αισιόδοξη προσέγγιση. Συνιστά στον κόμβο απόφασης να εξετάσει μόνο τις περιπτώσεις των αβέβαιων εκβάσεων που μεγιστοποιούν το κέρδος και στη συνέχεια, μεταξύ αυτών, να επιλέξει αυτή με το μεγαλύτερο κέρδος [14].

Κριτήριο MAXIMIN: Αποτελεί ένα συντηρητικό κριτήριο το οποίο αναπτύχθηκε από τον Wald. Συνιστά στον κόμβο απόφασης να εξετάσει μόνο τις περιπτώσεις των αβέβαιων εκβάσεων που ελαχιστοποιούν το κέρδος και στη συνέχεια, μεταξύ αυτών, να επιλέξει αυτή με το μικρότερο κέρδος [14].

Κριτήριο Ισοδύναμης Βεβαιότητας: Αποτελεί ένα κριτήριο το οποίο συνυπολογίζει όλα τα αβέβαια αποτελέσματα στον υπολογισμό των τιμών των αβέβαιων εκβάσεων. Συγκεκριμένα στους κόμβους ευκαιρίας, δηλαδή στις αβέβαιες εκβάσεις, η τιμή τους είναι το άθροισμα των γινομένων του κέρδους/κόστους κάθε έκβασης με την

πιθανότητα να επαληθευτεί αυτή η έκβαση. Οι κόμβοι απόφασης συνίσταται να επιλέξουν την επιλογή που επιφέρει το μεγαλύτερο κέρδος [15].

Με βάση το κριτήριο που επιλέγεται σε κάθε περίπτωση ολοκληρώνεται ο υπολογισμός του δέντρου αποφάσεων εκκινώντας από τους τερματικούς κόμβους προς τη ρίζα του δέντρου.

Σε περίπτωση που υπάρχουν διαθέσιμες πληροφορίες σχετικά με τις αβέβαιες εκβάσεις χρησιμοποιείται το Θεώρημα Bayes για τον υπολογισμό των βελτιωμένων πιθανοτήτων των εν λόγω εκβάσεων. Δεδομένου ότι οι πληροφορίες παρέχουν την πιθανότητα να γίνει η πρόβλεψη E_j υπό τον όρο η κατάσταση S_i . Με βάση το θεώρημα Bayes οι βελτιωμένες πιθανότητες υπολογίζονται από τον παρακάτω τύπο:

$$P(S_i|E_j) = \frac{P(E_j|S_i) \cdot P(S_i)}{\sum_i P(E_j|S_i) \cdot P(S_i)}$$

Όπου $P(S_i|E_j)$ οι βελτιωμένες πιθανότητες των αβέβαιων εκβάσεων, $P(E_j|S_i)$ η αξιοπιστία της μελέτης και $P(S_i)$ οι αρχικές πιθανότητες των αβέβαιων εκβάσεων [12].

2.3 Γραμμικός Προγραμματισμός

Ο Γραμμικός Προγραμματισμός επιλύει, κάτω από ορισμένες προϋποθέσεις, το πρόβλημα κατανομής πεπερασμένων πόρων ή μέσω ή γενικότερα αγαθών σε διάφορες εναλλακτικές και ανταγωνιστικές μεταξύ τους δραστηριότητες κατά τον καλύτερο δυνατό τρόπο. Κάνει δηλαδή επιλογή της στάθμης κάθε δραστηριότητας έτσι ώστε να βελτιστοποιείται ένα προκαθορισμένο κριτήριο επιλογής [16]. Το κριτήριο επιλογής ονομάζεται αντικειμενική συνάρτηση, οι μεταβλητές της αντικειμενικής συνάρτησης ονομάζονται μεταβλητές απόφασης και το πεπερασμένο των πόρων ορίζεται ως περιορισμοί των πόρων.

Στην παρούσα διπλωματική εργασία ένα πρόβλημα Γραμμικού Προγραμματισμού μπορεί να λυθεί και να αναπαρασταθεί με δύο τρόπους ανάλογα με το πλήθος των μεταβλητών απόφασης. Αν το πρόβλημα αποτελείται από δυο μεταβλητές απόφασης, η λύση αναπαρίσταται γραφικά. Αν το πρόβλημα αποτελείται από περισσότερες από δυο μεταβλητές απόφασης, που δεν είναι εφικτή η γραφική αναπαράσταση δύο επιπέδων, χρησιμοποιείται η μέθοδος Simplex. Στη συνέχεια θα αναλυθούν συνοπτικά οι αλγόριθμοι αυτοί.

2.3.1 Γραφική Επίλυση

Κατά τη γραφική επίλυση κατασκευάζεται ένα σύστημα ορθογώνιων αξόνων. Ο οριζόντιος άξονας αναπαριστά τις τιμές της μιας μεταβλητής απόφασης, ενώ ο κάθετος αναπαριστά τις τιμές της άλλης μεταβλητής απόφασης. Στη συνέχεια προστίθενται στο σύστημα οι ευθείες που περιγράφονται από τους περιορισμούς των μεταβλητών απόφασης. Έπειτα ακολουθεί η σχεδίαση των εξισώσεων των δραστηριοτήτων στο σύστημα, σεβόμενοι την ερμηνεία των ανισοτήτων, όπου και αν αυτές υπάρχουν [17]. Σε αυτό το σημείο στο σύστημα έχει οριστεί ένα επίπεδο επί του οποίου βρίσκεται η λύση του προβλήματος. Κρίσιμα σημεία του επιπέδου για τη λύση του προβλήματος Γραμμικού Προγραμματισμού αποτελούν οι γωνίες που ορίζονται από τις εξισώσεις των περιορισμών και των ανισοτήτων. Συνεπώς, ελέγχοντας ποιός συνδυασμός τιμών των μεταβλητών απόφασης μεγιστοποιεί ή ελαχιστοποιεί, ανάλογα με το είδος του προβλήματος, την αντικειμενική συνάρτηση εντοπίζεται η λύση του προβλήματος. Όπως είναι κατανοητό, ο παραπάνω τρόπος λύσης είναι απαγορευτικός για προβλήματα περισσότερων των δύο μεταβλητών διότι η παραπάνω ανάλυση θα έπρεπε να πραγματοποιηθεί σε πολυδιάστατο σύστημα, το οποίο κάνει την σχεδίαση και κατ' επέκταση τη γραφική επίλυση του προβλήματος πρακτικά αδύνατη.

2.3.2 Μέθοδος Simplex

Η μέθοδος Simplex είναι ένα θεμελιώδης αλγόριθμος επίλυσης προβλημάτων Γραμμικού Προγραμματισμού που κάνει χρήση της θεωρίας της Γραμμικής Άλγεβρας και θεωρείται ένας από τους 10 πιο διάσημους αλγόριθμους του 20^{ου} αιώνα [18]. Η μέθοδος Simplex είναι μια μητρική μέθοδος για την επίλυση γραμμικών προγραμμάτων στην κανονική μορφή. Ένα γραμμικό πρόγραμμα στην κανονική μορφή έχει ως εξής:

$$\text{βελτιστοποίηση: } z = \mathbf{C}^T \mathbf{X}$$

$$\text{με περιορισμούς: } \mathbf{A}\mathbf{X} = \mathbf{B}$$

$$\text{με: } \mathbf{X} \geq \mathbf{0}$$

Όπου $B \geq 0$ και X_0 μια γνωστή βασική εφικτή λύση. Ξεκινώντας από το X_0 , με τη μέθοδο αυτή υπολογίζουμε διαδοχικά άλλες βασικές εφικτές λύσεις που δίνουν καλύτερες τιμές της αντικειμενικής συνάρτησης μέχρι να βρούμε τη βέλτιστη λύση. Για προγράμματα ελαχιστοποίησης, η μέθοδος Simplex χρησιμοποιεί τον πίνακα στην *Εικόνα 6*, στον οποίο το C_0 συμβολίζει το διάνυσμα κόστους που σχετίζεται με τις μεταβλητές στο X_0 .

	X^T	
	C^T	
$X_0 \ C_0$	A	B
	$C_T - C_0^T A$	

Εικόνα 6- Πίνακας Simplex

Για προβλήματα μεγιστοποίησης, ο παραπάνω πίνακας ισχύει αν τα πρόσημα των στοιχείων της κάτω γραμμής είναι αντίστροφα [19].

Η παραπάνω πινακοποίηση της μεθόδου simplex μπορεί να απλοποιηθεί. Για κάθε j ($j = 1, 2, \dots, n$), ορίζουμε $z \equiv C_0^T A_j$, το εσωτερικό γινόμενο του C_0 με τη στήλη j της A . Το στοιχείο j της τελευταίας γραμμής του πίνακα στην *Εικόνα 6* είναι $c_j - z_j$ (ή για προγράμματα μεγιστοποίησης, $z_j - c_j$), όπου c_j είναι το κόστος στη δεύτερη γραμμή του πίνακα, ακριβώς πάνω από την A_j . Αφού βρεθεί αυτή η τελευταία γραμμή, η δεύτερη γραμμή και η δεύτερη στήλη του πίνακα, οι οποίες αντιστοιχούν στα C^T και C_0 , αντίστοιχα, δεν μας χρειάζονται και μπορούν να εξαλειφθούν [19].

Στη συνέχεια ακολουθούν τα βήματα της μεθόδου Simplex, όπως αυτά περιγράφονται στο βιβλίο Επιχειρησιακή Έρευνα [19].

Βήμα 1: Εντοπίζεται ο πιο αρνητικός αριθμός που υπάρχει στην κάτω γραμμή του πίνακα Simplex, με εξαίρεση την τελευταία στήλη, και η στήλη στην οποία εμφανίζεται ο συγκεκριμένος αριθμός ορίζεται ως *στήλη εργασίας*. Αν υπάρχουν περισσότεροι από ένας τέτοιοι αριθμοί, τότε επιλέγεται αυθαίρετα ένας από αυτούς.

Βήμα 2: Σχηματίζονται λόγοι, διαιρώντας κάθε θετικό αριθμό της στήλης εργασίας, με εξαίρεση την τελευταία γραμμή, με το στοιχείο της ίδιας γραμμής και τελευταίας στήλης. Το στοιχείο της στήλης εργασίας που δίνει τον πιο μικρό λόγο ορίζεται ως το *στοιχείο-οδηγό*. Αν τον ίδιο ελάχιστο λόγο δίνουν

περισσότερα από ένα στοιχεία, τότε επιλέγεται αυθαίρετα ένα από αυτά. Αν στη στήλη εργασίας δεν υπάρχει θετικό στοιχείο, το πρόγραμμα δεν έχει λύση.

Βήμα 3: Γίνονται οι απαραίτητες πράξεις γραμμών με τις γραμμές του πίνακα για να μετατραπεί το οδηγό στοιχείο σε μονάδα και στη συνέχεια όλα τα υπόλοιπα στοιχεία της στήλης εργασίας σε 0.

Βήμα 4: Η μεταβλητή x στην πρώτη στήλη της γραμμής-οδηγού αντικαθίσταται με τη μεταβλητή x στην πρώτη γραμμή της στήλης-οδηγού. Η νέα πρώτη στήλη είναι το προσωρινό σύνολο των βασικών μεταβλητών και ονομάζεται βάση.

Βήμα 5: Τα βήματα 1 έως 4 επαναλαμβάνονται μέχρι να μην υπάρχουν αρνητικοί αριθμοί στην τελευταία γραμμή, με εξαίρεση την τελευταία στήλη.

Βήμα 6: Η βέλτιστη λύση προκύπτει όταν σε κάθε μεταβλητή της πρώτης στήλης δοθεί η τιμή που υπάρχει στην αντίστοιχη γραμμή της τελευταίας στήλης. Σε όλες τις άλλες μεταβλητές δίνεται η τιμή μηδέν. Το αντίστοιχο z^* , η βέλτιστη τιμή της αντικειμενικής συνάρτησης, είναι ο αριθμός στην τελευταία γραμμή και τελευταία στήλη για ένα πρόγραμμα μεγιστοποίησης ή ο αντίθετός του για ένα πρόβλημα ελαχιστοποίησης.

Τα παραπάνω βήματα αφορούν προγράμματα των οποίων οι περιορισμοί δεν περιλαμβάνουν τεχνητές μεταβλητές. Ωστόσο πριν την αναφορά των τροποποιήσεων στη μέθοδο Simplex εξηγείται ο τρόπος με τον οποίο οι περιορισμοί ανισοτήτων μετατρέπονται σε ισότητες καθώς και ο όρος *τεχνητές μεταβλητές*.

Ένας γραμμικός περιορισμός της μορφής $\sum_{j=1}^n a_{ij} x_j \leq b_i$ μπορεί να μετατραπεί σε ισότητα με την προσθήκη μια νέας, μη αρνητικής μεταβλητής στο αριστερό μέλος της ανισότητας. Μια τέτοια μεταβλητή, που είναι αριθμητικά ίση με τη διαφορά του δεξιού μέλους και του αριστερού μέλους της ανισότητας, ονομάζεται *χαλαρή μεταβλητή*. Παριστάνει τη σπατάλη η οποία αφορά τη φάση του συστήματος που μοντελοποιείται από τον περιορισμό [20].

Ένας γραμμικός περιορισμός της μορφής $\sum_{j=1}^n a_{ij} x_j \geq b_i$ μπορεί να μετατραπεί σε ισότητα αν αφαιρεθεί μια νέα, μη αρνητική μεταβλητή από το αριστερό μέλος της ανισότητας. Μια τέτοια μεταβλητή, που είναι αριθμητικά ίση με τη διαφορά μεταξύ του αριστερού και του δεξιού μέλους της ανισότητας, ονομάζεται *πλεονασματική μεταβλητή*. Παριστάνει πρόσθετη είσοδο ή προσθήκη στη φάση του συστήματος που μοντελοποιείται από τον περιορισμό [20].

Αφού όλοι οι γραμμικοί περιορισμοί (με μη αρνητικά δεξιά μέλη) μετατραπούν σε εξισώσεις με την εισαγωγή χαλαρών και πλεονασματικών μεταβλητών, όπου αυτό είναι απαραίτητο, προστίθεται μια νέα μεταβλητή, που ονομάζεται *τεχνητή μεταβλητή*, στο αριστερό μέλος κάθε εξίσωσης περιορισμού που δεν περιλαμβάνει κάποια χαλαρή μεταβλητή. Με τον τρόπο αυτό, σε κάθε εξίσωση θα υπάρχει μια χαλαρή ή μια τεχνητή μεταβλητή [20].

Η εισαγωγή χαλαρών μεταβλητών και πλεονασματικών μεταβλητών δεν αλλάζει το χαρακτήρα των περιορισμών ή της αντικειμενικής συνάρτησης. Οι μεταβλητές αυτές εισάγονται στην αντικειμενική συνάρτηση με μηδενικούς συντελεστές. Ωστόσο, οι τεχνητές μεταβλητές αλλάζουν το χαρακτήρα των περιορισμών. Αφού προστίθενται μόνο στο ένα μέλος της ισότητας, το νέο σύστημα των περιορισμών είναι ισοδύναμο με το παλαιό αν και μόνο αν οι τεχνητές μεταβλητές είναι μηδέν. Για να εξασφαλιστεί ότι αυτό ισχύει στη βέλτιστη λύση, οι τεχνητές μεταβλητές ενσωματώνονται στην αντικειμενική συνάρτηση ενός προγράμματος ελαχιστοποίησης με πολύ μεγάλους θετικούς συντελεστές ή με πολύ μεγάλους αρνητικούς συντελεστές στην αντικειμενική συνάρτηση ενός προγράμματος μεγιστοποίησης. Αυτοί οι συντελεστές, που συμβολίζονται με M είτε με $-M$, όπου M ένας μεγάλος θετικός αριθμός, παριστάνουν την (αυστηρή) ποινή που επιβάλλεται κατά την εισαγωγή των τεχνητών μεταβλητών [20].

Αφού παραπάνω επεξηγήθηκαν όλα τα πιθανά μέρη ενός προβλήματος γραμμικού προγραμματισμού, ακολουθούν οι τροποποιήσεις στην παραπάνω μέθοδο, ώστε να ελαχιστοποιηθεί το σφάλμα στρογγυλοποίησης.

Τροποποίηση 1: Η τελευταία γραμμή του πίνακα της *Εικόνας 6* αναλύεται σε δύο γραμμές, η πρώτη από τις οποίες αποτελείται από τους όρους που δεν περιέχουν M , ενώ η δεύτερη αποτελείται από τους συντελεστές του M στους όρους που απομένουν.

Τροποποίηση 2: Στην τελευταία γραμμή που δημιουργήθηκε κατά την *Τροποποίηση 1* εφαρμόζεται το *Βήμα 1* της μεθόδου Simplex (και στη συνέχεια τα βήματα 2, 3 και 4) μέχρι η γραμμή αυτή να μην περιέχει αρνητικά στοιχεία. Στη συνέχεια, το *Βήμα 1* εφαρμόζεται στα στοιχεία της προτελευταίας στήλης που βρίσκονται επάνω από τα από τα μηδενικά στοιχεία της τελευταίας γραμμής.

Τροποποίηση 3: Κάθε φορά που μια τεχνητή μεταβλητή παύει να είναι βασική – δηλαδή όταν αφαιρείται από την πρώτη στήλη του πίνακα ως συνέπεια του *Βήματος 4* – διαγράφεται από την επάνω γραμμή του πίνακα, όπως και ολόκληρη η στήλη που βρίσκεται κάτω από αυτήν ή απλώς αγνοείται.

Τροποποίηση 4: Η τελευταία γραμμή μπορεί να διαγραφεί από τον πίνακα κάθε φορά που περιέχει μόνο μηδενικά στοιχεία.

Τροποποίηση 5: Αν στο τελικό σύνολο βασικών μεταβλητών υπάρχουν μη μηδενικές τεχνητές μεταβλητές, τότε το πρόγραμμα δεν έχει λύση. (Αντίθετα, στην τελική λύση μπορεί να εμφανιστούν τεχνητές μεταβλητές με μηδενική τιμή ως βασικές μεταβλητές όταν ένας ή περισσότεροι από τους αρχικούς περιορισμούς είναι πλεονασματικοί.

Ο αλγόριθμος που προκύπτει από είναι η μέθοδος δύο φάσεων.

2.4 Δυναμικός Προγραμματισμός

Ο Δυναμικός Προγραμματισμός είναι η συλλογή διάφορων μαθηματικών εργαλείων τα οποία χρησιμοποιούνται για την ανάλυση διαδοχικών διαδικασιών λήψης αποφάσεων. Για την ακρίβεια, το αντικείμενο που είναι γνωστό σήμερα ως Δυναμικός Προγραμματισμός γεννήθηκε κατά την ανάλυση των διαδοχικών διαδικασιών λήψης αποφάσεων, όταν παρατηρήθηκε ότι συγκεκριμένα χαρακτηριστικά επαναλαμβανόντουσαν ξανά και ξανά. Έτσι ο Δυναμικός Προγραμματισμός αναγνωρίζεται ως ένας κλάδος των εφαρμοσμένων μαθηματικών παρά κάτι πιο εξειδικευμένο. Η ύπαρξη του κλάδου είναι αποτέλεσμα του γεγονότος ότι βρίσκει εφαρμογές σε διάφορους τομείς. Κάποιοι από αυτούς είναι η έννοια του κράτους, αρχή της βελτιστοποίησης και λειτουργικές εξισώσεις [21]. Λόγω του εύρους των εφαρμογών του Δυναμικού Προγραμματισμού, η παρούσα διπλωματική επικεντρώνεται και αναλύει τρεις συγκεκριμένες εφαρμογές αυτού: την Εύρεση Βέλτιστη Διαδρομή, τον Προγραμματισμό Επενδύσεων και τον Προγραμματισμό Παραγωγής και Αποθήκευσης.

2.4.1 Εύρεση Βέλτιστης Διαδρομής

Ένα πρόβλημα συντομότερης διαδρομής περιλαμβάνει ένα συνεκτικό δίκτυο, με κάθε διακλάδωση του οποίου έχει συσχετιστεί ένα μη αρνητικό κόστος. Ένας κόμβος ορίζεται ως πηγή και ένας άλλος κόμβος ορίζεται ως απαγωγός ή καταβόθρα ή αρνητική πηγή. Ο στόχος είναι ο προσδιορισμός μιας διαδρομής που θα ενώνει την πηγή και τον απαγωγό τέτοιας ώστε το άθροισμα των τιμών κόστους που σχετίζονται με τους διακλαδώσεις της διαδρομής να είναι ελάχιστο [22].

Τα προβλήματα διαδρομών ελαχίστου κόστους λύνονται με τον παρακάτω αλγόριθμο, κατά την εφαρμογή του οποίου σε όλες τις ισότητες η επιλογή γίνεται αυθαίρετα.

Βήμα 1: Δημιουργείται ένας κύριος κατάλογος ταξινομώντας σε μορφή πίνακα κάτω από κάθε κόμβο, και με αύξουσα σειρά κόστους, των διακλαδώσεων που προσπίπτουν στον κόμβο. Η αναφορά σε κάθε διακλάδωση που βρίσκεται κάτω από έναν συγκεκριμένο κόμβο ξεκινά από τον κόμβο αυτό. Στον κατάλογο δεν συμπεριλαμβάνεται οποιαδήποτε διακλάδωση έχει την πηγή ως δεύτερο κόμβο της ή έχει τον απαγωγό ως πρώτο κόμβο.

Βήμα 2: Η πηγή επισημαίνεται με αστερίσκο και σε αυτή ανατίθεται η τιμή μηδέν. Εντοπίζεται η διακλάδωση που έχει το μικρότερο κόστος και εφάπτεται με την πηγή και επισημαίνεται. Επισημαίνεται με αστερίσκο ο δεύτερος κόμβος της διακλάδωσης αυτής και ανατίθεται σε αυτόν τιμή ίση με τον κόμβο της διακλάδωσης. Όλες οι άλλες διακλαδώσεις που έχουν τον κόμβο, ο οποίος

μόλις επισημάνθηκε με αστερίσκο, ως δεύτερο κόμβο τους διαγράφονται από τον κύριο κατάλογο.

Βήμα 3: Αν ο κόμβος, ο οποίος μόλις επισημάνθηκε με αστερίσκο, είναι ο απαγωγός, ο αλγόριθμος προχωράει στο *Βήμα 5*, αν όχι, προχωράει στο επόμενο βήμα.

Βήμα 4: Εξετάζονται όλοι οι κόμβοι με αστερίσκο, οι οποίοι έχουν μη επισημασμένες διακλαδώσεις στον ισχύοντα κύριο κατάλογο. Για κάθε κόμβο, προστίθεται η τιμή που έχει ανατεθεί στον κόμβο στο κόστος της συντομότερης διακλάδωσης χωρίς επισήμανση, η οποία βρίσκεται κάτω από αυτόν. Το μικρότερο από αυτά τα αθροίσματα συμβολίζεται με M και επισημαίνεται η διακλάδωση, το κόστος της οποίας συνείσφερε στο M . Επισημαίνεται με αστερίσκο ο δεύτερος κόμβος αυτής της διακλάδωσης και ανατίθεται σε αυτόν την τιμή M . Όλες οι άλλες διακλαδώσεις που έχουν αυτόν τον κόμβο, ο οποίος μόλις επισημάνθηκε με αστερίσκο, ως δεύτερο κόμβο διαγράφονται από τον κύριο κατάλογο.

Βήμα 5: z^* είναι η τιμή που ανατίθεται στον απαγωγό. Με αναδρομική διαδικασία βρίσκεται η διαδρομή ελαχίστου κόστους: ,ε σημείο εκκίνησης τον απαγωγό, συμπεριλαμβανομένων όλων των επισημασμένων διακλαδώσεων, ο δεύτερος κόμβος της οποίας ανήκει στη διαδρομή.

2.4.2 Προγραμματισμός Επενδύσεων

Ο Δυναμικός Προγραμματισμός βασίζεται στην αρχή βελτιστότητας του Bellman, με βάση την οποία μια βέλτιστη πολιτική έχει την ιδιότητα ότι ανεξάρτητα από τις αποφάσεις που έγιναν στη διάρκεια ενός συγκεκριμένου σταδίου για την είσοδο σε μια συγκεκριμένη κατάσταση, οι υπόλοιπες αποφάσεις πρέπει να συνιστούν μια βέλτιστη πολιτική για την έξοδο από αυτή την κατάσταση [23].

Στον προγραμματισμό επενδύσεων, λοιπόν, αν οι χρονικές περίοδοι οριστούν ως στάδια και οι μονάδες κεφαλαίου προς επένδυση ως οι πιθανές επιλογές σε κάθε στάδιο, η παραπάνω αρχή μπορεί να χρησιμοποιηθεί ως μέθοδος επίλυσης. Για την εφαρμογή αυτής της αρχής, ο αλγόριθμος εκκινεί από το τελευταίο στάδιο μιας διαδικασίας n σταδίων και προσδιορίζει για κάθε κατάσταση τη βέλτιστη πολιτική για την έξοδο από αυτή την κατάσταση και την ολοκλήρωση της διαδικασίας, θεωρώντας ότι όλα τα προηγούμενα στάδια έχουν ολοκληρωθεί. Στη συνέχεια, ανατρέχει αντίστροφα στη διαδικασία, στάδιο προς στάδιο. Σε κάθε στάδιο προσδιορίζεται η βέλτιστη πολιτική για την έξοδο από κάθε κατάσταση και την ολοκλήρωση της διαδικασίας, θεωρώντας ότι όλα τα προηγούμενα στάδια έχουν ολοκληρωθεί και χρησιμοποιώντας τα αποτελέσματα που έχουν ήδη βρεθεί από το επόμενο στάδιο. Έτσι υπολογίζονται οι καταχωρίσεις του πίνακα της *Εικόνας 7*, όπου

	u					
	0	1	2	3	...	
$m_n(u)$						} Τελευταίο στάδιο
$d_n(u)$						
$m_{n-1}(u)$						} Προτελευταίο στάδιο
$d_{n-1}(u)$						
...					
$m_1(u)$						} Πρώτο στάδιο
$d_1(u)$						

Εικόνα 7- Πίνακας επίλυσης προβλήματος προγραμματισμού επενδύσεων

$u \equiv$ η μεταβλητή κατάστασης, οι τιμές της οποίας προσδιορίζουν τις καταστάσεις

$m_j(u) \equiv$ η βέλτιστη απόδοση από την ολοκλήρωση της διαδικασίας ξεκινώντας από το στάδιο j στην κατάσταση u

$d_j(u) \equiv$ η απόφαση η οποία έγινε στο στάδιο j και επιτυγχάνει την $m_j(u)$

Οι καταχωρίσεις που αντιστοιχούν στο τελευταίο στάδιο της διαδικασίας, οι $m_n(u)$ και $d_n(u)$, είναι σε γενικές γραμμές εύκολο να υπολογιστούν. Οι υπόλοιπες βρίσκονται με αναδρομική διαδικασία υπολογισμού, δηλαδή οι καταχωρίσεις για το στάδιο j προσδιορίζονται ως προς τις καταχωρίσεις του σταδίου $j + 1$. Ο αναδρομικός τύπος υπολογισμού εξαρτάται από το εκάστοτε πρόβλημα και πρέπει να βρίσκεται εκ νέου για κάθε τύπο πολυσταδιακής διαδικασίας [23].

Ο πίνακας της Εικόνας 7 βασίζεται στην παραδοχή ότι κάθε στάδιο έχει το ίδιο σύνολο καταστάσεων. Με βάση αυτή την παραδοχή, οι τιμές του $m_n(u)$ για $u = 0, 1, \dots, b$ δίνονται από τον τύπο:

$$m_u(u) = \max_{0 \leq x \leq u} \{f_n(x)\}$$

Ο αναδρομικός τύπος για $j = n - 1, n - 2, \dots, 1$ είναι:

$$m_j = \max_{0 \leq x \leq u} \{f_j(x) + m_{j+1}(u - x)\}$$

Στην πρώτη σχέση, η μεταβλητή x , η οποία συμβολίζεται με x_n , παίρνει ακέραιες τιμές, όπως και το x ($\equiv x_j$) στη δεύτερη σχέση. Αν οποιοδήποτε από αυτά τα δύο

βέλτιστα δίνονται από πολλές τιμές του x , επιλέγεται αυθαίρετα μια από αυτές ως η βέλτιστη απόφαση. Η βέλτιστη λύση για το παραπάνω πρόγραμμα είναι η $z = m_1(b)$, η οποία είναι η βέλτιστη απόδοση από την ολοκλήρωση της διαδικασίας αν στο στάδιο 1 υπήρχαν αρχικά b διαθέσιμες μονάδες για κατανομή. Αφού έχει προσδιοριστεί το z , οι βέλτιστες αποφάσεις x_1^* , x_2^* , ..., x_n^* βρίσκονται διαδοχικά από τις παρακάτω σχέσεις:

$$\begin{aligned} x_1^* &= d_1(b) \\ x_2^* &= d_2(b - x_1^*) \\ x_3^* &= d_3(b - x_1^* - x_2^*) \\ &\dots\dots\dots \\ x_n^* &= d_n(b - x_1^* - x_2^* - \dots - x_{n-1}^*) \end{aligned}$$

2.4.3 Προγραμματισμός Παραγωγής και Αποθήκευσης

Τα προβλήματα αποθήκευσης είναι μια ειδική υποδιαίρεση της κλάσεως των στοχαστικών προβλημάτων βελτιστοποίησης πολλών σταδίων, όπου ένας πόρος διατηρείται ώστε να είτε να μπορεί να διατεθεί τώρα για ένα ορισμένο κέρδος ή στο μέλλον για αβέβαιο κέρδος [24]. Ωστόσο στην παρούσα διπλωματική εργασία, το κέρδος σε κάθε χρονική περίοδο θεωρείται γνωστό.

Σε κάθε περίοδο, ο υπεύθυνος για λήψη απόφασης πρέπει να υπολογίσει το όφελος της παραγωγής για την απλή κάλυψη της ζήτησης σε σχέση με την περαιτέρω παραγωγή ώστε να πουληθεί ο πόρος μεταγενέστερα. Όπως και στην περίπτωση του προγραμματισμού επενδύσεων, έτσι κι εδώ εφαρμόζεται η αρχή βελτιστότητας του Bellman. Σε αυτή την περίπτωση ως στάδια ($j = 1, 2, \dots, n$) ορίζονται οι χρονικές περιόδους για τις οποίες πρέπει να γίνει ο προγραμματισμός και το μέγεθος της παρτίδας παραγωγής σε κάθε στάδιο ως οι πιθανές επιλογές. Εδώ θεωρούνται γνωστά η ζήτηση ανά χρονική περίοδο, η δυναμικότητα παραγωγής και η δυναμικότητα αποθήκευσης. Σημαντική σημείωση αποτελεί ότι σε κάθε στάδιο πρέπει να καλύπτεται η απαιτούμενη ζήτηση. Επιπροσθέτως, αυτού του είδους τα προβλήματα έχουν δύο βασικές εξισώσεις εξάρτησης, το κόστος παραγωγής και το κόστος αποθήκευσης τα οποία θα επεξηγηθούν στη συνέχεια [10].

Παρόμοια με τον προγραμματισμό επενδύσεων ο αλγόριθμος εκκινεί από το τελευταίο στάδιο μιας διαδικασίας n σταδίων και προσδιορίζει για κάθε κατάσταση τη βέλτιστη πολιτική για την έξοδο από αυτή την κατάσταση και την ολοκλήρωση της διαδικασίας, θεωρώντας ότι όλα τα προηγούμενα στάδια έχουν ολοκληρωθεί. Στη συνέχεια ανατρέχει αντίστροφα στη διαδικασία, στάδιο προς στάδιο. Σε κάθε στάδιο προσδιορίζεται η βέλτιστη πολιτική για την έξοδο από κάθε κατάσταση και την

ολοκλήρωση της διαδικασίας, θεωρώντας ότι όλα τα προηγούμενα στάδια έχουν ολοκληρωθεί και χρησιμοποιώντας τα αποτελέσματα που έχουν ήδη βρεθεί από το επόμενο στάδιο [23]. Εδώ, τα στοιχεία του πίνακα της *Εικόνας 7* ορίζονται ως εξής:

$u \equiv$ η μεταβλητή κατάσταση. Συγκεκριμένα οι διαθέσιμες μονάδες στην αποθήκη στο στάδιο n

$m_j(u) \equiv$ το ελάχιστο κόστος παραγωγής και αποθήκευσης από την ολοκλήρωση της διαδικασίας ξεκινώντας από το στάδιο j στην κατάσταση u

$d_j(u) \equiv$ η απόφαση που έγινε στο στάδιο j και επιτυγχάνει τη $m_j(u)$, δηλαδή το μέγεθος της παρτίδας που επιλέχθηκε να παραχθεί.

Όπως αναφέρθηκε και παραπάνω, οι καταχωρίσει στο στάδιο n είναι εύκολο να υπολογιστούν. Οι υπόλοιπες ωστόσο βρίσκονται με αναδρομική διαδικασία υπολογισμού. Οι καταχωρίσεις για το στάδιο j προσδιορίζονται ως προς τις καταχωρίσεις του σταδίου $j + 1$. Σε αυτό το σημειωθεί αξίζει να σημειωθεί ότι ο αναδρομικός τύπος εξαρτάται από το εκάστοτε πρόβλημα και πρέπει να βρίσκεται εκ νέου για κάθε τύπο πολυσταδιακής διαδικασίας [23].

Στην παρούσα διπλωματική εργασία γίνεται η παραδοχή ότι κάθε στάδιο έχει ίδιο b ακέραιο αριθμό καταστάσεων. Αυτός ο αριθμός ορίζεται από τη μέγιστη δυναμικότητα αποθήκευσης και την ελάχιστη μερίδα πόρων που μπορούν να αποθηκευτούν. Με βάση αυτή την παραδοχή, οι τιμές του $m_n(u)$ για $u = 0, 1^*(\text{μερίδα αποθήκευσης}), 2^*(\text{μερίδα αποθήκευσης}), \dots, (b-1)^*(\text{μερίδα αποθήκευσης})$ δίνονται από τον τύπο:

$$m_u(u) = \min_{0 \leq x \leq u} \{C_{\text{παραγωγής}}(x) + C_{\text{αποθήκευσης}}(u)\}$$

Όπου $C_{\text{παραγωγής}}$ και $C_{\text{αποθήκευσης}}$ το κόστος παραγωγής και κόστος αποθήκευσης αντίστοιχα. Αυτές οι δύο συναρτήσεις διαφέρουν από πρόβλημα σε πρόβλημα και πιθανόν από στάδιο σε στάδιο.

Ο αναδρομικός τύπος για $j = n - 1, n - 2, \dots, 1$ είναι:

$$m_j = \min_{0 \leq x \leq u} \{C_{\text{παραγωγής},j}(x_j) + C_{\text{αποθήκευσης},j}(u_j) + m_{j+1}(x_j, u_j)\}$$

Στην πρώτη και στη δεύτερη σχέση η μεταβλητή x (x_j στη δεύτερη σχέση) παίρνουν τιμές ακέραιων πολλαπλάσιων της ελάχιστης μερίδας παραγωγής. Αν οποιοδήποτε από αυτά τα δύο βέλτιστα δίνονται από πολλές τιμές του x , επιλέγεται αυθαίρετα μια από αυτές ως βέλτιστη απόφαση. Η βέλτιστη λύση για το παραπάνω πρόγραμμα είναι η $z = m_1(b_1)$, αν στο στάδιο 1 υπήρχαν αρχικά b_1 ποσότητα πόρων στο απόθεμα. Αφού προσδιοριστεί το z , οι βέλτιστες αποφάσεις $x_1^*, x_2^*, \dots, x_n^*$ βρίσκονται διαδοχικά από τις παρακάτω σχέσεις:

$$x_1^* = d_1(b_1)$$

$$x_2^* = d_2(b_1 + x_1^* - w_1)$$

.....

$$x_n^* = d_n(b_{n-1} + x_{n-1}^* - w_{n-1})$$

Όπου w_j η ζήτηση στο στάδιο j , η οποία είναι γνωστή.

3 Σύγχρονες Σχετικές Εφαρμογές

Τα τελευταία χρόνια, πιο πολύ από ποτέ, η επιχειρησιακή έρευνα βρίσκεται στο επίκεντρο της λειτουργίας των επιχειρήσεων και της έρευνας σε πανεπιστημιακό επίπεδο. Αν και η επιστήμη έχει αναπτυχθεί σημαντικά, λίγα είναι τα εργαλεία υπολογισμού που είναι διαθέσιμα για τον κοινό χρήστη, καθώς συνήθως είναι η ανάπτυξη μοντέλων και εφαρμογών σε ατομικό επιχειρηματικό επίπεδο. Για αυτό το λόγο οι διαθέσιμες προς το κοινό εφαρμογές είτε είναι λίγες σε αριθμό, είτε περιορίζονται σε εκπαιδευτικό επίπεδο, είτε αποτελούν πλήρως εμπορικά προϊόντα επί πληρωμή.

3.1 Σύγχρονες Εφαρμογές

3.1.1 Lucidchart – Decision Tree Maker

Η εφαρμογή Decision Tree Maker [25] αποτελεί ένα εργαλείο σχεδιασμού Δέντρων Αποφάσεων που περιέχεται στη συνδρομητική διαδικτυακή εφαρμογή Lucidchart [26]. Η εφαρμογή αυτή παρέχει στον χρήστη που επιθυμεί να υπολογίσει τους κινδύνους, τις ανταμοιβές και τα χρηματικά κέρδη τη δυνατότητα δημιουργίας ενός δέντρου αποφάσεων. Είτε αυτός επιθυμεί να αναλύσει την επικινδυνότητα μιας επένδυσης είτε να αποτυπώσει τα πιθανά αποτελέσματα ως μια ακολουθία γεγονότων, το Decision Tree Maker τον διευκολύνει. Επιπλέον αποτελεί ένα επαγγελματικό εργαλείο όπου παρέχει πληθώρα πρωτοτύπων και προσαρμόσιμων επιλογών μορφοποίησης που καθιστούν την κατασκευή ενός δέντρου αποφάσεων ευκολότερη. Τέλος, ο χρήστης παρέχονται επιλογές εξατομίκευσης, όπως η επιλογή χρωμάτων και μεγεθών, που βελτιστοποιούν την κατανόηση του δέντρου.

3.1.2 WolframAlpha - Linear Programming Solver

Το Linear Programming Solver [27] αποτελεί μια μικροεφαρμογή που κατασκευάστηκε στην ιστοσελίδα WolframAlpha [28] από τον χρήστη vik_31415. Η εν λόγω εφαρμογή ζητάει από τον χρήστη να παρέχει τα δεδομένα ενός προβλήματος Γραμμικού Προγραμματισμού, μέχρι εννέα μεταβλητών, και να του επιστρέφει τη βέλτιστη λύση, αν υπάρχει. Δηλαδή, ο χρήστης μπορεί να ορίσει την αντικειμενική συνάρτηση ενός προβλήματος, τις δραστηριότητες και τους περιορισμούς που περιλαμβάνει. Κατά την επίδειξη του αποτελέσματος ο χρήστης λαμβάνει τις τιμές για τις οποίες το πρόβλημά του βελτιστοποιείται και την τιμή που παίρνει η αντικειμενική συνάρτηση στη βέλτιστη λύση, αν αυτή υπάρχει. Επιπλέον, αν πρόκειται για

πρόβλημα δύο μεταβλητών και υπάρχει λύση, η εφαρμογή παρουσιάζει στον χρήστη και τη γραφική απεικόνιση του προβλήματος. Ωστόσο επειδή αποτελεί μια μικροεφαρμογή που έχει κατασκευαστεί και κοινοποιηθεί από έναν χρήστη του WolframAlpha, δεν υπάρχει τεκμηρίωση για τον τρόπο επίλυσης των προβλημάτων και των αλγορίθμων που χρησιμοποιούνται σε αυτήν.

3.1.3 Visual LinProg

Το Visual LinProg [29] είναι ένα εκπαιδευτικό εργαλείο που επιλύει Γραμμικά Προβλήματα χρησιμοποιώντας τεχνικές απεικόνισης και κινουμένων σχεδίων. Ο πυρήνας του λογισμικού αυτό περιλαμβάνει τη γνωστή οικογένεια αλγορίθμων Simplex. Το Visual LinProg εφευρέθηκε για να χρησιμοποιηθεί αρχικά σε μαθήματα μαθηματικού προγραμματισμού ως συμπληρωματικό εκπαιδευτικό εργαλείο. Ο μαθητής-χρήστης μπορεί να επιλύει τα δικά του Γραμμικά Προβλήματα, να δει τη λύση τους βήμα-βήμα και να εισάγει και να εξάγει τα δεδομένα του με εύκολο και ευανάγνωστο τρόπο. Η διαδικασία επίλυσης συνοδεύεται από κείμενο επεξήγησης της λύσης σε κάθε βήμα.

3.1.4 Online Calculator

Το Online Calculator [30] είναι μια τόσο διαδικτυακή εφαρμογή, όσο και εφαρμογή κινητού που περιέχει εργαλεία επίλυσης προβλημάτων Συστημάτων Αποφάσεων. Συγκεκριμένα, ένας χρήστης μπορεί να εφαρμόσει μεθόδους θεωρίας παιγνίων, Simplex, διπλό Simplex, δυναμικού προγραμματισμού και άλλες. Η εν λόγω εφαρμογή καταφέρνει να συγκεντρώσει ένα σημαντικό αριθμό μεθόδων επιχειρησιακή έρευνας και αποτελεί ένα ολοκληρωμένο εργαλείο λήψης αποφάσεων. Ο χρήστης έχει τη δυνατότητα να πληκτρολογήσει τα δεδομένα του για ένα συγκεκριμένο πρόβλημα αποφάσεων και να του παρουσιαστεί η λύση αυτού. Επιπλέον διαθέτει σύστημα λογαριασμών χρηστών αν και δεν προσφέρει κάποια περαιτέρω υπηρεσία στον χρήστη. Τέλος, αν και έχει φιλικό περιβάλλον χρήσης, ο σχεδιασμός του είναι πεπαλαιωμένος.

3.1.5 DynaProg

Το DynaProg [31] είναι ένα εργαλείο ανοιχτού κώδικα της προγραμματιστικής και υπολογιστικής πλατφόρμας MATLAB [32]. Το MATLAB συνδυάζει ένα περιβάλλον ηλεκτρονικού υπολογιστή, συντονισμένο για επαναληπτική ανάλυση και σχεδιασμό διαδικασιών με γλώσσα προγραμματισμού που εκφράζει τις μήτρες και τους πίνακες μαθηματικών ευθέως. Περιλαμβάνει εργαλείο σύνταξης για τη συγγραφή κώδικα. Το DynaProg επιλύει πολυεπίπεδα αιτιοκρατικά προβλήματα βελτιστοποίησης

απόφασης. Αυτά τα προβλήματα ελέγχου βελτιστοποίησης μπορούν να λυθούν με τη μέθοδο του Δυναμικού Προγραμματισμού, η οποία προτείνεται σε μη γραμμικά δυναμικά συστήματα. Το εργαλείο παρέχει στον χρήστη έναν αριθμητικό γρήγορο κινητήρα εμφωλευμένο σε μια απλή στη χρήση διεπαφή, όπου μπορεί με ευθύ τρόπο να ορίσει ευθέως και σε ένα ευέλικτο περιβάλλον το πρόβλημά του. Επιπλέον, επιτρέπει στους ερευνητές να πραγματοποιήσουν με ευκολία έρευνα για τη χρήση του Δυναμικού Προγραμματισμού, ο καθένας στο αντικείμενο εξειδίκευσής του. Τέλος, ο βασισμένος σε κλάσεις πηγαίος κώδικας με τον οποίο έχει σχεδιαστεί το εργαλείο επιτρέπει στους έμπειρους ερευνητές του Δυναμικού Προγραμματισμού να πειραματιστούν και γιατί όχι να επεκτείνουν τις λειτουργικότητες του εργαλείου, αν αυτό κριθεί απαραίτητο.

3.2 Συμβολή της Διπλωματικής Εργασίας

Οι παραπάνω εφαρμογές, αν και παρέχουν κάποιες καλές λύσεις και υλοποιήσεις στα προβλήματα της επιχειρησιακής έρευνας, δεν παρέχουν ένα ολοκληρωμένο εργαλείο επίλυσης για πολλές περιπτώσεις προβλημάτων. Εξαίρεση αποτελεί η εφαρμογή Online Calculator, η οποία αν και παρέχει πληθώρα εργαλείων, διαθέτει ένα όχι τόσο φιλικό προς τον χρήστη περιβάλλον.

Στην παρούσα διπλωματική εργασία αναπτύσσεται ένα πολυπρισματικό εργαλείο το οποίο δεν εστιάζει στην επίλυση ενός μόνο είδους προβλημάτων. Αντιθέτως, δίνει τη δυνατότητα σχεδιασμού και επίλυσης προβλημάτων Δέντρων Αποφάσεων, Γραμμικού Προγραμματισμού και Δυναμικού Προγραμματισμού.

Όσον αφορά στα Δέντρα Αποφάσεων, υπάρχουν αρκετές εφαρμογές σχεδιασμού δέντρων αποφάσεων, τόσο δωρεάν όσο και επί πληρωμή. Ωστόσο, περιορισμένες είναι αυτές στις οποίες ο χρήστης μπορεί να επιλύσει το πρόβλημά του. Έτσι, η εργασία αυτή σχεδιάστηκε ώστε να παρέχει στον χρήστη τόσο εργαλείο σχεδιασμού ενός δέντρου αποφάσεων, αλλά και να επιλύει το δέντρο αυτό. Επιπλέον, η ενσωμάτωση μελέτης σε ένα δέντρο αποφάσεων μπορεί να φανεί χρονοβόρα σε έναν χρήστη. Για αυτό το λόγο κατά τη σχεδίαση ενός δέντρου ο χρήστης θα πρέπει απλά να παρέχει τα αποτελέσματα της έρευνας σε μορφή πίνακα. Έπειτα το εργαλείο θα σχεδιάζει το υποδέντρο της μελέτης μόνο του.

Σε σχέση με τις ήδη υπάρχουσες εφαρμογές Γραμμικού Προγραμματισμού, το εργαλείο που αναπτύχθηκε στο πλαίσιο της διπλωματικής εργασίας, καταφέρνει να συνδυάσει την απλότητα και ευκολία στην καταχώρηση των δεδομένων από τον χρήστη με τη πλήρη επεξηγηματική παρουσίαση της λύσης ενός προβλήματος Γραμμικού Προγραμματισμού. Όλα αυτά ενσωματώνονται σε ένα φιλικό και διαδραστικό για τον χρήστη περιβάλλον, όπου έχει τη δυνατότητα να πειραματιστεί με ευκολία. Με αυτόν τον τρόπο η εφαρμογή καταφέρνει να αντιμετωπίσει τη μη

φιλικότητα των δεδομένων προς τον χρήστη, λόγω του πλήθους των μεταβλητών και των εξισώσεων που διέπουν τα προβλήματα Γραμμικού Προγραμματισμού.

Τις σημαντικότερες ελλείψεις σε αριθμό και ποιότητα εργαλείων συναντά ο χρήστης στο πεδίο του Δυναμικού Προγραμματισμού. Τόσο οι εφαρμογές, όσο και τα προβλήματα που αυτές επιλύουν είναι περιορισμένα. Κατά συνέπεια, έρχεται το εργαλείο Δυναμικού Προγραμματισμού που υλοποιήθηκε στο πλαίσιο της διπλωματικής εργασίας να δώσει λύση σε τρία είδη προβλημάτων. Αυτά είναι η εύρεση βέλτιστης διαδρομής, ο προγραμματισμός επενδύσεων και ο προγραμματισμός παραγωγής και αποθήκευσης. Όπως και στις προηγούμενες περιπτώσεις, έτσι κι εδώ η εφαρμογή αποτελείται από ένα φιλικό προς τον χρήστη περιβάλλον στο οποίο μπορεί να κατασκευάσει τα προβλήματά του, αλλά και να δει αναλυτικά τα στάδια που ακολουθήθηκαν για την επίλυσή του.

Επιπλέον, πρέπει να τονιστεί ότι οι παραπάνω εφαρμογές δεν εστιάζουν σε συγκεκριμένο είδους δεδομένων εισόδου, αρκεί το πρόβλημα και αυτά να είναι συμβατά με το εργαλείο που χρησιμοποιείται στην εκάστοτε περίπτωση.

Σε όλες τις παραπάνω περιπτώσεις δίνεται ιδιαίτερη σημασία στη διεπαφή του χρήστη. Με αυτόν τον τρόπο κατασκευάστηκε ένα φιλικό προς αυτόν εργαλείο που όχι μόνο μπορεί να κατασκευάσει τα εν λόγω προβλήματα σε αυτό, αλλά να παρέχει και έτοιμα τα δεδομένα του ώστε να έχει άμεσα τη λύση, εφόσον το επιθυμεί.

Τέλος, οι παραπάνω υπάρχουσες εφαρμογές αποτέλεσαν πηγή έμπνευσης για τη δημιουργία της εφαρμογής που περιγράφεται στην εργασία αυτή, αλλά σε ορισμένες περιπτώσεις παρουσίαζαν είτε μη φιλικό προς τον χρήστη περιβάλλον, είτε μικρό εύρος χρήσης, είτε ακόμα και έλλειψη λύσης στα προβλήματα, αλλά μόνο σχεδιασμός. Η εφαρμογή που αναπτύχθηκε παρέχει λοιπόν ένα αρκετά ολοκληρωμένο εργαλείο σχεδιασμού και επίλυσης προβλημάτων επιχειρησιακής έρευνας.

Στα κεφάλαια που ακολουθούν αναλύονται τα εργαλεία που χρειάστηκαν κατά την υλοποίηση, οι επιλογές σχεδίασης και υλοποίησης που έγιναν καθώς και η παρουσίαση της χρήσης αυτών σε πραγματικά προβλήματα.

4 Εργαλεία Ανάπτυξης Λογισμικού

4.1 Περιβάλλον ανάπτυξης Visual Studio Code

Το Visual Studio Code [33] είναι ένας συντάκτης ο οποίος κέρδισε γρήγορα δημοτικότητα μεταξύ των προγραμματιστών, κατατάσσοντάς το ως το πιο δημοφιλές περιβάλλον ανάπτυξης, συνολικά, στην έρευνα προγραμματιστών του Stack Overflow του 2019 [34] στην οποία έλαβαν μέρος περισσότεροι από ενενήντα χιλιάδες προγραμματιστές. Η δημοσιότητά του οφείλεται στην ανάπτυξη του τομέα ανάπτυξης ιστοσελίδων τα τελευταία χρόνια και στην ανάγκη των προγραμματιστών να έχουν έναν ελαφρύ και καλοφτιαγμένο συντάκτη, με λίγα χαρακτηριστικά αλλά λιγότερο περίπλοκα από τα άλλα διαθέσιμα στην αγορά. Επιπλέον η χρήση του είναι δωρεάν και αναπτύσσεται και συντηρείται από τη Microsoft.

Το Visual Studio Code είναι ολοκληρωμένο περιβάλλον ανάπτυξης το οποίο χρησιμοποιείται για τη γραφή, τη δοκιμή και τον έλεγχο σφαλμάτων του λογισμικού. Όντας εφαρμογή ανοιχτού κώδικα, διαθέτει ένα μεγάλο οικοσύστημα επεκτάσεων κάνοντας εφικτή την ανάπτυξη εφαρμογών σε διάφορες γλώσσες προγραμματισμού, όπως Python, Java, Javascript, C/C++, Jinja2, PHP, R και άλλες.

Ένα ακόμη σημαντικό χαρακτηριστικό του είναι ότι διαθέτει ενσωματωμένο σύστημα διαχείρισης πολλαπλών παρόχων πηγής ελέγχου, όπως σύστημα ανάπτυξη βάσεων δεδομένων. Τα παραπάνω τον καθιστούν κατάλληλο για την ανάπτυξη της διαδικτυακής εφαρμογής της παρούσας διπλωματικής εργασίας.

4.2 Η γλώσσα προγραμματισμού Python

Η Python είναι μια ερμηνευμένη, αντικειμενοστραφής, υψηλού επιπέδου γλώσσα προγραμματισμού με δυναμική σημασιολογία. Η υψηλού επιπέδου ενσωματωμένη δομή δεδομένων, σε συνδυασμό με τη δυναμική πληκτρολόγηση (dynamic typing) και τη δυναμική δέσμευση (dynamic binding), την καθιστούν πολύ ελκυστική για την Ταχεία Ανάπτυξη Εφαρμογών (Rapid Application Development), καθώς και για χρήση ως γλώσσα προγραμματισμού για τη σύνδεση υπαρχόντων στοιχείων μεταξύ τους. Το συντακτικό της Python είναι απλό, εύκολο στην εκμάθηση και δίνει έμφαση στην αναγνωσιμότητα και συνεπώς μειώνει το κόστος συντήρησης του προγράμματος. Υποστηρίζει ενότητες και πακέτα, γεγονός που ενθαρρύνει τη λειτουργικότητα του προγράμματος και την επαναχρησιμοποίηση του κώδικα. Ο διερμηνέας της Python και η εκτεταμένη τυπική βιβλιοθήκη της διατίθενται σε μορφή κώδικα ή δυαδική μορφή χωρίς χρέωση για όλες τις μεγάλες πλατφόρμες [35].

4.3 Django Web Framework

Το Django [36] είναι ένα υψηλού επιπέδου διαδικτυακό πλαίσιο (web framework) της Python το οποίο επιτρέπει τον ταχύ προγραμματισμό ασφαλών και διατηρήσιμων ιστότοπων. Χτισμένο από έμπειρους προγραμματιστές, το Django φροντίζει για την αποφυγή της δυσκολίας ανάπτυξης ιστοσελίδων, ώστε ο χρήστης να επικεντρώνεται στη σύνταξη. Είναι ανοιχτού κώδικα, έχει ακμάζουσα και ενεργή κοινότητα, μεγάλη τεκμηρίωση και πολλές επιλογές υποστήριξης.

Το Django ακολουθεί τη φιλοσοφία «οι Μπαταρίες περιλαμβάνονται» (“Batteries included”) και παρέχει σχεδόν όλα τα εργαλεία που χρειάζεται ένας προγραμματιστής ενσωματωμένα. Καθώς όλα αυτά τα εργαλεία αποτελούν μέρος ενός προϊόντος, είναι σχεδιασμένα να λειτουργούν άψογα μαζί και να ακολουθούν με συνέπεια τις αρχές σχεδιασμού και διαθέτουν εκτεταμένη και ενημερωμένη τεκμηρίωση.

Επιπλέον μπορεί να χρησιμοποιηθεί και χρησιμοποιείται για τη δημιουργία σχεδόν κάθε τύπου ιστοτόπου – από συστήματα διαχείρισης περιεχομένου, μέχρι κοινωνικά δίκτυα και ειδησεογραφικούς ιστότοπους. Μπορεί να λειτουργήσει με οποιοδήποτε πλαίσιο από την πλευρά του χρήστη και μπορεί να του παρέχει περιεχόμενο σε σχεδόν οποιαδήποτε μορφή (συμπεριλαμβανομένων HTML, ροών RSS, JSON, XML κ.λπ.).

Ένα ακόμη σημαντικό χαρακτηριστικό του Django είναι ότι βοηθά τους προγραμματιστές να αποφύγουν κοινά λάθη ασφαλείας, παρέχοντας ένα πλαίσιο που έχει σχεδιαστεί για να κάνει αυτό που πρέπει για την αυτόματη προστασία του ιστοτόπου. Επιτρέπει την προστασία από πολλές ευπάθειες συμπεριλαμβανομένης της ένεσης SQL (SQL injection), της δέσμης ενεργειών μεταξύ ιστοτόπων (cross-site scripting), και της πλαστογραφίας αιτήματος μεταξύ ιστοτόπων, τα οποία το καθιστούν ασφαλή.

Επιπλέον το Django χρησιμοποιεί μια βασισμένη στα συστατικά “shared nothing” αρχιτεκτονική (κάθε κομμάτι της αρχιτεκτονικής είναι ανεξάρτητο από τα υπόλοιπα και κατ’ επέκταση μπορεί να αντικατασταθεί ή να αλλάξει αν χρειαστεί). Έχοντας διαχωρισμένα τα διάφορα μέρη σημαίνει ότι μπορεί να κλιμακωθεί για αυξημένη επισκεψιμότητα προσθέτοντας υλικό σε οποιοδήποτε επίπεδο: διακομιστές προσωρινής αποθήκευσης, διακομιστές βάσεων δεδομένων ή διακομιστές εφαρμογών. Χαρακτηριστικά παραδείγματα ιστοσελίδων που κλιμάκωσαν το Django για να ικανοποιήσουν τη ζήτησή τους είναι το Instagram και το Disqus.

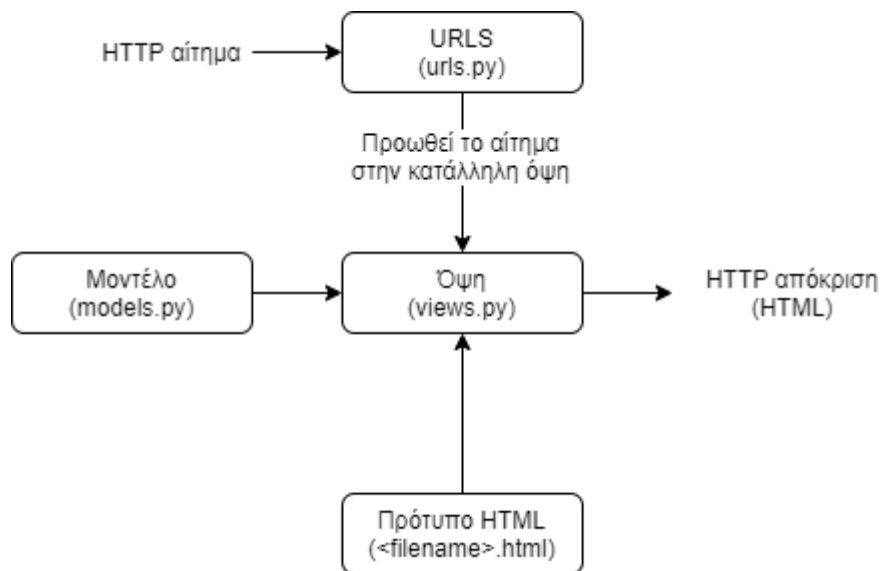
Ο κώδικας του Django γράφεται χρησιμοποιώντας αρχές σχεδιασμού και μοτίβα που ενθαρρύνουν τη δημιουργία διατηρήσιμου και επαναχρησιμοποιούμενου κώδικα. Συγκεκριμένα, κάνει χρήση της αρχής Μην Επαναλαμβάνεις τον Εαυτό σου (Don't Repeat Yourself), ώστε να μην υπάρχει περιττή επικάλυψη, μειώνοντας την έκταση του κώδικα. Το Django προωθεί επίσης την ομαδοποίηση των σχετικών λειτουργιών

σε επαναχρησιμοποιήσιμες εφαρμογές και, σε χαμηλότερο επίπεδο, ομαδοποιεί τον κώδικα σε ενότητες σύμφωνα με το μοτίβο του Model View Controller (MVC).

Πώς λειτουργεί το Django;

Σε έναν παραδοσιακό ιστότοπο που βασίζεται σε δεδομένα, μια εφαρμογή ιστού περιμένει αιτήματα HTTP από το πρόγραμμα περιήγησης ιστού. Όταν λαμβάνεται ένα αίτημα, η εφαρμογή εξηγεί τι χρειάζεται με βάση τη διεύθυνση ενιαίου εντοπιστή πόρου (URL) και πιθανώς τις πληροφορίες σε POST και GET δεδομένα. Ανάλογα με το τι απαιτείται, μπορεί στη συνέχεια να διαβάσει ή να γράψει πληροφορίες από μια βάση δεδομένων ή να εκτελέσει άλλες εργασίες που απαιτούνται για να ικανοποιήσει το αίτημα. Στη συνέχεια, η εφαρμογή θα επιστρέψει μια απάντηση στο πρόγραμμα περιήγησης ιστού, δημιουργώντας συχνά δυναμικά μια σελίδα HTML για εμφάνιση του προγράμματος περιήγησης εισάγοντας τα δεδομένα που ανακτήθηκαν σε σύμβολα κράτησης θέσης σε ένα πρότυπο HTML.

Οι διαδικτυακές εφαρμογές Django συνήθως ομαδοποιούν τον κώδικα που χειρίζεται καθένα από αυτά τα βήματα σε ξεχωριστά αρχεία:



Εικόνα 8- Τρόπος διαχείρισης αιτημάτων στο Django

- **Διευθύνσεις URL:** Ενώ είναι δυνατή η επεξεργασία αιτημάτων από κάθε διεύθυνση URL μέσω μιας συνάρτησης, είναι πολύ πιο διατηρήσιμο να γράφεται μια συνάρτηση – όψη – για τη διαχείριση κάθε πόρου. Ένας προσαρμογέας URL (URL mapper) χρησιμοποιείται για την ανακατεύθυνση αιτημάτων HTTP στην κατάλληλη όψη με βάση τη διεύθυνση URL του αιτήματος. Η αντιστοίχιση URL μπορεί επίσης να ταιριάζει με συγκεκριμένα

μοτίβα συμβολοσειρών ή ψηφίων που εμφανίζονται σε μια διεύθυνση URL και να τα μεταφέρει σε μια όψη ως δεδομένα.

- **Όψη:** Μια όψη είναι μια λειτουργία χειρισμού αιτημάτων, η οποία λαμβάνει αιτήματα HTTP και επιστρέφει απαντήσεις HTTP. Οι προβολές έχουν πρόσβαση στα δεδομένα που απαιτούνται για την ικανοποίηση αιτημάτων μέσω μοντέλων και αναθέτουν τη μορφοποίηση της απόκρισης σε πρότυπα HTML.
- **Μοντέλα:** Τα μοντέλα είναι αντικείμενα Python που καθορίζουν τη δομή των δεδομένων μιας εφαρμογής και παρέχουν μηχανισμούς για τη διαχείριση (προσθήκη, τροποποίηση, διαγραφή) και ερωτήσεων (queries) εγγραφών στη βάση δεδομένων.
- **Πρότυπα HTML:** Ένα πρότυπο HTML είναι ένα αρχείο κειμένου που καθορίζει τη δομή ή τη διάταξη ενός αρχείου (όπως μια σελίδα HTML), με τα σύμβολα κράτησης θέσης που χρησιμοποιούνται για την αναπαράσταση του πραγματικού περιεχομένου. Μια όψη μπορεί να δημιουργήσει δυναμικά μια σελίδα HTML χρησιμοποιώντας ένα πρότυπο HTML, συμπληρώνοντάς την με δεδομένα από ένα μοντέλο. Ένα πρότυπο μπορεί να χρησιμοποιηθεί για τον καθορισμό της δομής οποιουδήποτε τύπου αρχείου.

Η παραπάνω εκτεταμένη παρουσίαση του Django [37] είναι αναγκαία διότι πολλοί όροι και αντικείμενά του χρησιμοποιούνται στη συνέχεια για να επεξηγηθεί η ανάπτυξη της εφαρμογής της παρούσας διπλωματικής εργασίας.

4.4 Σύστημα διαχείρισης βάσεων δεδομένων PostgreSQL

Το PostgreSQL είναι ένα ισχυρό, ανοιχτού κώδικα αντικειμενοσχεσιακό σύστημα βάσης δεδομένων που χρησιμοποιεί και επεκτείνει τη γλώσσα SQL σε συνδυασμό με πολλές δυνατότητες που αποθηκεύουν και κλιμακώνουν με ασφάλεια τον πιο περίπλοκο φόρτο εργασίας δεδομένων. Η προέλευση του PostgreSQL χρονολογείται από το 1986 ως μέρος του POSTGRES στο Πανεπιστήμιο Μπέρκλεϊ της Καλιφόρνια και έχει περισσότερα από 30 χρόνια ενεργής ανάπτυξης στη βασική πλατφόρμα.

Το PostgreSQL έχει κερδίσει ισχυρή φήμη για την αποδεδειγμένη αρχιτεκτονική, την αξιοπιστία, την ακεραιότητα των δεδομένων, το ισχυρό σύνολο δυνατοτήτων, την επεκτασιμότητα και την αφοσίωση της κοινότητας ανοιχτού κώδικα πίσω από το λογισμικό για την προσφορά, με συνέπεια, επιδόσεων και καινοτόμων λύσεων [38].

Η χρήση του PostgreSQL έχει πολλά οφέλη για τον προγραμματιστή για αυτό και επιλέχθηκε ως σύστημα βάσεων δεδομένων της παρούσας διπλωματικής εργασίας. Αρχικά, το Django προσφέρει ένα αριθμό τύπων δεδομένων τα οποία λειτουργούν μόνο με το PostgreSQL. Επίσης το Django δίνει τη δυνατότητα στον προγραμματιστή να εκτελέσει λειτουργίες βάσης δεδομένων στο PostgreSQL. Τελευταίο και

σημαντικότερο χαρακτηριστικό της PostgreSQL είναι ότι διαθέτει το πλουσιότερο σύνολο δυνατοτήτων που υποστηρίζονται από το Django.

4.5 Οι γλώσσες ιστού HTML, CSS και Javascript

Οι γλώσσες ιστού HTML, CSS και Javascript θεωρούνται ο κεντρικός άξονας του παγκόσμιου ιστού. Όσον αφορά την ανάπτυξη ιστοσελίδων, υπάρχει το «μπροστινό μέρος» (front-end) ανάπτυξης και το «πίσω μέρος» (back-end). Αυτές οι τρεις γλώσσες είναι για την ανάπτυξη του «μπροστινού μέρους» και είναι υπεύθυνες για το τι βλέπει ένας χρήστης σε μια ιστοσελίδα. Αναφέρονται επίσης και ως «από τη μεριά του χρήστη» γλώσσες καθώς εκτελούνται στο πρόγραμμα περιήγησης του υπολογιστή. Το πρόγραμμα περιήγησης μεταφράζει αυτές τις γλώσσες και αποτέλεσμα αυτής της μετάφρασης είναι το οπτικό κομμάτι μιας ιστοσελίδας.

Αξίζει να σημειωθεί ότι οι HTML και CSS δεν θεωρούνται γλώσσες προγραμματισμού. Η HTML είναι μια γλώσσα σήμανσης και η CSS είναι μια στιλιστική γλώσσα. Ωστόσο, η Javascript είναι γλώσσα προγραμματισμού. Αν και όλες είναι γλώσσες ιστού, εκτελούν διαφορετικές εργασίες [39].

HyperText Markup Language (HTML)

Η γλώσσα σήμανσης HTML μπορεί να αναλυθεί σε HyperText, το οποίο παρέχει πρόσβαση σε άλλα κείμενα μέσω συνδέσμων και Markup που περιγράφει τη βασική δομή και εμφάνιση του ακατέργαστου κειμένου. Αυτό σημαίνει ότι η HTML περιγράφει και ορίζει το περιεχόμενο και τη βασική δομή μιας ιστοσελίδας. Αυτό γίνεται μέσω ειδικών ετικετών ή κωδικών που καθοδηγούν το πρόγραμμα περιήγησης στο τι πρέπει να κάνει. Η HTML είναι από τα πολύ βασικά στοιχεία μιας ιστοσελίδας [39].

Cascading Style Sheet (CSS)

Η CSS είναι «τα αξεσουάρ» μιας ιστοσελίδας. Είναι υπεύθυνη για τη σκιαγράφηση των χρωμάτων, τις γραμματοσειρές και τη θέση του περιεχομένου ενός ιστότοπου. Προσθέτει κάποιο ύφος και δομή στο περιεχόμενο. Προκειμένου να αξιοποιηθούν οι δυνατότητες της CSS, πρέπει να συνδεθεί το περιεχόμενο της HTML, ώστε να προστεθεί ύφος στην ιστοσελίδα. Το CSS στην πράξη θα ορίσει στο πρόγραμμα περιήγησης το πως πρέπει να παρουσιαστεί το περιεχόμενο της HTML [39].

Javascript

Η Javascript ελέγχει τη συμπεριφορά ενός ιστότοπου. Είναι σημαντικό να σημειωθεί ότι η Javascript και η Java είναι δυο διαφορετικά πράγματα. Η Javascript σχεδιάστηκε για να χειρίζεται ιστοσελίδες και χρησιμοποιείται για τη δημιουργία διαδραστικών λειτουργιών. Χωρίς Javascript, μια ιστοσελίδα θα εξακολουθεί να είναι λειτουργική, αλλά με περιορισμένο τρόπο. Η Javascript είναι αυτή που ζωντανεύει την HTML και τη CSS σε μια ιστοσελίδα [39].

4.6 Η βιβλιοθήκη οπτικοποίησης VisJS

Το VisJS είναι μια δυναμική, βασισμένη στο πρόγραμμα περιήγησης, βιβλιοθήκη οπτικοποίησης της Javascript. Η εν λόγω βιβλιοθήκη έχει σχεδιαστεί για να είναι εύκολη στη χρήση, να χειρίζεται μεγάλες ποσότητες δυναμικών δεδομένων και να επιτρέπει το χειρισμό και την αλληλεπίδραση με τα δεδομένα. Η βιβλιοθήκη αποτελείται από τα στοιχεία DataSet, Timeline, Network, Graph2d και Graph3d [40].

Στην παρούσα διπλωματική εργασία αξιοποιήθηκε το στοιχείο Network για την οπτικοποίηση των δέντρων αποφάσεων. Το Network, ή στα ελληνικά δίκτυο, είναι μια οπτικοποίηση για την εμφάνιση δικτύων και δικτύων που αποτελούνται από κόμβους και ακμές. Η απεικόνιση είναι εύκολη στη χρήση και υποστηρίζει προσαρμοσμένα σχήματα, στυλ, χρώματα, μεγέθη, εικόνες και πολλά άλλα. Η οπτικοποίηση δικτύου λειτουργεί ομαλά σε οποιοδήποτε σύγχρονο πρόγραμμα περιήγησης για έως και μερικές χιλιάδες κόμβους και ακμές. Από την έκδοση 4.0, το δίκτυο αποτελείται από μεμονωμένες μονάδες που χειρίζονται συγκεκριμένα τμήματα του δικτύου [41].

4.7 Η βιβλιοθήκη οπτικοποίησης Chart.js

Το Chart.js [42] είναι μια δημοφιλής, ανοιχτού κώδικα, βιβλιοθήκη της Javascript για την απεικόνιση δεδομένων. Υποστηρίζει οκτώ διαφορετικούς τύπους γραφημάτων (γραφήματα πίτας, γραμμικά γραφήματα, γραφήματα μπάρας και άλλα). Στην παρούσα διπλωματική χρησιμοποιήθηκε το γράφημα γραμμής για την απεικόνιση των γραφικών αποτελεσμάτων στην εφαρμογή γραμμικού προγραμματισμού. Η μεγάλη έκταση της βάσης τεκμηρίωσης της οποίας βοήθησε σημαντικά στην ανάπτυξη του παραπάνω εργαλείου.

4.8 Το πρότυπο Black Dashboard

Το πρότυπο Black Dashboard [43] είναι ένα σύγχρονο θέμα της Bootstrap 4 το οποίο αναπτύχθηκε για το Django από τη ομάδα Creative-Tim. Διαθέτει ένα τεράστιο

αριθμό εξαρτημάτων κατασκευασμένα για να ταιριάζουν μεταξύ τους και να δίνουν ένα όμορφο οπτικό αποτέλεσμα. Επιπλέον, η χρήση αυτών των εξαρτημάτων διευκολύνει τον προγραμματισμό προτύπων Django (Django templates). Αυτό το πρότυπο χρησιμοποιήθηκε και για τη δημιουργία των προτύπων της εφαρμογής που αναπτύχθηκε κατά την εκπόνηση της παρούσας διπλωματικής εργασίας.

5 Σχεδίαση και Υλοποίηση της Διαδικτυακής Εφαρμογής

5.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναλυθεί η σχεδίαση και η υλοποίηση της εφαρμογής. Δεδομένου ότι στόχος της είναι η επίλυση προβλημάτων Δέντρων Αποφάσεων, Γραμμικού Προγραμματισμού και Δυναμικού Προγραμματισμού έγινε η σχεδιαστική επιλογή να επιμεριστεί σε τέσσερις επιμέρους εφαρμογές. Τα μέρη αυτά είναι τα εξής:

- Βασική Εφαρμογή
- Εφαρμογή Δέντρων Αποφάσεων
- Εφαρμογή Γραμμικού Προγραμματισμού
- Εφαρμογή Δυναμικού Προγραμματισμού

Το κάθε μέρος είναι υπεύθυνο για την υλοποίηση και τη διεκπεραίωση ενός ξεχωριστού συνόλου λειτουργιών της εφαρμογής. Η Εφαρμογή Δέντρων Αποφάσεων είναι υπεύθυνη για τη συλλογή δεδομένων από τον χρήστη, την επίλυση των προβλημάτων Δέντρων Αποφάσεων και την απεικόνιση της λύσης τους. Η Εφαρμογή Γραμμικού Προγραμματισμού είναι υπεύθυνη για τη συλλογή από τον χρήστη, την επίλυση των προβλημάτων Γραμμικού Προγραμματισμού και την απεικόνιση της λύσης τους. Η Εφαρμογή Δυναμικού Προγραμματισμού είναι υπεύθυνη για τη συλλογή δεδομένων από τον χρήστη, την επίλυση προβλημάτων Δυναμικού Προγραμματισμού και την απεικόνιση της λύσης τους. Τέλος, η Βασική Εφαρμογή είναι υπεύθυνη για την περιήγηση του χρήστη στο περιβάλλον της Διαδικτυακής Εφαρμογής εκτός των προαναφερθέντων τριών εφαρμογών.

5.2 Δημιουργία Εφαρμογών της Διαδικτυακής Εφαρμογής

Το Django καθιστά σημαντικά εύκολη την κατασκευή των παραπάνω εφαρμογών. Συγκεκριμένα, για τη δημιουργία τους χρειάστηκαν οι παρακάτω εντολές στον διερμηνέα της python.

Για την κατασκευή της Βασικής Εφαρμογής:

➤ `python manage.py startapp mainApp`

Η εντολή αυτή κατασκευάζει αυτόματα τα βασικά αρχεία που θα χρησιμοποιηθούν στην υλοποίηση της Βασικής Εφαρμογής. Σε αυτό το σημείο θα παρουσιαστούν και θα αναλυθούν συνοπτικά τα αρχεία που παράγονται μετά την εκτέλεση της παραπάνω εντολής.

Αρχικά δημιουργείται ο φάκελος mainApp στο ευρετήριο της συνολικής εφαρμογής και περιέχει τα εξής αρχεία:

__init__.py: Αποτελεί ένα κενό Python αρχείο όπου με βάση το όνομά του καθίσταται γνωστό στην Python ότι ο φάκελος mainApp θα πρέπει να αντιμετωπίζεται ως ένα πακέτο.

admin.py: Δίνει τη δυνατότητα στον προγραμματιστή να εγγράψει σε αυτό το Python αρχείο τα μοντέλα τα οποία θα χρησιμοποιήσει στη συνέχεια στην εφαρμογή του.

apps.py: Σε αυτό το αρχείο τοποθετούνται οι ειδικές, για κάθε εφαρμογή, διαμορφώσεις.

models.py: Εδώ ο προγραμματιστής θα κληθεί να κατασκευάσει τα μοντέλα δεδομένων που θα χρησιμοποιήσει στην εφαρμογή του.

tests.py: Στο αρχείο αυτό ο προγραμματιστής μπορεί να κατασκευάσει συναρτήσεις με τις οποίες θα έχει τη δυνατότητα να ελέγξει τον κώδικα της εφαρμογής του.

views.py: Εδώ κατασκευάζονται οι συναρτήσεις που θα δώσουν εν τέλη λειτουργικότητα στην εφαρμογή. Δηλαδή περιέχει τις συναρτήσεις που διαχειρίζονται τα αιτήματα των χρηστών και επιστρέφουν τις επιθυμητές αντιδράσεις.

migrations/__init__.py: Τέλος, σε αυτό το φάκελο το Django θα αποθηκεύει αυτόματα όλες τις πληροφορίες της βάσης δεδομένων, όπως αυτές σχετίζονται με τα μοντέλα δεδομένων που χρησιμοποιεί η εφαρμογή. Το αρχείο __init__.py επιτελεί την ίδια λειτουργία όπως έχει αναλυθεί παραπάνω.

Ακολουθούν οι αντίστοιχες εντολές για τις υπόλοιπες εφαρμογές, αυτές που κατασκευάζουν την Εφαρμογή Δέντρων Αποφάσεων, την Εφαρμογή Γραμμικού Προγραμματισμού και την Εφαρμογή Δυναμικού Προγραμματισμού αντίστοιχα.

- `python manage.py startapp dectree`
- `python manage.py startapp linear`
- `python manage.py startapp dynamic`

Έπειτα από τη δημιουργία των εφαρμογών, για να είναι ορατές στο Django θα πρέπει να δηλωθούν στο αρχείο `setting.py`, όπου περιλαμβάνονται όλες οι ρυθμίσεις της συνολικής εφαρμογής. Όλες οι εφαρμογές που κατασκευάζονται πρέπει να δηλώνονται στη λίστα `INSTALLED_APPS`. Στη συγκεκριμένη περίπτωση η λίστα αυτή θα διαμορφωθεί ως εξής:

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'bootstrap4',  
    'mainApp',  
    'dectree',  
    'linear',  
    'dynamic'  
]
```

Αφού ολοκληρωθεί η διαδικασία της δημιουργίας και καταχώρησης των εφαρμογών που απαρτίζουν τη συνολική διαδικτυακή εφαρμογή, έπεται η σχεδίαση και η συγγραφή του κώδικα που θα δώσει, εν τέλη, λειτουργικότητα στην εκάστοτε εφαρμογή.

5.3 Σχεδίαση και Υλοποίηση της Βασικής Εφαρμογής

Ο ρόλος της Βασικής Εφαρμογής είναι να διεκπεραιώσει την πλοήγηση του χρήστη εκτός των υπόλοιπων εφαρμογών, αλλά και μέσω αυτής να επιλέξει ποιά εφαρμογή του έργου επιθυμεί να χρησιμοποιήσει. Επίσης σε αυτή δηλώνεται το Μοντέλο Δεδομένων που χρησιμοποιείται στο συνολικό έργο.

5.3.1 Μοντέλα Δεδομένων

Ένα ουσιώδες κομμάτι της Διαδικτυακής Εφαρμογής είναι η ικανότητα συλλογής δεδομένων από τον χρήστη, η επεξεργασία τους και η εισαγωγή τους στη βάση δεδομένων ώστε να είναι διαθέσιμα να ανακτηθούν και να παρουσιαστούν στον

χρήστη. Τα Μοντέλα Δεδομένων χρησιμοποιούνται για την ενσωμάτωση της βάσης δεδομένων στη Διαδικτυακή Εφαρμογή.

Σε αυτό το σημείο λήφθηκε η απόφαση να κατασκευαστεί ένα μοντέλο το οποίο θα καλύπτει όλες τις εφαρμογές που απαρτίζουν το συνολικό έργο, αφού το Django επιτρέπει τη χρήση κοινών Μοντέλων Δεδομένων μεταξύ των εφαρμογών. Υπό αυτό το πρίσμα, το Μοντέλο Δεδομένων κατασκευάστηκε στη Βασική Εφαρμογή και στη συνέχεια δηλώθηκε προς χρήση στο κάθε αρχείο `admin.py` των υπόλοιπων εφαρμογών.

Το Μοντέλο Δεδομένων που κατασκευάστηκε για τις ανάγκες του έργου περιέχει τα παρακάτω πεδία:

- `methodID = models.AutoField(primary_key=True)`: Είναι ένας αύξων θετικός και ακέραιος αριθμός ο οποίος και αποτελεί το πρωτεύον κλειδί του Μοντέλου. Μέσω αυτού έχουμε πρόσβαση σε όλα τα στοιχεία του πίνακα της βάσης δεδομένων.
- `method_type = models.CharField(blank=True, max_length=50)`: Εδώ δηλώνεται ποια μέθοδος (π.χ. Γραμμικός Προγραμματισμός) χρησιμοποιήθηκε για το πρόβλημα που επιλύθηκε.
- `title = models.CharField(blank=True, max_length=50)`: Σε αυτό το πεδίο αποθηκεύεται ο τίτλος που δίνει ο χρήστης στο πρόβλημα που θέλει να επιλύσει. Ο τίτλος μπορεί να έχει μήκος μέχρι 50 χαρακτήρων.
- `input_file = models.JSONField()`: Αποτελεί το πεδίο όπου αποθηκεύεται το αρχείο εισόδου που χρησιμοποιήθηκε στο πρόβλημα που επιλύθηκε. Το αρχείο αποθηκεύεται σε μορφή JSON.
- `output_file = models.JSONField()`: Αποτελεί το πεδίο όπου αποθηκεύεται το αρχείο εξόδου που παράχθηκε από το πρόβλημα που επιλύθηκε. Το αρχείο αποθηκεύεται σε μορφή JSON.
- `date = models.DateTimeField(auto_now_add=True)`: Στο πεδίο αυτό καταχωρείται αυτόματα κατά τη δημιουργία του μοντέλου η πλήρης ημερομηνία και ώρα που επιλύθηκε το πρόβλημα.

Το παραπάνω Μοντέλο Δεδομένων περιλαμβάνει όλες τις απαραίτητες πληροφορίες που χαρακτηρίζουν ένα πρόβλημα που θα θέσει ένας χρήστης προς επίλυση.

5.3.2 Λειτουργικότητα

Για να μπορεί ο χρήστης να περιηγηθεί μέσα στο συνολικό έργο και να έχει πρόσβαση τόσο στις εφαρμογές επίλυσης προβλημάτων, όσο και στις υπόλοιπες σελίδες της εφαρμογής κατασκευάστηκαν οι παρακάτω όψεις.

mainapp_index: Η όψη αυτή δέχεται μόνο GET αίτημα από τον χρήστη και του φορτώνει τη σελίδα που δίνει πρόσβαση στις εφαρμογές Δέντρων Αποφάσεων, Γραμμικού Προγραμματισμού και Δυναμικού Προγραμματισμού.

instructions: Η όψη αυτή δέχεται μόνο GET αίτημα από τον χρήστη και του φορτώνει τη σελίδα που περιλαμβάνει τα σχήματα JSON που έχουν οριστεί για τα αρχεία εισόδου της κάθε εφαρμογής.

info: Η όψη αυτή δέχεται μόνο GET αίτημα από το χρήστη και του φορτώνει τη σελίδα που περιλαμβάνει πληροφορίες για το συνολικό έργο.

Παρότι οι παραπάνω όψη δεν περιλαμβάνουν κάποια ουσιώδη λειτουργικότητα είναι απαραίτητες για την καλύτερη εμπειρία του χρήστη.

5.3.3 Διασύνδεση Λειτουργικότητας με τον χρήστη

Η πρόσβαση του χρήστη στις όψεις γίνεται μέσω των Ενιαίων Εντοπιστών Πόρων (URLs). Συνεπώς σε αυτό το σημείο ορίζονται τα URLs που χρειάστηκαν για τις ανάγκες της εφαρμογής.

- **url(r'^\$', views.mainapp_index, name='mainapp_index')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000` με την όψη `mainapp_index`.
- **url(r'^instructions/', views.instructions, name='instructions')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/instructions` με την όψη `instructions`.
- **url(r'^info/', views.info, name='info')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/info` με την όψη `info`.

Με την ολοκλήρωση της κατασκευής των URLs επιτυγχάνεται η ευέλικτη περιήγηση του χρήστη στην εφαρμογή καθώς και η παροχή σημαντικών πληροφοριών σε αυτόν.

5.4 Σχεδίαση και Υλοποίηση της Εφαρμογής Δέντρων Αποφάσεων

Η βασική λειτουργία της Εφαρμογής Δέντρων Αποφάσεων είναι να δίνει τη δυνατότητα στον χρήστη, μέσα σε ένα διαδραστικό και φιλικό προς αυτόν περιβάλλον, να εφαρμόζει τη μέθοδο των Δέντρων Αποφάσεων. Θα επιλέγει λοιπόν τον τρόπο που θα παρέχει τα δεδομένα του. Στη συνέχεια, η εφαρμογή θα επεξεργάζεται και θα αποθηκεύει τα δεδομένα αυτά, εμφανίζοντας στον χρήστη το δέντρο αποφάσεων επιλυμένο. Στην ενότητα αυτή θα αναλυθούν οι σχεδιαστικές επιλογές και οι υλοποιήσεις που έγιναν για την επίτευξη των παραπάνω λειτουργιών.

5.4.1 Συλλογή, Επεξεργασία και Αποθήκευση Δεδομένων

Για τη διευκόλυνση και την παροχή ευελιξίας στον χρήστη, του δίνεται η δυνατότητα να επιλέξει ανάμεσα σε δύο τρόπους παροχής των δεδομένων του. Ο πρώτος είναι μέσω ενός αρχείου JSON και ο δεύτερος κατασκευάζοντας πάνω στην εφαρμογή το δέντρο αποφάσεων που επιθυμεί να επιλύσει. Ανάλογα με τον τρόπο που θα επιλέξει γίνεται και διαφορετικά η συλλογή δεδομένων.

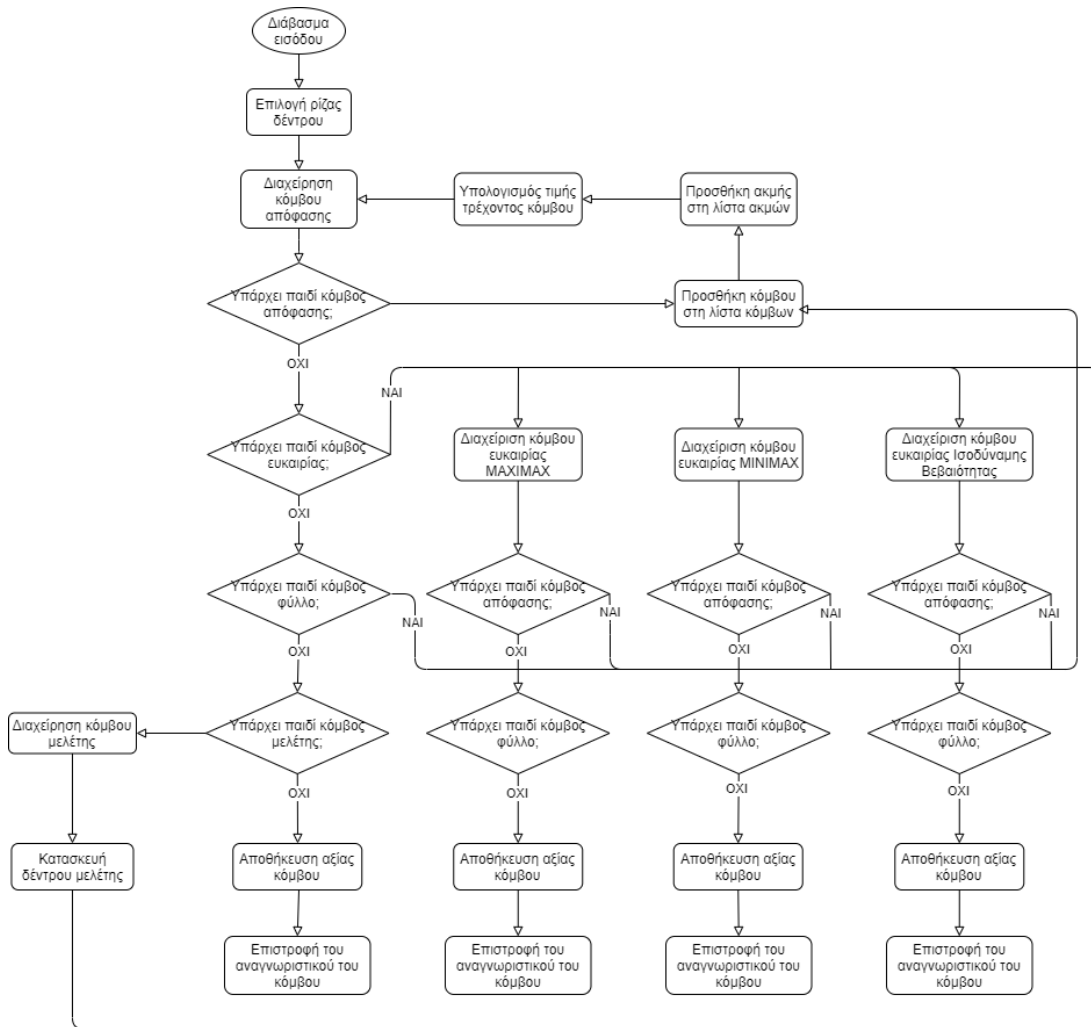
Για να μπορεί ο χρήστης να παρέχει τα δεδομένα του μέσω ενός JSON αρχείου, κατασκευάστηκε μια φόρμα στην οποία μπορεί, εφόσον το επιθυμεί, να δηλώσει τον τίτλο του προβλήματος που θέλει να επιλύσει και να επιλέξει από τον τοπικό του ηλεκτρονικό υπολογιστή το αρχείο εισόδου που περιέχει το Δέντρο Αποφάσεων που είναι προς επίλυση. Για τη διασφάλιση της εγκυρότητας των δεδομένων του χρήστη κατασκευάστηκαν δύο επίπεδα ελέγχου. Το πρώτο, κατά την επιλογή του αρχείου από τον χρήστη, ελέγχει τον τύπο του αρχείου. Ο χρήστης έχει τη δυνατότητα να ανεβάσει σε αυτό το σημείο μόνο αρχείου τύπου JSON (.json). Το δεύτερο επίπεδο ελέγχου εφαρμόζεται εφόσον ο χρήστης επιλέξει είτε να επεξεργαστεί το δέντρο αποφάσεων που περιέχεται στο αρχείο που ανέβασε στην εφαρμογή, είτε να επιλύσει το δέντρο αυτό. Σε αυτό το σημείο ελέγχεται κατά πόσο το περιεχόμενο του αρχείου του χρήστη επιβεβαιώνει το σχήμα JSON που έχει οριστεί. Έτσι διασφαλίζεται ότι ο αλγόριθμος επίλυσης που κατασκευάστηκε στη συνέχεια θα βρει τα απαιτούμε δεδομένα ώστε να εκτελεστεί ορθά. Έτσι ο χρήστης μπορεί να προχωρήσει είτε στην επεξεργασία του είτε στην άμεση επίλυση του δέντρου του.

Σε άλλη περίπτωση που χρήστης επιλέξει να κατασκευάσει το δέντρο αποφάσεων προς επίλυση πάνω στη εφαρμογή δεν χρειάζεται επικύρωση των δεδομένων καθώς η εγκυρότητά τους διασφαλίζεται από την ίδια την εφαρμογή. Οπότε με την εντολή εκτέλεσης του χρήστη προωθούνται προς επίλυση.

Εφόσον η μορφή των δεδομένων είναι συμβατή με αυτή που έχει οριστεί προχωράει η επεξεργασία τους. Τα δεδομένα τα οποία είναι τώρα σε μορφή JSON μετατρέπονται σε python λεξικό. Σε αυτό το σημείο τα δεδομένα εισόδου αποτελούνται από δύο λίστες. Η μια περιέχει πληροφορίες για τους κόμβους του δέντρου και η άλλη πληροφορίες για τις ακμές. Για να είναι τα δεδομένα αυτά ευκολότερα διαχειρίσιμα, κατασκευάστηκε αλγόριθμος που τις δύο λίστες τις μετατρέπει σε ένα python λεξικό το οποίο περιέχει όλες τις πληροφορίες των κόμβων και των ακμών δομημένες σε μορφή δέντρου. Αυτό έγινε εφικτό με τη χρήση εμφωλευμένων python λεξικών.

Έπεται η επίλυση του προβλήματος δέντρων αποφάσεων. Για τη λειτουργία αυτή κατασκευάστηκε αλγόριθμος που επιλύει τέτοια προβλήματα. Στην *Εικόνα 9* φαίνεται το διάγραμμα ροής του αλγορίθμου. Ο αλγόριθμος δέχεται ως είσοδο το python λεξικό με τα δεδομένα εισόδου κατασκευασμένα σε μορφή δέντρου. Τα δεδομένα εξόδου του είναι δυο λίστες python λεξικών. Η μια περιέχει όλες τις πληροφορίες των κόμβων και

η άλλη όλες τις πληροφορίες των ακμών του επιλυμένου δέντρου αποφάσεων. Ο αλγόριθμος εκκινεί μια αναδρομική διαδικασία ώστε να διατρεχθούν όλοι οι κόμβοι του δέντρου, εκκινώντας από τη ρίζα του δέντρου η οποία στα δέντρα αποφάσεων είναι και κόμβος απόφασης. Συνεπώς αρχικά μεταβαίνει σε μια διαδικασία διαχείρισης κόμβων απόφασης. Σε κάθε κόμβο ο αλγόριθμός ελέγχει αν και τι είδους παιδί διαθέτει. Ανάλογα με το είδος του κόμβου-παιδιού ο αλγόριθμος μεταβαίνει και σε διαφορετική διαδικασία. Εξάριση αποτελεί η περίπτωση που έχουμε κόμβο-φύλλο όπου δεν γίνεται μετάβαση σε κάποια διαδικασία. Για κόμβο απόφασης ο αλγόριθμος μεταβαίνει εκ νέου σε διαδικασία διαχείρισης κόμβων αποφάσεων. Για κόμβο ευκαιρίας ελέγχεται το είδος του κριτηρίου που έχει οριστεί και γίνεται μετάβαση στην αντίστοιχη διαδικασία. Τα κριτήρια που καλύπτονται από την εφαρμογή είναι τα Ισοδύναμης Αβεβαιότητας, MAXIMIN και MAXIMAX. Σε κάθε είδους διαδικασία διαχείρισης κόμβου ευκαιρίας, για κάθε παιδί του κόμβου ευκαιρίας, ο αλγόριθμος μεταβαίνει σε αντίστοιχη διαδικασία διαχείρισης, όπως έχει περιγραφεί πιο πάνω, ανάλογα με τον τύπο του κόμβου-παιδιού. Όταν όλες οι διαδικασίες που δημιουργήθηκαν στον κόμβο ευκαιρίας επιστρέψουν καταγράφεται ο κάθε κόμβος-παιδί στη λίστα των κόμβων καθώς και οι ακμές τους στη λίστα ακμών. Στη συνέχεια υπολογίζεται η τιμή του κόμβου ευκαιρίας ανάλογα με το κριτήριό του και η διαδικασία διαχείρισης κόμβου ευκαιρίας επιστρέφει. Ενδιαφέρον έχει η υλοποίηση στην περίπτωση κόμβου μελέτης. Στο σημείο αυτό να αναφερθεί ότι στα δεδομένα του χρήστη η μελέτη δεν απαιτείται να είναι σε μορφή υποδέντρου, αλλά απλώς να παρέχει σε μορφή πινάκων τα δεδομένα αυτής. Έτσι λοιπόν ο αλγόριθμος κατασκευάζει το υποδέντρο της μελέτης βασισμένος στις πληροφορίες που παρείχε ο χρήστης. Για τους κόμβους που δημιουργήθηκαν στο υποδέντρο της μελέτης ακολουθείται η ίδια διαδικασία με τους υπόλοιπους κόμβους, όπως αναλύθηκε πιο πάνω. Έπειτα, όταν όλες οι διαδικασίες που παρήγαγε ο κόμβος απόφασης επιστρέψουν, καταγράφονται οι κόμβοι-παιδιά του στη λίστα των κόμβων και οι ακμές του με αυτά στη λίστα ακμών. Τέλος, υπολογίζεται η τιμή του κόμβου απόφασης επιλέγοντας το παιδί με τη μεγαλύτερη τιμή και η διαδικασία επιστρέφει. Όταν ολοκληρωθούν οι παραπάνω διαδικασίες, στις λίστες των κόμβων και των ακμών θα περιέχονται όλες οι απαραίτητες πληροφορίες για το επιλυμένο τελικά δέντρο αποφάσεων.



Εικόνα 9- Αλγόριθμος επίλυσης δέντρων αποφάσεων

Έχοντας ολοκληρωθεί ο αλγόριθμος επίλυσης των δέντρων αποφάσεων, έχουν δημιουργηθεί όλα τα απαραίτητα δεδομένα για τη δημιουργία εγγραφής στη βάση δεδομένων με τη χρήση του Μοντέλου που έχει δημιουργηθεί στη Βασική Εφαρμογή. Η εγγραφή αυτή περιλαμβάνει τον τίτλο που δόθηκε από τον χρήστη, το αρχείο εισόδου σε μορφή JSON πριν τη μετατροπή του σε μορφή δέντρου, το αρχείο εξόδου σε μορφή JSON, όπως αυτό επιστράφηκε από τον αλγόριθμο επίλυσης και το λεκτικό “Δέντρα Αποφάσεων” ως ο τύπος της μεθόδου που χρησιμοποιήθηκε. Η ταυτότητα και η ημερομηνία της εγγραφής συμπληρώνονται αυτόματα από τη βάση δεδομένων.

Με αυτόν τον τρόπο ολοκληρώνεται η διαδικασία συλλογής, επεξεργασίας και αποθήκευσης των δεδομένων για ένα πρόβλημα Δέντρων Αποφάσεων που έθεσε ο χρήστης προς επίλυση.

5.4.2 Λειτουργικότητα της Εφαρμογής

Για να αποκτήσει η Εφαρμογή των Δέντρων Αποφάσεων λειτουργικότητα και να καταστεί εφικτό στον χρήστη να περιηγηθεί και να αξιοποιήσει τις λειτουργίες της είναι αναγκαία η κατασκευή των όψεων. Στη συνέχεια παρουσιάζονται οι όψεις που δημιουργήθηκαν καθώς και η λειτουργία που επιτελούν.

dectree_index: Η εν λόγω όψη δέχεται μόνο GET αίτημα από τον χρήστη και του φορτώνει την αρχική σελίδα της εφαρμογής, όπου μπορεί να επιλέξει με ποιο τρόπο θέλει να παρέχει τα δεδομένα του.

upload_form: Σε GET αίτημα η όψη επιστρέφει τη σελίδα με τη φόρμα συμπλήρωσης των δεδομένων εισόδου. Τα POST αιτήματα που μπορεί να δεχθεί η όψη είναι δύο ειδών. Αν στο αίτημα περιέχεται το λεκτικό “run” επιστρέφει GET αίτημα, με παράμετρο το αναγνωριστικό της εγγραφής που περιλαμβάνει τα δεδομένα προς εμφάνιση, στην όψη results για να εμφανίσει τα αποτελέσματα της εγγραφής αυτής. Αν στο αίτημα περιέχεται το λεκτικό “manage” επιστρέφει GET αίτημα, με παράμετρο το αναγνωριστικό της εγγραφής που περιλαμβάνει τα δεδομένα προς επεξεργασία, στην όψη manage ώστε ο χρήστης στη συνέχεια να επεξεργαστεί τα δεδομένα εισόδου που ανάρτησε.

results: Δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής της οποίας καλείται να εμφανίσει τα αποτελέσματα. Σε GET αίτημα η όψη ανακαλεί από τη βάση δεδομένων την εγγραφή που αντιστοιχεί στο αναγνωριστικό που δόθηκε ως παράμετρος. Στη συνέχεια συλλέγει τα δεδομένα εξόδου της εγγραφής και τα παρέχει στη σελίδα παρουσίασης αποτελεσμάτων την οποία και φορτώνει στον χρήστη. Τα POST αιτήματα που μπορεί να δεχθεί η όψη είναι δύο ειδών. Αν περιέχεται το λεκτικό “new” στο αίτημα, η όψη στέλνει ένα GET αίτημα στην dectree_index για την επίλυση νέου προβλήματος δέντρου αποφάσεων από τον χρήστη. Αν περιέχεται το λεκτικό “inspect” στο αίτημα, η όψη στέλνει ένα GET αίτημα, με παράμετρο το αναγνωριστικό της εγγραφής της οποίας τα αποτελέσματα κλήθηκε να εμφανίσει η όψη, στην όψη manage ώστε ο χρήστης να επεξεργαστεί εκ νέου το δέντρο που χρησιμοποιήσε ως είσοδο.

manage: Δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής της οποίας τα δεδομένα εισόδου θα υποστούν επεξεργασία από τον χρήστη. Σε GET αίτημα η όψη επιστρέφει στον χρήστη τη σελίδα στην οποία περιέχεται το εργαλείο απεικόνισης επεξεργασίας ενός δέντρου αποφάσεων με παραμέτρους τα δεδομένα εισόδου της εγγραφής που είναι προς επεξεργασία. Σε POST αίτημα η όψη στέλνει GET αίτημα, με παράμετρο το αναγνωριστικό της εγγραφής της οποίας τα δεδομένα εισόδου έχουν υποστεί επεξεργασία, στην όψη results ώστε να εμφανίσει τα νέα αποτελέσματα του ανανεωμένου προβλήματος δέντρων αποφάσεων.

create_tree: Σε GET αίτημα η όψη φορτώνει στον χρήστη τη σελίδα με το εργαλείο κατασκευής δέντρου αποφάσεων. Σε POST αίτημα η όψη στέλνει GET αίτημα στην όψη results, με παράμετρο το αναγνωριστικό της εγγραφής που είναι προς εμφάνιση.

ajax_create_and_calculate: Πρόκειται για μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα δημιουργίας νέου δέντρου. Το αίτημα περιέχει τα δεδομένα εισόδου που δημιούργησε ο χρήστης στο εργαλείο κατασκευής δέντρων αποφάσεων και ο τίτλος που έδωσε ο χρήστης. Τα δεδομένα συλλέγονται από την όψη. Τα δεδομένα εισόδου είναι σε μορφή JSON. Μετατρέπονται σε ένα rython λεξικό το οποίο περιέχει τις λίστες κόμβων και ακμών. Με τον αλγόριθμο που αναλύθηκε πιο πάνω μετατρέπονται σε μορφή δέντρου και δίνονται ως είσοδο στον αλγόριθμο επίλυσης Δέντρων Αποφάσεων. Η όψη στη συνέχεια δημιουργεί μια νέα εγγραφή στη βάση δεδομένων. Τέλος, η όψη επιστρέφει κωδικό επιτυχίας πίσω στη σελίδα από την οποία καλέστηκε.

ajax_calculate: Πρόκειται για μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα επεξεργασίας ενός υπάρχοντος δέντρου αποφάσεων. Το αίτημα περιέχει τα δεδομένα εισόδου που επεξεργάστηκε ο χρήστης στο εργαλείο επεξεργασίας δέντρων αποφάσεων, ο τίτλος που ενδεχομένως να άλλαξε ο χρήστης και το αναγνωριστικό της εγγραφής της οποίας τα δεδομένα εισόδου ο χρήστης επεξεργάστηκε. Τα δεδομένα συλλέγονται από την όψη. Τα δεδομένα εισόδου είναι σε μορφή JSON. Μετατρέπονται σε ένα rython λεξικό το οποίο περιέχει τις λίστες κόμβων και ακμών. Με τον αλγόριθμο που αναλύθηκε πιο πάνω μετατρέπονται σε μορφή δέντρου και δίνονται ως είσοδο στον αλγόριθμο επίλυσης Δέντρων Αποφάσεων. Έπειτα η όψη τροποποιεί τα πεδία της εγγραφής με βάση τα νέα.

ajax_upload_manage: Πρόκειται για μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα που ο χρήστης ανεβάζει σε μορφή αρχείου τα δεδομένα του εφόσον ο χρήστης έχει επιλέξει τον τίτλο, το αρχείο JSON που επιθυμεί να ανεβάσει στην εφαρμογή και την επεξεργασία των δεδομένων αυτών. Το αίτημα περιλαμβάνει το αρχείο που επέλεξε ο χρήστης και τον τίτλο που όρισε. Τα δεδομένα συλλέγονται από την όψη. Τα περιεχόμενα του αρχείου JSON διαβάζονται και μετατρέπονται σε rython λεξικό. Στη συνέχεια ελέγχεται αν ικανοποιούν το σχήμα JSON που έχει οριστεί. Αυτή η επικύρωση επιτυγχάνεται με χρήση του Draft7Validator από τη βιβλιοθήκη jsonschema. Αν η επικύρωση επιτύχει, δημιουργείται νέα εγγραφή στη βάση περιέχοντας τον τίτλο και τα δεδομένα που παρείχε ο χρήστης. Τέλος επιστρέφει το αναγνωριστικό της εγγραφής και κατάσταση επιτυχίας. Αν η επικύρωση των δεδομένων αποτύχει, η όψη επιστρέφει κατάσταση αποτυχίας.

ajax_upload_run: Πρόκειται για μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα που ο χρήστης ανεβάζει σε μορφή αρχείου τα δεδομένα του εφόσον ο χρήστης έχει επιλέξει τον τίτλο, το αρχείο JSON που επιθυμεί να ανεβάσει στην εφαρμογή και την εκτέλεση των δεδομένων αυτών. Το αίτημα περιλαμβάνει το αρχείο που επέλεξε ο χρήστης και τον τίτλο που όρισε. Τα δεδομένα συλλέγονται από την όψη. Έπειτα τα

περιεχόμενα του αρχείου JSON διαβάζονται και μετατρέπονται σε python λεξικό και ελέγχεται αν ικανοποιούν το σχήμα JSON που έχει οριστεί. Όπως και στην προηγούμενη όψη, έτσι κι εδώ η επικύρωση πραγματοποιείται με χρήση του Draft7Validator. Αν η επικύρωση ολοκληρωθεί με επιτυχία, δημιουργείται νέα εγγραφή στη βάση περιέχοντας τον τίτλο και τα δεδομένα που παρείχε ο χρήστης. Στη συνέχεια εκτελείται ο αλγόριθμος επίλυσης Δέντρων Αποφάσεων και το αποτέλεσμα του προστίθεται κι αυτό στην εγγραφή της βάσης που μόλις δημιουργήθηκε. Τέλος η όψη επιστρέφει το αναγνωριστικό της μεθόδου που δημιούργησε και κατάσταση επιτυχίας. Αν η επικύρωση των δεδομένων αποτύχει, η όψη επιστρέφει κατάσταση αποτυχίας.

5.4.3 Διασύνδεση Λειτουργικότητας με τον χρήστη

Για να μπορεί η λειτουργικότητα που αναλύθηκε πιο πάνω να είναι προσβάσιμη από τον χρήστη κατασκευάστηκαν Ενιαίοι Εντοπιστές Πόρων (URLs). Πριν την κατασκευή τους απαιτείται η σύνδεση τους με τα υπόλοιπα URLs του έργου. Αυτό επιτυγχάνεται δηλώνοντας το εξής μονοπάτι στο αρχείο urls.py του συνολικού έργου: `path('dectree/', include('dectree.urls'))`. Στη συνέχεια, κάθε URL συνδέεται με την αντίστοιχη όψη που θα παρέχει στον χρήστη την επιθυμητή λειτουργικότητα πάνω στη διαδικτυακή εφαρμογή. Επιπλέον σε αυτό το σημείο είναι απαραίτητο να δηλωθούν τα URLs που συνδέουν τις AJAX κλήσεις με τις αντίστοιχες όψεις τους. Τα URLs αυτά δεν παρέχουν κάποιο περιεχόμενο στον χρήστη αλλά επιτελούν εσωτερική λειτουργία της εφαρμογής. Για τις ανάγκες της Εφαρμογής των Δέντρων Αποφάσεων δημιουργήθηκαν τα παρακάτω URLs και δηλώθηκαν στο αρχείο urls.py της εφαρμογής:

- **url(r'^\$', views.dectree_index, name='dectree_index')**:
Συνδέει το μονοπάτι `http://127.0.0.1:8000/dectree` με την όψη `dectree_index` και αποτελεί την αρχική σελίδα της εφαρμογής.
- **url(r'^upload/', views.upload_form, name='dectree_upload')**:
Συνδέει το μονοπάτι `http://127.0.0.1:8000/dectree/upload` με την όψη `upload_form` και αποτελεί τη σελίδα όπου ο χρήστης μπορεί να συμπληρώσει τη φόρμα με τον τίτλο του προβλήματός του και το αρχείο εισόδου.
- **path('manage/<int:method_id>', views.manage, name='manage')**:
Συνδέει το μονοπάτι `http://127.0.0.1:8000/dectree/manage` με την όψη `manage`. Δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων για την οποία ο χρήστης επιθυμεί να επεξεργαστεί το αρχείο εισόδου. Η σελίδα αυτή περιλαμβάνει το εργαλείο επεξεργασίας δέντρων αποφάσεων όπου ο χρήστης μπορεί να επεξεργαστεί τα δεδομένα εισόδου.
- **url(r'^create/', views.create_tree, name='create_tree')**:

Συνδέει το μονοπάτι `http://127.0.0.1:8000/dectree/create` με την όψη `create_tree` και παρέχει στον χρήστη τη σελίδα με το εργαλείο δημιουργίας και επεξεργασίας δέντρων αποφάσεων.

- **`path('results/<int:method_id>', views.results, name='results')`:**
Συνδέει το μονοπάτι `http://127.0.0.1:8000/dectree/results` με την όψη `results`. Δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας ο χρήστης επιθυμεί να δει τα αποτελέσματα. Η σελίδα που παρέχεται στον χρήστη περιλαμβάνει την απεικόνιση της λύσης του προβλήματος που έθεσε προς επίλυση.
- **`url(r'^ajax/calculate', views.ajax_calculate, name='ajax_calculate')`:**
Συνδέει την AJAX κλήση `calculate` με την όψη `ajax_calculate`.
- **`url(r'^ajax/create_and_calculate', views.ajax_create_and_calculate, name='ajax_create_and_calculate')`:** Συνδέει την AJAX κλήση `calculate_and_create` με την όψη `ajax_create_and_calculate`.
- **`url(r'^ajax/upload_manage', views.ajax_upload_manage, name='ajax_upload_manage')`:** Συνδέει την AJAX κλήση `upload_manage` με την όψη `ajax_upload_manage`.
- **`url(r'^ajax/upload_run', views.ajax_upload_run, name='ajax_upload_run')`:** Συνδέει την AJAX κλήση `upload_run` με την όψη `ajax_upload_run`.

Μετά και τη δημιουργία των παραπάνω URLs η εφαρμογή είναι έτοιμη να χρησιμοποιηθεί από τον χρήστη και μέσω αυτών να περιηγηθεί σε αυτή και να αξιοποιήσει τις λειτουργικότητές της.

5.5 Σχεδίαση και Υλοποίηση της Εφαρμογής Γραμμικού Προγραμματισμού

Η Εφαρμογή του Γραμμικού Προγραμματισμού έχει ως στόχο να παρέχει στον χρήστη ένα εργαλείο πάνω στο οποίο θα μπορεί να λύνει προβλήματα Γραμμικού Προγραμματισμού. Η εφαρμογή χωρίζεται σε δύο σκέλη, ανάλογα με το πλήθος των μεταβλητών του προβλήματος που επιθυμεί να λύσει ο χρήστης. Στην περίπτωση δύο μεταβλητών το πρόβλημα λύνεται με τη μέθοδο Simplex και παρουσιάζεται γραφικά στον χρήστη. Ενώ στην περίπτωση περισσότερων των δύο μεταβλητών το πρόβλημα λύνεται με τη μέθοδο Simplex και παρουσιάζονται στον χρήστη πλήρως οι επαναλήψεις Simplex που χρειάστηκαν για το πρόβλημα. Ακολουθεί η ανάλυση των σχεδιαστικών αποφάσεων και πως αυτές υλοποιήθηκαν στο πλαίσιο της εφαρμογής.

5.5.1 Συλλογή, Επεξεργασία και Αποθήκευση Δεδομένων

Η συλλογή των δεδομένων μπορεί να γίνει με δύο διαφορετικούς τρόπους, όπου είναι στην επιλογή του χρήστη ποιον θα διαλέξει. Ο ένας είναι ανεβάζοντας ένα αρχείο json στην εφαρμογή και ο άλλος είναι κατασκευάζοντας την αντικειμενική συνάρτηση, τους περιορισμούς και τις δραστηριότητες του προβλήματος πάνω στην εφαρμογή.

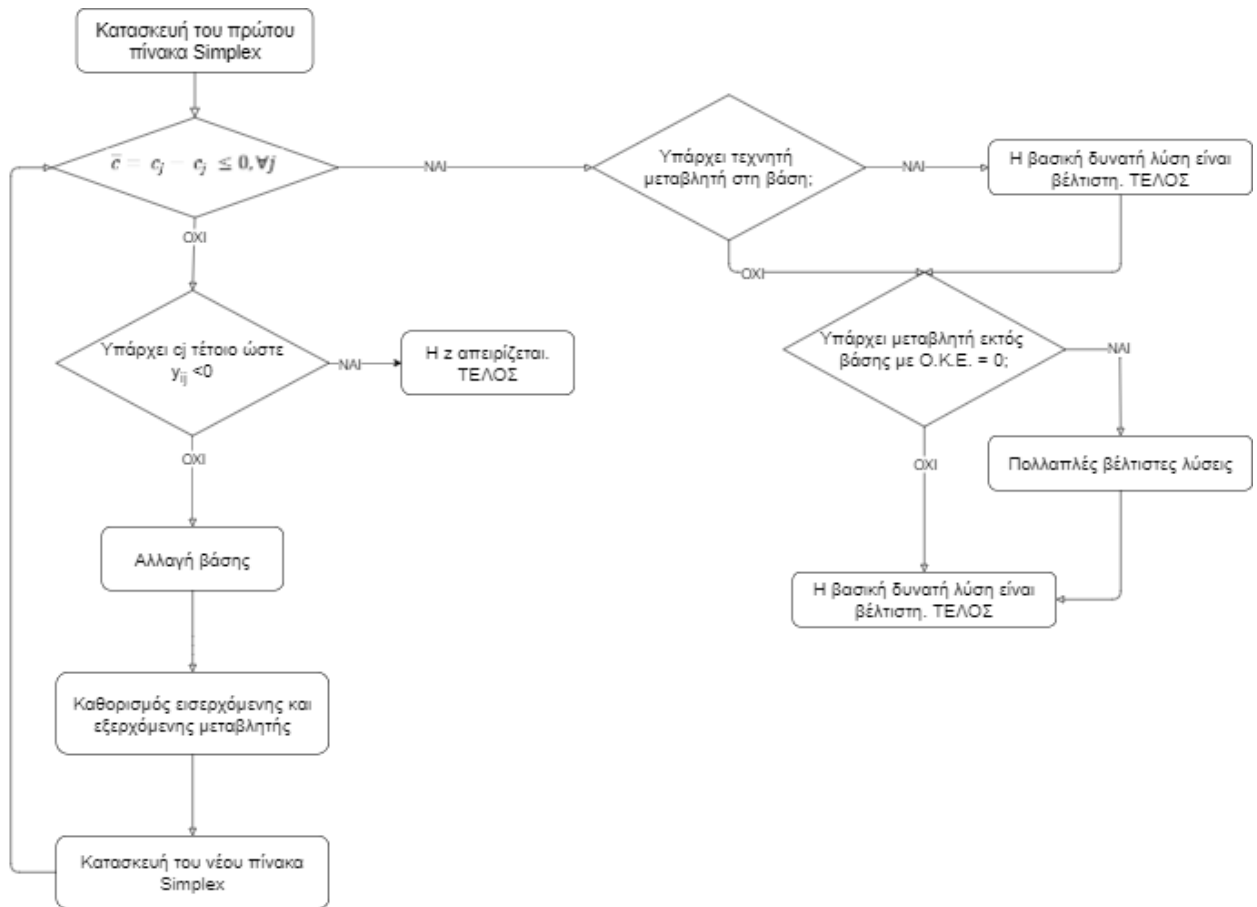
Για να μπορεί ο χρήστης να παράσχει τα δεδομένα του μέσω ενός αρχείου JSON δημιουργήθηκε μια φόρμα. Σε αυτή θα μπορεί να συμπληρώσει τον τίτλο του προβλήματός του, εφόσον το επιθυμεί και να επιλέξει ποιο αρχείο επιθυμεί να ανεβάσει στην εφαρμογή. Για τη διασφάλιση της εγκυρότητας των δεδομένων του χρήστη κατασκευάστηκαν δύο επίπεδα ελέγχου. Το πρώτο, κατά την επιλογή του αρχείου από τον χρήστη, ελέγχει τον τύπο του αρχείου. Ο χρήστης έχει τη δυνατότητα να ανεβάσει σε αυτό το σημείο μόνο αρχείου τύπου JSON (.json). Το δεύτερο επίπεδο ελέγχου εφαρμόζεται εφόσον ο χρήστης επιλέξει είτε να επεξεργαστεί τα δεδομένα εισόδου που περιέχονται στο αρχείο που ανέβασε στην εφαρμογή, είτε να επιλύσει το πρόβλημα Γραμμικού Προγραμματισμού. Σε αυτό το σημείο ελέγχεται κατά πόσο το περιεχόμενο του αρχείου του χρήστη επιβεβαιώνει το σχήμα JSON που έχει οριστεί. Έτσι διασφαλίζεται ότι ο αλγόριθμος επίλυσης που κατασκευάστηκε στη συνέχεια θα βρει τα απαιτούμενα δεδομένα ώστε να εκτελεστεί ορθά. Έτσι στη συνέχεια ο χρήστης μπορεί να προχωρήσει είτε στην επεξεργασία του είτε στην άμεση επίλυση του δέντρου του.

Αφού τα δεδομένα έχουν συλλεχθεί και ελεγχθεί ως προς την ορθότητά τους, προχωράει η επεξεργασία τους και έπειτα η επίλυση του προβλήματος Γραμμικού Προγραμματισμού. Η μόνη επεξεργασία που χρειάζονται σε αυτό το σημείο είναι η μετατροπή τους από αντικείμενο JSON σε python λεξικό και τροφοδοτούνται σε διαφορετικό αλγόριθμο επίλυσης ανάλογα με το πλήθος των μεταβλητών του προβλήματος.

Για την περίπτωση που το πρόβλημα περιλαμβάνει δύο μεταβλητές χρησιμοποιήθηκε η συνάρτηση `linprog` που περιέχεται στη βιβλιοθήκη `scipy.optimize` [44] της python. Έτσι το μόνο που απαιτείται είναι η μετατροπή των δεδομένων εισόδου ώστε να είναι συμβατά με αυτά που απαιτούνται από την παραπάνω συνάρτηση. Η συνάρτηση `linprog` επιστρέφει ένα python λεξικό που περιλαμβάνει το αποτέλεσμα της.

Για την περίπτωση που το πρόβλημα περιλαμβάνει 3 έως 8 μεταβλητές καταστρώθηκε αλγόριθμος Simplex που επιλύει το πρόβλημα με την τεχνική του μεγάλου M. Η επιλογή αυτή έγινε καθώς ο χρήστης μπορεί να επιθυμεί να περιλαμβάνει στα δεδομένα του δραστηριότητες με τελεστές “ \geq ” και “ $=$ ”. Πρώτο βήμα είναι η μετατροπή των δεδομένων εισόδου σε πρότυπη μορφή. Συνεπώς καταγράφονται οι τεχνικές μεταβλητές A και οι μεταβλητές απόκλισης S που πρέπει να προστεθούν. Αυτό επιτυγχάνεται ελέγχοντας το είδος των τελεστών των δραστηριοτήτων. Έπειτα

προστίθεται ο συντελεστής M (πολύ μεγάλος αριθμός). Στον συγκεκριμένο αλγόριθμο η τιμή του M ισούται με 100000. Το πρόσημο του συντελεστή ορίζεται από το είδος του προβλήματος Γραμμικού Προγραμματισμού. Αν πρόκειται για πρόβλημα ελαχιστοποίησης ο συντελεστής προστίθεται με θετικό πρόσημο, ενώ με αρνητικό προστίθεται στα προβλήματα μεγιστοποίησης. Αφού γίνουν οι παραπάνω παρεμβάσεις στα δεδομένα, ακολουθεί η επίλυση του προβλήματος με την υλοποίηση του αλγορίθμου Simplex. Το διάγραμμα ροής του αλγορίθμου Simplex φαίνεται στην *Εικόνα 10*.



Εικόνα 10- Αλγόριθμος simplex

Αφού ολοκληρωθεί η επίλυση του προβλήματος Γραμμικού Προγραμματισμού, τα αποτελέσματα επιστρέφουν ώστε να αποθηκευτούν στο πεδίο `output_file` της εγγραφής της βάσεις. Έτσι θα είναι στη συνέχεια διαθέσιμα προς εμφάνιση.

Μια ακόμα επεξεργασία που γίνεται στα δεδομένα η οποία ωστόσο δεν αποθηκεύεται στη βάση δεδομένων είναι στην περίπτωση των δύο μεταβλητών, στο στάδιο παρουσίασης των αποτελεσμάτων, όπου είναι αναγκαία η κατασκευή των σημείων

των γραφικών παραστάσεων, ώστε να σχεδιαστούν σωστά με τη χρήση του εργαλείου chart.js στον χρήστη.

5.5.2 Λειτουργικότητα της Εφαρμογής

Αφού καταστρώθηκαν οι αλγόριθμοι και οι διαδικασίες συλλογής, επεξεργασίας και αποθήκευσης των δεδομένων, επόμενο βήμα είναι η δημιουργία των όψεων ώστε να εξασφαλιστεί η ορθή λειτουργικότητα της εφαρμογής και η διασύνδεση των διαδικασιών που εκτελούνται στον εξυπηρετητή με την αλληλεπίδραση του χρήστη με την εφαρμογή. Παρακάτω παρουσιάζονται οι όψεις που κατασκευάστηκαν για τις ανάγκες της εφαρμογής καθώς και οι λειτουργίες που επιτελούν.

linear_index: Αυτή η όψη δέχεται μόνο GET αίτημα από τον χρήστη και φορτώνει την αρχική σελίδα της εφαρμογής, όπου επιλέγει με ποιο τρόπο επιθυμεί να παρέχει τα δεδομένα του.

upload_form: Σε GET αίτημα η όψη επιστρέφει τη σελίδα με τη φόρμα συμπλήρωσης των δεδομένων εισόδου. Τα POST αιτήματα που μπορεί να δεχθεί η όψη είναι δύο ειδών. Αν στο αίτημα περιλαμβάνεται το λεκτικό “run”, η όψη επιστρέφει GET αίτημα στην όψη results, με παράμετρο το αναγνωριστικό της εγγραφής της βάσης δεδομένων που περιλαμβάνει τα δεδομένα προς παρουσίαση. Αν στο αίτημα περιλαμβάνεται το λεκτικό “manage”, η όψη επιστρέφει GET αίτημα στην όψη manage, με παράμετρο το αναγνωριστικό της εγγραφής της βάσης δεδομένων που περιέχει τα δεδομένα εισόδου που είναι προς επεξεργασία.

results: Η εν λόγω όψη δέχεται GET αίτημα και περιμένει ως παράμετρο το αναγνωριστικό της εγγραφής της βάσης δεδομένων η οποία είναι προς παρουσίαση. Αν στα δεδομένα του πεδίου input_file της εγγραφής ο αριθμός των μεταβλητών του προβλήματος Γραμμικού Προγραμματισμού είναι ίσος με 2 εκτελείται ο αλγόριθμος κατασκευής των δεδομένων για τις γραφικές παραστάσεις. Έπειτα φορτώνεται η σελίδα παρουσίασης αποτελεσμάτων με γραφική αναπαράσταση της λύσης. Αν στα δεδομένα του πεδίου input_file της εγγραφής ο αριθμός των μεταβλητών είναι από 3 έως 8 φορτώνεται η σελίδα παρουσίασης των αποτελεσμάτων περιλαμβάνοντας όλους τους πίνακες simplex που κατασκευάστηκαν κατά της επίλυση του προβλήματος. Σε POST αίτημα η όψη ανταποκρίνεται ανάλογα με το λεκτικό που θα περιλαμβάνει το POST αίτημα. Αν περιλαμβάνει το λεκτικό “new”, η όψη αποστέλλει GET αίτημα στην όψη dynamic_index για την επίλυση νέου προβλήματος. Αν περιλαμβάνει το λεκτικό “inspect”, η όψη αποστέλλει GET αίτημα στην όψη manage έχοντας ως παράμετρο το αναγνωριστικό της εγγραφής της βάσης δεδομένων της οποίας τα δεδομένα είναι προς επεξεργασία.

manage: Η όψη αυτή δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής της οποίας τα δεδομένα εισόδου είναι προς επεξεργασία από τον χρήστη. Σε GET αίτημα

η όψη επιστρέφει τη σελίδα επεξεργασίας δεδομένων Γραμμικού Προγραμματισμού περιέχοντας τα δεδομένα του πεδίου `input_file` της παραπάνω εγγραφής ώστε να συμπεριληφθούν στη σελίδα. Σε POST αίτημα η όψη αποστέλλει GET αίτημα, με παράμετρο το αναγνωριστικό της εγγραφής της οποίας τα δεδομένα εισόδου έχουν υποστεί επεξεργασία, στην όψη `results` ώστε να εμφανίσει τα νέα αποτελέσματα του επικαιροποιημένου προβλήματος Γραμμικού Προγραμματισμού.

create_data: Σε GET αίτημα η όψη φορτώνει στον χρήστη τη σελίδα με το εργαλείο κατασκευής ενός προβλήματος Γραμμικού Προγραμματισμού. Σε POST αίτημα η όψη στέλνει GET αίτημα στην όψη `results`, με παράμετρο το αναγνωριστικό της εγγραφής που είναι προς εμφάνιση.

ajax_create_and_calculate: Είναι μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα δημιουργίας ενός νέου προβλήματος Γραμμικού Προγραμματισμού. Το αίτημα περιέχει τα δεδομένα εισόδου που δημιούργησε ο χρήστης πάνω στην εφαρμογή καθώς και ο τίτλος που όρισε. Τα δεδομένα είναι σε μορφή JSON. Αυτά συλλέγονται και μετατρέπονται σε ένα `rython` λεξικό που περιλαμβάνει όλα τα απαραίτητα δεδομένα για την εκτέλεση του αλγορίθμου επίλυσης προβλημάτων Γραμμικού Προγραμματισμού. Ανάλογα με το πλήθος των μεταβλητών του προβλήματος εκτελείται και ο αντίστοιχος αλγόριθμος επίλυσης προβλημάτων Γραμμικού Προγραμματισμού. Έπειτα το αποτέλεσμα αυτού, μαζί με τα δεδομένα εισόδου, αποθηκεύονται στη βάση δεδομένων δημιουργώντας μια νέα εγγραφή. Εξαιρεση αποτελεί η περίπτωση των δύο μεταβλητών όπου αν ο αλγόριθμος δεν βρει λύση στο πρόβλημα. Σε αυτή την περίπτωση η όψη επιστρέφει κατάσταση σφάλματος καθώς και τον μήνυμα που επεξηγεί γιατί δεν βρήκε λύση στο πρόβλημα ο αλγόριθμος. Στην άλλη περίπτωση που είτε ο αλγόριθμος εκτελεστεί με επιτυχία, είτε πρόκειται για πρόβλημα τριών έως οκτώ μεταβλητών η όψη επιστρέφει κατάσταση επιτυχίας συμπεριλαμβάνοντας το αναγνωριστικό της εγγραφής στη βάση δεδομένων..

ajax_calculate: Πρόκειται για μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα επεξεργασίας των δεδομένων εισόδου. Το αίτημα περιλαμβάνει τα επικαιροποιημένα δεδομένα εισόδου που κατασκεύασε ο χρήστης πάνω στην εφαρμογή, καθώς και τον τίτλο που έχει ορίσει. Τα δεδομένα αυτά συλλέγονται από την όψη και από JSON μορφή μετατρέπονται σε `rython` λεξικό ώστε να μπορούν να υποστούν επεξεργασία. Όπως στην προηγούμενη όψη, έτσι κι εδώ, ανάλογα με το πλήθος των μεταβλητών εκτελείται και ο ανάλογος αλγόριθμος επίλυσης προβλημάτων Γραμμικού Προγραμματισμού. Στη συνέχεια, το αποτέλεσμα του αλγορίθμου επίλυσης και τα δεδομένα εισόδου ενημερώνονται στα αντίστοιχα πεδία της υπάρχουσας εγγραφής στη βάση δεδομένων. Εξαιρεση κι αυτή τη φορά αποτελεί η περίπτωση όπου έχουμε πρόβλημα δυο μεταβλητών και ο αλγόριθμος επίλυσης αποτύχει. Σε αυτή την περίπτωση δεν ενημερώνεται η εγγραφή στη βάση και η όψη επιστρέφει κατάσταση σφάλματος περιέχοντας το μήνυμα σφάλματος που είχε ως έξοδο ο αλγόριθμος. Σε άλλη περίπτωση τα δεδομένα αποθηκεύονται στη βάση και η

όψη επιστρέφει κατάσταση επιτυχίας συμπεριλαμβάνοντας το αναγνωριστικό της εγγραφής στη βάση δεδομένων.

ajax_upload_manage: Είναι μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα μεταφόρτωσης αρχείου εφόσον ο χρήστης έχει ορίσει τον τίτλο που επιθυμεί, έχει ανεβάσει το JSON αρχείο που επιθυμεί κι έχει επιλέξει την διαδικασία επεξεργασίας των δεδομένων εισόδου. Το αίτημα περιλαμβάνει το αρχείο JSON που επέλεξε ο χρήστης και τον τίτλο που όρισε. Τα περιεχόμενα του αρχείου JSON διαβάζονται και μετατρέπονται σε rython λεξικό. Ακολουθεί ο έλεγχος της συμβατότητας των δεδομένων εισόδου με το σχήμα JSON που έχουμε ορίσει. Ο έλεγχος αυτός γίνεται με τη χρήση του Draft7Validator από τη βιβλιοθήκη jsonschema. Αν η επικύρωση αυτή επιτύχει, δημιουργείται νέα εγγραφή στη βάση δεδομένων με τα δεδομένα του JSON αρχείου και τον τίτλο που έλαβε η όψη. Τέλος, η όψη επιστρέφει κατάσταση επιτυχίας συμπεριλαμβανομένου του αναγνωριστικού της εγγραφής στη βάση δεδομένων. Αν η επικύρωση των δεδομένων αποτύχει, η όψη επιστρέφει κατάσταση αποτυχίας.

ajax_upload_run: Πρόκειται για μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα μεταφόρτωσης αρχείου, εφόσον ο χρήστης έχει ορίσει τον τίτλο που επιθυμεί, έχει επιλέξει το JSON αρχείο που περιλαμβάνει τα δεδομένα του προβλήματος Γραμμικού Προγραμματισμούς που θέλει να επιλύσει κι έχει επιλέξει την εκτέλεση της επίλυσης. Το αίτημα και σε αυτή την περίπτωση περιλαμβάνει το αρχείο JSON του χρήστη και τον τίτλο που έθεσε. Τα περιεχόμενα του αρχείου JSON διαβάζονται και μετατρέπονται σε rython λεξικό. Ακολουθεί ο έλεγχος της συμβατότητας των δεδομένων εισόδου με το σχήμα JSON που έχουμε ορίσει. Ο έλεγχος αυτός γίνεται με τη χρήση του Draft7Validator από τη βιβλιοθήκη jsonschema. Αν η επικύρωση των δεδομένων εισόδου ολοκληρωθεί με επιτυχία εκτελείται ο αλγόριθμος επίλυσης προβλημάτων Γραμμικού Προγραμματισμού, ανάλογα με το πλήθος των μεταβλητών του προβλήματος. Αφού επιλυθεί το πρόβλημα, δημιουργείται νέα εγγραφή στη βάση δεδομένων με τα δεδομένα εισόδου, τον τίτλο και το αποτέλεσμα του αλγόριθμου επίλυσης. Τέλος, η όψη επιστρέφει κατάσταση επιτυχίας συμπεριλαμβάνοντας το αναγνωριστικό της εγγραφής που δημιουργήθηκε. Αν η επικύρωση αποτύχει, η όψη επιστρέφει κατάσταση αποτυχίας.

5.5.3 Διασύνδεση Λειτουργικότητας με τον χρήστη

Απαραίτητη προϋπόθεση για τη διασύνδεση της λειτουργικότητας με τα αιτήματα που στέλνει ο χρήστης είναι η κατασκευή των Ενιαίων Εντοπιστών Πόρων (URLs). Πριν τη κατασκευή τους απαιτείται η διασύνδεσή των URLs της εφαρμογής Γραμμικού Προγραμματισμού με τα URLs του υπόλοιπου έργου. Αυτό επιτυγχάνεται προσθέτοντας το μονοπάτι `path('linear/', include('linear.urls'))` στο αρχείο `urls.py` του συνολικού έργου. Στη συνέχεια ορίζονται τα ονόματα του κάθε URL και η διασύνδεση του κάθε ενός από αυτά με την αντίστοιχη όψη που ικανοποιεί τη

λειτουργικότητά του. Σε αυτό το σημείο πρέπει να οριστούν και τα URLs που ικανοποιούν τις AJAX κλήσεις που γίνονται από τις σελίδες στον εξυπηρετητή. Παρακάτω παρουσιάζονται και επεξηγούνται τα URLs που κατασκευάστηκαν για την εφαρμογή Γραμμικού Προγραμματισμού.

- **url(r'^\$',views.linear_index,name='linear_index')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/linear` με την όψη `linear_index` και αποτελεί την αρχική σελίδα της Εφαρμογής Γραμμικού Προγραμματισμού.
- **url(r'^upload/',views.upload_form,name='linear_upload')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/linear/upload` με την όψη `linear_upload` και αποτελεί τη σελίδα όπου ο χρήστης μπορεί να συμπληρώσει τη φόρμα με τον τίτλο του προβλήματος του και το αρχείο εισόδου.
- **url(r'^create/',views.create_data,name='create_data')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/linear/create` με την όψη `create_data` και παρέχει στον χρήστη τη σελίδα δημιουργίας των δεδομένων ενός προβλήματος Γραμμικού Προγραμματισμού.
- **path('manage/<int:method_id>',views.manage, name='manage')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/linear/manage` με την όψη `manage`. Δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας το αρχείο εισόδου είναι προς επεξεργασία.
- **path('results/<int:method_id>',views.results,name='results')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/linear/results` με την όψη `results`. Δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας τα δεδομένα είναι προς εμφάνιση.
- **url(r'^ajax/calculate',views.ajax_calculate, name='ajax_calculate')**: Συνδέει την AJAX κλήση `calculate` με την όψη `ajax_calculate`.
- **url(r'^ajax/create_and_calculate',views.ajax_create_and_calculate,name='ajax_create_and_calculate')**: Συνδέει την AJAX κλήση `create_and_calculate` με την όψη `create_and_calculate`.
- **url(r'^ajax/upload_manage',views.ajax_upload_manage,name='ajax_upload_manage')**: Συνδέει την AJAX κλήση `upload_manage` με την όψη `ajax_upload_manage`.
- **url(r'^ajax/upload_run', views.ajax_upload_run, name='ajax_upload_run')**: Συνδέει την AJAX κλήση `upload_run` με την όψη `ajax_upload_run`.

Με την ολοκλήρωση της κατασκευής των URLs η εφαρμογή του Γραμμικού Προγραμματισμού μπορεί να χρησιμοποιηθεί πλήρως από τον χρήστη και να επιτελέσει το σκοπό της, δηλαδή την επίλυση προβλημάτων Γραμμικού Προγραμματισμού και η παρουσίαση των αποτελεσμάτων είτε γραφικά είτε παρουσιάζοντας τις επαναλήψεις που χρειάστηκαν για την επίλυση του προβλήματος.

5.6 Σχεδίαση και Υλοποίηση του Προγράμματος Δυναμικού Προγραμματισμού

Η Εφαρμογή του Δυναμικού Προγραμματισμού έχει ως στόχο την παροχή ενός φιλικού προς τον χρήστη εργαλείου με το οποίο θα μπορεί να κατασκευάζει και να επιλύει προβλήματα Δυναμικού Προγραμματισμού. Λόγω του εύρους των προβλημάτων Δυναμικού Προγραμματισμού επιλέχθηκαν τρία είδη προβλημάτων τα οποία θα μπορεί ο χρήστης να επεξεργαστεί στην εφαρμογή. Αυτά είναι η Εύρεση Βέλτιστης Διαδρομής, ο Προγραμματισμός Επενδύσεων και ο Προγραμματισμός Παραγωγής και Αποθήκευσης. Αφού επιλυθεί το εκάστοτε πρόβλημα, στον χρήστη παρουσιάζεται, εκτός από τη λύση του προβλήματος, το σύνολο των επαναλήψεων που χρειάστηκαν για την επίλυσή του. Στη συνέχεια παρουσιάζονται και αναλύονται οι σχεδιαστικές επιλογές καθώς και η υλοποίηση αυτών, που έγιναν στο πλαίσιο της εφαρμογής.

5.6.1 Συλλογή, Επεξεργασία και Αποθήκευση Δεδομένων

Κατά το σχεδιασμό της Εφαρμογής λήφθηκε η απόφαση ο χρήστης να μπορεί να παρέχει τα δεδομένα του με δυο διαφορετικούς τρόπους. Ο ένα είναι μέσω ενός αρχείου JSON και ο δεύτερος κατασκευάζοντας ο ίδιος στην εφαρμογή τα δεδομένα του προβλήματος Δυναμικού Προγραμματισμού που θέλει να επιλύσει.

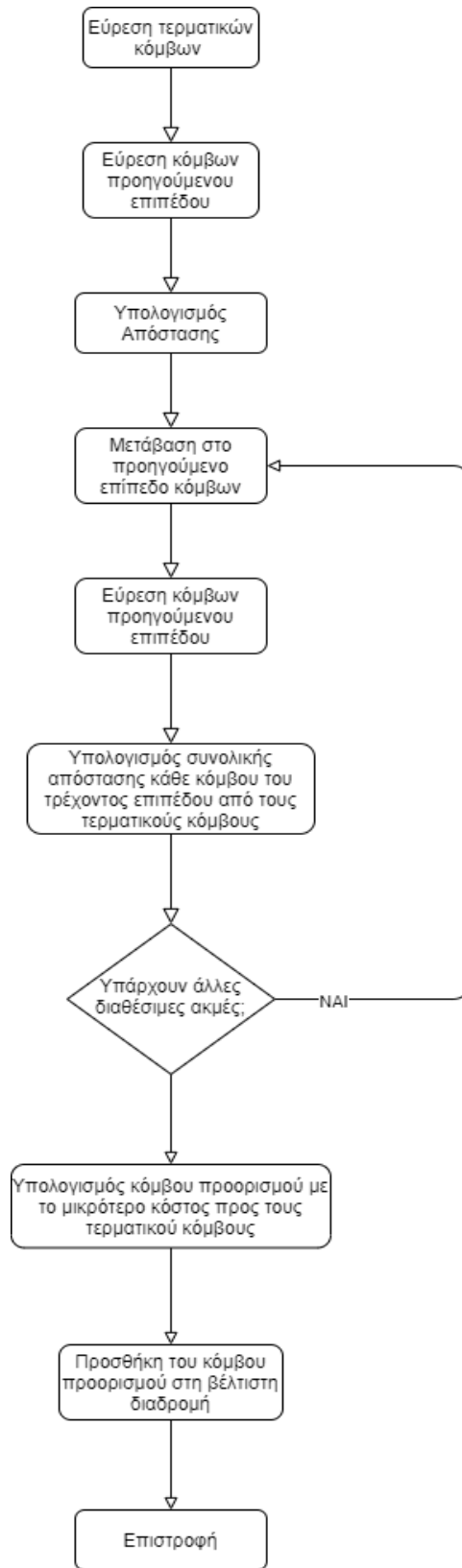
Για να καταστεί εφικτό στον χρήστη να παρέχει τα δεδομένα του μέσω ενός αρχείου JSON κατασκευάστηκε μια φόρμα στη οποία ο χρήστης πληκτρολογεί έναν τίτλο για το πρόβλημά του, στη συνέχεια επιλέγει το είδος του προβλήματος που επιθυμεί να λύσει και τέλος επιλέγει το αρχείο JSON από τον τοπικό του ηλεκτρονικό υπολογιστή. Έπειτα έπρεπε να διασφαλιστεί η εγκυρότητα της συμπλήρωσης της φόρμας και κυρίως το αρχείο που επέλεξε ο χρήστης. Η επικύρωση του αρχείου γίνεται σε δύο επίπεδα. Το ένα είναι κατά την επιλογή του αρχείου, όπου και ελέγχεται αν ο τύπος του αρχείου είναι ο επιθυμητός, δηλαδή JSON. Αν όχι, ο χρήστης ειδοποιείται να επιλέξει ένα αρχείο τύπου JSON. Το επόμενο επίπεδο είναι αφού ο χρήστης επιλέξει πως θα κινηθεί στη συνέχεια. Πριν μεταβεί στη σελίδα που επιθυμεί, ελέγχεται η εγκυρότητα των περιεχομένων του αρχείου JSON. Σε αυτό το επίπεδο δηλαδή, ελέγχεται κατά πόσο το περιεχόμενο του αρχείου JSON επιβεβαιώνει το σχήμα JSON που έχει οριστεί για τον κάθε τύπο προβλήματος. Με αυτόν τον τρόπο εξασφαλίζεται ότι ο αλγόριθμος επίλυσης θα βρει τα απαιτούμενα δεδομένα ώστε να λύσει το πρόβλημα Δυναμικού Προγραμματισμού.

Στην περίπτωση που ο χρήστης επιλέξει να κατασκευάσει τα δεδομένα του χρησιμοποιώντας την εφαρμογή, ο έλεγχος της εγκυρότητας των δεδομένων παραλείπεται διότι εξασφαλίζεται από την ίδια την εφαρμογή ότι περιέχονται οι

απαραίτητες πληροφορίες για να προχωρήσει η επίλυση του προβλήματος Δυναμικού Προγραμματισμού.

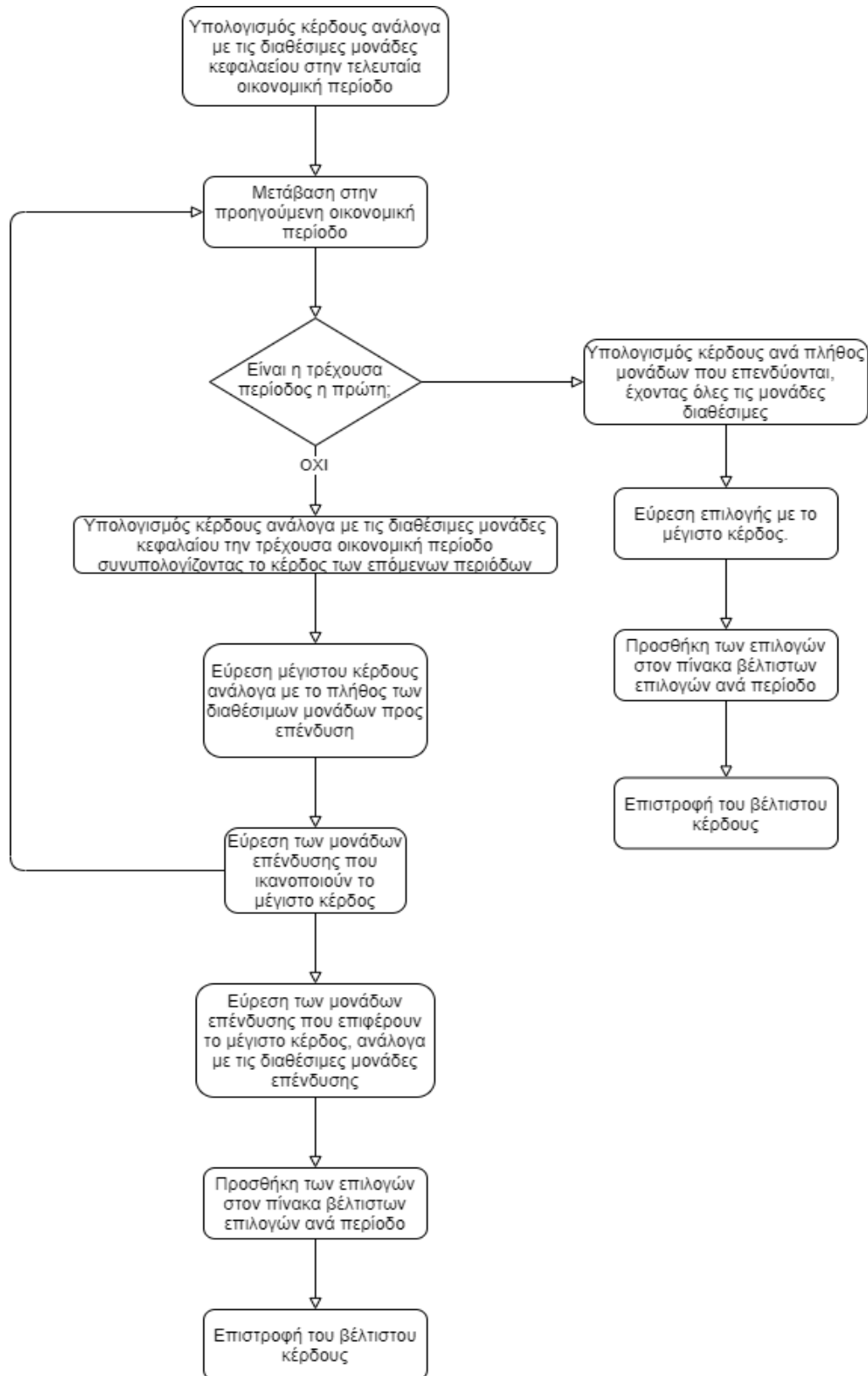
Εφόσον, λοιπόν, η μορφή των δεδομένων είναι η επιθυμητή αυτά συλλέγονται και μετατρέπονται σε rython λεξικό ώστε να προχωρήσει η επίλυση του προβλήματος. Ανάλογα με το είδος του προβλήματος, που ορίζεται από τον χρήστη, εκτελείται και διαφορετικός αλγόριθμος. Στη συνέχεια παρουσιάζονται οι αλγόριθμοι για την κάθε περίπτωση.

Εύρεση Βέλτιστης Διαδρομής: Τα δεδομένα αποτελούνται από δύο λίστες. Η μια λίστα περιέχει του κόμβους της διαδρομής, με ορισμένους τους κόμβους εκκίνησης και τους κόμβους τερματισμού, ενώ η άλλη περιέχει τις ακμές με την πληροφορία του κόστους μετάβασης από τον κόμβο αναχώρησης στον κόμβος προορισμού. Οι δύο αυτές λίστες προωθούνται σαν είσοδος στον αλγόριθμο Εύρεσης Βέλτιστης Διαδρομής. Στην *Εικόνα 11* φαίνεται το διάγραμμα ροής του αλγορίθμου. Ο αλγόριθμος βρίσκει από τη λίστα κόμβων τους τερματικούς κόμβους και θεωρεί ότι βρίσκονται στο ίδιο επίπεδο. Έχοντας ως κόμβους του τρέχοντος επιπέδου τους τερματικούς κόμβους εκκινεί μια αναδρομική διαδικασία, όπου χρησιμοποιώντας τη λίστα των ακμών, για καθένα από αυτούς τους κόμβους βρίσκει τους κόμβους που βρίσκονται σε μικρότερο επίπεδο, και τους χρίζει αυτούς κόμβους του τρέχοντος επιπέδου. Κατά την εύρεση των κόμβων μικρότερου επιπέδου ταυτόχρονα υπολογίζεται και το συνολικό κόστος που απαιτήθηκε για τη μετάβαση από τον τερματικό κόμβο στον τρέχοντα κόμβο που ελέγχεται. Η κάθε ακμή που χρησιμοποιήθηκε αφαιρείται από τη λίστα των ακμών. Η αναδρομική διαδικασία προχωράει τροφοδοτούμενη από τους κόμβους του νέου επιπέδου και τη λίστα των εναπομείναντων ακμών. Η αναδρομική σχέση κάθε επιπέδου επιστρέφει. Τέλος υπολογίζεται για κάθε επίπεδο αντίστροφα τώρα, κατά την επιστροφή η μέχρι τώρα βέλτιστη διαδρομή από τους κόμβους εκκίνησης προς τους κόμβους τερματισμού. Σε κάθε επίπεδο οι κόμβοι αναχώρησης, οι κόμβοι άφιξης, καθώς και το κόστος μετάβασης από τους τερματικούς κόμβους στους κόμβους αναχώρησης του επιπέδου διατηρούνται σε ένα rython λεξικό ώστε να παρουσιαστούν στη συνέχεια στον χρήστη. Όταν ολοκληρωθεί η διαδικασία έχουμε τις βέλτιστες δυνατές διαδρομές, τη διαδρομή των κόμβων που ακολουθούν και το ελάχιστο κόστος των βέλτιστων διαδρομών.



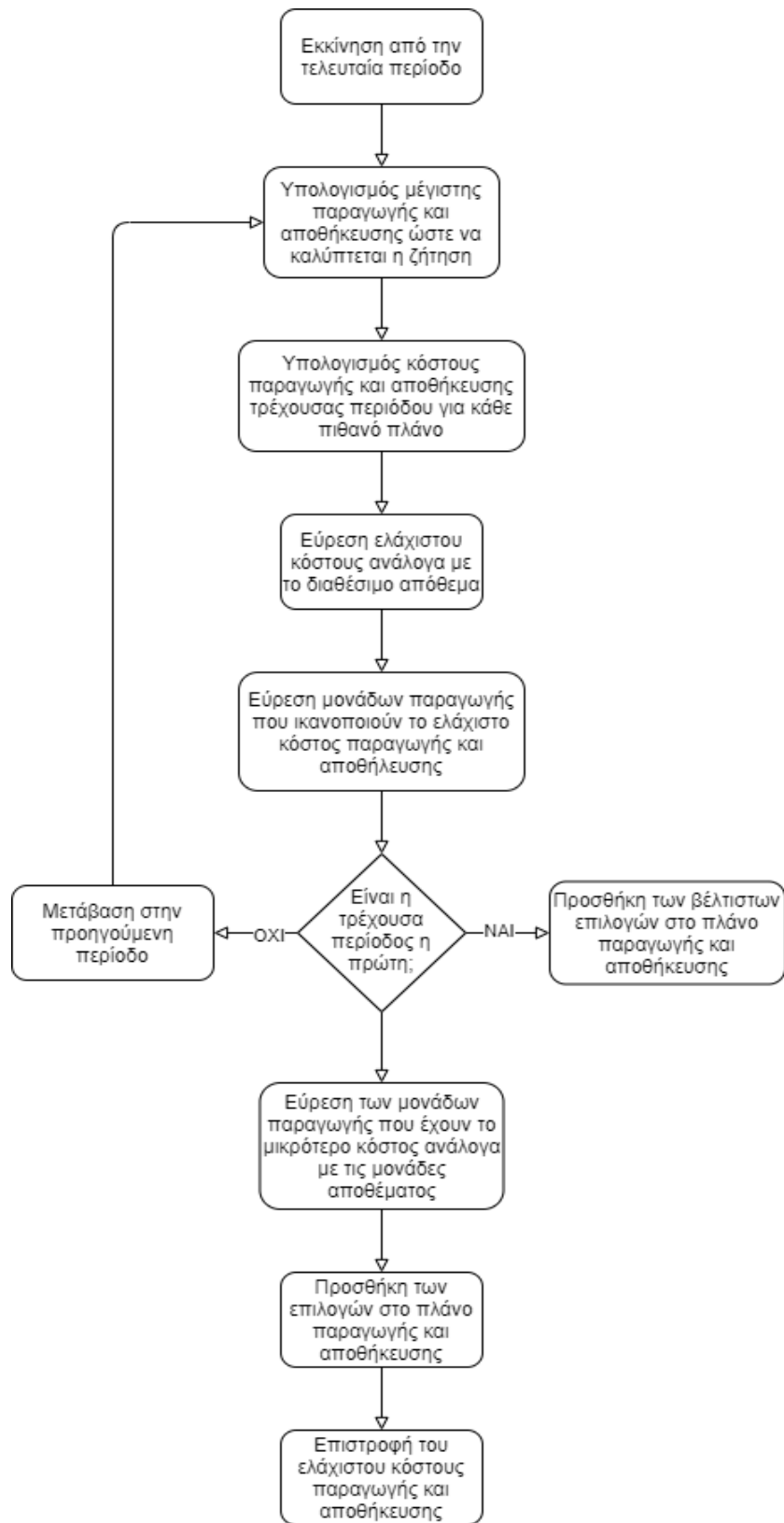
Εικόνα 11- Αλγόριθμος εύρεσης βέλτιστης διαδρομής

Προγραμματισμός Επενδύσεων: Τα δεδομένα εισόδου αποτελούνται από έναν πίνακα που περιλαμβάνει τα κέρδη που επιφέρει κάθε επενδυτική επιλογή την κάθε επενδυτική περίοδο. Το διάγραμμα ροής του αλγόριθμου αυτού φαίνεται στην *Εικόνα 12*. Ο πίνακας αυτός προωθείται στον αλγόριθμο εύρεσης βέλτιστου πλάνου επενδύσεων. Ο αλγόριθμος εκκινεί μια αναδρομική διαδικασία ξεκινώντας από τα κέρδη που επιφέρει η κάθε μια επενδυτική επιλογή την τελευταία οικονομική περίοδο. Για κάθε οικονομική περίοδο υπολογίζεται το συνολικό κέρδος που έχει επιφέρει η κάθε επιλογή την τρέχουσα οικονομική περίοδο. Δηλαδή από την τελευταία οικονομική περίοδο μέχρι και την τρέχουσα. Αυτό επιτυγχάνεται υπολογίζοντας τις εναπομείναντες οικονομικές μονάδες οι οποίες μπορούν να χρησιμοποιηθούν. Συνεπώς όταν η αναδρομή φτάσει στην πρώτη οικονομική περίοδο έχει υπολογιστεί το μέγιστο κέρδος για κάθε επενδυτική επιλογή. Έτσι όταν κάθε αναδρομή επιστρέψει υπολογίζεται ποια επενδυτική επιλογή επιφέρει το μεγαλύτερο κέρδος, ανάλογα με τις διαθέσιμες οικονομικές μονάδες προς επένδυση. Όταν όλες οι αναδρομές επιστρέψουν, έχει το συνολικό πλάνο που επιφέρει το μέγιστο κέρδος. Δηλαδή τις οικονομικές μονάδες που πρέπει να επενδυθούν σε κάθε οικονομική περίοδο ώστε να επιτευχθεί το μέγιστο κέρδος.



Εικόνα 12- Αλγόριθμος επίλυσης προβλημάτων προγραμματισμού επενδύσεων

Προγραμματισμός Παραγωγής και Αποθήκευσης: Τα δεδομένα εισόδου περιέχουν τους συντελεστές της συνάρτησης κόστους παραγωγής, τους συντελεστές της συνάρτησης κόστους αποθήκευσης και τους πίνακες ζήτησης, δυνατότητας παραγωγής και δυνατότητας αποθήκευσης ανά περίοδο. Το διάγραμμα ροής του αλγορίθμου φαίνεται στην *Εικόνα 13*. Τα παραπάνω δεδομένα περνάνε ως είσοδο στον αλγόριθμο εύρεσης βέλτιστου πλάνου παραγωγής και αποθήκευσης. Ο αλγόριθμος εκκινεί μια αναδρομική διαδικασία. Έχοντας ως αρχή την τελευταία περίοδο και ως γνώμονα σε κάθε περίοδο να καλύπτεται από το άθροισμα παραγωγής και αποθέματος η ζήτηση, χωρίς να παραβιάζεται το όριο παραγωγής και το όριο αποθήκευσης, υπολογίζεται το κόστος κάθε πλάνου παραγωγής και αποθήκευσης ανά περίπτωση. Σε κάθε αναδρομή πέρα από το κόστος της τρέχουσας περιόδου συνυπολογίζεται το κόστος της εκάστοτε επιλογής από την τελευταία περίοδο μέχρι και την τρέχουσα. Συνεπώς, όταν ο αλγόριθμος φτάσει στην πρώτη περίοδο έχει υπολογιστεί το κόστος της κάθε επιλογής της πρώτης περιόδου, συνυπολογίζοντας όλες τις επόμενες περιόδους. Έτσι επιλέγεται η στρατηγική με το μικρότερο κόστος και επιστρέφει. Το ίδιο πλάνο ακολουθείτε μέχρι να επιστρέψει και η τελευταία αναδρομική διαδικασία. Όταν όλες οι αναδρομές έχουν επιστρέψει, θα έχει υπολογιστεί η στρατηγική που έχει το μικρότερο κόστος παραγωγής και αποθήκευσης. Δηλαδή το ποσό το μονάδων που πρέπει να παραχθούν και να αποθηκευτούν ανά οικονομική περίοδο ώστε να καλύπτεται πλήρως η ζήτηση. Τέλος, αξίζει να σημειωθεί ότι οι πίνακες τους κόστους της κάθε περιόδου αποθηκεύονται ώστε να παρουσιαστεί στον χρήστη η όλη η διαδικασία επίλυσης του προβλήματος.



Εικόνα 13- Αλγόριθμος επίλυσης προβλημάτων προγραμματισμού παραγωγής και αποθήκευσης

Σε κάθε περίπτωση, όταν ολοκληρωθεί ο αντίστοιχος αλγόριθμος, τα δεδομένα που παράγονται αποθηκεύονται ώστε να παρουσιασθή στη συνέχεια η πλήρης λύση στον χρήστη. Δηλαδή όλη η δυναμική διαδικασία επίλυσης του προβλήματος.

5.6.2 Λειτουργικότητα της Εφαρμογής

Με την ολοκλήρωση του σχεδιασμού της συλλογής, επεξεργασίας και αποθήκευσης των δεδομένων, επόμενο βήμα είναι η προσθήκη λειτουργικότητας στην εφαρμογή. Αυτό επιτυγχάνεται με την κατασκευή των όψεων ώστε να διασυνδεθούν οι διαδικασίες που εκτελούνται στον εξυπηρετητή με τις ιστοσελίδες όπου θα έχει πρόσβαση ο χρήστης. Στη συνέχεια παρουσιάζονται και αναλύονται οι όψεις που κατασκευάστηκαν στο πλαίσιο της εφαρμογής, ώστε να καταστεί η διασύνδεση αυτή εφικτή.

dynamic_index: Αυτή η όψη δέχεται μόνο GET αίτημα από τον χρήστη και φορτώνει την αρχική σελίδα της Εφαρμογής Δυναμικού Προγραμματισμού, όπου ο χρήστης επιλέγει τον τρόπο με τον οποίο προτιμά να παρέχει τα δεδομένα του σε αυτή.

upload_form: Σε GET αίτημα η όψη επιστρέφει τη σελίδα με τη φόρμα συμπλήρωσης των δεδομένων εισόδου. Τα POST αιτήματα που μπορεί να δεχθεί η όψη είναι δύο ειδών. Αν στο αίτημα περιλαμβάνεται το λεκτικό “run”, η όψη επιστρέφει GET αίτημα στην όψη results με παράμετρο το αναγνωριστικό της εγγραφής της βάσης δεδομένων που περιλαμβάνει τα δεδομένα που είναι προς παρουσίαση. Αν στο αίτημα περιλαμβάνεται το λεκτικό “manage”, η όψη επιστρέφει GET αίτημα στην όψη manage με παράμετρο το αναγνωριστικό της εγγραφής της βάσης δεδομένων που περιέχει τα δεδομένα εισόδου που είναι προς επεξεργασία.

results: Η συγκεκριμένη όψη δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής της βάσης δεδομένων που είναι προς παρουσίαση. Σε GET αίτημα η όψη ανταποκρίνεται ανάλογα με την τιμή της ετικέτας “type” του πεδίου input_file της εγγραφής της βάσης δεδομένων της οποίας το αναγνωριστικό λαμβάνεται ως παράμετρος στην όψη. Αν η τιμή είναι “route”, η όψη επιστρέφει τη σελίδα παρουσίασης αποτελεσμάτων των προβλημάτων εύρεσης βέλτιστης διαδρομής. Αν η τιμή είναι “invest”, η όψη επιστρέφει τη σελίδα παρουσίασης αποτελεσμάτων των προβλημάτων εύρεσης βέλτιστου πλάνου προγραμματισμού επενδύσεων. Αν η τιμή είναι “store”, η όψη επιστρέφει τη σελίδα παρουσίασης αποτελεσμάτων των προβλημάτων εύρεσης βέλτιστου πλάνου προγραμματισμού παραγωγής και αποθήκευσης. Σε POST αίτημα η όψη ανταποκρίνεται ανάλογα με το λεκτικό που θα περιλαμβάνει το POST αίτημα. Αν περιλαμβάνει το λεκτικό “new”, η όψη αποστέλλει GET αίτημα στην όψη dynamic_index για την επίλυση νέου προβλήματος. Αν περιλαμβάνει το λεκτικό “inspect”, η όψη αποστέλλει GET αίτημα στην όψη manage,

η οποία αναλύεται στη συνέχεια, έχοντας ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας τα δεδομένα είναι προς επεξεργασία.

manage: Η εν λόγω όψη δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας τα δεδομένα εισόδου είναι προς επεξεργασία. Σε GET αίτημα η όψη επιστρέφει τη σελίδα επεξεργασίας των δεδομένων εισόδου για προβλήματα Δυναμικού Προγραμματισμού. Σε POST αίτημα η όψη αποστέλλει GET αίτημα στην όψη results, έχοντας ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας τα δεδομένα υπέστησαν επεξεργασία από τον χρήστη.

create_data: Σε GET αίτημα η όψη φορτώνει στον χρήστη τη σελίδα με το εργαλείο κατασκευής δεδομένων για ένα πρόβλημα Δυναμικού Προγραμματισμού. Σε POST αίτημα η όψη στέλνει GET αίτημα στην όψη results, με παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων όπου κατασκευάστηκε μόλις από τον χρήστη και είναι τα αποτελέσματά της προς εμφάνιση.

ajax_create_and_calculate: Είναι μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα δημιουργίας δεδομένων ενός νέου προβλήματος Δυναμικού Προγραμματισμού. Το αίτημα που δέχεται περιλαμβάνει τα δεδομένα εισόδου που δημιούργησε ο χρήστης και ο τίτλος που όρισε. Τα δεδομένα είναι σε μορφή JSON. Η όψη τα μετατρέπει σε python λεξικό και ανάλογα με την τιμή του πεδίου “type”, του λεξικού αυτού, περνάει τα δεδομένα αυτά ως είσοδο στον αντίστοιχο αλγόριθμο επίλυσης. Για τις τιμές “route”, “invest” και “store” χρησιμοποιούνται αντίστοιχα οι αλγόριθμοι εύρεσης βέλτιστης διαδρομής, εύρεσης βέλτιστου πλάνου προγραμματισμού επενδύσεων και εύρεσης βέλτιστου πλάνου προγραμματισμού παραγωγής και αποθήκευσης. Έπειτα δημιουργείται εγγραφή στη βάση δεδομένων χρησιμοποιώντας τα δεδομένα εισόδου, το αποτέλεσμα του εκάστοτε αλγορίθμου και τον τίτλο που ορίστηκε από τον χρήστη. Τέλος, η όψη επιστρέφει κατάσταση επιτυχίας έχοντας ως παράμετρος το αναγνωριστικό της εγγραφής.

ajax_calculate: Πρόκειται για μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα επεξεργασία δεδομένων εισόδου ενός προβλήματος Δυναμικού Προγραμματισμού. Το αίτημα περιλαμβάνει το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας τα δεδομένα εισόδου δέχτηκαν επεξεργασία, τα ανανεωμένα δεδομένα εισόδου, όπως τα επεξεργάστηκε ο χρήστης και ο τίτλος που όρισε ο χρήστης. Η όψη δέχεται τα δεδομένα σε μορφή JSON, τα μετατρέπει σε python λεξικό και ανάλογα με την τιμή του πεδίου “type” των δεδομένων εισόδου εκτελείται ο ανάλογος αλγόριθμος όπως ακριβώς στην όψη ajax_create_and_calculate. Στη συνέχεια ενημερώνονται τα πεδία title, input_file και output_file στην εγγραφή της οποίας το αναγνωριστικό λήφθηκε ως είσοδος. Τέλος, η όψη επιστρέφει κατάσταση επιτυχίας.

ajax_upload_manage: Είναι μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα που περιέχει τη σελίδα μεταφόρτωσης αρχείου, εφόσον έχει επιλεγεί από τον χρήστη

η διαδικασία επεξεργασίας των δεδομένων του αρχείου εισόδου. Το αίτημα περιλαμβάνει το αρχείο JSON που επέλεξε ο χρήστης και τον τίτλο που όρισε. Τα περιεχόμενα του αρχείου JSON διαβάζονται και μετατρέπονται σε python λεξικό. Ακολουθεί ο έλεγχος της συμβατότητας των δεδομένων εισόδου με το σχήμα JSON που έχουμε ορίσει. Ο έλεγχος αυτός γίνεται με τη χρήση του Draft7Validator από τη βιβλιοθήκη jsonschema. Αν η επικύρωση αυτή επιτύχει, δημιουργείται νέα εγγραφή στη βάση δεδομένων με τα δεδομένα του JSON αρχείου και τον τίτλο που έλαβε η όψη. Τέλος, η όψη επιστρέφει κατάσταση επιτυχίας συμπεριλαμβανομένου του αναγνωριστικού της εγγραφής στη βάση δεδομένων. Αν η επικύρωση των δεδομένων αποτύχει, η όψη επιστρέφει κατάσταση αποτυχίας.

ajax_upload_run: Πρόκειται για μια όψη που ικανοποιεί μια AJAX κλήση από τη σελίδα μεταφόρτωσης αρχείου, εφόσον ο χρήστης έχει ορίσει τον τίτλο που επιθυμεί, έχει επιλέξει το JSON αρχείο που περιλαμβάνει τα δεδομένα του προβλήματος Γραμμικού Προγραμματισμούς που θέλει να επιλύσει κι έχει επιλέξει την εκτέλεση της επίλυσης. Το αίτημα και σε αυτή την περίπτωση περιλαμβάνει το αρχείο JSON του χρήστη και τον τίτλο που έθεσε. Τα περιεχόμενα του αρχείου JSON διαβάζονται και μετατρέπονται σε python λεξικό. Ακολουθεί ο έλεγχος της συμβατότητας των δεδομένων εισόδου με το σχήμα JSON που έχουμε ορίσει. Ο έλεγχος αυτός γίνεται με τη χρήση του Draft7Validator από τη βιβλιοθήκη jsonschema. Αν η επικύρωση των δεδομένων εισόδου ολοκληρωθεί με επιτυχία εκτελείται ο αλγόριθμος επίλυσης προβλημάτων Γραμμικού Προγραμματισμού, ανάλογα με το πλήθος των μεταβλητών του προβλήματος. Αφού επιλυθεί το πρόβλημα, δημιουργείται νέα εγγραφή στη βάση δεδομένων με τα δεδομένα εισόδου, τον τίτλο και το αποτέλεσμα του αλγόριθμου επίλυσης. Τέλος, η όψη επιστρέφει κατάσταση επιτυχίας συμπεριλαμβάνοντας το αναγνωριστικό της εγγραφής που δημιουργήθηκε. Αν η επικύρωση αποτύχει, η όψη επιστρέφει κατάσταση αποτυχίας.

5.6.3 Διασύνδεση Λειτουργικότητας με τον χρήστη

Για τη διασύνδεση της λειτουργικότητας της Εφαρμογής Δυναμικού Προγραμματισμού με τα αιτήματα που δημιουργούνται από τον χρήστη είναι απαραίτητη η δήλωση των Ενιαίων Εντοπιστών Πόρων (URLs). Ωστόσο για να αναγνωριστούν τα URLs της εφαρμογής από το συνολικό έργο θα πρέπει πρώτα να προστεθεί το μονοπάτι `path('dynamic/', include(dynamic.urls'))` στο αρχείο `urls.py` του συνολικού έργου. Παρακάτω παρουσιάζονται τα μονοπάτια που δημιουργήθηκαν για την Εφαρμογή και η διασύνδεση του καθενός από αυτά με την όψη που αντιστοιχεί τη λειτουργικότητα της κάθε σελίδας, αλλά και τα μονοπάτια που κατασκευάστηκαν για την ικανοποίηση των AJAX κλήσεων που κάνουν οι σελίδες στον εξυπηρετητή.

- **url(r'^\$', views.dynamic_index, name='dynamic_index')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/dynamic` με την όψη `dynamic_index` και αποτελεί την αρχική σελίδα της Εφαρμογής Δυναμικού Προγραμματισμού.
- **url(r'^upload/', views.upload_form, name='dynamic_upload')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/dynamic/upload` με την όψη `dynamic_upload` και αποτελεί τη σελίδα όπου ο χρήστης μπορεί να συμπληρώσει τη φόρμα με τον τίτλο του προβλήματος του, τον τύπο του προβλήματος και το αρχείο εισόδου.
- **url(r'^create/', views.create_data, name='create_data')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/dynamic/create` με την όψη `create_data` και παρέχει στον χρήστη τη σελίδα δημιουργίας των δεδομένων ενός προβλήματος Δυναμικού Προγραμματισμού.
- **path('results/<int:method_id>', views.results, name='results')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/dynamic/results` με την όψη `results`. Δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας τα δεδομένα είναι προς εμφάνιση.
- **path('manage/<int:method_id>', views.manage, name='manage')**: Συνδέει το μονοπάτι `http://127.0.0.1:8000/dynamic/manage` με την όψη `manage`. Δέχεται ως παράμετρο το αναγνωριστικό της εγγραφής στη βάση δεδομένων της οποίας το αρχείο εισόδου είναι προς επεξεργασία.
- **url(r'^ajax/calculate', views.ajax_calculate, name='ajax_calculate')**: Συνδέει την AJAX κλήση `calculate` με την όψη `ajax_calculate`.
- **url(r'^ajax/create_and_calculate', views.ajax_create_and_calculate, name='ajax_create_and_calculate')**: Συνδέει την AJAX κλήση `create_and_calculate` με την όψη `create_and_calculate`.
- **url(r'^ajax/upload_manage', views.ajax_upload_manage, name='ajax_upload_manage')**: Συνδέει την AJAX κλήση `upload_manage` με την όψη `ajax_upload_manage`.
- **url(r'^ajax/upload_run', views.ajax_upload_run, name='ajax_upload_run')**: Συνδέει την AJAX κλήση `upload_run` με την όψη `ajax_upload_run`.

Με την ολοκλήρωση της κατασκευής των URLs η εφαρμογή του Δυναμικού Προγραμματισμού μπορεί να χρησιμοποιηθεί πλήρως από τον χρήστη και να επιτελέσει το σκοπό της, δηλαδή την επίλυση τριών ειδών προβλημάτων Δυναμικού Προγραμματισμού και η παρουσίαση των αποτελεσμάτων παρουσιάζοντας τις επαναλήψεις που χρειάστηκαν για την επίλυση του εκάστοτε προβλήματος.

6 Παρουσίαση Εφαρμογής

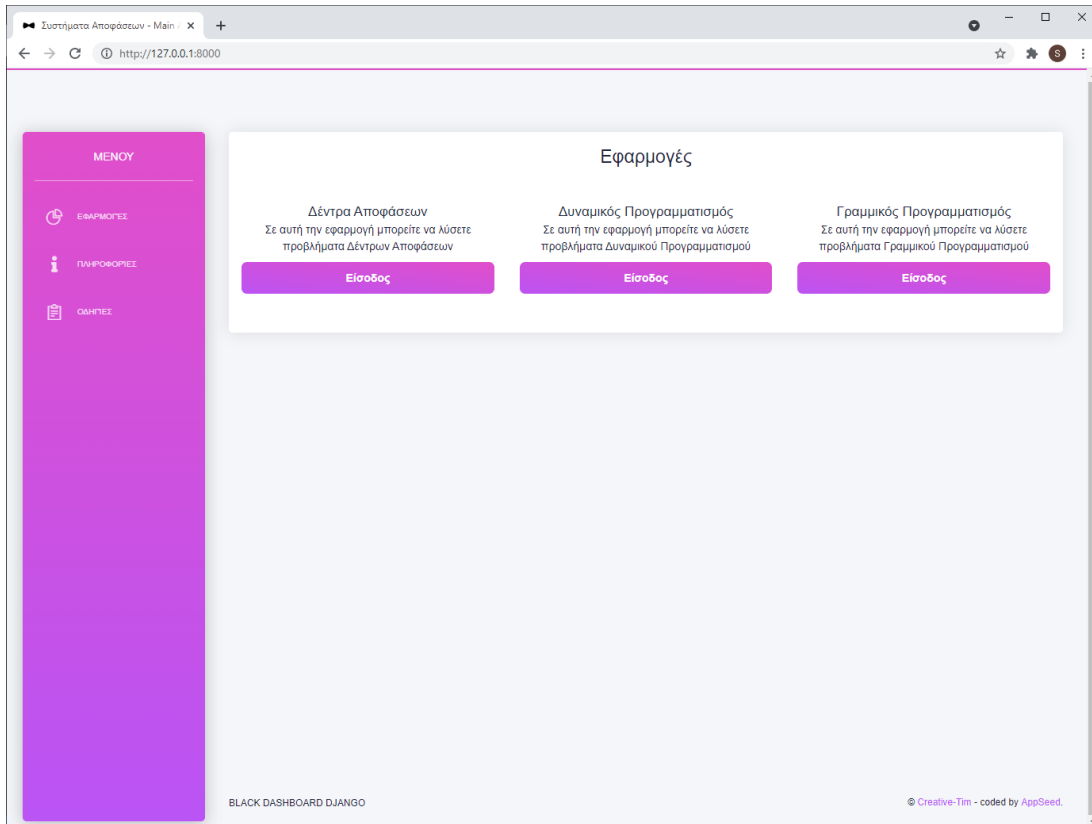
Στο κεφάλαιο αυτό θα γίνει αναλυτική περιγραφή της λειτουργικότητας των εφαρμογών που παρουσιάστηκαν παραπάνω και θα δοθούν σενάρια χρήσης για κάθε οθόνη που περιλαμβάνεται σε κάθε μια από αυτές. Συγκεκριμένα, θα περιγραφεί με λεπτομέρεια η διαδικασία χρήσης των εφαρμογών Δέντρων Αποφάσεων, Γραμμικού Προγραμματισμού και Δυναμικού Προγραμματισμού είτε με δεδομένα εισόδου από αρχείο είτε με την κατασκευή αυτών πάνω στην εκάστοτε εφαρμογή. Επιπλέον θα παρουσιαστούν και όλες οι υπόλοιπες λειτουργικότητες που συμβάλλουν σε μια ολοκληρωμένη εμπειρία από την πλευρά του χρήστη. Αρχικά, θα περιγράφονται οι οθόνες και το περιεχόμενό τους, θα εξηγείται η λειτουργία τους και οι λόγοι επιλογών του προγραμματιστή και θα αναλύεται σε βήματα κάθε κίνηση την οποία ο χρήστης θα μπορεί να εκτελέσει προκειμένου να επιτύχει το αποτέλεσμα που επιθυμεί.

6.1 Βασική Εφαρμογή

6.1.1 Αρχική Οθόνη

Η παρούσα οθόνη αποτελεί την αρχική σελίδα της συνολικής εφαρμογής και κατά συνέπεια αυτή που αντικρίζει ο χρήστης κατά την είσοδό του σε αυτή. Όπως φαίνεται στη *Εικόνα 14*, που είναι το στιγμιότυπο της παραπάνω οθόνης, αριστερά υπάρχει μια στήλη η οποία παραμένει διαθέσιμη στον χρήστη καθ' όλη την περιήγηση του στην εφαρμογή. Αυτή η στήλη αποτελεί και τον κατάλογο πλοήγησης της εφαρμογής. Μέσω αυτής γίνονται προσβάσιμες τρεις οθόνες. Η οθόνη εφαρμογών που περιλαμβάνονται οι εφαρμογές που μπορεί να χρησιμοποιηθούν για επίλυση προβλημάτων, η οθόνη πληροφοριών, όπου παρέχονται στον χρήστη πληροφορίες της εφαρμογής, και η οθόνη οδηγιών όπου ο χρήστης θα βρει χρήσιμες οδηγίες.

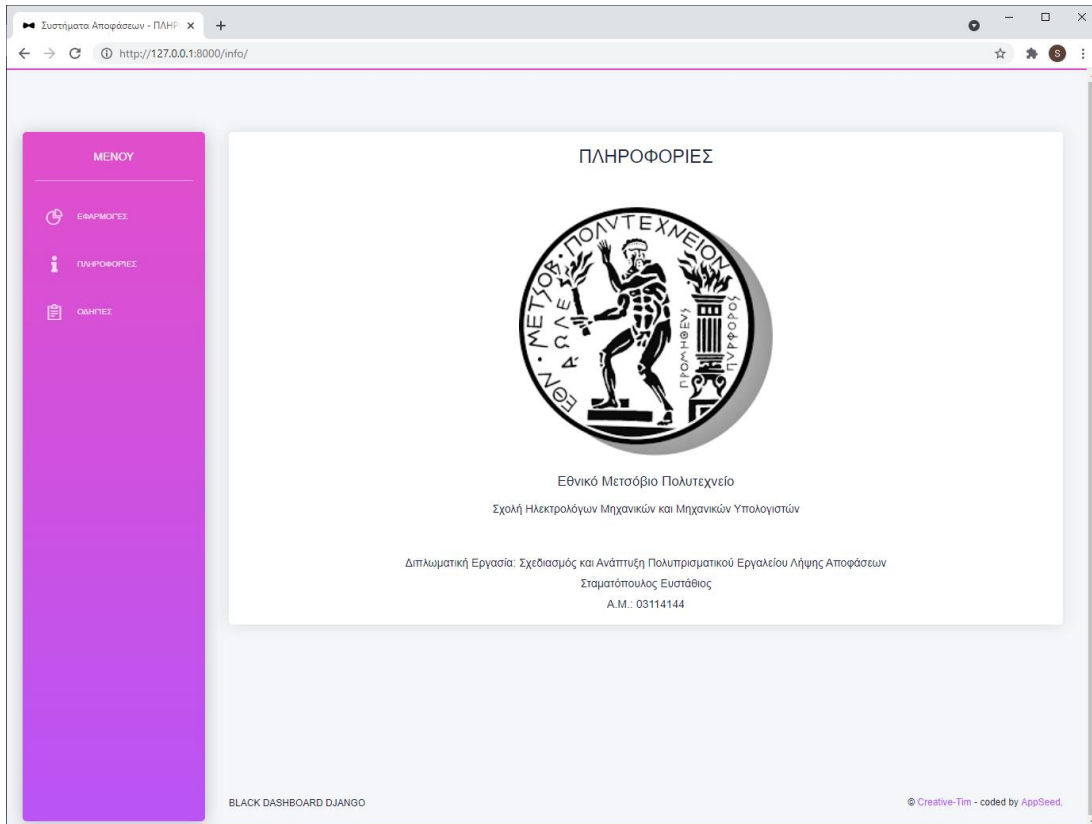
Η αρχική οθόνη αποτελεί ταυτόχρονα και την οθόνη εφαρμογών που αναφέρθηκε παραπάνω και ο χρήστης μπορεί άμεσα να προβεί στη χρήση όποιας εφαρμογής επιθυμεί μέσα σε αυτή. Σε αυτή τη σελίδα, λοιπόν, γίνεται μια συνοπτική παρουσίαση των προς χρήση εφαρμογών η οποία συνοδεύεται από ένα κουμπί Είσοδος που ανακατευθύνει τον χρήστη στην αρχική σελίδα της αντίστοιχης εφαρμογής. Μέσω αυτής της σελίδα είναι εφικτή η πρόσβαση σε τρεις εφαρμογές. Την εφαρμογή δέντρων αποφάσεων, την εφαρμογή γραμμικού προγραμματισμού και την εφαρμογή δυναμικού προγραμματισμού, όπου ο χρήστης μπορεί να λύσει τα αντίστοιχα προβλήματα συστημάτων αποφάσεων.



Εικόνα 14- Αρχική οθόνη συνολικής εφαρμογής

6.1.2 Οθόνη Πληροφοριών

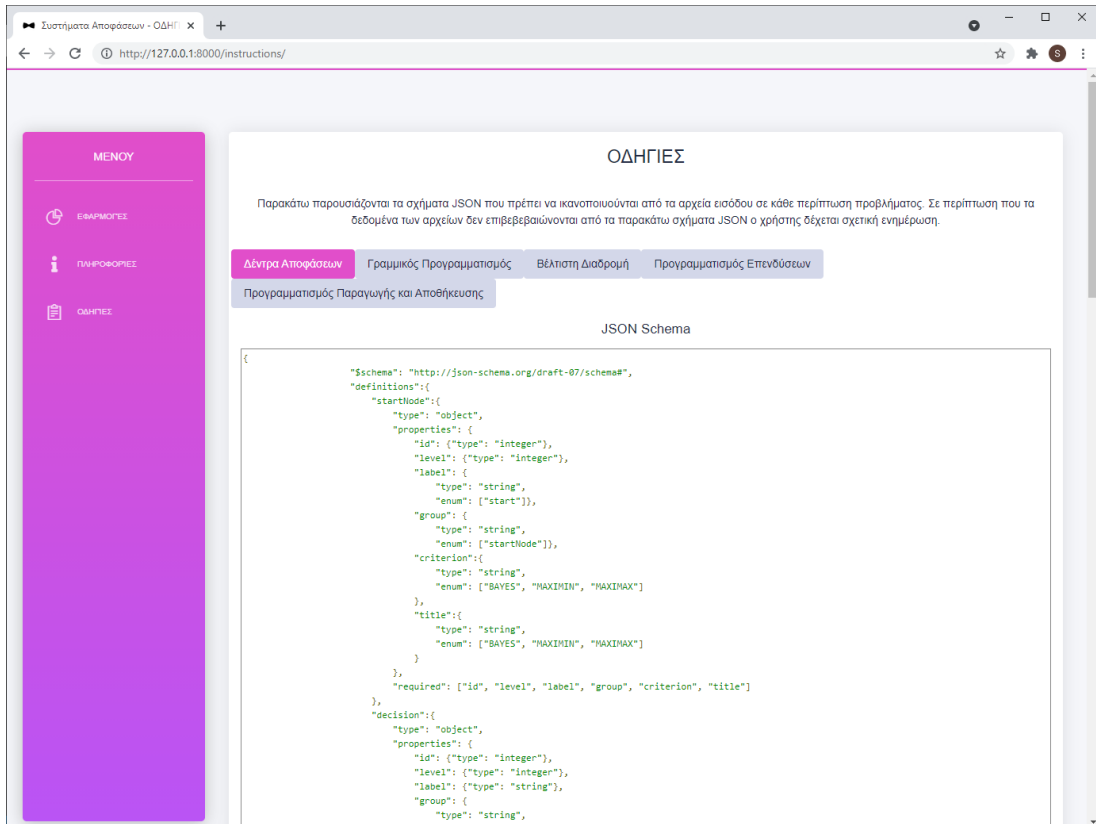
Σε αυτή τη σελίδα περιλαμβάνονται πληροφορίες σχετικά με την εφαρμογή και την ομάδα υλοποίησης, αλλά και το πανεπιστημιακό ίδρυμα υπό την επίβλεψη του οποίου κατασκευάστηκε η παρούσα εφαρμογή. Η πρόσβαση σε αυτή γίνεται μόνο μέσω της στήλης πλοήγησης που βρίσκεται στο αριστερό μέρος της διαδικτυακής εφαρμογής. Στην *Εικόνα 15* φαίνεται το στιγμιότυπο της οθόνης πληροφοριών.



Εικόνα 15- Οθόνη πληροφοριών

6.1.3 Οθόνη Οδηγιών

Στην οθόνη αυτή ο χρήστης θα βρει οδηγίες για τα αρχεία εισόδου και πώς πρέπει να διαμορφώνεται το περιεχόμενο αυτών. Η πρόσβαση σε αυτή γίνεται μόνο μέσω της στήλης πλοήγησης που βρίσκεται στο αριστερό μέρος της διαδικτυακής εφαρμογής. Συγκεκριμένα, εδώ ο χρήστης θα βρει τα σχήματα JSON τα οποία θα πρέπει να επιβεβαιώνουν τα αρχεία εισόδου. Για αυτό το σκοπό έχει τοποθετηθεί μια μπάρα πλοήγησης, όπως φαίνεται στην *Εικόνα 16*, με την οποία, ανάλογα με την επιλογή του τύπου προβλήματος προς επίλυση, στο παρακάτω πλαίσιο παρουσιάζεται και το αντίστοιχο σχήμα JSON.



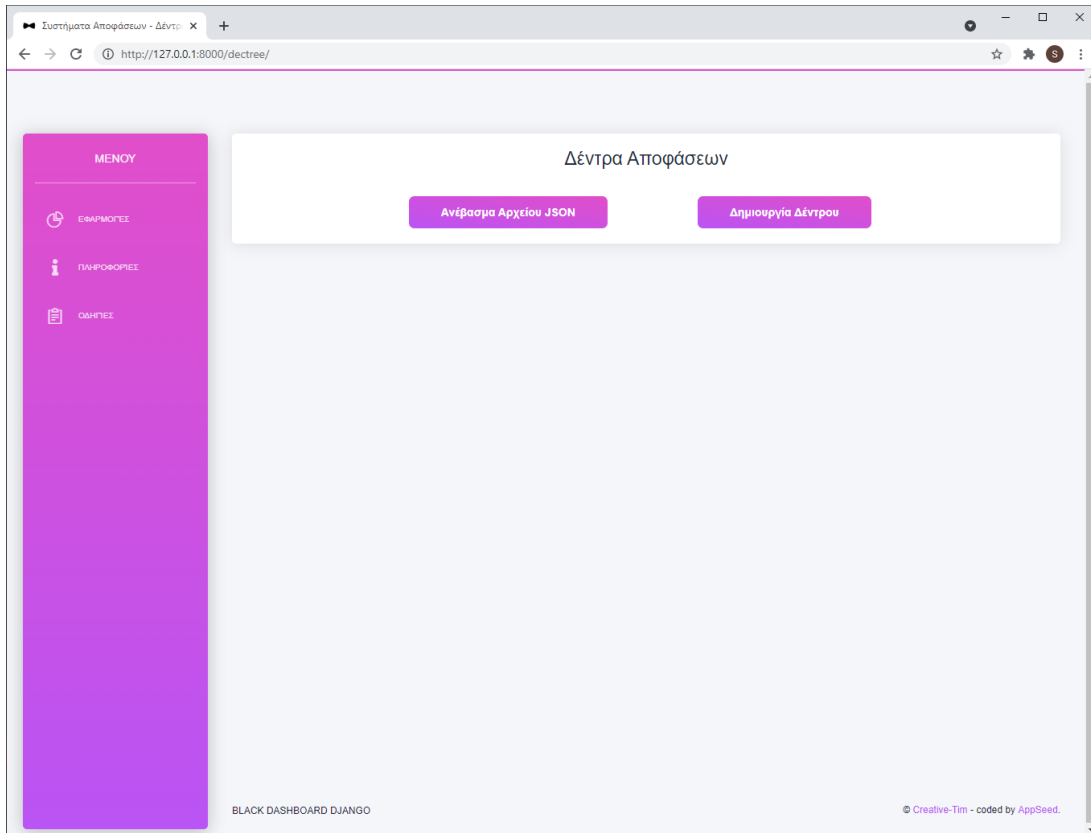
Εικόνα 16- Οθόνη οδηγιών

6.2 Εφαρμογή Δέντρων Αποφάσεων

Σε αυτό το σημείο θα περιγραφούν τα σενάρια χρήσης της εφαρμογής Δέντρων Αποφάσεων. Δηλαδή θα παρουσιαστούν οι δυνατότητες που παρέχει η εφαρμογή στον χρήστη και πώς αυτή ανταποκρίνεται σε κάθε κίνηση του χρήστη μέσα σε αυτή.

6.2.1 Αρχική Οθόνη

Εφόσον ο χρήστης έχει επιλέξει ότι θέλει να κάνει χρήση της εφαρμογής δέντρων αποφάσεων ανακατευθύνεται, στην αρχική οθόνη της εφαρμογής δέντρων αποφάσεων. Μια άλλη περίπτωση με την οποία ο χρήστης μπορεί να βρεθεί σε αυτή τη οθόνη είναι από την οθόνη αποτελεσμάτων σε περίπτωση που επιθυμεί να λύσει ένα νέο δέντρο αποφάσεων. Το στιγμιότυπο της αρχικής οθόνης φαίνεται στην *Εικόνα 17*.

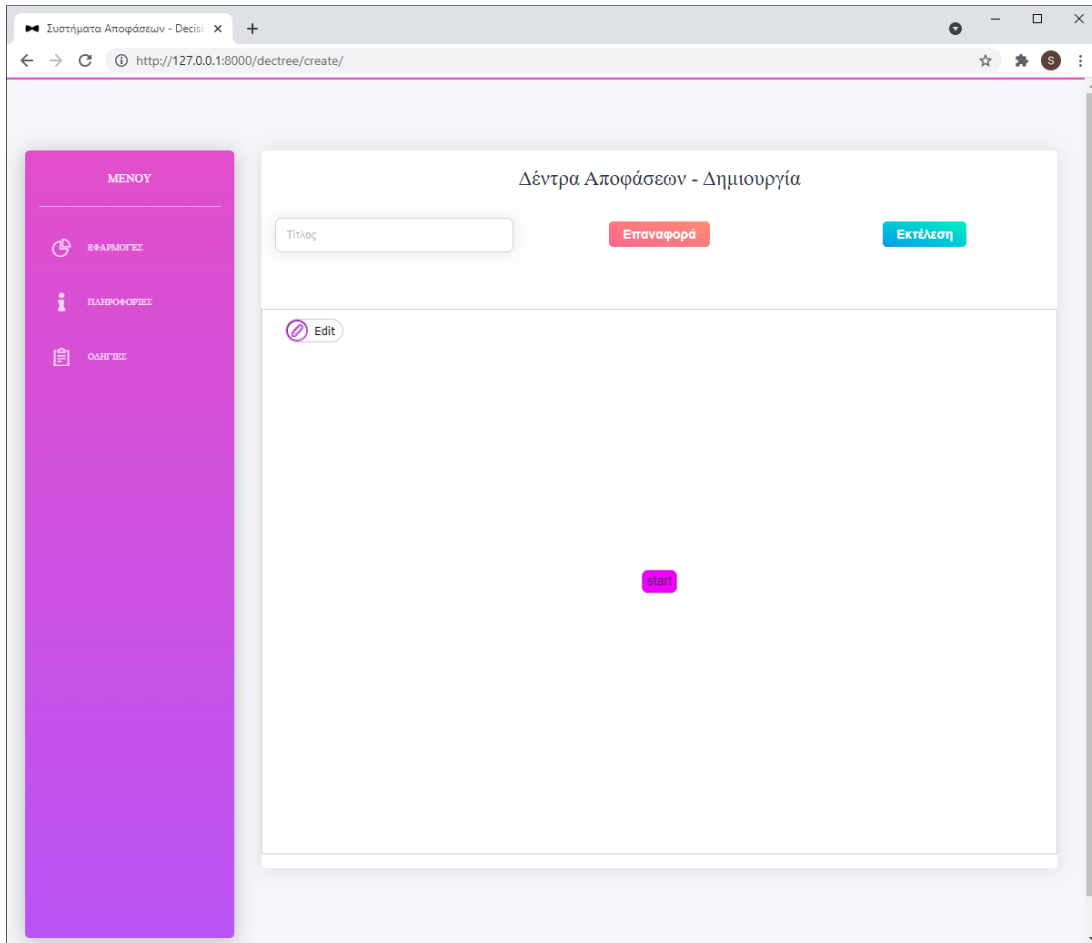


Εικόνα 17- Αρχική οθόνη εφαρμογής δέντρων αποφάσεων

Σε αυτή ο χρήστης καλείται να επιλέξει τον τρόπο με τον οποίο επιθυμεί να παρέχει τα δεδομένα του στην εφαρμογή. Επιλέγοντας το κουμπί *Ανέβασμα Αρχείου JSON* ο χρήστης θα ανακατευθυνθεί στη σελίδα ανεβάσματος αρχείου όπου θα κληθεί να παρέχει τα δεδομένα του μέσω ενός αρχείου JSON. Επιλέγοντας το κουμπί *Δημιουργία Δέντρου* ο χρήστης θα ανακατευθυνθεί στη σελίδα δημιουργίας ενός δέντρου αποφάσεων όπου περιέχεται το εργαλείο κατασκευής ενός δέντρου αποφάσεων.

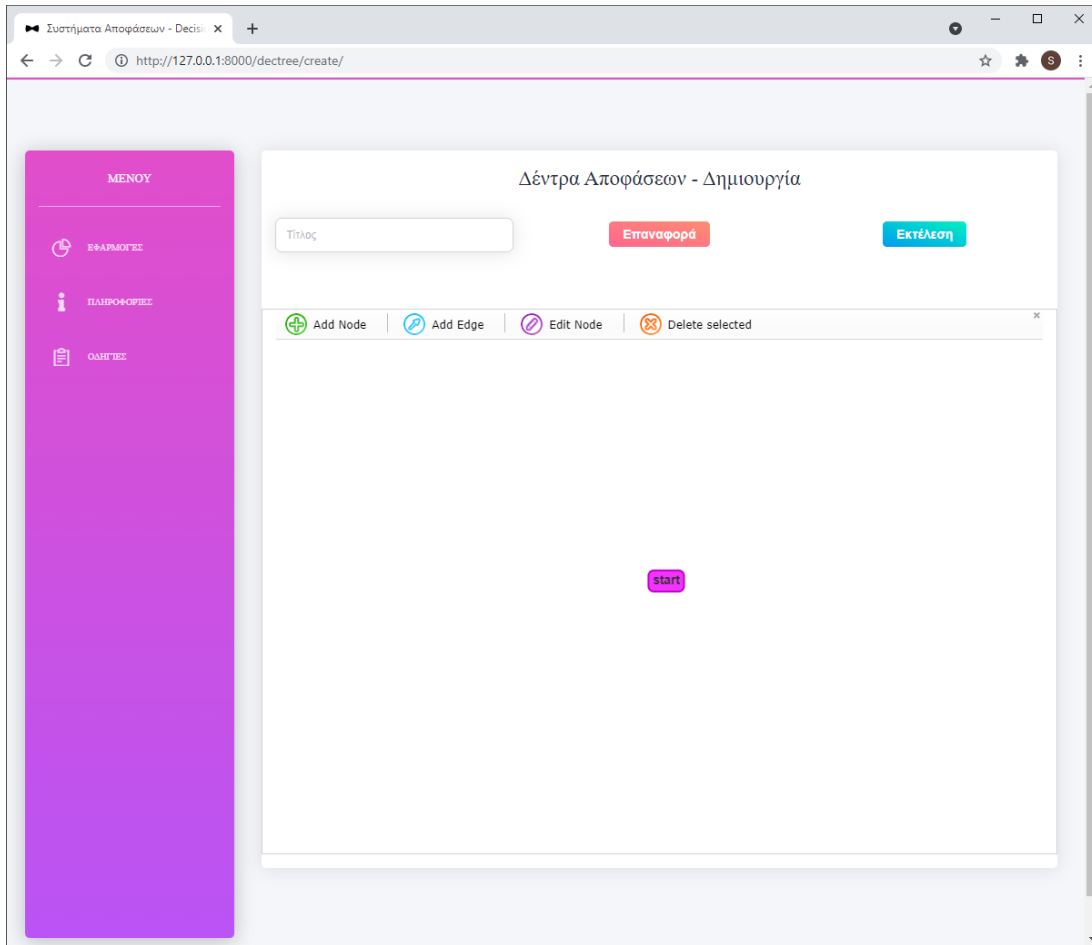
6.2.2 Οθόνη Δημιουργίας Δέντρου Αποφάσεων

Ο χρήστης μπορεί να ανακατευθυνθεί σε αυτή την οθόνη μόνο μέσω της αρχικής οθόνης της εφαρμογής δέντρων αποφάσεων. Το στιγμιότυπο της οθόνης φαίνεται στην *Εικόνα 18*.



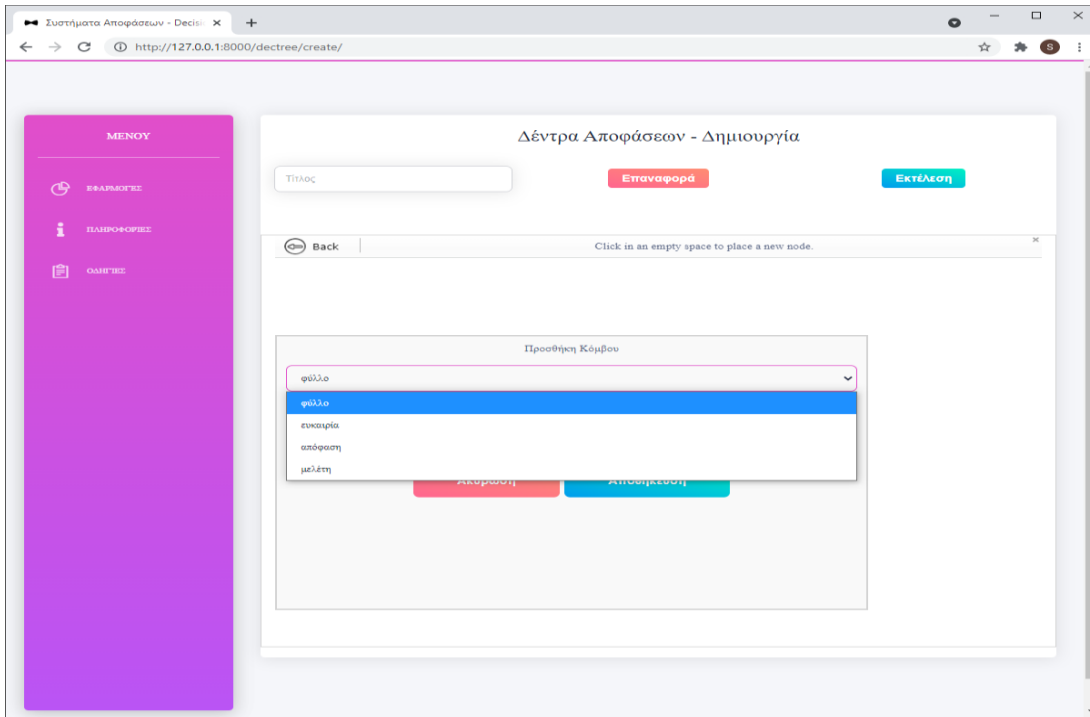
Εικόνα 18- Οθόνη δημιουργίας δέντρου αποφάσεων

Εδώ, ο χρήστης καλείται να συμπληρώσει τον τίτλο του προβλήματός του, αν το επιθυμεί, στο πεδίο *Τίτλος*. Επιπλέον, συναντά και το εργαλείο στο οποίο μπορεί να κατασκευάσει ένα δέντρο αποφάσεων. Σε αυτό υπάρχει ήδη ο αρχικός κόμβος του δέντρου, ο οποίος είναι ένας κόμβος απόφασης τον οποίο καλείται να χρησιμοποιήσει. Ο χρήστης μπορεί να ξεκινήσει να κατασκευάσει το δέντρο του επιλέγοντας το κουμπί *Edit*, ώστε να του εμφανιστούν τα εργαλεία που έχει στη διάθεσή του (Εικόνα 19). Όπως φαίνεται και στο στιγμιότυπο ο χρήστης μπορεί να επιλέξει να προσθέσει έναν κόμβο (κουμπί *Add Node*), να προσθέσει μια ακμή (κουμπί *Add Edge*), να επεξεργαστεί έναν επιλεγμένο κόμβο ή ακμή (κουμπί *Edit Node/Edge*) και να διαγράψει ένα επιλεγμένο κόμβο ή ακμή (κουμπί *Delete selected*). Στη συνέχεια παρουσιάζεται η διαδικασία κατασκευής ενός απλού δέντρου αποφάσεων το οποίο δεν περιέχει μελέτη.

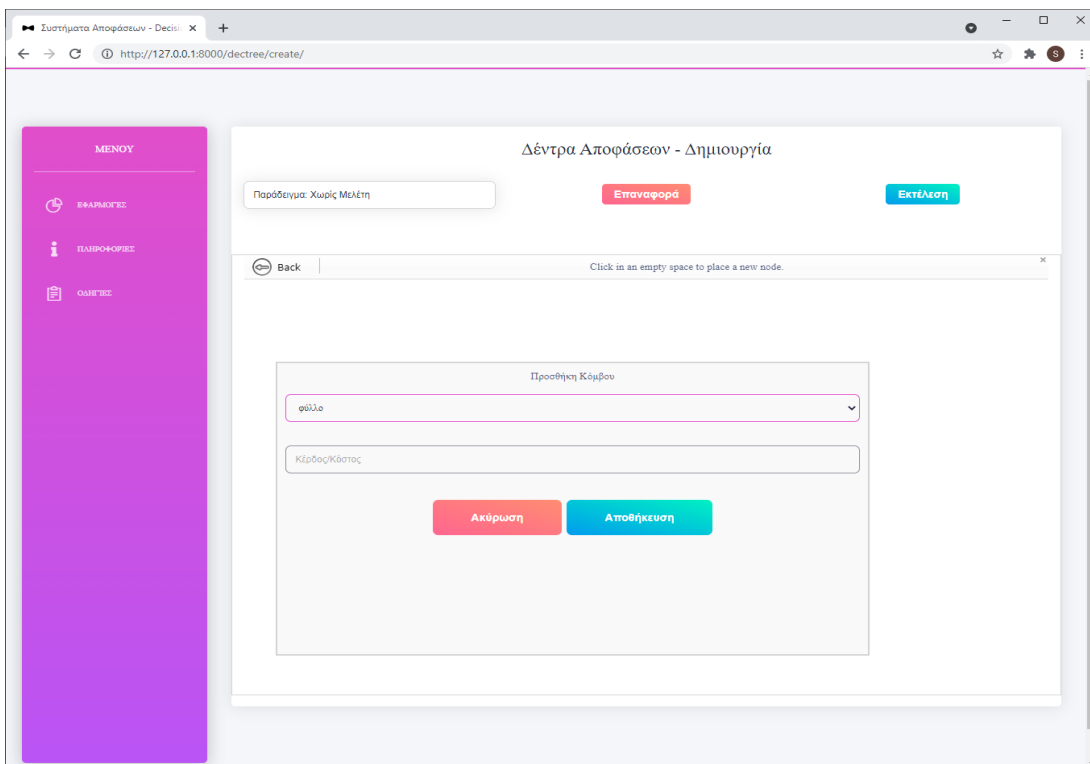


Εικόνα 19- Επιλογές επεξεργασίας δέντρου αποφάσεων

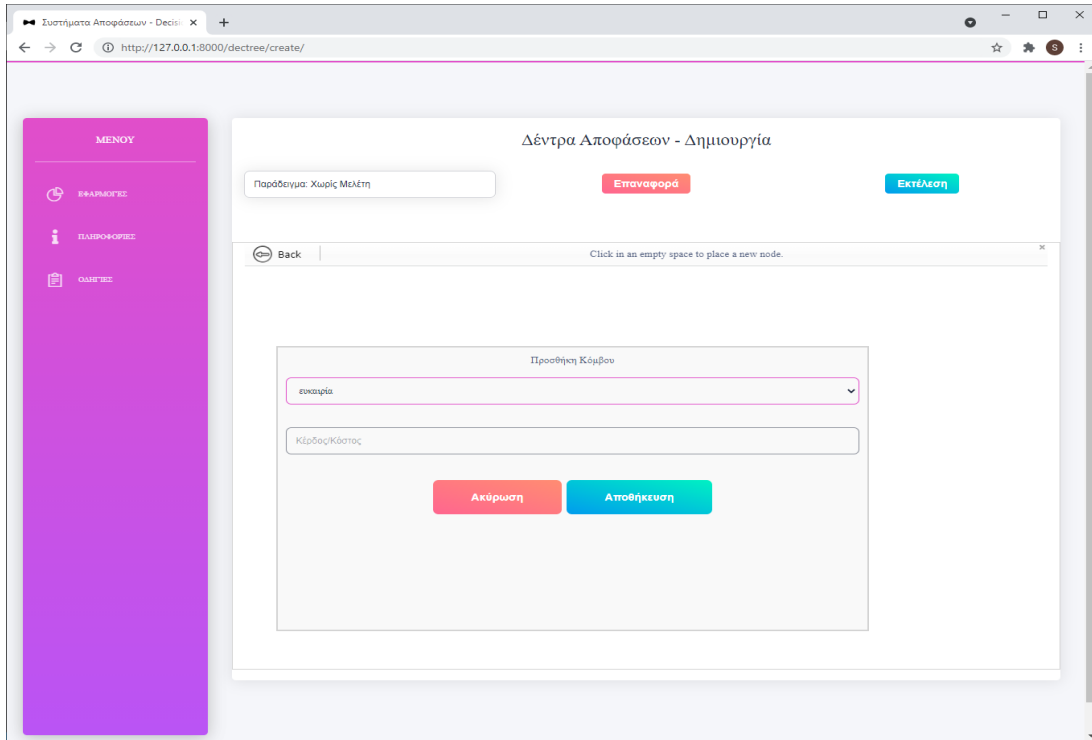
Επιλέγοντας ο χρήστης να προσθέσει έναν κόμβο στο δέντρο του, του εμφανίζεται η φόρμα όπως φαίνεται στο στιγμιότυπο της *Εικόνας 20*. Σε αυτή ο χρήστης αρχικά καλείται να επιλέξει το είδος του κόμβου που επιθυμεί να προσθέσει. Σε αυτό το πεδίο ο χρήστης μπορεί να επιλέξει ανάμεσα σε τέσσερα είδη κόμβων (κόμβο φύλλο, κόμβο ευκαιρίας, κόμβο απόφασης και κόμβο μελέτης) όπως φαίνεται στο στιγμιότυπο της *Εικόνας 21*. Ανάλογα με το είδος κόμβου που ο χρήστης επιλέξει η φόρμα συμπλήρωσης διαμορφώνεται και αναλόγως. Τα πιθανά σενάρια φαίνονται στις *Εικόνες 22, 23, 24, 25*.



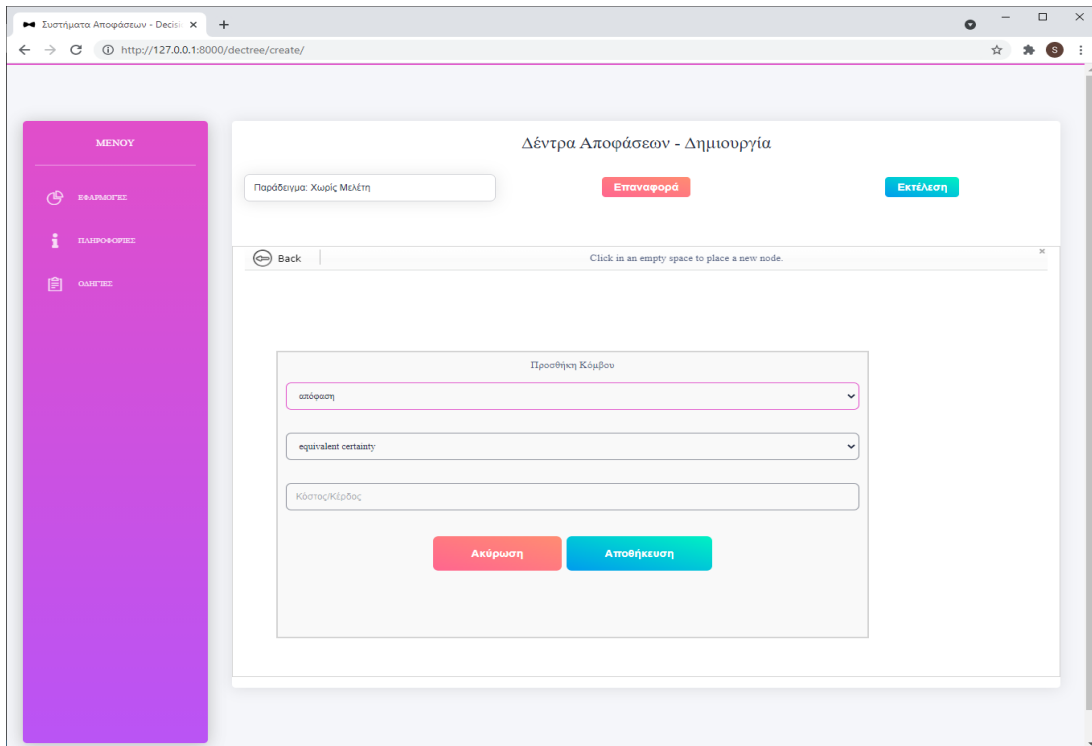
Εικόνα 20- Προσθήκη κόμβου



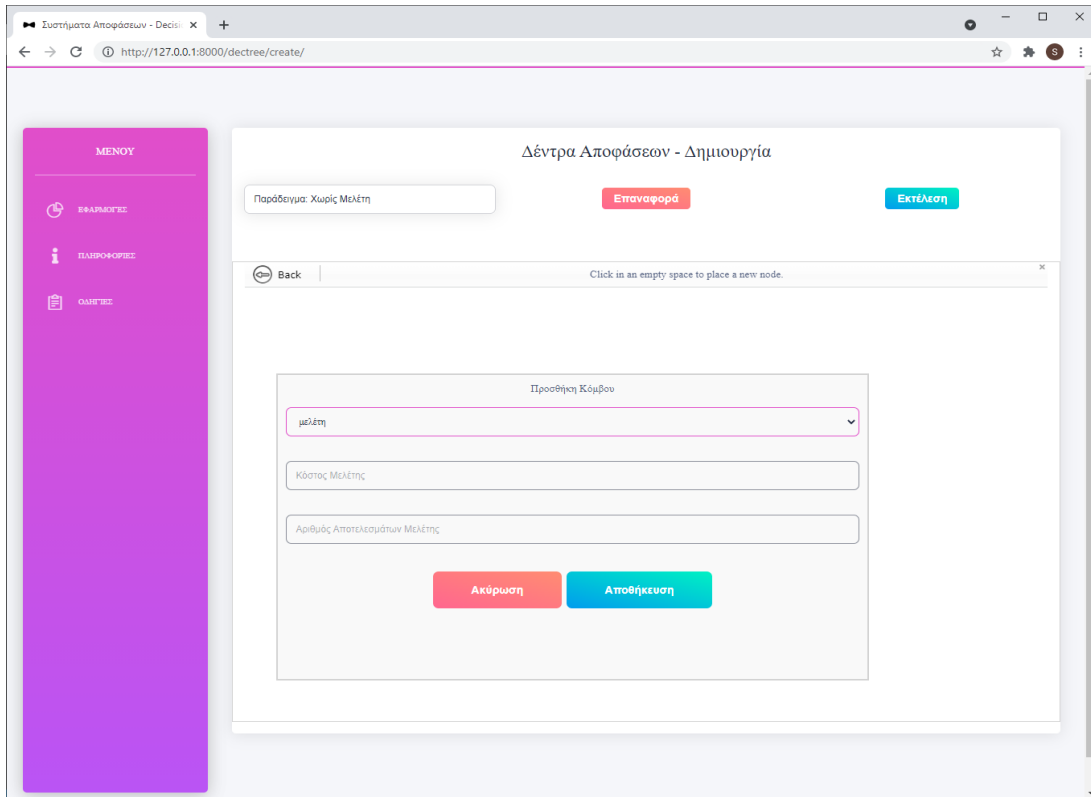
Εικόνα 21- Προσθήκη κόμβου φύλλο



Εικόνα 24- Προσθήκη κόμβου ευκαιρίας



Εικόνα 23- Προσθήκη κόμβου απόφασης



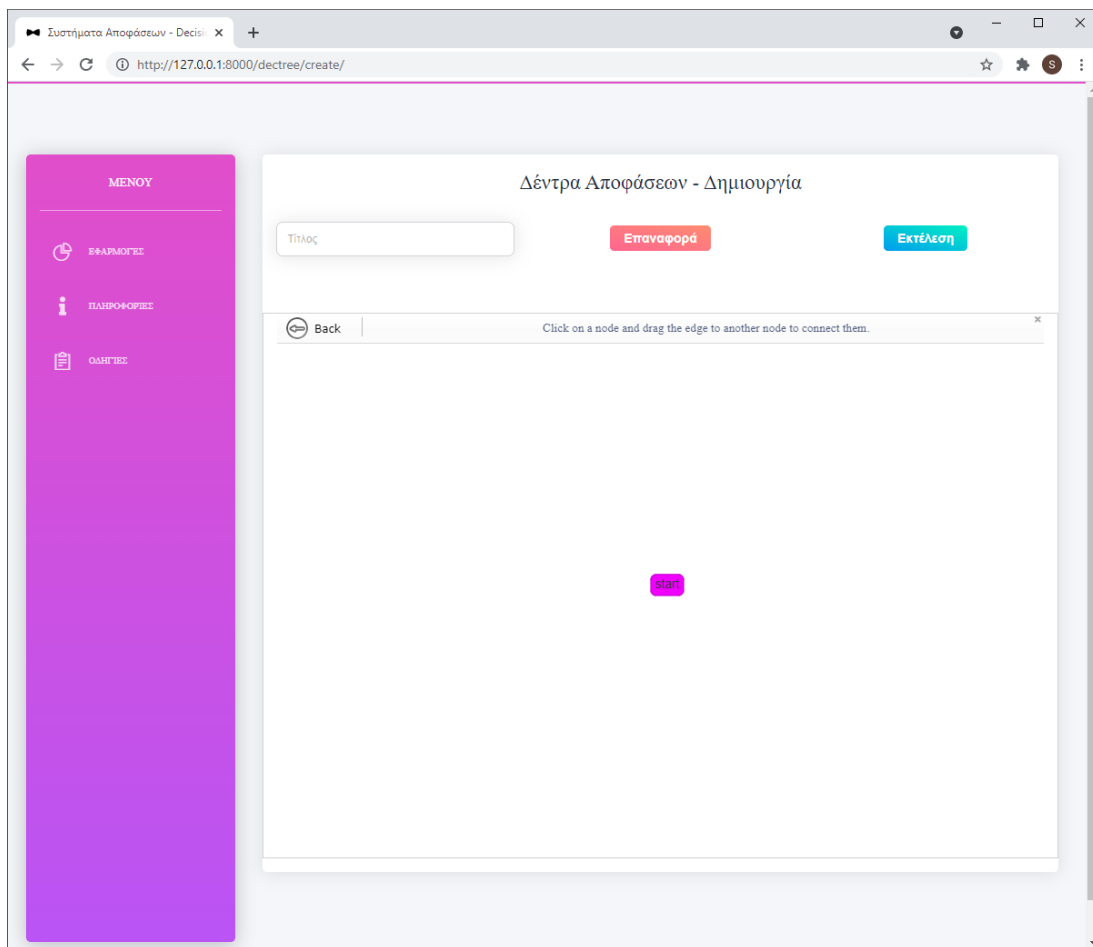
Εικόνα 24- Προσθήκη κόμβου μελέτης

Συνεπώς ανάλογα με την επιλογή του χρήστη στο είδος του κόμβου έχει τις εξής επιλογές:

- Προσθήκη κόμβου φύλλο: Όπως φαίνεται και στη *Εικόνα 22* ο χρήστης καλείται να συμπληρώσει σε αριθμητική τιμή το κόστος (αρνητική τιμή) ή το κέρδος (θετική τιμή) που επιφέρει αυτός ο τερματικός κόμβος. Ο κόμβος φύλλο παρουσιάζεται στο εργαλείο με στρόγγυλο σχήμα και πράσινο χρώμα.
- Προσθήκη κόμβου ευκαιρίας: Όπως φαίνεται στη *Εικόνα 23* ο χρήστης καλείται να συμπληρώσει σε αριθμητική τιμή το κόστος (αρνητική τιμή) ή το κέρδος (θετική τιμή) που επιφέρει αυτός ο κόμβος ευκαιρίας. Ο κόμβος ευκαιρίας παρουσιάζεται στο εργαλείο με στρόγγυλο σχήμα και κόκκινο χρώμα.
- Προσθήκη κόμβου απόφασης: Όπως φαίνεται στη *Εικόνα 24* ο χρήστης καλείται να συμπληρώσει σε αριθμητική τιμή το κόστος (αρνητική τιμή) ή το κέρδος (θετική τιμή) που επιφέρει αυτός ο κόμβος απόφασης. Στη συνέχεια πρέπει να επιλέξει το κριτήριο το οποίο θα καθορίσει την πολιτική με την οποία ο κόμβος αυτός θα χειριστεί τις αβέβαιες εκβάσεις παιδιά του, αν αυτά υπάρχουν. Ο κόμβος απόφασης παρουσιάζεται στο εργαλείο με τετράγωνο σχήμα και μπλε χρώμα.

- Προσθήκη κόμβου μελέτης: Όπως φαίνεται στην *Εικόνα 25* ο χρήστης καλείται να συμπληρώσει σε αριθμητική τιμή το κόστος της μελέτης (θετική τιμή) και στη συνέχεια να συμπληρώσει το πλήθος των αποτελεσμάτων που έχει αυτή η μελέτη. Ο κόμβος μελέτης παρουσιάζεται στο σχήμα με στρόγγυλο σχήμα και πράσινο χρώμα.

Επιλέγοντας ο χρήστης να προσθέσει μια ακμή στο δέντρο του, λαμβάνει το στιγμιότυπο της *Εικόνας 26* όπου τον καθοδηγεί εκκινώντας από έναν κόμβο να σύρει την ακμή μέχρι τον κόμβο που επιθυμεί να ενώσει. Σε αυτό το σημείο πρέπει να δώσει προσοχή διότι θα πρέπει να ενώσει τον κόμβο γονιό με τον κόμβο παιδί. Ανάλογα με τα είδη των κόμβων γονιού και παιδιού ο χρήστης καλείται να συμπληρώσει και διαφορετικού είδους δεδομένα.

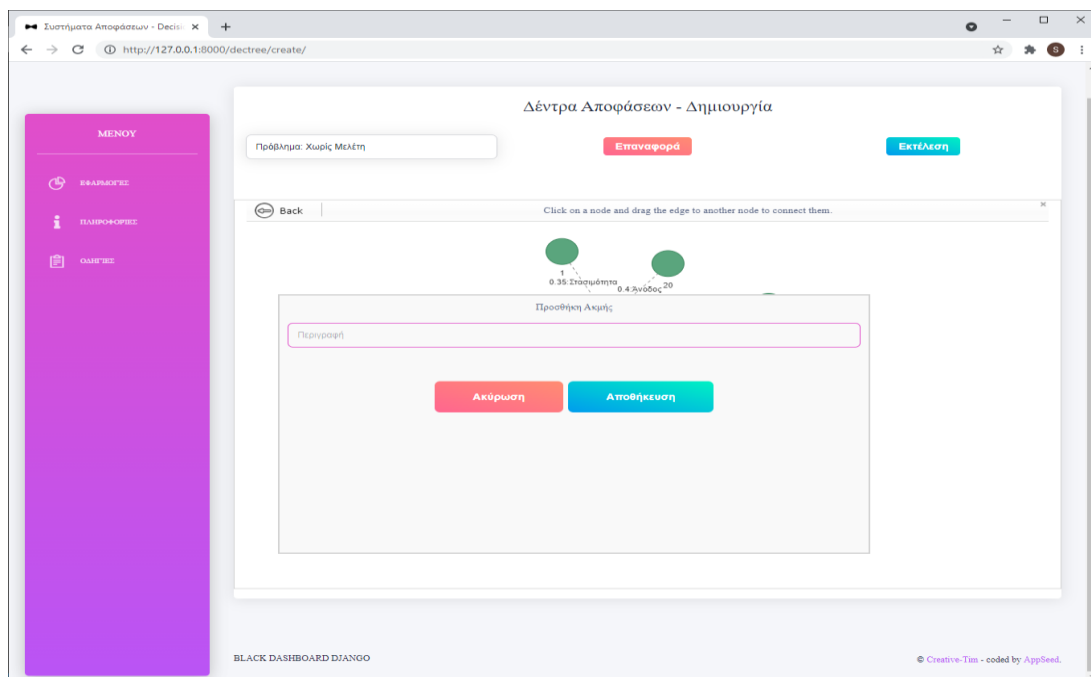


Εικόνα 25- Χάραξη ακμής

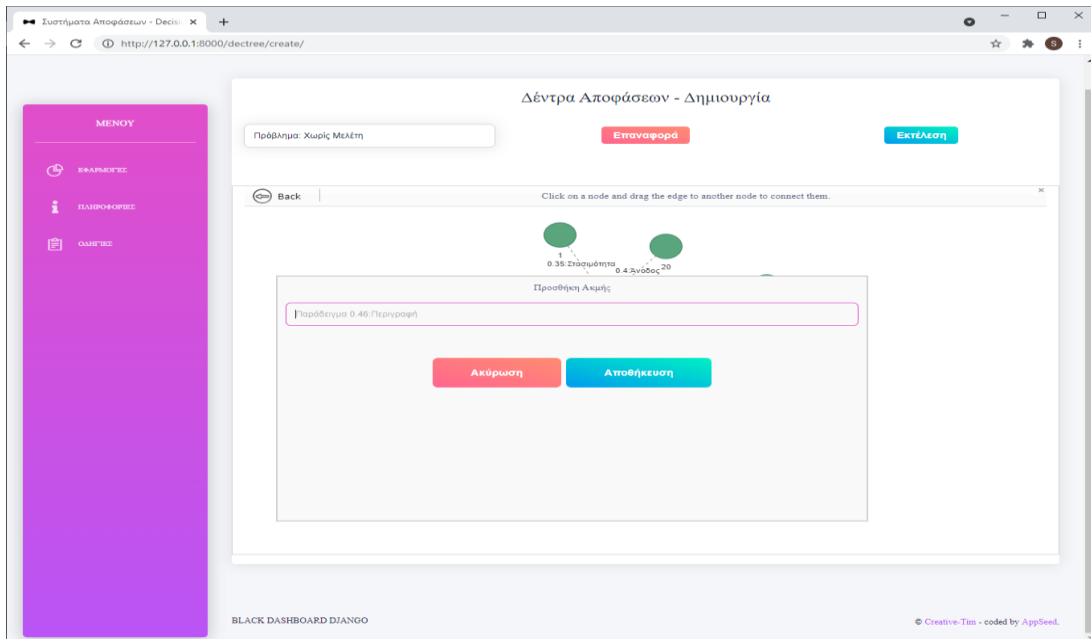
Οι διαφορετικοί συνδυασμοί κόμβων γονιού – παιδιού φαίνονται παρακάτω το πρώτο μέρος της δυάδας υποδηλώνει το γονιό και το δεύτερο το παιδί:

- Στις περιπτώσεις (απόφαση, φύλλο), (απόφαση, ευκαιρία) και (απόφαση, απόφαση) ο χρήστης λαμβάνει το στιγμιότυπο της *Εικόνας 27*, όπου καλείται να συμπληρώσει την περιγραφή της ακμής που επιθυμεί να κατασκευάσει.
- Στις περιπτώσεις (ευκαιρία, φύλλο), (ευκαιρία, ευκαιρία) και (ευκαιρία, απόφαση) ο χρήστης λαμβάνει το στιγμιότυπο της *Εικόνας 28*, όπου καλείται να συμπληρώσει την περιγραφή της ακμής που επιθυμεί να κατασκευάσει με τη διαφορά ότι λόγω ύπαρξης γονιού - ευκαιρίας θα πρέπει να περιλαμβάνει και την πιθανότητα να εκφραστεί η συγκεκριμένη έκβαση σε μορφή δεκαδικού, όπως φαίνεται στην περιγραφή του πεδίου.
- Στη περίπτωση (απόφαση, μελέτη) ο χρήστης λαμβάνει το στιγμιότυπο, στο οποίο καλείται αρχικά να συμπληρώσει τον τίτλο της μελέτης και στη συνέχεια να συμπληρώσει τον πίνακα αξιοπιστίας της μελέτης. Συγκεκριμένα ο χρήστης θα πρέπει να δώσει τις περιγραφές των πιθανών αποτελεσμάτων της μελέτης και τις δεσμευμένες πιθανότητες των εκβάσεων της μελέτης δεδομένων των αποτελεσμάτων των αβέβαιων εκβάσεων του κόμβου απόφασης. Το παραπάνω στιγμιότυπο παρουσιάζεται στην περιγραφή της σελίδας επεξεργασία δέντρου αποφάσεων *Εικόνα 36* όπου και θα καλυφθεί το σενάριο χρήσης κατασκευής δέντρου αποφάσεων με μελέτη.

Σε κάθε άλλο συνδυασμό ο χρήστης λαμβάνει ειδοποίηση που του αναφέρει ότι δεν του επιτρέπει να προχωρήσει σε αυτή την ενέργεια διότι παραβιάζει τους κανόνες ενός δέντρου αποφάσεων.

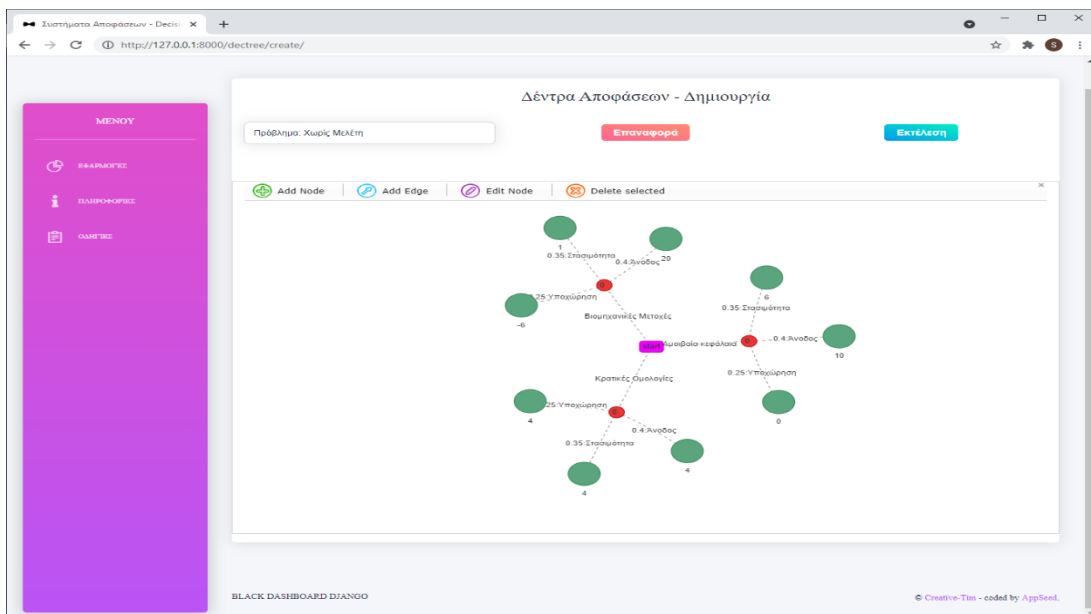


Εικόνα 26- Προσθήκη απλής ακμής



Εικόνα 27- Προσθήκη ακμής αθέβαιου γεγονότος

Έχοντας στη διάθεσή του τις παραπάνω πληροφορίες, ο χρήστης μπορεί να κατασκευάσει ένα πλήρες δέντρο αποφάσεων χωρίς περιορισμό στο πλήθος κόμβων και ακμών. Ένα παράδειγμα ενός απλού, αλλά πλήρους, δέντρου αποφάσεων είναι αυτό που παρουσιάζεται στην *Εικόνα 29*. Τα δεδομένα εισόδου λήφθηκαν από τις ασκήσεις του μαθήματος Συστήματα Αποφάσεων της ΣΗΜΜΥ του ΕΜΠ [7].



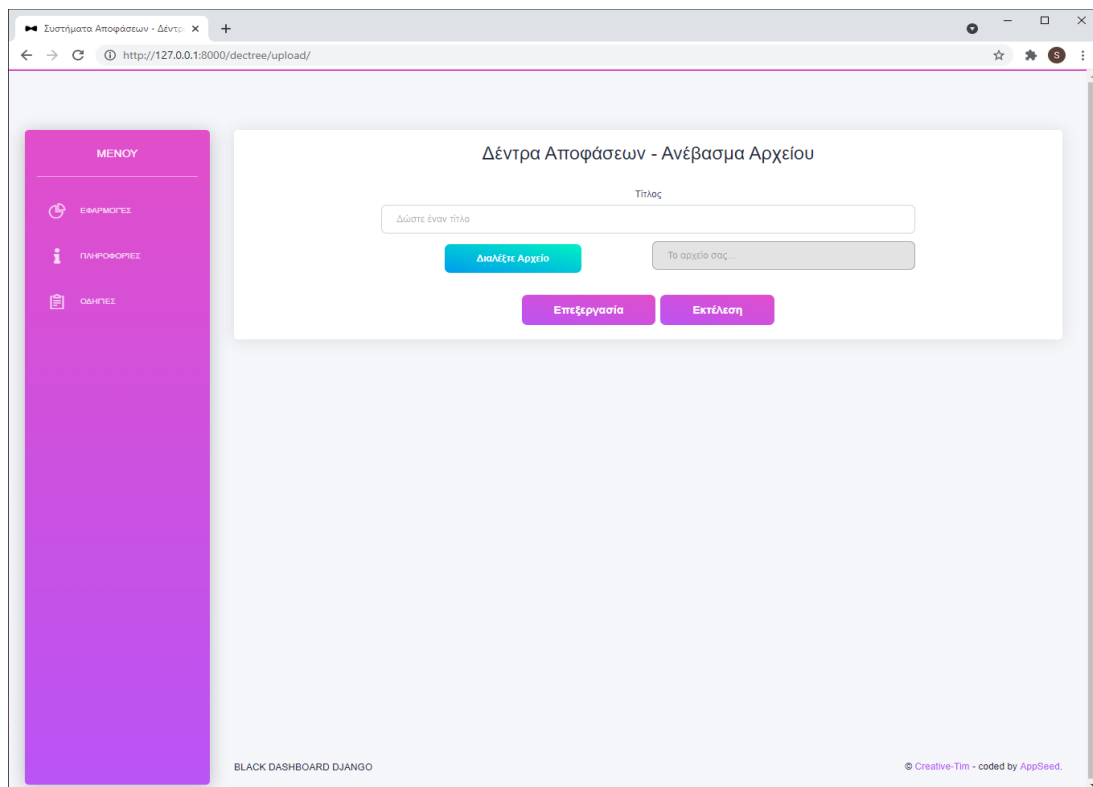
Εικόνα 28- Παράδειγμα δέντρου αποφάσεων χωρίς μελέτη

Σε αυτό το σημείο να αναφερθεί ότι ο χρήστης, σε οποιαδήποτε στιγμή, έχει τη δυνατότητα να επαναφέρει το δέντρο του στην αρχική κατάσταση, δηλαδή να περιέχει μόνο τον αρχικό κόμβο. Αυτό επιτυγχάνεται με το κουμπί *Επιβεβαίωση* που βρίσκεται στο πάνω μέρος της εφαρμογής.

Εφόσον ο χρήστης έχει ολοκληρώσει την κατασκευή του δέντρου του μπορεί να προχωρήσει στην επίλυση του δέντρου απόφασης επιλέγοντας το κουμπί *Εκτέλεση* που βρίσκεται στο πάνω μέρος της σελίδας. Επιλέγοντάς το θα ανακατευθυνθεί στη σελίδα παρουσίασης αποτελεσμάτων όπου και θα δει το δέντρο του επιλυμένο. Η λύση του δέντρου της *Εικόνας 29* βρίσκεται στο τμήμα 6.2.5 όπου παρουσιάζεται η οθόνη αποτελεσμάτων.

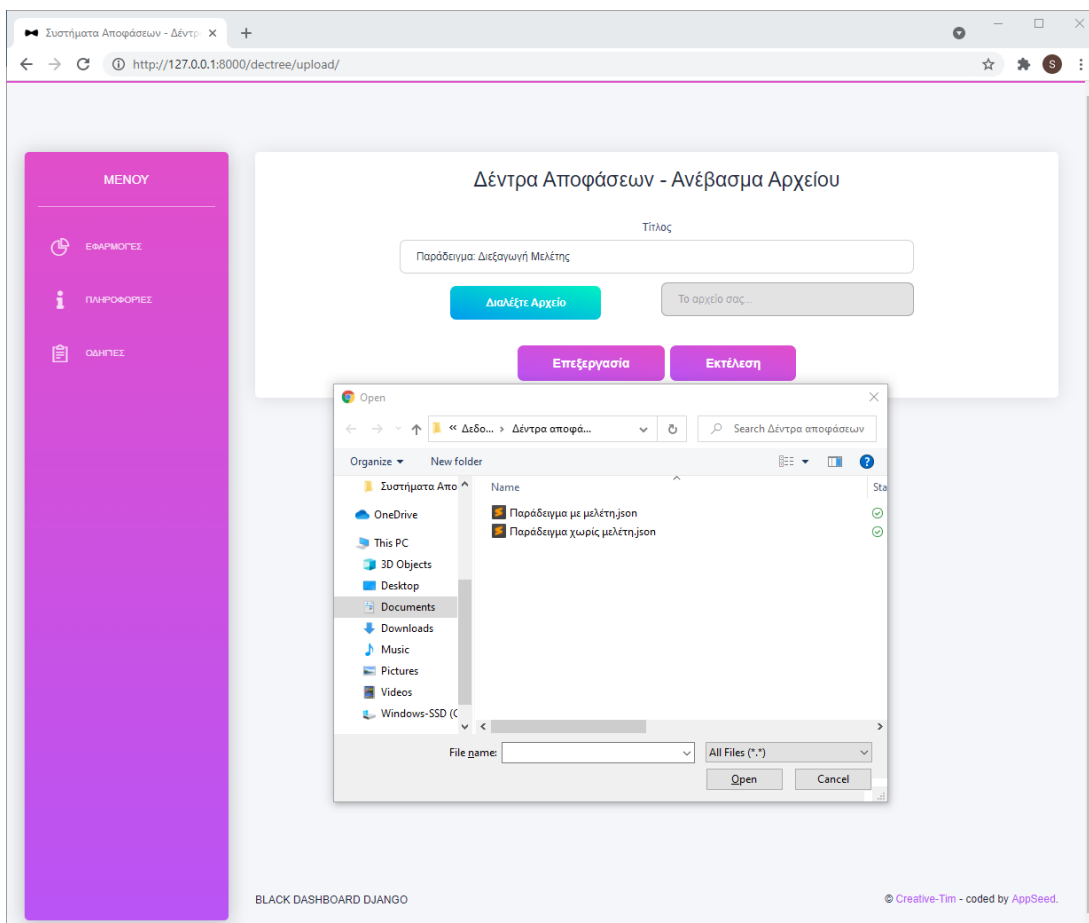
6.2.3 Οθόνη Ανεβάσματος Αρχείου

Ο χρήστης μπορεί να ανακατευθυνθεί σε αυτή την οθόνη μόνο μέσω της της αρχικής οθόνης της εφαρμογής δέντρων αποφάσεων, εφόσον επιλέξει να παρέχει τα δεδομένα εισόδου του μέσω αρχείου JSON. Το στιγμιότυπο της οθόνης φαίνεται στην *Εικόνα 30*.



Εικόνα 29- Εφαρμογή δέντρων αποφάσεων: Οθόνη ανεβάσματος αρχείου

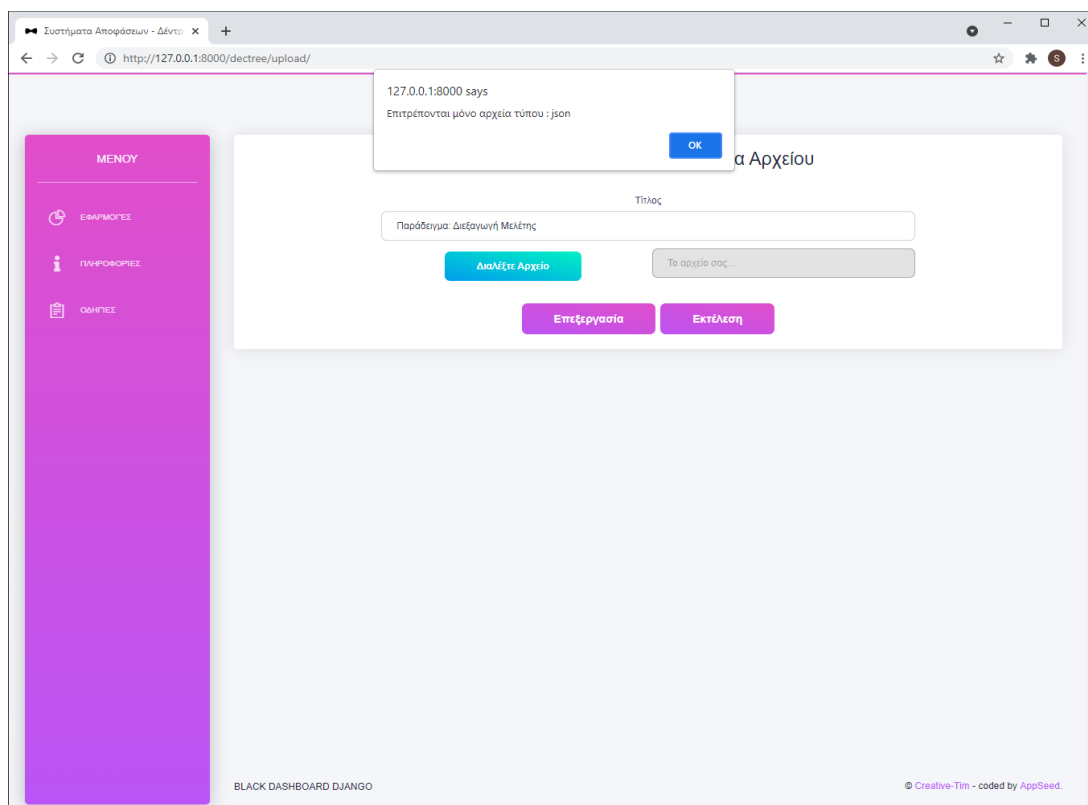
Σε αυτή ο χρήστης, εφόσον το επιθυμεί, καλείται να συμπληρώσει τον τίτλο του προβλήματος δέντρων αποφάσεων που επιθυμεί να επιλύσει στο πεδίο *Δώστε έναν τίτλο*. Στη συνέχεια, επιλέγοντας το κουμπί *Διαλέξτε Αρχείο* θα του εμφανιστεί ένα παράθυρο με τα αρχεία του τοπικού του υπολογιστή από όπου θα πρέπει να επιλέξει ένα αρχείο. Το στιγμιότυπο της παραπάνω ενέργειας φαίνεται στη *Εικόνα 31*.



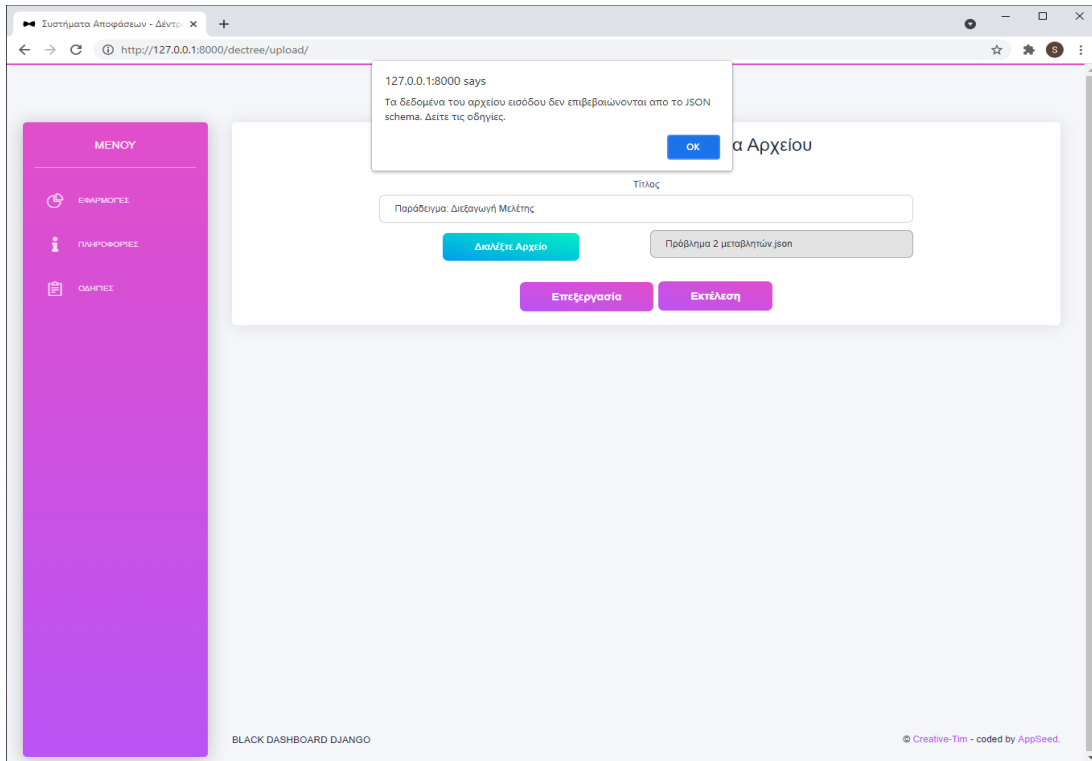
Εικόνα 30- Εφαρμογή δέντρων αποφάσεων: Επιλογή αρχείου

Σε αυτό το σημείο ο χρήστης θα πρέπει να επιλέξει ένα αρχείο με κατάληξη *.json*, το οποίο θα περιέχει τα δεδομένα του προβλήματος του. Αν το αρχείο που επιλέξει δεν είναι τύπου *JSON*, τότε θα λάβει την ειδοποίηση που φαίνεται στο στιγμιότυπο της *Εικόνας 32* και θα κληθεί να επιλέξει ένα άλλο αρχείο. Εφόσον ο χρήστης επιλέξει σωστού τύπου αρχείο (*Εικόνα 34*) καλείται να επιλέξει αν θέλει να λύσει το δέντρο του ως έχει, ή αν επιθυμεί να επεξεργαστεί το δέντρο του πριν την εκτέλεση. Με το κουμπί *Επεξεργασία* ο χρήστης ανακατευθύνεται στη σελίδα επεξεργασίας δέντρων αποφάσεων όπου θα δει το δέντρο που περιέχεται στο αρχείο εισόδου του απεικονισμένα σε μορφή δέντρου. Με το κουμπί *Εκτέλεση* ο χρήστης θα ανακατευθυνθεί στη σελίδα αποτελεσμάτων όπου θα δει το δέντρο του επιλυμένα. Τα

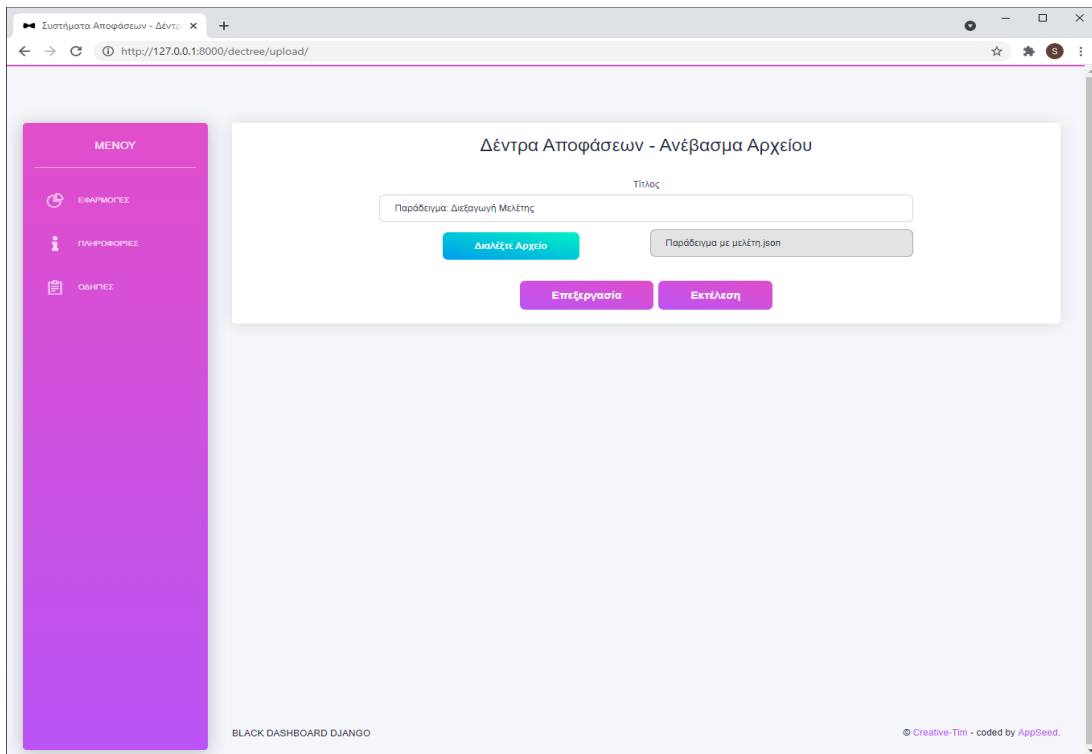
παραπάνω ισχύουν εφόσον τα δεδομένα του αρχείου εισόδου επιβεβαιώνουν το σχήμα JSON που έχει οριστεί. Εάν δεν το επιβεβαιώνει, τότε ο χρήστης θα λάβει το στιγμιότυπο της *Εικόνας 33* και θα κληθεί να επιλέξει ένα άλλο αρχείο. Στο πεδίο *Το αρχείο σας...* φαίνεται το αρχείο JSON που έχει επιλέξει ο χρήστης.



Εικόνα 31- Εφαρμογή δέντρων αποφάσεων: Σφάλμα τύπου αρχείου εισόδου



Εικόνα 32- Σφάλμα περιεχομένων αρχείου εισόδου

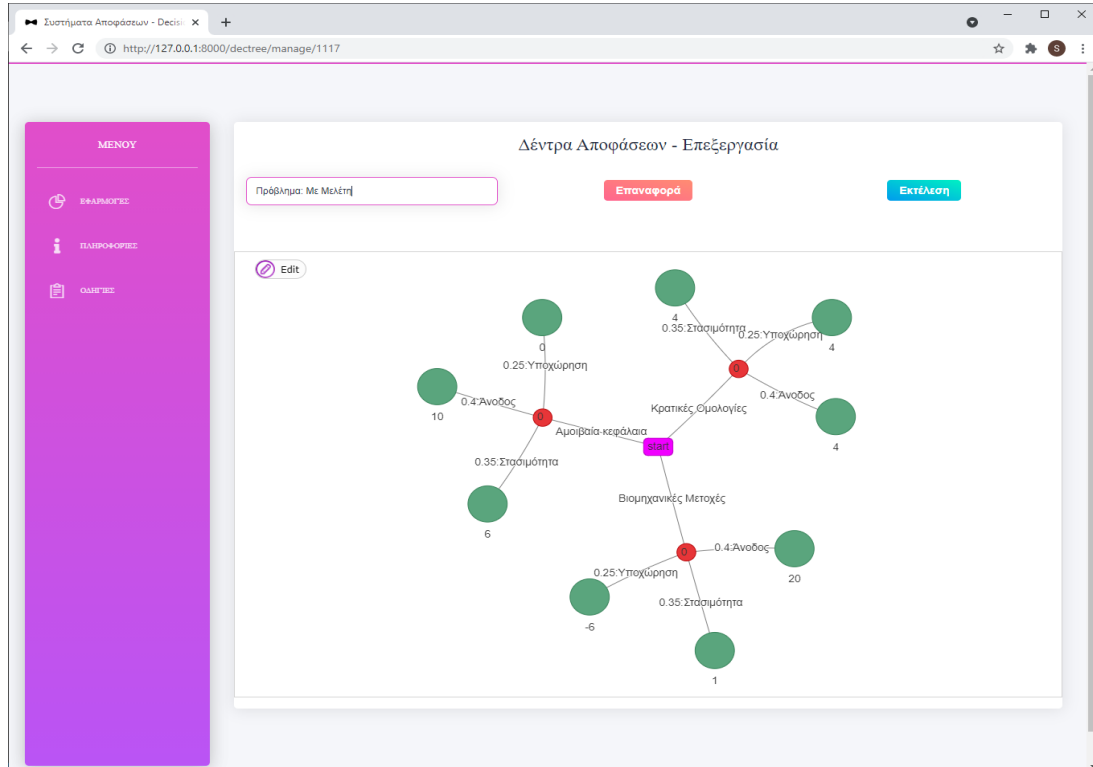


Εικόνα 33- Εφαρμογή δέντρων αποφάσεων: Φόρμα ανεβάσματος αρχείου συμπληρωμένη

6.2.4 Οθόνη Επεξεργασίας Δέντρων Αποφάσεων

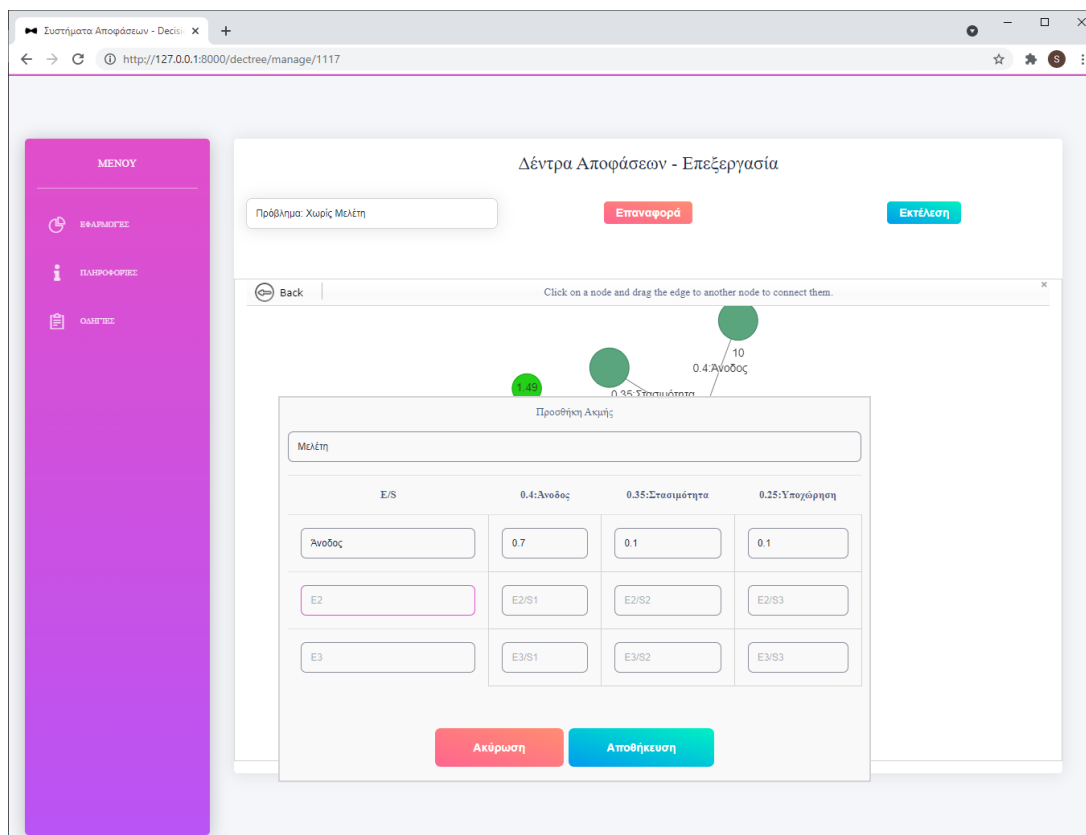
Σε αυτή την οθόνη ο χρήστης μπορεί να επεξεργαστεί τα δεδομένα εισόδου ενός προβλήματος δέντρων αποφάσεων. Υπάρχουν δύο τρόποι με τους οποίους μπορεί να ανακατευθυνθεί σε αυτή τη σελίδα. Είτε μέσω της σελίδας ανεβάσματος αρχείου είτε από τη σελίδα αποτελεσμάτων. Εδώ ο χρήστης μπορεί να αλλάξει τον τίτλο του προβλήματός του και να επεξεργαστεί ένα δέντρο αποφάσεων. Στο στιγμιότυπο της *Εικόνας 34* φαίνεται το πεδίο στο οποίο είναι εφικτή η αλλαγή του τίτλου.

Επιπλέον, σε αυτό το σημείο θα καλυφθεί και η περίπτωση χρήσης όπου ο χρήστης επιθυμεί να κατασκευάσει ένα δέντρο αποφάσεων που περιέχει μελέτη. Τα δεδομένα εισόδου λήφθηκαν από τις ασκήσεις του μαθήματος Συστήματα Αποφάσεων της ΣΗΜΜΥ του ΕΜΠ [7]. Για τη διευκόλυνση του χρήστη το εργαλείο κατασκευής και επεξεργασίας δέντρων αποφάσεων κατασκευάστηκε έτσι ώστε να μη αναγκάζει τον χρήστη να κατασκευάζει τους κόμβους και τις ακμές που δημιουργεί μια μελέτη στο δέντρο. Το μόνο που έχει να κάνει είναι κατά τη δημιουργία ακμής μεταξύ κόμβου απόφασης και κόμβου ακμής να συμπληρώσει σωστά τον πίνακα αξιοπιστίας της έρευνας που φαίνεται στο στιγμιότυπο της *Εικόνας 35* και του παρουσιάζεται κατά της χάραξη της εν λόγω ακμής. Επιπλέον, όλοι οι κανόνες σχεδιασμού που αναφέρθηκαν στο τμήμα δημιουργίας ενός δέντρου αποφάσεων ισχύουν κι εδώ.



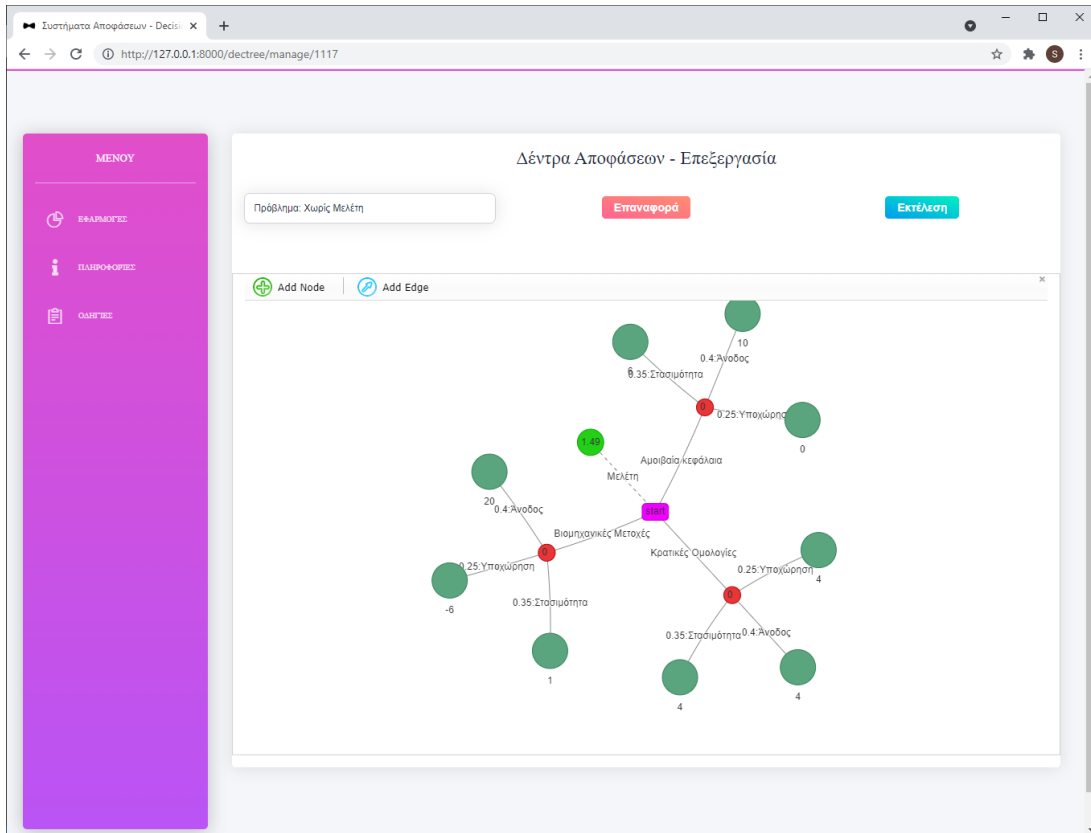
Εικόνα 34- Εφαρμογή δέντρων αποφάσεων: Οθόνη επεξεργασίας

Έστω ότι ο χρήστης επιθυμεί να προσθέσει μια μελέτη στον αρχικό κόμβο απόφασης του παραδείγματος που παρουσιάστηκε στην *Εικόνα 29*. Τότε θα πρέπει να συμπληρώσει τον πίνακα αξιοπιστίας της *Εικόνας 35*. Οι στήλες του εν λόγω πίνακα δημιουργούνται από των εκβάσεων των κόμβων ευκαιριών που είναι παιδιά του γονιού – κόμβου απόφασης – του κόμβου μελέτης.



Εικόνα 35- Προσθήκη ακμής μελέτης

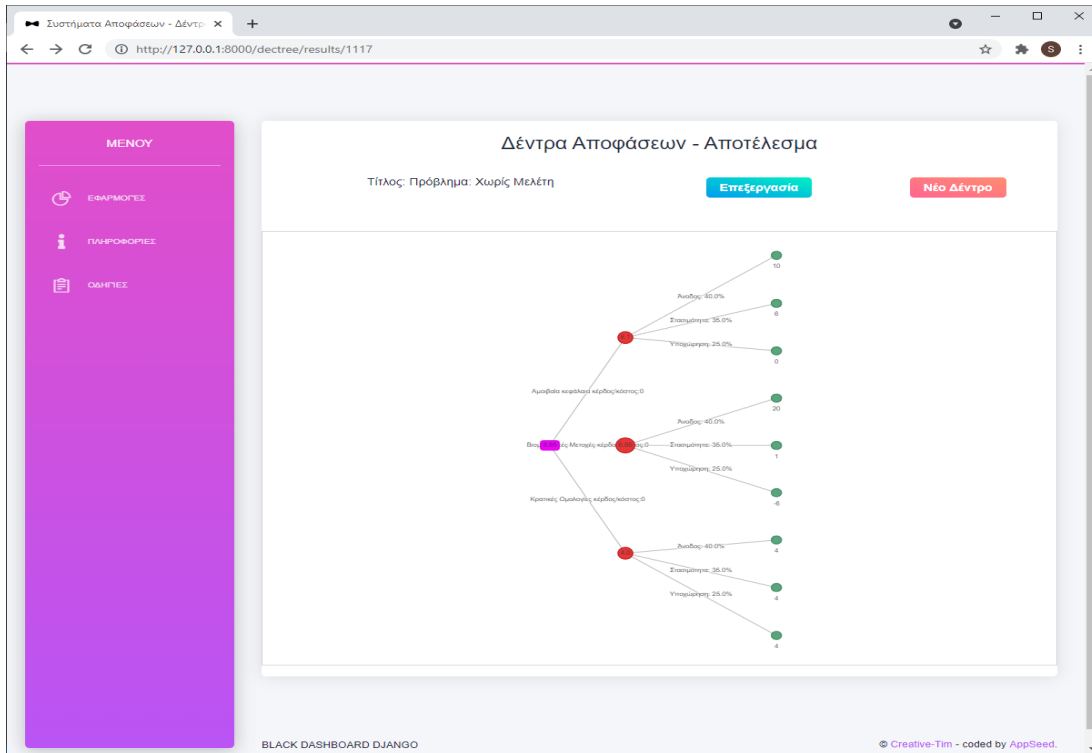
Εφόσον ο χρήστης συμπληρώσει σωστά τις αριθμητικές τιμές του πίνακα μπορεί να προχωρήσει στον υπολογισμό του δέντρου αποφάσεων που περιλαμβάνει μελέτη. Αυτό επιτυγχάνεται επιλέγοντας το κουμπί *Εκτέλεση*. Επιλέγοντάς το ανακατευθύνεται στη σελίδα αποτελεσμάτων όπου μπορεί να δει την επίλυσή του. Στην *Εικόνα 36* φαίνεται το στιγμιότυπο της οθόνης με το δέντρο αποφάσεων ολοκληρωμένο. Οι ακμές που προσέθεσε ο χρήστης στο δέντρο του παριστάνονται με διακεκομμένη γραμμή. Τέλος, όπως και στην περίπτωση της σελίδας δημιουργίας δέντρου αποφάσεων, έτσι κι εδώ, ανά πάσα στιγμή ο χρήστης μπορεί να επαναφέρει το δέντρο του στην αρχική μορφή επιλέγοντας το κουμπί *Επιναφορά*.



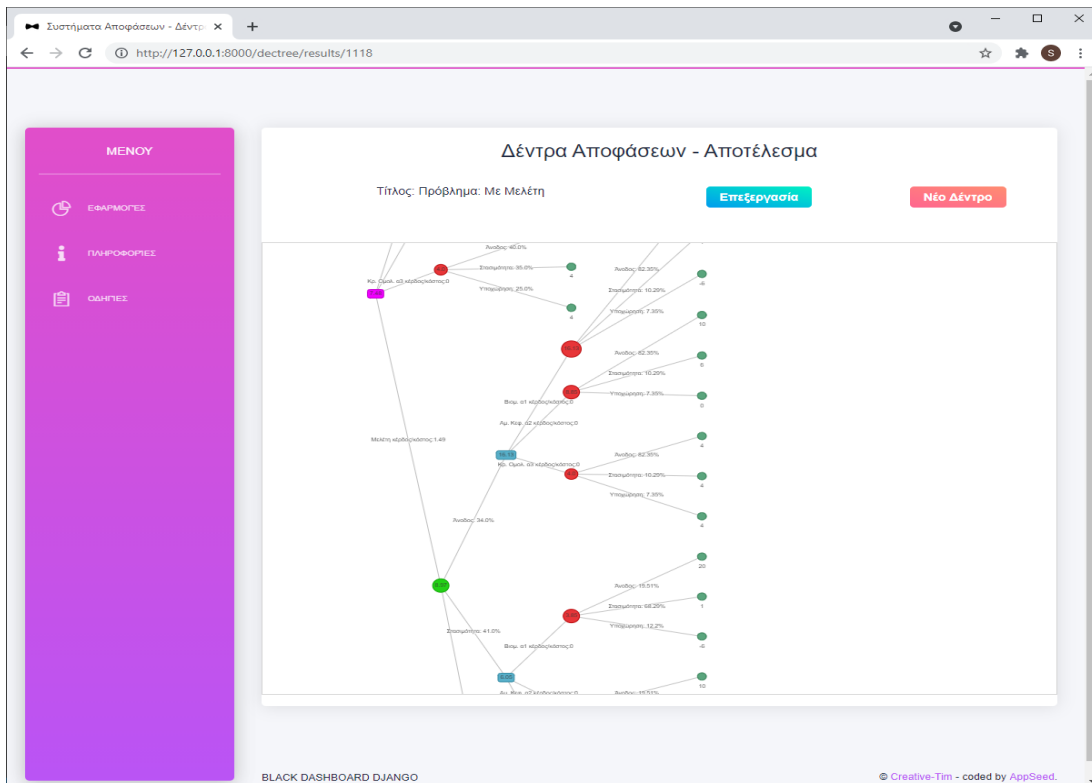
Εικόνα 36- Παράδειγμα δέντρων αποφάσεων με μελέτη

6.2.5 Οθόνη Αποτελεσμάτων

Σε αυτή σελίδα ο χρήστης μπορεί να βρεθεί είτε μέσω της σελίδας δημιουργίας δέντρου αποφάσεων, είτε μέσω της σελίδας επεξεργασίας ενός δέντρου αποφάσεων. Εδώ παρουσιάζονται τα αποτελέσματα της λύσης ενός δέντρου αποφάσεων, δηλαδή ένα δέντρο αποφάσεων με το δέντρο αποφάσεων επιλυμένο. Στο πάνω μέρος της σελίδας ο χρήστης μπορεί να δει τον τίτλο του προβλήματος, εφόσον τον έχει ορίσει. Επιπλέον έχει τη δυνατότητα να μεγεθύνει ή να σμικρύνει το δέντρο αποφάσεων ώστε να είναι ευανάγνωστες οι τιμές των κόμβων που υπολογίστηκαν. Στην *Εικόνα 37* φαίνεται η λύση του παραδείγματος που παρουσιάστηκε στην *Εικόνα 29*. Στην *Εικόνα 38* φαίνεται η λύση του παραδείγματος που παρουσιάστηκε στην *Εικόνα 36* μεγεθυμένο ώστε να φαίνεται το αποτέλεσμα της μελέτης και πώς αυτή εν τέλει διαμορφώθηκε στο δέντρο αποφάσεων από την εφαρμογή.



Εικόνα 37- Εφαρμογή δέντρων αποφάσεων: Οθόνη αποτελεσμάτων – Παράδειγμα χωρίς μελέτη



Εικόνα 38- Εφαρμογή δέντρων αποφάσεων: Οθόνη αποτελεσμάτων – Παράδειγμα με μελέτη

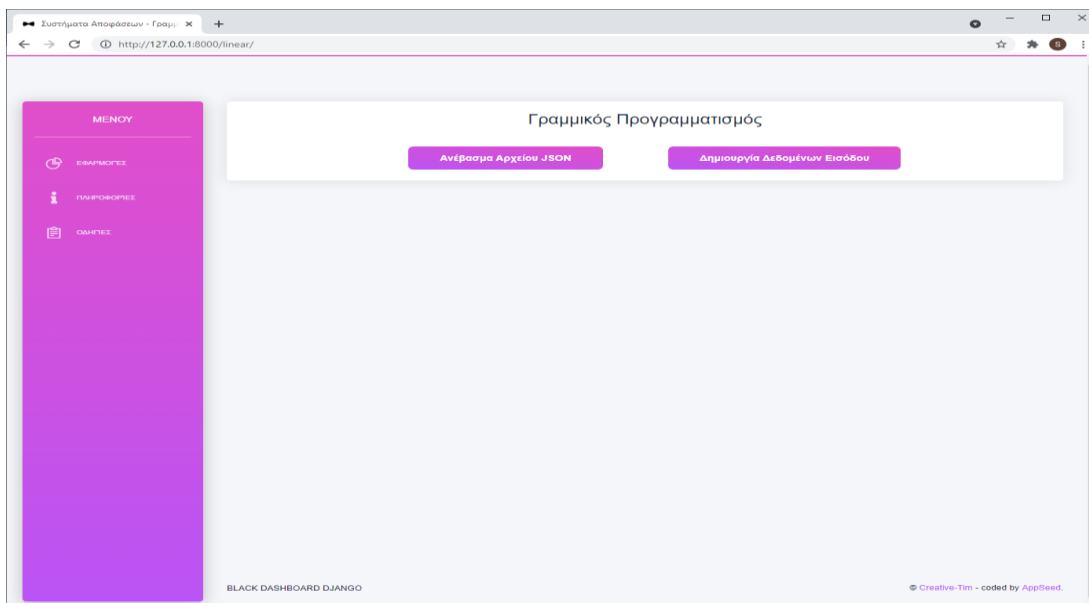
Με αυτό τον τρόπο ο χρήστης έχει ολοκληρώσει την επίλυση ενός προβλήματος δέντρων αποφάσεων. Έπειτα μπορεί είτε να κατασκευάσει ένα νέο δέντρο είτε να επεξεργαστεί τα δεδομένα εισόδου του επιλυμένου δέντρου. Η πρώτη λειτουργία επιτυγχάνεται επιλέγοντας το κουμπί *Νέο Δέντρο*, το οποίο τον ανακατευθύνει στην αρχική οθόνη της εφαρμογής των δέντρων αποφάσεων. Η δεύτερη λειτουργία επιτυγχάνεται επιλέγοντας το κουμπί *Επεξεργασία* το οποίο τον ανακατευθύνει στην οθόνη επεξεργασίας ενός δέντρου αποφάσεων.

6.3 Εφαρμογή Γραμμικού Προγραμματισμού

Σε αυτό το σημείο θα περιγραφούν τα σενάρια χρήσης της εφαρμογής Γραμμικού Προγραμματισμού. Δηλαδή θα παρουσιαστούν οι δυνατότητες που παρέχει η εφαρμογή στον χρήστη και πώς αυτή ανταποκρίνεται σε κάθε κίνηση του χρήστη μέσα σε αυτή.

6.3.1 Αρχική Οθόνη

Εφόσον ο χρήστης έχει επιλέξει ότι θέλει να κάνει χρήση της εφαρμογής γραμμικού προγραμματισμού, ανακατευθύνεται στην αρχική οθόνη της εφαρμογής γραμμικού προγραμματισμού. Μια άλλη περίπτωση με την οποία ο χρήστης μπορεί να βρεθεί σε αυτή την οθόνη είναι από την οθόνη αποτελεσμάτων στην περίπτωση που επιθυμεί να λύσει ένα νέο πρόβλημα γραμμικού προγραμματισμού. Το στιγμιότυπο της αρχικής οθόνης φαίνεται στην *Εικόνα 39*.

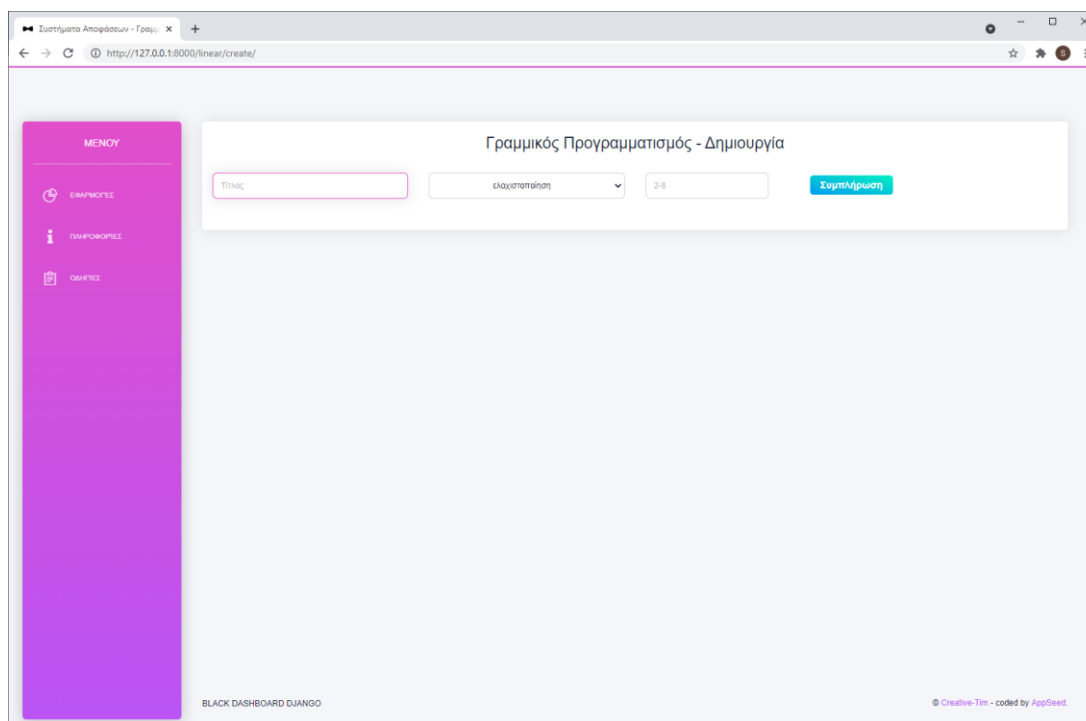


Εικόνα 39 - Εφαρμογή γραμμικού προγραμματισμού: Αρχική οθόνη

Σε αυτή την οθόνη ο χρήστης καλείται να επιλέξει τον τρόπο με τον οποίο επιθυμεί να παρέχει τα δεδομένα του στην εφαρμογή. Επιλέγοντας το κουμπί *Ανέβασμα Αρχείου JSON* ο χρήστης θα ανακατευθυνθεί στη σελίδα ανεβάσματος αρχείου, όπου θα κληθεί να παρέχει τα δεδομένα του μέσω ενός αρχείου JSON. Επιλέγοντας το κουμπί *Δημιουργία Δεδομένων Εισόδου*, ο χρήστης θα ανακατευθυνθεί στη σελίδα δημιουργίας ενός προβλήματος γραμμικού προγραμματισμού, όπου θα μπορέσει να συμπληρώσει τα δεδομένα του πάνω στην εφαρμογή.

6.3.2 Οθόνη Δημιουργία Προβλήματος Γραμμικού Προγραμματισμού

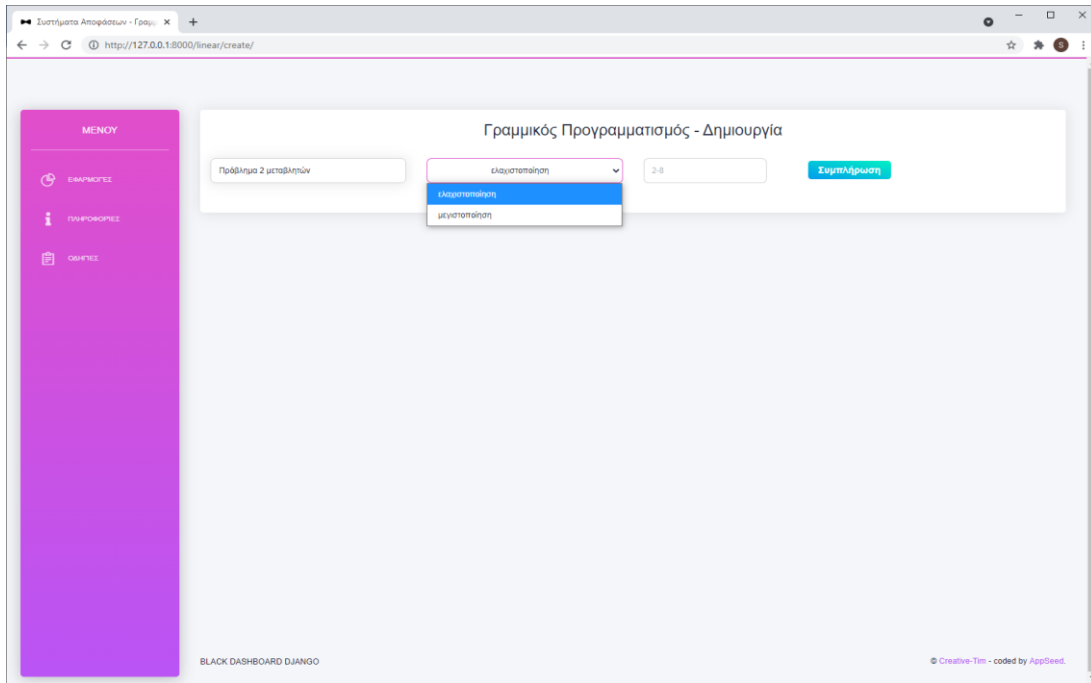
Ο χρήστης μπορεί να βρεθεί σε αυτή την οθόνη μόνο μέσω της αρχικής οθόνης της εφαρμογής γραμμικού προγραμματισμού. Το στιγμιότυπο της οθόνης φαίνεται στην *Εικόνα 40*.



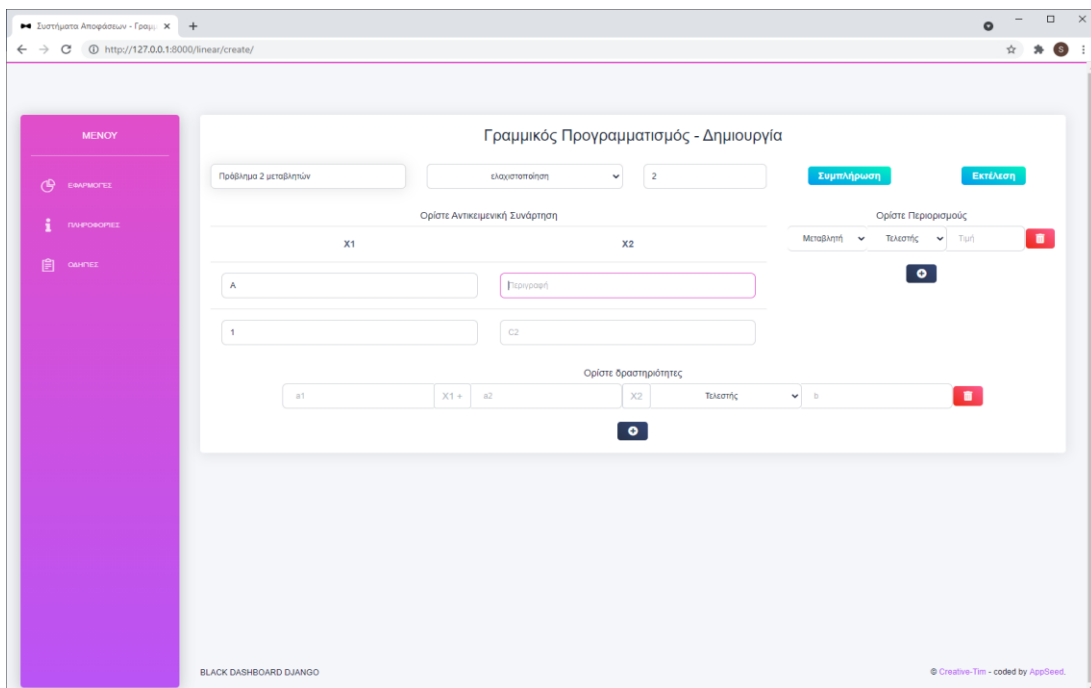
Εικόνα 40- Εφαρμογή γραμμικού προγραμματισμού: Δημιουργία προβλήματος

Εδώ, ο χρήστης καλείται αρχικά να συμπληρώσει τον τίτλο του προβλήματός του, αν το επιθυμεί. Στη συνέχεια, πρέπει να επιλέξει τη μέθοδο με την οποία θέλει να λυθεί το πρόβλημά του. Οι επιλογές του φαίνονται στην *Εικόνα 41*. Έπειτα συμπληρώνει το πλήθος των μεταβλητών απόφασης που θα έχει το πρόβλημά του. Εφόσον έχει συμπληρώσει σωστά τα παραπάνω, πατώντας το κουμπί *Συμπλήρωση* του εμφανίζεται στην οθόνη ένα σύνολο εισόδου (*Εικόνα 42*) όπου ο χρήστης θα πρέπει να

συμπληρώσει. Αν ο χρήστης είτε δεν συμπληρώσει το πεδίο των μεταβλητών απόφασης, είτε το συμπληρώσει εσφαλμένα, λαμβάνει την ειδοποίηση της *Εικόνας 43* με το αντίστοιχο σφάλμα του.



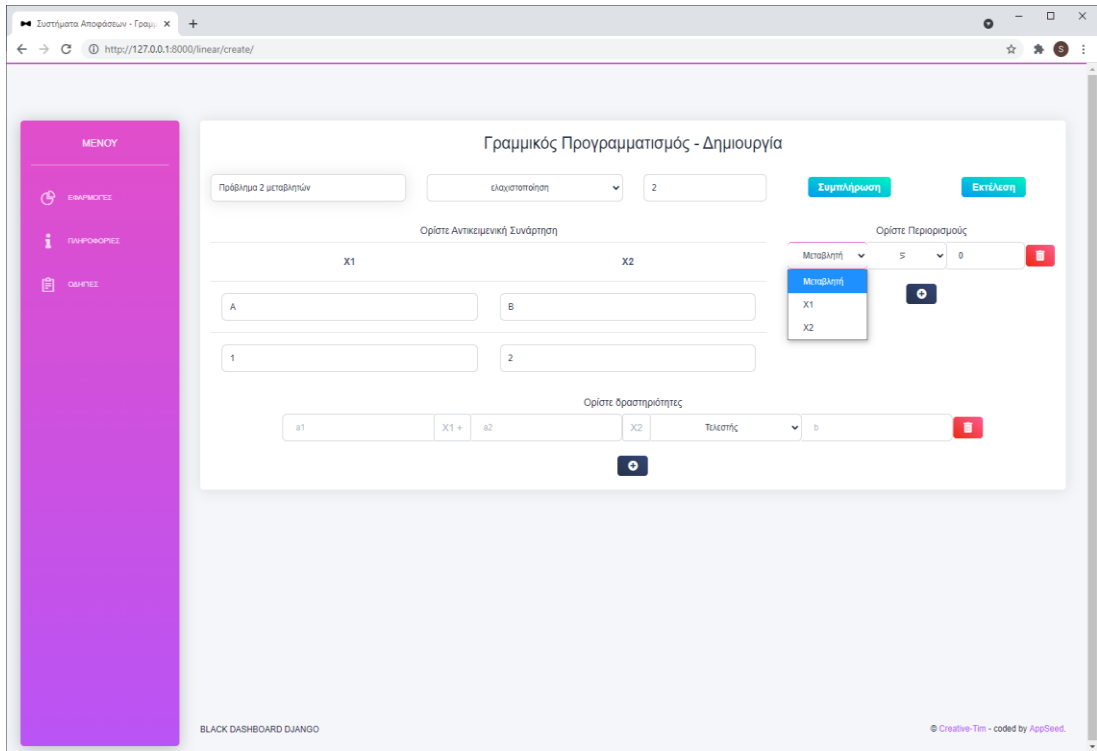
Εικόνα 41- Εφαρμογή γραμμικού προγραμματισμού: Επιλογή κριτηρίου επίλυσης



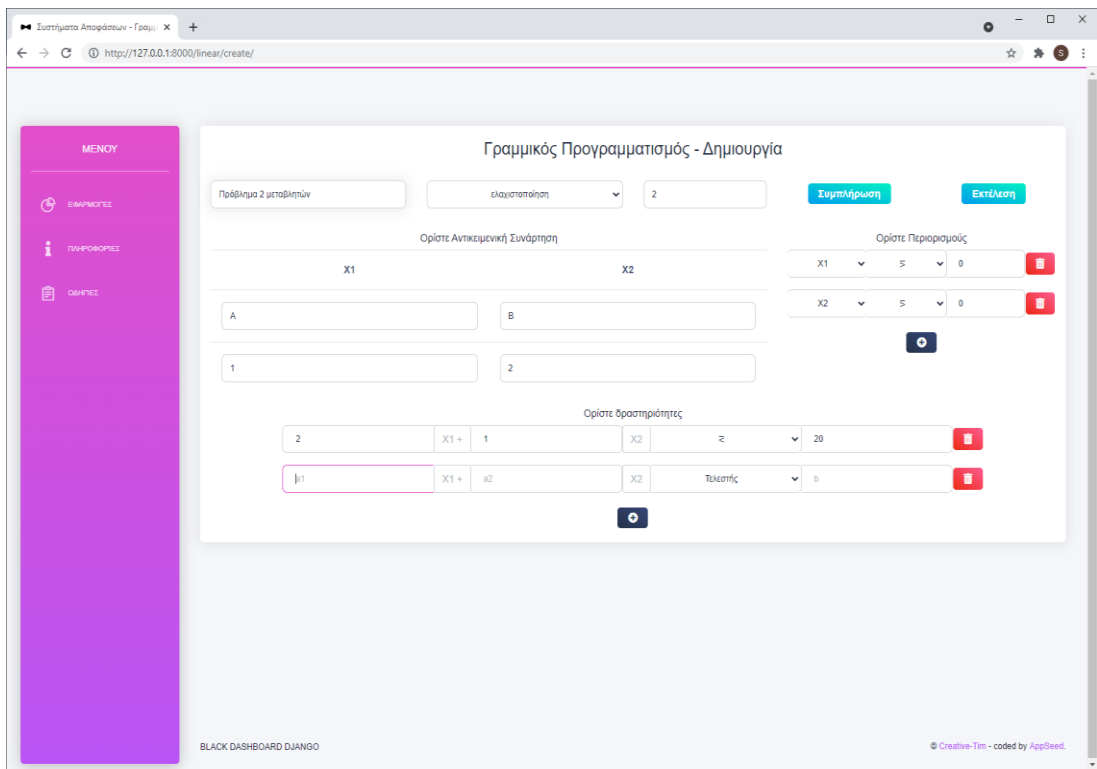
Εικόνα 42- Εφαρμογή γραμμικού προγραμματισμού: Συμπλήρωση δεδομένων αντικειμενικής συνάρτησης

Στο σύνολο εισόδων που εμφανίζεται στον χρήστη συμπεριλαμβάνονται τα δεδομένα της αντικειμενικής συνάρτησης, τα δεδομένα των περιορισμών και τα δεδομένα των δραστηριοτήτων. Παρακάτω παρουσιάζονται τα βήματα που πρέπει να ακολουθήσει ο χρήστης ώστε να συμπληρώσει καθένα από αυτά πεδία.

- Αντικειμενική συνάρτηση: Σε αυτό το σύνολο εισόδων ο χρήστης καλείται να συμπληρώσει τα δεδομένα της αντικειμενικής συνάρτησης του προβλήματός του. Αρχικά μπορεί να δώσει ένα όνομα στις μεταβλητές απόφασης συμπληρώνοντας το πεδίο *Περιγραφή*. Στη συνέχεια στο πεδίο *C* συμπληρώνει τον συντελεστή της κάθε μεταβλητής απόφασης στην αντικειμενική συνάρτηση. Το συγκεκριμένο πεδίο δέχεται μόνο αριθμητικές τιμές.
- Περιορισμοί: Σε αυτό το σύνολο εισόδων ο χρήστης καλείται να συμπληρώσει τα δεδομένα των περιορισμών του. Όπως φαίνεται στην *Εικόνα 43*, ο χρήστης στο πεδίο *Μεταβλητή* επιλέγει για ποια μεταβλητή απόφασης επιθυμεί να ορίσει περιορισμό. Στη συνέχεια στο πεδίο *Τελεστής* επιλέγει το τελεστή του περιορισμού. Τέλος, πρέπει να συμπληρώσει την αριθμητική τιμή του περιορισμού στο πεδίο *Τιμή*. Δίπλα από κάθε περιορισμό υπάρχει ένα κουμπί με σύμβολο ένα κάδο απορριμμάτων όπου επιλέγοντάς το ο χρήστης μπορεί να διαγράψει τον εν λόγω περιορισμό. Επιπλέον, στο τέλος των περιορισμών, υπάρχει ένα κουμπί με το σύμβολο της πρόσθεσης με το οποίο ο χρήστης μπορεί να προσθέσει έναν νέο περιορισμό.
- Δραστηριότητες: Σε αυτό το σύνολο εισόδων ο χρήστης καλείται να συμπληρώσει τα δεδομένα των δραστηριοτήτων του. Όπως φαίνεται στη *Εικόνα 44*, ο χρήστης στα πεδία *a* ορίζει τους συντελεστές των μεταβλητών απόφασης στην εκάστοτε δραστηριότητα. Οι τιμές των *a* μπορούν να είναι μόνο αριθμητικές. Στη συνέχεια, στο πεδίο *Τελεστής* επιλέγει τον τελεστή της δραστηριότητας. Τέλος, πρέπει να συμπληρώσει στο πεδίο *b* το δεξί μέρος της δραστηριότητας. Όπως και στις εισόδους των περιορισμών, έτσι κι εδώ με το κουμπί με το σύμβολο του κάδου απορριμμάτων ο χρήστης μπορεί να διαγράψει μια δραστηριότητα. Με το κουμπί με το σύμβολο της πρόσθεσης, στο τέλος των δραστηριοτήτων, ο χρήστης μπορεί να προσθέσει μια νέα δραστηριότητα.



Εικόνα 43- Εφαρμογή γραμμικού προγραμματισμού: Κατασκευή περιορισμών

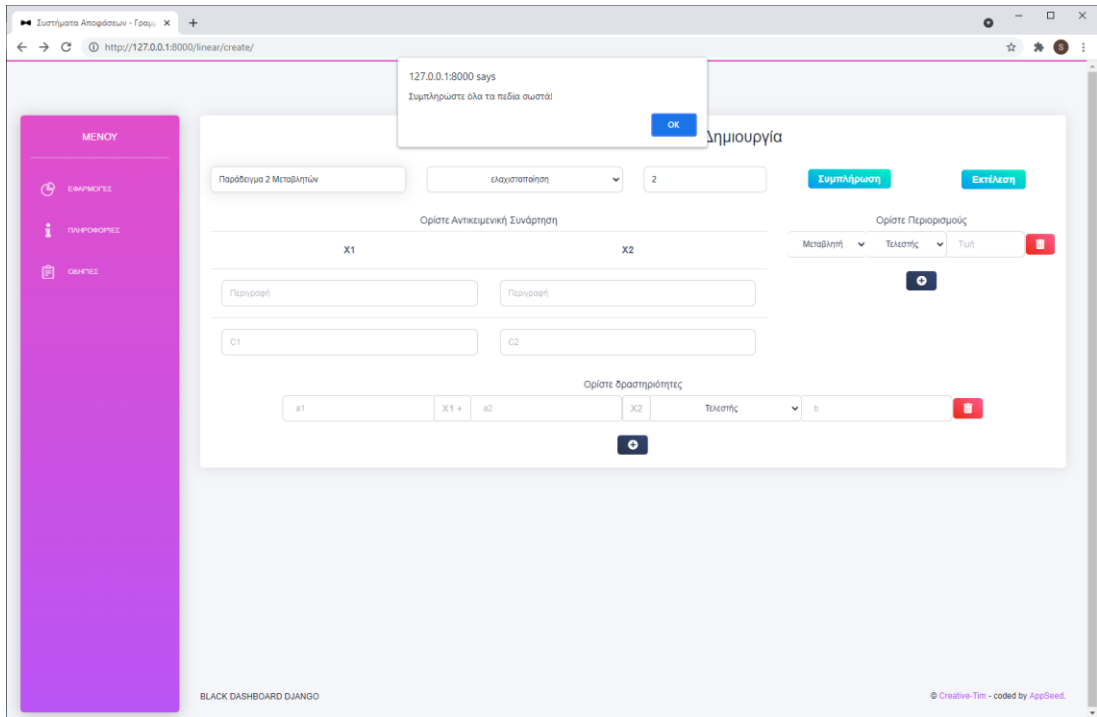


Εικόνα 44- Εφαρμογή γραμμικού προγραμματισμού: Κατασκευή δραστηριοτήτων

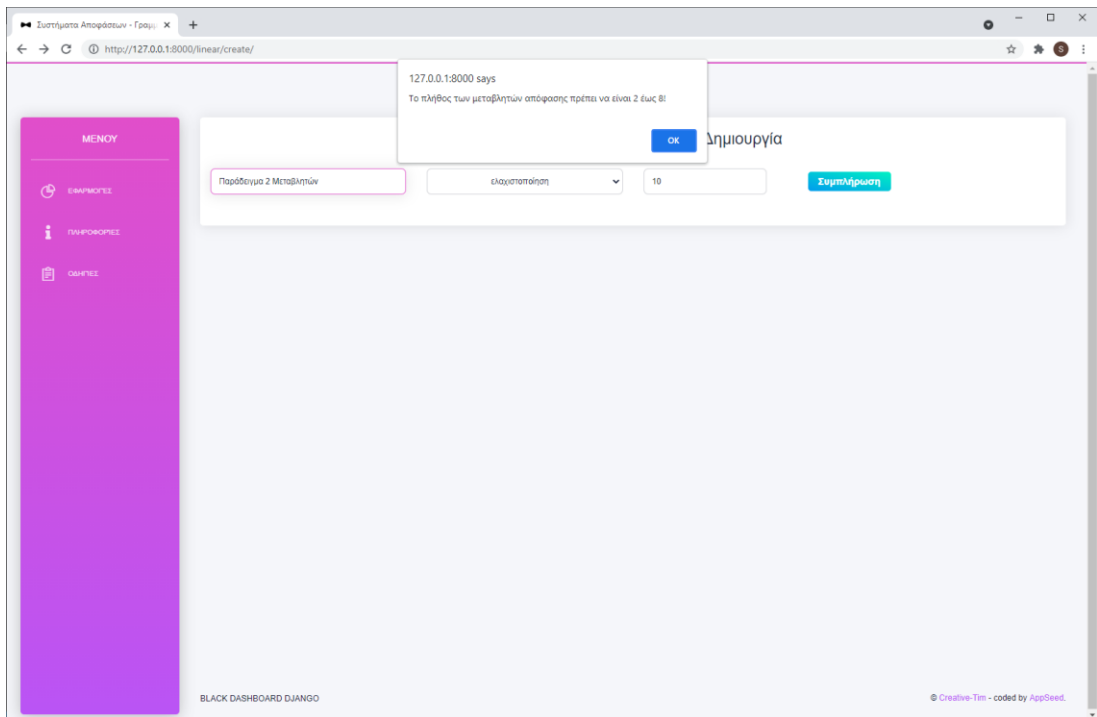
Εφόσον ο χρήστης έχει συμπληρώσει σωστά όλα του τα δεδομένα(Εικόνα 45), μπορεί να επιλέξει το κουμπί *Εκτέλεση* για να λύσει το πρόβλημά του. Επιλέγοντάς το θα ανακατευθυνθεί στη σελίδα αποτελεσμάτων της εφαρμογής γραμμικού προγραμματισμού. Τα αποτελέσματα του προβλήματος δύο μεταβλητών του οποίου τα δεδομένα έχουν συμπληρωθεί θα παρουσιαστούν στην ενότητα 6.3.5 *Οθόνη Αποτελεσμάτων*. Επιπλέον, μπορεί επιλέγοντας το κουμπί *Συμπλήρωση* να σβήσει όλα τα δεδομένα που έχει συμπληρώσει και να καταχωρήσει νέα.

Εικόνα 45- Παράδειγμα προβλήματος δύο μεταβλητών

Αν ο χρήστης δεν έχει συμπληρώσει όλα τα πεδία της αντικειμενικής συνάρτησης, των περιορισμών και των δραστηριοτήτων και επιλέξει το κουμπί *Εκτέλεση*, η εφαρμογή δεν θα προχωρήσει στη λύση του προβλήματος, αλλά θα ενημερώσει τον χρήστη με ειδοποίηση ότι τα πεδία δεν είναι σωστά συμπληρωμένα(Εικόνα 46). Επιπλέον ο χρήστης λαμβάνει ειδοποίηση για τη λάθος ή τη μη συμπλήρωση του πεδίου του πλήθους των μεταβλητών απόφασης. Όπως φαίνεται και σε αυτό, οι μεταβλητές πρέπει να είναι από δύο έως οκτώ. Σε άλλη περίπτωση λαμβάνει την ειδοποίηση που φαίνεται στη *Εικόνα 47*.



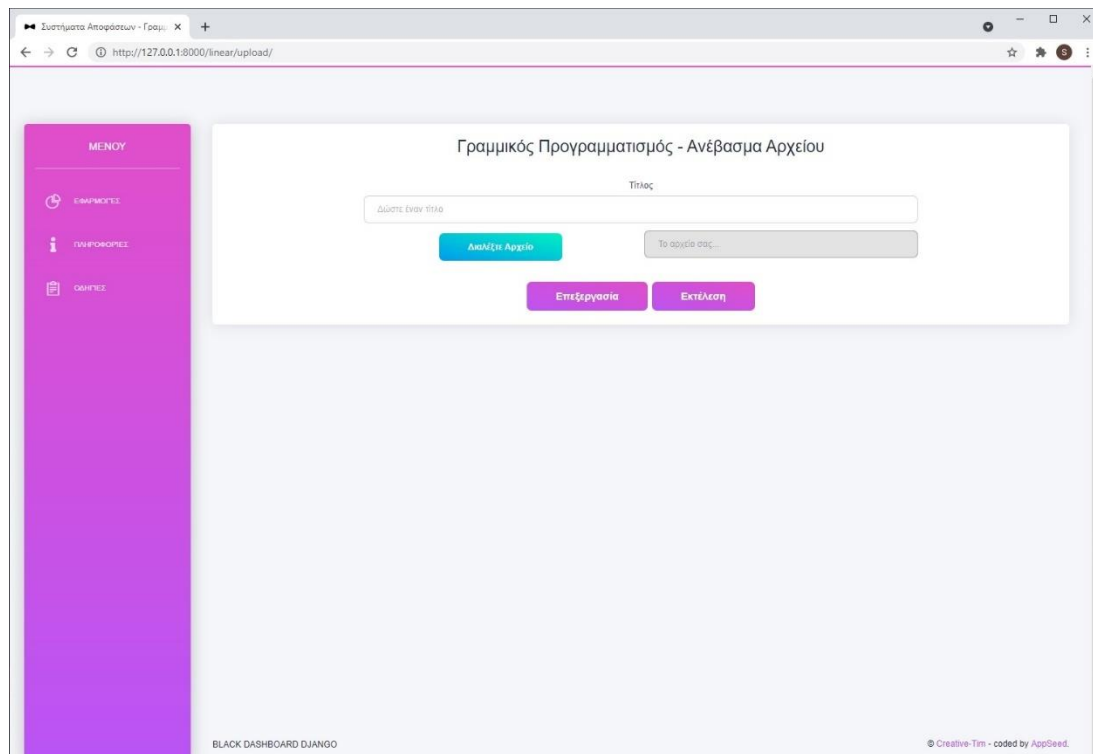
Εικόνα 46- Εφαρμογή γραμμικού προγραμματισμού: Λανθασμένη συμπλήρωση και εκτέλεση



Εικόνα 47- Εφαρμογή γραμμικού προγραμματισμού: Σφλαμα μη υποστηριζόμενου πλήθους μεταβλητών απόφασης

6.3.3 Οθόνη Ανεβάσματος Αρχείου

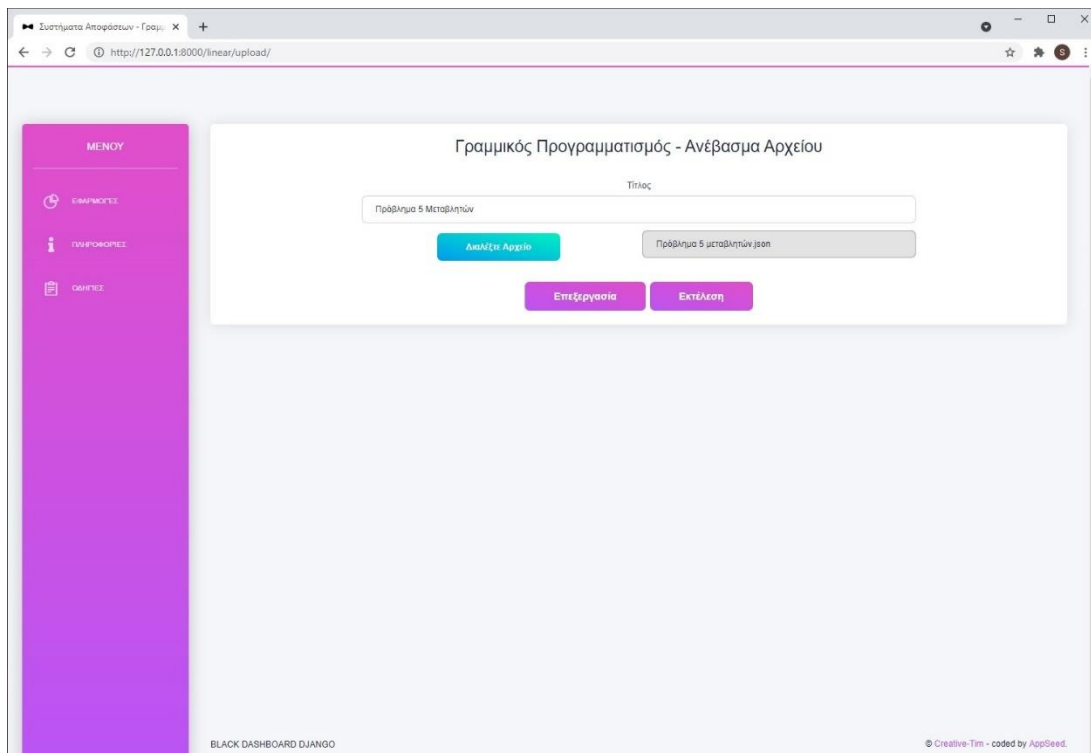
Ο χρήστης μπορεί να ανακατευθυνθεί σε αυτή τη σελίδα μόνο μέσω της αρχικής σελίδας της αρχικής οθόνης της εφαρμογής γραμμικού προγραμματισμού, εφόσον έχει επιλέξει να παρέχει τα δεδομένα του μέσω αρχείου JSON. Το στιγμιότυπο της οθόνης φαίνεται στην *Εικόνα 48*.



Εικόνα 48- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη ανεβάσματος αρχείου

Σε αυτή ο χρήστης, εφόσον το επιθυμεί, καλείται να συμπληρώσει τον τίτλο του προβλήματος γραμμικού προγραμματισμού που επιθυμεί να επιλύσει. Στη συνέχεια στο πεδίο *Διαλέξτε Αρχείο* μπορεί να επιλέξει ένα αρχείο τύπου JSON από τον τοπικό του υπολογιστή. Οι περιορισμοί του τύπου του αρχείου και του περιεχομένου του που αναφέρθηκαν στην ενότητα *6.2.3 Οθόνη Ανεβάσματος Αρχείου* ισχύουν κι εδώ.

Για να καλυφθεί η περίπτωση χρήσης της εφαρμογής σε πρόβλημα περισσότερων των δύο μεταβλητών επιλέγεται αρχείο με δεδομένα προβλήματος γραμμικού προγραμματισμού με πέντε μεταβλητές. Το συγκεκριμένο πρόβλημα αποτελεί παράδειγμα του μαθήματος Συστήματα Αποφάσεων της ΣΗΜΜΥ του ΕΜΠ [9]. Στην *Εικόνα 49* φαίνεται η φόρμα συμπληρωμένη.



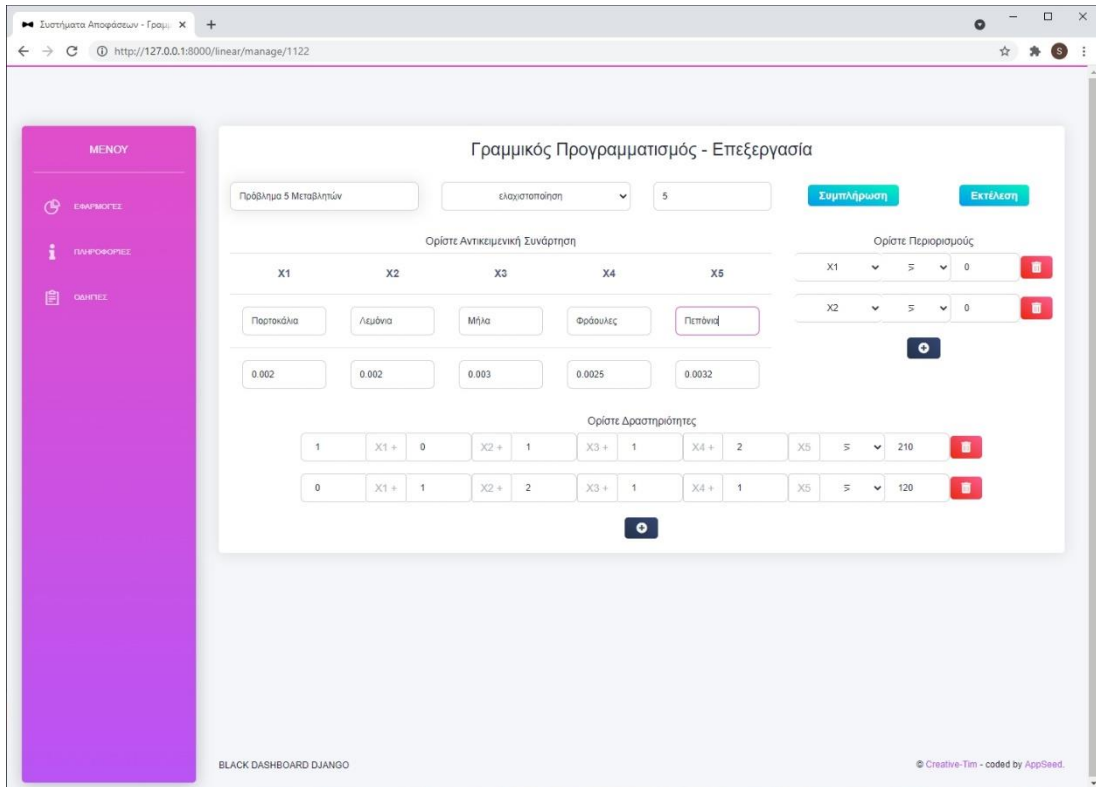
Εικόνα 49- Εφαρμογή γραμμικού προγραμματισμού: Φόρμα ανεβάσματος αρχείου συμπληρωμένη

Εφόσον η φόρμα έχει συμπληρωθεί σωστά, με τη επιλογή του κουμπιού *Επεξεργασία* ο χρήστης ανακατευθύνεται στη σελίδα επεξεργασία δεδομένων ενός προβλήματος γραμμικού προγραμματισμού. Σε άλλη περίπτωση που ο χρήστης επιθυμεί να επιλύσει άμεσα το πρόβλημά του, επιλέγει το κουμπί *Εκτέλεση*, όπου τον ανακατευθύνει στην οθόνη των αποτελεσμάτων.

6.3.4 Οθόνη Επεξεργασίας Δεδομένων Γραμμικού Προγραμματισμού

Σε αυτή την οθόνη ο χρήστης μπορεί να επεξεργαστεί τα δεδομένα εισόδου ενός προβλήματος γραμμικού προγραμματισμού. Υπάρχουν δύο τρόποι με τους οποίους μπορεί να ανακατευθυνθεί σε αυτή τη σελίδα. Είτε μέσω της σελίδας ανεβάσματος αρχείου, εφόσον επιθυμεί να επεξεργαστεί το αρχείο εισόδου, είτε μέσω της σελίδας αποτελεσμάτων, εφόσον επιθυμεί να επεξεργαστεί τα δεδομένα εισόδου ενός ήδη λυμένου προβλήματος.

Σε αυτό το σημείο θα καλυφθεί επίσης η περίπτωση χρήσης ενός προβλήματος πέντε μεταβλητών. Τα δεδομένα εισόδου λήφθηκαν από τις ασκήσεις του μαθήματος Συστήματα Αποφάσεων της ΣΗΜΜΥ του ΕΜΠ [9]. Στην *Εικόνα 50* φαίνονται τα δεδομένα εισόδου του παραπάνω παραδείγματος συμπληρωμένα.



Εικόνα 50- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη επεξεργασίας

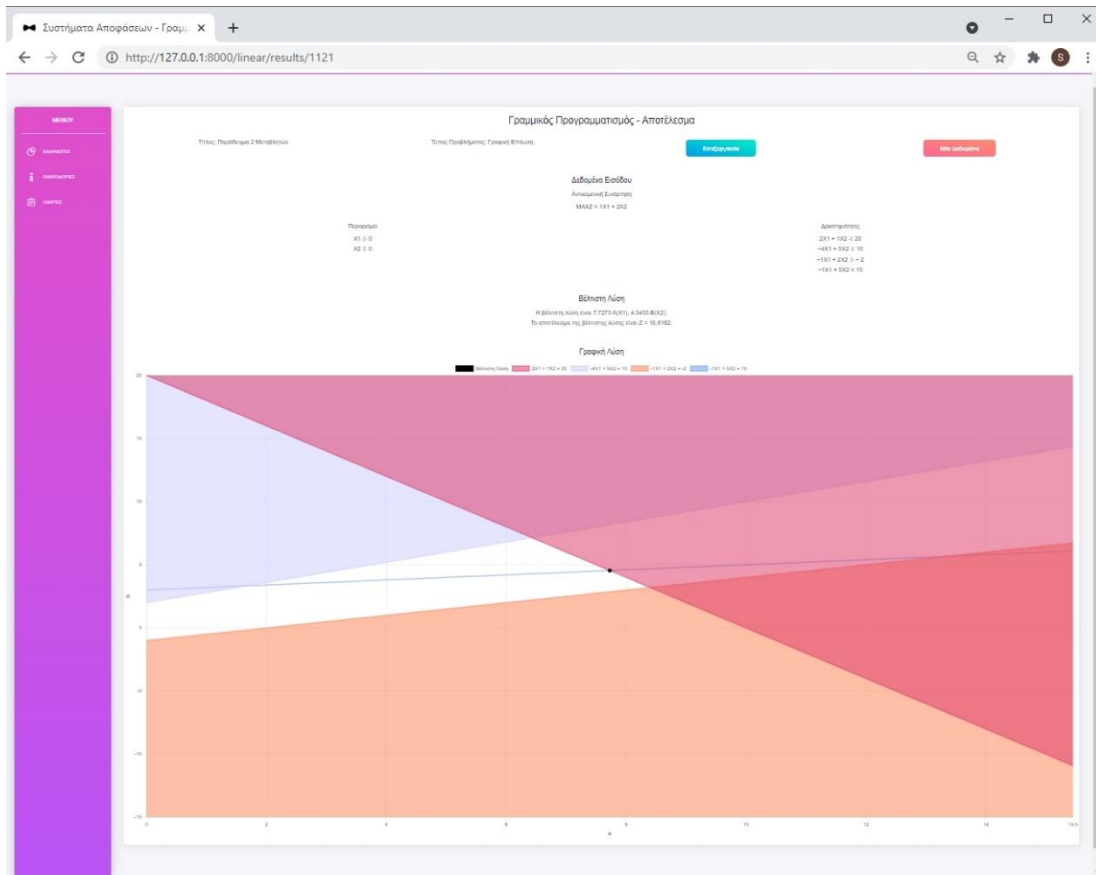
Σε αυτή την οθόνη, λοιπόν, ο χρήστης θα βρει όλα τα δεδομένα τα οποία έχει χρησιμοποιήσει ως είσοδο. Όπως και στη οθόνη δημιουργίας, έτσι κι εδώ ο χρήστης μπορεί να αλλάξει τον τίτλο του προβλήματός του, το κριτήριο επίλυσης αλλά και να αλλάξει τον αριθμό των μεταβλητών απόφασης και να τα εφαρμόσει χρησιμοποιώντας το κουμπί *Συμπλήρωση*. Επιπλέον, του δίνεται η δυνατότητα να επεξεργαστεί την αντικειμενική συνάρτηση, τους περιορισμούς, αλλά και τις δραστηριότητες. Οι λειτουργίες επεξεργασίας που ισχύουν στην οθόνη δημιουργίας ισχύουν κι εδώ.

6.3.5 Οθόνη Αποτελεσμάτων

Εδώ ο χρήστης μπορεί να δει τα αποτελέσματα της λύσης ενός προβλήματος γραμμικού προγραμματισμού. Αν το πρόβλημα που έχει λυθεί είναι πρόβλημα δύο μεταβλητών, η λύση αναπαρίσταται γραφικά. Αν το πρόβλημα που έχει λυθεί είναι πρόβλημα περισσότερων των δύο μεταβλητών, τότε παρουσιάζονται οι πίνακες Simplex που κατασκεύασε ο αλγόριθμος κατά τη διαδικασία επίλυσης του προβλήματος. Για αυτό το λόγο η παρουσίαση της οθόνης αποτελεσμάτων θα χωριστεί σε δύο τμήματα, ανάλογα με το πλήθος των μεταβλητών απόφασης στα δεδομένα εισόδου.

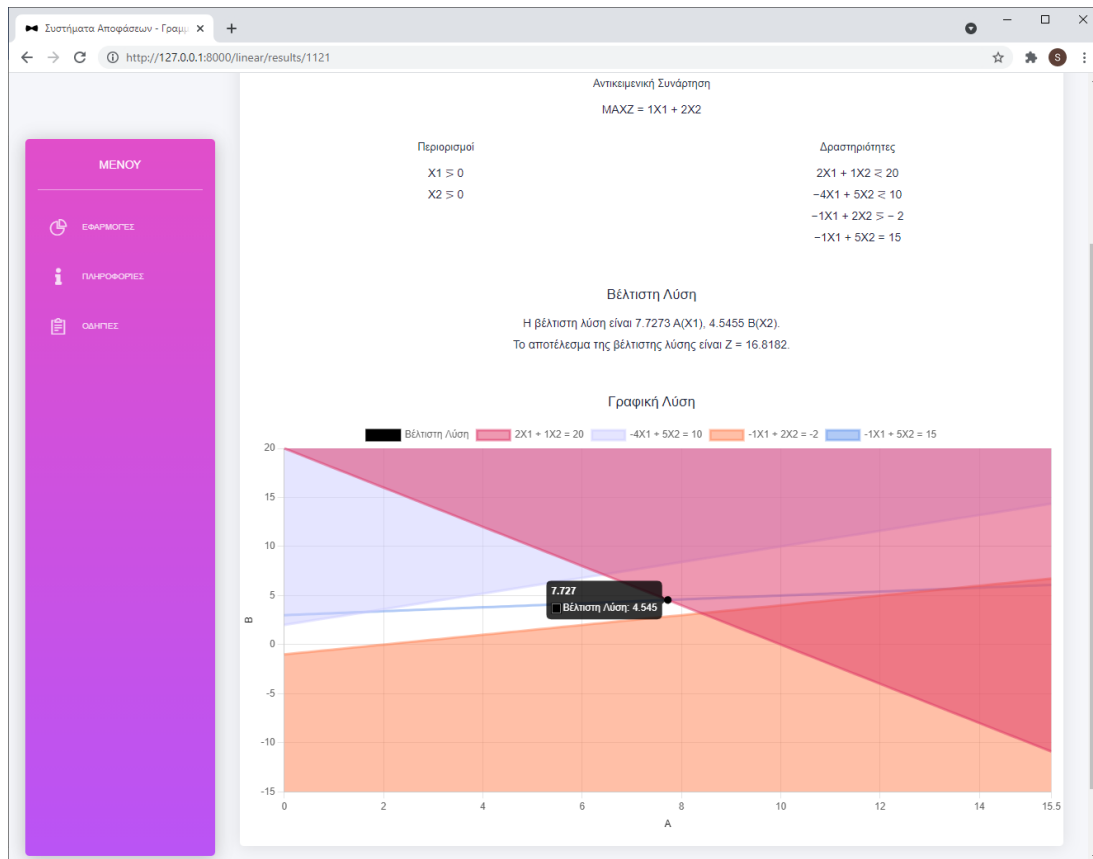
Πρόβλημα δύο μεταβλητών

Σε αυτό το σημείο θα παρουσιαστούν τα αποτελέσματα του παραδείγματος της *Εικόνας 45*. Στην *Εικόνα 51* φαίνεται η επίλυση αυτού του προβλήματος.



Εικόνα 51- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη αποτελεσμάτων – γραφική επίλυση

Στο πάνω μέρος της οθόνης ο χρήστης μπορεί να δει τον τίτλο του προβλήματός του καθώς και τον τύπο του. Έπειτα ακολουθούν τα δεδομένα εισόδου που όρισε ο χρήστης. Πρώτα του παρουσιάζεται η αντικειμενική συνάρτηση που επέλεξε και ακολουθούν οι περιορισμοί και οι δραστηριότητες. Στη συνέχεια εξηγείται το σημείο της βέλτιστης λύσης καθώς και η τιμή της αντικειμενικής συνάρτησης για αυτή. Τέλος, στον χρήστη παρουσιάζεται το γράφημα που περιλαμβάνει χαραγμένες τις δραστηριότητες του προβλήματος και επισημασμένη τη λύση του προβλήματος. Επιπλέον, ο χρήστης με την αιώρηση του δρομέα στις γραφικές παραστάσεις της λύσεις μπορεί να δει τις τιμές των μεταβλητών απόφασης σε κάθε σημείο του γραφήματος. Παράδειγμα αυτού φαίνεται στη *Εικόνα 52*.

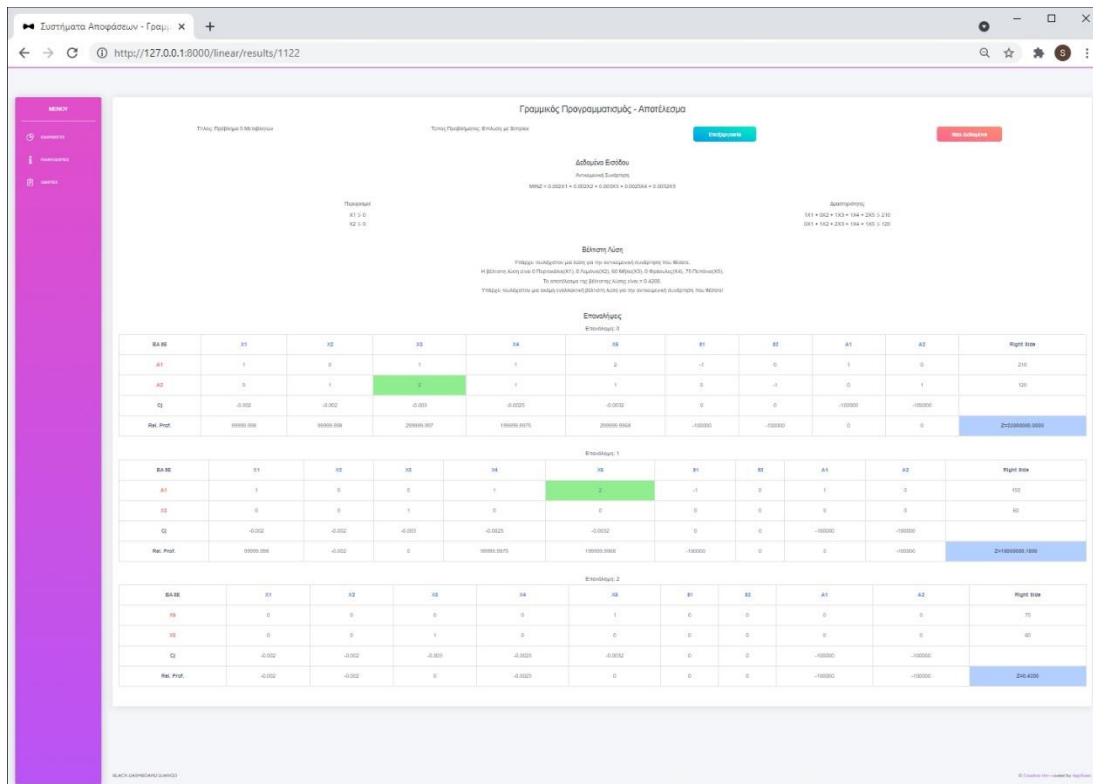


Εικόνα 52- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη αποτελεσμάτων –απεικόνιση σημείων

Από αυτή την οθόνη ο χρήστης μπορεί να πραγματοποιήσει δύο ενέργειες. Είτε να επεξεργαστεί τα δεδομένα εισόδου επιλέγοντας το κουμπί *Επεξεργασία* που φαίνεται στην 51. Αυτή η ενέργεια θα τον ανακατευθύνει στην οθόνη επεξεργασίας δεδομένων. Είτε να παρέχει νέα δεδομένα εισόδου, επιλέγοντας το κουμπί *Νέα Δεδομένα* της Εικόνας 51, εφόσον επιθυμεί να επιλύσει ένα νέο πρόβλημα γραμμικού προγραμματισμού. Αυτή η ενέργεια θα τον ανακατευθύνει στην αρχική οθόνη της εφαρμογής γραμμικού προγραμματισμού ώστε να επιλέξει τη μέθοδο με την οποία επιθυμεί να παρέχει τα δεδομένα του.

Πρόβλημα με περισσότερες από δύο μεταβλητές

Σε αυτό το σημείο θα παρουσιαστούν τα αποτελέσματα του παραδείγματος της Εικόνας 50. Στην Εικόνα 53 φαίνεται η επίλυση αυτού του προβλήματος.



Εικόνα 53- Εφαρμογή γραμμικού προγραμματισμού: Οθόνη αποτελεσμάτων –μέθοδος simplex

Όπως στο προηγούμενο παράδειγμα, έτσι κι εδώ στο πάνω μέρος της οθόνης ο χρήστης μπορεί να δει τον τίτλο του προβλήματός του και τον τύπο του. Στη συνέχεια παρουσιάζονται τα δεδομένα εισόδου που όρισε ο χρήστης. Αρχικά, η αντικειμενική του συνάρτηση και στη συνέχεια οι περιορισμοί και οι δραστηριότητες. Ακολουθεί η επεξήγηση της βέλτιστης λύσης, αν αυτή υπάρχει και τη διαμόρφωση της τιμής της αντικειμενικής συνάρτησης με τις τιμές της βέλτιστης λύσης. Τέλος, παρουσιάζονται στον χρήστη οι πίνακες Simplex που κατασκευάστηκαν κατά τη διαδικασία επίλυσης του προβλήματος.

Στους πίνακες Simplex με πράσινο χρώμα επισημαίνονται τα κελιά τα οποία επιλέχθηκαν ως στοιχεία περιστροφής (pivot elements) και με πράσινο η τιμή που παίρνει η αντικειμενική συνάρτηση σε κάθε επανάληψη.

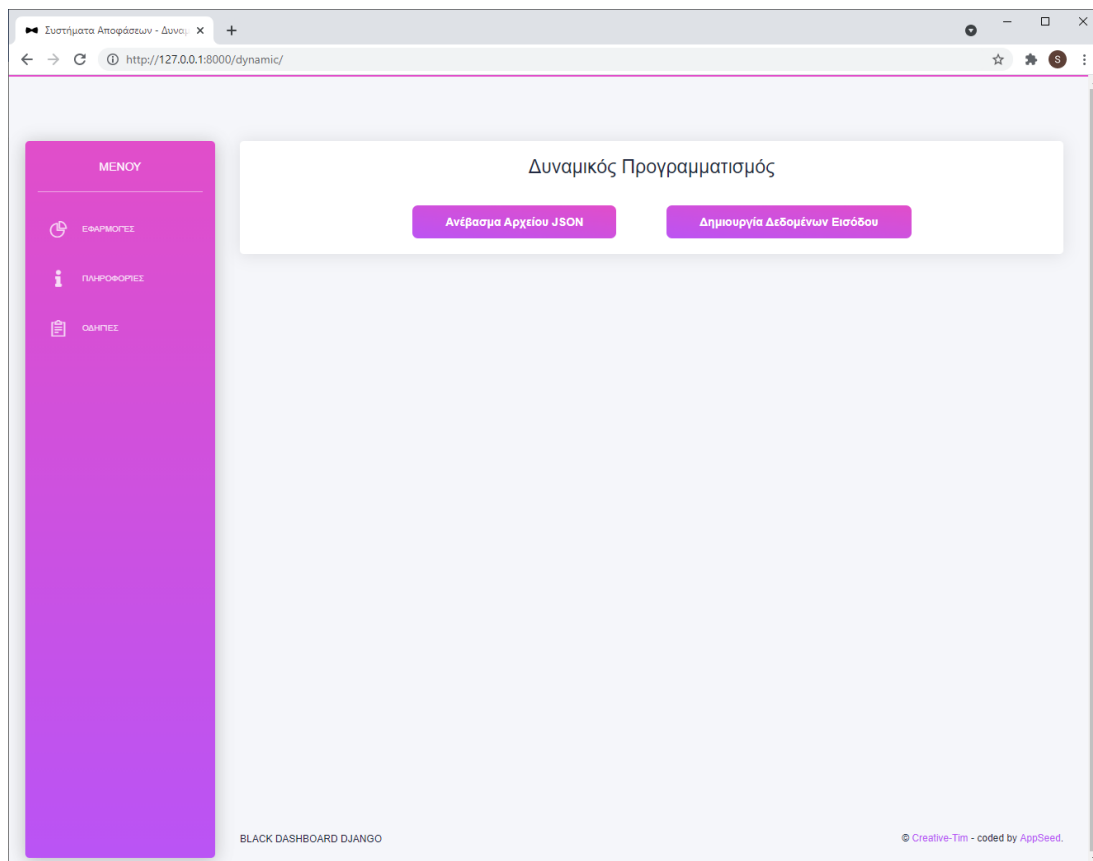
Από αυτή την οθόνη ο χρήστης μπορεί να πραγματοποιήσει δύο ενέργειες. Είτε να επεξεργαστεί τα δεδομένα εισόδου επιλέγοντας το κουμπί *Επεξεργασία* που φαίνεται στην *Εικόνα 53*. Αυτή η ενέργεια θα τον ανακατευθύνει στην οθόνη επεξεργασίας δεδομένων. Είτε να παρέχει νέα δεδομένα εισόδου, επιλέγοντας το κουμπί *Νέα Δεδομένα* της *Εικόνας 53*, εφόσον επιθυμεί να επιλύσει ένα νέο πρόβλημα γραμμικού προγραμματισμού. Αυτή η ενέργεια θα τον ανακατευθύνει στην αρχική οθόνη της εφαρμογής γραμμικού προγραμματισμού ώστε να επιλέξει τη μέθοδο με την οποία επιθυμεί να παρέχει τα δεδομένα του.

6.4 Εφαρμογή Δυναμικού Προγραμματισμού

Σε αυτό το σημείο θα περιγραφούν τα σενάρια χρήσης της εφαρμογής Δυναμικού Προγραμματισμού. Δηλαδή θα παρουσιαστούν οι δυνατότητες που παρέχει η εφαρμογή στον χρήστη και πώς αυτή ανταποκρίνεται σε κάθε κίνηση του χρήστη μέσα σε αυτή.

6.4.1 Αρχική Οθόνη

Εφόσον ο χρήστης έχει επιλέξει ότι θέλει να κάνει χρήση της εφαρμογής δυναμικού προγραμματισμού, ανακατευθύνεται στην αρχική οθόνη της εφαρμογής δυναμικού προγραμματισμού. Μια άλλη περίπτωση με την οποία ο χρήστης μπορεί να βρεθεί σε αυτή την οθόνη είναι μέσω της οθόνης αποτελεσμάτων, στην περίπτωση που επιθυμεί να παρέχει δεδομένα για ένα νέο πρόβλημα δυναμικού προγραμματισμού. Το στιγμιότυπο της αρχικής οθόνης φαίνεται στην *Εικόνα 54*.

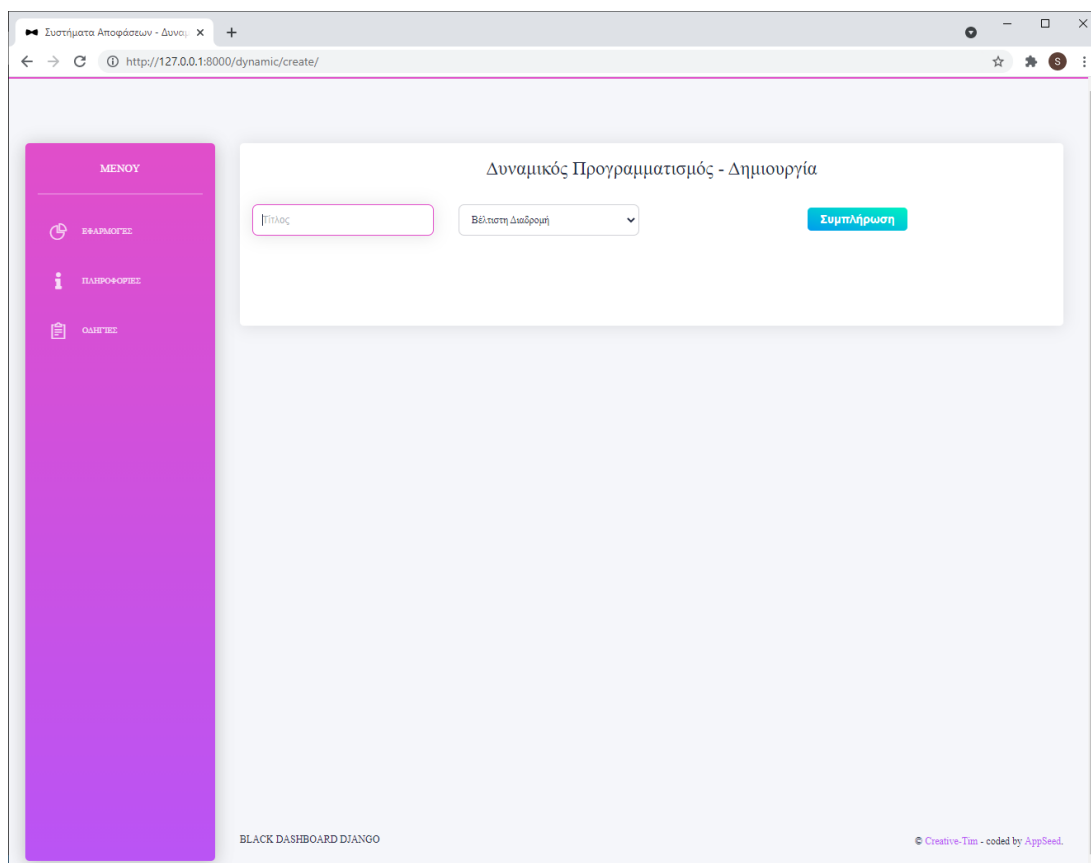


Εικόνα 54- Εφαρμογή δυναμικού προγραμματισμού: Αρχική οθόνη

Σε αυτή την οθόνη ο χρήστης καλείται να επιλέξει τον τρόπο με τον οποίο επιθυμεί να παρέχει τα δεδομένα του στην εφαρμογή. Επιλέγοντας το κουμπί *Ανέβασμα Αρχείου JSON* ο χρήστης θα ανακατευθυνθεί στη σελίδα ανεβάσματος αρχείου, όπου θα κληθεί να παρέχει τα δεδομένα του μέσω ενός αρχείου JSON. Επιλέγοντας το κουμπί *Δημιουργία Δεδομένων Εισόδου* ο χρήστης θα ανακατευθυνθεί στη σελίδα δημιουργίας ενός προβλήματος δυναμικού προγραμματισμού, όπου θα μπορεί να κατασκευάσει τα δεδομένα του πάνω στην εφαρμογή.

6.4.2 Οθόνη Δημιουργίας Προβλήματος Δυναμικού Προγραμματισμού

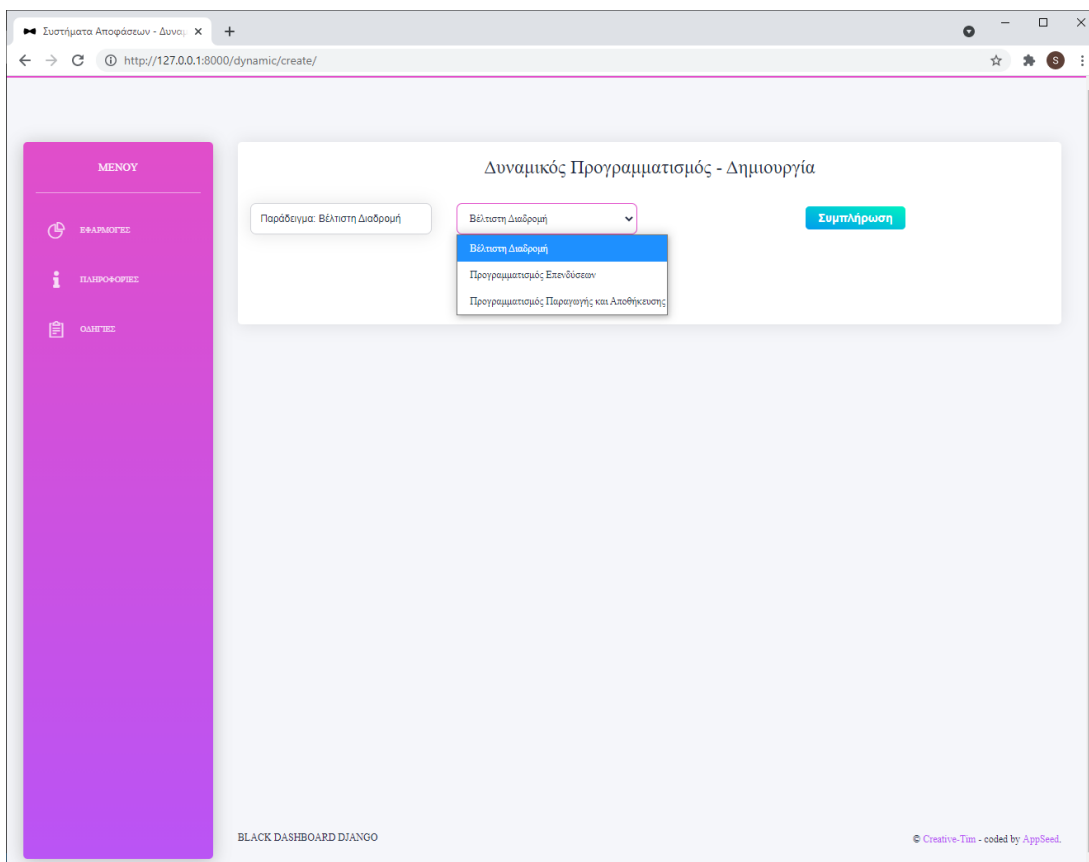
Εδώ ο χρήστης μπορεί να κατασκευάσει τα δεδομένα ενός προβλήματος δυναμικού προγραμματισμού. Μπορεί να βρεθεί σε αυτή τη σελίδα μόνο μέσω της αρχικής οθόνης της εφαρμογής δυναμικού προγραμματισμού. Το στιγμιότυπο της οθόνης, κατά την επίσκεψη του χρήστη, φαίνεται στην *Εικόνα 55*.



Εικόνα 55- Εφαρμογή δυναμικού προγραμματισμού: Δημιουργία προβλήματος

Σε αυτή, ο χρήστης καλείται αρχικά να συμπληρώσει τον τίτλο του προβλήματός του, εφόσον το επιθυμεί, στο πεδίο *Τίτλος*. Στη συνέχεια, μπορεί να επιλέξει το είδος

προβλήματος του οποίου τα δεδομένα επιθυμεί να συμπληρώσει. Οι δυνατές επιλογές του φαίνονται στην *Εικόνα 56*. Αφού έχει συμπληρώσει τα παραπάνω στοιχεία, επιλέγοντας το κουμπί *Συμπλήρωση*, ανάλογα με τον τύπο προβλήματος που έχει επιλέξει θα του παρουσιαστούν διαφορετικοί τύποι εισόδων που μπορεί να συμπληρώσει τα δεδομένα του. Στη συνέχεια θα παρουσιαστούν τα σενάρια χρήσης για κάθε τύπο προβλήματος.



Εικόνα 56- Εφαρμογή δυναμικού προγραμματισμού: Επιλογή τύπου προβλήματος

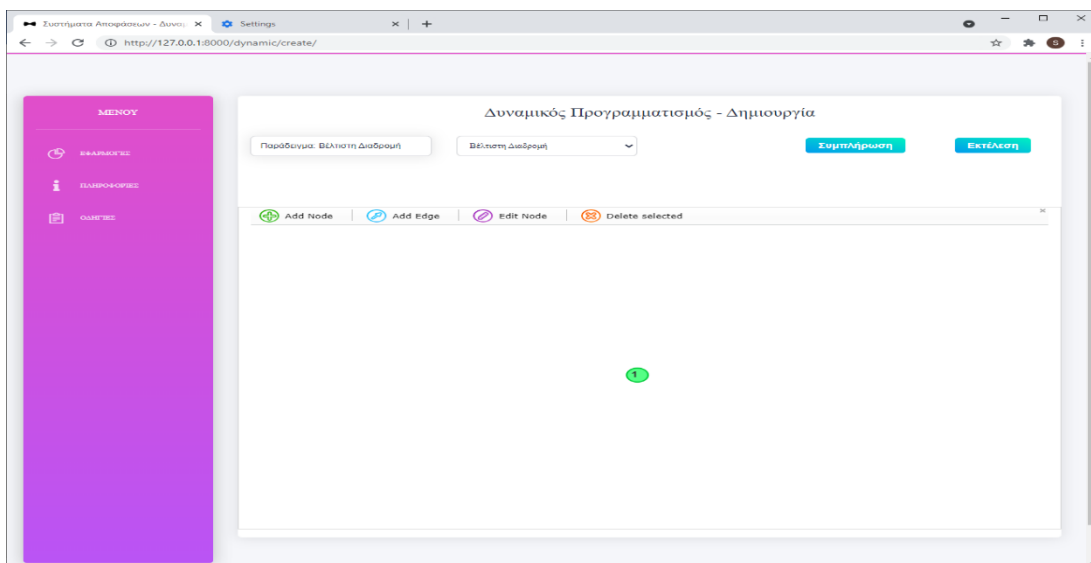
Τύπος Προβλήματος: Βέλτιστη Διαδρομή

Εφόσον ο χρήστης επιλέξει να συμπληρώσει τα δεδομένα για ένα πρόβλημα τύπου βέλτιστης διαδρομής, θα εμφανιστεί σε αυτόν το εργαλείο κατασκευής γράφου. Αυτό φαίνεται στην *Εικόνα 57*. Το εργαλείο έχει ήδη καταχωρημένο έναν κόμβο εκκίνησης. Οι δυνατότητες που παρέχονται στον χρήστη μέσω αυτού του εργαλείου είναι οι εξής:

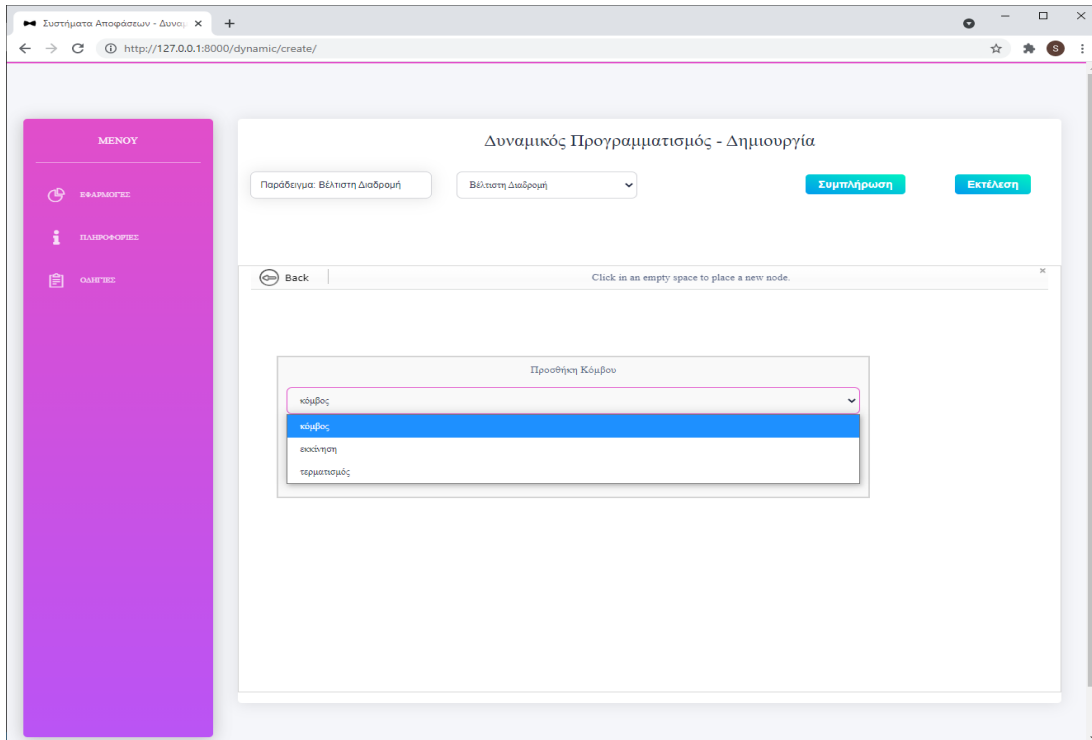
- Προσθήκη κόμβου (Add Node): Όπως φαίνεται στη *Εικόνα 58*, όταν ο χρήστης επιλέξει το κουμπί *Add Node* και κάνει κλικ σε έναν κενό χώρο μέσα στο εργαλείο θα του εμφανιστεί ένα αναδυόμενο παράθυρο όπου πρέπει να επιλέξει το είδος του κόμβου που επιθυμεί να προσθέσει στον γράφο. Εφόσον θέλει να προχωρήσει σε αυτή την ενέργεια, αφού έχει επιλέξει το είδος του

κόμβου, επιλέγει το κουμπί *Αποθήκευση* για να ολοκληρωθεί η ενέργεια. Όταν ολοκληρωθεί η ενέργεια στο γράφο θα εμφανιστεί στο εργαλείο ο κόμβος και εντός του κύκλου του παρουσιάζεται το αναγνωριστικό του. Αν θέλει να ακυρωθεί η ενέργεια επιλέγει το κουμπί *Ακύρωση*.

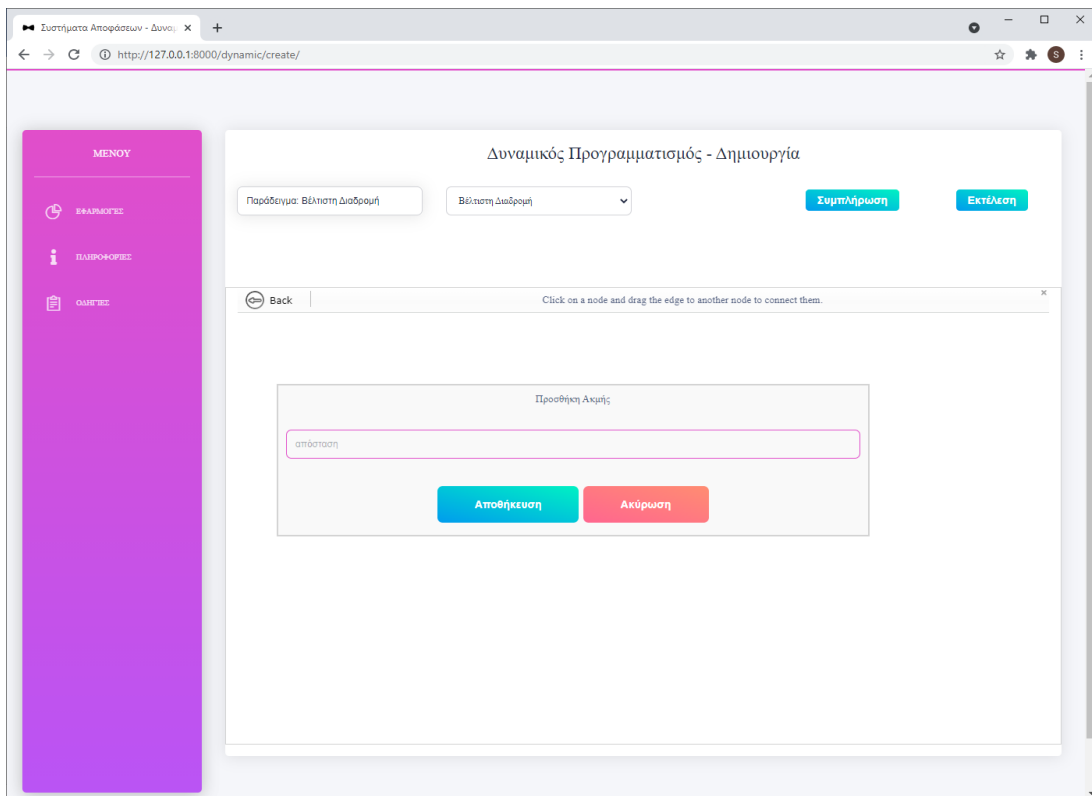
- Προσθήκη ακμής (Add Edge): Όπως φαίνεται στην *Εικόνα 59*, όταν ο χρήστης επιλέξει το κουμπί *Add Edge* και σχεδιάσει μια κατευθυνόμενη ακμή μεταξύ των κόμβων που επιθυμεί να ενώσει θα του εμφανιστεί ένα αναδυόμενο παράθυρο όπου καλείται να συμπληρώσει την αριθμητική τιμή της απόστασης των δύο κόμβων. Εφόσον θέλει να προχωρήσει σε αυτή την ενέργεια επιλέγει το κουμπί *Αποθήκευση*. Αν θέλει να την ακυρώσει, επιλέγει το κουμπί *Ακύρωση*.
- Επεξεργασία ακμής (Edit Edge): Όταν ο χρήστης έχει επιλεγμένη μια ακμή στον γράφο του, έχει τη δυνατότητα να επεξεργαστεί την απόσταση που έχει δηλωθεί για την ακμή αυτή. Επιλέγοντας, λοιπόν, το κουμπί *Edit Edge* εμφανίζεται ένα αναδυόμενο παράθυρο όπως στην *Εικόνα 59*, όπου εμφανίζεται στον χρήστη η ήδη δηλωμένη απόσταση της ακμής. Εφόσον επιθυμεί να την αλλάξει, πληκτρολογεί μια νέα αριθμητική τιμή και επιλέγει το κουμπί *Αποθήκευση* ώστε να καταχωρηθεί η νέα τιμή. Αν επιθυμεί να ακυρώσει την ενέργεια αυτή, επιλέγει το κουμπί *Ακύρωση*.
- Διαγραφή Επιλεγμένου (Delete Selected): Όταν ο χρήστης έχει επιλεγμένο έναν κόμβο ή μια ακμή, έχει τη δυνατότητα να διαγράψει το επιλεγμένο στοιχείο επιλέγοντας το κουμπί *Delete Selected*. Αν το επιλεγμένο στοιχείο είναι κόμβος και επιλεγεί η διαγραφή του, διαγράφονται και όλες οι ακμές που εκκινούν ή καταλήγουν σε αυτόν τον κόμβο, εφόσον υπάρχουν.



Εικόνα 57- Εφαρμογή δυναμικού προγραμματισμού: Δημιουργία προβλήματος βέλτιστης διαδρομής

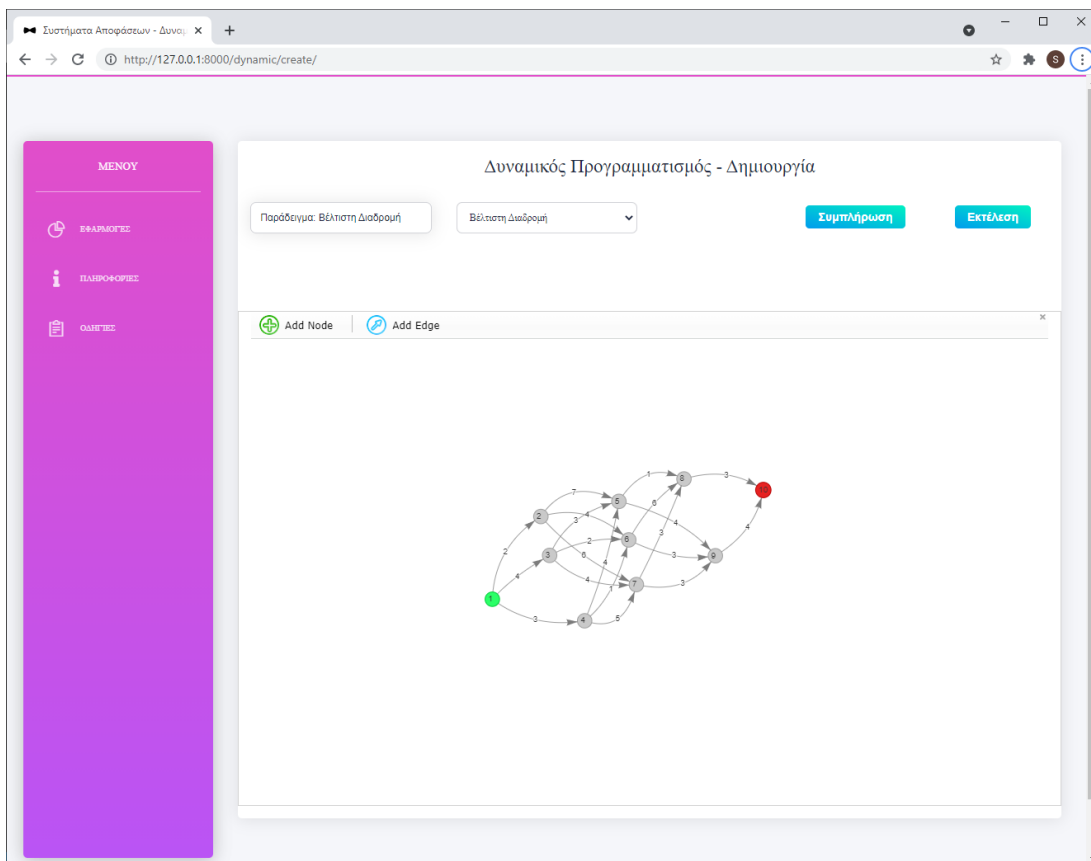


Εικόνα 58- Εφαρμογή δυναμικού προγραμματισμού: Προσθήκη κόμβου



Εικόνα 59- Εφαρμογή δυναμικού προγραμματισμού: Προσθήκη ακμής

Αφού χρήστης κατασκευάσει τον γράφο που επιθυμεί μπορεί στη συνέχεια να προχωρήσει στην επίλυση του προβλήματός του επιλέγοντας το κουμπί *Εκτέλεση*. Μόνη προϋπόθεση για να προχωρήσει η ενέργεια αυτή είναι όλες οι ακμές του γράφου του να έχουν αριθμητική τιμή της απόστασης των κόμβων που συνδέουν. Ένας ολοκληρωμένος γράφος ενός προβλήματος δυναμικού προγραμματισμού φαίνεται στην *Εικόνα 60*. Τα δεδομένα εισόδου του προβλήματος αυτού είναι από το μάθημα Συστήματα Αποφάσεων της ΣΗΜΜΥ του ΕΜΠ [10]. Τα αποτελέσματα της εκτέλεσης του παραπάνω παραδείγματος θα παρουσιαστούν στην ενότητα 6.4.5, όπου θα επεξηγηθεί η οθόνη των αποτελεσμάτων.



Εικόνα 60- - Εφαρμογή δυναμικού προγραμματισμού: Παράδειγμα βέλτιστης διαδρομής

Τύπος Προβλήματος: Προγραμματισμός Επενδύσεων

Εφόσον ο χρήστης επιλέξει να συμπληρώσει τα δεδομένα για ένα πρόβλημα τύπου προγραμματισμού επενδύσεων, θα του εμφανιστούν οι επιλογές που πρέπει να ορίσει για το πρόβλημά του. Οι επιλογές αυτές φαίνονται στην *Εικόνα 61*. Εκεί πρέπει να συμπληρώσει την ονομασία που θέλει να δώσει στις οικονομικές περιόδους του προβλήματος και το πλήθος αυτών, στα πεδία *Τύπος Οικονομικών Περιόδων* και *Αριθμός Οικονομικών Περιόδων* αντίστοιχα. Στη συνέχεια, πρέπει να συμπληρώσει τον τύπο των οικονομικών μονάδων του προβλήματός του και το πλήθος αυτών, στα

πεδία *Τύπος Οικονομικών Μονάδων* και *Αριθμός Οικονομικών Μονάδων*. Αν έχει συμπληρώσει σωστά τις τιμές στα παραπάνω πεδία, επιλέγοντας το κουμπί *Ορισμός Επιλογών* θα του εμφανιστεί ο πίνακας με τα κέρδη που επιφέρει η επένδυση κάθε συνδυασμού οικονομικών μονάδων ανά οικονομική περίοδο. Ο πίνακας αυτός φαίνεται στην *Εικόνα 62*. Αν κάποια τιμή των επιλογών δεν συμπληρωθεί σωστά, κατά την επιλογή του κουμπιού *Ορισμός Επιλογών* θα λάβει την ειδοποίηση που φαίνεται στην *Εικόνα 63*.

Δυναμικός Προγραμματισμός - Δημιουργία

Παράδειγμα: Προγραμματισμός Επενδύσει Προγραμματισμός Επενδύσεων Συμπλήρωση

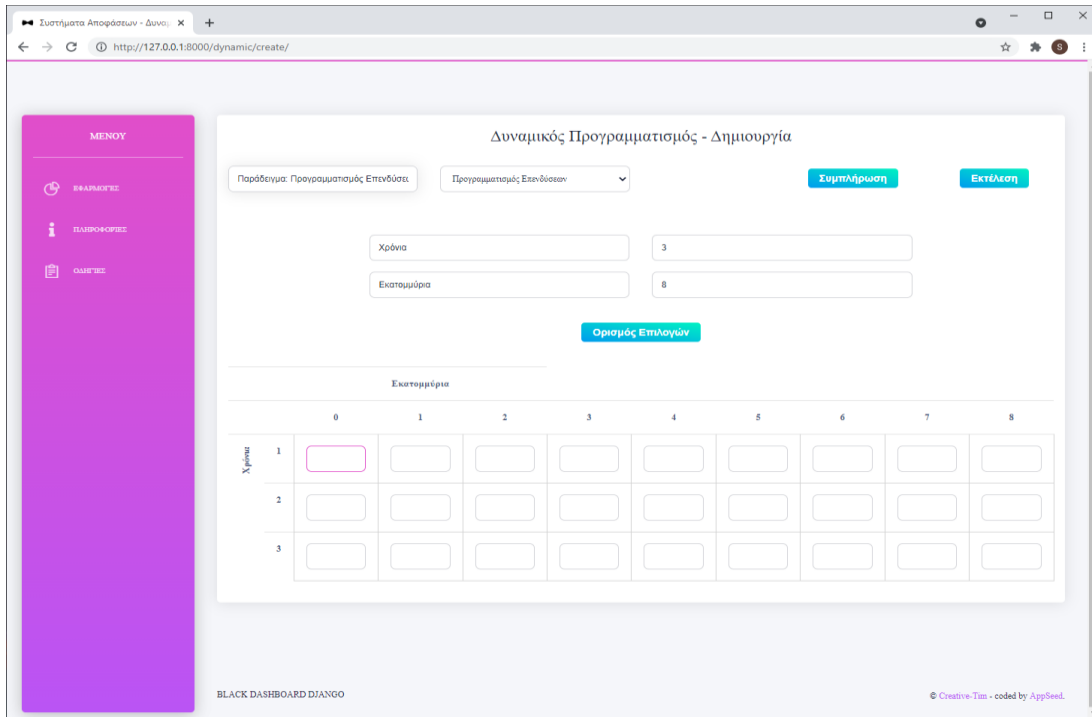
Τύπος Οικονομικών Περιόδων Αριθμός Οικονομικών Περιόδων

Τύπος Οικονομικών Μονάδων Αριθμός Οικονομικών Μονάδων

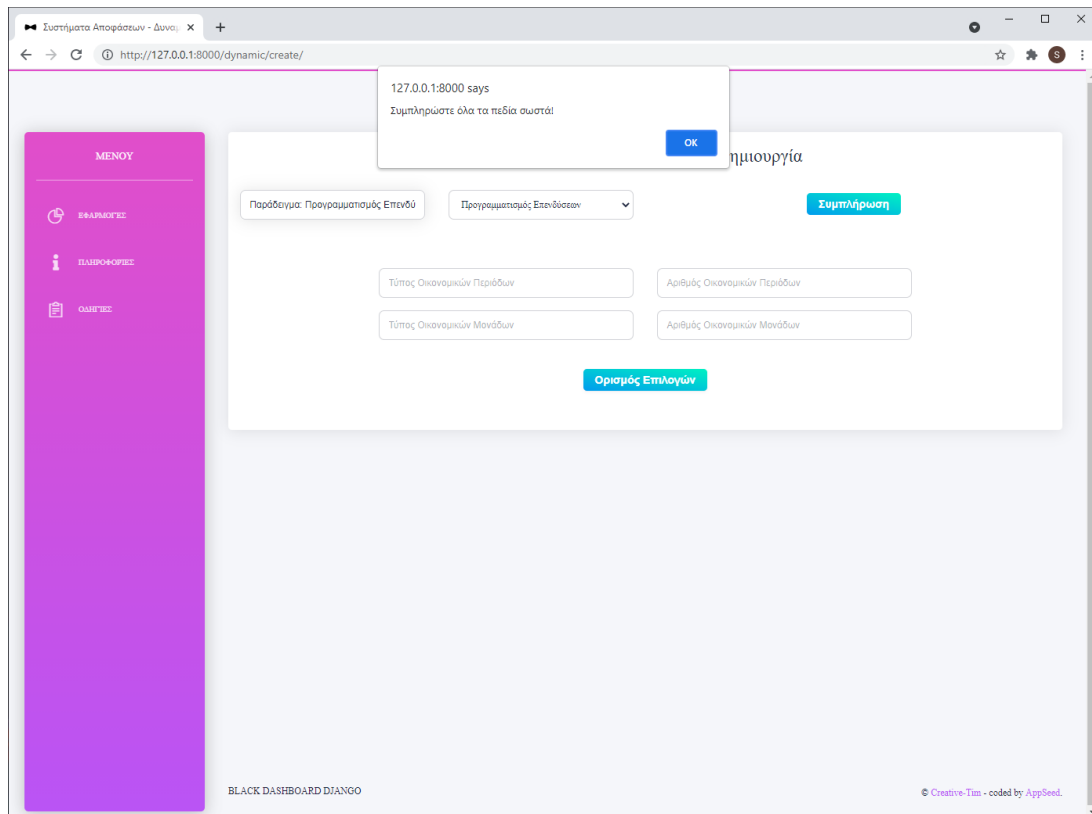
Ορισμός Επιλογών

BLACK DASHBOARD DJANGO © Creative-Tim - coded by AppSeed.

Εικόνα 61- Εφαρμογή δυναμικού προγραμματισμού: Δημιουργία προβλήματος προγραμματισμού επενδύσεων



Εικόνα 62- Συμπλήρωση δεδομένων προβλήματος προγραμματισμού επενδύσεων



Εικόνα 63- Εφαρμογή δυναμικού προγραμματισμού: Σφάλμα λανθασμένης συμπλήρωσης

Αφού ο χρήστης έχει συμπληρώσει τα δεδομένα του όπως φαίνεται στο παράδειγμα της *Εικόνας 63* μπορεί να προχωρήσει στην επίλυση του προβλήματος επιλέγοντας το κουμπί *Εκτέλεση*. Αν τα δεδομένα εισόδου είναι σωστά συμπληρωμένα, ο χρήστης θα ανακατευθυνθεί στην οθόνη αποτελεσμάτων, όπου θα του παρουσιαστούν τα αποτελέσματα της λύσης του προβλήματος του. Αν τα δεδομένα εισόδου του παρουσιάζουν κάποια παράληψη, θα του παρουσιαστεί ειδοποίηση αντίστοιχη με αυτό της *Εικόνας 63*. Τα δεδομένα του παραδείγματος της *Εικόνας 64* εκμαιεύθηκαν από τις ασκήσεις του μαθήματος Συστήματα Αποφάσεων της ΣΗΜΜΥ του ΕΜΠ [10].

Δυναμικός Προγραμματισμός - Δημιουργία

Πρόβλημα: Προγραμματισμός Επενδύσει Προγραμματισμός Επενδύσεων **Συμπλήρωση** **Εκτέλεση**

Χρόνια: 3
Εκατομύρια: 8

Ορισμός Επιλογών

Εκατομύρια

		0	1	2	3	4	5	6	7	8
Χρόνια	1	0	6	16	40	80	90	95	98	100
	2	0	5	12	35	60	70	73	74	75
	3	0	4	26	40	45	50	51	52	53

BLACK DASHBOARD DJANGO © Creative-Tim - coded by AppSeed.

Εικόνα 64- Παράδειγμα προβλήματος προγραμματισμού επενδύσεων

Τύπος Προβλήματος: Προγραμματισμός Παραγωγής και Αποθήκευσης

Εφόσον ο χρήστης επιλέξει να συμπληρώσει τα δεδομένα εισόδου για ένα πρόβλημα τύπου προγραμματισμού παραγωγής και αποθήκευσης, θα εμφανιστούν οι επιλογές που πρέπει να ορίσει για το πρόβλημά του, όπως αυτές φαίνονται στην *Εικόνα 65*. Αρχικά θα πρέπει να ορίσει του συντελεστές στο πεδία *Συνάρτηση Κόστους Παραγωγής* και *Συνάρτηση Κόστους Αποθήκευσης*. Αυτές οι συναρτήσεις πρώτου βαθμού ορίζουν το κόστος της παραγωγής και της αποθήκευσης ανάλογα με το πλήθος των προϊόντων που παράχθηκαν ή αποθηκεύτηκαν και την περίοδο, αντίστοιχα. Ακολουθούν τα πεδία *Παρτίδα Παραγωγής* και *Παρτίδα Αποθήκευσης*

όπου αναφέρονται στο πλήθος προϊόντων που περιέχει η ελάχιστη παρτίδα παραγωγής και αποθήκευσης, αντίστοιχα. Τέλος, θα πρέπει να ορίσει το πλήθος των περιόδων του προβλήματος στο πεδίο *Αριθμός Περιόδων*. Αφού συμπληρώσει σωστά όλα τα παραπάνω πεδία, επιλέγοντας το κουμπί *Ορισμός Επιλογών* θα του εμφανιστεί ο πίνακας των δεδομένων προς συμπλήρωση, όπως αυτός φαίνεται στην *Εικόνα 66*. Αν κάποιο από τα πεδία των επιλογών δεν έχει συμπληρωθεί θα του εμφανιστεί αντίστοιχη ειδοποίηση όπως αυτή της *Εικόνας 63*.

Εικόνα 65- Εφαρμογή δυναμικού προγραμματισμού: Δημιουργία προβλήματος προγραμματισμού παραγωγής και αποθήκευσης

Συστήματα Αποφάσεων - Δυναμ. x

← → http://127.0.0.1:8000/dynamic/create/

Δυναμικός Προγραμματισμός - Δημιουργία

Παράδειγμα: Προγραμματισμός Παραγωγής Προγραμματισμός Παραγωγής και Αποθ. **Συμπλήρωση** **Εκτέλεση**

Συνάρτηση Κόστους Παραγωγής: $C_p = (0.1 + v \times x) \times x + (0 + v \times x) \times 0$

Συνάρτηση Κόστους Αποθήκευσης: $C_a = (1 + v \times x) \times 2 \times a + (0 + v \times x) \times 0$

Παρτίδα Παραγωγής: 10 Παρτίδα Αποθήκευσης: 10

Αρχικό Απόθεμα: 0 Αριθμός Περιόδων: 5

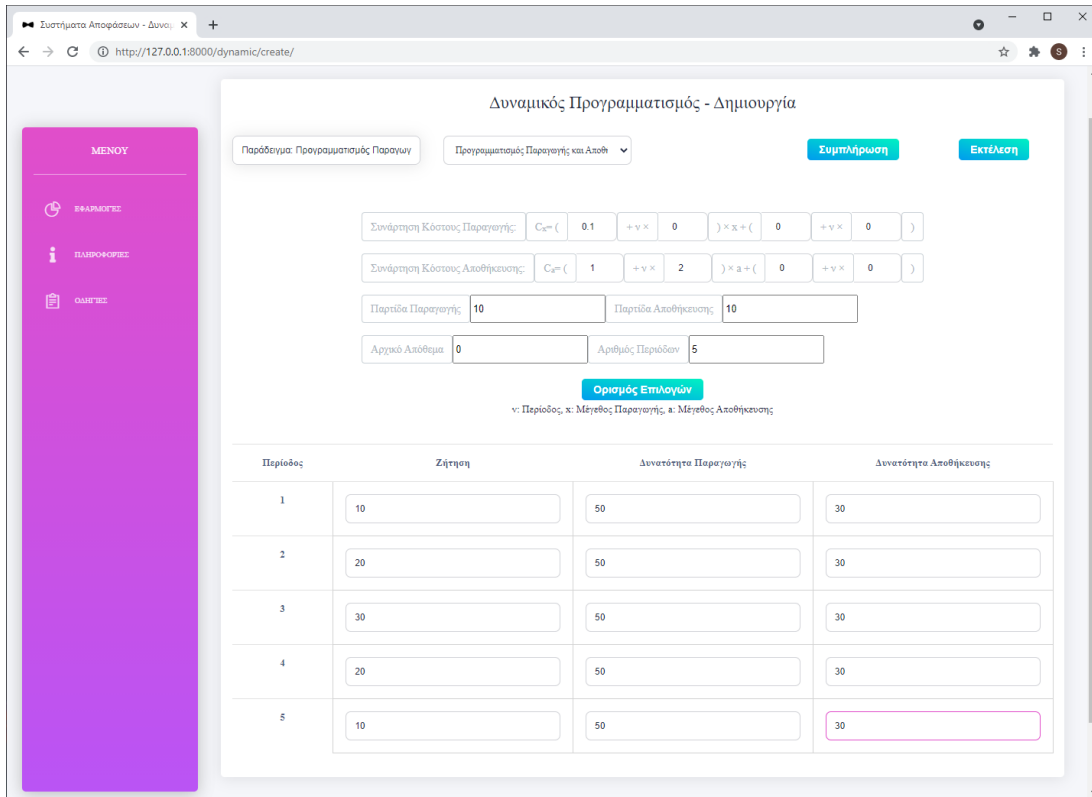
Ορισμός Επιλογών
v: Περίοδος, x: Μέγεθος Παραγωγής, a: Μέγεθος Αποθήκευσης

Περίοδος	Ζήτηση	Δυνατότητα Παραγωγής	Δυνατότητα Αποθήκευσης
1	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>	<input type="text"/>

Εικόνα 66- Συμπλήρωση δεδομένων προβλήματος προγραμματισμού παραγωγής και αποθήκευσης

Στον παραπάνω πίνακα ο χρήστης καλείται να συμπληρώσει στα στοιχεία της στήλης *Ζήτηση* το πλήθος των προϊόντων που ζητούνται προς παράδοση σε κάθε οικονομική περίοδο. Στα στοιχεία της στήλης *Δυναμικότητα Παραγωγής* θα πρέπει να συμπληρώσει το πλήθος των προϊόντων που είναι εφικτό να παραχθούν σε κάθε περίοδο. Στα στοιχεία της στήλης *Δυναμικότητα Αποθήκευσης* θα πρέπει να συμπληρώσει το πλήθος των προϊόντων που είναι εφικτό να αποθηκευτούν σε κάθε περίοδο.

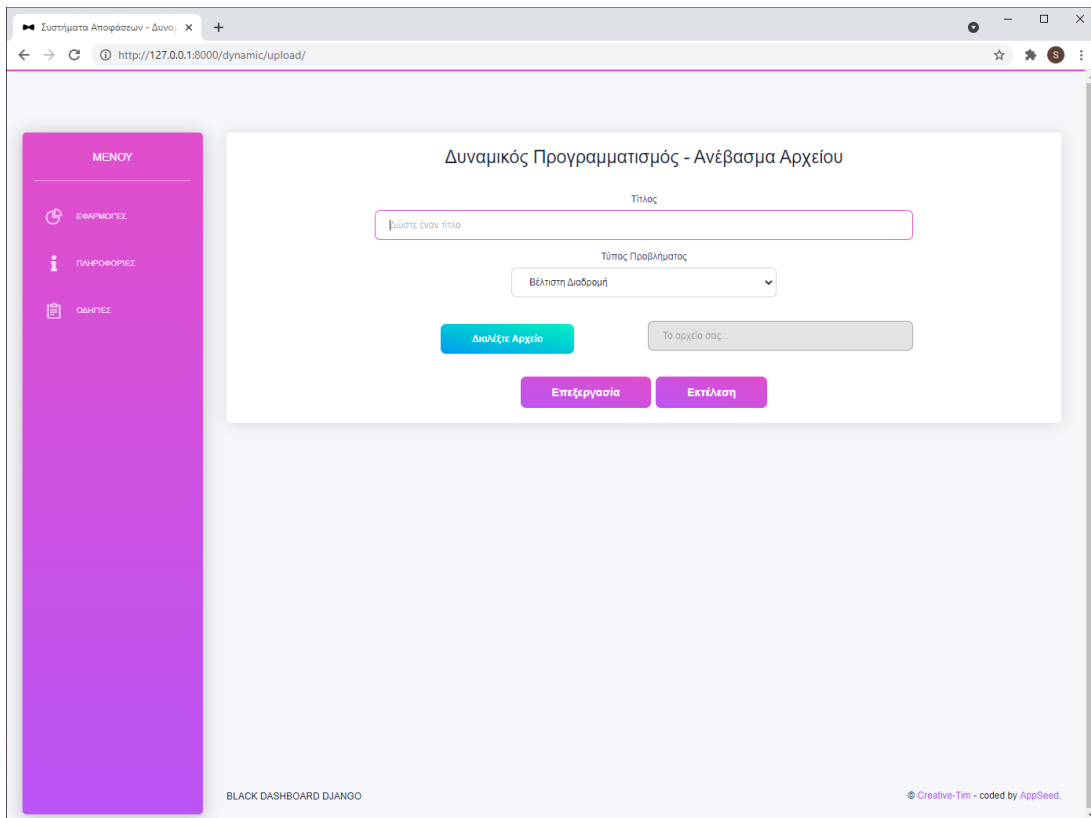
Αφού ο χρήστης συμπληρώσει σωστά όλα τα πεδία μπορεί να προχωρήσει στην επίλυση του προβλήματος επιλέγοντας το κουμπί *Εκτέλεση*. Με αυτόν τον τρόπο θα ανακατευθυνθεί στη σελίδα αποτελεσμάτων, όπου θα του παρουσιαστεί η επίλυση του προβλήματός του. Ένα παράδειγμα προβλήματος τύπου προγραμματισμού παραγωγής και αποθήκευσης είναι αυτό της *Εικόνας 67*. Αν ο χρήστης έχει παραλείψει να συμπληρώσει κάποιο πεδίο του πίνακα, θα λάβει ειδοποίηση αντίστοιχη με αυτή της *Εικόνας 63*.



Εικόνα 67- Παράδειγμα προβλήματος προγραμματισμού παραγωγής και αποθήκευσης

6.4.3 Οθόνη Ανεβάσματος Αρχείου

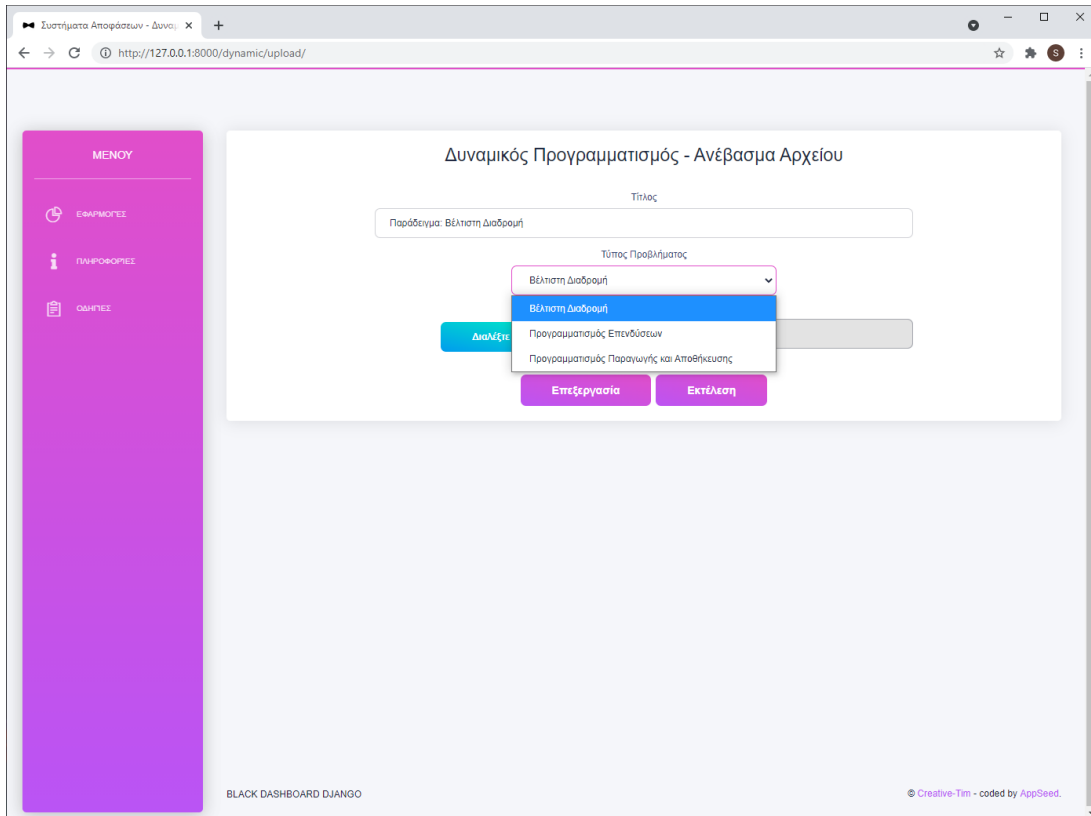
Σε αυτή την οθόνη ο χρήστης μπορεί να παρέχει τα δεδομένα εισόδου ενός προβλήματος δυναμικού προγραμματισμού με μορφή αρχείου. Ο χρήστης μπορεί να ανακατευθυνθεί σε αυτή μόνο μέσω της αρχικής οθόνης της εφαρμογής δυναμικού προγραμματισμού εφόσον έχει επιλέξει να παρέχει τα δεδομένα του μέσω ενός αρχείου JSON. Το στιγμιότυπο της οθόνης φαίνεται στη *Εικόνα 68*.



Εικόνα 68- Εφαρμογή δυναμικού προγραμματισμού: Οθόνη ανεβάσματος αρχείου

Σε αυτή ο χρήστης, εφόσον το επιθυμεί, καλείται να συμπληρώσει τον τίτλο του προβλήματός δυναμικού προγραμματισμού στο πεδίο *Δώστε έναν τίτλο*. Στη συνέχεια στο πεδίο *Τύπος Προβλήματος* πρέπει να επιλέξει τον τύπο του προβλήματος που επιθυμεί να επιλέξει. Οι δυνατές επιλογές, όπως φαίνονται στην *Εικόνα 69*, είναι: Βέλτιστη Διαδρομή, Προγραμματισμός Επενδύσεων, Προγραμματισμός Παραγωγής και Αποθήκευσης. Στη συνέχεια στο πεδίο *Διαλέξτε Αρχείο* μπορεί να επιλέξει ένα αρχείο τύπου JSON από τον τοπικό του υπολογιστή. Οι περιορισμοί του τύπου του αρχείου και του περιεχομένου του που αναφέρθηκαν στην ενότητα 6.2.3 ισχύουν κι εδώ.

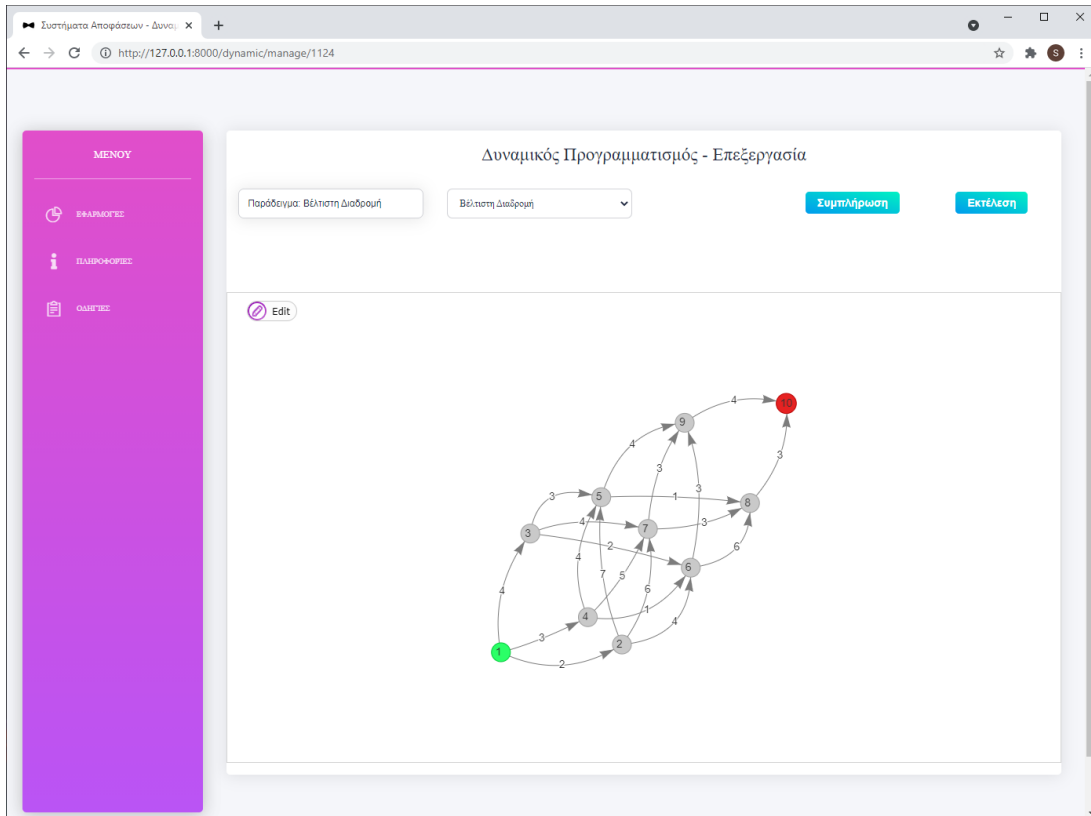
Εφόσον η φόρμα έχει συμπληρωθεί σωστά, με την επιλογή του κουμπιού *Επεξεργασία* ο χρήστης ανακατευθύνεται στη σελίδα επεξεργασίας δεδομένων ενός προβλήματος δυναμικού προγραμματισμού. Σε άλλη περίπτωση που ο χρήστης επιθυμεί να επιλύσει άμεσα το πρόβλημά του, επιλέγει το κουμπί *Εκτέλεση*, το οποίο τον ανακατευθύνει στην οθόνη αποτελεσμάτων.



Εικόνα 69- Εφαρμογή δυναμικού προγραμματισμού: Οθόνη ανεβάσματος αρχείου – Επιλογή τύπου προβλήματος

6.4.4 Οθόνη Επεξεργασίας Δεδομένων Δυναμικού Προγραμματισμού

Σε αυτή την οθόνη ο χρήστης μπορεί να επεξεργαστεί τα δεδομένα εισόδου ενός προβλήματος δυναμικού προγραμματισμού. Υπάρχουν δύο τρόποι με τους οποίους μπορεί να βρεθεί σε αυτή τη σελίδα. Είτε μέσω της σελίδας ανεβάσματος αρχείου, είτε μέσω της σελίδας αποτελεσμάτων. Σε κάθε περίπτωση ο χρήστης λαμβάνει τα δεδομένα εισόδου πλήρως απεικονισμένα και μπορεί να τα επεξεργαστεί, όπως ακριβώς παρουσιάστηκε στην ενότητα 6.4.3. Στην *Εικόνα 70* φαίνεται το παράδειγμα της *Εικόνας 60* όπως αυτό εμφανίζεται στην οθόνη επεξεργασίας δεδομένων.



Εικόνα 70- Εφαρμογή δυναμικού προγραμματισμού: Οθόνη επεξεργασίας

6.4.5 Οθόνη Αποτελεσμάτων

Εδώ απεικονίζονται και παρουσιάζονται στον χρήστη η λύση ενός προβλήματος δυναμικού προγραμματισμού. Ανάλογα με τον τύπο του προβλήματος στον χρήστη εμφανίζονται διαφορετικού είδους δεδομένα. Στη συνέχεια θα περιγραφεί το περιεχόμενο της οθόνης αποτελεσμάτων για κάθε περίπτωση. Σε κάθε περίπτωση, στο πάνω μέρος της οθόνης εμφανίζεται ο τίτλος του προβλήματος και στη συνέχεια δύο κουμπιά. Επιλέγοντας το κουμπί *Επεξεργασία* ο χρήστης μπορεί να επεξεργαστεί εκ νέου τα δεδομένα εισόδου του στην οθόνη επεξεργασίας δεδομένων, ενώ επιλέγοντας το κουμπί *Νέα Δεδομένα έχει* τη δυνατότητα να καταχωρήσει νέα δεδομένα για ένα πρόβλημα δυναμικού προγραμματισμού και ανακατευθύνεται στην αρχική οθόνη της εφαρμογής δυναμικού προγραμματισμού ώστε να επιλέξει τον τρόπο με τον οποίο θα παρέχει τα δεδομένα του.

Τύπος Προβλήματος: Βέλτιστη Διαδρομή

Στην *Εικόνα 71* φαίνονται τα αποτελέσματα της επίλυσης του προβλήματος της *Εικόνας 60*.

Συστήματα Αποφάσεων - Δυναμικό

http://127.0.0.1:8000/dynamic/results/1124

Δυναμικός Προγραμματισμός - Αποτέλεσμα

Τίτλος Παράδειγμα: Βέλτιστη Διαδρομή Τύπος Προβλήματος: Βέλτιστη Διαδρομή [Επιβεβαιώστε](#) [Νέα Διάγνωση](#)

Γράφος Εισόδου

Βέλτιστη Λύση

Υπάρχουν 3 βέλτιστες διαδρομές:
 1 - 4 - 5 - 8 - 10, 1 - 4 - 6 - 9 - 10, 1 - 3 - 5 - 8 - 10.
 Η καλύτερη απόσταση είναι 11.

Επαναλήψεις

Επανάληψη: 1

B_1	9	8
$F_1^*(B_1)$	4	3
x_1^*	10	10

Επανάληψη: 2

$B_2 \setminus x_2$	9	8	$F_2^*(B_2)$	x_2^*
7	7	6	6	8
6	8	4	4	8
8	7	9	7	9

Επανάληψη: 3

$B_3 \setminus x_3$	7	6	8	$F_3^*(B_3)$	x_3^*
4	11	8	8	8	5,6
2	12	11	11	11	5,6
5	10	7	9	7	5

Επανάληψη: 4

$B_4 \setminus x_4$	4	2	8	$F_4^*(B_4)$	x_4^*
1	11	13	11	11	4,3

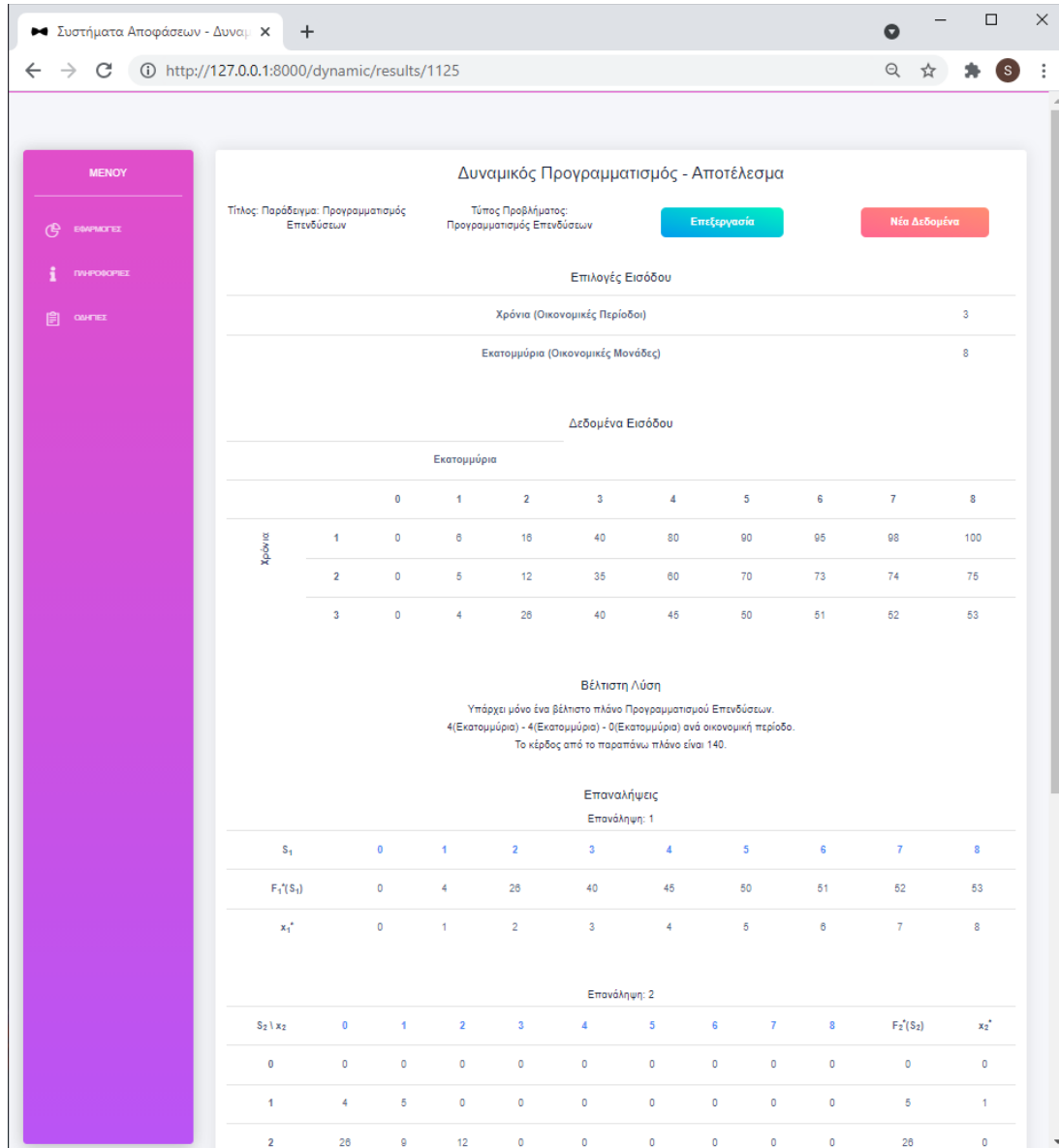
© Creative-101 - coded by AppSquad

Εικόνα 71- Εφαρμογή δυναμικού προγραμματισμού: Οθόνη αποτελεσμάτων – παράδειγμα βέλτιστης διαδρομής

Αρχικά, στο τμήμα *Γράφος Εισόδου* απεικονίζεται στον χρήστη ο γράφος που επέλεξε ως είσοδο στο πρόβλημά του. Στη συνέχεια, στο τμήμα *Βέλτιστη Λύση* του παρουσιάζονται οι βέλτιστες λύσεις του προβλήματος καθώς και η τιμή της βέλτιστης διαδρομής από ένα κόμβο εκκίνησης σε έναν τερματικό κόμβο. Τέλος, στο τμήμα *Επαναλήψεις* ο χρήστης μπορεί να δει τους πίνακες που κατασκευάστηκαν κατά την επίλυση του προβλήματος και πώς έφτασε η εφαρμογή στη βέλτιστη λύση.

Τύπος Προβλήματος: Προγραμματισμός Επενδύσεων

Στην *Εικόνα 72* φαίνονται τα αποτελέσματα της επίλυσης του προβλήματος της *Εικόνας 64*.



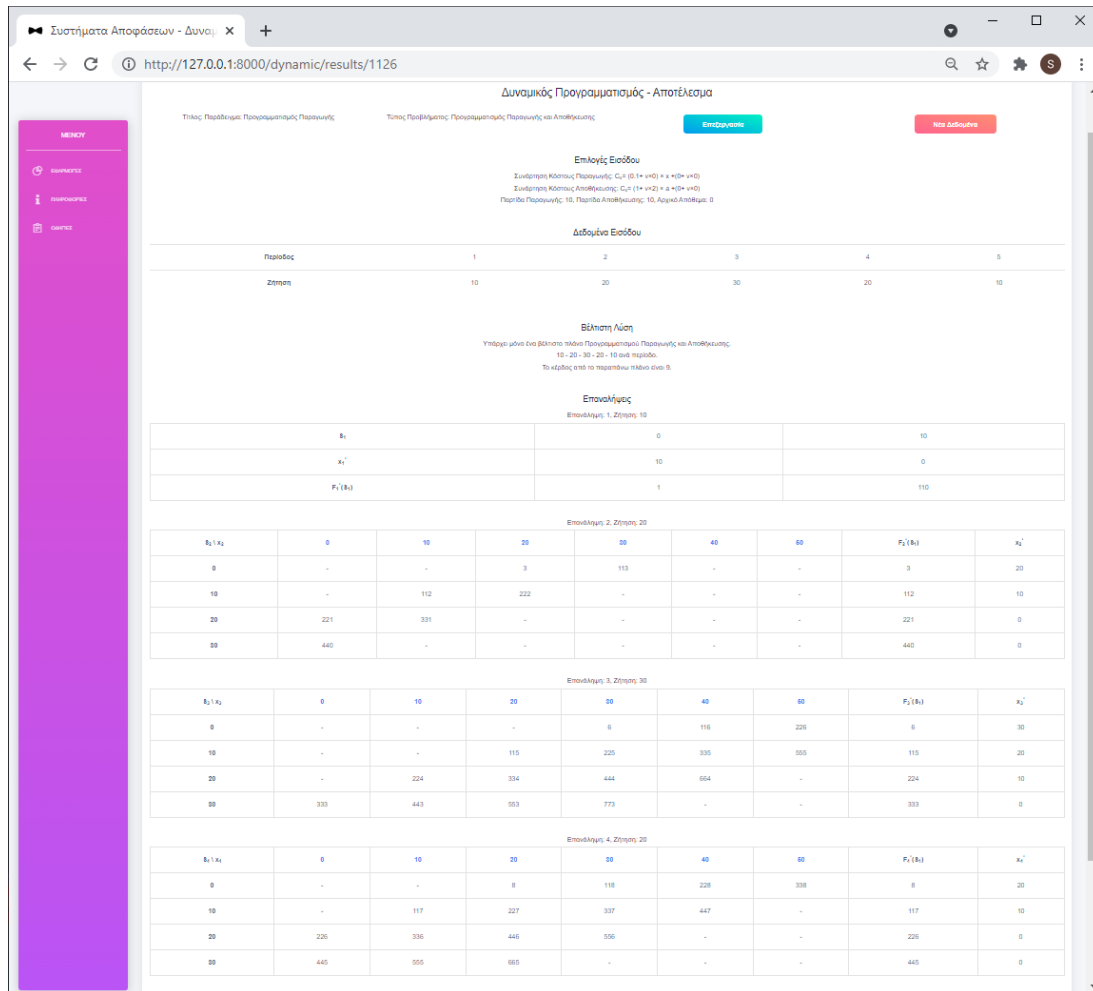
Εικόνα 72 - Εφαρμογή δυναμικού προγραμματισμού: οθόνη αποτελεσμάτων – παράδειγμα προγραμματισμού επενδύσεων

Αρχικά, στο τμήμα *Επιλογές Εισόδου* ο χρήστης μπορεί να δει τις επιλογές που έκανε για τις οικονομικές περιόδους και τις οικονομικές μονάδες. Στη συνέχεια, στο τμήμα *Δεδομένα Εισόδου* του παρουσιάζεται ο πίνακας δεδομένων εισόδου που έχει ορίσει για το πρόβλημά του. Στο επόμενο τμήμα, *Βέλτιστη Λύση*, του επεξηγείται η βέλτιστη λύση του προβλήματός του. Δηλαδή, οι οικονομικές μονάδες που πρέπει να

επενδυθούν ανά οικονομική περίοδο ώστε το κέρδος να είναι μέγιστο. Τέλος, στο τμήμα *Επαναλήψεις*, ο χρήστης μπορεί να δει του πίνακες που κατασκεύασε η εφαρμογή κατά τη διαδικασία επίλυσης του προβλήματός του.

Τύπος Προβλήματος: Προγραμματισμός Παραγωγής και Αποθήκευσης

Στην *Εικόνα 73* φαίνονται τα αποτελέσματα της επίλυσης του προβλήματος της *Εικόνας 67*.



Εικόνα 73- - Εφαρμογή δυναμικού προγραμματισμού: Οθόνη αποτελεσμάτων - παράδειγμα προγραμματισμού παραγωγής και αποθήκευσης

Αρχικά, στο τμήμα *Επιλογές Εισόδου* παρουσιάζονται στον χρήστη η Συνάρτηση Κόστους Παραγωγής και η Συνάρτηση Κόστους Αποθήκευσης, όπως αυτός τις όρισε. Στη συνέχεια, στο πεδίο *Δεδομένα Εισόδου* του απεικονίζεται ο πίνακας της ζήτησης ανά περίοδο. Στο τμήμα *Βέλτιστη Λύση* του παρουσιάζεται η βέλτιστη λύση. Δηλαδή, οι μονάδες που πρέπει να παραχθούν ανά περίοδο ώστε να επιτευχθεί το ελάχιστο κόστος παραγωγής και αποθήκευσης. Τέλος, στο τμήμα *Επαναλήψεις*, ο χρήστης

μπορεί να δει του πίνακες που κατασκεύασε η εφαρμογή κατά τη διαδικασία επίλυσης του προβλήματός του.

7 Συμπεράσματα, Περιορισμοί και Προτάσεις Μελλοντικής Εξέλιξης

Στο κεφάλαιο αυτό γίνεται περιγραφή των αποτελεσμάτων και των συμπερασμάτων που εξάχθηκαν κατά την υλοποίηση της διαδικτυακής εφαρμογής λήψης αποφάσεων. Αρχικά αναλύεται η γενική εικόνα της εφαρμογής, δηλαδή σε ποιο βαθμό εξυπηρετεί το στόχο της και υπό ποιες προϋποθέσεις. Στη συνέχεια, παρουσιάζονται τα πλεονεκτήματα και τα μειονεκτήματα του εργαλείου λήψης αποφάσεων, καθώς και οι περιορισμοί οι οποίοι εμφανίστηκαν κατά την υλοποίηση. Τέλος, γίνονται εισηγήσεις εξέλιξης για την παρούσα εφαρμογή.

7.1 Συμπεράσματα και Αποτελέσματα

Η διαδικτυακή εφαρμογή που αναπτύχθηκε, στο πλαίσιο της παρούσας διπλωματικής εργασίας, έχει ως στόχο τη συγκέντρωση και την υλοποίηση διαφόρων μεθόδων επίλυσης προβλημάτων επιχειρησιακής, καθιστώντας τη ένα ολοκληρωμένο εργαλείο λήψης αποφάσεων. Ανώτερος σκοπός του εργαλείου είναι η διευκόλυνση της εκπαιδευτικής διαδικασίας πάνω σε θέματα επιχειρησιακής ερευνάς και λήψης αποφάσεων, αλλά και η εκτενής πειραματική χρήση του σε εμπορικό/επαγγελματικό επίπεδο (και σε διάφορους τομείς), ώστε να επιλύονται αποτελεσματικά τα ανάλογα προβλήματα αποφάσεων, ανεξαρτήτως του τύπου της πηγής των δεδομένων εισόδου.

Υπάρχουν διάφορες εφαρμογές οι οποίες είναι ικανές είτε να κατασκευάσουν, είτε και να λύσουν διάφορα προβλήματα συστημάτων αποφάσεων, όπως είναι το WolframAlpha - Linear Programming Solver, το οποίο καταφέρνει να λύσει με επιτυχία προβλήματα γραμμικού προγραμματισμού, Lucidchart – Decision Tree Maker, το οποίο αποτελεί ένα σύγχρονο και φιλικό προς τη χρήση εργαλείο κατασκευής δέντρων αποφάσεων, και το Online Calculator, το οποίο καταφέρνει να συγκεντρώσει ένα αξιολογικό σύνολο εργαλείων λήψης αποφάσεων σε ένα φιλικό προς τον χρήστη περιβάλλον. Προτού εκκινήσει η υλοποίηση της παρούσας εφαρμογής, μελετήθηκαν και αναλύθηκαν οι υλοποιήσεις των παραπάνω εφαρμογών. Αφού διαχωρίστηκαν τα προτερήματα και τα μειονεκτήματα της κάθε εφαρμογής, αποτέλεσαν σημαντική πηγή έμπνευσης και ιδεών στο στάδιο του σχεδιασμού της δικής μας εφαρμογής.

Ύστερα από την παραπάνω ανάλυση, τέθηκε ως στόχος, και μάλιστα επιτεύχθηκε, πως ανεξάρτητα από το πλήθος των μεθόδων που θα καλύπτει η εφαρμογή, θα πρέπει να είναι φιλική προς τον χρήστη χωρίς να απαιτεί την προσθήκη πολύπλοκων και μαθηματικοποιημένων εισόδων από αυτόν, όπως επίσης να είναι ενσωματωμένη σε ένα σύγχρονο περιβάλλον χρήσης. Μετά από προσεκτικό σχεδιασμό, αλλά και

παραχωρήσεις στο εύρος των σεναρίων χρήσης που θα καλύπτει, δημιουργήθηκαν τρεις καλαίσθητες και εύχρηστες εφαρμογές, οι εφαρμογή δέντρων αποφάσεων, εφαρμογή γραμμικού προγραμματισμού και η εφαρμογή δυναμικού προγραμματισμού. Αποκορύφωμα της επίτευξης του προαναφερθέντος στόχου αποτελεί η καινοτόμα διαχείριση των πληροφοριών για τις καταστάσεις της φύσης, δηλαδή των μελετών, που προέκυψε από την κατάλληλη παραμετροποίηση του εργαλείου διαχείρισης γράφων VisJS. Αξιόλογη ήταν και η υλοποίηση του εργαλείου συλλογής δεδομένων για ένα πρόβλημα γραμμικού προγραμματισμού. Αν και περιορίζει τον χρήστη σε προβλήματα μέχρι οκτώ μεταβλητών απόφασης, η συμπλήρωση των δεδομένων είναι πλήρως κατανοητή και ανεξάρτητη, όσο ήταν εφικτό, από τις πολύπλοκες μαθηματικές παραστάσεις που συμπεριλαμβάνουν τέτοιους είδους προβλήματα. Συμπερασματικά, τόσο η ομάδα προγραμματισμού, όσο και χρήστες που ήρθαν σε επαφή με την εφαρμογή έμεινα πλήρως ικανοποιημένοι από το οπτικό αποτέλεσμά της, αλλά και την ευκολία στη χρήση της.

Κάνοντας αναφορά σε μαθηματικές παραστάσεις, αξίζει να σημειωθεί σε αυτό το σημείο πως λόγω του όγκου των δεδομένων ενός προβλήματος επιχειρησιακής έρευνας, ενέχει ο κίνδυνος εισόδου άκυρων δεδομένων από κάποιον χρήστη. Προς αποφυγήν τέτοιου είδους καταστάσεων, μελετήθηκαν διεξοδικά τα σενάρια χρήσης ώστε να αποφευχθούν τέτοιου είδους καταστάσεις, εμποδίζοντας τη διαδικασία επίλυσης και παρέχοντας τις απαραίτητες πληροφορίες στον χρήστη για τη σωστή συμπλήρωση των πεδίων. Το ίδιο συμβαίνει και κατά την εισαγωγή αρχείων εισόδου, όπου ελέγχεται πλήρως εγκυρότητά τους από την εφαρμογή. Συνεπώς, σε κάθε βήμα διασφαλίζεται η συμβατότητα των δεδομένων εισόδου του χρήστη, αλλά και η πλήρης καθοδήγηση αυτού στη σωστή παροχή αυτών.

Η εφαρμογή δεν θα μπορούσε να παρέχει μια ευχάριστη εμπειρία στον χρήστη αν δεν κατάφερνε να επιλύσει τα προβλήματα που απαιτούσε ο χρήστης. Για αυτό το σκοπό, πραγματοποιήθηκε ένας σημαντικός αριθμός δοκιμών ώστε να εξασφαλιστεί η ορθότητα των αλγόριθμων επίλυσης. Από τις παραπάνω δοκιμές πραγματοποιήθηκαν οι κατάλληλες διορθώσεις ώστε να παραδοθεί ένα εργαλείο το οποίο θα επιτελεί το βασικό στόχο του. Δηλαδή, τόσο να επιλύει προβλήματα λήψης αποφάσεων, αλλά και να παρουσιάζει τα στάδια επίλυσης στον χρήστη. Με τη διαμόρφωση κατάλληλων αλγορίθμων επιτεύχθηκε όχι μόνο η επιτυχής επίλυση προβλημάτων δέντρων αποφάσεων, γραμμικού προγραμματισμού και ορισμένες περιορισμένες περιπτώσεις δυναμικού προγραμματισμού, αλλά να το καταφέρνει σε αποδοτικό χρόνο χωρίς η διαδικασία να ενέχει χρόνους αναμονής για τον χρήστη και η εφαρμογή να είναι άμεσα ανταποκρινόμενη.

Τέλος, ένας ακόμη βασικός στόχος της εν λόγω εφαρμογής ήταν η διάθεσή της ως ένα εργαλείο ανοιχτού κώδικα ώστε να αναπτυχθεί μια κοινότητα η οποία θα μπορεί να προσθέσει νέες λειτουργικότητες, καθώς και να συντηρήσει και να αναβαθμίσει τις ήδη υπάρχουσες. Επομένως, σε κάθε στάδιο υλοποίησης έπρεπε να διασφαλιστεί η δυνατότητα επεκτασιμότητας της εφαρμογής. Η επιλογή του Django ως διαδικτυακό

πλαίσιο (web framework) για την κατασκευή της εφαρμογής όχι μόνο συνέβαλε στην κατασκευή μιας πλήρως επεκτάσιμης εφαρμογής, αλλά και καθιστά της διαδικασία προσθήκης νέων λειτουργικοτήτων σε αυτή εύκολη και άμεση, καθώς τα διαφορετικά τμήματά της λειτουργούν και έχουν αναπτυχθεί ανεξάρτητα μεταξύ τους.

7.2 Πλεονεκτήματα και Μειονεκτήματα της Υλοποίησης

Η χρήση του εργαλείου που κατασκευάστηκε είχε σημαντικά πλεονεκτήματα σε σχέση με άλλες εφαρμογές. Αυτά που εντοπίστηκαν μετά την ολοκλήρωση και τη χρήση του είναι τα εξής:

- Η εφαρμογή είναι λειτουργική, γρήγορη και επιλύει με επιτυχία τα προβλήματα για τα οποία κατασκευάστηκε. Μάλιστα, οι αλγόριθμοι που κατασκευάστηκαν είναι αποδοτικοί χωρίς να υπάρχουν χρόνοι αναμονής για τον χρήστη.
- Το περιβάλλον χρήσης του εργαλείου είναι καλαίσθητο και διαδραστικό παρέχοντας στον χρήστη την ευελιξία να πειραματιστεί με τα δεδομένα εισόδου του.
- Η εφαρμογή καταφέρνει να συγκεντρώσει εργαλεία για την επίλυση τριών διαφορετικών ειδών προβλημάτων λήψης απόφασης, αλλά και υποπεριπτώσεις αυτών, κάνοντάς το ένα ολοκληρωμένο εργαλείο λήψης αποφάσεων.
- Ο τρόπος με τον οποίο έχει δομηθεί η εφαρμογή επιτρέπει και διευκολύνει την προσθήκη περισσότερων εργαλείων λήψης αποφάσεων καλύπτοντας περισσότερες μεθόδους των συστημάτων αποφάσεων. Αυτό το χαρακτηριστικό καθιστά την εφαρμογή πλήρως επεκτάσιμη.
- Το εργαλείο δίνει τη δυνατότητα στον χρήστη να παρέχει τα δεδομένα του με περισσότερους από έναν τρόπο, δίνοντάς του ελευθερία κινήσεων. Επιπλέον, του παρέχονται φιλικά εργαλεία, μη μαθηματικοποιημένα, με πλήρως κατανοητά πεδία προς συμπλήρωση για την κατασκευή των προβλημάτων του στην εφαρμογή.
- Η διαδικτυακή εφαρμογή δίνει βάση μόνο στις αριθμητικές τιμές των δεδομένων εισόδου, αδιαφορώντας για την προέλευση τους, καθιστώντας της ένα πολυπρισματικό εργαλείο λήψης αποφάσεων.
- Η λύση παρουσιάζεται αναλυτικά στον χρήστη μαζί με τα στάδια που πραγματοποιήθηκαν κατά τη διαδικασία επίλυσης ώστε να γίνεται κατανοητή η επεξεργασία των δεδομένων μέχρι τη λύση.
- Η εφαρμογή δέντρων αποφάσεων διαχειρίζεται κατά ιδιαίτερο και καινοτόμο τρόπο τις μελέτες ενός προβλήματος δέντρων αποφάσεων, διευκολύνοντας σημαντικά τον χρήστη στο σχεδιασμό τους.

- Προορίζεται για εργαλείο ανοιχτού κώδικα, συνεπώς δεν προβλέπεται η χρέωση για τη χρήση της εφαρμογής. Επιπλέον, αυτό το χαρακτηριστικό δίνει προοπτικές εξέλιξης στην εφαρμογή, από την κοινότητα χρήσης της.

Ωστόσο, στην υλοποίηση δεν θα μπορούσαν να λείπουν τα μειονεκτήματα:

- Η αδυναμία χρήσης διαδικτυακού εξυπηρετητή για την απομακρυσμένη χρήση της εφαρμογής, την περιορίζει σε τοπική χρήση.
- Δεν διαθέτει σύστημα διαχείρισης χρηστών το οποίο θα μπορούσε να παρέχει σημαντικά ακόμη πλεονεκτήματα σε αυτή, τα οποία αναφέρονται σε επόμενη ενότητα.
- Λόγω της φύσης του δυναμικού προβληματισμού, που η λύση διαμορφώνεται εξατομικευμένα για κάθε πρόβλημα, η εφαρμογή του καταφέρνει να καλύψει μόλις τρία είδη προβλημάτων και αυτά υπό συγκεκριμένες προϋποθέσεις.
- Αν και αλγοριθμικά επιτρέπεται η επίλυση προβλημάτων γραμμικού προγραμματισμού χωρίς περιορισμό στον αριθμό των μεταβλητών απόφασης, η εφαρμογή περιορίζεται στον μέγιστο αριθμό των οκτώ για τη διασφάλιση της καλαισθησίας και ευχρηστίας της.

7.3 Περιορισμοί

7.3.1 Περιορισμοί Συνολική Εφαρμογής

Ο σκοπός της εφαρμογής, όπως έχει αναφερθεί παραπάνω, είναι να αποτελεί ένα ολοκληρωμένο και πολυπρισματικό εργαλείο λήψης αποφάσεων. Αν και σχεδιάστηκε και υλοποιήθηκε ώστε να καλύπτει τρεις βασικές μεθόδους της επιχειρησιακής έρευνας και των συστημάτων αποφάσεων, δεν παύει να αποτελεί ένα μικρό μέρος των μεθόδων που πραγματεύεται η εν λόγω επιστήμη. Το γεγονός ότι η ομάδα υλοποίησης αποτελούνταν από ένα άτομο κατέστησε χρονικά αδύνατη την υλοποίηση περισσότερων μεθόδων. Για αυτό το λόγο επιλέχθηκαν τρεις υπολογιστικές μέθοδοι που θα υποστηρίζονται από την εφαρμογή που υλοποιήθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας. Παρ' όλα αυτά δόθηκε η δέουσα προσοχή στην επεκτασιμότητα αυτής, ώστε να είναι εφικτή η ενσωμάτωση επιπλέον μεθόδων.

7.3.2 Περιορισμοί Εφαρμογής Δέντρων Αποφάσεων

Δεδομένου ότι η συνολική εφαρμογή δεν επικεντρώνεται στην υλοποίηση μιας μόνο μεθόδου λήψης αποφάσεων, αλλά σε τρεις, δεν ήταν εφικτή η κατασκευή ενός εξατομικευμένου εργαλείου διαχείρισης δέντρων για την εφαρμογή δέντρων αποφάσεων. Για να επιτευχθεί η υλοποίηση της εφαρμογής χρησιμοποιήθηκε μια

βιβλιοθήκη της javascript, η visjs. Η συγκεκριμένη βιβλιοθήκη αν και προορίζεται για την απλή διαχείριση γράφων και προσθαφαίρεση κόμβων και ακμών, δεν κάλυπτε τις ανάγκες της εφαρμογής. Συνεπώς, έπρεπε να επεξεργαστεί καταλλήλως ώστε να επιτελεί τις επιθυμητές λειτουργίες. Ωστόσο, η ελλιπής τεκμηρίωση αυτής της βιβλιοθήκης, σε συνδυασμό με την περιορισμένη χρήση της από την προγραμματιστική κοινότητα, δυσχέρανε σημαντικά το έργο της προγραμματιστικής ομάδας. Παρότι εν τέλη δημιουργήθηκε ένα εύχρηστο και καλαίσθητο αποτέλεσμα με το εργαλείο να επιτελεί τις επιθυμητές λειτουργίες, η διαδικασία αποδείχθηκε σημαντικά χρονοβόρα και επίπονη.

7.3.3 Περιορισμοί Εφαρμογής Γραμμικού Προγραμματισμού

Τα προβλήματα γραμμικού προγραμματισμού μπορούν να αποτελούνται από μεγάλο αριθμό μεταβλητών απόφασης. Αν και ο αλγόριθμος που κατασκευάστηκε για την επίλυση προβλημάτων περισσότερων των δύο μεταβλητών δεν έχει άνω όριο στο πλήθος αυτών, δεν ήταν δυνατή η υποστήριξή τους από τη διεπαφή του χρήστη. Συγκεκριμένα, για περισσότερες από οκτώ μεταβλητές απόφασης επηρεαζόταν σημαντικά το οπτικό κομμάτι της εφαρμογής. Συνεπώς, έγινε η σύμβαση η παραπάνω εφαρμογή να καλύπτει προβλήματα δύο έως οκτώ μεταβλητών απόφασης για τη διασφάλιση της καλαίσθητης εμπειρίας κατά τη χρήση της.

7.3.4 Περιορισμοί Εφαρμογής Δυναμικού Προγραμματισμού

Το αντικείμενο του δυναμικού προγραμματισμού είναι σημαντικά ευρύ. Σύνηθες φαινόμενο σε προβλήματα τέτοιου τύπου είναι να απαιτούν εξατομικευμένες λύσεις. Επιπλέον, αυτή ιδιαιτερότητα μπορεί να παρουσιαστεί σε προβλήματα ίδιου αντικειμένου. Κατ' επέκταση ήταν αρκετά πολύπλοκο να κατασκευαστεί μια εφαρμογή η οποία να καλύπτει όλο το φάσμα των προβλημάτων του δυναμικού προγραμματισμού. Επομένως, ελήφθη η απόφαση να καλυφθούν ορισμένες συγκεκριμένες αναδρομικές σχέσεις και κατά συνέπεια ένα περιορισμένο σύνολο προβλημάτων. Ωστόσο, ο τρόπος με τον οποίο έχει σχεδιαστεί και δομηθεί η εφαρμογή επιτρέπει την προσθήκη υποστηρίξης περισσότερων τύπων προβλημάτων.

7.4 Προτάσεις Εξέλιξης

Η διαδικτυακή εφαρμογή που αναπτύχθηκε αποτελεί μεν ένα ολοκληρωμένο εργαλείο λήψης αποφάσεων, ωστόσο είναι εφικτή η προσθήκη επιπλέον λειτουργιών, ώστε να εμπλουτιστεί η λειτουργικότητα της και η εμπειρία του χρήστη σε αυτή.

7.4.1 Προτάσεις σχετικά με το περιεχόμενο

Το εργαλείο που κατασκευάστηκε στο πλαίσιο της διπλωματική εργασίας υλοποιήθηκε με τέτοιο τρόπο ώστε να είναι η εφικτή όχι μόνο η προσθήκη εξ' ολοκλήρου νέων μεθόδων, αλλά και η επέκταση των ήδη υπαρχόντων μεθόδων για την κάλυψη περισσότερων σεναρίων χρήσης. Επιπλέον, η διαδικτυακή εφαρμογή και ο κώδικάς της προορίζονται για ανοιχτή χρήση γεγονός που επιτρέπει σε όποιον προτίθεται να λάβει ένα αντίγραφο του κώδικα και να τον επεκτείνει. Κάποιες επεκτάσεις που προτείνονται είναι οι εξής:

- Η προσθήκη νέων μεθόδων λήψης αποφάσεων θα μπορούσε να διευρύνει το πεδίο εφαρμογής του εργαλείου και δώσει λύσεις σε ακόμα περισσότερα είδη προβλημάτων της επιχειρησιακής έρευνας. Κάποιες από τις μεθόδους που θα μπορούσαν να προστεθούν είναι ο ακέραιος προγραμματισμός, οι προβλέψεις και η θεωρία παιγνίων. Επιπλέον θα μπορούσαν να χρησιμοποιηθούν και πολυκριτηριακές μέθοδοι όπως είναι η μέθοδος Promethee [45], η μέθοδος Electre [46], η μέθοδος Vikor [47] και άλλες.
- Η προσθήκη επιπλέον κριτηρίων λήψης αποφάσεων σε συνθήκες αβεβαιότητας θα ενίσχυε τα σενάρια χρήσης της εφαρμογής των δέντρων αποφάσεων. Κάποια επιπλέον κριτήρια που θα μπορούσαν να χρησιμοποιηθούν είναι το κριτήριο Laplace, το κριτήριο Hurwicz και το κριτήριο minimax regret [48].
- Σημαντική επέκταση στην εφαρμογή του δυναμικού προγραμματισμού θα ήταν η προσθήκη επιπλέον αναδρομικών σχέσεων και αλγορίθμων ώστε να διευρύνει το φάσμα των προβλημάτων που επιλύει.
- Αν και οι μέθοδοι του γραμμικού προγραμματισμού έχουν καλυφθεί σε σημαντικό ποσοστό, η εύρεση της χρυσής τομής του αισθητικού τομέα και την εξυπηρέτηση προβλημάτων περισσότερων των οκτώ μεταβλητών απόφασης θα μπορούσε να ελαχιστοποιήσει ή και να εκμηδενίσει τον περιορισμό χρήσης της εν λόγω εφαρμογής.

7.4.2 Προτάσεις σχετικά με τη λειτουργικότητα

Αν και η παρούσα διαδικτυακή εφαρμογή εξυπηρετεί πιστά τον σκοπό για τον οποίο κατασκευάστηκε και είναι πλήρως λειτουργική, υπάρχει πάντα χώρος για προσθήκη νέων λειτουργιών οι οποίες θα βελτιώσουν την εμπειρία του χρήστη σε αυτή. Η σημαντικότερη, που προτείνεται, είναι η ενσωμάτωση ενός συστήματος διαχείρισης χρηστών. Συγκεκριμένα, η προσθήκη λογαριασμών χρηστών ανοίγει διάπλατα έναν δρόμο περαιτέρω επεκτάσεων στις λειτουργικότητες τις εφαρμογής. Ορισμένες από αυτές είναι οι εξής:

- Δυνατότητα καταγραφής των προβλημάτων που επίλυσε ο χρήστης και διατήρηση ιστορικού.
- Δυνατότητα πρόσβασης του χρήστη στα δεδομένα εισόδου και δεδομένα εξόδου.
- Συλλογή στατιστικών δεδομένων των χρηστών και των εφαρμογών που χρησιμοποιούν.
- Δημιουργία ομάδων εργασίας μεταξύ των χρηστών όπου θα μπορούν να αλληλεπιδρούν μεταξύ τους για τη δημιουργία και την επίλυση προβλημάτων.

Για να μπορούν τα παραπάνω να εφαρμοστούν θα πρέπει η εφαρμογή να εγκατασταθεί και να εκτελεστεί σε έναν διαδικτυακό εξυπηρετητή ώστε να είναι εφικτή η απομακρυσμένη χρήση της.

8 Βιβλιογραφία

- [1] DSSLab-NTUA 2021, «Συστήματα Αποφάσεων -Εισαγωγή,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 3 08 2021].
- [2] QuIC Solutions, Inc, «What is Operations Research,» QuIC Solutions, Inc, [Ηλεκτρονικό]. Available: www.hsor.org. [Πρόσβαση 29 11 2021].
- [3] J. F. McCLOSKEY, «BRITISH OPERATIONAL RESEARCH IN WORLD WAR II,» *INFORMS*, pp. 453-470, 1987.
- [4] H. Luss και M. Rosenwein, «Operations Research applications: Opportunities and accomplishments,» *European Journal of Operational Research*, pp. 220-244, 1997.
- [5] K. B. Zandin, «87. PRINCIPLES AND APPLICATIONS OF OPERATIONS RESEARCH,» σε *Maynard's Industrial Engineering Handbook, Fifth Edition*, The McGraw-Hill Companies, Inc., 2001.
- [6] DSSLab-NTUA, «Συστήματα Αποφάσεων,» 2021. [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 18 08 2021].
- [7] DSSLab-NTUA, «02. Δένδρα Αποφάσεων - Ασκήσεις,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 02 07 2021].
- [8] DSSLab-NTUA, «07. Γραμμικός Προγραμματισμός - Ασκήσεις,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 17 07 2021].
- [9] DSSLab-NTUA, «Επαναληπτικές Ασκήσεις - Γραμμικός Προγραμματισμός,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 17 07 2021].
- [10] DSSLab-NTUA, «04. Δυναμικός Προγραμματισμός - Ασκήσεις,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 20 07 2021].
- [11] D. L. Keefer, J. Corner και C. W. Kirkwood, «Decision Analysis Applications in the Operations Research Literature, 1990-1999,» 2000.
- [12] DSSLab-NTUA, «Συστήματα Αποφάσεων - Δέντρα Αποφάσεων,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 26 06 2021].

- [13 «What is a Decision Tree Diagram,» Lucid Software Inc., 2021. [Ηλεκτρονικό].
] Available: <https://www.lucidchart.com/pages/decision-tree>. [Πρόσβαση 22 12 2020].
- [14 K. Pazek και C. Rozman, «DECISION MAKING UNDER CONDITIONS OF UNCERTAINTY IN
] AGRICULTURE: A CASE STUDY OF OIL CROPS,» Maribor, 2009.
- [15 J. B. Davis, D. W. Hands και U. Maki, *The Handbook of Economic Methodology*,
] Cheltenham, ENG: Edward Elgar Publishing, 1998.
- [16 DSSLab-NTUA, «05. Γραμμικός Προγραμματισμός,» [Ηλεκτρονικό]. Available:
] https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 15 07 2021].
- [17 R. James και L. Scott, «Using the Graphical Method to Solve Linear Programs,»
] *Performance Excellence in the Wood Products Industry*, p. 24, October 1998.
- [18 DSSLab-NTUA, «06. Η μέθοδος SIMPLEX,» [Ηλεκτρονικό]. Available:
] https://academics.epu.ntua.gr/sites/default/files/2021-07/%CE%A0%CE%B1%CF%81%CE%BF%CF%85%CF%83%CE%AF%CE%B1%CF%83%CE%B7_SIMPLEX.pdf. [Πρόσβαση 29 07 2021].
- [19 R. Bronson και G. Naadimuthu, «Γραμμικός Προγραμματισμός: Η μέθοδος Simplex και
] η δυϊκή μέθοδος Simplex,» σε *ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ Δεύτερη αμερικάνικη έκδοση*), ΚΛΕΙΔΑΡΙΘΜΟΣ, 2010, pp. 44-67.
- [20 R. Bronson και G. Naadimuthu, «Γραμμικός Προγραμματισμός: Βασικές έννοιες,» σε
] *ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ Δεύτερη αμερικάνικη έκδοση*, ΚΛΕΙΔΑΡΙΘΜΟΣ, 2010, pp. 29-43.
- [21 E. V. Denardo, *Dynamic Programming: Models and Applications*, Courier Corporation,
] 2012.
- [22 R. Bronson και G. Naadimuthu, «Ανάλυση Δικτύων,» σε *ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ
] Δεύτερη αμερικανική έκδοση*, ΚΛΕΙΔΑΡΙΘΜΟΣ, 2010, pp. 243-258.
- [23 R. Bronson και G. Naadimuthu, «Δυναμικός προγραμματισμός,» σε *ΕΠΙΧΕΙΡΗΣΙΑΚΗ
] ΕΡΕΥΝΑ Δεύτερη αμερικάνικη έκδοση*, ΚΛΕΙΔΑΡΙΘΜΟΣ, 2010, pp. 371-397.
- [24 H. A. Lauren και D. B. Dunson, «Approximate Dynamic Programming for Storage
] Problems,» Bellevue, Washington, USA, 2011.
- [25 «Decision tree maker,» [Ηλεκτρονικό]. Available:
] <https://www.lucidchart.com/pages/examples/decision-tree-maker>. [Πρόσβαση 01 08 2021].
- [26 «Lucid,» [Ηλεκτρονικό]. Available: <https://lucid.app/>. [Πρόσβαση 23 06 2021].
]

- [27 vik_31415, «Linear Programming Solver,» [Ηλεκτρονικό]. Available:
] <https://www.wolframalpha.com/widgets/view.jsp?id=daa12bbf5e4daec7b363737d6d496120>. [Πρόσβαση 28 04 2021].
- [28 «WolframAlpha: Computational Intelligence,» Wolfram, 2021. [Ηλεκτρονικό].
] Available: <https://www.wolframalpha.com/>. [Πρόσβαση 24 04 2021].
- [29 V. LAZARIDIS, K. PAPARRIZOS, N. SAMARAS και A. SIFALERAS, «Visual LinProg: A Web-
] based Educational Software for Linear Programming,» Wiley Periodicals Inc., 2007.
- [30 «Online Calculator,» [Ηλεκτρονικό]. Available: <https://linprog.com/main-dynamic-programming>. [Πρόσβαση 12 08 2021].
- [31 F. Miretti, D. Misul και E. Spessa, «DynaProg: Deterministic Dynamic Programming
] solver for finite horizon multi-stage decision problems,» Elsevier B.V., Torino, Italy, 2021.
- [32 The MathWorks, Inc., «MATLAB,» The MathWorks, Inc., 1994. [Ηλεκτρονικό].
] Available: <https://www.mathworks.com/products/matlab.html>. [Πρόσβαση 27 06 2021].
- [33 «Visual Studio Code,» Microsoft, 2021. [Ηλεκτρονικό]. Available:
] <https://code.visualstudio.com/>. [Πρόσβαση 23 01 2021].
- [34 «Developer Survey Results 2019,» Stack Overflow, [Ηλεκτρονικό]. Available:
] <https://insights.stackoverflow.com/survey/2019>. [Πρόσβαση 20 8 2021].
- [35 «Python,» Python Software Foundation, 2001-2021. [Ηλεκτρονικό]. Available:
] <https://www.python.org/>. [Πρόσβαση 17 03 2021].
- [36 «django,» Django Software Foundation and individual contributors, 2005-2021.
] [Ηλεκτρονικό]. Available: <https://www.djangoproject.com/>. [Πρόσβαση 21 06 2021].
- [37 «Django introduction,» Mozilla and individual contributors, 2005-2021. [Ηλεκτρονικό].
] Available: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>. [Πρόσβαση 11 06 2021].
- [38 «About,» The PostgreSQL Global Development Group, 1996-2021. [Ηλεκτρονικό].
] Available: <https://www.postgresql.org/about/>.
- [39 «How Do HTML, CSS and JavaScript Work Together?,» ITonlinelearning Limited, 11
] May 2017. [Ηλεκτρονικό]. Available: <https://www.itonlinelearning.com/blog/how-do-html-css-and-javascript-work-together/>. [Πρόσβαση 17 07 2021].
- [40 «vis.js,» vis.js community, [Ηλεκτρονικό]. Available: <https://visjs.org/>. [Πρόσβαση 12
] 03 2021].

- [41 «Network,» vis.js community, [Ηλεκτρονικό]. Available: <https://visjs.github.io/vis-network/docs/network/>. [Πρόσβαση 02 07 2021].
- [42 «Chart.js,» [Ηλεκτρονικό]. Available: <https://www.chartjs.org/>. [Πρόσβαση 14 08 2021].
- [43 «Black Dashboard Django,» Creative Tim, [Ηλεκτρονικό]. Available: <https://www.creative-tim.com/product/black-dashboard-django>. [Πρόσβαση 28 07 2021].
- [44 «Optimization and root finding (scipy.optimize),» The SciPy community, 2008-2021. [Ηλεκτρονικό]. Available: <https://docs.scipy.org/doc/scipy/reference/optimize.html>. [Πρόσβαση 22 04 2021].
- [45 DSSLab-NTUA, «ΠΑΑ - Π8. Σχέσεις Υπεροχής - PROMETHEE,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/multicriteria_decision_analysis. [Πρόσβαση 22 08 2021].
- [46 DSSLab-NTUA, «ΠΑΑ - Π7. Σχέσεις Υπεροχής - Electre I & Tri,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/multicriteria_decision_analysis. [Πρόσβαση 22 08 2021].
- [47 DSSLab-NTUA, «ΠΑΑ - Π4. Η μέθοδος VIKOR,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/multicriteria_decision_analysis. [Πρόσβαση 23 08 2021].
- [48 DSSLab-NTUA, «11. Αποφάσεις σε συνθήκες αβεβαιότητας,» [Ηλεκτρονικό]. Available: https://academics.epu.ntua.gr/en/courses/desicion_systems. [Πρόσβαση 2 08 2021].