



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

**ΒΕΛΤΙΣΤΗ ΑΝΑΘΕΣΗ ΔΙΑΦΘΡΩΜΕΝΩΝ ΣΕ
ΔΙΑΦΟΡΕΤΙΚΟΥΣ ΤΟΜΕΙΣ ΔΙΠΛΩΜΑΤΙΚΩΝ ΕΡΓΑΣΙΩΝ
ΥΠΟ ΑΥΣΤΗΡΗ ΠΡΟΤΕΡΑΙΟΤΗΤΑ**
υπό

ΑΠΟΣΤΟΛΟΥ ΤΕΡΛΕΞΗ

Διπλωματούχου Μηχανολόγου Μηχανικού Πανεπιστημίου Θεσσαλίας 2022

Μεταπτυχιακή Εργασία

Υπεβλήθη για την εκπλήρωση μέρους των
απαιτήσεων για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης

Βόλος, 2025

© 2025 Απόστολος Τερλεξής

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής (Επιβλέπων)	Δρ. Γεώργιος Κοζανίδης Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Δεύτερος Εξεταστής	Δρ. Δημήτριος Παντελής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Τρίτος Εξεταστής	Δρ. Ιωάννης Λυχναρόπουλος Μέλος Ε.ΔΙ.Π, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Καταρχήν, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή, κ. Γεώργιο Κοζανίδη, για την πολύτιμη καθοδήγηση που μου παρείχε, καθώς και για την βοήθεια που μου προσέφερε στα προβλήματα που αντιμετώπισα. Επιπλέον, θα ήθελα να ευχαριστήσω τα μέλη της επιτροπής, κ. Παντελή Δημήτριο και κ. Ιωάννη Λυχναρόπουλο, για την ανάγνωση της εργασίας μου, καθώς και για τις πολύτιμες υποδείξεις που μου έκαναν. Επίσης, θα ήθελα να αναφερθώ στους φίλους και στην κοπέλα μου, Κωνσταντίνο Ντελή, Κωνσταντίνο Τσουμπαριώτη, Γεώργιο Παπαδοπεράκη και Κασσιανή Κοτζαθανάση, που για ακόμη μία φορά στάθηκαν δίπλα μου κατά την διάρκεια αυτής της διπλωματικής εργασίας. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου, Ιωάννα και Ιωάννη Τερλεξή και την αδερφή μου, Μαρία Τερλεξή, που σε όλη την διάρκεια των σπουδών μου, με στήριζαν και με βοηθούσαν σε όλες τις δυσκολίες που παρουσιάστηκαν.

ΒΕΛΤΙΣΤΗ ΑΝΑΘΕΣΗ ΔΙΑΦΘΡΩΜΕΝΩΝ ΣΕ ΔΙΑΦΟΡΕΤΙΚΟΥΣ ΤΟΜΕΙΣ ΔΙΠΛΩΜΑΤΙΚΩΝ ΕΡΓΑΣΙΩΝ ΥΠΟ ΑΥΣΤΗΡΗ ΠΡΟΤΕΡΑΙΟΤΗΤΑ

ΑΠΟΣΤΟΛΟΣ ΤΕΡΛΕΞΗΣ

Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας, 2025

Επιβλέπων Καθηγητής: Δρ. Γεώργιος Κοζανίδης,
Αναπληρωτής Καθηγητής Μεθόδων Βελτιστοποίησης

Περίληψη

Οι αλγόριθμοι στην σημερινή εποχή αποτελούν ένα από τα πιο βασικά εργαλεία στην επίλυση των προβλημάτων, καθώς χρησιμοποιούνται τόσο σε απλά καθημερινά προβλήματα, όσο και σε πιο πολύπλοκα υπολογιστικά προβλήματα. Από την άλλη μεριά, η μοντελοποίηση προβλημάτων είναι ένα ευρύ πεδίο, το οποίο αποτελείται από την αναγνώριση, αναπαράσταση και επίλυση προβλημάτων. Τα παραπάνω δύο εργαλεία επίλυσης προβλημάτων συνδυάζονται στην σημερινή εποχή και χρησιμοποιούνται όλο και περισσότερο σε διάφορες κατηγορίες, όπως στην επιχειρησιακή έρευνα, στην τεχνητή νοημοσύνη και στην βελτιστοποίηση προβλημάτων.

Το πρόβλημα που παρουσιάζεται στην παρούσα διπλωματική επιλύεται με τον παραπάνω τρόπο, δηλαδή στην αρχή μορφοποιείται ένα μαθηματικό μοντέλο για την μοντελοποίησή του και μετά αναπτύσσεται ένας αλγόριθμος για την επίλυσή του. Το πρόβλημα αφορά την ανάπτυξη ενός αλγορίθμου βελτιστοποίησης, μέσω του οποίου θα γίνεται η ανάθεση διπλωματικών εργασιών στο τμήμα Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας.

Αρχικά, αναλύεται η δομή του προβλήματος, παρουσιάζοντας τις μεταβλητές απόφασης και τους περιορισμούς. Στην συνέχεια, περιγράφονται οι δύο μέθοδοι που θα χρησιμοποιηθούν για την αντιμετώπισή του, καθώς επίσης και ο κώδικας που χρησιμοποιείται για την επίλυση του προβλήματος. Τέλος, παρατίθενται διάφοροι πίνακες και διαγράμματα που χρησιμοποιούνται για να συγκριθούν οι δύο παραλλαγές του προβλήματος, ενώ στο τέλος παρουσιάζεται και ένα πολύ ενδιαφέρον παράδειγμα που θα κάνει πλήρως κατανοητή την λογική του αλγορίθμου που παρουσιάζεται.

Optimal assignment of diploma theses associated with different divisions subject to strict priority.

APOSTOLOS TERLEXIS

Department of Mechanical Engineering, University of Thessaly, 2025

Supervisor: Dr George Kozanidis

Associate Professor of Optimization Methods

Abstract

Algorithms today constitute one of the most fundamental tools for problem solving, as they are applied to a wide range of cases, from simple everyday problems to highly complex computational challenges. On the other hand, mathematical modeling is a broad field that encompasses the identification, representation, and resolution of problems. These two problem-solving tools are increasingly being combined and applied across various domains, such as operational research, artificial intelligence, and optimization.

The problem addressed in this thesis is approached as follows: it is first modeled and then solved using an algorithm. Specifically, the objective is the development of an optimization algorithm for the automatic assignment of diploma theses in the Department of Mechanical Engineering at the University of Thessaly.

Initially, the structure of the problem is analyzed in order to present the decision variables and the constraints. Then, the two problem variations are described, along with the code implemented for their solution. Finally, various tables and charts are presented to compare them and the thesis concludes with a particularly illustrative example that makes the logic of the proposed algorithm fully comprehensible.

Πίνακας Περιεχομένων

Κεφάλαιο 1.ΕΙΣΑΓΩΓΗ	1
1.1 Κίνητρο και Υπόβαθρο	1
1.2 Βιβλιογραφική Ανασκόπηση	1
1.3 Οργάνωση Μεταπτυχιακής Εργασίας	3
Κεφάλαιο 2.ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΑΝΑΘΕΣΗΣ ΔΙΠΛΩΜΑΤΙΚΩΝ ΕΡΓΑΣΙΩΝ	4
2.1 Παρουσίαση του προβλήματος	4
2.2 Αναφορά μεταβλητών απόφασης, περιορισμών και παραλλαγών	5
2.3 Παρουσίαση κώδικα	7
2.4 Συμπεράσματα	9
Κεφάλαιο 3. ΣΥΓΚΡΙΣΗ ΤΩΝ 2 ΠΑΡΑΛΛΑΓΩΝ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	10
3.1 Ποσοστό ανάθεσης	10
3.2 Ποσοστό ανάθεσης ανά σειρά προτίμησης φοιτητή	12
3.3 Χρονομέτρηση επίλυσης	17
3.4 Ένα πολύ ενδιαφέρον παράδειγμα.....	19
Κεφάλαιο 4.ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΡΟΤΑΣΕΙΣ.....	21
Βιβλιογραφία	23
ΠΑΡΑΡΤΗΜΑ	24

Κατάλογος Πινάκων

Πίνακας 1: Ποσοστό Ανάθεσης ανά Αριθμό Διπλωματικών για $\Phi = 20$	10
Πίνακας 2: Ποσοστό Ανάθεσης ανά Αριθμό Διπλωματικών για $\Phi = 30$	11
Πίνακας 3: Ποσοστό Ανάθεσης ανά Αριθμό Φοιτητών για $\Delta = 20$	11
Πίνακας 4: Ποσοστό Ανάθεσης ανά Αριθμό Φοιτητών για $\Delta = 30$	11
Πίνακας 5: Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό διπλωματικών για αριθμό φοιτητών = 20, πρώτη παραλλαγή.....	12
Πίνακας 6: Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό διπλωματικών για αριθμό φοιτητών = 20, δεύτερη παραλλαγή	13
Πίνακας 7: Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό διπλωματικών για αριθμό φοιτητών = 30, πρώτη παραλλαγή.....	13
Πίνακας 8: Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό διπλωματικών για αριθμό φοιτητών = 30, δεύτερη παραλλαγή.....	14
Πίνακας 9: Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό φοιτητών για αριθμό διπλωματικών = 20, πρώτη παραλλαγή.....	14
Πίνακας 10: Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό φοιτητών για αριθμό διπλωματικών = 20, παραλλαγή	15
Πίνακας 11: Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό φοιτητών για αριθμό διπλωματικών = 30, πρώτη παραλλαγή.....	16
Πίνακας 12: Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό φοιτητών για αριθμό διπλωματικών = 30, δεύτερη παραλλαγή	16

Κατάλογος Διαγραμμάτων

Διάγραμμα 1: Χρόνος επίλυσης πρώτης παραλλαγής ανά αριθμό φοιτητών/διπλωματικών 18

Διάγραμμα 2: Χρόνος επίλυσης δεύτερης παραλλαγής ανά αριθμό φοιτητών/διπλωματικών
..... 18

Κεφάλαιο 1. ΕΙΣΑΓΩΓΗ

1.1 Κίνητρο και Υπόβαθρο

Στην σημερινή εποχή ο τρόπος ανάθεσης διπλωματικών εργασιών είναι διαφορετικός από πανεπιστήμιο σε πανεπιστήμιο. Σε κάποια ιδρύματα υπάρχουν τράπεζες θεμάτων όπου εισέρχεται ο φοιτητής και δηλώνει τις προτιμήσεις του, ενώ σε κάποια άλλα πανεπιστήμια, είναι σύνηθες το φαινόμενο να προτείνει ο ίδιος ο φοιτητής σε έναν καθηγητή ένα θέμα το οποίο θα μπορούσε να αναπτύξει ως διπλωματική εργασία. Κάποιοι άλλοι τρόποι που μπορεί να γίνει μία ανάθεση διπλωματικής εργασίας σε έναν φοιτητή, είναι η ένταξή του σε μία ερευνητική ομάδα, ή να γίνει μία ανάθεση που μπορεί να προκύψει από μία ενδιαφερόμενη επιχείρηση.

Στην παρούσα διπλωματική εργασία προτείνεται ένα νέο μοντέλο ανάθεσης διπλωματικών εργασιών για το τμήμα Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας. Για την δημιουργία του παραπάνω μοντέλου θα αναπτυχθεί ένας αλγόριθμος βελτιστοποίησης ανάθεσης διπλωματικών εργασιών. Πιο συγκεκριμένα, ο αλγόριθμος θα κάνει ανάγνωση των δεδομένων για τον τομέα των φοιτητών, τον τομέα των διπλωματικών, και τις επιθυμίες του κάθε φοιτητή. Στην συνέχεια σύμφωνα με τους περιορισμούς που θα ορίσουμε παρακάτω, και έχοντας την δυνατότητα να επιλυθεί με 2 διαφορετικές συνθήκες, θα εκτυπώνει σε ένα αρχείο τις αναθέσεις του κάθε φοιτητή, το ποσοστό των φοιτητών που τους έγινε ανάθεση, τα ποσοστά αναθέσεων των προτιμήσεων των φοιτητών και τέλος το χρόνο που χρειάστηκε ο αλγόριθμος έτσι ώστε να κάνει τις αναθέσεις.

1.2 Βιβλιογραφική Ανασκόπηση

Στην σημερινή εποχή, έχουν υπάρξει αρκετές μελέτες πάνω στον τομέα των αναθέσεων διπλωματικών εργασιών. Οι Zhao, Liu και Yang [1], στο άρθρο τους έκαναν μία έρευνα στην οποία συμμετείχαν 88 φοιτητές με σκοπό να βρεθούν τα ενδιαφέροντα των φοιτητών, έτσι ώστε να καθοδηγηθούν οι διδάσκοντες στην καλύτερη προετοιμασία θεμάτων διατριβής. Τα αποτελέσματα της μελέτης είναι ότι:

- 1) Στην επιλογή ενός θέματος διατριβής, σημαντικό ρόλο έχει η μελέτη πολλών ερευνητικών άρθρων.
- 2) Οι καθηγητές χρειάζεται να αυξήσουν τον αριθμό των θεμάτων και να μεγαλώσουν το εύρος τους.
- 3) Η εξισορρόπηση της δυσκολίας των θεμάτων κρίνεται αναγκαία.

Από την πλευρά τους, οι Sanchez- Anguix, Chalumuri, Alberola και Aydogan [2], αναφέρουν ότι το πρόβλημα της κατανομής των φοιτητών σε καθηγητές για την εκπόνηση διπλωματικών εργασιών είναι ένα αρκετά περίπλοκο πρόβλημα που εξαρτάται από τον αριθμό των φοιτητών και των διαθέσιμων καθηγητών, τις προτιμήσεις των φοιτητών και των καθηγητών και τους περιορισμούς φόρτου εργασίας του προσωπικού. Έτσι, αναπτύσσουν ένα μοντέλο τεχνητής νοημοσύνης που έχει ως στόχο την κατανομή των φοιτητών σε επιβλέποντες καθηγητές για την διπλωματική εργασία τους. Το εν λόγω μοντέλο, λαμβάνει υπόψη τις προτιμήσεις των φοιτητών και των καθηγητών, ενώ επίσης φροντίζει να εξισορροπείται ο φόρτος εργασίας μεταξύ των καθηγητών.

Ακόμα μία μελέτη πάνω στην ανάθεση διπλωματικών εργασιών έκαναν οι Serek και Zhararov [3]. Αυτοί στο άρθρο τους ανέπτυξαν έναν αλγόριθμο όπου βελτιστοποιείται η διαδικασία ανάθεσης, μεγιστοποιώντας την ικανοποίηση των προτιμήσεων των φοιτητών και των καθηγητών λαμβάνοντας υπόψη την ψυχολογική συμβατότητα μεταξύ των δύο.

Ένας γενετικός αλγόριθμός αναπτύσσεται και από τους Harper, Valter de Senna, Vieira και Shahani [4]. Στην συγκεκριμένη μελέτη οι φοιτητές δηλώνουν εκ των προτέρων τις προτιμήσεις τους από μία λίστα διαθέσιμων διπλωματικών. Στην συνέχεια, γίνεται μία σύγκριση ανάμεσα στον αλγόριθμο που αναπτύχθηκε και με μία βέλτιστη προσέγγιση ακέραιου προγραμματισμού. Το πλεονέκτημα του αλγορίθμου, είναι ότι παράγει αρκετές εφικτές λύσεις ανάθεσης εργασιών.

Οι Kwanashie, Irving, Manlove και Colin [5], επίσης παρουσιάζουν έναν αλγόριθμο αντιστοίχισης φοιτητών με διπλωματικές εργασίες. Αρχικά παρουσιάζουν έναν αλγόριθμο που προσπαθεί να αντιστοιχίσει τους φοιτητές με τις υψηλότερες προτιμήσεις τους, ενώ στην συνέχεια κάνει μία σύγκριση με ένα αλγόριθμο που έχει ως στόχο την ελαχιστοποίηση των αντιστοιχίσεων των φοιτητών σε προτιμήσεις που είναι στις τελευταίες προτεραιότητες τους.

Επιπλέον οι Sanchez-Anguix, Chalumuri και Julian [6] από κοινού αλλά και σε συνεργασία με τον Aydoğan [7], σε δύο διαφορετικά άρθρα αναπτύσσουν δύο αλγόριθμους Pareto για την ανάθεση διπλωματικών εργασιών σε φοιτητές με βάση τις προτιμήσεις των φοιτητών και των καθηγητών.

Τέλος οι Laporte και Desroches [8], αναπτύσσουν ένα πιο γενικό υπολογιστικό πρόγραμμα που αφορά την δημιουργία τμημάτων για την παρακολούθηση μαθημάτων ενός Πανεπιστημίου. Οι μαθητές δηλώνουν τα μαθήματα που θέλουν να παρακολουθήσουν αφού ήδη γνωρίζουν την ώρα διεξαγωγής κάθε μαθήματος και τον διδάσκοντα. Μετά μέσω του προγράμματος δημιουργούνται τμήματα που εξισορροπείται ο αριθμός των φοιτητών σε κάθε τμήμα και εξασφαλίζει ικανοποιητικά προγράμματα για τους φοιτητές.

1.3 Οργάνωση Μεταπτυχιακής Εργασίας

Το υπόλοιπο αυτής της μεταπτυχιακής εργασίας χωρίζεται σε δύο ενότητες που καταλαμβάνουν τα Κεφάλαιο 2 - 3, αντίστοιχα. Συγκεκριμένα:

Στο Κεφάλαιο 2, γίνεται παρουσίαση του προβλήματος που καλούμαστε να επιλύσουμε, ορισμός των μεταβλητών απόφασης, των περιορισμών, των παραλλαγών του προβλήματος, παρουσίαση του κώδικα και τέλος αναφορά για τα συμπεράσματα που μπορεί να προκύψουν κατά την παρουσίαση του κώδικα.

Στο Κεφάλαιο 3, γίνεται η σύγκριση των 2 παραλλαγών αρχικά με βάση το ποσοστό ανάθεσης, μετά με το ποσοστό ανάθεσης ανά σειρά προτίμησης φοιτητή και στην συνέχεια σύμφωνα με τον χρόνο επίλυσης. Τέλος, παρουσιάζεται ένα πολύ ενδιαφέρον παράδειγμα, που θα βοηθήσει στο να γίνει πλήρως κατανοητή η λειτουργία του κώδικα.

Τέλος, στο Κεφάλαιο 4, παρουσιάζονται τα συμπεράσματα της διπλωματικής εργασίας.

Κεφάλαιο 2. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΑΝΑΘΕΣΗΣ ΔΙΠΛΩΜΑΤΙΚΩΝ ΕΡΓΑΣΙΩΝ

Στο κεφάλαιο αυτό θα παρουσιαστεί και θα αναλυθεί το πρόβλημα ανάθεσης διπλωματικών εργασιών, καθώς επίσης και ο αλγόριθμος βελτιστοποίησης του προβλήματος αυτού.

Το συγκεκριμένο κεφάλαιο οργανώνεται ως εξής. Στην Ενότητα 2.1 θα γίνει η παρουσίαση του προβλήματος. Στην Ενότητα 2.2 θα γίνει η μοντελοποίησή του και ο ορισμός των μεταβλητών απόφασης αλλά και των περιορισμών. Στην Ενότητα 2.3 θα γίνει η επεξήγηση του κώδικα. Τέλος, στην ενότητα 2.4 θα παρουσιαστούν κάποια συμπεράσματα που θα προκύψουν μετά την παρουσίαση του κώδικα.

2.1 Παρουσίαση του προβλήματος

Το πρόβλημα το οποίο θα επιλυθεί έχει την παρακάτω δομή. Μέσα από ένα σύστημα, οι φοιτητές έχουν την δυνατότητα να δηλώσουν με σειρά προτίμησης, τις επιθυμίες που έχουν για τις διπλωματικές που θέλουν να τους ανατεθούν. Κάθε φοιτητής και κάθε διπλωματική, έχουν ένα δικό τους id το οποίο είναι μοναδικό για οποιαδήποτε διπλωματική και φοιτητή. Επίσης κάθε φοιτητής και κάθε διπλωματική ανήκουν σε έναν από τους 3 τομείς που υπάρχουν στο τμήμα Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας. Τέλος, πρέπει να συμπεριληφθεί και η περίπτωση όπου μία διπλωματική μπορεί να ανατεθεί σε δύο φοιτητές. Παράλληλα, πρέπει να είναι σίγουρο, ότι θα ισχύσει ένας κανόνας, που αφορά την δυνατότητα των φοιτητών να τους γίνεται ανάθεση διπλωματικής σε διαφορετικό τομέα από αυτόν που ανήκουν. Ο κανόνας είναι ο παρακάτω. Ένας φοιτητής, μπορεί να λάβει μία διπλωματική εργασία από έναν άλλον τομέα από αυτόν που ανήκει, μόνο αν η συγκεκριμένη διπλωματική δεν ανατεθεί σε κάποιον φοιτητή που ανήκει στον ίδιο τομέα, με τον τομέα της διπλωματικής. Τέλος, μέσω του συστήματος δήλωσης των διπλωματικών εργασιών, βγαίνει μία σειρά κατάταξης (με βάση τις βαθμολογίες που έχει ο κάθε φοιτητής), και βάσει αυτής της σειράς θα γίνει η ανάθεση των διπλωματικών εργασιών. Δηλαδή, πρέπει να γίνει προσπάθεια έτσι ώστε να ικανοποιηθεί η πρώτη επιθυμία, του φοιτητή που είναι πρώτος στην παραπάνω κατάταξη. Μετά του δεύτερου, μέχρι να ολοκληρωθεί η λίστα των φοιτητών που

δήλωσαν προτιμήσεις μέσω του συστήματος. Πάντα όμως πρέπει να ισχύει ο περιορισμός για τις αναθέσεις φοιτητών που δεν ανήκουν στον ίδιο τομέα με τις διπλωματικές που δηλώσαν.

2.2 Αναφορά μεταβλητών απόφασης, περιορισμών και μεθόδων επίλυσης

Αρχικά, για να επιλυθεί το πρόβλημα, πρέπει πρώτα να οριστούν οι μεταβλητές απόφασης. Θα οριστεί μόνο μία μεταβλητή απόφασης, η Y_{ij} , η οποία θα είναι δυαδική μεταβλητή και θα παίρνει την τιμή 1 όταν γίνεται ανάθεση μίας διπλωματικής σε έναν φοιτητή, και 0 όταν δεν γίνεται ανάθεση μίας διπλωματικής σε έναν φοιτητή. Ο δείκτης i είναι ο μοναδικός αριθμός id που έχει ο κάθε φοιτητής από το Σύστημα δηλώσεων διπλωματικών εργασιών, ενώ ο δείκτης j αφορά τον αντίστοιχο αριθμό id που έχουν οι διπλωματικές εργασίες. Οπότε πρέπει να οριστεί μία μεταβλητή απόφασης, για κάθε μία επιθυμία που θα δηλώσει ο κάθε φοιτητής στο σύστημα.

Στην συνέχεια πρέπει να οριστούν οι περιορισμοί του προβλήματος. Ο πρώτος περιορισμός, αφορά το κομμάτι τον φοιτητών και πιο συγκεκριμένα, ότι ένας φοιτητής μπορεί να αναλάβει το πολύ μία διπλωματική εργασία. Ο παραπάνω περιορισμός διατυπώνεται ως εξής:

$j = \text{το } id \text{ διπλωματικής της λίστας επιθυμιών φοιτητή}$

$$\sum_{j=1 \text{ ο } id \text{ διπλωματικής της λίστας επιθυμιών φοιτητή}} Y_{ij} \leq 1 \quad \forall \text{ } id \text{ φοιτητή}$$

Στην συνέχεια, ο επόμενος περιορισμός, είναι ότι κάθε διπλωματική, μπορεί να ανατεθεί μόνο σε έναν ή δύο φοιτητές, ανάλογα με το αν μία διπλωματική μπορεί να οριστεί μόνο σε έναν φοιτητή, ή μπορεί να οριστεί και σε 2 φοιτητές. Για να γίνει αυτό, χρειάζεται να οριστεί μία παράμετρος την οποία την ονομάζουμε $M(j)$ και ισούται με τον αριθμό των φοιτητών που μπορεί να ανατεθεί. Ο παραπάνω περιορισμός διατυπώνεται ως εξής:

$j = \text{το } id \text{ φοιτητή που έχει δηλώσει την διπλωματική}$

$$\sum_{j=1 \text{ ο } id \text{ φοιτητή που έχει δηλώσει την διπλωματική}} Y_{ij} \leq M(j) \quad \forall \text{ } id \text{ διπλωματικής}$$

Τέλος, πρέπει να εκφραστεί ο περιορισμός αναφορικά με τους φοιτητές που δηλώνουν στις επιθυμίες τους διπλωματικές από διαφορετικούς τομείς από αυτούς που βρίσκονται οι ίδιοι. Για να γίνει αυτό, πρέπει να εντοπιστεί κάθε διπλωματική που έχει

δηλωθεί από δυο ή περισσότερα άτομα. Στην συνέχεια να δημιουργηθούν όλα τα πιθανά ζεύγη φοιτητών, έτσι ώστε αφού συγκριθούν μεταξύ τους να αποφασιστεί ποιος φοιτητής θα έχει προτεραιότητα για την ανάθεση της διπλωματικής.

Για την σύγκριση των ζευγών των φοιτητών που δηλώσαν μία διπλωματική, πρέπει αρχικά να παρατηρηθεί αν κάποιος φοιτητής είναι στον ίδιο τομέα με τον τομέα της διπλωματικής. Σε περίπτωση, που ένας από τους δύο βρίσκεται στον ίδιο τομέα με την διπλωματική, τότε πρέπει να διασφαλιστεί ότι αυτός θα έχει προτεραιότητα σε σχέση με τον φοιτητή που δεν βρίσκεται στον ίδιο τομέα με την διπλωματική. Όταν όμως, και οι δύο φοιτητές βρίσκονται στον ίδιο τομέα με την διπλωματική, ή και οι δύο φοιτητές δεν βρίσκονται στον ίδιο τομέα με την διπλωματική, προτεραιότητα έχει ο φοιτητής που βρίσκεται πιο ψηλά στην σειρά κατάταξης. Έστω ότι ο φοιτητής που είναι ο ισχυρός του ζεύγους έχει δείκτη 1, ο ανίσχυρος του ζεύγους έχει δείκτη 2 και η διπλωματική που έχουν δηλώσει και οι δύο έχει δείκτη k. Ο περιορισμός εκφράζεται με τον παρακάτω τρόπο:

$$Y_{2k} \leq Y_{1k} + \sum Y_{1j}$$

Όπου $\sum Y_{1j}$, είναι το σύνολο των διπλωματικών που έχει ορίσει ο φοιτητής 1 πιο ψηλά στις προτεραιότητες του από την διπλωματική k.

Με τον παραπάνω περιορισμό εξασφαλίζουμε την προτεραιότητα του ισχυρότερου φοιτητή διότι:

- Αν $Y_{2k} = 1$, τότε για να ισχύει ο περιορισμός θα πρέπει πρώτα ο φοιτητής 1 να έχει ανατεθεί σε μία διπλωματική που είναι πιο ψηλά στις προτιμήσεις του.
- Αν $Y_{1k} = 1$, τότε ο φοιτητής 2 θα μπορεί να ανατεθεί στην διπλωματική k μόνο εάν η συγκεκριμένη διπλωματική μπορεί να ανατεθεί σε 2 φοιτητές και δεν υπάρχει κάποιος άλλος (τρίτος) φοιτητής που να είναι πιο ισχυρός από τον φοιτητή 2. Σε διαφορετική περίπτωση στον φοιτητή 2 δεν θα γίνει ανάθεση της διπλωματικής k.

Η διαδικασία ανάθεσης γίνεται με τον παρακάτω τρόπο:

Με μία επαναληπτική διαδικασία, αρχικά γίνεται η προσπάθεια να ανατεθεί στον πρώτο φοιτητή της κατάταξης η πρώτη του επιθυμία, θέτοντας την μεταβλητή που εκφράζει την παραπάνω επιθυμία ίση με 1. Αν το πρόβλημα παραμένει εφικτό, τότε ο κώδικας προχωράει στο να ελέγξει την εφικτότητα του προβλήματος όταν θέσει την μεταβλητή που εκφράζει την πρώτη επιθυμία του δεύτερου φοιτητή ίση με την μονάδα. Όμως αν παραβιάζεται κάποιος περιορισμός, ο κώδικας θα προχωρήσει θέτοντας την μεταβλητή της δεύτερης επιθυμίας του πρώτου φοιτητή ίση με 1, πριν προχωρήσει στις επιθυμίες του δεύτερου φοιτητή. Αυτή η διαδικασία επαναλαμβάνεται για όλες τις επιθυμίες όλων των φοιτητών. Είναι σημαντικό να αναφερθεί, ότι στην συγκεκριμένη περίπτωση δεν υπάρχει

αντικειμενική συνάρτηση και κάθε φορά εξετάζεται το αν το πρόβλημα είναι εφικτό όταν τίθεται μία μεταβλητή ίση με την μονάδα.

Στον κώδικα όμως δίνεται άλλη μια δυνατότητα. Πιο συγκεκριμένα, εκτός από το πραγματικό πρόβλημα εξετάζεται μία χαλάρωση του προβλήματος, όπου υπάρχουν μόνο οι δύο πρώτοι περιορισμοί και όχι οι περιορισμοί που δηλώνουν προτεραιότητα βάσει κατάταξης ή τομέα και γίνεται μεγιστοποίηση των αναθέσεων. Έτσι, σε αυτήν την περίπτωση υπάρχει αντικειμενική συνάρτηση και έχει την παρακάτω μορφή:

$$\text{MAX } \sum Y_{ij} \forall i, j$$

Αυτό συμβαίνει έτσι ώστε να γίνει μία σύγκριση για το πόσο πολύ επηρεάζει το ποσοστό αναθέσεων ο περιορισμός των τομέων και της ανάθεσης σύμφωνα με την σειρά κατάταξης.

2.3 Παρουσίαση κώδικα

Αρχικά, πρέπει να αναφερθεί ότι ο κώδικας γράφτηκε σε γλώσσα προγραμματισμού Python και πιο συγκεκριμένα στο περιβάλλον της Spyder, στην έκδοση 4.1.5. Όταν ο χρήστης τρέξει τον κώδικα, αρχικά ο κώδικας θα του ζητήσει να δηλώσει τον ακριβή αριθμό φοιτητών και διπλωματικών εργασιών. Στην συνέχεια, ο χρήστης θα έχει την επιλογή να ζητήσει από τον κώδικα να δημιουργήσει ένα τυχαίο αρχείο δεδομένων (πατώντας το 0), ή να διαβάσει ένα έτοιμο αρχείο δεδομένων που είναι ήδη αποθηκευμένο (πατώντας το 1). Το αρχείο ανάγνωσης, που περιέχει τα δεδομένα του προβλήματος δηλαδή, έχει την παρακάτω μορφή:

9 16

366 422 460 546 590 611 682 734 735

74 706 736 737 739 742 782 783 803 811 906 908 909 955 956 970

1 2 3 2 2 1 1 3 1

2 3 1 1 1 3 3 1 2 1 1 1 1 1 2

1 1 2 2 2 1 1 1 1 1 1 1 1 1 2

783 782

74

736 737 739

706

909

803 970

811 906 909 908 955 956 783

742

956

Το συγκεκριμένο αρχείο αφορά 9 φοιτητές και 16 διπλωματικές. Στην πρώτη σειρά, αναγράφονται ο αριθμός των φοιτητών και των διπλωματικών εργασιών. Μετά, υπάρχει μία σειρά κενό και στην συνέχεια αναφέρονται τα id των φοιτητών και των διπλωματικών, έχοντας ανάμεσα τους μια κενή σειρά. Συνεχίζοντας, και αφήνοντας μια κενή σειρά, το αρχείο ανάγνωσης περιέχει τους τομείς τους οποίους βρίσκονται οι φοιτητές και μετά από μία ακόμη κενή σειρά τους τομείς που βρίσκονται οι διπλωματικές (αριθμοί 1,2 ή 3 σύμφωνα με τους 3 τομείς που υπάρχουν στο τμήμα). Μετά αφήνεται άλλη μία κενή σειρά, και στην συνέχεια αναγράφεται ο αριθμός που θα καθορίσει αν μία διπλωματική μπορεί να ανατεθεί μόνο σε ένα άτομο, ή μπορεί να ανατεθεί και σε 2 άτομα (αριθμοί 1 και 2 αντίστοιχα). Τέλος, υπάρχει μία τελευταία κενή σειρά, και το αρχείο τελειώνει αναφέροντας τις επιθυμίες των φοιτητών. Κάθε γραμμή περιέχει τις επιθυμίες του κάθε φοιτητή, από τον πρώτο μέχρι και τον τελευταίο στην κατάταξη, σύμφωνα με τα id των διπλωματικών που δηλώσαν στο σύστημα. Στην περίπτωση που ο χρήστης ζητήσει από το πρόγραμμα να φτιάξει ένα τυχαίο πρόβλημα, λαμβάνεται η υπόθεση ότι οι φοιτητές μπορούν να δηλώσουν μέχρι 8 προτιμήσεις. Τέλος, πρέπει να αναφερθεί ότι σε κάθε γραμμή μπορούν να γραφούν μέχρι 50 χαρακτήρες. Σε περίπτωση που σε ένα πρόβλημα χρειαστεί να γραφούν παραπάνω αριθμοί σε κάποια γραμμή, τότε θα γράφουν οι πρώτοι 50 αριθμοί σε μία γραμμή, και οι υπόλοιποι αριθμοί θα αναφερθούν στην επόμενη γραμμή, χωρίς την προσθήκη κενής γραμμής.

Το επόμενο που θα ζητηθεί από τον χρήστη, είναι να δηλώσει εάν επιθυμεί το πρόβλημα να επιλυθεί με όλους τους περιορισμούς (πατώντας 0), ή εάν επιθυμεί να γίνει μέγιστη ανάθεση διπλωματικών, (πατώντας το 1).

Ολοκληρώνοντας και το παραπάνω βήμα, ο κώδικας επιλύει το πρόβλημα σύμφωνα με τα δεδομένα που του εισήγαγε ο χρήστης, και δημιουργεί ένα αρχείο αποτελεσμάτων που έχει την παρακάτω μορφή:

366 783

422 74

460 736

546 706

590 909

611 803

682 811

734 742

735 956

Σύνολο αναθέσεων: 9

Ποσοστό ανάθεσης: 100.00%

Προτίμηση 1: 9 φοιτητές (100.00%)

Ποσοστό φοιτητών που δεν τους έγινε ανάθεση διπλωματικής: 0.00%

Χρόνος επίλυσης: 0.04 δευτερόλεπτα

Πρώτα, αναφέρονται τα id των φοιτητών, και δίπλα τα id των διπλωματικών που τους ανατέθηκαν. Σε περίπτωση που σε ένα φοιτητή δεν γίνει κάποια ανάθεση, τότε δίπλα από το id του, θα γράφεται η λέξη null και όχι το id κάποιας διπλωματικής. Επίσης αναφέρονται, το σύνολο και το ποσοστό αναθέσεων, τα επιμέρους ποσοστά των προτιμήσεων των φοιτητών που τους έγιναν ανάθεση, το ποσοστό φοιτητών που δεν τους έγινε ανάθεση κάποιας διπλωματικής, και ο χρόνος που χρειάστηκε ο αλγόριθμος για να επιλύσει το πρόβλημα.

2.4 Συμπεράσματα

Σε αυτό το κεφάλαιο αναφέρθηκαν όλα τα δεδομένα του προβλήματος, οι μεταβλητές απόφασής και οι περιορισμοί του. Επίσης, παρουσιάστηκαν οι 2 παραλλαγές του, καθώς και τα δεδομένα και η φύση του αλγόριθμου επίλυσης. Τέλος, αναλύθηκε η μορφή του αρχείου εισόδου, και του αρχείου των αποτελεσμάτων.

Κεφάλαιο 3. ΣΥΓΚΡΙΣΗ ΤΩΝ 2 ΠΑΡΑΛΛΑΓΩΝ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Στο κεφάλαιο αυτό τρέχουμε στον κώδικα διάφορα παραδείγματα, έτσι ώστε να ληφθούν αποτελέσματα, που θα παρουσιαστούν κάποια συμπεράσματα πάνω στις 2 μεθόδους επίλυσης. Θα εξαχθούν διάφορα συμπεράσματα που θα αφορούν τον χρόνο επίλυσης, το ποσοστό ανάθεσης, και το ποσοστό ανάθεσης ανά σειρά προτίμησης φοιτητή. Τέλος, θα παρουσιαστεί ένα πολύ ενδιαφέρον παράδειγμα που θα δείξει πόσο σημαντικός είναι ο περιορισμός προτεραιότητας για διπλωματικές και φοιτητές που ανήκουν σε διαφορετικό τομέα.

3.1 Ποσοστό ανάθεσης

Στην συγκεκριμένη ενότητα, θα γίνει ανάλυση κάποιων παραδειγμάτων πάνω στην σύγκριση του ποσοστού αναθέσεων στις 2 παραλλαγές του προβλήματος. Στα παραδείγματα, αρχικά κρατήθηκε σταθερός ο αριθμός των φοιτητών και μεταβάλαμε των αριθμό των διπλωματικών. Ο αριθμός των φοιτητών πρώτα τοποθετήθηκε στους 20 και στην συνέχεια μεταβλήθηκε στους 30. Για αυτά τα δύο ενδεχόμενα, οι διπλωματικές αυξάνονταν από 10 έως 60 με βήμα 5 και εξήχθησαν τα αποτελέσματα που θα παρουσιαστούν παρακάτω. Στην συνέχεια έγινε το αντίστροφο. Κρατήθηκε ο αριθμός των διπλωματικών σταθερός, πρώτα ίσο με 20 και μετά ίσο με 30, και μεταβάλλονταν ο αριθμός των φοιτητών από 10 έως 60 με βήμα 5. Τα αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες, όπου Π1 είναι τα αποτελέσματα για τα παραδείγματα της πρώτης παραλλαγής και Π2 τα αποτελέσματα για τα παραδείγματα της δεύτερης παραλλαγής:

Φοιτητές =20 :

Δ	10	15	20	25	30	35	40	45	50	55	60
Π1*	60	80	85	95	95	100	90	95	100	100	100
%											

Π2*	60	85	100	100	100	100	100	100	100	100	100
%											

Πίνακας 1 Ποσοστό Ανάθεσης ανά Αριθμό Διπλωματικών για $\Phi = 20$

Φοιτητές = 30:

Δ	10	15	20	25	30	35	40	45	50	55	60
Π1	40	56,67	70	83,33	90	86,67	83,33	93,33	96,67	93,33	100
*%											
Π2	40	56,67	73,33	90	100	100	100	100	100	100	100
*%											

Πίνακας 2 Ποσοστό Ανάθεσης ανά Αριθμό Διπλωματικών για $\Phi = 30$

Διπλωματικές = 20:

Φ	10	15	20	25	30	35	40	45	50	55	60
Π1*	90	100	85	76	73,33	62,86	55	48,89	44	40	36,67
%											
Π2*	100	100	100	88	73,33	62,86	55	48,89	44	40	36,67
%											

Πίνακας 3 Ποσοστό Ανάθεσης ανά Αριθμό Φοιτητών για $\Delta = 20$

Διπλωματικές = 30:

Φ	10	15	20	25	30	35	40	45	50	55	60
Π1*	100	100	95	96	90	88,57	75	68,88	62	58,18	53,33
%											
Π2*	100	100	100	100	100	91,43	80	71,11	64	58,18	53,33
%											

Πίνακας 4 Ποσοστό Ανάθεσης ανά Αριθμό Φοιτητών για $\Delta = 30$

Στους παραπάνω πίνακες, παρατηρείται ότι το μεγαλύτερο ποσοστό ανάθεσης έχει η δεύτερη παραλλαγή του προβλήματος, όπως είναι λογικό, καθώς επιλύεται χωρίς περιορισμούς και με την λογική της μέγιστης ανάθεσης. Παρόλα αυτά, φαίνεται ότι τα δύο ποσοστά σε γενικές γραμμές δεν έχουν μεγάλες αποκλίσεις, κάτι που δηλώνει ότι και η πρώτη παραλλαγή είναι αρκετά καλή. Η μεγαλύτερη απόκλιση παρατηρείται στην περίπτωση όπου οι φοιτητές είναι ίσοι με 30, και οι διπλωματικές ισούνται με 40. Εκεί, παρατηρείται με διαφορά της τάξης του 16,67 %.

3.2 Ποσοστό ανάθεσης ανά σειρά προτίμησης φοιτητή

Με παρόμοιο τρόπο, παρουσιάζονται κάποια παραδείγματα, αλλά τώρα αφορούν την σειρά προτίμησης των φοιτητών, και πιο συγκεκριμένα, το ποσοστό που ικανοποιείται για κάθε προτίμηση των φοιτητών. Εδώ, αρχικά πρέπει να αναφερθεί ότι ελήφθη η απόφαση η φοιτητές να μπορούν να δηλώσουν μέχρι 8 προτιμήσεις. Με την ίδια λογική όπως και στην προηγούμενη ενότητα, παρουσιάζονται αποτελέσματα πρώτα για σταθερό αριθμό φοιτητών, και στην συνέχεια για σταθερό αριθμό διπλωματικών.

Φοιτητές = 20, Πρώτη παραλλαγή:

Δ	1	2	3	4	5	6	7	8	Χωρίς Ανάθεση
10	20	15	15	0	10	0	0	0	40
15	40	20	15	0	5	0	0	0	20
20	50	20	10	0	5	0	0	0	15
25	70	10	10	0	5	0	0	0	5
30	60	15	10	10	0	0	0	0	5
35	80	20	0	0	0	0	0	0	0
40	60	25	5	0	0	0	0	0	10
45	85	10	0	0	0	0	0	0	5
50	95	5	0	0	0	0	0	0	0
55	90	10	0	0	0	0	0	0	0
60	85	15	0	0	0	0	0	0	0

Πίνακας 5 Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό διπλωματικών για αριθμό φοιτητών = 20, πρώτη παραλλαγή

Φοιτητές = 20, Δεύτερη παραλλαγή:

Δ	1	2	3	4	5	6	7	8	Χωρίς Ανάθεση
10	35	5	10	5	0	0	5	0	40
15	40	25	10	10	0	0	0	0	15
20	30	20	25	10	10	0	5	0	0
25	65	20	10	0	0	0	5	0	0
30	30	10	20	15	5	15	0	5	0
35	45	30	10	10	5	0	0	0	0
40	40	25	20	15	0	0	0	0	0
45	40	25	5	10	15	0	0	5	0
50	45	15	10	15	5	10	0	0	0
55	45	25	15	0	15	0	0	0	0
60	40	30	5	10	10	5	0	0	0

Πίνακας 6 Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό διπλωματικών για αριθμό φοιτητών = 20, δεύτερη παραλλαγή

Φοιτητές = 30, Πρώτη παραλλαγή:

Δ	1	2	3	4	5	6	7	8	Χωρίς Ανάθεση
10	20	10	6,67	0	0	3,33	0	0	60
15	13,33	13,33	3,33	10	3,33	6,67	6,67	0	43,33
20	43,33	16,67	10	0	0	0	0	0	30
25	33,33	30	10	0	6,67	3,33	0	0	16,67
30	56,67	26,67	6,67	0	0	0	0	0	10
35	50	20	6,67	10	0	0	0	0	13,33
40	56,67	16,67	3,33	6,67	0	0	0	0	16,67

45	70	20	3,33	0	0	0	0	0	6,67
50	70	16,67	10	0	0	0	0	0	3,33
55	66,67	26,67	0	0	0	0	0	0	6,67
60	73,33	10	10	6,67	0	0	0	0	0

Πίνακας 7 Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό διπλωματικών για αριθμό φοιτητών = 30, πρώτη παραλλαγή

Φοιτητές = 30, Δεύτερη παραλλαγή:

Δ	1	2	3	4	5	6	7	8	Χωρίς Ανάθεση
10	23,33	6,67	3,33	3,33	0	3,33	0	0	60
15	30	10	6,67	0	0	3,33	6,67	0	43,33
20	40	10	10	6,67	3,33	0	3,33	0	26,67
25	26,67	20	6,67	20	10	3,33	0	3,33	10
30	23,33	23,33	20	6,67	16,67	0	6,67	3,33	0
35	56,67	16,67	20	3,33	3,33	0	0	0	0
40	30	23,33	16,67	13,33	3,33	0	3,33	10	0
45	40	30	6,67	6,67	10	3,33	0	3,33	0
50	40	40	13,33	3,33	0	3,33	0	0	0
55	43,33	20	20	6,67	10	0	0	0	0
60	43,33	6,67	20	10	6,67	6,67	6,67	0	0

Πίνακας 8 Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό διπλωματικών για αριθμό φοιτητών = 30, δεύτερη παραλλαγή

Διπλωματικές = 20, Πρώτη μέθοδος παραλλαγή:

Φ	1	2	3	4	5	6	7	8	Χωρίς Ανάθεση
10	60	30	0	0	0	0	0	0	10
15	73,33	20	6,67	0	0	0	0	0	0

20	60	5	10	10	0	0	0	0	15
25	40	8	16	4	8	0	0	0	24
30	26,67	23,33	13,33	6,67	0	3,33	0	0	26,67
35	17,14	17,14	11,43	5,71	5,71	2,86	2,86	0	37,14
40	20	15	5	10	2,5	2,5	0	0	45
45	22,22	15,56	2,22	0	4,44	2,22	0	2,22	51,11
50	16	10	4	8	4	2	0	0	56
55	10,91	10,91	5,45	5,45	3,64	0	3,64	0	60
60	11,67	5	8,33	0	8,33	1,67	1,67	0	63,33

Πίνακας 9 Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό φοιτητών για αριθμό διπλωματικών = 20, πρώτη παραλλαγή

Διπλωματικές = 20, Δεύτερη μέθοδος επίλυσης:

Φ	1	2	3	4	5	6	7	8	Χωρίς Ανάθεση
10	10	30	10	20	10	10	10	0	0
15	66,67	26,67	0	6,67	0	0	0	0	0
20	40	15	20	20	0	5	0	0	0
25	46	16	8	0	4	8	4	0	0
30	30	13,33	16,67	10	3,33	0	0	0	26,67
35	28,57	8,57	14,29	2,86	5,71	2,86	0	0	37,14
40	22,5	17,5	2,5	5	0	5	2,5	0	45
45	37,78	2,22	2,22	6,67	0	0	0	0	51,11
50	22	4	2	6	4	4	2	0	56
55	23,64	5,45	5,45	3,64	1,82	0	0	0	60
60	15	11,67	0	3,33	1,67	5	0	0	63,33

Πίνακας 10 Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό φοιτητών για αριθμό διπλωματικών = 20, δεύτερη παραλλαγή

Διπλωματικές = 30, Πρώτη παραλλαγή:

Φ	1	2	3	4	5	6	7	8	Χωρίς Ανάθεση
10	100	0	0	0	0	0	0	0	0
15	73,33	20	6,67	0	0	0	0	0	0
20	70	15	10	0	0	0	0	0	5
25	76	12	4	0	4	0	0	0	4
30	50	26,67	6,67	0	6,67	0	0	0	10
35	57,17	17,14	5,71	8,57	0	0	0	0	11,43
40	50	15	7,5	2,5	0	0	0	0	25
45	33,33	22,22	11,11	0	2,22	0	0	0	31,11
50	32	4	14	6	2	4	0	0	38
55	20	18,18	9,09	1,82	3,64	1,82	3,64	0	41,82
60	21,67	16,67	5	6,67	1,67	1,67	0	0	46,67

Πίνακας 11 Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό φοιτητών για αριθμό διπλωματικών = 30, πρώτη παραλλαγή

Διπλωματικές = 30, Δεύτερη παραλλαγή:

Φ	1	2	3	4	5	6	7	8	Χωρίς Ανάθεση
10	40	50	10	0	0	0	0	0	0
15	46,67	26,67	6,67	13,33	6,67	0	0	0	0
20	40	20	15	5	5	0	15	0	0
25	68	20	12	0	0	0	0	0	0
30	26,67	30	16,67	10	13,33	3,33	0	0	0
35	20	31,43	20	8,57	2,86	5,71	2,86	0	8,57

40	27,5	25	7,5	5	5	5	5	0	20
45	28,89	13,33	6,67	6,67	6,67	2,22	4,44	2,22	28,89
50	36	18	6	2	0	0	0	2	36
55	29,09	5,45	7,27	9,09	3,64	1,82	0	1,82	41,82
60	28,33	10	10	5	0	0	0	0	46,67

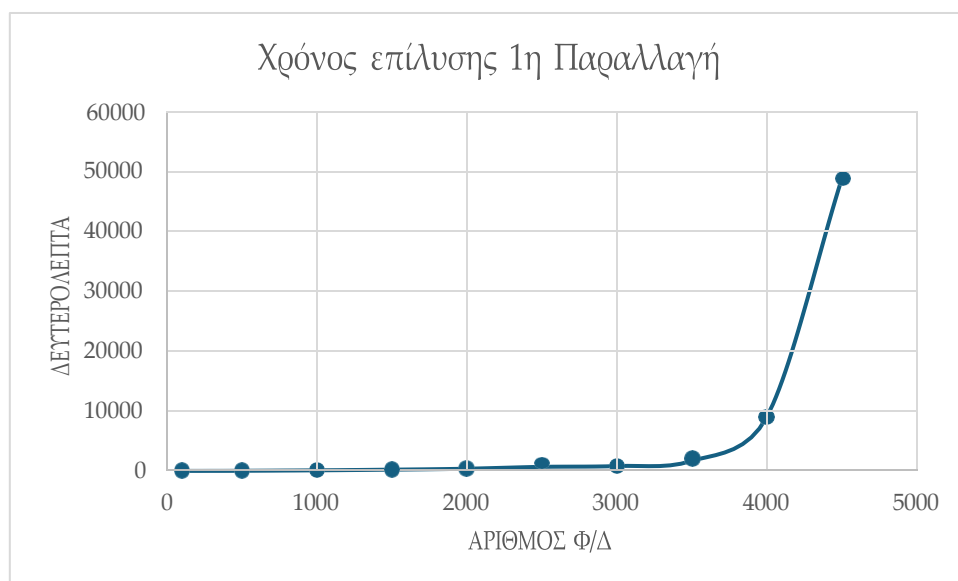
Πίνακας 12 Ποσοστό Ανάθεσης προτιμήσεων ανά αριθμό φοιτητών για αριθμό διπλωματικών = 30, δεύτερη παραλλαγή

Το συμπέρασμα που προκύπτει από τους παραπάνω πίνακες είναι ότι στην πρώτη παραλλαγή τα ποσοστά ανάθεσης στις πρώτες προτιμήσεις των φοιτητών είναι αρκετά υψηλά, σε αντίθεση με την δεύτερη παραλλαγή, όπου τα ποσοστά ανάθεσης είναι πιο ομοιόμορφα κατανεμημένα. Αυτό συμβαίνει λόγω των μεθόδων επίλυσης που εφαρμόζονται στις παραλλαγές του προβλήματος. Στην πρώτη παραλλαγή, η ανάθεση γίνεται με σκοπό αρχικά να ικανοποιηθούν οι πρώτες προτιμήσεις των φοιτητών, και στην συνέχεια αν δεν είναι εφικτό να ανατεθούν οι πρώτες προτιμήσεις να συνεχιστεί ο έλεγχος για την ανάθεση των δεύτερων επιθυμιών. Στην δεύτερη παραλλαγή, μεγιστοποιείται ο αριθμός αναθέσεων, χωρίς να υπάρχει κάποιος έλεγχος πάνω στην σειρά προτίμησης. Έτσι, ενώ η δεύτερη παραλλαγή εγγυάται μεγαλύτερο ποσοστό ανάθεσης, η πρώτη παραλλαγή προσφέρει μεγαλύτερο ποσοστό ανάθεσης στις πρώτες προτιμήσεις των φοιτητών.

3.3 Χρονομέτρηση επίλυσης

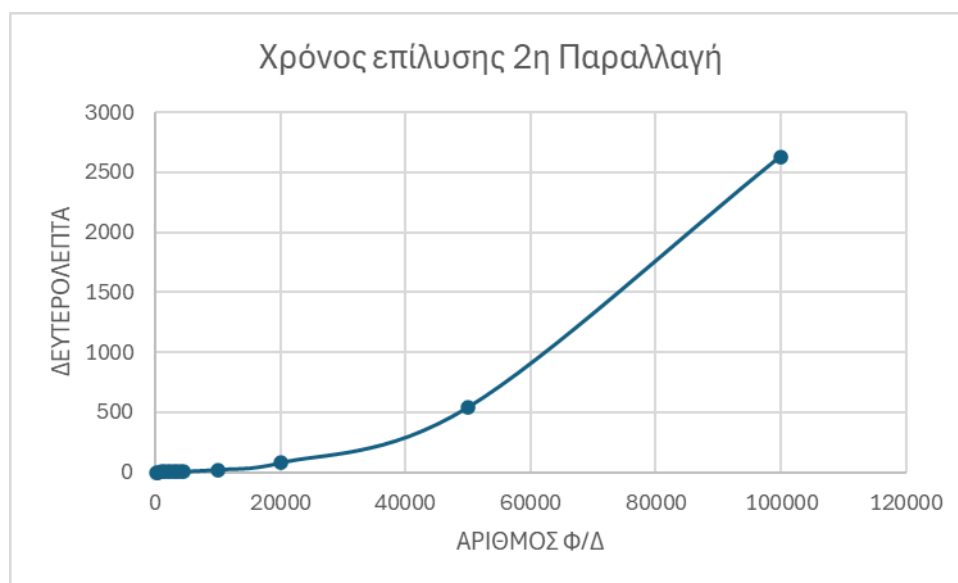
Ολοκληρώνοντας την σύγκριση μεταξύ των 2 παραλλαγών, παρουσιάζονται δύο διαγράμματα πάνω στο χρόνο που χρειάζονται οι δύο παραλλαγές για να επιλύσουν ένα πρόβλημα. Σε αυτά τα παραδείγματα θεωρήθηκαν ο αριθμός των φοιτητών και των διπλωματικών ίσος. Παρακάτω, παρουσιάζονται τα 2 διαγράμματα πάνω στον χρόνο επίλυσης των δύο μεθόδων.

Πρώτη παραλλαγή:



Διάγραμμα 1 Χρόνος επίλυσης πρώτης παραλλαγής ανά αριθμό φοιτητών/ διπλωματικών

Δεύτερη παραλλαγή:



Διάγραμμα 2 Χρόνος επίλυσης δεύτερης παραλλαγής ανά αριθμό φοιτητών/ διπλωματικών

Από τα παραπάνω διαγράμματα, προκύπτει ότι ο χρόνος επίλυσης στις περιπτώσεις που εφαρμόζεται η πρώτη παραλλαγή, αυτή με τους περιορισμούς τομέα και κατάταξης δηλαδή, είναι αρκετά μεγαλύτερος από την περίπτωση της μέγιστης ανάθεσης διπλωματικών εργασιών. Ακόμα και όταν υπάρχουν 10.000 φοιτητές και διπλωματικές, ο χρόνος επίλυσης στην περίπτωση της μέγιστης ανάθεσης είναι μικρότερος από όταν έχουμε 4.500 φοιτητές και διπλωματικές και επιλέγεται το πρόβλημα να λυθεί με τους περιορισμούς τομέα και κατάταξης.

3.4 Ένα πολύ ενδιαφέρον παράδειγμα

Κλείνοντας το παρόν κεφάλαιο, παρουσιάζεται ένα πολύ ενδιαφέρον παράδειγμα στην συγκεκριμένη ενότητα. Για να εξηγηθεί σωστά το εν λόγω παράδειγμα, παρουσιάζεται η πιο απλή μορφή του. Αφορά την ανάθεση σε τρεις φοιτητές με δύο διαθέσιμες διπλωματικές. Παρακάτω, παρουσιάζεται το αρχείο εισόδου:

3 2

1 2 3

4 5

1 2 3

2 3

1 1

4

5 4

4 5

Εξηγώντας το αρχείο στην πρώτη γραμμή δηλώνεται ότι υπάρχουν τρεις φοιτητές και δύο διπλωματικές. Μετά για λόγους ευκολίας, τα id των φοιτητών έστω ότι είναι 1,2,3 για τους 3 φοιτητές με την σειρά, ενώ τα id των διπλωματικών είναι 4 και 5 αντίστοιχα. Επιπλέον, φαίνεται ότι ο πρώτος φοιτητής ανήκει στον πρώτο τομέα, ο δεύτερος στον δεύτερο τομέα και ο τρίτος φοιτητής στον τρίτο τομέα. Επίσης, η πρώτη διπλωματική ανήκει στον δεύτερο τομέα και η τρίτη διπλωματική στον τρίτο τομέα. Τόσο η πρώτη, όσο και η

δεύτερη διπλωματική, μπορούν να ανατεθούν μόνο σε ένα άτομο. Όσον αφορά τις επιθυμίες των φοιτητών, ο πρώτος έχει δηλώσει μόνο την διπλωματική 4, ο δεύτερος πρώτα την διπλωματική 5 και μετά την διπλωματική 4, ενώ ο τρίτος πρώτα την διπλωματική 4 και μετά την διπλωματική 5. Χρησιμοποιώντας τον αλγόριθμο που αναπτύχθηκε, βγαίνει το παρακάτω αρχείο αποτελεσμάτων:

1 null

2 4

3 5

Σύνολο αναθέσεων: 2

Ποσοστό ανάθεσης: 66.67%

Προτίμηση 2: 2 φοιτητές (66.67%)

Ποσοστό φοιτητών που δεν τους έγινε ανάθεση διπλωματικής: 33.33%

Χρόνος επίλυσης: 0.02 δευτερόλεπτα

Το παραπάνω αρχείο, αναφέρει ότι στον φοιτητή 1 δεν έγινε κάποια ανάθεση, στον φοιτητή 2 ανατέθηκε η διπλωματική 4 και τέλος στον φοιτητή 3 η διπλωματική 5. Δηλαδή, τόσο στον φοιτητή 2 όσο και στον φοιτητή 3 ανατέθηκαν οι δεύτερες προτιμήσεις τους και όχι οι πρώτες. Σε ένα υποθετικό σενάριο, οι δύο αυτοί φοιτητές θα συζητούσαν και θα θέλανε να ανταλλάξουν διπλωματικές. Αυτό όμως ενώ με μία πρώτη ματιά φαίνεται σωστό, στην πραγματικότητα δεν είναι. Οι δύο φοιτητές, σας πρώτη τους επιλογή, δηλώσανε μία διπλωματική που δεν ανήκει στον τομέα τους. Το ίδιο έκανε όμως και ο πρώτος φοιτητής, καθώς δήλωσε την διπλωματική 4, που ανήκει στον τομέα 2 και σε σειρά προτεραιότητας βρίσκεται πιο μπροστά και από τους δύο άλλους φοιτητές. Αυτό σημαίνει ότι σε περίπτωση που η διπλωματική 4 δεν ανατεθεί στον φοιτητή που βρίσκεται στον ίδιο τομέα, στην συγκεκριμένη περίπτωση στον φοιτητή 2, τότε δεν πρέπει να ανατεθεί στον φοιτητή 3, άλλα πρέπει να ανατεθεί στον φοιτητή 1 καθώς βρίσκεται πιο ψηλά στην λίστα των φοιτητών. Έτσι η ανάθεση γίνεται σε φοιτητές και διπλωματικές που ανήκουν στον ίδιο τομέα καθώς σε διαφορετική περίπτωση, ο φοιτητής που θα είχε μείνει χωρίς ανάθεση διπλωματικής, θα ήταν ο φοιτητής 3 και όχι ο φοιτητής 1. Αυτό το παράδειγμα, παρουσιάζει το πόσο ισχυρός είναι ο περιορισμός του τομέα ανάμεσα σε φοιτητές και διπλωματικές, καθώς οι διπλωματικές ανατέθηκαν σε φοιτητές που ήταν πιο χαμηλά σε σειρά προτεραιότητας, καθώς αυτοί είχαν δηλώσει διπλωματικές του τομέα τους.

Κεφάλαιο 4. ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΡΟΤΑΣΕΙΣ

Στην παρούσα διπλωματική εργασία, μελετήθηκε ένα πρόβλημα ανάθεσης διπλωματικών εργασιών του τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας. Πιο συγκεκριμένα, μοντελοποιήθηκε ένας αλγόριθμός που αφού διαβάσει ένα αρχείο, επιλύει το πρόβλημα σύμφωνα με τους περιορισμούς του και εκτυπώνει τα αποτελέσματα σε ένα αρχείο εξαγωγής. Οι περιορισμοί προκύπτουν από την φύση του προβλήματος.

Αρχικά, ο πρώτος περιορισμός είναι ότι σε έναν φοιτητή μπορεί να ανατεθεί μία μόνο διπλωματική. Ο δεύτερος περιορισμός είναι, ότι μία διπλωματική μπορεί να οριστεί σε έναν ή και σε δύο φοιτητές, ανάλογα με το αν η συγκεκριμένη διπλωματική μπορεί να εκπονηθεί από έναν ή μπορεί να εκπονηθεί και σε συνεργασία δύο φοιτητών. Τέλος, ο πιο σημαντικός περιορισμός είναι ο περιορισμός των τομέων και της σειράς προτεραιότητας. Αυτός ο περιορισμός εφαρμόζεται όταν υπάρχουν δύο φοιτητές που έχουν δηλώσει την ίδια διπλωματική. Για να αποφασιστεί ποιος από τους δύο θα έχει την προτεραιότητα, σημασία έχει ο τομέας που βρίσκονται οι δύο φοιτητές, και ο τομέας της διπλωματικής. Εάν ένας από τους δύο φοιτητές βρίσκεται στον ίδιο τομέα με τον τομέα της διπλωματικής, τότε αυτός ο φοιτητής έχει προτεραιότητα. Αν όμως, και οι δύο φοιτητές βρίσκονται στον ίδιο τομέα με την διπλωματική, ή κανένας από τους δύο δεν βρίσκεται στον ίδιο τομέα με την διπλωματική, τότε προτεραιότητα έχει ο φοιτητής που βρίσκεται πιο ψηλά στην σειρά προτεραιότητας.

Στην συνέχεια, αναλύθηκε ο κώδικας της διπλωματικής. Ο κώδικας αρχικά διαβάσει ένα εισαγωγικό αρχείο που περιέχει των αριθμό των φοιτητών και των διπλωματικών, τα id των φοιτητών και των διπλωματικών, τους τομείς που ανήκουν οι φοιτητές και οι διπλωματικές, την πληροφορία σχετικά με το αν μία διπλωματική εργασία μπορεί να ανατεθεί σε έναν ή και σε δύο φοιτητές και οι δηλώσεις των επιθυμιών των διπλωματικών εργασιών των φοιτητών. Ο κώδικας ζητά από τον χρήστη αρχικά να δηλώσει των αριθμό των φοιτητών και των διπλωματικών εργασιών. Στην συνέχεια, ζητά να του δηλώσει εάν επιθυμεί να δημιουργηθεί ένα τυχαίο αρχείο εισαγωγής, η να διαβάσει ένα έτοιμο αρχείο. Τέλος, δίνει τη δυνατότητα στον χρήστη να επιλέξει εάν το πρόβλημα θα επιλυθεί με τους περιορισμούς

τομέα και σειράς προτεραιότητας, ή εάν δεν θα εφαρμοστούν οι συγκεκριμένοι περιορισμοί και θα γίνει απλώς μέγιστη ανάθεση διπλωματικών εργασιών. Στο αρχείο που εκτυπώνεται, παρουσιάζονται οι αναθέσεις των φοιτητών με τις διπλωματικές, το ποσοστό ανάθεσης, το ποσοστό ανάθεσης ανά σειρά προτίμησης, το ποσοστό των φοιτητών που δεν τους έγινε ανάθεση κάποιας διπλωματικής, και η χρονομέτρηση επίλυσης του προβλήματος.

Στην συνέχεια, παρουσιάστηκαν κάποια παραδείγματα που συγκρίθηκαν οι δύο παραλλαγές του προβλήματος. Τα συγκεκριμένα παραδείγματα οδηγούν στο συμπέρασμα ότι η επίλυση μέσω της παραλλαγής της μέγιστης ανάθεσης δίνει μεγαλύτερο ποσοστό ανάθεσης και πιο μικρό χρόνο επίλυσης, άλλα η παραλλαγή με τους περιορισμούς τομέα και σειράς προτεραιότητας δίνει μεγαλύτερο ποσοστό ανάθεσης στις πρώτες προτιμήσεις των φοιτητών. Τέλος, παρουσιάστηκε ένα παράδειγμα που αποδεικνύει το πόσο ισχυρός είναι ο περιορισμός τομέα στις αναθέσεις διπλωματικών εργασιών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Zhao J. , Liu Y. and Yang X. (2019) To Inspire and Enhance Research Interests with Reading Assignments before Dissertation Topic Selection: An Empirical Study of Master Students. *ICIEI '19: Proceedings of the 4th International Conference on Information and Education Innovations*, 58-61.
- [2] Sanchez- Anguix V. , Chalumuri R., Alberola J.M and Aydogan R.(2020) Artificial Intelligence Tools for Academic Management: Assigning Students to Academic Supervisors. *INTED2020 Proceedings*, 4638-4644
- [3] Serek A. and Zhaparov M. (2024) Optimizing preference satisfaction with genetic algorithm in matching students to supervisors. *Applied Mathematics & Information Sciences*, 18(1), 133-138.
- [4] Paul R. Harper, Valter de Senna, Israel T. Vieira, Arjan K. Shahani (2005) A genetic algorithm for the project assignment problem. *Computers & Operations Research*, 32(5), 1255-1265.
- [5] Augustine Kwanashie, Robert W. Irving, David F. Manlove & Colin T.S. Song (2015) Profile-Based Optimal Matchings in the Student/Project Allocation Problem. *Combinatorial Algorithms*, 213-225.
- [6] Sanchez-Anguix V. , Chalumuri R. and Julian V. (2018) A Multi-objective Evolutionary Proposal for Matching Students to Supervisors. *Distributed Computing and Artificial Intelligence, 15th International Conference*, 94-102.
- [7] Sanchez-Anguix V. , Chalumuri R. , Aydoğan R. and Julian V. (2019) A near Pareto optimal approach to student-supervisor allocation with two sided preferences and workload balance. *Applied Soft Computing*, 1-15.
- [8] Laporte G. and Desroches S. (1986) The problem of assigning students to course sections in a large engineering school. *Computers & Operations Research*, 387-394.

Κεφάλαιο 5. ΠΑΡΑΡΤΗΜΑ

Σε αυτό το Παράρτημα, παρουσιάζεται ο πλήρης κώδικας της παρούσας διπλωματικής εργασίας:

```
# -*- coding: utf-8 -*-  
from docplex.mp.model import Model  
import random  
import time  
  
P = Model(name="DiplomatikiProblem")  
  
num_students = int(input("Εισάγετε τον αριθμό των φοιτητών: "))  
num_theses = int(input("Εισάγετε τον αριθμό των διπλωματικών: "))  
  
print('Παρακαλώ επιλέξτε 0 για τυχαία δημιουργία δεδομένων, 1 για ανάγνωση  
δεδομένων από αρχείο')  
data_choice = int(input())  
  
def write_line_with_limit(datafile, data_list, limit=50):  
    line = []  
    for i, item in enumerate(data_list):  
        line.append(str(item))  
        if len(line) == limit or i == len(data_list) - 1:  
            datafile.write(" ".join(line) + "\n")  
            line = []  
  
foitites_tomoi = {}  
diplomatikes_tomoi = {}
```

```

wishes = {}
diplomatikes_ed = {}
student_ids = []
thesis_ids = []

if data_choice == 0:
    with open('diloseis.txt', 'w', encoding='utf-8') as datafile:
        toimoi = [1, 2, 3]

        student_ids = [100 + i for i in range(1, num_students + 1)]
        thesis_ids = [200 + i for i in range(1, num_theses + 1)]

        foitites_tomoi = {stud_id: random.choice(toimoi) for stud_id in student_ids}
        diplomatikes_tomoi = {thesis_id: random.choice(toimoi) for thesis_id in
thesis_ids}

        diplomatikes_ed = {thesis_id: 1 for thesis_id in thesis_ids}
        twos_sample = random.sample(thesis_ids, min(2, len(thesis_ids)))
        for thesis_id in twos_sample:
            diplomatikes_ed[thesis_id] = 2

        wishes = {}
        for stud_id in student_ids:
            prefs_count = random.randint(1, min(8, num_theses))
            wishes[stud_id] = random.sample(thesis_ids, prefs_count)

        datafile.write(f"{num_students} {num_theses}\n\n")
        write_line_with_limit(datafile, student_ids)
        datafile.write("\n")
        write_line_with_limit(datafile, thesis_ids)
        datafile.write("\n")
        write_line_with_limit(datafile, [foitites_tomoi[stud_id] for stud_id in
student_ids])

```

```

        datafile.write('\n')
        write_line_with_limit(datafile, [diplomatikes_tomoi[thesis_id] for thesis_id in
thesis_ids])
        datafile.write('\n')
        write_line_with_limit(datafile, [diplomatikes_ed[thesis_id] for thesis_id in
thesis_ids])
        datafile.write('\n')
        for stud_id in student_ids:
            datafile.write(" ".join(map(str, wishes[stud_id])) + "\n")
        print("Δεδομένα αποθηκεύτηκαν στο diloseis.txt")

elif data_choice == 1:
    try:
        with open('diloseis.txt', 'r', encoding='utf-8') as datafile:
            line = datafile.readline()
            num_students, num_theses = map(int, line.strip().split())

            student_ids = []
            while len(student_ids) < num_students:
                student_ids += list(map(int, datafile.readline().split()))
            datafile.readline()

            thesis_ids = []
            while len(thesis_ids) < num_theses:
                thesis_ids += list(map(int, datafile.readline().split()))
            datafile.readline()

            foitites_sectors = []
            while len(foitites_sectors) < num_students:
                foitites_sectors += list(map(int, datafile.readline().split()))
            datafile.readline()

```

```

diplomatikes_sectors = []
while len(diplomatikes_sectors) < num_theses:
    diplomatikes_sectors += list(map(int, datafile.readline().split()))
datafile.readline()

diplomatikes_size = []
while len(diplomatikes_size) < num_theses:
    diplomatikes_size += list(map(int, datafile.readline().split()))
datafile.readline()

wishes = {}
for i in range(num_students):
    line = datafile.readline().split()
    wishes[student_ids[i]] = list(map(int, line)) if line else []

foitites_tomoi = {student_ids[i]: foitites_sectors[i] for i in
range(num_students)}
diplomatikes_tomoi = {thesis_ids[i]: diplomatikes_sectors[i] for i in
range(num_theses)}
diplomatikes_ed = {thesis_ids[i]: diplomatikes_size[i] for i in
range(num_theses)}

print("Δεδομένα φορτώθηκαν από το αρχείο diloseis.txt")

except Exception as e:
    print(f"Σφάλμα: {e}")
    exit()
else:
    print("Μη έγκυρη επιλογή.")
    exit()

```

```
print('Παρακαλώ επιλέξτε 0 για επίλυση με περιορισμούς, 1 για απλή μέγιστη  
ανάθεση')
```

```
solving_method = int(input())
```

```
if solving_method == 0:
```

```
    start_time = time.time()
```

```
    Y = {(s, t): P.binary_var(name=f"Y_{s}_{t}") for s in student_ids for t in  
wishes.get(s, [])}
```

```
    for s in student_ids:
```

```
        P.add_constraint(P.sum(Y[s, t] for t in wishes.get(s, []) if (s, t) in Y) <= 1)
```

```
    for t in thesis_ids:
```

```
        P.add_constraint(P.sum(Y[s, t] for s in student_ids if (s, t) in Y) <=  
diplomatikes_ed.get(t, 0))
```

```
def get_preference_penalty_expr(student_id, target_thesis_id):
```

```
    expr = P.linear_expr()
```

```
    if student_id not in wishes or target_thesis_id not in wishes[student_id]:
```

```
        return expr
```

```
    target_rank = wishes[student_id].index(target_thesis_id)
```

```
    for other_thesis_id in wishes[student_id]:
```

```
        if (student_id, other_thesis_id) in Y:
```

```
            if wishes[student_id].index(other_thesis_id) < target_rank:
```

```
                expr.add_term(Y[student_id, other_thesis_id], 1)
```

```
    return expr
```

```
for thesis_id in thesis_ids:
```

```
    interested = [s for s in student_ids if thesis_id in wishes.get(s, [])]
```

```
    for i in range(len(interested)):
```

```

for j in range(i + 1, len(interested)):
    s1, s2 = interested[i], interested[j]
    s1_sector = foitites_tomoi.get(s1) == diplomatikes_tomoi.get(thesis_id)
    s2_sector = foitites_tomoi.get(s2) == diplomatikes_tomoi.get(thesis_id)

    if s1_sector and not s2_sector:
        higher, lower = s1, s2
    elif s2_sector and not s1_sector:
        higher, lower = s2, s1
    else:
        higher, lower = sorted((s1, s2))

    if (higher, thesis_id) in Y and (lower, thesis_id) in Y:
        penalty_expr = get_preference_penalty_expr(higher, thesis_id)
        P.add_constraint(Y[lower, thesis_id] <= Y[higher, thesis_id] +
penalty_expr)

```

```

assigned_theses = {s: None for s in student_ids}
with open('apotelesmata.txt', 'w', encoding='utf-8') as resfile:
    assigned = 0
    preference_matches = {}

    for s in student_ids:
        assigned_for_this = False
        for t in wishes.get(s, []):
            if (s, t) not in Y:
                continue
            constraint_name = f"assign_{s}_{t}"
            P.add_constraint(Y[s, t] == 1, cname=constraint_name)
            if P.solve():
                assigned_theses[s] = t
                resfile.write(f"{s} {t}\n")
                rank = wishes[s].index(t) + 1
                preference_matches[rank] = preference_matches.get(rank, 0) + 1

```

```

        assigned += 1
        assigned_for_this = True
        break
    else:
        P.remove_constraint(constraint_name)
    if not assigned_for_this:
        resfile.write(f"{s} null\n")

resfile.write(f"\nΣύνολο αναθέσεων: {assigned}\n")
resfile.write(f"Ποσοστό ανάθεσης: {assigned / num_students * 100:.2f}%\n")
for rank in sorted(preference_matches):
    count = preference_matches[rank]
    label = "φοιτητής" if count == 1 else "φοιτητές"
    resfile.write(f"Προτίμηση {rank}: {count} {label} ({count / num_students *
100:.2f}%) \n")
    resfile.write(f"Ποσοστό φοιτητών που δεν τους έγινε ανάθεση διπλωματικής: {(1
- assigned / num_students) * 100:.2f}%\n")
    end_time = time.time()
    solving_time = end_time - start_time
    resfile.write(f"Χρόνος επίλυσης: {solving_time:.2f} δευτερόλεπτα\n")
    print(f"Χρόνος επίλυσης: {solving_time:.2f} δευτερόλεπτα")
    print("Η επίλυση ολοκληρώθηκε. Δες τα αποτελέσματα στο 'apotelesmata.txt'")

elif solving_method == 1:
    start_time = time.time()
    Y = {(s, t): P.binary_var(name=f"Y_{s}_{t}") for s in student_ids for t in
wishes.get(s, [])}
    for s in student_ids:
        P.add_constraint(P.sum(Y[s, t] for t in wishes.get(s, [])) <= 1)
    for t in thesis_ids:
        P.add_constraint(P.sum(Y[s, t] for s in student_ids if (s, t) in Y) <=
diplomatikes_ed.get(t, 0))
    P.maximize(P.sum(Y[s, t] for (s, t) in Y))
    P.solve()

```

```

assigned = 0
preference_matches = {}
with open('apotelesmata2.txt', 'w', encoding='utf-8') as resfile:
    for s in student_ids:
        assigned_thesis = None
        for t in wishes.get(s, []):
            if (s, t) in Y and Y[s, t].solution_value > 0.5:
                assigned_thesis = t
                assigned += 1
                break
        if assigned_thesis is not None:
            rank = wishes[s].index(assigned_thesis) + 1
            preference_matches[rank] = preference_matches.get(rank, 0) + 1
            resfile.write(f"{s} {assigned_thesis}\n")
        else:
            resfile.write(f"{s} null\n")
    resfile.write(f"\nΣύνολο αναθέσεων: {assigned}\n")
    resfile.write(f"Ποσοστό ανάθεσης: {assigned / num_students * 100:.2f}%\n")
    for rank in sorted(preference_matches):
        count = preference_matches[rank]
        label = "φοιτητής" if count == 1 else "φοιτητές"
        resfile.write(f"Προτίμηση {rank}: {count} {label} ({count / num_students * 100:.2f}%)\n")
    resfile.write(f"Ποσοστό φοιτητών που δεν τους έγινε ανάθεση διπλωματικής: {(1 - assigned / num_students) * 100:.2f}%\n")
    end_time = time.time()
    solving_time = end_time - start_time
    resfile.write(f"Χρόνος επίλυσης: {solving_time:.2f} δευτερόλεπτα\n")
    print(f"Χρόνος επίλυσης: {solving_time:.2f} δευτερόλεπτα")
    print("Η επίλυση ολοκληρώθηκε. Δες τα αποτελέσματα στο 'apotelesmata2.txt'")

else:
    print("Μη έγκυρη επιλογή μεθόδου επίλυσης.")

```