Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Αλγόριθμοι βελτιστοποίησης για το πρόβλημα ανάθεσης

Ιωάννης Σφακιανάκης (ΑΜ: 1559)

Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

Εργαστήριο Ευφυών Συστημάτων & Βελτιστοποίησης

14 Οκτωβρίου 2024

Περίληψη

Η παρούσα διπλωματική εργασία επικεντρώνεται στην επίλυση ενός κλασικού προβλήματος βελτιστοποίησης, του προβλήματος ανάθεσης, με ευρείες εφαρμογές στη διαχείριση έργων, παραγωγής και κατανομής πόρων. Η εργασία αυτή εξετάζει διάφορους αλγόριθμους εστιάζοντας στην απόδοσή τους, συγκριτικά τόσο σε πυκνά όσο και σε αραιά σύνολα προβλημάτων. Επιπλέον, αναλύεται η χρήση μνήμης και οι χρόνοι εκτέλεσης κάθε αλγορίθμου, με στόχο την κατανόηση των πλεονεκτημάτων και μειονεκτημάτων κάθε προσέγγισης. Με την ανάλυση αυτή, παρέχεται μια σαφή κατεύθυνση, όσον αφορά την επιλογή του κατάλληλου αλγορίθμου, ανάλογα με τις απαιτήσεις των δεδομένων και των διαθέσιμων πόρων.

Λέξεις κλειδιά: Πρόβλημα ανάθεσης, αλγόριθμοι βελτιστοποίησης, MIP, CP-SAT, Ουγγρικός αλγόριθμος, Jonker-Volgenant, LAPMOD.

Abstract

This thesis focuses on solving a classical optimization problem, the assignment problem, with broad applications in project management, production and resource allocation. This study examines several algorithms, focusing on their performance comparatively on both dense and sparse problem sets. In addition, the memory usage and execution times of each algorithm are analyzed, in order to understand the advantages and disadvantages of each approach. With this analysis, a clear guidance is provided in terms of selecting the appropriate algorithm, depending on the data requirements and available resources.

Keywords: Assignment problem, optimization algorithms, MIP, CP-SAT, Hungarian algorithm, Jonker-Volgenant, LAPMOD.

Δήλωση Πνευματικών Δικαιωμάτων

Υπογραφή Φοιτητή

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος αλγορίθμων

Κατάλογος πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός του Προβλήματος

Το πρόβλημα ανάθεσης αποτελεί μια κλασική πρόκληση στη βελτιστοποίηση, όπου στόχος είναι η αντιστοίχιση ενός συνόλου πόρων σε ένα σύνολο εργασιών με τέτοιο τρόπο, με στόχο την ελαχιστοποίηση του συνολικού κόστους ή της μεγιστοποίησης του κέρδους ανάθεσης. Οι εφαρμογές του προβλήματος ανάθεσης καλύπτουν ένα πολύ μεγάλο εύρος από τομείς, όπως η διαχείριση έργων, η παραγωγή και η κατανομή πόρων. Η μαθηματική του μοντελοποίηση περιλαμβάνει τη διαχείριση πινάκων κόστους, όπου πρέπει να επιλεγεί η καλύτερη δυνατή κατανομή πόρων σε δραστηριότητες, περιορίζοντας τις αλληλεπιδράσεις και εξασφαλίζοντας ότι κάθε εργασία εκτελείται από έναν μοναδικό πόρο.

1.2 Κίνητρα και Στόχοι Υλοποίησης

Τα κίνητρα για την υλοποίηση αυτής της διπλωματικής εργασίας προκύπτουν από την ανάγκη εύρεσης αποδοτικών λύσεων για το πρόβλημα ανάθεσης, χρησιμοποιώντας διάφορους αλγορίθμους βελτιστοποίησης, με τους στόχους να είναι η ανάπτυξη, υλοποίηση και σύγκριση διαφορετικών αλγορίθμων. Επιπλέον, ιδιαίτερη έμφαση δίνεται στη μελέτη της απόδοσης αυτών των αλγορίθμων, τόσο σε όρους ταχύτητας όσο και κατανάλωσης μνήμης, καθώς και στην ικανότητά τους να διαχειρίζονται πυκνά και αραιά προβλήματα. Τέλος, στόχος είναι η καταγραφή των αποτελεσμάτων και η εξαγωγή συμπερασμάτων που θα επιτρέψουν την καλύτερη κατανόηση της αποτελεσματικότητας κάθε μεθόδου.

1.3 Διάρθρωση Κειμένου

Τα υπόλοιπα κεφάλαια οργανώνονται ως εξής. Στο Κεφάλαιο 2 γίνεται ανασκόπηση της σχετικής βιβλιογραφίας, όπου παρουσιάζονται οι βασικοί αλγόριθμοι και μέθοδοι που έχουν αναπτυχθεί για την επίλυση του προβλήματος ανάθεσης, με αναφορά τόσο σε κλασικές όσο και σε πιο σύγχρονες προσεγγίσεις. Στο Κεφάλαιο 3 αναλύονται οι υλοποιήσεις των αλγορίθμων που χρησιμοποιήθηκαν στην εργασία. Παρουσιάζονται τα τεχνικά χαρακτηριστικά των αλγορίθμων και οι διαδικασίες που ακολουθήθηκαν για την εφαρμογή τους στο πρόβλημα ανάθεσης. Στο Κεφάλαιο 4 παρουσιάζονται τα αποτελέσματα των πειραματικών δοκιμών, γίνεται σύγκριση της απόδοσης των αλγορίθμων σε διάφορα σύνολα δεδομένων, με έμφαση στον χρόνο εκτέλεσης και τη χρήση μνήμης. Τέλος, στο Κεφάλαιο 5 εξάγονται τα συνολικά συμπεράσματα της εργασίας.

Κεφάλαιο 2

Βιβλιογραφική ανασκόπηση

Το πρόβλημα ανάθεσης είναι ένα θεμελιώδες πρόβλημα βελτιστοποίησης, που ανήκει στην κατηγορία των προβλημάτων διαχείρισης πόρων, με εφαρμογές σε διάφορους τομείς, όπως η επιχειρησιακή έρευνα, τα οικονομικά και η επιστήμη των υπολογιστών. Το πρόβλημα ανάθεσης είναι μια ειδική περίπτωση του προβλήματος μεταφοράς, όπου ο στόχος είναι η ανάθεση ενός αριθμού πόρων σε ίσο αριθμό δραστηριοτήτων, ώστε να ελαχιστοποιηθεί το συνολικό κόστος ή να μεγιστοποιηθεί το συνολικό κέρδος της ανάθεσης. Το πρόβλημα αυτό προκύπτει διότι οι διαθέσιμοι πόροι, όπως οι άνθρωποι, τα μηχανήματα κ.λπ. έχουν διαφορετικό βαθμό απόδοσης για την εκτέλεση διαφορετικών δραστηριοτήτων, επομένως, το κόστος, το κέρδος ή η ζημία από την εκτέλεση των διαφορετικών δραστηριοτήτων είναι διαφορετικά. Συνεπώς, το ερώτημα που θέτει το πρόβλημα ανάθεσης είναι: "Πώς πρέπει να γίνουν οι αναθέσεις ώστε να βελτιστοποιηθεί ο συγκεκριμένος στόχος".

Ορισμός και μαθηματική μοντελοποίηση:

Το κάθε πρόβλημα ανάθεσης συνοδεύεται από τον δικό του πίνακα. Γενικότερα, οι γραμμές περιέχουν τους πόρους που θέλουμε να αναθέσουμε, ενώ οι στήλες περιέχουν τις εργασίες που θέλουμε να εκτελεστούν. Υποθέστε ένα πρόβλημα ανάθεσης n πόρων σε m δραστηριότητες με σκοπό την ελαχιστοποίηση του συνολικού κόστους ή χρόνου, με τέτοιο τρόπο ώστε κάθε πόρος να μπορεί να συσχετιστεί με μία και μόνο μία εργασία. Ο πίνακας κόστους c δίνεται ως εξής [36]:

Έστω x_{ij} να δηλώνει την ανάθεση κάθε i πόρων σε κάθε j δραστηριότητες, έτσι

Πίνακας 2.1: Cost Matrix

		Activity				
		A_1	A_2	$\cdots$	A_n	Available
	R_1	c_{11}	c_{12}	$\cdots$	c_{1n}	1
	R_2	c_{21}	c_{22}	$\cdots$	c_{2n}	1
Resource	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	R_n	c_{n1}	c_{n2}	$\cdots$	c_{nm}	1
	Required	1	1	$\cdots$	1	

ώστε:

$$x_{ij} = \begin{cases} 1, & \text{Εάν ο πόρος } i \text{ ανατίθεται σε δραστηριότητα } j \\ 0, & \text{Αλλιώς} \end{cases} \tag{2.1}$$

Με περιορισμούς:

1. Κάθε εργασία ανατίθεται ακριβώς σε έναν πράκτορα:

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \forall j = 1, 2, \ldots, n$$

2. Σε κάθε πράκτορα ανατίθεται ακριβώς μία εργασία:

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \forall i = 1, 2, \ldots, n$$

3. Δυαδικός περιορισμός:

$$x_{ij} \in \{0, 1\}, \quad \forall i, j$$

Τότε η μαθηματική φόρμουλα είναι ως εξής:

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

υπόκειται σε:

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \forall j = 1, 2, \ldots, n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \forall i = 1, 2, \ldots, n$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j$$

2.1 Ουγγρικός Αλγόριθμος-The Hungarian Algorithm

Ο Ουγγρικός αλγόριθμος αναπτύχθηκε το 1955 από τον H.W. Kuhn [31] και είναι ένας αλγόριθμος συνδυαστικής βελτιστοποίησης, όπου σχεδιάστηκε για την επίλυση του προβλήματος ανάθεσης, όπου περιλαμβάνει την ανάθεση n εργαζομένων σε n εργασίες με τρόπο που ελαχιστοποιεί το συνολικό κόστος ή μεγιστοποιεί το συνολικό κέρδος των αναθέσεων. Ο αλγόριθμος αρχικά βασίστηκε στις ιδέες δύο Ούγγρων μαθηματικών, του Dénes Kőnig και του Jenő Egerváry, εξού και το όνομά του.

Ο αλγόριθμος λειτουργεί κάνοντας κάποια διαδοχικά βήματα:

1. **Ελαχιστοποίηση γραμμών και στηλών**: Αφαιρεί τη μικρότερη τιμή από κάθε γραμμή και στήλη σε όλες τις καταχωρήσεις, όπου υπάρχει η αντίστοιχη γραμμή και στήλη.

2. **Κάλυψη μηδενικών τιμών**: Ο αλγόριθμος καλύπτει όλα τα μηδενικά από τον πίνακα που προκύπτει, χρησιμοποιώντας έναν ελάχιστο αριθμό οριζόντιων και κάθετων γραμμών.

3. **Βέλτιστη ανάθεση**: Εάν n γραμμές καλύπτουν όλα τα μηδενικά, μπορεί να προσδιοριστεί μια βέλτιστη ανάθεση. Διαφορετικά, προσαρμόζει τον πίνακα και επαναλαμβάνει.

Η ουγγρική μέθοδος έχει υπολογιστική πολυπλοκότητα $O(n^3)$, καθιστώντας την κατάλληλα αποτελεσματική για σχετικά μεγάλες περιπτώσεις προβλημάτων. Με το πέρασμα των χρόνων, έχουν προταθεί διάφορες βελτιώσεις και παραλλαγές του αρχικού αλγορίθμου για την ενίσχυση της απόδοσής του. Για παράδειγμα, οι Jonker και Volgenant [30] ενσωμάτωσαν τροποποιήσεις που μείωσαν σημαντικά τους χρόνους υπολογισμού, ειδικά για προβλήματα μεγάλης κλίμακας. Επιπλέον, ο Munkres παρείχε μια εναλλακτική διατύπωση του αλγορίθμου, που μερικές φορές αναφέρεται ως αλγόριθμος Kuhn-Munkres [33]. Η υλοποίηση του αλγορίθμου έχει μελετηθεί εκτενώς, με αποτελεσματικές εκδόσεις να είναι διαθέσιμες στη βιβλιογραφία. Για παράδειγμα, οι Carpaneto και Toth [18], προσέφεραν μια πρακτική λύση που υλοποιήθηκε σε FORTRAN. Η εργασία τους είναι αρκετά σημαντική καθώς προσαρμόζει τις θεωρητικές έννοιες που έχουν οι αλγόριθμοι του προβλήματος ανάθεσης σε αποδοτικό κώδικα, αποδεικνύοντας έτσι την αποτελεσματικότητα του αλγορίθμου τόσο για πυκνούς όσο και για αραιούς πίνακες. Αυτή η υλοποίηση είχε ένα σημείο αναφοράς για τη σύγκριση της απόδοσης διαφόρων αλγορίθμων, συμπεριλαμβανομένων και εκείνων που βασίζονται στην ουγγρική μέθοδο. Επιπλέον, υπογράμμισε τη σημασία των αποτελεσματικών πρακτικών κωδικοποίησης για την επίτευξη βέλτιστης υπολογιστικής απόδοσης. Περαιτέρω εξελίξεις περιέχουν οι Bertsekas και Castanon [15], οι οποίοι στη μελέτη τους συζήτησαν την παράλληλη ασύγχρονη εφαρμογή της ουγγρικής μεθόδου για την επίλυση του κλασικού προβλήματος ανάθεσης. Η προσαρμογή αυτή επιτρέπει την ταυτόχρονη προσπάθεια για πολλαπλές ενισχύσεις και αυξήσεις τιμών από διάφορες μη ανατεθείσες πηγές, ακόμη και με τη χρήση πιθανώς ξεπερασμένων πληροφοριών τόσο για τις τιμές όσο και για τις πληροφορίες ανάθεσης. Τα αποτελέσματα τους έδειξαν ότι αυτή η ασύγχρονη υλοποίηση μπορεί συχνά να είναι ταχύτερη από τη σύγχρονη αντίστοιχη υλοποίηση, μειώνοντας έτσι τις ποινές συγχρονισμού και καθιστώντας την αποτελεσματική προσέγγιση για προβλήματα μεγάλης κλίμακας σε μηχανές που έχουν κοινή μνήμη. Αυτές οι συνεισφορές έχουν εδραιώσει τη θέση που κατέχει ο ουγγρικού αλγορίθμου ως ένα θεμελιώδες εργαλείο στην επιχειρησιακή έρευνα και στη βελτιστοποίηση.

2.2 Auction Algorithm

Ο αλγόριθμος δημοπρασίας, που αναπτύχθηκε από τον Dimitri P. Bertsekas στα τέλη της δεκαετίας του 1970 [10] και περιγράφεται λεπτομερώς στην εργασία του [11], είναι μια μέθοδος κατανεμημένης χαλάρωσης που έχει σχεδιαστεί για την επίλυση του κλασικού προβλήματος ανάθεσης. Αναγνωρισμένος για την ισχυρή παραλληλότητά του, ο Αλγόριθμος Δημοπρασίας είναι κατάλληλος για υλοποίηση σε συστήματα παράλληλων υπολογιστών, επιτρέποντάς του να χειρίζεται προβλήματα ανάθεσης μεγάλης κλίμακας με μεγαλύτερη αποτελεσματικότητα από τις παραδοσιακές μεθόδους. Ο αλγόριθμος δημοπρασίας λειτουργεί σαν μια ανταγωνιστική διαδικασία υποβολής προσφορών, όπου οι πράκτορες, όπως εργασίες ή άτομα, υποβάλλουν προσφορές για πόρους, για παράδειγμα θέσεις εργασίας ή αντικείμενα. Τα βασικά βήματα του αλγορίθμου περιλαμβάνουν:

1. **Αρχικοποίηση**: Ο κάθε εκπρόσωπος ξεκινά με μια αρχική τιμή και μια αρχική ανάθεση.

2. **Φάση υποβολής προσφορών**: Οι εκπρόσωποι που δεν έχουν ανατεθεί υποβάλλουν προσφορά για πόρους αυξάνοντας την τιμή, με στόχο εξασφαλίσουν τον επιθυμητό πόρο.

3. **Φάση ανάθεσης**: Οι πόροι ανατίθενται στον πλειοδότη, και γίνεται συνεχή προσαρμογή των αναθέσεων και των τιμών.

4. **Επαναληπτική διαδικασία**: Η διαδικασία επαναλαμβάνεται, με τους εκπροσώπους να προσαρμόζουν συνεχώς τις προσφορές και τις αναθέσεις τους, μέχρι να μην μπορούν να γίνουν περαιτέρω βελτιώσεις και να επιτευχθεί η βέλτιστη ανάθεση.

Ο αλγόριθμος μπορεί να υλοποιηθεί τόσο συγχρονισμένα όσο και ασύγχρονα, με τον ασύγχρονο τρόπο να επιτρέπει μεγαλύτερο παραλληλισμό και αποτελεσματικότητα σε κατανεμημένα υπολογιστικά περιβάλλοντα. Ο αλγόριθμος δημοπρασίας χαρακτηρίζεται από τον υψηλό παραλληλισμό του, καθιστώντας τον κατάλληλο για υλοποίηση σε συστήματα παράλληλων υπολογιστών. Αυτή η ιδιότητα επιτρέπει στον αλγόριθμο να χειρίζεται προβλήματα κατανομής μεγάλης κλίμακας πιο αποτελεσματικά από τις παραδοσιακές μεθόδους. Ο αλγόριθμος μπορεί να ερμηνευθεί ως μια μέθοδος χαλάρωσης τύπου Jacobi για την επίλυση ενός δυϊκού προβλήματος, όπου κάθε αντιπρόσωπος προσαρμόζει επαναληπτικά την τιμή του με βάση τοπικές πληροφορίες, με στόχο τη βελτίωση της συνολικής λύσης.

Η υπολογιστική πολυπλοκότητα του αλγορίθμου δημοπρασίας, ιδίως όταν έχουμε κλιμάκωση, είναι $O(NAlog(NC))$, όπου N είναι ο αριθμός των ατόμων, A είναι ο αριθμός από ζεύγη ατόμων και αντικειμένων, και C είναι η μέγιστη απόλυτη τιμή ενός αντικειμένου. Η πολυπλοκότητα αυτή καθιστά αλγόριθμο ιδιαίτερα ανταγωνιστικό σε σχέση με τις άλλες υπάρχουσες μεθόδους, ιδίως για μεγάλα προβλήματα. Επιπλέον, ο αλγόριθμος παρουσιάζει σημαντική επιτάχυνση όταν εκτελείται σε παράλληλες μηχανές.

Αρκετές εργασίες έχουν διερευνήσει διάφορες πτυχές του αλγορίθμου δημοπρασίας είναι:

- **Οι Bertsekas και Castañon [14]**, όπου συζήτησαν διάφορες σύγχρονες και ασύγχρονες υλοποιήσεις σε μηχανές με κοινή μνήμη, τονίζοντας τα συμβιβαστικά μεταξύ συγχρονισμού και υπολογιστικής απόδοσης.
- **Οι Bertsekas και Castañon [13]**, οι οποίοι επέκτειναν τον Αλγόριθμο Δημοπρασίας για την επίλυση γραμμικών προβλημάτων μεταφοράς μετατρέποντάς τα σε προβλήματα ανάθεσης και εκμεταλλευόμενοι την ειδική δομή τους.
- **Ο Bertsekas [10]**, όπου παρείχε μια λεπτομερή εξέταση των θεμελιωδών πτυχών και της κατανεμημένης φύσης του αλγορίθμου, ενισχύοντας έτσι τη δυνατότητα εφαρμογής του σε παράλληλους υπολογισμούς.
- **Ο Bertsekas [12]**, με την παρουσίαση μιας ασύγχρονης μεθόδου για προβλήματα ροής δικτύου, αποδεικνύοντας περαιτέρω την ευελιξία του αλγορίθμου.
- **Οι Wein και Zenios [40]**, οι οποίοι μελέτησαν την απόδοση του αλγορίθμου σε μαζικά παράλληλους υπολογιστές, επιβεβαιώνοντας την αποδοτικότητα και την επεκτασιμότητά του.

Η εργασία του Bertsekas [11] εισήγαγε έναν νέο αλγόριθμο, που σε ορισμένα σημεία ήταν παρόμοιος με την ουγγρική μέθοδο, αλλά διαφέρει από άλλες απόψεις. Αυτός ο νέος αλγόριθμος, ο οποίος μοιράζεται θεμελιώδεις αρχές με τον Αλγόριθμο Δημοπρασίας, εμφανίζει ανώτερη μέση υπολογιστική πολυπλοκότητα και ξεπερνά σταθερά την αποτελεσματική κωδικοποιημένη έκδοση της ουγγρικής μεθόδου σε αρκετά προβλήματα τυχαίας παραγωγής.

Συνολικά, ο αλγόριθμος δημοπρασίας έχει αποδειχθεί ότι είναι ένα ευέλικτο και αποτελεσματικό εργαλείο για την επίλυση προβλημάτων ανάθεσης, ιδίως σε περιβάλλοντα παράλληλων υπολογιστών, καθώς η ικανότητά του να χειρίζεται προβλήματα μεγάλης κλίμακας με σημαντική επιτάχυνση τον καθιστά μια πολύτιμη προσθήκη στον τομέα της συνδυαστικής βελτιστοποίησης.

2.3 The Shortest Path Algorithms

2.3.1 Successive Shortest Path Algorithm

Το 1982, ο Engquist [22] ανάλυσε έναν αλγόριθμο με σκοπό την επίλυση του προβλήματος ανάθεσης, ώστε να μετατραπεί σε μια σειρά απλούστερων προβλημάτων ροής δικτύου. Το όνομα αυτού είναι ο αλγόριθμος διαδοχικής συντομότερης διαδρομής.

Τα βήματα του αλγορίθμου έχουν ως εξής:

1. **Αρχικοποίηση**: Ξεκινάει με μια αρχική ανάθεση και κατασκευή ενός υπολειμματικού δικτύου.

2. **Εύρεση συντομότερης διαδρομής**: Έπειτα γίνεται χρήση τυπικών τεχνικών συντομότερης διαδρομής, όπως ο αλγόριθμος του Dijkstra, για τον εντοπισμό της συντομότερης διαδρομής στο υπολειμματικό δίκτυο.

3. **Ενίσχυση**: Επαύξηση της τρέχουσας ανάθεσης κατά μήκος του συντομότερου μονοπατιού που βρέθηκε στο προηγούμενο βήμα, ώστε να βελτιωθεί η συνολική λύση.

4. **Επανάληψη**: Γίνεται επανάληψη των βημάτων 2 και 3 έως ότου δεν υπάρχουν πλέον διαθέσιμα μονοπάτια επαύξησης στο εναπομείναν δίκτυο.

5. **Τερματισμός**: Η διαδικασία ολοκληρώνεται όταν δεν μπορούν να βρεθούν άλλα μονοπάτια επαύξησης, υποδεικνύοντας ότι έχει επιτευχθεί η βέλτιστη λύση.

Αυτή η μέθοδος είναι ιδιαίτερα αποτελεσματική για πυκνούς πίνακες κόστους όπου οι συντελεστές κόστους ποικίλλουν. Η υπολογιστική πολυπλοκότητα αυτού του αλγορίθμου είναι $O(n^3)$, καθιστώντας τον ανταγωνιστικό σε σχέση με άλλους αλγορίθμους ανάθεσης, όπως η ουγγρική μέθοδος.

2.3.2 Shortest Augmenting Path Algorithm

Ο Αλγόριθμος συντομότερης αυξητικής διαδρομής, που διερευνήθηκε εκτενώς από τους Derigs [21] και Jonker και Volgenant [29], εστιάζει στη γρήγορη εύρεση του συντομότερου μονοπατιού αύξησης σε ένα διμερές γράφημα που αντιπροσωπεύει το πρόβλημα ανάθεσης.

Τα βήματα του αλγορίθμου έχουν ως εξής:

1. **Αρχικοποίηση**: Γίνεται έναρξη μιας αρχικής εφικτής ανάθεσης.
2. **Υπολογισμός ετικετών/αποστάσεων**: Έπειτα, γίνεται χρήση τεχνικών επισήμανσης ή υπολογισμών απόστασης για τον εντοπισμό της συντομότερης διαδρομής από μια μη ανατεθειμένη γραμμή σε μια μη ανατεθειμένη στήλη στο διμερές γράφημα που αναπαριστά το πρόβλημα ανάθεσης.
3. **Εύρεση συντομότερου μονοπατιού**: Άμεση εύρεση του συντομότερου μονοπατιού αύξησης χρησιμοποιώντας τις αναγνωρισμένες ετικέτες ή αποστάσεις.
4. **Επαύξηση**: Επαύξηση της ανάθεσης κατά μήκος του συντομότερου μονοπατιού που βρέθηκε στο προηγούμενο βήμα.
5. **Ενημέρωση ετικετών/δυνατοτήτων**: Ενημέρωση των ετικετών ή των δυνατοτήτων με βάση τη νέα ανάθεση.
6. **Επανάληψη**: Επανάληψη των βημάτων 2 έως 5 έως ότου δεν μπορούν να βρεθούν άλλα μονοπάτια επαύξησης.
7. **Τερματισμός**: Η διαδικασία ολοκληρώνεται όταν δεν υπάρχουν άλλα επαυξητικά μονοπάτια, εξασφαλίζοντας ότι η τελική ανάθεση είναι βέλτιστη.

Η αποδοτικότητα του αλγορίθμου έχει επίσης πολυπλοκότητα $O(n^3)$ και έχει αποδειχθεί η απόδοση του είναι εξαιρετικά καλή τόσο για πυκνά όσο και για αραιά προβλήματα γραμμικής ανάθεσης [29].

2.3.3 Ομοιότητες

1. **Εύρεση διαδρομής**: Και οι δύο αλγόριθμοι επικεντρώνονται στην εύρεση επαυξητικών μονοπατιών για τη βελτίωση μιας αρχικής εφικτής ανάθεσης.
2. **Επαναληπτική βελτίωση**: Κάθε μέθοδος βελτιώνει επαναληπτικά την τρέχουσα ανάθεση με αύξηση κατά μήκος των συντομότερων μονοπατιών που βρέθηκαν, οδηγώντας τελικά σε μια βέλτιστη λύση.

3. **Αποδοτικότητα**: Και οι δύο αλγόριθμοι έχουν υπολογιστική πολυπλοκότητα $O(n^3)$, καθιστώντας τους κατάλληλους για την αποτελεσματική επίλυση προβλημάτων ανάθεσης μεγάλης κλίμακας.

2.3.4 Διαφορές

1. **Προσέγγιση της εύρεσης διαδρομής**:
   - **Αλγόριθμος διαδοχικής συντομότερης διαδρομής**: Χρησιμοποιεί ένα μοντέλο ροής δικτύου και βρίσκει επανειλημμένα τις συντομότερες διαδρομές στο εναπομείναν δίκτυο.
   - **Αλγόριθμος συντομότερης αυξητικής διαδρομής**: Προσδιορίζει άμεσα το συντομότερο μονοπάτι επαύξησης στο διμερές γράφημα χρησιμοποιώντας τεχνικές επισήμανσης.
2. **Αρχικοποίηση και επισήμανση**:
   - **Αλγόριθμος διαδοχικής συντομότερης διαδρομής**: Ξεκινά με μια αρχική εφικτή ανάθεση και κατασκευάζει ένα υπολειμματικό δίκτυο.
   - **Αλγόριθμος συντομότερης αυξητικής διαδρομής**: Διασφαλίζει ότι η διαδρομή που βρίσκεται είναι η συντομότερη μέσω εξελιγμένων τεχνικών επισήμανσης.
3. **Εφαρμογές και παραλλαγές**:
   - **Αλγόριθμος διαδοχικής συντομότερης διαδρομής**: Αποτελεσματικός σε περιβάλλοντα ροής δικτύου, όπως οι μεταφορές και τα εφοδιαστικά.
   - **Αλγόριθμος συντομότερης επαυξητικής διαδρομής**: Προσαρμόζεται τόσο σε πυκνά όσο και σε αραιά προβλήματα ανάθεσης, με εφαρμογές σε διάφορα σενάρια συνδυαστικής βελτιστοποίησης.

2.3.5 Εφαρμογή και δοκιμή

Και οι δύο αλγόριθμοι έχουν υλοποιηθεί σε διάφορες γλώσσες προγραμματισμού και έχουν δοκιμαστεί εις βάθος σε υπολογιστικά πειράματα. Οι Engquist [22] και Jonker & Volgenant [29] παρουσίασαν λεπτομερείς υλοποιήσεις και υπολογιστικά αποτελέσματα, δείχνοντας την αποτελεσματικότητα και την ανθεκτικότητα των αντίστοιχων αλγορίθμων τους για ένα ευρύ φάσμα περιπτώσεων προβλημάτων.

2.4 Dual Algorithms

Οι δυϊκοί αλγόριθμοι εστιάζουν στην επίλυση του δυϊκού της διατύπωσης του γραμμικού προγραμματισμού του προβλήματος ανάθεσης. Αυτοί οι αλγόριθμοι λειτουργούν με επαναληπτική προσαρμογή των δυϊκών (dual) μεταβλητών, ώστε η αντικειμενική συνάρτηση να είναι εφικτή και βελτιωμένη.

Οι Weintraub και Barahona [41], παρουσίασαν έναν δυϊκο αλγόριθμο για το πρόβλημα ανάθεσης, ο οποίος προσαρμόζει τις δυϊκές μεταβλητές με σκοπό να εντοπίσει τη βέλτιστη λύση. Το 1986 ο Goldfarb [25] παρουσίασε μερικούς αποδοτικούς αλγορίθμους δυϊκής απλής μεθόδου (dual simplex algorithms) για το πρόβλημα ανάθεσης, παρουσιάζοντας υπολογιστικά όρια $O(n^3)$. Επιπλέον, ο Balinski [4], πρότεινε μια ανταγωνιστική δυϊκή απλή μέθοδο για το πρόβλημα ανάθεσης, η οποία κατάφερε να διατηρήσει τη δυϊκή εφικτότητα επαναλαμβάνοντας τη βέλτιστη λύση, κάνοντας τη μια αρκετά ανταγωνιστική προσέγγιση.

2.4.1 Primal-Dual Algorithms

Οι πρωτεύοντες-δυϊκοί αλγόριθμοι (Primal-dual algorithms) εξετάζουν ταυτόχρονα τόσο την πρωτεύουσα (primal) όσο και τη δυϊκή (dual) διατύπωση του προβλήματος ανάθεσης, κάνοντας προσαρμογές για τη διατήρηση της εφικτής του λειτουργίας, τόσο για τις δύο περιπτώσεις όσο και για τη σταδιακή βελτίωση της λύσης.

Αρχικά, ο McGinnis [32], ανέπτυξε και παρουσίασε μια υπολογιστική σύγκριση ενός πρωτεύοντος-δυϊκού αλγορίθμου για το πρόβλημα της ανάθεσης, υπογραμμίζοντας την αποτελεσματικότητα και την ανθεκτικότητά του. Έπειτα, οι Carpaneto και Toth δημιούργησαν [19] πρωτεύοντες-δυϊκούς αλγορίθμους όπου ενσωματώθηκαν στοιχεία της Ουγγρικής μεθόδου και της συντομότερης επαυξητικής διαδρομής. Αυτοί οι αλγόριθμοι κάνουν μετασχηματισμό του πίνακα κόστους σε μια πιο αραιή μορφή. Έπειτα, λύνουν το αραιό πρόβλημα και στη συνέχεια επαληθεύουν τη βέλτιστη λύση, σε σχέση με το αρχικό πρόβλημα.

2.4.2 Other Primal or Dual Algorithms

Έχουν αναπτυχθεί αρκετοί άλλοι αλγόριθμοι που χρησιμοποιούν πρωτεύουσες ή δυϊκές μεθόδους για την επίλυση του προβλήματος ανάθεσης, ο καθένας με συγκεκριμένες στρατηγικές και βελτιστοποιήσεις.

Οι Balinski και Gomory [5] περιέγραφαν μια πρωτεύουσα μέθοδο (primal method), για τα προβλήματα ανάθεσης και μεταφοράς, παρέχοντας μια εφικτή ανάθεση σε κάθε στάδιο υπολογισμού. Αυτή η μέθοδος είναι δυϊκή ως προς τη Ουγγρική μέθοδο και προσφέρει συγκρίσιμα υπολογιστικά όρια. Έπειτα, ο Hung [26] παρουσίασε έναν πολυωνυμικό πρωτεύων απλό αλγόριθμο (primal simplex algorithm) για το πρόβλημα ανάθεσης. Ο αλγόριθμος αυτός μπορεί και παράγει έναν περιορισμένο αριθμό βάσεων πριν τελικά φτάσει στη βέλτιστη λύση. Το 1988, ο Akgül [2] εισήγαγε έναν διαδοχικό δυϊκό απλό αλγόριθμο (sequential dual simplex algorithm) που λειτουργεί με ισχυρά εφικτά δέντρα (strongly feasible trees), παρουσιάζοντας μια πολυωνυμική πολυπλοκότητα για ορθογώνια συστήματα. Τέλος, οι Akgül και Ekin [1] πρότειναν έναν αλγόριθμο δυϊκής εφικτής δασικής δομής (dual feasible forest algorithm) για το γραμμικό πρόβλημα ανάθεσης, ο οποίος διατηρεί τη δυϊκή εφικτότητα (dual feasibility) κατασκευάζοντας και διατηρώντας μια δασική δομή εντός του δυϊκού χώρου λύσεων.

2.5 Branch and Bound Algorithm

Ο αλγόριθμος Branch and Bound είναι μια ευέλικτη μέθοδος, που χρησιμοποιείται συνήθως για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης, συμπεριλαμβανομένου και του προβλήματος ανάθεσης. Διερευνά τον χώρο λύσεων συστηματικά, τον διαιρεί σε μικρότερα υπό-προβλήματα (διακλαδώσεις) και στη συνέχεια υπολογίζει κάποια όρια, με σκοπό την εξάλειψη των υπό-βέλτιστων λύσεων (οριοθέτηση), ώστε να εντοπίσει τη βέλτιστη λύση αποτελεσματικά.

Ο αλγόριθμος Branch and Bound περιλαμβάνει τα ακόλουθα βήματα:

1. **Αρχικοποίηση**: Έναρξη με μια αρχική εφικτή λύση και υπολογισμός του κόστους της.

2. **Διακλάδωση**: Χωρισμός του προβλήματος σε μικρότερα υπό-προβλήματα με τον καθορισμό αναθέσεων για μία ή περισσότερες μεταβλητές.

3. **Οριοθέτηση**: Υπολογισμός ενός κατώτερου ορίου για κάθε υπό-πρόβλημα. Περικοπή των υπό-προβλημάτων, των οποίων τα όρια είναι χειρότερα από την τρέχουσα καλύτερη λύση.

4. **Εξερεύνηση**: Επιλέξτε το πιο υποσχόμενο υπό-πρόβλημα και επαναλάβετε τα βήματα διακλάδωσης και οριοθέτησης.

5. **Βέλτιστη λύση**: Η καλύτερη γνωστή λύση είναι η βέλτιστη αφού διερευνηθούν ή περικόπτουν όλα τα υπό-προβλήματα.

Η αποτελεσματικότητα του αλγορίθμου εξαρτάται από την τεχνική οριοθέτησης και τη δομή του προβλήματος, καθιστώντας τον ιδιαίτερα χρήσιμο για μεγάλα και πολύπλοκα προβλήματα ανάθεσης, όπου άλλες μέθοδοι, όπως ο ουγγρικός αλγόριθμος, μπορεί να είναι λιγότερο αποτελεσματικές. Ο αλγόριθμος Branch and Bound χρησιμοποιείται ευρέως στην επιχειρησιακή έρευνα, τον προγραμματισμό και των logistic, λόγω της ευελιξίας και της ευρωστίας του.

Οι Violina et al. [38] ανέλυσαν τον αλγόριθμο Branch and Bound για την επίλυση προβλημάτων ανάθεσης εργασιών, αποδεικνύοντας την αποτελεσματικότητά του στο χειρισμό περιπτώσεων μεγάλης κλίμακας και την υπεροχή του έναντι των παραδοσιακών μεθόδων σε κάποια συγκεκριμένα σενάρια. Η μελέτη αυτή απέδειξε την ικανότητα του αλγορίθμου να βρίσκει αποτελεσματικά βέλτιστες λύσεις μέσω αποτελεσματικών στρατηγικών περικοπής και οριοθέτησης.

Ενώ η πολυπλοκότητα της χειρότερης περίπτωσης του αλγορίθμου Branch and Bound μπορεί να είναι εκθετική, οι πρακτικές υλοποιήσεις συχνά επιτυγχάνουν πολύ καλύτερες επιδόσεις μέσω αποτελεσματικών στρατηγικών περικοπής και οριοθέτησης.

Ο αλγόριθμος Branch and Bound είναι μια ισχυρή και αποδοτική προσέγγιση για την επίλυση προβλημάτων ανάθεσης, ικανή να διαχειρίζεται πολύπλοκες και μεγάλης κλίμακας περιπτώσεις με υψηλή ακρίβεια. Η προσαρμοστικότητα και η αποτελεσματικότητά του τον καθιστούν πολύτιμο εργαλείο στη συνδυαστική βελτιστοποίηση.

2.6 Threshold Assignment Algorithm

Ο αλγόριθμος ανάθεσης ορίων, που εισήχθη από τους Glover et al. [24], είναι μια πολυωνυμικά περιορισμένη μέθοδος που έχει σχεδιαστεί και αυτή για την αποτελεσματική επίλυση του προβλήματος ανάθεσης. Ο αλγόριθμος αυτός αξιοποιεί ένα νέο θεώρημα που σχετίζεται με label-correcting αλγορίθμους συντομότερης διαδρομής, παρέχοντας μια ταχύτερη εναλλακτική λύση σε σχέση με παραδοσιακές μεθόδους όπως ο πρωτεύοντας αλγόριθμος απλής μεθόδου.

Βασικές έννοιες και μεθοδολογία:

1. **Συντομότερη διαδρομή με διόρθωση επισήμανσης**: Ο αλγόριθμος χρησιμοποιεί μια προσέγγιση διόρθωσης επισήμανσης για τη διαχείριση των υπολογισμών της συντομότερης διαδρομής. Αυτό περιλαμβάνει τη δυναμική προσαρμογή των ετικετών (ή των αποστάσεων) για να διασφαλιστεί ότι η συντομότερη διαδρομή διατηρείται καθ' όλη τη διάρκεια της διαδικασίας.

2. **Τιμές ορίων (Threshold Values)**: Ο αλγόριθμος εισάγει την έννοια των τιμών ορίων, οι οποίες χρησιμοποιούνται για τον έλεγχο και τον περιορισμό του χώρου αναζήτησης για βέλτιστες αναθέσεις. Αυτά τα κατώτατα όρια βοηθούν στο να περιορισθεί ο χώρος αναζήτησης και να επικεντρωθεί μόνο στα μονοπάτια με τις μεγαλύτερες προοπτικές, μειώνοντας έτσι την υπολογιστική επιβάρυνση.

3. **Επαναληπτική διαδικασία**: Η μέθοδος επαναλαμβάνει τις πιθανές αναθέσεις, προσαρμόζοντας συνεχώς τις ετικέτες και τις τιμές ορίων για την εύρεση της βέλτιστης λύσης. Κάθε επανάληψη βελτιώνει την τρέχουσα λύση, κινούμενη προοδευτικά πιο κοντά στη βέλτιστη ανάθεση.

Ο αλγόριθμος ανάθεσης ορίων προσφέρει αρκετά αξιοσημείωτα πλεονεκτήματα:

- **Αποδοτικότητα**: Η χρήση των τιμών ορίων και των τεχνικών διόρθωσης ετικετών ενισχύει σημαντικά την υπολογιστική απόδοση, καθιστώντας τον ταχύτερο από τις παραδοσιακές μεθόδους, όπως ο αλγόριθμος primal simplex.

- **Πολυωνυμική δέσμευση**: Ο αλγόριθμος λειτουργεί σε πολυωνυμικά περιορισμένο χρονικό πλαίσιο, εξασφαλίζοντας τη δυνατότητα ανάπτυξης και εφαρμογής σε προβλήματα ανάθεσης μεγάλης κλίμακας.

- **Πρακτική απόδοση**: Τα αρχικά υπολογιστικά ευρήματα, μας δείχνουν ότι ο αλγόριθμος ανάθεσης ορίων έχει πολύ καλή απόδοση στην πράξη, ξεπερνώντας συχνά τις επιδόσεις άλλων καθιερωμένων μεθόδων.

Οι Glover et al. [24] παρείχαν ολοκληρωμένα υπολογιστικά ευρήματα που κατέδειξαν την ταχύτητα και την ανθεκτικότητα του αλγορίθμου σε σύγκριση με τον πρωτεύοντα αλγόριθμο απλής μεθόδου. Η εργασία τους έδειξε ότι ο αλγόριθμος ανάθεσης ορίων μπορούσε να χειριστεί αποτελεσματικά ένα ευρύ φάσμα προβλημάτων ανάθεσης, καθιστώντας τον πολύτιμο εργαλείο για πρακτικές εφαρμογές στην επιχειρησιακή έρευνα και τη συνδυαστική βελτιστοποίηση.

Η βασική καινοτομία του αλγορίθμου έγκειται στη χρήση τιμών ορίων για την αποδοτικότητα της αναζήτησης βέλτιστων λύσεων, γεγονός που τον καθιστά ιδιαίτερα αποτελεσματικό για προβλήματα μεγάλης κλίμακας.

2.7 Recursive Methods

Οι αναδρομικές μέθοδοι για την επίλυση προβλημάτων ανάθεσης αντιπροσωπεύουν μια κατηγορία αλγορίθμων που επιλύουν το πρόβλημα επεκτείνοντας προοδευτικά τις διαστάσεις του. Αυτές οι μέθοδοι είναι πολυωνυμικά περιορισμένες και προσφέροντας έτσι μια μοναδική προσέγγιση σε σύγκριση με τις πιο παραδοσιακές τεχνικές συνδυαστικής βελτιστοποίησης, επιτρέποντάς τους να λύσουν αποτελεσματικά μια σειρά προβλημάτων κατανομής.

Ο Thompson [23] εισήγαγε μια αναδρομική προσέγγιση για την επίλυση προβλημάτων ανάθεσης, παρέχοντας μια εναλλακτική στην παραδοσιακή προσέγγιση στους πολυωνυμικούς περιορισμούς. Ο αλγόριθμος βρίσκει πρώτα την καλύτερη λύση στο πρόβλημα που ορίζεται από την πρώτη σειρά και μετά περιλαμβάνει διαδοχικά τις άλλες σειρές. Η μέθοδος έχει πολυπλοκότητα $O(n^3)$ και έχει αποδειχθεί ότι είναι ανταγωνιστική με άλλους αλγόριθμους κατά την αντιμετώπιση διαφόρων τύπων προβλημάτων ανάθεσης. Η εκτεταμένη υπολογιστική εμπειρία του Thompson μας έχει δείξει την αποτελεσματικότητα του αλγορίθμου σε διάφορα συστήματα υπολογιστών, αναδεικνύοντας την πρακτική εφαρμογή του.

Ο αυξητικός χαρακτήρας του αλγορίθμου του επιτρέπει να χειρίζεται αποτελεσματικά μεγάλες περιπτώσεις προβλημάτων, με επιδόσεις συγκρίσιμες με άλλους πολυωνυμικά περιορισμένους αλγόριθμους, όπως είναι η ουγγρική μέθοδος.

Οι αναδρομικές μέθοδοι παρέχουν ισχυρές και αποτελεσματικές τεχνικές προσέγγισης, για την επίλυση προβλημάτων ανάθεσης. Με την προοδευτική επέκταση των διαστάσεων του προβλήματος και τη σταδιακή επίλυση, οι μέθοδοι αυτές επιτυγχάνουν βέλτιστες λύσεις με πολυωνυμική χρονική πολυπλοκότητα. Το έργο του Thompson σε αυτόν τον τομέα έχει προσφέρει πολύτιμες γνώσεις και πρακτικά εργαλεία, αποδεικνύοντας έτσι την αποτελεσματικότητα των αναδρομικών μεθόδων στη συνδυαστική βελτιστοποίηση. Οι μέθοδοι αυτές είναι ιδιαίτερα κατάλληλες για μεγάλης κλίμακας και πολύπλοκα προβλήματα ανάθεσης, εξασφαλίζοντας ισχυρές και αποτελεσματικές λύσεις.

2.8 Relaxation Techniques

Οι τεχνικές χαλάρωσης αποτελούν ένα σύνολο μεθόδων που χρησιμοποιούνται για την επίλυση του προβλήματος ανάθεσης, με σκοπό να το μετατρέψουν σε ένα απλούστερο πρόβλημα, το οποίο μπορεί στη συνέχεια να επιλυθεί πιο αποτελεσματικά. Αυτές οι τεχνικές συνήθως περιλαμβάνουν τη χαλάρωση των περιορισμών ή την τροποποίηση της δομής του προβλήματος για να επιτρέψουν απλούστερες λύσεις, οι οποίες στη συνέχεια μπορούν να βελτιωθούν για να επιτευχθεί βελτιστοποίηση του αρχικού προβλήματος.

Οι τεχνικές χαλάρωσης είναι ιδιαίτερα χρήσιμες για μεγάλα και πολύπλοκα προβλήματα ανάθεσης, όπου οι παραδοσιακές μέθοδοι μπορεί να είναι υπολογιστικά δαπανηρές. Με την αποσύνθεση του προβλήματος και την επίλυση μιας σειράς απλούστερων υποπροβλημάτων, αυτές οι τεχνικές μπορούν να παρέχουν αποτελεσματικές και επεκτάσιμες λύσεις. Χρησιμοποιούνται ευρέως σε διάφορους τομείς όπως η επιχειρησιακή έρευνα, η επιμελητεία και ο σχεδιασμός.

Το 1980, οι Hung και Rom [27] παρουσίασαν έναν αλγόριθμο χαλάρωσης για την επίλυση του προβλήματος ανάθεσης μετατρέποντάς το σε μια σειρά απλών προβλημάτων ροής δικτύου. Η προσέγγισή τους εκμεταλλεύεται τις πρωτεύουσες και δυϊκές ιδιότητες του προβλήματος ανάθεσης, παρέχοντας ένα υπολογιστικό όριο $O(n^3)$ και επιδεικνύοντας καλύτερους μέσους χρόνους υπολογισμού σε σύγκριση με τους περισσότερους εξειδικευμένους αλγορίθμους ανάθεσης. Έπειτα, ο Bertsekas [9] εισήγαγε τον αλγόριθμο δημοπρασίας το 1988, ο οποίος μπορεί να θεωρηθεί ως μια μορφή τεχνικής χαλάρωσης. Αυτός ο αλγόριθμος λειτουργεί όπως μια δημοπρασία, όπου οι μη ανατεθειμένοι πράκτορες υποβάλλουν προσφορές για εργασίες, προσαρμόζοντας επαναληπτικά τις προσφορές τους μέχρι να η βέλτιστη ανάθεση να πραγματοποιηθεί. Η μέθοδος αυτή έχει την δυνατότητα να γίνει παράλληλη και είναι αποτελεσματική για προβλήματα μεγάλης κλίμακας.

Η υπολογιστική πολυπλοκότητα των τεχνικών χαλάρωσης ποικίλλει ανάλογα με τη συγκεκριμένη μέθοδο που χρησιμοποιείται. Γενικά, οι τεχνικές αυτές έχουν σχεδιαστεί για να επιτυγχάνουν πολυωνυμική χρονική πολυπλοκότητα, γεγονός που τις καθιστά κατάλληλες για μεγάλες περιπτώσεις προβλημάτων. Για παράδειγμα, ο αλγόριθμος χαλάρωσης των Hung και Rom [27] έχει πολυπλοκότητα $O(n^3)$, ευθυγραμμιζόμενος με άλλους αποδοτικούς αλγορίθμους ανάθεσης.

2.9 Algebraic Approach

Η αλγεβρική προσέγγιση, για την επίλυση προβλημάτων ανάθεσης, εισήχθη από τους Burkard et al. [17] και χρησιμοποιεί αλγεβρικές δομές για την αντιμετώπιση ενός ευρύτερου φάσματος προβλημάτων συνδυαστικής βελτιστοποίησης πέραν της παραδοσιακής γραμμικής ανάθεσης.

Η αλγεβρική προσέγγιση βασίζεται στην έννοια μιας ολικής εντολής ανταλλακτικής ημι-ομάδας, η οποία επιτρέπει μια πιο ευέλικτη αναπαράσταση των δομών κόστους στα προβλήματα ανάθεσης. Η μέθοδος περιλαμβάνει τα ακόλουθα βασικά βήματα:

1. **Αναπαράσταση αντικειμενικής συνάρτησης**: Το πρόβλημα ανάθεσης αναπαρίσταται χρησιμοποιώντας μια αλγεβρική αντικειμενική συνάρτηση. Η συνάρτηση αυτή ορίζεται εντός μιας πλήρως διατεταγμένης αντιμεταθετικής ημι-ομάδας, η οποία γενικεύει τις συμβατικές γραμμικές συναρτήσεις κόστους.

2. **Αλγεβρικές πράξεις**: Οι πράξεις εντός της ημι-ομάδας, όπως η πρόσθεση και ο πολλαπλασιασμός, χρησιμοποιούνται για τη συγκέντρωση του κόστους και τη σύγκριση διαφορετικών αναθέσεων. Οι πράξεις αυτές πρέπει να τηρούν τις αλγεβρικές ιδιότητες της αντιμεταθετικότητας και της προσεταιριστικότητα.

3. **Μετασχηματισμός και λύση**: Το πρόβλημα μετατρέπεται σε μια σειρά αλγεβρικών εξισώσεων ή ανισώσεων, οι οποίες μπορούν να επιλυθούν με τη

χρήση πολυωνυμικά περιορισμένων αλγορίθμων. Αυτοί οι μετασχηματισμοί αξιοποιούν τις αλγεβρικές ιδιότητες για να απλοποιήσουν το πρόβλημα και να επιτρέψουν τον αποδοτικό υπολογισμό.

4. **Γενίκευση σε άλλα προβλήματα**: Αυτή η προσέγγιση είναι ευέλικτη και μπορεί να εφαρμοστεί σε διάφορους τύπους προβλημάτων ανάθεσης, συμπεριλαμβανομένων των γραμμικών προβλημάτων συμφόρησης, των λεξικογραφικών πολυκριτηριακών προβλημάτων και των προβλημάτων ανάθεσης p-norm. Το αλγεβρικό πλαίσιο επιτρέπει τη μοντελοποίηση διαφορετικών αντικειμενικών συναρτήσεων και περιορισμών με ενιαίο τρόπο.

Η αλγεβρική προσέγγιση παρέχει αρκετά πλεονεκτήματα:

1. **Ευελιξία**: Με τη χρήση αλγεβρικών δομών, η μέθοδος αυτή μπορεί να χειριστεί μια ευρύτερη ποικιλία συναρτήσεων κόστους και περιορισμών, καθιστώντας την εφαρμόσιμη σε ένα ευρύτερο φάσμα προβλημάτων ανάθεσης.

2. **Αποτελεσματικότητα**: Οι αλγεβρικοί μετασχηματισμοί και πράξεις οδηγούν σε πολυωνυμικά περιορισμένους αλγορίθμους, εξασφαλίζοντας ότι τα προβλήματα μπορούν να επιλυθούν αποτελεσματικά ακόμη και για μεγάλες περιπτώσεις.

3. **Γενίκευση**: Η προσέγγιση αυτή δεν περιορίζεται σε γραμμικό κόστος ή σε συγκεκριμένους τύπους προβλημάτων. Μπορεί να προσαρμοστεί για την αντιμετώπιση διαφόρων προβλημάτων συνδυαστικής βελτιστοποίησης που μοιράζονται παρόμοιες αλγεβρικές ιδιότητες.

Οι Burkard et al. [17] ήταν οι πρωτοπόροι αυτής της προσέγγισης. Η εργασία τους παρέχει ένα ολοκληρωμένο πλαίσιο για την επίλυση προβλημάτων ανάθεσης χρησιμοποιώντας αλγεβρικές μεθόδους. Αποδεικνύουν ότι πολλά κοινά προβλήματα ανάθεσης μπορούν να μοντελοποιηθούν αποτελεσματικά και να λυθούν χρησιμοποιώντας ένα αλγεβρικό πλαίσιο. Οι συγγραφείς δείχνουν επίσης ότι αυτή η προσέγγιση μπορεί να επεκταθεί και σε άλλα προβλήματα συνδυαστικής βελτιστοποίησης, διευρύνοντας έτσι το εύρος των εφαρμογών της.

Η αλγεβρική προσέγγιση στα προβλήματα ανάθεσης που προτάθηκε από τους Burkard et al. [17] ήταν μια σημαντική πρόοδος στον τομέα της συνδυαστικής βελτιστοποίησης. Με την αξιοποίηση των αλγεβρικών δομών, αυτή η μέθοδος παρέχει έναν ευέλικτο και αποτελεσματικό τρόπο επίλυσης διαφόρων προβλημάτων ανάθεσης. Η γενικευσιμότητα της και οι πολυωνυμικά περιορισμένοι αλγόριθμοι την καθιστούν πολύτιμο εργαλείο τόσο για τη θεωρητική έρευνα όσο και για τις πρακτικές εφαρμογές της βελτιστοποίησης.

2.10 Ones Assignment Method

Η εργασία του Basirzadeh [7], εισάγει τη Μέθοδο Ανάθεσης Μονάδων, μια αρκετά νέα προσέγγιση για την επίλυση προβλημάτων ανάθεσης. Η μέθοδος αυτή απλοποιεί τη διαδικασία εύρεσης της βέλτιστης ανάθεσης μετασχηματίζοντας τον πίνακα κόστους με μοναδικό τρόπο.

Η Μέθοδος Ανάθεσης Μονάδων περιλαμβάνει τα ακόλουθα βασικά βήματα:

1. **Μετασχηματισμός του πίνακα κόστους**: Η μέθοδος μετασχηματίζει τα στοιχεία του πίνακα κόστους σε μονάδες. Ο μετασχηματισμός αυτός έχει σχεδιαστεί για να απλοποιήσει τους επακόλουθους υπολογισμούς που απαιτούνται για την εύρεση της βέλτιστης ανάθεσης.

2. **Απλοποιημένοι υπολογισμοί**: Με τη μετατροπή του πίνακα κόστους σε μονάδες, η μέθοδος μειώνει την πολυπλοκότητα του προβλήματος, καθιστώντας την επίλυσή του ευκολότερη και ταχύτερη. Αυτό περιλαμβάνει βασικές αριθμητικές πράξεις και λογικά βήματα που βελτιστοποιούν τη διαδικασία.

3. **Επαναληπτική διαδικασία**: Η μέθοδος προσαρμόζει επαναληπτικά τις αναθέσεις με βάση τον μετασχηματισμένο πίνακα κόστους μέχρι να βρεθεί η βέλτιστη λύση. Η διαδικασία αυτή διασφαλίζει ότι η τελική ανάθεση ελαχιστοποιεί αποτελεσματικά το συνολικό κόστος.

Η Μέθοδος Ανάθεσης Μονάδων προσφέρει πολλά πλεονεκτήματα:

- **Αποδοτικότητα**: Ο μετασχηματισμός σε μονάδες απλοποιεί τις σχετικές μαθηματικές πράξεις, οδηγώντας ενδεχομένως σε ταχύτερους και ευκολότερους υπολογισμούς σε σύγκριση με τις παραδοσιακές μεθόδους.

- **Ευελιξία**: Η μέθοδος αυτή είναι ευέλικτη και μπορεί να εφαρμοστεί σε ένα ευρύ φάσμα προβλημάτων ανάθεσης, καθιστώντας τη ένα ευέλικτο εργαλείο για διάφορα σενάρια.

- **Αποτελεσματικότητα**: Τα υπολογιστικά αποτελέσματα που παρέχει ο Basirzadeh αποδεικνύουν την αποτελεσματικότητα της μεθόδου στην αποτελεσματική εύρεση βέλτιστων λύσεων.

Η μέθοδος του Basirzadeh [7] ξεχωρίζει για την απλότητα και την καινοτομία της. Ο μετασχηματισμός του πίνακα κόστους σε μονάδες αντιπροσωπεύει μια μοναδική προσέγγιση που μειώνει τον υπολογιστικό φόρτο και ενισχύει την αποτελεσματικότητα. Η μέθοδος ανάθεσης με μονάδες παρέχει μια εναλλακτική λύση σε πιο πολύπλοκους αλγορίθμους, προσφέροντας μια πρακτική και απλή λύση για την επίλυση προβλημάτων ανάθεσης.

2.11 Two New Effective Methods

Το 2020, η εργασία των Hussein και Shiker [28] παρουσίασε δύο νέες μεθόδους για την εύρεση βέλτιστων λύσεων σε προβλήματα ανάθεσης. Οι μέθοδοι αυτές αποσκοπούν στη βελτίωση τόσο της αποτελεσματικότητας όσο και της ακρίβειας της επίλυσης τέτοιων προβλημάτων, παρέχοντας πολύτιμες εναλλακτικές λύσεις στις υπάρχουσες τεχνικές.

1. **Πρώτη μέθοδος**: Αυτή η μέθοδος ενσωματώνει προηγμένες τεχνικές βελτιστοποίησης για τη βελτιστοποίηση της διαδικασίας ανάθεσης, εστιάζοντας στη μείωση του υπολογιστικού χρόνου, διασφαλίζοντας παράλληλα τη βέλτιστη λειτουργία. Οι ιδιαιτερότητες του αλγορίθμου περιλαμβάνουν καινοτόμους τρόπους χειρισμού των πινάκων κόστους και των περιορισμών ανάθεσης.

2. **Δεύτερη μέθοδος**: Η δεύτερη μέθοδος βασίζεται σε ευρετικές προσεγγίσεις για να συγκλίνει γρήγορα σε σχεδόν βέλτιστες λύσεις. Η μέθοδος αυτή αξιοποιεί πιθανολογικές τεχνικές για την αποτελεσματική εξερεύνηση του χώρου λύσεων, γεγονός που την καθιστά ιδιαίτερα αποτελεσματική για προβλήματα μεγάλης κλίμακας όπου οι παραδοσιακές μέθοδοι μπορεί να παραπαίουν.

Και στις δυο μέθοδούς έγινε δοκιμή σε μια ποικιλία προβλημάτων ανάθεσης και στην απόδοσή τους, η οποία αξιολογήθηκε με βάση τον υπολογιστικό χρόνο, την ποιότητα των λύσεων και την επεκτασιμότητα.

Η μελέτη διαπιστώνει ότι και οι δύο μέθοδοι υπερτερούν σημαντικά έναντι των υφιστάμενων αλγορίθμων όσον αφορά την ταχύτητα και την ακρίβεια για τα εξεταζόμενα σενάρια. Η πρώτη μέθοδος, ειδικότερα, επιδεικνύει σημαντική βελτίωση της υπολογιστικής απόδοσης, καθιστώντας την κατάλληλη για εφαρμογές πραγματικού χρόνου. Η δεύτερη μέθοδος, αν και ελαφρώς λιγότερο βέλτιστη σε ορισμένες περιπτώσεις, προσφέρει ισχυρή απόδοση σε ένα εύρος μεγεθών και πολυπλοκότητας προβλημάτων.

Αυτές οι μέθοδοι παρέχουν πολύτιμες εναλλακτικές λύσεις στους παραδοσιακούς αλγόριθμους, ενισχύοντας την ικανότητα αποτελεσματικής επίλυσης πολύπλοκων και μεγάλης κλίμακας προβλημάτων κατανομής.

2.12 Alternating Basis Algorithm

Η εργασία των Barr et al. [6] εισήγαγε τον αλγόριθμο εναλλασσόμενης βάσης, έναν νέο πρωτόγονο αλγόριθμο ακραίων σημείων που σχεδιάστηκε για την επίλυση προβλημάτων ανάθεσης. Η μέθοδος αυτή αντιμετωπίζει τα ζητήματα εκφυλισμού στις παραδοσιακές μεθόδους simplex εστιάζοντας σε ένα υποσύνολο βάσεων που οδηγούν σε βέλτιστες λύσεις.

Ο αλγόριθμος εναλλασσόμενης βάσης λειτουργεί με την παράκαμψη και την εκμετάλλευση του εκφυλισμού. Οι παραδοσιακές μέθοδοι simplex αντιμετωπίζουν συχνά δυσκολίες λόγω της επιθεώρησης πολλαπλών αναπαραστάσεων ισοδύναμων βάσεων των ακραίων σημείων. Ο Αλγόριθμος Εναλλασσόμενης Βάσης το αποφεύγει αυτό:

1. **Χαρακτηρισμός των βάσεων**: Προσδιορισμός ενός υποσυνόλου Q όλων των πιθανών βάσεων που μπορεί να οδηγήσει σε βέλτιστη λύση, εάν υπάρχει. Αυτός ο χαρακτηρισμός βοηθά στον περιορισμό του χώρου αναζήτησης.

2. **Αλγόριθμος ακραίων σημείων**: Ανάπτυξη αλγορίθμου ακραίου σημείου που εξετάζει μόνο τις βάσεις του υποσυνόλου Q. Αυτή η εστιασμένη προσέγγιση μειώνει τους περιττούς υπολογισμούς και βελτιώνει την αποτελεσματικότητα.

3. **Διαχείριση εκφυλισμού**: Αποτελεσματική διαχείριση του εκφυλισμού διασφαλίζοντας ότι εξετάζονται μόνο οι σχετικές βάσεις, βελτιώνοντας έτσι την υπολογιστική διαδικασία.

Ο αλγόριθμος εναλλασσόμενης βάσης έχει σημαντική βελτίωση στην απόδοση σε σύγκριση με την παραδοσιακή μέθοδο simplex. Τα αποτελέσματα δείχνουν ότι ο νέος αλγόριθμος είναι σημαντικά αποτελεσματικός, ειδικά όσον αφορά τον χειρισμό του εκφυλισμού και την επίτευξη ταχύτερης σύγκλισης προς τη βέλτιστη λύση. Ο κεντρικός έλεγχος των αντίστοιχων βάσεων μειώνει το υπολογιστικό φορτίο και βελτιώνει την απόδοση.

2.13 Signature Methods

Το άρθρο του Balinski [3] εισάγει τη χαρακτηριστική μέθοδο, μια νέα προσέγγιση για την επίλυση προβλημάτων ανάθεσης χρησιμοποιώντας διανυσματικές αναπαραστάσεις διπλών εφαρμοστέων βάσεων. Η μέθοδος στοχεύει στη βελτίωση της αποτελεσματικότητας και της συστηματικής επίλυσης προβλημάτων κατανομής μέσω μιας δομημένης αλγοριθμικής διαδικασίας.

Η Χαρακτηριστικές Μέθοδος (Signature Method) βασίζεται στην έννοια της χαρακτηριστικής (concept of a signature), η οποία ορίζεται ως ένα διάνυσμα n του οποίου η i-οστή συνιστώσα είναι ο αριθμός των μη βασικών δραστηριοτήτων τύπου (i,j) σε μια δυϊκή εφικτή βάση. Η μεθοδολογία περιλαμβάνει:

1. **Ορισμός χαρακτηρισμού**: Χρήση του χαρακτηριστικού δείκτη μιας δυϊκής εφικτής βάσης για τον χαρακτηρισμό της κατάστασης του προβλήματος ανάθεσης (Utilizing the signature of a dual feasible basis to characterize the state of the assignment problem). Αυτός ο χαρακτηρισμός βοηθά στον εντοπισμό της βέλτιστης διαδρομής προς τη λύση.

2. **Βήματα περιστροφής (Pivot Steps)**: Ανάπτυξη δομημένου αλγορίθμου που τερματίζει σε πεπερασμένο αριθμό βημάτων περιστροφής (pivot), συγκεκριμένα το πολύ σε $(n-1)(n-2)/2$ βήματα. Αυτό εξασφαλίζει μια συστηματική προσέγγιση για την επίτευξη της βέλτιστης λύσης.

3. **Αποδοτικότητα**: Υλοποίηση της χαρακτηριστικής μεθόδου με τέτοιο τρόπο ώστε να εκτελεί το πολύ $O(n^3)$ εργασίες, καθιστώντας την υπολογιστικά αποδοτική.

Η συγκεκριμένη μέθοδος προσφέρει ένα καθαρό και αποδοτικό πλαίσιο για την επίλυση προβλημάτων ανάθεσης. Η συστηματική εφαρμογή των χαρακτηριστικών δεικτών (signatures) μειώνει την πολυπλοκότητα του προβλήματος, επιταχύνοντας τη διαδικασία εύρεσης της βέλτιστης λύσης. Τα υπολογιστικά αποτελέσματα αποδεικνύουν ότι αυτή η μέθοδος είναι αποτελεσματική στην αντιμετώπιση προβλημάτων ανάθεσης μεγάλης κλίμακας, με περιορισμένο και προβλέψιμο αριθμό βημάτων.

2.14 Network Simplex Method

Η εργασία του Cunningham [20] παρουσιάζει τη Μέθοδο Δικτύου Simplex, μια εξειδικευμένη μέθοδο για την επίλυση προβλημάτων ροής δικτύου, συμπεριλαμβανομένου του προβλήματος ανάθεσης. Αυτή η μέθοδος τροποποιεί την παραδοσιακή μέθοδο simplex, προκειμένου να διαχειρίζεται αποτελεσματικά τα ιδιαίτερα χαρακτηριστικά των ροών δικτύου.

Η Μέθοδος Δικτύου Simplex περιλαμβάνει αρκετές καινοτομίες που διασφαλίζουν την περατότητα και την αποτελεσματικότητα στην επίλυση προβλημάτων ροής δικτύου:

1. **Ισχυρά εφικτές βάσεις (Strongly Feasible Bases)**: Η μέθοδος περιλαμβάνει τη διατήρηση "ισχυρά εφικτών" βάσεων καθ' όλη τη διάρκεια των επαναλήψεων. Αυτή η έννοια εξασφαλίζει ότι η μέθοδος primal simplex παραμένει πεπερασμένη και αποτελεσματική για ροές δικτύου.

2. **Επαναλήψεις Simplex**: Η μέθοδος χρησιμοποιεί επαναλήψεις simplex προσαρμοσμένες για δομές δικτύων, εστιάζοντας στην επιλογή και διατήρηση εφικτών βάσεων που οδηγούν σε βέλτιστες λύσεις χωρίς προβλήματα εκφυλισμού.

3. **Αλγόριθμος μετατροπής**: Παρέχεται ένας αποτελεσματικός αλγόριθμος για τη μετατροπή οποιασδήποτε εφικτής βάσης σε ισχυρά εφικτή. Αυτό εξασφαλίζει ότι η μέθοδος μπορεί να ξεκινήσει με οποιαδήποτε αρχική εφικτή λύση και να κινηθεί προοδευτικά προς τη βέλτιστη λύση.

Η Μέθοδος Δικτύου Simplex προσφέρει πολλά πλεονεκτήματα:

- **Αποδοτικότητα**: Η χρήση ισχυρά εφικτών βάσεων και εξειδικευμένων επαναλήψεων simplex καθιστά τη μέθοδο ιδιαίτερα αποτελεσματική για προβλήματα ροής δικτύου.

- **Πεπερασμένο**: Η διασφάλιση της πεπερατότητας μέσω της διατήρησης ισχυρά εφικτών βάσεων αποτρέπει την ανακύκλωση και εγγυάται τη σύγκλιση στη βέλτιστη λύση.

- **Κλιμακωσιμότητα**: Η μέθοδος είναι ιδιαίτερα αποτελεσματική για προβλήματα ροής δικτύου μεγάλης κλίμακας, συμπεριλαμβανομένων των προβλημάτων ανάθεσης, λόγω της δομημένης προσέγγισής της.

Κεφάλαιο 3

Υλοποίηση

3.1 MIP Solver

Ο μικτός ακέραιος προγραμματισμός (MIP) είναι μια ισχυρή μαθηματική τεχνική βελτιστοποίησης που επεκτείνει τον γραμμικό προγραμματισμό, επιτρέποντας σε κάποιες μεταβλητές απόφασης να λαμβάνουν ακέραιες τιμές. Αυτό το χαρακτηριστικό είναι ιδιαίτερα χρήσιμο σε προβλήματα που περιλαμβάνουν διακριτές αποφάσεις, όπως για παράδειγμα, η ανάθεση εργασιών σε πράκτορες, όπου οι μεταβλητές αντιπροσωπεύουν δυαδικές επιλογές, δηλαδή η εργασία είτε θα ανατίθεται είτε δεν ανατίθεται [42] [35].

Ο MIP μπορεί να μοντελοποιήσει με ακρίβεια τη δυαδική φύση των αναθέσεων εργασιών, κάνοντας τον ιδιαίτερα κατάλληλο για την επίλυση του προβλήματος ανάθεσης. Η ευελιξία που έχει ο μικτός ακέραιος προγραμματισμός, του επιτρέπει να ενσωματώνει πρόσθετους περιορισμούς, όπως περιορισμούς χωρητικότητας ή προτεραιότητας, καθιστώντας τον ένα ευέλικτο εργαλείο για την εύρεση βέλτιστων λύσεων σε σύνθετα σενάρια ανάθεσης. Μέσω της εφαρμογής του, η επίλυση του προβλήματος ανάθεσης γίνεται με τέτοιον τρόπο ώστε να εξασφαλίζει τη βέλτιστη λύση, ακόμη και όταν η κλίμακα ή η πολυπλοκότητα του προβλήματος αυξάνεται [16].

Η υλοποίηση πραγματοποιήθηκε με τη χρήση της βιβλιοθήκης Google OR-Tools. Οι μεταβλητές απόφασης δημιουργούνται ως δυαδικές μεταβλητές, οι περιορισμοί προστίθενται ώστε κάθε εργασία να ανατίθεται σε έναν και μόνο πράκτορα, και κάθε πράκτορας να αναλαμβάνει μία εργασία. Τέλος, η συνάρτηση κόστους ελαχιστοποιείται, ώστε να επιτεύξουμε την εξασφάλιση της βέλτιστης λύσης.

```
Initialize model;
Initialize decision variables x;
for i from 0 to n - 1 do
    for j from 0 to n - 1 do
        Create a binary integer variable x[i, j];
    end
end
for i from 0 to n - 1 do
    Add constraint that sum of x[i, j] for j from 0 to n - 1 = 1;
end
for j from 0 to n - 1 do
    Add constraint that sum of x[i, j] for i from 0 to n - 1 = 1;
end
Initialize objective function;
for i from 0 to n - 1 do
    for j from 0 to n - 1 do
        Append costs[i][j] * x[i, j] to objective function;
    end
end
Set objective to minimize sum of objective function;
Solve the problem using Google OR-Tools MIP Solver;
```

Αλγόριθμος 1: MIP Solver

3.2 CP-SAT Solver

Στα πεδία της επιστήμης των υπολογιστών και της επιχειρησιακής έρευνας, ο Προγραμματισμός Περιορισμών (CP) ξεχωρίζει ως μια στιβαρή προσέγγιση για την αντιμετώπιση συνδυαστικών προβλημάτων. Ο CP επικεντρώνεται στον εντοπισμό λύσεων που πληρούν ένα προκαθορισμένο σύνολο περιορισμών που εφαρμόζονται στις μεταβλητές απόφασης, σε αντίθεση με τις συμβατικές μεθόδους βελτιστοποίησης, όπου στοχεύουν κυρίως στη μεγιστοποίηση ή την ελαχιστοποίηση μιας αντικειμενικής συνάρτησης. Αυτή η προσέγγιση καθιστά τον CP κατάλληλο για την αντιμετώπιση περίπλοκων προβλημάτων που αφορούν τον προγραμματισμό, την κατανομή πόρων και την ανάθεση εργασιών, ιδίως όταν οι σχέσεις μεταξύ των μεταβλητών είναι ιδιαίτερα δομημένες και δεσμεύονται από πολυάριθμους περιορισμούς [34].

Ο Προγραμματισμός Περιορισμών, υπερέχει στην αντιμετώπιση του προβλήματος ανάθεσης λόγω της ικανότητάς του να διαχειρίζεται τους δυαδικούς και λογικούς περιορισμούς που χαρακτηρίζουν την ανάθεση εργασιών. Όταν εφαρμόζεται στο πρόβλημα ανάθεσης, ο CP επιτρέπει στο μοντέλο να εφαρμόζει αυστηρούς κανόνες, όπως η ανάθεση κάθε εργασίας σε ακριβώς έναν πράκτορα και ο περιορισμός κάθε πράκτορα σε ακριβώς μία εργασία. Αυτό επιτυγχάνετε, μέσω περιορισμών και όχι πολύπλοκων αντικειμενικών συναρτήσεων. Αυτή η μέθοδος μπορεί να οδηγήσει σε πιο απλοποιημένες διαδικασίες επίλυσης, ειδικά όταν το πρόβλημα περιλαμβάνει περίπλοκους περιορισμούς, όπου μπορεί να είναι δύσκολο να εκφραστούν σε αμιγώς μαθηματικά πλαίσια βελτιστοποίησης, όπως για παράδειγμα, ο μικτός ακέραιος προγραμματισμός [37].

Η υλοποίηση πραγματοποιήθηκε μέσω της βιβλιοθήκης Google OR-Tools, με παρόμοιο τρόπο όπως στον MIP Solver. Ωστόσο, ο CP-SAT Solver χειρίζεται με διαφορετικό τρόπο τις δυαδικές μεταβλητές και τους περιορισμούς, προσφέροντας καλύτερη διαχείριση όταν το πρόβλημα περιλαμβάνει πολλούς και σύνθετους περιορισμούς.

```
Initialize solver;
Initialize decision variables x;
for i from 0 to n - 1 do
    for j from 0 to n - 1 do
        Create a binary decision variable x[i, j];
    end
end
for i from 0 to n - 1 do
    Add constraint that sum of x[i, j] for j from 0 to n - 1 = 1;
end
for j from 0 to n - 1 do
    Add constraint that sum of x[i, j] for i from 0 to n - 1 = 1;
end
Initialize objective function;
for i from 0 to n - 1 do
    for j from 0 to n - 1 do
        Append costs[i][j] * x[i, j] to objective function;
    end
end
Set objective to minimize sum of objective function;
Solve the problem using Google OR-Tools CP-SAT solver;
```

Αλγόριθμος 2: CP-SAT Solver

3.3 Hungarian Method

Ο Ουγγρικός Αλγόριθμος είναι μια κλασική μέθοδος για την επίλυση του προβλήματος ανάθεσης, η οποία παρέχει βέλτιστες λύσεις σε πολυωνυμικό χρόνο. Η υλοποίηση του αλγορίθμου έγινε μέσω της συνάρτησης linear_sum_assignment της βιβλιοθήκης scipy.optimize.

Η μέθοδος αυτή επιλύει το πρόβλημα ανάθεσης με το μικρότερο δυνατό κόστος και έχει τη δυνατότητα να βρίσκει τη βέλτιστη λύση με γρήγορο τρόπο ακόμα και για μεγάλα προβλήματα.

```
for i from 0 to n - 1 do
    Subtract the minimum value from row i in the cost matrix;
end
for j from 0 to n - 1 do
    Subtract the minimum value from column j in the cost matrix;
end
while True do
    Cover all zero elements in the cost matrix using the minimum number of
      horizontal and vertical lines;
    if the number of covering lines not equals n then
        Identify the smallest uncovered value;
        Subtract it from all uncovered elements;
        Add it to all elements covered twice;
        Cover all zeros again using the minimum number of lines;
    else
        break; // Found optimal solution
    end
end
Assign tasks to agents by selecting uncovered zeros, ensuring no row or
  column is used more than once;
Return the optimal assignment;
```

Αλγόριθμος 3: Hungarian Method

3.4 Jonker-Volgenant Αλγόριθμος

Ο αλγόριθμος Jonker-Volgenant (LAPJV) είναι μια πιο βελτιωμένη εκδοχή του αλγορίθμου συντομότερης επαυξητικής διαδρομής (Shortest Augmenting Path) και κυρίως χρησιμοποιείται για την επίλυση πυκνών προβλημάτων ανάθεσης [29]. Η πολυπλοκότητά του είναι $O(n^3)$, γεγονός που τον καθιστά ιδιαίτερα αποδοτικό για πυκνά προβλήματα. Σε αυτή τη διπλωματική, η υλοποίηση του αλγορίθμου LAPJV πραγματοποιήθηκε με τη χρήση της βιβλιοθήκης lap. Ο αλγόριθμος αυτός, όπως αποδείχθηκε, μπορεί να επιλύσει μόνο πυκνά προβλήματα, καθώς δεν είναι σχεδιασμένος για αραιά δεδομένα.

Ο αλγόριθμος υλοποιήθηκε με τα εξής βήματα:

- Αρχικοποίηση του πίνακα κόστους

- Υπολογισμός της βέλτιστης ανάθεσης χρησιμοποιώντας τη συνάρτηση lapjv
- Επιστροφή του συνολικού κόστους και του χρόνου εκτέλεσης

```
Initialize labels u and v to 0;
for i from 0 to n - 1 do
    for j from 0 to n - 1 do
        Calculate reduced cost r[i, j] = C[i, j] - u[i] - v[j];
    end
end
while there are unassigned rows do
    Find the row i without an assignment;
    Find the column j that minimizes r[i, j];
    if no zero is found in r then
        Adjust labels u and v based on the smallest uncovered value;
end
Return the optimal assignment;
```

Αλγόριθμος 4: Jonker-Volgenant Method

3.5 LAPMOD

Ο LAPMOD [39] είναι ένας αλγόριθμος που επίσης βασίζεται στον αλγόριθμο συντομότερης αυξητικής διαδρομής, αλλά έχει σχεδιαστεί για την επίλυση αραιών προβλημάτων ανάθεσης, σε αντίθεση με τον LAPJV.

Η υλοποίηση του αλγορίθμου LAPMOD πραγματοποιήθηκε επίσης με τη βιβλιοθήκη lap. Σε αυτόν τον αλγόριθμο έγιναν μερικές παραμετροποιήσεις, κάνοντας τους πυκνούς πίνακες αραιούς, ώστε ο αλγόριθμος να μπορεί να λειτουργήσει σωστά. Αυτό αποδείχθηκε απαραίτητο, καθώς ο LAPMOD δεν μπορεί να λύσει πυκνά προβλήματα χωρίς αυτήν την προεπεξεργασία.

Initialize labels u and v to 0;
for *i from* 0 *to* *n* - 1 **do**
    Subtract the minimum value from row i in the sparse cost matrix;
end
for *j from* 0 *to* *n* - 1 **do**
    Subtract the minimum value from column j in the sparse cost matrix;
end
while *assignment is not feasible* **do**
    Identify the smallest uncovered value;
    Subtract it from all uncovered elements;
    Add it to elements covered twice;
end
Return the optimal assignment;

Αλγόριθμος 5: LAPMOD Method

Κεφάλαιο 4

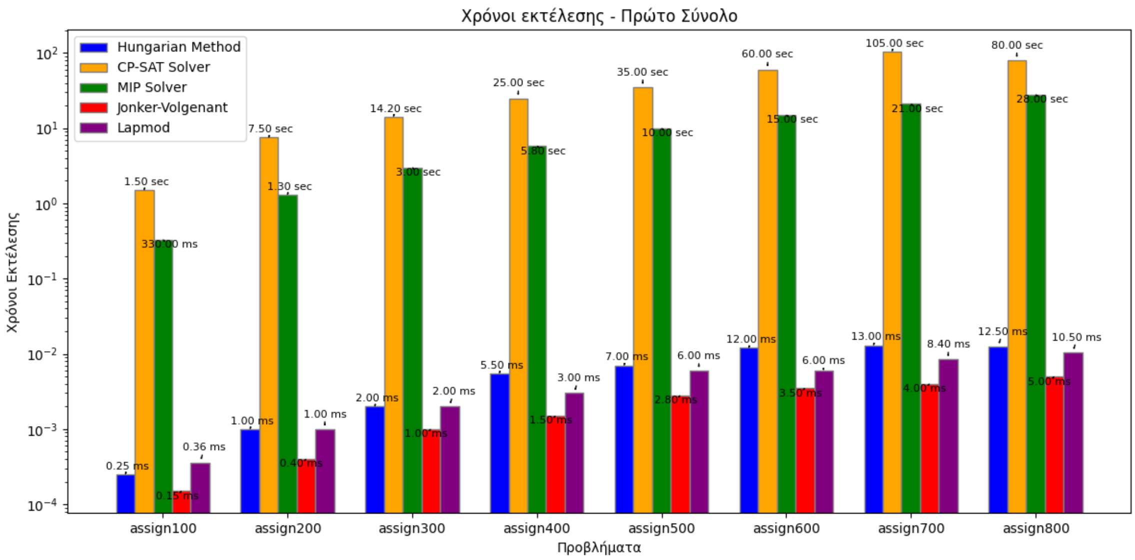
Υπολογιστική μελέτη

Η υλοποίηση του κώδικα έγινε με τη χρήση της γλώσσας προγραμματισμού Python. Για την επίλυση των προβλημάτων που εξετάζονται στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν δεδομένα από τη βιβλιοθήκη OR-Library. Η OR-Library παρέχει ένα σύνολο τυποποιημένων δεδομένων για διάφορα προβλήματα βελτιστοποίησης, συμπεριλαμβανομένου του προβλήματος ανάθεσης. Αυτά τα σύνολα δεδομένων επιλέχθηκαν λόγω της συχνής χρήσης τους στην υπάρχουσα βιβλιογραφία, επιτρέποντας τη σύγκριση μεταξύ των ευρημάτων της παρούσας διπλωματικής εργασίας και προηγούμενων μελετών.

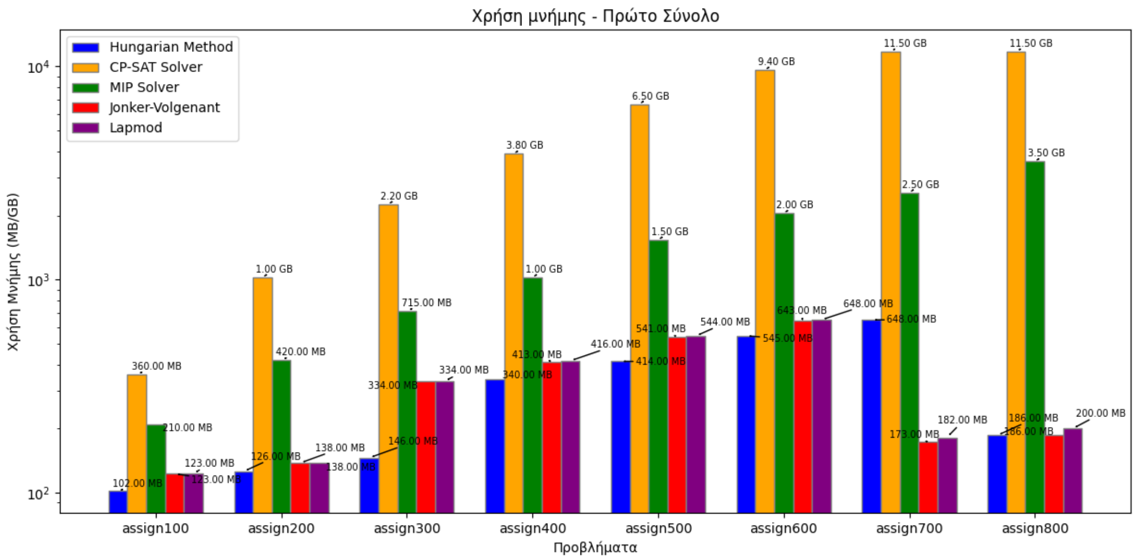
Όλοι οι αλγόριθμοι υλοποιήθηκαν και εκτελέστηκαν εντός ενός ενιαίου script, προκειμένου να διατηρηθεί η συνοχή κατά τη δοκιμή και τη συλλογή δεδομένων. Αυτό επέτρεψε μια ομαλή διαδικασία σύγκρισης του χρόνου εκτέλεσης και της χρήσης μνήμης των διαφορετικών solvers υπό πανομοιότυπες συνθήκες. Ενδέχεται να προκύψουν μικρές αποκλίσεις στη χρήση μνήμης, από τους κοινούς πόρους του συστήματος, κατά την εκτέλεση διαδοχικών αλγορίθμων, κάτι που είναι αναπόφευκτο όταν εκτελούνται πολλαπλοί solvers στο ίδιο script.

4.1 Πρώτο Σύνολο Προβλημάτων

Το πρώτο σύνολο προβλημάτων περιλαμβάνει οκτώ προβλήματα (assign100 έως assign800) που βασίζονται στον πίνακα 2 της εργασίας του J.E. Beasley, «Linear programming on Cray supercomputers», που δημοσιεύθηκε στο Journal of the Operational Research Society [8]. Αυτά τα προβλήματα είναι πυκνά και περιέχουν πλήρεις πίνακες κόστους για την ανάθεση. Οι αλγόριθμοι εκτελέστηκαν απευθείας με δεδομένα σε πυκνή μορφή, εκτός από τον Lapmod, ο οποίος χρειάστηκε να εκτελεστεί μετατρέποντας τα πυκνά προβλήματα σε αραιά μορφή, καθώς έχει σχεδιαστεί για την επίλυση αραιών προβλημάτων.



Σχήμα 4.1: Χρόνοι Εκτέλεσης Πρώτου Συνόλου



Σχήμα 4.2: Χρήση Μνήμης Πρώτου Συνόλου

Στα αποτελέσματα βλέπουμε ότι όλοι οι αλγόριθμοι – MIP, CP-SAT, Ουγγρικός αλγόριθμος, Jonker-Volgenant και Lapmod – βρήκαν τα αναμενόμενα βέλτιστα κόστη με βάση τα δεδομένα από την OR-Library. Ωστόσο, οι κύριες διαφορές μεταξύ τους εντοπίζονται στον χρόνο εκτέλεσης και την κατανάλωση μνήμης.

- **Ουγγρικός Αλγόριθμος**: Εξαιρετικά ταχύς με χρόνους από 0,25 ms (assign100) έως 12,5 ms (assign800). Είναι ένας από τους πιο αποτελεσματικούς αλγό-

Πίνακας 4.1: Πρώτο Σύνολο

assign100			
Αλγόριθμοι	**Κόστος**	**Χρόνος Εκτέλεσης**	**Χρήση Μνήμης**
MIP Solver	305	330 ms	210 MB
CP-SAT Solver	305	1,5 sec	360 MB
Hungarian Method	305	0,25 ms	102 MB
Jonker-Volgenant	305	0,15 ms	123 MB
Lapmod	305	0,36 ms	123 MB
assign200			
MIP Solver	475	1,3 sec	420 MB
CP-SAT Solver	475	7,5 sec	1 GB
Hungarian Method	475	1 ms	126 MB
Jonker-Volgenant	475	0,4 ms	138 MB
Lapmod	475	1 ms	138 MB
assign300			
MIP Solver	626	3 sec	715 MB
CP-SAT Solver	626	14,2 sec	2,2 GB
Hungarian Method	626	2 ms	146 MB
Jonker-Volgenant	626	1 ms	334 MB
Lapmod	626	2 ms	334 MB
assign400			
MIP Solver	804	5,8 sec	1 GB
CP-SAT Solver	804	25 sec	3,8 GB
Hungarian Method	804	5,5 ms	340 MB
Jonker-Volgenant	804	1,5 ms	413 MB
Lapmod	804	3 ms	416 MB
assign500			
MIP Solver	991	10 sec	1,5 GB
CP-SAT Solver	991	35 sec	6,5 GB
Hungarian Method	991	7 ms	414 MB
Jonker-Volgenant	991	2,8 ms	541 MB
Lapmod	991	6 ms	544 MB
assign600			
MIP Solver	1176	15 sec	2 GB
CP-SAT Solver	1176	60 sec	9,4 GB
Hungarian Method	1176	12 ms	545 MB
Jonker-Volgenant	1176	3,5 ms	643 MB
Lapmod	1176	6 ms	648 MB
assign700			
MIP Solver	1362	21 sec	2,5 GB
CP-SAT Solver	1362	105 sec	11,5 GB
Hungarian Method	1362	13 ms	648 MB
Jonker-Volgenant	1362	4 ms	173 MB
Lapmod	1362	8,4 ms	182 MB
assign800			
MIP Solver	1552	28 sec	3,5 GB
CP-SAT Solver	1552	80 sec	11,5 GB
Hungarian Method	1552	12,5 ms	186 MB
Jonker-Volgenant	1552	5 ms	186 MB
Lapmod	1552	10,5 ms	200 MB

ριθμους για μικρού και μεσαίου μεγέθους προβλήματα. Στη χρήση μνήμης είναι επίσης πολύ αποδοτικός, χρησιμοποιώντας από 102 MB (assign100) έως 186 MB (assign800). Αυτό τον καθιστά ιδανικό για εφαρμογές που απαιτούν χαμηλή κατανάλωση πόρων.

- **Jonker-Volgenant**: Πολύ κοντά σε αποδόσεις με τον Ουγγρικό αλγόριθμο. Οι χρόνοι εκτέλεσης κυμαίνονται από 0,15 ms (assign100) έως 5 ms (assign800), γεγονός που τον καθιστά επίσης ιδανικό για ταχεία επίλυση. Κινείται σε παρόμοια επίπεδα με τον Ουγγρικό αλγόριθμο, χρησιμοποιώντας από 123 MB (assign100) έως 186 MB (assign800).

- **Lapmod**: Ελαφρώς πιο αργός από τον Jonker-Volgenant, με χρόνους από 0,36 ms (assign100) έως 10,5 ms (assign800). Παραμένει γρήγορος αλλά δείχνει να υστερεί ελαφρώς σε απόδοση. Παρουσιάζει ελαφρώς αυξημένη χρήση μνήμης, ξεκινώντας από 123 MB και φτάνοντας στα 200 MB για το assign800.

- **MIP Solver**: Χρησιμοποιεί περισσότερο χρόνο, από 330 ms (assign100) έως 28 sec (assign800). Παρόλο που είναι ακριβής, δεν είναι τόσο γρήγορος όσο οι άλλοι αλγόριθμοι, ιδιαίτερα σε μεγαλύτερα προβλήματα. Απαιτεί μεγαλύτερη μνήμη, από 210 MB (assign100) έως 3,5 GB (assign800), γεγονός που αυξάνει το κόστος σε πόρους όσο μεγαλώνει το πρόβλημα.

- **CP-SAT Solver**: Είναι ο πιο αργός αλγόριθμος, με χρόνους εκτέλεσης που αυξάνονται δραματικά, από 1,5 sec (assign100) έως 80 sec (assign800). Αυτό δείχνει ότι δεν είναι κατάλληλος για μεγάλα προβλήματα, όσον αφορά την ταχύτητα. Ο πιο απαιτητικός αλγόριθμος σε χρήση μνήμης, με καταναλώσεις από 360 MB (assign100) έως 11,5 GB (assign800), καθιστώντας τον λιγότερο αποδοτικό σε μεγάλα προβλήματα όσον αφορά την κατανάλωση πόρων.

4.2 Δεύτερο Σύνολο Προβλημάτων

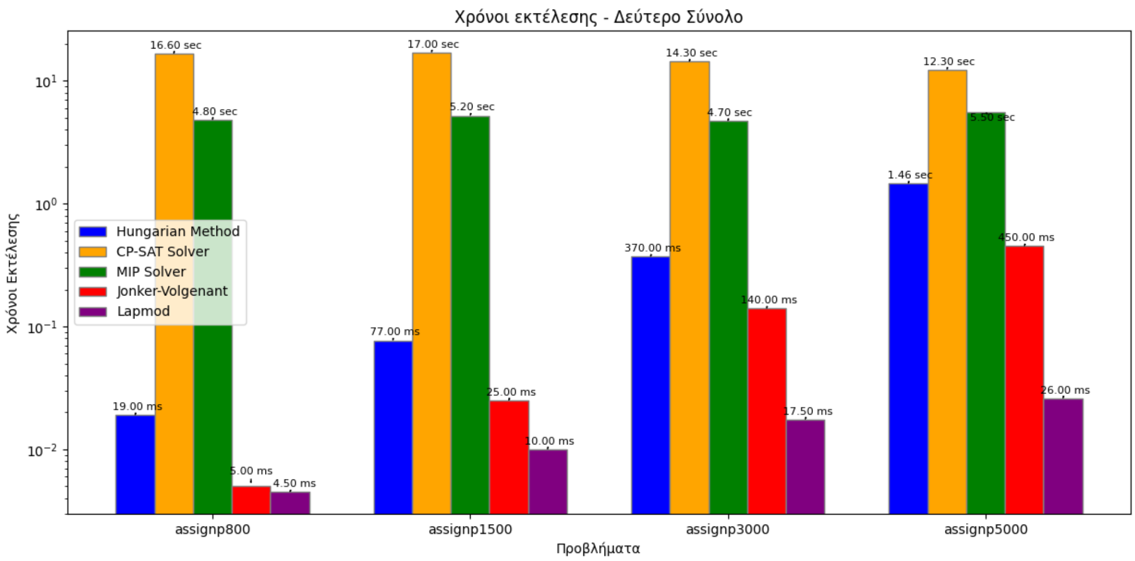
Το δεύτερο σετ προβλημάτων (assignp800 έως assignp5000) βασίζεται στον πίνακα 3 της ίδιας εργασίας και περιλαμβάνει αραιούς πίνακες κόστους. Εδώ, για να επιλυθούν τα προβλήματα με τους αλγόριθμους Jonker-Volgenant και Ουγγρικό, τα προβλήματα μετατράπηκαν από αραιά σε πυκνά, αφού αυτοί οι αλγόριθμοι δεν είναι σχεδιασμένοι για την άμεση επίλυση αραιών προβλημάτων.

Πίνακας 4.2: Δεύτερο Σύνολο

assignp800			
Αλγόριθμοι	**Κόστος**	**Χρόνος Εκτέλεσης**	**Χρήση Μνήμης**
MIP Solver	2239	4,8 sec	1 GB
CP-SAT Solver	2239	16,6 sec	2,5 GB
Hungarian Method	2239	19 ms	117 MB
Jonker-Volgenant	2239	5 ms	157 MB
Lapmod	2239	4,5 ms	155 MB
assignp1500			
MIP Solver	5839	5,2 sec	1 GB
CP-SAT Solver	5839	17 sec	2,5 GB
Hungarian Method	5839	77 ms	201 MB
Jonker-Volgenant	5839	25 ms	417 MB
Lapmod	5839	10 ms	178 MB
assignp3000			
MIP Solver	18696	4,7 sec	953 MB
CP-SAT Solver	18696	14,3 sec	2,5 GB
Hungarian Method	18696	370 ms	525 MB
Jonker-Volgenant	18696	140 ms	533 MB
Lapmod	18696	17,5 ms	240 MB
assignp5000			
MIP Solver	48533	5,5 sec	1 GB
CP-SAT Solver	48533	12,3 sec	2,7 GB
Hungarian Method	48533	1,46 sec	777 MB
Jonker-Volgenant	48533	450 ms	789 MB
Lapmod	48533	26 ms	607 MB

- Ο LAPMOD αποδεικνύεται ο πιο αποδοτικός αλγόριθμος στο δεύτερο σύνολο προβλημάτων, με εξαιρετικά χαμηλούς χρόνους εκτέλεσης που κυμαίνονται από 4,5 ms για το assignp800 έως 26 ms για το assignp5000. Η κατανάλωση μνήμης παραμένει χαμηλή, από 155 MB έως 607 MB, κάτι που επιβεβαιώνει την αποδοτικότητά του τόσο σε χρόνο όσο και σε πόρους.

- Ο Jonker-Volgenant παρουσιάζει επίσης υψηλές επιδόσεις, με χρόνους εκτέλεσης από 5 ms για το assignp800 έως 450 ms για το assignp5000. Ωστόσο, απαιτεί ελαφρώς περισσότερη μνήμη, από 157 MB έως 789 MB, κάτι που τον καθιστά λιγότερο αποδοτικό από τον LAPMOD, αλλά παραμένει πολύ αποτελεσματικός για μεγάλα προβλήματα.

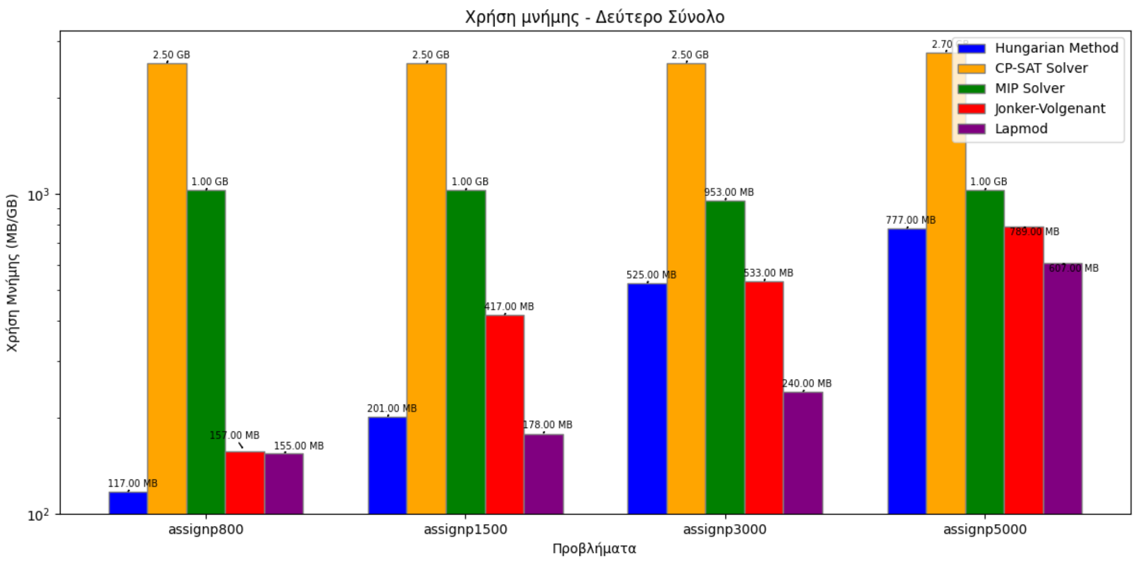


Σχήμα 4.3: Χρόνοι Εκτέλεσης Δεύτερου Συνόλου

- Ο Ουγγρικός αλγόριθμος διατηρεί καλές επιδόσεις, αν και οι χρόνοι εκτέλεσης είναι μεγαλύτεροι σε σχέση με τους Jonker-Volgenant και LAPMOD. Για το assignp800, ο χρόνος εκτέλεσης είναι 19 ms και για το assignp5000 είναι 1,46 sec. Η κατανάλωση μνήμης κυμαίνεται από 117 MB έως 777 MB, γεγονός που τον καθιστά καλή επιλογή, αν και λιγότερο αποδοτικό σε σύγκριση με τους άλλους δύο αλγόριθμους.

- Ο MIP Solver δίνει αξιόπιστες λύσεις, αλλά με σημαντικά μεγαλύτερους χρόνους εκτέλεσης, παραμένοντας σταθερά γύρω στα 5 sec για όλα τα προβλήματα. Η κατανάλωση μνήμης είναι σταθερή στα 1 GB, καθιστώντας τον πιο απαιτητικό σε πόρους και με χαμηλότερη απόδοση σε σύγκριση με τους άλλους αλγόριθμους.

- Ο CP-SAT Solver είναι ο πιο αργός από όλους, με χρόνους εκτέλεσης από 16,6 sec για το assignp800 έως 12,3 sec για το assignp5000. Η κατανάλωση μνήμης είναι υψηλή, σταθερά στα 2,5 GB για όλα τα προβλήματα. Παρά την ευελιξία του στη διαχείριση σύνθετων περιορισμών, η απόδοσή του σε χρόνους και πόρους τον κατατάσσει ως τον λιγότερο αποδοτικό αλγόριθμο στο σύνολο αυτό.



Σχήμα 4.4: Χρήση Μνήμης Δεύτερου Συνόλου

4.3 Συμπεράσματα Αποτελεσμάτων

Βάση των αποτελεσμάτων που προέκυψαν από τις δοκιμές των αλγορίθμων, τα συμπεράσματα παρουσιάζονται ως εξής:

Στο πρώτο σύνολο προβλημάτων, τα οποία είναι πυκνά, ο Ουγγρικός αλγόριθμος και ο αλγόριθμος Jonker-Volgenant απέδωσαν πιο αποτελεσματικά, όσον αφορά τον χρόνο και τη χρήση μνήμης. Με τον Ουγγρικός αλγόριθμος να διατηρεί σταθερά χαμηλή χρήση μνήμης και χρόνο εκτέλεσης, κάνοντάς τον ένα από τους πιο αποδοτικούς αλγόριθμους για πυκνά προβλήματα μικρού και μεσαίου μεγέθους. Με σχεδόν μηδαμινή διαφορά, ο Jonker-Volgenant παρουσίασε παρόμοια απόδοση με τον Ουγγρικό, και στις δύο μετρήσεις. Έπειτα, ο αλγόριθμος LAPMOD χρειάστηκε να μετατρέψει τα πυκνά προβλήματα σε αραιά για να τα επιλύσει, πράγμα που είχε μικρή επίδραση στη χρήση μνήμης σε σχέση με τους υπόλοιπους αλγορίθμους, αλλά εντούτοις ολοκλήρωσε τις λύσεις σε ικανοποιητικό χρόνο. Τέλος, οι MIP Solver και CP-SAT Solver, ενώ κατάφεραν να βρουν τις βέλτιστες λύσεις, παρουσίασαν υψηλότερους χρόνους εκτέλεσης και μεγαλύτερη κατανάλωση μνήμης σε σύγκριση με τους υπόλοιπους αλγόριθμους. Ο CP-SAT ήταν ο πιο αργός και πιο απαιτητικός σε πόρους από όλους τους αλγόριθμους, ενώ ο MIP ήταν αποδοτικότερος από τον CP-SAT, αλλά όχι τόσο όσο οι Ουγγρικός και Jonker-Volgenant.

Στο δεύτερο σύνολο προβλημάτων, τα οποία είναι αραιά, ο LAPMOD αναδείχθηκε ως ο πιο αποδοτικός αλγόριθμος, παρέχοντας εξαιρετικά χαμηλούς χρόνους εκτέλεσης και κατανάλωση μνήμης. Έπειτα ακολούθησε ο Jonker-Volgenant με υψηλή αποδοτικότητα. Ο Ουγγρικός αλγόριθμος συνέχισε να προσφέρει ικανοποιητική απόδοση, ωστόσο υστερούσε σε χρόνο σε σύγκριση με τους LAPMOD και Jonker-Volgenant. Ο MIP Solver παρείχε αξιόπιστες λύσεις, αλλά οι χρόνοι εκτέλεσης και η κατανάλωση μνήμης ήταν σημαντικά υψηλότεροι, καθιστώντας τον λιγότερο αποδοτικό. Τέλος, ο CP-SAT Solver ήταν ο πιο αργός αλγόριθμος, με υψηλές απαιτήσεις σε μνήμη, τοποθετώντας τον ως τη λιγότερο αποδοτική επιλογή.

Συνολικά, τα αποτελέσματα δείχνουν ότι οι αλγόριθμοι Ουγγρικός, Jonker-Volgenant και LAPMOD προσφέρουν την καλύτερη αποδοτικότητα, με χαμηλή χρήση μνήμης και γρήγορους χρόνους εκτέλεσης, ειδικά σε προβλήματα μικρού και μεσαίου μεγέθους. Ο LAPMOD ξεχωρίζει για την απόδοσή του στα αραιά προβλήματα, καθώς είναι σχεδιασμένος αποκλειστικά για αυτά, ενώ στο πρώτο σύνολο προβλημάτων χρειάστηκε μετατροπή των πυκνών προβλημάτων σε αραιά για να τα επιλύσει. Ο Jonker-Volgenant και ο Ουγγρικός αλγόριθμος είναι σχεδιασμένοι για πυκνά προβλήματα, και στο δεύτερο σύνολο έγινε μετατροπή των αραιών προβλημάτων σε πυκνά για να μπορέσουν να τα λύσουν. Από την άλλη, οι MIP και CP-SAT Solvers ξεχωρίζουν για την ευελιξία τους, καθώς είναι σχεδιασμένοι να επιλύουν τόσο πυκνά όσο και αραιά προβλήματα χωρίς να χρειαστούν μετατροπή, αν και έχουν υψηλότερες απαιτήσεις σε χρόνο και μνήμη.

Κεφάλαιο 5

Συμπεράσματα

Σε αυτή τη διπλωματική εργασία, ασχοληθήκαμε με το κλασικό πρόβλημα της ανάθεσης, το οποίο γενικότερα εφαρμόζεται σε διάφορους τομείς όπως, η διαχείριση έργων, η κατανομή πόρων και η παραγωγή. Στην ουσία, το πρόβλημα ανάθεσης αντιστοιχεί ένα σύνολο πόρων σε ένα σύνολο εργασιών, με τέτοιο τρόπο ώστε να ελαχιστοποιείται το συνολικό κόστος ή να μεγιστοποιείται το κέρδος. Κάποιες από τις πιο γνωστές μαθηματικές μεθόδους που χρησιμοποιήθηκαν για να λύσουν το συγκεκριμένο πρόβλημα, βασίζονται κυρίως σε αλγορίθμους βελτιστοποιήσεις, όπως ο Ουγγρικός αλγόριθμος, οι αλγόριθμοι Branch and Bound, οι τεχνικές χαλάρωσης και άλλοι. Στη βιβλιογραφία, έχουν αναπτυχθεί αρκετές μέθοδοι για την επίλυση του προβλήματος ανάθεσης, καθεμία με τα δικά της πλεονεκτήματα και μειονεκτήματα. Ειδικότερα, οι αλγόριθμοι που εξετάστηκαν στην εργασία αυτή περιλαμβάνουν τους κλασικούς αλγόριθμους όπως ο Ουγγρικός και κάποιους άλλους σύγχρονους αλγορίθμους, όπως οι Jonker-Volgenant και LAPMOD, καθώς και οι λύσεις με χρήση MIP και CP-SAT solvers. Αυτές οι μέθοδοι συγκρίθηκαν με βάση την ταχύτητα και την κατανάλωση μνήμης, ειδικά σε πυκνά και αραιά σύνολα προβλημάτων, με στόχο να προσδιοριστεί η καταλληλόλητα κάθε προσέγγισης. Η συνεισφορά σε αυτή την εργασία ήταν η σύγκριση αυτών των αλγορίθμων σε πραγματικά δεδομένα, τόσο για πυκνά όσο και για αραιά σύνολα προβλημάτων ανάθεσης. Επιπλέον, πραγματοποιήθηκαν υπολογιστικές μελέτες, με στόχο την αξιολόγηση της αποδοτικότητάς τους σε σχέση με τον χρόνο εκτέλεσης και τη χρήση μνήμης. Οι πειραματικές δοκιμές έδειξαν ότι οι πιο σύγχρονοι αλγόριθμοι, όπως ο LAPMOD και ο Jonker-Volgenant, προσφέρουν καλύτερες επιδόσεις σε σχέση με τις παραδοσιακές μεθόδους σε μεγάλα προβλήματα ανάθεσης, ενώ οι MIP και CP-SAT solvers είναι κατάλληλοι για συγκεκριμένες κατηγορίες προβλημάτων, όπως προβλήματα με πρόσθετους περιορισμούς. Τα αποτελέσματα έδειξαν ότι η κατάλληλη επιλογή για τον κάθε αλγορίθμου εξαρτάται από τη φύση των δεδομένων. Για παράδειγμα, σε πυκνά προβλήματα, ο Ουγγρικός αλγόριθμος και ο Jonker-Volgenant(LAPJV) ήταν πιο αποτελεσματικοί, ενώ για αραιά προβλήματα, ο LAPMOD παρουσίασε σαφείς βελτιώσεις, καθώς είναι σχεδιασμένος για αυτά. Μελλοντικές επεκτάσεις αυτής της εργασίας θα μπορούσαν να περιλαμβάνουν την εφαρμογή αυτών των αλγορίθμων σε ακόμα μεγαλύτερα σύνολα δεδομένων, την ανάπτυξη υβριδικών μεθόδων που συνδυάζουν διαφορετικές προσεγγίσεις, ή την ενσωμάτωση νέων τεχνικών βελτιστοποίησης που αξιοποιούν πιο σύγχρονες εξελίξεις στους τομείς της τεχνητής νοημοσύνης και των παράλληλων υπολογιστικών συστημάτων.

Βιβλιογραφία

[1] M. Akgül and O. Ekin. A dual feasible forest algorithm for the linear assignment problem. *RAIRO - Operations Research*, 25(4):403–411, 1991. Number: 4 Publisher: EDP Sciences.

[2] Mustafa Akgül. A sequential dual simplex algorithm for the linear assignment problem. *Operations Research Letters*, 7(3):155–158, June 1988.

[3] M. L. Balinski. Signature Methods for the Assignment Problem. *Operations Research*, 33(3):527–536, June 1985. Publisher: INFORMS.

[4] M. L. Balinski. A competitive (dual) simplex method for the assignment problem. *Mathematical Programming*, 34(2):125–141, March 1986.

[5] M. L. Balinski and R. E. Gomory. A Primal Method for the Assignment and Transportation Problems. *Management Science*, 10(3):578–593, April 1964. Publisher: INFORMS.

[6] R. S. Barr, F. Glover, and D. Klingman. The alternating basis algorithm for assignment problems. *Mathematical Programming*, 13(1):1–13, December 1977.

[7] Hadi Basirzadeh. Ones assignment method for solving assignment problems. *Applied Mathematical Sciences*, 6(47):2345–2355, 2012.

[8] J. E. Beasley. Linear Programming on Cray Supercomputers. *The Journal of the Operational Research Society*, 41(2):133–139, 1990. Publisher: Palgrave Macmillan Journals.

[9] D. P. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14(1):105–123, December 1988.

[10] Dimitri P. Bertsekas. A distributed algorithm for the assignment problem. *Lab. for Information and Decision Systems Working Paper, MIT*, 1979.

[11] Dimitri P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, December 1981.

[12] Dimitri P. Bertsekas. A distributed asynchronous relaxation algorithm for the assignment problem. In *1985 24th IEEE Conference on Decision and Control*, pages 1703–1704, December 1985.

[13] Dimitri P. Bertsekas and David A. Castanon. The auction algorithm for the transportation problem. *Annals of Operations Research*, 20(1):67–96, December 1989.

[14] Dimitri P. Bertsekas and David A. Castañon. Parallel synchronous and asynchronous implementations of the auction algorithm. *Parallel Computing*, 17(6):707–732, September 1991.

[15] Dimitri P. Bertsekas and David A. Castañon. Parallel Asynchronous Hungarian Methods for the Assignment Problem. *ORSA Journal on Computing*, 5(3):261–274, August 1993. Publisher: ORSA.

[16] Robert Bixby and Edward Rothberg. Progress in computational mixed integer programming—A look back from the other side of the tipping point. *Annals of Operations Research*, 149(1):37–41, February 2007.

[17] Rainer E. Burkard, Willi Hahn, and Uwe Zimmermann. An algebraic approach to assignment problems. *Mathematical Programming*, 12(1):318–327, December 1977.

[18] Giorgio Carpaneto and Paolo Toth. Algorithm 548: Solution of the Assignment Problem [H]. *ACM Transactions on Mathematical Software*, 6(1):104–111, March 1980.

[19] Giorgio Carpaneto and Paolo Toth. Primal-dual algrorithms for the assignment problem. *Discrete Applied Mathematics*, 18(2):137–153, November 1987.

[20] W. H. Cunningham. A network simplex method. *Mathematical Programming*, 11(1):105–116, December 1976.

[21] U. Derigs. The shortest augmenting path method for solving assignment problems — Motivation and computational experience. *Annals of Operations Research*, 4(1):57–102, December 1985.

[22] Michael Engquist. A Successive Shortest Path Algorithm for The Assignment Problem. *INFOR: Information Systems and Operational Research*, 20(4):370–384, November 1982. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/03155986.1982.11731874.

[23] Gerald L. Thompson and Gerald L. Thompson. A Recursive Method for Solving Assignment Problems. *North-holland Mathematics Studies*, 59:319–343, January 1981. MAG ID: 1828206908 S2ID: 2b19185fcfa6fe4db658477d14cfab9e99398368.

[24] Fred Glover, Randy Glover, and Darwin Klingman. Threshold assignment algorithm. In Giorgio Gallo and Claudio Sandi, editors, *Netflow at Pisa*, Mathematical Programming Studies, pages 12–37. Springer, Berlin, Heidelberg, 1986.

[25] D. Goldfarb. Efficient dual simplex algorithms for the assignment problem. *Mathematical Programming*, 33(2):187–203, November 1985.

[26] Ming S. Hung. Technical Note—A Polynomial Simplex Method for the Assignment Problem. *Operations Research*, 31(3):595–600, June 1983. Publisher: INFORMS.

[27] Ming S. Hung and Walter O. Rom. Solving the Assignment Problem by Relaxation. *Operations Research*, 28(4):969–982, August 1980. Publisher: INFORMS.

[28] H. A. Hussein and M. a. K. Shiker. Two New Effective Methods to Find the Optimal Solution for the Assignment Problems. *Journal of Advanced Research in Dynamic and Control Systems*, Volume 12(Issue 7):49–54, 2020.

[29] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, December 1987.

[30] Roy Jonker and Ton Volgenant. Improving the Hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175, October 1986.

[31] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109.

[32] Leon F. McGinnis. Implementation and Testing of a Primal-Dual Algorithm for the Assignment Problem. *Operations Research*, 31(2):277–291, April 1983. Publisher: INFORMS.

[33] James Munkres. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, March 1957. Publisher: Society for Industrial and Applied Mathematics.

[34] Edited F Rossi, P van Beek, and T Walsh. Handbook of Constraint Programming.

[35] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., USA, May 1986.

[36] Shweta Singh, G. C. Dubey, and Rajesh Shrivastava. A comparative analysis of assignment problem. *IOSR Journal of Engineering*, 2(8):01–15, 2012.

[37] Edward Tsang. *Foundations of constraint satisfaction: the classic text*. BoD–Books on Demand, 2014.

[38] Sriyani Violina and Et Al. Branch And Bound Algorithm Analysis For Solving Job Assignment Problems. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(11):1272–1276, May 2021. Number: 11.

[39] A. Volgenant. Linear and semi-assignment problems: A core oriented approach. *Computers & Operations Research*, 23(10):917–932, October 1996.

[40] Joel M. Wein and Stavros A. Zenios. On the massively parallel solution of the assignment problem. *Journal of Parallel and Distributed Computing*, 13(2):228–236, October 1991.

[41] A. Weintraub and F. Barahona. A dual algorithm for the assignment problem. *Departamento de Industrias, Universidad de Chile-Sede Occidente (April)*, 1979.

[42] Laurence A. Wolsey. *Integer programming / Laurence A. Wolsey.* Wiley-Interscience series in discrete mathematics and optimization. J. Wiley, New York, 1998. Publication Title: Integer programming.