

Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών
Υπολογιστών

Υλοποίηση και υπολογιστική σύγκριση
αλγορίθμων για προβλήματα ανάθεσης
πόρων

Αικατερίνη Ειρήνη Βαφειάδου
Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

10 Οκτωβρίου 2021

*Πάντα στον νου σου να 'χεις την Ιθάκη.
Το φθάσιμον εκεί είν' ο προορισμός σου.
Αλλά μη βιάζεις το ταξίδι διόλου.
Καλύτερα χρόνια πολλά να διαρκέσει
Και γέρος πια ν' αράξεις στο νησί,
Πλούσιος με όσα κέρδισες στον δρόμο,
Μη προσδοκώντας πλούτη να σε δώσει η Ιθάκη.
24-30, Καβάφης, Ιθάκη*

Περίληψη

Το Γενικευμένο Πρόβλημα Ανάθεσης Πόρων είναι ένα σημαντικό πρόβλημα της Επιχειρησιακής Έρευνας. Είναι ένα πρόβλημα που περιγράφει την κατανομή πόρων σε διεργασίες, με διάφορους περιορισμούς συστήματος, και σκοπό έχει την ελαχιστοποίηση του γενικού κόστους ή αλλιώς τη μεγιστοποίηση του οφέλους. Πιο συγκεκριμένα, το Γενικευμένο Πρόβλημα Ανάθεσης Πόρων μπορεί να περιγράψει παραδείγματα και να χρησιμοποιηθεί σε διάφορους τομείς της σύγχρονης ζωής, όπως είναι η επιστήμη των υπολογιστικών συστημάτων, οι τηλεπικοινωνίες, ο χρονοπρογραμματισμός επιχειρήσεων και η βέλτιστη απόδοση βιομηχανικών μηχανημάτων.

Στην παρούσα εργασία μελετάται το πρόβλημα αυτό, αναπτύσσοντας σε προγραμματιστικό περιβάλλον ένα πρόγραμμα επίλυσης που αποτελείται από γνωστούς μεθευρετικούς αλγορίθμους. Οι αλγόριθμοι αυτοί θεωρούνται βασικοί αλγόριθμοι επίλυσης του γενικευμένου προβλήματος ανάθεσης πόρων. Με την ανάπτυξη του προγράμματος αυτού, σκοπός της εργασίας είναι η εξαγωγή αποδεκτών λύσεων βάση των προβλημάτων που επιλύθηκαν, αλλά και η ανάλυση και σύγκριση των μεθόδων βάση των αποτελεσμάτων τους.

Λέξεις κλειδιά: Γενικευμένο Πρόβλημα Ανάθεσης Πόρων, Μεθευρετικοί Αλγόριθμοι, Επιχειρησιακή Έρευνα, Βελτιστοποίηση, Ελάχιστο Κόστος

Abstract

The Generalized Problem of Resource Assignment is a major problem of Operational Research. It is a problem that describes the allocation of resources in processes, with various system constraints, and aims to minimize the overall cost or otherwise maximize the benefit. More specifically, the Generalized Resource Assignment Problem can be described as examples and used in various areas of modern life, such as computer science, telecommunications, business scheduling, and the optimal performance of industrial machinery.

In the present study, this problem is studied, developing in a programming environment a solution computer program consisting of known metaheuristic algorithms. These algorithms are considered as basic algorithms for solving the generalized assignment problem. With the development of this solver, the purpose of the study is to extract acceptable solutions based on the problems solved, but also to analyze and compare the methods based on their results.

Keywords: Generalized Assignment Problem, Metaheuristic Algorithms, Operational Research, Optimization, Minimum Cost

Δήλωση Πνευματικών Δικαιωμάτων

Δήλωση Πνευματικών Δικαιωμάτων Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο "Υλοποίηση και υπολογιστική σύγκριση αλγορίθμων για προβλήματα ανάθεσης πόρων" καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Νικόλαου Πλόσκα αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

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

Copyright (C) Αικατερίνη-Ειρήνη Βαφειάδου & Νικόλαος Πλόσκας, 2021, Κοζάνη

Υπογραφή Φοιτητή

Περιεχόμενα

1	Εισαγωγή	11
1.1	Ορισμός του προβλήματος	11
1.2	Κίνητρα και στόχοι υλοποίησης	12
1.3	Διάρθρωση κειμένου	13
2	Θεωρητικό υπόβαθρο	14
2.1	Επιχειρησιακή έρευνα	14
2.2	Πρόβλημα ανάθεσης πόρων	15
2.3	Η Ουγγρική μέθοδος	18
2.4	Λοιποί μέθοδοι επίλυσης	21
3	Αλγόριθμοι επίλυσης γενικευμένου προβλήματος ανάθεσης πόρων	23
3.1	Μέθοδος προσομοιωμένης ανόπτωσης	23
3.2	Μέθοδος επαναληπτικής τοπικής αναζήτησης	26
3.3	Μέθοδος μεταβλητής γειτονιάς αναζήτησης	28
3.4	Μέθοδος άπληστης τυχαιοποιημένης τοπικής αναζήτησης	30
3.5	Μέθοδος περιορισμένης αναζήτησης	32
3.6	Μέθοδος αποδοχής κατωφλιού	34
4	Υπολογιστική μελέτη	37
4.1	Δεδομένα προβλημάτων	37
4.2	Υλοποίηση	40
4.3	Αποτελέσματα και σύγκριση μεθόδων	42
5	Συμπεράσματα και μελλοντικές επεκτάσεις	53
5.1	Συμπεράσματα	53
5.2	Μελλοντικές επεκτάσεις	55

Κατάλογος σχημάτων

2.1	Παράδειγμα αριθμητικού παραδείγματος για το πρόβλημα ανάθεσης πόρων	17
2.2	Παράδειγμα επίλυσης Ουγγρικής μεθόδου	18
2.3	Επιλογή μέγιστων κερδών	18
2.4	Υπολογισμός πίνακα	19
2.5	Επιλογή ελάχιστων κερδών	19
2.6	Υπολογισμός πίνακα	19
2.7	Κάλυψη μηδενικών με γραμμές	20
2.8	Εύρεση ελάχιστης τιμής	20
2.9	Υπολογισμός πίνακα	20
2.10	Τελική ανάθεση διεργασιών	21
3.1	Παράδειγμα προβλήματος	25
3.2	Λύση του προβλήματος με τη μέθοδο προσομοιωμένης ανόπτωσης . . .	25
3.3	Λύση του προβλήματος με τη μέθοδο επαναληπτικής τοπικής αναζήτησης	27
3.4	Λύση του προβλήματος με τη μέθοδο αναζήτησης μεταβλητής γειτονιάς	29
3.5	Λύση του προβλήματος με τη μέθοδο άπληστης τυχαιοποιημένης τοπικής αναζήτησης	31
3.6	Λύση του προβλήματος με τη μέθοδο περιορισμένης αναζήτησης . . .	33
3.7	Λύση του προβλήματος με τη μέθοδο αποδοχής κατωφλιού	36
4.1	Δήλωση δεδομένων προβλήματος - μέρος αρχείου gap1.txt	38
4.2	Δήλωση αποτελεσμάτων ανά μέθοδο για τα προβλήματα του gap1.txt	41
4.3	Διάγραμμα ελαχίστου κόστους - gap1.txt	42
4.4	Διάγραμμα χρόνων εκτέλεσης - gap1.txt	43
4.5	Διάγραμμα επιδόσεων αλγορίθμων - gap1.txt	44

4.6	Διάγραμμα ελαχίστου λόστους - gar6.txt	45
4.7	Διάγραμμα χρόνων εκτέλεσης - gar6.txt	46
4.8	Διάγραμμα επιδόσεων αλγορίθμων - gar6.txt	47
4.9	Διάγραμμα ελαχίστου κόστους - gar12.txt	48
4.10	Διάγραμμα χρόνων εκτέλεσης - gar12.txt	49
4.11	Διάγραμμα επιδόσεων αλγορίθμων - gar12.txt	50

Κατάλογος αλγορίθμων

1	Ψευδοκώδικας μεθόδου προσομοιωμένης ανόπτωσης	24
2	Ψευδοκώδικας μεθόδου επαναληπτικής τοπικής αναζήτησης	27
3	Ψευδοκώδικας μεθόδου αναζήτησης μεταβλητής γειτονιάς	29
4	Ψευδοκώδικας μεθόδου άπληστης τυχαιοποιημένης τοπικής αναζήτησης	31
5	Ψευδοκώδικας μεθόδου περιορισμένης αναζήτησης	33
6	Ψευδοκώδικας μεθόδου αποδοχής κατωφλιού	35

Κατάλογος πινάκων

4.1	Δεδομένα και λύσεις των προβλημάτων	39
4.2	Αποτελέσματα μεθόδων	51
4.3	Αποτελέσματα μεθόδων	51
1	Αποτελέσματα προβλημάτων αρχείου gap1.txt	57
2	Αποτελέσματα προβλημάτων αρχείου gap2.txt	58
3	Αποτελέσματα προβλημάτων αρχείου gap3.txt	60
4	Αποτελέσματα προβλημάτων αρχείου gap4.txt	62
5	Αποτελέσματα προβλημάτων αρχείου gap5.txt	64
6	Αποτελέσματα προβλημάτων αρχείου gap6.txt	66
7	Αποτελέσματα προβλημάτων αρχείου gap7.txt	68
8	Αποτελέσματα προβλημάτων αρχείου gap8.txt	70
9	Αποτελέσματα προβλημάτων αρχείου gap9.txt	72
10	Αποτελέσματα προβλημάτων αρχείου gap10.txt	74
11	Αποτελέσματα προβλημάτων αρχείου gap11.txt	76
12	Αποτελέσματα προβλημάτων αρχείου gap12.txt	79

Κεφάλαιο 1

Εισαγωγή

Στην παρούσα διπλωματική εργασία παρουσιάζεται και επιλύεται το γενικευμένο πρόβλημα ανάθεσης πόρων (Generalized Assignment Problem). Πρόκειται για ένα πρόβλημα συνδυαστικής βελτιστοποίησης που περιγράφει την κατανομή πόρων ενός συστήματος σε διεργασίες, χρησιμοποιώντας διάφορους περιορισμούς στο σύστημα και έχοντας σκοπό την ελαχιστοποίηση του κόστους ή τη μεγιστοποίηση του οφέλους. Επίσης, ανήκει στην επιστήμη της Επιχειρησιακής Έρευνας.

Γενικά, το πρόβλημα αυτό θεωρείται ένα πολύ σημαντικό πρόβλημα σε διάφορους επιστημονικούς τομείς, καθώς έχει διάφορες εφαρμογές στον τομέα των υπολογιστών, τις τηλεπικοινωνίες, την οργάνωση και διοίκηση επιχειρήσεων και τη λειτουργία μηχανημάτων στη βιομηχανία.

Μέσω της μελέτης πλήθους μεθυστικών μεθόδων επίλυσης προβλημάτων ανάθεσης πόρων γίνεται μια κατηγοριοποίηση και αξιολόγηση των αλγορίθμων αυτών. Η μελέτη των αλγορίθμων αυτών αρχικά απαιτεί την υλοποίησή τους, ενώ στη συνέχεια οι αλγόριθμοι αυτοί συγκρίνονται σύμφωνα με τα ιδιαίτερα χαρακτηριστικά διεξαγωγής του εκάστοτε αλγορίθμου και τέλος, αναλύονται και αξιολογούνται τα αποτελέσματά τους σύμφωνα με προκαθορισμένα κριτήρια.

1.1 Ορισμός του προβλήματος

Το γενικευμένο πρόβλημα ανάθεσης πόρων είναι ένα πρόβλημα στον χώρο της επιχειρησιακής έρευνας και των εφαρμοσμένων μαθηματικών και είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης. Η συνδυαστική βελτιστοποίηση είναι ένας σημαντικός υπό-κλάδος της μαθηματικής βελτιστοποίησης και έχει ως στόχο της εύρεση ενός βέλτιστου αντικειμένου από ένα πεπερασμένο σύνολο στοιχείων.

Σε μία γενική μορφή περιγραφής του συγκεκριμένου προβλήματος, έστω ότι υπάρχουν συγκεκριμένου αριθμού εργασίες προς διεκπεραίωση, καθώς και πεπερασμένου αριθμού πράκτορες, οι οποίοι αναλαμβάνουν την ολοκλήρωση των εργασιών. Ειδικότερα, για την υλοποίηση της κάθε εργασίας, πρέπει να ανατεθεί σε οποιονδήποτε πράκτορα, έχοντας συγκεκριμένο κόστος πόρων και κέρδος, σύμφωνα με τον εκάστοτε πράκτορα. Στόχος είναι η βέλτιστη επιλογή σύμφωνα με δύο κριτήρια κατανομής. Το πρώτο αφορά τη βέλτιστη κατανάλωση των πόρων του εκάστοτε πράκτορα, χωρίς αυτή να υπερβεί το μέγιστο όριο. Το δεύτερο κριτήριο αφορά το μέγιστο δυνατό κέρδος του συστήματος.

Σε μία απλοποίηση του προβλήματος ανάθεσης πόρων, μπορούμε να περιγράψουμε το πρόβλημα πολλαπλού σακιδίου, όπου όλοι οι πράκτορες έχουν ίσα κέρδη και κόστη. Εάν στο πρόβλημα υπάρχει μόνο ένας πράκτορας, το πρόβλημα απλοποιείται στο πρόβλημα σακιδίου [1].

Το γενικευμένο πρόβλημα ανάθεσης πόρων εφαρμόζεται σε πολλούς τομείς, όπως τα δίκτυα τηλεπικοινωνιών, τους πόρους ηλεκτρονικών υπολογιστών, την πλοήγηση οχημάτων, αλλά και τον συντονισμό και σχεδιασμό ομάδων παραγωγής κάποιας εργασίας. Για αυτόν τον λόγο καθίσταται ένα πρόβλημα μείζονος ερευνητικού ενδιαφέροντος. Χάρη στην επίλυση και πρακτική εφαρμογή του προβλήματος αυτού στους διάφορους τομείς που προαναφέρθηκαν, υπάρχουν χρήσιμες και καλά σχεδιασμένες εφαρμογές.

1.2 Κίνητρα και στόχοι υλοποίησης

Στο πλαίσιο ανάπτυξης της τεχνολογίας και της επιστήμης της επιχειρησιακής έρευνας, το γενικευμένο πρόβλημα ανάθεσης πόρων γίνεται αντικείμενο μελέτης περίπου από το 1975 [2]. Όπως προαναφέρθηκε, το συγκεκριμένο πρόβλημα έχει εφαρμογές σε διάφορους τομείς και επιστημονικά πεδία, οπότε βασικό κίνητρο μελέτης του είναι η βελτιστοποίηση και η ανάλυση των μεθόδων επίλυσής του.

Κίνητρο της παρούσας εργασίας είναι να μελετηθεί και να δημιουργηθεί ένας υπολογιστικός επιλυτής προβλημάτων που βασίζεται στο πρόβλημα ανάθεσης πόρων με διάφορες μεθόδους αναζήτησης των λύσεων τους. Τα προβλήματα αυτά είναι εκφρασμένα με μαθηματικές εξισώσεις, αλλά μπορούν να συμβολίζουν τις μεταβλητές οποιουδήποτε προβλήματος μπορεί να περιγραφεί ως πρόβλημα ανάθεσης

πόρων.

Με αυτόν τον τρόπο θα επιτευχθούν και οι στόχοι της παρούσας εργασίας που είναι η υλοποίηση βασικών μεθόδων επίλυσης του προβλήματος σε προγραμματιστικό περιβάλλον, αλλά και η μελέτη και σύγκριση των αποτελεσμάτων αυτών. Η αξιολόγηση αυτή βασίζεται σε δύο βασικά επίπεδα, το πρώτο είναι η αξιολόγηση και ανάλυση της εξαγόμενης λύσης, και το δεύτερο σημείο αξιολόγησης είναι σε σχέση με τον τελικό χρόνο εκτέλεσης.

1.3 Διάρθρωση κειμένου

Στο Κεφάλαιο 2 δίνεται το θεωρητικό υπόβαθρο που μελετήθηκε για την υλοποίηση της εργασίας. Βασικές έννοιες όπως ο επιστημονικός κλάδος της επιχειρησιακής έρευνας, αλλά και η περαιτέρω ανάλυση του γενικευμένου προβλήματος ανάθεσης πόρων αναλύονται ως θεωρητικό υπόβαθρο της εργασίας.

Οι μέθοδοι επίλυσης που υλοποιήθηκαν και συγκρίθηκαν, παρουσιάζονται στο Κεφάλαιο 3. Οι αλγόριθμοι των μεθόδων αυτών είναι μεθευρετικοί, και δίνεται για τον καθένα η περιγραφή του καθώς και βασικές ιδιότητες. Στη συνέχεια, παραθέτοντας τους ψευδοκώδικες που χρησιμοποιήθηκαν για την υλοποίηση του εκάστοτε προγράμματος για τον κάθε αλγόριθμο, παρουσιάζεται ο βασικός σχεδιασμός της επίλυσης της κάθε μεθόδου.

Στο Κεφάλαιο 4 δίνεται η περιγραφή της υλοποίησης της υπολογιστικής μελέτης της εργασίας καθώς και η περιγραφή των δεδομένων των προβλημάτων που χρησιμοποιήθηκαν προς επίλυση. Επίσης, συγκρίνονται τα αποτελέσματα που προκύπτουν από τις μεθόδους που περιγράφονται και δίνονται συγκριτικά διαγράμματα βάση κάποιων προβλημάτων που επιλύθηκαν.

Τα τελικά συμπεράσματα και μελλοντικές επεκτάσεις που θα μπορούν να διεκπεραιωθούν βάση αυτής της εργασίας αναλύονται στο Κεφάλαιο 5. Συγκεκριμένα, βάση των αποτελεσμάτων που προέκυψαν, δίνονται συμπεράσματα σύγκρισης των μεθόδων που υλοποιήθηκαν, και αναφέρονται οι τομείς που μπορούν να επεκταθούν πάνω στο γενικευμένο πρόβλημα ανάθεσης πόρων.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό θα αναφερθούν βασικές έννοιες που χρησιμοποιήθηκαν ως θεωρητικό υπόβαθρο της ανάλυσης και επίλυσης της παρούσας εργασίας. Αρχικά περιγράφεται γενικά η επιστήμη της επιχειρησιακής έρευνας και οι τομείς της. Στη συνέχεια αναλύονται το πρόβλημα ανάθεσης πόρων και οι αλγόριθμοι επίλυσής του. Για την υλοποίηση της εργασίας αυτής, είναι σημαντικό να κατανοηθούν οι βασικές έννοιες του τομέα αυτού, σε θεωρητικό και τεχνικό επίπεδο. Για τον λόγο αυτό, παρακάτω δίνονται ορισμοί βασικών εννοιών όπως είναι η επιχειρησιακή έρευνα.

2.1 Επιχειρησιακή έρευνα

Επιχειρησιακή έρευνα ονομάζεται η επιστήμη που σε συνδυασμό με άλλες επιστήμες, όπως τα μαθηματικά, επιλύει προβλήματα για τη διεύθυνση και διοίκηση μεγάλων συστημάτων ενός οργανισμού, με βέλτιστους τρόπους. Τέτοια συστήματα μπορούν να αποτελούνται από βιομηχανικά υλικά, μηχανήματα, υπηρεσίες ή και ανθρώπινο δυναμικό [3].

Η επιστήμη αυτή ξεκίνησε στον Β' Παγκόσμιο Πόλεμο, όπου οι επιστήμονες της εποχής έπρεπε να λύσουν προβλήματα σε σχέση με στρατιωτικές επιχειρήσεις, όπως επιθέσεις και διαχείριση υποβρυχίων. Μετά τον πόλεμο, αποτέλεσε νέο επιστημονικό πεδίο, όπου τις δεκαετίες του 1950 και 1960 αναπτύχθηκαν οι θεμελιώδεις μέθοδοι που χρησιμοποιούμε για επίλυση μέχρι και σήμερα. Η επιχειρησιακή έρευνα μπορεί να αναφερθεί επίσης και ως Διοικητική Επιστήμη (Management Science) και αποτελεί την επιστημονική ανάλυση και σχεδιασμό αποφάσεων για τον καλύτερο συντονισμό συστημάτων που χρειάζονται κατανομή κάποιων παραγωγικών πόρων.

2.2 Πρόβλημα ανάθεσης πόρων

Στην πιο γενική του μορφή, το πρόβλημα ανάθεσης πόρων μπορεί να θεωρηθεί ως ένα πρόβλημα συνδυαστικής βελτιστοποίησης. Το πλήθος των πρακτόρων και των διεργασιών έχουν ένα μέγεθος, και μπορεί να είναι διαφορετικό μεταξύ τους. Σε οποιονδήποτε πράκτορα, μπορεί να ανατεθεί οποιαδήποτε διεργασία, με διαφορετικό κόστος υλοποίησής της. Στόχος της επίλυσης του προβλήματος είναι η ανάθεση και υλοποίηση όλων των διεργασιών από τους πράκτορες με το ελάχιστο δυνατό κόστος και μέγιστο δυνατό κέρδος.

Οι πιο γνωστές παραλλαγές του προβλήματος ανάθεσης πόρων αφορούν τις ιδιότητες των πρακτόρων. Εάν οι πράκτορες έχουν ίσα κέρδη και κόστη στις αναθέσεις της εκάστοτε διεργασίας το πρόβλημα μετατρέπεται στο πρόβλημα πολλαπλού σακιδίου. Επιπρόσθετα, εάν το πρόβλημα ορίζεται με μόνο έναν πράκτορα, το πρόβλημα θεωρείται το γενικό πρόβλημα σακιδίου [1].

Το γενικευμένο πρόβλημα ανάθεσης πόρων έχει μελετηθεί αρκετά στο επιστημονικό πεδίο της επιχειρησιακής έρευνας τις τελευταίες δεκαετίες. Η ανάπτυξη της επιχειρησιακής έρευνας και η ανάγκη για βελτιστοποίηση προβλημάτων, προσελκύει μεγάλη προσοχή στην ερευνητική κοινότητα. Λεπτομερής περιγραφή του προβλήματος αλλά και των βασικών μεθόδων επίλυσης του έχουν διατυπωθεί από τους Martello και Toth στο βιβλίο "Knapsack Problems, Algorithms and Computer Implementations" [1].

Ποικίλουν οι τομείς που έχουν ερευνηθεί για την εφαρμογή του γενικευμένου προβλήματος ανάθεσης πόρων. Πιο συγκεκριμένα, κάποια θέματα που έχουν εφαρμογές είναι ο προγραμματισμός εργασιών [4], η δρομολόγηση [5], ο προγραμματισμός χειροκίνητων βιομηχανικών συστημάτων [6], αλλά και η τοποθέτηση εγκαταστάσεων [7]. Περισσότερα προβλήματα που μπορούν να περιγραφούν ως προβλήματα ανάθεσης πόρων έχουν μελετηθεί από τον Pentico [8].

Μέθοδοι επίλυσης του προβλήματος μελετώνται με βασικό στόχο τη βελτιστοποίηση και καλύτερη διαχείριση του εκάστοτε προβλήματος. Ο Cattrysse et al. [9] μελέτησε λεπτομερώς διάφορους αλγόριθμους επίλυσης του γενικευμένου προβλήματος ανάθεσης με μεθόδους Branch and Bound, με ευρετικές συναρτήσεις, αλλά και με μεθόδους χαλάρωσης της αρχικής αποδεκτής λύσης. Ο Naus [10] μελέτησε

το γενικευμένο πρόβλημα ανάθεσης πόρων με σκοπό τη δημιουργία μίας μεθόδου επίλυσης με χρήση ευρετικής μεθόδου. Επίσης, έχουν μελετηθεί τροποποιήσεις του προβλήματος, όπως διατύπωσαν οι Krumke και Thielen [11] σχετικά με το γενικευμένο πρόβλημα ανάθεσης πόρων με ελάχιστες ποσότητες.

Το έργο της παρούσας εργασίας διαφέρει από τις προαναφερθέντες μελέτες σε δύο σημεία. Πρώτον, γίνεται υπολογιστική σύγκριση μεταξύ των ήδη υπαρχόντων μεθόδων επίλυσης προκειμένου να αξιολογηθούν ενδελεχώς. Δεύτερον, εισάγονται δεδομένα προβλημάτων, τα οποία χρησιμοποιούνται ευρέως σε διάφορες έρευνες, αποδίδοντας με αυτόν τον τρόπο αποτελέσματα μεγαλύτερης αξιοπιστίας και σωστότερης σύγκρισης μεταξύ των αλγορίθμων.

Σε μία μαθηματική προσέγγιση και περιγράφοντας το πρόβλημα ως πρόβλημα ακέραιου προγραμματισμού, έστω ότι έχουμε m πράκτορες και n διεργασίες. Η κάθε διεργασία έχει κέρδος p_{ij} και μία δαπάνη πόρων r_{ij} από τον κάθε πράκτορα. Επιπλέον, ο εκάστοτε πράκτορας έχει χωρητικότητα πόρων προς δαπάνη c_{ij} για την υλοποίηση των διεργασιών. Η εξίσωση μεγιστοποίησης του κέρδους ορίζεται στην εξίσωση 2.1 και οι περιορισμοί περιγράφονται στις εξισώσεις 2.2.

$$\text{maximize } \sum_{i=1}^m \sum_{j=1}^n p_{ij} x_{ij} \quad (2.1)$$

$$\begin{aligned} \text{subject to } \sum_{j=1}^n r_{ij} x_{ij} &\leq c_{ij} & i = 1, \dots, m \\ \sum_{i=1}^m x_{ij} &= 1 & j = 1, \dots, n \\ x_{ij} &\in \{0, 1\} \end{aligned} \quad (2.2)$$

Στο Σχήμα 2.1 δίνεται αριθμητικό παράδειγμα ενός γενικευμένου προβλήματος ανάθεσης πόρων. Στον πρώτο πίνακα ορίζονται τα κέρδη ανά διεργασία και πράκτορα, στον δεύτερο οι δαπάνες των διεργασιών ανά πράκτορα και τέλος, δίνεται η χωρητικότητα δαπανών ανά πράκτορα.

Σχήμα 2.1: Παράδειγμα αριθμητικού παραδείγματος για το πρόβλημα ανάθεσης πόρων

$p_{ij} =$

	agent 1	agent 2	agent 3
job 1	3	5	7
job 2	2	3	2
job 3	5	3	6
job 4	7	5	3
job 5	2	2	5

$r_{ij} =$

	agent 1	agent 2	agent 3
job 1	3	5	3
job 2	4	3	5
job 3	5	3	2
job 4	4	5	5
job 5	6	2	4

$c_{ij} =$

	agent 1	agent 2	agent 3
	28	30	26

Το παράδειγμα του Σχήματος 2.1 μπορεί να εκφραστεί στην εξίσωση μεγιστοποίησης κέρδους όπως ορίζεται στην εξίσωση 2.3, αλλά και οι περιορισμοί του σύμφωνα με τις εξισώσεις 2.4.

$$\begin{aligned}
 \text{maximize : } & 3x_{11} + 2x_{12} + 3x_{13} + 7x_{14} + 2x_{15} \\
 & + 5x_{21} + 3x_{22} + 3x_{23} + 5x_{24} + 2x_{25} \\
 & + 7x_{31} + 2x_{32} + 6x_{33} + 3x_{34} + 5x_{35}
 \end{aligned} \tag{2.3}$$

$$\begin{aligned}
 3x_{11} + 4x_{12} + 5x_{13} + 4x_{14} + 6x_{15} & \leq 28 \\
 5x_{21} + 3x_{22} + 3x_{23} + 5x_{24} + 2x_{25} & \leq 30 \\
 3x_{31} + 5x_{32} + 2x_{33} + 5x_{34} + 4x_{35} & \leq 26 \\
 x_{ij} & \in \{0, 1\}
 \end{aligned} \tag{2.4}$$

2.3 Η Ουγγρική μέθοδος

Για την επίλυση προβλημάτων τέτοιου τύπου, υπάρχουν διάφοροι μέθοδοι επίλυσης. Εάν το πρόβλημα είναι εύκολο και με λίγες μεταβλητές, μπορεί να χρησιμοποιηθεί η Ουγγρική μέθοδος [12]. Η Ουγγρική μέθοδος είναι μία γραφική μέθοδος επίλυσης, όπου ο αλγόριθμος της αναπτύχθηκε από τον Harold Kuhn το 1955. Η Ουγγρική μέθοδος ξεχωρίζει σε προβλήματα με ίσους πράκτορες και διεργασίες, χωρίς περιορισμούς, και η επίλυση του προβλήματος γίνεται γραφικά. Στο Σχήμα 2.2 μπορούμε να δούμε ένα παράδειγμα επίλυσης ενός προβλήματος με πέντε πράκτορες και πέντε διεργασίες, με στόχο την εύρεση του μέγιστου κέρδους στις αναθέσεις των διεργασιών.

Σχήμα 2.2: Παράδειγμα επίλυσης Ουγγρικής μεθόδου

jobs	agents				
	a1	a2	a3	a4	a5
j1	26	14	10	12	9
j2	31	27	30	14	16
j3	20	19	16	25	10
j4	17	12	21	30	25
j5	15	18	16	25	30

Αρχικά, στο Σχήμα 2.3 βρίσκουμε το μέγιστο κέρδος για κάθε ανάθεση διεργασίας.

Σχήμα 2.3: Επιλογή μέγιστων κερδών

jobs	agents				
	a1	a2	a3	a4	a5
j1	26	14	10	12	9
j2	31	27	30	14	16
j3	20	19	16	25	10
j4	17	12	21	30	25
j5	15	18	16	25	30

Για κάθε άλλη μικρότερη τιμή, την αντικαθιστούμε με την αφαίρεση της μικρότερης τιμής από τη μέγιστη. Οπότε ο πίνακας των αναθέσεων γίνεται όπως απεικονίζεται στο Σχήμα 2.4.

Σχήμα 2.4: Υπολογισμός πίνακα

jobs	agents				
	a1	a2	a3	a4	a5
j1	0	12	16	14	17
j2	0	4	1	17	15
j3	5	6	9	0	15
j4	13	18	9	0	5
j5	15	12	14	5	0

Στη συνέχεια, στο Σχήμα 2.5 βρίσκουμε την ελάχιστη τιμή κέρδους για κάθε πράκτορα (στήλη).

Σχήμα 2.5: Επιλογή ελάχιστων κερδών

jobs	agents				
	a1	a2	a3	a4	a5
j1	0	12	16	14	17
j2	0	4	1	17	15
j3	5	6	9	0	15
j4	13	18	9	0	5
j5	15	12	14	5	0

Βρίσκουμε τα ελάχιστα κέρδη για να κάνουμε αντικατάσταση των τιμών ανά πράκτορα, με την αφαίρεση της ελάχιστης τιμής από αυτών. Το αποτέλεσμα δίνεται στο Σχήμα 2.6.

Σχήμα 2.6: Υπολογισμός πίνακα

jobs	agents				
	a1	a2	a3	a4	a5
j1	0	8	15	14	17
j2	0	0	0	17	15
j3	5	2	8	0	15
j4	13	14	8	0	5
j5	15	8	13	5	0

Στο επόμενο βήμα, γίνεται έλεγχος της βέλτιστης λύσης. Καλύπτουμε με όσο το δυνατόν ελάχιστες γραμμές τα μηδενικά του πίνακα. Εάν οι γραμμές ήταν σε πλήθος ίσες με τους πράκτορες και τις διεργασίες, δηλαδή πέντε, θα μπορούσαμε να προσδιορίσουμε τη βέλτιστη λύση. Όμως, όπως απεικονίζεται στο Σχήμα 2.7, έχουμε τέσσερις γραμμές, οπότε θα πρέπει να γίνει μία αναπροσαρμογή στον πίνακα.

Σχήμα 2.7: Κάλυψη μηδενικών με γραμμές

jobs	agents				
	a1	a2	a3	a4	a5
j1	0	8	15	14	17
j2	0	0	0	17	15
j3	5	2	8	0	15
j4	13	14	8	0	5
j5	15	8	13	5	0

Στην αναπροσαρμογή του πίνακα στο Σχήμα 2.8, βρίσκουμε τη μικρότερη τιμή κέρδους που δεν είναι καλυπτόμενη από τις υπάρχουσες γραμμές.

Σχήμα 2.8: Εύρεση ελάχιστης τιμής

jobs	agents				
	a1	a2	a3	a4	a5
j1	0	8	15	14	17
j2	0	0	0	17	15
j3	5	2	8	0	15
j4	13	14	8	0	5
j5	15	8	13	5	0

Στις μη καλυπτόμενες τιμές, αφαιρούμε την ελάχιστη τιμή κέρδους, και καλύπτουμε τη νέα μηδενική τιμή με γραμμή. Έτσι, προκύπτουν οι απαιτούμενες πέντε γραμμές στον πίνακα για να βρεθεί το βέλτιστο κέρδος, όπως φαίνεται στο Σχήμα 2.9.

Σχήμα 2.9: Υπολογισμός πίνακα

jobs	agents				
	a1	a2	a3	a4	a5
j1	0	8	15	14	17
j2	0	0	0	17	15
j3	3	0	6	0	15
j4	11	12	6	0	5
j5	13	6	11	5	0

Στο Σχήμα 2.10 προχωράμε στη βέλτιστη ανάθεση της κάθε διεργασίας στον εκάστοτε πράκτορα. Ξεκινώντας από τις διεργασίες με μόνο ένα μηδενικό στις αναθέσεις τους, και αφήνοντας τελευταίες τις διεργασίες που έχει παραπάνω μηδενικά.

Σχήμα 2.10: Τελική ανάθεση διεργασιών

jobs	agents				
	a1	a2	a3	a4	a5
j1	0	8	15	14	17
j2	0	0	0	17	15
j3	3	0	6	0	15
j4	11	12	6	0	5
j5	13	6	11	5	0

Για να γίνει ο υπολογισμός του βέλτιστου κέρδους, πρέπει να ξαναγυρίσουμε στο Σχήμα 2.2 και στις θέσεις που βρίσκονται τα μηδενικά, να προσθέσουμε τα κέρδη της εκάστοτε ανάθεσης. Οπότε το κέρδος υπολογίζεται ως: $26 + 30 + 19 + 30 + 30 = 135$.

2.4 Λοιποί μέθοδοι επίλυσης

Η δημοφιλέστερη μέθοδος που χρησιμοποιείται για την επίλυση προβλημάτων ανάθεσης πόρων και βρίσκει τη λύση με το βέλτιστο κόστος είναι η μέθοδος Simplex [13]. Η μέθοδος Simplex αποτελείται από μία συγκεκριμένη σειρά επαναλαμβανόμενων υπολογισμών, όπου ξεκινώντας από μία αρχική κατάσταση του μοντέλου, επαναλαμβάνονται μέχρι να βρεθεί η καλύτερη τιμή κόστους της αντικειμενικής συνάρτησης. Γενικά, η μέθοδος αυτή, ξεχωρίζει για τα πλεονεκτήματά της και την πολυπλοκότητά της σε ένα μέσο πρόβλημα. Τα πλεονεκτήματα της μεθόδου Simplex είναι ότι εντοπίζει πάντα τη βέλτιστη λύση (με το ελάχιστο κόστος) και στις περισσότερες περιπτώσεις (μέσα προβλήματα) η ταχύτητα εκτέλεσης του αλγορίθμου είναι μικρή. Όσον αφορά την πολυπλοκότητα του αλγορίθμου της μεθόδου αυτής είναι εκθετική στη χειρότερη περίπτωση.

Σημαντικό είναι να αναφερθεί ότι για την επίλυση του προβλήματος ανάθεσης πόρων συνήθως χρησιμοποιούνται ευρετικοί αλγόριθμοι (heuristics). Οι αλγόριθμοι αυτοί χρησιμοποιούνται για αναζήτηση λύσης κάποιου προβλήματος και χαρακτηριστικό τους είναι η ευρετική συνάρτηση του εκάστοτε αλγορίθμου. Η ευρετική

συνάρτηση επιστρέφει στη μέθοδο μία εκτίμηση του κόστους ανάμεσα σε δύο καταστάσεις. Σημαντικό είναι τα εκτιμώμενα κόστη της ευρετικής συνάρτησης να είναι πολύ κοντά στα πραγματικά κόστη του προβλήματος. Το πλεονέκτημα των αλγορίθμων αυτών είναι ότι ο χρόνος εκτέλεσής τους και επίλυσης του προβλήματος είναι εξαιρετικά μικρός σε σχέση με τις προαναφερθείσες μεθόδους, ειδικά σε σύνθετα προβλήματα. Από την άλλη, οι λύσεις που δίνουν οι ευρετικοί αλγόριθμοι, μπορεί να μην είναι η ίδια η βέλτιστη λύση, αλλά μία προσεγγιστική τιμή κόστους, πολύ κοντά στη βέλτιστη. Για να θεωρηθούν τα αποτελέσματα ενός τέτοιου αλγορίθμου αποδεκτά, πρέπει η απόκλιση της λύσης από τη βέλτιστη να είναι πολύ μικρή.

Για την υλοποίηση της εργασίας αυτής και την επίλυση των προβλημάτων χρησιμοποιήθηκαν μεθόδευτικοί αλγόριθμοι, οι οποίοι θα αναλυθούν στο επόμενο κεφάλαιο.

Κεφάλαιο 3

Αλγόριθμοι επίλυσης γενικευμένου προβλήματος ανάθεσης πόρων

Στο κεφάλαιο αυτό θα δοθούν οι περιγραφές των μεθόδων που υλοποιήθηκαν. Γενικά όπως προαναφέρθηκε, οι αλγόριθμοι που υλοποιήθηκαν είναι μεθευρετικοί. Πιο συγκεκριμένα, για να επιλύσουν το γενικευμένο πρόβλημα ανάθεσης πόρων χρησιμοποιούν μεθόδους τοπικής αναζήτησης αλλά και άλλες μεθόδους αναζήτησης υψηλού επιπέδου για να συγκλίνουν στην πραγματική λύση του εκάστοτε παραδείγματος του προβλήματος αυτού [14]. Ο σχεδιασμός της κάθε μεθόδου δίνεται σε ψευδοκώδικα.

3.1 Μέθοδος προσομοιωμένης ανόπτωσης

Η μέθοδος προσομοιωμένης ανόπτωσης (simulated annealing method) είναι ένας δημοφιλής μεθευρετικός αλγόριθμος τοπικής αναζήτησης. Χρησιμοποιείται για την επίλυση διακριτών προβλημάτων με συνεχή και συνδυαστική βελτιστοποίηση. Στη μέθοδο αυτή, αρχικά τίθεται μία τυχαία λύση και αρχικοποιείται με τη μέγιστη τιμή της μία μεταβλητή που ονομάζεται θερμοκρασία T . Η συγκεκριμένη μεταβλητή μειώνεται βάση μίας συνάρτησης $a(T)$, όπου σύμφωνα με την αλλαγή της μεταβλητής γίνεται και η αλλαγή των καταχωρήσεων των εργασιών στους πράκτορες. Οι βασικοί τρόποι χρήσης της συνάρτησης μείωσης της θερμοκρασίας είναι είτε γραμμικά, εκθετικά ή λογαριθμικά.

Στη μηχανική, ανόπτωση ονομάζεται η θερμική επεξεργασία ενός μεταλλικού υλικού που βρίσκεται σε στερεή μορφή μέχρι να φτάσει στο σημείο τήξης και να συνεχίσει έπειτα με κάποια συνάρτηση η ψύξη του. Η μέθοδος προσομοιωμένης

ανόπτησης διαχειρίζεται το εκάστοτε πρόβλημα ως ένα πρόβλημα ανόπτησης ενός υλικού. Ο αλγόριθμος τερματίζει σε μία λύση του προβλήματος όταν το υλικό φτάσει στο τελικό σημείο ισορροπίας.

Στην εργασία αυτή, ο αλγόριθμος ανόπτησης χρησιμοποιείται για την εύρεση της καταχώρησης εργασιών στους πράκτορες. Η μέθοδος αυτή αποτελεί μία επαναληπτική διεργασία, που στην εκάστοτε επανάληψη, διερευνά εάν μία διαφορετική από την προηγούμενη ανάθεση έχει λιγότερο κόστος από την προηγούμενη κατάσταση του συστήματος.

Στον Αλγόριθμο 1 δίνεται ο ψευδοκώδικας της μεθόδου αυτής.

```
Input: initial solution  $s$   
Output: solution  $s$   
 $s = \text{random initial solution};$   
 $T = \text{max temperature};$   
 $a(T) = \text{decrease temperature function};$   
 $s' = s;$   
 $N = \text{update temperature iterations};$   
while (not termination condition reached) do  
  for ( $i$  of  $N$ ) do  
     $s' = \text{perturbation of } s';$   
    if ( $s'.\text{cost} < s.\text{cost}$ ) then  
       $s = s';$   
    else  
       $s = s' \text{ with probability } e^{(-s'.\text{cost}+s.\text{cost})/T};$   
    end  
     $T = a(T);$   
  end  
end  
return  $s;$ 
```

Αλγόριθμος 1: Ψευδοκώδικας μεθόδου προσομοιωμένης ανόπτησης

Στο Σχήμα 3.1 δίνεται σε γραφική αναπαράσταση ένα πρόβλημα όπου ορίζονται δύο πράκτορες και επτά διεργασίες. Στον πίνακα p_{ij} υπάρχουν τα κέρδη ανά πράκτορα και διεργασία. Στον δεύτερο πίνακα w_{ij} δηλώνονται οι δαπάνες πόρων των πρακτόρων ανά διεργασία, ενώ στον τρίτο πίνακα c_{ij} αναφέρονται οι συνολικοί πόροι των πρακτόρων του συστήματος.

Σχήμα 3.1: Παράδειγμα προβλήματος

$\rho_{ij} =$		job 1	job 2	job 3	job 4	job 5	job 6	job 7
	agent 1	6	9	4	2	10	3	6
	agent 2	4	8	9	1	7	5	4

$w_{ij} =$		job 1	job 2	job 3	job 4	job 5	job 6	job 7
	agent 1	4	1	2	1	4	3	8
	agent 2	9	9	8	1	3	8	7

$c_{ij} =$	agent 1	11
	agent 2	22

Η λύση του παραπάνω προβλήματος με τη μέθοδο προσομοιωμένης απόπτωσης δίνεται στο Σχήμα 3.2. Στη μέθοδο αυτή, κάνοντας διαφορετικές αναθέσεις ανά διεργασία, μελετάται το αντίστοιχο κόστος. Σε αντίθεση με άλλες μεθόδους επίλυσης, παρατηρούμε ότι με τη μέθοδο αυτή μπορεί να επιλεγθεί μία ανάθεση με μεγαλύτερο κόστος, με μία πιθανότητα p .

Σχήμα 3.2: Λύση του προβλήματος με τη μέθοδο προσομοιωμένης απόπτωσης

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7	
	1	1	2	2	1	1	2	
Cost:								243

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7	
	1	1	1	2	1	1	2	
Cost:								138

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7	
	2	1	1	2	1	1	2	
Cost:								35

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7	
	2	1	1	1	1	1	2	approved new solution with probability $e^{-(s'.Cost + s.Cost)/T}$
Cost:								36

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7	
	2	1	1	2	2	1	2	approved new solution
Cost:								32

3.2 Μέθοδος επαναληπτικής τοπικής αναζήτησης

Η μέθοδος επαναληπτικής τοπικής αναζήτησης (Iterated Local Search) εντάσσεται σε ένα σύνολο αλγοριθμικών εργαλείων που στόχο έχουν την εύρεση της βέλτιστης λύσης σ' ένα σαφέστατα οριοθετημένο πρόβλημα [15]. Ειδικότερα, η αλγοριθμική διαδικασία εξειδικεύεται στην αναζήτηση του τοπικού ελαχίστου της οριζόμενης περιοχής. Τα βασικά στοιχεία της επαναληπτικής τοπικής αναζήτησης παραθέτονται παρακάτω:

- Αλγόριθμος τοπικής αναζήτησης. Στον αλγόριθμο της τοπικής αναζήτησης, χρησιμοποιούνται διαφορετικοί τρόποι εκλογής πιθανής βέλτιστης λύσης, προκειμένου να ελεγχθεί με την τρέχουσα βέλτιστη λύση. Οι μέθοδοι αυτοί έχουν ως αποτέλεσμα την ιδανικότερη διαχείριση των υπολογιστικών πόρων του συστήματος.
- Μέθοδος διαταραχής της υπάρχουσας κατάστασης. Η χρήση των συγκεκριμένων συναρτήσεων διαταράσσουν τη δεδομένη κατάσταση του συστήματος, προκειμένου να προκύψει μία καλύτερη κατανομή των πόρων και άρα ένα μικρότερο κόστος. Οι συναρτήσεις αυτές μπορούν να κατηγοριοποιηθούν σύμφωνα με τον τρόπο διαταραχής που δημιουργούν στο σύστημα. Οι δύο βασικότερες κατηγορίες που χρησιμοποιούνται ευρέως στην αναζήτηση αυτή είναι είτε οι καθορισμένες είτε οι τυχαίες ή αλλιώς ημικαθορισμένες διαταραχές. Στην πρώτη κατηγορία, οι αλλαγές έχουν έναν τυποποιημένο τρόπο επιλογής των πόρων που θα γίνουν ανάθεση. Αντίθετα, στη δεύτερη κατηγορία, η επιλογή των πόρων που θα αλλάξουν πράκτορα, βασίζεται στην τυχαία επιλογή αυτών, ή έστω ημι-τυχαία (για παράδειγμα, τυχαία επιλογή πόρων, αλλά βάσει του μεγέθους τους).
- Κριτήριο αποδοχής της λύσης. Το τελικό στάδιο του αλγορίθμου αυτού είναι το κριτήριο αποδοχής ή απόρριψης της λύσης που δημιουργήθηκε στην εκάστοτε επανάληψη. Οι προϋποθέσεις του κριτηρίου ποικίλουν ανάλογα με τα χαρακτηριστικά του προβλήματος.

Στον Αλγόριθμο 2 δίνεται ο ψευδοκώδικας της μεθόδου αυτής.

Input: initial solution s_0
Output: solution s
 $s_0 = \text{random initial solution};$
 $T = \text{localSearch}(s);$
while (not termination condition reached) **do**
 $s_{Pert} = \text{perturbation of } s;$
 $s_{Pert} = \text{localSearch}(s_{Pert});$
 $s = \text{best}(s, s_{Pert});$
end
return $s;$

Αλγόριθμος 2: Ψευδοκώδικας μεθόδου επαναληπτικής τοπικής αναζήτησης

Για τη μέθοδο επαναληπτικής τοπικής αναζήτησης λύνεται το πρόβλημα που ορίζεται στο Σχήμα 3.1. Η λύση του προβλήματος απεικονίζεται στο Σχήμα 3.3. Με τη μέθοδο αυτή, παρατηρούμε ότι σε κάθε επανάληψη γίνεται διαταραχή της υπάρχουσας κατάστασης, και όχι μόνο μερική εναλλαγή των αναθέσεων. Επιλέγεται πάντα η λύση με το ελάχιστο κόστος.

Σχήμα 3.3: Λύση του προβλήματος με τη μέθοδο επαναληπτικής τοπικής αναζήτησης

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	2	1	2	2	1	2

Cost: 731

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	1	1	2	2	2	2	2

Cost: 533

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	1	1	2	1	1	2

Cost: 32

3.3 Μέθοδος μεταβλητής γειτονιάς αναζήτησης

Ο αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search) αποτελεί ένα υποσύνολο αλγοριθμικών τοπικών αναζητήσεων με σκοπό την εύρεση της βέλτιστης λύσης. Κύρια χρήση του αλγόριθμου αυτού είναι για την επίλυση σύνθετων προβλημάτων συνδυαστικής βελτιστοποίησης [16]. Τα βήματα της μεθόδου αυτής είναι τρία, η ανακίνηση του συστήματος, η τοπική αναζήτηση που αναλύεται παρακάτω και η κίνηση. Χρησιμοποιούνται διαφορετικές συναρτήσεις αναζήτησης τοπικών αναζητήσεων που επιστρέφουν τα τοπικά ελάχιστα και αυτό έχει έως πλεονέκτημα να βρίσκεται η βέλτιστη λύση ανάμεσα στα τοπικά βέλτιστα. Οι μέθοδοι αναζήτησης κατηγοριοποιούνται ως εξής:

- Μέθοδοι τοπικής αναζήτησης με ντετερμινιστικό τρόπο. Με τις συναρτήσεις αυτές χρησιμοποιείται ο αλγόριθμος Κατάβασης Μεταβλητής Γειτονιάς (VND) όπου η τοπική αναζήτηση εκτελείται με αιτιοκρατικό τρόπο στην τρέχουσα λύση.
- Μέθοδοι τοπικής αναζήτησης με στοχαστικό τρόπο. Στη συγκεκριμένη αναζήτηση χρησιμοποιείται ο αλγόριθμος Μειωμένης Έρευνας Μεταβλητής Γειτονιάς (RVNS). Στον αλγόριθμο αυτό, για την ανανέωση των γειτονιών, εάν υπάρχει κάποια βελτίωση στη λύση, επιλέγεται η εκάστοτε τυχαία αλλαγή. Οι αλλαγές αυτές επιτυγχάνονται με μία μέθοδο τυχαίας διαταραχής της ανάθεσης των πόρων.
- Μέθοδοι τοπικής αναζήτησης με ντετερμινιστικό και στοχαστικό τρόπο. Πρόκειται για μία σύνθετη τοπική αναζήτηση, όπου υλοποιείται ο αλγόριθμος Βασικής Έρευνας Μεταβλητής Γειτονιάς (BVNS). Η αναζήτηση αρχικά, επιλέγει μια τυχαία αρχική λύση και στη συνέχεια χρησιμοποιείται μία νέα ανάθεση που βελτιώνει τη λύση του προβλήματος αιτιοκρατικά. Η μέθοδος αυτή μπορεί να εκτελεστεί κατ' επανάληψη, μέχρι η τελική λύση να έχει βέλτιστο κόστος από την αρχική λύση.

Στον Αλγόριθμο 3 δίνεται ο ψευδοκώδικας της μεθόδου αυτής.

Input: initial solution s

Output: solution s

$neighborhood = choose\ a\ set\ of\ neighborhood;$

$s = initial\ solution;$

for (i of N iterations) **do**

$s^* = neighborhood(i);$

$s^{**} = localSearch(s);$

if ($s^{**}.cost < s^*.cost$) **then**

$s = s^{**};$

$i = 1;$

end

end

return $s;$

Αλγόριθμος 3: Ψευδοκώδικας μεθόδου αναζήτησης μεταβλητής γειτονιάς

Μέρος λύσης του προβλήματος του Σχήματος 3.1 με τη μέθοδο Μεταβλητής Γειτονιάς Αναζήτησης δίνεται στο Σχήμα 3.4. Με τη μέθοδο αυτή, αναζητούνται οι γειτονικές καταστάσεις της τρέχουσας ανάθεσης και έπειτα από την εφαρμογή τοπικής αναζήτησης, γίνεται αποδεκτή η κατάσταση με το ελάχιστο κόστος.

Σχήμα 3.4: Λύση του προβλήματος με τη μέθοδο αναζήτησης μεταβλητής γειτονιάς

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	2	1	2	2	1	2

Cost: 731

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	1	1	2	2	2	2

Cost: 634

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	1	1	2	2	2	2	2

Cost: 533

•
•
•

3.4 Μέθοδος άπληστης τυχαιοποιημένης τοπικής αναζήτησης

Η μέθοδος της άπληστης τυχοποιημένης προσαρμοστικής αναζήτησης (Greedy Randomized Adaptive Search) χρησιμοποιείται κυρίως σε προβλήματα συνδυαστικής βελτιστοποίησης, για την εύρεση μίας προσεγγιστικής λύσης [17]. Ο αλγόριθμος της μεθόδου αποτελείται από δύο φάσεις, και επαναλαμβάνεται μέχρι κάποιο κριτήριο τερματισμού να ικανοποιηθεί (αριθμός επαναλήψεων, μέγεθος σφάλματος, κλπ).

Στην πρώτη φάση όπου την ονομάζουμε φάση κατασκευής (construction phase), χρησιμοποιείται μία μέθοδος κατασκευής άπληστης λύσης. Στη δεύτερη φάση που ονομάζεται τοπικής αναζήτησης (local search phase), υλοποιείται μία τοπική αναζήτηση για τη βελτίωση της λύσης που επιλέχθηκε παραπάνω. Οι φάσεις αυτές επαναλαμβάνονται για νέες λύσεις, και η επιλογή των στοιχείων των λύσεων αυτών γίνεται σύμφωνα με τυχαία επιλογή από μία λίστα υποψήφιων αλλαγών – στοιχείων (restricted candidate list).

Η λίστα υποψήφιων στοιχείων μπορεί να θεωρηθεί πολύ σημαντική για τον αλγόριθμο αυτό, καθώς από αυτή γίνεται έλεγχος της διασποράς των λύσεων που θα δημιουργηθούν στην εκάστοτε επανάληψη. Προβλήματα που θα μπορούσε να δημιουργηθούν στη μέθοδο αυτή εξαιτίας της λίστας υποψηφίων θα μπορούσαν να αφορούν το μήκος της λίστας. Για παράδειγμα, μία πολύ μικρή λίστα θα δημιουργούσε σε κάθε επανάληψη σχεδόν την ίδια αρχική λύση, ενώ μία πολύ μεγάλη λίστα θα δημιουργούσε τελείως τυχαίες λύσεις μεταξύ τους. Η λίστα υποψήφιων στοιχείων μπορεί να είναι είτε πεπερασμένου μήκους είτε δυναμική και να εντάσσονται σε αυτήν μόνο στοιχεία που βρίσκονται μέσα σε ένα διάστημα ή είναι μικρότερα από μία τιμή κατωφλίου.

Σημαντικό είναι να αναφερθεί ότι η μέθοδος αυτή επιστρέφει μία εφικτή λύση στο τέλος κάθε επανάληψης, ενώ η τελική λύση της μεθόδου είναι η βέλτιστη λύση μεταξύ όλων των επαναλήψεων του αλγορίθμου. Βασικό μειονέκτημα της μεθόδου της άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης είναι ότι οι επαναλήψεις είναι ανεξάρτητες μεταξύ τους [18], δηλαδή η μέθοδος δε μαθαίνει από τις προηγούμενες επαναλήψεις, καθώς δεδομένα από οποιαδήποτε λύση πλην της βέλτιστης δεν κρατούνται.

Στον Αλγόριθμο 4 δίνεται ο ψευδοκώδικας της μεθόδου αυτής.

Input: initial solution s

Output: solution s^*

while (*not termination condition reached*) **do**

$s = \text{initial solution};$

$s^* = \text{localSearch}(s);$

if ($s.\text{cost} < s^*.\text{cost}$) **then**

$s^* = s;$

end

end

return $s^*;$

Αλγόριθμος 4: Ψευδοκώδικας μεθόδου άπληστης τυχαιοποιημένης τοπικής αναζήτησης

Το πρόβλημα του Σχήματος 3.1 επιλύεται με τη μέθοδο άπληστης τυχαιοποιημένης τοπικής αναζήτησης στο Σχήμα 3.5. Όπως απεικονίζεται στο σχήμα αυτό, η μέθοδος αυτή, έχοντας μία αρχική κατάσταση, εφαρμόζει επαναληπτικά τοπική αναζήτηση σε αυτή και επιλέγει την κατάσταση με το ελάχιστο κόστος.

Σχήμα 3.5: Λύση του προβλήματος με τη μέθοδο άπληστης τυχαιοποιημένης τοπικής αναζήτησης

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	2	1	2	2	1	2

Cost: 731

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	1	1	2	2	2	2

Cost: 634

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	1	1	2	2	2	2	2

Cost: 541

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	1	1	2	2	1	2

Cost: 32

3.5 Μέθοδος περιορισμένης αναζήτησης

Η περιορισμένη αναζήτηση (Tabu Search) είναι ένας μεθευρετικός αλγόριθμος αναζήτησης που χρησιμοποιεί τοπική αναζήτηση [19]. Αποτελεί μία μέθοδο αναζήτησης και σε πολλές περιπτώσεις προβλημάτων βελτιστοποίησης επιστρέφει αποδεκτές και συγκλίνουσες λύσεις. Παρουσιάστηκε τη δεκαετία του '80 από τον Αμερικανό επιστήμονα Fred W. Glover [19].

Ο αλγόριθμος αναζήτησης αυτός, χρησιμοποιεί έναν ευρετικό αλγόριθμο επιλογής για να βρει τις διαφορετικές λύσεις, όπου όμως υπάρχει μεγάλη πιθανότητα να παγιδευτεί σε τοπικό ελάχιστο. Για τον λόγο αυτόν, δεν επιλέγεται μία τυχαία λύση για την επόμενη επανάληψη, αλλά χρησιμοποιείται μία λίστα (Tabu List) ως μνήμη των επαναλήψεων. Οι κινήσεις της εκάστοτε επανάληψης του αλγορίθμου αναγράφονται στη λίστα αυτή, που μπορεί να έχει πολύ περιορισμένο μήκος (τελευταίες επαναλήψεις) ή δυναμικά μεγάλο μήκος όπου εγγράφονται όλες οι επαναλήψεις. Με τον τρόπο αυτόν αποφεύγεται ο αλγόριθμος να επισκεφθεί ξανά ίδιες γειτονιές, και η αναζήτηση γίνεται πιο εκτεταμένη.

Ο αλγόριθμος της περιορισμένης αναζήτησης δε συγκλίνει με φυσικό τρόπο στη βέλτιστη λύση και για τον λόγο αυτόν χρησιμοποιούνται κριτήρια τερματισμού των επαναλήψεων του [20].

Στον Αλγόριθμο 5 δίνεται ο ψευδοκώδικας της μεθόδου αυτής.

```
Input: initial solution  $s_0$   
Output: solution  $s_{Best}$   
 $s_0 = \text{initial solution};$   
 $s_{Best} = s_0;$   
 $bestCandidate = s_0;$   
 $tabuList = [];$   
 $tabuList.push(s_0);$   
while (not termination condition reached) do  
   $s_{Neighborhood} = getNeighbors(bestCandidate);$   
   $bestCandidate = s_{Neighborhood}[0];$   
  for ( $s_{Candidate}$  in  $s_{Neighborhood}$ ) do  
    if ((not  $tabuList.contains(s_{Candidate})$ ) and  $fitness(s_{Candidate}) >$   
       $fitness(bestCandidate)$ ) then  
      |  $bestCandidate = s_{Candidate};$   
    end  
  end  
  if ( $fitness(bestCandidate) > fitness(s_{Best})$ ) then  
  |  $s_{Best} = bestCandidate;$   
  end  
   $tabuList.push(bestCandidate);$   
  if ( $tabuList.size > maxTabuSize$ ) then  
  |  $tabuList.removeFirst();$   
  end  
end  
return  $s_{Best};$ 
```

Αλγόριθμος 5: Ψευδοκώδικας μεθόδου περιορισμένης αναζήτησης

Στο Σχήμα 3.6 δίνεται η λύση του προβλήματος του Σχήματος 3.1 με τη μέθοδο περιορισμένης αναζήτησης. Παρατηρούμε πως η μέθοδος αυτή μπορεί με λιγότερες επαναλήψεις να βρει νωρίτερα τη βέλτιστη λύση.

Σχήμα 3.6: Λύση του προβλήματος με τη μέθοδο περιορισμένης αναζήτησης

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	2	1	2	2	1	2

Cost: 731

assignment	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	1	1	2	2	1	2

Cost: 32

3.6 Μέθοδος αποδοχής κατωφλιού

Στον αλγόριθμο αναζήτησης της μεθόδου Αποδοχής Κατωφλιού (Threshold Accepted Method) δημιουργούνται βάση μίας επαναληπτικής διαδικασίας κύκλοι αποδοχής μίας λύσης βάση της αρχικής λύσης που της δίνεται. Πρόκειται για μία μέθοδο αιτιοκρατικής ανόπτωσης και αποτελεί μία διαφορετική έκδοση της μεθόδου αυτής. Η κύρια διαφορά της μεθόδου αποδοχής κατωφλιού είναι ότι το τοπικό ελάχιστο αλλάζει μέσω μίας τιμής αποδοχής – κατωφλιού T που ορίζεται ως άνω όριο για την αύξηση της τιμής στην αντικειμενική συνάρτηση για την επόμενη επανάληψη.

Σημαντικό είναι να αναφερθεί ότι η μέθοδος αυτή αποτελεί αποτελεσματικότερη λύση της μεθόδου ανόπτωσης. Στη μέθοδο ανόπτωσης παράγεται ένας τυχαίος αριθμός και συμβαδίζει με την εκθετική συνάρτηση, ενώ στη μέθοδο αποδοχής κατωφλιού, ο αριθμός γίνεται αποδεκτός μέσω της τιμής κατωφλιού. Με αυτόν τον τρόπο, η επόμενη κίνηση εγγυάται ότι δε θα κάνει την τρέχουσα λύση αρκετά χειρότερη.

Στην έναρξη του αλγορίθμου, η τιμή του κατωφλιού είναι αρκετά υψηλή και με αυτόν τον τρόπο μπορούν να ερευνηθούν στις πρώτες επαναλήψεις διαφορετικές γειτονιές λύσεων, όπως στην περίπτωση της μεθόδου ανόπτωσης. Στη συνέχεια όμως, το κατώφλι μειώνεται σημαντικά και οι επαναλήψεις της έρευνας εφαρμόζονται στην περιοχή λύσεων του προβλήματος.

Καθώς η τιμή κατωφλιού είναι πολύ σημαντική στον αλγόριθμο αυτό, μια παραλλαγή της μεθόδου είναι η ανατροφοδότηση της συνάρτησης μείωσης του κατωφλιού με καινούργιες τιμές, όταν οι επιλογές αποδοχής λύσεων με βάση το κριτήριο επιλογής του κατωφλιού έχουν μηδενιστεί. Σε αντίθετη περίπτωση, οι αποδεκτές καινούργιες λύσεις είναι πολύ λίγες ή μηδανιές και δεν υπάρχουν ουσιαστικά νέες αποδεκτές λύσεις σε κάθε επανάληψη. Με την ανατροφοδότηση του κατωφλιού με νέες τιμές, δίνεται η δυνατότητα διερεύνησης και μετατόπισης του αλγορίθμου αναζήτησης σε νέες περιοχές με σκοπό τον εντοπισμό καλύτερων λύσεων.

Στον Αλγόριθμο 6 δίνεται ο ψευδοκώδικας της μεθόδου αυτής.

```
Input: initial solution  $s_0$   
Output: solution  $s_0$   
 $s_0 = \text{random initial solution};$   
 $T = \text{max temperature};$   
 $a(T) = \text{decrease Temperature function};$   
 $N = \text{threshold iterations};$   
while (not termination condition reached) do  
  for ( $i$  of  $N$ ) do  
     $s = \text{random neighbor of } s_0;$   
     $\text{delta} = s.\text{cost} - s_0.\text{cost};$   
    if ( $\text{delta} < T$ ) then  
       $s_0 = s;$   
    end  
  end  
   $T = a(T);$   
end  
return  $s_0;$ 
```

Αλγόριθμος 6: Ψευδοκώδικας μεθόδου αποδοχής κατωφλιού

Μέρος της επίλυσης με τη μέθοδο αποδοχής κατωφλιού του προβλήματος του Σχήματος 3.1 δίνεται στο Σχήμα 3.7. Όπως παρατηρείται, οι επαναλήψεις της μεθόδου επίλυσης αυτής μοιάζουν αρκετά με τη μέθοδο προσομοιωμένης ανόπτωσης, με τη διαφορά ότι δεν επιλέγεται κατάσταση με κόστος μεγαλύτερο της ήδη υπάρχουσας κατάστασης.

Σχήμα 3.7: Λύση του προβλήματος με τη μέθοδο αποδοχής κατωφλιού

		•					
		•					
		•					
assignmen	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	2	1	1	1	1	2
Cost:	335						
assignmen	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	1	1	1	1	1	2
Cost:	36						
assignmen	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	1	1	2	2	1	2
Cost:	35						
assignmen	job 1	job 2	job 3	job 4	job 5	job 6	job 7
	2	1	1	2	2	1	2
Cost:	32						

Κεφάλαιο 4

Υπολογιστική μελέτη

Στο κεφάλαιο αυτό θα παρουσιαστούν αναλυτικά τα αποτελέσματα που προέκυψαν από την υπολογιστική υλοποίηση των παραπάνω μεθωρετικών αλγορίθμων. Αρχικά, δίνονται πληροφορίες σχετικά με τα προβλήματα που χρησιμοποιήθηκαν για την επίλυση από τις μεθόδους αναζήτησης. Στη συνέχεια, περιγράφεται τεχνικά η υλοποίηση των αλγορίθμων αναζήτησης. Τέλος, δίνονται τα αποτελέσματα των προβλημάτων που υποβλήθηκαν σε λύση και συγκρίνονται με βάση τους αλγορίθμους που υλοποιήθηκαν.

4.1 Δεδομένα προβλημάτων

Για την υλοποίηση της παρούσας υπολογιστικής μελέτης, χρησιμοποιήθηκαν τα προβλήματα της βιβλιοθήκης OR-Library για το πρόβλημα ανάθεσης πόρων [21]. Επιλέχθηκε η συλλογή προβλημάτων αυτή, γιατί είναι μία ευρέως εφαρμοσμένη βιβλιοθήκη προβλημάτων που έχει χρησιμοποιηθεί σε διάφορες ακαδημαϊκές μελέτες. Κάποιες αναφορές εργασιών που βασίστηκαν στα δεδομένα αυτά είναι των Talaván & Yáñez [22] και των Albareda-Sambola et al. [23].

Στην εργασία αυτή, χρησιμοποιήθηκαν τα προβλήματα που μελετώνται για το γενικευμένο πρόβλημα ανάθεσης πόρων και βρίσκονται στα αρχεία gap1.txt έως gap12.txt. Στο εκάστοτε αρχείο, ορίζονται προβλήματα μαθηματικά διατυπωμένα και τίθενται προς επίλυση για την εύρεση του ελαχίστου κόστους. Τα προβλήματα που ορίζονται ανά αρχείο είναι πέντε και όσο αυξάνεται ο αριθμός του αρχείου, τα προβλήματα μεγαλώνουν σε μέγεθος και πολυπλοκότητα.

Επίσης, η σειρά δήλωσης των δεδομένων των προβλημάτων μέσα στα αρχεία είναι συγκεκριμένη. Αρχικά, ορίζεται ο αριθμός προβλημάτων που υπάρχουν στο

αρχείο και στη συνέχεια δηλώνονται το πλήθος των πρακτόρων, το πλήθος των διεργασιών, το κόστος ανάθεσης κάθε διεργασίας στον εκάστοτε πράκτορα, οι πόροι που χρησιμοποιούνται για την κάθε ανάθεση διεργασίας ανά πράκτορα και οι συνολικοί πόροι του εκάστοτε πράκτορα. Στο Σχήμα 4.1 δίνεται παράδειγμα δήλωσης ενός προβλήματος από το αρχείο gap1.txt.

Σχήμα 4.1: Δήλωση δεδομένων προβλήματος - μέρος αρχείου gap1.txt

```
5
5 15
17 21 22 18 24 15 20 18 19 18 16 22 24 24 16
23 16 21 16 17 16 19 25 18 21 17 15 25 17 24
16 20 16 25 24 16 17 19 19 18 20 16 17 21 24
19 19 22 22 20 16 19 17 21 19 25 23 25 25 25
18 19 15 15 21 25 16 16 23 15 22 17 19 22 24
8 15 14 23 8 16 8 25 9 17 25 15 10 8 24
15 7 23 22 11 11 12 10 17 16 7 16 10 18 22
21 20 6 22 24 10 24 9 21 14 11 14 11 19 16
20 11 8 14 9 5 6 19 19 7 6 6 13 9 18
8 13 13 13 10 20 25 16 16 17 10 10 5 12 23
36 34 38 27 33
```

Με τον τρόπο που είναι δηλωμένα τα δεδομένα των προβλημάτων είναι εύκολο να αναπτυχθεί πρόγραμμα που να διαβάζει τα δεδομένα αυτόματα.

Ειδικότερα, στο Πίνακα 4.1 παρουσιάζονται ο αριθμός των πρακτόρων, ο αριθμός των διεργασιών και το πραγματικό ελάχιστο κόστος του προβλήματος για όλα τα προβλήματα της βιβλιοθήκης. Το κόστος υπολογίστηκε με τη μέθοδο Simplex με τη χρήση του προγράμματος επίλυσης του Lorenzo Gatto [24].

Πίνακας 4.1: Δεδομένα και λύσεις των προβλημάτων

Πρόβλημα	Αρ. Πρακτόρων	Αρ. Διεργασιών	Πραγματικό ελαχ. κόστος
gap1.1	5	15	254
gap1.2	5	15	253
gap1.3	5	15	247
gap1.4	5	15	266
gap1.5	5	15	246
gap2.1	5	20	269
gap2.2	5	20	259
gap2.3	5	20	253
gap2.4	5	20	260
gap2.5	5	20	264
gap3.1	5	25	431
gap3.2	5	25	412
gap3.3	5	25	441
gap3.4	5	25	422
gap3.5	5	25	408
gap4.1	5	30	415
gap4.2	5	30	415
gap4.3	5	30	417
gap4.4	5	30	387
gap4.5	5	30	398
gap5.1	8	24	398
gap5.2	8	24	384
gap5.3	8	24	379
gap5.4	8	24	378
gap5.5	8	24	388
gap6.1	8	32	518
gap6.2	8	32	518
gap6.3	8	32	513
gap6.4	8	32	513
gap6.5	8	32	517
gap7.1	8	40	639
gap7.2	8	40	654
gap7.3	8	40	654
gap7.4	8	40	640
gap7.5	8	40	643
gap8.1	8	48	790
gap8.2	8	48	775
gap8.3	8	48	790
gap8.4	8	48	784
gap8.5	8	48	785

Πρόβλημα	Αρ. Πρακτόρων	Αρ. Διεργασιών	Πραγματικό ελαχ. κόστος
gap9.1	10	30	475
gap9.2	10	30	470
gap9.3	10	30	487
gap9.4	10	30	484
gap9.5	10	30	478
gap10.1	10	40	628
gap10.2	10	40	631
gap10.3	10	40	644
gap10.4	10	40	630
gap10.5	10	40	632
gap11.1	10	50	567
gap11.2	10	50	578
gap11.3	10	50	581
gap11.4	10	50	570
gap11.5	10	50	578
gap12.1	10	60	968
gap12.2	10	60	948
gap12.3	10	60	937
gap12.4	10	60	946
gap12.5	10	60	940

4.2 Υλοποίηση

Για την υλοποίηση των αλγορίθμων αναπτύχθηκε ένα πρόγραμμα επίλυσης προβλημάτων που εμπεριέχει τους αλγορίθμους των μεθόδων που προαναφέρθηκαν στο Κεφάλαιο 3. Το πρόγραμμα αναπτύχθηκε στη γλώσσα προγραμματισμού c, καθώς η γλώσσα αυτή έχει πολλά πλεονεκτήματα σε σχέση με άλλες γλώσσες προγραμματισμού. Πρόκειται για μία γλώσσα προγραμματισμού υψηλού επιπέδου, εύκολα αναγνώσιμη από τον άνθρωπο και ευρέως διαδεδομένη. Ένα κύριο χαρακτηριστικό της που είναι πολύ σημαντικό σε σχέση με άλλες γλώσσες προγραμματισμού υψηλού επιπέδου είναι ότι είναι γρήγορη και δε χρειάζεται πολύ μνήμη για να εκτελεστούν τα προγράμματα που είναι γραμμένα στη γλώσσα c.

Το πρόγραμμα αποτελείται από δύο βασικά μέρη. Το κυρίως πρόγραμμα, όπου εκτελείται για τον υπολογισμό των αποτελεσμάτων και τα αρχεία δήλωσης των μεθόδων που είναι χωρισμένα σε επιμέρους αρχεία για την ευκολότερη διατύπωση των αλγορίθμων αναζήτησης. Αρχικά, το κυρίως πρόγραμμα διαβάζει τα δεδομένα από έναν φάκελο "data" και όλα τα αρχεία του προγράμματος βρίσκονται στον ίδιο κατάλογο αρχείων. Για το εκάστοτε αρχείο δεδομένων, το πρόγραμμα ανοίγει και

διαβάζει τα δεδομένα που είναι ορισμένα μέσα του και τα αποθηκεύει σε πίνακες. Στη συνέχεια και για κάθε πρόβλημα, καλεί τη συνάρτηση της εκάστοτε μεθόδου, εισάγοντας της τα δεδομένα του προβλήματος. Οι λύσεις που βρίσκουν οι μέθοδοι και οι χρόνοι εκτέλεσής τους ανά πρόβλημα και αρχείο, αποθηκεύονται σε νέα αρχεία απλού κειμένου, στον φάκελο "results".

Στο Σχήμα 4.2 δίνεται μέρος του αρχείου λύσεων για τα προβλήματα του αρχείου gap1.txt.

Σχήμα 4.2: Δήλωση αποτελεσμάτων ανά μέθοδο για τα προβλήματα του gap1.txt

```
SIMULATED ANNEALING
cost: 270
assignment: 1 2 3 5 4 4 4 5 1 4 2 2 3 1 3
time: 0.046000
cost: 273
assignment: 3 5 4 3 3 5 1 1 5 2 3 1 2 4 2
time: 0.047000
cost: 256
assignment: 4 3 2 1 2 1 2 4 4 5 4 1 3 3 5
time: 0.032000
cost: 280
assignment: 2 2 3 5 4 5 1 3 2 3 4 1 5 5 4
time: 0.031000
cost: 257
assignment: 5 4 2 3 4 5 3 3 3 1 1 5 2 1 4
time: 0.031000
#####
TABU SEARCH
cost: 270
assignment: 1 2 3 5 4 4 4 5 1 4 2 1 3 2 3
time: 0.078000
cost: 278
assignment: 3 1 1 3 4 5 3 2 5 4 3 1 2 2 5
time: 0.063000
cost: 259
assignment: 4 3 5 1 2 1 2 4 2 3 4 1 3 5 5
```

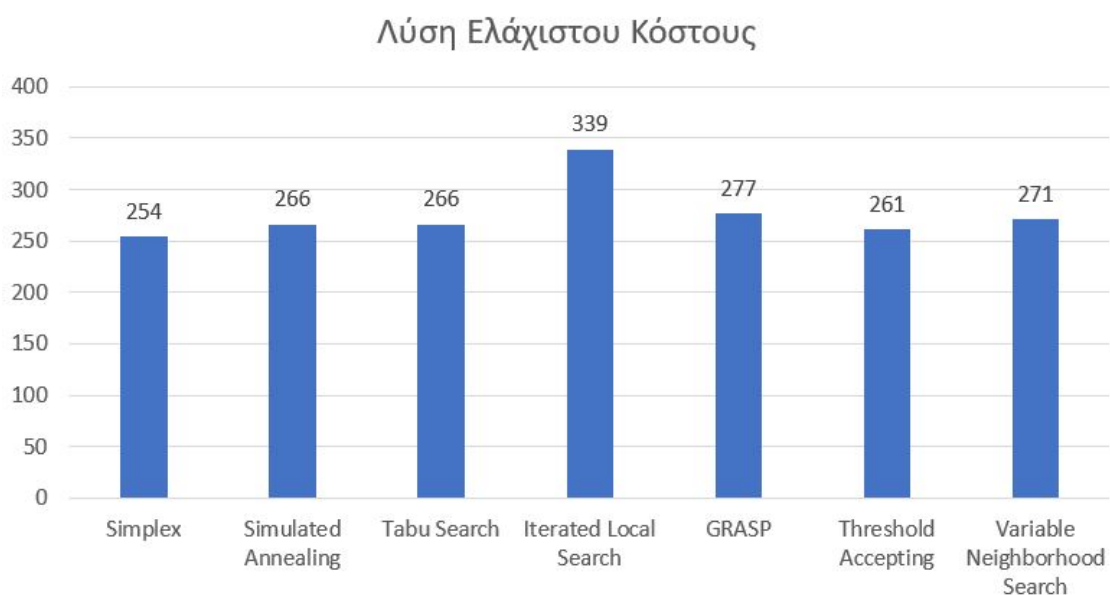
4.3 Αποτελέσματα και σύγκριση μεθόδων

Για τη σύγκριση και εξαγωγή αποτελεσμάτων βάση των αλγορίθμων, εκτελέσθηκε το πρόγραμμα για όλα τα προβλήματα πέντε φορές και βγήκε ένας μέσος όρος των αποτελεσμάτων τους. Οι αλγόριθμοι μπορούν να συγκριθούν βάση δύο στοιχείων. Πρώτον, βάση του τελικού αποτελέσματος, μεταξύ των αλγορίθμων αλλά και κατά πόσο συγκλίνει στο πραγματικό ελάχιστο κόστος. Το πραγματικό ελάχιστο κόστος υπολογίσθηκε με τη μέθοδο Simplex με το πρόγραμμα επίλυσης του Lorenzo Gatto [24]. Το δεύτερο στοιχείο σύγκρισης είναι ο χρόνος εκτέλεσης της εκάστοτε μεθόδου, ο χρόνος δηλαδή που χρειάστηκε για την εύρεση του ελάχιστου κόστους ανάθεσης.

Αναλυτικά τα αποτελέσματα των αλγορίθμων ανά πρόβλημα και αρχείο δίνονται στο παράρτημα της εργασίας. Στο κεφάλαιο αυτό, θα μελετηθούν συγκεκριμένα προβλήματα όπου υπάρχουν σημεία ενδιαφέροντος προς σύγκριση.

Ειδικότερα, στο πρώτο αρχείο, το gap1.txt, τα προβλήματα έχουν 5 πράκτορες και 15 διεργασίες προς ανάθεση. Συγκρίνοντας έστω το πρώτο πρόβλημα του αρχείου ως προς την επιστρεφόμενη τιμή λύσης της κάθε μεθόδου παρατηρούμε τα κόστη που δίνονται στο Σχήμα 4.3.

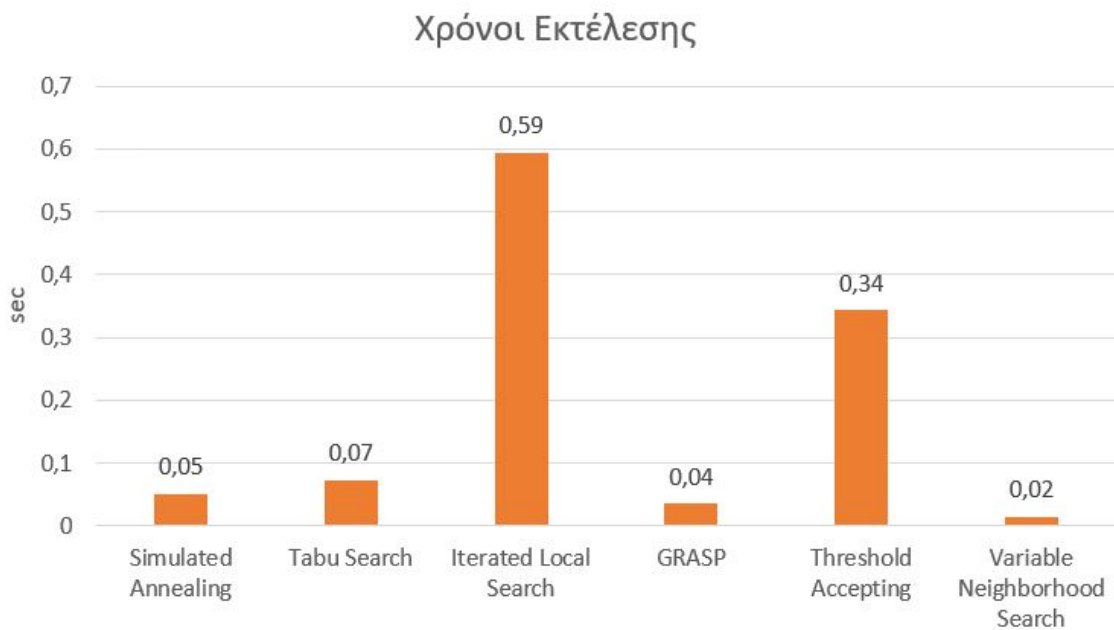
Σχήμα 4.3: Διάγραμμα ελαχίστου κόστους - gap1.txt



Παρατηρούμε ότι το ελάχιστο κόστος που επιστρέφεται με τη μέθοδο Simplex είναι 254 και η μέθοδος αποδοχής κατωφλιού συγκλίνει καλύτερα με την πραγματική λύση με τιμή 261. Από την άλλη, η μέθοδος επαναληπτικής τοπικής αναζήτησης επιστρέφει αρκετά μεγαλύτερο κόστος, την τιμή 339, σε σχέση με τη λύση που βρέθηκε με τη μέθοδο Simplex. Οι υπόλοιποι αλγόριθμοι έχουν αποδεκτά αποτελέσματα.

Σε σχέση με τον χρόνο εκτέλεσης των αλγορίθμων δεν μπορούμε να τους συγκρίνουμε με τη μέθοδο Simplex, αλλά ακολουθεί μεταξύ τους σύγκριση στο Σχήμα 4.4.

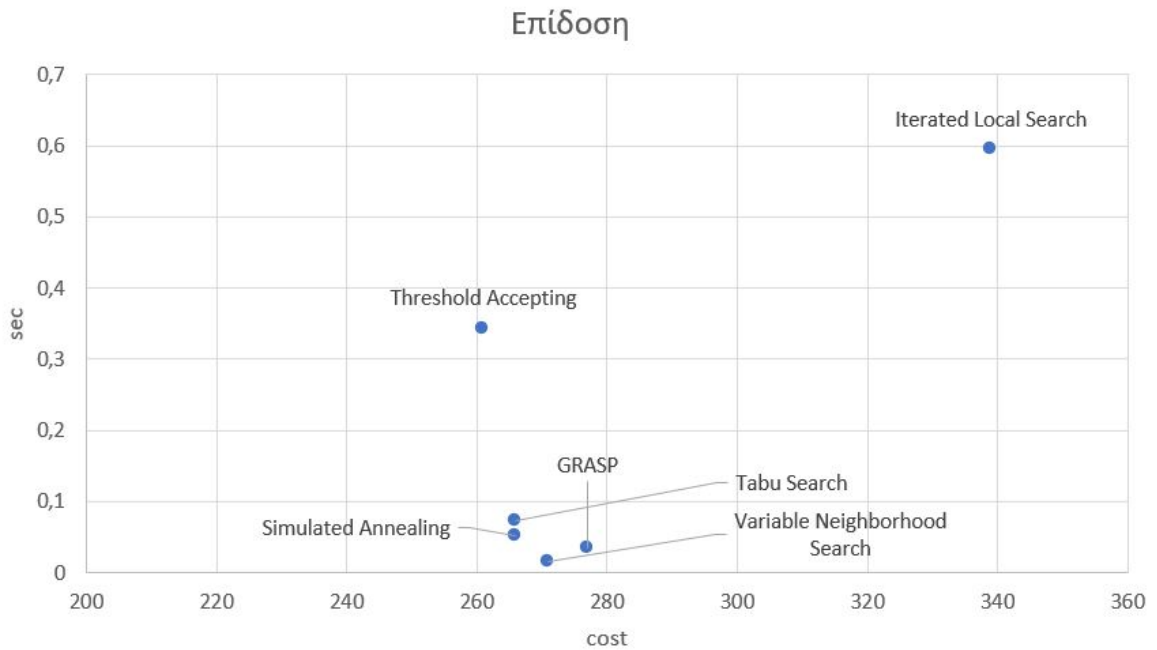
Σχήμα 4.4: Διάγραμμα χρόνων εκτέλεσης - gap1.txt



Οι χρόνοι εκτέλεσης των μεθόδων είναι αρκετά διαφορετικοί. Μετρώντας τους χρόνους εκτέλεσης σε δευτερόλεπτα, η μέθοδος μεταβλητής γειτονιάς αναζήτησης εκτελείται γρηγορότερα από τους υπόλοιπους, με μέσο χρόνο εκτέλεσης 0.02 sec. Αντίθετα, η μέθοδος επαναληπτικής τοπικής αναζήτησης είναι πιο αργή, με μέσο χρόνο εκτέλεσης 0.59 sec.

Για να συγκριθούν οι μέθοδοι και να υπάρξει μία συσχέτιση ανάμεσα στο κόστος και τον χρόνο εκτέλεσης της εκάστοτε μεθόδου, δίνεται στο Σχήμα 4.5 το διάγραμμα διασποράς του ίδιου προβλήματος.

Σχήμα 4.5: Διάγραμμα επιδόσεων αλγορίθμων - gap1.txt

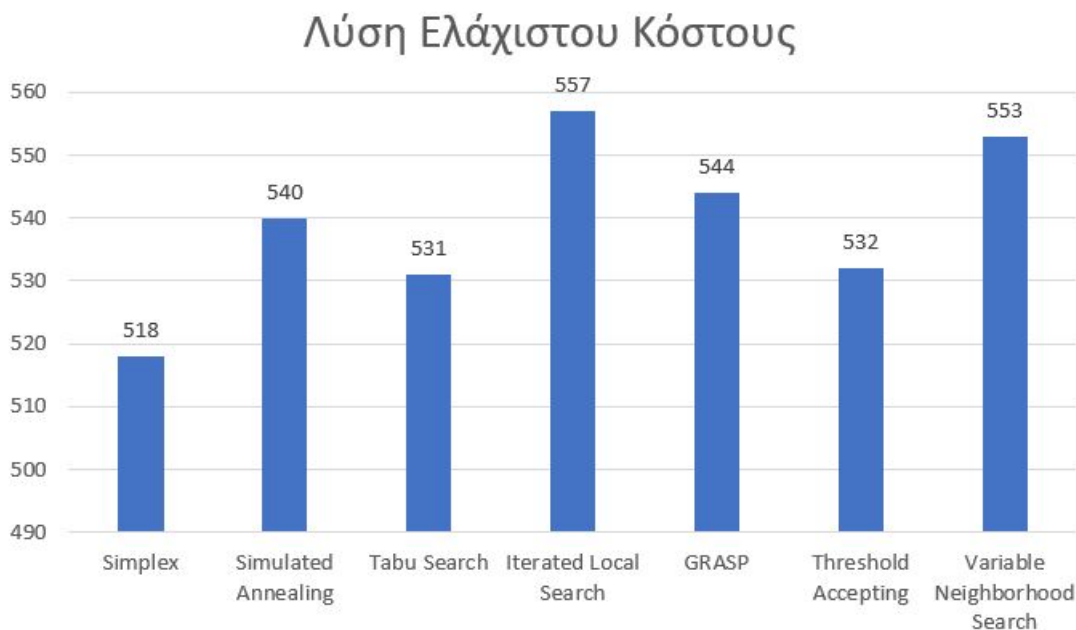


Επειδή το πρόβλημα είναι εύρεσης ελαχίστου κόστους και μας ενδιαφέρει ο ελάχιστος χρόνος εκτέλεσης, μπορούμε να θεωρήσουμε ως αποδοτικότερους αλγόριθμους αυτούς που το σημείο τους βρίσκεται αριστερά και χαμηλά στο διάγραμμα. Έτσι, ενώ η μέθοδος αποδοχής κατωφλιού επιστρέφει μικρότερο κόστος, οι μέθοδοι προσομοιωμένης ανόπτωσης και περιορισμένης αναζήτησης έχουν μικρότερους χρόνους εκτέλεσης οπότε μπορούν να θεωρηθούν επίσης αποδοτικοί για το πρόβλημα αυτό. Επιπλέον, όπως παρατηρήσαμε και παραπάνω, η μέθοδος επαναληπτικής τοπικής αναζήτησης έχει μεγάλο κόστος ανάθεσης και αρκετό χρόνο εκτέλεσης ως αλγόριθμος.

Μεσαίας πολυπλοκότητας προβλήματα, μπορούν να θεωρηθούν τα προβλήματα που είναι δηλωμένα στο αρχείο gar6.txt. Στο αρχείο αυτό, τα προβλήματα έχουν ως δεδομένα 8 πράκτορες και 32 διεργασίες προς ανάθεση. Παρακάτω παρουσιάζονται τα αποτελέσματα του πρώτου προβλήματος του έκτου αρχείου. Σκοπός είναι να αναλυθούν οι μέθοδοι βάσει ενός προβλήματος μεσαίας πολυπλοκότητας.

Όπως παραπάνω, έτσι και στο πρόβλημα αυτό, θα συγκρίνουμε τις μεθόδους σύμφωνα με τις λύσεις τους σε σχέση με την πραγματική ελάχιστη λύση που δίνεται από τη μέθοδο Simplex, αλλά και σε σχέση με τον χρόνο εκτέλεσής τους. Στο Σχήμα 4.6 δίνεται το διάγραμμα των λύσεων των μεθόδων.

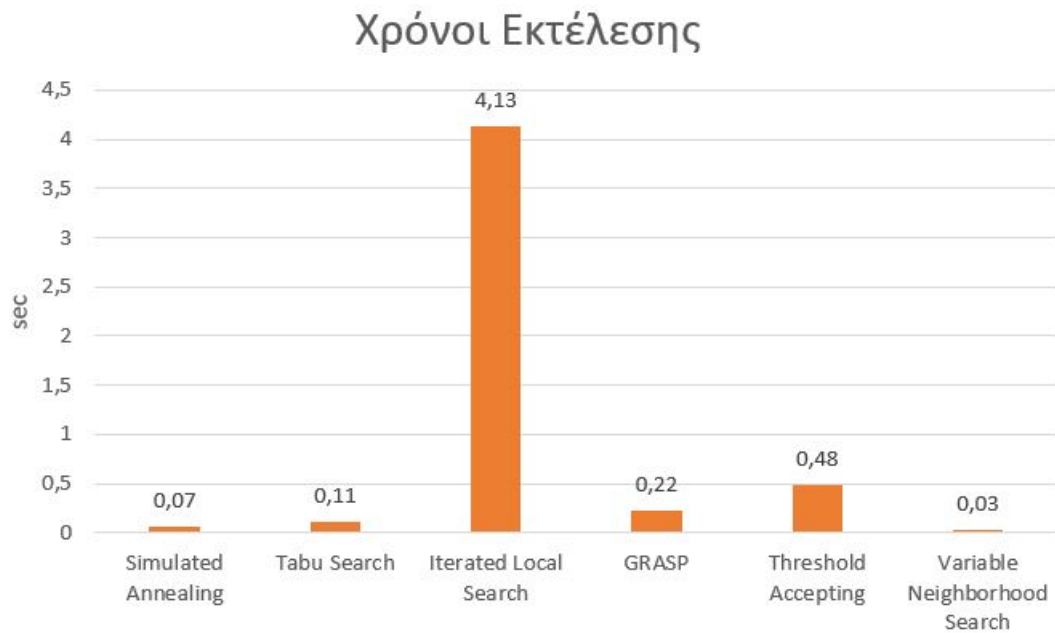
Σχήμα 4.6: Διάγραμμα ελαχίστου κόστους - gar6.txt



Όπως φαίνεται στο διάγραμμα, το ελάχιστο κόστος που υπολογίστηκε με τη μέθοδο Simplex είναι 518. Στο παράδειγμα αυτό, παρατηρούμε ότι όλοι οι αλγόριθμοι που έχουν υλοποιηθεί έχουν μεγαλύτερες διαφορές σε σχέση με το προηγούμενο, πιο απλό πρόβλημα. Οι μέθοδοι που συγκλίνουν στη βέλτιστη λύση είναι η μέθοδος της άπληστης τυχαιοποιημένης τοπικής αναζήτησης με μέσο κόστος 531 και η μέθοδος αποδοχής κατωφλιού με μέσο κόστος 532. Από την άλλη, οι μέθοδοι επαναληπτικής τοπικής αναζήτησης και μεταβλητής γειτονιάς αναζήτησης επιστρέφουν τις μεγαλύτερες λύσεις, 557 και 553, αντίστοιχα.

Στο Σχήμα 4.7 παρουσιάζεται το διάγραμμα με τους χρόνους εκτέλεσης των υλοποιημένων αλγορίθμων για το πρόβλημα αυτό με τη μέση πολυπλοκότητα. Και στο διάγραμμα αυτό οι τιμές των χρόνων έχουν μονάδα μέτρησης τα δευτερόλεπτα.

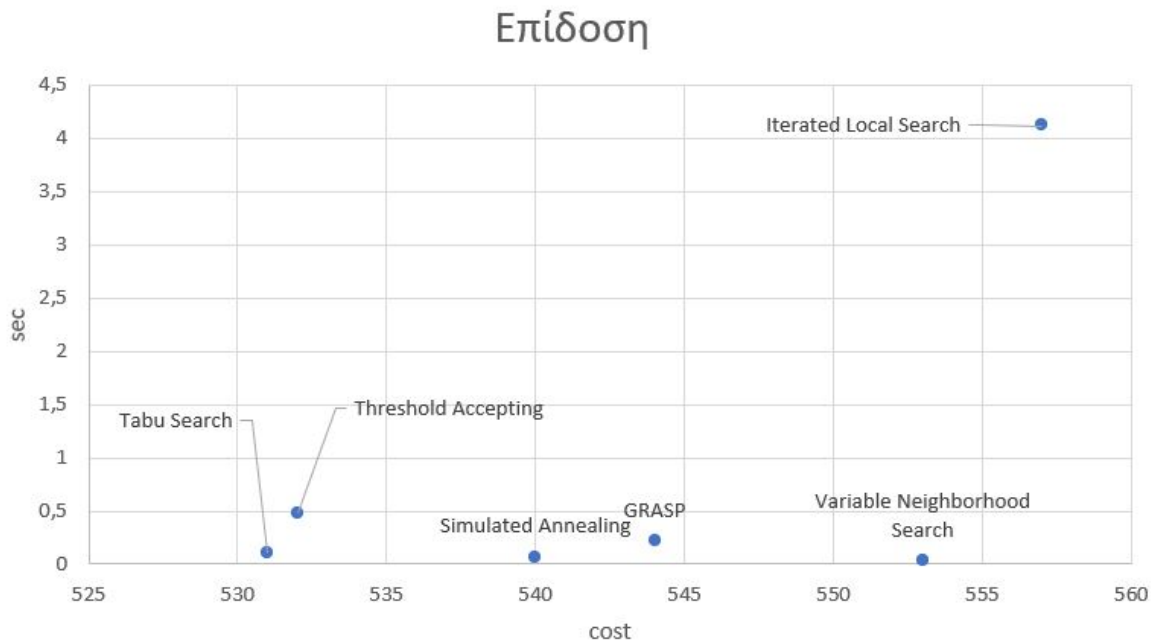
Σχήμα 4.7: Διάγραμμα χρόνων εκτέλεσης - gap6.txt



Παρατηρούμε ότι στο πρόβλημα αυτό η μέθοδος μεταβλητής γειτονιάς αναζήτησης, χρειάστηκε τον ελάχιστο χρόνο εκτέλεσης, ενώ η μέθοδος επαναληπτικής τοπικής αναζήτησης χρειάστηκε πολύ περισσότερο χρόνο για να ολοκληρώσει τη διαδικασία ανάθεσης και εύρεσης του ελαχίστου κόστους. Οι χρόνοι εκτέλεσης των υπόλοιπων μεθόδων είναι κανονικοί για την εύρεση λύσης ενός προβλήματος αυτού του μεγέθους και πολυπλοκότητας.

Το διάγραμμα διασποράς των επιδόσεων των μεθόδων με βάση τη λύση και τον χρόνο εκτέλεσης παρουσιάζεται στο Σχήμα 4.8.

Σχήμα 4.8: Διάγραμμα επιδόσεων αλγορίθμων - gap6.txt

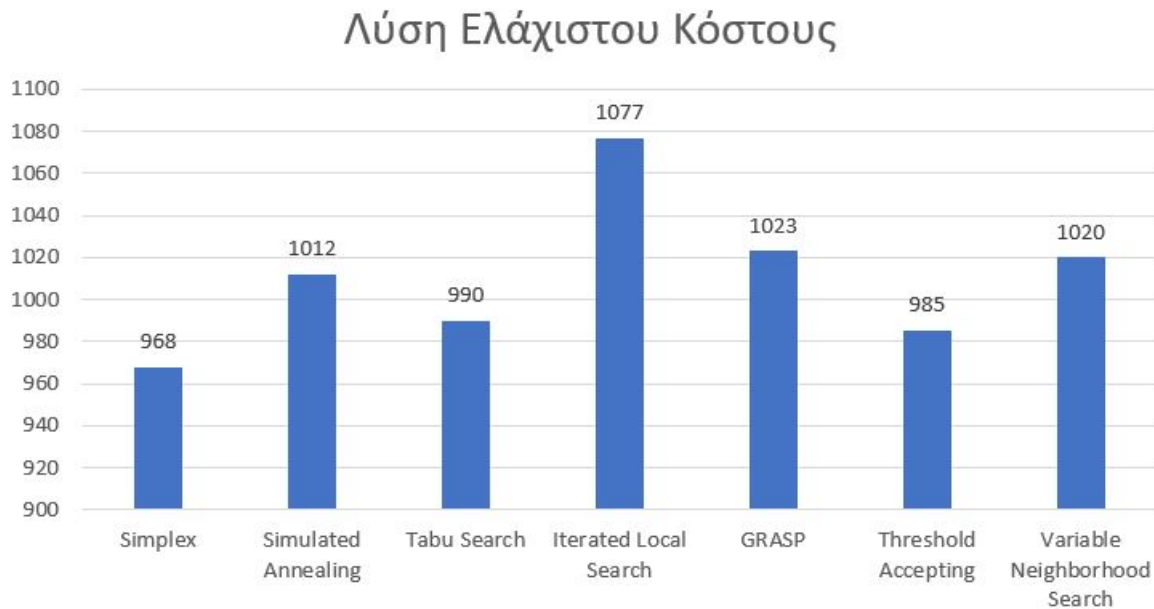


Και σε αυτή τη περίπτωση, αποδοτικότερες μέθοδοι θεωρούνται αυτές που έχουν το ελάχιστο κόστος και με ελάχιστο χρόνο εκτέλεσης, δηλαδή τα σημεία που βρίσκονται χαμηλά και αριστερά βάση των αξόνων. Όπως είναι φανερό, η μέθοδος περιορισμένης αναζήτησης θεωρείται αποδοτικότερη σύμφωνα με τη λύση και τον χρόνο εκτέλεσης. Αντίθετα, η μέθοδος επαναληπτικής τοπικής αναζήτησης δίνοντας μεγαλύτερο κόστος ανάθεσης αλλά και όντας πιο αργή δε θεωρείται αποδοτική σε σχέση με τις υπόλοιπες μεθόδους.

Τέλος, θα αποτυπωθούν τα αποτελέσματα των μεθόδων για μεγαλύτερο και πολυπλοκότερο πρόβλημα. Πιο συγκεκριμένα, δίνονται παρακάτω οι λύσεις και οι χρόνοι εκτέλεσης των αλγορίθμων βάση του πρώτου προβλήματος που είναι δηλωμένο στο αρχείο gap12.txt. Τα προβλήματα στο αρχείο gap12.txt, έχουν 10 πράκτορες και 60 διεργασίες προς ανάθεση, και είναι τα μεγαλύτερα προβλήματα της βιβλιοθήκης αυτής.

Στο Σχήμα 4.9 δίνονται τα κόστη ανάθεσης που επιστρέφει η κάθε μέθοδος ως λύση και γίνεται σύγκριση με βάση το βέλτιστο ελάχιστο κόστος που βρέθηκε μέσω της μεθόδου Simplex.

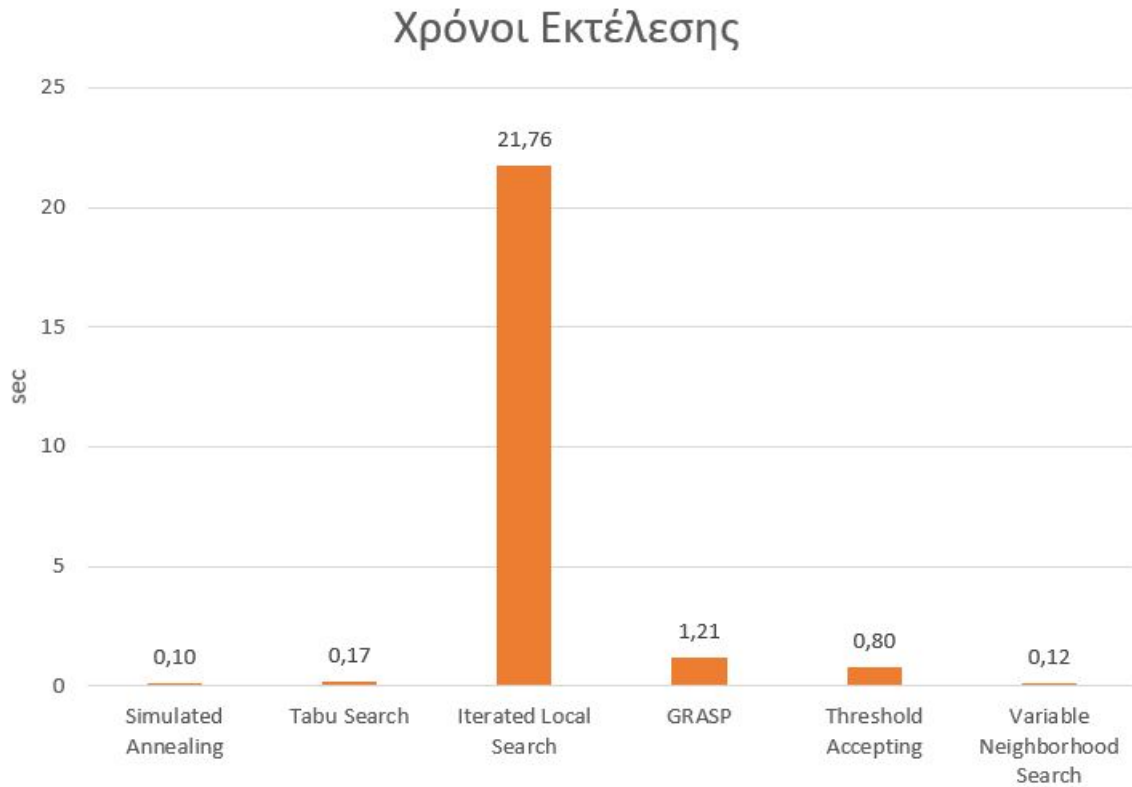
Σχήμα 4.9: Διάγραμμα ελάχιστου κόστους - gap12.txt



Το ελάχιστο κόστος ανάθεσης που υπολογίστηκε για το πρόβλημα αυτό με τη μέθοδο Simplex είναι 968. Παρατηρούμε ότι οι λύσεις των μεθόδων αποδοχής κατωφλιού και περιορισμένης αναζήτησης συγκλίνουν στη βέλτιστη λύση, ενώ η μέθοδος επαναληπτικής αναζήτησης και πάλι επιστρέφει το μεγαλύτερο κόστος.

Στο Σχήμα 4.10 δίνονται οι χρόνοι εκτέλεσης που χρειάστηκαν οι αλγόριθμοι για να υπολογίσουν τη λύση του παραπάνω προβλήματος.

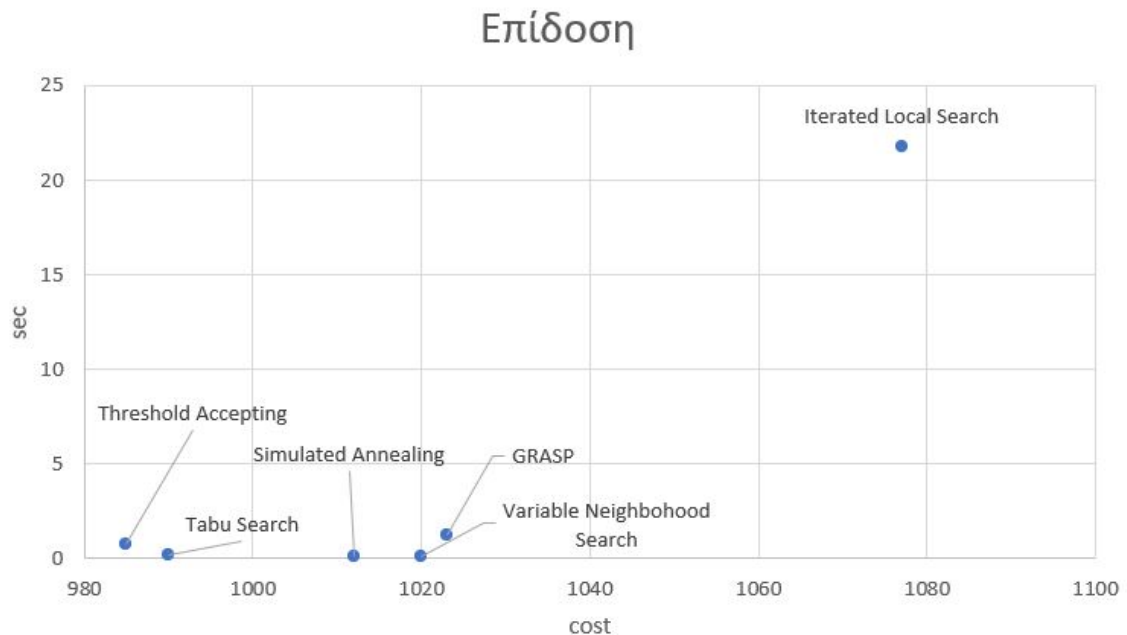
Σχήμα 4.10: Διάγραμμα χρόνων εκτέλεσης - gap12.txt



Όπως φαίνεται από το παραπάνω διάγραμμα, οι μέθοδοι προσομοιωμένης απόψησης, μεταβλητής γειτονιάς αναζήτησης και περιορισμένης αναζήτησης έχουν τους χαμηλότερους χρόνους εκτέλεσης. Αντίθετα, η μέθοδος επαναληπτικής τοπικής αναζήτησης είναι αρκετά αργή στον τερματισμό και εύρεση λύσης από τις υπόλοιπες μεθόδους.

Για να βρούμε την αποδοτικότερη μέθοδο για το πρόβλημα του αρχείου gap12.txt, δίνεται το διάγραμμα επίδοσης στο Σχήμα 4.11.

Σχήμα 4.11: Διάγραμμα επιδόσεων αλγορίθμων - gap12.txt



Από το διάγραμμα αυτό, βρίσκουμε τη μέθοδο που είναι πιο αποδοτική για το πρόβλημα αυτό, όπου όπως προαναφέρθηκε, για να θεωρηθεί αποδοτικότερη μία μέθοδος πρέπει η λύση κόστους ανάθεσης που επιστρέφει και ο χρόνος επίλυσης να είναι μικρός. Δηλαδή, το αντίστοιχο σημείο να βρίσκεται χαμηλά και αριστερά σε σχέση με τους άξονες χρόνου – κόστους. Παρατηρούμε ότι η μέθοδος αποδοχής κατωφλιού και η μέθοδος περιορισμένης αναζήτησης είναι αποδοτικότερες σε σχέση με τις υπόλοιπες μεθόδους. Από την άλλη, το σημείο της μεθόδου επαναληπτικής τοπικής αναζήτησης βρίσκεται πάνω και δεξιά στο διάγραμμα, οπότε το κόστος ανάθεσης είναι μεγαλύτερο και ο χρόνος εκτέλεσης επίσης μεγάλος.

Στους Πίνακες 4.2 και 4.3 παρουσιάζονται οι μέσοι όροι των αποτελεσμάτων των αλγορίθμων καθώς και οι τιμές εκτέλεσης τους ανά αρχείο προβλημάτων. Τα αποτελέσματα, οι αναθέσεις και οι χρόνοι εκτέλεσης όλων των προβλημάτων δίνονται στο Παράρτημα.

Πίνακας 4.2: Αποτελέσματα μεθόδων

Πρόβλημα	SIMULATED ANNEALING		TABU SEARCH		ITERATED LOCAL SEARCH	
	Κόστος	Χρόνος	Κόστος	Χρόνος	Κόστος	Χρόνος
gap1	266.705	0.041085	289.4435	0.062117	284.5627	0.613661
gap2	275.8453	0.034053	278.4209	0.060241	281.6432	1.093932
gap3	433.7294	0.045834	431.3271	0.076933	446.0015	1.978021
gap4	430.0582	0.045782	423.9151	0.082068	450.2733	3.307873
gap5	400.9793	0.047333	398.6631	0.074707	413.238	1.66508
gap6	532.9307	0.056824	530.9953	0.09456	547.556	3.608275
gap7	676.0895	0.062081	662.3278	0.109308	662.4232	6.672796
gap8	840.9098	0.071532	841.2932	0.130101	1013.517	11.50701
gap9	509.3353	0.055853	500.5212	0.088057	533.1535	2.903018
gap10	664.2792	0.084493	656.3292	0.137152	711.5438	6.976616
gap11	615.8635	0.085046	599.9147	0.1539	669.3251	16.07032
gap12	987.9804	0.103882	972.1934	0.166008	1046.259	21.39888
M.O	552.8921	0.0612	548.7787	0.1029	588.2913	6.4829

Πίνακας 4.3: Αποτελέσματα μεθόδων

Πρόβλημα	GRASP		THRESHOLD ACCEPTING		VARIABLE NEIGHBORHOOD SEARCH	
	Κόστος	Χρόνος	Κόστος	Χρόνος	Κόστος	Χρόνος
gap1	273.9721	0.040341	262.2	0.345934	260.3631	0.016353
gap2	280.1877	0.057079	269.3312	0.318738	278.3413	0.018032
gap3	439.5281	0.112224	429.8657	0.38954	439.1243	0.020136
gap4	439.1053	0.177905	422.3882	0.4188	420.4602	0.025
gap5	401.0599	0.112514	393.1325	0.404763	406.0233	0.026645
gap6	539.2823	0.228022	525.7982	0.4844	543.5284	0.03826564
gap7	683.8286	0.416398	660.5297	0.569595	681.5587	0.049803
gap8	866.4628	0.727809	807.7788	0.684227	829.1325	0.071932
gap9	509.4283	0.20929	494.7217	0.495746	517.3845	0.039515
gap10	669.6455	0.475381	652.8615	0.674111	677.2344	0.080252
gap11	626.3665	0.765771	597.1813	0.758349	640.2695	0.101964
gap12	999.5864	1.209364	967.1279	0.803113	980.0735	0.11509
M.O	560.7044	0.377675	540.2430	0.528943	556.1245	0.050249

Γενικά, κάνοντας μία σύγκριση των αποτελεσμάτων των προβλημάτων που επιλύθηκαν, μπορούμε να εξάγουμε κάποια συμπεράσματα για τις μεθόδους σε σχέση με το μέγεθος το προβλήματος που επιλύουν.

Αρχικά, καταβαίνουμε ότι οι λύσεις, ή αλλιώς κόστη ανάθεσης, όσο μεγαλώνουν τα προβλήματα και γίνονται πιο σύνθετα και περίπλοκα ως προς την ανάθεση των διεργασιών στους πράκτορες, ανάλογα μεγαλώνει και η διαφορά ανάμεσα στο πραγματικό ελάχιστο κόστος και της λύσης ανάθεσης των μεθόδων. Επιπλέον, όπως φαίνεται και από το μέγεθος των προβλημάτων, όσο μεγαλύτερο είναι το πρόβλημα, είναι πιο περίπλοκο στην αναζήτηση βέλτιστης ανάθεσης και άρα ο χρόνος εκτέλεσης της κάθε μεθόδου είναι μεγαλύτερος.

Επίσης, από τους μέσους όρους των Πινάκων 4.2 και 4.3 παρατηρούμε ότι η βέλτιστης μέθοδος γενικά είναι η μέθοδος Αποδοχής Κατωφλιού και η μέθοδος Προσομοιωμένης Ανόπτωσης. Η μέθοδος Αποδοχής Κατωφλιού συγκλίνει περισσότερο στην βέλτιστη λύση, ενώ η μέθοδος Προσομοιωμένης Ανόπτωσης επιστρέφει αποδεκτά αποτελέσματα και με μικρότερο χρόνο εκτέλεσης κατά μέσο όρο.

Στο επόμενο κεφάλαιο δίνονται αναλυτικά περισσότερα συμπεράσματα σχετικά με τις επιδόσεις των μεθόδων που υλοποιήθηκαν.

Κεφάλαιο 5

Συμπεράσματα και μελλοντικές επεκτάσεις

Στο κεφάλαιο αυτό θα παρουσιαστούν τα τελικά συμπεράσματα για τις μεθευρετικές μεθόδους που υλοποιήθηκαν στην εργασία αυτή. Βάση των προβλημάτων που επιλύθηκαν και αναλύθηκαν στο προηγούμενο κεφάλαιο, συγκρίνονται και αξιολογούνται οι μέθοδοι. Στη συνέχεια, αναφέρονται μελλοντικές επεκτάσεις της παρούσας εργασίας.

5.1 Συμπεράσματα

Με την εξαγωγή των αποτελεσμάτων των προβλημάτων που αναλύθηκαν στο προηγούμενο κεφάλαιο, μπορούμε να εξάγουμε συμπεράσματα σχετικά με τους αλγόριθμους των μεθόδων που υλοποιήθηκαν στην παρούσα εργασία.

Οι αλγόριθμοι που υλοποιήθηκαν είναι μεθευρετικοί και τα αποτελέσματά τους προσεγγίζουν τη βέλτιστη λύση. Όμως, επειδή εμπεριέχουν στοιχεία τυχαιότητας, σημαντικό είναι να εφαρμοστούν και να επιλύσουν πολλές φορές μια μεγάλη ποικιλία προβλημάτων. Με τον τρόπο αυτό, τα αποτελέσματα και τα συμπεράσματα είναι πιο σαφή και τεκμηριωμένα αντίστοιχα. Ένα παράδειγμα στοιχείου τυχαιότητας που περιέχεται στις ευρετικές μεθόδους των αλγόριθμων είναι η τυχαία επιλογή αλλαγής ανάθεσης διεργασίας για νέα αλλαγή αναθέσεων και σύγκριση του νέου κόστους που προκύπτει.

Τα συμπεράσματα που μπορούν να προκύψουν για τις υλοποιημένες μεθόδους αναζήτησης της λύσης του γενικευμένου προβλήματος ανάθεσης πόρων μπορούν να χωριστούν σε δύο κατηγορίες. Η πρώτη αφορά τη λύση της εκάστοτε μεθόδου και

πόσο κοντά είναι στην πραγματική βέλτιστη λύση, ενώ η δεύτερη κατηγορία αφορά την πολυπλοκότητα και τον χρόνο που χρειάζεται η κάθε μέθοδος να βρει τη λύση στο πρόβλημα προς επίλυση.

Τα συμπεράσματα που παρουσιάζονται παρακάτω ανά μέθοδο αφορούν όλα τα προβλήματα της βιβλιοθήκης προβλημάτων ανάθεσης πόρων OR-Library [21] και αφορούν προβλήματα ελαχιστοποίησης κόστους. Τα σχετικά διαγράμματα του προηγούμενου κεφαλαίου, μπορούν να βοηθήσουν στην απεικόνιση των επιδόσεων των υλοποιημένων μεθόδων.

Η μέθοδος αποδοχής κατωφλιού επιστρέφει λύση των προβλημάτων που συγκλίνει αρκετά με την πραγματική βέλτιστη λύση, χωρίς να παίζει ρόλο το μέγεθος και η πολυπλοκότητα του προβλήματος. Όμως σε σύγκριση με τις άλλες μεθόδους, ο χρόνος εκτέλεσης και εύρεσης της λύσης δεν είναι ο βέλτιστος, καθώς υπάρχουν και άλλοι μέθοδοι με μικρότερους χρόνους εκτέλεσης.

Επιπλέον και η μέθοδος περιορισμένης αναζήτησης προσφέρει αποτελέσματα που συγκλίνουν με την πραγματική βέλτιστη λύση ανάθεσης. Στην περίπτωση της μεθόδου αυτής όμως, όσο μεγαλώνουν τα προβλήματα, παρατηρούμε ότι η μέθοδος αυτή γίνεται αποδοτικότερη ως προς το επιστρεφόμενο κόστος και τον χρόνο εκτέλεσης. Γενικά, η μέθοδος αυτή μπορεί να θεωρηθεί γρήγορη. Υπάρχουν όμως και άλλες μεθόδους με μικρότερους χρόνους εκτέλεσης στην επίλυση των ίδιων προβλημάτων.

Μία άλλη μέθοδος που επιλύει και επιστρέφει λύση πολύ κοντά στην πραγματική, είναι η μέθοδος προσομοιωμένης ανόπτωσης. Στην περίπτωση αυτής της μεθόδου τα αποτελέσματα είναι αρκετά αποδεκτά, όμως η μέση τιμή του κόστους που αποδίδει, είναι μεγαλύτερη από αυτές των προαναφερόμενων μεθόδων. Παρ' όλα αυτά, είναι μία μέθοδος με χαμηλό χρόνο εκτέλεσης, ακόμα και σε μεγάλης πολυπλοκότητας προβλήματα.

Η μέθοδος άπληστης τυχαιοποιημένης τοπικής αναζήτησης προσφέρει λύσεις όπου το κόστος ανάθεσης είναι μεγαλύτερο από το κόστος που επιστρέφουν οι παραπάνω μέθοδοι. Επίσης, ο χρόνος εκτέλεσης της μεθόδου αυξάνεται σε σχέση με των άλλων αλγορίθμων όσο αυξάνεται και η πολυπλοκότητα των προβλημάτων.

Τα συμπεράσματα της παραπάνω μεθόδου μοιάζουν με τα συμπεράσματα που προκύπτουν για τη μέθοδο μεταβλητής γειτονιάς αναζήτησης. Η διαφορά είναι ότι

η μέθοδος αυτή είναι γρηγορότερη, σε οποιοδήποτε μεγέθους προβλήματα. Αυτό μπορεί να εξηγηθεί στο ότι δε ξεκινάει από τελείως μακρινή αρχική κατάσταση, αλλά με μία αρχική ανάθεση των διεργασιών στους πράκτορες.

Τέλος, η μέθοδος επαναληπτικής τοπικής αναζήτησης σε οποιοδήποτε μεγέθους προβλήματα επιστρέφει κόστος μεγαλύτερο από τις άλλες μεθόδους. Επιπλέον, επειδή πρόκειται για μία πολύπλοκη μέθοδο, ο χρόνος εκτέλεσης της είναι αρκετά μεγάλος ακόμα και σε μικρά προβλήματα. Πιο γενικά, η μέθοδος αυτή δεν μπορεί να θεωρηθεί αρκετά αποδοτική σε σχέση με τα προβλήματα ανάθεσης πόρων που επιλύθηκαν.

5.2 Μελλοντικές επεκτάσεις

Η παρούσα εργασία μπορεί να επεκταθεί σε διάφορους τομείς. Οι τομείς αυτοί αφορούν νέους αλγόριθμους, την τεχνική επίλυση των μεθόδων και τα προβλήματα που επιλύονται.

Σε σχέση με τους άλλους αλγόριθμους, μπορούν να μελετηθούν επιπλέον εξελιγμένοι αλγόριθμοι όπως αυτοί που ερευνήθηκαν από τους Shmoys και Tardos [25], των Jörnsten και Värbrand [26] ή των Tasgetiren et al. [27]. Στόχος της παρούσας εργασίας ήταν να μελετηθούν βασικοί μεθευρετικοί μέθοδοι, όμως η εργασία αυτή, μπορεί να επεκταθεί και να υλοποιηθούν αλγόριθμοι που έχουν ερευνηθεί σε ερευνητικό πλαίσιο για το γενικευμένο πρόβλημα ανάθεσης πόρων.

Στον τομέα της τεχνικής επίλυσης, ως μελλοντική επέκταση μπορούν να μελετηθούν καλύτερα δομές δεδομένων σε προγραμματιστικό επίπεδο. Αυτό έχει ως αποτέλεσμα οι μέθοδοι να έχουν μικρότερους χρόνους εκτέλεσης, και να είναι ανάλογοι της πολυπλοκότητας της εκάστοτε μεθόδου.

Τέλος, στην εργασία αυτή μελετήθηκαν προβλήματα από μία συγκεκριμένη βιβλιοθήκη προβλημάτων ανάθεσης πόρων. Όμως, θα μπορούσε να επεκταθεί η εργασία και να μπορούν να εισάγονται τα δεδομένα και από άλλες βιβλιοθήκες με διαφορετικά προβλήματα.

Παραρτήματα

Πίνακας 1: Αποτελέσματα προβλημάτων αρχείου gap1.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	261	3 2 3 5 4 4 4 3 1 4 2 5 5 2 1	0.063000
	2	287	3 1 1 3 2 4 3 2 4 4 3 1 5 2 5	0.047000
	3	256	4 3 2 1 2 1 2 4 4 5 4 1 3 3 5	0.046000
	4	274	2 3 3 3 2 5 1 4 2 3 4 5 5 1 4	0.047000
	5	251	3 5 4 3 4 5 3 3 2 1 1 5 2 1 4	0.047000
Περιορισμένη Αναζήτηση	1	261	5 2 3 5 4 4 4 3 1 4 2 5 3 2 1	0.063000
	2	270	3 5 1 3 4 5 3 1 5 2 3 1 2 4 2	0.078000
	3	256	4 3 2 1 2 1 2 4 4 5 4 1 3 3 5	0.062000
	4	280	2 3 3 3 1 5 4 4 2 3 2 5 5 1 4	0.078000
	5	255	5 5 4 3 4 2 3 3 3 1 1 5 2 1 4	0.063000
Επαναληπτική Τοπική Αναζήτηση	1	271	5 2 3 5 4 4 4 2 1 4 2 5 3 3 1	0.703000
	2	279	3 1 1 3 4 5 3 2 5 4 3 1 2 5 2	0.688000
	3	265	4 3 5 1 2 1 2 4 2 2 4 5 3 3 1	0.671000
	4	281	2 3 3 4 1 5 2 4 2 3 3 5 5 1 4	0.672000
	5	255	5 5 4 4 4 3 3 3 2 1 4 5 2 1 1	0.657000
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	274	1 2 3 5 2 4 4 3 1 1 2 5 5 4 3	0.046000
	2	374	3 5 5 3 3 4 1 2 5 1 3 2 4 4 2	0.032000
	3	256	4 3 2 1 2 1 2 4 4 5 4 1 3 3 5	0.047000
	4	274	2 3 3 3 2 5 1 4 2 3 4 1 5 5 4	0.031000
	5	255	5 5 4 3 4 2 3 3 3 1 1 5 2 1 4	0.047000
Αποδοχής Κατωφλιού	1	261	5 2 3 5 4 3 4 3 1 4 2 5 3 2 1	0.375000
	2	269	3 1 1 3 4 5 3 2 5 2 3 1 5 4 2	0.390000
	3	256	4 3 1 2 5 1 2 4 4 2 4 1 3 3 5	0.391000
	4	274	2 3 3 3 2 5 1 4 2 3 4 1 5 5 4	0.375000
	5	251	3 5 4 3 4 5 3 3 2 1 1 5 2 1 4	0.391000
Μεταβλητής Γειτονιάς Αναζήτησης	1	273	1 2 3 5 2 4 4 3 1 4 2 5 5 1 3	0.015000
	2	269	3 1 1 3 4 5 3 2 5 5 3 1 2 4 2	0.016000
	3	268	4 5 5 2 2 3 1 4 1 3 4 1 3 3 5	0.031000
	4	275	2 3 3 5 2 5 1 4 2 3 4 1 5 5 4	0.016000
	5	257	3 5 1 4 4 5 3 3 2 1 1 5 2 3 4	0.015000

Πίνακας 2: Αποτελέσματα προβλημάτων αρχείου gap2.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	295	5 4 2 3 5 3 5 3 4 5 4 1 1 1 2 2 4 1 3 2	0.046000
	2	272	2 5 1 5 3 2 2 2 4 2 5 3 4 1 1 1 4 3 5 5	0.032000
	3	260	2 1 3 1 5 2 3 5 4 3 5 5 1 2 4 2 1 3 5 4	0.031000
	4	273	1 3 3 3 3 3 5 1 5 4 2 5 4 4 2 2 1 1 2 5	0.031000
	5	278	4 3 3 2 1 3 2 1 5 2 1 2 5 5 1 1 3 4 5 5	0.031000
Περιορισμένη Αναζήτηση	1	302	4 4 1 3 2 3 5 3 4 4 2 5 1 2 2 1 4 1 3 5	0.063000
	2	271	2 4 4 1 3 2 2 2 1 2 5 3 4 1 4 1 5 3 5 5	0.062000
	3	260	2 1 3 1 5 2 3 5 4 3 5 5 1 2 4 2 1 3 5 4	0.063000
	4	273	1 3 3 1 3 3 5 1 5 4 2 5 4 4 2 2 4 1 2 5	0.062000
	5	278	4 2 3 5 1 3 4 1 5 2 1 2 5 5 1 1 2 4 5 3	0.063000
Επαναληπτική Τοπική Αναζήτηση	1	298	5 4 4 2 2 3 5 3 4 4 5 1 1 1 2 3 5 1 2 3	1.062000
	2	275	2 5 1 5 3 2 2 2 5 2 5 3 4 1 1 1 4 3 4 5	1.500000
	3	280	2 2 4 1 5 5 3 3 4 3 2 4 1 2 1 4 1 3 5 5	1.016000
	4	280	5 3 3 2 3 1 4 1 5 4 2 5 4 4 2 5 1 1 2 3	0.984000
	5	267	4 2 3 1 1 3 2 4 5 2 1 2 5 5 4 1 3 3 5 5	0.985000
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	292	5 1 2 3 5 3 1 3 4 4 2 5 1 2 2 4 4 5 2 3	0.062000
	2	301	2 5 1 3 3 2 4 2 5 2 5 2 4 2 1 1 1 4 5 3	0.047000
	3	260	2 1 3 1 5 2 3 5 4 3 5 5 1 2 4 2 1 3 5 4	0.047000
	4	275	1 3 3 3 3 3 5 4 5 5 2 3 4 4 2 2 1 1 2 1	0.062000
	5	280	4 5 3 2 1 3 2 4 5 2 1 2 5 5 1 1 3 3 5 1	0.047000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Αποδοχής Κατωφλιού	1	282	5 4 1 3 5 3 5 3 4 5 2 1 1 2 2 4 4 1 3 2	0.328000
	2	272	2 5 1 5 3 2 2 2 4 2 5 3 4 1 1 1 4 3 5 5	0.313000
	3	260	2 1 3 1 5 2 3 5 4 3 5 5 1 2 4 2 1 3 5 4	0.312000
	4	269	1 3 3 2 3 1 5 1 5 4 5 3 4 4 3 2 4 1 2 5	0.328000
	5	267	4 2 3 1 1 3 2 4 5 2 1 2 5 5 4 1 3 3 5 5	0.313000
Μεταβλητής Γειτονιάς Αναζήτησης	1	294	1 4 1 3 5 3 5 3 2 4 5 2 1 2 2 4 5 1 2 3	0.031000
	2	274	2 4 1 3 4 2 2 2 5 5 5 3 3 1 1 1 4 3 5 5	0.000000
	3	271	2 1 4 1 5 2 3 5 4 3 5 4 1 2 4 2 1 3 5 3	0.016000
	4	232	4 3 3 3 3 1 4 1 1 4 2 5 4 4 2 2 1 5 2 3	0.031000
	5	277	1 5 3 2 1 3 4 4 5 2 1 2 5 5 4 1 3 4 5 1	0.000000

Πίνακας 3: Αποτελέσματα προβλημάτων αρχείου gap3.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	446	3 4 2 1 2 2 5 4 5 4 1 1 5 2 3 1 2 5 3 3 5 1 2 3 4	0.062000
	2	417	4 3 2 3 1 4 4 2 5 2 4 1 2 3 5 5 3 2 2 4 2 3 5 1 1	0.032000
	3	451	3 4 4 1 5 4 4 5 2 5 3 4 2 4 1 3 1 3 2 1 3 1 5 2 2	0.031000
	4	436	2 4 2 1 5 2 5 1 1 2 5 3 3 5 2 4 4 4 4 3 3 5 1 3 1	0.047000
	5	412	2 3 1 4 3 3 4 3 1 4 3 2 4 5 1 2 4 3 5 2 1 4 5 2 5	0.031000
Περιορισμένη Αναζήτηση	1	449	3 4 2 2 2 2 3 4 5 4 1 1 2 1 5 1 2 5 3 4 4 3 3 5 1	0.078000
	2	415	4 3 2 3 1 4 4 2 5 3 2 1 2 4 5 5 3 2 2 4 2 3 5 1 1	0.063000
	3	453	3 1 4 2 5 4 4 1 2 5 3 4 2 4 1 3 1 3 2 5 4 1 3 5 3	0.078000
	4	433	1 4 2 1 3 5 5 1 4 2 5 3 3 5 2 4 4 2 1 5 3 5 1 3 4	0.078000
	5	416	2 3 1 4 3 3 4 3 1 4 3 2 4 5 1 2 4 3 5 5 5 4 1 2 5	0.078000
Επαναληπτική Τοπική Αναζήτηση	1	488	5 4 1 3 5 1 3 4 5 1 4 1 1 4 3 1 2 2 3 5 4 3 2 5 2	2.703000
	2	433	4 3 2 3 1 4 3 5 2 1 2 1 2 5 5 2 3 4 2 4 4 3 5 1 5	2.266000
	3	458	2 1 4 1 5 5 4 1 2 5 3 4 3 2 3 3 1 3 3 5 4 1 5 2 4	1.844000
	4	463	5 4 5 1 4 5 5 1 1 3 5 3 2 1 2 1 4 3 4 5 4 1 2 3 2	1.859000
	5	413	2 3 1 4 3 3 4 3 1 4 3 2 4 4 1 5 4 3 1 2 1 2 5 2 5	1.828000
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	455	3 4 3 5 5 2 3 4 5 3 1 1 2 1 5 1 2 4 3 4 5 1 2 3 2	0.109000
	2	427	2 3 2 3 1 4 3 5 4 1 2 1 1 3 5 5 3 4 2 4 2 2 5 1 5	0.110000
	3	454	3 1 4 2 3 4 4 2 2 5 5 4 1 4 1 3 1 3 2 5 3 1 5 5 2	0.109000
	4	441	5 4 2 1 2 5 5 1 4 4 4 3 1 1 3 4 4 2 1 3 3 5 1 3 5	0.094000
	5	412	2 3 1 4 3 3 4 3 1 4 3 2 4 5 1 2 4 3 5 2 1 4 5 2 5	0.109000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Αποδοχής Κατωφλιού	1	442	3 4 2 1 5 2 3 4 5 4 1 1 2 2 3 1 2 5 3 4 4 3 2 5 1	0.360000
	2	420	5 3 4 3 1 4 4 5 1 2 2 1 2 4 5 5 3 2 2 4 2 3 5 1 3	0.375000
	3	450	3 4 4 1 5 5 4 2 2 5 3 4 1 2 3 3 1 3 2 1 4 4 5 5 2	0.359000
	4	431	1 4 2 1 5 2 5 1 4 2 5 3 3 5 2 4 4 2 1 3 3 5 1 3 4	0.359000
	5	411	2 3 1 4 3 3 4 3 1 4 3 2 4 2 1 5 4 3 1 2 1 4 5 2 5	0.375000
Μεταβλητής Γειτονιάς Αναζήτησης	1	442	3 2 2 1 2 2 3 4 5 4 1 3 2 1 5 1 2 5 3 1 4 3 2 5 4	0.016000
	2	425	4 3 4 3 1 2 4 2 5 1 4 1 2 5 5 4 3 3 2 4 2 1 5 1 3	0.031000
	3	453	3 4 4 2 5 5 3 1 2 5 3 4 1 2 1 3 1 3 1 5 4 4 5 2 3	0.016000
	4	462	5 4 5 1 3 2 1 1 3 4 5 2 3 2 2 1 4 4 4 3 1 5 1 3 2	0.016000
	5	426	3 3 1 5 3 3 4 3 1 2 5 2 2 2 1 2 4 3 5 4 1 4 5 2 5	0.015000

Πίνακας 4: Αποτελέσματα προβλημάτων αρχείου gar4.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	443	2 5 5 4 1 1 2 5 4 3 4 4 3 1 5 4 4 4 2 3 3 2 1 5 5 3 2 2 5 5	0.047000
	2	439	1 1 4 3 3 4 5 4 3 2 1 4 1 4 2 2 4 4 2 2 4 3 1 1 5 5 5 5 2 1	0.047000
	3	442	3 4 1 5 2 1 4 5 2 3 3 1 4 3 4 3 1 2 2 3 1 5 2 2 4 1 5 3 5 1	0.047000
	4	399	1 5 5 4 1 3 1 2 4 4 3 2 3 3 2 4 1 4 5 2 4 4 1 3 5 2 3 5 1 2	0.031000
	5	416	4 5 4 2 3 5 3 4 1 1 2 3 4 5 4 2 1 5 5 4 2 1 1 5 5 3 3 2 4 1	0.047000
Περιορισμένη Αναζήτηση	1	446	2 5 5 3 1 1 4 5 4 3 2 4 1 2 5 4 4 4 2 3 3 2 5 5 2 3 1 2 5 1	0.078000
	2	436	1 4 4 3 5 4 2 5 3 5 1 4 1 4 4 2 4 3 5 1 2 5 1 1 5 4 3 2 2 1	0.094000
	3	428	3 4 1 5 2 1 4 3 2 3 3 5 4 3 3 2 4 4 1 3 1 5 2 1 4 1 5 2 3 1	0.078000
	4	407	5 4 3 4 1 3 2 5 4 5 4 1 4 3 2 4 3 2 5 3 4 1 1 2 5 2 3 5 1 2	0.078000
	5	427	2 5 4 2 3 4 3 4 5 4 1 3 4 5 2 1 2 3 5 1 2 3 1 4 5 5 3 2 4 1	0.094000
Επαναληπτική Τοπική Αναζήτηση	1	452	2 5 5 4 1 2 5 1 4 3 4 4 3 1 5 4 4 5 2 3 3 2 2 5 2 3 1 4 5 1	3.375000
	2	458	1 5 3 3 3 4 5 2 3 2 1 1 2 4 2 4 2 4 4 1 4 3 1 1 5 5 5 2 2 1	3.078000
	3	468	4 5 1 2 2 1 4 3 4 3 2 5 4 3 3 3 1 1 1 3 5 3 2 4 4 2 5 3 2 1	3.078000
	4	420	5 1 3 4 1 2 1 2 4 4 3 5 4 3 5 4 1 2 5 3 4 4 5 3 5 4 1 5 2 2	3.078000
	5	445	2 5 4 2 3 4 5 4 4 1 1 1 4 5 4 2 2 3 5 4 3 1 1 3 5 2 3 2 5 1	3.110000
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	471	2 2 4 3 1 3 2 5 4 1 3 4 2 2 5 1 5 4 4 5 5 2 1 5 5 3 1 4 5 3	0.203000
	2	449	1 5 4 3 5 3 2 5 3 3 1 3 4 4 2 2 4 4 2 1 5 1 1 1 5 4 5 2 2 4	0.172000
	3	448	3 4 1 5 2 1 4 3 2 3 3 1 2 3 3 2 1 5 2 3 1 5 4 2 4 1 5 4 3 1	0.171000
	4	407	4 5 3 4 1 3 2 5 4 4 4 5 3 3 2 4 1 2 5 2 4 1 1 3 5 2 3 5 1 2	0.172000
	5	451	2 5 3 5 3 4 5 4 5 4 4 3 2 4 1 2 1 1 5 3 3 1 1 4 5 2 3 2 2 1	0.172000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Αποδοχής Κατωφλιού	1	438	1 5 5 4 1 3 2 5 4 2 4 4 3 2 5 4 4 4 2 3 1 2 5 5 3 1 1 2 5 1	0.422000
	2	434	5 1 4 4 5 4 2 2 3 5 1 4 1 4 2 5 4 3 4 1 2 3 1 1 5 5 3 2 2 1	0.422000
	3	436	4 4 1 2 2 1 4 3 2 3 3 5 4 3 5 4 1 2 2 3 5 5 5 1 4 1 3 2 3 1	0.422000
	4	397	5 5 3 4 1 3 2 2 4 4 4 5 4 3 2 4 1 2 5 3 4 4 1 3 5 2 1 5 1 2	0.406000
	5	423	4 5 3 2 3 4 5 4 5 1 1 1 4 5 4 5 3 2 2 3 2 1 1 4 5 3 3 2 4 1	0.422000
Μεταβλητής Γειτονιάς Αναζήτησης	1	359	2 5 2 4 1 1 2 5 3 3 3 4 4 5 1 4 4 4 1 5 1 2 2 5 5 3 3 4 5 3	0.031000
	2	455	4 1 4 3 3 5 2 4 1 3 1 4 1 4 2 2 4 3 2 1 5 3 2 1 5 1 5 5 5 4	0.016000
	3	453	3 4 1 5 2 3 4 4 4 3 3 1 2 3 3 2 4 2 1 3 1 5 5 1 4 1 5 2 2 1	0.031000
	4	429	5 2 3 5 1 3 1 2 4 4 3 2 3 1 1 4 2 2 5 5 4 4 1 3 5 4 1 5 4 2	0.016000
	5	436	2 5 4 2 3 4 3 4 1 4 1 1 4 4 2 5 3 3 2 5 3 1 1 5 5 2 3 2 2 1	0.031000

Πίνακας 5: Αποτελέσματα προβλημάτων αρχείου gar5.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	410	8 4 8 1 1 7 2 3 6 5 7 4 6 4 8 5 4 6 2 7 2 3 2 1	0.046000
	2	395	5 1 1 4 6 7 5 7 6 8 3 3 4 4 2 6 6 8 2 2 1 3 7 5	0.032000
	3	385	5 2 1 4 2 8 4 8 7 6 6 6 5 2 1 5 3 3 7 4 7 4 8 3	0.047000
	4	396	4 3 2 5 8 3 1 1 8 4 7 2 3 8 2 2 7 5 6 5 4 3 6 6	0.031000
	5	403	2 8 3 7 5 4 5 4 3 7 1 8 1 2 7 5 8 3 2 6 1 6 4 2	0.062000
Περιορισμένη Αναζήτηση	1	412	4 3 8 1 1 8 7 3 2 4 1 4 6 6 8 5 5 7 5 7 6 3 2 1	0.063000
	2	395	5 1 1 2 3 7 5 6 6 8 3 4 4 8 2 5 1 8 2 4 7 6 7 5	0.062000
	3	387	1 2 8 4 2 8 4 8 7 8 6 6 5 2 1 5 7 3 6 7 7 4 5 3	0.078000
	4	392	4 1 2 4 1 2 1 8 6 4 7 7 3 8 2 2 5 5 2 5 6 3 4 8	0.063000
	5	404	7 5 2 7 5 4 6 5 3 7 8 8 1 2 6 7 8 4 3 1 1 4 4 2	0.078000
Επαναληπτική Τοπική Αναζήτηση	1	430	8 6 8 8 1 7 2 3 1 4 7 4 3 6 1 5 4 6 2 7 2 8 5 2	2.031000
	2	410	5 3 1 4 4 7 5 6 1 8 4 3 4 2 3 6 8 7 1 2 6 3 7 5	1.594000
	3	399	8 6 1 4 2 8 2 8 1 6 7 4 5 2 1 5 2 6 5 7 7 4 5 3	1.547000
	4	409	4 3 2 6 8 5 1 1 3 4 2 2 3 8 7 7 8 5 6 5 7 3 4 6	1.531000
	5	417	2 4 3 7 5 4 2 5 3 8 8 1 6 1 6 5 8 3 2 5 7 6 4 2	1.547000
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	411	8 4 8 1 1 7 2 3 4 5 1 4 2 6 8 5 4 3 2 7 2 3 6 1	0.109000
	2	391	5 5 1 2 6 7 4 6 1 8 3 7 4 8 2 5 6 8 1 4 7 3 3 5	0.125000
	3	394	1 5 1 4 2 6 7 8 3 6 7 5 5 2 1 4 3 6 6 7 8 4 8 2	0.110000
	4	392	4 3 2 8 1 2 1 8 6 4 7 2 3 8 7 2 5 5 6 5 8 3 4 6	0.109000
	5	408	7 1 4 7 5 4 6 5 5 1 7 8 1 2 6 2 8 3 5 3 1 8 4 2	0.110000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Αποδοχής Κατωφλιού	1	406	8 4 8 1 1 3 2 1 4 5 7 8 2 4 6 5 4 7 2 7 6 3 2 1	0.406000
	2	389	5 5 1 4 6 7 4 6 1 8 3 7 4 2 2 5 6 8 1 4 7 3 3 5	0.406000
	3	385	1 2 1 4 2 8 4 8 7 6 6 5 5 2 1 5 3 3 7 4 7 4 8 3	0.406000
	4	388	4 1 2 8 1 2 1 8 8 4 7 2 3 7 6 2 5 5 6 5 7 3 4 6	0.407000
	5	399	7 8 3 7 5 4 5 5 3 7 1 8 6 2 6 2 8 3 4 5 1 6 4 2	0.484000
Μεταβλητής Γειτονιάς Αναζήτησης	1	421	4 4 6 1 1 3 7 5 2 4 3 8 6 5 8 5 5 7 2 7 6 3 2 1	0.031000
	2	397	5 3 1 2 6 7 5 6 1 8 3 3 4 4 2 5 6 8 1 4 2 3 7 5	0.031000
	3	394	2 4 6 4 2 8 7 3 1 5 4 6 5 2 1 8 7 1 6 4 8 2 7 3	0.032000
	4	404	1 3 2 4 8 1 1 8 3 6 7 2 3 8 6 5 5 5 7 4 8 3 4 1	0.031000
	5	405	2 8 1 7 2 4 5 7 2 7 8 8 6 3 6 5 3 4 6 1 1 4 4 2	0.031000

Πίνακας 6: Αποτελέσματα προβλημάτων αρχείου gar6.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	536	7 5 4 3 1 3 1 2 5 4 4 4 8 5 3 5 6 3 6 7 5 2 4 7 2 1 2 8 8 1 6 6	0.078000
	2	549	1 1 4 4 8 8 7 2 8 5 3 3 6 3 5 5 7 2 1 6 3 2 7 6 7 6 8 8 7 1 4 4	0.063000
	3	529	3 1 4 8 1 5 5 3 2 8 4 5 8 7 2 4 6 7 1 4 3 6 7 2 1 8 6 6 2 8 7 7	0.078000
	4	523	6 6 4 1 7 1 3 8 3 7 1 2 2 4 2 5 6 3 5 3 1 2 8 1 3 8 7 3 6 7 4 3	0.047000
	5	529	1 5 6 4 6 7 1 1 3 8 3 1 5 1 2 7 4 5 8 8 4 7 3 8 6 4 2 2 5 2 3 6	0.062000
Περιορισμένη Αναζήτηση	1	531	7 5 4 3 1 3 7 2 7 7 4 4 8 5 3 5 6 6 6 4 5 2 4 7 2 1 2 8 8 1 3 5	0.125000
	2	541	6 4 5 2 8 6 7 1 8 5 1 3 4 1 3 7 8 4 1 5 3 2 7 6 7 3 8 8 7 4 6 4	0.094000
	3	528	7 5 6 8 2 5 1 3 2 4 4 8 5 7 8 4 6 7 1 4 3 6 6 2 1 1 5 6 2 3 7 7	0.078000
	4	520	3 8 4 1 7 6 3 5 3 7 1 2 2 7 2 6 1 6 5 8 1 2 3 1 4 8 8 3 6 7 4 3	0.094000
	5	529	1 7 4 7 6 3 1 1 3 7 3 1 8 2 5 8 4 5 8 8 5 7 7 2 6 4 2 2 5 3 3 6	0.094000
Επαναληπτική Τοπική Αναζήτηση	1	537	7 5 8 3 7 3 8 2 4 7 4 4 2 6 3 5 6 1 6 3 5 2 4 7 2 1 2 8 8 1 5 6	3.953000
	2	550	8 4 6 4 4 3 7 1 8 5 1 3 8 6 3 5 2 4 5 7 3 7 7 1 2 6 6 4 7 6 1 4	3.390000
	3	566	7 5 6 4 2 3 8 6 1 8 7 5 8 7 5 4 8 3 1 7 3 7 6 2 1 8 4 1 2 4 7 3	3.407000
	4	529	6 6 4 1 5 1 3 8 3 7 1 2 2 7 2 6 1 8 4 8 1 2 3 6 3 8 7 3 5 7 4 3	3.406000
	5	537	1 5 4 7 6 5 1 7 3 6 3 7 8 1 2 8 5 2 8 8 4 1 7 4 3 4 2 2 5 3 3 6	3.406000
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	561	7 7 2 3 1 6 7 2 4 7 4 4 5 3 8 5 8 3 6 6 5 2 1 7 8 5 2 8 8 1 4 6	0.219000
	2	543	6 4 6 4 4 8 7 5 8 3 1 3 8 2 5 7 2 4 5 5 3 1 1 6 7 3 6 4 7 1 2 4	0.234000
	3	533	3 5 6 8 2 5 5 3 2 4 5 8 8 7 2 4 7 4 1 4 8 6 1 3 1 2 6 6 1 8 7 7	0.235000
	4	527	3 6 4 1 7 6 7 8 3 3 1 2 2 7 2 4 6 6 4 1 1 2 8 5 3 8 6 5 5 7 4 1	0.234000
	5	566	7 7 4 4 5 1 2 1 6 2 3 4 5 8 2 4 3 6 8 5 1 3 8 7 5 8 2 2 5 8 3 6	0.219000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Αποδοχής Κατωφλιού	1	535	6 5 8 3 1 3 8 2 7 7 7 4 2 5 3 5 8 6 6 4 5 2 4 7 2 1 2 8 8 4 1 5	0.484000
	2	530	8 4 6 4 4 8 7 2 8 5 1 1 4 3 3 5 2 7 5 7 3 1 1 6 7 6 6 8 7 6 2 4	0.485000
	3	523	3 5 6 8 2 5 5 3 2 4 4 8 8 7 2 4 8 7 1 4 3 6 1 2 1 1 6 6 7 5 7 8	0.484000
	4	516	7 8 4 1 7 1 3 8 3 4 1 2 2 7 2 6 6 8 5 1 1 2 3 5 3 8 8 3 6 7 4 3	0.484000
	5	529	1 7 6 7 6 3 1 7 3 8 3 1 8 1 5 2 5 4 5 8 1 3 7 4 6 4 2 2 5 2 8 6	0.485000
Μεταβλητής Γειτονιάς Αναζήτησης	1	557	7 5 4 1 2 6 8 2 7 7 4 7 2 3 3 5 6 1 6 3 8 2 4 8 2 1 5 8 8 4 6 5	0.031000
	2	558	2 1 6 5 8 5 1 5 8 3 1 2 4 6 3 2 7 7 5 8 3 1 7 6 7 6 1 8 7 6 4 4	0.047000
	3	537	3 2 6 8 2 5 5 7 2 3 6 4 8 7 2 4 6 5 1 7 8 6 1 1 4 8 5 6 3 8 7 7	0.031000
	4	524	3 8 4 1 7 1 3 8 4 1 1 2 8 7 6 5 1 8 5 3 1 2 3 6 2 5 7 3 6 7 4 3	0.031000
	5	535	1 7 4 7 5 3 1 7 3 6 1 5 5 8 5 2 4 4 8 8 4 7 3 2 6 8 2 2 1 8 3 6	0.032000

Πίνακας 7: Αποτελέσματα προβλημάτων αρχείου gap7.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	669	8 5 1 3 4 4 6 5 2 7 7 7 8 1 8 1 8 4 7 1 1 4 6 3 6 3 2 3 6 2 2 7 6 3 7 8 1 4 5 7	0.093000
	2	689	4 2 6 7 8 7 5 6 1 7 2 4 4 7 8 3 4 6 1 3 2 4 5 3 6 7 2 8 7 5 3 6 1 1 8 5 2 5 5 1	0.047000
	3	678	5 4 8 6 1 7 3 2 2 6 4 7 8 4 3 7 4 1 3 5 7 8 2 1 4 6 8 6 5 1 7 2 5 1 5 1 7 2 6 6	0.063000
	4	650	7 8 1 8 8 5 5 5 7 3 8 7 2 1 6 1 4 5 3 7 3 3 2 6 4 7 4 8 2 2 5 6 1 3 4 1 4 4 8 8	0.062000
	5	666	1 1 7 2 5 8 3 5 6 3 6 2 6 7 3 1 7 7 8 1 5 5 2 4 1 2 5 4 5 4 3 8 1 3 7 4 7 6 4 8	0.047000
Περιορισμένη Αναζήτηση	1	659	4 5 3 1 4 6 6 8 7 7 7 5 3 1 8 7 8 4 2 2 1 4 6 3 4 3 2 1 6 2 6 7 5 3 7 8 1 1 5 7	0.109000
	2	669	4 2 6 7 8 6 5 6 6 7 2 3 4 7 8 4 6 1 1 7 1 8 2 3 1 1 2 8 4 5 3 8 5 2 5 5 7 1 4 3	0.110000
	3	680	2 8 3 8 3 7 7 2 2 6 4 5 8 4 3 7 4 1 6 1 8 5 2 1 4 2 8 6 5 1 7 2 4 1 5 1 7 3 6 6	0.094000
	4	651	6 8 1 3 7 5 5 5 7 3 8 7 2 1 2 1 4 5 7 7 3 4 6 6 4 7 7 8 2 2 5 6 1 3 4 1 4 4 8 8	0.109000
	5	654	1 1 4 2 5 2 5 3 8 7 6 2 6 7 5 1 7 7 4 1 5 5 2 4 4 6 3 8 6 8 3 4 1 3 7 8 2 6 1 8	0.125000
Επαναληπτική Τοπική Αναζήτηση	1	737	2 2 3 1 4 4 5 2 5 7 7 5 8 6 2 3 8 4 8 4 1 8 6 1 4 3 4 6 7 2 6 7 3 5 7 5 7 7 5 8	7.562000
	2	687	4 2 6 7 7 7 5 6 1 3 2 3 4 7 8 4 4 1 4 4 8 2 5 3 6 7 2 8 8 5 3 6 1 5 5 5 2 1 6 1	6.625000
	3	780	2 4 7 8 7 6 3 2 2 1 8 8 3 8 3 6 4 6 6 2 2 1 1 4 8 6 5 5 6 5 3 5 5 1 3 7 4 4 8 6	6.907000
	4	714	7 8 4 3 8 3 4 7 8 3 7 5 1 5 2 1 4 5 7 5 5 6 2 6 2 7 8 2 8 6 3 2 1 6 1 2 4 4 8 1	7.625000
	5	670	1 1 8 5 3 2 3 5 8 7 2 2 6 7 3 2 5 7 8 8 5 1 2 4 4 6 5 4 6 8 3 2 1 4 7 4 7 6 1 5	7.125000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	721	8 2 5 5 7 6 1 6 4 7 7 8 3 5 2 3 2 4 3 1 1 1 6 6 4 3 4 3 8 2 7 7 8 2 4 8 1 5 4 7	0.422000
	2	691	3 3 6 7 5 7 5 6 1 6 7 6 4 7 3 4 2 1 1 4 8 5 5 3 2 7 2 1 8 8 4 8 6 2 5 5 7 5 4 1	0.453000
	3	693	8 4 8 8 3 7 6 2 1 6 5 7 2 5 3 6 4 1 3 1 7 7 2 1 4 6 4 8 1 2 8 5 5 1 6 1 4 2 6 7	0.422000
	4	684	6 3 1 8 7 6 5 7 6 7 8 7 8 6 4 1 4 5 3 5 4 3 2 6 8 1 7 2 2 2 5 2 1 3 4 2 8 5 4 1	0.421000
	5	678	1 2 7 8 5 8 5 3 8 3 2 2 6 3 2 3 5 7 4 8 5 1 2 4 1 6 5 4 6 7 3 2 1 4 5 7 7 6 1 8	0.407000
Αποδοχής Κατωφλιού	1	658	4 2 3 1 4 8 1 4 2 7 7 5 3 8 3 1 8 4 5 5 7 4 6 6 4 7 2 1 6 2 6 7 6 3 7 8 1 2 5 7	0.625000
	2	678	4 2 6 7 6 7 5 6 1 6 2 6 4 8 3 4 1 3 3 7 1 8 2 3 7 7 2 8 8 5 4 8 1 2 5 5 5 5 4 7	0.593000
	3	666	2 8 8 5 3 7 3 2 1 6 4 5 2 5 3 7 4 6 5 1 7 8 2 1 4 6 8 6 5 1 2 1 4 1 7 1 4 3 6 7	0.610000
	4	647	6 8 1 8 7 5 5 5 6 7 8 7 2 1 6 1 2 5 7 5 3 3 6 4 4 7 7 8 8 2 3 2 1 3 4 1 4 4 4 8	0.625000
	5	655	1 1 7 2 5 2 5 3 8 1 2 2 6 7 7 8 7 7 8 3 5 5 2 4 4 6 5 4 5 8 3 4 1 3 6 4 7 6 1 8	0.594000
Μεταβλητής Γειτονιάς Αναζήτησης	1	679	8 5 7 1 4 6 1 4 4 7 8 8 4 8 2 7 2 1 3 2 5 4 6 3 6 3 2 1 3 2 6 8 6 3 7 8 1 7 5 7	0.062000
	2	679	2 3 6 7 8 6 5 2 4 7 5 1 6 8 3 4 4 1 3 7 6 8 5 4 1 1 2 8 7 5 8 6 6 2 8 5 7 1 2 1	0.047000
	3	697	5 8 8 8 8 7 7 2 2 6 5 5 2 4 6 7 4 1 3 5 8 1 4 1 4 6 1 7 5 4 2 1 3 7 3 6 3 3 6 4	0.047000
	4	665	6 7 1 3 1 5 5 5 7 3 8 1 2 6 2 1 4 7 7 7 3 3 6 4 4 7 7 8 2 3 5 6 2 5 8 1 4 4 8 8	0.047000
	5	678	1 2 5 5 3 2 4 3 8 1 2 2 6 3 7 8 7 7 8 1 5 2 2 3 1 6 5 8 6 8 3 4 1 6 6 5 2 4 4 5	0.047000

Πίνακας 8: Αποτελέσματα προβλημάτων αρχείου gar8.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	889	6 2 1 1 8 8 7 6 8 4 4 1 2 4 5 8 3 4 6 4 3 7 6 7 5 7 5 6 5 5 7 7 6 3 4 5 2 2 3 1 1 2 3 4 1 2 8 8	0.093000
	2	811	4 4 8 4 4 8 2 6 5 1 3 6 2 8 3 7 2 1 5 8 7 2 6 5 5 1 7 6 7 1 1 7 3 3 8 8 4 7 3 7 6 3 5 6 5 1 1 2	0.078000
	3	846	6 2 4 8 4 7 8 4 1 8 8 5 6 5 1 5 4 4 7 2 1 7 1 2 2 2 6 7 5 8 3 3 6 3 5 6 1 3 8 1 6 3 7 6 2 2 7 3	0.063000
	4	828	6 1 2 5 8 3 6 7 2 3 5 1 8 4 1 8 6 7 7 8 5 8 3 6 4 3 2 1 6 5 8 4 7 6 1 3 4 1 2 1 5 7 5 4 3 3 7 4	0.078000
	5	850	4 3 1 6 6 3 1 6 7 1 4 2 1 5 5 6 2 3 5 7 2 7 8 8 7 4 5 7 8 6 3 4 3 5 2 8 2 6 1 3 8 7 4 1 3 7 5 8	0.078000
Περιορισμένη Αναζήτηση	1	827	8 5 2 1 8 3 7 1 8 4 1 5 1 6 5 7 2 8 2 3 8 7 6 7 4 7 2 6 3 5 3 7 2 3 4 8 5 5 6 1 4 4 3 4 7 2 6 5	0.125000
	2	812	2 6 8 4 4 2 7 1 4 1 5 6 6 5 2 8 5 8 5 8 7 7 1 7 3 1 3 6 8 7 2 5 3 3 8 8 1 1 3 7 6 3 6 4 5 1 4 6	0.125000
	3	829	6 8 4 8 3 5 8 2 6 8 7 7 1 5 1 5 2 3 7 6 6 3 1 4 4 8 6 5 5 7 1 3 6 3 2 7 1 2 8 1 3 4 7 6 2 2 4 3	0.140000
	4	809	6 8 2 2 8 3 6 7 2 1 6 8 8 4 1 3 2 7 4 7 5 5 3 6 3 3 4 4 6 5 8 5 7 1 1 5 6 8 4 2 3 7 5 1 8 5 1 4	0.140000
	5	1030	3 1 1 8 6 3 2 4 2 6 8 6 5 3 5 6 8 4 3 7 2 2 8 7 7 7 5 7 3 1 6 4 3 5 2 8 2 6 1 3 5 7 5 4 1 4 1 8	0.141000
Επαναληπτική Τοπική Αναζήτηση	1	1041	1 2 2 1 7 3 1 6 3 3 3 5 6 4 8 7 6 8 7 3 4 7 2 7 5 1 5 8 5 5 5 8 5 6 4 8 8 2 1 1 4 4 8 2 7 4 6 3	13.125000
	2	878	5 4 8 4 6 2 2 2 4 7 5 6 6 2 5 8 7 1 5 8 7 7 1 8 3 1 3 4 8 6 1 7 1 3 7 8 3 8 3 4 6 3 5 5 2 6 2 2	13.563000
	3	922	3 8 4 4 7 1 8 4 4 4 6 4 1 5 1 2 5 3 2 1 6 2 1 4 6 8 6 5 2 7 3 3 6 8 2 3 1 5 8 7 6 6 7 6 7 2 8 5	12.000000
	4	838	6 5 2 4 8 3 3 6 2 1 5 3 6 4 1 4 2 7 3 7 1 5 3 6 2 3 2 1 6 5 4 6 7 1 1 5 8 8 4 6 4 7 8 1 8 7 8 5	11.390000
	5	1105	1 6 7 2 3 3 2 1 7 2 5 6 5 2 5 6 8 7 1 7 7 6 8 7 2 4 6 1 4 8 5 5 3 3 3 4 6 6 4 3 8 4 8 4 3 1 2 8	13.328000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	831	2 4 1 1 5 5 7 8 6 3 8 5 6 6 5 8 2 8 2 3 7 7 6 7 3 4 4 2 3 5 5 7 7 3 4 8 3 5 1 1 4 6 7 4 1 2 8 1	1.344000
	2	889	5 1 8 4 8 5 2 6 4 1 3 1 7 8 5 4 7 4 8 4 8 7 6 2 3 1 7 6 5 6 1 3 7 3 5 3 3 6 4 7 6 2 5 8 2 2 1 2	0.672000
	3	874	2 8 4 8 5 5 6 4 4 6 8 7 1 5 1 8 5 1 7 3 6 7 4 2 7 1 2 3 5 8 3 8 6 3 3 7 1 2 4 1 4 3 2 1 2 2 5 6	1.078000
	4	869	6 1 2 2 1 4 1 5 2 3 8 8 7 4 1 4 2 7 6 5 2 5 3 6 8 8 6 3 6 5 4 3 7 4 2 5 7 7 4 6 3 7 3 1 5 6 1 4	0.766000
	5	854	2 6 5 4 6 3 1 4 5 8 2 7 3 5 5 1 1 3 1 7 6 7 8 7 7 2 5 8 4 8 3 6 3 5 2 8 8 6 7 3 4 7 5 4 3 1 2 1	0.687000
Αποδοχής Κατωφλιού	1	802	8 4 1 1 5 3 7 4 8 8 8 5 1 2 5 7 2 8 6 3 3 3 6 7 6 4 5 6 3 4 5 7 2 5 1 8 5 4 3 1 1 6 7 4 7 2 7 2	0.688000
	2	797	4 4 8 7 6 4 2 2 5 1 5 6 7 8 8 8 2 3 5 8 7 7 5 5 3 1 3 6 4 1 4 2 3 3 8 7 1 1 3 7 6 2 6 1 5 2 6 2	1.015000
	3	830	6 3 4 8 4 1 3 6 6 8 7 8 7 5 1 8 4 1 5 7 1 7 1 2 2 2 2 3 4 8 3 6 6 3 2 7 1 5 4 1 3 6 5 6 2 5 8 2	0.703000
	4	802	6 1 2 5 8 3 3 6 2 4 4 8 6 4 1 6 1 7 7 8 5 5 3 6 2 3 7 2 6 5 8 6 7 1 1 3 4 8 4 2 3 7 7 4 3 5 8 5	0.672000
	5	812	4 3 5 7 6 3 2 4 5 6 8 1 1 3 1 1 8 5 5 7 2 7 6 7 7 7 4 8 4 8 3 6 3 3 2 8 5 6 1 3 2 1 5 1 3 4 2 8	0.672000
Μεταβλητής Γειτονιάς Αναζήτησης	1	846	2 2 3 1 5 6 7 3 8 4 8 5 6 2 5 7 6 8 2 3 7 7 6 4 3 4 5 6 5 7 1 6 2 5 1 8 5 8 3 1 1 4 7 4 2 8 4 1	0.078000
	2	816	4 5 6 7 8 4 2 1 4 1 6 2 7 8 3 8 3 1 5 8 7 3 5 4 5 1 6 6 8 5 4 2 1 3 8 1 4 7 3 7 6 3 5 2 2 7 6 6	0.078000
	3	831	6 8 4 5 4 4 3 6 6 7 4 8 7 5 1 5 4 3 5 6 1 3 5 2 2 1 2 3 5 7 7 8 7 3 2 6 1 2 8 1 3 3 7 6 2 2 8 1	0.063000
	4	819	6 8 2 2 8 8 6 7 2 3 5 2 7 4 1 6 1 7 3 8 5 5 3 6 7 3 2 4 4 5 7 5 7 1 1 3 6 8 1 6 3 7 3 4 5 5 8 2	0.078000
	5	812	1 8 5 4 6 3 1 4 6 6 8 1 1 5 5 1 8 4 5 7 2 2 8 7 7 7 4 3 4 8 6 3 2 8 2 8 2 6 7 3 2 7 5 1 3 1 7 5	0.062000

Πίνακας 9: Αποτελέσματα προβλημάτων αρχείου gar9.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	490	10 10 9 5 8 8 7 4 3 9 3 3 8 5 4 9 2 1 7 8 1 2 6 4 10 2 5 10 7 6	0.063000
	2	488	5 5 3 9 6 1 7 6 3 10 7 8 9 1 9 10 1 4 8 10 5 9 4 2 2 9 2 7 8 5	0.047000
	3	515	8 1 5 7 6 5 9 3 9 10 8 9 4 5 3 4 2 10 8 3 7 9 1 1 2 8 10 4 2 6	0.062000
	4	511	8 5 1 2 10 7 3 10 7 9 10 5 2 5 9 4 6 1 10 2 4 8 8 4 3 6 3 5 7 3	0.047000
	5	513	1 2 10 8 5 5 4 10 5 1 10 7 4 2 2 9 7 6 1 4 7 1 6 9 3 6 8 9 3 2	0.062000
Περιορισμένη Αναζήτηση	1	490	8 1 7 5 8 8 3 4 3 1 2 3 8 5 4 10 2 4 7 5 3 9 6 7 10 2 9 10 6 6	0.094000
	2	500	5 5 3 9 6 10 1 9 8 3 7 2 10 3 6 10 1 7 6 5 5 9 10 2 4 8 7 7 1 4	0.078000
	3	505	10 4 5 7 5 1 9 3 6 7 8 7 6 4 8 4 3 10 1 3 6 9 1 2 7 8 10 2 2 9	0.125000
	4	501	8 5 3 2 9 7 3 1 2 7 7 5 10 4 9 2 8 1 10 2 4 8 6 4 5 6 3 5 10 3	0.094000
	5	500	1 5 5 4 6 10 4 10 3 8 8 7 4 9 1 8 7 6 10 5 2 1 6 9 1 7 2 9 3 2	0.094000
Επαναληπτική Τοπική Αναζήτηση	1	538	1 10 9 6 7 8 3 10 9 8 2 8 3 5 5 3 2 1 7 2 6 9 1 10 10 4 4 10 7 6	3.297000
	2	524	3 5 3 9 10 2 7 6 6 1 7 3 3 1 6 5 9 7 8 10 8 9 4 2 2 8 2 4 9 5	2.828000
	3	525	8 4 2 4 5 7 9 3 9 10 8 7 4 1 1 4 3 10 3 8 6 9 10 5 7 6 1 2 2 6	2.844000
	4	521	8 5 1 3 10 9 3 6 2 9 7 5 8 4 7 10 10 3 10 2 4 8 6 4 3 2 3 5 7 1	2.828000
	5	516	1 5 5 3 6 10 7 10 7 4 8 9 5 2 9 8 6 6 1 10 7 1 2 9 3 1 5 9 6 2	2.781000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	499	1 2 9 5 4 8 3 4 9 1 3 1 8 5 4 9 3 8 7 8 7 10 6 7 10 2 5 10 6 6	0.203000
	2	494	5 5 3 9 6 10 1 4 8 1 7 8 10 1 6 6 3 7 8 10 9 2 4 2 3 9 2 7 9 2	0.219000
	3	517	8 2 5 1 4 1 9 3 9 7 8 7 4 1 4 5 9 10 3 3 7 6 10 5 5 6 10 2 8 9	0.187000
	4	511	8 5 3 2 9 7 4 8 7 9 10 5 1 4 10 2 6 3 10 9 4 1 6 2 5 2 3 5 7 3	0.219000
	5	497	1 8 5 9 6 10 7 8 3 1 8 7 4 2 4 2 6 9 10 10 7 1 6 9 3 7 5 5 1 2	0.219000
Αποδοχής Κατωφλιού	1	490	10 2 7 5 8 8 3 1 9 4 6 3 8 5 3 9 2 8 7 1 3 9 4 10 10 2 5 10 6 6	0.484000
	2	478	5 5 3 9 8 2 1 7 8 6 7 8 10 3 9 10 1 7 6 10 5 4 4 2 2 9 2 9 1 5	0.500000
	3	506	10 4 9 1 7 1 9 2 6 7 8 7 6 4 8 4 3 10 8 3 5 9 1 5 7 6 10 2 3 6	0.485000
	4	505	3 9 2 2 8 7 5 10 6 1 10 5 1 4 7 4 8 3 10 2 9 7 8 4 3 6 3 5 2 3	0.500000
	5	494	1 8 5 9 3 10 7 10 7 4 8 9 6 2 1 5 6 4 10 4 6 1 2 9 3 7 8 5 1 2	0.500000
Μεταβλητής Γειτονιάς Αναζήτησης	1	512	10 3 9 5 8 8 3 4 3 4 6 9 1 5 4 7 2 1 7 6 8 2 7 6 10 2 5 10 7 1	0.031000
	2	513	6 4 3 9 6 2 3 1 8 1 10 5 5 3 7 6 1 4 7 10 8 9 10 2 4 1 2 7 8 9	0.047000
	3	537	6 10 5 1 4 8 9 3 10 7 9 9 6 1 7 9 2 3 4 3 6 1 10 5 7 2 5 7 2 8	0.031000
	4	531	8 5 2 2 10 7 3 6 2 6 7 5 9 4 9 2 4 3 10 9 4 1 10 4 3 6 10 5 7 1	0.047000
	5	510	4 2 10 9 6 10 7 1 5 4 3 9 9 10 2 5 7 6 1 10 6 1 7 8 3 7 2 5 3 2	0.031000

Πίνακας 10: Αποτελέσματα προβλημάτων αρχείου gap10.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	667	1 10 4 8 9 9 7 6 3 1 8 10 5 5 4 6 5 4 3 2 4 6 9 9 2 1 2 7 3 10 5 1 8 8 2 4 2 9 9 1	0.125000
	2	703	8 4 9 7 3 9 1 10 2 8 7 2 7 6 10 1 4 3 9 8 5 5 3 4 10 6 8 1 7 2 5 4 2 10 3 5 6 6 9 7	0.078000
	3	681	2 1 2 10 4 3 5 6 3 6 9 4 7 10 7 9 2 3 4 5 7 6 8 10 9 4 10 4 5 6 1 1 7 3 8 1 5 8 3 2	0.094000
	4	648	1 2 5 9 3 10 9 3 10 7 8 4 7 7 10 8 7 3 2 7 6 1 9 8 10 6 9 4 5 4 10 6 8 2 8 6 1 2 3 5	0.110000
	5	652	7 5 1 8 7 4 5 10 9 10 6 5 8 10 7 9 10 1 7 3 7 7 1 10 2 2 3 6 3 9 4 6 4 4 8 9 4 2 6 6	0.093000
Περιορισμένη Αναζήτηση	1	652	1 6 10 8 2 4 4 6 9 1 4 7 5 3 6 10 5 8 8 7 1 6 3 9 2 1 2 5 8 10 9 10 2 3 2 5 9 7 9 1	0.157000
	2	654	8 6 6 3 3 9 7 5 2 9 5 2 7 6 10 1 1 9 10 3 5 9 5 4 10 6 2 7 7 1 2 4 8 6 8 5 4 4 10 2	0.171000
	3	667	7 1 1 6 4 5 5 3 3 6 9 4 4 1 8 10 2 3 3 5 10 6 8 7 9 5 7 4 2 6 7 1 10 3 8 9 2 6 10 2	0.157000
	4	638	2 2 8 6 3 10 9 2 10 7 8 8 7 7 10 5 7 9 2 10 6 1 9 4 5 1 9 4 5 4 6 10 8 4 8 3 1 3 3 6	0.172000
	5	662	9 7 1 5 4 4 6 10 8 5 9 8 6 10 7 10 10 1 7 3 4 7 2 1 2 8 3 6 1 9 4 10 2 7 3 1 3 2 5 6	0.171000
Επαναληπτική Τοπική Αναζήτηση	1	696	10 5 4 10 10 7 6 6 3 3 1 9 2 5 4 2 5 8 3 7 1 7 8 9 2 1 5 6 1 10 5 10 2 3 8 4 9 7 9 4	9.172000
	2	717	1 4 9 3 1 1 7 5 2 7 5 2 6 3 10 9 2 9 10 8 10 9 6 6 3 8 2 10 9 9 5 4 4 8 7 3 5 4 10 2	8.031000
	3	720	7 1 3 5 5 4 5 3 4 7 3 6 9 1 7 9 4 3 2 6 4 1 8 2 7 8 10 6 2 5 1 1 10 8 10 9 2 6 10 10	6.250000
	4	673	10 3 2 10 2 7 9 4 10 7 8 8 5 7 1 4 6 8 2 10 6 1 4 8 5 6 9 4 5 7 9 10 3 2 9 3 1 2 1 6	6.516000
	5	714	6 8 4 8 4 7 8 3 9 5 9 8 6 10 9 10 10 1 5 1 7 7 1 5 2 10 3 6 3 9 2 5 4 1 3 6 4 2 10 2	6.422000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	664	1 10 10 8 2 9 4 7 9 3 4 1 6 5 4 2 5 8 3 2 1 6 3 9 7 1 2 3 5 10 7 10 8 8 2 5 9 7 9 4	0.406000
	2	660	8 4 6 7 3 9 7 5 2 2 5 1 7 6 10 1 8 7 10 8 5 9 9 4 4 6 2 8 10 2 4 5 1 6 9 3 4 6 3 2	0.422000
	3	711	10 1 3 9 4 5 5 5 2 6 8 7 3 1 8 8 2 3 3 6 10 6 7 10 6 4 7 4 2 9 2 1 1 3 10 8 2 4 7 9	0.437000
	4	660	2 2 5 9 3 10 9 3 10 7 1 8 7 7 10 3 6 4 2 9 7 4 4 8 8 6 8 4 5 10 6 10 8 2 8 2 1 5 1 6	0.422000
	5	682	7 7 6 4 7 6 8 10 9 5 9 5 8 1 3 10 10 1 9 2 4 5 7 2 2 1 3 2 3 5 4 5 4 6 3 1 4 1 8 10	0.453000
Αποδοχής Κατωφλιού	1	655	1 4 5 5 9 7 8 6 9 1 4 10 2 3 7 6 5 4 3 7 4 6 9 9 3 1 2 1 8 10 7 10 8 3 2 5 2 7 9 4	0.610000
	2	652	8 9 6 3 3 1 7 5 2 7 2 2 7 6 10 1 8 2 10 1 5 9 9 4 4 6 2 6 7 2 4 5 4 6 3 5 8 4 9 10	0.609000
	3	673	2 9 2 7 4 3 5 3 3 6 9 7 4 6 8 8 2 1 3 8 7 5 10 10 4 5 10 4 9 6 1 1 1 3 10 9 5 2 10 10	0.609000
	4	641	2 6 8 9 3 3 9 10 10 7 8 8 7 7 1 3 5 5 2 6 7 1 9 4 10 2 9 4 5 4 6 10 8 4 8 1 1 2 3 5	0.610000
	5	656	2 5 1 5 4 4 6 8 9 5 9 5 6 10 7 9 10 1 7 3 7 7 1 10 2 8 4 2 3 9 10 6 2 7 8 1 4 3 2 6	0.609000
Μεταβλητής Γειτονιάς Αναζήτησης	1	672	1 10 10 8 10 9 6 6 3 1 4 7 1 2 5 6 5 8 8 7 1 4 3 9 2 1 2 5 8 10 9 4 2 8 5 3 9 9 6 4	0.063000
	2	675	8 6 9 7 3 1 7 5 2 9 5 3 7 6 10 3 7 9 4 8 2 4 3 4 4 6 3 6 7 1 2 5 1 6 8 10 8 4 9 10	0.062000
	3	689	8 1 8 5 9 6 5 3 3 6 9 7 7 1 5 8 2 2 3 9 10 5 8 10 7 4 10 4 6 6 1 2 1 3 10 8 6 4 10 2	0.063000
	4	709	5 2 6 9 2 10 7 2 1 5 8 4 5 7 1 4 5 3 3 7 7 1 6 1 8 8 2 9 9 10 6 10 10 8 3 2 1 9 4 9	0.078000
	5	681	9 7 5 8 4 8 5 4 9 5 6 1 6 10 3 3 10 1 7 3 7 8 1 2 2 1 2 6 3 5 2 6 2 7 8 9 4 10 6 4	0.078000

Πίνακας 11: Αποτελέσματα προβλημάτων αρχείου gap11.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	605	2 4 4 9 2 8 6 4 3 2 7 5 5 5 9 1 3 10 4 2 3 8 6 1 2 7 2 2 1 8 5 6 1 10 9 3 3 5 4 9 10 10 7 10 7 1 5 9 6 3	0.094000
	2	644	2 4 5 9 4 7 5 1 9 9 10 3 8 6 10 2 1 10 4 1 3 6 6 1 4 6 8 10 3 2 5 1 5 2 7 5 7 6 2 4 5 3 7 8 3 10 9 8 2 6	0.062000
	3	628	1 1 10 3 8 7 8 10 1 5 1 1 7 6 4 8 10 2 7 3 7 5 4 1 2 2 6 9 4 6 6 9 9 8 10 4 5 6 3 5 7 2 3 3 8 3 2 5 1 6	0.078000
	4	605	10 8 10 10 8 9 7 3 6 4 4 10 7 7 1 1 4 7 8 5 6 6 1 9 10 2 4 2 6 3 4 9 4 8 7 2 7 5 3 5 3 5 5 1 3 2 1 10 2 9	0.078000
	5	599	1 7 4 3 8 5 7 4 3 9 1 8 7 2 7 7 9 5 2 5 5 10 4 9 3 2 6 9 9 2 1 1 3 8 6 4 7 1 3 6 9 6 8 10 8 5 8 2 10 10	0.063000
Περιορισμένη Αναζήτηση	1	595	2 4 3 5 6 8 6 10 3 8 5 5 5 6 5 10 3 2 4 2 3 5 3 1 1 5 2 2 1 7 2 6 10 7 9 3 1 4 1 9 9 10 7 10 7 1 5 4 8 3	0.141000
	2	598	7 5 4 9 3 7 5 1 9 9 10 3 8 4 8 10 1 10 4 4 3 1 5 1 5 6 8 3 3 2 4 3 6 7 7 8 6 6 4 5 10 9 7 2 2 10 3 2 3 6	0.140000
	3	608	4 10 10 9 8 2 8 4 1 5 1 1 7 5 4 8 10 9 7 6 7 2 7 1 10 5 3 9 9 6 6 1 3 6 10 4 1 6 3 5 7 2 3 3 8 3 2 5 1 6	0.125000
	4	591	8 8 10 2 8 3 4 9 4 4 6 10 7 7 1 3 4 5 1 5 6 10 1 7 10 6 4 2 6 3 8 9 7 2 7 3 7 9 3 5 5 5 5 9 1 2 1 10 2 9	0.125000
	5	591	1 7 4 3 7 5 7 8 2 9 1 8 7 5 7 9 9 5 6 2 5 6 4 9 9 2 6 10 3 2 1 1 3 8 8 4 7 1 3 10 6 6 8 10 8 2 10 9 4 1	0.141000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Επαναληπτική Τοπική Αναζήτηση	1	706	2 8 3 9 6 4 8 4 6 8 10 2 3 1 10 1 3 6 4 2 10 2 7 1 8 7 3 6 5 6 5 6 10 8 9 5 9 4 1 9 5 3 7 2 7 1 5 3 2 2	12.078000
	2	676	7 1 3 7 4 7 3 2 2 2 3 7 8 4 4 10 10 2 6 4 9 1 7 3 4 9 1 10 5 2 4 1 6 2 7 8 5 8 6 5 10 9 4 6 6 10 9 8 3 1	11.672000
	3	677	10 2 10 5 6 2 8 7 1 3 1 9 4 7 4 1 4 2 7 3 6 5 1 1 3 9 6 9 3 5 6 9 4 8 10 2 8 8 1 5 5 6 7 3 8 3 2 9 1 2	11.656000
	4	680	8 8 10 7 2 2 5 6 10 1 6 8 7 5 3 5 4 7 9 7 3 1 6 3 5 6 9 2 9 10 1 1 4 8 7 10 7 5 4 2 3 10 5 9 3 4 2 3 2 9	11.672000
	5	627	1 7 4 3 1 5 6 8 2 9 1 2 7 5 5 9 9 6 2 2 5 6 4 9 4 4 3 5 9 1 1 6 8 8 6 8 7 1 3 10 7 5 8 10 7 9 10 2 7 10	11.625000
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	621	1 4 4 1 6 4 6 9 3 8 5 7 5 4 5 8 6 10 2 2 3 2 7 1 4 7 2 2 1 9 5 6 10 10 9 3 3 9 8 6 3 10 6 10 7 4 5 2 5 3	0.750000
	2	623	7 5 1 7 4 7 10 3 9 9 5 3 8 4 8 6 10 10 6 1 7 1 6 7 4 10 8 3 3 2 4 2 5 2 7 8 3 6 2 5 1 9 4 4 9 10 3 2 5 6	0.734000
	3	646	10 1 10 6 5 2 8 1 1 3 9 7 7 5 4 1 3 9 7 3 6 9 7 1 3 2 6 9 8 6 6 9 4 5 10 2 1 10 3 5 7 2 1 3 8 10 2 5 1 4	0.766000
	4	606	10 8 2 5 1 3 5 9 9 4 4 10 7 8 1 3 4 8 3 5 9 10 1 7 6 8 4 2 6 6 4 4 7 6 7 2 7 9 3 5 3 5 5 1 1 10 2 10 7 9	0.734000
	5	622	3 7 10 5 6 7 7 4 2 9 6 8 3 5 5 7 9 5 1 10 10 10 4 5 9 4 6 5 7 2 1 9 3 8 8 1 7 1 8 6 6 1 8 9 8 2 8 2 10 1	0.750000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Αποδοχής Κατωφλιού	1	605	2 4 4 5 6 8 6 4 3 9 5 5 9 5 9 3 7 10 2 2 9 4 7 1 1 5 2 2 1 1 2 6 10 10 9 3 8 5 1 10 10 10 6 8 7 4 7 3 6 8	0.719000
	2	597	7 5 4 7 3 7 5 1 9 9 10 3 8 4 8 10 1 6 4 1 7 9 9 1 5 6 8 10 3 1 5 2 7 2 7 8 6 6 4 5 10 9 3 4 2 10 3 2 3 6	0.703000
	3	599	7 10 10 6 8 7 8 4 1 3 1 1 7 5 4 1 10 5 7 3 2 2 8 1 10 5 3 9 3 6 6 7 4 8 10 4 5 6 1 5 7 9 9 3 8 3 2 5 9 6	0.703000
	4	589	8 8 10 10 1 9 4 6 1 1 4 10 7 8 1 7 4 8 9 5 6 10 1 3 6 6 4 2 6 3 4 9 4 2 7 3 7 9 3 5 5 5 5 9 3 2 2 10 7 2	0.719000
	5	586	1 7 10 3 7 5 6 8 2 9 1 8 7 5 5 7 9 5 2 2 10 6 4 9 7 4 6 9 7 2 1 1 3 8 6 4 7 1 3 4 9 6 8 10 8 2 10 9 3 1	0.703000
Μεταβλητής Γειτονιάς Αναζήτησης	1	608	2 4 4 5 6 8 6 10 3 8 5 4 9 4 7 10 3 10 6 2 3 2 7 1 1 5 6 2 1 4 5 6 1 7 9 3 9 5 1 9 10 2 7 8 7 4 5 2 8 3	0.110000
	2	626	2 5 4 1 4 5 5 1 9 2 8 6 9 4 4 10 10 2 6 4 7 8 2 1 5 6 8 2 3 1 4 1 7 7 5 9 10 6 3 9 10 9 7 7 3 2 3 5 3 6	0.093000
	3	627	1 1 4 5 8 7 8 4 1 3 1 9 7 6 4 1 10 5 7 2 2 2 7 1 10 5 6 9 3 6 6 10 4 8 10 2 8 6 3 5 8 9 7 3 8 3 2 3 9 3	0.110000
	4	641	8 8 10 5 1 2 7 9 7 4 4 10 7 8 4 3 4 8 3 5 10 10 1 7 1 4 3 2 6 2 4 9 7 6 2 2 2 5 4 1 3 10 1 9 3 9 1 3 6 9	0.093000
	5	627	1 3 4 4 6 5 6 8 2 9 1 8 7 5 7 7 9 5 1 2 5 6 4 8 3 3 3 9 1 2 1 6 3 7 6 4 2 4 8 10 9 7 8 10 3 9 10 9 7 1	0.094000

Πίνακας 12: Αποτελέσματα προβλημάτων αρχείου gap12.txt

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Προσομοιωμένη Ανόπτηση	1	1010	10 4 1 4 6 8 1 5 2 10 3 5 6 9 4 1 8 5 3 1 8 9 6 7 4 8 10 7 10 9 6 10 10 2 4 2 5 6 8 2 4 9 7 8 7 9 2 6 3 8 9 3 1 3 2 1 8 5 7 4	0.110000
	2	985	1 5 7 5 9 2 4 1 10 3 3 8 4 1 9 3 5 3 2 9 3 10 1 7 6 2 7 8 6 7 2 5 10 6 4 7 4 5 9 4 1 10 8 3 2 8 1 3 8 6 6 3 9 4 8 4 7 8 1 6	0.093000
	3	969	2 3 6 9 9 6 5 8 6 10 1 10 4 6 9 4 3 1 6 5 7 7 8 1 9 5 10 5 8 1 7 3 8 5 3 2 7 4 7 1 9 1 9 4 1 2 8 6 5 8 6 4 7 2 3 2 1 8 6 4	0.078000
	4	982	2 9 1 9 6 5 8 9 5 9 10 6 4 4 3 10 7 3 2 10 4 7 7 4 2 3 7 7 4 9 7 3 9 10 6 8 9 6 8 3 5 7 1 3 2 10 2 4 5 1 10 1 3 5 8 6 1 6 4 8	0.079000
	5	990	2 10 9 6 3 7 7 2 5 3 3 5 8 4 3 6 9 10 5 10 8 9 7 6 8 1 6 9 3 4 2 5 1 8 8 7 7 4 9 2 4 6 5 7 6 6 1 9 7 10 1 9 2 10 3 2 6 8 4 1	0.093000
Περιορισμένη Αναζήτηση	1	998	6 8 1 4 6 3 6 5 4 4 10 5 9 10 9 1 8 5 3 8 2 9 6 8 3 8 10 5 7 2 6 10 10 2 4 2 9 2 7 2 4 8 9 7 7 3 3 4 2 1 1 5 1 3 2 7 8 5 9 10	0.141000
	2	974	1 10 10 5 9 2 4 1 10 3 3 8 4 7 7 3 5 3 6 9 5 3 1 7 9 4 7 2 6 6 2 5 10 7 8 7 3 9 9 6 4 10 8 3 4 2 1 3 8 5 5 1 9 4 8 4 7 8 1 6	0.172000
	3	957	6 3 2 8 9 6 5 3 9 2 3 5 6 6 2 3 5 1 4 5 7 7 7 4 1 5 10 5 10 8 7 2 7 9 3 2 7 4 5 1 9 1 9 4 1 1 8 6 4 8 6 3 7 2 10 2 1 10 8 4	0.140000
	4	980	6 9 3 9 6 7 9 9 4 5 10 6 4 10 3 10 7 5 2 10 4 10 7 9 2 3 9 3 4 9 2 3 9 7 6 8 1 8 5 6 1 7 8 1 2 7 5 4 4 8 10 1 3 5 8 6 1 6 4 7	0.157000
	5	960	6 3 6 6 3 7 7 2 6 2 9 8 8 9 7 5 9 10 5 10 8 3 1 3 8 5 4 9 3 4 2 5 10 8 10 4 7 4 9 4 4 1 9 7 10 1 1 2 6 8 1 9 5 10 3 5 1 8 7 2	0.172000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Επαναληπτική Τοπική Αναζήτηση	1	1082	2 10 1 1 8 3 1 9 5 4 8 5 3 9 9 10 5 9 6 2 7 9 8 7 10 10 6 10 4 9 7 5 5 7 4 2 10 6 4 6 6 8 3 8 6 4 2 2 6 8 1 5 7 3 2 7 1 9 2 10	20.375000
	2	1052	8 2 9 10 9 2 6 1 10 3 10 8 4 7 4 3 5 3 1 9 5 1 2 7 6 7 7 5 6 8 6 5 2 7 8 10 5 9 10 4 10 10 8 9 3 7 1 3 6 2 4 3 10 4 8 4 3 5 9 5	19.546000
	3	1042	2 9 2 1 8 5 6 6 7 3 10 3 8 4 9 3 1 1 4 9 7 1 4 4 5 5 3 5 10 1 2 2 8 5 3 2 7 8 5 10 9 2 4 4 6 10 6 6 6 5 6 1 7 1 9 2 5 7 7 4	19.547000
	4	1037	9 9 1 1 6 6 2 7 5 9 4 8 4 4 8 10 7 3 5 10 2 7 5 9 2 3 7 3 5 4 2 3 9 10 6 8 1 3 3 6 1 9 5 6 4 3 8 4 4 2 10 5 10 5 8 6 10 6 7 7	19.594000
	5	1028	2 6 7 6 3 4 7 2 2 2 9 8 8 1 3 6 2 4 5 10 1 5 5 9 3 3 6 1 5 4 10 5 1 8 5 4 10 8 7 2 7 10 6 7 6 6 4 9 6 10 9 9 9 10 5 2 6 4 8 8	19.656000
Άπληστη Τυ- χαιοποιημένη Αναζήτηση	1	1022	6 2 2 4 6 3 1 5 5 4 8 8 6 9 8 1 8 5 3 3 7 9 7 7 4 8 10 5 1 9 10 10 10 1 4 2 8 2 8 6 6 9 9 4 3 4 2 4 6 10 9 5 1 3 2 7 10 5 7 4	1.203000
	2	998	1 10 10 5 9 2 1 1 10 2 3 8 9 1 8 3 9 3 2 9 3 2 1 7 6 2 7 8 6 6 2 5 3 9 7 6 10 7 5 4 4 10 8 3 4 4 6 3 8 5 5 3 8 4 8 4 7 5 1 6	1.188000
	3	990	10 6 2 8 9 5 6 3 7 2 3 3 10 7 2 3 1 1 6 1 7 7 8 1 1 5 6 6 8 1 7 2 8 5 2 4 7 9 10 10 5 9 9 4 6 2 8 6 10 8 3 4 7 4 3 2 1 9 5 4	1.203000
	4	994	7 9 10 1 6 5 9 9 4 9 8 3 4 10 3 10 9 5 8 10 4 7 6 4 2 3 7 2 4 9 7 3 9 7 6 3 1 6 8 6 7 7 4 3 2 2 2 5 8 1 10 1 10 6 4 6 1 6 8 5	1.203000
	5	970	2 4 9 6 3 10 7 2 1 2 9 8 8 1 3 6 9 8 1 10 8 2 1 3 6 5 6 9 3 4 10 7 10 5 5 4 7 4 9 2 7 3 6 7 6 6 8 9 1 4 9 9 10 5 5 5 6 8 7 2	1.188000

Αλγόριθμος	Πρόβλημα	Κόστος	Αναθέσεις	Χρόνος
Αποδοχής Κατωφλιού	1	990	10 8 1 4 6 3 6 1 2 4 7 9 6 9 4 8 8 5 3 1 7 9 6 7 3 8 10 5 1 5 6 5 10 2 4 2 5 7 8 6 4 9 9 10 3 4 2 4 2 10 9 5 1 3 2 1 8 8 7 10	0.812000
	2	972	1 10 10 5 9 2 4 1 10 2 7 8 2 7 1 3 5 3 6 9 3 3 1 7 6 2 7 3 6 6 2 5 10 9 10 7 4 9 5 4 8 6 8 3 4 1 1 8 8 2 5 3 9 4 8 4 7 3 9 6	0.797000
	3	947	6 3 2 9 9 5 6 3 9 5 3 3 5 6 2 3 1 1 6 5 7 7 8 4 1 5 10 5 8 1 7 2 8 9 2 2 7 4 7 10 9 1 8 4 1 2 8 6 4 8 6 4 7 1 3 2 1 10 6 4	0.813000
	4	965	6 9 2 9 6 8 5 9 4 9 10 3 4 4 3 10 7 3 2 10 4 7 7 4 2 3 9 1 4 9 2 3 9 10 6 8 1 6 8 6 5 7 1 3 4 7 8 5 4 1 10 1 3 6 7 6 10 6 8 7	0.812000
	5	957	2 3 7 6 3 1 7 2 6 2 9 8 8 9 6 6 9 10 5 10 1 5 1 3 8 5 4 9 3 4 10 5 10 8 2 4 7 4 9 4 4 8 5 2 6 7 1 9 6 3 1 9 10 10 3 5 6 8 1 2	0.813000
Μεταβλητής Γειτονιάς Αναζήτησης	1	1020	6 3 1 1 8 3 3 5 4 6 9 5 6 9 5 10 10 5 3 1 2 9 4 7 4 8 10 10 1 2 6 10 10 2 4 2 8 10 8 2 6 8 2 7 7 4 3 8 1 10 9 5 1 3 4 7 8 8 7 4	0.140000
	2	996	1 6 10 5 8 2 4 1 10 2 3 6 4 7 7 3 9 10 2 9 7 2 9 2 9 2 7 3 6 3 6 5 10 9 8 7 4 9 5 4 1 10 8 1 4 7 1 9 8 5 6 3 8 4 8 5 7 3 1 5	0.156000
	3	988	6 7 2 1 4 5 6 3 1 5 9 5 10 10 2 3 7 6 2 5 8 7 8 4 1 2 4 6 7 1 7 2 8 1 3 2 4 9 7 10 9 1 8 4 1 10 8 8 4 8 6 9 5 10 10 2 3 9 6 4	0.141000
	4	1004	10 9 10 9 6 7 6 9 5 3 10 3 4 4 10 9 7 3 3 6 4 3 7 9 8 6 8 3 4 9 2 3 2 2 7 8 8 4 8 1 5 7 4 1 2 9 2 5 7 1 10 4 3 5 8 6 10 6 1 7	0.156000
	5	985	2 3 9 6 3 7 2 2 4 3 3 8 4 9 7 6 9 8 5 10 8 3 1 6 9 4 6 9 3 10 10 4 7 8 1 7 9 4 9 4 7 4 5 7 10 1 1 2 6 8 1 7 10 5 5 2 6 8 7 2	0.157000

Βιβλιογραφία

- [1] Silvano Martello and Paolo Toth. *Knapsack problems, algorithms and computer implementations*. John Wiley and Sons, 1st edition, 1990.
- [2] G. Terry Ross and Richard Soland. A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming*, 8:91–103, 12 1975.
- [3] Κωνσταντίνος Κουνετάς and Νικόλαος Χατζησταμούλου. *Εισαγωγή στην επιχειρησιακή έρευνα και στον γραμμικό προγραμματισμό. Λύσεις προβλημάτων με το πρόγραμμα R*. Εκδόσεις Κάλλιπος, 1 edition, 2015.
- [4] Kashi R. Balachandran and Suresh Radhakrishnan. Cost of congestion, operational efficiency and management accounting. *European Journal of Operational Research*, 89(2):237–245, 1996.
- [5] Marshall L. Fisher and Ramchandran Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.
- [6] G. Mazzola, H.-G. Wieser, V. Brunner, and D. Muzzolini. A symmetry-oriented mathematical model of classical counterpoint and related neurophysiological investigations by depth eeg. *Computers & Mathematics with Applications*, 17(4):539–594, 1989. Symmetry 2.
- [7] G. Terry Ross and Richard M. Soland. Modeling facility location problems as generalized assignment problems. *Management Science*, 24:345–357, 1977.
- [8] David W. Pentico. Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176(2):774–793, 2007.
- [9] Dirk G. Cattrysse and Luk N. Van Wassenhove. A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research*, 60(3):260–272, 1992.
- [10] Robert M. Nauss. Solving the generalized assignment problem: An optimizing and heuristic approach. *INFORMS Journal on Computing*, 15:249–266, 2003.
- [11] Sven O Krumke and Clemens Thielen. The generalized assignment problem with minimum quantities. *European Journal of Operational Research*, 228(1):46–55, 2013.
- [12] H. W. Kuhn. Variants of the hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 3(4):253–258, 1956.
- [13] Richard E. Stone and Craig A. Tovey. The simplex and projective scaling algorithms as iteratively reweighted least squares methods. *SIAM Rev.*, 33(2):220–237, May 1991.

-
- [14] I. Μαρινάκης, M. Μαρινάκη, N. Ματσατσίνης, and K. Ζοπουνίδης. *Μεθευρετικοί και εξελικτικοί αλγόριθμοι σε προβλήματα διοικητικής επιστήμης*. Κλειδάριθμος, 1 edition, 2017.
- [15] Παναγιώτης Καρακώστας. Υλοποίηση ενός αλγόριθμου βασισμένου στη γενική αναζήτηση μεταβλητής γειτονιάς για την επίλυση του προβλήματος του πλανόδιου πωλητή. Master's thesis, Πανεπιστήμιο Μακεδονίας, 2014.
- [16] Σπυρίδων Σπηλιόπουλος. Μέθοδοι επίλυσης του προβλήματος δρομολόγησης οχημάτων αστικών εμπορευματικών μεταφορών και εφαρμογή τους σε πληροφοριακό σύστημα. Master's thesis, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Μηχανολόγων Μηχανικών, 2019.
- [17] Στέλλα Νικολακάκη. Αλγόριθμος άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης για το πρόβλημα δρομολόγησης οχημάτων σε περιορισμένη απόσταση. Master's thesis, Πολυτεχνείο Κρήτης, Σχολή Μηχανικών Παραγωγής και Διοίκησης, 2015.
- [18] Ειρήνη Τσαμπουνάρη. Αλγόριθμος βελτιστοποίησης ζευγαρώματος μελισσών για το ανοιχτό πρόβλημα δρομολόγησης οχημάτων. Master's thesis, Πολυτεχνείο Κρήτης, Σχολή Μηχανικών Παραγωγής και Διοίκησης, 2017.
- [19] Fred Glover. Tabu search—part i. *ORSA Journal on Computing*, 1:190–206, 1989.
- [20] Fred Glover. Tabu search—part ii. *ORSA Journal on Computing*, 2:4–32, 1990.
- [21] J E Beasley. Welcome to or-library. <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>. [Online; accessed 10-October-2021].
- [22] Pedro M. Talaván and Javier Yáñez. The generalized quadratic knapsack problem. a neuronal network approach. *Neural Networks*, 19(4):416–428, 2006.
- [23] A. Volgenant. Solving the k-cardinality assignment problem by transformation. *European Journal of Operational Research*, 157(2):322–331, September 2004.
- [24] Lorenzo Gatto. lorenzogatto/gapsolvers: Metaheuristics for the generalized assignment problem written in javascript. <https://github.com/lorenzogatto/GAPsolvers>. [Online; accessed 10-October-2021].
- [25] David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62:461–474, 1993.
- [26] Kurt Jörnsten and Peter Värbrand. A hybrid algorithm for the generalized assignment problem. *Optimization*, 22:273–282, 01 1991.
- [27] Mehmet Tasgetiren, Ponnuthurai Suganthan, Tay Chua, and Abdullah Al Hajri. Differential evolution algorithms for the generalized assignment problem. pages 2606–2613, 05 2009.

