

Πανεπιστήμιο Θεσσαλίας Πολυτεχνική Σχολή

Τμήμα Μηχανολόγων Μηχανικών



Διπλωματική Εργασία

Στρατηγικές μείωσης του υπολογιστικού φόρτου για την επίλυση του master προβλήματος σε αλγόριθμους δημιουργίας στηλών

Επίθετο: Λιανός

Όνομα: Μάριος – Δημήτριος

A. M: 01976

Επίθετο: Παπαδάκης

Όνομα: Σταύρος

A. M: 01492

Επιβλέπων Καθηγητής: Κοζανίδης Γεώργιος

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ

1^ο Μέλος Επιτροπής (Επιβλέπων): Αναπληρωτής Καθηγητής Γεώργιος Κοζανίδης
Τμήμα Μηχανολόγων Μηχανικών

2^ο Μέλος Επιτροπής: Καθηγητής Παντελής Δημήτριος
Τμήμα Μηχανολόγων Μηχανικών

3^ο Μέλος Επιτροπής: Καθηγητής Μποντόζογλου Βασίλης
Τμήμα Μηχανολόγων Μηχανικών

Περίληψη

Στο πλαίσιο της διπλωματικής, θα γίνει εισαγωγή στην εξέλιξη του τομέα της αεροπορικής βιομηχανίας αλλά και σε μεθοδολογίες που εφαρμόζονται σε κώδικα για την βελτιστοποίηση του χρονοπρογραμματισμού ιπταμένου προσωπικού, με σκοπό την εξοικείωση του αναγνώστη με το πρόβλημα. Στη συνέχεια θα αναπτυχθεί συμπληρωματικός κώδικας με στόχο τη σύγκριση δύο εναλλακτικών στρατηγικών για τη μείωση του υπολογιστικού φόρτου που απαιτείται για την επίλυση του master προβλήματος σε αλγόριθμους δημιουργίας στηλών. Οι στρατηγικές που θα εξεταστούν είναι οι εξής: Αφαίρεση από το master πρόβλημα στηλών με βάση το ευκαιριακό κόστος τους. Στη συγκεκριμένη διαδικασία εξάγονται τα ευκαιριακά κόστη των μεταβλητών. Έπειτα κατανέμονται ανάλογα με τα αρνητικά τους κόστη, και τέλος βάσει ενός ποσοστού που έχει ορισθεί από τον χρήστη διαγράφεται ένας αριθμός μεταβλητών από το master πρόβλημα. Η δεύτερη στρατηγική αφορά την επιλογή για προσθήκη στο master πρόβλημα ενός υποσυνόλου στηλών μεταξύ αυτών που έχουν δημιουργηθεί από το υποπρόβλημα δημιουργίας στηλών με βάση την επίλυση κατάλληλα διαμορφωμένου προβλήματος κάνοντας χρήση κατάλληλων κριτηρίων “ομοιότητας/ποικιλίας”. Τέλος θα παρουσιαστούν αποτελέσματα και θα πραγματοποιηθεί σύγκριση των χρόνων εκτέλεσης του αρχικού κώδικα με τους χρόνους μετά την εισαγωγή κάθε διαδικασίας ξεχωριστά, ώστε να διαπιστωθεί κατά πόσο η εισαγωγή τους βελτίωσε τον υπολογιστικό φόρτο του προβλήματος.

Περιεχόμενα

1.	Εισαγωγή.....	1
1.1	Ιστορική Αναδρομή	1
1.1	Η Επιχειρησιακή Έρευνα στις Αερομεταφορές	3
1.2	Χρονοπρογραμματισμός Αεροσκαφών και Ιπταμένων.....	3
1.3	Γενική Περιγραφή του Προβλήματος.....	4
2.	Βιβλιογραφική Ανασκόπηση.....	5
2.1	Μέθοδοι βελτιστοποίησης και Αλγόριθμοι	6
2.1.1	Δημιουργία Στηλών (Column Generation)	6
2.1.2	Branch & Bound	6
2.1.3	Branch and Cut.....	7
2.1.4	Branch & Price.....	7
2.1.5	Ranking Paths Algorithm	7
3.	Ανάπτυξη Μοντέλου.....	8
3.1	Master Problem.....	9
3.1.1	Σύγκριση Set-Partition και Set-Cover	9
3.1.2	Set-Partition Μορφοποίηση.....	10
3.1.3	Set-Cover Μορφοποίηση.....	11
4.	Μείωση Υπολογιστικού Φόρτου	12
4.1	Διαδικασία 1: Αφαίρεση από το Master Πρόβλημα Στηλών με Βάση το Ενκαιριακό κόστος τους.....	13
4.1.1	Περιγραφή της Διαδικασίας 1	13
4.2	Διαδικασία 2: Επιλογή για Προσθήκη στο Master Πρόβλημα ενός Υποσυνόλου Στηλών με Βάση Κατάλληλων Κριτηρίων Ομοιότητας.....	17
4.2.1	Περιγραφή της Διαδικασίας 2	17
5.	Αποτελέσματα Υπολογισμών	20
5.1	Line 1 Set-Cover	21
5.1.1	Σύγκριση Αποτελεσμάτων Set-Cover [C1]: Table	21
5.1.2	Σύγκριση Αποτελεσμάτων Set-Cover [C1]: Line Chart	21
5.1.3	Σύγκριση Αποτελεσμάτων Set-Cover [C1]: Bar Chart.....	22
5.1.4	Σύγκριση Αποτελεσμάτων Set-Cover [C2]: Table	23
5.1.5	Σύγκριση Αποτελεσμάτων Set-Cover [C2]: Line Chart	23
5.1.6	Σύγκριση Αποτελεσμάτων Set-Cover [C2]: Bar Chart.....	24

5.1.7	Σύγκριση Αποτελεσμάτων Set-Cover [C3]: Table	25
5.1.8	Σύγκριση Αποτελεσμάτων Set-Cover [C2]: Line Chart	25
5.1.9	Σύγκριση Αποτελεσμάτων Set-Cover [C3]: Bar Chart.....	26
5.2	Line 2 Set-Partition	27
5.2.1	Σύγκριση Αποτελεσμάτων Set-Partition [C1]: Table	27
5.2.2	Σύγκριση Αποτελεσμάτων Set-Partition [C1]: Line Chart	27
5.2.3	Σύγκριση Αποτελεσμάτων Set-Partition [C1]: Bar Chart.....	28
5.2.4	Σύγκριση Αποτελεσμάτων Set-Partition [C2]: Table	29
5.2.5	Σύγκριση Αποτελεσμάτων Set-Partition [C2]: Line Chart	29
5.2.6	Σύγκριση Αποτελεσμάτων Set-Partition [C2]: Bar Chart.....	30
5.2.7	Σύγκριση Αποτελεσμάτων Set-Partition [C3]: Table	31
5.2.8	Σύγκριση Αποτελεσμάτων Set-Partition [C3]: Line Chart	31
5.2.9	Σύγκριση Αποτελεσμάτων Set-Partition [C3]: Bar Chart.....	32
5.3	Line 1 Set-Cover $\bar{x}_{IP}$	33
5.3.1	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C1]: Table	33
5.3.2	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C1]: Bar Chart.....	33
5.3.3	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C2]: Table	34
5.3.4	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C2]: Table	34
5.3.5	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C3]: Table	35
5.3.6	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C3]: Bar Chart.....	35
5.4	Line 1 Set-Partition $\bar{x}_{IP}$	36
5.4.1	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C1]: Table.....	36
5.4.2	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C1]: Bar Chart.....	36
5.4.3	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C2]: Table.....	37
5.4.4	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C2]: Bar Chart.....	37
5.4.5	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C3]: Table.....	38
5.4.6	Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C3]: Bar Chart.....	38
5.5	Σχολιασμός Αποτελεσμάτων.....	39
5.5.1	Line 1 Set-Cover	39
5.5.2	Line 2 Set-Partition	39
5.6	Συμπεράσματα	40
6.	Μελλοντική Έρευνα	41
7.	Βιβλιογραφία	42

Λίστα εικόνων

Εικόνα 1.1: Αεροπορικές Μεταφορές – Μεταφορά επιβατών 1970 – Σήμερα(Σε δισεκατομμύρια) (Πηγή: International Civil Aviation Organization, Civil Aviation Statistics of the World and ICAO staff estimates, https://data.worldbank.org/indicator/IS.AIR.PSGR)	1
Εικόνα 1.2: Αεροπορικές Μεταφορές – Μεταφορά φορτίου εμπορικών πτήσεων (Εκατομμύρια τόνοι-Χιλιόμετρο). (Πηγή: International Civil Aviation Organization, Civil Aviation Statistics of the World and ICAO staff estimates, https://data.worldbank.org/indicator/IS.AIR.PSGR)	2
Εικόνα 2.1: Column Generation Διάγραμμα Ροής (Πηγή : [13])	6
Εικόνα 3.1: Set-Partition pairing (Πηγή : [13])	9
Εικόνα 3.2: Set-Cover pairing (Πηγή : [13])	9
Εικόνα 3.3: Set-Partition Formulation (Πηγή : [13]).....	10
Εικόνα 3.4: Set-Cover Formulation (Πηγή : [13]).....	11
Εικόνα 4. 1: Πίνακας δυικών τιμών.....	13
Εικόνα 4.2: Δημιουργία πίνακα ευκαιριακό κόστους.....	14
Εικόνα 4.3: Ταξινόμηση πίνακα με βάση το ευκαιριακό κόστος	15
Εικόνα 4.4 : Διαγραφή στηλών με βάση την τιμή του “perc”	16
Εικόνα 4.5: Σύνοψη Διαδικασίας 1.....	16
Εικόνα 4. 6 Κώδικας Διαδικασίας 2	18
Εικόνα 4. 7 Συνέχεια κώδικα	19
Εικόνα 5.1: Solution time comparison [C1] Line 1 Set-Cover Table.....	21
Εικόνα 5.2: Solution time comparison [C1] Line 1 Set-Cover Line Chart.....	21
Εικόνα 5.3: Solution time comparison [C1] Line 1 Set-Cover Bar Chart	22
Εικόνα 5.4: Solution time comparison [C2] Line 1 Set-Cover Table.....	23
Εικόνα 5.5: Solution time comparison [C2] Line 1 Set-Cover Line Chart.....	23
Εικόνα 5.6: Solution time comparison [C2] Line 1 Set-Cover Bar Chart	24
Εικόνα 5.7: Solution time comparison [C3] Line 1 Set-Cover Table.....	25
Εικόνα 5.8: Solution time comparison [C3] Line 1 Set-Cover Line Chart.....	25
Εικόνα 5.9: Solution time comparison [C3] Line 1 Set-Cover Bar Chart	26
Εικόνα 5.10: Solution time comparison [C1] Line 2 Set-Partition Table	27
Εικόνα 5.11: Solution time comparison [C1] Line 2 Set-Partition Line Chart.....	27
Εικόνα 5.12: Solution time comparison [C1] Line 2 Set-Partition Bar Chart	28
Εικόνα 5.13: Solution time comparison [C2] Line 2 Set-Partition Table	29
Εικόνα 5.14: Solution time comparison [C2] Line 2 Set-Partition Line Chart.....	29
Εικόνα 5.15: Solution time comparison [C2] Line 2 Set-Partition Bar Chart	30
Εικόνα 5.16: Solution time comparison [C3] Line 2 Set-Partition Table	31
Εικόνα 5.17: Solution time comparison [C3] Line 2 Set-Partition Line Chart.....	31
Εικόνα 5.18: Solution time comparison [C3] Line 2 Set-Partition Bar Chart	32
Εικόνα 5.19: Line 1 Set-Cover [C1]max Table	33

Εικόνα 5.20: Line 1 Set-Cover [C3]max Bar Chart.....	33
Εικόνα 5.21: Line 1 Set-Cover [C2]max Table	34
Εικόνα 5.22: Line 1 Set-Cover [C2]max Bar Chart.....	34
Εικόνα 5.23: Line 1 Set-Cover [C3]max Table	35
Εικόνα 5.24: Line 1 Set-Cover [C3]max Bar Chart.....	35
Εικόνα 5.25: Line 2 Set-Partition [C1]max Table	36
Εικόνα 5.26: Line 2 Set-Partition [C1]max Table	36
Εικόνα 5.27: Line 2 Set-Partition [C2]max Table	37
Εικόνα 5.28: Line 2 Set-Partition [C2]max Table	37
Εικόνα 5.29: Line 2 Set-Partition [C3]max Table	38
Εικόνα 5.30: Line 2 Set-Partition [C3]max Bar Chart.....	38
Εικόνα 5.31: Συντελεστές σύγκρισης διαδικασιών Line 1	39
Εικόνα 5.32: Συντελεστές σύγκρισης διαδικασιών Line 2	39

*Στρατηγικές μείωσης του υπολογιστικού φόρτου για την επίλυση του master προβλήματος
σε αλγόριθμους δημιουργίας στηλών*

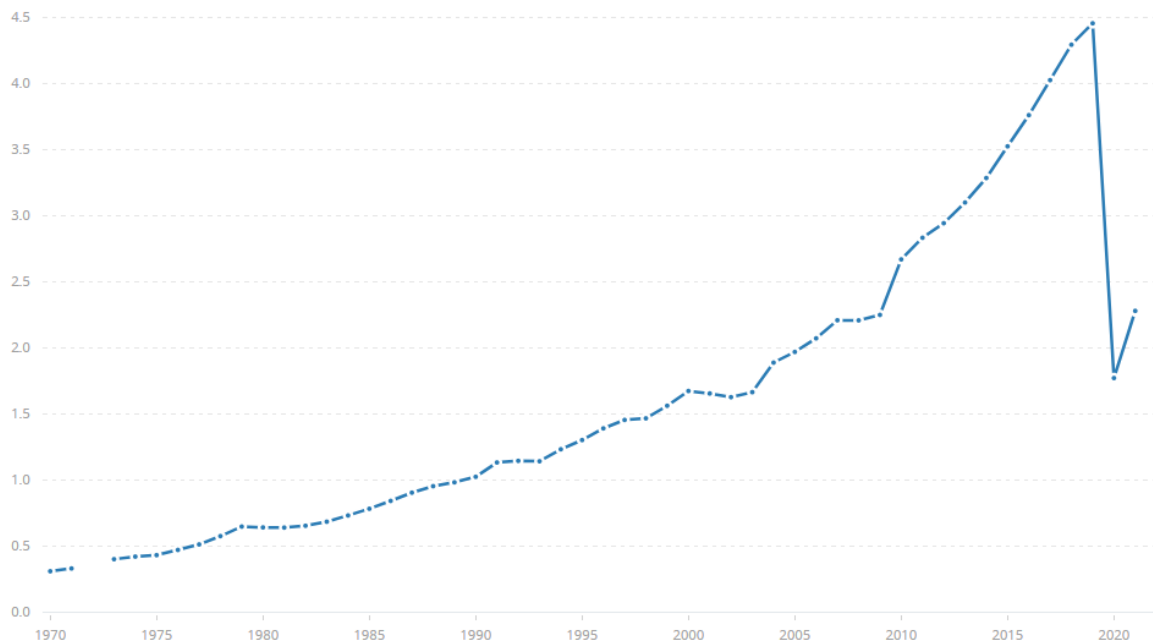
Κεφάλαιο 1

1. Εισαγωγή

1.1 Ιστορική Αναδρομή

Η αεροπορική βιομηχανία έχει γνωρίσει σημαντική εξέλιξη κατά τη διάρκεια του 20ού και 21ου αιώνα. Από την πρώτη πτήση των αδελφών Wright το 1903 μέχρι σήμερα, ο τομέας έχει μετασχηματιστεί σε έναν από τους κυριότερους τομείς της παγκόσμιας οικονομίας. Οι σημαντικότερες εξελίξεις περιλαμβάνουν την τεχνολογική πρόοδο, την ανάπτυξη των αεροπορικών εταιρειών, την αύξηση της αεροπορικής κυκλοφορίας και τις βελτιώσεις στον τομέα της αεροπορικής ασφάλειας.[1]

Στις αρχές του 20ού αιώνα, οι αεροπορικές εταιρείες επικεντρώθηκαν κυρίως στην αμυντική βιομηχανία, κατασκευάζοντας μικρά αεροσκάφη για στρατιωτική χρήση. Κατά τη διάρκεια του Πρώτου Παγκοσμίου Πολέμου, η τεχνολογική πρόοδος οδήγησε στην κατασκευή μεγαλύτερων και πιο προηγμένων αεροσκαφών, ενώ οι πτήσεις άρχισαν να χρησιμοποιούνται και για ανθρωπιστικούς σκοπούς. Μετά τον Πρώτο Παγκόσμιο Πόλεμο, η αεροπορία ανέπτυξε μια νέα διάσταση με την έναρξη εμπορικών πτήσεων. Μεγάλες αεροπορικές εταιρείες όπως η Boeing, η Douglas και η Lockheed ανέπτυξαν προηγμένα αεροσκάφη που επέτρεπαν ταξίδια σε μακρινούς προορισμούς. Οι πτήσεις αποτέλεσαν σύντομα έναν γρήγορο και αποτελεσματικό τρόπο μεταφοράς επιβατών και εμπορευμάτων, επιτρέποντας τη σύνδεση των ανθρώπων και των οικονομιών σε παγκόσμιο επίπεδο.

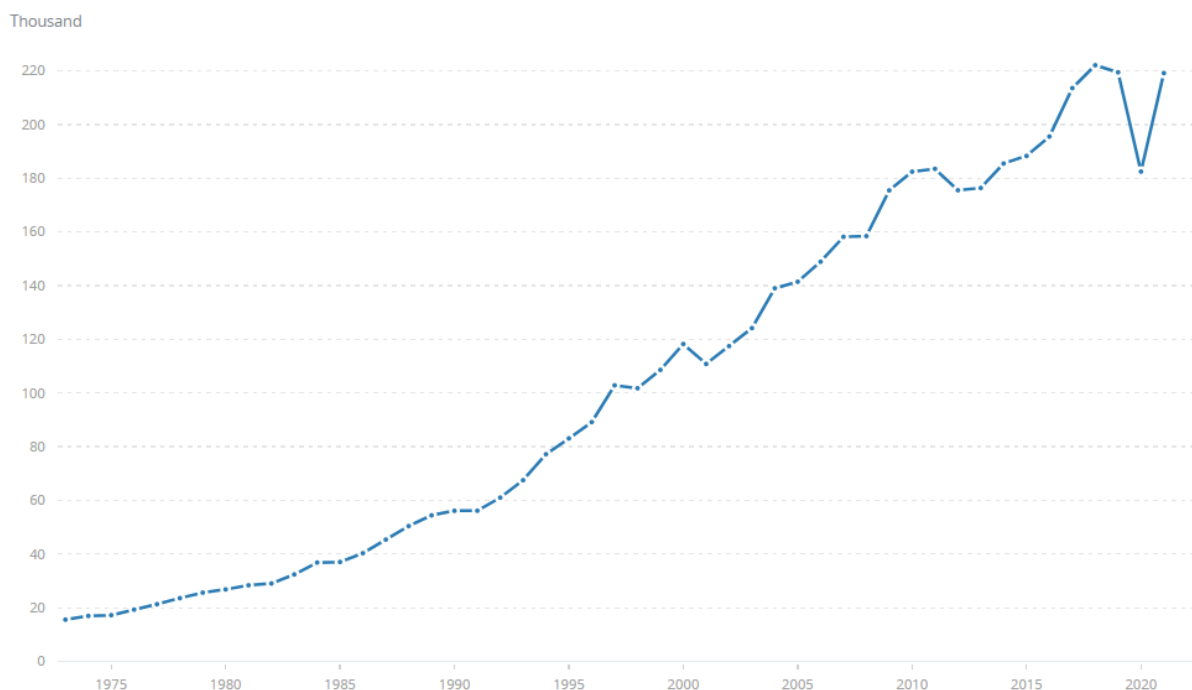


Εικόνα 1.1: Αεροπορικές Μεταφορές – Μεταφορά επιβατών 1970 – Σήμερα(Σε δισεκατομμύρια)
(Πηγή: International Civil Aviation Organization, Civil Aviation Statistics of the World and ICAO staff estimates, <https://data.worldbank.org/indicator/IS.AIR.PSGR>)

Κατά τη δεκαετία του '70 , η αεροπορική βιομηχανία βίωσε μεγάλες αλλαγές. Εισήχθησαν νέες τεχνολογίες και αεροσκάφη που επέτρεψαν μεγαλύτερες ταχύτητες, αυξημένη χωρητικότητα και μειωμένο κόστος λειτουργίας. Το αεροσκάφος Concorde, το οποίο μπορούσε να πετάξει με υπερηχητικές ταχύτητες, κυκλοφόρησε και ξεκίνησε εμπορικές πτήσεις. Ωστόσο, παρά την επιτυχία του, η λειτουργία του Concorde διακόπηκε .

Η εισαγωγή στην δεκαετία του '80 , των αεροσκαφών τύπου Boeing 747 και Airbus A300 άνοιξε νέους ορίζοντες στην εμπορική αεροπορία, επιτρέποντας στις μεγάλες αεροπορικές εταιρείες να προσφέρουν μεγαλύτερη χωρητικότητα και να επεκτείνουν τις διεθνείς τους πτήσεις (Εικόνα 1.2). Επίσης, η δεκαετία αυτή είδε την εμφάνιση της πρώτης γενιάς αυτόνομων πολυκινητήριων αεροσκαφών, όπως το Boeing 757 και το Airbus A320.

Τις επόμενες δεκαετίες έως και σήμερα , η αεροπορική βιομηχανία αντιμετώπισε μεγάλες αλλαγές λόγω της ραγδαίας παγκόσμιας οικονομικής ανάπτυξης. Εισήχθησαν νέα αεροσκάφη όπως το Boeing 777 και 737 MAX , το Airbus A350 και A330, τα οποία προσέφεραν αυξημένη απόδοση και καλύτερη οικονομία καυσίμων. Πλέον, η βιομηχανία επενδύει σε πράσινες τεχνολογίες με στόχο τη μείωση των εκπομπών αερίων του θερμοκηπίου και τη βελτίωση της αποδοτικότητας των αεροσκαφών.[11]



Εικόνα 1.2: Αεροπορικές Μεταφορές – Μεταφορά φορτίου εμπορικών πτήσεων (Εκατομμύρια τόνοι-Χιλιόμετρο). (Πηγή: International Civil Aviation Organization, Civil Aviation Statistics of the World and ICAO staff estimates, <https://data.worldbank.org/indicator/IS.AIR.PSGR>)

1.1 Η Επιχειρησιακή Έρευνα στις Αερομεταφορές

Στην πάροδο του χρόνου οι εταιρείες της αεροπορικής βιομηχανίας αντιμετώπισαν προκλήσεις σε πολλούς τομείς της επιχείρησής τους. Αυτές οι προκλήσεις περιλάμβαναν την ανάγκη για αποτελεσματική διαχείριση της παραγωγής, την ανάπτυξη νέων τεχνολογιών και προϊόντων, τη βελτίωση της ασφάλειας και τη μείωση του κόστους. Για την αντιμετώπιση αυτών των προκλήσεων, η επιχειρησιακή έρευνα έπαιξε σημαντικό ρόλο στην αεροπορική βιομηχανία. Η επιχειρησιακή έρευνα αφορά τη μελέτη και την ανάλυση των επιχειρηματικών διαδικασιών, την εξεύρεση τρόπων βελτίωσης της απόδοσης και την εφαρμογή αποτελεσματικών λύσεων. Παραδείγματα περιλαμβάνουν την ανάλυση και βελτιστοποίηση των αεροπορικών δρομολογίων, τον σχεδιασμό και τη βελτιστοποίηση των αεροσκαφών και των κινητήρων τους, την ανάπτυξη και τη βελτίωση των αεροπορικών συστημάτων ελέγχου και πληροφορικής, καθώς και την ανάπτυξη και την εφαρμογή νέων τεχνολογιών, όπως οι αεροδυναμικές βελτιώσεις και οι αεροπορικές υπηρεσίες των μη επανδρωμένων αεροσκαφών.[2]

Η βελτιστοποίηση των αεροπορικών δρομολογίων και του χρονοπρογραμματισμού του ιπτάμενου προσωπικού (Aircraft and Crew Schedule Planning) θέτει ως στόχο την αποτελεσματική εφαρμογή της επιχειρησιακής έρευνας στο σημείο αυτό. Ένα μεγάλο και σύνθετο κομμάτι που αποτελεί παράγοντα υψηλού κόστους. Για το λόγο αυτό, γίνεται χρήση μαθηματικών μοντέλων, αλγορίθμων προσομοίωσης και άλλων προηγμένων τεχνικών για την εύρεση της βέλτιστης λύσης.

1.2 Χρονοπρογραμματισμός Αεροσκαφών και Ιπταμένων

Αποτελεί ένα μείζον ζήτημα για τις αεροπορικές εταιρείες, καθώς επηρεάζει την αποδοτικότητα, την ασφάλεια και το κόστος των πτήσεων τους. Το πρόβλημα αυτό είναι πολύπλοκο και περιλαμβάνει πολλούς παράγοντες που πρέπει να ληφθούν υπόψη.

Κάποιοι από αυτούς τους παράγοντες, είναι η ασφάλεια, οι κανονισμοί της αεροπορίας, οι χρόνοι ανάπαυσης του προσωπικού, οι προτιμήσεις των επιβατών, οι οικονομικοί περιορισμοί και οι απαιτήσεις για την αποδοτική λειτουργία της εταιρείας.

Ο σχεδιασμός αυτός απαιτεί τη χρήση εξειδικευμένων λογισμικών και αλγορίθμων βελτιστοποίησης για την επίτευξη του βέλτιστου αποτελέσματος. Όμως, μπορεί να διαιρεθεί σε ένα σύνολο υποπροβλημάτων[2] με σκοπό να οριστεί καλύτερα το πρόβλημα, όπως παρουσιάζεται παρακάτω :

1. Προγραμματισμός πτήσεων: Αρχικά, πρέπει να γίνει ο προγραμματισμός των πτήσεων για ένα συγκεκριμένο χρονικό διάστημα. Αυτό περιλαμβάνει τον καθορισμό των πτήσεων που πρέπει να δρομολογηθούν, των αεροδρομίων αναχώρησης και άφιξης, και του χρόνου διάρκειας κάθε πτήσης.

2. Δρομολόγηση της συντήρησης των αεροσκαφών: Καθορισμός του τρόπου δρομολόγησης των αεροσκαφών για να εξασφαλιστεί η συντήρηση των αεροσκαφών βάσει προγράμματος.

3. Αντιστοίχιση αεροσκαφών: Έπειτα, πρέπει να γίνει η αντιστοίχιση των αεροσκαφών σε κάθε πτήση, λαμβάνοντας υπόψη τις απαιτήσεις κάθε πτήσης (όπως οι διαθέσιμες θέσεις και οικονομικές κλάσεις) και τις δυνατότητες και περιορισμούς των αεροσκαφών (όπως οι αποστάσεις πτήσης και οι τεχνικές προδιαγραφές).

4. Χρονοπρογραμματισμός πληρώματος: Τέλος, πρέπει να γίνει η αντιστοίχιση των μελών του πληρώματος (πιλότοι, αεροσυνοδοί κ.λπ.) σε κάθε πτήση, λαμβάνοντας υπόψη τις

απαιτούμενες ικανότητες, πιστοποιήσεις και περιορισμούς του προσωπικού με στόχο την ελαχιστοποίηση του κόστους.[3]

1.3 Γενική Περιγραφή του Προβλήματος

Το πρόβλημα με το οποίο ασχολούμαστε αποτελεί ένα ρεαλιστικό κομμάτι του προγραμματισμού μιας αεροπορικής εταιρίας. Στόχος είναι η βελτιστοποίηση της ανάθεσης ιπτάμενου προσωπικού, για την πραγματοποίηση των δρομολογίων αεροπορικών εταιριών. Κάθε αεροσκάφος βρίσκεται στη βάση του (αρχική τοποθεσία της πτητικής διαδρομής), έχει συγκεκριμένη ώρα αναχώρησης, ενώ γίνεται καθορισμός και των ενδιάμεσων στάσεων που θα εκτελέσει. Τελικό προορισμό κάθε πτητικής διαδρομής αποτελεί πάντοτε η βάση της. Προτεραιότητα στο πρόβλημα μας είναι να καλυφθούν όσο το δυνατόν περισσότερες πτητικές διαδρομές. Σε περίπτωση που δεν καλυφθεί κάποια διαδρομή αυτό συνεπάγεται μεγάλο κόστος για την αεροπορική εταιρία. Επομένως το πρόγραμμα του πτητικού προσωπικού θα πρέπει να διαμορφωθεί με τέτοιο τρόπο ώστε να αποφευχθεί αυτή η κατάσταση.[12]

Κεφάλαιο 2

2. Βιβλιογραφική Ανασκόπηση

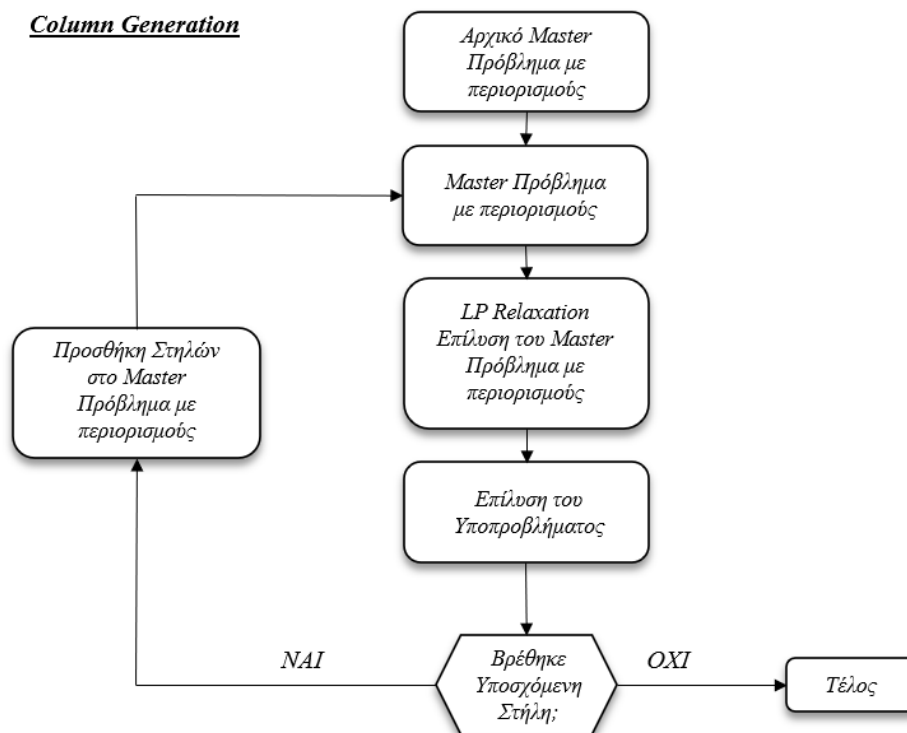
Μέθοδοι επιχειρησιακής έρευνας, όπως αναφέρθηκε, χρησιμοποιούνται σε όλο το φάσμα των αεροπορικών δραστηριοτήτων. Στον παρόν κεφάλαιο και για τους σκοπούς της παρούσας εργασίας, θα γίνει αναφορά σε συγκεκριμένα άρθρα, έτσι ώστε ο αναγνώστης να μπορέσει να κατανοήσει περισσότερο τις μεθοδολογίες και τα σημεία του κώδικα του προβλήματος, στα οποία γίνεται η παρέμβαση. Για την υλοποίηση της επιχειρησιακής έρευνας, συνήθως χρησιμοποιούνται προηγμένες τεχνικές ανάλυσης δεδομένων, αλγόριθμοι βελτιστοποίησης, μοντέλα γραμμικού/μη γραμμικού προγραμματισμού, καθώς και τεχνικές τεχνητής νοημοσύνης όπως οι γενετικοί αλγόριθμοι. Παρακάτω θα αναφερθούμε σε μεθόδους βελτιστοποίησης και σε αλγορίθμους που χρησιμοποιούμε στο πρόγραμμα.

2.1 Μέθοδοι βελτιστοποίησης και Αλγόριθμοι

2.1.1 Δημιουργία Στηλών (Column Generation)

Η μεθοδολογία column generation χρησιμοποιείται για την επίλυση προβλημάτων με πολύ μεγάλο αριθμό μεταβλητών απόφασης, εξετάζοντας ένα μικρότερο σύνολο αυτών. Η διαδικασία είναι ένας τρόπος επίλυσης γραμμικών προβλημάτων προγραμματισμού προσθέτοντας στήλες κατά τη διάρκεια καθορισμού τιμών της simplex μεθοδολογίας επίλυσης του προβλήματος.

Η διαδικασία δημιουργίας στηλών (column generation) εκμεταλλεύεται το γεγονός ότι στη μεθοδολογία της simplex, κατά την επίλυση δε χρειάζεται πρόσβαση σε όλες τις μεταβλητές του προβλήματος ταυτόχρονα. Στη πραγματικότητα, για να ξεκινήσει η επίλυση χρειάζεται ένα συγκεκριμένο υποσύνολο μεταβλητών, και ακολούθως χρησιμοποιώντας τις τιμές των ευκαιριακών κοστών καθορίζει ποιες άλλες μεταβλητές θα πάρουν μέρος. Στη [5], πρωτοπόροι στο τομέα των τεχνικών βελτιστοποίησης και ειδικότερα στους αλγορίθμους δημιουργίας στηλών (column generation), παρουσιάζουν πολλές πετυχημένες εφαρμογές αυτής της τεχνικής, σε διάφορα πλαίσια της βιομηχανίας. Στην Εικόνα 2.1, παρουσιάζεται το διάγραμμα ροής της Column Generation.



Εικόνα 2.1: Column Generation Διάγραμμα Ροής (Πηγή : [13])

2.1.2 Branch & Bound

Η βασική δομή του Branch and Bound είναι ένα δένδρο απαρίθμησης. Ο αρχικός κόμβος του δένδρου αντιστοιχεί στο αρχικό πρόβλημα. Καθώς ο αλγόριθμος προχωρά, το δένδρο μεγαλώνει μέσω μιας διαδικασίας που ονομάζεται διακλάδωση, η οποία δημιουργεί δύο ή περισσότερους κόμβους-παιδιά από τον γονικό κόμβο. Κάθε πρόβλημα στους κόμβους-παιδιά δημιουργείται προσθέτοντας περιορισμούς στο πρόβλημα του γονικού κόμβου. Η δομή του

Branch and Bound αποτελεί μια θεμελιώδη μεθοδολογία για τη παραγωγή λύσεων ακριβείας σε προβλήματα βελτιστοποίησης.[9] Χρησιμοποιούνται κανόνες για την αποκοπή περιοχών της αναζήτησης που αποδεικνύεται ότι δεν είναι βέλτιστες. Μόλις εξερευνηθεί ολόκληρο το δέντρο, η αναζήτηση τελειώνει και δίνεται η βέλτιστη λύση.[7]

2.1.3 Branch and Cut

Η μέθοδος του Branch and Cut είναι εξαιρετικά επιτυχής και αποδοτική για την επίλυση μιας πληθώρας προβλημάτων ακέραιου προγραμματισμού, παρέχοντας παράλληλα εγγύηση για τη βελτιστότητα. Η βασική ιδέα του αλγόριθμου Branch and Cut είναι απλή. Η Branch and Cut είναι μια γενίκευση της Branch and Bound με χαλαρώσεις LP και επιτρέπει να εφαρμοστούν διαχωρισμοί και αποκοπές στο δέντρο του Branch and Bound. Το Branching ξεκινάει όταν δεν υπάρχουν παραβιασμένες ανισότητες και μη εφικτές λύσεις.[6]

2.1.4 Branch & Price

Η φιλοσοφία της μεθόδου Branch and Price είναι παρόμοια με αυτή της Branch and Cut, εκτός από το ότι η διαδικασία επικεντρώνεται στη δημιουργία στηλών αντί για τη δημιουργία γραμμών.[8] Η μέθοδος αποτελεί συνδυασμό αλγορίθμων Branch and Bound και Column Generation. Πιο αναλυτικά, η Branch and Price είναι μια μέθοδος διακλάδωσης και δέσμευσης στην οποία σε κάθε κόμβο του δέντρου αναζήτησης, μπορούν να προστεθούν στήλες στο γραμμικό πρόβλημα χαλάρωσης (LP relaxation) ενός κύριου προβλήματος, επιτρέποντας στην διαδικασία δημιουργίας στηλών (Column Generation) να εφαρμόζεται σε όλο το δένδρο Branch and Bound.

Μια σημαντική πηγή σχετικά με τη διαδικασία δημιουργίας στηλών και την ενσωμάτωση της σε λύση π.χ. branch and price, που αντιμετωπίζει προβλήματα προγραμματισμού ακεραίων αριθμών είναι η [4].

2.1.5 Ranking Paths Algorithm

Η λειτουργία του αλγόριθμου, βασίζεται στην δημιουργία ενός διαγράμματος που περιλαμβάνει όλα τα πιθανά μονοπάτια από ένα αρχικό σημείο σε ένα τελικό σημείο. Στη συνέχεια, τα μονοπάτια "κατατάσσονται" με βάση κάποιο κριτήριο, το οποίο συνήθως αντιπροσωπεύει το "κόστος" του μονοπατιού. Η βασική ιδέα του αλγόριθμου είναι η εύρεση της συντομότερης διαδρομής, η οποία καθορίζεται από το δίκτυο με τέτοιο τρόπο, ώστε να μην σχηματιστεί νέα διαδρομή και να μην αποκλείονται άλλα μονοπάτια. Αυτό οδηγεί σε ένα διευρυμένο δίκτυο όπου όλα τα μονοπάτια, αλλά και η συντομότερη διαδρομή, μπορούν να καθοριστούν.[15]

Κεφάλαιο 3

3. *Ανάπτυξη Μοντέλου*

Σε αυτό το κεφάλαιο, θα εξετάσουμε τα μαθηματικά μοντέλα που συμβάλλουν στην κατασκευή μιας αποτελεσματικής διαδικασίας για την επίλυση του προβλήματος χρονοπρογραμματισμού ιπτάμενου προσωπικού μέσω της μεθόδου branch and price. Τα μοντέλα βελτιστοποίησης, δηλαδή τα master problems, έχουν ως στόχο να καλύψουν με τον πιο αποδοτικό τρόπο τον μέγιστο αριθμό ζευγαριών ενός συγκεκριμένου συνόλου, του ιπτάμενου προσωπικού. Θα αναλυθούν οι set-partition και set-cover μορφοποιήσεις που αποτελούν τα master problems. Θα τονιστούν οι διαφορές τους και θα παρουσιαστούν τα προβλεπόμενα πλεονεκτήματα της χρήσης της set-cover φόρμουλας για το πρόβλημα που μελετάμε.

3.1 Master Problem

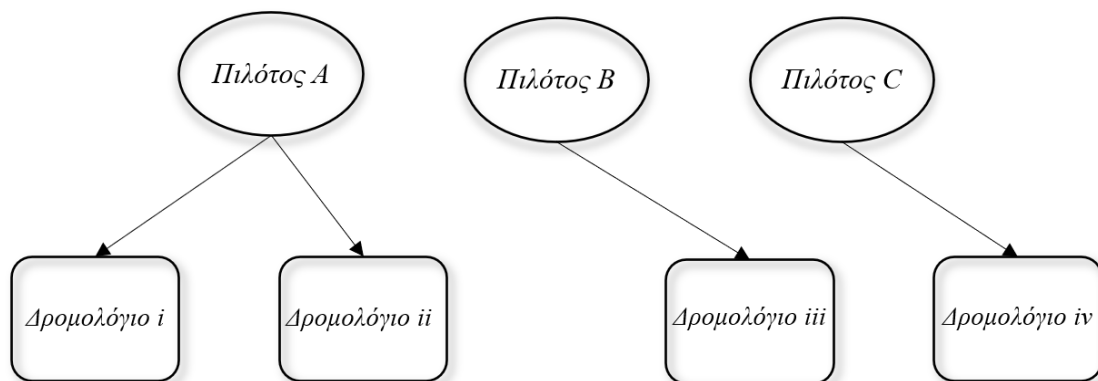
3.1.1 Σύγκριση Set-Partition και Set-Cover

Προτού παρουσιαστούν τα μαθηματικά μοντέλα, χρησιμότερο για τον αναγνώστη θα ήταν να κατανοήσει την διαφορά ανάμεσα στους δύο τύπους προβλημάτων: του set-partition και του set-cover.

Στην περίπτωση του set-partition απαιτούμε να καλύπτεται η κάθε πτητική διαδρομή ακριβώς από τον αριθμό των μελών του πληρώματος που απαιτούνται για την πραγματοποίηση αυτών των πτήσεων. Οι πτήσεις απαιτούν πλήρωμα σε διάφορες θέσεις. Για παράδειγμα, χρειάζεται πάντα ένας κυβερνήτης, ένας συγκυβερνήτης και κάποιος αριθμός αεροσυνοδών.

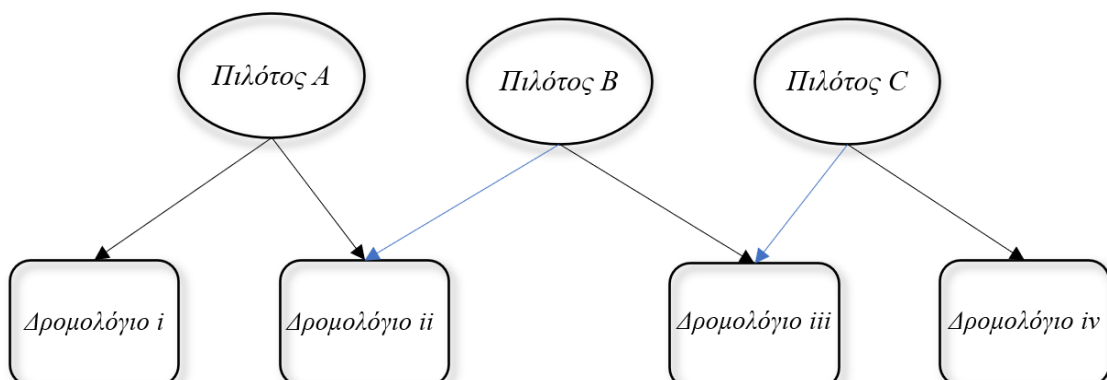
Από την άλλη πλευρά, στο πρόβλημα set-cover επιτρέπουμε την εκχώρηση μια διαδρομής σε περισσότερους πιλότους από τον πραγματικά απαιτούμενο αριθμό. Στην πραγματικότητα αυτό θα σήμαινε ότι ένας πιλότος A και ένας πιλότος B θα εκτελούσαν μια πτήση και οι δύο μαζί ενώ θα χρειαζόταν μόνο ο ένας από τους δύο, κάτι που δεν έχει λογική.

Set Partition



Εικόνα 3.1: Set-Partition pairing (Πηγή : [13])

Set Cover



Εικόνα 3.2: Set-Cover pairing (Πηγή : [13])

3.1.2 Set-Partition Μορφοποίηση

Σύνολα :

I : σύνολο των μελών του πληρώματος (πυλότες , αεροσυννοδοί κλπ.)

R_i : σύνολο των προγραμμάτων για κάθε μέλος του πληρώματος i

F : σύνολο πτητικών διαδρομών

Παράμετροι:

c_{ij} : κόστος του προγράμματος j για το μέλος πληρώματος i

h : κόστος για κάθε πτητική διαδρομή που παραμένει ακάλυπτη.

b_f : αριθμός των μελών του πληρώματος που απαιτούνται για κάθε πτητική διαδρομή f ,

a_{ijf} : δυαδική παράμετρος η οποία παίρνει την τιμή 1 όταν το πρόγραμμα j του πληρώματος i καλύπτει την πτητική διαδρομή f , και τιμή 0 όταν δεν την καλύπτει, όπου $i \in I, j \in R_i, f \in F$

Μεταβλητές Απόφασης:

x_{ij} : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 εαν στο πρόγραμμα j τοποθετείται το μέλος του πληρώματος i , αλλιώς παίρνει την τιμή 0, όπου $i \in I, j \in R_i, f \in F$

y_f : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 εαν η πτητική διαδρομή παραμένει ακάλυπτη , αλλιώς παίρνει την τιμή 0, όπου $f \in F$

Με βάση τον παραπάνω σχεδιασμό ,η μορφοποίηση του master προβλήματος διαμορφώνεται ως εξής:

$$\text{Min} \sum_{i \in I} \sum_{j \in R_i} c_{ij} x_{ij} + \sum_{f \in F} h y_f \quad (1)$$

$$\text{s.t} \quad \sum_{j \in R_i} x_{ij} = 1, \quad \forall i \in I \quad (2)$$

$$y_f + \sum_{i \in I} \sum_{j \in R_i} a_{ij} x_{ij} = b_f, \quad \forall f \in F \quad (3)$$

$$x_{ij} \text{ δυαδική}, y_f \text{ ακέραια}, \quad \forall i, j, f \quad (4)$$

Εικόνα 3.3: Set-Partition Formulation (Πηγή : [13])

Η αντικειμενική συνάρτηση (1) ελαχιστοποιεί το συνολικό κόστος, το οποίο περιλαμβάνει το κόστος των ακάλυπτων ζευγαριών συν το κόστος ποιότητας του προγράμματος. Το σύνολο περιορισμών (2) διασφαλίζει ότι ακριβώς ένα πρόγραμμα εκχωρείται σε κάθε μέλος του πληρώματος. Το σύνολο περιορισμών (3) δηλώνει ότι κάθε πτήση-διαδρομή πρέπει να καλύπτεται ακριβώς από τον απαιτούμενο αριθμό μελών του πληρώματος. Η μεταβλητή y_f γίνεται θετική όταν το δρομολόγιο της πτήσης f δεν καλύπτεται πλήρως, οπότε επιβάλλεται η αντίστοιχη ποινή h . Τέλος το σύνολο των περιορισμών (4) περιορίζει τις μεταβλητές απόφασης του προβλήματος να είναι ακέραιες τιμές.[13]

3.1.3 Set-Cover Μορφοποίηση

Σύνολα :

I : σύνολο των μελών του πληρώματος (πιλότοι , αεροσυννοδοί κλπ.)

R_i : σύνολο των προγραμμάτων για κάθε μέλος του πληρώματος i

F : σύνολο πτητικών διαδρομών

Παράμετροι:

h : κόστος για κάθε πτητική διαδρομή που παραμένει ακάλυπτη.

b_f : αριθμός των μελών του πληρώματος που απαιτούνται για κάθε πτητική διαδρομή f ,

a_{ij} : δυαδική παράμετρος η οποία παίρνει την τιμή 1 όταν το πρόγραμμα j του πληρώματος i καλύπτει την πτητική διαδρομή f , και τιμή 0 όταν δεν την καλύπτει, όπου $i \in I, j \in R_i, f \in F$

Μεταβλητές Απόφασης:

x_{ij} : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 εαν στο πρόγραμμα j τοποθετείται το μέλος του πληρώματος i , αλλιώς παίρνει την τιμή 0, όπου $i \in I, j \in R_i, f \in F$

y_f : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 εαν η πτητική διαδρομή παραμένει ακάλυπτη , αλλιώς παίρνει την τιμή 0, όπου $f \in F$

Με βάση τον παραπάνω σχεδιασμό ,η μορφοποίηση του master προβλήματος διαμορφώνεται ως εξής:

$$\text{Min} \sum_{f \in F} h y_f \quad (5)$$

$$\text{s.t} \sum_{j \in R_i} x_{ij} = 1, \forall i \in I \quad (6)$$

$$y_f + \sum_{i \in I} \sum_{j \in R_i} a_{ij} x_{ij} \geq b_f, \forall f \in F \quad (7)$$

$$x_{ij} \text{ δυαδική}, y_f \text{ ακέραια}, \forall i, j, f \quad (8)$$

Εικόνα 3.4: Set-Cover Formulation (Πηγή : [13])

Υπάρχουν δύο διαφορές μεταξύ του Set-Cover και του Set-Partition. Η πρώτη διαφορά είναι η αντικειμενική συνάρτηση (5). Στη Set-Cover λύση περιλαμβάνεται μόνο το κόστος κάλυψης, ενώ το κόστος ποιότητας παραλείπεται σκόπιμα. Η δεύτερη διαφορά είναι το σύνολο περιορισμών (7). Αντί για ισότητα, εδώ χρησιμοποιούμε ανισότητα ($\geq$). Υπενθυμίζουμε ότι αυτό το σύνολο περιορισμών αφορά την κάλυψη των ζευγαριών.[13]

Κεφάλαιο 4

4. Μείωση Υπολογιστικού Φόρτου

Για την παραπάνω διπλωματική εργασία έχει γραφεί κώδικας ο οποίος χρησιμοποιώντας δύο μαθηματικές μεθόδους, έχει στόχο την μείωση του υπολογιστικού φόρτου εργασίας του προβλήματος. Οι δύο αυτές μέθοδοι επεμβαίνουν στην διαδικασία δημιουργίας στηλών (Column Generation) και με μαθηματικές θεωρίες ακεραίου μαθηματικού προγραμματισμού αφαιρούν μεταβλητές (κορδόνια) του προβλήματος. Η πρώτη διαδικασία αφορά την αφαίρεση από το master πρόβλημα στηλών με βάση το ευκαιριακό κόστος τους. Στη συγκεκριμένη διαδικασία εξάγονται οι δυικές τιμές των μεταβλητών. Έπειτα κατανέμονται ανάλογα με τα αρνητικά τους κόστη, και τέλος βάση ενός ποσοστού που έχει οριστεί από τον χρήστη διαγράφεται ένας αριθμός μεταβλητών από το master πρόβλημα. Η δεύτερη διαδικασία αφορά την επιλογή για προσθήκη στο master πρόβλημα ενός υποσυνόλου στηλών μεταξύ αυτών που έχουν δημιουργηθεί από το υποπρόβλημα δημιουργίας στηλών, κάνοντας χρήση κατάλληλων κριτηρίων “ομοιότητας/ποικιλίας”.

4.1 Διαδικασία 1: Αφαίρεση από το Master Πρόβλημα Στηλών με Βάση το Ευκαιριακό κόστος τους

4.1.1 Περιγραφή της Διαδικασίας 1

Η πρώτη διαδικασία αφορά την αφαίρεση από το master πρόβλημα στηλών με βάση το ευκαιριακό κόστος τους. Στη συγκεκριμένη διαδικασία εξάγονται οι δυικές τιμές των μεταβλητών. Έπειτα κατανομούνται ανάλογα με τα αρνητικά τους κόστη, και τέλος βάση ενός ποσοστού που έχει οριστεί από τον χρήστη διαγράφεται ένας αριθμός μεταβλητών από το master πρόβλημα.

Η παραπάνω διαδικασία λειτουργεί στο εσωτερικό της διαδικασίας δημιουργίας στηλών (Column Generation). Παρακάτω παρουσιάζεται ο κώδικας που δημιουργήθηκε για την πραγματοποίηση της Διαδικασίας 1, και μαζί θα γίνει περιγραφή των συναρτήσεων (bool) που χρησιμοποιήθηκαν για την λειτουργία του.

‘GetMasterDuals’:

Η συνάρτηση ‘GetMasterDuals’ αναλαμβάνει να λάβει τις δυικές τιμές από το τρέχον κύριο πρόβλημα. Αυτό επιτυγχάνεται πρώτα ελέγχοντας αν η λύση βελτιστοποίησης του προβλήματος είναι επιτυχής μέσω της συνάρτησης ‘LPOpt’. Στη συνέχεια, ανακτά τον αριθμό των γραμμών του προβλήματος και δεσμεύει την απαιτούμενη μνήμη για την αποθήκευση των δυικών μεταβλητών. Τέλος, καλεί τη συνάρτηση ‘GetDuals’ για να αποκτήσει τις δυικές μεταβλητές. Εάν κάποιο από τα παραπάνω βήματα αποτύχει, η συνάρτηση επιστρέφει ‘false’, ενώ επιστρέφει true σε περίπτωση επιτυχίας.

```
bool GetMasterDuals(CPXENVptr env, CPXLPptr lp, double** DualArr)
{
    if (!LPOpt(env, lp))
        return false;
    int numrows = CPXgetnumrows(env, lp);
    *DualArr = (double*)malloc(numrows * sizeof(double));
    if (!GetDuals(env, lp, DualArr))
        return false;

    return true;
}
```

Εικόνα 4. 1: Πίνακας δυικών τιμών

‘ComputeReducedCost’:

Η συνάρτηση ‘ComputeReducedCost’ υπολογίζει το ευκαιριακό κόστος για κάθε μεταβλητή του κύριου προβλήματος με βάση τις δυικές μεταβλητές που έχουν ληφθεί από τη συνάρτηση ‘GetMasterDuals’. Αρχικά, ανακτά τον αριθμό των γραμμών και των στηλών του προβλήματος και δεσμεύει την απαιτούμενη μνήμη για την αποθήκευση των μειωμένων κοστών. Στη συνέχεια, πραγματοποιεί μια επανάληψη για κάθε στήλη του προβλήματος και υπολογίζει το ευκαιριακό κόστος με βάση τις δυικές μεταβλητές και τους συντελεστές των μεταβλητών, χρησιμοποιώντας τη συνάρτηση ‘GetVariableCoefficient’. Εάν οι τιμές των ευκαιριακών κοστών είναι θετικές, αυτές αποθηκεύονται στον πίνακα ‘MixedArr’. Η

συνάρτηση επιστρέφει **‘true’** εάν η διαδικασία ολοκληρωθεί με επιτυχία και **‘false’** εάν παρουσιαστεί κάποιο πρόβλημα.

```
bool ComputeReducedCost(CPXENVptr env, CPXLPptr lp, double* DualArr, int* MixedArrSize, CplexCol** MixedArr)
{
    int numrows = CPXgetnumrows(env, lp);
    int numcols = CPXgetnumcols(env, lp);
    int numvars = numcols - (N + R);

    (*MixedArr) = (CplexCol*)malloc(numvars * sizeof(CplexCol));
    double RedCost;

    double coef;
    *MixedArrSize = 0;
    for (int col = N + R; col < numcols; col++) {
        RedCost = 0;
        for (int row = 0; row < numrows; row++) {
            if (!GetVariableCoefficient(env, lp, col, row, &coef))
                return false;
            if (coef > 0.5)
                RedCost = RedCost - DualArr[row];
        }
        if (RedCost < 0) continue;
        (*MixedArrSize)++;
        (*MixedArr)[col - N - R].CplexIdx = col;
        (*MixedArr)[col - N - R].RCost = RedCost;
    }
    return true;
}
```

Εικόνα 4.2: Δημιουργία πίνακα ευκαιριακό κόστους

‘Sort_idx’:

Η συνάρτηση **‘Sort_idx’** είναι υπεύθυνη για την ταξινόμηση των μεταβλητών με βάση το ευκαιριακό κόστος τους. Αρχικά, εκτελεί την ταξινόμηση των μεταβλητών που αποθηκεύτηκαν στον πίνακα **‘MixedArr’** χρησιμοποιώντας την συνάρτηση **‘qsort’** και την συνάρτηση σύγκρισης **‘compRC’**. Στη συνέχεια, υπολογίζει το ελάχιστο αριθμό των μεταβλητών που πρέπει να διαγραφούν και αποθηκεύει τους δείκτες τους στον πίνακα **‘IdxArray’**. Τέλος, ταξινομεί τον πίνακα **‘IdxArray’** χρησιμοποιώντας την συνάρτηση **‘qsort’** και την συνάρτηση σύγκρισης **‘compIdx’**. Η συνάρτηση επιστρέφει true αν η διαδικασία ολοκληρωθεί επιτυχώς και **‘false’** εάν παρουσιαστεί κάποιο πρόβλημα κατά τη διάρκεια της ταξινόμησης.

```

int compRC(const void* elem1, const void* elem2)
{
    double f = (((CplexCol*)elem1)->RCost);
    double s = (((CplexCol*)elem2)->RCost);
    if (f > s) return 1;
    if (f < s) return -1;
    return 0;
}

int compIdx(const void* elem1, const void* elem2)
{
    int f = *((int*)elem1);
    int s = *((int*)elem2);
    if (f < s) return 1;
    if (f > s) return -1;
    return 0;
}

bool Sort_idx(CPXENVptr env, CPXLPptr lp, int** IdxArray, CplexCol** MixedArr, int MixedArrSize)
{
    int numcols = CPXgetnumcols(env, lp);
    int numvars = numcols - (N + R);
    int VarsToDelete = round(numvars * perc);
    bool flag = false;

    qsort(*MixedArr, MixedArrSize, sizeof(CplexCol), compRC);

    int MinVars = std::min(VarsToDelete, MixedArrSize);

    (*IdxArray) = (int*)malloc(MinVars * sizeof(int));

    for (int i = 0; i < MinVars; i++)
    {
        if ((*MixedArr)[MixedArrSize - 1 - i].CplexIdx < N + R)
        {
            VarsToDelete = i;
            flag = true;
            break;
        }
        (*IdxArray)[i] = (*MixedArr)[MixedArrSize - 1 - i].CplexIdx;
    }

    if (flag)
        MinVars = VarsToDelete;
    qsort(*IdxArray, MinVars, sizeof(int), compIdx);

    return true;
}

```

Εικόνα 4.3: Ταξινόμηση πίνακα με βάση το ευκαιριακό κόστος

‘DeleteColumnsReduce’:

Η συνάρτηση **‘DeleteColumnsReduce’** αναλαμβάνει να διαγράψει τις στήλες του προβλήματος που αντιστοιχούν σε μεταβλητές με υψηλότερο ευκαιριακό κόστος, όπως προκύπτει από τον πίνακα **‘IdxArray’**. Αρχικά, αποκτά τον αριθμό των στηλών που πρέπει να διαγραφούν και υπολογίζει το ελάχιστο αριθμό των μεταβλητών που πρέπει να διαγραφούν. Στη συνέχεια, εκτελεί μια επανάληψη για κάθε στήλη που πρέπει να διαγραφεί και καλεί τη συνάρτηση **‘CPXdelcols’** για να διαγράψει τη στήλη από το πρόβλημα. Εάν η διαδικασία διαγραφής είναι επιτυχής, η συνάρτηση συνεχίζει με την επόμενη στήλη, ενώ επιστρέφει **‘false’** εάν παρουσιαστεί κάποιο πρόβλημα. Η συνάρτηση επιστρέφει **‘true’** αν όλες οι στήλες διαγραφούν επιτυχώς.

```

bool DeleteColumnsReduce(CPXENVptr env, CPXLPptr lp, int* IdxArray, int MixedArrSize) {
    int status;
    int numcols = CPXgetnumcols(env, lp);
    int numvars = numcols - (N + R);
    int VarsToDelete = round(numvars * perc);
    int MinVars = std::min(VarsToDelete, MixedArrSize);

    for (int col = 0; col < MinVars; col++)
    {
        status = CPXdelcols(env, lp, IdxArray[col], IdxArray[col]);
        if (status != 0) {
            dsp_cpx_error("CPXdelcols", status);
            return false;
        }
    }
    return true;
}

```

Εικόνα 4.4 : Διαγραφή στηλών με βάση την τιμή του “perc”

Η παραπάνω διαδικασία επιλέγεται από τον χρήστη, στην αρχή του κώδικα ως #define Procedure == 1.

```

if (Procedure == 1) {

    if (PrintProblemToFile(MasterEnv, (*Node)->lp, "bug1.lp") == false)
        return false;
    if (!GetMasterDuals(MasterEnv, (*Node)->lp, &DualArr))
        return false;
    if (!ComputeReducedCost(MasterEnv, (*Node)->lp, DualArr, &MixedArrSize, &MixedArr))
        return false;
    if (!Sort_idx(MasterEnv, (*Node)->lp, &IdxArray, &MixedArr, MixedArrSize))
        return false;
    if (!DeleteColumnsReduce(MasterEnv, (*Node)->lp, IdxArray, MixedArrSize))
        return false;
    if (PrintProblemToFile(MasterEnv, (*Node)->lp, "bug2.lp") == false)
        return false;

}
if (Procedure == 1) {
    free(IdxArray);
    free(MixedArr);
    free(DualArr);
}

```

Εικόνα 4.5: Σύνοψη Διαδικασίας 1

4.2 Διαδικασία 2: Επιλογή για Προσθήκη στο Master Πρόβλημα ενός Υποσυνόλου Στηλών με Βάση Κατάλληλων Κριτηρίων Ομοιότητας

4.2.1 Περιγραφή της Διαδικασίας 2

Η Διαδικασία 2 αφορά την προσθήκη στο master πρόβλημα ενός στοχευμένα επιλεγμένου υποσυνόλου στηλών, μεταξύ αυτών που έχουν δημιουργηθεί από το υποπρόβλημα δημιουργίας στηλών (column generation), κάνοντας χρήση κατάλληλων κριτηρίων “ομοιότητας/ποικιλίας”.

Η Διαδικασία ουσιαστικά είναι μια εναλλακτική τεχνική για την προσθήκη υποσχόμενων προγραμμάτων στο master πρόβλημα, και πραγματοποιείται μέσα στην συνάρτηση **‘CG_AddPromisingRostersToMaster’**.

Η Διαδικασία 2 αποτελείται από τα εξής βήματα:

1. Δημιουργεί έναν πίνακα **‘Rosters’**, όπου κάθε γραμμή αντιπροσωπεύει ένα υποσχόμενο πρόγραμμα και κάθε στήλη αντιπροσωπεύει τη συμμετοχή του συγκεκριμένου προγράμματος στον κάθε περιορισμό του master προβλήματος.
2. Για κάθε υποσχόμενο πρόγραμμα:
 - Υπολογίζει τη συμμετοχή του προγράμματος στον κάθε περιορισμό του master προβλήματος και αποθηκεύει το αποτέλεσμα στον πίνακα **‘Rosters’**.
 - Συγκρίνει το τρέχον πρόγραμμα με τα προηγούμενα υποσχόμενα προγράμματα που έχουν προστεθεί στον **‘Rosters’**. Αν το πρόγραμμα είναι πολύ παρόμοιο με κάποιο από τα προηγούμενα (δηλαδή η ομοιότητα είναι μεγαλύτερη από ένα καθορισμένο ποσοστό π.χ. 0.7), τότε αποφασίζει να μην το προσθέσει στο master πρόβλημα, αλλάζοντας την τιμή του **‘ShouldAddToMaster[i]’** σε **‘false’**. Αυτό βοηθά στη μείωση του αριθμού των προγραμμάτων που προστίθενται στο master πρόβλημα, μειώνοντας την πολυπλοκότητα του προβλήματος και επιταχύνοντας τη σύγκλιση.
3. Αφού έχουν ελεγχθεί όλα τα υποσχόμενα προγράμματα και έχει αποφασιστεί ποια θα προστεθούν στο master πρόβλημα, η Διαδικασία 2 καλεί την συνάρτηση **‘CG_AddRosterToMaster_UpdateDualsManually’** για να προσθέσει τα επιλεγμένα προγράμματα στο master πρόβλημα και να ενημερώσει τα μέλη χειροκίνητα.

```

int CG_AddPromisingRostersToMaster(struct TreeNode** Node, struct NetworkNode** IdNode, double** Duals, bool SetCover) {
    int i, nRostersAdded = 0, pathcnt = 0;
    bool ProperRoster = false, RosterAddedSuccessfully = false;
    struct NetworkNode* Fict = NULL;
    bool** Rosters;
    int k, n;
    int numcol, totalflighthours, IdUB, IdLB, PreviousBucketIndex, temp, K = 0;
    bool tempDual[N + R]; // One element for every row of the Master lp
    struct NetworkNode* Previous = NULL;

    Fict = (*Node)->Networklist;
    while (Fict->Rtind != -1) Fict = Fict->rightnode; // Find fictitious

    for (i = 0; i <= fictbuckets - 1; i++) {
        if (Fict->Prev[i] != NULL) pathcnt++;
    }

    bool* ShouldAddToMaster;
    ShouldAddToMaster = (bool*)malloc(fictbuckets * sizeof(bool));
    for (i = 0; i < fictbuckets; i++)
        ShouldAddToMaster[i] = true;

    if (Procedure == 2)
    {
        //bool** Rosters;
        Rosters = (bool**)malloc(fictbuckets * (sizeof(bool*)));
        for (int j = 0; j < fictbuckets; j++)
        {
            Rosters[j] = (bool*)malloc((N + R) * (sizeof(bool)));
            for (k = 0; k < N + R; k++)
                Rosters[j][k] = false;
        }
    }

    for (i = 0; i < fictbuckets; i++) { // For every bucket of the fictitious node

        ProperRoster = CG_IsRosterPromising(&Fict, i);
        if (ProperRoster == false)
        {
            ShouldAddToMaster[i] = false;
            continue;
        }

        if (Procedure == 2) {
            Rosters[i][(*IdNode)->index] = true;

            Previous = (Fict)->Prev[i];
            PreviousBucketIndex = (Fict)->PrevPath[i];
            while (Previous->Rtind != -2) {
                Rosters[i][N + Previous->Rtind] = true;
                temp = PreviousBucketIndex;
                PreviousBucketIndex = Previous->PrevPath[temp];
                Previous = Previous->Prev[temp];
            }
        }
    }
}

```

Εικόνα 4. 6 Κώδικας Διαδικασίας 2

```

    if (i == 0)
        ShouldAddToMaster[i] = true;
    else
    {
        for (int e = 0; e < (i - 1); e++)
        {
            if (!ShouldAddToMaster[e])
                continue;
            k = 0; n = 0;
            for (int j = 0; j < (N + R); j++)
            {
                if (Rosters[i][j] == true)
                {
                    n++;
                    if (Rosters[i][j] == Rosters[e][j])
                        k++;
                }
            }
            if ((k / n) > 0.7) {
                ShouldAddToMaster[i] = false;
                break;
            }
        }
    }
}
if (ShouldAddToMaster[i] == true)
{
    RosterAddedSuccessfully = CG_AddRosterToMaster_UpdateDualsManually(Node, &Fict, IdNode, i, &nRostersAdded, Duals, pathcnt, SetCover);
    if (RosterAddedSuccessfully == false) {
        free(ShouldAddToMaster);
        return -1;
    }
}
}
if (Procedure == 2)
    free(Rosters);
    free(ShouldAddToMaster);
return nRostersAdded;
}

```

Εικόνα 4. 7 Συνέχεια κώδικα

Κεφάλαιο 5

5. Αποτελέσματα Υπολογισμών

Στο παραπάνω κεφάλαιο παρουσιάζονται τα αποτελέσματα από την εκτέλεση του προγράμματος για διαφορετικές περιπτώσεις φόρτου πληρώματος και πτητικών διαδρομών. Συγκρίνονται οι χρόνοι ολοκλήρωσης του IP (ακέραιου προγραμματισμού) προβλήματος για Set – Cover λύση και Set – Partition λύση. Εξετάζεται η περίπτωση τριών διαφορετικών συνθηκών, C1 μικρού φόρτου πληρώματος και πτητικών διαδρομών, C2 μεσαίου φόρτου και τέλος C3 μεγάλου φόρτου. Στη συνέχεια δίνονται οι χρόνοι εκτέλεσης του προγράμματος των μέγιστων τιμών για κάθε συνθήκη από δέκα φορές.

Για κάθε διαδικασία οι συντελεστές καθορισμού του αριθμού των μεταβλητών που θα αφαιρεθούν από το πρόβλημα στη Διαδικασία 1, και για το ποσοστό ομοιότητας για τον αριθμό των μεταβλητών που θα εισαχθούν στο πρόβλημα στην Διαδικασία 2 καθορίστηκαν από μια όχι τόσο εκτενή σύγκριση χρόνων για διαφορετικές τιμές αυτών, καθώς δεν αποτελεί κομμάτι της συγκεκριμένης διπλωματικής εργασίας. Οι τιμές που δόθηκαν είναι $perc = 0.3$ και $k/n > 0.7$ για την Διαδικασία 1 και Διαδικασία 2 αντίστοιχα.

Η Διαδικασία 1 με συνθήκη λύσεως Line 2 Set-Partition, δεν ήταν δυνατό να πραγματοποιηθεί λόγω δυσκολίας στην αντιμετώπιση προβλημάτων στο κομμάτι προγραμματισμού της.

Όλες οι εκτελέσεις του προγράμματος πραγματοποιήθηκαν σε υπολογιστή 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz processor και RAM – 16.0 GB.

Οι βιβλιοθήκες βελτιστοποίησης που χρησιμοποιήθηκαν ήταν της CPLEX ILOG IBM σε Microsoft Visual Studio 2022 C++.

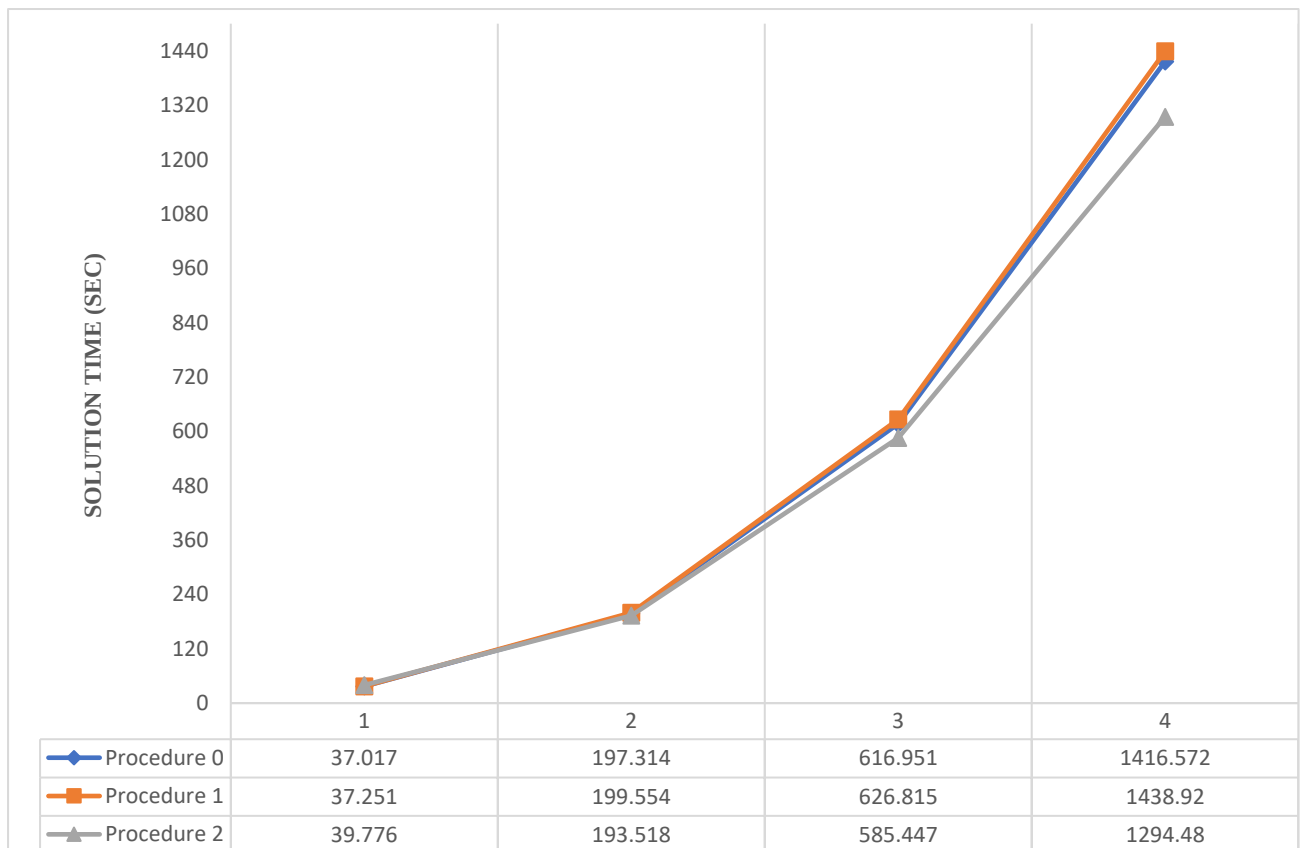
5.1 Line 1 Set-Cover

5.1.1 Σύγκριση Αποτελεσμάτων Set-Cover [C1]: Table

	Procedure 0		Procedure 1		Procedure 2	
	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
N = 100, R = 400	3,452	37,017	3,391	37,251	3,563	39,776
N = 150, R = 600	12,012	197,314	12,061	199,554	12,319	193,518
N = 200, R = 800	26,897	616,181	26,749	626,815	26,728	585,447
N = 250, R = 1000	49,598	1416,572	50,289	1438,92	49,672	1294,48

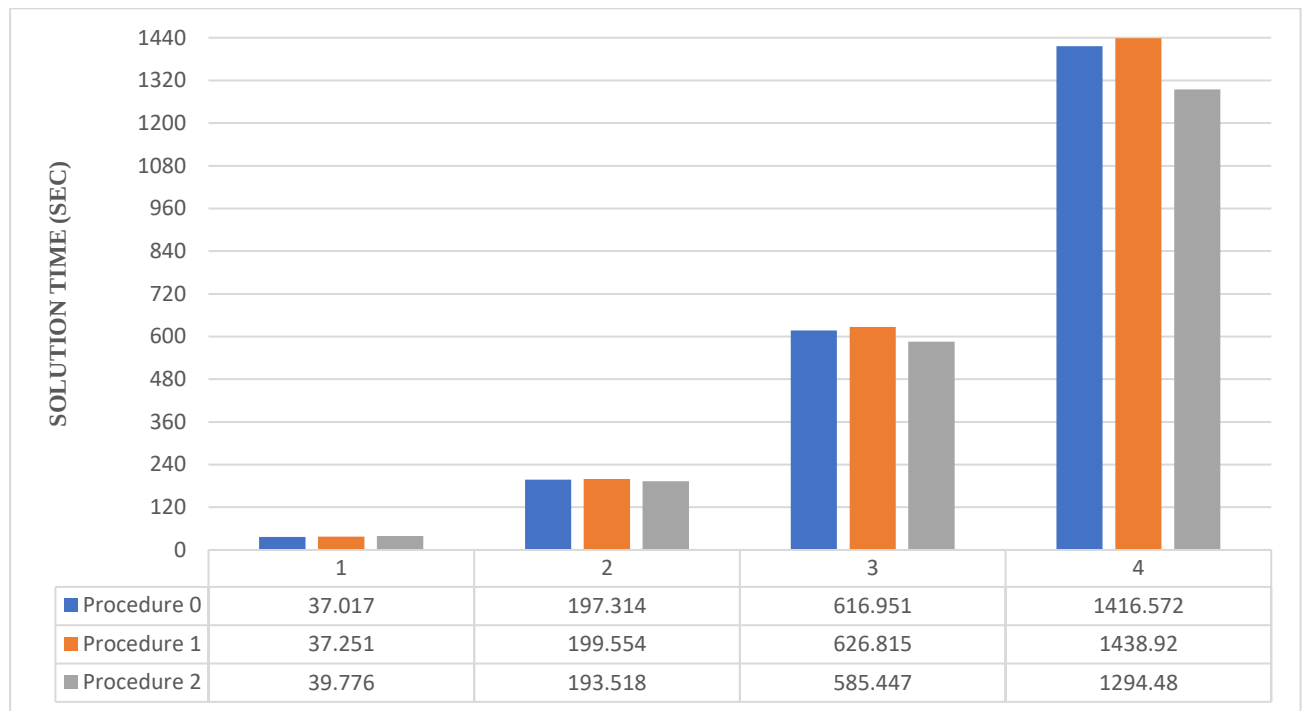
Εικόνα 5.1: Solution time comparison [C1] Line 1 Set-Cover Table

5.1.2 Σύγκριση Αποτελεσμάτων Set-Cover [C1]: Line Chart



Εικόνα 5.2: Solution time comparison [C1] Line 1 Set-Cover Line Chart

5.1.3 Σύγκριση Αποτελεσμάτων Set-Cover [C1]: Bar Chart



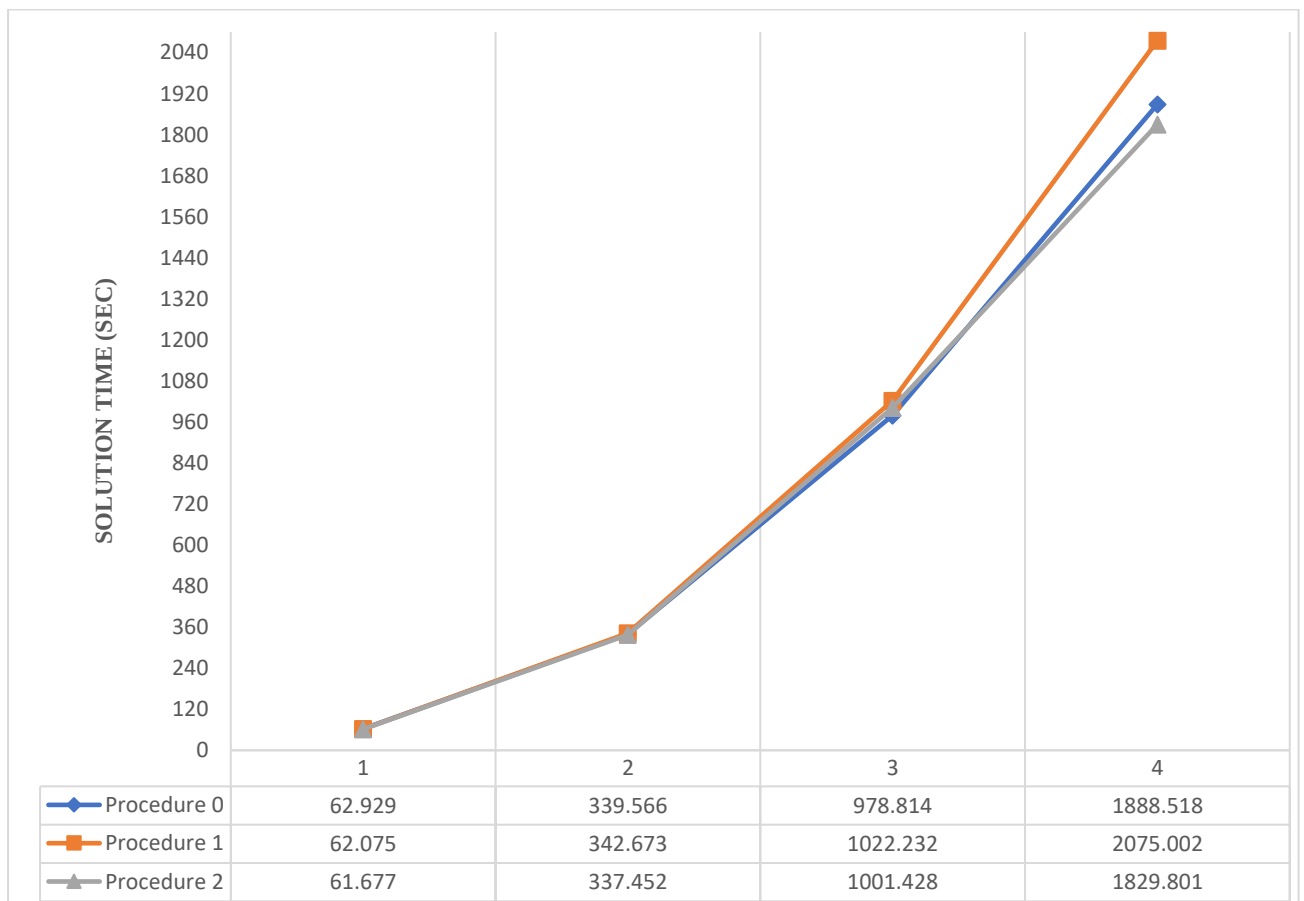
Εικόνα 5.3: Solution time comparison [C1] Line 1 Set-Cover Bar Chart

5.1.4 Σύγκριση Αποτελεσμάτων Set-Cover [C2]: Table

	Procedure 0		Procedure 1		Procedure 2	
	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
N = 100, R = 450	6,348	62,929	6,463	62,075	6,409	61,677
N = 150, R = 675	23,165	339,814	23,239	342,673	22,857	337,452
N = 200, R = 900	50,425	978,814	51,225	1022,232	50,692	1001,428
N = 250, R = 1125	95,724	1888,518	95,439	2075,002	62,239	1829,801

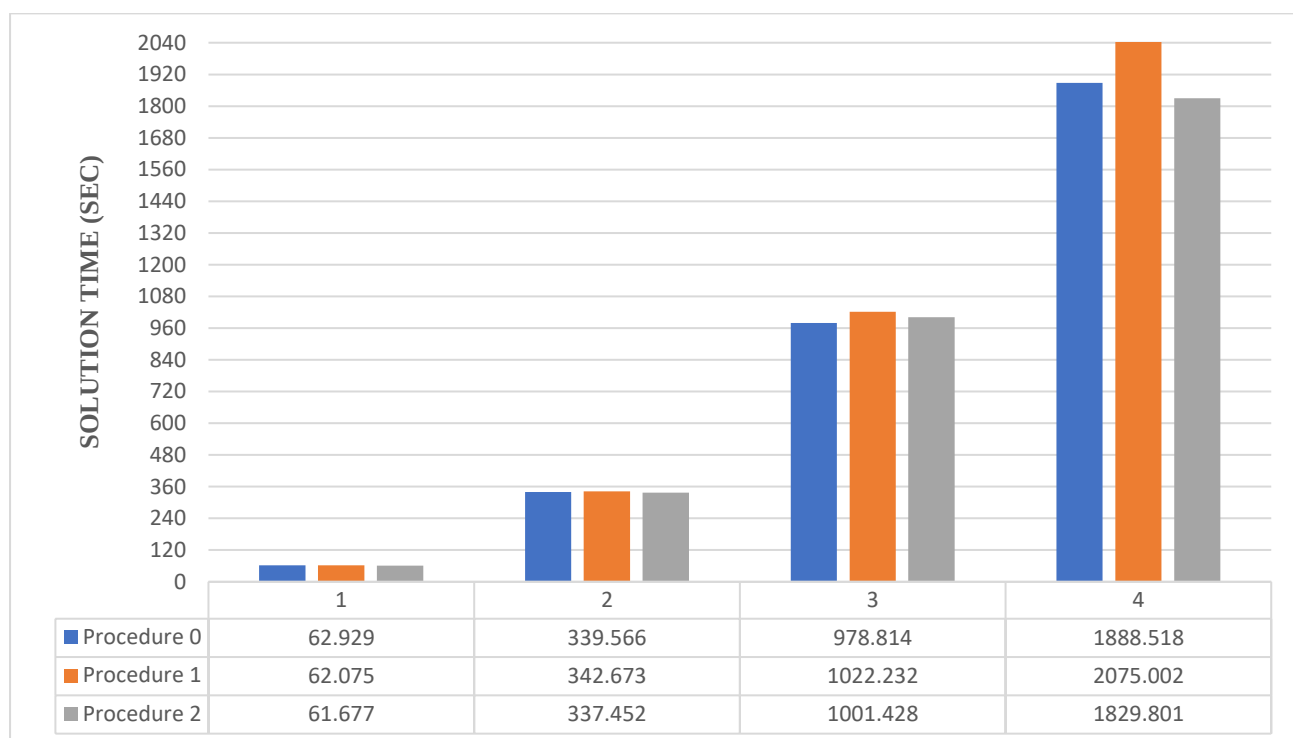
Εικόνα 5.4: Solution time comparison [C2] Line 1 Set-Cover Table

5.1.5 Σύγκριση Αποτελεσμάτων Set-Cover [C2]: Line Chart



Εικόνα 5.5: Solution time comparison [C2] Line 1 Set-Cover Line Chart

5.1.6 Σύγκριση Αποτελεσμάτων Set-Cover [C2]: Bar Chart



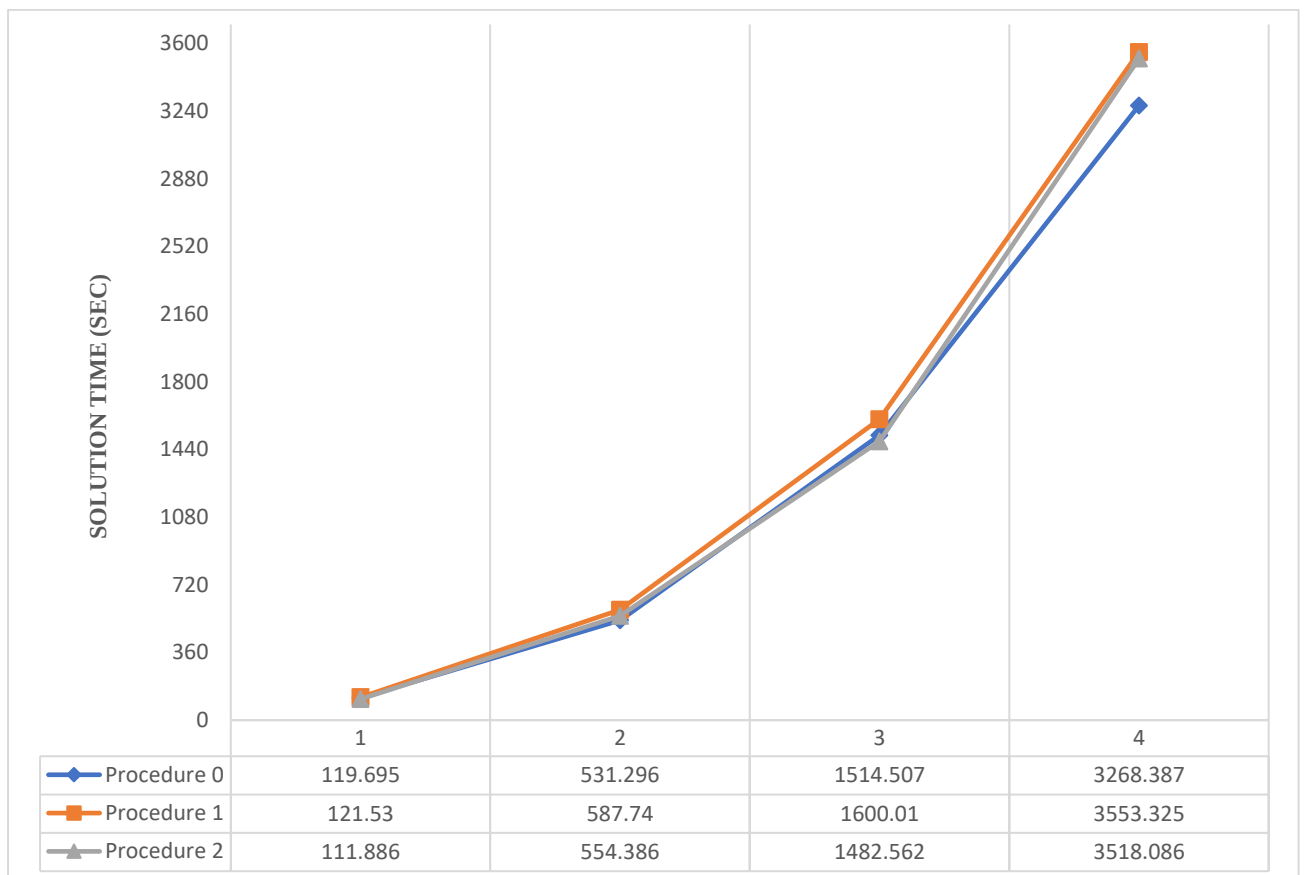
Εικόνα 5.6: Solution time comparison [C2] Line 1 Set-Cover Bar Chart

5.1.7 Σύγκριση Αποτελεσμάτων Set-Cover [C3]: Table

	Procedure 0		Procedure 1		Procedure 2	
	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
N = 100, R = 500	15,047	119,831	18,497	121,530	15,754	111,886
N = 150, R = 750	45,243	531,484	51,392	587,740	45,848	554,533
N = 200, R = 1000	99,012	1514,507	142,483	1600,010	97,055	1482,856
N = 250, R = 1250	180,820	3268,387	252,526	3553,325	180,510	3518,086

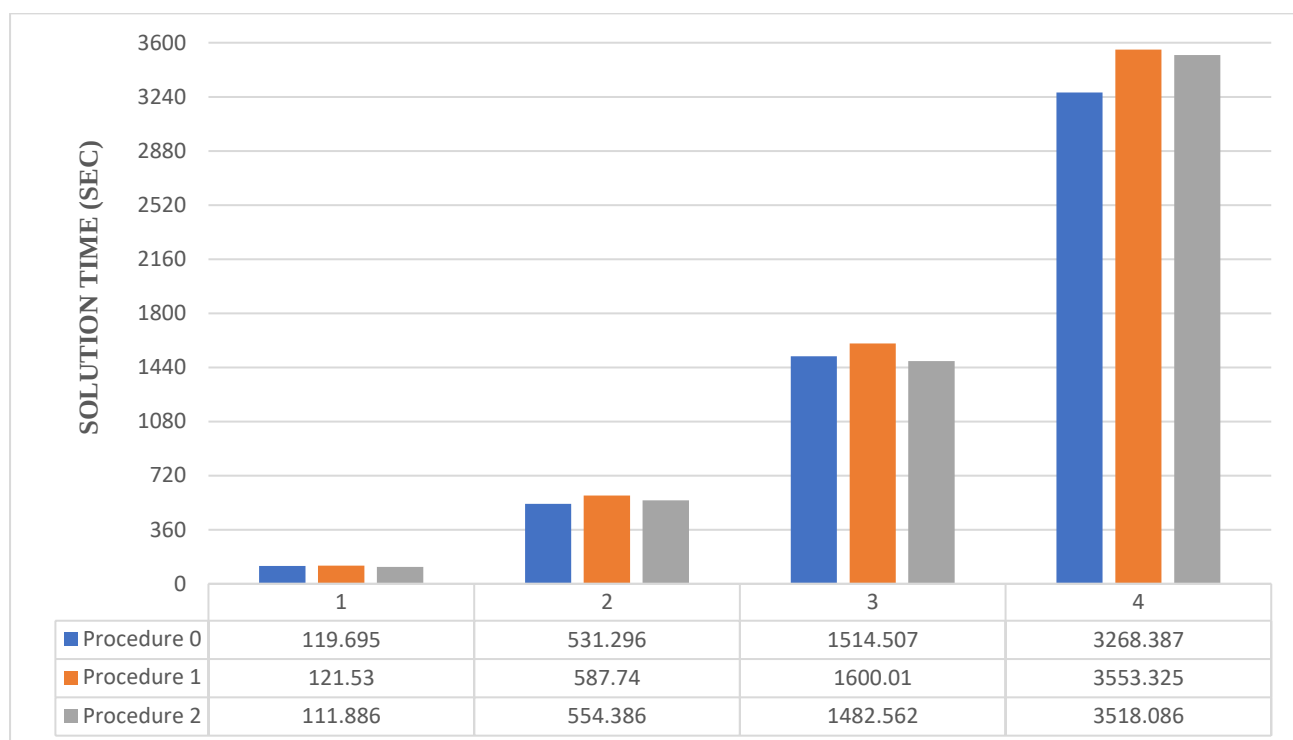
Εικόνα 5.7: Solution time comparison [C3] Line 1 Set-Cover Table

5.1.8 Σύγκριση Αποτελεσμάτων Set-Cover [C2]: Line Chart



Εικόνα 5.8: Solution time comparison [C3] Line 1 Set-Cover Line Chart

5.1.9 Σύγκριση Αποτελεσμάτων Set-Cover [C3]: Bar Chart



Εικόνα 5.9: Solution time comparison [C3] Line 1 Set-Cover Bar Chart

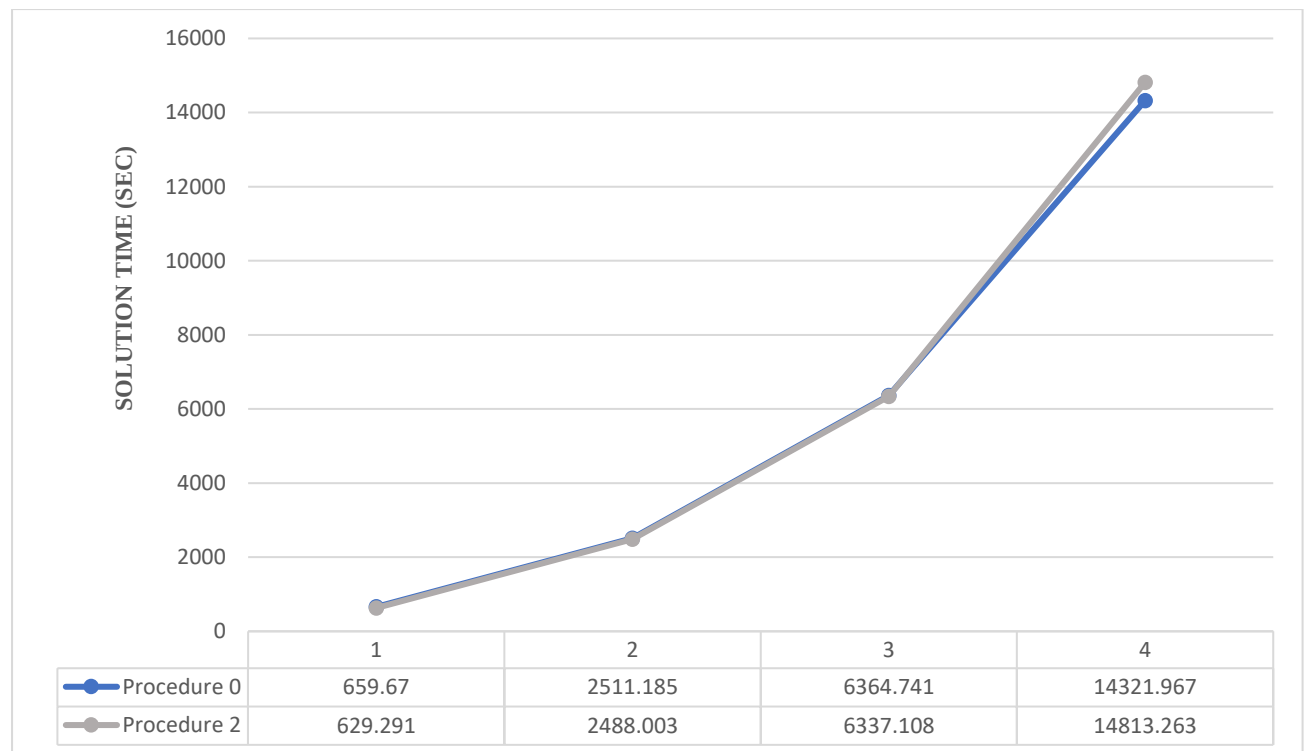
5.2 Line 2 Set-Partition

5.2.1 Σύγκριση Αποτελεσμάτων Set-Partition [C1]: Table

	Procedure 0		Procedure 2	
	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
N = 100, R = 400	342,171	659,670	331,426	629,291
N = 150, R = 600	1167,457	2511,185	1160,626	2588,003
N = 200, R = 800	2866,847	6364,741	2867,870	6337,108
N = 250, R = 1000	6151,095	14321,967	6080,561	14813,263

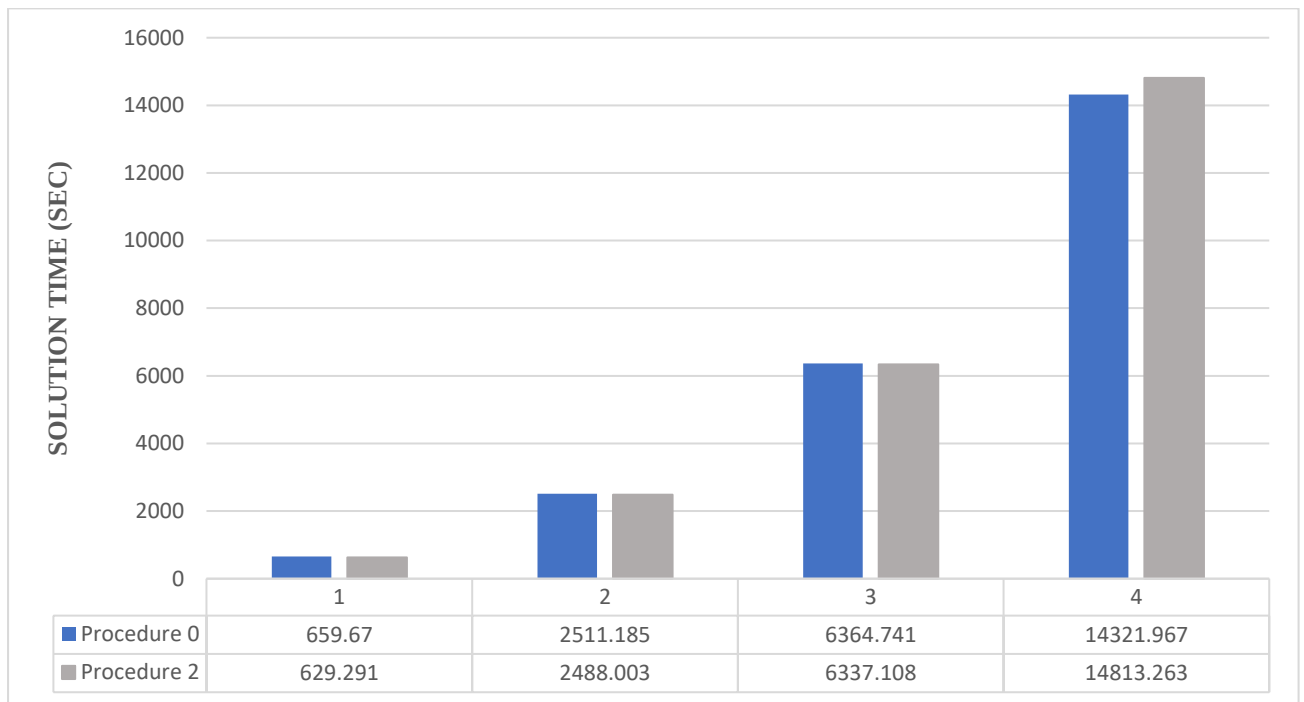
Εικόνα 5.10: Solution time comparison [C1] Line 2 Set-Partition Table

5.2.2 Σύγκριση Αποτελεσμάτων Set-Partition [C1]: Line Chart



Εικόνα 5.11: Solution time comparison [C1] Line 2 Set-Partition Line Chart

5.2.3 Σύγκριση Αποτελεσμάτων Set-Partition [C1]: Bar Chart



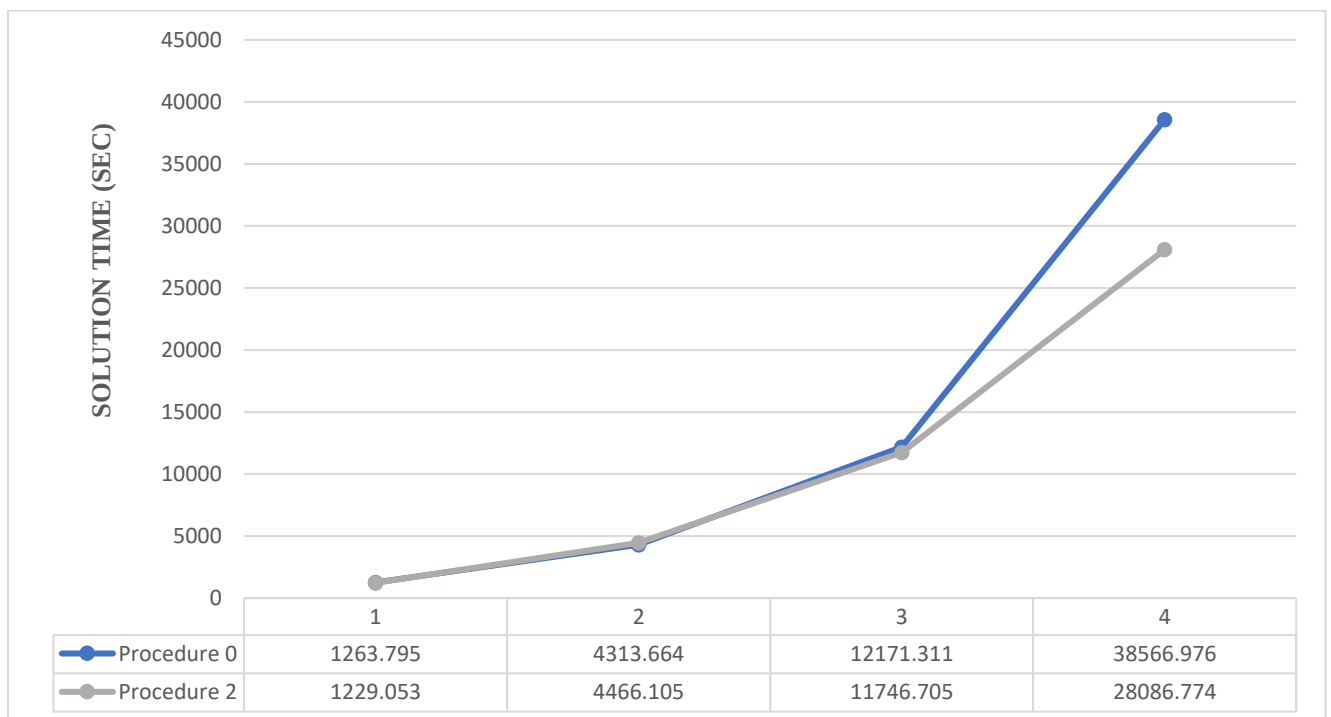
Εικόνα 5.12: Solution time comparison [C1] Line 2 Set-Partition Bar Chart

5.2.4 Σύγκριση Αποτελεσμάτων Set-Partition [C2]: Table

	Procedure 0		Procedure 2	
	<i>LP Coverage Achieved</i>	<i>IP Coverage Achieved</i>	<i>LP Coverage Achieved</i>	<i>IP Coverage Achieved</i>
N = 100, R = 450	477,781	1263,795	423,514	1229,053
N = 150, R = 675	1608,561	4313,664	1541,661	4466,105
N = 200, R = 900	3929,046	12171,311	3863,685	11746,705
N = 250, R = 1125	8695,208	38566,976	7919,553	28086,774

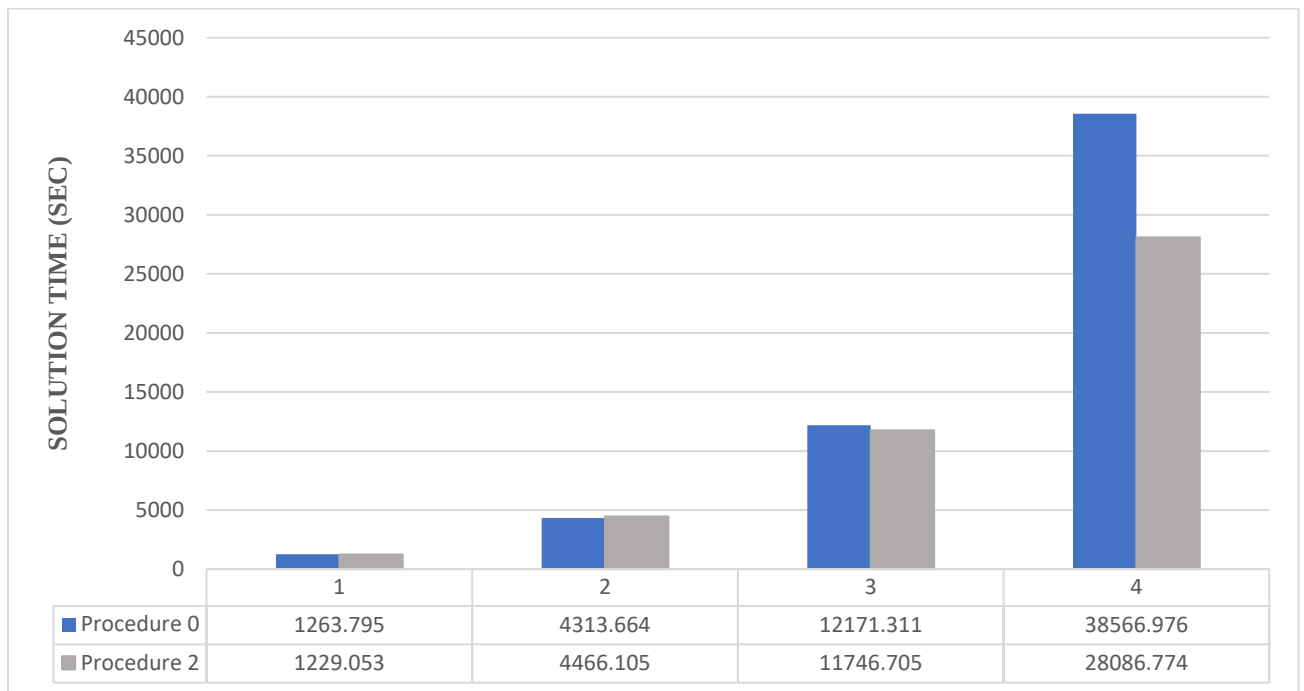
Εικόνα 5.13: Solution time comparison [C2] Line 2 Set-Partition Table

5.2.5 Σύγκριση Αποτελεσμάτων Set-Partition [C2]: Line Chart



Εικόνα 5.14: Solution time comparison [C2] Line 2 Set-Partition Line Chart

5.2.6 Σύγκριση Αποτελεσμάτων Set-Partition [C2]: Bar Chart



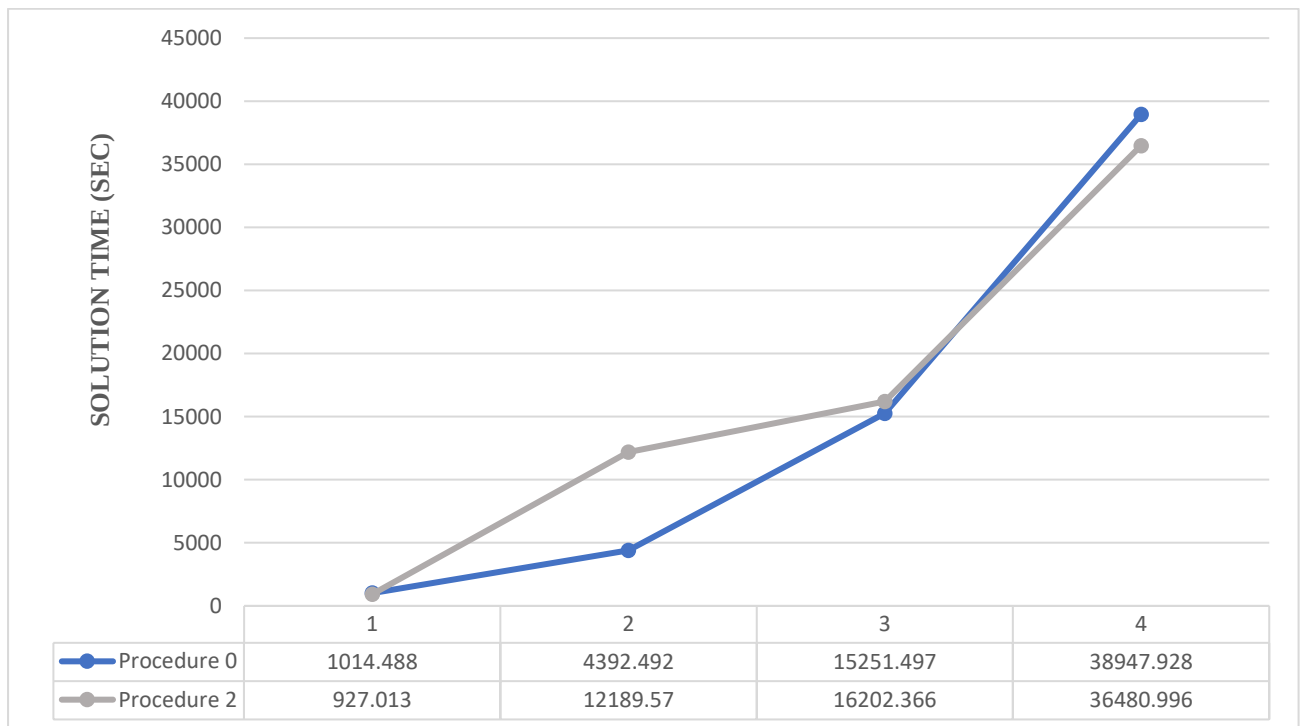
Εικόνα 5.15: Solution time comparison [C2] Line 2 Set-Partition Bar Chart

5.2.7 Σύγκριση Αποτελεσμάτων Set-Partition [C3]: Table

	Procedure 0		Procedure 2	
	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
N = 100, R = 500	472,344	1014,488	439,984	927,013
N = 150, R = 750	1585,692	4392,492	1627,312	12189,570
N = 200, R = 1000	4317,988	15251,497	4314,660	16202,366
N = 250, R = 1250	8836,924	38947,926	8704,44	36480,996

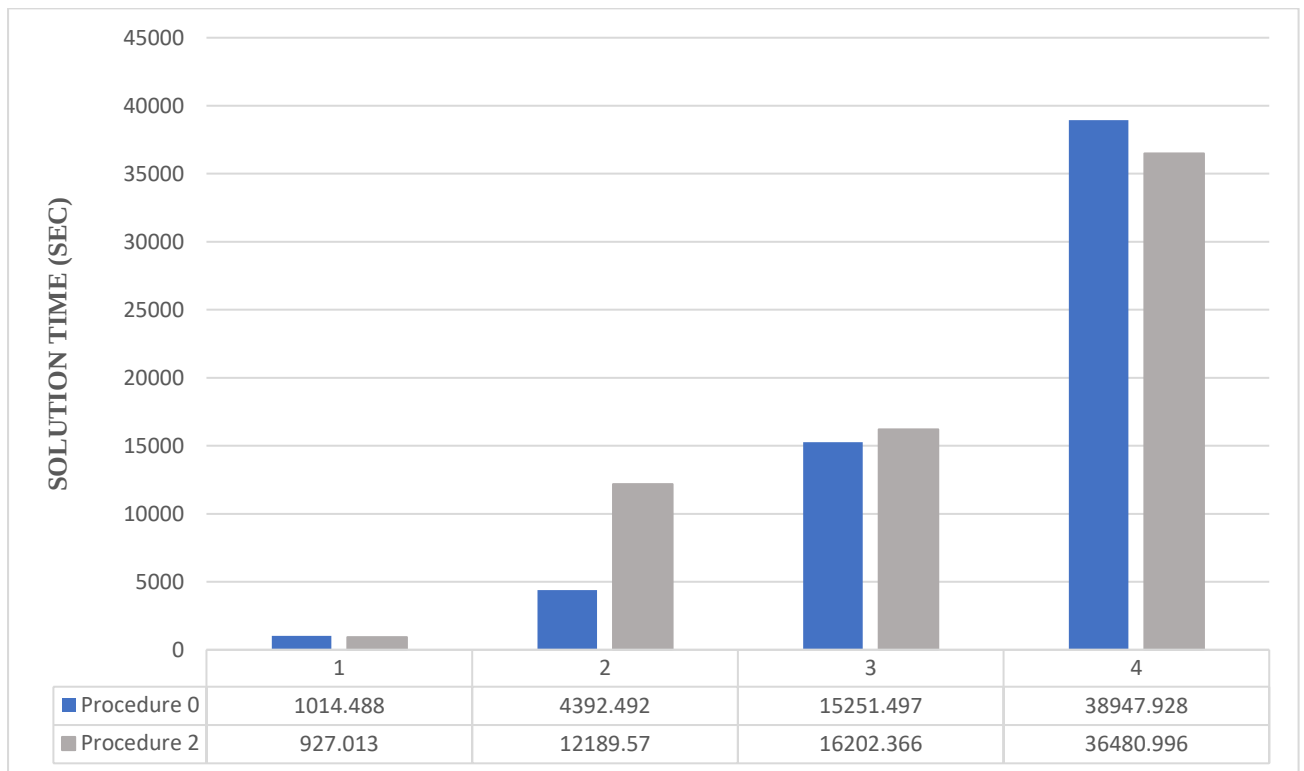
Εικόνα 5.16: Solution time comparison [C3] Line 2 Set-Partition Table

5.2.8 Σύγκριση Αποτελεσμάτων Set-Partition [C3]: Line Chart



Εικόνα 5.17: Solution time comparison [C3] Line 2 Set-Partition Line Chart

5.2.9 Σύγκριση Αποτελεσμάτων Set-Partition [C3]: Bar Chart



Εικόνα 5.18: Solution time comparison [C3] Line 2 Set-Partition Bar Chart

5.3 Line 1 Set-Cover $\bar{x}_{IP}$

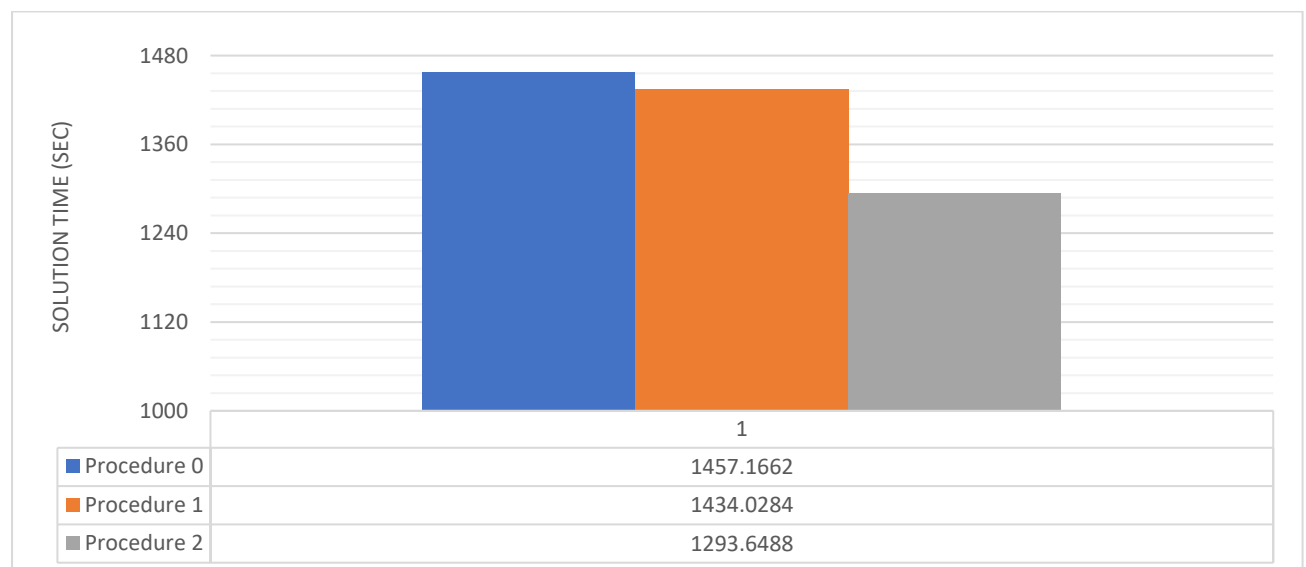
Στη συνέχεια παρουσιάζονται τα αποτελέσματα εκτέλεσης του προγράμματος για τα μέγιστα φορτία πληρώματος και πτητικών διαδρομών για μέθοδο λύσεως Line 1: Set-cover και Line 2: Set-Partition από δέκα φορές. Συγκρίνοντας τους μέσους όρους ($\bar{x}_{IP}$), θα γίνει ξεκάθαρο αν οι διαδικασίες που εισήχθησαν έχουν πραγματικά ως αποτέλεσμα την μείωση του υπολογιστικού φόρτου του προγράμματος.

5.3.1 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C1]: Table

N = 250, R = 1000					
Procedure 0		Procedure 1		Procedure 2	
LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
49,259	1417,375	55,705	1432,788	54,191	1297,519
49,200	1411,936	50,085	1437,631	49,436	1293,830
56,395	1480,101	49,860	1436,700	49,485	1299,285
50,648	1443,138	49,962	1436,794	49,283	1301,286
48,872	1403,548	49,633	1428,729	49,422	1297,422
50,907	1474,244	49,820	1442,763	49,947	1295,317
50,801	1479,226	50,289	1429,936	50,520	1285,114
50,829	1478,093	49,560	1428,187	49,759	1289,278
51,717	1487,673	49,829	1427,704	49,657	1285,176
54,207	1496,328	50,265	1439,052	49,499	1292,261
$\bar{x}_{IP} = 1457,166$		$\bar{x}_{IP} = 1434,028$		$\bar{x}_{IP} = 1293,649$	

Εικόνα 5.19: Line 1 Set-Cover [C1]max Table

5.3.2 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C1]: Bar Chart



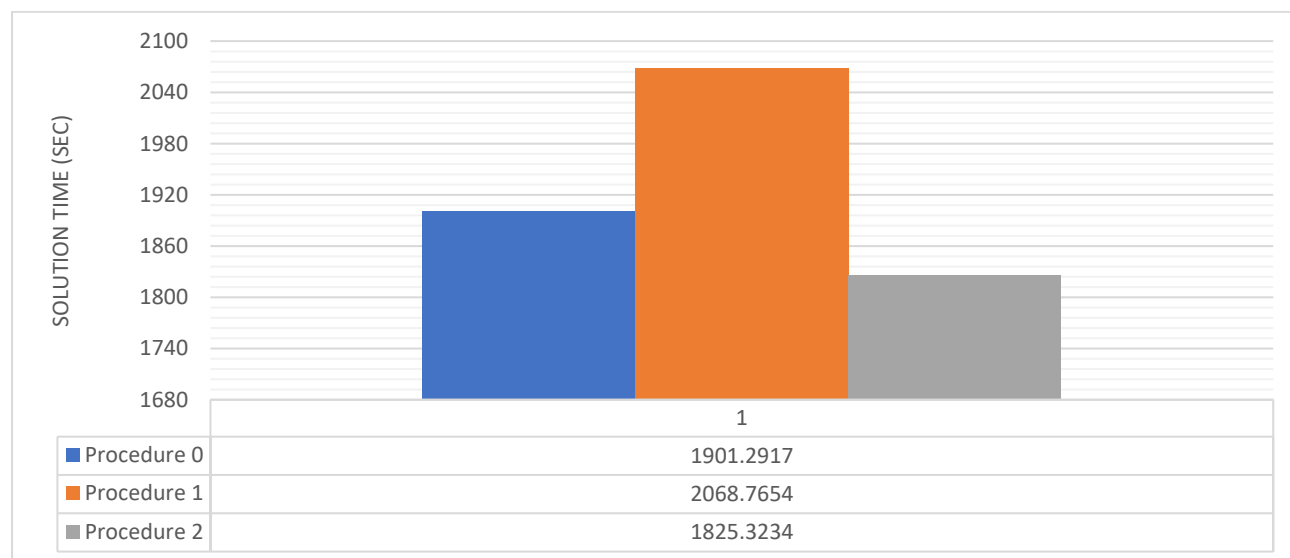
Εικόνα 5.20: Line 1 Set-Cover [C3]max Bar Chart

5.3.3 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C2]: Table

N = 250, R = 1125					
Procedure 0		Procedure 1		Procedure 2	
LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
49,259	1417,375	55,705	1432,788	54,191	1297,519
49,200	1411,936	50,085	1437,631	49,436	1293,830
56,395	1480,101	49,860	1436,700	49,485	1299,285
50,648	1443,138	49,962	1436,794	49,283	1301,286
48,872	1403,548	49,633	1428,729	49,422	1297,422
50,907	1474,244	49,820	1442,763	49,947	1295,317
50,801	1479,226	50,289	1429,936	50,520	1285,114
50,829	1478,093	49,560	1428,187	49,759	1289,278
51,717	1487,673	49,829	1427,704	49,657	1285,176
54,207	1496,328	50,265	1439,052	49,499	1292,261
$\bar{x}_{IP} = 1901,292$		$\bar{x}_{IP} = 2068,765$		$\bar{x}_{IP} = 1825,323$	

Εικόνα 5.21: Line 1 Set-Cover [C2]max Table

5.3.4 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C2]: Table



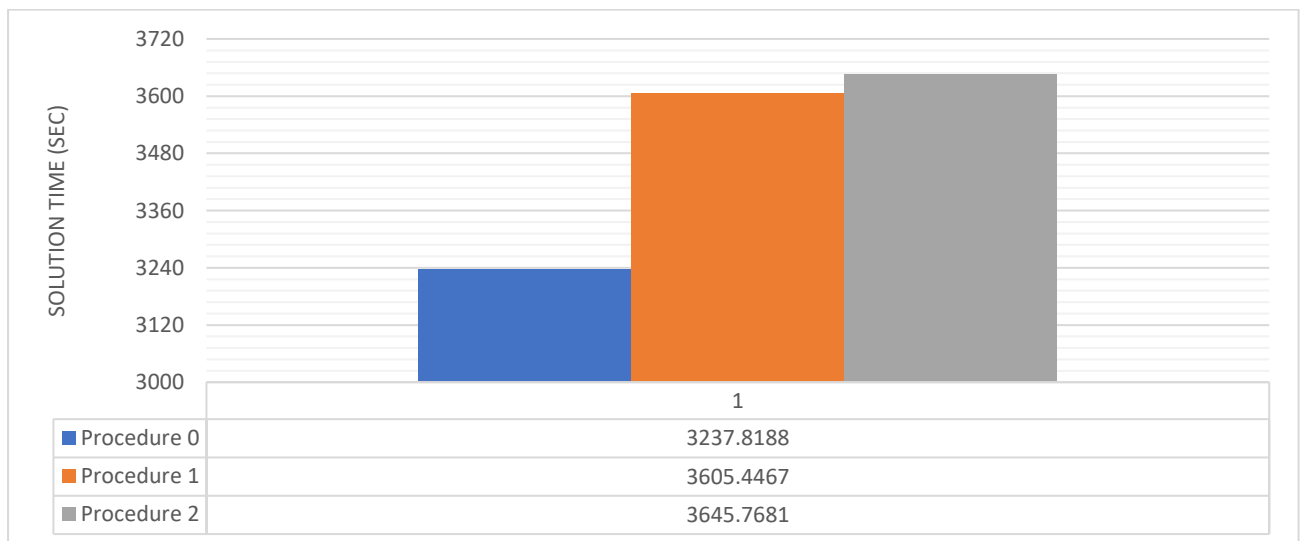
Εικόνα 5.22: Line 1 Set-Cover [C2]max Bar Chart

5.3.5 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C3]: Table

N = 250, R = 1250					
Procedure 0		Procedure 1		Procedure 2	
LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
180,820	3340,529	265,088	3642,535	202,461	3640,346
178,741	3224,461	266,178	3674,875	193,504	3651,918
179,333	3249,130	268,568	3670,554	193,746	3683,522
178,567	3234,042	251,686	3503,349	194,492	3639,982
178,013	3232,059	252,344	3527,317	193,114	3667,980
180,121	3235,189	254,138	3530,417	193,650	3652,963
178,520	3232,909	253,014	3523,612	193,620	3620,897
178,452	3230,297	254,836	3533,483	191,182	3607,507
178,521	3235,468	270,909	3724,318	192,466	3599,376
178,298	3236,246	269,075	3724,007	182,090	3693,190
$\bar{x}_{IP} = 3237,819$		$\bar{x}_{IP} = 3605,447$		$\bar{x}_{IP} = 3645,768$	

Εικόνα 5.23: Line 1 Set-Cover [C3]max Table

5.3.6 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Cover [C3]: Bar Chart



Εικόνα 5.24: Line 1 Set-Cover [C3]max Bar Chart

5.4 Line 1 Set-Partition $\bar{x}_{IP}$

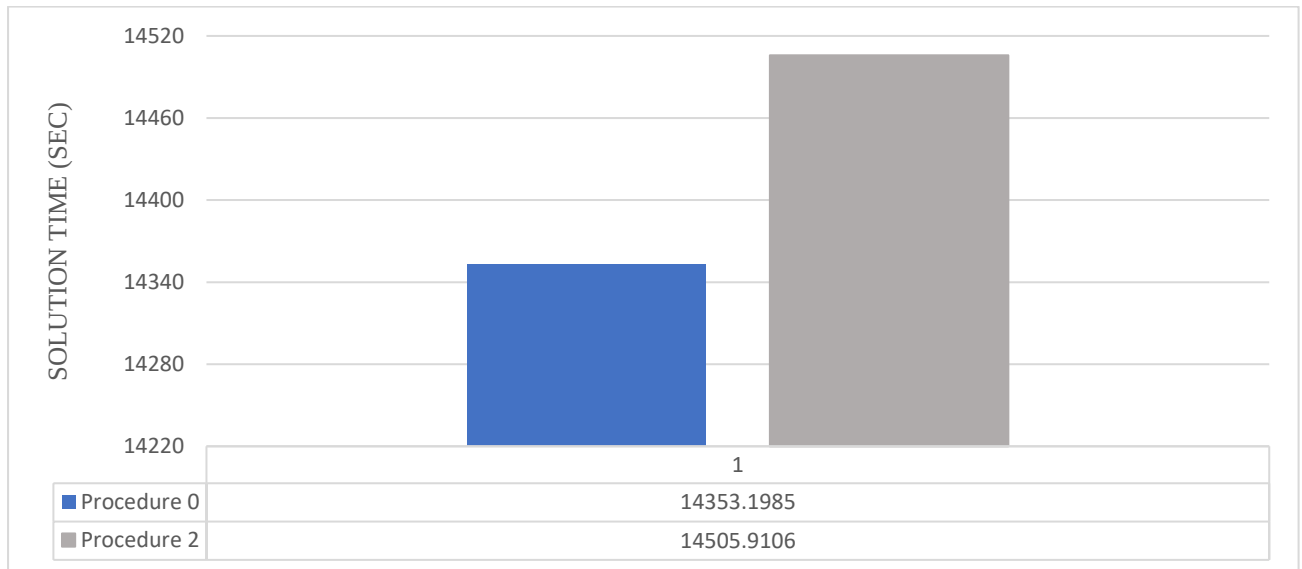
5.4.1 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C1]: Table

N = 250, R = 1000

Procedure 1		Procedure 2	
LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
6151,095	14321,967	6008,561	14637,097
6047,153	14182,277	6284,751	15363,606
6122,112	14382,034	5997,527	14310,319
6139,335	14356,507	6010,833	14331,546
6242,471	14418,002	6035,654	14395,232
6188,892	14301,416	6032,801	14345,331
6243,593	14228,663	6032,601	14425,543
6031,765	14114,529	6124,410	14310,038
6364,183	14587,670	6031,444	14468,109
6348,236	14638,920	6085,838	14472,210
$\bar{x}_{IP} = 14353,199$		$\bar{x}_{IP} = 14505,911$	

Εικόνα 5.25: Line 2 Set-Partition [C1]max Table

5.4.2 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C1]: Bar Chart



Εικόνα 5.26: Line 2 Set-Partition [C1]max Table

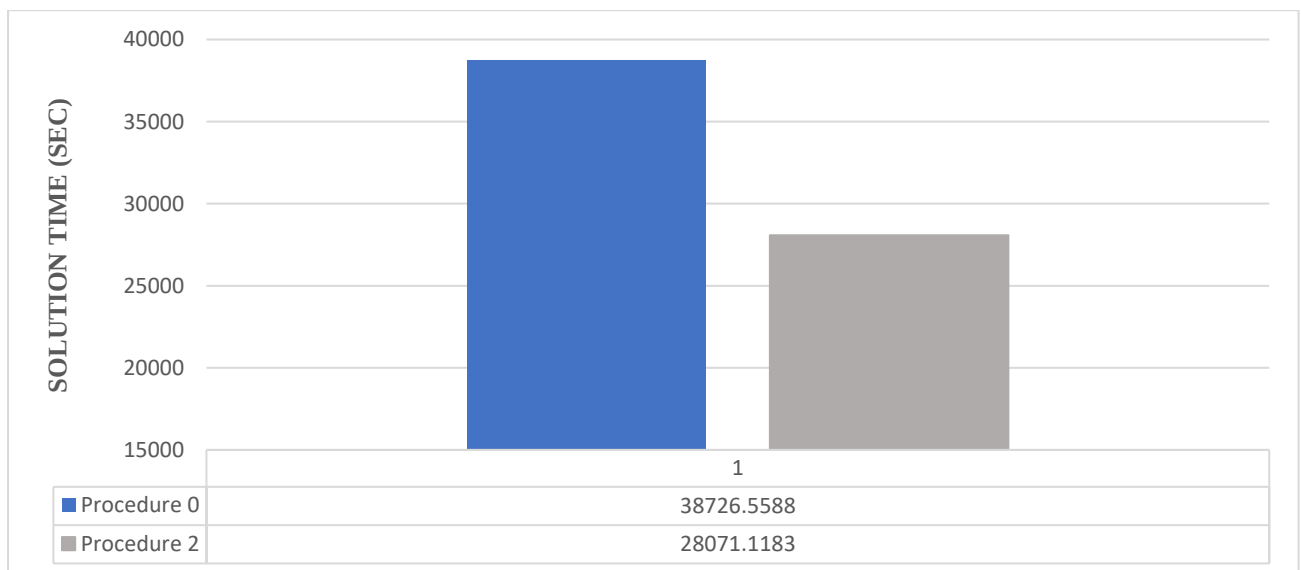
5.4.3 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C2]: Table

N = 250, R = 1125

Procedure 1		Procedure 2	
LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
8695,208	38566,976	7919,553	28086,774
8781,743	38894,328	7917,182	27899,941
8704,109	38649,658	7621,203	27851,930
8746,116	38801,411	7884,407	28022,319
8672,600	38589,747	7946,192	28139,008
8759,582	38756,636	7901,337	28022,670
8688,251	38891,020	7991,632	28240,011
8634,715	38604,809	7934,608	28102,844
8736,274	38804,991	7871,665	28041,569
8602,871	38706,012	7983,688	28304,117
$\bar{x}_{IP} = 38726.559$		$\bar{x}_{IP} = 28071.118$	

Εικόνα 5.27: Line 2 Set-Partition [C2]max Table

5.4.4 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C2]: Bar Chart



Εικόνα 5.28: Line 2 Set-Partition [C2]max Table

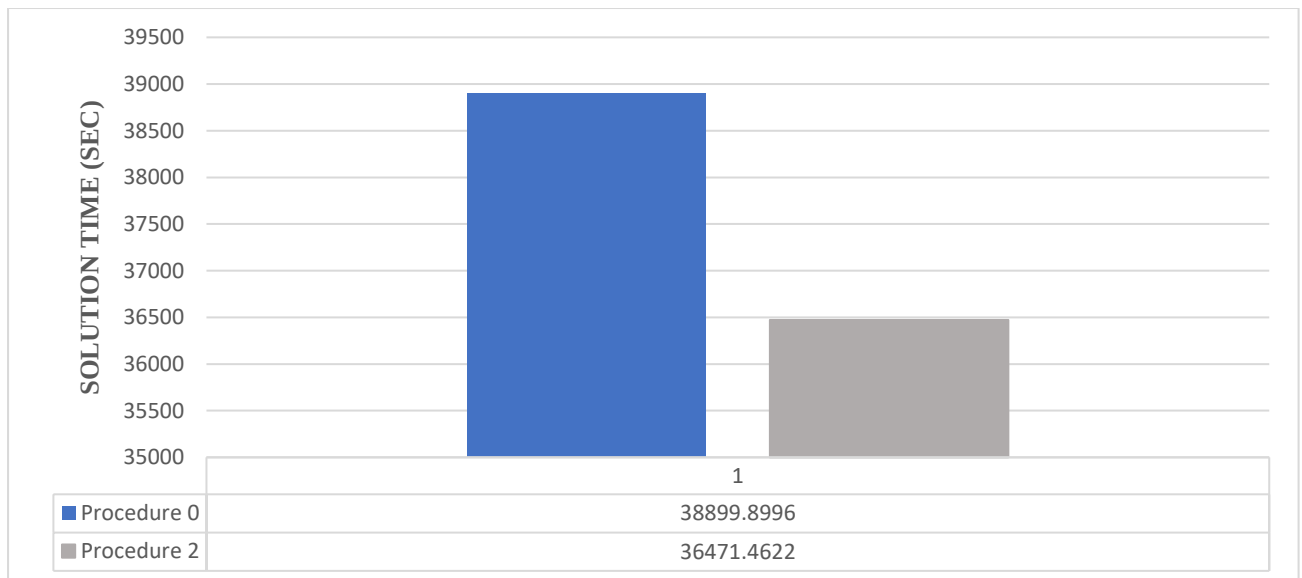
5.4.5 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C3]: Table

N = 250, R = 1250

Procedure 1		Procedure 2	
LP Coverage Achieved	IP Coverage Achieved	LP Coverage Achieved	IP Coverage Achieved
8836,924	38947,928	8704,440	36480,996
8852,341	39018,012	8621,583	36411,702
8797,816	38851,227	8789,723	36575,244
8834,022	38913,890	8637,409	36338,760
8830,554	38937,026	8784,731	36501,082
8819,015	38826,891	8852,368	36647,883
8792,971	38763,309	8603,656	36262,944
8972,601	39054,102	8772,359	36424,773
8818,006	38822,648	8837,946	36675,630
8829,719	38863,963	8654,304	36395,608
$\bar{x}_{IP} = 38899,900$		$\bar{x}_{IP} = 36471,4462$	

Εικόνα 5.29: Line 2 Set-Partition [C3]max Table

5.4.6 Σύγκριση Αποτελεσμάτων $\bar{x}_{IP}$ Set-Partition [C3]: Bar Chart



Εικόνα 5.30: Line 2 Set-Partition [C3]max Bar Chart

5.5 Σχολιασμός Αποτελεσμάτων

Αρχικά αξίζει να σημειωθεί πως θα επικεντρώσουμε την προσοχή μας στα αποτελέσματα που προέκυψαν από τις περιπτώσεις που απαιτούν τον περισσότερο υπολογιστικό φόρτο από τις C1, C2 και C3, αφού αυτά αντικατοπτρίζουν ρεαλιστικά προβλήματα των αεροπορικών εταιρειών. Για πιο ακριβή και ασφαλή συμπεράσματα ο σχολιασμός θα γίνει πάνω στους μέσους όρους ($\bar{X}_{IP}$) για κάθε μέθοδο λύσεως (Set-Cover και Set-Partition).

Για όλες τις περιπτώσεις λύσεως δεν παρατηρήθηκε, όπως ήταν αναμενόμενο, σημαντική μεταβολή της τιμής της αντικειμενικής συνάρτησης του master προβλήματος.

5.5.1 Line 1 Set-Cover

	PROCEDURE 1 /PROCEDURE 0	PROCEDURE 2 /PROCEDURE 0	PROCEDURE 2 /PROCEDURE 1
$\bar{X}_{IPC1MAX}$	0,98	0,89	0,90
$\bar{X}_{IPC2MAX}$	1,09	0,96	0,88
$\bar{X}_{IPC3MAX}$	1,11	1,13	1,01

Εικόνα 5.31: Συντελεστές σύγκρισης διαδικασιών Line 1

Παραπάνω παρουσιάζεται ο πίνακας με τους συντελεστές που προκύπτουν από την σύγκριση των χρόνων εκτέλεσης κάθε μεθόδου ανά ζεύγος. Με βάση την αρχική Διαδικασία του προγράμματος, παρατηρούμε πως η Διαδικασία 1 είναι κατά 2% γρηγορότερη στο C1max, 9% και 11% πιο αργή για C2max και C3max αντίστοιχα. Η Διαδικασία 2 για C1max και C2max είναι 11% και 4% γρηγορότερη στην εκτέλεση της, και για C3max είναι 13% πιο αργή.

Συγκρίνοντας την Διαδικασία 1 με την Διαδικασία 2 παρατηρούμε πως η Διαδικασία 2 είναι στις περιπτώσεις C1max και C2max γρηγορότερη ενώ στην περίπτωση C3max οι χρόνοι εκτέλεσης είναι σχεδόν όμοιοι.

5.5.2 Line 2 Set-Partition

PROCEDURE 2/PROCEDURE 0

$\bar{X}_{IPC1MAX}$	1,01
$\bar{X}_{IPC2MAX}$	0,73
$\bar{X}_{IPC3MAX}$	0,94

Εικόνα 5.32: Συντελεστές σύγκρισης διαδικασιών Line 2

Με ανάλογο τρόπο υπολογίζουμε και τους αντίστοιχους συντελεστές για την αρχική Διαδικασία με την Διαδικασία 2 στην περίπτωση του Line 2 Set-Partition. Για C1max οι χρόνοι εκτέλεσης είναι σχεδόν ίδιοι, ενώ για C2max υπάρχει μείωση του χρόνου κατά 27%. Τέλος για C3max η Διαδικασία 2 είναι γρηγορότερη κατά 6%.

5.6 Συμπεράσματα

Στην προσπάθειά μας να αναλύσουμε την αποτελεσματικότητα των δύο διαφορετικών στρατηγικών που εφαρμόστηκαν στην παρούσα διπλωματική εργασία, δεν παρουσιάζεται μια ξεκάθαρη εικόνα στα αποτελέσματα. Αυτό πηγάζει από την παρατήρηση ότι, στη περίπτωση της Line 1 Set-Cover λύσεως, σε κάποιες περιπτώσεις οι διαδικασίες που εφαρμόστηκαν μειώνουν το χρόνο εκτέλεσης του προγράμματος, ενώ σε άλλες δεν επιβεβαιώνουν αυτήν την τάση. Όμως, όταν εξετάζουμε τη μέθοδο λύσεως Line 2 Set-Partition, η Διαδικασία 2 φαίνεται να μειώνει αισθητά τον υπολογιστικό φόρτο του προγράμματος στις δύο τελευταίες περιπτώσεις, C2max και C3max. Παρ' όλα αυτά, όταν καταλήγουμε στην περίπτωση C1max, οι χρόνοι εκτέλεσης της Διαδικασίας 2 είναι σχεδόν όμοιοι με εκείνους της αρχικής Διαδικασίας.

Συνεπώς, εάν κάποια από τις δύο στρατηγικές είναι περισσότερο υποσχόμενη, αυτή είναι εκείνη της “επιλογής για προσθήκη στο Master πρόβλημα ενός υποσυνόλου στηλών με βάση κατάλληλων κριτηρίων Ομοιότητας”.

Κεφάλαιο 6

6. Μελλοντική Έρευνα

Τροποποίηση των Παραμέτρων των Εισαχθέντων Στρατηγικών Μείωσης του Υπολογιστικού Φόρτου

Όπως αναφέραμε στην αρχή του 5^{ου} Κεφαλαίου, ο συντελεστής ρ_{perc} καθορίζει τον αριθμό των μεταβλητών που πρόκειται να αφαιρεθούν από το master πρόβλημα στην Διαδικασία 1. Το ποσοστό αυτό ορίζεται καθορίζεται από το χρήστη και έχει συγκεκριμένη τιμή. Για παράδειγμα, στη περίπτωση του προβλήματος μας, ισούται με 0,3. Παρατηρήθηκε ότι η μεταβολή αυτής της τιμής επιφέρει και αλλαγές στους χρόνους εκτέλεσης του προγράμματος, εφόσον αλλάζει τον αριθμό των μεταβλητών που αφαιρούνται από το πρόβλημα. Συνεπώς θα μπορούσαν να αποτελέσουν σημαντικό κομμάτι για μελλοντική έρευνα, δοκιμές στο πρόγραμμα διαφορετικών τιμών της μεταβλητής αυτής.

Αντιστοίχως, για την Διαδικασία 2, η τιμή k/n μπορεί να μεταβληθεί και να προκύψουν διαφορετικοί χρόνοι επίλυσης. Άλλωστε όπως διαπιστώθηκε και από την παρούσα διπλωματική εργασία η συγκεκριμένη μέθοδος μας έδωσε τα περισσότερα υποσχόμενα αποτελέσματα. Με σωστή τροποποίηση του ποσοστού ομοιότητας τα αποτελέσματα αναμένεται να βελτιωθούν.

Κεφάλαιο 7

7. Βιβλιογραφία

- [1] Severin Borenstein, Nancy L. Rose, *How Airline Markets Work or...Do They, Regulatory Reform in the Airline Industry* (p. 63 – 135)
- [2] Air Transport Association of America. *The Airline Handbook*. Air Transport Association of America, 2001.
- [3] Cynthia Barnhart, Amy M Cohn, Ellis L Johnson, Diego Klabjan, George L Nemhauser, and Pamela H Vance. *Airline crew scheduling*. In *Handbook of transportation science*, pages 517–560. Springer, 2003
- [4] Gang Yu. *Operations research in the airline industry*, volume 9. Springer Science & Business Media, 2012
- [5] Guy Desaulniers, Jacques Desrosiers, and Marius M Solomon. *Column generation*, volume 5. Springer Science & Business Media, 2006.
- [6] Naud, O., Taylor, J., Colizzi, L., Giroudeau, R., Guillaume, S., Bourreau, E., ... Tisseyre, B. (2020). Support to decision-making. *Agricultural Internet of Things and Decision Support for Precision Smart Farming*, 183–224.
- [7] David R Morrison, Sheldon H Jacobson, Jason J Sauppe, and Edward C Sewell. *Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning*. *Discrete Optimization*, 19:79–102, 2016
- [8] Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., & Vance, P. H. (1998). *Branch-and-Price: Column Generation for Solving Huge Integer Programs*. *Operations Research*, 46(3), 316–329.
- [9] Johnson, E. L., Nemhauser, G. L., & Savelsbergh, M. W. P. (2000). *Progress in Linear Programming-Based Algorithms for Integer Programming: An Exposition*. *INFORMS Journal on Computing*, 12(1), 2–23.
- [10] E. L. Lawler, D. E. Wood, *Branch-And-Bound Methods: A Survey*
- [11] Belobaba, P., Odoni A., Barnhart C., (2009) *The Global Airline Industry*. John Wiley & Sons.
- [12] George Kozanidis. *Branch and price for covering shipments in a logistic distribution network with a fleet of aircraft*. *Optimization Methods and Software*, 33(2):221–248, 2018.
- [13] Odysseas C. M. Moschopoulos, 2018, *Set-Partition versus Set-Cover formulations for column generation based solution methodologies for optimal crew scheduling in the airline industry*, Department of Mechanical Engineering Systems Optimization Laboratory, University of Thessaly.
- [14] Freling, R., Wagelmans, A. P. M., & Paixão, J. M. P. (2001). *Models and Algorithms for Single-Depot Vehicle Scheduling*. *Transportation Science*, 35(2), 165–180.