



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και
μέσω Σμήνους Κοτών για τον Χρονοπρογραμματισμό
Πληρωμάτων Αεροπορικών Εταιρειών**

Αναστάσιος Δ. Τσόπελας

Επιβλέπων: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2022

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

Αναστάσιος Δ. Τσόπελας

A.M.: 1115200900153

ΕΠΙΒΛΕΠΟΝΤΕΣ: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής

ΠΕΡΙΛΗΨΗ

Ο αλγόριθμος βελτιστοποίησης μέσω σμήνους κοτών (CSO) ανήκει στη γενικότερη οικογένεια αλγορίθμων νοημοσύνης σμήνους, ο οποίος μιμείται τον τρόπο με τον οποίο ένα σμήνος από κότες αναζητά τροφή. Αναπτύχθηκε από τους Meng, Liu, Gao και Zhang το 2014 και έκτοτε έχει γίνει αρκετά δημοφιλής και έχει εφαρμοστεί επιτυχώς σε ένα ευρύ φάσμα προβλημάτων βελτιστοποίησης. Έχουν αναπτυχθεί πολλές παραλλαγές του αλγορίθμου, καθώς και υβριδικοί αλγόριθμοι που τον χρησιμοποιούν.

Ο αλγόριθμος βελτιστοποίησης του Αρχιμήδη (AOA) είναι ένας καινούργιος αλγόριθμος, ο οποίος αναπτύχθηκε από τους Hashim, Hussain, Houssein, Mabrouk και Al-Atabany και δημοσιεύτηκε τον Σεπτέμβριο του 2020. Βασίζεται στην αρχή της άνωσης του Αρχιμήδη και προσομοιώνει την βύθιση σε νερό αντικειμένων με διαφορετικά βάρη.

Ο σκοπός αυτής της πτυχιακής εργασίας είναι να εφαρμόσει τους δύο προαναφερθέντες αλγορίθμους βελτιστοποίησης (CSO και AOA) στο πρόβλημα του χρονοπρογραμματισμού πληρωμάτων αεροπορικών εταιρειών. Εξετάζουμε κατά πόσο είναι αποδοτική η χρήση τους στο συγκεκριμένο πρόβλημα, καθώς επιθυμούμε την εύρεση της καλύτερης δυνατής λύσης, ενώ παράλληλα σεβόμαστε τους περιορισμούς που μας τίθενται. Τέλος, παρουσιάζουμε τα αποτελέσματα για κάθε σύνολο αρχικών παραμέτρων που εξετάσαμε και για τον κάθε αλγόριθμο ξεχωριστά.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Τεχνητή Νοημοσύνη

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: νοημοσύνη σμήνους, βελτιστοποίηση σμήνους κοτών, αλγόριθμος βελτιστοποίησης του Αρχιμήδη, πρόβλημα δημιουργίας ροστερ σε πληρώματα αεροπορικών εταιρειών, πρόβλημα ικανοποίησης περιορισμών

ABSTRACT

The chicken swarm optimization algorithm (CSO) belongs to the more broad family of swarm intelligence algorithms and it mimicks the way that a swarm of chickens searches for food. It was developed by Meng, Liu, Gao and Zhang in 2014 and since then it has become quite popular and has been applied successfully to a wide range of optimization problems. There have been developed a lot of variants of the algorithm, as well as hybrid algorithms that utilize it.

The archimedes optimization algorithm (AOA) is a novel algorithm which was developed by Hashim, Hussain, Houssein, Mabrouk and Al-Atabany and was pushished in September of 2020. It is based on the Archimedes' principle and it simulates the immersion of objects with different weights in water.

The purpose of this thesis is to apply the two aforementioned algorithms (CSO and AOA) to the airline crew rostering problem. We examine if their use is efficient for this specific problem, as we want to find the best possible solution, while we respect the constraints given. Finally, we present our results for every set of initial parameters we examined and for each algorithm separately.

SUBJECT AREA: Artificial Intelligence

KEYWORDS: swarm intelligence, chicken swarm optimization, Archimedes optimization algorithm, airline crew rostering, constraint satisfaction problem

Στην μητέρα μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	10
1. ΕΙΣΑΓΩΓΗ.....	11
1.1 Βελτιστοποίηση μέσω σμήνους κοτών	11
1.1.1 Βιολογία.....	11
1.1.2 Βασική εκδοχή του αλγορίθμου.....	12
1.1.3 Παράμετροι.....	14
1.1.4 Πλεονεκτήματα και αδυναμίες	15
1.1.5 Πολύ-βηματική CSO.....	15
1.1.6 Παραλλαγές CSO	16
1.2 Αλγόριθμος Βελτιστοποίησης του Αρχιμήδη	18
1.2.1 Έμπνευση	18
1.2.2 Θεωρητικό υπόβαθρο	19
1.2.3 Αλγόριθμος.....	20
1.2.4 Παράμετροι.....	23
1.2.5 Πειράματα και αποτελέσματα.....	23
1.3 Χρονοπρογραμματισμός πληρωμάτων αεροπορικών εταιρειών	24
1.3.1 Υπόβαθρο	24
1.3.2 Περιορισμοί	25
1.3.3 Στόχοι	27
1.3.4 Μαθηματικό μοντέλο	28
1.3.5 Προσεγγίσεις της λύσης	28
2. ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΙ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΛΗΡΩΜΑΤΩΝ	30
2.1 Ορισμός προβλήματος	30
2.2 Περιγραφή εφαρμογής.....	31
2.2.1 Είσοδος εφαρμογής	32
2.2.2 Αεροπορική εταιρεία.....	33
2.2.3 Ανανέωση θέσης.....	35
2.2.4 Κανόνες.....	38
2.2.5 Κατασκευή λύσης.....	42
2.2.6 Ανάθεση συνδυασμού.....	43
2.2.7 Βελτίωση ίσου καταμερισμού.....	45

3. ΚΡΙΤΗΡΙΑ ΑΞΙΟΛΟΓΗΣΗΣ	47
4. ΑΠΟΤΕΛΕΣΜΑΤΑ.....	50
4.1 Αξιολόγηση πολύ-βηματικής CSO.....	52
4.2 Αξιολόγηση βελτιστοποίησης του Αρχιμήδη	56
4.2.1 Εκτίμηση παραμέτρου C1	56
4.2.2 Εκτίμηση παραμέτρου C2	57
4.2.3 Εκτίμηση παραμέτρου C3	58
4.2.4 Εκτίμηση παραμέτρου C4	59
4.3 Εκτέλεση πειραμάτων	60
4.3.1 Πείραμα πολύ-βηματικής CSO για 45 πιλότους.....	60
4.3.2 Πείραμα πολύ-βηματικής CSO για 40 πιλότους.....	61
4.3.3 Πείραμα πολύ-βηματικής CSO για 50 πιλότους.....	61
4.3.4 Πείραμα βελτιστοποίησης του Αρχιμήδη για 45 πιλότους	61
4.3.5 Πείραμα βελτιστοποίησης του Αρχιμήδη για 40 πιλότους	63
4.3.6 Πείραμα βελτιστοποίησης του Αρχιμήδη για 50 πιλότους	63
5. ΣΥΜΠΕΡΑΣΜΑΤΑ	64
ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	65
ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ	67
ΠΑΡΑΡΤΗΜΑ Ι.....	68
ΠΑΡΑΡΤΗΜΑ ΙΙ: ΣΤΑΤΙΣΤΙΚΑ.....	69
ΑΝΑΦΟΡΕΣ	96

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Ιεραρχία σμήνουσ κοτών	12
Σχήμα 2: Βύθιση αντικειμένου σε υγρό και ο όγκος του εκτοπισμένου υγρού	19
Σχήμα 3: Βύθιση πολλαπλών αντικειμένων σε υγρό	20
Σχήμα 4: Γράφος γνώσης για τρεις συνδυασμούς και δύο πράκτορες	36
Σχήμα 5: FL=0 (αριστερά), FL=0.1 (δεξιά).....	53
Σχήμα 6: FL=0.2 (αριστερά), FL=0.3 (δεξιά).....	53
Σχήμα 7: FL=0.4 (αριστερά), FL=0.5 (δεξιά).....	53
Σχήμα 8: FL=0.6 (αριστερά), FL=0.7 (δεξιά).....	54
Σχήμα 9: FL=0.8 (αριστερά), FL=0.9 (δεξιά).....	54
Σχήμα 10: FL=1 (αριστερά), FL=1.1 (δεξιά).....	54
Σχήμα 11: FL=1.2 (αριστερά), FL=1.3 (δεξιά).....	55
Σχήμα 12: FL=1.4 (αριστερά), FL=1.5 (δεξιά).....	55
Σχήμα 13: FL=1.6 (αριστερά), FL=1.7 (δεξιά).....	55
Σχήμα 14: FL=1.8 (αριστερά), FL=1.9 (δεξιά).....	56
Σχήμα 15: FL=2.0	56
Σχήμα 16: C1=1 (αριστερά), C1=2 (δεξιά).....	57
Σχήμα 17: C2=2 (αριστερά), C2=4 (δεξιά).....	57
Σχήμα 18: C2=6.....	58
Σχήμα 19: C3=1 (αριστερά), C3=2 (δεξιά).....	59
Σχήμα 20: C4=0.5 (αριστερά), C4=1 (δεξιά).....	59
Σχήμα 21: Επίδραση παραμέτρου FL.....	61
Σχήμα 22: Επίδραση συνδυασμού παραμέτρων C1, C2, C3, C4.....	62
Σχήμα 23: Επίδραση παραμέτρου C1 (αριστερά) και C2 (δεξιά).....	62
Σχήμα 24: Επίδραση παραμέτρου C3 (αριστερά) και C4 (δεξιά).....	63
Σχήμα 25: Πρόγραμμα Πιλότων.....	68

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Στατιστικά έγκυρου προγράμματος.....	50
Πίνακας 2: Αποτελέσματα πολύ-βηματικής CSO για 40 πιλότους, FL=1.....	69
Πίνακας 3: Αποτελέσματα πολύ-βηματικής CSO για 50 πιλότους, FL=1.....	70
Πίνακας 4: Αποτελέσματα βελτιστοποίησης του Αρχιμήδη για 40 πιλότους	71
Πίνακας 5: Αποτελέσματα βελτιστοποίησης του Αρχιμήδη για 50 πιλότους	72
Πίνακας 6: Αποτελέσματα πολύ-βηματικής CSO για 45 πιλότους.....	73
Πίνακας 7: Αποτελέσματα βελτιστοποίησης του Αρχιμήδη για 45 πιλότους,	82

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία συντάχθηκε στο πλαίσιο εκπόνησης της πτυχιακής διατριβής του τμήματος Πληροφορικής και Τηλεπικοινωνιών, με σκοπό να εξεταστεί η αποτελεσματικότητα δύο αλγορίθμων τεχνητής νοημοσύνης, του αλγορίθμου βελτιστοποίησης μέσω σμήνους κοτών και του αλγορίθμου βελτιστοποίησης του Αρχιμήδη, στην επίλυση ενός προβλήματος βελτιστοποίησης, συγκεκριμένα την ανάθεση πληρωμάτων για πτήσεις αεροπορικών εταιρειών. Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Σταματόπουλο για την βοήθειά του τόσο στην επιλογή του θέματος αυτού, όσο και καθ' όλη την διάρκεια της εργασίας.

1. ΕΙΣΑΓΩΓΗ

1.1 Βελτιστοποίηση μέσω σμήνους κοτών

Ο αλγόριθμος βελτιστοποίησης μέσω σμήνους κοτών (chicken swarm optimization - CSO) ανήκει στην οικογένεια αλγορίθμων τεχνητής νοημοσύνης που αντλούν την έμπνευσή τους από τη συμπεριφορά ζώων κατά την αναζήτηση τροφής. Η οικογένεια αυτή είναι ευρέως γνωστή με το όνομα νοημοσύνη σμήνους, όρος που χρησιμοποιήθηκε για πρώτη φορά από τους Gerardo Beni και Jing Wang το 1989. Έχουν αναπτυχθεί αρκετοί αλγόριθμοι που μιμούνται πολλά διαφορετικά ζώα, όπως μυρμηγκια, μέλισσες, ελέφαντες και λύκους. Κατά βάση, όλοι τους προσπαθούν να βρουν μία ισορροπία ανάμεσα στην εξερεύνηση και στην εκμετάλλευση του χώρου αναζήτησης ενός προβλήματος βελτιστοποίησης, ώστε να επιταχυνθεί η διαδικασία εύρεσης της καλύτερης λύσης.

Οι αλγόριθμοι νοημοσύνης σμήνους έχουν πολλές πρακτικές εφαρμογές, με τομείς όπως την ιατρική, τη βιομηχανία, την ενέργεια ή τις τηλεπικοινωνίες. Η βελτιστοποίηση μέσω σμήνους κοτών, συγκεκριμένα, έχει κερδίσει δημοτικότητα τα τελευταία χρόνια. Αυτό οφείλεται στο γεγονός ότι ο αλγόριθμος μπορεί να συνδυαστεί εύκολα με άλλα εργαλεία της τεχνητής νοημοσύνης, αλλά και στην εφαρμογή του σε ένα μεγάλο εύρος προβλημάτων βελτιστοποίησης, όπως η τοποθέτηση σταθμών φόρτισης και η πρόβλεψη της επίδοσης δεδομένων που αναρτώνται στο Facebook, η οποία έχει δείξει πολύ υποσχόμενα αποτελέσματα.

1.1.1 Βιολογία

Οι κότες είναι από τα πιο συνηθισμένα οικόσιτα ζώα παγκοσμίως, τα οποία οι άνθρωποι κρατούν και μεγαλώνουν ως πηγή τροφής, αφού καταναλώνουν τόσο το κρέας τους όσο και τα αυγά τους, αλλά και ως κατοικίδια. Πρόκειται για κοινωνικά και εξωστρεφή ζώα που ζουν και αναζητούν τροφή σε σμήνη. Έχουν την ικανότητα να αναγνωρίζουν πάνω από 100 διαφορετικά πρόσωπα, ακόμη και αν έχουν περάσει αρκετοί μήνες από την τελευταία φορά που τα είδαν. Επικοινωνούν μεταξύ τους με περισσότερους από 30 διαφορετικούς ήχους, όπως κρωγμοί (κακαρίσματα), τιτιβίσματα και ουρλιαχτά. Αυτοί οι ήχοι χρησιμοποιούνται κατά την εύρεση φωλιάς ή τροφής, την εύρεση συντρόφου ή την παρουσία κάποιου κινδύνου, ώστε να ειδοποιηθούν όλα τα μέλη του σμήνους. Όταν βρεθεί τροφή, είναι πιθανό ένας κόκορας να καλέσει άλλες κότες, ώστε να φάνε πρώτες. Το ίδιο μπορεί να παρατηρηθεί και σε κότες που είναι μητέρες, οι οποίες μπορεί να καλέσουν τα κοτόπουλά τους και να τα ενθαρρύνουν να φάνε. Επίσης, χρησιμοποιούν τις προηγούμενες εμπειρίες τους όταν αναζητούν τροφή, ώστε να επιταχύνουν τη διαδικασία. Όταν παρουσιάζεται κάποιος κίνδυνος, διαθέτουν διαφορετικούς ήχους, όχι μόνο για να ειδοποιήσουν το σμήνος για την απειλή, αλλά και για το είδος της, για παράδειγμα αν πρόκειται για εναέρια ή απειλή από εδάφους.

Ως ζώα με κοινωνική νοημοσύνη, κάποια μέλη του σμήνους κυριαρχούν απέναντι στα υπόλοιπα, εγκαθιδρύοντας εν τέλει μια ιεραρχία (Σχήμα 1), όπου τα πιο ισχυρά μέλη βρίσκονται στην κορυφή. Αυτή η ιεραρχία είναι πολύ σημαντική για το σμήνος, καθώς αν αφαιρέσουμε από αυτό κότες ή κόκορες, θα διαταραχθεί η ισορροπία του σμήνους μέχρι να εγκαθιδρυθεί μια νέα ιεραρχία. Στην κορυφή βρίσκονται οι κόκορες, τους οποίους ακολουθούν οι πιο ισχυρές κότες. Οι κόκορες έχουν πρόσβαση πρώτοι σε τροφή και σε τοποθεσίες όπου μπορούν να γίνουν φωλιές. Επίσης, θα παλέψουν με άλλους κόκορες ή κότες, αν αυτοί εισβάλουν στην περιοχή τους. Στη δεύτερη θέση

βρίσκονται οι κότες, οι οποίες ακολουθούν τους κόκορες που είναι στην ίδια ομάδα με αυτές κατά την αναζήτηση τροφής. Επίσης, οι κότες ανταγωνίζονται μεταξύ τους κατά την αναζήτηση τροφής. Στη βάση της ιεραρχίας βρίσκονται τα κοτόπουλα, τα οποία απλώς ακολουθούν τις μητέρες τους όταν ψάχνουν για τροφή.

Οι κότες, ως μεμονωμένες οντότητες δεν μπορούν να συνεργαστούν μεταξύ τους. Παρόλα αυτά, ως σμήνος, και δεδομένης μιας συγκεκριμένης ιεραρχίας μπορούν να συγχρονιστούν ως ομάδα, ώστε να αναζητήσουν τροφή. Αυτή η συμπεριφορά μπόρεσε να υιοθετηθεί, ώστε να αναπτυχθεί ο αλγόριθμος νοημοσύνης σμήνους.



Σχήμα 1: Ιεραρχία σμήνους κοτών

1.1.2 Βασική εκδοχή του αλγορίθμου

Η βελτιστοποίηση μέσω σμήνους κοτών αναπτύχθηκε από τους Meng et al. (2014). Ο αλγόριθμος μιμείται τη συμπεριφορά που επιδεικνύει ένα σμήνος κοτών κατά την αναζήτηση τροφής, με σκοπό να φτάσει με αποδοτικό τρόπο στην καλύτερη λύση ενός προβλήματος βελτιστοποίησης. Επίσης, υιοθετεί τους μηχανισμούς που χρησιμοποιεί το σμήνος κατά τη διαδικασία αυτή, καθώς και την ιεραρχία που υπάρχει σε αυτό. Κάθε μέλος του σμήνους αποτελεί έναν ορθολογικό πράκτορα, του οποίου ο σκοπός είναι να βρει μια λύση του προβλήματος. Η λύση αυτή θεωρείται ως η θέση του μέλους στον χώρο αναζήτησης του προβλήματος. Αποτιμώντας την λύση που βρίσκει κάθε μέλος, βρίσκουμε την τιμή της καταλληλότητάς του και με βάση αυτήν χαρακτηρίζεται ως κόκορας, κότα ή κοτόπουλο. Τα μέλη με τις μεγαλύτερες τιμές χαρακτηρίζονται ως κόκορες, τα μέλη με τις μικρότερες ως κοτόπουλα και τα μέλη που οι τιμές τους είναι στη μέση χαρακτηρίζονται ως κότες. Επίσης, εφαρμόζεται η σχέση μητέρας-παιδιού ανάμεσα σε κοτόπουλα και κότες. Η σχέση αυτή, καθώς και η ιεραρχία του σμήνους, ανανεώνεται κάθε φορά που ο αλγόριθμος ολοκληρώνει G βήματα. Η συμπεριφορά των κοτών να ακολουθούν τους κόκορες που είναι στην ίδια ομάδα με αυτές και των κοτόπουλων να ακολουθούν τις μητέρες τους υλοποιείται αποδοτικά από τον αλγόριθμο. Επιπλέον, υλοποιείται ο ανταγωνισμός ανάμεσα σε κότες και η προσπάθεια τους να αρπάξουν την τροφή που έχει βρει κάποια άλλη κότα. Συγκεντρώνοντας όλα τα παραπάνω, ο αλγόριθμος χωρίζεται σε δύο φάσεις. Η πρώτη είναι η αρχικοποίηση, ενώ η δεύτερη είναι η ανανέωση.

Κατά την αρχικοποίηση, ορίζουμε το μέγεθος του σμήνους και τις παραμέτρους που σχετίζονται με τον αλγόριθμο, όπως είναι ο αριθμός των κοκόρων, των κοτών και των κοτόπουλων στο σμήνος, ο αριθμός των κοτών που είναι μητέρες και ο αριθμός των βημάτων που εκτελούμε πριν ανανεώσουμε την ιεραρχία του σμήνους. Επίσης, βρίσκουμε τις τιμές καταλληλότητας των αρχικών λύσεων των μελών του σμήνους και εγκαθιδρύουμε την ιεραρχία με βάση αυτές.

Ο αλγόριθμος κάνει κάποιες παραδοχές ώστε να λειτουργήσει. Καταρχάς, υποθέτει ότι οι κότες είναι περισσότερες από τους κόκορες και περισσότερες από τα κοτόπουλα. Η ομάδα κάθε μέλους εκλέγεται τυχαία. Δεν είναι όλες οι κότες μητέρες και οι μητέρες εκλέγονται τυχαία από τις διαθέσιμες κότες. Τέλος, το πλήθος των κοτόπουλων είναι μικρότερο από το πλήθος των μητέρων. Χρησιμοποιώντας τη θεωρία συνόλων, μπορούμε να αναπαραστήσουμε όλες αυτές τις παραδοχές ως εξής:

$$MN \subset HN, \quad (1)$$

$$PN = RN \cup HN \cup CN, \quad (2)$$

$$|HN| \geq |RN|, \quad (3)$$

$$|HN| \geq |CN|, \quad (4)$$

όπου MN αναπαριστά το σύνολο των κοτών που είναι μητέρες, HN το σύνολο των κοτών, RN το σύνολο των κοκόρων, CN το σύνολο των κοτόπουλων και PN το σύνολο όλων των μελών του σμήνους.

Κατά το βήμα της ανανέωσης, η θέση κάθε μέλους μεταβάλλεται ανάλογα με την κατηγορία στην οποία ανήκει (κόκορας, κότα ή κοτόπουλο), αποτυπώνοντας με αυτόν τον τρόπο τη διαφορετική ικανότητα αναζήτησης τροφής κάθε μέλους. Οι τιμές καταλληλότητας ανανεώνονται επίσης βάσει αυτής της ικανότητας. Η ικανότητα ενός κόκορα να αναζητήσει τροφή βασίζεται στην τιμή καταλληλότητάς του και η μαθηματική αποτύπωση αυτής της ικανότητας είναι η εξής:

$$x_{i,j}^{t+1} = x_{i,j}^t \times (1 + \text{randn}(0, \sigma^2)) \quad (5)$$

$$\text{Αν } f_i \leq f_k$$

$$\sigma^2 = 1 \quad (6)$$

Αλλιώς

$$\sigma^2 = \exp\left(\frac{(f_k - f_i)}{|f_i| + \varepsilon}\right) \quad (7)$$

Όπου $x_{i,j}^t$, $j \in [1, \dots, D]$ είναι η j συνιστώσα της θέσης του μέλους με δείκτη i του σμήνους που ψάχνει τροφή στον D -διάστατο χώρο αναζήτησης του προβλήματος την χρονική στιγμή t , f_i είναι η τιμή καταλληλότητας του μέλους του οποίου ανανεώνουμε την θέση και f_k είναι η τιμή καταλληλότητας ενός άλλου κόκορα του σμήνους, ο οποίος εκλέγεται τυχαία. Η $\text{randn}(0, \sigma^2)$ είναι μια συνάρτηση που μας δίνει μια τυχαία μεταβλητή από μια κατανομή Gauss με μέση τιμή 0 και διακύμανση σ^2 . Τέλος, η ε είναι μια πολύ μικρή σταθερά που χρησιμεύει στην αποφυγή διαίρεσης με το 0.

Οι κότες ακολουθούν τον κόκορα που είναι ο επικεφαλής της ομάδας τους. Επιπλέον, έχουν την τάση να κλέβουν τροφή από άλλους κόκορες ή κότες. Μαθηματικά αυτό αποτυπώνεται ως εξής:

$$x_{i,j}^{t+1} = x_{i,j}^t + S1 \times \text{rand} \times (x_{r1,j}^t - x_{i,j}^t) + S2 \times \text{rand} \times (x_{r2,j}^t - x_{i,j}^t) \quad (8)$$

$$S1 = \exp\left(\frac{(f_i - f_{r1})}{|f_i| + \varepsilon}\right) \quad (9)$$

$$S2 = \exp(f_{r2} - f_i) \quad (10)$$

Όπου rand είναι ένας τυχαίος αριθμός μεταξύ 0 και 1. Το $r1 \in [1, N]$ είναι ο δείκτης του επικεφαλής κόκορα της ομάδας της κότας της οποίας τη θέση ανανεώνουμε. Το $r2 \in [1, N]$ είναι ο δείκτης ενός κόκορα ή κότας, ο οποίος εκλέγεται τυχαία με τέτοιο τρόπο ώστε το $r2$ να μην είναι ίσο με το $r1$.

Τέλος, τα κοτόπουλα έχουν την τάση να ακολουθούν τις μητέρες τους και αυτήν την τάση την αποτυπώνουμε ως εξής:

$$x_{i,j}^{t+1} = x_{i,j}^t + FL \times (x_{m,j}^t - x_{i,j}^t) \quad (11)$$

Όπου $x_{m,j}^t$ είναι η θέση της μητέρας του κοτόπουλου του οποίου τη θέση ανανεώνουμε. Η FL είναι μία παράμετρος του αλγορίθμου με τυπικές τιμές της να ανήκουν στο διάστημα $[0,2]$.

1.1.3 Παράμετροι

Εκτός από τις τυπικές παραμέτρους που συναντάμε σε έναν εξελικτικό αλγόριθμο, όπως είναι το πλήθος των πρακτόρων ή, στην προκειμένη περίπτωση, το μέγεθος του σμήνους (N) και ο αριθμός των γενεών (gen), η βελτιστοποίηση μέσω σμήνους κοτών έχει έξι παραμέτρους που είναι ειδικές για τον αλγόριθμο. Αυτές καθορίζουν το διαχωρισμό του σμήνους σε κόκορες (RN), κότες (HN) και κοτόπουλα (CN), το αριθμό των κοτών που είναι μητέρες (MN), τον αριθμό των βημάτων ανάμεσα σε δύο διαδοχικές αναθεωρήσεις της ιεραρχίας (G) και, τέλος, την παράμετρο FL που εμφανίζεται στην εξίσωση (11).

Σε αντιστοιχία με ένα πραγματικό σμήνος, οι κότες καταλαμβάνουν το μεγαλύτερο ποσοστό του πληθυσμού, καθώς είναι πιο ωφέλιμο λόγω της δυνατότητάς τους να κάνουν αυγά. Επειδή, όμως, δεν κάνουν όλες οι κότες αυγά ταυτόχρονα, δεν είναι όλες οι κότες μητέρες. Μια διαφορά με ένα πραγματικό σμήνος είναι ότι παρόλο που μια μητέρα μπορεί να μεγαλώσει πολλά κοτόπουλα, ο αριθμός των κοτόπουλων είναι μικρότερος των ενήλικων κοτών και κοκόρων.

Όσον αφορά την παράμετρο G , χρειάζεται προσοχή κατά την επιλογή της, καθώς αν η τιμή της είναι πολύ μεγάλη, μπορεί να επηρεαστεί η σύγκλιση του αλγορίθμου. Αντιθέτως, αν είναι μικρή, μπορεί να παγιδευτούμε σε τοπικά μέγιστα του χώρου αναζήτησης. Αυτό συμβαίνει διότι καθορίζει τις σχέσεις ανάμεσα στα μέλη του σμήνους και συνεπώς από ποιον επηρεάζεται το καθένα. Γενικά, η καταλληλότερη τιμή εξαρτάται κυρίως από το πρόβλημα που επιλύουμε. Η παράμετρος FL καθορίζει σε ποιον βαθμό κάθε κοτόπουλο επηρεάζεται από τη μητέρα του. Σε πειράματα που εκτελέστηκαν σε 12 διαφορετικά προβλήματα (Meng et al. 2014) επιβεβαιώθηκε ότι τιμές του G που ανήκουν στο διάστημα $[2,20]$ λειτουργούν καλά για τα περισσότερα προβλήματα.

Επίσης, απόρροια των πειραμάτων ήταν ότι ο αλγόριθμος μπορεί να επιτύχει καλά αποτελέσματα για τιμές της παραμέτρου FL στο διάστημα $[0.4, 1]$.

1.1.4 Πλεονεκτήματα και αδυναμίες

Ο αλγόριθμος βελτιστοποίησης μέσω σμήνους κοτών έχει μια ειδοποιό διαφορά που τον διαχωρίζει από άλλους αλγορίθμους νοημοσύνης σμήνους, η οποία είναι ο διαχωρισμός του πληθυσμού σε τρεις ομάδες (κόκορες, κότες και κοτόπουλα). Αυτό επιτρέπει την αποδοτική χρήση όλου του σμήνους, καθώς και την επίτευξη μίας καλής ισορροπίας ανάμεσα στην εξερεύνηση και στην εκμετάλλευση του χώρου αναζήτησης. Με τον όρο εξερεύνηση εννοούμε τη διερεύνηση προηγούμενως ανεξερεύνητων περιοχών του χώρου που δεν έχουν κοινά σημεία με τις περιοχές που έχουμε ήδη επισκεφτεί, ενώ η εκμετάλλευση αναφέρεται στη διερεύνηση περιοχών κοντινών σε αυτές που έχουμε ήδη επισκεφτεί με στόχο τη βελτιστοποίηση καλών λύσεων που έχουν βρεθεί προηγούμενως. Η ισορροπία εξερεύνησης και εκμετάλλευσης, επομένως, είναι πολύ σημαντική για έναν εξελικτικό αλγόριθμο και επηρεάζει σαφώς την απόδοσή του. Στον CSO, ο τρόπος με τον οποίο οι κόκορες αναζητούν τροφή αντιστοιχεί στην εξερεύνηση του χώρου αναζήτησης, ενώ ο τρόπος με τον οποίο αναζητούν τροφή οι κότες και τα κοτόπουλα συμβολίζει την εκμετάλλευση του. Δηλαδή, οι λύσεις που βρίσκουν οι κόκορες βελτιώνονται ακόμη περισσότερο από τις κότες και τα κοτόπουλα. Με αυτόν τον τρόπο, επιτυγχάνεται εν τέλει η επιθυμητή ισορροπία.

Η βελτιστοποίηση μέσω σμήνους κοτών θεωρείται ότι υπερτερεί σε σχέση με άλλους αλγορίθμους εμπνευσμένους από τη φύση, όπως είναι ο αλγόριθμος βελτιστοποίησης μέσω σμήνους σωματιδίων (PSO – Particle Swarm Optimization) και ο DE (Differential Evolution). Στην περίπτωση του PSO, αν στο σμήνος έχουμε μόνο κότες και αντιστοιχίσουμε τις παραμέτρους $S1$ και $S2$ του CSO στους συντελεστές εκμάθησης $c1$ και $c2$ του PSO, τότε ο αλγόριθμος μας μετατρέπεται στον αλγόριθμο PSO. Όσον αφορά τον αλγόριθμο DE, μπορούμε να συσχετίσουμε τον τύπο με τον οποίο ανανεώνουμε τη θέση ενός κοτόπουλου με τον τύπο μετάλλαξης του DE.

Ενώ, γενικά, ο αλγόριθμος φέρνει καλά αποτελέσματα και υπερτερεί έναντι άλλων αλγορίθμων, λόγω του μοναδικού του χαρακτηριστικού να διαχωρίζει το σμήνος σε κατηγορίες, εμφανίζει κάποια μειονεκτήματα. Ο αλγόριθμος εξαρτάται από έξι διαφορετικές παραμέτρους για τις οποίες πρέπει να επιλέξουμε τιμές. Ένας άστοχος συνδυασμός παραμέτρων μπορεί εύκολα να οδηγήσει ένα μέλος του σμήνους να παγιδευτεί σε τοπικό μέγιστο. Επιπλέον, τα κοτόπουλα ακολουθούν τις μητέρες τους και οι κότες ακολουθούν τους κόκορες. Συνεπώς, αν οι κόκορες παγιδευτούν σε τοπικά μέγιστα τότε και οι κότες και τα κοτόπουλα θα παγιδευτούν με τη σειρά τους, οδηγώντας δυστυχώς τον αλγόριθμο σε πρόωρη σύγκλιση. Επίσης, τα κοτόπουλα επικοινωνούν μόνο με τις μητέρες τους και όχι με τους κόκορες. Επομένως, ακόμη και στην περίπτωση που μόνο οι κότες παγιδευτούν σε τοπικά μέγιστα, θα οδηγηθούμε και στην παγίδευση των κοτόπουλων. Αυτό έχει ως αποτέλεσμα τη μη αποδοτική εκμετάλλευση του χώρου αναζήτησης και τη διατάραξη της ισορροπίας εξερεύνησης και εκμετάλλευσης.

1.1.5 Πολύ-βηματική CSO

Η πολύ-βηματική βελτιστοποίηση μέσω σμήνους κοτών (multi-step CSO) είναι μια παραλλαγή της βασικής εκδοχής του αλγορίθμου που επιχειρεί να αντιμετωπίσει την κύρια αδυναμία της βασικής εκδοχής του αλγορίθμου, δηλαδή το γεγονός ότι μπορεί να

παγιδευτεί σε τοπικά μέγιστα. Προτάθηκε για πρώτη φορά από τους Irsalinda et al. το 2017. Ο τρόπος με τον οποίο επιτυγχάνει να ξεπεράσει αυτό το μειονέκτημα είναι να εφαρμόσει τους μαθηματικούς τύπους για την ανανέωση θέσης κοκόρων, κοτών και κοτόπουλων σε όλο τον πληθυσμό. Στον τύπο ανανέωσης θέσης της κότας, αντί για τον επικεφαλής κόκορα ($r1$) διαλέγουμε τυχαία ένα μέλος του σμήνους με τέτοιο τρόπο ώστε να μην είναι το μέλος του οποίου τη θέση ανανεώνουμε ή με το άλλο μέλος που επίσης διαλέγουμε τυχαία ($r1 \neq r2 \neq i$). Αντίστοιχα, στον τύπο ανανέωσης θέσης του κοτόπουλου, αντί για τη μητέρα του, διαλέγουμε ένα άλλο μέλος τυχαία. Δεδομένων αυτών των αλλαγών στους τύπους, δεν υπάρχει πλέον ανάγκη για τις παραμέτρους που σχετίζονται με την ιεραρχία, δηλαδή τις RN , HN , CN , MN και G . Αυτό απλοποιεί τον αλγόριθμο, αφού η μόνη παράμετρος από την οποία εξαρτάται πλέον είναι η FL και, επομένως, αντιμετωπίζεται το πρόβλημα άστοχου συνδυασμού παραμέτρων.

Το βήμα της ανανέωσης του αλγορίθμου διαιρείται σε δύο υποβήματα. Το πρώτο είναι η διαφοροποίηση, που αντιστοιχεί στην εξερεύνηση του χώρου αναζήτησης. Σε αυτό το βήμα, εφαρμόζουμε τον τύπο που αναφέρεται στην εξίσωση (8) σε όλα τα μέλη του σμήνους. Στο δεύτερο βήμα της ανανέωσης, που είναι η εντατικοποίηση και αντιστοιχεί στην εκμετάλλευση του χώρου αναζήτησης, εφαρμόζουμε διαδοχικά τις εξισώσεις (5) και (11) σε όλα τα μέλη του σμήνους. Συνεπώς, κάθε μέλος αναζητεί αρχικά το ολικό μέγιστο μέσω της εξίσωσης (8) και στη συνέχεια, βελτιώνουμε τις λύσεις που βρίσκουμε και αποτελούν τοπικά μέγιστα μέσω των εξισώσεων (5) και (11).

Η πολύ-βηματική βελτιστοποίηση εξετάστηκε πάνω σε 7 διαφορετικές συναρτήσεις (Irsalinda et al. 2017) και συγκρίθηκε με αλγορίθμους όπως ο PSO, ο CS (Cuckoo Search), ο γενετικός αλγόριθμος (GA - Genetic algorithm) και η βασική εκδοχή του CSO. Τα αποτελέσματα έδειξαν ότι η πολύ-βηματική εκδοχή του αλγορίθμου βρίσκει καλύτερες λύσεις και έχει μεγαλύτερη ταχύτητα σύγκλισης.

1.1.6 Παραλλαγές CSO

Από το 2014 όπου εφευρέθηκε η βασική εκδοχή του αλγορίθμου CSO έχουν προταθεί αρκετές παραλλαγές του. Παρόλο που ο αλγόριθμος έχει αποδείξει την αποτελεσματικότητά του στην επίλυση πλήθους προβλημάτων, ο κύριος στόχος των παραλλαγών είναι να αντιμετωπίσουν τις βασικές αδυναμίες του αλγορίθμου, όπως το γεγονός ότι παγιδευτεί σε τοπικά μέγιστα. Κάποιες από αυτές τις παραλλαγές είναι οι εξής:

1. Τροποποιημένη CSO (mCSO – modified CSO)

Προτάθηκε από τους Chen et al. (2015). Στην εξίσωση (8) οι τυχαίοι αριθμοί που χρησιμοποιούμε προέρχονται από ομοιόμορφες κατανομές.

2. Βελτιωμένη CSO (ICSO – Improved CSO)

Προτάθηκε από τους Wu et al. (2015). Σε αυτή την εκδοχή του αλγορίθμου η εξίσωση ανανέωσης της θέσης ενός κοτόπουλου μετατρέπεται ώστε να μπορεί να μαθαίνει εκτός από τη μητέρα του και από κάποιον τυχαίο κόκορα του σμήνους.

3. Μεταλλαγμένη CSO (MCSO – Mutation CSO)

Προτάθηκε από τους Wang et al. (2017). Η διαφορά της παραλλαγής αυτής είναι ότι αφού βρούμε τη νέα θέση ενός κοτόπουλου μέσω της εξίσωσης (11), την πολλαπλασιάζουμε με τον παράγοντα $(1 + 0.5\eta)$, όπου η είναι μια τυχαία μεταβλητή που ακολουθεί την Γκαουσιανή κατανομή.

4. Μεταλλαγμένη CSO βασισμένη σε μη γραμμικά αδρανή βάρη (NW-MCSO –

Mutation CSO based on non linear inertia weight)

Αυτή η παραλλαγή προτάθηκε από τους Wang et al. (2017) και μετατρέπει την εξίσωση ανανέωσης θέσης του κόκορα, ώστε η θέση του να μεταβάλλεται αναλόγως με ένα μη γραμμικό βάρος $w(t)$, το οποίο φθίνει με την πάροδο του χρόνου.

5. CSO βασισμένη σε μονομερή αναταραχή και σωματιδιακή ανανέωση (MPCSO – Monomer turbulence and particle renovation based CSO)

Οι Shi et al. (2018) επιχειρούν να βελτιώσουν τον βασικό αλγόριθμο εισάγοντας μονομερή αναταραχή και σωματιδιακή ανανέωση, την οποία δανείζονται από τον αλγόριθμο PSO, στους μηχανισμούς ανανέωσης θέσης κοκόρων και κοτών.

6. Χαοτική CSO (CCSO – Chaotic CSO)

Η παραλλαγή αυτή από τους Ahmed et al. (2017) συνδυάζει τη θεωρία του χάους με την CSO. Χρησιμοποιήθηκε για την επίλυση προβλημάτων επιλογής χαρακτηριστικών.

7. Δυαδική βελτιωμένη CSO (BGCSO – Binary Improved CSO)

Οι Han και Liu (2017) πρότειναν αυτήν την παραλλαγή, όπου εφαρμόζουμε έναν τελεστή μετάλλαξης στο μέλος του σμήνους με τη χειρότερη τιμή καταλληλότητας.

8. CSO με βελτιωμένα όρια (IBCSO – Improved Boundary CSO)

Οι Chen et al. (2016) πρότειναν μια βελτιωμένη εκδοχή του αλγορίθμου, η οποία χειρίζεται με καλύτερο τρόπο τους περιορισμούς του χώρου αναζήτησης.

9. CSO βασισμένη στην εκμάθηση μέσω αντιπαλότητας (EOCSO – CSO based on elite opposition based learning)

Σε αυτή την εκδοχή του αλγορίθμου από τους Qu et al. (2017) έχουμε δύο αλλαγές. Πρώτον, ο μηχανισμός ανανέωσης θέσης των κοκόρων χρησιμοποιεί τυχαίες μεταβλητές που ακολουθούν την προσαρμοστική κατανομή t (adaptive t -distribution) αντί της κατανομής Gauss. Δεύτερον, στην εξίσωση ανανέωσης θέσης κοτών εισάγουμε την εκμάθηση μέσω αντιπαλότητας με σκοπό να ενισχύσουμε την ποικιλομορφία του σμήνους.

10. Προσαρμοστική CSO (ACSO – Adaptive CSO)

Οι Ahmed et al. (2016) πρότειναν την προσαρμοστική CSO για την επίλυση του προβλήματος αναγνώρισης κοινωνικών ομάδων σε κοινωνικά δίκτυα. Πρόκειται για μια διακριτή μορφή του βασικού αλγορίθμου, όπου σε κάθε επανάληψη αντικαθιστούμε το χειρότερο μέλος του σμήνους με το καλύτερο. Αυτό οδηγεί σε καλύτερη εξερεύνηση του χώρου αναζήτησης.

11. Κβαντική CSO (QCSO – Quantum CSO)

Η κβαντική CSO προτάθηκε από τους Meng και Li (2017) και βασίζεται στην θεωρία των κβάντων. Προσπαθεί να βελτιώσει την ικανότητα αναζήτησης τροφής των κοτόπουλων, ώστε να βελτιωθεί η ικανότητα αναζήτησης του σμήνους συνολικά. Ο μηχανισμός ανανέωσης θέσης κοτόπουλων μετατρέπεται με δύο τρόπους. Αφ' ενός, εισάγουμε στην εξίσωση έναν τυχαίο αριθμό και αφ' ετέρου, η παράμετρος FL μεταβάλλεται πλέον σε κάθε επανάληψη με φθίνοντα τρόπο.

Εκτός από όλες τις προαναφερθείσες παραλλαγές η βελτιστοποίηση μέσω σμήνους κοτών έχει συνδυαστεί με πολλούς άλλους αλγορίθμους, όπως είναι ο αλγόριθμος νυχτερίδων (BA - Bat Algorithm), ο Cuckoo Search (CS), ο αλγόριθμος TLBO και ο DE, με σκοπό την παραγωγή υβριδικών αλγορίθμων.

1.2 Αλγόριθμος Βελτιστοποίησης του Αρχιμήδη

Ο αλγόριθμος βελτιστοποίησης του Αρχιμήδη (Archimedes Optimization Algorithm – AOA) είναι ένας καινούργιος αλγόριθμος, ο οποίος ανήκει στην οικογένεια μετα-ευρετικών αλγορίθμων τεχνητής νοημοσύνης βασισμένων σε φυσικά φαινόμενα. Συγκεκριμένα, πρόκειται για έναν αλγόριθμο που προσπαθεί να προσομοιώσει έναν νόμο της Φυσικής, την αρχή της άνωσης του Αρχιμήδη. Τις τελευταίες δεκαετίες, έχουν αναπτυχθεί πολλοί αλγόριθμοι των οποίων το επίκεντρο είναι η μίμηση φυσικών νόμων, όπως είναι οι νόμοι του Coulomb και του Newton, ο νόμος του Henry, αλγόριθμοι βασισμένοι στην βαρυτική θεωρία ή στην θεωρία ακτίνων. Έχουμε, επίσης, αλγορίθμους που αντλούν την έμπνευσή τους από αλλά πεδία της επιστήμης, όπως τη χημεία, όπου υπάρχει αλγόριθμος που μιμείται τις μοριακές αλληλεπιδράσεις σε μία χημική αντίδραση, ή τα μαθηματικά, όπου έχουμε αλγορίθμους που χρησιμοποιούν τριγωνομετρικές συναρτήσεις ή μορφοκλάσματα (fractals).

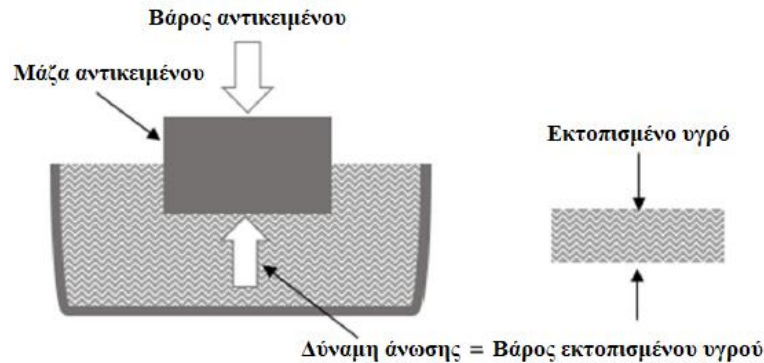
Οι αλγόριθμοι που βασίζονται σε φυσικά φαινόμενα όταν μετατρέπονται σε αλγορίθμους βελτιστοποίησης παράγουν σημαντικά και ικανοποιητικά αποτελέσματα. Για αυτό το λόγο, προτείνονται συνεχώς καινούργιοι αλγόριθμοι βασισμένοι σε προηγουμένως ανεξερεύνητα φαινόμενα. Παρόλα αυτά, επειδή κάποια βασικά στοιχεία που πρέπει να διαθέτει ένας μετα-ευρετικός αλγόριθμος είναι η απλότητά του, η ευκολία χρήσης του και η ικανότητά του να μην παγιδευτεί σε τοπικά μέγιστα, οι περισσότεροι δεν καταφέρνουν να κρατήσουν το ενδιαφέρον της επιστημονικής κοινότητας και συχνά εξαφανίζονται από τη βιβλιογραφία.

1.2.1 Έμπνευση

Σύμφωνα με την Ιστορία, ο Αρχιμήδης έζησε στις Συρακούσες τον 3^ο αιώνα π.Χ. Μία από τις σημαντικότερες ανακαλύψεις του, αν όχι η σημαντικότερη, είναι η αρχή του Αρχιμήδη, η οποία εκφράζει τον νόμο της άνωσης. Η ιστορία γύρω από την ανακάλυψη του νόμου της άνωσης, ισχυρίζεται ότι ο βασιλιάς Ιέρων Β' των Συρακουσών είχε παραγγείλει να του φτιάξουν ένα στέμμα από ατόφιο χρυσάφι. Επειδή, υποψιαζόταν ότι ο χρυσοχόος που του είχε αναθέσει την κατασκευή του στέμματος τον είχε εξαπατήσει και είχε νοθεύσει τον χρυσό με ασήμι, κάλεσε τον Αρχιμήδη και του ανέθεσε να εξακριβώσει αυτό το θέμα. Φυσικά, αυτό έπρεπε να γίνει χωρίς να καταστραφεί το στέμμα. Καθώς έκανε μπάνιο, παρατήρησε ότι η στάθμη του νερού ανέβαινε όταν βυθιζόταν ο ίδιος στο νερό. Σκέφτηκε, ότι με τον ίδιο τρόπο θα μπορούσε να βυθίσει το στέμμα στο νερό και μια ποσότητα χρυσού, ίση με αυτήν που θα έπρεπε να περιέχει κανονικά το στέμμα, ώστε να διαπιστώσει αν υπάρχει κάποια διαφορά. Από τον ενθουσιασμό του για την ανακάλυψή του, λέγεται ότι βγήκε γυμνός στον δρόμο και άρχισε να φωνάζει «Εύρηκα, Εύρηκα!». Η ιστορία τελειώνει λέγοντας ότι ο χρυσός είχε όντως νοθευτεί.

Ο νόμος της άνωσης, λοιπόν, συνδέει ένα αντικείμενο που βυθίζεται σε κάποιο υγρό με τη δύναμη άνωσης που του ασκείται. Αντίστοιχα, η άνωση ενός αντικειμένου εξαρτάται από μια δύναμη που του ασκείται προς τα πάνω και είναι ίση με το βάρος του υγρού που εκτοπίζει. Αν το βάρος του αντικειμένου είναι μεγαλύτερο από το βάρος του εκτοπισμένου υγρού, τότε το αντικείμενο θα βυθιστεί. Αντίθετα, αν τα δύο βάρη είναι ίσα, το αντικείμενο θα ισορροπήσει στο βάθος του υγρού που ήδη βρίσκεται. Η ιστορία γύρω από την ανακάλυψη του νόμου δεν αναφέρεται σε κάποιο από τα έργα του Αρχιμήδη. Η αρχή του αναφέρεται στο έργο του «Περί επιπλεόντων σωμάτων». Η αρχή, όπως περιγράφεται εκεί, θα μπορούσε να οδηγήσει στη σύγκριση της πυκνότητας του

στέμματος με ένα κομμάτι στερεού χρυσού με την εξισορρόπηση του στέμματος και του χρυσού αναφοράς σε ένα ζυγό και, εν συνεχεία, τη βύθιση του συστήματος στο νερό. Η διαφορά πυκνότητας των δύο σωμάτων θα προκαλούσε την αντίστοιχη κλίση του ζυγού.



Σχήμα 2: Βύθιση αντικειμένου σε υγρό και ο όγκος του εκτοπισμένου υγρού

1.2.2 Θεωρητικό υπόβαθρο

Ο αλγόριθμος βελτιστοποίησης του Αρχιμήδη αναπτύχθηκε από τους Hashim et al. και προτάθηκε τον Σεπτέμβριο του 2020. Ο τρόπος με τον οποίο αναζητεί την καλύτερη δυνατή λύση ενός προβλήματος βελτιστοποίησης είναι μέσω της προσομοίωσης του νόμου της άνωσης. Όπως και σε άλλους αλγόριθμους που ανήκουν στην ίδια κατηγορία, έχουμε ένα σύνολο πρακτόρων, τον πληθυσμό, όπου ο κάθε πράκτορας προσπαθεί να βρει μια λύση του προβλήματος. Στον συγκεκριμένο αλγόριθμο, ο πληθυσμός μας είναι μια συλλογή αντικειμένων, τα μέλη του είναι αντικείμενα τα οποία έχουμε βυθίσει σε κάποιο υγρό και το υγρό αντιπροσωπεύει τον χώρο αναζήτησης του προβλήματος. Η πυκνότητα, ο όγκος και η επιτάχυνση είναι οι τρεις σημαντικότερες ιδιότητες που χρησιμοποιούνται εκτενώς για τον προσδιορισμό της θέσης κάθε αντικειμένου. Η βασική ιδέα του αλγορίθμου είναι να φτάσει ένα σημείο ισορροπίας, όπου η συνολική δύναμη που ασκείται πάνω σε κάθε αντικείμενο από το υγρό να είναι μηδενική.

Η αρχή του Αρχιμήδη, όπως αναφέρθηκε προηγουμένως, δηλώνει ότι όταν βυθίσουμε ένα αντικείμενο ολικώς ή μερικώς σε κάποιο υγρό, τότε το υγρό θα ασκήσει μια δύναμη προς τα πάνω στο αντικείμενο ίση με το βάρος του υγρού που εκτοπίζεται από το αντικείμενο. Αν υποθέσουμε ότι έχουμε πολλά αντικείμενα βυθισμένα στο ίδιο υγρό, τότε, το καθένα από αυτά προσπαθεί να φτάσει σε μία κατάσταση ισορροπίας και, λόγω της διαφορετικής πυκνότητας και όγκου του κάθε αντικειμένου, θα έχουν και διαφορετικές επιταχύνσεις. Όταν ένα αντικείμενο φτάσει στην κατάσταση ισορροπίας, η δύναμη άνωσης είναι ίση με το βάρος του αντικειμένου:

$$F_b = W_o,$$
$$\rho_b v_b a_b = \rho_o v_o a_o \quad (12)$$

$$a_o = \frac{p_b v_b a_b}{p_o v_o} \quad (13)$$

Όπου F_b είναι η δύναμη άνωσης, W_o το βάρος του αντικειμένου, p η πυκνότητα, v ο όγκος, a_o η επιτάχυνση του αντικειμένου, a_b η επιτάχυνση της βαρύτητας και οι δείκτες b και o αναφέρονται στο υγρό και στο αντικείμενο αντίστοιχα. Αν υπάρχει και δύναμη που οφείλεται σε σύγκρουση με κάποιο γειτονικό αντικείμενο (r), τότε η κατάσταση ισορροπίας αλλάζει και εκφράζεται ως:

$$\begin{aligned} F_b &= W_o, \\ W_b - W_r &= W_o, \\ p_b v_b a_b - p_r v_r a_r &= p_o v_o a_o \end{aligned} \quad (14)$$



Σχήμα 3: Βύθιση πολλαπλών αντικειμένων σε υγρό

1.2.3 Αλγόριθμος

Ο αλγόριθμος βελτιστοποίησης του Αρχιμήδη, λειτουργεί γενικά όπως οι περισσότεροι αλγόριθμοι που βασίζονται σε έναν πληθυσμό πρακτόρων. Χωρίζεται σε δύο φάσεις, με την πρώτη να είναι η αρχικοποίηση των αντικειμένων και τη δεύτερη να είναι η ανανέωση της θέσης τους μέσα στο υγρό. Σε κάθε επανάληψη (γενιά) του αλγορίθμου, κάθε αντικείμενο βρίσκει μια λύση του προβλήματος, την οποία αποτιμά μέσω της συνάρτησης καταλληλότητας και υιοθετεί εφόσον είναι καλύτερη από την καλύτερη λύση που έχει βρει μέχρι τώρα. Πριν προχωρήσουμε, όμως, στην κατασκευή μιας καινούργιας λύσης, ανανεώνουμε την πυκνότητα και τον όγκο κάθε αντικειμένου. Επίσης, ανανεώνουμε την επιτάχυνση κάθε αντικειμένου με βάση τις συγκρούσεις του με γειτονικά του αντικείμενα. Για αυτόν τον λόγο, σημειώνεται σε κάθε επανάληψη το αντικείμενο που έχει βρει την καλύτερη λύση μέχρι τώρα από όλη τη συλλογή.

Κατά τη φάση της αρχικοποίησης, κάθε αντικείμενο αρχικοποιείται με τυχαίες τιμές πυκνότητας, όγκου, επιτάχυνσης και θέσης και προχωράει στην κατασκευή της πρώτης λύσης του. Οι μαθηματικοί τύποι για τις αρχικοποιήσεις είναι οι ακόλουθοι:

$$O_i = lb_i + rand \times (ub_i - lb_i) \quad (15)$$

$$den_i = rand$$

$$vol_i = rand \quad (16)$$

$$acc_i = lb_i + rand \times (ub_i - lb_i) \quad (17)$$

Όπου O_i είναι το αντικείμενο με δείκτη i στη συλλογή μας με $i = 1, 2, \dots, N$ και N να είναι το πλήθος των αντικειμένων της συλλογής. Η πυκνότητα και ο όγκος του είναι den_i και vol_i αντίστοιχα. Η επιτάχυνσή του είναι acc_i και, τέλος, $rand$ είναι ένας τυχαίος αριθμός στο διάστημα $[0,1]$. Αφού τελειώσουμε με την αρχικοποίηση όλων των αντικειμένων, επιλέγουμε το αντικείμενο με την καλύτερη τιμή καταλληλότητας και αναθέτουμε τη θέση, την πυκνότητα, τον όγκο και την επιτάχυνσή του στις μεταβλητές x_{best} , den_{best} , vol_{best} και acc_{best} αντίστοιχα.

Στη συνέχεια, προχωρούμε στο δεύτερο κομμάτι του αλγορίθμου, την ανανέωση, όπου σε κάθε επανάληψη, ανανεώνουμε τις τιμές πυκνότητας και όγκου μέσω των ακόλουθων μαθηματικών τύπων:

$$\begin{aligned} den_i^{t+1} &= den_i^t + rand \times (den_{best} - den_i^t) \\ vol_i^{t+1} &= vol_i^t + rand \times (vol_{best} - vol_i^t) \end{aligned} \quad (18)$$

Το στάδιο της ανανέωσης χωρίζεται περαιτέρω σε δύο υποστάδια, το στάδιο της εξερεύνησης και το στάδιο της εκμετάλλευσης. Αυτά οριοθετούνται μέσω του τελεστή μεταφοράς (transfer operator - TF) και του συντελεστή πυκνότητας (density factor - d). Ο τελεστής μεταφοράς προσομοιώνει την εμφάνιση πολλαπλών συγκρούσεων στην αρχή του πειράματος, οι οποίες σταδιακά μειώνονται. Παρομοίως, ο συντελεστής πυκνότητας βοηθάει τον αλγόριθμο στην αλλαγή από καθολική σε τοπική αναζήτηση. Οι παράμετροι αυτές μεταβάλλονται με τον χρόνο και οι μαθηματικοί τύποι τους είναι οι εξής:

$$TF = \exp\left(\frac{t - t_{max}}{t_{max}}\right) \quad (19)$$

$$d^{t+1} = \exp\left(\frac{t_{max} - t}{t_{max}}\right) - \left(\frac{t}{t_{max}}\right) \quad (20)$$

Όπου t και t_{max} είναι ο αριθμός της τρέχουσας επανάληψης και του μεγίστου αριθμού επαναλήψεων, αντίστοιχα. Ο τελεστής μεταφοράς αυξάνεται σταδιακά από e^{-1} έως 1, ενώ ο συντελεστής πυκνότητας μειώνεται σταδιακά από e έως 0, δίνοντας τη δυνατότητα σύγκλισης σε μία υποσχόμενη περιοχή του χώρου. Αξίζει να σημειωθεί ότι ο σωστός χειρισμός του συντελεστή πυκνότητας εξασφαλίζει ισορροπία ανάμεσα σε εξερεύνηση και εκμετάλλευση.

Όταν $TF \leq 0.5$, βρισκόμαστε στο στάδιο της εξερεύνησης, όπου έχουμε συγκρούσεις μεταξύ των αντικειμένων, όποτε κατά την ανανέωση της επιτάχυνσης ενός αντικειμένου συνυπολογίζουμε τη σύγκρουση του με κάποιο τυχαίο αντικείμενο της συλλογής:

$$acc_i^{t+1} = \frac{den_{mr} + vol_{mr} \times acc_{mr}}{den_i^{t+1} \times vol_i^{t+1}} \quad (21)$$

Όπου mr είναι ο δείκτης του τυχαίου αντικειμένου και den_{mr} , vol_{mr} και acc_{mr} είναι αντίστοιχα η πυκνότητα, ο όγκος και η επιτάχυνσή του. Το όριο που βάζουμε στον τελεστή μεταφοράς εξασφαλίζει ότι περίπου το ένα τρίτο των επαναλήψεων θα σπαταληθεί στο στάδιο της εξερεύνησης. Αν αλλάξουμε το όριο, αλλάζει και το ποσοστό των επαναλήψεων που καταναλώνεται σε κάθε στάδιο.

Αν $TF > 0.5$ βρισκόμαστε στο στάδιο της εκμετάλλευσης, όπου δεν υπάρχουν πλέον συγκρούσεις και η επιτάχυνση υπολογίζεται ως εξής:

$$acc_i^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_i^{t+1} \times vol_i^{t+1}} \quad (22)$$

Στη συνέχεια, κανονικοποιούμε τις επιταχύνσεις. Ο λόγος της κανονικοποίησης είναι η μετάβαση από τη φάση της εξερεύνησης στη φάση της εκμετάλλευσης. Ένα αντικείμενο που απέχει πολύ από τη βέλτιστη λύση θα έχει μεγάλη επιτάχυνση και θα βρίσκεται ακόμα στη φάση της εξερεύνησης. Η κανονικοποιημένη επιτάχυνση καθορίζει τη μεταβολή της θέσης κάθε αντικειμένου. Φυσιολογικά, η τιμή της επιτάχυνσης αρχικά είναι μεγάλη και με την πάροδο του χρόνου μειώνεται, οδηγώντας με αυτόν τον τρόπο τα αντικείμενα προς τη βέλτιστη λύση και μακριά από τοπικά μέγιστα. Παρόλα αυτά, κάποια αντικείμενα μπορεί να χρειάζονται περισσότερο χρόνο εξερευνώντας τον χώρο από το φυσιολογικό. Τελικά, επιτυγχάνεται με αυτόν τον τρόπο η ισορροπία εξερεύνησης και αναζήτησης. Ο τύπος της κανονικοποίησης έχει ως εξής:

$$acc_{i-norm}^{t+1} = u \times \frac{acc_i^{t+1} + \min(acc)}{\max(acc) - \min(acc)} + l \quad (23)$$

Όπου u και l είναι το εύρος της κανονικοποίησης και έχουν τιμές 0.9 και 0.1 αντίστοιχα, $\min(acc)$ και $\max(acc)$ είναι οι επιταχύνσεις των αντικειμένων με τη μικρότερη και μεγαλύτερη επιτάχυνση και acc_{i-norm}^{t+1} είναι η κανονικοποιημένη επιτάχυνση.

Μπορούμε να υπολογίσουμε πλέον την καινούργια θέση του αντικειμένου. Ο υπολογισμός αυτός εξαρτάται από το στάδιο της ανανέωσης στο οποίο βρισκόμαστε. Όταν $TF \leq 0.5$, βρίσκουμε τη νέα θέση μέσω της εξίσωσης (24), διαφορετικά χρησιμοποιούμε την εξίσωση (25):

$$x_i^{t+1} = x_i^t + C_1 \times rand \times acc_{i-norm}^{t+1} \times d \times (x_{rand} - x_i^t) \quad (24)$$

$$x_i^{t+1} = x_{best}^t + F \times C_2 \times rand \times acc_{i-norm}^{t+1} \times d \times (T \times x_{best} - x_i^t) \quad (25)$$

$$T = C_3 \times TF \quad (26)$$

$$F = \begin{cases} 1, & P \leq 0.5 \\ -1, & P > 0.5 \end{cases} \quad (27)$$

$$P = 2 \times rand - C_4 \quad (28)$$

Όπου C_1 , C_2 , C_3 , C_4 είναι σταθερές παράμετροι του αλγορίθμου, των οποίων τις τιμές έχουμε διαλέξει εκ των προτέρων, η μεταβλητή T μεταβάλλεται σε αναλογία με τον

τελεστή μεταφοράς, ενώ η μεταβλητή F σηματοδοτεί την αλλαγή κατεύθυνσης της κίνησης. Η θέση του αντικειμένου συμβολίζεται με x_i και x_{rand} είναι η θέση ενός τυχαίου αντικειμένου της συλλογής. Ο τυχαίος αριθμός $rand$ δίνει τιμές στο διάστημα $[0,1]$. Η μεταβλητή T χρησιμοποιείται ώστε να πάρουμε ένα ποσοστό της καλύτερης θέσης στο στάδιο της εκμετάλλευσης και αυξάνεται με τον χρόνο στο διάστημα $[C_3 \times e^{-1}, C_3]$. Αρχικά, παίρνουμε ένα μικρό ποσοστό, κάτι που οδηγεί σε μεγάλη διαφορά ανάμεσα στην τρέχουσα και στην καλύτερη θέση και, επομένως, το βήμα του τυχαίου περιπάτου θα είναι μεγάλο. Σταδιακά, αυξάνουμε το ποσοστό που παίρνουμε και, συνεπώς, μειώνεται η διαφορά, κάτι που ενισχύει περισσότερο την ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης.

Τέλος, εκτιμούμε κάθε αντικείμενο μέσω της αντικειμενικής συνάρτησης και σημειώνουμε τη θέση, πυκνότητα, όγκο και επιτάχυνση αυτού που κατέχει την καλύτερη λύση.

1.2.4 Παράμετροι

Οι Hashim et al. στη δημοσίευσή τους εξέτασαν διάφορες τιμές παραμέτρων, με σκοπό να βρεθούν τόσο ένα περιορισμένο σύνολο τιμών για την κάθε παράμετρο, όσο και ο καλύτερος δυνατός συνδυασμός τους. Ο αλγόριθμος βελτιστοποίησης του Αρχιμήδη εξαρτάται από τέσσερις παραμέτρους συνολικά (C_1, C_2, C_3, C_4) οι οποίες ορίζονται από τον χρήστη, ενώ οι υπόλοιπες παράμετροι είτε προκύπτουν από αυτές τις αρχικές, είτε υπολογίζονται βάσει της επανάληψης στην οποία βρισκόμαστε. Η ευαισθησία των παραμέτρων αναλύθηκε εφαρμόζοντας τον αλγόριθμο σε τρεις συναρτήσεις από τρεις διαφορετικές κατηγορίες του CEC' 17 test suite. Βάσει των αποτελεσμάτων των πειραμάτων οι Hashim et al. κατέληξαν σε κάποιες τιμές για την κάθε παράμετρο, οι οποίες δίνουν τα περισσότερα υποσχόμενα αποτελέσματα. Συνεπώς, έχουμε τα εξής σύνολα τιμών τα οποία λειτουργούν καλά για τα περισσότερα προβλήματα: $C_1 \in \{1,2\}$, $C_2 \in \{2,4,6\}$, $C_3 \in \{1,2\}$, $C_4 \in \{0.5,1\}$. Βεβαίως, αναλόγως του προβλήματος πάνω στο οποίο εργαζόμαστε, μπορούν να δοκιμαστούν και άλλες τιμές. Παρόλα αυτά, ο συνδυασμός $C_1 = 2$, $C_2 = 6$, $C_3 = 2$ και $C_4 = 0.5$ έδειξε να παράγει τις καλύτερες τιμές της συνάρτησης κόστους.

1.2.5 Πειράματα και αποτελέσματα

Οι Hashim et al. εκτός από τα πειράματα που εκτέλεσαν για την ανάλυση των παραμέτρων, εξέτασαν και διάφορες άλλες ιδιότητες που μας ενδιαφέρει να έχει ένας αλγόριθμος βελτιστοποίησης, όπως είναι η ισορροπία εξερεύνησης και εκμετάλλευσης του χώρου αναζήτησης του προβλήματος, η σύγκλιση και η ταχύτητα της σύγκλισης. Επίσης, τα αποτελέσματα συγκρίθηκαν με αυτά άλλων επιλεγμένων μετα-ευρετικών αλγορίθμων. Συνολικά, ο αλγόριθμος εφαρμόστηκε πάνω σε 29 δοκιμαστικές συναρτήσεις του CEC' 17 test suite, που αποτελούν προβλήματα ελαχιστοποίησης, και πάνω σε τέσσερα προβλήματα περιορισμών σχεδιασμού της μηχανικής.

Τα τελικά συμπεράσματα ήταν ότι ο αλγόριθμος βελτιστοποίησης του Αρχιμήδη, λόγω του μικρού αριθμού παραμέτρων, είναι απλός. Ξεκινάει εξερευνώντας τον χώρο και κυρίως αργότερα προσπαθεί να εκμεταλλευτεί και να βελτιώσει τις καλύτερες λύσεις που έχει βρει. Τα πειράματα κατέδειξαν ότι διατηρεί την ισορροπία εξερεύνησης και εκμετάλλευσης και διακρίνεται από την προσαρμοστικότητα των υποψήφιων λύσεων, ώστε να αποφεύγονται οι παγιδεύσεις σε τοπικά μέγιστα. Τέλος, η σύγκρισή του με

αλγορίθμους, όπως ο γενετικός, ο PSO καθώς και άλλους πιο πρόσφατους, έδειξαν την υπεροχή της βελτιστοποίησης του Αρχιμήδη τόσο όσον αφορά την απόδοση, όσο και την ταχύτητα σύγκλισης στη βέλτιστη λύση. Βεβαίως, είναι αναγκαία η εφαρμογή του σε πολύ περισσότερα προβλήματα του πραγματικού κόσμου, προκειμένου να επιβεβαιωθεί ή όχι αυτή η υπεροχή.

1.3 Χρονοπρογραμματισμός πληρωμάτων αεροπορικών εταιρειών

Ο χρονοπρογραμματισμός πληρωμάτων αεροπορικών εταιρειών (airline crew rostering) είναι ένα πρόβλημα ικανοποίησης περιορισμών, για το οποίο προσπαθούμε να βρούμε τη βέλτιστη δυνατή λύση. Ανήκει στη γενικότερη κατηγορία χρονοπρογραμματισμού πληρωμάτων (crew rostering) και είναι επίσης γνωστό ως πρόβλημα ανάθεσης πληρωμάτων (crew assignment). Όπως διαφαίνεται και από το όνομα του, είναι ένα πρόβλημα που αφορά τη διαχείριση προσωπικού σε συστήματα μεταφορών μεγάλης κλίμακας. Ο στόχος μας είναι να αναθέσουμε μια σειρά καθηκόντων και εργασιών, που πρέπει να εκτελεστούν στη διάρκεια μιας χρονικής περιόδου, σε διαθέσιμα μέλη του προσωπικού, που μπορούν να επανδρώσουν ένα πλήρωμα, ενώ παράλληλα σεβόμαστε τους περιορισμούς που μας τίθενται είτε από την ίδια την εταιρεία, είτε από άλλους παράγοντες. Εν τέλει, ζητούμε να καλύπτονται όλα τα καθήκοντα με επαρκή αριθμό ατόμων, ενώ παράλληλα προσπαθούμε να ελαχιστοποιήσουμε όποιο κόστος έχει σχέση με έξοδα προσωπικού. Ο χρονοπρογραμματισμός πληρωμάτων θεωρείται ως ένα πρόβλημα NP-δύσκολο και για αυτόν τον λόγο έχει τραβήξει το ενδιαφέρον πλήθους ερευνών.

1.3.1 Υπόβαθρο

Μια αεροπορική εταιρεία έχει έσοδα από τα εισιτήρια που κόβει και έξοδα που αφορούν τη συντήρηση των αεροσκαφών της, τα καύσιμα, το ενοίκιο που πληρώνει στα αεροδρόμια και τους μισθούς του προσωπικού. Τα έξοδα προσωπικού είναι το δεύτερο μεγαλύτερο κόστος που επωμίζεται, με πρώτο να είναι το κόστος καυσίμων. Συνεπώς, ακόμα και μία μικρή ποσοστιαία μείωση στο κόστος του πληρώματος μπορεί να σημαίνει μία τεράστια εξοικονόμηση για μια μεγάλη αεροπορική εταιρεία. Είναι εύκολα κατανοητό, ότι αυτή η δυνατότητα μείωσης του κόστους προσέλκυσε το ενδιαφέρον των εταιρειών ήδη από τα πρώτα στάδια ανάπτυξής τους, με αρκετές τεχνικές να εφαρμόζονται για την εύρεση ενός αποδοτικού προγράμματος.

Ο χρονοπρογραμματισμός πληρωμάτων είναι μέρος ενός ακόμη πιο σύνθετου προβλήματος, του σχεδιασμού πληρωμάτων (crew planning). Ο σχεδιασμός πληρωμάτων, λοιπόν, χωρίζεται σε δύο υποπροβλήματα. Το πρώτο ονομάζεται συνδυασμός πληρώματος (crew pairing), όπου δημιουργούνται ανώνυμοι συνδυασμοί πτήσεων (pairings), με το κάθε μέλος ενός συνδυασμού να είναι μια πτήση χωρίς ενδιάμεσες στάσεις (flight legs). Ο κύριος περιορισμός, που πρέπει να ικανοποιηθεί σε αυτό το στάδιο, είναι το πλήρωμα να επαρκεί για την εκτέλεση κάθε πτήσης του συνδυασμού. Αυτό δεν αναφέρεται μόνο στον αριθμό των ατόμων που χρειάζονται, αλλά και στον βαθμό κάθε μέλους του πληρώματος που είναι απαραίτητος για τη διεκπεραίωση όλων των καθηκόντων που προκύπτουν κατά τη διάρκεια μιας πτήσης. Ο αριθμός των ατόμων που πρέπει να διατεθούν για έναν συνδυασμό, καθώς και οι βαθμοί τους, ορίζεται ως συμπλήρωμα πληρώματος (crew complement). Το δεύτερο υποπρόβλημα, το βασικό αντικείμενο αυτής της πτυχιακής εργασίας, είναι ο χρονοπρογραμματισμός πληρωμάτων. Σε αυτό το στάδιο, χρησιμοποιούμε τους συνδυασμούς που δημιουργήσαμε στον συνδυασμό πληρώματος, μαζί με άλλα

καθήκοντα, όπως είναι εργασίες εδάφους, εφεδρείες ή ρεπό, για τη δημιουργία συνθέσεων (rosters) τις οποίες αναθέτουμε σε συγκεκριμένα μέλη πληρώματος. Κατά την επίλυση των δύο υποπροβλημάτων, πρέπει να ικανοποιούνται κανόνες, οι οποίοι προκύπτουν είτε βάσει της νομοθεσίας, είτε με υποχρεώσεις που πρέπει να εκπληρώσει η εταιρεία βάσει συμβολαίων που έχει υπογράψει.

Η διαφορά ανάμεσα στα δύο υποπροβλήματα και ο κύριος λόγος για τον οποίο οι περισσότερες έρευνες στρέφονται προς την πιο αποδοτική επίλυση του συνδυασμού πληρώματος είναι ότι στον συνδυασμό πληρώματος υπάρχουν περισσότερες δυνατότητες για μείωση κόστους σε σχέση με τον χρονοπρογραμματισμό. Για αυτόν τον λόγο, ενώ στον χρονοπρογραμματισμό πληρωμάτων ο κύριος στόχος συνεχίζει να είναι η μείωση κόστους, παράλληλα λαμβάνεται υπόψη και η ποιότητα ζωής των μελών του πληρώματος. Δηλαδή, ο στόχος είναι ένα πρόγραμμα που αφενός έχει όσο το δυνατόν πιο μειωμένο κόστος και αφετέρου ικανοποιεί τις απαιτήσεις του προσωπικού. Κάποιες από τις προσεγγίσεις του προβλήματος, μάλιστα, λαμβάνουν υπόψη όχι μόνο τις υποχρεώσεις που έχει η εταιρεία απέναντι στο προσωπικό της, αλλά και τις προτιμήσεις των μελών των πληρωμάτων.

1.3.2 Περιορισμοί

Κάθε αεροπορική εταιρεία προσπαθεί να επιτύχει τους δικούς της στόχους και να εκπληρώσει τις δικές της υποχρεώσεις. Επομένως, η οριοθέτηση του προβλήματος του συνδυασμού πληρώματος διαφέρει για κάθε εταιρεία. Παρόλα αυτά, τα κύρια χαρακτηριστικά του παραμένουν τα ίδια. Οι κανόνες που πρέπει να ικανοποιηθούν συγκρίνονται εύκολα ανάμεσα σε δύο διαφορετικές υλοποιήσεις και οι αντικειμενικές συναρτήσεις βασίζονται σε πραγματικά κόστη και σε μισθούς. Αντιθέτως, το πρόβλημα του χρονοπρογραμματισμού μπορεί να προσεγγιστεί με διαφορετικούς τρόπους, όπως η δημιουργία συνθέσεων όπου για κάθε σύνθεση τα μέλη του πληρώματος δηλώνουν διαθεσιμότητα (bidlines approach) ή η δημιουργία προσωποποιημένων συνθέσεων (personalized rosters), όπου κάθε σύνθεση δημιουργείται για συγκεκριμένο άτομο.

Το κοινό σημείο ανάμεσα σε όλες τις προσεγγίσεις είναι ο σεβασμός στους κανόνες και στους κανονισμούς. Αυτοί είναι που καθορίζουν τη νομιμότητα μιας σύνθεσης. Μπορούν να προέρχονται από την κυβέρνηση, από την ίδια την εταιρεία ή από τις όποιες συμφωνίες έχουν υπογραφεί μεταξύ της εταιρείας και τα σωματεία εργαζομένων. Οι κανόνες και οι κανονισμοί σε αυτό το πλαίσιο είναι οι περιορισμοί του προβλήματος. Χωρίζονται σε τρεις κατηγορίες, στους οριζόντιους, στους κάθετους και στους τεχνητούς κανόνες.

Οι οριζόντιοι κανόνες εφαρμόζονται σε μία μόνο σύνθεση και δεν λαμβάνουν υπόψη τους τις υπόλοιπες συνθέσεις της λύσης. Κάποιοι από τους συνηθισμένους οριζόντιους κανόνες είναι οι εξής:

1. Η συμβατότητα ατόμου-καθήκοντος-χρόνου είναι ο πιο συνηθισμένος οριζόντιος κανόνας. Ασυμβατότητα μπορεί να προκύψει λόγω έλλειψης των απαραίτητων προσόντων ή λόγω πρότερων αναθέσεων άλλων καθηκόντων στο ίδιο άτομο, στα οποία καθήκοντα συμπεριλαμβάνονται και οι προγραμματισμένες άδειες. Η εγκυρότητα του κανόνα συμβατότητας μπορεί να επαληθευτεί πολύ εύκολα.
2. Ο χρόνος ανάπαυσης μεταξύ εργασιών είναι ένας ακόμα συνηθισμένος οριζόντιος κανόνας που επαληθεύεται εύκολα, καθώς ορίζει το χρονικό διάστημα που πρέπει να παρέλθει ανάμεσα σε δύο διαδοχικές εργασίες που εκτελεί το ίδιο άτομο. Παρόλα αυτά, ο υπολογισμός του χρονικού διαστήματος μπορεί να είναι πιο

σύνθετος, αφού είναι πιθανό εξαρτάται από περισσότερους παράγοντες πέρα από τον χρόνο λήξης και έναρξης των εργασιών.

3. Τα μοτίβα των ρεπτό είναι ένας κανόνας που ορίζει τόσο την περίοδο εργασίας όσο και την περίοδο ανάπαυσης. Η περίοδος εργασίας είναι το χρονικό διάστημα όπου το μέλος του πληρώματος εκτελεί εργασίες χωρίς διάλειμμα (με εξαίρεση τον χρόνο ανάπαυσης που αναφέρεται στον προηγούμενο κανόνα). Αυτή η περίοδος έχει ένα άνω όριο, στο τέλος του οποίου ο εργαζόμενος έχει μια περίοδο ανάπαυσης, όπου δεν του ανατίθεται καμία εργασία. Οι δύο αυτές περιόδους υπολογίζονται σε ημέρες και ο συνηθισμένος χρόνος μιας περιόδου ανάπαυσης είναι δύο ημέρες. Επίσης, είναι πιθανό να απαιτείται η ικανοποίηση ενός συγκεκριμένου μοτίβου σε ένα μεγαλύτερο χρονικό διάστημα.
4. Οι συσσωρευμένες τιμές είναι ένας κανόνας που έχει σχέση με τις υποχρεώσεις της εταιρείας προς το προσωπικό όσον αφορά τον χρόνο απασχόλησης. Κάποια παραδείγματα συσσωρευμένων τιμών είναι ο μέγιστος συνεχόμενος αριθμός ωρών εργασίας, ο μέγιστος αριθμός «βαριών» καθηκόντων που μπορεί να αναλάβει ένα άτομο σε μια περίοδο εργασίας και ο ελάχιστος αριθμός αδειών που πρέπει να πάρει ένα μέλος του πληρώματος.

Οι κάθετοι κανόνες αφορούν περισσότερες από μία συνθέσεις. Μπορεί να αφορούν ένα υποσύνολο των συνθέσεων ή ακόμη και ολόκληρο το πρόγραμμα. Οι βασικότεροι κάθετοι κανόνες είναι:

1. Το συμπλήρωμα πληρώματος είναι ο πιο σημαντικός κάθετος κανόνας. Συμπλήρωμα πληρώματος ονομάζεται ο αριθμός των ατόμων που είναι απαραίτητα για ένα πλήρωμα για τη διεκπεραίωση μιας δραστηριότητας, καθώς και ο βαθμός που πρέπει να κατέχει κάθε μέλος του πληρώματος. Ενώ το συμπλήρωμα πληρώματος θα μπορούσε να κατηγοριοποιηθεί ως περιορισμός σχετιζόμενος με προσόντα (qualification-type constraint), αναφέρεται ξεχωριστά λόγω της σημασίας του.
2. Οι περιορισμοί σχετιζόμενοι με προσόντα εφαρμόζονται σε δραστηριότητες που απαιτούν άτομα με συγκεκριμένη τεχνογνωσία. Τα άτομα αυτά, ανήκουν κυρίως στο πλήρωμα καμπίνας του αεροπλάνου, όμως είναι δυνατό ο περιορισμός να αναφέρεται στο πλήρωμα του πιλοτηρίου. Παραδείγματα τέτοιου είδους περιορισμών μπορεί να αφορούν τον μέγιστο αριθμό μη έμπειρων μελών του πληρώματος, μέλη του πληρώματος που πρέπει να έχουν κοινές πτήσεις, ασυμβατότητα μεταξύ δύο μελών ή η απαραίτητη γνώση μιας γλώσσας.
3. Οι καθολικοί περιορισμοί είναι η τρίτη κατηγορία κάθετων κανόνων. Οι προηγούμενοι κανόνες μπορεί να εφαρμόζονται σε όλες τις δραστηριότητες. Για παράδειγμα, μπορεί όλες οι εργασίες να έχουν την απαίτηση για γνώση κάποιας γλώσσας. Παρόλα αυτά, μόνο ένα υποσύνολο των συνθέσεων θα έχει την απαίτηση για μια συγκεκριμένη γλώσσα και θα είναι οι συνθέσεις αυτές που περιέχουν τους συνδυασμούς με αυτήν την απαίτηση. Αντιθέτως, ένας καθολικός περιορισμός επιβάλλει τις απαιτήσεις του σε όλες τις συνθέσεις της λύσης. Κάποιοι καθολικοί περιορισμοί είναι το άνω φράγμα του κόστους της λύσης, η συνολική ικανοποίηση των αιτήσεων για ανάληψη συνθέσεων (όταν ακολουθούμε την αντίστοιχη προσέγγιση του προβλήματος) και οριζόντιοι περιορισμοί που ξεπερνούν την διάρκεια ενός προγράμματος.
4. Άλλοι κάθετοι περιορισμοί μπορεί να αφορούν την ανάθεση μιας εργασίας σε κάποιο μέλος του πληρώματος που έχει παραπάνω προσόντα από ότι χρειάζεται ή περιορισμοί που σχετίζονται με εκπαιδευτικά προγράμματα. Για παράδειγμα, είναι δυνατό να αναθέσουμε τον ρόλο του συγκυβερνήτη σε κάποιον με τον βαθμό του πιλότου, όμως να υπάρχει ένα όριο στον αριθμό των φορών που μπορεί να γίνει αυτή η ανάθεση.

Οι τεχνητοί κανόνες είναι τυπικά περιορισμοί που ορίζονται από την αεροπορική εταιρεία και δεν επιβάλλονται από κάποια νομοθεσία ή κάποια συμφωνία. Είναι βασισμένοι στην πείρα του εκάστοτε υπεύθυνου σχεδιασμού του προγράμματος. Ο στόχος είναι η βελτίωση της ποιότητας του προγράμματος και η υποστήριξη των μεθόδων που εφαρμόζονται για τη λύση. Ένα παράδειγμα τεχνητού κανόνα είναι ο σχεδιασμός του προγράμματος εφαρμόζοντας κάποια χρονικά περιθώρια, ώστε σε περίπτωση απρόοπτου γεγονότος να μην καταστεί όλο το πρόγραμμα αδύνατο.

1.3.3 Στόχοι

Ο βασικός στόχος κατά την επίλυση του προβλήματος χρονοπρογραμματισμού είναι να σχεδιάσουμε ένα πρόγραμμα στο οποίο όλοι οι δοσμένοι συνδυασμοί έχουν ανατεθεί σε επαρκή αριθμό μελών του πληρώματος. Παρόλα αυτά, μπορεί να υπάρχουν και άλλοι στόχοι οι οποίοι χωρίζονται τυπικά σε τέσσερις κατηγορίες. Συνήθως, απαιτείται η επίτευξη ενός συνδυασμού στόχων από αυτές τις κατηγορίες, κάτι που γίνεται εφικτό είτε μέσω της εισαγωγής τους ως καθολικούς περιορισμούς είτε μέσω της συμπερίληψής τους στην αντικειμενική συνάρτηση. Τα είδη των στόχων είναι τα εξής:

1. Στόχοι που σχετίζονται με πραγματικά κόστη. Αυτά τα κόστη μπορεί να προέρχονται από δραστηριότητες που δεν έχουν ανατεθεί. Ο όρος που χρησιμοποιείται για τις δραστηριότητες αυτές είναι ανοιχτός χρόνος (open time). Η παρουσία ανοιχτού χρόνου στην λύση δεν είναι απαγορευτική για την εγκυρότητάς της, καθώς υπάρχουν τρόποι για την εξάλειψή του, όπως η εκτέλεση υπερωριών, η πρόσληψη προσωπικού ή ακόμη και η ακύρωση των μη ανατιθέμενων δραστηριοτήτων. Είναι εμφανές, ωστόσο, ότι η ελαχιστοποίηση ή ακόμη και η αποφυγή του ανοιχτού χρόνου είναι ζωτικής σημασίας για την ποιότητα της λύσης. Άλλα παραδείγματα πραγματικού κόστους, είναι οι πληρωμές υπερωριών και οι αμοιβές ωρομίσθιων πιλότων.
2. Στόχοι που σχετίζονται με την ευρωστία της λύσης. Ένας τέτοιος στόχος μπορεί να τεθεί όταν θέλουμε να εισάγουμε στον σχεδιασμό μας κάποια περιθώρια. Αντί να επιβάλλουμε αυστηρά περιθώρια μπορούμε να επιτρέψουμε λύσεις που τα παραβιάζουν, αλλά να τις κατατάσσουμε ως χαμηλότερης ποιότητας. Ένα άλλο παράδειγμα είναι ο ίσος καταμερισμός των «βαριών» καθηκόντων, καθώς έχει παρατηρηθεί το φαινόμενο δήλωσης ασθενείας των ατόμων που έχουν αναλάβει «βαριές» συνθέσεις.
3. Στόχοι που σχετίζονται με συγκεκριμένες ιδιότητες των συνθέσεων. Ο πιο συνήθης στόχος αυτής της κατηγορίας είναι ο ίσος καταμερισμός του φόρτου εργασίας ανάμεσα σε όλα τα μέλη του πληρώματος. Αν και η επίτευξη του στόχου δεν θεωρείται απαραίτητη στον χρόνο που διαρκεί ένα πρόγραμμα, είναι αναγκαία σε ετήσια βάση. Τα μοτίβα των ωρών εργασίας είναι άλλος στόχος αυτού του είδους, καθώς μπορεί να είναι επιθυμητά από το προσωπικό.
4. Στόχοι που σχετίζονται με προσωπικές προτιμήσεις. Κάποιες αεροπορικές εταιρείες επιτρέπουν στα μέλη των πληρωμάτων τους να εκφράσουν τις προτιμήσεις τους σε σχέση με το προσωπικό τους πρόγραμμα. Οι προτιμήσεις αυτές λαμβάνονται υπόψη κατά τον σχεδιασμό του προγράμματος αν και δεν είναι απαραίτητο να ικανοποιηθούν. Κατά κανόνα, εξυπηρετούνται πρώτα οι εργαζόμενοι με μεγαλύτερη παλαιότητα.

1.3.4 Μαθηματικό μοντέλο

Για την αναπαράσταση του προβλήματος χρειάζεται να ορίσουμε τα σύνολα των οντοτήτων που το αποτελούν καθώς και κάποιες βοηθητικές μεταβλητές. Έστω ότι K είναι το σύνολο όλων των διαθέσιμων εργαζομένων και S^k είναι το σύνολο όλων των συνθέσεων που μπορεί να αναλάβει ένα μέλος $k \in K$. Επίσης, P συμβολίζει το σύνολο όλων των δοσμένων συνδυασμών βάσει των οποίων κατασκευάζουμε την λύση. Ο αριθμός των ατόμων που χρειαζόμαστε για την κάλυψη ενός συνδυασμού $p \in P$ είναι n_p . Η μεταβλητή γ_s^p ισούται με 1, όταν ο συνδυασμός $p \in P$ ανήκει στην σύνθεση s , και 0 διαφορετικά. Η μεταβλητή x_s^k είναι μια μεταβλητή απόφασης, η οποία ορίζεται ως 1 εφόσον το μέλος του πληρώματος $k \in K$ αναλαμβάνει την σύνθεση $s \in S^k$, και 0 σε διαφορετική περίπτωση. Το κόστος της σύνθεσης συμβολίζεται με c_s^k και αντιπροσωπεύει το πραγματικό λειτουργικό κόστος ανάθεσης της σύνθεσης στον συγκεκριμένο εργαζόμενο, καθώς η ανάθεση της ίδιας σύνθεσης σε διαφορετικό μέλος μπορεί να αποδώσει και διαφορετικό κόστος. Συνοψίζοντας όλα τα παραπάνω, το μοντέλο βελτιστοποίησης γράφεται ως εξής:

$$\min \sum_{k \in K} \sum_{s \in S^k} c_s^k x_s^k \quad (29)$$

Υπό τον περιορισμό ότι:

$$\sum_{k \in K} \sum_{s \in S^k} \gamma_s^p x_s^k \geq n_p \quad \forall p \in P \quad (30)$$

$$\sum_{s \in S^k} x_s^k = 1 \quad \forall k \in K \quad (31)$$

$$x_s^k \in \{0,1\} \quad \forall s \in S^k, \forall k \in K \quad (32)$$

Η εξίσωση (29) είναι η αντικειμενική συνάρτηση μέσω της οποίας ελαχιστοποιούμε τα λειτουργικά κόστη όλων των συνθέσεων της λύσης. Η εξίσωση (30) εξασφαλίζει ότι όλοι οι συνδυασμοί καλύπτονται τουλάχιστον από τον απαραίτητο αριθμό ατόμων. Η εξίσωση (31) εξασφαλίζει ότι σε κάθε μέλος του προσωπικού ανατίθεται ακριβώς μια έγκυρη σύνθεση. Τέλος, η εξίσωση (32) ορίζει τις μεταβλητές απόφασης για κάθε εργαζόμενο.

1.3.5 Προσεγγίσεις της λύσης

Ο χρονοπρογραμματισμός πληρωμάτων απασχόλησε τις αεροπορικές εταιρείες από νωρίς. Ως αποτέλεσμα, έχει γίνει εκτενής έρευνα γύρω από την επίλυση του προβλήματος, με πλήθος μεθόδων να έχουν εφαρμοστεί. Μία προσέγγιση είναι η χρήση κατασκευαστικών ευρετικών αλγορίθμων (constructive heuristics), οι οποίοι είτε θα βρουν μια λύση η οποία θα αρκεί για την επίτευξη των στόχων είτε αυτή η λύση θα χρησιμοποιηθεί ως βάση για πιο σύνθετους υπολογισμούς. Παρόλα αυτά, σπάνια θα

βρεθεί η βέλτιστη λύση. Ο τρόπος που κατασκευάζουμε τη λύση μπορεί να είναι μέρα με τη μέρα ή ανά πιλότο.

Ο μαθηματικός προγραμματισμός μπορεί επίσης να εφαρμοστεί στο πρόβλημα. Εδώ ο χρονοπρογραμματισμός εκφράζεται ως πρόβλημα διαχωρισμού συνόλων (set partitioning problem) ή ως πρόβλημα κάλυψης (covering problem). Κάποιες μέθοδοι μαθηματικού προγραμματισμού είναι η διακλάδωση και δέσμευση (branch-and-bound), η διακλάδωση και αποκοπή (branch-and-cut), η διακλάδωση και αποτίμηση (branch-and-price) και η παραγωγή στηλών (column generation). Η διακλάδωση και δέσμευση είναι ένας αλγόριθμος με την ίδια φιλοσοφία με τους αλγορίθμους διαίρει και βασίλευε (divide-and-conquer) και λειτουργεί σε χαλαρώσεις LP . Η παραγωγή στηλών χρησιμοποιείται σε γραμμικά προγράμματα μεγάλης κλίμακας. Η διακλάδωση και αποκοπή επιλύει χαλαρώσεις LP με λιγότερους περιορισμούς, ενώ η διακλάδωση και αποτίμηση εφαρμόζει παρόμοια τακτική σε περιπτώσεις που έχουμε λιγότερες μεταβλητές.

Έχουν επίσης προταθεί μοντέλα βασισμένα σε δίκτυο, όπου το πρόβλημα διατυπώνεται ως ένα δίκτυο ροών. Τα δίκτυα ταξιδιού ως κόμβος (trip-as-node networks) επικεντρώνονται κυρίως στις σχέσεις μεταξύ των πτήσεων που αποτελούν έναν συνδυασμό. Τα ταξίδια δίνονται ως κόμβοι και οι ακμές αντιπροσωπεύουν πιθανές συνδέσεις. Ο όρος ταξίδι, γενικά, μπορεί να ορίζεται ως μια πτήση, ένας συνδυασμός ή ένα καθήκον που πρέπει να ολοκληρωθεί κατά τη διάρκεια μιας πτήσης. Τα δίκτυα ταξιδιού ως ακμή (trip-as-arc networks), γνωστά και ως δίκτυα χώρου και χρόνου (time-space networks) ή δίκτυα καθήκοντος/πτήσης, προσπαθούν να εκμεταλλευτούν τα χαρακτηριστικά του χώρου και του χρόνου του προβλήματος. Σε αυτά τα δίκτυα τα ταξίδια ή οι δραστηριότητες είναι οι ακμές του δικτύου, ενώ οι κόμβοι αντιπροσωπεύουν τοποθεσίες σε συγκεκριμένες χρονικές στιγμές.

Οι μετα-ευρετικοί αλγόριθμοι αποτελούν μία άλλη προσέγγιση του προβλήματος. Αλγόριθμοι όπως ο γενετικός, η προσομοιωμένη απόκτηση και η αναζήτηση Tabu (Tabu search) έχουν εφαρμοστεί με επιτυχία στο παρελθόν. Η γενική στρατηγική αυτών των αλγορίθμων είναι η αναπαράσταση μιας λύσης, η οποία δεν είναι απαραίτητα έγκυρη, από κάθε μέλος του πληθυσμού τους. Η σύγκριση της ποιότητας των λύσεων γίνεται μέσω των τιμών καταλληλότητας.

Άλλες τεχνικές που έχουν προταθεί είναι η τοπική βελτιστοποίηση, όπου επικεντρωνόμαστε στην βελτίωση ενός τμήματος της λύσης, ο λογικός προγραμματισμός, όπου δίνονται σύνολα επιτρεπόμενων τιμών στις μεταβλητές, και ο δυναμικός προγραμματισμός, όπου χρησιμοποιείται αναδρομή για την εύρεση της βέλτιστης λύσης.

Οι μετα-ευρετικοί αλγόριθμοι βελτιστοποίησης μέσω σμήνους κοτών, και οι παραλλαγές της, και βελτιστοποίησης του Αρχιμήδη δεν έχουν δοκιμαστεί στο πρόβλημα χρονοπρογραμματισμού πληρωμάτων αεροπορικών εταιρειών, καθώς δεν υπάρχει κάποια σχετική έρευνα στην τρέχουσα βιβλιογραφία.

2. ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΙ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΛΗΡΩΜΑΤΩΝ

Σε αυτό το κεφάλαιο αναλύουμε την εφαρμογή του αλγορίθμου βελτιστοποίησης μέσω σμήνους κοτών και βελτιστοποίησης του Αρχιμήδη για την επίλυση του προβλήματος χρονοπρογραμματισμού πληρωμάτων μιας αεροπορικής εταιρείας. Ορίζουμε τα πλαίσια του προβλήματος, το μοντέλο που ακολουθήθηκε, τις παραδοχές που υιοθετήθηκαν και τους περιορισμούς που έπρεπε να ικανοποιηθούν. Τέλος, παρουσιάζουμε τις δομές και τις συναρτήσεις που χρησιμοποιήθηκαν, καθώς και τα αλγοριθμικά βήματα.

2.1 Ορισμός προβλήματος

Στο προηγούμενο κεφάλαιο δείξαμε ότι ο χρονοπρογραμματισμός πληρωμάτων είναι ένα αρκετά σύνθετο πρόβλημα, το οποίο μπορεί να έχει πολλαπλούς στόχους πέρα από την ανάθεση όλων των δραστηριοτήτων. Επίσης, απαιτεί συνήθως την ικανοποίηση πολλών περιορισμών που υπαγορεύονται από κανονισμούς ή συμφωνίες που έχει συνάψει η εταιρεία. Για να μπορέσουμε να αποδείξουμε την ικανότητα των μετα-ευρετικών αλγορίθμων να επιλύσουν το πρόβλημα, προτείνουμε μια απλοποιημένη εκδοχή του, όπου εισάγουμε δύο κανόνες ως περιορισμούς και κάνουμε κάποιες παραδοχές. Αυτές οι παραδοχές είναι:

1. Οι δραστηριότητες που μας δίνονται είναι μόνο συνδυασμοί πτήσεων.
2. Για την κάλυψη ενός συνδυασμού αρκεί η ανάθεσή του σε ένα άτομο.
3. Τα μέλη του πληρώματος θεωρούνται ότι είναι όλα πιλότοι.
4. Όλοι οι πιλότοι έχουν τα ίδια προσόντα.
5. Ένας συνδυασμός δεν έχει κάποια ειδική απαίτηση σχετικά με τα προσόντα του ατόμου που θα τον αναλάβει, δηλαδή μπορεί να ανατεθεί σε οποιοδήποτε μέλος του πληρώματος.

Όσον αφορά τους κανόνες που θα ακολουθήσουμε, αυτοί είναι:

1. Ανάμεσα σε δύο διαδοχικούς συνδυασμούς που ανατίθενται σε έναν πιλότο πρέπει να μεσολαβούν τουλάχιστον έντεκα ώρες ανάπαυσης (κανόνας ανάπαυσης).
2. Σε οποιοδήποτε διάστημα επτά ημερών του προγράμματος διαλέξουμε, κάθε πιλότος πρέπει να έχει τουλάχιστον δύο ημέρες που είναι ρεπό. Μια ημέρα είναι ρεπό για έναν πιλότο αν από τα μεσάνυχτα της μέχρι τα μεσάνυχτα της επομένης δεν υπάρχει στο πρόγραμμα του κάποιος συνδυασμός (κανόνας ρεπό).

Στη συνέχεια, πρέπει να ορίσουμε την αντικειμενική συνάρτηση ή συνάρτηση καταλληλότητας βάσει της οποίας αξιολογούμε τις λύσεις. Όπως έχουμε πει, ο κύριος στόχος είναι να βρεθεί λύση η οποία είναι πλήρης ή, αλλιώς, έγκυρη, δηλαδή μια λύση όπου δεν υπάρχει μη ανατιθέμενος συνδυασμός. Μη έγκυρες λύσεις ονομάζονται μερικές ή άκυρες. Παράλληλα με αυτόν τον στόχο, επιθυμούμε τον ίσο καταμερισμό του φόρτου εργασίας. Ως φόρτο εργασίας θεωρούμε τον συνολικό χρόνο που ένας πιλότος είναι σε πτήση. Αυτοί οι δύο στόχοι συνδυάζονται και εισάγονται στην αντικειμενική συνάρτηση.

Η ποιότητα των λύσεων εξαρτάται σε μεγάλο βαθμό από τον αριθμό έγκυρων λύσεων που υπάρχουν. Για παράδειγμα, αν ο αριθμός των πιλότων οριακά αρκεί για να καλύψει τους συνδυασμούς, τότε είναι πιθανό να μην μπορέσουμε να βρούμε πλήρη λύση. Σε αυτήν την περίπτωση, ο αλγόριθμος θα επιστρέψει την καλύτερη μερική λύση, δηλαδή αυτή που καλύπτει τους περισσότερους συνδυασμούς.

Ο αλγόριθμος δέχεται ως είσοδο το αρχείο με τους συνδυασμούς προς ανάθεση, τον αριθμό των διαθέσιμων πιλότων και την αρχή και την λήξη του προγράμματος της αεροπορικής εταιρείας (βλ. ενότητα 2.2.1). Η έξοδος είναι το πρόγραμμα με την σύνθεση κάθε πιλότου.

2.2 Περιγραφή εφαρμογής

Ο σκελετός της εφαρμογής μας αποτελείται από δύο φάσεις, σε συμφωνία με τις δομές των αλγορίθμων βελτιστοποίησης. Αρχικά, προετοιμάζουμε τις δομές που θα μας βοηθήσουν στην κατασκευή της λύσης και στη συνέχεια προχωρούμε στην εκτέλεση του κυρίου σώματος του αλγορίθμου βελτιστοποίησης που διαλέγουμε για το τρέχον πείραμα. Όσον αφορά τη βελτιστοποίηση μέσω σμήνους κοτών, πειραματιστήκαμε με τη βασική εκδοχή στα πρώτα στάδια ανάπτυξης της εφαρμογής, αλλά στην τελική υλοποίηση χρησιμοποιούμε την πολύ-βηματική CSO, καθώς ξεπερνάει αποδοτικά την παγίδευση σε τοπικά μέγιστα. Αναλύουμε σε βάθος την κάθε πτυχή της εφαρμογής στις αντίστοιχες ενότητες.

Αλγόριθμος 1: Γενική δομή εφαρμογής

Είσοδος: *pairs, numberOfPilots, maxGenerations, numberOfAgents, parameters*

Έξοδος: *rosters*

```
1  airlineSetup(pairs, numberOfPilots);
2  graphSetup(pairs, numberOfAgents);
3  if chosen algorithm is multi-step CSO then
4  |  swarmSetup(maxGenerations , numberOfAgents, parameters);
5  |  multiCSO();
6  else if chosen algorithm is AOA then
7  |  collectionSetup(maxGenerations , numberOfAgents, parameters);
8  |  AOA();
9  end
10 return rosters;
```

Το πρόγραμμα παίρνει την είσοδο και μέσω των συναρτήσεων *airlineSetup* και *graphSetup* αρχικοποιεί την δομή που αναπαριστά την αεροπορική εταιρεία και τον γράφο αντίστοιχα (βλ. ενότητες 2.2.2 και 2.2.3). Έπειτα, ανάλογα με τον αλγόριθμο βελτιστοποίησης που επιλέξαμε αρχικοποιούμε είτε το σμήνος κοτών είτε την συλλογή αντικειμένων (*swarmSetup* και *collectionSetup* αντίστοιχα), ορίζοντας τις κατάλληλες παραμέτρους για τον κάθε αλγόριθμο. Το κύριο σώμα της πολύ-βηματικής CSO έχει παρόμοια δομή με την AOA και μπορούν να περιγραφούν αμφότερες από τον «Αλγόριθμο 2».

Αλγόριθμος 2: Γενική δομή αλγορίθμων βελτιστοποίησης

Είσοδος: *maxGenerations, pairs, agents*

Έξοδος: *rosters*

```
1 for i = 1, 2, ..., maxGenerations do
2   foreach agent in agents do
3     constructSolution(pairs);
4     evaluate new solution and assign it to agent if it has better fitness
5   end
6   updatePositions(agents);
7 end
8 return rosters;
```

Ο αλγόριθμος σε κάθε γενιά κατασκευάζει λύσεις για κάθε πράκτορα βάσει της τρέχουσας θέσης του, αξιολογεί την καινούρια λύση και αντικαθιστά την ήδη υπάρχουσα, εφόσον η τιμή καταλληλότητας είναι καλύτερη και στο τέλος της επανάληψης ανανεώνει τις θέσεις των πρακτόρων. Οι πράκτορες ανάλογα με τον αλγόριθμο που χρησιμοποιούμε είναι είτε κότες είτε αντικείμενα. Στο τέλος του αλγορίθμου επιστρέφουμε την καλύτερη λύση.

2.2.1 Είσοδος εφαρμογής

Η είσοδος του προγράμματος παίρνει μια σειρά από ορίσματα, από τα οποία μόνο δύο πρέπει να δοθούν υποχρεωτικά από τον χρήστη:

- `multiCSO/AOA`: είναι εντολές που ορίζουν ποιον αλγόριθμο βελτιστοποίησης θα χρησιμοποιήσουμε. Πρέπει να είναι το πρώτο όρισμα που δίνεται κατά την κλήση του προγράμματος. Επιπλέον, είναι το πρώτο υποχρεωτικό όρισμα και πρέπει να διαλέξουμε ακριβώς μία από τις διαθέσιμες επιλογές.
- `-f, --filename`: είναι το αρχείο των συνδυασμών πτήσεων, το οποίο πρέπει να δοθεί απαραίτητως από τον χρήστη ώστε να μπορούμε να προχωρήσουμε. Μπορούμε να δώσουμε το σχετικό ή το απόλυτο μονοπάτι προς το αρχείο.
- `--results`: είναι προαιρετικό όρισμα, το οποίο δίνει το όνομα του αρχείου excel εξόδου. Το αρχείο αποθηκεύεται στον υποφάκελο `output` του φακέλου όπου βρίσκεται η εφαρμογή. Εξ ορισμού, το όνομα του αρχείου είναι `Output.xlsx`.
- `-p, --pilots`: είναι ο αριθμός των διαθέσιμων πιλότων και μπορεί να παραλειφθεί. Σε αυτήν την περίπτωση θεωρούμε ότι έχουμε 45 πιλότους.
- `--startDate`: σηματοδοτεί την ημερομηνία της αρχής του προγράμματος της εταιρείας και δίνεται στη μορφή έτος-μήνας-ημέρα. Η ώρα που αρχίζει το πρόγραμμα είναι τα μεσάνυχτα της ημερομηνίας που δίνεται. Οι δοσμένοι συνδυασμοί που ξεκινούν πριν από αυτήν την ημερομηνία δεν υπολογίζονται στον σχεδιασμό του προγράμματος. Αν δεν δοθεί το όρισμα, θεωρούμε ότι η αρχή του προγράμματος είναι 2001-01-01/00:00

- --endDate: αντίστοιχα με το προηγούμενο όρισμα σηματοδοτεί τη λήξη του προγράμματος και δίνεται με την ίδια μορφή. Η ώρα λήξης είναι τα μεσάνυχτα της ημερομηνίας που δίνεται. Οι συνδυασμοί που τελειώνουν μετά από αυτήν την ημερομηνία δεν υπολογίζονται. Πρέπει να προσέξουμε την ημερομηνία που θα δώσουμε, καθώς αν επιθυμούμε, για παράδειγμα, να υπολογίσουμε στον σχεδιασμό μας συνδυασμούς που τελειώνουν μέχρι και τις 2020-07-31, τότε θα πρέπει να δώσουμε ως 2020-08-01 την ημερομηνία λήξης. Αν δεν δοθεί αυτό το όρισμα, τότε ημερομηνία λήξης θεωρείται 2021-01-01/00:00
- --seed: είναι ένας ακέραιος που χρησιμοποιείται για την αρχικοποίηση της γεννήτριας τυχαίων αριθμών.
- --generations: είναι ο αριθμός των επαναλήψεων που θα εκτελέσει η εφαρμογή πριν τερματίσει. Αν δεν δοθεί, ο αλγόριθμος θα εκτελέσει 150 επαναλήψεις.

Αν επιλέξουμε την πολύ-βηματική CSO, τότε μπορούν να δοθούν προαιρετικά τα παρακάτω ορίσματα:

- --chickens: είναι ο αριθμός των κοτών που χρησιμοποιούνται από τον αλγόριθμο. Αν δεν δοθεί, χρησιμοποιούνται 20 κόττες.
- --FL: είναι η παράμετρος του αλγορίθμου. Εξ ορισμού, αρχικοποιείται σε 0.5.

Αντιθέτως, αν επιλέξουμε τη βελτιστοποίηση του Αρχιμήδη, τότε μπορούμε να ορίσουμε τα εξής:

- --objects: είναι ο αριθμός των αντικειμένων που χρησιμοποιούνται από τον αλγόριθμο. Αν δεν δοθεί, χρησιμοποιούνται 20 αντικείμενα.
- --C1, --C2, --C3, --C4: είναι οι παράμετροι του ΑΟΑ. Καμία δεν είναι υποχρεωτικό να δοθεί και μπορεί να παραλειφθεί ο ορισμός κάποιων ή και όλων των παραμέτρων. Σε αυτήν την περίπτωση, οι τιμές τους ορίζονται ως 2, 6, 1, 0.5 αντίστοιχα.

Το αρχείο των συνδυασμών πτήσεων είναι τύπου csv, όπου κάθε γραμμή αναπαριστά μια πτήση ενός συνδυασμού και έχει την ακόλουθη μορφή:

- ένα τετραψήφιο κωδικό, το αναγνωριστικό του συνδυασμού που ανήκει η πτήση
- έναν τριψήφιο κωδικό, τον αριθμό της πτήσης
- τρία γράμματα, το όνομα αεροδρομίου αναχώρησης
- τρία γράμματα, το όνομα αεροδρομίου άφιξης
- ημερομηνία αναχώρησης, στην μορφή έτος-μήνας-ημέρα
- ώρα αναχώρησης, στην μορφή ώρα:λεπτά
- ημερομηνία άφιξης, στην μορφή έτος-μήνας-ημέρα
- ώρα άφιξης, στην μορφή ώρα:λεπτά

2.2.2 Αεροπορική εταιρεία

Για την αναπαράσταση της αεροπορικής εταιρείας χρησιμοποιούμε τρεις δομές. Μια για τους συνδυασμούς, μια για τους πιλότους και μια για την ίδια την εταιρεία. Η δομή των συνδυασμών πέρα από τα πεδία που προκύπτουν από το αρχείο συνδυασμών, έχει και

κάποια άλλα βοηθητικά πεδία. Οι πληροφορίες που μπορούν να δοθούν από τη δομή είναι:

- id: αναγνωριστικός κωδικός του συνδυασμού
- start: ημερομηνία και ώρα έναρξης του συνδυασμού
- end: ημερομηνία και ώρα λήξης του συνδυασμού
- startDay: αριθμός ημερών από την αρχή του προγράμματος μέχρι την ημέρα έναρξης του συνδυασμού
- endDay: αριθμός ημερών από την αρχή του προγράμματος μέχρι την ημέρα λήξης του συνδυασμού
- duration: άθροισμα της διάρκειας των πτήσεων του συνδυασμού

Η δεύτερη δομή που χρησιμοποιούμε αφορά τους πιλότους. Η δομή των πιλότων έχει τα εξής πεδία:

- id: ακέραιος αριθμός, το αναγνωριστικό του πιλότου
- assignedPairs: λίστα με τους συνδυασμούς που έχει αναλάβει ο πιλότος, η σύνθεση, ταξινομημένη σε χρονολογική σειρά βάσει της έναρξης των συνδυασμών
- assignedLength: αριθμός συνδυασμών που έχει αναλάβει ο πιλότος
- flightTime: ώρες πτήσεις του πιλότου, προκύπτει από τις διάρκειες των συνδυασμών
- workdays: λίστα μήκους όσες και οι ημέρες του προγράμματος, που δείχνει πόσους συνδυασμούς έχουν ανατεθεί στον πιλότο κάθε μέρα.

Η τελευταία δομή αναπαριστά την αεροπορική εταιρεία. Αυτή η δομή χρησιμεύει περισσότερο ως μια ομαδοποίηση κάποιων χρήσιμων μεγεθών καθώς και στην χρήση αυτής της υλοποίησης από άλλη εφαρμογή με παρόμοιους κανόνες (π.χ. διαφορετική περίοδος ανάπαυσης ανάμεσα σε διαδοχικά καθήκοντα). Τα πεδία της είναι:

- pairsArray: πίνακας των συνδυασμών βάσει των οποίων φτιάχνουμε τις συνθέσεις
- numberOfPilots: αριθμός διαθέσιμων πιλότων
- scheduleStart: ημερομηνία αρχής προγράμματος
- scheduleEnd: ημερομηνία λήξης προγράμματος
- scheduleDuration: διάρκεια προγράμματος σε ημέρες
- averageWorkload: ιδανικός φόρτος εργασίας ανά πιλότο
- restPeriod: χρόνος ανάπαυσης σε λεπτά ανάμεσα σε δύο διαδοχικούς συνδυασμούς μιας σύνθεσης (εδώ 660 λεπτά)
- timespan: χρονική περίοδος σε ημέρες, στην οποία θέλουμε κάθε πιλότος να έχει τουλάχιστον έναν αριθμό ρεπό (εδώ, μια εβδομάδα).
- minimumDaysOff: ελάχιστος αριθμός ρεπό σε ημέρες που πρέπει να έχει η περίοδος timespan (εδώ 2 ημέρες).

Κατά την αρχικοποίηση, διαβάζουμε όλους τους συνδυασμούς από το αρχείο, τους φιλτράρουμε με βάση την ημερομηνία αρχής και λήξης του προγράμματος, τους ταξινομούμε βάσει της ημερομηνίας έναρξης τους και στην συνέχεια τους αποθηκεύουμε στον πίνακα συνδυασμών της εταιρείας. Αξίζει να σημειωθεί ότι σε αυτή την φάση μπορούν να οριστούν και τα πεδία restPeriod, timespan, minimumDaysOff. Ο

λόγος ταξινόμησης των συνδυασμών είναι ότι θέλουμε να τους εξετάσουμε με χρονολογική σειρά. Επίσης, ορίζουμε έναν ειδικό συνδυασμό ρίζα (root) με αναγνωριστικό 0, χωρίς πτήσεις και ημερομηνίες αρχής και λήξης ίσες με την ημερομηνία αρχής του προγράμματος. Είναι καθαρά βοηθητικός, χωρίς φυσική σημασία, και διευκολύνει τους υπολογισμούς. Τέλος, υπολογίζουμε τον ιδανικό φόρτο εργασίας ανά πιλότο, δηλαδή τον ιδανικό αριθμό ωρών πτήσης, βάσει του διαθέσιμου προσωπικού και του αριθμού των συνδυασμών.

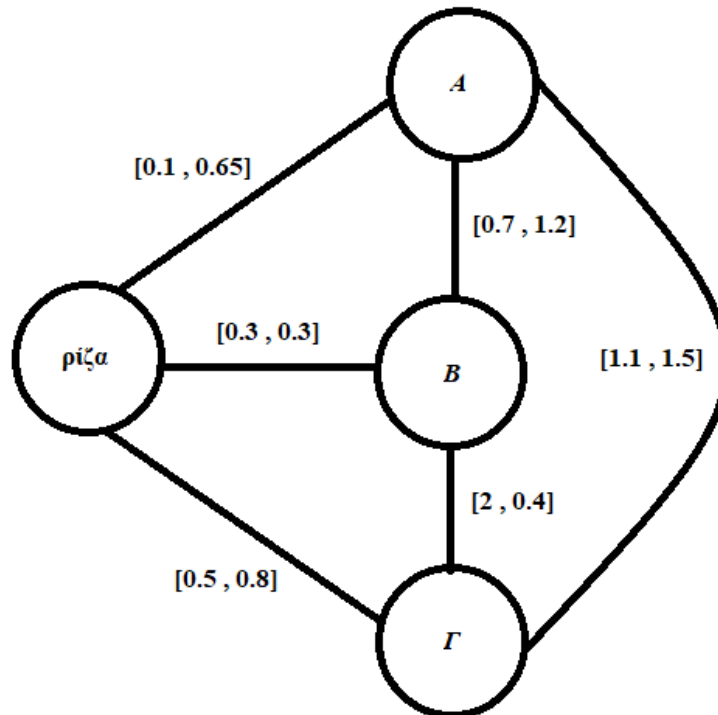
2.2.3 Ανανέωση θέσης

Ο μηχανισμός ανανέωσης θέσης είναι η βασική διαφορά μεταξύ της πολύ-βηματικής CSO και της βελτιστοποίησης του Αρχιμήδη. Παρόλα αυτά, και για τους δύο μπορεί να χρησιμοποιηθεί ένας γράφος γνώσης για την αποθήκευση και ενημέρωση της θέσης κάθε πράκτορα. Αυτή η μοντελοποίηση του χώρου αναζήτησης υιοθετήθηκε με επιτυχία στην πτυχιακή εργασία του Γ. Χαϊλαζόπουλου (2018) κατά την εφαρμογή της βελτιστοποίησης μέσω αποικίας μυρμηγκιών (ant colony optimization) στο πρόβλημα χρονοπρογραμματισμού πληρωμάτων και με την κατάλληλη προσαρμογή λειτουργεί ικανοποιητικά για τους παρόντες αλγορίθμους.

Η θέση ενός πράκτορα καθορίζεται από τις συνθέσεις που σχηματίζουν τη λύση του. Οι κόμβοι του γράφου αναπαριστούν τους συνδυασμούς που επιθυμούμε να αναθέσουμε, ενώ οι ακμές συμβολίζουν τις συνδέσεις αυτών των συνδυασμών. Δύο συνδυασμοί συνδέονται μεταξύ τους όταν υπάρχει μία σύνθεση στη λύση όπου οι δύο συνδυασμοί είναι διαδοχικοί. Επιπλέον, επειδή η σύνθεση ενός πιλότου είναι ταξινομημένη χρονολογικά, η σειρά με την οποία δίνονται οι συνδυασμοί είναι σταθερή για συνθέσεις που αποτελούνται από τους ίδιους συνδυασμούς. Θεωρούμε ότι το σημείο εκκίνησης κάθε σύνθεσης είναι ένας ειδικός κόμβος ρίζα (root), ο οποίος αντιστοιχεί στον ειδικό συνδυασμό ρίζα που αναφέραμε στην ενότητα 2.2.2. Για παράδειγμα, αν ένας πιλότος έχει αναλάβει μια σύνθεση με τρεις συνδυασμούς A, B, Γ , τότε θα αποθηκεύσουμε στο γράφο το μονοπάτι: $\text{ρίζα} \rightarrow A \rightarrow B \rightarrow \Gamma$. Οι ακμές κρατούν, επίσης, μία λίστα με τη θέση του κάθε πράκτορα, δηλαδή κατά πόσο η διαδοχική ανάθεση των κόμβων στα άκρα της ακμής οδηγεί σε ευνοϊκότερες λύσεις. Η λειτουργία του γράφου, τελικά, είναι να συγκεντρώνει όλα τα μονοπάτια που σχηματίζονται από τις συνθέσεις της λύσης κάθε πράκτορα και να ενημερώνει τις θέσεις κατά μήκος αυτών των μονοπατιών. Εδώ αξίζει να σημειωθεί ότι αν σχηματίσουμε ένα μονοπάτι βάσει της λύσης ενός πράκτορα, ενημερώνονται οι θέσεις όλων των πρακτόρων στο μονοπάτι αυτό.

Ο γράφος υλοποιείται με τη βοήθεια τριών δομών. Η πρώτη αναπαριστά τον ίδιο τον γράφο και περιέχει έναν πίνακα κατακερματισμού των κόμβων του, μία λίστα με τις ακμές του και τρία μεγέθη που σημειώνουν το πλήθος των κόμβων, το ακμών και των πρακτόρων. Ο πίνακας των κόμβων έχει ως κλειδιά τα αναγνωριστικά των συνδυασμών που αναπαριστούν οι κόμβοι. Η δεύτερη δομή είναι οι κόμβοι του γράφου και περιέχει έναν πίνακα κατακερματισμού με όλες τις ακμές οι οποίες έχουν το ένα τους άκρο στον κόμβο. Ως κλειδί του πίνακα χρησιμοποιείται το αναγνωριστικό του κόμβου που βρίσκεται στο άλλο άκρο της εκάστοτε ακμής. Η τρίτη δομή αντιπροσωπεύει την ακμή του γράφου και έχει ως πεδία έναν αναγνωριστικό αριθμό, τα αναγνωριστικά των κόμβων στα άκρα της και μια λίστα που αποθηκεύει τη θέση κάθε πράκτορα. Το αναγνωριστικό της ακμής προέρχεται από τη θέση της στη λίστα ακμών της δομής του γράφου. Συγκεντρώνοντας όλα τα παραπάνω, ο αριθμός των κόμβων του γράφου είναι ίσος με τον αριθμό των συνδυασμών (συμπεριλαμβανομένης και της ρίζας), ενώ οι ακμές κάθε κόμβου είναι το πολύ ίσες με τον αριθμό των συνδυασμών πλην ένα. Στο σχήμα 4, βλέπουμε την αναπαράσταση ενός γράφου με τρεις συνδυασμούς (A, B, Γ)

και δύο πράκτορες, όπου κάθε ζεύγος συνδυασμών μπορεί να υπάρξει στην ίδια σύνθεση. Στην πραγματικότητα, ένα ζεύγος που δεν υπακούει στον κανόνα ανάπαυσης μεταξύ διαδοχικών κόμβων δεν θα συνδέεται στον γράφο. Οι αριθμοί στις ακμές είναι οι θέσεις των πρακτόρων.



Σχήμα 4: Γράφος γνώσης για τρεις συνδυασμούς και δύο πράκτορες

Κατά την αρχικοποίηση του γράφου, δημιουργούμε όλους τους κόμβους, αφού τους γνωρίζουμε εκ των προτέρων, ενώ οι ακμές δημιουργούνται δυναμικά κατά την εισαγωγή προηγουμένως ανεξερεύνητων μονοπατιών. Η προκαθορισμένη τιμή κάθε καινούριας ακμής είναι 1 για την πολύ-βηματική CSO και ένας τυχαίος αριθμός στο διάστημα $[0.95, 1)$ για τη βελτιστοποίηση του Αρχιμήδη. Παραθέτουμε τις δομές του γράφου και τους μηχανισμούς ενημέρωσης κάθε αλγορίθμου.

Αλγόριθμος 3: Γράφος

```
1 Struct Graph contains
2   | map[int]*Node nodes;
3   | []*Edge edges;
4   | int numberOfNodes;
5   | int numberOfEdges;
6   | int agents;
7 end
```

Αλγόριθμος 4: Κόμβος γράφου

```
1 Struct Node contains
2   | map[int]*Edge Edges;
```

3 **end**

Αλγόριθμος 5: Ακμή γράφου

1 **Struct** *Edge* **contains**

2 | *int id*;

3 | *int sourceId*;

4 | *int goalId*;

5 | [*float64 position*];

6 **end**

Αλγόριθμος 6: Ανανέωση θέσης αλγορίθμου βελτιστοποίησης

Είσοδος: *graph, agents*

1 *edgeToUpdate* ← *make(map[int]*Edge)*

2 **foreach** *agent* **in** *agents* **do**

3 | *setUpdateParameters(agent, agents)*;

4 **end**

5 **foreach** *agent* **in** *agents* **do**

6 | **foreach** *roster* **in** *agent solution* **do**

7 | | **for** *i=0; i < len(roster); i++* **do**

8 | | | *pair1* ← *roster[i]*;

9 | | | *pair2* ← *roster[i+1]*;

10 | | | **if** *pair2* **not in** *graph.nodes[pair1].edges*

11 | | | | *edge* ← *new Edge*;

12 | | | | *graph.nodes[pair1].edges[pair2]* ← *edge*;

13 | | | | *graph.nodes[pair2].edges[pair1]* ← *edge*;

14 | | | | *graph.edges.append(edge)*;

15 | | | **end**

16 | | | *edgesToUpdate[edge.id]* ← *edge*;

17 | | **end**

18 **end**

19 **end**

20 **foreach** *edge* **in** *edgesToUpdate* **do**

21 | | **foreach** *agent* **in** *agents* **do**

22 | | | *updatePosition(agent, edge)*;

23 **end**

Η ανανέωση θέσης ξεκινάει με τον καθορισμό των απαιτούμενων παραμέτρων. Στην περίπτωση της πολύ-βηματικής CSO αυτές είναι τα τυχαία μέλη του σμήνους που χρησιμοποιούνται κατά τον υπολογισμό της καινούριας θέσης, ενώ στη βελτιστοποίηση του Αρχιμήδη είναι το τυχαίο αντικείμενο, όταν είμαστε στη φάση της εξερεύνησης, οι παράμετροι που εξαρτώνται από τον αριθμό της επανάληψης, καθώς και αυτές που εξαρτώνται από τις παραμέτρους που ορίζονται από τον χρήστη. Επίσης, υπολογίζουμε τις νέες πυκνότητες, όγκους και κανονικοποιημένες επιταχύνσεις των αντικειμένων. Αφού συγκεντρώσουμε όλες τις ακμές που πρέπει να ενημερωθούν σε έναν πίνακα, τις ανανεώνουμε βάσει των αλγορίθμων 7 και 8.

Αλγόριθμος 7: Ανανέωση θέσης μίας ακμής στην πολύ-βηματική CSO

Είσοδος: *chicken, edge*

- 1 *position* ← *edge.position[chicken.id]*;
 - 2 *position* ← *update using equation (8)*;
 - 3 *position* ← *update using equation (5)*;
 - 4 *position* ← *update using equation (11)*;
 - 5 *edge.position[chicken.id]* ← *position*;
-

Αλγόριθμος 8: Ανανέωση θέσης μίας ακμής στην βελτιστοποίηση του Αρχιμήδη

Είσοδος: *object, edge*

- 1 *position* ← *edge.position[object.id]*;
 - 2 **if** $TF \leq 0.5$ **do**
 - 3 | *position* ← *update using equation (24)*;
 - 5 **else do**
 - 6 | *position* ← *update using equation (25)*;
 - 19 **end**
 - 23 *edge.position[object.id]* ← *position*;
-

2.2.4 Κανόνες

Κατά τον ορισμό του προβλήματος, αναφέραμε ότι ένα έγκυρο πρόγραμμα συνθέσεων πρέπει να ικανοποιεί δύο συγκεκριμένους κανόνες. Ο πρώτος, τον οποίο ονομάσαμε κανόνα ανάπαυσης, δηλώνει ότι ανάμεσα σε δύο διαδοχικά ζευγάρια μιας σύνθεσης πρέπει να μεσολαβούν 11 ώρες (660 λεπτά). Ο κανόνας επαληθεύεται εύκολα με μία απλή σύγκριση των ωρών έναρξης και λήξης δύο συνδυασμών μπορούμε να συμπεράνουμε αν ισχύει. Επομένως, αν επιθυμούμε να προσθέσουμε έναν συνδυασμό σε μια σύνθεση, τον συγκρίνουμε με τους συνδυασμούς που ήδη υπάρχουν σ' αυτή και

έτσι μπορούμε να βρούμε όχι μόνο αν ισχύει ο κανόνας, αλλά και σε ποιο σημείο της σύνθεσης μπορούμε να τον προσθέσουμε για να τηρηθεί η χρονολογική σειρά της σύνθεσης.

Αλγόριθμος 9: Κανόνας ανάπαυσης (restPeriodRule)

Είσοδος: *pilot*, *pair*

Έξοδος: *index*

```
1  currentOverlap ← -1;
2  previousOverlap ← -1;
3  if pilot.assignedLength > 0 do
4    | currentOverlap = overlappingPairs(pilot.assignedPairs[pilot.assignedLength], pair)
5  end
6  if pilot.AssignedLength == 0 or currentOverlap == 1 do
7    | return pilot.assignedLength + 1;
8  else if currentOverlap == 0 do
9    | return -1;
10 end
11 for index = 1; index <= pilot.AssignedLength; index++ do
12 | pilotPair ← pilot.assignedPairs[index];
13 | currentOverlap = overlappingPairs(pilotPair, pair);
14 | if currentOverlap == 2 do
15 | | if index == 1 or previousOverlap == 1 do
16 | | | return index;
17 | | end
18 | else if currentOverlap == 0 do
19 | | break;
20 | end
21 | previousOverlap = currentOverlap;
22 end
23 return -1;
```

Η συνάρτηση *overlappingPairs* συγκρίνει δύο συνδυασμούς και μπορεί να επιστρέψει τρεις διαφορετικές τιμές, μηδέν αν υπάρχει επικάλυψη, δηλαδή αν χρονικά απέχουν λιγότερο από έντεκα ώρες, ένα αν δεν υπάρχει επικάλυψη και ο πρώτος συνδυασμός προηγείται του δεύτερου και δύο αν δεν υπάρχει επικάλυψη και ο πρώτος ακολουθεί τον δεύτερο. Ο αλγόριθμος επιστρέφει τη θέση που πρέπει να εισαχθεί ο συνδυασμός στο πρόγραμμα του πιλότου για να τηρηθεί η χρονική σειρά του και -1 σε περίπτωση που η ανάθεση θα οδηγήσει σε παραβίαση του κανόνα.

Ο δεύτερος κανόνας, τον οποίο ονομάσαμε κανόνα ρεπό, είναι λίγο πιο σύνθετος στην επαλήθευσή του, καθώς πρέπει να ελέγξουμε όλες τις περιόδους εφτά ημερών του προγράμματος ενός πιλότου που περιλαμβάνουν τις μέρες που διαρκεί ο συνδυασμός και να διαπιστώσουμε αν οι ημέρες ρεπό σε αυτά τα διαστήματα είναι τουλάχιστον δύο μετά την ανάθεση του στον πιλότο. Υπενθυμίζουμε ότι μια ημέρα είναι ρεπό για έναν πιλότο αν από τα μεσάνυχτα της μέχρι τα μεσάνυχτα της επομένης δεν υπάρχει στο πρόγραμμα του κάποιος συνδυασμός. Έστω ότι έχουμε το ακόλουθο πρόγραμμα διάρκειας 14 ημερών.

----- Σ Σ -----

Όπου Σ είναι ο καινούριος συνδυασμός που ξεκινάει την εβδομή και τελειώνει την όγδοη ημέρα. Προκειμένου να επιβεβαιώσουμε την εγκυρότητα της νέας σύνθεσης πρέπει να ελέγξουμε όλα τα διαστήματα 7 ημερών που περιλαμβάνουν την εβδομή ή και την όγδοη ημέρα. Στην χειρότερη περίπτωση, ο συνδυασμός δεν ξεκινά στις πρώτες έξι ημέρες και δεν τελειώνει στις έξι τελευταίες. Τότε, πρέπει να ελέγξουμε τουλάχιστον εφτά χρονικά διαστήματα. Στο παράδειγμά μας, τα διαστήματα αυτά είναι: 1-7, 2-8, 3-9, 4-10, 5-11, 6-12, 7-13 και 8-14. Αν σε οποιοδήποτε από αυτά τα διαστήματα έχουμε λιγότερες από δύο ημέρες ρεπό, τότε μπορούμε να συμπεράνουμε ότι η ανάθεση του συνδυασμού παραβιάζει τον κανόνα.

Ο τρόπος που περιγράψαμε παραπάνω για την εξέταση των διαστημάτων περιέχει αρκετούς επαναλαμβανόμενους υπολογισμούς τους οποίους μπορούμε να αποφύγουμε. Ας πάρουμε τα διαστήματα 1-7 και 2-8. Εύκολα παρατηρούμε ότι η μόνη τους διαφορά είναι ότι στο διάστημα 2-8 αντί για την πρώτη ημέρα, έχουμε την όγδοη. Ας υποθέσουμε ότι στο διάστημα 1-7 έχουμε τέσσερις εργάσιμες ημέρες και η πρώτη ημέρα είναι ρεπό. Τότε στο διάστημα 2-7 έχουμε δύο ρεπό και μένει απλώς να εξετάσουμε αν η όγδοη ημέρα είναι εργάσιμη για να αποφανθούμε για το διάστημα 2-8. Αυτό επεκτείνεται σε οποιοδήποτε ζεύγος διαδοχικών διαστημάτων και, επομένως, αν έχουμε έναν μετρητή για τα ρεπό, αρκεί να εξετάσουμε κάθε ημέρα του προγράμματος μία φορά για να επαληθεύσουμε τον κανόνα. Επιπλέον, παρατηρούμε ότι επειδή η μόνη αλλαγή στο πρόγραμμα αφορά τις ημέρες 7 και 8, αν διαπιστώνουμε παραβίαση του κανόνα στον διάστημα 8-14, τότε θα υπάρχει παραβίαση και στο διάστημα 7-13, δηλαδή για συνδυασμούς που διαρκούν περισσότερες από μία ημέρες αρκεί να ελέγξουμε τα διαστήματα που ξεκινούν έξι μέρες πριν την έναρξη του μέχρι το διάστημα που ξεκινά την ημέρα της έναρξής του.

Ο δεύτερος τρόπος και σημαντικότερος που μειώνουμε τους υπολογισμούς, προκύπτει από το γεγονός ότι εξετάζουμε τους συνδυασμούς με χρονολογική σειρά. Γίνεται εύκολα κατανοητό ότι όταν προσθέτουμε έναν συνδυασμό, τότε στην σύνθεση δεν ακολουθούν άλλοι, δεδομένου ότι έχουμε επαληθεύσει πρώτα τον κανόνα ανάπαυσης. Συνεπώς, αρκεί να ελέγξουμε τα διαστήματα που ξεκινούν μέχρι και έξι ημέρες πριν την έναρξη του συνδυασμού και τελειώνουν την ημέρα λήξης του. Στο παράδειγμά μας βλέπουμε ότι είναι τα διαστήματα 1-7 και 2-8. Στον κανόνα ανάπαυσης, δεν έχουμε κάποιο μηχανισμό που να λαμβάνει υπόψη τη χρονολογική σειρά εξέτασης, καθώς ο αλγόριθμος ελέγχει πρώτα αν ο συνδυασμός προστίθεται στο τέλος της σύνθεσης οπότε και τερματίζει, άρα δεν γίνονται μη αναγκαίοι υπολογισμοί.

Όλα τα παραπάνω, συνδυάζονται στον Αλγόριθμο 10. Ο λόγος που ο σχεδιασμός του αλγορίθμου προβλέπει για χρονολογική ή μη σειρά είναι ότι παρόλο που κατά την κατασκευή της λύσης εξετάζουμε με χρονολογική σειρά, στο τέλος της κατασκευής κάνουμε χρήση μιας συνάρτησης για την βελτίωση της λύσης (βλ. ενότητα 2.2.7), όπου η σειρά εξέτασης των συνδυασμών είναι διαφορετική.

Αλγόριθμος 10: Κανόνας ρεπό (*daysOffRule*)

Είσοδος: *airline, pilot, pair*

Έξοδος: *boolean*

```
1  daysOff ← airline.timespan;  
2  startPoint ← pair.startDay - airline.timespan;  
3  middlePoint ← pair.startDay;  
4  endPoint ← pair.startDay + airline.timespan;  
5  add 1 to pilot.workdays[i] for  $i \in [pair.startDay, pair.endDay]$   
6  if we examine in chronological order do  
7    | endPoint ← pair.endDay + 1;  
8  end  
9  for  $i = startPoint; i < middlePoint; i++$  do  
10   | if pilot.workdays[i] > 0 do  
11     | daysOff - -;  
12   | end  
13 end  
14 if daysOff < airline.minimumDaysOff do  
15   | return false;  
16 end  
17 for  $i = middlePoint; i < endPoint; i++$  do  
18   | if pilot.workdays[i-airline.timespan] > 0 do  
19     | daysOff + +;  
20   | end  
21   | if pilot.workdays[i] > 0 do  
22     | daysOff - -;  
23   | end  
24   | if daysOff < airline.minimumDaysOff do  
25     | return false;  
26   | end
```

```
27 end  
28 return true;
```

2.2.5 Κατασκευή λύσης

Η συνάρτηση κατασκευής λύσης που παρουσιάζεται στον αλγόριθμο 11, βρίσκει μια έγκυρη λύση του προβλήματος για έναν πράκτορα. Η συνάρτηση είναι κοινή τόσο για την πολύ-βηματική CSO όσο και για τη βελτιστοποίηση του Αρχιμήδη.

Αλγόριθμος 11: Κατασκευή λύσης

Είσοδος: *airline, graph, agentId*

Έξοδος: *pilotsArray, validSolution*

```
1 pilotsArray ← array of available pilots with empty rosters;  
2 foreach pair in airline.pairsArray do  
3   foreach pilot in pilotsArray do  
4     index ← restPeriodRule(pilot, pair);  
5     if index > -1 and daysOffRule(airline, pilot, pair) do  
6       previousPair ← pilot.AssignedPairs[index-1];  
7       edge ← graph.nodes[previousPair.Id].edges[pair.Id];  
8       position ← edge.Position[id];  
9       restPeriod ← time between pair and previousPair in hours - 10;  
10      propability ← position / restPeriod;  
11      candidates.append([pilot,index, propability]);  
12    end  
13  end  
14  selectedPilot ← selectPilot(candidates);  
15  add pair to selected pilot's roster;  
17 end  
17 if the solution is valid do  
18   equalizeWorkload(pilotsArray);  
19 end  
20 equalizeWorkload(pilotsArray);  
21 return pilotsArray, validSolution;
```

Ο τρόπος που λειτουργεί ο αλγόριθμος είναι να προσπαθεί να αναθέσει σε κάθε επανάληψη έναν συνδυασμό σε κάποιον πιλότο από τον πίνακα συνδυασμών της εταιρείας. Οι συνδυασμοί λαμβάνονται από τον πίνακα με χρονολογική σειρά με βάση την ημερομηνία και ώρα έναρξης τους. Στην συνέχεια, βρίσκει όλους τους πιλότους που μπορούν να τον αναλάβουν και τη θέση του κάθε πιλότου βάσει του προηγούμενου συνδυασμού της σύνθεσης του. Αφού συγκεντρώσουμε όλους τους υποψήφιους πιλότους σε μια λίστα καλούμε την συνάρτηση επιλογής υποψηφίου (*selectPilot*). Τέλος, αφού κατασκευάσουμε τη λύση καλούμε μια συνάρτηση (*equalizeWorkload*) που βελτιώνει ακόμα περισσότερο τη λύση. Εφόσον η λύση είναι έγκυρη, καλείται μία δεύτερη φορά, καθώς τότε αξίζει η διάθεση των πόρων για την βελτίωση της. Η εγκυρότητα μπορεί εύκολα να εντοπιστεί, αφού, αν κατά την κατασκευή δεν μπορούμε να αναθέσουμε τουλάχιστον ένα συνδυασμό σε κάποιον πιλότο, τότε η λύση είναι μερική.

2.2.6 Ανάθεση συνδυασμού

Η επιλογή πιλότου για την ανάθεση ενός συνδυασμού είναι το βασικό βήμα της συνάρτησης κατασκευής λύσης και παρομοίως με αυτή, είναι η ίδια για τους δύο αλγορίθμους βελτιστοποίησης. Αυτό που επιθυμούμε από τον μηχανισμό επιλογής είναι να καθιστά πιθανότερη την ανάθεση σε πιλότους με καλύτερες θέσεις. Για αυτόν τον λόγο, για κάθε πιλότο υπολογίζουμε την πιθανότητα να αναλάβει τον συνδυασμό βάσει της ακόλουθης εξίσωσης:

$$P = p \times h \quad (33)$$

Όπου P είναι η πιθανότητα ανάθεσης, p είναι η θέση του πιλότου και h είναι ένας ευρετικός μηχανισμός. Η πιθανότητα που αναφέρουμε εδώ δεν είναι η πιθανότητα ακριβώς όπως ορίζεται στα μαθηματικά, καθώς οι τιμές της μπορεί να είναι εκτός του διαστήματος $[0, 1]$. Για να βρεθεί η θέση ενός πιλότου, αρχικά βρίσκουμε τη θέση της σύνθεσης όπου εισάγεται ο καινούριος συνδυασμός καθώς και τον συνδυασμό που είναι στην ακριβώς προηγούμενη θέση της σύνθεσης. Έπειτα, βρίσκουμε τη σύνδεση στον γράφο γνώσης ανάμεσα στους δύο συνδυασμούς και με βάση το αναγνωριστικό του πράκτορα λαμβάνουμε τη θέση του πιλότου. Η θέση αυτή αντιπροσωπεύει τη γνώση που έχει αποκτήσει ο πράκτορας από προηγούμενες επαναλήψεις του αλγορίθμου βελτιστοποίησης, καθώς και από την αλληλεπίδραση του με τους υπόλοιπους πράκτορες και δείχνει κατά πόσο η διαδοχική ανάθεση δύο συνδυασμών σε έναν πιλότο οδήγησε σε καλύτερες λύσεις σε προηγούμενες επαναλήψεις.

Ας ξαναδούμε το παράδειγμα του γράφου στο σχήμα 4 για τον πρώτο πράκτορα, όπου μπορούμε εύκολα να βρούμε τη θέση όπου έχουμε τον συνδυασμό A μετά τη ρίζα σε μια σύνθεση ή τη θέση όπου ο B είναι μετά τον A , ακολουθώντας τις αντίστοιχες ακμές του γράφου. Ας υποθέσουμε ότι οι διαθέσιμοι πιλότοι για την κάλυψη των A , B , Γ είναι δύο (x και y). Η σειρά με την οποία εξετάζουμε τους συνδυασμούς είναι $B \rightarrow A \rightarrow \Gamma$, δηλαδή πρώτα ξεκινάει ο συνδυασμός B , ακολουθεί ο A και τέλος ο Γ . Επίσης, η λύση μέχρι τώρα έχει κατασκευαστεί ώστε ο πιλότος x να έχει αναλάβει τον συνδυασμό B και ο πιλότος y τον A :

¹ $x.assignedPairs = [root, B]$

2 $y.assignedPairs = [root, A]$

Απομένει η ανάθεση του συνδυασμού Γ σε έναν από τους δύο πιλότους. Αν αναζητήσουμε στον γράφο το μονοπάτι που ακολουθεί ο πιλότος x τότε αυτό είναι $ρίζα \rightarrow B \rightarrow \Gamma$ και η θέση του x έχει την τιμή 2. Αντίστοιχα ο πιλότος y ακολουθεί το μονοπάτι $ρίζα \rightarrow A \rightarrow \Gamma$ με αποτίμηση θέσης 1.1. Συνεπώς, η ανάθεση του συνδυασμού Γ στον πιλότο x είναι ευνοϊκότερη από την ανάθεση στον y . Σημειώνουμε ότι έχουμε υποθέσει ότι όλα τα μονοπάτια του παραδείγματος δεν παραβιάζουν τους κανόνες που ορίσαμε.

Ο υπολογισμός της πιθανότητας ενός πιλότου δεν βασίζεται μόνο στη θέση, άλλα και σε έναν ευρετικό μηχανισμό. Για το πρόβλημα μας επιλέξαμε ο μηχανισμός μας να εξαρτάται από το χρονικό διάστημα που μεσολαβεί ανάμεσα στον προηγούμενο συνδυασμό της σύνθεσης και στον καινούριο συνδυασμό. Με αυτόν τον τρόπο, ενθαρρύνουμε τον αλγόριθμο να δημιουργήσει προγράμματα με μικρά κενά (χρόνους ανάπαυσης) μεταξύ διαδοχικών συνδυασμών, κάτι που οδηγεί αφενός στην μείωση του ανεκμετάλλετου χρόνου και αφετέρου στη μεγαλύτερη διαθεσιμότητα των πιλότων για την ανάληψη επόμενων συνδυασμών. Ο ευρετικός μηχανισμός δίνεται από την εξίσωση (34), όπου έχουμε υπολογίσει τον χρόνο ανάπαυσης (*restPeriod*) σε ώρες και έχουμε λάβει υπόψη ότι θα είναι τουλάχιστον έντεκα ώρες.

$$h = \frac{1}{restPeriod - 10} \quad (34)$$

Έχοντας υπολογίσει τις πιθανότητες των υποψηφίων, μπορούμε να προχωρήσουμε στη διαδικασία επιλογής πιλότου, η οποία παρουσιάζεται στον αλγόριθμο 12:

Αλγόριθμος 12: Επιλογή πιλότου (*selectPilot*)

Είσοδος: *candidates* //list of candidate pilots

Έξοδος: *pilot*

```
1 offset ← | minimum propability from candidates list |;
2 if offset < 0 do
3   | add offset to each candidate pilot's propability;
4 end
5 sum ← sum of all candidate pilot's propabilities;
6 sumOfPropabilities ← rand * sum;
7 foreach pilot in candidates do
8   | sumOfPropabilities = sumOfPropabilities - propability of pilot;
9   | if sumOfPropabilities < 0 do
10  |   | return pilot;
11  | end
```

```
12 end
13 if no candidates are found do
14     | return none;
15 end
```

Η συνάρτηση λειτουργεί με απλό τρόπο. Αρχικά, επειδή οι πιθανότητες κάποιων πιλότων μπορεί να είναι αρνητικές πρέπει να κανονικοποιήσουμε όλες τις πιθανότητες ώστε η ελάχιστη τιμή τους να είναι το 0. Αυτό γίνεται βρίσκοντας τον πιλότο με την μικρότερη πιθανότητα και, εφόσον αυτή είναι αρνητική, την προσθέτουμε σε όλες τις πιθανότητες, συμπεριλαμβανομένης και της ίδιας. Έπειτα, υπολογίζουμε το άθροισμα όλων των κανονικοποιημένων πιθανοτήτων και το πολλαπλασιάζουμε με έναν τυχαίο αριθμό στο διάστημα $[0, 1]$. Ακολουθεί το κύριο σώμα της συνάρτησης όπου αφαιρούμε επαναληπτικά την πιθανότητα κάθε πιλότου από το πολλαπλασιασμένο άθροισμα μέχρι να μηδενιστεί, οπότε και διαλέγουμε τον πιλότο εκείνης της επανάληψης. Αυτή η τεχνική προσομοιώνει τον τροχό μιας ρουλέτας, η οποία επιλέγει τυχαία έναν αριθμό. Οι αριθμοί της ρουλέτας ισοδυναμούν με τις πιθανότητες των πιλότων, ενώ παράλληλα ευνοούνται οι αναθέσεις με μεγαλύτερες πιθανότητες. Η στιγμή που μηδενίζεται το πολλαπλασιασμένο άθροισμα ισοδυναμεί με την στιγμή που η μπίλια πέφτει σε κάποια ανάθεση και κάθε ανάθεση καταλαμβάνει μέρος της ρουλέτας ανάλογο της πιθανότητάς της. Τέλος, αν η λίστα των υποψηφίων είναι κενή, επιστρέφουμε την ανάλογη τιμή για την αδυναμία εύρεσης πιλότου.

2.2.7 Βελτίωση ίσου καταμερισμού

Στο τελευταίο βήμα της συνάρτησης κατασκευής λύσης καλούμε μια συνάρτηση (*equalizeWorkload*), η οποία έχει ως σκοπό να βελτιώσει περαιτέρω τη λύση που βρέθηκε. Η συνάρτηση προσπαθεί με άμεσο τρόπο να επηρεάσει το κριτήριο της αντικειμενικής συνάρτησης, που είναι η ελαχιστοποίηση της απόκλισης κάθε πιλότου από τον ιδανικό φόρτο εργασίας. Συνεπώς, οδηγούμαστε σε καλύτερες λύσεις με χαμηλότερο κόστος. Επίσης, επειδή η τιμή της αντικειμενικής συνάρτησης έχει άμεσο αντίκτυπο στη θέση των πρακτόρων των δύο αλγορίθμων βελτιστοποίησης που χρησιμοποιούμε, επηρεάζονται όλοι οι πράκτορες του πληθυσμού. Η συνάρτηση βελτίωσης ίσου καταμερισμού που εφαρμόζουμε είναι αρκετά απλή και γρήγορη και συνήθως, μετά τη δεύτερη εφαρμογή της στην ίδια λύση δεν είναι δυνατό να τη βελτιώσει άλλο. Η συνάρτηση εξαρτάται σε μεγάλο βαθμό από την αναλογία καθηκόντων και μελών του πληρώματος, καθώς όσο περισσότερα καθήκοντα έχουμε, τόσο μεγαλύτερος είναι ο ιδανικός φόρτος εργασίας και, επομένως, ο αλγόριθμος δεν μπορεί να κάνει πολλές και καλές αλλαγές στα προγράμματα των πιλότων.

Η συνάρτηση βρίσκει τους πιλότους με φόρτο εργασίας μεγαλύτερο του ιδανικού και προσπαθεί εξετάζοντας επαναληπτικά τους συνδυασμούς των προγραμμάτων τους να αφαιρέσει όσους περισσότερους μπορεί και να τους αναθέσει σε άλλα μέλη, τα οποία μπορούν να τον αναλάβουν βάσει των κανόνων. Για να γίνει μια αλλαγή στο πρόγραμμα ενός πιλότου, θα πρέπει η αλλαγή να οδηγήσει σε μείωση της απόκλισης των ωρών πτήσης από τις ιδανικές τόσο για εκείνον, όσο και για τον πιλότο που θα αναλάβει τον συνδυασμό που αφαιρέσει. Αυτή η τεχνική λειτουργεί καλύτερα σε πιο «αραιά» προγράμματα, δηλαδή με μεγαλύτερα κενά, καθώς τότε υπάρχουν περιθώρια για αλλαγές. Αντιθέτως, προγράμματα με μικρά κενά δεν έχουν πολύ χώρο για

μεταφορά των συνδυασμών. Θα μπορούσαν να υλοποιηθούν πιο πολύπλοκες συναρτήσεις, όπως συναρτήσεις όπου γίνονται ανταλλαγές συνδυασμών, προκειμένου να μην εξαρτώμαστε στον ίδιο βαθμό από τον διαθέσιμο κενό χώρο, παρόλα αυτά η συγκεκριμένη υλοποίηση λειτουργεί ικανοποιητικά για το πρόβλημά μας. Ο αλγόριθμος της συνάρτησης περιγράφεται παρακάτω.

Αλγόριθμος 13: Βελτίωση ίσου καταμερισμού (equalizeWorkload)

Είσοδος: *airline, pilots*

Έξοδος: *pilots*

```
1  foreach pilot1 in pilots do
2      if pilot1.flightTime > averageWorkload do
3          foreach pair in pilot1.assignedPairs do
4              difference ← pilot1.flightTime – averageWorkload;
5              if  $|difference - pair.Duration| < |difference|$  do
6                  foreach pilot2 != pilot1 in pilots do
7                      difference2 ← averageWorkload - pilot2.flightTime;
8                      if  $|difference2 - pair.Duration| < |difference2|$  do
9                          if pilot2 can add the pair to his roster do
10                             remove pair from pilot1's roster;
11                             add pair to pilot2's roster;
12                             break;
13                         end
14                     end
15                 end
16             end
17         end
18     end
19 end
```

3. ΚΡΙΤΗΡΙΑ ΑΞΙΟΛΟΓΗΣΗΣ

Για την αξιολόγηση των αποτελεσμάτων της εφαρμογής χρησιμοποιούνται μια σειρά κριτηρίων, τα οποία εκτιμούν τόσο τις παραμέτρους του εκάστοτε αλγορίθμου βελτιστοποίησης, όσο και διάφορες πτυχές όλων των λύσεων που βρέθηκαν κατά την εκτέλεση της εφαρμογής.

1. Έγκυρες και μερικές λύσεις

Υπολογίζουμε τον αριθμό των λύσεων που βρέθηκαν κατά τη διάρκεια εκτέλεσης, καθώς και πόσες από αυτές ήταν έγκυρες.

2. Μέση περίοδος ανάπαυσης

Υπολογίζουμε τον μέσο χρόνο ανάπαυσης ανά ζεύγος συνδυασμών για τη βέλτιστη λύση που προτάθηκε από τον αλγόριθμο, δηλαδή πόσο χρόνο έχει κάθε πιλότος για ξεκούραση μεταξύ δύο διαδοχικών συνδυασμών. Λαμβάνουμε υπόψη τον περιορισμό του κανόνα ανάπαυσης ώστε να έχουμε μια πιο ακριβή μέτρηση του διαστήματος αυτού. Γενικά, όσο μικρότερος ο χρόνος τόσο πιο μικρά κενά υπάρχουν στα προγράμματα των πιλότων και πιθανόν να μειώνεται και το κόστος της λύσης.

3. Μέσος αριθμός ρεπό

Επιπλέον υπολογίζουμε τον μέσο αριθμό ημερών ρεπό την εβδομάδα ανά πιλότο. Λόγω του κανόνα των ρεπό, ο αριθμός αυτός θα είναι πάντα μεγαλύτερος του 2. Σε αντιστοιχία με το προηγούμενο κριτήριο, όσο μικρότερος είναι ο αριθμός, τόσο μικρότερα είναι τα κενά και πιθανόν το κόστος του προγράμματος.

4. Κόστος βέλτιστης λύσης

Το κόστος αυτό προκύπτει από την εκτίμηση της καλύτερης λύσης που βρήκε ο αλγόριθμος κατά την εκτέλεση της εφαρμογής.

5. Καλύτερο και χειρότερο κόστος λύσης ανά επανάληψη

Υπολογίζουμε και το κόστος της καλύτερης και της χειρότερης λύσης σε κάθε επανάληψη.

6. Μέσο κόστος λύσης ανά επανάληψη

Σε κάθε επανάληψη, βρίσκουμε τον μέσο όρο του κόστους όλων των λύσεων που κατασκευάστηκαν από όλους τους πράκτορες στη δεδομένη γενιά.

7. Άλματα

Θεωρούμε ότι ο αλγόριθμος εκτελεί ένα άλμα όταν βρίσκει νέα βέλτιστη λύση σε μία επανάληψή του.

8. Μοναδικές λύσεις

Κατά τη διάρκεια εκτέλεσης της εφαρμογής, κρατάμε το πλήθος των διαφορετικών λύσεων που κατασκευάζει ο αλγόριθμος. Το κριτήριο δείχνει με ξεκάθαρο τρόπο κατά πόσο ο αλγόριθμος εξερευνά διαφορετικές περιοχές του χώρου αναζήτησης, καθώς επιθυμούμε να μην εξετάζουμε λύσεις που έχουμε βρει προηγουμένως ξανά. Για το κριτήριο δεν έχει σημασία η εγκυρότητα των λύσεων. Αναμένεται ότι εφόσον ο αριθμός συνδυασμών είναι επαρκώς μεγάλος, κάθε λύση θα είναι διαφορετική. Επειδή η αναπαράσταση των λύσεων, όπως την παρουσιάσαμε στο

κεφάλαιο 2, συντελεί στη χρονοβόρα σύγκριση μεταξύ τους, επιλέγουμε μια διαφορετική αναπαράσταση για την σύγκριση.

Αρχικά, κατασκευάζουμε μια λίστα μήκους όσο το πλήθος των συνδυασμών, όπου κάθε θέση της λίστας αντιπροσωπεύει έναν συνδυασμό. Η τιμή της θέσης είναι ο αναγνωριστικός αριθμός πιλότου στον οποίο ανατέθηκε ο συνδυασμός. Επειδή, οι συνδυασμοί εξετάζονται με την ίδια σειρά κάθε φορά, αν δύο λίστες είναι ίσες τότε αναφέρονται στην ίδια λύση.

Στη συνέχεια, συμπύσσουμε όλες τις τιμές της λίστας σε μία συμβολοσειρά, προκειμένου η σύγκριση να είναι πιο άμεση και γρήγορη. Σε αυτό το σημείο, πρέπει να αναφέρουμε ένα πρόβλημα που προκύπτει. Από τον ορισμό του προβλήματος γνωρίζουμε ότι δεν υπάρχει διαφοροποίηση μεταξύ των πιλότων, αφού θεωρούμε ότι έχουν όλοι τις ίδιες ικανότητες. Αυτό σημαίνει ότι μπορεί να σχηματιστούν συμμετρικές λύσεις. Για να αντιμετωπίσουμε το πρόβλημα αυτό και να μπορούμε να εντοπίζουμε τις συμμετρικές λύσεις ως ίδιες, δίνουμε σε κάθε πιλότο ένα ψευδώνυμο, το οποίο είναι ο αναγνωριστικός αριθμός του πρώτου συνδυασμού του προγράμματός του. Συνεπώς, πριν προχωρήσουμε στην κατασκευή της συμβολοσειράς αντικαθιστούμε τις τιμές της λίστας με τα αντίστοιχα ψευδώνυμα. Επειδή εξετάζουμε τους συνδυασμούς με την ίδια σειρά κάθε φορά, οι συμμετρικές λύσεις θα παράγουν τελικά την ίδια συμβολοσειρά.

Τέλος, αποθηκεύουμε τις διαφορετικές συμβολοσειρές σε έναν πίνακα κατακερματισμού, του οποίου τα κλειδιά είναι οι ίδιες οι συμβολοσειρές. Επειδή, ο πίνακας δεν κρατάει διπλότυπα, αρκεί να σημειώνουμε κάθε φορά το πλήθος των κλειδιών, το οποίο είναι το ίσο με το πλήθος των διαφορετικών λύσεων.

Αλγόριθμος 14: Κωδικοποίηση λύσης

Είσοδος: *condesedSolution, completeSolution, uniqueSolsMap, keyCount*

Έξοδος: *normalisedSolution*

```
1  normalisedSolution ← []string;
2  pilotAliases ← map[int]string;
3  foreach pilot in completeSolution do
4  | pilotAliases[pilot.Id] ← pilot.AssignedPairs[1].Id as string;
5  end
6  foreach pilot in condesedSolution do
7  | normalisedSolution.append(pilotAliases[pilotId]);
8  end
9  solutionString = normalisedSolution.join();
10 If solutionString not in uniqueSolsMap do
11 | keyCount ++;
12 | uniqueSolsMap[solutionString] = true;
13 end
14 return normalisedSolution;
```

Ο αλγόριθμος 14 συγκεντρώνει όλα τα παραπάνω. Αφού δημιουργήσει τη λίστα με τα ψευδώνυμα και την αντίστοιχη συμβολοσειρά, την αποθηκεύει στον πίνακα μοναδικών λύσεων, εφόσον πρόκειται για νέα λύση και ενημερώνει τον μετρητή των κλειδιών του πίνακα. Στο τέλος, επιστρέφει τη λίστα με τα ψευδώνυμα.

9. Ομοιότητα λύσεων

Το τελευταίο κριτήριο που χρησιμοποιούμε είναι η ομοιότητα κάθε λύσης που βρίσκουμε με τη βέλτιστη λύση και καταδεικνύει την αναλογία εξερεύνησης και εκμετάλλευσης του χώρου αναζήτησης. Χρησιμοποιούμε τις λίστες ψευδωνύμων που επιστρέφονται από τον αλγόριθμο 14 και υπολογίζουμε το ποσοστό ομοιότητας βάσει του πλήθους των συνδυασμών που ανατέθηκαν σε διαφορετικούς πιλότους στις συγκρινόμενες λύσεις.

Αλγόριθμος 15: Σύγκριση λύσεων

Είσοδος: *solutionA*, *solutionB*

Έξοδος: *similarity*

```
1 smallestLength ← min(len(solutionA), len(solution));
2 biggestLength ← max(len(solutionA), len(solution));
3 difference ← biggestLength – smallestLength;
4 for i = 0; i < smallestLength; i++ do
5   | if solutionA[i] != solutionB[i] do
6   |   | difference ++;
7   | end
8 end
9 similarity = (1.0 - difference/biggestLength) * 100.0
10 return similarity;
```

4. ΑΠΟΤΕΛΕΣΜΑΤΑ

Για την παραγωγή των αποτελεσμάτων και την εξαγωγή συμπερασμάτων, διεξάγαμε μια σειρά πειραμάτων για τον κάθε αλγόριθμο βελτιστοποίησης παίρνοντας ως είσοδο ένα αρχείο 3304 συνδυασμών που καλύπτουν μια περίοδο 123 ημερών. Ο πρώτος συνδυασμός ξεκινάει την 1/11/2011 και ο τελευταίος τελειώνει την 3/3/2012. Για την πολύ-βηματική CSO εξετάσαμε κάθε δυνατή τιμή της παραμέτρου FL , ενώ για την βελτιστοποίηση του Αρχιμήδη εξετάσαμε κάθε δυνατό συνδυασμό των παραμέτρων $C1$, $C2$, $C3$, $C4$. Χρησιμοποιήσαμε συνολικά 44 διαφορετικές αρχικοποιήσεις της γεννήτριας τυχαίων αριθμών και για τις δύο περιπτώσεις. Παρουσιάζουμε διάφορα στατιστικά στοιχεία των αλγορίθμων, χωρίς να δείχνουμε το παραγόμενο πρόγραμμα των πιλότων για λόγους συντομίας. Η μορφή του προγράμματος περιγράφεται στο Παράρτημα Ι. Λόγω της τυχαιότητας που εισάγεται από τους αλγορίθμους χρησιμοποιούμε ένα τυχαίο πρόγραμμα που παράγεται από τον πολύ-βηματικό αλγόριθμο βελτιστοποίησης μέσω σμήνους κοτών, το οποίο θεωρούμε αποδεκτό ως προς την ποιότητά του, ως μέτρο σύγκρισης της αποδοτικότητας των λύσεων.

Πιλότοι: 45, Κότες: 20, Επαναλήψεις: 200, Αρχείο εισόδου: Pairings.csv

Αρχή προγράμματος: 2011-11-1 00:00:00, Λήξη: 2012-03-04 00:00:00,

FL: 1, αρχικοποίηση γεννήτριας: 1622631426338403900

Μέση περίοδος ανάπαυσης: 12.73 ώρες

Μέσος αριθμός ρεπό: 3.04 μέρες, Μέση ομοιότητα λύσεων: 2.25%

Μέσος φόρτος εργασίας: 20468, Συνδυασμοί: 3304

Κόστος λύσης: 30891

Έγκυρες λύσεις: 3026, Μη έγκυρες λύσεις: 974

Χρόνος εκτέλεσης: 1 min. 15.8151488 sec.

Πίνακας 1: Στατιστικά έγκυρου προγράμματος

Πιλότοι	Χρόνος πτήσης (σε λεπτά)	Απόκλιση από τον μέσο φόρτο εργασίας (σε λεπτά)	Ανατιθέμενοι Συνδυασμοί
0	20450	18	72
1	20445	23	72
2	20515	47	78
3	20485	17	64
4	20490	22	72
5	20455	13	77
6	20435	33	75
7	20445	23	67

8	20460	8	77
9	20435	33	70
10	20465	3	88
11	20470	2	77
12	20490	22	68
13	20455	13	73
14	20490	22	69
15	20445	23	78
16	20495	27	75
17	20470	2	73
18	20490	22	73
19	20430	38	75
20	20430	38	66
21	20430	38	83
22	20465	3	74
23	20465	3	78
24	20475	7	68
25	20450	18	70
26	20465	3	79
27	20450	18	74
28	20470	2	71
29	20515	47	70
30	20495	27	78
31	20460	8	66
32	20450	18	75
33	20470	2	65

34	20465	3	79
35	20425	43	80
36	20505	37	83
37	20510	42	76
38	20495	27	70
39	20465	3	67
40	20480	12	70
41	20480	12	72
42	20510	42	66
43	20520	52	74
44	20420	48	77

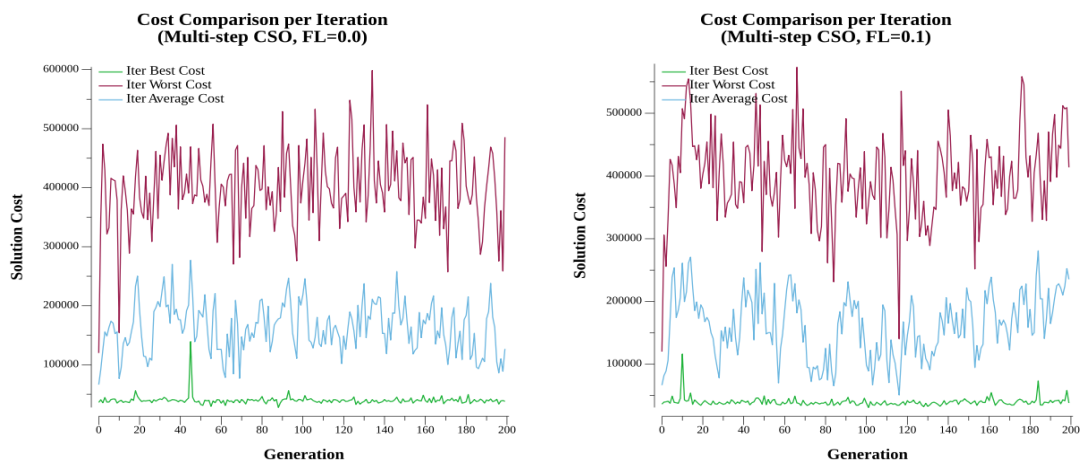
Ο λόγος που θεωρούμε την λύση αυτή ως αποδεκτή είναι ότι η απόκλιση κάθε πιλότου από τον ιδανικό, δηλαδή τον μέσο, χρόνο πτήσης είναι λιγότερο από μία ώρα. Επομένως, μπορούμε να θέσουμε ένα άνω όριο στο κόστος της λύσης βάσει του οποίου να αξιολογήσουμε τις λύσεις που προκύπτουν από τις διάφορες τιμές των παραμέτρων.

Όλα τα πειράματα χρησιμοποιούν τον ίδιο αριθμό πρακτόρων και επαναλήψεων, ενώ διαφοροποιούνται μόνο στις τιμές παραμέτρων και στην αρχικοποίηση της γεννήτριας τυχαίων αριθμών. Ως επί το πλείστον χρησιμοποιούμε 45 πιλότους, ενώ αναφέρουμε όποτε χρησιμοποιούμε διαφορετικό πλήθος πιλότων.

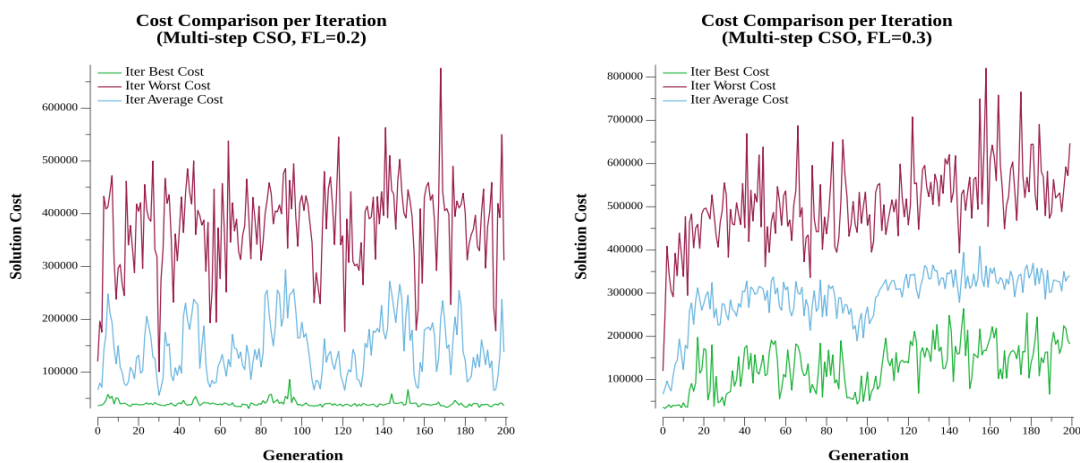
4.1 Αξιολόγηση πολύ-βηματικής CSO

Η αξιολόγηση του αλγορίθμου είναι αρκετά απλή, αφού αυτή η εκδοχή του εξαρτάται μόνο από την παράμετρο FL . Αυτή καθορίζει ως ένα βαθμό κατά πόσο μία κότα επηρεάζεται από κάποια άλλη. Αν η τιμή της είναι πολύ μικρή τότε η επίδραση των άλλων κοτών εφαρμόζεται μόνο μέσω του μηχανισμού ανανέωσης θέσης κόκκορα, ενώ αν είναι πολύ μεγάλη τότε μια κότα μπορεί να ακολουθήσει μια άλλη σε ένα κακό σημείο του χώρου αναζήτησης. Εξετάσαμε τιμές στο εύρος $[0,2]$ προκειμένου να διαπιστώσουμε ποιες είναι οι καλύτερες τιμές της παραμέτρου και παραθέτουμε παρακάτω γραφήματα που περιγράφουν την συμπεριφορά του αλγορίθμου. Παρατηρούμε ότι κάποιες τιμές ξεχωρίζουν, κυρίως η τιμή 1, όπου έχουμε το χαμηλότερο μέσο κόστος λύσης καθ' όλη την εκτέλεση και επιπλέον όλα τα μεγέθη μένουν σχετικά σταθερά, που σημαίνει ότι δεν ξεφεύγουμε σε κακές περιοχές του χώρου. Την ίδια σταθερότητα, αλλά σε μικρότερο βαθμό βλέπουμε και για τις τιμές 0, 1.4, 1.5 και 1.6.

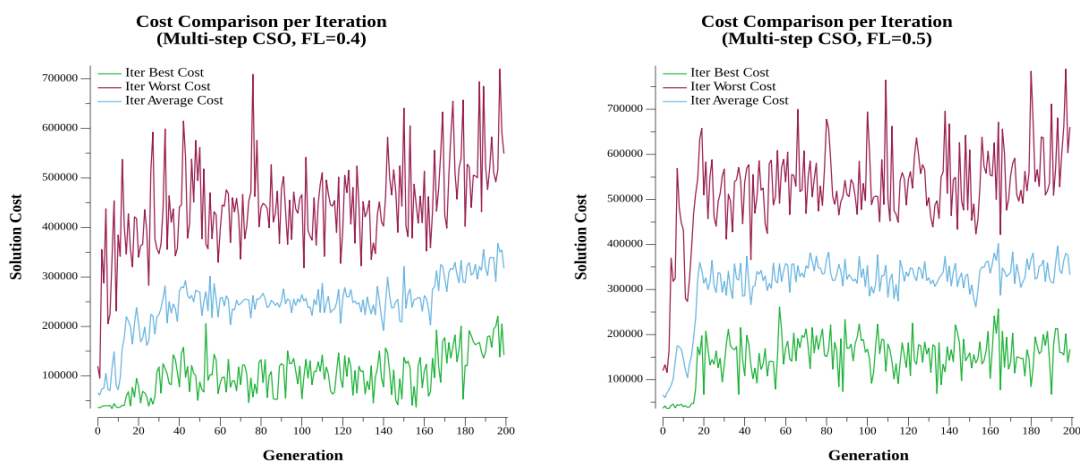
Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών



Σχήμα 5: FL=0 (αριστερά), FL=0.1 (δεξιά)

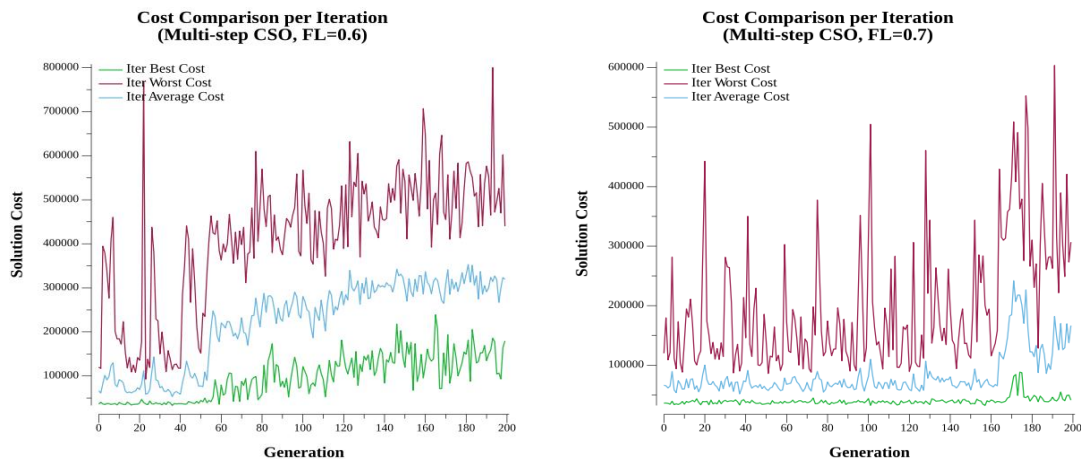


Σχήμα 6: FL=0.2 (αριστερά), FL=0.3 (δεξιά)

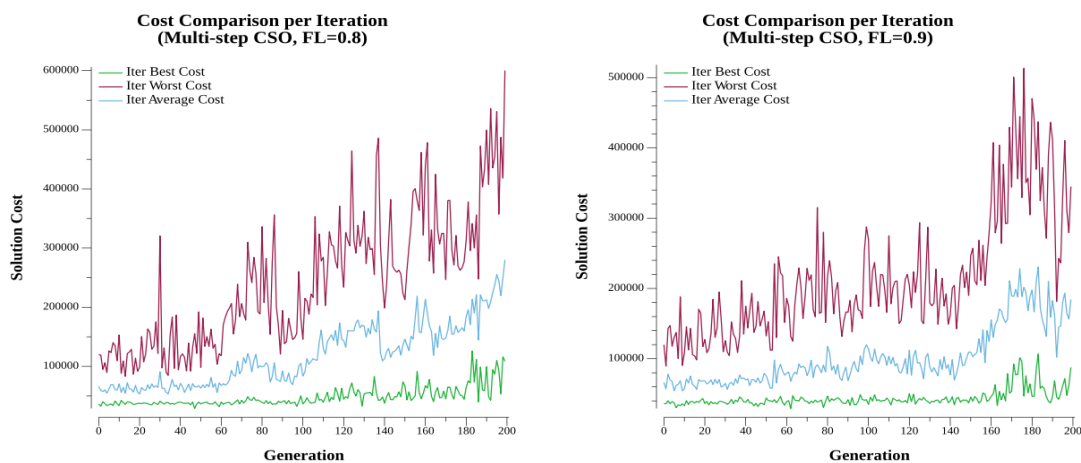


Σχήμα 7: FL=0.4 (αριστερά), FL=0.5 (δεξιά)

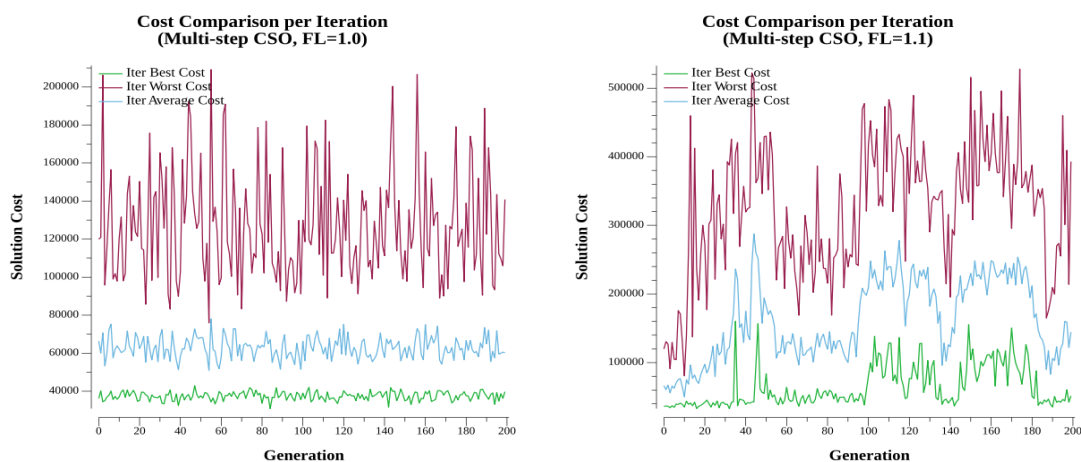
Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών



Σχήμα 8: FL=0.6 (αριστερά), FL=0.7 (δεξιά)

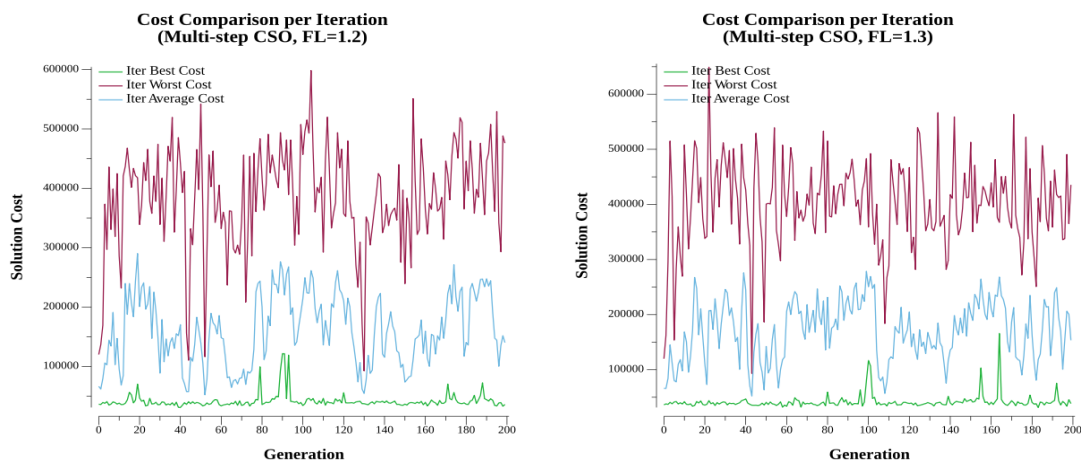


Σχήμα 9: FL=0.8 (αριστερά), FL=0.9 (δεξιά)

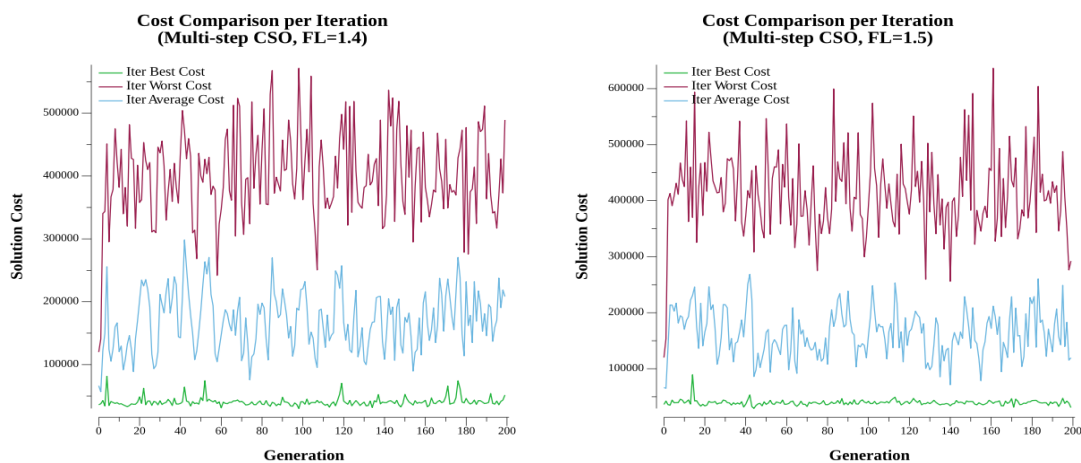


Σχήμα 10: FL=1 (αριστερά), FL=1.1 (δεξιά)

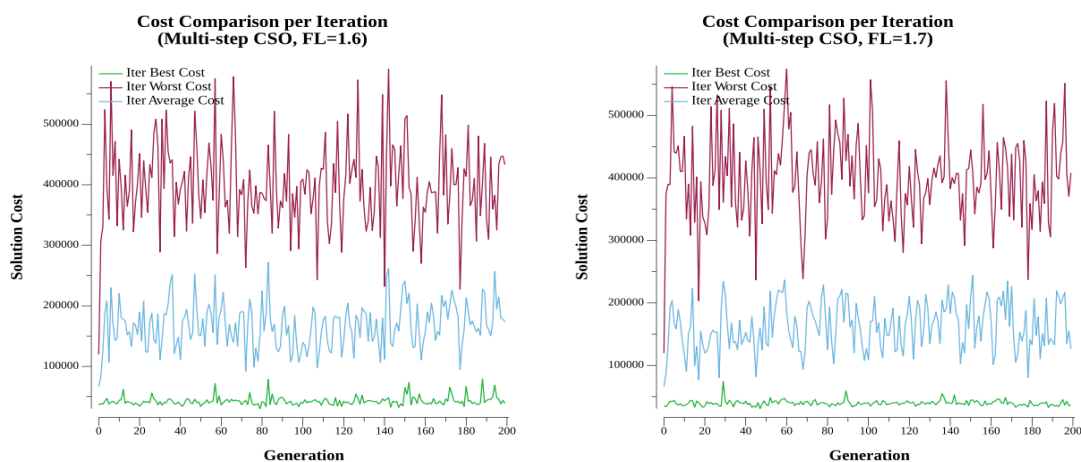
Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών



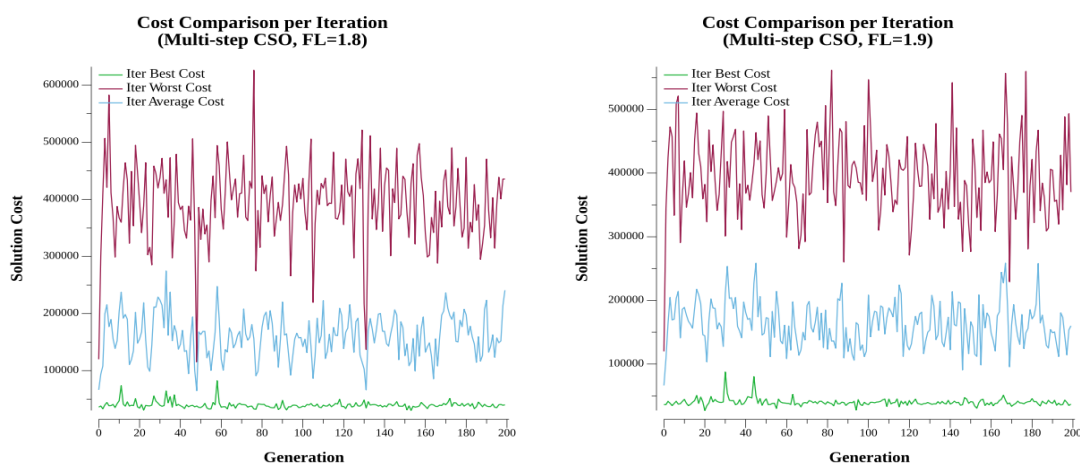
Σχήμα 11: FL=1.2 (αριστερά), FL=1.3 (δεξιά)



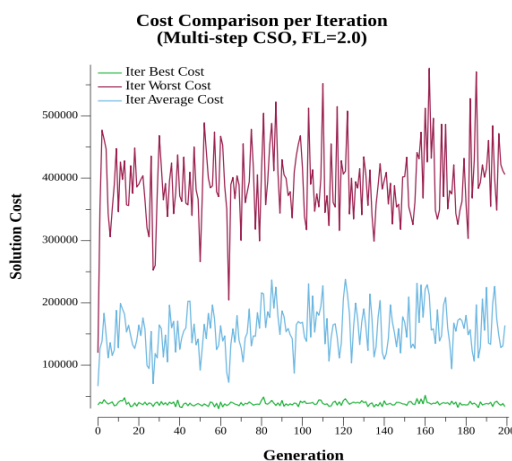
Σχήμα 12: FL=1.4 (αριστερά), FL=1.5 (δεξιά)



Σχήμα 13: FL=1.6 (αριστερά), FL=1.7 (δεξιά)



Σχήμα 14: FL=1.8 (αριστερά), FL=1.9 (δεξιά)



Σχήμα 15: FL=2.0

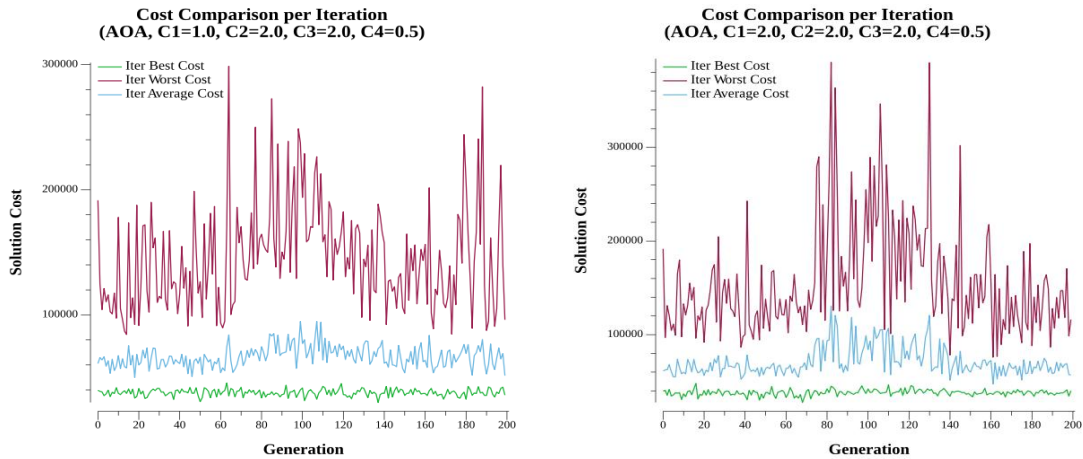
4.2 Αξιολόγηση βελτιστοποίησης του Αρχιμήδη

Στην περίπτωση του αλγορίθμου βελτιστοποίησης του Αρχιμήδη η αξιολόγηση είναι λίγο πιο σύνθετη, καθώς πρέπει να εκτιμήσουμε την επίδραση τεσσάρων διαφορετικών παραμέτρων ($C1$, $C2$, $C3$, $C4$) στα αποτελέσματα του αλγορίθμου, ξεχωριστά για την καθεμία, προτού προχωρήσουμε στην εύρεση του καταλληλότερου συνδυασμού τιμών τους.

4.2.1 Εκτίμηση παραμέτρου $C1$

Η παράμετρος $C1$ χρησιμοποιείται στη φάση εξερεύνησης του χώρου αναζήτησης και καθορίζει σε ποιον βαθμό επηρεάζεται κάθε αντικείμενο από κάποιο άλλο. Επηρεάζει τον αλγόριθμο άμεσα κατά τη διάρκεια της εξερεύνησης, η οποία διαρκεί για περίπου το ένα τρίτο των μέγιστων επαναλήψεων του αλγορίθμου, καθώς και έμμεσα την αρχή της φάσης εκμετάλλευσης. Εξετάσαμε δύο τιμές της παραμέτρου (1 και 2) και παραθέτουμε δύο διαγράμματα για αυτές τις τιμές με μέγιστο αριθμό επαναλήψεων τις 200. Σε αυτά παρατηρούμε ότι η τιμή 1 οδηγεί σε χαμηλότερα κόστη κατά την εξερεύνηση, καθώς και

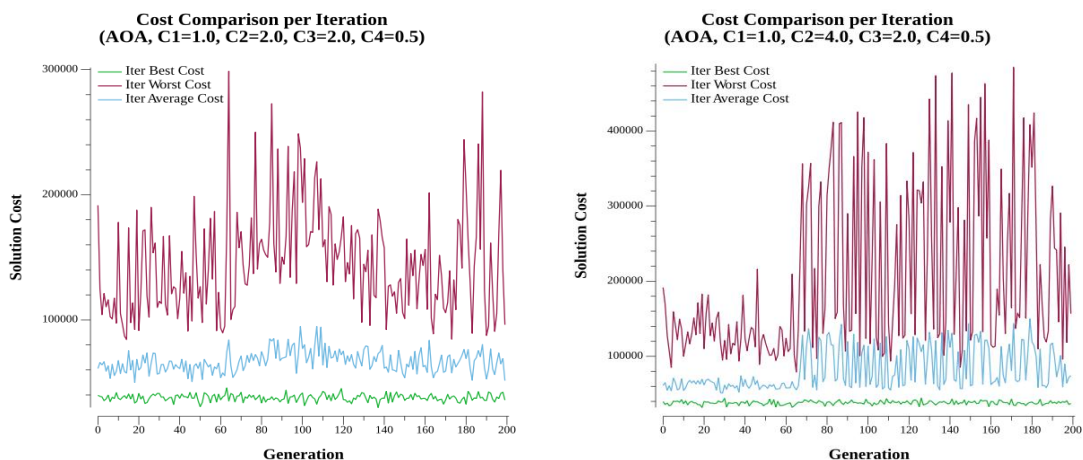
μια ομαλή μετάβαση στο στάδιο της εκμετάλλευσης δημιουργεί. Αντίθετα, η τιμή 2 εμφανίζει μεγαλύτερα κόστη κατά την εξερεύνηση και επιπλέον μια αύξηση των κοστών λύσης στο πρώτο κομμάτι της φάσης εκμετάλλευσης. Αναμένουμε ότι η τιμή 1 θα είναι καταλληλότερη καθώς φαίνεται να εκτελεί την εξερεύνηση καλύτερα, ενώ παράλληλα θέτει μια καλή βάση για τον αλγόριθμο, ώστε να προχωρήσει στη δεύτερη φάση της εκτέλεσής του. Ταυτόχρονα, μια καλή επιλογή των υπόλοιπων τριών παραμέτρων θα εξασφαλίσει τα βέλτιστα δυνατά αποτελέσματα.



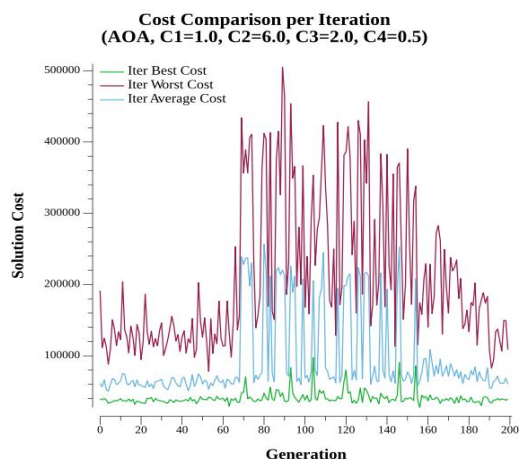
Σχήμα 16: C1=1 (αριστερά), C1=2 (δεξιά)

4.2.2 Εκτίμηση παραμέτρου C2

Η παράμετρος C2 χρησιμοποιείται κατά τη διάρκεια του σταδίου εκμετάλλευσης και καθορίζει τον βαθμό στον οποίο μεταβάλλεται η θέση ενός αντικειμένου. Μεγαλύτερες τιμές οδηγούν σε μεγαλύτερες μεταβολές. Δοκιμάζουμε τρεις διαφορετικές τιμές (2, 4 και 6) και παραθέτουμε διαγράμματα ώστε να αντιληφθούμε τις διαφορές τους.



Σχήμα 17: C2=2 (αριστερά), C2=4 (δεξιά)



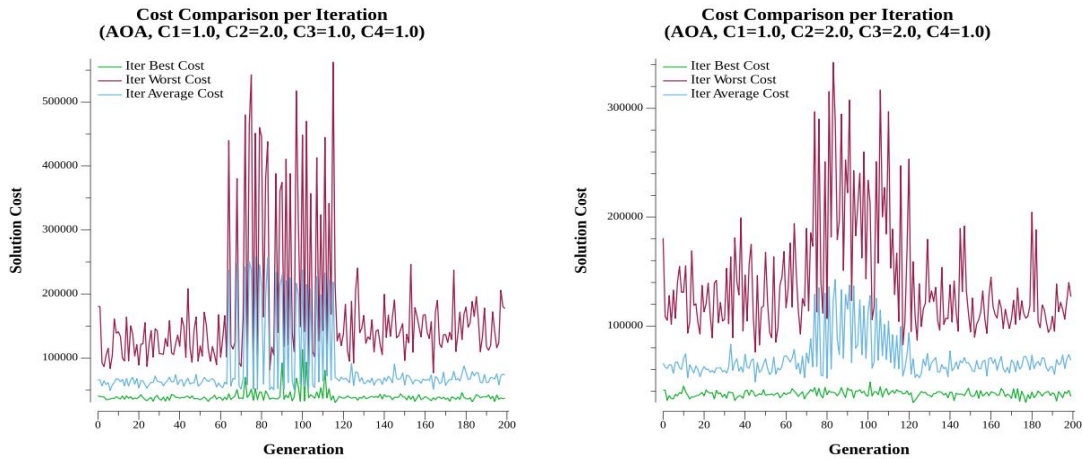
Σχήμα 18: C2=6

Παρατηρούμε ότι στο πρώτο τρίτο των επαναλήψεων (φάση εξερεύνησης), τα διαγράμματα είναι ίδια. Η τιμή 2 εμφανίζει μικρές διακυμάνσεις στα κόστη μεταξύ των επαναλήψεων, αφού οι μεταβολές των θέσεων είναι μικρότερες. Η τιμή 4 εμφανίζει την χειρότερη απόδοση από πλευράς διακυμάνσεων, καθώς υπάρχει αστάθεια καθ' όλη τη διάρκεια της εκμετάλλευσης, κάτι που εξηγείται από τις μεγαλύτερες μεταβολές των θέσεων. Η συμπεριφορά της τιμής 6 είναι η πιο ενδιαφέρουσα, αφού στο πρώτο μισό της φάσης εκμετάλλευσης υπάρχουν μεγάλες διακυμάνσεις, ενώ στο δεύτερο μισό παρουσιάζεται εξομάλυνση. Αυτό συμβαίνει διότι στο πρώτο μισό καλύπτεται γρήγορα η απόσταση των θέσεων μεταξύ τρέχοντος και καλύτερου αντικειμένου, ενώ στη συνέχεια ο αλγόριθμος εξερευνά περιοχές που βρίσκονται κοντά στην καλύτερη λύση, με αποτέλεσμα οι καινούργιες λύσεις που βρίσκουμε να έχουν κόστη με μικρές διαφορές από τα ήδη υπάρχοντα.

4.2.3 Εκτίμηση παραμέτρου C3

Η παράμετρος C3 χρησιμοποιείται κατά τη φάση της εκμετάλλευσης και καθορίζει σε συνδυασμό με την παράμετρο TF το ποσοστό της θέσης του καλύτερου αντικειμένου που θα χρησιμοποιηθεί για τη μεταβολή της θέσης του τρέχοντος αντικειμένου. Εξετάζουμε δύο τιμές της παραμέτρου (1 και 2) και παρουσιάζουμε τα αντίστοιχα διαγράμματα στο σχήμα 19.

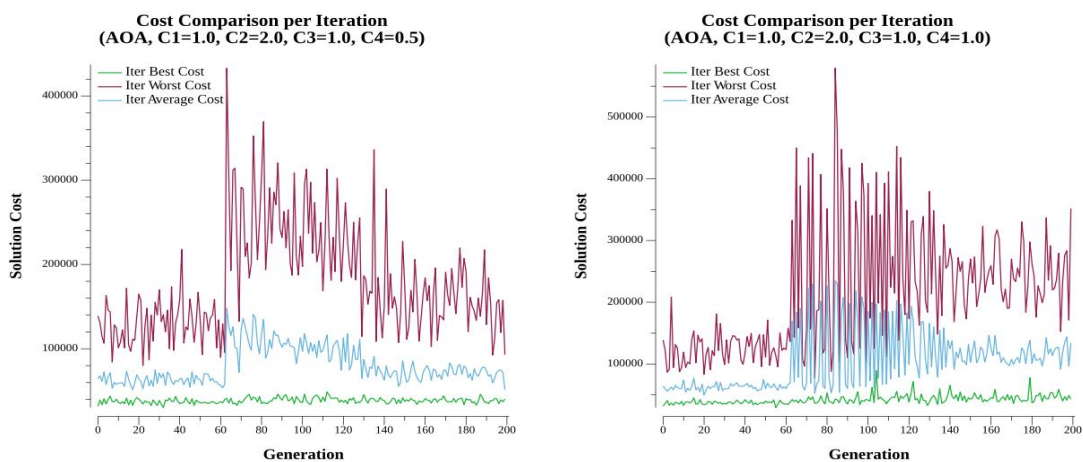
Γενικά, φαίνεται ότι η παράμετρος επηρεάζει κυρίως το πρώτο κομμάτι του σταδίου εκμετάλλευσης. Η τιμή 1 εμφανίζει σημαντικά αυξημένα κόστη λύσεων και μεγαλύτερες διακυμάνσεις μεταξύ των επαναλήψεων, κάτι που μας οδηγεί στο συμπέρασμα ότι δεν είναι κατάλληλη για την καλύτερη απόδοση του αλγορίθμου. Αντιθέτως, η τιμή 2 εμφανίζει μικρότερα κόστη στο πρώτο μέρος του σταδίου εκμετάλλευσης, παρόλο που εμφανίζει μια διακύμανση, παρόμοια με τη συμπεριφορά της τιμής 1, η οποία, όμως, παρουσιάζεται σε μικρότερο βαθμό. Αυτό συμβαίνει διότι, στην πρώτη περίπτωση παίρνουμε ένα μικρό ποσοστό της καλύτερης θέσης με αποτέλεσμα μια μεγάλη μεταβολή της θέσης του εκάστοτε αντικειμένου, ενώ προς το τέλος της εκτέλεσης του αλγορίθμου οι μεταβολές των θέσεων είναι πολύ μικρές, με αποτέλεσμα να μην ξεφεύγουμε αρκετά από τις λύσεις που έχουν ήδη βρεθεί. Στη δεύτερη περίπτωση, ξεκινούμε με μεγαλύτερο ποσοστό της καλύτερης θέσης, με αποτέλεσμα να καλύπτεται γρήγορα η απόσταση από αυτήν, ενώ στη συνέχεια οι μεταβολές που προκαλούνται είναι στα κατάλληλα πλαίσια ώστε να μην προκαλούνται μεγάλες διακυμάνσεις και τα κόστη των καινούργιων λύσεων να παραμένουν χαμηλά.



Σχήμα 19: C3=1 (αριστερά), C3=2 (δεξιά)

4.2.4 Εκτίμηση παραμέτρου C4

Η παράμετρος $C4$ χρησιμοποιείται, μαζί με τις προηγούμενες δύο παραμέτρους, στον μηχανισμό ανανέωσης θέσης των αντικειμένων κατά τη φάση της εκμετάλλευσης του χώρου αναζήτησης και προσομοιώνει την αλλαγή κατεύθυνσης ενός αντικειμένου. Εξετάσαμε δύο τιμές της παραμέτρου (0.5 και 1). Η τιμή 0.5 καθιστά ισοπίθανη την αλλαγή ή μη της κατεύθυνσης, ενώ η τιμή 1 ευνοεί την αλλαγή. Παραθέτουμε δύο ενδεικτικά διαγράμματα που επιδεικνύουν την επίδραση της παραμέτρου παρακάτω.



Σχήμα 20: C4=0.5 (αριστερά), C4=1 (δεξιά)

Παρατηρούμε, όπως με τις παραμέτρους $C2$ και $C3$, ότι στο πρώτο τρίτο των επαναλήψεων (φάση εξερεύνησης), δεν εμφανίζονται ιδιαίτερες διαφορές μεταξύ των διαγραμμάτων. Η τιμή 1 φαίνεται να οδηγεί σε μια έντονη παλινδρόμηση μεταξύ των επαναλήψεων, κάτι που εξηγείται από το γεγονός ότι τα αντικείμενα αλλάζουν συχνότερα την κατεύθυνσή τους, με αποτέλεσμα να μεταβάλλονται πιο εύκολα οι θέσεις τους και ορισμένες φορές να εξασθενούν θέσεις οι οποίες είναι καλές. Η τιμή 0.5 παρουσιάζει μικρότερες παλινδρομήσεις, αφού είναι πιθανότερο τα αντικείμενα να συνεχίσουν στην ίδια κατεύθυνση και, επομένως, να ενισχύουν τις θέσεις που οδηγούν στις καλύτερες λύσεις.

4.3 Εκτέλεση πειραμάτων

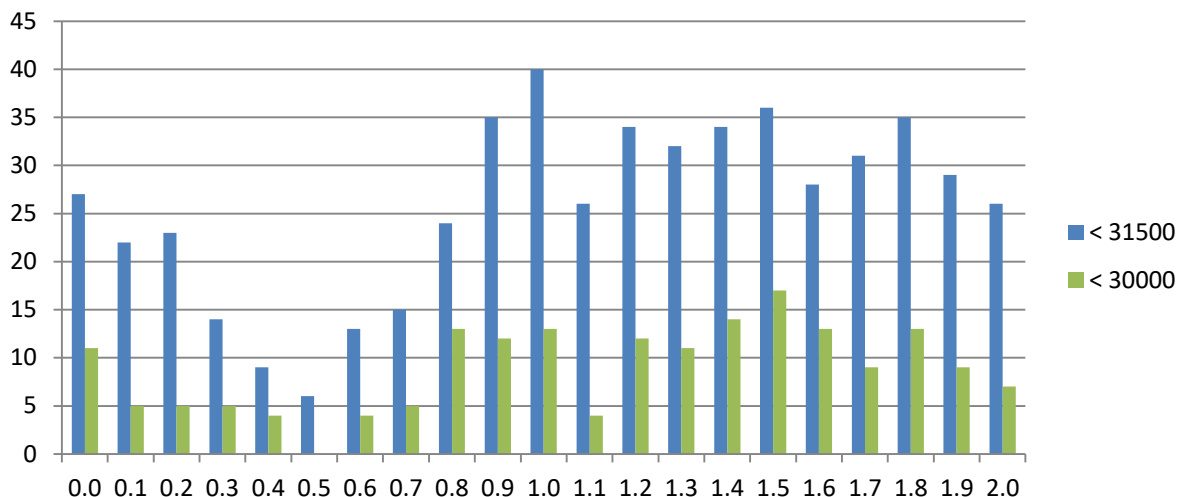
Προκειμένου να αξιολογήσουμε την απόδοση των αλγορίθμων και να εκτιμήσουμε τις παραμέτρους τους, εφαρμόσαμε τρία διαφορετικά σενάρια του προβλήματος χρονοπρογραμματισμού πτήσεων, στα οποία καταλήξαμε βάσει του πλήθους και της μορφής των συνδυασμών πτήσεων του αρχείου εισόδου. Σκοπός των πειραμάτων είναι να συμπεράνουμε τις ιδανικότερες τιμές ή τον ιδανικότερο συνδυασμό τιμών παραμέτρων των αλγορίθμων βελτιστοποίησης. Δεν αναζητούμε απλώς τις τιμές που αποδίδουν το καλύτερο κόστος λύσης του πειράματος, αλλά τις τιμές εκείνες, οι οποίες παράγουν αποδεκτές ή καλές λύσεις του προβλήματος τις περισσότερες φορές. Δηλαδή, επιθυμούμε να έχουμε παραμέτρους, οι οποίες θα μπορούν με μία σχετική αξιοπιστία να κατασκευάσουν μια καλή λύση. Ο τρόπος με τον οποίο ορίζουμε μια λύση ως αποδεκτή βασίζεται στο κόστος της. Δεδομένων όσων αναφέραμε στην εισαγωγή του κεφαλαίου για τα πλαίσια των πειραμάτων θεωρούμε ότι μια λύση με κόστος μικρότερο του 31500 είναι αποδεκτή. Αντίστοιχα, μια λύση της οποίας το κόστος είναι μικρότερο του 30000 θεωρούμε ότι είναι αρκετά καλή.

Όσον αφορά τα σενάρια, στο πρώτο, θεωρούμε ότι πρέπει να διαμοιράσουμε τους συνδυασμούς μεταξύ 45 πιλότων. Σε αυτό το σενάριο, ο χώρος των λύσεων είναι αρκούτσως μεγάλος, ώστε να βρίσκονται λύσεις με ευκολία, με τις ιδανικότερες τιμές παραμέτρων να αποδίδουν καλύτερα κόστη λύσεων την πλειονότητα των φορών. Για αυτόν τον λόγο, αυτό είναι το κύριο σενάριο, βάσει του οποίου εξαγάγουμε συμπεράσματα για τις διαφορετικές τιμές των παραμέτρων. Στο δεύτερο σενάριο έχουμε 40 διαθέσιμους πιλότους και το μέγεθος του χώρου λύσεων είναι αρκετά περιορισμένο. Σε αυτήν την περίπτωση, εξετάζουμε τις επιλεγμένες τιμές παραμέτρων που προκύπτουν από το πρώτο σενάριο ώστε να αξιολογήσουμε την απόδοσή τους. Στο τελευταίο σενάριο, κατασκευάζουμε πρόγραμμα για 50 πιλότους. Σε αυτό το σενάριο, ο χώρος λύσεων είναι αρκετά χαλαρός, οπότε αναμένουμε οι περισσότεροι πράκτορες να μπορούν να βρουν μια έγκυρη λύση. Όπως και στη δεύτερη περίπτωση, αξιολογούμε την επίδοση των επιλεγμένων τιμών του πρώτου σεναρίου σε έναν διαφορετικό χώρο λύσεων.

4.3.1 Πείραμα πολύ-βηματικής CSO για 45 πιλότους

Παρουσιάζουμε στον πίνακα 6 του παραρτήματος II τα στατιστικά στοιχεία κάθε εκτέλεσης του αλγορίθμου πολύ-βηματικής CSO, για τις περιπτώσεις όπου το κόστος λύσης είναι μικρότερο του 31500. Τα αποτελέσματα είναι ταξινομημένα βάσει κόστους. Για το πείραμα διεξάγαμε συνολικά 924 εκτελέσεις του αλγορίθμου για 21 διαφορετικές τιμές τις παραμέτρου FL , με κάθε τιμή να δοκιμάζεται 44 φορές. Επομένως, για λόγους συντομίας δεν περιλαμβάνουμε στον πίνακα λύσεις με μεγαλύτερο κόστος του 31500. Στο σχήμα 21 παρουσιάζουμε τον αριθμό εμφανίσεων κάθε εξεταζόμενης τιμής σε λύσεις κόστους μικρότερου του 31500 και μικρότερου του 30000. Δίνουμε μεγαλύτερη βαρύτητα στην πρώτη κατηγορία καθώς μας ενδιαφέρει πρωτίστως να βρίσκουμε αποδεκτές λύσεις, ενώ χρησιμοποιούμε τη δεύτερη προκειμένου να συγκρίνουμε τιμές οι οποίες έχουν περίπου τον ίδιο αριθμό εμφανίσεων στην πρώτη.

Παρατηρούμε ότι για την τιμή 1 έχουμε τις περισσότερες εμφανίσεις για την πρώτη κατηγορία (40 λύσεις), με την τιμή 1.5 να ακολουθεί (36 λύσεις), ενώ στη δεύτερη κατηγορία η τιμή 1.5 είναι πρώτη (17 λύσεις) και η τιμή 1 είναι η τρίτη καλύτερη (13 λύσεις). Λόγω της σημαντικής διαφοράς στην πρώτη κατηγορία, μπορούμε να συμπεράνουμε ότι η κατάλληλη τιμή της παραμέτρου FL είναι 1.



Σχήμα 21: Επίδραση παραμέτρου FL

σε αποδεκτές (μπλε) και πολύ καλές (πράσινο) λύσεις

4.3.2 Πείραμα πολύ-βηματικής CSO για 40 πιλότους

Στον πίνακα 2 του παραρτήματος II παρουσιάζουμε τα αποτελέσματα του αλγορίθμου όταν έχουμε διαθέσιμους 40 πιλότους για την τιμή 1 της FL , βάσει του συμπεράσματος που εξάγαμε στο προηγούμενο πείραμα. Ο πίνακας είναι ταξινομημένος βάσει του πλήθους λύσεων που βρέθηκαν κατά τη διάρκεια της εκτέλεσης. Σε αυτό το σενάριο, αξιολογούμε την ικανότητα του αλγορίθμου να βρίσκει τουλάχιστον μία έγκυρη λύση, όταν ο χώρος λύσεων είναι πολύ περιορισμένος. Παρατηρούμε, ότι η κατασκευή μιας έγκυρης λύσης είναι αρκετά δύσκολη, με τους περισσότερους πράκτορες να αποτυγχάνουν. Παρόλα αυτά, ο αλγόριθμος σε ένα πλήθος 44 εκτελέσεων κατάφερε να βρει έγκυρη λύση στην πλειονότητα τους (41 εκτελέσεις).

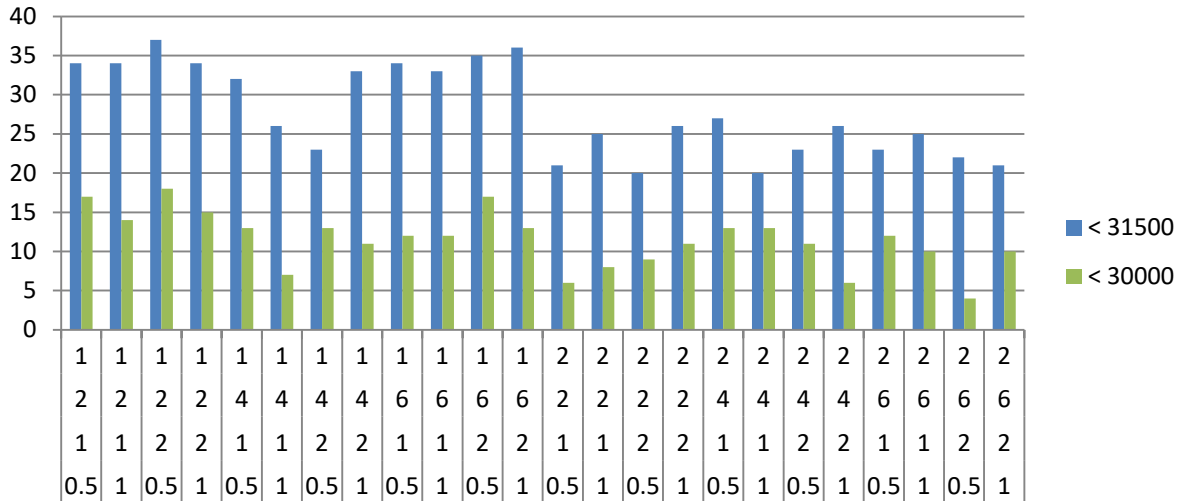
4.3.3 Πείραμα πολύ-βηματικής CSO για 50 πιλότους

Σε αυτό το σενάριο, λόγω του γεγονότος ότι ο χώρος λύσεων είναι υπερβολικά μεγάλος, αξιολογούμε τόσο την ικανότητα των κοτών να βρίσκουν έγκυρες λύσεις σε κάθε γενιά, όσο και τα κόστη των λύσεων να είναι χαμηλά. Όπως και στο προηγούμενο πείραμα, εξετάζουμε τον αλγόριθμο για την τιμή 1 της FL , 44 φορές και παρουσιάζουμε τα αποτελέσματα στον ταξινομημένο κατά βέλτιστο κόστος πίνακα 3 του παραρτήματος II. Παρατηρούμε ότι σε όλες τις εκτελέσεις δεν βρέθηκε καμία άκυρη λύση, ενώ το καλύτερο κόστος λύσης όλων των εκτελέσεων ήταν 29325.

4.3.4 Πείραμα βελτιστοποίησης του Αρχιμήδη για 45 πιλότους

Παρουσιάζουμε στον πίνακα 7 του παραρτήματος II τα αποτελέσματα κάθε εκτέλεσης του αλγορίθμου βελτιστοποίησης του Αρχιμήδη, για τις περιπτώσεις όπου το κόστος λύσης είναι μικρότερο του 31500. Τα αποτελέσματα είναι ταξινομημένα βάσει κόστους. Για το πείραμα διεξάγαμε συνολικά 1056 εκτελέσεις του αλγορίθμου για 24 διαφορετικούς συνδυασμούς τιμών των παραμέτρων $C1$, $C2$, $C3$ και $C4$, με κάθε

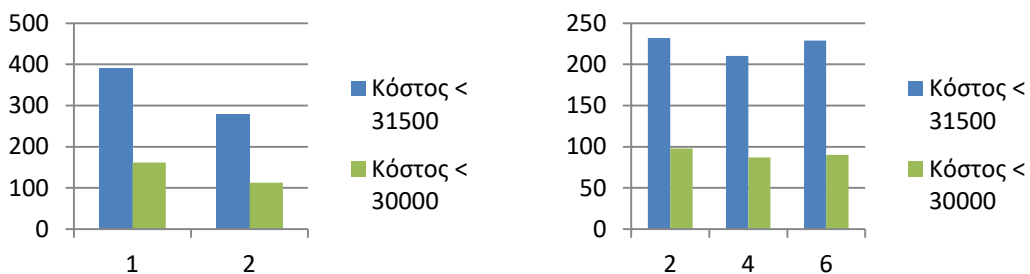
συνδυασμό να δοκιμάζεται 44 φορές. Επομένως, για λόγους συντομίας δεν περιλαμβάνουμε στον πίνακα λύσεις με μεγαλύτερο κόστος του 31500. Στο σχήμα 22 παρουσιάζουμε τον αριθμό εμφανίσεων κάθε εξεταζόμενου συνδυασμού σε λύσεις κόστους μικρότερου του 31500 και μικρότερου του 30000. Οι ετικέτες του οριζόντιου άξονα διαβάζονται από πάνω προς τα κάτω και αντιστοιχούν με αυτήν την σειρά στις παραμέτρους $C1$, $C2$, $C3$ και $C4$. Όπως και στο πείραμα της ενότητας 4.3.1, δίνουμε μεγαλύτερη βαρύτητα στην πρώτη κατηγορία για τους ίδιους λόγους.



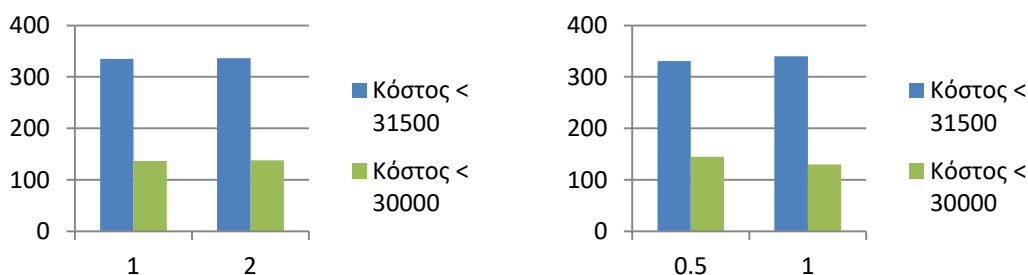
Σχήμα 22: Επίδραση συνδυασμού παραμέτρων $C1$, $C2$, $C3$, $C4$ σε αποδεκτές (μπλε) και πολύ καλές (πράσινο) λύσεις

Αναπαριστούμε έναν συνδυασμό ως $(C1, C2, C3, C4)$ και παρατηρούμε ότι τρεις ξεχωρίζουν, οι $(1, 2, 2, 0.5)$, $(1, 6, 2, 0.5)$ και $(1, 6, 2, 1)$ με 37, 35 και 36 εμφανίσεις στην πρώτη κατηγορία και 18, 17 και 13 στην δεύτερη. Συμπεραίνουμε, επομένως, ότι ο συνδυασμός $(1, 2, 2, 0.5)$ είναι καταλληλότερος, εφόσον υπερέχει και στις δύο κατηγορίες, παρόλο που ο συνδυασμός $(1, 6, 2, 0.5)$ δεν απέχει πολύ σε απόδοση. Εντούτοις, οδηγούμαστε στην επιλογή παραμέτρων $C1 = 1$, $C2 = 2$, $C3 = 2$, $C4 = 0.5$, τις οποίες και αξιολογούμε στα εναπομείναντα δύο σενάρια.

Στην συνέχεια, παραθέτουμε στα σχήματα 23 και 24 διαγράμματα που δείχνουν την επίδραση κάθε παραμέτρου στον αλγόριθμο. Από αυτά, τα διαγράμματα του σχήματος 23 παρουσιάζουν το μεγαλύτερο ενδιαφέρον, καθώς φαίνεται ότι για $C1 = 1$ έχουμε σαφώς καλύτερα αποτελέσματα, ενώ για $C2 = 2$ υπάρχει μια ελαφρώς μεγαλύτερη αξιοπιστία στην εύρεση καλών και αποδεκτών λύσεων. Δυστυχώς, δεν μπορούμε να συμπεράνουμε κάτι για τις παραμέτρους $C3$ και $C4$ από τα διαγράμματα του σχήματος 24.



Σχήμα 23: Επίδραση παραμέτρου $C1$ (αριστερά) και $C2$ (δεξιά)



Σχήμα 24: Επίδραση παραμέτρου C3 (αριστερά) και C4 (δεξιά)

4.3.5 Πείραμα βελτιστοποίησης του Αρχιμήδη για 40 πιλότους

Σε αντιστοιχία των πειραμάτων της πολύ-βηματικής CSO, αξιολογούμε την απόδοση του καλύτερου συνδυασμού, $(1, 2, 2, 0.5)$, που βρήκαμε στο προηγούμενο πείραμα, ώστε να διαπιστώσουμε την ικανότητα του αλγορίθμου να βρίσκει έγκυρες λύσεις. Στον πίνακα 4 του παραρτήματος II παρουσιάζονται όλα τα αποτελέσματα, ταξινομημένα κατά πλήθος έγκυρων λύσεων. Παρατηρούμε ότι από τις 44 εκτελέσεις του αλγορίθμου μόνο οι 4 δεν μπόρεσαν να οδηγηθούν σε έγκυρη λύση, ενώ το πλήθος τους ανά εκτέλεση είναι σαφώς μεγαλύτερο σε σχέση με την πολύ-βηματική CSO. Όσον αφορά τα κόστη τα περισσότερα είναι αρκετά υψηλά.

4.3.6 Πείραμα βελτιστοποίησης του Αρχιμήδη για 50 πιλότους

Στο τελευταίο σενάριο που εξετάζουμε, αξιολογούμε την ικανότητα των αντικειμένων να βρίσκουν έγκυρες λύσεις σε κάθε γενιά, καθώς και τα βέλτιστα κόστη λύσεων του αλγορίθμου για τον συνδυασμό $(1, 2, 2, 0.5)$. Τα αποτελέσματα αναγράφονται στον πίνακα 5 του παραρτήματος II. Παρατηρούμε ότι μόλις 3 εκτελέσεις από τις 44 έχουν άσχημες επιδόσεις ως προς το πλήθος λύσεων, 2 βρίσκουν 1 με 2 άκυρες λύσεις, ενώ οι υπόλοιπες εκτελέσεις δεν βρίσκουν καμία άκυρη λύση. Όσον αφορά τα κόστη, παραμένουν χαμηλά, με το καλύτερο να είναι 30298.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Τα αποτελέσματα των πειραμάτων μας αποδεικνύουν ότι οι δύο αλγόριθμοι βελτιστοποίησης που εφαρμόζουμε, δηλαδή η πολύ-βηματική βελτιστοποίηση μέσω σμήνους κοτών και η βελτιστοποίηση του Αρχιμήδη, είναι ικανοί να επιλύσουν ένα σύνθετο πρόβλημα πολλών διαστάσεων όπως είναι ο χρονοπρογραμματισμός πληρωμάτων. Οι αλγόριθμοι στην πλειονότητα των περιπτώσεων μπόρεσαν να φτάσουν με αποδοτικό τρόπο σε καλές λύσεις στις όποιες η απόκλιση κάθε πιλότου από τις ιδανικές ώρες εργασίας ήταν μόλις κάποια λεπτά. Η ταχύτητα των αλγορίθμων ήταν αρκετά γρήγορη, καθώς σε έναν οκταπύρηνο επιτραπέζιο υπολογιστή, οι χρόνοι εκτέλεσης κυμαίνονταν περίπου στο 1 λεπτό και 10 δευτερόλεπτα ως 1 λεπτό και 20 δευτερόλεπτα, όπως φαίνεται στους πίνακες του παραρτήματος II. Εδώ αξίζει να σημειωθεί ότι τρέχαμε ταυτόχρονα 4 με 5 εκτελέσεις της εφαρμογής κάθε φορά, ενώ σε περιπτώσεις όπου τρέχαμε μόνο 1 εκτέλεση, οι χρόνοι ήταν σαφώς μειωμένοι (50 με 55 δευτερόλεπτα).

Το μεγαλύτερο μέρος αυτής της πτυχιακής εργασίας επικεντρώθηκε στην ικανότητα των δύο προτεινόμενων αλγορίθμων βελτιστοποίησης να επιλύσουν μία απλοποιημένη μορφή του προβλήματος χρονοπρογραμματισμού πληρωμάτων. Η εφαρμογή που αναπτύχθηκε μπορεί εύκολα να επεκταθεί, με την συμπερίληψη άλλων κανόνων, ώστε να δοκιμάσουμε τους αλγορίθμους σε πιο πολύπλοκα προβλήματα. Επιπλέον, θα μπορούσαν εξεταστούν διαφορετικοί ευρετικοί μηχανισμοί, καθώς και πιο σύνθετες συναρτήσεις που βελτιστοποιούν τις λύσεις των πρακτόρων στο τέλος κάθε γενιάς. Τελευταίο άλλα και σημαντικότερο, επιλέξαμε την γλώσσα Golang για την ανάπτυξη του κώδικα για δύο λόγους, την ταχύτητά της και την ευκολία παραλληλοποίησης. Παρόλα αυτά, η εφαρμογή μας δεν είναι παραλληλοποιημένη, για λόγους διευκόλυνσης της εκτέλεσης των πειραμάτων και εξαγωγής συμπερασμάτων. Υπάρχει, όμως, η υποδομή στον υπάρχοντα κώδικα, ώστε η συνάρτηση κατασκευής λύσης κάθε πράκτορα να εκτελείται στον δικό της πυρήνα, με αποτέλεσμα να οδηγηθούμε σε ακόμη πιο γρήγορες ταχύτητες.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Chicken swarm optimization	Βελτιστοποίηση μέσω σμήνους κοτών
Archimedes optimization algorithm	αλγόριθμος βελτιστοποίησης του Αρχιμήδη
Particle Swarm Optimization	βελτιστοποίηση μέσω σμήνους σωματιδίων
multi-step CSO	πολύ-βηματική βελτιστοποίηση μέσω σμήνους κοτών
modified CSO	Τροποποιημένη CSO
Improved CSO	Βελτιωμένη CSO
Mutation CSO	Μεταλλαγμένη CSO
Mutation CSO based on non linear inertia weight	Μεταλλαγμένη CSO βασισμένη σε μη γραμμικά αδρανή βάρη
Monomer turbulence and particle renovation based CSO	CSO βασισμένη σε μονομερή αναταραχή και σωματιδιακή ανανέωση
Chaotic CSO	Χαστική CSO
Binary Improved CSO	Διαδική βελτιωμένη CSO
Improved Boundary CSO	CSO με βελτιωμένα όρια
CSO based on elite opposition based learning	CSO βασισμένη στην εκμάθηση μέσω αντιπαλότητας
Adaptive CSO	Προσαρμοστική CSO
Κβαντική CSO	Quantum CSO
Bat Algorithm	αλγόριθμος νυχτερίδων
fractals	μορφοκλάσματα
transfer operator	τελεστής μεταφοράς
density factor	συντελεστής πυκνότητας
airline crew rostering	χρονοπρογραμματισμός πληρωμάτων αεροπορικών εταιρειών
crew rostering	χρονοπρογραμματισμός πληρωμάτων
crew assignment	ανάθεση πληρωμάτων
crew planning	σχεδιασμός πληρωμάτων
crew pairing	συνδυασμός πληρώματος
pairing	συνδυασμός πτήσεων
flight leg	πτήση χωρίς ενδιάμεσες στάσεις
crew complement	συμπλήρωμα πληρώματος
roster	σύνθεση
personalized roster	προσωποποιημένη σύνθεση
qualification-type constraint	περιορισμός σχετιζόμενος με προσόντα
open time	ανοιχτός χρόνος
constructive heuristic	κατασκευαστικός ευρετικός αλγόριθμος
set partitioning problem	πρόβλημα διαχωρισμού συνόλων
covering problem	πρόβλημα κάλυψης
branch-and-bound	διακλάδωση και δέσμευση
branch-and-cut	διακλάδωση και αποκοπή
branch-and-price	διακλάδωση και αποτίμηση
column generation	παραγωγή στηλών
divide-and-conquer	διαίρει και βασίλευε
trip-as-node networks	δίκτυα ταξιδιού ως κόμβος
trip-as-arc networks	δίκτυα ταξιδιού ως ακμή

time-space networks	δίκτυα χώρου και χρόνου
Tabu search	αναζήτηση Tabu
root	ρίζα
ant colony optimization	βελτιστοποίηση μέσω αποικίας μυρμηγκιών

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

CSO	Chicken swarm optimization (βελτιστοποίηση μέσω σμήνους κοτών)
AOA	Archimedes optimization algorithm (αλγόριθμος βελτιστοποίησης του Αρχιμήδη)
PSO	Particle Swarm Optimization (βελτιστοποίηση μέσω σμήνους σωματιδίων)
DE	Differential Evolution
CS	Cuckoo Search
GA	Genetic algorithm
mCSO	modified CSO (Τροποποιημένη CSO)
ICSO	Improved CSO (Βελτιωμένη CSO)
MCSO	Mutation CSO (Μεταλλαγμένη CSO)
NW-MCSO	Mutation CSO based on non linear inertia weight (Μεταλλαγμένη CSO βασισμένη σε μη γραμμικά αδρανή βάρη)
MPCSO	Monomer turbulence and particle renovation based CSO (CSO βασισμένη σε μονομερή αναταραχή και σωματιδιακή ανανέωση)
CCSO	Chaotic CSO (Χαοτική CSO)
BGCSO	Binary Improved CSO (Διαδική βελτιωμένη CSO)
IBCSO	Improved Boundary CSO (CSO με βελτιωμένα όρια)
EOCSO	CSO based on elite opposition based learning (CSO βασισμένη στην εκμάθηση μέσω αντιπαλότητας)
ACSO	Adaptive CSO (Προσαρμοστική CSO)
QCSO	Quantum CSO (Κβαντική CSO)
BA	Bat Algorithm (αλγόριθμος νυχτερίδων)
CS	Cuckoo Search
TLBO	Teaching Learning based Optimization
TF	transfer operator (τελεστής μεταφοράς)

ΠΑΡΑΡΤΗΜΑ Ι: ΥΛΟΠΟΙΗΣΗ

Στα πλαίσια της εργασίας αναπτύχθηκε μία εφαρμογή η οποία εκτελούνταν μέσω ενός τερματικού και αναλόγως των ορισμάτων εφαρμόζε έναν από τους δύο αλγορίθμους πάνω στο πρόβλημα. Στο τέλος της εκτέλεσης της, ελέγχαμε αν η προτεινόμενη λύση υπάκουε στους κανόνες. Το πρόγραμμα των πιλότων μαζί με διάφορα στατιστικά στοιχεία της λύσης καταγράφονταν αυτόματα από την εφαρμογή σε ένα αρχείο excel. Ένα τμήμα του προγράμματος των πιλότων φαίνεται στο σχήμα 25 και το χρησιμοποιούμε για καλύτερη οπτικοποίηση της μορφής του. Η εφαρμογή αναπτύχθηκε σε γλώσσα Golang, καθώς θέλαμε να εκμεταλλευτούμε την ταχύτητά της.

Pilots	November 2011																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	F	F	F	F	F	F				F	F	F	F	F	F		F	F	F	F	F	F			F	F	F	F	F	F
1						F			F		F	F	F	F	F		F	F	F	F	F			F	F	F	F	F		F
2	F	F	F	F	F			F	F	F	F	F	F	F	F	F							F	F	F	F	F	F		F
3		F	F	F	F		F		F	F	F	F	F	F	F	F						F	F	F	F	F	F	F	F	F
4			F	F	F	F	F		F	F	F	F	F	F	F	F						F	F	F	F	F	F	F	F	F
5	F		F	F	F	F	F		F		F	F	F	F	F	F						F	F	F	F	F	F	F	F	F
6		F	F	F	F	F	F		F		F	F	F	F	F	F						F	F	F	F	F	F	F	F	F
7						F			F						F	F		F	F	F	F			F	F	F	F	F		F
8		F	F	F	F	F			F	F	F	F	F	F	F	F						F	F	F	F	F	F	F	F	F
9									F	F	F	F	F	F	F	F						F	F	F	F	F	F	F	F	F
10																														
11	F		F	F	F																									
12		F	F	F	F																									
13		F	F	F	F																									
14	F	F	F	F	F																									
15	F	F	F	F	F																									
16	F	F	F	F	F																									
17																														
18	F		F	F	F																									
19	F	F	F	F	F																									
20		F	F	F	F																									
21		F	F	F	F																									
22	F	F	F	F	F																									
23	F	F	F	F	F																									
24																														
25	F	F	F	F	F																									
26	F	F	F	F	F																									
27	F		F	F	F																									
28	F	F	F	F	F																									
29	F	F	F	F	F																									
30	F	F	F	F	F																									
31	F	F	F	F	F																									
32	F	F	F	F	F																									
33	F	F	F	F	F																									
34	F	F	F	F	F																									
35	F	F	F	F	F																									
36																														
37	F	F	F	F	F																									
38	F	F	F	F	F																									
39	F		F	F	F																									
40	F		F	F	F																									
41	F		F	F	F																									
42	F	F	F	F	F																									
43	F	F	F	F	F																									
44	F	F	F	F	F																									

Σχήμα 25: Πρόγραμμα Πιλότων

Το τελευταίο κομμάτι κώδικα που αναπτύξαμε αφορούσε τα πειράματα. Εδώ χρησιμοποιήσαμε την γλώσσα Python, λόγω της απλότητάς της. Γράψαμε κώδικα ο οποίος μπορούσε εύκολα να παράγει έναν αριθμό για την αρχικοποίηση της γεννήτριας τυχαίων αριθμών, κώδικα που μπορούσε να τρέχει την εφαρμογή μαζικά για αυτές τις αρχικοποιήσεις και για διάφορες τιμές παραμέτρων, κώδικα που μας επιτρέπει να εκτελούμε πολλές φορές την εφαρμογή για συγκεκριμένη παράμετρο και, τέλος, κώδικα που συλλέγει ορισμένα στοιχεία από πολλά αρχεία εξόδου και τα οργανώνει σε έναν πίνακα του Microsoft Word.

ΠΑΡΑΡΤΗΜΑ ΙΙ: ΣΤΑΤΙΣΤΙΚΑ

Σε αυτό το παράρτημα παραθέτουμε όλα τα αποτελέσματα συγκεντρώσαμε για αυτήν την εργασία. Σε κάθε πίνακα έχουμε χρησιμοποιήσει 44 διαφορετικές αρχικοποιήσεις της γεννήτριας τυχαίων αριθμών.

Πίνακας 2: Αποτελέσματα πολύ-βηματικής CSO για 40 πιλότους, FL=1

Χρόνος Εκτέλεσης	Καλύτερη Λύση	Χειρότερη Λύση	Έγκυρες Λύσεις	Μονοπάτια	Ομοιότητα	Άλματα
1 λ. 14.6023755 δ.	116448	691232	5	4000	2.67	8
1 λ. 12.6548750 δ.	140928	723520	5	4000	2.62	4
1 λ. 15.7566895 δ.	104320	643904	4	4000	2.46	4
1 λ. 14.2855758 δ.	115584	722112	4	4000	2.69	5
1 λ. 11.7110790 δ.	128672	706816	4	4000	2.55	6
1 λ. 14.1575969 δ.	94176	775904	3	4000	2.56	7
1 λ. 15.6223841 δ.	112064	700736	3	4000	2.5	6
1 λ. 14.9695252 δ.	121728	715168	3	4000	2.5	8
1 λ. 11.1118889 δ.	127872	704928	3	4000	2.57	5
1 λ. 5.51731060 δ.	134080	719456	3	4000	2.71	2
1 λ. 14.0275861 δ.	139584	644000	3	4000	2.68	7
1 λ. 15.5278909 δ.	142464	725760	3	4000	2.64	5
1 λ. 10.1579927 δ.	150304	657600	3	4000	2.59	7
1 λ. 8.96549560 δ.	180864	701152	3	4000	2.58	7
1 λ. 14.4602361 δ.	69728	720096	2	4000	2.72	4
1 λ. 14.2071288 δ.	96544	704544	2	4000	2.69	4
1 λ. 12.3273960 δ.	119488	736960	2	4000	2.53	2
1 λ. 15.4083030 δ.	148864	702784	2	4000	2.66	9
1 λ. 14.3381171 δ.	154976	715264	2	4000	2.58	6
1 λ. 14.5834263 δ.	58240	730944	1	4000	2.54	10
1 λ. 15.5552358 δ.	84160	706752	1	4000	2.62	7
1 λ. 14.1048774 δ.	97696	685440	1	4000	2.64	6
1 λ. 11.0268938 δ.	100768	792064	1	4000	2.62	2
1 λ. 13.3220547 δ.	104512	790432	1	4000	2.57	9
1 λ. 15.6271114 δ.	104832	738432	1	4000	2.51	7
1 λ. 14.9539379 δ.	108768	786048	1	4000	2.57	6
1 λ. 15.3898834 δ.	110080	682784	1	4000	2.46	3
1 λ. 13.4564699 δ.	110880	672032	1	4000	2.47	9
1 λ. 11.3982754 δ.	114368	677440	1	4000	2.69	6
1 λ. 15.4990677 δ.	114656	764544	1	4000	2.53	8
1 λ. 15.6898337 δ.	115520	766144	1	4000	2.66	6
1 λ. 13.9657933 δ.	117152	742912	1	4000	2.52	7
1 λ. 13.7908747 δ.	122080	687488	1	4000	2.54	6
1 λ. 14.5396534 δ.	122624	690336	1	4000	2.68	7
1 λ. 13.5389674 δ.	124576	817056	1	4000	2.61	11
1 λ. 14.3659645 δ.	128000	722368	1	4000	2.62	3
1 λ. 10.1282918 δ.	129472	678112	1	4000	2.42	3
1 λ. 14.8663726 δ.	134624	785216	1	4000	2.54	11
1 λ. 14.9081180 δ.	138432	761088	1	4000	2.43	5
1 λ. 13.9999981 δ.	163488	686048	1	4000	2.58	6
1 λ. 16.7257935 δ.	169824	710304	1	4000	2.59	11
1 λ. 12.2732506 δ.	64128	712384	0	4000	2.61	7
1 λ. 15.9150569 δ.	130368	671776	0	4000	2.59	8
1 λ. 14.5326505 δ.	136064	691168	0	4000	2.58	6

Πίνακας 3: Αποτελέσματα πολύ-βηματικής CSO για 50 πιλότους, FL=1

Χρόνος Εκτέλεσης	Καλύτερη Λύση	Χειρότερη Λύση	Έγκυρες Λύσεις	Μονοπάτια	Ομοιότητα	Άλματα
1 λ. 28.9342876 δ.	29325	145178	4000	4000	1.89	8
1 λ. 25.0026399 δ.	30502	118618	4000	4000	1.82	8
1 λ. 31.0551795 δ.	30912	154112	4000	4000	1.83	5
1 λ. 32.7613303 δ.	31565	135373	4000	4000	1.88	7
1 λ. 29.6611993 δ.	31565	183770	4000	4000	1.9	5
1 λ. 29.6772098 δ.	31578	146253	4000	4000	1.97	6
1 λ. 29.0340133 δ.	31667	142707	4000	4000	1.97	7
1 λ. 28.5784163 δ.	32000	174848	4000	4000	1.83	6
1 λ. 32.4589819 δ.	32218	148390	4000	4000	1.92	4
1 λ. 28.7011500 δ.	32307	114214	4000	4000	1.84	3
1 λ. 24.2669097 δ.	32525	165862	4000	4000	1.92	4
1 λ. 23.8794888 δ.	32538	211622	4000	4000	1.88	7
1 λ. 30.1739858 δ.	32627	143795	4000	4000	1.93	5
1 λ. 31.8225954 δ.	32627	170662	4000	4000	1.96	4
1 λ. 29.5369620 δ.	32730	178227	4000	4000	1.93	5
1 λ. 25.3202502 δ.	32960	139635	4000	4000	1.82	6
1 λ. 31.3509777 δ.	33050	141645	4000	4000	1.91	4
1 λ. 25.4563343 δ.	33062	202880	4000	4000	1.92	7
1 λ. 29.8560508 δ.	33165	156160	4000	4000	2.01	6
1 λ. 29.3246161 δ.	33178	141005	4000	4000	1.91	5
1 λ. 30.6433155 δ.	33280	235520	4000	4000	1.81	4
1 λ. 33.0603475 δ.	33370	186470	4000	4000	2.01	6
1 λ. 32.0352483 δ.	33370	159437	4000	4000	1.88	5
1 λ. 30.4711417 δ.	33382	121920	4000	4000	1.92	6
1 λ. 24.5977981 δ.	33498	107098	4000	4000	2.01	3
1 λ. 31.2187113 δ.	33498	170790	4000	4000	1.96	9
1 λ. 30.7562381 δ.	33600	156992	4000	4000	1.88	6
1 λ. 29.4541487 δ.	33600	153805	4000	4000	1.87	11
1 λ. 32.6143099 δ.	33702	126272	4000	4000	1.92	9
1 λ. 24.5345825 δ.	33702	175475	4000	4000	2.03	5
1 λ. 28.9330041 δ.	33805	225382	4000	4000	1.92	5
1 λ. 30.6683052 δ.	33805	148493	4000	4000	1.87	5
1 λ. 29.2716486 δ.	33818	203725	4000	4000	1.87	7
1 λ. 28.9899773 δ.	33907	126835	4000	4000	1.88	5
1 λ. 31.6922591 δ.	33920	178867	4000	4000	1.89	5
1 λ. 29.6663636 δ.	34022	187187	4000	4000	1.83	3
1 λ. 29.8498996 δ.	34125	160883	4000	4000	1.88	7
1 λ. 29.9962322 δ.	34150	142067	4000	4000	1.83	6
1 λ. 30.5279930 δ.	34227	160102	4000	4000	1.99	9
1 λ. 30.3548294 δ.	34342	143795	4000	4000	1.85	4
1 λ. 31.9814431 δ.	34342	148173	4000	4000	1.9	9
1 λ. 30.2914520 δ.	34432	144346	4000	4000	1.82	5
1 λ. 30.2671433 δ.	34458	183565	4000	4000	1.9	7
1 λ. 24.7415290 δ.	34560	137280	4000	4000	1.98	3

**Πίνακας 4: Αποτελέσματα βελτιστοποίησης του Αρχιμήδη για 40 πιλότους,
C1=1, C2=2, C3=2, C4=0.5**

Χρόνος Εκτέλεσης	Καλύτερη Λύση	Χειρότερη Λύση	Έγκυρες λύσεις	Μονοπάτια	Ομοιότητα	Άλματα
1 λ. 22.8808133 δ.	160896	810144	135	4000	2.68	13
1 λ. 23.5340412 δ.	137920	967392	88	4000	2.6	14
1 λ. 21.3835582 δ.	93440	882592	86	4000	2.62	9
1 λ. 27.4147097 δ.	235392	858752	63	4000	2.64	11
1 λ. 19.3546826 δ.	89472	800384	52	4000	2.56	6
1 λ. 25.1376276 δ.	85120	944000	41	4000	2.66	8
1 λ. 20.0145572 δ.	139520	913760	41	4000	2.65	7
1 λ. 29.0488006 δ.	209728	863264	38	4000	2.58	9
1 λ. 24.1729092 δ.	189184	822560	36	4000	2.51	6
1 λ. 21.0246642 δ.	122432	838816	28	4000	2.54	9
1 λ. 20.2808505 δ.	131904	785472	25	4000	2.6	11
1 λ. 25.4021510 δ.	166912	839936	24	4000	2.6	10
1 λ. 24.6218295 δ.	244512	842912	24	4000	2.51	5
1 λ. 25.4784961 δ.	206784	874720	23	4000	2.58	4
1 λ. 20.3719250 δ.	149824	864544	22	4000	2.57	5
1 λ. 19.4436819 δ.	152832	935456	21	4000	2.53	8
1 λ. 26.9006743 δ.	202400	873984	18	4000	2.6	6
1 λ. 20.3931803 δ.	197312	814272	14	4000	2.62	9
1 λ. 20.5135564 δ.	138816	956928	12	4000	2.54	8
1 λ. 19.2455768 δ.	171264	994336	11	4000	2.55	4
1 λ. 22.4833222 δ.	179904	823968	11	4000	2.66	10
1 λ. 21.3729766 δ.	218080	788096	11	4000	2.57	7
1 λ. 19.4424312 δ.	131840	826304	10	4000	2.57	3
1 λ. 20.5902528 δ.	135584	802336	9	4000	2.48	5
1 λ. 23.6267711 δ.	281024	895808	9	4000	2.56	9
1 λ. 18.1772680 δ.	147680	984160	8	4000	2.6	8
1 λ. 17.8557357 δ.	70080	970432	6	4000	2.55	5
1 λ. 18.0022671 δ.	124288	862944	6	4000	2.56	12
1 λ. 17.7790418 δ.	158400	808448	5	4000	2.48	6
1 λ. 13.2181503 δ.	134048	829056	4	4000	2.6	4
1 λ. 18.9050500 δ.	168192	917312	4	4000	2.55	8
1 λ. 16.4818361 δ.	172288	1060000	4	4000	2.5	9
1 λ. 16.8450499 δ.	133504	760416	3	4000	2.64	4
1 λ. 16.7838094 δ.	134432	768544	2	4000	2.63	5
1 λ. 17.5118345 δ.	145696	850912	2	4000	2.58	6
1 λ. 21.7288484 δ.	22051360	22814400	2	4000	2.56	6
1 λ. 18.0827369 δ.	84416	22743520	1	4000	2.12	7
1 λ. 19.1170174 δ.	112064	22817760	1	4000	0.8	6
1 λ. 21.7846143 δ.	163488	22915520	1	4000	0.81	6
1 λ. 24.2541505 δ.	21894720	22911040	1	4000	4.72	6
1 λ. 22.4368071 δ.	86400	22809120	0	4000	0.81	6
1 λ. 23.8669877 δ.	101696	22945280	0	4000	0.8	6
1 λ. 22.6304889 δ.	118336	22796480	0	4000	0.81	6
1 λ. 18.6923982 δ.	141152	874752	0	4000	2.47	6

**Πίνακας 5: Αποτελέσματα βελτιστοποίησης του Αρχιμήδη για 50 πιλότους,
C1=1, C2=2, C3=2, C4=0.5**

Χρόνος Εκτέλεσης	Καλύτερη Λύση	Χειρότερη Λύση	Έγκυρες λύσεις	Μονοπάτια	Ομοιότητα	Άλματα
1 λ. 31.3617432 δ.	30298	180378	4000	4000	1.78	7
1 λ. 25.3883811 δ.	31142	163942	4000	4000	1.76	8
1 λ. 33.2892258 δ.	31578	165235	4000	4000	1.82	8
1 λ. 29.8782002 δ.	31680	141248	4000	4000	1.81	7
1 λ. 30.4294204 δ.	31885	160666	4000	4000	1.83	5
1 λ. 35.847366 δ.	31987	181952	4000	4000	1.85	4
1 λ. 27.8418715 δ.	32000	203520	4000	4000	1.87	6
1 λ. 33.4075889 δ.	32192	166093	4000	4000	1.83	5
1 λ. 33.7165827 δ.	32205	189338	4000	4000	1.84	5
1 λ. 28.4964325 δ.	32307	176858	4000	4000	1.79	5
1 λ. 31.5342894 δ.	32525	148378	4000	4000	1.9	4
1 λ. 32.1677482 δ.	32538	163098	4000	4000	2.02	3
1 λ. 39.9697966 δ.	32730	23576000	1360	4000	0.65	7
1 λ. 31.8830436 δ.	32742	239027	4000	4000	2.04	3
1 λ. 38.4501744 δ.	32845	273306	4000	4000	1.82	5
1 λ. 32.8349927 δ.	32858	221466	4000	4000	1.88	5
1 λ. 25.9415242 δ.	32960	242138	4000	4000	1.82	5
1 λ. 31.6901114 δ.	33062	146458	4000	4000	1.93	4
1 λ. 33.3149892 δ.	33152	156595	4000	4000	1.89	6
1 λ. 32.9671676 δ.	33165	142387	4000	4000	1.93	5
1 λ. 32.582585 δ.	33267	227520	4000	4000	1.87	5
1 λ. 32.1758119 δ.	33280	154790	4000	4000	1.96	3
1 λ. 34.3340323 δ.	33485	164698	4000	4000	1.89	7
1 λ. 34.4024955 δ.	33587	158080	4000	4000	1.96	7
1 λ. 26.1825792 δ.	33690	147750	4000	4000	1.98	4
1 λ. 34.2295003 δ.	33907	194675	4000	4000	1.83	10
1 λ. 39.2483437 δ.	33920	23436480	1260	4000	0.61	6
1 λ. 32.121988 δ.	33920	172378	4000	4000	1.89	3
1 λ. 30.338399 δ.	34022	197747	3999	4000	1.78	8
1 λ. 34.4913847 δ.	34022	198195	4000	4000	1.9	7
1 λ. 32.467248 δ.	34022	139110	4000	4000	1.81	3
1 λ. 33.2285692 δ.	34125	145933	4000	4000	1.91	8
1 λ. 39.5735488 δ.	34227	23613440	1260	4000	0.56	6
1 λ. 34.5426322 δ.	34342	180390	4000	4000	1.79	5
1 λ. 32.2075131 δ.	34355	198912	4000	4000	2.01	5
1 λ. 35.0620427 δ.	34445	241715	4000	4000	1.81	7
1 λ. 33.179121 δ.	34458	207667	4000	4000	1.83	6
1 λ. 32.3728265 δ.	34547	182208	4000	4000	1.81	5
1 λ. 31.7059773 δ.	34560	190515	4000	4000	1.81	7
1 λ. 30.1852969 δ.	34662	156698	4000	4000	1.98	8
1 λ. 26.2747451 δ.	34662	179738	4000	4000	1.83	8
1 λ. 33.9355024 δ.	34995	231821	4000	4000	1.86	5
1 λ. 33.8440619 δ.	36045	257843	3998	4000	1.85	5
1 λ. 33.0452903 δ.	36800	168218	4000	4000	1.98	2

Πίνακας 6: Αποτελέσματα πολύ-βηματικής CSO για 45 πιλότους

Χρόνος Εκτέλεσης	FL	Καλύτερη Λύση	Χειρότερη Λύση	Έγκυρες Λύσεις	Μονοπάτια	Ομοιότητα	Άλματα
1 λ. 13.9478132 δ.	1.4	24668	618428	2029	4000	2.03	6
1 λ. 21.0511853 δ.	1.5	26731	608690	2105	4000	2.16	8
1 λ. 15.4054883 δ.	1.3	26731	640270	2393	4000	2.03	7
1 λ. 15.5549661 δ.	1.7	26830	572629	2103	4000	1.99	4
1 λ. 16.6955903 δ.	1.9	26930	683790	1893	4000	2.25	7
1 λ. 16.3169232 δ.	0.9	26930	283371	3015	4000	2.11	3
1 λ. 23.3147343 δ.	2	27008	623250	2292	4000	2.09	8
1 λ. 23.2664773 δ.	1.9	27129	561490	1998	4000	2.08	4
1 λ. 36.8190607 δ.	0.7	27129	974507	2466	4000	1.95	3
1 λ. 21.791748 δ.	1.6	27250	642212	2068	4000	2.01	5
1 λ. 13.1793299 δ.	2	27271	601550	2049	4000	2.11	5
1 λ. 14.2345715 δ.	1.5	27371	601849	1921	4000	2.07	7
1 λ. 21.8510793 δ.	1.5	27470	593472	2516	4000	2.11	2
1 λ. 21.2288152 δ.	1.1	27669	680572	1562	4000	2.0	8
1 λ. 24.2211931 δ.	1.7	27669	639708	2542	4000	2.08	10
1 λ. 13.094208 δ.	1.9	27669	642869	1695	4000	1.97	4
1 λ. 16.3711293 δ.	1.6	27769	669550	2653	4000	1.97	4
1 λ. 21.7797079 δ.	1.2	27769	596668	2193	4000	1.92	7
1 λ. 16.5613782 δ.	1.4	27812	600512	2687	4000	2.18	6
1 λ. 26.9927875 δ.	1.9	27868	670229	2374	4000	2.1	3
1 λ. 17.3624796 δ.	0.8	27890	435470	2633	4000	2.19	6
1 λ. 26.4189433 δ.	1.2	27890	599989	2953	4000	1.95	3
1 λ. 15.1915633 δ.	1.2	27911	622692	2952	4000	2.06	5
1 λ. 16.4834724 δ.	1.6	27968	619008	2246	4000	2.05	7
1 λ. 13.4443518 δ.	1.8	27989	636331	2181	4000	2.01	5
1 λ. 16.2537669 δ.	1.6	28011	677671	2474	4000	2.21	4
1 λ. 22.4367558 δ.	1.4	28068	648149	2527	4000	2.19	6
1 λ. 12.4998577 δ.	1.5	28089	611449	2168	4000	2.16	7
1 λ. 11.4046107 δ.	1	28089	240668	2991	4000	2.35	5
1 λ. 14.7949366 δ.	1.5	28089	679652	2299	4000	2.21	6
1 λ. 17.4672263 δ.	1	28110	295950	3002	4000	2.21	5
1 λ. 11.5321188 δ.	0.8	28210	459890	2772	4000	2.3	5
1 λ. 22.3488861 δ.	0.6	28210	713472	2904	4000	2.0	6
1 λ. 15.0074272 δ.	1.8	28210	710212	2685	4000	2.06	5
1 λ. 28.2904078 δ.	0.8	28288	686549	3107	4000	1.82	4
1 λ. 18.7903051 δ.	1.1	28309	603367	2690	4000	2.05	6
1 λ. 14.4282942 δ.	1.6	28309	619911	2165	4000	2.08	7
1 λ. 14.5618194 δ.	1.4	28309	631509	2677	4000	1.97	4
1 λ. 22.5237164 δ.	0.6	28309	821390	3210	4000	1.87	4
1 λ. 14.4631754 δ.	0.9	28409	256626	3007	4000	2.26	6
1 λ. 23.5382884 δ.	1.3	28409	634389	2002	4000	2.16	7
1 λ. 26.7166778 δ.	1.8	28430	585849	2873	4000	2.09	8
1 λ. 18.1759059 δ.	0	28430	614272	2331	4000	1.95	6
1 λ. 16.5683328 δ.	1.5	28430	650510	2521	4000	1.98	7
1 λ. 22.2609179 δ.	1.4	28452	629131	2586	4000	1.95	4
1 λ. 13.357183 δ.	1.2	28508	644188	2609	4000	2.2	8
1 λ. 11.7185044 δ.	0.8	28530	287332	2333	4000	2.22	7
1 λ. 22.6435696 δ.	1.3	28530	667449	1918	4000	2.03	2
1 λ. 21.6399392 δ.	0.1	28530	616331	2691	4000	1.97	5
1 λ. 17.3163138 δ.	1	28629	286108	3034	4000	2.24	5
1 λ. 17.7442421 δ.	1.5	28629	602830	1896	4000	2.03	5

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 14.5304764 δ.	0.1	28629	632551	2480	4000	2.06	5
1 λ. 18.3409421 δ.	0.4	28708	615068	2424	4000	1.99	6
1 λ. 12.4968855 δ.	1	28828	226229	3012	4000	2.3	5
1 λ. 17.9759161 δ.	1.6	28828	661589	2201	4000	2.06	3
1 λ. 15.2363585 δ.	0.3	28828	614350	3122	4000	1.97	4
1 λ. 10.5887594 δ.	0.9	28850	245312	2889	4000	2.27	6
1 λ. 22.5138388 δ.	0.8	28850	600210	2780	4000	2.15	3
1 λ. 18.9167566 δ.	1	28871	222471	2931	4000	2.27	5
1 λ. 21.9464965 δ.	1.1	28928	627570	2830	4000	1.98	10
1 λ. 28.3668864 δ.	0.1	28928	716629	2730	4000	1.91	4
1 λ. 10.2848582 δ.	0.9	28949	224910	2715	4000	2.06	5
1 λ. 25.367726 δ.	1.5	28949	618549	2379	4000	2.05	6
1 λ. 16.7897444 δ.	1.6	28949	649749	3098	4000	1.97	6
1 λ. 20.3308991 δ.	0.7	28971	626830	2931	4000	2.03	4
1 λ. 22.9824481 δ.	2	28971	668011	2349	4000	2.19	9
1 λ. 14.8388883 δ.	2	28971	624633	2067	4000	2.05	3
1 λ. 23.652598 δ.	0.4	28971	789412	2597	4000	1.98	3
1 λ. 13.9695933 δ.	1.5	29028	628651	1454	4000	2.04	3
1 λ. 11.4972914 δ.	1	29049	230471	2993	4000	2.25	7
1 λ. 15.579447 δ.	1.4	29049	590791	2515	4000	2.07	5
1 λ. 24.6592508 δ.	1.4	29049	609173	2455	4000	2.09	4
1 λ. 18.9583286 δ.	0.6	29049	789568	1943	4000	1.87	2
1 λ. 16.3744076 δ.	0.9	29070	323328	3053	4000	2.18	4
1 λ. 12.0998291 δ.	1	29070	217628	3039	4000	2.18	6
1 λ. 22.4843415 δ.	0.9	29070	347570	3135	4000	2.12	6
1 λ. 17.6516463 δ.	0	29070	597931	2566	4000	2.02	5
1 λ. 23.8318022 δ.	1.5	29092	636270	2755	4000	1.92	6
1 λ. 15.1351481 δ.	0.8	29092	349468	2753	4000	2.17	7
1 λ. 14.5099082 δ.	0.9	29148	413689	3038	4000	2.34	6
1 λ. 28.6214146 δ.	0.3	29148	760370	2337	4000	1.9	2
1 λ. 21.7550722 δ.	0	29148	592171	2390	4000	1.98	4
1 λ. 15.1551873 δ.	1.6	29148	570731	2200	4000	2.09	3
1 λ. 21.6234914 δ.	1.4	29170	591090	2341	4000	1.98	5
1 λ. 26.7743046 δ.	0.3	29170	641991	2504	4000	1.95	6
1 λ. 19.6455134 δ.	1.7	29170	609049	1933	4000	2.16	8
1 λ. 14.4633426 δ.	0.8	29170	436068	2966	4000	2.27	4
1 λ. 24.0791387 δ.	0	29170	620331	2710	4000	2.08	9
1 λ. 15.7476847 δ.	0.1	29191	606628	2865	4000	2.04	5
1 λ. 14.9601122 δ.	0.1	29191	594791	2484	4000	2.04	4
1 λ. 13.4589424 δ.	1.9	29248	609330	2485	4000	2.01	4
1 λ. 21.2041482 δ.	1.8	29269	615829	2506	4000	2.07	7
1 λ. 22.7306461 δ.	0.9	29269	513191	3092	4000	2.24	4
1 λ. 18.5391211 δ.	0.6	29269	688992	2621	4000	1.99	5
1 λ. 21.5567943 δ.	1.6	29269	556292	2770	4000	2.0	5
1 λ. 28.4825981 δ.	1.8	29291	646770	2017	4000	1.97	4
1 λ. 20.1862447 δ.	1.8	29291	613753	1519	4000	2.16	7
1 λ. 23.4338359 δ.	1.2	29369	640249	2659	4000	2.07	6
1 λ. 13.1784858 δ.	1.7	29369	649230	1903	4000	2.1	9
1 λ. 11.8120597 δ.	1	29369	229209	2987	4000	2.17	3
1 λ. 12.4057458 δ.	1	29369	255289	3023	4000	2.22	2
1 λ. 23.5091342 δ.	0	29369	628171	2255	4000	2.05	4
1 λ. 12.9230134 δ.	1.6	29390	585312	1986	4000	2.15	6
1 λ. 12.9230134 δ.	1.6	29390	585312	1986	4000	2.15	6
1 λ. 26.9906715 δ.	0.8	29390	660430	2900	4000	2.11	3
1 λ. 19.1961241 δ.	1.4	29390	671989	1678	4000	2.09	4
1 λ. 22.5905484 δ.	1.2	29390	583090	3093	4000	2.0	9

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 27.0591278 δ.	1.2	29390	607328	2696	4000	1.99	7
1 λ. 18.7902136 δ.	0.3	29412	630791	2900	4000	2.02	5
1 λ. 15.5075146 δ.	1.7	29468	654108	1434	4000	2.2	6
1 λ. 20.490204 δ.	1.4	29468	640448	2712	4000	2.05	7
1 λ. 14.8055416 δ.	0.8	29490	660892	2757	4000	2.18	6
1 λ. 22.5049275 δ.	0.4	29490	688647	3065	4000	1.94	5
1 λ. 15.8930365 δ.	0	29490	598571	2646	4000	1.96	6
1 λ. 19.6021408 δ.	1.3	29490	599950	1996	4000	2.2	7
1 λ. 12.6967764 δ.	1.7	29490	595790	1485	4000	2.17	6
1 λ. 15.6226491 δ.	1.5	29490	605170	2172	4000	2.1	5
1 λ. 15.8944764 δ.	1.3	29490	687250	2489	4000	2.04	4
1 λ. 15.8515416 δ.	1.5	29490	703708	2206	4000	2.0	3
1 λ. 15.0967384 δ.	0.9	29511	257529	3027	4000	2.14	7
1 λ. 11.9762635 δ.	0.9	29511	303908	2991	4000	2.15	2
1 λ. 11.4361153 δ.	0.7	29511	507989	2437	4000	2.21	5
1 λ. 15.269519 δ.	1.2	29511	623250	1906	4000	1.91	2
1 λ. 20.3613206 δ.	1.6	29511	625550	2282	4000	2.14	7
1 λ. 23.3167095 δ.	0.7	29589	712569	2845	4000	1.95	6
1 λ. 20.5746895 δ.	0.8	29589	542670	3015	4000	2.17	7
1 λ. 23.7949915 δ.	1.7	29589	616530	2576	4000	2.04	4
1 λ. 13.5925481 δ.	1.4	29589	621550	2569	4000	2.1	6
1 λ. 10.631008 δ.	0.8	29589	462052	1928	4000	2.28	7
1 λ. 14.2142036 δ.	0.8	29589	489628	2892	4000	1.99	4
1 λ. 24.9123454 δ.	1.4	29589	602510	2097	4000	2.19	6
1 λ. 13.3791852 δ.	1.8	29589	616789	2200	4000	2.08	5
1 λ. 20.3452714 δ.	2	29589	624491	1793	4000	2.17	3
1 λ. 19.592116 δ.	1.8	29589	676850	2275	4000	2.08	6
1 λ. 25.3174373 δ.	0.4	29589	712348	3069	4000	1.98	4
1 λ. 15.1465061 δ.	1.3	29611	587431	2632	4000	2.1	7
1 λ. 22.6365366 δ.	0.2	29611	591268	2916	4000	2.02	6
1 λ. 15.9561493 δ.	1.3	29689	626830	2207	4000	2.1	4
1 λ. 18.9760497 δ.	0.9	29689	251250	2681	4000	2.28	6
1 λ. 12.1965471 δ.	1	29689	251790	2998	4000	2.21	5
1 λ. 23.8385998 δ.	1.5	29689	549191	2901	4000	1.96	6
1 λ. 29.7561022 δ.	1.5	29689	603332	2588	4000	1.99	7
1 λ. 23.7963181 δ.	1.4	29689	610108	1736	4000	2.04	4
1 λ. 23.5737045 δ.	0	29689	620750	2211	4000	2.07	3
1 λ. 21.0370833 δ.	0.6	29689	761451	2792	4000	2.01	4
1 λ. 16.3426167 δ.	0	29710	574770	2718	4000	2.07	7
1 λ. 13.3828277 δ.	1.4	29710	606030	2138	4000	2.18	5
1 λ. 12.6462665 δ.	1.9	29710	608270	2037	4000	2.09	4
1 λ. 13.4882087 δ.	1.9	29710	617771	1918	4000	2.15	5
1 λ. 18.9143848 δ.	1.6	29710	625550	2649	4000	2.14	2
1 λ. 12.9606384 δ.	1.7	29710	667669	1529	4000	1.99	6
1 λ. 20.1402506 δ.	2	29710	562212	1925	4000	2.0	6
1 λ. 19.1364908 δ.	0.2	29710	588469	2731	4000	2.03	4
1 λ. 21.1020929 δ.	0.3	29710	602951	2872	4000	2.02	4
1 λ. 16.7471643 δ.	0	29788	598528	2576	4000	1.97	4
1 λ. 16.7471643 δ.	0	29788	598528	2576	4000	1.97	4
1 λ. 22.4998609 δ.	1.3	29788	706268	2901	4000	1.98	5
1 λ. 24.2138392 δ.	1.5	29788	709710	2256	4000	2.12	5
1 λ. 14.0811381 δ.	1.8	29810	617550	2657	4000	2.02	3
1 λ. 14.0811381 δ.	1.8	29810	617550	2657	4000	2.02	3
1 λ. 18.58144 δ.	1.5	29810	629429	2728	4000	2.03	6
1 λ. 29.4261691 δ.	0.2	29810	585269	2719	4000	1.98	7
1 λ. 21.6744788 δ.	1.3	29810	562830	2614	4000	1.97	7

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 13.5953413 δ.	1.9	29810	695150	2574	4000	2.14	4
1 λ. 17.1420541 δ.	1.3	29810	559630	2338	4000	2.05	4
1 λ. 13.1220364 δ.	2	29810	591950	1854	4000	2.08	7
1 λ. 16.4796844 δ.	0.7	29810	599189	2784	4000	2.3	5
1 λ. 27.5705297 δ.	1.1	29810	676309	2348	4000	2.08	5
1 λ. 17.2823855 δ.	1.8	29831	574869	2806	4000	2.08	5
1 λ. 19.9932262 δ.	1.8	29888	725269	1771	4000	2.16	5
1 λ. 12.9034744 δ.	1	29888	221049	2998	4000	2.35	6
1 λ. 27.9119149 δ.	0.2	29888	652032	2746	4000	2.14	4
1 λ. 16.267577 δ.	0.9	29909	303211	3091	4000	2.26	9
1 λ. 22.4323127 δ.	1.9	29909	557888	2468	4000	2.19	8
1 λ. 15.724526 δ.	0.2	29909	583812	2342	4000	2.09	5
1 λ. 21.6250642 δ.	1.2	29909	588910	1531	4000	2.01	5
1 λ. 25.0700485 δ.	1.7	29909	630670	2169	4000	2.18	8
1 λ. 14.4360153 δ.	1.2	29909	670990	1937	4000	2.07	3
1 λ. 26.0017674 δ.	1.2	29931	569070	3053	4000	1.91	4
1 λ. 17.1413966 δ.	0	29931	683890	2154	4000	2.12	2
1 λ. 26.4749749 δ.	1.2	29931	575929	2792	4000	2.0	5
1 λ. 14.2917997 δ.	1	29931	229269	3012	4000	2.16	8
1 λ. 13.6235688 δ.	1	29931	259271	3047	4000	2.25	8
1 λ. 11.3216736 δ.	0.8	29931	545291	2623	4000	2.25	3
1 λ. 14.8838624 δ.	1.5	29931	645269	2504	4000	2.09	7
1 λ. 23.7835223 δ.	1.3	29931	646130	2472	4000	1.96	5
1 λ. 12.4052633 δ.	1.8	29988	592171	2206	4000	2.18	7
1 λ. 15.6571793 δ.	0.1	30009	580508	2776	4000	1.99	5
1 λ. 15.6571793 δ.	0.1	30009	580508	2776	4000	1.99	5
1 λ. 19.3872878 δ.	0.2	30009	622471	2936	4000	1.96	3
1 λ. 19.3872878 δ.	0.2	30009	622471	2936	4000	1.96	3
1 λ. 26.3697118 δ.	0.6	30009	787812	2778	4000	1.91	5
1 λ. 11.4233034 δ.	0.7	30009	540430	2942	4000	2.27	7
1 λ. 14.3507273 δ.	1	30009	277767	2997	4000	2.37	7
1 λ. 18.4718559 δ.	1.2	30009	624128	2240	4000	2.02	4
1 λ. 15.209116 δ.	1.5	30009	661888	2303	4000	2.21	7
1 λ. 18.4598145 δ.	0.9	30009	268469	2622	4000	2.15	6
1 λ. 26.4187071 δ.	0	30009	660807	2684	4000	2.05	5
1 λ. 22.7102119 δ.	1.2	30009	676530	2374	4000	2.04	12
1 λ. 14.35821 δ.	1.8	30009	690510	2282	4000	2.16	5
1 λ. 22.3686272 δ.	1.1	30009	699250	1960	4000	1.92	6
1 λ. 12.8708106 δ.	1	30030	258091	3011	4000	2.21	5
1 λ. 18.0225367 δ.	0	30030	607630	2581	4000	2.0	5
1 λ. 15.9994063 δ.	1	30030	280569	3018	4000	2.18	7
1 λ. 28.0571228 δ.	0.7	30030	707890	3038	4000	1.99	4
1 λ. 19.2050686 δ.	0	30052	640988	2614	4000	1.92	3
1 λ. 12.5446586 δ.	0.6	30108	526812	2359	4000	2.26	5
1 λ. 20.724248 δ.	1.4	30108	596309	2322	4000	2.05	2
1 λ. 14.3909938 δ.	1.5	30108	658091	1852	4000	2.1	7
1 λ. 20.266991 δ.	1.9	30108	649372	2571	4000	1.91	3
1 λ. 17.4468411 δ.	1.8	30108	566052	2029	4000	2.11	5
1 λ. 23.0525605 δ.	1.1	30108	623150	3118	4000	2.05	8
1 λ. 15.3440509 δ.	1.8	30108	537831	2664	4000	2.04	8
1 λ. 26.7165977 δ.	2	30108	610830	2391	4000	2.13	3
1 λ. 15.3860675 δ.	1.3	30108	673870	2422	4000	2.13	3
1 λ. 18.1696449 δ.	1.9	30108	677308	1866	4000	2.07	6
1 λ. 27.1893056 δ.	0.2	30108	694770	2750	4000	1.98	6
1 λ. 13.9185906 δ.	0.9	30130	224512	2721	4000	2.22	5
1 λ. 10.5236873 δ.	0.9	30130	249252	2572	4000	2.2	8

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 15.777528 δ.	0.9	30130	272348	3003	4000	2.15	2
1 λ. 12.2320833 δ.	1	30130	316708	2967	4000	2.31	4
1 λ. 23.6587278 δ.	0.1	30130	606471	2837	4000	1.95	6
1 λ. 14.1203903 δ.	1.8	30130	626930	1512	4000	2.06	4
1 λ. 15.8665572 δ.	1.3	30130	649010	2295	4000	2.0	4
1 λ. 20.149532 δ.	2	30130	651929	2046	4000	2.21	7
1 λ. 15.7214602 δ.	1.3	30130	669689	2766	4000	2.1	7
1 λ. 26.8710202 δ.	1.1	30130	694389	2834	4000	2.06	5
1 λ. 23.1426059 δ.	0.3	30130	733589	3106	4000	2.03	4
1 λ. 19.3905563 δ.	1.3	30151	591509	2090	4000	2.01	4
1 λ. 15.9939966 δ.	1.7	30151	643492	2528	4000	2.17	6
1 λ. 21.0477459 δ.	1.5	30151	601870	2544	4000	2.09	6
1 λ. 16.3512741 δ.	1.1	30151	572530	2775	4000	2.01	3
1 λ. 15.6799159 δ.	1.8	30208	614670	2429	4000	2.09	4
1 λ. 21.1849219 δ.	1.5	30208	583730	1392	4000	2.17	4
1 λ. 20.8674669 δ.	1.5	30208	567431	1490	4000	2.1	7
1 λ. 27.1182431 δ.	1.7	30208	664171	2246	4000	2.01	4
1 λ. 22.850207 δ.	0.1	30229	635691	2582	4000	1.95	6
1 λ. 13.8488919 δ.	1.8	30229	663808	2238	4000	2.06	5
1 λ. 18.8708304 δ.	1	30229	258190	2998	4000	2.23	5
1 λ. 23.7850753 δ.	0.2	30229	658510	2480	4000	2.01	8
1 λ. 20.9262216 δ.	0.1	30229	526969	2639	4000	1.95	4
1 λ. 24.1036054 δ.	1	30229	217148	2991	4000	2.2	5
1 λ. 22.194402 δ.	0.9	30229	363890	2960	4000	2.25	6
1 λ. 15.9776248 δ.	0.9	30229	481308	3069	4000	2.16	7
1 λ. 15.9534297 δ.	1.7	30229	596370	2618	4000	1.92	5
1 λ. 13.8626311 δ.	1.7	30229	629931	1595	4000	2.08	4
1 λ. 21.7443626 δ.	1.7	30229	633550	2499	4000	1.93	4
1 λ. 23.5172706 δ.	0.2	30251	601671	2657	4000	2.02	5
1 λ. 22.0349898 δ.	1.4	30251	571271	2260	4000	2.05	5
1 λ. 22.1361297 δ.	1.5	30251	619989	2689	4000	2.05	4
1 λ. 20.6676612 δ.	0.8	30251	517650	2887	4000	2.14	5
1 λ. 20.17921 δ.	0.8	30251	560910	3100	4000	2.02	5
1 λ. 28.1964123 δ.	0.9	30251	249408	3012	4000	2.29	3
1 λ. 16.9123257 δ.	0.9	30329	317269	2847	4000	2.18	2
1 λ. 22.5884611 δ.	1.8	30329	625849	1729	4000	2.31	4
1 λ. 28.6345162 δ.	0.2	30329	606208	2692	4000	1.97	7
1 λ. 17.9909629 δ.	1	30329	241010	2989	4000	2.19	5
1 λ. 21.7458163 δ.	1.5	30329	627851	1796	4000	2.04	5
1 λ. 13.9936814 δ.	1	30329	237028	2958	4000	2.17	8
1 λ. 27.5831374 δ.	0.7	30329	454649	2738	4000	2.27	5
1 λ. 24.4787434 δ.	1.3	30329	620928	2402	4000	2.05	6
1 λ. 19.881641 δ.	2	30329	632128	1981	4000	2.24	5
1 λ. 12.3385956 δ.	1.2	30350	547250	2007	4000	2.09	4
1 λ. 20.4932476 δ.	1.2	30350	799872	2940	4000	1.99	7
1 λ. 26.0171357 δ.	0.2	30350	607349	2721	4000	2.05	6
1 λ. 14.7417631 δ.	2	30350	624348	2212	4000	2.23	5
1 λ. 17.1871355 δ.	1.6	30350	540231	2850	4000	2.13	7
1 λ. 15.3342069 δ.	1.9	30350	591232	2262	4000	2.15	5
1 λ. 22.7042713 δ.	1.8	30350	606428	2284	4000	2.14	5
1 λ. 28.783041 δ.	0	30350	660971	2769	4000	2.04	5
1 λ. 15.1714244 δ.	1.7	30350	702930	2174	4000	2.18	3
1 λ. 23.2129556 δ.	1.6	30372	619570	2100	4000	2.11	6
1 λ. 34.5494198 δ.	1.2	30372	663108	2881	4000	2.06	7
1 λ. 12.1248808 δ.	0.9	30428	222549	2822	4000	2.26	6
1 λ. 15.5269246 δ.	1.4	30428	596629	3048	4000	2.09	7

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 30.2601197 δ.	1.6	30428	745490	2655	4000	1.97	5
1 λ. 17.6838345 δ.	0.5	30428	786268	2629	4000	1.84	4
1 λ. 16.8191316 δ.	1	30450	277611	3005	4000	2.18	1
1 λ. 15.9900125 δ.	0.7	30450	413031	1426	4000	2.22	1
1 λ. 20.5289362 δ.	1.8	30450	581049	2209	4000	1.97	1
1 λ. 20.3865328 δ.	1.9	30450	584910	2768	4000	1.96	1
1 λ. 25.1095967 δ.	1.2	30450	591289	3507	4000	1.9	1
1 λ. 21.1203436 δ.	1.1	30450	593490	2875	4000	1.98	1
1 λ. 28.2997043 δ.	0.3	30450	615531	3072	4000	1.89	1
1 λ. 17.020498 δ.	1.3	30450	633170	2376	4000	2.0	8
1 λ. 20.5187693 δ.	1.7	30450	635271	2087	4000	1.98	1
1 λ. 20.0319153 δ.	2	30450	646933	1942	4000	1.96	1
1 λ. 27.3518752 δ.	0.1	30450	654912	3262	4000	1.93	1
1 λ. 11.8417387 δ.	1.6	30450	657248	1458	4000	2.2	4
1 λ. 25.9368752 δ.	0.5	30450	659890	2911	4000	1.9	1
1 λ. 24.3342886 δ.	0.8	30450	700572	2782	4000	2.04	1
1 λ. 26.9954418 δ.	0.2	30450	703332	2564	4000	1.87	1
1 λ. 24.165593 δ.	0.6	30450	708508	2901	4000	2.03	1
1 λ. 28.8603254 δ.	0.4	30450	797248	2662	4000	1.76	1
1 λ. 21.1843659 δ.	1.6	30450	590752	1561	4000	2.24	3
1 λ. 18.3231995 δ.	1	30450	228708	2986	4000	2.2	8
1 λ. 22.6336101 δ.	1.3	30450	615090	1868	4000	2.01	4
1 λ. 12.4888248 δ.	1.8	30450	677568	2148	4000	2.14	2
1 λ. 19.9810117 δ.	0.2	30450	637070	3190	4000	2.0	5
1 λ. 11.8598643 δ.	1	30450	233429	2970	4000	2.19	5
1 λ. 20.5816882 δ.	0.9	30450	270709	2614	4000	2.15	8
1 λ. 23.7871515 δ.	1.1	30450	599989	2605	4000	2.02	5
1 λ. 16.8233697 δ.	2	30450	636871	1980	4000	2.21	4
1 λ. 17.2590009 δ.	1.3	30450	640071	2924	4000	2.0	5
1 λ. 14.1548691 δ.	2	30450	705529	2042	4000	2.23	6
1 λ. 23.1958717 δ.	1.8	30471	547531	2168	4000	2.09	8
1 λ. 13.1255702 δ.	0	30471	703250	2071	4000	1.91	3
1 λ. 13.1719937 δ.	0.8	30471	448469	2948	4000	2.09	8
1 λ. 19.0601846 δ.	1	30471	234610	3001	4000	2.08	4
1 λ. 19.460406 δ.	1.9	30471	642670	2065	4000	2.14	4
1 λ. 21.95315 δ.	0.3	30528	668629	2640	4000	2.07	3
1 λ. 18.7380701 δ.	1	30549	282809	3028	4000	2.11	9
1 λ. 19.0864625 δ.	0.2	30549	567630	2792	4000	2.04	3
1 λ. 15.9962812 δ.	1.4	30549	600448	2305	4000	2.17	7
1 λ. 14.6657716 δ.	1.4	30549	607552	1409	4000	2.23	5
1 λ. 13.6244254 δ.	1.5	30549	608789	1841	4000	2.23	5
1 λ. 19.9062631 δ.	0.9	30549	402731	3068	4000	2.23	5
1 λ. 24.5520927 δ.	2	30549	576348	2697	4000	2.14	6
1 λ. 14.2023597 δ.	1.7	30549	563509	2165	4000	2.05	9
1 λ. 15.5127755 δ.	1.4	30549	564391	2167	4000	2.02	6
1 λ. 21.0711222 δ.	1	30549	232270	3030	4000	2.23	7
1 λ. 22.3179353 δ.	1.4	30549	560590	2848	4000	2.05	5
1 λ. 19.9204726 δ.	0	30549	614670	2953	4000	1.98	6
1 λ. 19.2566348 δ.	1.2	30549	633394	2159	4000	1.95	8
1 λ. 31.8528242 δ.	0.4	30549	734009	2919	4000	1.84	3
1 λ. 25.1993738 δ.	0.3	30549	794830	3064	4000	1.94	2
1 λ. 14.5377947 δ.	1.4	30571	699868	2637	4000	2.11	6
1 λ. 16.2062553 δ.	0	30571	594709	2441	4000	2.01	3
1 λ. 25.064511 δ.	1.7	30571	663250	1468	4000	2.07	2
1 λ. 13.7057042 δ.	0.9	30571	328832	3119	4000	2.05	5
1 λ. 21.2071431 δ.	1.9	30571	590311	1557	4000	2.06	7

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 14.4500011 δ.	1.7	30571	700469	2260	4000	2.27	4
1 λ. 25.6597386 δ.	1	30628	203730	2970	4000	2.25	10
1 λ. 11.1951493 δ.	1	30649	236288	3041	4000	2.22	3
1 λ. 14.2705222 δ.	1.4	30649	691189	2505	4000	2.03	3
1 λ. 14.2705222 δ.	1.4	30649	691189	2505	4000	2.03	3
1 λ. 18.468429 δ.	1.8	30649	590791	2164	4000	2.16	4
1 λ. 19.6135899 δ.	0.2	30649	651250	2332	4000	1.85	5
1 λ. 17.9181327 δ.	1.2	30649	575310	2924	4000	1.94	5
1 λ. 20.8601813 δ.	1.7	30649	597828	2151	4000	2.19	4
1 λ. 14.8411629 δ.	1.4	30649	632412	2265	4000	2.13	8
1 λ. 17.2255737 δ.	1.1	30649	635911	2701	4000	2.13	4
1 λ. 14.446908 δ.	1.5	30649	695250	1936	4000	2.16	5
1 λ. 12.844077 δ.	1	30670	208128	3031	4000	2.19	3
1 λ. 12.844077 δ.	1	30670	208128	3031	4000	2.19	3
1 λ. 14.5277407 δ.	0.8	30670	712309	2903	4000	2.13	2
1 λ. 14.5277407 δ.	0.8	30670	712309	2903	4000	2.13	2
1 λ. 18.9729337 δ.	0.3	30670	545248	2811	4000	2.01	4
1 λ. 22.0913067 δ.	1.3	30670	561671	2637	4000	2.06	7
1 λ. 12.7460804 δ.	1.7	30670	575669	2070	4000	2.02	4
1 λ. 27.8172291 δ.	0.2	30670	586091	2583	4000	1.92	5
1 λ. 23.3472322 δ.	0.2	30670	690969	3482	4000	1.96	3
1 λ. 25.4651882 δ.	1.6	30692	575868	2835	4000	2.02	9
1 λ. 22.1809346 δ.	1.9	30692	629632	2214	4000	2.17	4
1 λ. 26.8486139 δ.	0.1	30748	606389	2630	4000	2.11	5
1 λ. 18.0676649 δ.	0.1	30748	596811	2745	4000	1.93	3
1 λ. 13.1452996 δ.	0.9	30748	270770	3117	4000	2.18	7
1 λ. 18.3642665 δ.	1.1	30748	617988	2508	4000	2.07	4
1 λ. 18.5504436 δ.	1.1	30748	635449	3369	4000	2.28	4
1 λ. 32.5810765 δ.	0.6	30748	645689	2922	4000	2.05	5
1 λ. 20.9160191 δ.	1.9	30770	609131	2027	4000	2.21	8
1 λ. 23.8234113 δ.	1.2	30770	598510	2585	4000	2.04	6
1 λ. 25.6539374 δ.	1.4	30770	692331	2735	4000	1.95	6
1 λ. 11.4578347 δ.	2	30770	613191	2362	4000	2.06	5
1 λ. 14.161871 δ.	1.3	30770	570731	2302	4000	2.08	6
1 λ. 13.7529315 δ.	1.9	30770	605611	1920	4000	2.11	7
1 λ. 12.7403659 δ.	1	30770	254869	2968	4000	2.29	7
1 λ. 31.9064129 δ.	0.1	30770	524651	3173	4000	1.95	5
1 λ. 15.5889825 δ.	1.4	30770	591812	2110	4000	2.25	5
1 λ. 23.1178673 δ.	0	30770	606272	2470	4000	2.03	3
1 λ. 18.134718 δ.	1.7	30770	626670	2368	4000	2.13	8
1 λ. 14.7766009 δ.	1.4	30770	638108	1778	4000	2.07	4
1 λ. 25.3863169 δ.	0.4	30770	642048	2716	4000	1.99	3
1 λ. 26.0157016 δ.	0.2	30770	652352	2948	4000	2.03	3
1 λ. 19.2909588 δ.	0.1	30770	656249	2174	4000	2.04	3
1 λ. 26.8925903 δ.	1.2	30770	658112	2711	4000	2.05	6
1 λ. 14.6844496 δ.	1.8	30770	660370	1929	4000	2.14	8
1 λ. 22.0473065 δ.	1.6	30791	622350	1991	4000	2.13	5
1 λ. 23.3427369 δ.	1.1	30791	574869	3196	4000	2.12	8
1 λ. 19.6369684 δ.	0.9	30791	293369	3032	4000	2.12	6
1 λ. 21.5602018 δ.	1.6	30791	584811	1860	4000	2.18	6
1 λ. 17.2018598 δ.	1.6	30791	597508	1719	4000	2.11	7
1 λ. 15.923303 δ.	0	30791	630229	2366	4000	1.96	4
1 λ. 14.8936749 δ.	1	30848	241589	2943	4000	2.19	4
1 λ. 22.0381097 δ.	1.7	30848	630411	1870	4000	2.06	4
1 λ. 14.0823998 δ.	0.8	30869	300971	1782	4000	2.29	4
1 λ. 16.4976196 δ.	2	30869	584828	1707	4000	2.07	5

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 16.4953221 δ.	1.2	30869	610069	2550	4000	2.08	6
1 λ. 20.8895997 δ.	0	30869	630290	2092	4000	1.89	6
1 λ. 15.5130424 δ.	1.9	30869	656491	2162	4000	2.19	3
1 λ. 15.5130424 δ.	1.9	30869	656491	2162	4000	2.19	3
1 λ. 14.9231234 δ.	1.3	30869	662009	2420	4000	2.1	2
1 λ. 14.9231234 δ.	1.3	30869	662009	2420	4000	2.1	2
1 λ. 22.9486347 δ.	1.8	30869	578631	2058	4000	2.11	5
1 λ. 23.9449722 δ.	0.6	30869	645689	3223	4000	1.98	5
1 λ. 16.1395388 δ.	0.6	30869	664988	2849	4000	2.06	7
1 λ. 13.1031997 δ.	0.9	30869	281451	2968	4000	2.25	6
1 λ. 24.7024598 δ.	0.8	30869	551431	2859	4000	2.2	5
1 λ. 20.0078386 δ.	1.3	30869	569828	2495	4000	2.07	8
1 λ. 23.9422725 δ.	1.3	30869	573749	2312	4000	1.86	4
1 λ. 16.4938687 δ.	1.2	30869	602510	2594	4000	2.14	8
1 λ. 16.7126114 δ.	1.8	30869	606450	2366	4000	2.06	5
1 λ. 27.1749323 δ.	0.5	30869	611570	3070	4000	2.0	6
1 λ. 23.2748107 δ.	0.4	30869	620729	2783	4000	1.9	4
1 λ. 21.5217733 δ.	1.4	30869	651929	2207	4000	2.16	5
1 λ. 18.2362581 δ.	1.5	30869	691012	1946	4000	2.19	4
1 λ. 12.9678434 δ.	1.7	30891	592690	1847	4000	2.05	2
1 λ. 12.9678434 δ.	1.7	30891	592690	1847	4000	2.05	2
1 λ. 12.2901998 δ.	1.7	30891	619492	2380	4000	2.11	3
1 λ. 21.5115206 δ.	0	30891	595989	1891	4000	1.86	6
1 λ. 19.0790616 δ.	0.1	30891	573230	2672	4000	2.05	5
1 λ. 19.0288289 δ.	1	30891	209131	3026	4000	2.25	7
1 λ. 15.8303836 δ.	1	30891	288171	3028	4000	2.25	7
1 λ. 19.9689191 δ.	0	30891	643029	2397	4000	2.02	5
1 λ. 13.2804279 δ.	0.9	30912	271829	2964	4000	2.15	7
1 λ. 14.4857073 δ.	1.2	30969	620750	2112	4000	2.13	4
1 λ. 14.4857073 δ.	1.2	30969	620750	2112	4000	2.13	4
1 λ. 26.7194114 δ.	1.4	30969	626130	2729	4000	2.15	7
1 λ. 12.3441864 δ.	1.7	30969	566332	2151	4000	2.15	8
1 λ. 24.6024549 δ.	0.5	30969	890311	2006	4000	1.94	4
1 λ. 18.0008727 δ.	1	30969	203730	2984	4000	2.33	4
1 λ. 24.703519 δ.	1.6	30969	572149	2468	4000	1.96	4
1 λ. 25.2951704 δ.	0.1	30969	604132	2788	4000	2.01	6
1 λ. 23.661773 δ.	1.8	30969	609970	2502	4000	2.09	9
1 λ. 21.6719738 δ.	0.6	30990	762773	2750	4000	1.94	5
1 λ. 28.3519217 δ.	0.2	30990	675371	3009	4000	1.99	5
1 λ. 16.2105373 δ.	0.9	30990	231289	2697	4000	2.19	6
1 λ. 24.5870643 δ.	1.6	30990	653888	2153	4000	2.18	6
1 λ. 14.6685821 δ.	1.5	30990	644832	2575	4000	2.11	4
1 λ. 18.1832242 δ.	0.7	30990	668729	2840	4000	2.11	9
1 λ. 12.9801664 δ.	2	30990	583570	2106	4000	2.27	9
1 λ. 20.0203557 δ.	0.2	30990	563648	3024	4000	1.98	4
1 λ. 25.2460347 δ.	1.7	30990	575730	2499	4000	2.07	9
1 λ. 19.290734 δ.	1.8	30990	588651	2704	4000	2.08	8
1 λ. 20.746497 δ.	1.9	30990	595392	2497	4000	2.04	4
1 λ. 18.6167249 δ.	1.1	30990	612309	2526	4000	1.87	5
1 λ. 15.5518345 δ.	1.3	30990	632011	2540	4000	2.01	6
1 λ. 31.3386431 δ.	0.3	30990	641308	3154	4000	2.0	4
1 λ. 32.4269261 δ.	1.4	30990	671851	2814	4000	2.17	6
1 λ. 11.5771296 δ.	1	31012	221070	3024	4000	2.18	3
1 λ. 24.5292882 δ.	1.9	31068	574489	2421	4000	1.98	4
1 λ. 17.954911 δ.	0.9	31068	300828	2961	4000	2.29	9
1 λ. 26.5824054 δ.	1.3	31068	637589	2905	4000	2.06	4

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 15.8254455 δ.	0.9	31068	290510	3065	4000	2.25	5
1 λ. 14.9215694 δ.	1.9	31068	543950	1989	4000	2.04	9
1 λ. 19.2587719 δ.	1.2	31090	677412	2228	4000	2.11	5
1 λ. 24.0754296 δ.	1.6	31090	557952	2090	4000	2.15	5
1 λ. 24.1131131 δ.	0.4	31090	748032	3155	4000	1.92	3
1 λ. 26.4420696 δ.	1.1	31090	661369	2484	4000	1.97	4
1 λ. 23.8697302 δ.	1.5	31090	662148	2214	4000	2.11	6
1 λ. 18.6511238 δ.	1	31090	231730	3012	4000	2.24	6
1 λ. 21.0300717 δ.	0.8	31090	487829	2890	4000	2.2	6
1 λ. 18.5089252 δ.	1.2	31090	601010	2432	4000	1.93	5
1 λ. 15.6474558 δ.	1.3	31090	633230	2389	4000	1.98	4
1 λ. 17.2364894 δ.	0.8	31090	641429	2731	4000	2.06	4
1 λ. 16.7768854 δ.	0	31090	650048	2823	4000	2.05	7
1 λ. 20.4702959 δ.	0.8	31090	791730	2728	4000	2.05	3
1 λ. 19.194071 δ.	0.6	31090	817732	2628	4000	2.16	2
1 λ. 21.8412528 δ.	1.4	31111	611150	2169	4000	2.04	6
1 λ. 16.7797819 δ.	1.1	31132	594091	2971	4000	2.05	7
1 λ. 15.0286983 δ.	1.8	31168	584089	2463	4000	2.0	4
1 λ. 24.2663513 δ.	1.5	31168	644530	2791	4000	2.02	4
1 λ. 18.3237765 δ.	0.9	31168	305892	2988	4000	2.17	6
1 λ. 18.1911724 δ.	1.2	31168	591189	3251	4000	1.99	5
1 λ. 16.7561134 δ.	1.8	31168	615332	2070	4000	2.11	6
1 λ. 23.2301147 δ.	1.2	31168	637291	1981	4000	2.12	4
1 λ. 24.0041542 δ.	1.8	31168	701291	1543	4000	2.16	4
1 λ. 15.6587985 δ.	0.7	31189	577650	2800	4000	2.16	8
1 λ. 12.4962548 δ.	2	31189	592028	1364	4000	2.02	4
1 λ. 16.5054011 δ.	0.1	31189	682091	2397	4000	1.99	8
1 λ. 34.6252888 δ.	0.5	31189	718748	2902	4000	1.98	4
1 λ. 19.0059918 δ.	0.9	31189	274809	2971	4000	2.31	9
1 λ. 21.6785281 δ.	1.7	31189	573710	1870	4000	2.1	6
1 λ. 14.8273072 δ.	0.7	31189	530652	3045	4000	2.2	4
1 λ. 22.4247105 δ.	1.6	31189	588811	1976	4000	2.1	6
1 λ. 15.8128164 δ.	1.2	31189	605732	2704	4000	2.15	3
1 λ. 31.2430638 δ.	0.3	31189	630428	2702	4000	1.95	4
1 λ. 23.1648553 δ.	1.1	31189	630852	2589	4000	1.94	4
1 λ. 15.0765127 δ.	0	31189	723769	2427	4000	2.03	6
1 λ. 20.6537118 δ.	1.9	31211	601010	2172	4000	2.06	9
1 λ. 28.6083159 δ.	0.7	31211	873568	2307	4000	2.07	5
1 λ. 13.6331217 δ.	1	31211	321287	2999	4000	2.18	2
1 λ. 14.5496354 δ.	0.1	31211	606009	2405	4000	2.03	5
1 λ. 21.8651944 δ.	1.6	31232	669092	2000	4000	2.01	4
1 λ. 28.0639819 δ.	0.1	31268	580370	3114	4000	2.03	5
1 λ. 17.4602503 δ.	1.1	31268	639652	2106	4000	1.98	4
1 λ. 25.3405278 δ.	1.5	31289	631772	2703	4000	2.26	4
1 λ. 13.5550281 δ.	1.5	31289	643271	2009	4000	2.18	6
1 λ. 15.8418373 δ.	1.3	31289	599908	2848	4000	2.07	4
1 λ. 18.3466084 δ.	1.4	31289	790891	2482	4000	2.11	5
1 λ. 15.5165546 δ.	0.9	31289	270848	2934	4000	2.13	4
1 λ. 18.6047673 δ.	0.7	31289	452430	2168	4000	2.28	7
1 λ. 17.0031723 δ.	1.8	31289	594268	1961	4000	2.07	5
1 λ. 13.3279562 δ.	1.9	31289	613248	2258	4000	2.21	6
1 λ. 20.7410413 δ.	2	31289	625390	1286	4000	2.14	5
1 λ. 15.4019722 δ.	1	31310	252107	2993	4000	2.2	8
1 λ. 10.8589605 δ.	0.5	31310	559552	2352	4000	2.12	5
1 λ. 14.6533316 δ.	1.2	31310	592373	2331	4000	2.02	5
1 λ. 17.8116633 δ.	1.5	31310	607950	1732	4000	2.24	5

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 27.4141047 δ.	0.3	31310	649970	2243	4000	1.99	7
1 λ. 14.7079467 δ.	2	31310	652711	1738	4000	2.19	6
1 λ. 21.4367264 δ.	1.2	31310	624789	1806	4000	2.03	7
1 λ. 15.3420911 δ.	0.9	31310	307108	2985	4000	2.15	3
1 λ. 19.8540136 δ.	1.1	31310	670727	2489	4000	2.04	5
1 λ. 23.5420195 δ.	0.2	31310	645529	2427	4000	1.87	6
1 λ. 13.7629347 δ.	0	31310	570930	2471	4000	1.94	5
1 λ. 16.1381631 δ.	1.1	31310	585728	3052	4000	1.93	5
1 λ. 17.4343477 δ.	2	31310	589152	2190	4000	2.05	3
1 λ. 27.7026428 δ.	1.6	31310	599310	2795	4000	1.98	5
1 λ. 14.4329512 δ.	1.4	31310	599691	2121	4000	2.26	6
1 λ. 13.0446732 δ.	0.1	31310	622471	2809	4000	2.08	3
1 λ. 18.8457447 δ.	1.1	31310	633529	2192	4000	2.07	6
1 λ. 16.1270633 δ.	1.5	31310	717109	2651	4000	1.92	5
1 λ. 19.2979296 δ.	1.3	31310	746112	2505	4000	2.17	5
1 λ. 14.488011 δ.	1.5	31332	618510	1823	4000	2.07	6
1 λ. 15.1647161 δ.	1.9	31388	630809	2151	4000	2.12	7
1 λ. 22.5806137 δ.	2	31388	709170	2707	4000	2.05	2
1 λ. 20.6578411 δ.	0.1	31388	573849	2888	4000	2.09	2
1 λ. 13.7912917 δ.	1.5	31388	584850	2170	4000	2.15	8
1 λ. 16.7962405 δ.	1.1	31410	589131	2438	4000	1.99	4
1 λ. 16.7962405 δ.	1.1	31410	589131	2438	4000	1.99	4
1 λ. 20.8178624 δ.	1.9	31410	674770	1823	4000	2.16	7
1 λ. 21.1523076 δ.	1.1	31410	572971	2559	4000	2.05	7
1 λ. 20.4353735 δ.	0.2	31410	615908	2936	4000	1.96	6
1 λ. 22.4885386 δ.	1.9	31410	603310	2120	4000	2.12	7
1 λ. 12.2670386 δ.	1.7	31410	639410	1610	4000	2.16	3
1 λ. 13.4168628 δ.	1.7	31410	658169	2276	4000	2.16	7
1 λ. 24.437519 δ.	1.3	31410	681892	2962	4000	2.05	3
1 λ. 15.0239697 δ.	2	31410	683950	1689	4000	2.13	5
1 λ. 20.3339664 δ.	1.1	31410	687189	2074	4000	2.03	4
1 λ. 13.7454379 δ.	1.8	31431	578130	2335	4000	2.12	6
1 λ. 19.3645123 δ.	0	31431	628949	2561	4000	2.13	4
1 λ. 24.9086837 δ.	1.2	31431	589248	2642	4000	2.01	6
1 λ. 23.0659777 δ.	1.9	31431	626212	2183	4000	2.15	4
1 λ. 31.0407883 δ.	0.3	31431	773092	2947	4000	2.02	4
1 λ. 13.93634 δ.	2	31488	610549	1926	4000	2.25	9
1 λ. 21.2160823 δ.	1.3	31488	648789	2379	4000	2.09	3
1 λ. 13.0159985 δ.	2	31488	573831	1972	4000	2.1	10
1 λ. 13.8221387 δ.	0.7	31488	632430	2657	4000	2.05	6

Πίνακας 7: Αποτελέσματα βελτιστοποίησης του Αρχιμήδη για 45 πιλότους,

Χρόνος Εκτέλεσης	C1	C2	C3	C4	Καλύτερη Λύση	Χειρότερη Λύση	Έγκυρες λύσεις	Μονοπάτια	Ομοιότητα	Άλλατα
1 λ. 24.4224078 δ.	1	6	2	0.5	24171	582468	3362	4000	2.09	9
1 λ. 23.6831517 δ.	1	4	2	0.5	25451	597412	3067	4000	2.16	7
1 λ. 24.4158635 δ.	1	4	2	0.5	25628	509212	3367	4000	2.24	3
1 λ. 22.1383256 δ.	2	2	1	0.5	26268	511609	3387	4000	2.09	6
1 λ. 22.8133595 δ.	1	4	1	0.5	26290	554208	3458	4000	2.19	6
1 λ. 29.4437907 δ.	1	6	2	0.5	26311	545692	3045	4000	2.17	6
1 λ. 25.734947 δ.	1	6	1	0.5	26389	528910	3297	4000	2.23	6
1 λ. 21.2503237 δ.	1	2	2	1	26411	282290	2908	4000	2.31	4
1 λ. 23.345097 δ.	1	4	1	1	26510	473490	3393	4000	2.25	6

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 31.5750891 δ.	1	6	1	1	26588	675051	3210	4000	2.12	3
1 λ. 23.4036411 δ.	2	4	1	1	26709	594108	3389	4000	2.21	5
1 λ. 31.7818892 δ.	1	6	1	0.5	26731	584469	3173	4000	2.17	10
1 λ. 33.7807871 δ.	1	6	2	0.5	26731	23108000	971	4000	0.72	5
1 λ. 21.604385 δ.	1	2	2	1	26809	308388	2996	4000	2.27	5
1 λ. 21.2410381 δ.	2	6	2	1	26809	547189	2385	4000	2.19	3
1 λ. 23.5827438 δ.	1	6	1	0.5	26809	486809	3093	4000	2.23	5
1 λ. 25.8349544 δ.	1	6	1	1	26908	671531	3204	4000	2.18	3
1 λ. 25.1107068 δ.	2	2	1	1	26930	442688	3134	4000	2.19	4
1 λ. 19.7386466 δ.	1	4	2	1	27250	498652	3557	4000	2.35	5
1 λ. 24.2541346 δ.	1	4	1	0.5	27271	508729	2691	4000	2.25	10
1 λ. 23.8144332 δ.	2	4	2	0.5	27449	541291	3131	4000	2.1	2
1 λ. 21.4542138 δ.	1	2	2	0.5	27470	347371	3281	4000	2.25	7
1 λ. 21.5447515 δ.	1	4	2	1	27492	476452	2670	4000	2.22	9
1 λ. 27.4472007 δ.	1	6	1	1	27548	801788	2791	4000	2.08	4
1 λ. 28.6265778 δ.	1	2	1	1	27548	549148	3133	4000	2.11	2
1 λ. 32.5354885 δ.	2	4	2	1	27570	23412480	3018	4000	1.94	5
1 λ. 23.7626537 δ.	1	6	2	0.5	27570	504789	3360	4000	2.17	6
1 λ. 22.5662313 δ.	2	2	2	1	27570	369870	3492	4000	2.3	7
1 λ. 25.9837974 δ.	2	2	2	0.5	27669	338830	3223	4000	2.31	4
1 λ. 21.8144452 δ.	2	2	2	0.5	27669	390670	3318	4000	2.15	7
1 λ. 26.4539674 δ.	1	6	1	1	27790	531868	3483	4000	2.27	3
1 λ. 24.5064677 δ.	2	4	2	1	27790	625710	3159	4000	2.27	6
1 λ. 28.0028222 δ.	1	4	1	1	27890	732313	2877	4000	2.03	6
1 λ. 24.2458147 δ.	1	4	2	0.5	27890	504292	3126	4000	2.13	10
1 λ. 24.0411857 δ.	2	6	2	1	27911	588469	3027	4000	2.16	8
1 λ. 25.9499969 δ.	2	4	2	0.5	27989	380971	3844	4000	2.15	3
1 λ. 24.8694473 δ.	1	6	2	1	28011	570930	2943	4000	2.13	7
1 λ. 27.9940118 δ.	2	6	1	0.5	28068	22599840	1731	4000	1.14	3
1 λ. 23.0379319 δ.	2	2	1	0.5	28089	342649	3218	4000	2.14	4
1 λ. 22.7621225 δ.	2	6	1	0.5	28110	532789	3216	4000	2.29	7
1 λ. 22.1925457 δ.	2	2	1	0.5	28210	455431	3539	4000	2.17	3
1 λ. 23.6813538 δ.	1	6	2	1	28231	471609	3611	4000	2.18	7
1 λ. 19.6694403 δ.	1	2	1	0.5	28288	458528	3156	4000	2.28	6
1 λ. 18.6754828 δ.	1	2	1	0.5	28309	435769	3532	4000	2.26	4
1 λ. 31.082642 δ.	2	4	1	1	28309	23598880	1152	4000	0.63	2
1 λ. 28.8542145 δ.	1	4	1	1	28309	23152320	956	4000	0.71	3
1 λ. 21.4167786 δ.	1	4	2	1	28309	509490	2577	4000	2.17	4
1 λ. 16.1825173 δ.	1	2	1	0.5	28331	504171	3006	4000	2.11	3
1 λ. 21.3393069 δ.	2	6	2	0.5	28331	404487	3625	4000	2.22	7
1 λ. 18.3599144 δ.	1	2	1	0.5	28409	448932	3157	4000	2.24	6
1 λ. 30.7419901 δ.	2	2	2	1	28409	23619520	1187	4000	0.65	3
1 λ. 24.691073 δ.	2	2	2	0.5	28430	319211	3838	4000	2.03	5
1 λ. 26.9705975 δ.	1	2	1	1	28430	511609	3224	4000	2.11	6
1 λ. 26.136307 δ.	1	2	2	0.5	28430	343047	3180	4000	2.26	7
1 λ. 24.8992925 δ.	1	6	2	0.5	28430	466027	2956	4000	2.09	5
1 λ. 27.0638771 δ.	1	6	1	0.5	28430	508750	3006	4000	2.15	5
1 λ. 21.9202746 δ.	1	6	2	1	28430	470208	2157	4000	2.22	5
1 λ. 30.0001167 δ.	2	4	2	0.5	28530	403168	3232	4000	2.15	6
1 λ. 30.9506964 δ.	2	6	1	1	28530	659168	3407	4000	2.08	3
1 λ. 26.3069172 δ.	1	6	1	1	28530	476949	3434	4000	2.22	8

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 23.1717219 δ.	2	4	1	0.5	28530	538830	2859	4000	2.23	4
1 λ. 23.7011051 δ.	2	6	1	1	28530	627470	3081	4000	2.14	7
1 λ. 21.1159235 δ.	1	2	2	0.5	28530	459790	3174	4000	2.2	4
1 λ. 21.5632108 δ.	2	2	2	1	28530	371228	3486	4000	2.05	4
1 λ. 27.0894699 δ.	2	4	1	1	28530	633970	3716	4000	2.3	8
1 λ. 21.4142139 δ.	1	4	1	0.5	28551	538752	3324	4000	2.28	10
1 λ. 26.7288324 δ.	2	4	1	1	28551	479652	3851	4000	2.0	4
1 λ. 29.6953094 δ.	1	4	2	1	28608	23190240	1327	4000	0.94	9
1 λ. 20.1015894 δ.	1	4	2	1	28608	531712	2455	4000	2.24	6
1 λ. 31.3928887 δ.	2	2	2	1	28629	23328320	1003	4000	0.72	5
1 λ. 28.4192324 δ.	1	6	2	1	28651	532487	3211	4000	2.3	5
1 λ. 28.4239997 δ.	2	4	1	0.5	28708	505490	3091	4000	2.26	3
1 λ. 22.4063063 δ.	1	6	2	0.5	28729	405689	3067	4000	2.33	4
1 λ. 21.9392174 δ.	2	2	2	0.5	28729	401572	3466	4000	2.12	6
1 λ. 22.2157565 δ.	1	4	2	1	28750	498148	2659	4000	2.24	2
1 λ. 28.4614507 δ.	2	2	1	1	28750	486990	3676	4000	2.1	6
1 λ. 26.7181064 δ.	2	6	1	0.5	28750	541589	2974	4000	2.27	4
1 λ. 27.4045916 δ.	2	4	1	0.5	28750	456729	3623	4000	2.06	6
1 λ. 26.7712824 δ.	2	4	1	1	28750	541408	3183	4000	2.07	3
1 λ. 22.893056 δ.	2	4	1	0.5	28750	533269	3211	4000	2.33	5
1 λ. 16.7317652 δ.	1	2	1	0.5	28772	486930	3217	4000	2.2	6
1 λ. 28.0301869 δ.	2	2	2	0.5	28828	470549	3852	4000	2.24	4
1 λ. 26.3529751 δ.	1	2	1	0.5	28828	485948	3226	4000	2.23	4
1 λ. 23.9409897 δ.	1	2	1	1	28828	645170	3193	4000	2.08	4
1 λ. 23.8391826 δ.	1	4	2	1	28850	534848	3000	4000	2.15	7
1 λ. 24.1442105 δ.	1	6	1	0.5	28850	531431	2957	4000	2.16	9
1 λ. 25.0865954 δ.	1	4	2	0.5	28850	574912	3054	4000	2.09	3
1 λ. 28.6672631 δ.	2	4	2	0.5	28850	599890	3519	4000	2.16	8
1 λ. 21.1655031 δ.	1	2	1	1	28850	488789	3296	4000	2.15	5
1 λ. 27.7693844 δ.	1	4	1	1	28850	683790	3069	4000	2.04	3
1 λ. 24.1770853 δ.	1	6	2	1	28850	514311	3474	4000	2.1	6
1 λ. 22.7485824 δ.	2	6	1	0.5	28850	571790	2978	4000	2.17	10
1 λ. 20.3142554 δ.	1	2	2	1	28850	597049	2888	4000	2.17	10
1 λ. 26.3974121 δ.	2	2	2	1	28850	349689	3781	4000	2.1	6
1 λ. 25.8499761 δ.	1	2	2	0.5	28871	410468	3158	4000	2.08	3
1 λ. 22.8073145 δ.	1	6	2	0.5	28928	519090	3059	4000	2.32	7
1 λ. 27.4592035 δ.	1	6	2	1	28928	563968	2759	4000	2.12	6
1 λ. 27.6870844 δ.	2	2	1	0.5	28928	408988	3503	4000	2.06	2
1 λ. 15.4297106 δ.	2	2	2	0.5	28949	441330	3595	4000	2.18	2
1 λ. 23.5135579 δ.	1	6	2	0.5	28949	546809	3065	4000	2.1	6
1 λ. 23.0474873 δ.	1	4	2	0.5	28949	459989	3571	4000	2.25	3
1 λ. 21.5452374 δ.	2	4	2	1	28949	376612	3054	4000	2.07	3
1 λ. 25.0280715 δ.	1	4	1	0.5	28949	506951	3480	4000	2.33	6
1 λ. 21.5338388 δ.	2	2	1	0.5	28949	424231	3243	4000	2.17	3
1 λ. 21.8772457 δ.	1	6	1	0.5	28949	444050	2639	4000	2.15	3
1 λ. 25.1110531 δ.	1	6	1	1	28949	610389	2939	4000	2.01	5
1 λ. 24.3847171 δ.	1	6	2	0.5	28971	580910	3114	4000	2.1	5
1 λ. 24.8487158 δ.	2	6	2	1	28971	531470	3272	4000	2.14	4
1 λ. 20.5767015 δ.	1	2	2	1	28971	402069	3032	4000	2.22	7
1 λ. 28.620271 δ.	1	2	1	1	28971	505308	3261	4000	2.29	3
1 λ. 25.1681846 δ.	2	4	2	1	28971	343929	3747	4000	2.24	5

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 27.208997 δ.	1	2	1	0.5	28971	463829	3394	4000	2.11	6
1 λ. 26.3441339 δ.	1	4	2	0.5	28971	568711	3523	4000	2.21	7
1 λ. 23.6389721 δ.	1	2	2	0.5	29049	397049	3346	4000	2.15	5
1 λ. 27.2903112 δ.	2	6	2	1	29049	396011	3865	4000	1.94	3
1 λ. 23.1148385 δ.	1	2	1	0.5	29049	477589	3293	4000	2.14	3
1 λ. 24.6028035 δ.	2	6	1	1	29049	622670	3327	4000	2.25	7
1 λ. 22.7377824 δ.	2	4	2	0.5	29049	433607	3513	4000	2.35	6
1 λ. 26.5415742 δ.	2	6	1	1	29049	605092	3060	4000	2.21	4
1 λ. 21.5212568 δ.	1	2	2	1	29049	472811	2970	4000	2.18	7
1 λ. 22.2511232 δ.	2	2	2	0.5	29070	451989	3384	4000	2.03	2
1 λ. 27.5170696 δ.	2	6	1	0.5	29070	520530	3331	4000	2.12	6
1 λ. 25.0909392 δ.	1	6	1	0.5	29070	520932	3064	4000	2.1	8
1 λ. 23.3597263 δ.	1	4	2	1	29070	520512	2749	4000	2.07	2
1 λ. 28.1880454 δ.	2	2	1	1	29148	559332	3068	4000	2.2	5
1 λ. 23.3974821 δ.	1	6	1	1	29148	547029	3321	4000	2.17	7
1 λ. 22.4748213 δ.	2	4	2	0.5	29148	430891	3095	4000	2.3	4
1 λ. 23.3990294 δ.	1	2	2	1	29170	357689	3077	4000	2.24	5
1 λ. 21.6564154 δ.	1	4	1	0.5	29170	524871	3137	4000	2.29	6
1 λ. 23.9009824 δ.	2	2	1	1	29170	503310	3257	4000	2.15	3
1 λ. 30.1842543 δ.	1	4	1	1	29170	643691	3001	4000	2.13	9
1 λ. 28.8559228 δ.	1	6	1	1	29170	585010	3007	4000	2.06	6
1 λ. 23.7539216 δ.	1	2	1	0.5	29170	385668	3035	4000	2.29	5
1 λ. 24.9395165 δ.	2	4	1	0.5	29170	590748	3300	4000	2.34	3
1 λ. 23.9425673 δ.	2	4	2	0.5	29170	568348	3318	4000	2.2	7
1 λ. 24.7457715 δ.	2	6	2	1	29170	516807	3675	4000	2.13	6
1 λ. 23.0064859 δ.	1	2	1	1	29170	485148	3273	4000	2.23	7
1 λ. 23.5225131 δ.	2	4	2	1	29248	416811	3804	4000	2.04	4
1 λ. 23.2147727 δ.	2	6	1	0.5	29248	530247	3261	4000	2.19	5
1 λ. 22.0941739 δ.	2	6	1	1	29269	538112	3427	4000	2.28	10
1 λ. 22.1229459 δ.	2	2	2	0.5	29269	372430	3321	4000	2.2	3
1 λ. 20.3013385 δ.	1	4	2	0.5	29269	431410	3074	4000	2.04	4
1 λ. 22.326285 δ.	1	4	1	0.5	29269	519029	3333	4000	2.42	8
1 λ. 21.6881651 δ.	2	2	2	1	29269	435328	3555	4000	2.36	3
1 λ. 27.8043185 δ.	2	4	1	1	29269	519068	3367	4000	2.06	4
1 λ. 21.5051018 δ.	2	6	2	1	29269	601892	2138	4000	2.27	9
1 λ. 21.6718578 δ.	1	2	2	0.5	29269	469291	3249	4000	2.2	8
1 λ. 20.2075867 δ.	1	2	1	0.5	29269	477312	3134	4000	2.24	7
1 λ. 22.1112608 δ.	2	6	2	1	29291	439730	2553	4000	2.16	4
1 λ. 20.9309578 δ.	2	2	1	0.5	29291	481408	3481	4000	2.22	5
1 λ. 25.7571982 δ.	2	4	1	0.5	29291	558770	3021	4000	2.09	6
1 λ. 26.6051802 δ.	1	2	1	1	29369	502610	3309	4000	2.11	4
1 λ. 19.666472 δ.	1	2	2	1	29369	273308	2788	4000	2.29	2
1 λ. 26.793939 δ.	1	2	1	1	29369	548971	3102	4000	2.15	2
1 λ. 26.5679476 δ.	1	4	1	0.5	29369	542891	3137	4000	2.22	6
1 λ. 28.1188967 δ.	2	6	2	0.5	29369	463630	3702	4000	2.04	6
1 λ. 27.5232949 δ.	1	6	1	1	29390	672348	2986	4000	2.13	3
1 λ. 23.8096989 δ.	1	2	1	0.5	29390	533113	3282	4000	2.14	3
1 λ. 23.9167277 δ.	2	4	1	0.5	29390	392270	3672	4000	2.17	2
1 λ. 26.2589752 δ.	2	4	2	0.5	29390	470969	3461	4000	2.11	7
1 λ. 27.6842527 δ.	2	4	1	1	29390	553109	3467	4000	1.98	3
1 λ. 20.9790703 δ.	2	2	2	1	29390	309490	3615	4000	2.07	5

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 25.4497044 δ.	1	4	1	1	29390	534190	3523	4000	2.23	5
1 λ. 27.2236171 δ.	1	6	1	1	29390	722169	2883	4000	1.97	3
1 λ. 19.7047536 δ.	1	2	1	0.5	29412	386389	2842	4000	2.23	7
1 λ. 24.1035611 δ.	2	4	1	0.5	29468	443029	3838	4000	2.1	6
1 λ. 22.0615515 δ.	1	2	1	1	29468	450809	3451	4000	2.28	5
1 λ. 25.4383758 δ.	1	2	1	0.5	29468	487829	3322	4000	2.1	7
1 λ. 17.885388 δ.	2	2	1	1	29490	430969	3743	4000	2.14	4
1 λ. 19.5329358 δ.	1	2	2	1	29490	388331	2539	4000	2.18	7
1 λ. 29.504806 δ.	2	6	1	1	29490	669852	3602	4000	2.05	5
1 λ. 20.0513238 δ.	1	2	2	0.5	29490	485390	3052	4000	2.32	5
1 λ. 32.1716332 δ.	1	2	1	1	29490	579172	3199	4000	2.13	2
1 λ. 26.557796 δ.	2	4	1	0.5	29490	494030	3298	4000	2.19	2
1 λ. 20.8548473 δ.	1	2	2	0.5	29490	462891	3266	4000	2.18	4
1 λ. 22.8327088 δ.	1	4	2	0.5	29511	550588	3228	4000	2.13	5
1 λ. 22.2243658 δ.	2	6	2	0.5	29568	496590	3320	4000	2.22	6
1 λ. 21.8793019 δ.	1	4	1	0.5	29568	465749	3271	4000	2.31	9
1 λ. 25.1449223 δ.	1	6	2	0.5	29589	463851	3324	4000	2.15	3
1 λ. 21.5893746 δ.	2	6	1	0.5	29589	639851	2913	4000	2.19	4
1 λ. 23.2445503 δ.	1	2	2	0.5	29589	343829	3339	4000	2.2	4
1 λ. 26.5442995 δ.	1	4	2	0.5	29589	508971	3255	4000	2.22	3
1 λ. 20.4841953 δ.	2	6	1	1	29589	554389	3535	4000	2.2	3
1 λ. 26.9243794 δ.	1	4	1	0.5	29589	514891	3039	4000	2.1	8
1 λ. 32.5445482 δ.	1	2	1	1	29589	23104480	927	4000	0.68	3
1 λ. 30.2573665 δ.	2	6	2	1	29589	23128640	1561	4000	1.11	8
1 λ. 28.3649582 δ.	1	6	1	1	29589	612231	2806	4000	2.06	3
1 λ. 23.3869944 δ.	1	2	2	1	29589	377771	3065	4000	2.17	3
1 λ. 26.1327958 δ.	1	2	2	0.5	29589	353650	3348	4000	2.15	4
1 λ. 27.9209548 δ.	1	4	1	0.5	29589	533511	3065	4000	2.15	2
1 λ. 26.4846621 δ.	2	4	1	0.5	29589	576071	3029	4000	2.14	6
1 λ. 22.7858692 δ.	2	6	1	0.5	29589	523470	3023	4000	2.17	2
1 λ. 32.8205427 δ.	1	6	2	0.5	29611	23223680	965	4000	0.67	3
1 λ. 32.6962507 δ.	2	4	2	1	29611	471310	3643	4000	2.04	4
1 λ. 22.707382 δ.	1	2	2	0.5	29611	529429	3204	4000	2.24	4
1 λ. 27.2737532 δ.	1	4	2	0.5	29611	522069	3038	4000	2.29	7
1 λ. 26.5978944 δ.	1	2	1	1	29668	505628	3029	4000	2.08	3
1 λ. 18.7993791 δ.	1	4	2	0.5	29689	478969	3465	4000	2.2	2
1 λ. 18.0729279 δ.	2	4	2	0.5	29689	506212	3537	4000	2.24	2
1 λ. 25.5523138 δ.	2	4	1	1	29689	763691	3064	4000	1.97	3
1 λ. 29.7194575 δ.	2	6	2	0.5	29689	424612	3831	4000	2.01	3
1 λ. 28.9239137 δ.	1	6	2	1	29689	514091	3063	4000	2.22	6
1 λ. 22.2097878 δ.	1	6	1	0.5	29689	624932	2981	4000	2.09	7
1 λ. 25.5642938 δ.	2	2	1	1	29689	481308	3155	4000	2.11	7
1 λ. 21.5607006 δ.	2	2	2	1	29689	386731	3061	4000	2.08	4
1 λ. 22.016112 δ.	2	2	2	1	29689	337451	3737	4000	2.19	5
1 λ. 25.5011261 δ.	2	4	1	0.5	29689	526791	3479	4000	2.12	6
1 λ. 24.6112935 δ.	2	6	2	1	29689	418069	3304	4000	2.22	3
1 λ. 24.0262425 δ.	1	4	2	1	29689	456612	3247	4000	2.13	4
1 λ. 25.9797265 δ.	2	6	1	1	29689	661013	3143	4000	2.01	6
1 λ. 25.5937721 δ.	2	6	2	1	29689	520868	3820	4000	2.11	2
1 λ. 22.7522955 δ.	2	2	2	1	29689	326329	3528	4000	2.13	4
1 λ. 22.2421676 δ.	1	2	1	0.5	29710	407047	3238	4000	2.34	5

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 8.7210154 δ.	1	6	2	1	29710	468210	2864	4000	2.31	9
1 λ. 21.721015 δ.	1	2	1	0.5	29710	480690	3281	4000	2.08	3
1 λ. 20.910173 δ.	1	2	2	1	29710	318450	2922	4000	2.44	7
1 λ. 23.7742589 δ.	1	6	1	0.5	29710	527829	3077	4000	2.07	8
1 λ. 23.589053 δ.	1	6	2	0.5	29710	471630	3361	4000	1.97	4
1 λ. 28.3805533 δ.	1	2	1	0.5	29710	527431	3352	4000	2.18	9
1 λ. 26.3374067 δ.	2	4	2	0.5	29710	421369	3822	4000	1.9	4
1 λ. 20.1810602 δ.	1	2	2	0.5	29710	322489	3031	4000	2.16	5
1 λ. 27.7086753 δ.	1	6	2	0.5	29732	528249	3121	4000	2.06	3
1 λ. 29.627489 δ.	1	6	2	0.5	29788	560270	3113	4000	2.12	7
1 λ. 22.739598 δ.	1	2	2	1	29788	342108	2923	4000	2.2	4
1 λ. 21.151242 δ.	1	2	2	0.5	29788	387954	3143	4000	2.13	4
1 λ. 25.4311243 δ.	2	2	1	1	29788	489650	3685	4000	1.92	6
1 λ. 30.3973876 δ.	2	6	1	1	29788	651790	3131	4000	2.11	5
1 λ. 24.0669286 δ.	2	4	1	1	29788	568988	3291	4000	2.17	5
1 λ. 22.4703185 δ.	1	6	2	1	29810	556110	2615	4000	2.25	3
1 λ. 24.8409587 δ.	1	6	1	0.5	29810	617010	2985	4000	2.11	3
1 λ. 29.1096065 δ.	1	4	1	1	29810	527431	3183	4000	2.15	8
1 λ. 23.9642604 δ.	1	6	2	1	29810	512889	2900	4000	2.1	5
1 λ. 20.1707879 δ.	2	4	2	0.5	29810	345529	3480	4000	2.1	8
1 λ. 27.3846707 δ.	2	6	1	0.5	29810	462471	3859	4000	2.03	3
1 λ. 27.8006999 δ.	1	6	2	0.5	29810	571349	2948	4000	2.15	4
1 λ. 31.6156654 δ.	2	4	1	1	29810	513767	3129	4000	2.18	5
1 λ. 25.1770072 δ.	2	6	1	0.5	29810	466731	3072	4000	2.21	6
1 λ. 30.8123904 δ.	1	2	2	1	29810	23350880	940	4000	0.65	2
1 λ. 23.3506318 δ.	1	2	2	0.5	29810	562652	3247	4000	2.27	10
1 λ. 22.1809965 δ.	1	2	1	1	29810	492750	3105	4000	2.1	6
1 λ. 25.0258896 δ.	1	6	1	1	29810	542791	3488	4000	2.16	6
1 λ. 29.370931 δ.	2	2	1	1	29831	440910	3816	4000	2.0	3
1 λ. 29.674498 δ.	2	4	1	1	29831	615310	3486	4000	2.12	5
1 λ. 27.514927 δ.	1	6	1	0.5	29831	600850	3086	4000	2.14	5
1 λ. 25.3460631 δ.	1	6	2	1	29831	648807	2639	4000	2.14	5
1 λ. 21.3316594 δ.	1	2	2	0.5	29888	406492	3149	4000	2.24	5
1 λ. 17.1710499 δ.	1	4	1	0.5	29909	569209	3044	4000	2.24	2
1 λ. 21.8232628 δ.	2	4	1	1	29909	513572	3152	4000	2.03	2
1 λ. 12.3721305 δ.	1	2	2	0.5	29909	393870	3115	4000	2.22	4
1 λ. 18.6556113 δ.	1	2	2	1	29909	367872	2881	4000	2.26	5
1 λ. 21.0658871 δ.	1	4	2	0.5	29909	23197120	2791	4000	1.98	4
1 λ. 22.048632 δ.	1	2	1	1	29909	347449	3195	4000	2.23	6
1 λ. 22.9742102 δ.	1	4	2	1	29909	418848	2533	4000	2.13	6
1 λ. 25.6318211 δ.	2	4	1	0.5	29909	502489	3322	4000	2.13	4
1 λ. 23.6063171 δ.	2	6	1	1	29909	533490	3563	4000	2.15	8
1 λ. 23.5884482 δ.	1	4	1	0.5	29909	524309	3066	4000	2.06	3
1 λ. 23.1215212 δ.	2	4	1	0.5	29909	400811	3104	4000	2.09	4
1 λ. 18.678015 δ.	1	4	1	0.5	29909	506631	3116	4000	2.31	4
1 λ. 24.0974387 δ.	1	6	2	1	29909	532192	3170	4000	2.26	7
1 λ. 20.8961977 δ.	1	6	1	0.5	29909	489070	3529	4000	2.34	4
1 λ. 26.539162 δ.	1	2	2	1	29909	361671	3382	4000	2.17	6
1 λ. 21.0544196 δ.	1	2	2	0.5	29909	464732	3233	4000	2.07	7
1 λ. 23.2077082 δ.	1	6	2	0.5	29909	422990	3065	4000	2.23	6
1 λ. 23.5907097 δ.	1	6	2	1	29909	486990	3229	4000	2.31	4

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 24.4740254 δ.	2	2	2	0.5	29909	301170	3333	4000	2.22	4
1 λ. 23.7321926 δ.	1	4	2	0.5	29931	509511	2926	4000	2.19	4
1 λ. 23.5330232 δ.	2	6	1	0.5	29931	432370	3855	4000	2.11	4
1 λ. 21.3015439 δ.	1	4	2	1	29931	387029	2613	4000	2.33	5
1 λ. 22.7128157 δ.	2	6	1	0.5	29931	495552	2838	4000	2.22	4
1 λ. 21.0734181 δ.	1	2	1	0.5	29931	398048	3091	4000	2.17	5
1 λ. 22.7216131 δ.	1	2	2	0.5	29931	324388	3355	4000	2.1	2
1 λ. 22.4657044 δ.	2	2	2	1	29931	329152	2823	4000	2.2	4
1 λ. 23.5022169 δ.	1	6	2	0.5	29988	514190	3011	4000	2.12	4
1 λ. 11.6324374 δ.	1	2	2	1	29988	397447	3464	4000	2.25	3
1 λ. 23.0371906 δ.	2	4	1	1	29988	506588	3385	4000	2.28	3
1 λ. 17.2441895 δ.	2	4	2	1	30009	487972	2695	4000	2.19	2
1 λ. 12.9919398 δ.	1	2	1	0.5	30009	520668	3198	4000	2.17	2
1 λ. 20.3482661 δ.	1	6	1	1	30009	566169	3394	4000	2.19	6
1 λ. 25.814701 δ.	1	6	2	1	30009	568612	2780	4000	2.07	2
1 λ. 19.3988485 δ.	1	6	2	0.5	30009	507371	3095	4000	2.25	4
1 λ. 26.3274336 δ.	1	6	2	0.5	30009	534770	3209	4000	2.22	3
1 λ. 25.1329453 δ.	1	4	2	1	30030	472832	3070	4000	2.21	3
1 λ. 26.0821265 δ.	2	2	2	1	30030	369429	3885	4000	2.27	4
1 λ. 12.6319553 δ.	1	2	1	0.5	30030	644370	3300	4000	2.13	4
1 λ. 27.895064 δ.	1	4	2	0.5	30030	551552	3207	4000	2.07	6
1 λ. 22.6020921 δ.	1	4	2	1	30030	554212	2904	4000	2.05	4
1 λ. 25.8031622 δ.	1	6	1	1	30030	23105440	947	4000	0.73	3
1 λ. 19.1495223 δ.	1	2	2	0.5	30030	298667	3162	4000	2.19	9
1 λ. 30.2789044 δ.	2	4	1	0.5	30030	416910	3849	4000	2.13	3
1 λ. 24.9625274 δ.	1	6	2	1	30030	521888	3306	4000	2.12	6
1 λ. 30.4056001 δ.	2	6	1	1	30030	646009	3057	4000	1.95	3
1 λ. 28.851874 δ.	1	6	1	0.5	30030	540409	2919	4000	2.23	6
1 λ. 27.2448938 δ.	2	2	1	1	30052	481131	3175	4000	1.99	5
1 λ. 20.4752568 δ.	1	2	2	1	30052	293547	3294	4000	2.21	5
1 λ. 28.1906097 δ.	1	6	1	1	30108	476487	3527	4000	2.17	6
1 λ. 24.5204369 δ.	1	4	2	1	30108	442069	3338	4000	2.16	2
1 λ. 18.9165147 δ.	1	6	2	0.5	30130	461127	3086	4000	2.2	2
1 λ. 22.43641 δ.	1	2	2	1	30130	293831	2737	4000	2.27	7
1 λ. 30.9137288 δ.	2	2	1	1	30130	23762720	1073	4000	0.64	4
1 λ. 19.1625836 δ.	1	6	2	0.5	30130	468352	3556	4000	2.17	4
1 λ. 25.3510496 δ.	1	4	1	1	30130	604231	2957	4000	2.18	5
1 λ. 19.0060709 δ.	1	6	2	1	30130	500331	2103	4000	2.18	6
1 λ. 21.6866417 δ.	2	2	2	1	30130	372551	3824	4000	2.04	5
1 λ. 22.2345093 δ.	1	2	1	1	30130	487509	3294	4000	2.23	7
1 λ. 27.397553 δ.	2	6	2	0.5	30130	385330	3828	4000	2.13	6
1 λ. 24.7002459 δ.	2	4	2	1	30130	439929	2676	4000	2.14	4
1 λ. 25.3544121 δ.	2	2	2	1	30130	439730	2881	4000	2.06	7
1 λ. 22.2331682 δ.	1	6	2	1	30130	561088	2868	4000	2.13	4
1 λ. 23.4709724 δ.	1	4	1	0.5	30130	505490	3560	4000	2.34	6
1 λ. 23.6046901 δ.	2	2	1	0.5	30130	523250	3752	4000	2.07	3
1 λ. 25.8163821 δ.	1	2	1	0.5	30151	456768	3263	4000	2.23	2
1 λ. 24.2181431 δ.	2	2	2	0.5	30151	311012	3239	4000	2.19	2
1 λ. 24.8385472 δ.	2	4	2	1	30151	402930	2878	4000	2.24	9
1 λ. 28.1781806 δ.	2	6	1	1	30151	638990	3638	4000	2.06	3
1 λ. 21.3057207 δ.	1	2	1	0.5	30208	563748	3209	4000	2.06	1

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 25.6411098 δ.	1	2	1	1	30208	528448	3271	4000	2.0	1
1 λ. 21.3646135 δ.	1	2	2	0.5	30208	347591	3319	4000	2.08	1
1 λ. 19.4902763 δ.	1	4	1	1	30208	579748	3078	4000	1.99	1
1 λ. 20.3208326 δ.	1	6	1	0.5	30208	691492	3068	4000	2.0	1
1 λ. 21.4598902 δ.	1	6	1	1	30208	619349	3003	4000	1.95	1
1 λ. 21.1229196 δ.	1	6	2	1	30208	475150	2875	4000	2.04	1
1 λ. 23.5511402 δ.	2	2	1	0.5	30208	507211	3307	4000	2.04	1
1 λ. 21.6225497 δ.	2	2	1	1	30208	469148	3086	4000	2.01	1
1 λ. 15.8631044 δ.	2	2	2	1	30208	284971	3039	4000	2.09	1
1 λ. 19.6634399 δ.	2	4	1	0.5	30208	547790	3209	4000	2.02	1
1 λ. 19.1853847 δ.	2	6	1	0.5	30208	573952	3097	4000	2.02	1
1 λ. 20.9927772 δ.	2	6	1	1	30208	731292	2995	4000	1.96	1
1 λ. 18.5794792 δ.	2	6	2	0.5	30208	439950	3297	4000	2.03	1
1 λ. 18.2418498 δ.	2	6	2	1	30208	601671	2927	4000	2.03	1
1 λ. 16.8731094 δ.	1	2	1	0.5	30229	401131	3199	4000	2.15	6
1 λ. 27.0601334 δ.	1	6	2	0.5	30229	454990	3587	4000	2.08	6
1 λ. 24.7054864 δ.	1	6	1	0.5	30229	669049	3213	4000	2.27	4
1 λ. 23.0604026 δ.	1	4	2	1	30229	505572	2964	4000	2.08	4
1 λ. 26.0518977 δ.	1	4	1	1	30229	630308	3017	4000	2.01	7
1 λ. 22.3372167 δ.	1	2	2	1	30229	307591	2867	4000	2.24	4
1 λ. 22.4816632 δ.	1	6	2	0.5	30229	477852	3248	4000	2.22	9
1 λ. 24.4851872 δ.	2	4	2	1	30229	562951	3287	4000	2.2	4
1 λ. 24.2514356 δ.	1	4	2	0.5	30229	524793	2927	4000	2.17	4
1 λ. 23.9164802 δ.	2	6	2	0.5	30251	514112	3103	4000	2.03	3
1 λ. 28.5634616 δ.	1	6	1	0.5	30251	542812	3121	4000	2.01	4
1 λ. 26.127913 δ.	1	2	1	0.5	30251	392050	3264	4000	2.2	7
1 λ. 21.5932311 δ.	1	2	1	0.5	30251	432832	3324	4000	2.24	2
1 λ. 25.6075459 δ.	2	2	2	0.5	30251	395150	3166	4000	2.27	8
1 λ. 21.4628595 δ.	1	6	2	0.5	30251	560089	3131	4000	2.22	7
1 λ. 24.4197801 δ.	1	2	1	0.5	30329	512370	3252	4000	2.19	5
1 λ. 22.9943331 δ.	1	4	1	0.5	30329	581611	3061	4000	2.06	4
1 λ. 22.0726973 δ.	2	2	1	0.5	30329	437668	3458	4000	2.19	5
1 λ. 30.2394054 δ.	2	4	1	1	30329	625550	3499	4000	2.28	5
1 λ. 25.1274253 δ.	2	6	2	0.5	30329	494229	3483	4000	2.07	6
1 λ. 28.2492724 δ.	2	6	1	0.5	30329	547509	3144	4000	2.25	5
1 λ. 30.5362635 δ.	2	4	2	0.5	30329	477810	3378	4000	2.1	5
1 λ. 33.3407402 δ.	1	6	1	0.5	30329	546631	3035	4000	2.12	6
1 λ. 25.5124591 δ.	1	4	2	1	30329	540210	2946	4000	2.15	4
1 λ. 26.2396266 δ.	2	6	2	0.5	30329	463509	3761	4000	2.11	3
1 λ. 23.0298773 δ.	1	4	1	0.5	30329	477632	3191	4000	1.97	2
1 λ. 22.5463739 δ.	1	2	2	0.5	30350	402972	3035	4000	2.31	4
1 λ. 11.8653029 δ.	2	2	1	0.5	30350	334450	3715	4000	2.25	3
1 λ. 22.5680265 δ.	1	6	1	0.5	30350	531790	2967	4000	2.08	4
1 λ. 24.6433197 δ.	2	2	1	1	30350	477390	3130	4000	2.18	7
1 λ. 25.6799188 δ.	2	4	2	1	30350	523989	3457	4000	2.19	5
1 λ. 27.967677 δ.	2	4	2	1	30350	382670	3431	4000	2.17	4
1 λ. 24.9192028 δ.	1	6	1	0.5	30350	526169	3294	4000	2.17	11
1 λ. 24.8364855 δ.	1	2	2	0.5	30350	403911	3104	4000	2.27	5
1 λ. 25.6761912 δ.	1	4	1	0.5	30350	568711	3192	4000	2.18	3
1 λ. 22.7333828 δ.	1	6	2	1	30350	578069	3186	4000	2.17	3
1 λ. 22.8427352 δ.	2	2	2	0.5	30350	422770	3723	4000	2.18	3

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 24.8649241 δ.	2	6	2	0.5	30350	454329	3475	4000	2.14	6
1 λ. 20.7566452 δ.	1	2	2	0.5	30350	408469	3312	4000	2.28	6
1 λ. 25.787687 δ.	2	4	1	0.5	30350	485070	3446	4000	2.14	5
1 λ. 18.2183804 δ.	1	6	1	0.5	30372	463090	3542	4000	2.24	9
1 λ. 26.0976851 δ.	2	4	2	0.5	30372	520028	3321	4000	2.14	4
1 λ. 20.4670336 δ.	1	6	2	1	30428	398247	2529	4000	2.25	4
1 λ. 21.0553015 δ.	1	2	1	1	30428	468167	3499	4000	2.22	5
1 λ. 25.4302442 δ.	2	2	2	1	30428	332452	3553	4000	2.17	5
1 λ. 19.566855 δ.	1	2	2	0.5	30428	295289	3063	4000	2.17	7
1 λ. 22.123263 δ.	2	4	2	1	30428	471029	3595	4000	2.07	3
1 λ. 26.6137614 δ.	2	2	1	0.5	30428	461092	3844	4000	2.17	6
1 λ. 23.6405169 δ.	1	6	2	1	30450	488469	3095	4000	2.22	7
1 λ. 26.5225767 δ.	1	6	2	1	30450	478670	3490	4000	2.14	5
1 λ. 20.8028326 δ.	1	2	2	1	30450	331570	3027	4000	2.23	7
1 λ. 25.8835811 δ.	1	6	1	0.5	30450	519588	2957	4000	2.08	4
1 λ. 23.2641602 δ.	1	4	1	0.5	30450	625671	3469	4000	2.21	8
1 λ. 25.2418423 δ.	1	2	1	1	30450	562069	3172	4000	2.2	4
1 λ. 28.6498587 δ.	1	2	1	1	30450	557191	3319	4000	2.11	5
1 λ. 34.115822 δ.	1	4	2	1	30450	23166560	997	4000	0.72	5
1 λ. 20.9170885 δ.	2	4	2	0.5	30450	315890	3069	4000	2.3	5
1 λ. 23.1454611 δ.	1	6	2	0.5	30450	587029	3015	4000	2.12	8
1 λ. 23.076755 δ.	1	2	1	0.5	30450	403328	3173	4000	2.24	6
1 λ. 25.3041729 δ.	1	6	1	0.5	30450	501070	3501	4000	2.37	4
1 λ. 24.5682422 δ.	2	4	2	0.5	30450	475492	3210	4000	2.21	6
1 λ. 27.3393782 δ.	1	2	1	1	30450	556530	3018	4000	2.11	3
1 λ. 24.3138647 δ.	1	2	2	0.5	30450	359630	3346	4000	2.22	5
1 λ. 23.4916954 δ.	1	2	2	0.5	30450	523189	3072	4000	2.12	5
1 λ. 22.6178391 δ.	1	4	1	0.5	30450	540548	3078	4000	2.15	2
1 λ. 21.4715048 δ.	1	4	2	0.5	30450	500750	3175	4000	2.17	5
1 λ. 31.3370381 δ.	1	2	2	0.5	30450	23102400	966	4000	0.72	5
1 λ. 22.395228 δ.	1	4	2	0.5	30471	494350	3141	4000	2.19	6
1 λ. 24.2327287 δ.	1	6	2	1	30471	532608	3259	4000	1.97	5
1 λ. 23.0612898 δ.	2	4	2	0.5	30471	437390	3178	4000	2.09	3
1 λ. 23.0591026 δ.	1	4	2	1	30471	372487	3334	4000	2.04	3
1 λ. 28.3230593 δ.	1	2	2	1	30528	446770	3560	4000	2.17	8
1 λ. 24.4341263 δ.	1	2	1	0.5	30528	538649	3197	4000	2.21	7
1 λ. 24.7869757 δ.	1	4	1	0.5	30549	573031	3319	4000	2.17	4
1 λ. 17.1481841 δ.	1	4	1	1	30549	584391	3108	4000	2.18	3
1 λ. 18.1804558 δ.	2	4	1	1	30549	642809	3060	4000	2.09	6
1 λ. 13.9346849 δ.	1	2	1	0.5	30549	537970	3165	4000	2.19	6
1 λ. 26.1625954 δ.	2	6	1	1	30549	626229	3026	4000	2.01	7
1 λ. 20.9725802 δ.	1	2	1	1	30549	443246	3177	4000	2.23	7
1 λ. 27.0826108 δ.	2	6	1	1	30549	573148	3112	4000	2.13	2
1 λ. 27.1168197 δ.	2	4	2	0.5	30549	511730	3360	4000	1.95	3
1 λ. 21.0161926 δ.	2	2	2	0.5	30549	451591	3194	4000	2.28	5
1 λ. 25.3636444 δ.	1	2	1	1	30549	421191	3080	4000	2.23	7
1 λ. 29.2227396 δ.	1	6	2	1	30549	537668	3142	4000	2.13	5
1 λ. 27.7007706 δ.	2	2	1	1	30549	465152	3386	4000	2.14	3
1 λ. 25.5252613 δ.	2	6	2	1	30549	489771	2760	4000	2.3	2
1 λ. 24.1978428 δ.	1	2	2	0.5	30549	430290	3304	4000	2.11	2
1 λ. 26.608167 δ.	1	4	2	1	30549	516551	3169	4000	2.07	6

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 22.8178874 δ.	2	2	1	0.5	30549	364210	3307	4000	2.3	5
1 λ. 26.9116455 δ.	1	6	1	1	30549	548772	3477	4000	2.39	10
1 λ. 24.7328596 δ.	2	4	2	1	30549	512370	3704	4000	2.03	4
1 λ. 26.5321739 δ.	2	2	1	1	30549	504850	3001	4000	2.18	3
1 λ. 23.1197203 δ.	1	4	2	1	30549	626752	3038	4000	2.06	4
1 λ. 27.7039646 δ.	2	4	1	1	30549	602247	3109	4000	2.19	7
1 λ. 25.7864088 δ.	2	6	2	0.5	30549	469668	3375	4000	2.16	3
1 λ. 29.2461896 δ.	1	6	1	1	30571	646329	2650	4000	1.95	5
1 λ. 15.181168 δ.	1	4	1	0.5	30571	511652	3156	4000	2.1	7
1 λ. 23.2104152 δ.	1	4	2	0.5	30571	456370	3421	4000	2.19	6
1 λ. 24.0623013 δ.	1	2	1	1	30571	561970	3413	4000	2.28	10
1 λ. 22.1789139 δ.	2	4	1	0.5	30571	570091	2791	4000	2.24	6
1 λ. 26.1286813 δ.	2	2	2	0.5	30571	377188	3808	4000	2.11	3
1 λ. 32.5199673 δ.	1	2	1	1	30571	420608	3243	4000	2.21	4
1 λ. 20.6183966 δ.	1	4	1	1	30571	521849	3369	4000	2.29	5
1 λ. 25.1718506 δ.	1	2	2	1	30571	298091	2831	4000	2.25	3
1 λ. 26.7242519 δ.	2	6	1	0.5	30571	600789	3147	4000	2.09	2
1 λ. 23.1367057 δ.	2	2	2	0.5	30571	323349	3321	4000	2.23	4
1 λ. 23.5509541 δ.	2	4	2	0.5	30571	471310	3224	4000	2.33	5
1 λ. 19.0029737 δ.	1	2	2	1	30649	317589	2881	4000	2.08	5
1 λ. 24.6930652 δ.	1	4	2	1	30649	409948	2936	4000	2.17	5
1 λ. 30.3669701 δ.	1	6	1	1	30649	608889	3178	4000	2.02	2
1 λ. 24.722642 δ.	2	2	2	0.5	30649	415310	3752	4000	2.04	5
1 λ. 26.4901266 δ.	2	4	2	0.5	30649	359531	3785	4000	2.01	6
1 λ. 26.9537657 δ.	2	4	1	0.5	30649	617131	3533	4000	2.19	10
1 λ. 24.3599537 δ.	2	4	1	1	30649	638869	3744	4000	2.29	5
1 λ. 25.0435155 δ.	1	4	1	1	30649	441152	3520	4000	2.15	6
1 λ. 27.4735858 δ.	2	2	2	1	30649	398251	3749	4000	2.2	2
1 λ. 27.165791 δ.	1	2	1	1	30649	439452	3261	4000	2.2	5
1 λ. 23.6507568 δ.	2	2	1	1	30670	412629	3430	4000	2.14	5
1 λ. 11.0940924 δ.	2	6	1	0.5	30670	544988	3566	4000	2.18	5
1 λ. 21.4372846 δ.	2	4	1	0.5	30670	605429	2903	4000	2.25	4
1 λ. 24.8409382 δ.	2	6	2	0.5	30670	503150	3371	4000	2.13	5
1 λ. 23.5117558 δ.	1	4	1	0.5	30670	641152	3205	4000	2.21	3
1 λ. 23.5552529 δ.	2	2	1	0.5	30670	570610	3216	4000	2.14	8
1 λ. 23.3147502 δ.	2	4	1	0.5	30670	518788	3193	4000	2.19	6
1 λ. 22.0549901 δ.	1	2	2	0.5	30670	351068	3299	4000	2.12	4
1 λ. 24.3476069 δ.	1	4	1	0.5	30670	480953	3062	4000	2.29	3
1 λ. 23.7095817 δ.	1	6	2	1	30670	512370	2738	4000	2.19	5
1 λ. 26.1674808 δ.	2	2	1	1	30670	516971	3310	4000	2.14	3
1 λ. 21.5384041 δ.	2	2	1	0.5	30670	486571	3439	4000	2.1	5
1 λ. 32.0701216 δ.	2	2	1	1	30670	23934080	1175	4000	0.67	6
1 λ. 22.8817082 δ.	1	6	2	1	30670	427570	3586	4000	2.02	8
1 λ. 24.5924113 δ.	1	2	1	1	30670	501248	3170	4000	2.19	5
1 λ. 21.9482705 δ.	1	2	2	0.5	30670	378553	3311	4000	2.14	3
1 λ. 23.8542232 δ.	2	4	2	0.5	30670	390329	3243	4000	2.21	5
1 λ. 27.5490982 δ.	1	2	1	1	30670	477092	3200	4000	2.08	3
1 λ. 21.3595021 δ.	2	2	2	1	30692	439950	3272	4000	2.15	4
1 λ. 19.9611519 δ.	2	2	2	1	30748	277148	2798	4000	2.31	3
1 λ. 26.6783297 δ.	1	6	1	1	30748	711211	2843	4000	1.92	2
1 λ. 22.4482691 δ.	1	4	2	1	30748	470549	3021	4000	2.29	9

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 21.9886361 δ.	1	6	1	0.5	30748	524750	2936	4000	2.13	7
1 λ. 25.7214748 δ.	1	4	1	1	30748	616171	2995	4000	2.02	2
1 λ. 21.4223924 δ.	1	2	2	1	30748	336149	3020	4000	2.18	2
1 λ. 29.7999134 δ.	1	4	1	1	30770	23290560	943	4000	0.7	3
1 λ. 22.8725748 δ.	2	6	1	0.5	30770	575972	2797	4000	2.14	7
1 λ. 24.6859213 δ.	1	2	1	0.5	30770	448789	3381	4000	2.17	6
1 λ. 27.2079888 δ.	1	4	2	0.5	30770	462791	3494	4000	2.19	5
1 λ. 24.2568443 δ.	2	4	2	1	30770	393771	3150	4000	2.17	3
1 λ. 20.6218788 δ.	2	6	2	0.5	30770	458268	3036	4000	2.14	5
1 λ. 20.0143644 δ.	1	2	2	1	30770	243812	2515	4000	2.01	6
1 λ. 21.1465716 δ.	1	2	2	0.5	30770	479929	3512	4000	2.3	6
1 λ. 26.73218 δ.	1	4	2	1	30770	478450	3128	4000	2.11	4
1 λ. 28.1626201 δ.	1	6	2	0.5	30770	503410	3240	4000	2.22	3
1 λ. 28.6312133 δ.	1	4	2	1	30770	542151	3315	4000	2.01	6
1 λ. 24.4012792 δ.	2	4	2	1	30770	455289	3883	4000	2.06	5
1 λ. 25.5499787 δ.	2	6	2	1	30770	453668	2899	4000	2.23	4
1 λ. 21.3403746 δ.	1	6	2	1	30770	537628	2368	4000	2.18	3
1 λ. 24.9459464 δ.	1	4	1	1	30770	681010	3165	4000	2.25	4
1 λ. 26.4682314 δ.	2	2	1	1	30770	534990	3357	4000	2.18	3
1 λ. 23.7208568 δ.	1	2	2	1	30770	327367	3177	4000	2.2	9
1 λ. 24.9396326 δ.	2	2	2	0.5	30770	354731	3624	4000	2.17	5
1 λ. 25.0931672 δ.	2	4	2	1	30791	635932	3079	4000	2.13	4
1 λ. 34.2954691 δ.	1	6	1	0.5	30791	23353920	941	4000	0.71	7
1 λ. 30.1984582 δ.	2	4	2	1	30791	420530	3659	4000	2.23	5
1 λ. 26.7275309 δ.	2	6	1	1	30791	627691	3025	4000	2.11	4
1 λ. 21.0325803 δ.	2	2	2	1	30791	332672	3405	4000	2.23	7
1 λ. 21.9798406 δ.	2	4	2	0.5	30848	411989	3263	4000	2.08	2
1 λ. 29.5486436 δ.	1	4	1	0.5	30848	23177120	1120	4000	0.79	5
1 λ. 21.3470284 δ.	1	6	2	1	30869	500850	2243	4000	2.15	7
1 λ. 21.9355148 δ.	1	2	2	1	30869	392448	3040	4000	2.22	8
1 λ. 27.8187189 δ.	2	6	1	1	30869	593372	3085	4000	2.03	7
1 λ. 27.8365177 δ.	1	6	1	0.5	30869	629931	2905	4000	2.17	5
1 λ. 21.8022601 δ.	2	4	2	0.5	30869	491428	3334	4000	2.17	3
1 λ. 29.2957817 δ.	1	4	1	0.5	30869	612011	2913	4000	2.25	8
1 λ. 28.5619645 δ.	1	2	1	1	30869	483470	3198	4000	2.17	6
1 λ. 25.4145945 δ.	2	2	1	0.5	30869	456050	3800	4000	2.0	4
1 λ. 30.9672543 δ.	2	6	1	0.5	30869	504988	3680	4000	2.1	3
1 λ. 21.4466648 δ.	1	2	1	0.5	30869	339890	3105	4000	2.35	3
1 λ. 25.5207475 δ.	2	6	1	0.5	30869	517269	3670	4000	2.14	5
1 λ. 24.3559837 δ.	1	6	2	0.5	30891	566151	3400	4000	2.12	7
1 λ. 34.9156361 δ.	1	2	1	0.5	30891	23282400	936	4000	0.7	4
1 λ. 28.8544525 δ.	1	4	1	1	30891	554411	3000	4000	2.07	5
1 λ. 22.1117893 δ.	1	2	2	0.5	30891	440292	3100	4000	2.18	6
1 λ. 29.5948011 δ.	1	2	2	1	30891	23217760	948	4000	0.71	2
1 λ. 24.7693701 δ.	1	6	2	1	30891	494670	3289	4000	2.08	5
1 λ. 23.6951722 δ.	2	2	1	0.5	30891	523812	3458	4000	2.16	3
1 λ. 31.6207764 δ.	1	6	1	1	30891	23195040	1112	4000	0.85	4
1 λ. 24.4956325 δ.	2	4	2	0.5	30912	501732	3282	4000	2.15	7
1 λ. 27.2749282 δ.	2	6	2	0.5	30912	441593	3683	4000	2.09	3
1 λ. 28.5776603 δ.	2	2	1	1	30969	531548	3772	4000	2.07	3
1 λ. 16.8177582 δ.	1	6	1	1	30969	647708	2931	4000	2.08	4

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 24.6028634 δ.	2	6	2	0.5	30969	406251	3108	4000	2.29	5
1 λ. 21.0351491 δ.	1	6	2	1	30969	523428	2767	4000	2.26	5
1 λ. 22.993426 δ.	2	2	1	0.5	30969	494450	3416	4000	2.33	5
1 λ. 28.4627587 δ.	2	6	1	0.5	30969	616071	3489	4000	2.03	3
1 λ. 25.0372603 δ.	2	6	1	0.5	30969	652011	3311	4000	2.19	2
1 λ. 28.6554928 δ.	2	6	1	1	30969	569952	3046	4000	2.05	2
1 λ. 23.5473251 δ.	2	4	2	1	30969	380089	3784	4000	2.11	6
1 λ. 30.6013222 δ.	1	4	1	1	30969	640128	2971	4000	1.98	5
1 λ. 33.1863103 δ.	2	6	2	0.5	30969	23500960	1141	4000	0.73	2
1 λ. 24.4290359 δ.	2	2	2	1	30969	295609	3805	4000	2.17	6
1 λ. 20.9269688 δ.	1	6	2	1	30969	542208	2500	4000	2.24	4
1 λ. 24.63517 δ.	1	4	1	1	30990	620910	3260	4000	2.25	6
1 λ. 14.4573222 δ.	1	2	1	1	30990	467691	3188	4000	2.2	3
1 λ. 11.434803 δ.	1	6	2	0.5	30990	505170	3011	4000	2.15	5
1 λ. 21.0318682 δ.	2	2	2	1	30990	369828	3108	4000	2.19	4
1 λ. 25.7137791 δ.	2	6	1	0.5	30990	500309	3785	4000	2.15	7
1 λ. 22.968156 δ.	1	2	2	1	30990	393870	3421	4000	2.31	3
1 λ. 34.8049747 δ.	1	2	2	1	30990	23216640	967	4000	0.71	3
1 λ. 23.7957868 δ.	1	4	1	0.5	30990	622990	2778	4000	2.09	11
1 λ. 24.3295784 δ.	1	4	1	1	30990	536331	3313	4000	2.27	4
1 λ. 26.1396554 δ.	1	4	1	0.5	30990	611271	2952	4000	2.04	4
1 λ. 25.6665351 δ.	1	6	2	0.5	30990	594610	3067	4000	2.31	7
1 λ. 25.4648648 δ.	1	6	1	1	30990	485191	3430	4000	2.09	8
1 λ. 29.003462 δ.	1	6	1	1	30990	608249	3081	4000	1.97	7
1 λ. 26.3193275 δ.	1	2	2	0.5	30990	502713	3278	4000	2.16	5
1 λ. 24.6478648 δ.	1	2	1	1	31012	531292	3247	4000	2.12	7
1 λ. 23.9695969 δ.	2	2	1	0.5	31012	441287	3780	4000	2.13	7
1 λ. 22.8951064 δ.	1	4	2	0.5	31068	380331	3129	4000	2.25	6
1 λ. 24.8673094 δ.	2	4	1	0.5	31068	22247840	2980	4000	2.18	9
1 λ. 25.4855269 δ.	1	4	1	1	31068	682752	3414	4000	2.27	4
1 λ. 26.2188524 δ.	1	4	1	0.5	31068	630151	2981	4000	2.07	2
1 λ. 29.0790434 δ.	2	4	1	1	31068	662030	3105	4000	1.97	4
1 λ. 23.3736833 δ.	2	2	2	0.5	31068	389831	3352	4000	2.25	4
1 λ. 23.8278447 δ.	1	4	2	1	31068	488249	3301	4000	2.12	4
1 λ. 28.4725572 δ.	2	6	2	1	31090	438389	3696	4000	1.99	7
1 λ. 26.1237898 δ.	2	4	1	0.5	31090	22286560	2245	4000	1.6	8
1 λ. 21.4576619 δ.	1	4	2	1	31090	523250	2585	4000	2.06	8
1 λ. 22.6198364 δ.	2	2	1	1	31090	356352	3413	4000	2.25	2
1 λ. 22.4929234 δ.	1	2	1	1	31090	526770	3150	4000	2.19	4
1 λ. 26.7506731 δ.	1	4	1	0.5	31090	563652	3110	4000	2.32	7
1 λ. 25.7478496 δ.	1	6	1	1	31090	574692	3191	4000	2.13	8
1 λ. 20.6714351 δ.	1	2	2	1	31090	350891	3076	4000	2.3	6
1 λ. 19.3188083 δ.	1	4	1	0.5	31090	573070	3296	4000	2.15	6
1 λ. 23.2485728 δ.	1	4	2	1	31090	22988800	2247	4000	2.07	6
1 λ. 24.7303309 δ.	1	6	2	0.5	31090	584110	2994	4000	2.1	5
1 λ. 24.3741338 δ.	2	2	1	0.5	31090	423310	3425	4000	2.12	2
1 λ. 27.3697712 δ.	2	2	1	1	31090	491228	3501	4000	2.17	7
1 λ. 23.5061113 δ.	1	2	1	0.5	31090	434731	3015	4000	2.18	6
1 λ. 23.0569816 δ.	1	2	2	0.5	31090	411172	3174	4000	2.3	4
1 λ. 25.1098756 δ.	1	4	2	1	31090	475150	3144	4000	2.06	3
1 λ. 21.3288375 δ.	1	2	1	0.5	31090	440647	3321	4000	2.19	7

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 22.2077511 δ.	1	2	2	1	31090	22744320	2651	4000	1.84	6
1 λ. 27.482675 δ.	2	6	2	1	31090	482290	3445	4000	2.15	3
1 λ. 21.1011702 δ.	1	2	2	0.5	31111	439652	3192	4000	2.16	8
1 λ. 21.6147387 δ.	2	2	1	0.5	31111	577508	3479	4000	2.31	6
1 λ. 24.351253 δ.	1	6	2	1	31111	620110	2849	4000	2.18	3
1 λ. 24.5989178 δ.	2	4	2	1	31111	521209	3330	4000	2.13	5
1 λ. 24.7437568 δ.	1	6	1	0.5	31111	518891	3078	4000	2.17	2
1 λ. 24.9953807 δ.	1	2	2	1	31111	23074400	2012	4000	1.53	8
1 λ. 22.6100289 δ.	2	6	2	1	31111	412629	2591	4000	2.21	5
1 λ. 28.3046956 δ.	2	6	2	1	31168	385671	3884	4000	2.04	3
1 λ. 27.0162717 δ.	1	6	2	0.5	31168	520668	3195	4000	2.12	2
1 λ. 25.4004007 δ.	1	6	1	1	31189	721312	3069	4000	2.11	6
1 λ. 22.6869085 δ.	1	2	1	1	31189	458748	3194	4000	2.14	3
1 λ. 31.0733113 δ.	1	4	1	1	31189	23153760	910	4000	0.72	3
1 λ. 26.0530438 δ.	2	6	2	1	31189	23147040	997	4000	0.71	4
1 λ. 21.2117007 δ.	1	2	2	0.5	31189	442930	3253	4000	2.31	4
1 λ. 25.6157482 δ.	1	4	2	1	31189	404871	3510	4000	2.18	3
1 λ. 23.6322048 δ.	2	6	1	1	31189	570311	3602	4000	2.01	3
1 λ. 28.0228391 δ.	2	4	1	1	31189	554148	3069	4000	2.04	8
1 λ. 25.0883209 δ.	2	4	1	1	31189	522290	3241	4000	2.15	4
1 λ. 26.5878794 δ.	2	6	2	0.5	31189	540128	3185	4000	2.11	6
1 λ. 27.0950773 δ.	1	4	1	1	31189	612850	3215	4000	2.12	3
1 λ. 27.1488555 δ.	1	6	1	1	31189	606891	3012	4000	2.07	5
1 λ. 28.3554096 δ.	2	2	1	1	31189	483470	3637	4000	2.11	4
1 λ. 32.6796339 δ.	2	4	1	0.5	31189	599808	3034	4000	2.08	5
1 λ. 25.4666609 δ.	1	6	1	0.5	31189	529330	3010	4000	2.13	8
1 λ. 26.3767205 δ.	1	6	1	1	31189	821892	2843	4000	1.94	4
1 λ. 22.9767576 δ.	1	6	2	0.5	31189	605010	3313	4000	2.19	5
1 λ. 27.9307791 δ.	2	4	1	0.5	31189	482272	3644	4000	2.09	3
1 λ. 25.4465011 δ.	2	2	1	1	31189	479150	3479	4000	2.22	6
1 λ. 28.8582314 δ.	2	6	2	0.5	31189	425650	3807	4000	2.04	3
1 λ. 27.392564 δ.	1	4	1	1	31189	680690	2959	4000	2.13	4
1 λ. 26.5101939 δ.	2	6	1	1	31189	608711	3452	4000	1.97	3
1 λ. 27.0576426 δ.	2	6	2	0.5	31211	415253	3865	4000	2.16	6
1 λ. 22.7835916 δ.	2	6	1	1	31211	451012	3572	4000	2.25	8
1 λ. 26.1012317 δ.	1	2	2	1	31211	320071	3421	4000	2.21	7
1 λ. 22.6325795 δ.	1	2	1	0.5	31211	411989	3296	4000	2.14	8
1 λ. 25.0415645 δ.	2	2	2	0.5	31211	416533	3893	4000	2.16	7
1 λ. 26.6838201 δ.	2	6	2	1	31289	421732	3604	4000	2.17	7
1 λ. 23.8863397 δ.	1	2	2	0.5	31289	474190	3162	4000	2.26	3
1 λ. 26.5914387 δ.	1	6	1	1	31289	638891	2803	4000	2.11	2
1 λ. 26.0132224 δ.	2	4	2	1	31289	393828	3672	4000	2.13	5
1 λ. 28.0355007 δ.	2	6	2	1	31289	493788	3682	4000	2.06	5
1 λ. 20.9840852 δ.	2	4	2	1	31289	609508	3001	4000	2.15	5
1 λ. 27.2168131 δ.	1	6	1	0.5	31289	542510	2921	4000	2.12	4
1 λ. 26.1602499 δ.	1	6	1	0.5	31289	569692	3015	4000	2.08	2
1 λ. 25.7307366 δ.	1	6	1	0.5	31289	498951	3184	4000	2.34	7
1 λ. 22.076421 δ.	1	6	1	0.5	31289	536811	3142	4000	2.0	5
1 λ. 26.9947398 δ.	2	6	1	1	31310	472391	3829	4000	2.12	5
1 λ. 23.8924373 δ.	2	2	1	1	31310	498631	3320	4000	2.15	5
1 λ. 22.8732437 δ.	1	4	2	0.5	31310	368907	3178	4000	2.2	6

Εφαρμογή Αλγορίθμων Βελτιστοποίησης του Αρχιμήδη και μέσω Σμήνους
Κοτών για τον Χρονοπρογραμματισμό Πληρωμάτων Αεροπορικών Εταιρειών

1 λ. 24.4587793 δ.	2	4	2	1	31310	417230	3828	4000	2.04	8
1 λ. 30.2689889 δ.	2	6	1	1	31310	610610	3010	4000	2.05	6
1 λ. 24.6381852 δ.	1	4	2	1	31310	502891	3376	4000	2.11	2
1 λ. 24.4479496 δ.	2	2	1	1	31310	444011	3459	4000	2.09	5
1 λ. 23.6063744 δ.	1	4	2	1	31310	446770	3081	4000	2.2	8
1 λ. 26.5961402 δ.	2	2	2	1	31310	399111	3663	4000	2.28	8
1 λ. 26.4729392 δ.	1	4	2	0.5	31310	544811	3128	4000	2.06	2
1 λ. 27.1375512 δ.	2	6	1	1	31310	742372	3011	4000	2.02	2
1 λ. 21.0103118 δ.	1	4	1	1	31310	579890	2971	4000	2.13	3
1 λ. 25.3949416 δ.	1	6	1	0.5	31310	565870	3196	4000	2.36	10
1 λ. 24.3744091 δ.	1	6	2	0.5	31310	577771	3561	4000	2.21	6
1 λ. 31.1967904 δ.	2	6	1	1	31310	754951	2925	4000	2.09	6
1 λ. 27.1777057 δ.	2	6	1	0.5	31310	567410	3414	4000	2.05	2
1 λ. 29.8287551 δ.	1	6	2	1	31310	582428	3217	4000	2.08	9
1 λ. 26.1072683 δ.	2	4	2	1	31310	501433	3265	4000	2.11	6
1 λ. 27.0780327 δ.	2	2	1	0.5	31310	424292	3841	4000	2.01	4
1 λ. 24.9914494 δ.	1	6	1	1	31310	611271	3178	4000	2.09	6
1 λ. 24.1014818 δ.	1	6	2	1	31310	528288	3400	4000	2.12	7
1 λ. 25.0337546 δ.	2	2	1	1	31310	491051	3216	4000	2.1	5
1 λ. 24.1637719 δ.	1	2	2	1	31332	393188	2965	4000	2.24	6
1 λ. 23.0918557 δ.	1	2	1	1	31332	515851	3157	4000	2.14	5
1 λ. 13.1170485 δ.	2	6	2	0.5	31367	427591	3879	4000	2.15	4
1 λ. 29.0151066 δ.	2	4	1	0.5	31388	463289	3848	4000	2.02	6
1 λ. 30.2275245 δ.	1	6	1	0.5	31388	23123520	961	4000	0.7	7
1 λ. 18.8697559 δ.	1	4	1	1	31388	613511	3056	4000	2.2	5
1 λ. 23.4371803 δ.	1	6	2	1	31388	436807	3485	4000	2.2	5
1 λ. 26.5663892 δ.	2	6	2	0.5	31388	482212	3828	4000	2.07	3
1 λ. 24.0278975 δ.	2	4	2	1	31388	443349	3081	4000	2.1	8
1 λ. 27.7471334 δ.	2	6	2	0.5	31410	434268	3818	4000	2.02	2
1 λ. 14.1126569 δ.	2	2	2	1	31410	317611	3721	4000	2.19	5
1 λ. 24.2708471 δ.	1	4	2	1	31410	518009	3159	4000	2.12	2
1 λ. 22.8549898 δ.	2	2	2	0.5	31410	384213	3256	4000	2.09	7
1 λ. 24.9193169 δ.	1	6	1	0.5	31410	23118720	2034	4000	1.44	4
1 λ. 24.5384516 δ.	1	6	2	0.5	31410	554948	3065	4000	2.05	4
1 λ. 23.8325431 δ.	1	2	1	1	31410	520533	3117	4000	2.12	4
1 λ. 24.0545712 δ.	1	6	2	1	31410	570489	2921	4000	2.16	6
1 λ. 24.1802164 δ.	2	4	1	0.5	31410	550372	3086	4000	2.19	5
1 λ. 28.1641953 δ.	1	4	1	0.5	31410	548789	3087	4000	2.15	7
1 λ. 28.3305047 δ.	1	6	1	1	31410	763868	3007	4000	2.07	4
1 λ. 24.1834015 δ.	1	6	2	0.5	31410	526891	2984	4000	2.13	5
1 λ. 22.2436623 δ.	2	2	2	1	31410	331008	3553	4000	2.21	3
1 λ. 29.8659292 δ.	2	4	1	0.5	31410	457408	3771	4000	1.91	4
1 λ. 28.7227503 δ.	2	6	2	1	31410	408171	3633	4000	2.08	5
1 λ. 26.6744815 δ.	1	4	2	1	31410	515570	2880	4000	2.14	3
1 λ. 22.959922 δ.	2	4	2	1	31410	564452	3578	4000	2.11	5
1 λ. 22.1241949 δ.	1	6	1	1	31431	457810	3415	4000	2.19	5
1 λ. 23.0854352 δ.	1	6	1	1	31431	583289	3370	4000	2.36	4
1 λ. 26.2039241 δ.	2	2	2	1	31488	432110	3626	4000	2.18	4
1 λ. 32.745762 δ.	1	4	2	0.5	31488	23241920	947	4000	0.69	4
1 λ. 24.1220828 δ.	1	4	1	0.5	31488	510812	3163	4000	2.2	3

ΑΝΑΦΟΡΕΣ

- [1] Xianbing Meng, Yu Liu Xiaozhi Gao and Hengzhen Zhang, “A New Bio-inspired Algorithm: Chicken Swarm Optimization”, in International Conference in Swarm Intelligence Oct. 2014, pp. 86-94, doi: 10.1007/978-3-319-11857-4_10.
- [2] Sanchari Deb, Xiao-Zhi Gao, Kari Tammi, Karuna Kalita, Pinakeswar Mahanta, “Recent Studies on Chicken Swarm Optimization algorithm: a review (2014–2018)”, Springer Nature B.V. 2019 , 23 May 2019.
- [3] Nursyiva Irsalinda, Aris Thobirin, Dian Eka Wijayanti, “Chicken Swarm as a Multi Step Algorithm for Global Optimization”, International Journal of Engineering Science Invention, Vol. 6 Issue 1, Jan. 2017.
- [4] Marino, L., "Thinking chickens: a review of cognition, emotion, and behavior in the domestic chicken", Anim Cogn 20, 127–147, 2017.
- [5] Fatma A. Hashim, Kashif Hussain, Essam H. Houssein, Mai S. Mabrouk, Walid Al-Atabany, “Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems”, Springer Science+Business Media, LLC, 29 Sept. 2020.
- [6] Niklas Kohl, Stefan E. Karisch, “Airline Crew Rostering: Problem Types, Modeling, and Optimization”, Annals of Operations Research 127, 223–257, Mar. 2004.
- [7] Markus Thiel. “Team-oriented airline crew scheduling and rostering : problem description, solution approaches, and decision support”, Doctoral Thesis, Faculty of Business Administration and Economics, University of Paderborn, Germany, Jun. 2018.
- [8] George A. Chailazopoulos, “Ant colony optimization algorithms for airline crew rostering”, Πτυχιακή Εργασία, Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, 2018.
- [9] Atoosa Kasirzadeh, Mohammed Saddoune, Francois Soumis, “Airline crew scheduling: models, algorithms, and data sets”, EURO J Transp Logist, 2017.
- [10] Zhiqiang Tian, Huimin Niu, “Modeling and Algorithms of the Crew Rostering Problem with Given Cycle on High-Speed Railway Lines”, Mathematical Problems in Engineering, vol. 2012, Oct. 2012.
- [11] Alberto Caprara, Paolo Toth, Daniele Vigo, Matteo Fischetti, “Modeling and Solving the Crew Rostering Problem”, Operations Research, vol. 46, no. 6, pp. 820-830, 1 Dec. 1998.