



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ
ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη Μεθοδολογίας Εκτίμησης Ελαχίστων Σημείων του Εύρους
Αντικειμενικών Συναρτήσεων σε Προβλήματα Πολυστοχικού
Προγραμματισμού υπό Αβεβαιότητα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιώτης-Φίλιππος Παπαβασιλόπουλος

Επιβλέπων: **Χρυσόστομος (Χάρης) Δούκας**
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ
ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη Μεθοδολογίας Εκτίμησης Ελαχίστων Σημείων του Εύρους
Αντικειμενικών Συναρτήσεων σε Προβλήματα Πολυστοχικού
Προγραμματισμού υπό Αβεβαιότητα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιώτης-Φίλιππος Παπαβασιλόπουλος

Επιβλέπων: **Χρυσόστομος (Χάρης) Δούκας**
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16^η Ιουλίου 2024.

.....
Χρυσόστομος Δούκας
Καθηγητής Ε.Μ.Π.

.....
Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

.....
Ευάγγελος Μαρινάκης
Επ. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024

.....
Παπαβασιλόπουλος Παναγιώτης-Φίλιππος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Παπαβασιλόπουλος Παναγιώτης-Φίλιππος, 2024
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσοβίου Πολυτεχνείου.

Περίληψη

Ο πολυστοχικός προγραμματισμός είναι αρκετά διαδεδομένος στον τομέα της ενέργειας και πραγματοποιούνται διαρκώς μελέτες για την ανάπτυξη μεθόδων για την επίλυση τέτοιου τύπου προβλημάτων. Από την πληθώρα αυτών εστιάζουμε στην οικογένεια μεθόδων AUGMECON και συγκεκριμένα στην AUGMECON-Py. Η AUGMECON-Py διακριτοποιεί τον χώρο εφικτών λύσεων κάθε αντικειμενικής συνάρτησης και βρίσκει τις κατά Pareto βέλτιστες λύσεις με αρκετά αποδοτικό τρόπο. Ωστόσο, επειδή εκφράζονται όλες οι αντικειμενικές συναρτήσεις εκτός από μία ως περιορισμοί του πολυστοχικού προβλήματος μπορεί η αλλαγή της θέσης τους στο πρόβλημα, να επηρεάσει τον χρόνο εκτέλεσης του αλγορίθμου και τον αριθμό των βέλτιστων κατά Pareto λύσεων που θα βρεθούν. Τα παραπάνω επηρεάζονται καθοριστικά και από την εκτίμηση των ελαχίστων τιμών -για τις οποίες υπάρχει αβεβαιότητα- κάθε αντικειμενικής συνάρτησης καθώς το εύρος του χώρου εφικτών λύσεών τους θα αλλάζει.

Τα σημεία nadir μέχρι στιγμής εκτιμούνται πολλαπλασιάζοντας με έναν συντελεστή τις ελάχιστες τιμές κάθε αντικειμενικής συνάρτησης έπειτα από την λεξικογραφική βελτιστοποίησή τους, τον “nadir_cutoff”. Ιδανικά οι εκτιμώμενες τιμές των σημείων nadir θα πρέπει να είναι μικρότερες ή ίσες από τις πραγματικές έτσι ώστε να μην χαθεί καμία βέλτιστη λύση. Η πρόκληση έγκειται στην εύρεση ενός συντελεστή που θα είναι αρκετά μικρός ώστε να περιλαμβάνονται όλες οι λύσεις αλλά όχι τόσο ώστε να αυξάνεται σημαντικά ο χρόνος εκτέλεσης του προγράμματος.

Στην παρούσα εργασία ορίζεται μια μεθοδολογία αποτελεσματικής εκτίμησης του συντελεστή nadir_cutoff που ονομάστηκε 2-Stage AUGMECON-Py και έπειτα εφαρμόζεται σε ενεργειακά προβλήματα εύρεσης βέλτιστων χαρτοφυλακίων χρηματοδότησης τεχνολογιών χαμηλών εκπομπών με πραγματικά δεδομένα. Η νέα μέθοδος χρησιμοποιεί μέρος των δεδομένων του προβλήματος για να εκτιμήσει τον συντελεστή nadir_cutoff με αρκετά μεγάλη ακρίβεια. Η μέθοδος αυτή ελαχιστοποιεί την πιθανότητα απώλειας λύσεων και διαθέτει σημαντικά χρονικά πλεονεκτήματα ιδιαίτερα όταν γίνεται ανάλυση Monte Carlo του προβλήματος.

Λέξεις κλειδιά: Πολυκριτηριακή Ανάλυση Αποφάσεων, AUGMECON-Py, Αντικειμενικές συναρτήσεις, σημείο nadir, ενεργειακή πολιτική, ανανεώσιμες πηγές ενέργειας

Abstract

Multi-objective programming is widely used in the energy sector, with ongoing research dedicated to developing methods for solving such problems. Among these methods, we focus on the AUGMECON family, particularly AUGMECON-Py. AUGMECON-Py discretizes the feasible solution space of each objective function, efficiently finding Pareto optimal solutions. However, since all but one of the objective functions are expressed as constraints, the algorithm's runtime and the number of Pareto optimal solutions can be influenced by the order of these functions in the program formulation. Additionally, the estimation of the minimum values—uncertain by nature—of each objective function impacts the range of their feasible solution space.

Traditionally, these nadir values have been estimated by multiplying the minimum value of each objective function after lexicographic optimization by a specific coefficient, known as the “nadir_cutoff”. Ideally, these estimated nadir values should be less than or equal to the actual values to ensure no optimal solutions are missed. The challenge lies in finding a coefficient that is small enough to capture all Pareto optimal solutions without significantly increasing runtime.

This diploma thesis presents a specific methodology for efficiently estimating the nadir_cutoff, termed 2-Stage AUGMECON-Py. This method is applied to energy problems related to the portfolio analysis of subsidies for renewable energy technologies. The new method leverages part of the problem dataset to precisely estimate the nadir_cutoff coefficient. This approach minimizes the likelihood of missing solutions and offers significant runtime advantages, particularly when conducting a Monte Carlo analysis.

Key words: Multi-Criteria Decision Analysis, AUGMECON-Py, objective functions, nadir point, energy policy, renewable energy

Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους τους ανθρώπους που έχουν συμβάλει στη μέχρι τώρα πορεία μου και να τους αφιερώσω την εκπόνηση αυτής της διπλωματικής εργασίας.

Κατ' αρχήν τον καθηγητή μου και επιβλέποντα την παρούσα διπλωματική εργασία κ. Χρυσόστομο (Χάρη) Δούκα που με τη διδασκαλία του και τις εποικοδομητικές συζητήσεις που είχαμε με εισήγαγε σε πολύ ενδιαφέρουσες επιστημονικές περιοχές που επηρέασαν καθοριστικά την ακαδημαϊκή μου πορεία. Επίσης, για τις πολύτιμες συμβουλές που μου έδωσε σχετικά με τη συνέχεια των σπουδών μου.

Ιδιαίτερες ευχαριστίες στον Δρ. Κωνσταντίνο Κοασίδη για την πολύτιμη βοήθεια που μου προσέφερε και τον χρόνο που αφιέρωσε σε όλη τη διάρκεια της διπλωματικής εργασίας. Το αποτέλεσμα δεν θα ήταν το ίδιο χωρίς τις παραγωγικές συζητήσεις μας και τις εύστοχες παρατηρήσεις του και για τον λόγο αυτόν αισθάνομαι τυχερός και ευγνώμων που συνεργάστηκα μαζί του.

Τα μέλη ΔΕΠ του Εργαστηρίου Συστημάτων και Αποφάσεων Διοίκησης που με τη διδασκαλία τους και το πάθος τους για το αντικείμενο κατάφεραν να διεγείρουν ακόμα περισσότερο το ενδιαφέρον μου για τον τομέα της επιχειρησιακής έρευνας.

Τον Δρ. Αλέξανδρο Νίκα για τη βοήθεια και καθοδήγησή του.

Τους καθηγητές μου στο Πολυτεχνείο που μου έδωσαν πολύτιμες γνώσεις και με έκαναν να αγαπήσω πολλές επιστημονικές περιοχές.

Τους καθηγητές μου κ.κ. Ι. Ψαρρά και Ε. Μαρινάκη που δέχθηκαν να είναι μέλη της τριμελούς επιτροπής της παρούσης διπλωματικής εργασίας.

Τους φίλους μου και τους ανθρώπους που γνώρισα στο Πολυτεχνείο που έκαναν τα χρόνια των σπουδών μου μία πολύ ευχάριστη εμπειρία που θα μείνει για πάντα στην καρδιά μου.

Την οικογένειά μου, που είναι πάντα δίπλα μου και με στηρίζει σε κάθε προσπάθειά μου, την αδελφή μου, τη μητέρα μου, τον πατέρα μου και τη γιαγιά μου.

*Παπαβασιλόπουλος Παναγιώτης-Φίλιππος,
Αθήνα Ιούλιος 2024*

Πίνακας Περιεχομένων

Κεφάλαιο 1. Εισαγωγή	17
1.1 Αντικείμενο-Στόχος	19
1.2 Δομή Εργασίας.....	21
 Κεφάλαιο 2. Θεωρητικό Υπόβαθρο Μεθοδολογιών Πολυστοχικού Προγραμματισμού.....	22
2.1 Βελτιστοποίηση (Optimization).....	24
2.1.1 Γενικότερα για την Βελτιστοποίηση	24
2.1.2 Βελτιστοποίηση με ένα μόνο κριτήριο.....	25
2.1.3 Πολυκριτηριακή Ανάλυση αποφάσεων (Multi-Criteria Decision Analysis - MCDA).....	25
2.2 Η έννοια του Pareto.....	27
2.3 Πολυκριτήρια Ανάλυση Αποφάσεων για $p=2$	27
2.4 Μετατροπή Πολυκριτηριακού Προβλήματος σε Περισσότερα Προβλήματα Ενός Κριτηρίου (Scalarization).....	30
2.5 Μέθοδος Σταθμισμένου Αθροίσματος (Weighted Sum)	32
2.6 Μέθοδος ϵ -constraint	36
2.7 Παραλλαγές της μεθόδου ϵ -constraint	40
2.7.1 Augmented ϵ -constraint method (AUGMECON)	40
2.7.2 Μέθοδος AUGMECON2	43
2.7.3 AUGMECON-R	44
2.7.4 AUGMECON-Py.....	45
2.8 Μέθοδοι Βελτιστοποίησης σε Ενεργειακά Προβλήματα.....	46
 Κεφάλαιο 3. Ελάχιστα (ναδίρ) Σημεία Εύρους Αντικειμενικών σε Προβλήματα Πολυστοχικού Προγραμματισμού και Νέα Μεθοδολογία Εκτίμησής τους.....	49
3.1 Απώλεια Λύσεων και Χρονική Επιβάρυνση από την Επιλογή Ναδίρ Σημείων	51
3.2 Ανάλυση AUGMECON-Py - Εκτιμητής nadir cutoff c_i	52
3.2.1 Εμπειρική Ανάδειξη Προκλήσεων Εύρεσης Κατάλληλων Σημείων Ναδίρ και Σειράς Αντικειμενικών	52
3.3 Βελτίωση μεθόδου AUGMECON-Py για την Εύρεση Κατάλληλων Σημείων Ναδίρ	59
3.4 Ενσωμάτωση της Μεθοδολογίας στο Εργαλείο της AUGMECON-Py.....	62
3.5 Πιθανές Μελλοντικές Βελτιώσεις της AUGMECON-Py ως Πρόγραμμα H/Y	62

Κεφάλαιο 4. Εφαρμογή της Μεθοδολογίας σε Ενεργειακά Προβλήματα Εύρεσης Βέλτιστων Χαρτοφυλακίων Χρηματοδότησης Τεχνολογιών Χαμηλών Εκπομπών.....	66
4.1 Εφαρμογή της Νέας Μεθοδολογίας σε ένα Υπολογιστικά Πολύπλοκο Πρόβλημα Ενεργειακής Βελτιστοποίησης.....	68
4.1.1: Πειραματική Μελέτη Ενεργειακού Προβλήματος με το 10% των Αρχικών Δεδομένων.....	68
4.1.2: Πειραματική Μελέτη Ενεργειακού Προβλήματος με το 20% των Αρχικών Δεδομένων.....	73
4.2 Εφαρμογή της Μεθοδολογίας στο Ενεργειακό Πρόβλημα με Αβεβαιότητα ως προς τη Μορφή των Δεδομένων	75
 Κεφάλαιο 5. Συμπεράσματα και Προοπτικές	82
5.1 Συνολικά Συμπεράσματα	84
5.2 Γενικά Συμπεράσματα για τη Βελτιστοποίηση.....	86
5.3 Προοπτικές Εργασίας.....	87
 Βιβλιογραφία	89
 Παράρτημα.....	95
Κώδικας για την Επιλογή Μέρους των Δεδομένων.....	97
Κώδικας για την Εύρεση της Ελάχιστης Τιμής Pareto λύσης για κάθε Αντικειμενική Συνάρτηση.....	98

Ευρετήριο Γραφημάτων

Γράφημα 2.1: Αναπαράσταση των τριών κριτηρίων δύο αντικειμενικών συναρτήσεων	28
Γράφημα 2.2: Χώρος εφικτών λύσεων δύο συναντήσεων f_1, f_2	28
Γράφημα 2.3: Σχηματική αναπαράσταση μετώπου Pareto	29
Γράφημα 2.4: Εύρεση μετώπου Pareto μέσω των λύσεων των υποπροβλημάτων	31
Γράφημα 2.5: Η περίπτωση που η λύση του υποπροβλήματος δεν ανήκει στο μέτωπο Pareto	32
Γράφημα 2.6: Οι παράλληλες γραμμών που προκύπτουν από συγκεκριμένες τιμές των βαρών	33
Γράφημα 2.7: Ουτοπικό σημείο βελτιστοποίησης	34
Γράφημα 2.8: Ανομοιογενής συγκέντρωση λύσεων	34
Γράφημα 2.9: Μη-κυρτό feasible set με α) Απώλεια Pareto λύσεων β) Αδυναμία εύρεσης περεταίρω λύσεων μεταξύ των σημείων P και Q	35
Γράφημα 2.10: Αδυναμία εύρεσης των λύσεων ακόμα και αν ξεκινήσει η διαδικασία από το σημείο T	36
Γράφημα 2.11: α) Αναζήτησή σημείου Q με $\varepsilon=\varepsilon_1$, β) Εύρεση σημείου Q	38
Γράφημα 2.12: α) Εύρεση σημείου T με $\varepsilon=\varepsilon_2$, β) Εύρεση σημείου P με $\varepsilon=\varepsilon_3$, γ) Εύρεση σημείου F1 με $\varepsilon=\varepsilon_4$	39
Γράφημα 2.13: Αύξηση του ε ομοιογενώς και παρατήρηση των λύσεων	39
Γράφημα 2.14: Μέθοδος ε -constraint α) χωρίς λεξικογραφική βελτιστοποίηση β) με λεξικογραφική βελτιστοποίηση	41
Γράφημα 3.1: Αναπαράσταση Pareto λύσεων του πρώτου πειράματος για nadir cutoff 0.4	54
Γράφημα 3.2: Αναπαράσταση Pareto λύσεων πρώτου πειράματος για nadir cutoff 0.66	55
Γράφημα 3.3: Βελτίωση μεθόδου AUGMECON-Py	61
Γράφημα 4.1: Αναπαράσταση Pareto λύσεων πρώτου πειράματος για nadir cutoff 0.1	70
Γράφημα 4.2: Αναπαράσταση Pareto λύσεων πρώτου πειράματος για nadir cutoff 0.96	71
Γράφημα 4.3: Αναπαράσταση Pareto λύσεων δεύτερου πειράματος για nadir cutoff 0.1	75
Γράφημα 4.4: Αναπαράσταση Pareto λύσεων δεύτερου πειράματος για nadir cutoff 0.95	75

Γράφημα 4.5: Αναπαράσταση Pareto λύσεων πειράματος τυχαιοποιημένων δεδομένων για πολύ μικρό nadir_cutoff = 0.1	77
Γράφημα 4.6: Οι λύσεις που έδωσαν τα κάτω όρια για τον υπολογισμό του cmin στο πρώτο στάδιο με το 50% των δεδομένων.....	79
Γράφημα 4.7: Οι 136 κατά Pareto βέλτιστες λύσεις μετά το δεύτερο στάδιο με nadir_cutoff=0.9	79
Γράφημα 5.1: Χρονικό Μειονέκτημα (%) με την Χρήση της AUGMECON-Py έναντι της 2-Stage AUGMECON-Py	86

Ευρετήριο Πινάκων

Πίνακας 3.1: Εναλλαγή σειράς αντικειμενικών με nadir cutoff 0.1	53
Πίνακας 3.2: Εναλλαγή σειράς αντικειμενικών με nadir cutoff 0.4	54
Πίνακας 3.3: Εναλλαγή σειράς αντικειμενικών με nadir cutoff 0.66	55
Πίνακας 3.4: Payoff Table για nadir_cutoff = 0.4	56
Πίνακας 3.5: Μέγιστες Τιμές	56
Πίνακας 3.6: Εκτίμηση Σημείων Ναδír με Συντελεστή nadir_cutoff = 0.4	56
Πίνακας 3.7: Εκτίμηση Σημείων Ναδír με Συντελεστή nadir_cutoff = 0.66	56
Πίνακας 3.8: Εκτίμηση Σημείων Ναδír με Συντελεστή nadir_cutoff = 0.7	56
Πίνακας 3.9: Payoff Table με Σειρά Αντικειμενικών F1, F2, F3 (nadir_cutoff = 0.4) 57	
Πίνακας 3.10: Μέγιστες Τιμές	57
Πίνακας 3.11: Εκτίμηση Σημείων Ναδír με Συντελεστή nadir_cutoff = 0.4	57
Πίνακας 3.12: Εκτίμηση Σημείων Ναδír με Συντελεστή nadir_cutoff = 0.66	57
Πίνακας 3.13: Εκτίμηση Σημείων Ναδír με Συντελεστή nadir_cutoff = 0.7	58
Πίνακας 4.1: Payoff Table για nadir_cutoff = 0.4	68
Πίνακας 4.2: Μέγιστες Τιμές	69
Πίνακας 4.3: Δοκιμές για Nadir Cutoff 0.96	70
Πίνακας 4.4: Σύγκριση χρόνου εκτέλεσης 2-Stage AUGMECON-Py	72
Πίνακας 4.5: Σύγκριση χρόνου εκτέλεσης 2-Stage AUGMECON-Py με Monte Carlo 72	
Πίνακας 4.6: Ποσοστό Αύξησης Χρόνου με την Χρήση Μόνο της AUGMECON-Py 72	
Πίνακας 4.7: Payoff Table για το 20% των Δεδομένων (nadir_cutoff = 0.1)	73
Πίνακας 4.8: Ελάχιστες τιμές αντικειμενικής από Pareto Λύσεις	73
Πίνακας 4.9: Συντελεστές c_i	73
Πίνακας 4.10: Σύγκριση χρόνων σταδίου 1 (nadir_cutoff = 0.1)	74
Πίνακας 4.11: Payoff Table Πειράματος με Τυχαίοποιημένα Δεδομένα	76
Πίνακας 3.12: Ελάχιστες τιμές αντικειμενικής από Pareto Λύσεις	76
Πίνακας 4.13: Συντελεστές c_i	76
Πίνακας 4.14: Αποτελέσματα με nadir cutoff 0.1	77

Πίνακας 4.15: Αποτελέσματα με nadir cutoff 0.84	77
Πίνακας 4.16: Payoff Table από το 50% των δεδομένων	78
Πίνακας 4.17: Ελάχιστες τιμές αντικειμενικής από Pareto Λύσεις.....	78
Πίνακας 4.18: Σύγκριση 2-Stage AUGMECON-Py.....	80
Πίνακας 4.19: Σύγκριση 2-Stage AUGMECON-Py σε Monte Carlo Ανάλυση	80
Πίνακας 4.20: Σύγκριση χρόνου εκτέλεσης 2-Stage AUGMECON-Py.....	80
Πίνακας 4.21: Σύγκριση χρόνου εκτέλεσης 2-Stage AUGMECON-Py με Monte Carlo	80
Πίνακας 4.22: Ποσοστά Αύξησης Χρόνου Εκτέλεσης 2-Stage AUGMECON-Py με Monte Carlo	81

Κεφάλαιο 1.

Εισαγωγή

1.1 Αντικείμενο-Στόχος

Ζούμε σε έναν κόσμο που πολλοί φυσικοί πόροι που χρησιμοποιούμε καθημερινά είναι περιορισμένοι ενώ η ζήτησή τους διαρκώς αυξάνεται με την αύξηση του πληθυσμού. Η στροφή σε εναλλακτικές μορφές ενέργειας συμβάλλει στη βελτίωση της κατάστασης ωστόσο η άμεση μετάβαση σε αυτές δεν είναι απλή υπόθεση καθώς εξαρτάται από πολλούς παράγοντες ιδίως οικονομικούς και κοινωνικούς. Επομένως, είναι κρίσιμο ο άνθρωπος να εκμεταλλεύεται τους πόρους που διαθέτει με μεγάλη αποδοτικότητα προκειμένου η εξάλειψη τους να αποφευχθεί ή τουλάχιστον να καθυστερήσει επαρκώς έτσι ώστε να έχουν προλάβει να διαμορφωθούν κατάλληλα οι υποδομές, οι κοινωνικές πολιτικές, το νομοθετικό πλαίσιο, η οικονομία και η κοινωνία για τις εναλλακτικές μορφές ενέργειας. Ουσιαστικά πρόκειται για προβλήματα βελτιστοποίησης που μπορούν να επιλυθούν με μαθηματική ακρίβεια δίνοντας τις καλύτερες λύσεις. Εξαιτίας της πολυδιάστατης φύσεως των προβλημάτων ενδείκνυται συνεργασία ειδικών στην οικονομία, στο περιβάλλον, στην ενέργεια, στα μαθηματικά και στον προγραμματισμό για την εύρεση μεθόδων επίλυσης και την ανάλυση των αποτελεσμάτων των προβλημάτων.

Ο πολυστοχικός προγραμματισμός υπό αβεβαιότητα στον τομέα της ενέργειας είναι αρκετά διαδεδομένος και πραγματοποιούνται διαρκώς μελέτες για την ανάπτυξη των μεθόδων που χρησιμοποιούνται για την επίλυση προβλημάτων στον τομέα αυτόν και στην παρούσα διπλωματική εργασία αναφέρονται και αναλύονται ορισμένες από αυτές. Υπάρχουν σημαντικές διαφορές μεταξύ των μεθόδων αυτών διαθέτοντας προτερήματα και μειονεκτήματα ανάλογα με το είδος του προβλήματος, με διαφορές μεταξύ τους ως προς την ορθότητα και την πληρότητα των λύσεων αλλά και την χρονική τους πολυπλοκότητα όταν εκτελούνται ως προγράμματα σε ηλεκτρονικό υπολογιστή. Η παρούσα διπλωματική εργασία δίνει βάση στη μέθοδο πολυκριτηριακού προγραμματισμού AUGMECON-Py (Forouli et al, 2022) και έχει ως σκοπό να αναπτύξει τη μεθοδολογία εκτίμησης των ελαχίστων σημείων (σημείων ναδύρ) του εύρους των αντικειμενικών συναρτήσεων και να αναλύσει το χρονικό όφελος από χρήση της σε προβλήματα πολυστοχικού προγραμματισμού υπό αβεβαιότητα.

Η εύρεση βέλτιστων κατά Pareto λύσεων σε προβλήματα πολυκριτηρίας ανάλυσης αποφάσεων είναι πολλές φορές μία σύνθετη διαδικασία και η επιλογή μίας μεθόδου που θα βρίσκει όλες τις αυτές τις λύσεις είναι καθοριστικής σημασίας για τον αποφασίζοντα. Η AUGMECON-Py παρουσιάζει σημαντική υπεροχή έναντι αντίστοιχων μεθόδων σε αυτή την επιστημονική περιοχή. Τα σημαντικότερα ζητούμενα που τέθηκαν στην παρούσα εργασία ήταν ο χαμηλός χρόνος εκτέλεσης αλλά και η εύρεση όλων των κατά Pareto βέλτιστων λύσεων. Όπως παρατηρήθηκε έπειτα από τα πειράματα, η σειρά των αντικειμενικών συναρτήσεων στο πολυκριτηριακό πρόβλημα βελτιστοποίησης επηρεάζει σημαντικά το σύνολο των λύσεων. Επιβεβαιώθηκε ότι η φθίνουσα ταξινόμηση των αντικειμενικών συναρτήσεων σύμφωνα με τα εύρη τους έπειτα από λεξικογραφική βελτιστοποίηση,

ήταν ένας αποτελεσματικός τρόπος να αποφασιστεί ποια αντικειμενική συνάρτηση θα είναι η αντικειμενική του προβλήματος και με ποια σειρά θα βρίσκονται στους περιορισμούς οι υπόλοιπες.

Ένα μεγάλο μέρος της εργασίας αφιερώθηκε στην εκτίμηση της χειρότερης τιμής που μπορεί να πάρει η κάθε αντικειμενική (σημείο ναδέρ). Η πραγματική τιμή του σημείου ναδέρ όπου είναι δύσκολο να υπολογιστεί οπότε γίνεται εκτίμησή του ως ποσοστό της χειρότερης τιμής του payoff table για κάθε αντικειμενική έπειτα από λεξικογραφική βελτιστοποίηση των αντικειμενικών συναρτήσεων. Αυτό το ποσοστό ονομάστηκε `nadir_cutoff` και ακόμα και μετά από αυτό δεν ήταν σίγουρο πως θα ήταν μικρότερη εκτιμώμενη τιμή του σημείου ναδέρ από την πραγματική του τιμή. Μεγαλύτερη πληρότητα των λύσεων μπορούσε να εξασφαλιστεί όταν η παράμετρος `nadir_cutoff` μειωνόταν σημαντικά όμως επέφερε σημαντική αύξηση του χρόνου εκτέλεσης του προγράμματος. Όταν η παράμετρος `nadir_cutoff` ξεπερνούσε ένα συγκεκριμένο όριο τότε ο χώρος εφικτών λύσεων της συνάντησης μειωνόταν με αποτέλεσμα να χάνονται λύσεις, ενώ όταν ήταν πολύ μικρότερος τότε πραγματοποιούνταν πολύ περισσότερες επαναλήψεις στο πρόγραμμα με αποτέλεσμα την αύξηση του χρόνου.

Ο προσδιορισμός αυτού του ορίου, που ονομάστηκε c_{min} , δεν μπορούσε να εκτιμηθεί εκ των προτέρων από τον χρήστη του προγράμματος. Ωστόσο, με την μελέτη μέρους των δεδομένων του προβλήματος μπορεί αυτός να προσεγγιστεί σε ικανοποιητικό βαθμό μελετώντας ένα μικρότερο δείγμα των δεδομένων με τρόπο που ωστόσο θα είναι αντιπροσωπευτικό του συνόλου των δεδομένων. Η μέθοδος 2-Stage AUGMECON-Py όπως ονομάζεται στην παρούσα διπλωματική εργασία εφαρμόζει την παραπάνω ιδέα.

Επειδή τα ενεργειακά προβλήματα που μελετώνται γενικότερα έχουν δεδομένα που προέρχονται από τον πραγματικό κόσμο και διαθέτουν μία συγκεκριμένη δομή, η 2-Stage AUGMECON-Py κατάφερε να κάνει τη διαφορά πετυχαίνοντας πολύ καλούς χρόνους εκτέλεσης. Ιδιαίτερα όταν είναι επιθυμητή και η Monte Carlo ανάλυση του προβλήματος για λόγους robustness, οι χρόνοι που καταγράφονταν με την χρήση της 2-Stage AUGMECON-Py παρουσίαζαν μεγάλη διαφορά με άλλες εκτελέσεις του προγράμματος όπου ο χρήστης χρησιμοποιεί μία δική του εκτίμηση για τον συντελεστή `nadir_cutoff`. Ειδικότερα στα πειράματα που πραγματοποιήθηκαν με δεδομένα που είχαν διαφορετική κατανομή από τα συνήθη ενεργειακά προβλήματα το χρονικό όφελος ήταν ακόμα μεγαλύτερο.

1.2 Δομή Εργασίας

Η διπλωματική εργασία έχει την ακόλουθη δομή. Αρχικά, στο Κεφάλαιο 2, παρουσιάζονται οι βασικές θεωρητικές γνώσεις που απαιτούνται για την μελέτη του προβλήματος, γενικότερα για τον τομέα της βελτιστοποίησης, του γραμμικού προγραμματισμού και του πολυστοχικού προγραμματισμού. Ακόμα εξετάζονται και άλλοι αλγόριθμοι σχετικοί με αυτόν της μεθόδου AUGMECON-Py. Έπειτα, στο Κεφάλαιο 3 αναλύεται περισσότερο η μέθοδος AUGMECON-Py και προτείνεται μία βελτίωση της σχετικά με την αποτελεσματικότερη εκτίμηση των σημείων ναδύρ. Στη συνέχεια, στο Κεφάλαιο 4, παρουσιάζονται τα πειράματα που έγιναν με βάση αυτή τη βελτίωση και τα πορίσματα που εξάγονται άμεσα από αυτά. Στο Κεφάλαιο 5 αναφέρονται τα γενικά συμπεράσματα και στο τέλος είναι οι πηγές και η βιβλιογραφία. Μία μελλοντική κατεύθυνση της παρούσας εργασίας είναι η εκτέλεση ενός case study με τη χρήση της μεθόδου AUGMECON-Py και η αξιολόγηση των αποτελεσμάτων.

Κεφάλαιο 2.

Θεωρητικό Υπόβαθρο Μεθοδολογιών Πολυστοχικού Προγραμματισμού

2.1 Βελτιστοποίηση (Optimization)

2.1.1 Γενικότερα για την Βελτιστοποίηση

Η βελτιστοποίηση, επίσης γνωστή ως μαθηματική βελτιστοποίηση ή μαθηματικός προγραμματισμός, είναι μια περιοχή των εφαρμοσμένων μαθηματικών και της επιχειρησιακής έρευνας που ασχολείται με τον εντοπισμό της καλύτερης δυνατής λύσης ανάμεσα σε όλες τις διαφορετικές εναλλακτικές για ένα συγκεκριμένο πρόβλημα. Στόχος της βελτιστοποίησης είναι να μεγιστοποιήσει ή να ελαχιστοποιήσει μια συγκεκριμένη συνάρτηση-στόχο (αντικειμενική συνάρτηση), ικανοποιώντας παράλληλα ένα σύνολο περιορισμών.

Σε απλούστερους όρους, η βελτιστοποίηση αποσκοπεί στον εντοπισμό της "βέλτιστης" λύσης που είτε μεγιστοποιεί είτε ελαχιστοποιεί ένα συγκεκριμένο ποσό, με την προϋπόθεση ότι ικανοποιούνται ορισμένοι περιορισμοί ή προϋποθέσεις. Δηλαδή την επιλογή του καλύτερου στοιχείου ή του καλύτερου συνόλου στοιχείων σε σχέση με κάποιο κριτήριο και κάποιου συνόλου εναλλακτικών επιλογών.

Τα βασικά στοιχεία ενός προβλήματος βελτιστοποίησης είναι τα εξής (Bertsimas, D., & Tsitsiklis, J. N., 1997):

- Αντικειμενική συνάρτηση: η συνάρτηση που πρέπει να μεγιστοποιηθεί ή να ελαχιστοποιηθεί. Αντιπροσωπεύει το ποσό που θέλουμε να βελτιστοποιήσουμε, όπως κέρδη, κόστος, χρόνος ή οποιοδήποτε άλλο μέτρο επίδοσης.
- Μεταβλητές απόφασης: οι μεταβλητές που μπορούμε να ελέγχουμε και να προσαρμόσουμε για να επιτύχουμε το βέλτιστο αποτέλεσμα. Συνήθως έχουν τιμές που ανήκουν σε συγκεκριμένο πεδίο ορισμού.
- Περιορισμοί: οι περιορισμοί ή οι συνθήκες που πρέπει να ικανοποιούνται από τις μεταβλητές. Αντιπροσωπεύουν τους περιορισμούς που επιβάλλονται στις μεταβλητές απόφασης και συνήθως εκφράζονται ως ανισότητες ή ισότητες.

Η διαδικασία της βελτιστοποίησης περιλαμβάνει τη μαθηματική διατύπωση της αντικειμενικής συνάρτησης και των περιορισμών, την επιλογή κατάλληλων αλγορίθμων ή μεθόδων για την επίλυση του προβλήματος βελτιστοποίησης όσο το δυνατόν αποδοτικότερα και στη συνέχεια την εύρεση των τιμών των μεταβλητών απόφασης που δίνουν τη βέλτιστη λύση.

Διακρίνεται σε προβλήματα διακριτής βελτιστοποίησης (discrete optimization) και συνεχούς (continuous optimization). Στη διακριτή βελτιστοποίηση, όλες ή οι περισσότερες τιμές των μεταβλητών της αντικειμενικής συνάντησης είναι διακριτές. Τρεις χαρακτηριστικές περιπτώσεις της είναι: combinatorial optimization, integer programming, constraint programming. Στην συνεχή βελτιστοποίηση από την άλλη,

οι τιμές της αντικειμενικής συνάντησης είναι συνεχείς μεταβλητές, επομένως επιλέγονται από το σύνολο των πραγματικών αριθμών ως συνεχή διαστήματα.

2.1.2 Βελτιστοποίηση με ένα μόνο κριτήριο

Υπάρχουν προβλήματα σε τομείς όπως τα μαθηματικά, η επιστήμη υπολογιστών, η οικονομία κλπ. που ενδιαφερόμαστε για την εύρεση της βέλτιστης λύσης από ένα σύνολο ικανών λύσεων δηλαδή λύσεων που ικανοποιούν τους περιορισμούς που έχουμε θέσει. Σε αυτή την περίπτωση θέλουμε είτε να μεγιστοποιήσουμε είτε να ελαχιστοποιήσουμε μία συνάρτηση-στόχο, την αντικειμενική συνάρτηση. Αυτή θα μπορούσε να εκφράζει το συνολικό κέρδος μίας επένδυσης οπότε θέλουμε να το μεγιστοποιήσουμε είτε το κόστος της οπότε θέλουμε να το ελαχιστοποιήσουμε.

Έστω ότι έχουμε τη συνάρτηση f προς μεγιστοποίηση και πρέπει να ικανοποιούνται οι περιορισμοί g, h τότε θα έχουμε τη μορφή:

$$\begin{array}{ll}\max_x & f(x) \\ \text{s.t.} & g_i(x) \leq 0, i = 1, \dots, m \\ & h_j(x) = 0, j = 1, \dots, p\end{array}$$

Όπου

- x : το διάνυσμα των μεταβλητών απόφασης που θέλουμε να βρούμε
- $f(x)$: η αντικειμενική συνάρτηση (εδώ προς βελτιστοποίηση)
- $g_i(x)$: αντιπροσωπεύει τους περιορισμούς που πρέπει να ικανοποιούνται ως ανισότητες
- $h_j(x)$: αντιπροσωπεύει τους περιορισμούς που πρέπει να ικανοποιούνται ως ισότητες

Ανάλογα με τη μορφή των συναντήσεων προς βελτιστοποίηση και των περιορισμών τους, υπάρχουν διαφορετικές μαθηματικές μοντελοποιήσεις με κάθε μία να διαθέτει πολλές γνωστές μεθόδους επίλυσης. Για παράδειγμα, όταν οι συναρτήσεις είναι γραμμικές τότε έχουμε πρόβλημα γραμμικού προγραμματισμού όπου η γνωστότερη μέθοδος επίλυσης είναι η μέθοδος Simplex.

2.1.3 Πολυκριτηριακή Ανάλυση αποφάσεων (Multi-Criteria Decision Analysis - MCDA)

Όταν τα κριτήρια είναι περισσότερα σε ένα πρόβλημα βελτιστοποίησης τότε το πρόβλημα γίνεται πολυκριτηριακό (Hwang et al, 1980). Η Πολυκριτηριακή Ανάλυση αποφάσεων είναι μία σημαντική περιοχή της επιχειρησιακής έρευνας. Πρόκειται για μία προσέγγιση που χρησιμοποιείται για να αξιολογήσει και να συγκρίνει

διαφορετικές εναλλακτικές βάσει πολλαπλών κριτηρίων ή στόχων και βοηθά άτομα ή οργανισμούς να πάρουν ενημερωμένες αποφάσεις όταν αντιμετωπίζουν πολύπλοκες και αντικρουόμενες παραμέτρους.

Σε πολλές καταστάσεις της πραγματικής ζωής, οι αποφάσεις περιλαμβάνουν πολλαπλά κριτήρια που δεν μπορούν να αξιολογηθούν εύκολα ή να συγκριθούν απευθείας. Για παράδειγμα, στην επιλογή ενός νέου αυτοκινήτου, μπορεί να ληφθούν υπόψιν παράγοντες όπως η τιμή, η οικονομία καυσίμου, η ασφάλεια, οι επιδόσεις και η φήμη της μάρκας. Κάθε κριτήριο μπορεί να έχει διαφορετικό βαθμό σημασίας και μερικά ακόμη μπορεί να βρίσκονται σε σύγκρουση μεταξύ τους (π.χ. υψηλή απόδοση έναντι χαμηλής κατανάλωσης καυσίμου).

Η MCDA παρέχει ένα πλαίσιο για να αντιμετωπίσει τέτοια προβλήματα λήψης αποφάσεων, με τους παρακάτω τρόπους:

- Καθορισμός των κριτηρίων: Καταγραφή όλων των σχετικών κριτηρίων που είναι σημαντικά για τη λήψη της απόφασης.
- Ορισμός των εναλλακτικών: Καθορισμός των διαφορετικών επιλογών ή εναλλακτικών που είναι διαθέσιμες για εξέταση.
- Ποσοτικοποίηση κριτηρίων: Ανάθεση σχετικής σημασίας ή βαρών σε κάθε κριτήριο για να αντικατοπτρίζει τη σημασία τους στη διαδικασία λήψης αποφάσεων.
- Αξιολόγηση των εναλλακτικών σε σχέση με τα κριτήρια: Αξιολόγηση και βαθμολόγηση κάθε εναλλακτικής έναντι κάθε κριτηρίου. Μπορούν να χρησιμοποιηθούν διαφορετικές τεχνικές αξιολόγησης, όπως ποιοτικές βαθμολογίες ή ποσοτικά μετρήσεις.
- Συγκέντρωση των σκορ: Συνδυασμός των σκορ κάθε εναλλακτικής με βάση τα βάρη των κριτηρίων για να δημιουργηθεί ένα συνολικό σκορ για κάθε εναλλακτική
- Ανάλυση ευαισθησίας: Εξέταση πόση ευαισθησία έχει το συνολικό σκορ μίας εναλλακτικής σε αλλαγές στα βάρη των κριτηρίων ή σε άλλες αξιολογήσεις.

Για παράδειγμα, θέλουμε να ελαχιστοποιήσουμε ένα σύνολο αντικειμενικών συναντήσεων ταυτόχρονα:

$$\begin{aligned} \min_x \quad & (f_1(x), f_2(x), \dots, f_p(x)) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

Όπου:

$$\begin{aligned} f &: R^n \rightarrow R \\ g &: R^n \rightarrow R^m \\ h &: R^n \rightarrow R^k \end{aligned}$$

$f_1, f_2, \dots, f_p$ είναι οι αντικειμενικές συναρτήσεις

Η διανυσματική μορφή της αντικειμενικής συνάρτησης οδηγεί οι λύσεις να έχουν και αυτές διανυσματική μορφή. Θα πρέπει να γίνει αναζήτησή για μη-κυριαρχούμενες λύσεις έτσι ώστε να έχουμε τελικά ένα Pareto set λύσεων.

Οι μέθοδοι πολυκριτήριας ανάλυσης αποφάσεων, όπως εκείνοι που προαναφέρθηκαν, έχουν μεγάλη χρησιμότητα σε πολλές εφαρμογές. Για παράδειγμα, στη διαχείριση εφοδιαστικής αλυσίδας (supply chain management).

2.2 Η έννοια του Pareto

Σημαντική αναφορά θα γίνει σε λύσεις κατά Pareto και στο σύνολο Pareto σε όλη τη διάρκεια της εργασίας. Βέλτιστη λύση κατά Pareto είναι αυτή που δεν μπορεί να βελτιωθεί η τιμή ως προς μία αντικειμενική χωρίς να χειροτερέψει η τιμή τουλάχιστον σε άλλη μία αντικειμενική. Το σύνολο των κατά Pareto βέλτιστων λύσεων λέγεται σύνολο Pareto (Pareto set).

Γενικότερα, ως Pareto efficiency ή Pareto optimality ορίζουμε μία κατάσταση όπου δεν υπάρχει η δυνατότητα με οποιαδήποτε δυνατή ενέργεια να βελτιώσουμε μία τιμή χωρίς να χειροτερεύσουμε μία άλλη.

Pareto λύση: $P(S, d) = \{x \in S \mid x' \geq x \implies x' \notin S\}$

Υπάρχει και η έννοια weak Pareto efficiency που ορίζουμε μία κατάσταση όπου δεν μπορεί να βελτιωθεί αυστηρά για κάθε τιμή.

Weak Pareto λύση: $WP(S, d) = \{x \in S \mid x' > x \implies x' \notin S\}$

2.3 Πολυκριτήρια Ανάλυση Αποφάσεων για p=2

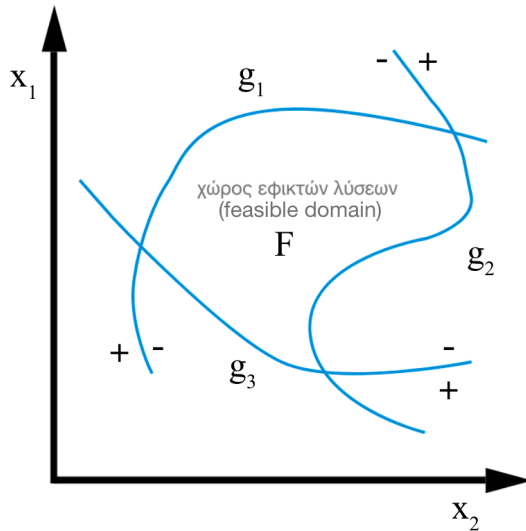
Ένας τρόπος να οπτικοποιήσουμε την πολυστοχική βελτιστοποίηση είναι ο ακόλουθος για p=2:

Αρχικά έστω ότι αυτός είναι ο χώρος των μεταβλητών της βελτιστοποίησης. Έστω ότι έχουμε δύο μεταβλητές x_1, x_2 .

Στην περίπτωση που για παράδειγμα έχουμε πρόβλημα με περιορισμούς ανισότητας της μορφής:

$$F = \{x : g_1(x) \leq 0, g_2(x) \leq 0, g_3(x) \leq 0, x \in R^2\},$$

θα πρέπει να ικανοποιούνται ταυτόχρονα όλοι οι περιορισμοί. Το Γράφημα 2.1 βοηθάει στην οπτικοποίηση τριών περιορισμών στους άξονες x-y. Το σύνολο των σημείων που θα είναι στην feasible side των περιορισμών αυτών θα αποτελούν το feasible domain.



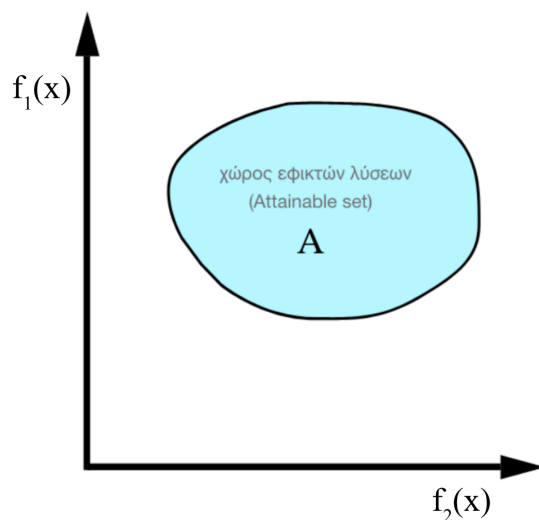
Γράφημα 2.1: Αναπαράσταση των τριών κριτηρίων δύο αντικειμενικών συναρτήσεων

Μπορούμε στη συνέχεια να φανταστούμε τις αντικειμενικές συναρτήσεις ως τον τρόπο να γίνεται αντιστοίχιση (mapping) των στοιχείων από τον χώρο των μεταβλητών βελτιστοποίησης στον χώρο των αντικειμενικών συναρτήσεων.

Το Γράφημα 2.2 οπτικοποιεί τον χώρο των αντικειμενικών συναρτήσεων:

$$A = \{[f_1(x), f_2(x)] : x \in F\}$$

Παρουσιάζει το attainable set δηλαδή το σύνολο των σημείων τα οποία αντιστοιχούν το κάθε ένα σε ένα σημείο του feasible domain της περιοχής των μεταβλητών.

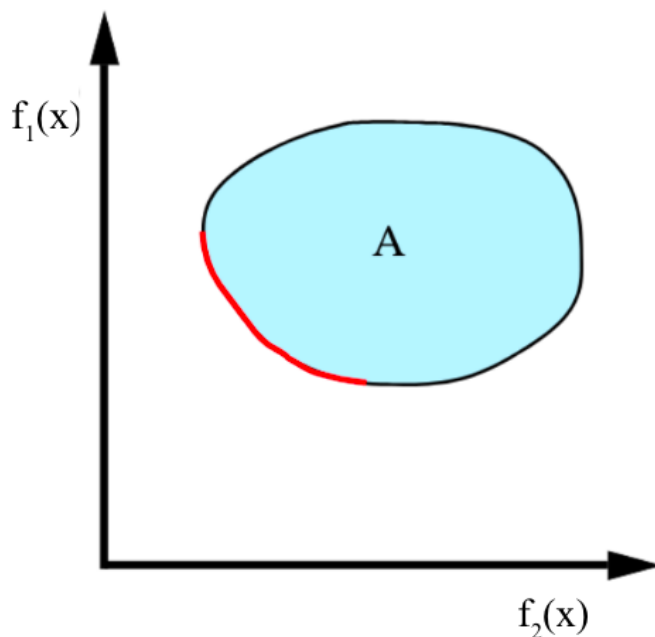


Γράφημα 2.2: Χώρος εφικτών λύσεων δύο συναντήσεων f_1, f_2

Η περιοχή έξω από το attainable set σημαίνει ότι δεν υπάρχει σημείο στο feasible domain των μεταβλητών τέτοιο ώστε να μπορεί μέσω της αντικειμενικής συνάρτησης να παραγάγει αυτό το σημείο εκτός της περιοχής.

Στη συγκεκριμένη περίπτωση που θέλουμε να ελαχιστοποιήσουμε τις δύο συναρτήσεις f_1 , f_2 θέλουμε ιδανικά να κινηθούμε κάτω και αριστερά.

Έτσι παίρνουμε το μέτωπο Pareto Γράφημα 2.3. Παρατηρούμε ότι αν κινηθούμε λίγο πάνω σε αυτό μειώνοντας π.χ. την τιμή της f_1 τότε η τιμή της f_2 θα μεγαλώσει εφόσον πρέπει να παραμείνουμε εντός του attainable set.



Γράφημα 2.3: Σχηματική αναπαράσταση μετώπου Pareto

Η πολυκριτηριακή ανάλυση αποφάσεων μπορεί να γίνει με διάφορες μεθόδους. Την Analytic Hierarchy Process (AHP), την Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), την Preference Ranking Organization METHod for Enrichment Evaluation (PROMETHEE), την VIKOR και πολλές άλλες. Η επιλογή της εκάστοτε μεθόδου εξαρτάται από το συγκεκριμένο πρόβλημα απόφασης και τα χαρακτηριστικά των κριτηρίων και των εναλλακτικών που περιλαμβάνει.

Αξιοσημείωτες μέθοδοι για την Πολυκριτήρια Ανάλυση Αποφάσεων είναι οι μέθοδοι ϵ -constraint και weighted sum όπως και οι επεκτάσεις τους AUGMECON, AUGMECON2, AUGMECON-R όπου με τις τελευταίες θα ασχοληθούμε διεξοδικά σε αυτή την εργασία.

Τέλος, είναι σημαντικό να αναφερθεί ότι οι μέθοδοι απόφασης διακρίνονται σε τρεις κατηγορίες, στις a priori, interactive και a posteriori. Στις a priori μεθόδους ο αποφασίζων έχει τη δυνατότητα να εκφράσει τις προτιμήσεις του ή τα βάρη των αντικειμενικών συναντήσεων πρώτου αρχίσει η διαδικασία επίλυσης του

προβλήματος. Στις διαδραστικές μεθόδους (interactive) επικρατεί μία συνεχής επικοινωνία μεταξύ του αποφασίζοντα και των αναλυτών που οδηγούν σε διαδοχικές συγκλίσεις προς την τελική απόφαση. Τέλος, στις a posteriori μεθόδους, το πρόβλημα επιλύεται και οι ικανές λύσεις παρουσιάζονται στον αποφασίζοντα προκειμένου να πάρει την τελική απόφαση με βάση τις προτιμήσεις του.

2.4 Μετατροπή Πολυκριτηριακού Προβλήματος σε Περισσότερα Προβλήματα Ενός Κριτηρίου (Scalarization)

Στην περίπτωση που έχουμε ένα πρόβλημα βελτιστοποίησης πολλών κριτηρίων όπως το ακόλουθο:

$$\begin{aligned} \min_x \quad & (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

και δεν μπορούμε να το επιλύσουμε άμεσα, το μετατρέπουμε σε ένα σύνολο από προβλήματα μίας αντικειμενικής συνάρτησης (Messac, 2015). Κάθε ξεχωριστό σύνολο τιμών παραμέτρων $\mathbf{p}_i$ θα δώσει ένα ξεχωριστό υποπρόβλημα J_{p_i} . Έστω τα σύνολα των παραμέτρων $\mathbf{p}_i: \{p_1, p_2, \dots, p_{N_p}\}$ όπου i θα είναι ο αριθμός των τιμών των παραμέτρων ή αλλιώς τα κατά Pareto βέλτιστα σημεία (ένα για κάθε υποπρόβλημα). Για παράδειγμα μπορούμε να μεταβάλλουμε τη σημασία της κάθε αντικειμενικής συνάντησης ασφαλώς όμως έτσι θα αποδοθεί διαφορετική βαρύτητα σε κάθε αντικειμενική συνάρτηση.

Έχουμε έτσι τα ακόλουθα υποπροβλήματα:

$$\begin{aligned} \min_x \quad & J_{p_1}(x) \\ \text{s.t.} \quad & g_{p_1}(x) \leq 0 \quad (\text{Πρόβλημα με μία αντικειμενική συνάρτηση}) \\ & h_{p_1}(x) = 0 \end{aligned}$$

$$\begin{aligned} \min_x \quad & J_{p_2}(x) \\ \text{s.t.} \quad & g_{p_2}(x) \leq 0 \\ & h_{p_2}(x) = 0 \end{aligned}$$

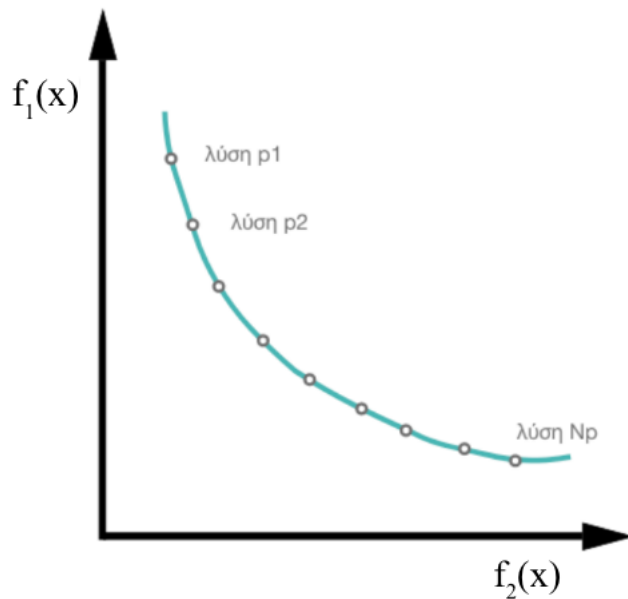
...

...

$$\begin{aligned}
& \min_x J_{N_p}(x) \\
& \text{s.t. } g_{N_p}(x) \leq 0 \\
& \quad h_{N_p}(x) = 0
\end{aligned}$$

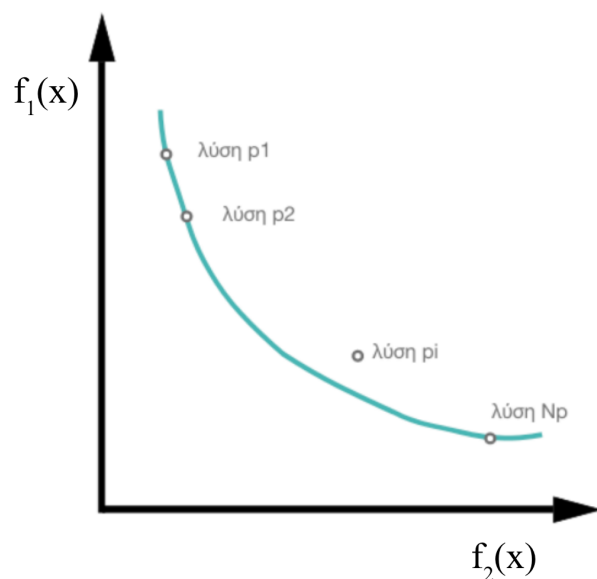
Όπου το J_i είναι η αλλαγμένη αντικειμενική συνάρτηση και τα g_{p_i} , h_{p_i} θα ορίζουν το καινούριο σύνολο περιορισμών

Αν λύσουμε το κάθε ένα από αυτά ξεχωριστά θα μας δώσει και από μία λύση στο Pareto set όπως φαίνεται στο Γράφημα 2.4.



Γράφημα 2.4: Εύρεση μετώπου Pareto μέσω των λύσεων των υποπροβλημάτων

Υπάρχει όμως η περίπτωση το υποπρόβλημα p_i να μη δώσει Pareto λύση μετά από την εκτέλεση του (Γράφημα 2.5).



Γράφημα 2.5: Η περίπτωση που η λύση του υποπροβλήματος δεν ανήκει στο μέτωπο Pareto

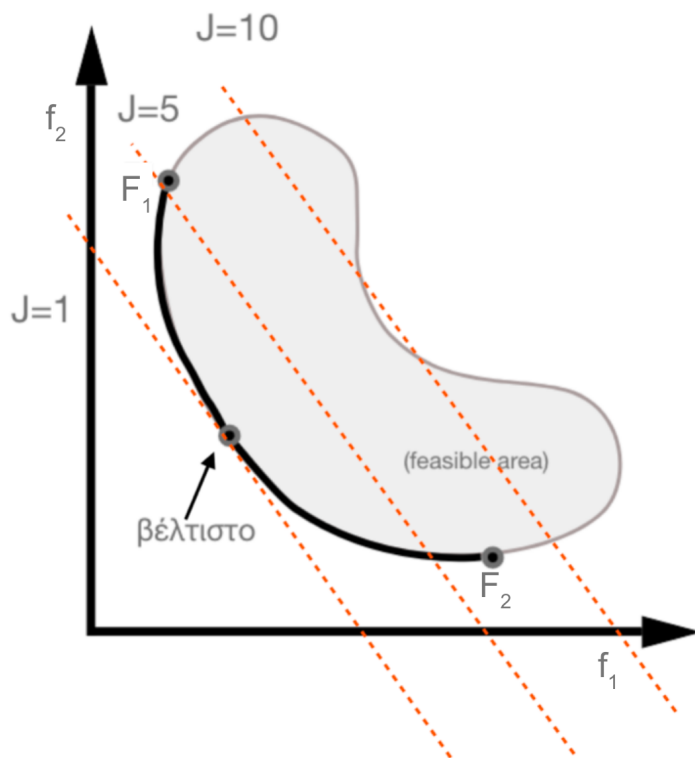
Επομένως σε αυτή την περίπτωση θα πρέπει να υπάρξει ένα στάδιο ανάλυσης της λύσης προκειμένου να διαπιστωθεί ο λόγος για αυτό το αποτέλεσμα. Επίσης, ειδικά στην περίπτωση που το Pareto set δεν είναι κυρτό σχήμα να οδηγηθούμε με τη βελτιστοποίηση σε περιπτώσεις τοπικών ελαχίστων από τα οποία να μην μπορούμε να φύγουμε επομένως να είναι πάλι απαραίτητη η ανάλυση των λύσεων και της μεθόδου που χρησιμοποιήσαμε.

2.5 Μέθοδος Σταθμισμένου Αθροίσματος (Weighted Sum)

Στη μέθοδο weighted sum (Nature-Inspired optimization algorithms, 2021), (Messac, 2015), (Marler et al, 2009), όπως άλλωστε και σε πολλές άλλες χρησιμοποιείται η έννοια του scalarization. Γίνεται αναφορά στη συγκεκριμένη μέθοδο λόγω της μεγαλύτερης ευκολίας ως προς την αντίληψη των εννοιών αυτών. Η βασική ιδέα είναι ότι χρησιμοποιείται μία γραμμική μέθοδος βαρών για να ποσοτικοποιηθεί η σημασία της κάθε αντικειμενικής συνάντησης. Θα δημιουργηθεί μία αθροιστική αντικειμενική συνάρτηση ως γραμμικός συνδυασμός των υπόλοιπων αντικειμενικών συναρτήσεων.

$$\begin{aligned} \min_x \quad & w_1 f_1(x) + w_2 f_2(x) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

Ουσιαστικά μία συγκεκριμένη τιμή των βαρών w_1 , w_2 αντιστοιχεί και σε μία παράλληλη γραμμή ίδιας κλίσης (Γράφημα 2.6).



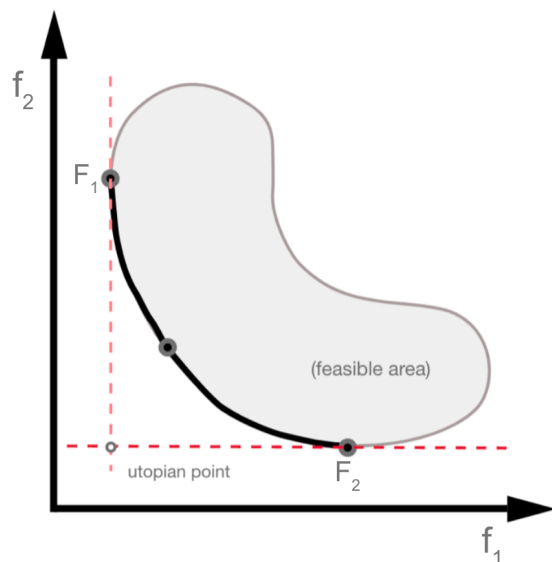
Γράφημα 2.6: Οι παράλληλες γραμμών που προκύπτουν από συγκεκριμένες τιμές των βαρών

Θεωρούμε ότι τα βάρη κανονικοποιούνται: $0 \leq w_i$ και $\sum w_i = 1$

Άρα έχουμε $n-1$ βαθμούς ελευθερίας στα βάρη αν θεωρήσουμε ότι ο αριθμός του είναι n . Στην προκειμένη περίπτωση, έχουμε 2 βάρη άρα έναν βαθμό ελευθερίας. Βάζοντας τιμές από το διάστημα $[0,1]$ στο w_1 βγαίνει αντίστοιχα και η τιμή $w_2=1-w_1$. Έτσι, ελαχιστοποιώντας την συνάρτηση η ευθεία πηγαίνει προς κάτω και αριστερά και τελικά εφάπτεται σε ένα σημείο Pareto της feasible area. Αλλάζοντας το w_1 και κάνοντας την ίδια διαδικασία θα πάρουμε τα υπόλοιπα σημεία Pareto. Με αυτή τη διαδικασία παίρνουμε το Pareto set αφού μετατρέψαμε το πρόβλημα πολλών αντικειμενικών σε υποπροβλήματα μίας αντικειμενικής συνάντησης.

Εάν ορίσουμε το w_1 ως 0 τότε οι ευθείες θα είναι παράλληλες ως προς τον άξονα της f_1 . Μετά την ελαχιστοποίηση το βέλτιστο σημείο όπως φαίνεται και στο σχήμα 6 θα είναι το F_2 . Το F_2 είναι το βέλτιστο σημείο όταν ελαχιστοποιούμε τη $f_2(x)$ και αγνοούμε την $f_1(x)$ παντελώς. Αντίστοιχα συμβαίνει αυτό και για το σημείο F_1 και το πιο σημαντικό είναι ότι τα σημεία F_1, F_2 μπορούν να υποδείξουν τα άκρα του μετώπου Pareto.

Το Γράφημα 2.7 δείχνει το ουτοπικό σημείο της βελτιστοποίησης (το οποίο συνήθως δεν υπάρχει), δηλαδή το σημείο όπου είναι το ελάχιστο για όλες τις αντικειμενικές συναρτήσεις και άρα το καλύτερο σημείο.

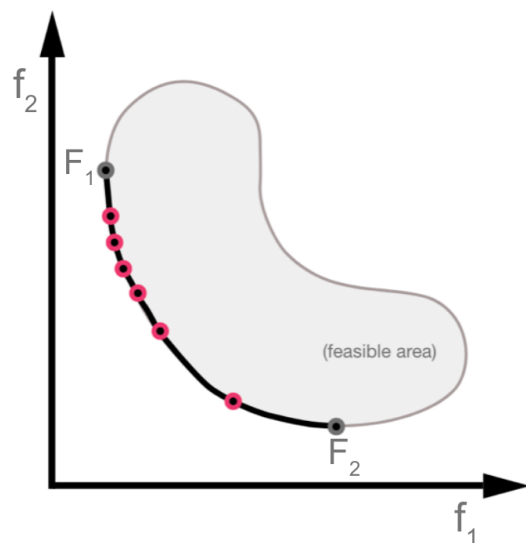


Γράφημα 2.7: Ουτοπικό σημείο βελτιστοποίησης

Βρίσκεται από την τομή των ευθειών με $w_1=0$ και με $w_2=0$ αν προεκταθούν αντίστοιχα. Ωστόσο, πολύ συχνά δεν μπορεί να υπάρξει ως λύση καθώς θα βρίσκεται εκτός της feasible area σε ένα ρεαλιστικό πρόβλημα βελτιστοποίησης όπου θα συγκρούονται μεταξύ τους τα συμφέροντα των αντικειμενικών συναρτήσεων. Εάν το ουτοπικό σημείο βρισκόταν εντός του συνόλου των πιθανών λύσεων τότε δεν θα είχαμε αντικρουόμενες αντικειμενικές συναρτήσεις και επομένως το πρόβλημα βελτιστοποίησης θα θεωρούνταν εκφυλισμένο αφού το μέτωπο Pareto θα είχε εκφυλιστεί σε ένα μόνο σημείο.

Ωστόσο παρουσιάζονται σημαντικά ελαττώματα με τη μέθοδο weighted sum.

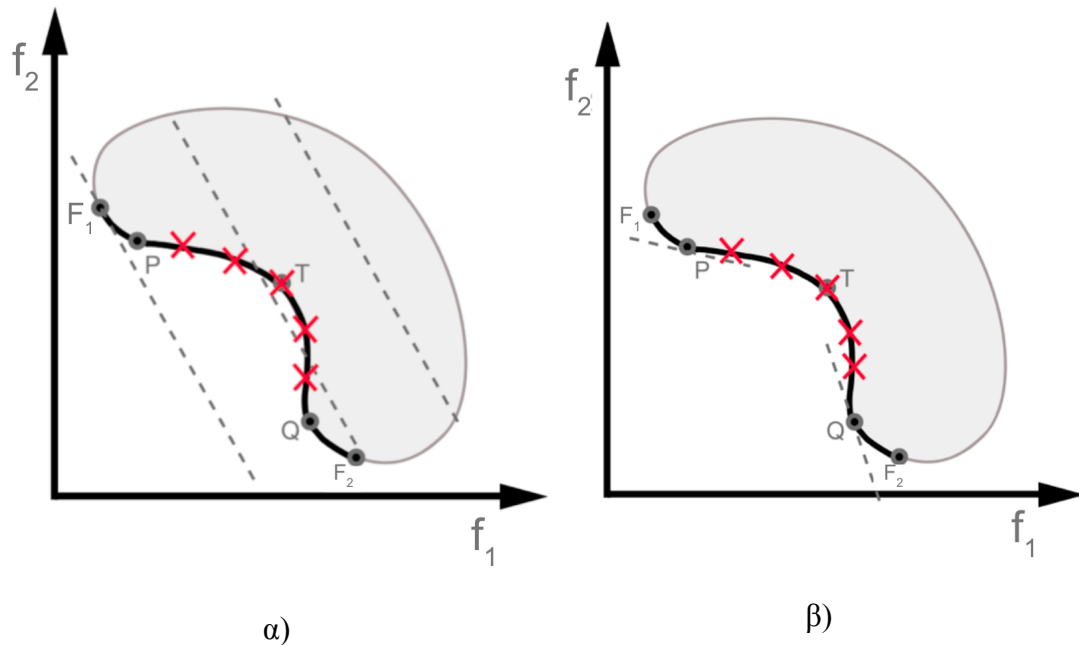
Ένα από αυτά είναι η ανομοιογενής συγκέντρωση λύσεων σε συγκεκριμένα σημεία (Γράφημα 2.8) με αποτέλεσμα να μην έχουμε σαφή εικόνα για τις διαφορές μεταξύ των λύσεων μίας περιοχής και να απαιτείται επιμέρους μελέτη με άλλες μεθόδους.



Γράφημα 2.8: Ανομοιογενής συγκέντρωση λύσεων

Ένα ακόμα βασικό πρόβλημα είναι η αδυναμία της μεθόδου weighted sum να εντοπίζει οποιαδήποτε Pareto λύση σε μη κυρτή περιοχή. Κάποιες λύσεις που μπορούν να βρεθούν φαίνονται στο Γράφημα 2.9.

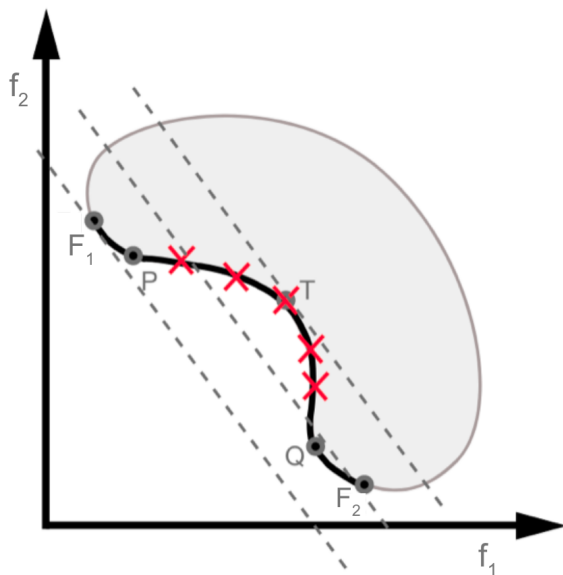
Η λύση T δεν θα μπορέσει να βρεθεί όπως και όλα τα σημεία μεταξύ των P και Q τα οποία ανήκουν στο μέτωπο Pareto (Γράφημα 2.10).



Γράφημα 2.9: Μη-κυρτό feasible set με α) Απώλεια Pareto λύσεων β) Αδυναμία εύρεσης περαιτέρω λύσεων μεταξύ των σημείων P και Q

Η μη-κυρτότητα (non-convexity) του feasible set μπορεί να ενφανιστεί αν τουλάχιστον μία από τις συναρτήσεις f_i είναι μη-κυρτή (non-convex).

Αυτό συμβαίνει γιατί δεν μπορεί να βρεθεί συνδυασμός βαρών άρα και ευθεία άρα και που να εφάπτεται σε αυτά τα σημεία. Ακόμα και να φέρναμε μία ευθεία που να εφάπτεται στο T κατά τη διαδικασία θα συνέχιζε να κατεβαίνει κάτω και αριστερά αφού βρίσκει σημεία με μικρότερες τιμές και άρα αγνοείται ένα σημαντικό μέρος του μετώπου Pareto. (Γράφημα 2.11).



Γράφημα 2.10: Αδυναμία εύρεσης των λύσεων ακόμα και αν ξεκινήσει η διαδικασία από το σημείο T

2.6 Μέθοδος ϵ -constraint

Υπάρχουν μέθοδοι που θα αντιμετωπίζουν τις παραπάνω δυσκολίες της μεθόδου weighted sum και μία πολύ σημαντική από αυτές είναι η ϵ -constraint (Mavrotas et al, 2009), (Messac, 2015), (Goodarzi et al, 2014).

Η βασική ιδέα είναι η μετατροπή όλων των αντικειμενικών συναντήσεων εκτός από μίας σε περιορισμούς. Έπειτα με τη συστηματική αλλαγή των ορίων των περιορισμών γεννάται το μέτωπο Pareto του προβλήματος. Επομένως δεν πρόκειται για δημιουργία συνολικής συνάρτησης αθροίζοντας τις αντικειμενικές συναρτήσεις όπως στη μέθοδο weighted sum.

Έστω ότι έχουμε ένα πρόβλημα πολυστοχικού μαθηματικού προγραμματισμού (Multi-Objective Mathematical Programming):

$$\begin{aligned} \max_x \quad & f_1(x), f_2(x), \dots, f_p(x) \\ \text{s.t.} \quad & x \in S \end{aligned}$$

Όπου:

$f_1, f_2, \dots, f_p$ είναι οι αντικειμενικές συναρτήσεις
 S το πεδίο ορισμού (feasible region)

Κάνουμε βελτιστοποίηση σε μία από τις αντικειμενικές συναρτήσεις χρησιμοποιώντας τις άλλες αντικειμενικές ως περιορισμούς:

$$\begin{aligned}
& \max_x f_1(x) \\
& \text{s.t. } f_2(x) \geq e_2 \\
& \quad f_3(x) \geq e_3 \\
& \quad \vdots \\
& \quad f_p(x) \geq e_p \\
& x \in S
\end{aligned}$$

Με την αλλαγή των παραμέτρων στα δεξιά μέρη των ισοτήτων (e_i) παίρνουμε τις επαρκείς (efficient) λύσεις του προβλήματος.

Το πρόβλημα επιλύεται σε βήματα με βάση ένα $N_2 \times N_3 \times \dots \times N_p$ πλέγμα (grid) όπου N_i είναι το ακέραιο εύρος της αντικειμενικής συνάρτησης f_i .

Ορισμένα πλεονεκτήματα της ϵ -constraint:

- Βρίσκονται με αξιοπιστία οι λύσεις σε μη-κυρτό μέτωπο Pareto εφόσον φυσικά λυθούν ορθά και τα αντίστοιχα υποπροβλήματα.
- Η κατανομή των σημείων Pareto είναι ομοιογενής.
- Για τα γραμμικά προβλήματα η weighted sum εφαρμόζεται στην αρχική feasible region και δίνει αποτελέσματα ακραίας φύσεως. Από την άλλη, η ϵ -constraint, μεταβάλλει τη feasible region και μπορεί να βρει και μη-ακραίες λύσεις.
- Στην ϵ -constraint μπορούμε να επιλέξουμε πόσες επαρκείς λύσεις θέλουμε να λάβουμε πολύ πιο εύκολα από ότι στη weighted sum. Αυτό συμβαίνει, όπως θα αναφερθεί και έπειτα στη δυνατότητα προσαρμογής του αριθμού των grid-points για κάθε εύρος κάθε αντικειμενικής συνάντησης.

Για την καλύτερη κατανόηση της μεθόδου θα μελετηθεί η περίπτωση με δύο αντικειμενικές συναρτήσεις.

$$\begin{aligned}
& \min_x (f_1(x), f_2(x)) \\
& \text{s.t. } g(x) \leq 0
\end{aligned}$$

Στη συγκεκριμένη περίπτωση αντί να έχουμε βάρη, οι παράμετροι μετατροπής θα είναι τα όρια στους περιορισμούς που θα προκύψουν.

$$\{p_1, p_2, \dots, p_{N_p}\} \implies \epsilon_1 < \epsilon_2 < \dots < \epsilon_{N_p}$$

Έτσι δημιουργούνται N_p υποπροβλήματα προς επίλυση:

$$\begin{aligned} \min_x \quad & f_1(x) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & f_2(x) - \varepsilon_1 \leq 0 \end{aligned}$$

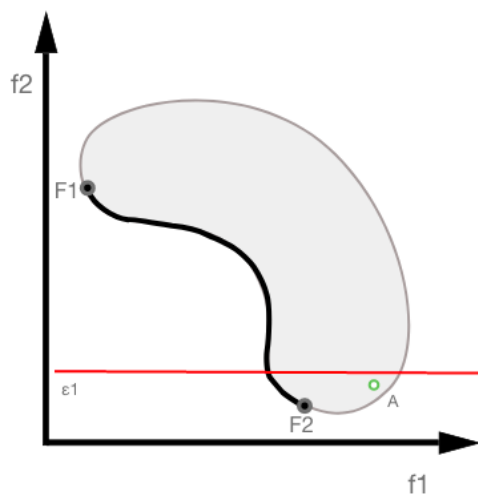
$$\begin{aligned} \min_x \quad & f_1(x) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & f_2(x) - \varepsilon_2 \leq 0 \end{aligned}$$

·
·
·

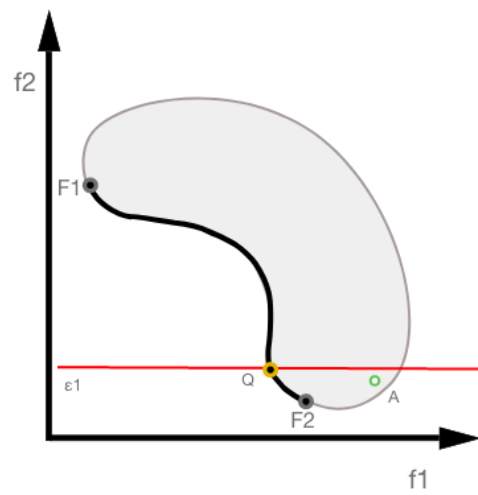
$$\begin{aligned} \min_x \quad & f_1(x) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & f_2(x) - \varepsilon_{Np} \leq 0 \end{aligned}$$

Έστω αρχικά λοιπόν ότι βάζουμε άνω όριο ε_2 στη συνάρτηση f_2 στο προηγούμενο πρόβλημα και ξεκινάμε τη διαδικασία από το σημείο A με τιμές $f_1(x_0)$, $f_2(x_0)$:

Το άνω όριο της f_2 θα μας οδηγήσει στο να ελαχιστοποιήσουμε την f_1 και να βρούμε την τιμή Q του μετώπου Pareto. (Γραφήματα 2.12.1, 2.12.2)



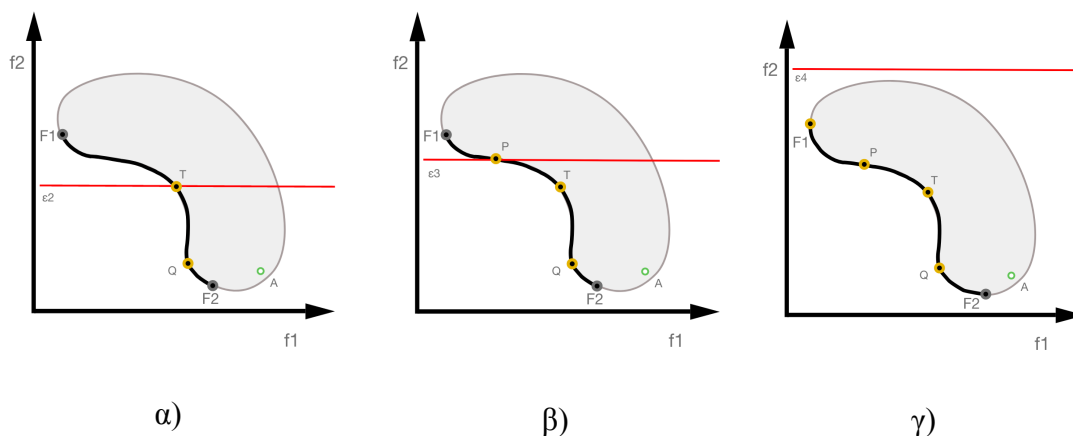
α)



β)

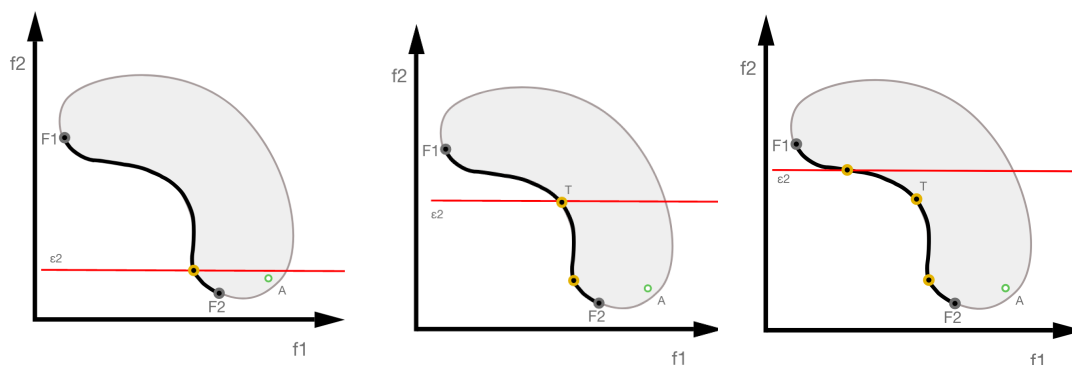
Γράφημα 2.11: α) Αναζήτηση σημείου Q με $\varepsilon = \varepsilon_1$, β) Εύρεση σημείου Q

Στη συνέχεια αλλάζουμε το όριο ε_1 σε ε_2 , ε_3 , ε_4 και χρησιμοποιώντας πάλι ως σημείο αναφοράς το A βρίσκουμε τα επόμενα σημεία (Γραφήματα 2.13.1, 2.13.2, 2.13.3):



Γράφημα 2.12: α) Εύρεση σημείου T με $\epsilon=\epsilon_2$, β) Εύρεση σημείου P με $\epsilon=\epsilon_3$, γ) Εύρεση σημείου F1 με $\epsilon=\epsilon_4$

Αξιοσημείωτη είναι η παρατήρηση ότι ακόμα και οι διαφορές μεταξύ των ϵ_i να είναι ομοιογενής, δεν σημαίνει ότι θα παρατηρήσουμε την ίδια ομοιογένεια και στις λύσεις που θα βρίσκουμε κάθε φορά. Εδώ επιλέξαμε να βάλουμε όριο στην f_2 και επομένως λόγω του σχήματος και των κλίσεων που παρουσιάζει (π.χ. σημεία ανάμεσα στα T, Q ή κοντά στο P) πήραμε τιμές που είτε μπορεί να περιλαμβάνουν μεγάλα άλματα μεταξύ τους ή να είναι αρκετά συσσωρευμένες μεταξύ τους (clustered) (Γράφημα 2.14).



Γράφημα 2.13: Αύξηση του ϵ ομοιογενώς και παρατήρηση των λύσεων

Παρά τα πλεονεκτήματα πρέπει να δοθεί προσοχή στα εξής σημεία:

- Στον υπολογισμό του εύρους των αντικειμενικών συναρτήσεων στο σύνολο των επαρκών λύσεων.
- Στο να εξασφαλιστεί η αξιοπιστία των λύσεων που θα πάρουμε
- Στον υψηλό χρόνο που έχει η διαδικασία ιδιαίτερα όταν αυξάνεται ο αριθμός των αντικειμενικών συναρτήσεων

2.7 Παραλλαγές της μεθόδου ϵ -constraint

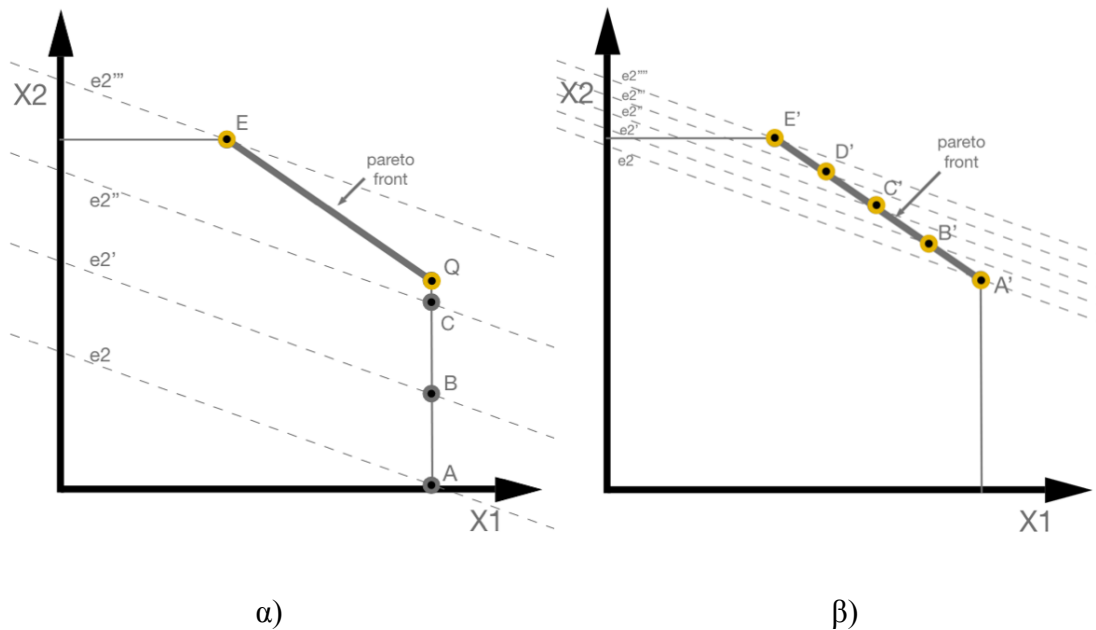
2.7.1 Augmented ϵ -constraint method (AUGMECON)

Μία βελτιωμένη μέθοδος της ϵ -constraint που αναφέρθηκε είναι η Augmented ϵ -constraint ή αλλιώς AUGMECON (Mavrotas et al, 2009).

Γενικότερα για την εφαρμογή της μεθόδου ϵ -constraint, είναι σημαντικός ο υπολογισμός του εύρους (range) των αντικειμενικών συναντήσεων ή τουλάχιστον των $p-1$ αν είναι p στο σύνολο. Η εύρεση της βέλτιστης τιμής κάθε αντικειμενικής είναι σχετικά εύκολη υπόθεση καθώς μπορεί να προκύψει από την ανεξάρτητη βελτιστοποίηση της από κάποιον απλό optimizer. Η εύρεση της χειρότερης τιμής ωστόσο (τιμή ναδίρ) δεν είναι τόσο απλή όπως δείχνουν άλλωστε και πολλές σχετικές μελέτες σχετικά με τον υπολογισμό ή την εκτίμησή τους (Alves et al, 2009), (Dessouky et al, 1986), (Isermann et al, 1988), (Jorge et al, 2009), (Metev et al, 2003), (Reeves et al, 1988), (Steuer et al, 1997). Συνήθως προσεγγίζεται με την μικρότερη τιμή της ανεξάρτητης βελτιστοποίησης στην περίπτωση προβλήματος μεγιστοποίησης (η μεγαλύτερη αντίστοιχα για πρόβλημα ελαχιστοποίησης). Θα πρέπει όμως να είμαστε σίγουροι ότι με τις συγκεκριμένες τιμές βέλτιστων και ναδίρ θα δώσει ο αλγόριθμος της ϵ -constraint βέλτιστες κατά Pareto λύσεις και όχι κυριαρχούμενος από άλλες. Μία πρόταση λοιπόν έγινε για λεξικογραφική βελτιστοποίηση για κάθε αντικειμενική συνάρτηση έτσι ώστε να κατασκευαστεί πίνακας τιμών (payoff table) με μόνο βέλτιστες κατά Pareto λύσεις.

Ένας τρόπος αντιμετώπισης του προβλήματος των σημείων ναδίρ είναι ο ορισμός μεταβλητών (reservation values) που κρατούν τα κάτω όρια των αντικειμενικών ως κατώτερο όριο (αντίστοιχα ανώτερο όριο στην περίπτωση ελαχιστοποίησης).

Στο Γράφημα 2.15.1 παρατηρούμε ότι από τις τιμές (A, B, C, E) που δίνει η μέθοδος ϵ -constraint σε ένα παράδειγμα χωρίς λεξικογραφική βελτιστοποίηση, οι A, B, C είναι ασθενείς Pareto λύσεις (weakly efficient solutions) καθώς η τιμή Q κυριαρχεί και στις τρεις. Αντίθετα, στο Γράφημα 2.15.2 όπου εφαρμόστηκε σε ένα παράδειγμα η μέθοδος ϵ -constraint αφότου έγινε λεξικογραφική βελτιστοποίηση στις αντικειμενικές συναρτήσεις φαίνεται ότι έδωσε λύσεις που βρίσκονται πάνω στο μέτωπο Pareto (A', B', C', D', E') και είναι όλες βέλτιστες (Pareto optimal solutions). Δηλαδή με τον τρόπο αυτό έγιναν οι επαναλήψεις στην περιοχή των βέλτιστων λύσεων.



Γράφημα 2.14: Μέθοδος ϵ -constraint α) χωρίς λεξικογραφική βελτιστοποίηση β) με λεξικογραφική βελτιστοποίηση

Η παραπάνω διαδικασία διασφαλίζει παίρνουμε βέλτιστες λύσεις δεδομένου ότι οι περιορισμοί που αντιστοιχούν στις αντικειμενικές συναρτήσεις είναι δεσμευτικοί (binding constraints). Γενικότερα δεσμευτικός περιορισμός $g_i(x) \leq 0$ (binding constraint) είναι αυτός που ικανοποιείται με ισότητα $g(x)=0$ στην βέλτιστη λύση. Στην περίπτωση που οι περιορισμοί δεν ήταν δεσμευτικοί τότε θα υπήρχαν άλλα βέλτιστα που θα ήταν καλύτερα από την τιμή της ισότητας και επομένως η λύση που θα παίρναμε από την τιμή της ισότητας θα ήταν ασθενής (weakly efficient).

Προκειμένου να ξεπεραστεί η παραπάνω περίπτωση, μετατρέπονται οι ανισότητες σε ισότητες με τη χρήση slack variables. Σε ένα πρόβλημα βελτιστοποίησης, οι slack variables είναι μεταβλητές που προστίθενται σε περιορισμούς ανισότητας έτσι ώστε να γίνουν ισότητες. Με την εισαγωγή της slack variable ο περιορισμός ανισότητας αντικαθίσταται από έναν περιορισμό ισότητας και έναν περιορισμό για τη μη-αρνητικότητα της slack variable.

$$a \cdot x > b \iff s = a \cdot x - b \\ s \geq 0$$

Μετά την εισαγωγή των slack variables και στην αντικειμενική συνάρτηση προς βελτιστοποίηση ως δεύτερος όρος με μικρότερη προτεραιότητα αλλά και στους περιορισμούς, το πρόβλημα έχει τη μορφή:

$$\begin{aligned}
\max_x \quad & f_1(x) + \varepsilon \cdot (s_2 + s_3 + \dots + s_p), \quad \varepsilon \in (10^{-6}, 10^{-3}) \\
\text{s.t.} \quad & f_2(x) - s_2 = e_2 \\
& f_3(x) - s_3 = e_3 \\
& \dots \\
& f_p(x) - s_p = e_p \\
& x \in S \\
& s_1, s_2, \dots, s_p \geq 0
\end{aligned}$$

Επιπλέον, για να αντιμετωπιστούν προβλήματα κανονικοποίησης (scaling problems) διαιρείται η κάθε slack variable με το αντίστοιχο range της κάθε αντικειμενικής συνάρτησης. Επομένως η αντικειμενική συνάρτηση παίρνει τη μορφή:

$$\begin{aligned}
\max_x \quad & f_1(x) + \varepsilon \cdot (s_2/r_2 + s_3/r_3 + \dots + s_p/r_p) \\
& r_i \geq 0
\end{aligned}$$

Οι παραπάνω μετατροπές εξασφαλίζουν ότι μόνο κατά Pareto βέλτιστες λύσεις θα εμφανίζονται και η εξελιγμένη μέθοδος αυτή ονομάστηκε “Augmented ε-constraint method” ή αλλιώς “AUGMECON”.

Μία ακόμη καινοτομία του αλγορίθμου της AUGMECON είναι η δυνατότητα πρόωρης εξόδου από την εκάστοτε επανάληψη κατά τη διάρκεια του τρεξίματος του προγράμματος όταν το πρόβλημα γίνεται infeasible. Με αυτόν τον τρόπο επιτυγχάνεται σημαντική επιτάχυνση του αλγορίθμου ιδίως όταν οι αντικειμενικές συναρτήσεις είναι περισσότερες από τρεις. Ξεκινάει η διαδικασία με μία ελαστική μορφή περιορισμών των αντικειμενικών συναρτήσεων και σταδιακά αυστηροποιούνται τα όρια. Συγκεκριμένα από την ελάχιστη τιμή (στην περίπτωση της μεγιστοποίησης) και σταδιακά αυξάνεται η τιμή στο δεξί μέρος της ανισότητας του αντίστοιχου περιορισμού. Όταν το πρόβλημα μετά από κάποιο σημείο γίνει ανέφικτο (infeasible) τότε δεν υπάρχει λόγος για περαιτέρω περιορισμούς στην αντίστοιχη αντικειμενική συνάρτηση. Επομένως, ο αλγόριθμος εξέρχεται από την συγκεκριμένη εσωτερική επανάληψη και προχωράει στην επόμενη επανάληψη, δηλαδή στο επόμενο εξωτερικό σημείο του grid που αντιστοιχεί στην επόμενη αντικειμενική συνάρτηση.

Γενικά όσο πιο πυκνό είναι το πλέγμα (grid) των πιθανών λύσεων τόσο περισσότερες λύσεις λαμβάνονται, αλλά τόσο αυξάνεται ο απαιτούμενος χρόνος εκτέλεσης του αλγορίθμου. Με τις πρόωρες εξόδους σημειώνεται σημαντική βελτίωση στον χρόνο επιτρέποντας την δημιουργία πιο πυκνού πλέγματος που σε περιπτώσεις με παραπάνω από τρεις αντικειμενικές μπορεί να φάνταζε απαγορευτική.

2.7.2 Μέθοδος AUGMECON2

Η μέθοδος AUGMECON2 (Mavrotas et al, 2013) είναι μία επέκταση της AUGMECON που εκμεταλλεύεται την πληροφορία που εμπεριέχουν οι slack variables και αποφεύγει ακόμα περισσότερες επαναλήψεις πετυχαίνοντας σημαντικά υπολογιστικά χρονικά οφέλη. Ειδικότερα στην περίπτωση που οι τιμές του προβλήματος είναι διακριτές και η feasible περιοχή όχι κυρτή, η AUGMECON2 είναι πολύ χρήσιμο εργαλείο.

Στην μέθοδο αυτή η συνάρτηση τροποποιείται ελαφρώς:

$$\begin{aligned} \max_x \quad & f_1(x) + \varepsilon \cdot (s_2/r_2 + s_3/r_3 \cdot 10^{-1} + \dots + s_p/r_p \cdot 10^{-(p-2)}) \quad \varepsilon \in (10^{-6}, 10^{-3}) \\ \text{s.t.} \quad & f_2(x) - s_2 = e_2 \\ & f_3(x) - s_3 = e_3 \\ & \dots \\ & f_p - s_p = e_p \\ & x \in S \\ & s_1, s_2, \dots, s_p > 0 \end{aligned}$$

Γίνεται δηλαδή μία λεξικογραφική βελτιστοποίηση στις υπόλοιπες αντικειμενικές συναρτήσεις και επιβάλλεται μία σειριακή βελτιστοποίηση.

Η διαδικασία πηγαίνει αρχικά όπως και στη AUGMECON. Υπολογίζονται τα εύρη για τις αντικειμενικές 2, ..., p και διαιρείται το εύρος της k αντικειμενικής σε q_k διαστήματα που ορίζονται από q_{k+1} σημεία του πλέγματος (grid points) $k=2, \dots, p$. Έτσι, οι συνολικές επαναλήψεις θα είναι $(q_2 + 1) \cdot (q_3 + 1) \cdot \dots \cdot (q_p + 1)$.

Έστω πως ορίζουμε για τις αντικειμενικές συναρτήσεις:

$$\begin{aligned} step_k &= \frac{r_k}{q_k} \\ e_{kt} &= f_{min} + t \cdot step_k \\ b &= \text{int}\left(\frac{s_k}{step_k}\right) \end{aligned}$$

όπου k η αντικειμενική συνάρτηση και t ο μετρητής της επανάληψης της αντικειμενικής. Με την συνάρτηση int παίρνουμε το ακέραιο μέρος του αριθμού.

Όταν η μεταβλητή s_k είναι μεγαλύτερη από το $step_k$ τότε υποθέτουμε ότι και στην επόμενη επανάληψη θα λάβουμε την ίδια λύση με μοναδική διαφορά την τιμή της s_k . Επομένως η επανάληψη μπορεί να παρακαμφθεί αφού δεν θα οδηγήσει σε νέα λύση Pareto. Ο συντελεστής b ουσιαστικά δείχνει πόσες συνεχόμενες επαναλήψεις μπορούμε να παρακάμψουμε και αυτή είναι η μεγάλη καινοτομία της AUGMECON2.

2.7.3 AUGMECON-R

Η μέθοδος AUGMECON2 (Nikas et al, 2020) απέτελεσε σημαντική βελτίωση στην εύρεση των σημείων του μετώπου Pareto. Ωστόσο, ορισμένα μειονεκτήματα της μεθόδου γέννησαν την ανάγκη για την επίτευξη ακόμα καλύτερου χρόνου υπολογισμού. Ορισμένα από αυτά είναι τα ακόλουθα. Στην περίπτωση προβλημάτων μεγάλης κλίμακας δηλαδή παραπάνω από δύο αντικειμενικών συναρτήσεων ο χρόνος εκτέλεσης είναι ακόμα μεγάλος. Ένα πρόβλημα m αντικειμενικών συναρτήσεων με μέσο όρο εύρους n θα έχει πολυπλοκότητα $O(n^{m-1})$. Επιπλέον, πρέπει οι συντελεστές των αντικειμενικών συναρτήσεων να είναι ακέραιοι αριθμοί. Στην περίπτωση που δεν συμβαίνει αυτό, είναι αναγκαίος ο πολλαπλασιασμός αυτών με την κατάλληλη δύναμη του 10. Ένα πρόβλημα m αντικειμενικών συναρτήσεων, μέσου όρου εύρους n και μέσο αριθμό δεκαδικών ψηφίων k θα έχει πολυπλοκότητα $O(n^{m-1} * 10^{k*(m-1)})$. Ακόμα, είναι απαραίτητη η γνώση των σημείων ναδίρ εξαρχής που όμως είναι διαδικασία σύνθετη και χρονοβόρα. Τα υπάρχοντα προγράμματα που τα βρίσκουν, γενικότερα μπορούν να διαχειριστούν μέχρι τρεις αντικειμενικές συναρτήσεις και δίνουν αποτέλεσμα σε χρόνο συγκρίσιμο με εκείνον που απαιτεί το τρέξιμο του προγράμματος της AUGMECON2. Επομένως, η AUGMECON2, κάνει μία υποεκτίμηση των σημείων ναδίρ πολλαπλασιάζοντας τις ελάχιστες τιμές από τα payoff tables με έναν αυθαίρετο συντελεστή π.χ. 90% που και αυξάνει αισθητά τον χρόνο υπολογισμού αλλά και βασίζεται στην πιθανότητα να περιλαμβάνεται το πραγματικό σημείο ναδίρ σε αυτό το πλέγμα. Εάν δεν τύχει να βρίσκεται σε αυτήν την περιοχή τότε είναι αρκετά πιθανό να χαθούν βέλτιστες λύσεις. Η ανάγκη να μειωθεί αρκετά αυτός ο συντελεστής έρχεται απέναντι σε τεράστιους χρόνους υπολογισμού. Ως τελευταίο ελάττωμα, η AUGMECON2 δεν διαχειρίζεται το γεγονός ότι υπάρχει άμεση σχέση ανάμεσα στην σειρά των βροχών επαναλήψεων των αντικειμενικών συναρτήσεων που θέτονται ως περιορισμοί και στον χρόνο εκτέλεσης. Από τη στιγμή που υπολογίζονται τα payoff tables, η καλύτερη στρατηγική για την αποφυγή όσων περισσότερων περιττών επαναλήψεων είναι δυνατόν είναι οι συναρτήσεις με τα μεγαλύτερα εύρη να αποτελούν τους εσωτερικότερους βρόγχους επαναλήψεων του προγράμματος. Επομένως είναι ιδιαίτερα σημαντική η δυνατότητα αλλαγής σειράς των αντικειμενικών συναρτήσεων.

Η μέθοδος AUGMECON-R προτείνει μία καινοτόμα προσθήκη στην ήδη αρκετά αποδοτική AUGMECON2 που μειώνει δραστικά τον χρόνο εκτέλεσης. Αντί να παρακάμπτονται στάδια μόνο στην εσωτερικότερη επανάληψη, θα μπορούν να παρακάμπτονται διαμέσου όλου του πλέγματος. Μία σημαντική παρατήρηση που

έγινε στην βιβλιογραφία (Nikas et al, 2020) είναι πως εάν για οποιαδήποτε τιμή των $e^*_2, e^*_3, \dots, e^*_p$ του δεξιού μέρους των περιορισμών από τις αντικειμενικές συναρτήσεις, υπάρχει infeasibility, τότε για μία αύξηση οποιουδήποτε εκ των $e_2, e_3, \dots, e_p$ με όλα τα άλλα μεγαλύτερα ή ίσα των $e^*_2, e^*_3, \dots, e^*_p$ θα υπάρχει πάλι infeasibility. Επομένως για $\delta \in \mathbb{N}$ (φυσικοί αριθμοί) κάθε συνδυασμός $\{e^*_i + \delta_i\}$ στο δεξί μέρος του περιορισμού θα δώσει infeasibility. Η AUGMECON2 αποφεύγει infeasibilities για $\{e^*_2 + \delta_2, e^*_3, \dots, e^*_p\}$, $\delta_2 > 0$. Η AUGMECON-R θα αποφεύγει infeasibilities για $\{e^*_2 + \delta_2, e^*_3 + \delta_3, \dots, e^*_p + \delta_p\}$, $\delta_1, \delta_2, \dots, \delta_p > 0$.

Η AUGMECON-R Εντάσσει στο μοντέλο τόσους συντελεστές b (bypass coefficient) όσες και οι αντικειμενικές συναρτήσεις:

$$b_2 = \text{int}(s_2/\text{step}_2)$$

$$b_3 = \text{int}(s_3/\text{step}_3)$$

...

$$b_p = \text{int}(s_p/\text{step}_p)$$

Για p αντικειμενικές συναρτήσεις θα χρειαστεί πίνακας $p-1$ διαστάσεων για την αποθήκευση ακέραιων τιμών ελέγχου (flag values). Ο πίνακας αυτός αρχικοποιείται με μηδενικά σε όλες τις θέσεις και πριν από κάθε βελτιστοποίηση αντικειμενικής συνάντησης k ο αλγόριθμος ελέγχει την αντίστοιχη θέση στον πίνακα. Αν η τιμή στη θέση αυτή είναι 0 τότε γίνεται κανονικά η βελτιστοποίηση. Σε περίπτωση που είναι κάποια τιμή μεγαλύτερη, ο αλγόριθμος παρακάμπτοντας επαναλήψεις, πηγαίνει στην εσωτερικότερη επανάληψη σε αριθμό βημάτων (steps) όσα δείχνει και η τιμή στον πίνακα.

2.7.4 AUGMECON-Py

Στη παρούσα εργασία χρησιμοποιήθηκε εκτενώς η AUGMECON-Py (Forouli et al, 2022), μία μέθοδος που επιλύει σύνθετα και μεγάλα προβλήματα πολυκριτηριακού γραμμικού προγραμματισμού υπό συνθήκες αβεβαιότητας. Το framework της AUGMECON-Py επεξεργάζεται τα δεδομένα που εισάγονται, βρίσκει όλες τις λύσεις στο εκάστοτε grid και κάνει μία γραφική αναπαράστασή των αποτελεσμάτων για καλύτερη κατανόηση. Ουσιαστικά ενσωματώνει τον αλγόριθμο AUGMECON-R σε συνδυασμό με ανάλυση Monte-Carlo προκειμένου να είναι οι λύσεις πιο εύρωστες (robust) καθώς υπάρχουν οι συνθήκες αβεβαιότητας που προαναφέρθηκαν. Γενικότερα η μελέτη του robustness των λύσεων που βρίσκει μία μέθοδος βελτιστοποίησης υπό αβεβαιότητα είναι μία αρκετά συνήθης ανάλυση (Van Dao et al, 2020), (Bertsimas et al, 2009), (Bertsimas et al, 2004). Όσον αφορά τις απαιτήσεις του προγράμματος ως προς την μνήμη, η γλώσσα προγραμματισμού Python στην οποία είναι γραμμένο το πρόγραμμα τις ικανοποιεί σε μεγάλο βαθμό.

Έγινε χρήση του Gurobi Solver που είναι ένα από τα πιο γνωστά λογισμικά βελτιστοποίησης. Χρησιμοποιείται ευρέως σε προβλήματα βελτιστοποίησης λόγω

του ταχέως και αξιόπιστου υπολογισμού μοντέλων που προσφέρει. Εκμεταλλεύεται την μοντέρνα αρχιτεκτονική υπολογιστών και τους πολυπύρηνους επεξεργαστές και αποτελεί ίσως τον καλύτερο solver μαζί με τον CPLEX προς το παρόν. Επομένως η έρευνα που έγινε για την εύρεση ενός γενικά καλύτερου solver δεν οδήγησε στην αλλαγή του στο συγκεκριμένο πρόγραμμα της AUGMECON-Py.

2.8 Μέθοδοι Βελτιστοποίησης σε Ενεργειακά Προβλήματα

Οι μέθοδοι βελτιστοποίησης αποδεικνύονται ιδιαίτερα χρήσιμα εργαλεία για την εύρεση της κατάλληλης ενεργειακής πολιτικής. Ο ενεργειακός τομέας περιέχει σύνθετα προβλήματα όπως η κατανομή πόρων και η αξιολόγηση έργων υποδομής ως προς το περιβαλλοντικό αντίκτυπο και το οικονομικό όφελος ή επιβάρυνση. Η χρήση των μεθόδων αυτών βοηθάει τον αποφασίζοντα να πάρει εμπεριστατωμένες αποφάσεις μεγιστοποιώντας τα οφέλη και ικανοποιώντας πληθώρα περιορισμών. Πιο συγκεκριμένα, ορισμένοι τρόποι που οι μέθοδοι βελτιστοποίησης εφαρμόζονται σε ενεργειακές πολιτικές είναι οι ακόλουθοι:

- Κατανομή πόρων: Οι ενεργειακές πολιτικές συχνά αφορούν αποφάσεις κατανομής περιορισμένων πόρων όπως χρηματικού προϋπολογισμού, έκτασης γης για υποδομές ανανεώσιμης ενέργειας ή κρατικές επιχορηγήσεις. Επομένως, οι μέθοδοι βελτιστοποίησης οδηγούν στην εύρεση της ιδανικής κατανομής των πόρων αυτών για να μεγιστοποιηθεί η παραγωγή ενέργειας, ελαχιστοποιηθούν τα κόστη ή να επιτευχθούν οι υπόλοιποι στόχοι της ενεργειακής πολιτικής.
- Προγραμματισμός έργων υποδομής: Οι αποφάσεις αφορούν δημιουργία ή επέκταση ενεργειακών υποδομών όπως ενός εργοστασίου παραγωγής ηλεκτρικής ενέργειας, καλωδίων μεταφοράς ηλεκτρισμού ή ενός κέντρου αποθήκευσης. Οι μέθοδοι βελτιστοποίησης μπορούν να βοηθήσουν στην εύρεση της κατάλληλης τοποθεσίας, της ιδανικής χωρητικότητας και του καλύτερου συνδυασμού τεχνολογιών που απαιτούν συγκεκριμένες δομές ενώ παράλληλα λαμβάνονται υπόψιν οι οικονομικοί και περιβαλλοντικοί περιορισμοί (Drezner et al, 2004).
- Ενσωμάτωση ανανεώσιμων πηγών ενέργειας: Συχνά οι ενεργειακές πολιτικές τείνουν να αυξάνουν το ποσοστό των ανανεώσιμων πηγών ενέργειας ως προς τον συνολικό συνδυασμό ενεργειακών πηγών. Οι μέθοδοι βελτιστοποίησης βοηθούν στην δημιουργία κατάλληλων στρατηγικών για την ενσωμάτωση των ανανεώσιμων πηγών ενέργειας με τη σωστή διαχείριση των πηγών ενέργειας που παρουσιάζουν διακυμάνσεις (όπως η αιολική και η ηλιακή) και βελτιστοποιώντας την χρήση των μέσων αποθήκευσης (Van De Ven et al, 2022), (Forouli, Gkonis et al, 2019), (Forouli, Doukas et al 2019).

- Οικονομική αποτελεσματικότητα: Οι ενεργειακές πολιτικές επιδιώκουν την οικονομική αποτελεσματικότητα μειώνοντας τα κόστη και αυξάνοντας τα οφέλη διαφορετικών επιλογών. Οι αποφασίζοντες χρησιμοποιούν τα μοντέλα βελτιστοποίησης για να εντοπίσουν τις πιο αποδοτικές ως προς το κόστος επιλογές λαμβάνοντας υπόψιν παράγοντες όπως το κόστος κεφαλαίου, τα λειτουργικά κόστη, οι τιμές της ενέργειας και οι εξωτερικότητες (Yang et al, 2023).
- Περιβαλλοντικό αντίκτυπο: Η βιωσιμότητα και ο σεβασμός στο περιβάλλον έχουν καθοριστική σημασία στις αποφάσεις μίας ενεργειακής πολιτικής. Οι μέθοδοι βελτιστοποίησης μπορούν να χρησιμοποιηθούν για να ελαχιστοποιήσουν τις αρνητικές επιπτώσεις στο περιβάλλον όπως είναι η εκπομπές ρύπων με τον ιδανικό συνδυασμό πηγών ενέργειας και τεχνολογιών (Fu et al, 2020), (Forouli, Gkonis et al, 2019).
- Διαχείριση ζήτησης ενέργειας: Η διαχείριση ζήτησης ενέργειας (demand-side management) είναι η μεταβολή της ζήτησης για ενέργεια από τον καταναλωτή μέσω διαφόρων μεθόδων όπως οικονομικά κίνητρα και ή η εκπαίδευση σχετικά με τη σημασία της. Πολιτικές που ενθαρρύνουν την αποδοτικότερη χρήση της ενέργειας και ασχολούνται με τη διαχείριση της ζήτησης αυτής είναι καίριας σημασίας για ένα μέλλον ενεργειακά βιώσιμο (Esther et al, 2016). Οι μέθοδοι βελτιστοποίησης βοηθούν στον σχεδιασμό καλύτερων προγραμμάτων διαχείρισης ενέργειας κάνοντας χρήση του έξυπνου δικτύου ηλεκτρικής ενέργειας (smart grid) για τον προσδιορισμό των μοτίβων κατανάλωσης ρεύματος. Έτσι επιτυγχάνεται αποσυμφόρηση του δικτύου διανομής και χρησιμοποιούνται αποδοτικότερα οι αποθήκες ενέργειας.
- Εύρεση συμβιβασμών ανάμεσα σε διαφορετικές ενεργειακές πολιτικές: Είναι αρκετά σύνηθες ο αποφασίζων να πρέπει να ικανοποιήσει αντικρουόμενους στόχους, όπως το οικονομικό όφελος και η αειφόρος ανάπτυξη που προαναφέρθηκαν, και επομένως να είναι αναγκαία η εύρεση συμβιβασμών μεταξύ τους. Η πολυκριτηριακή ανάλυση αποφάσεων βοηθάει στην ανάλυση και την ποσοτικοποίηση αυτών των συμβιβασμών παρέχοντας στους αποφασίζοντες μία πιο ξεκάθαρη αποτύπωση των επιπτώσεων των διαφορετικών ενεργειακών πολιτικών (Zerrahn et al, 2017).

Οι μέθοδοι που μελετήθηκαν στις υποενότητες 2.5-2.7 έχουν χρησιμοποιηθεί εκτενώς σε ενεργειακά προβλήματα. Συγκεκριμένα, η μέθοδος weighted sum έχει χρησιμοποιηθεί σε έρευνες που έχουν γίνει σχετικά με την επιλογή του σωστού τύπου ανεμογεννήτριας καθώς διαφορετικά χαρακτηριστικά μίας ανεμογεννήτριας όπως το ύψος, η διάμετρος και η τοποθεσία της εγκατάστασής τους μπορούν να επηρεάσουν την απόδοση σε ενέργεια (Rehman et al, 2017). Ένα ακόμα παράδειγμα εφαρμογής της μεθόδου weighted sum είναι στην επιλογή της βέλτιστης εναλλακτικής φυτών για την παραγωγή βιοκαυσίμου (Mishra et al 2021).

Όσον αφορά την μέθοδο ε-constraint, μία εφαρμογή της είναι στην μελέτη της λειτουργίας ενός συστήματος παροχής ενέργειας υβριδικού τύπου. Ο στόχος είναι να είναι οικονομικά ωφέλιμο, περιορίζοντας παράλληλα τις εκπομπές διοξειδίου του άνθρακα (CO₂) (Nojavan et al, 2017). Ακόμα, χρησιμοποιήθηκε η ε-constraint σε ένα πρόβλημα εκμετάλλευσης άεργου ισχύος (Reactive Power Planning) υπό συνθήκες αβεβαιότητας σχετικά με το συνολικό φορτίο ζήτησης και το μέγεθος της παραγωγής της αιολικής ενέργειας (Shojaei et al, 2020).

Τέλος, η μέθοδος AUGMECON και οι παραλλαγές της, παρουσιάζουν πληθώρα εφαρμογών στον τομέα της ενέργειας. Ορισμένες από αυτές είναι: η επιλογή βέλτιστης πολιτικής για την παραγωγή ενέργειας στην Ευρωπαϊκή Ένωση με στόχο την μείωση των επιπτώσεων της κλιματικής αλλαγής μέχρι το έτος 2050 (Forouli, Doukas et al 2019), η εύρεση της καλύτερης τοποθεσίας για την επεξεργασία τοξικών αποβλήτων (Yu et al 2016), η αποτελεσματικότερη διαχείριση του φυσικού αερίου και των κατανεμημένων ενεργειακών συστημάτων μεταξύ βιομηχανικών πάρκων σε περιόδους ελλείψεων (Xing et al 2019).

Συμπεραίνοντας, με την χρήση μεθόδων βελτιστοποίησης, οι αποφασίζοντες μπορούν να αναπτύσσουν ενεργειακές πολιτικές που είναι αποδοτικότερες, πιο επωφελείς οικονομικά και περιβαλλοντικά βιώσιμες. Μέσω μίας συστηματικής και ποσοτικής προσέγγισης αντιμετωπίζονται καλύτερα προβλήματα που εκ φύσεως είναι συχνά σύνθετα και περιέχουν πολλές αλληλεξαρτήσεις.

Κεφάλαιο 3.

Ελάχιστα (ναδίρ) Σημεία Εύρους
Αντικειμενικών σε Προβλήματα
Πολυστοχικού Προγραμματισμού και Νέα
Μεθοδολογία Εκτίμησής τους

3.1 Απώλεια Λύσεων και Χρονική Επιβάρυνση από την Επιλογή Ναδír Σημείων

Ένα χαρακτηριστικό της AUGMECON-Py είναι ότι βάζει τους περιορισμούς που προκύπτουν από τις αντικειμενικές συναρτήσεις, όπως και την εξωτερική αντικειμενική συνάρτηση με συγκεκριμένη σειρά. Οι αντικειμενικές συναρτήσεις με μεγαλύτερα εύρη (ranges) αντιστοιχούν στους πλέον εξωτερικούς βρόγχους προκειμένου να μην μείνει κάποια βέλτιστη λύση εκτός (Nikas et al, 2020). Έχει αναφερθεί στη βιβλιογραφία πως οι περιορισμοί που αντιστοιχούν σε αντικειμενικές συναρτήσεις με μεγαλύτερα εύρη είναι καλύτερο να μπαίνουν σε εσωτερικότερο βρόγχο για να είναι λιγότερος ο χρόνος τρεξίματος του προγράμματος (Forouli et al. 2022). Η ιδανική σειρά των αντικειμενικών θα μελετηθεί σε αυτό κεφάλαιο τις εργασίας με δοκιμές, παρατηρήσεις και θεωρητική ανάλυση για να διαπιστωθεί ποια είναι η σωστή αντιμετώπιση των αντικειμενικών συναρτήσεων για να έχουμε τον ταχύτερο χρόνο ολοκλήρωσης του προγράμματος και να μην παραλείπεται κάποια βέλτιστη λύση.

Θα κοιτάξουμε να δούμε πως η σχέση των δεδομένων επηρεάζει τον χρόνο. Ένας λόγος για τον αυξημένο χρόνο ασφαλώς είναι το μέγεθος των δεδομένων. Θα γίνουν δοκιμές αλλάζοντας την σειρά των αντικειμενικών συναντήσεων στην AUGMECON-Py και καταγράφοντας τον χρόνο. Θα γίνουν παρατηρήσεις σχετικά με το εύρος της κάθε αντικειμενικής σχετικά με το εύρος των υπολοίπων. Θεωρήθηκε σημαντικό επίσης να καταγράφεται σε κάθε πείραμα ο αριθμός των μοντέλων που υπολογίστηκαν, οι infeasibilities που έγιναν σε κάθε run και ο αριθμός των λύσεων κατά Pareto. Είναι σημαντικό να μην χάνεται καμία λύση ακόμα και αν ο χρόνος μειώνεται δραστικά.

Ένα κρίσιμο κομμάτι της επιτυχίας του αλγόριθμου είναι η εκτίμηση των σημείων ναδír. Η AUGMECON-Py στην περίπτωση που δεν γνωρίζουμε τα πραγματικά σημεία ναδír των αντικειμενικών συναρτήσεων, επιλέγει τις μικρότερες τιμές από τα payoff tables και τις πολλαπλασιάζει με έναν συντελεστή. Αλλάζοντας την τιμή του συντελεστή (σε 0.9, 0.4, 0.1, 0.01 κλπ.) θα λάβουμε τα αποτελέσματα που θα παρουσιαστούν και θα αναλυθούν στη συνέχεια. Θα μελετηθούν παραδείγματα με διαφορές ως προς το μέγεθος των δεδομένων.

Ένα ακόμα σημαντικό σημείο του αλγόριθμου είναι η σειρά που θα έχουν οι αντικειμενικές συναρτήσεις. Εκ προεπιλογής, η AUGMECON-Py αφού λύσει κάθε αντικειμενική συνάρτηση ξεχωριστά, τις ταξινομεί κατά φθίνουσα σειρά ως προς το μέγεθος του εύρους τους καθώς υπάρχει ο προαναφερθείς ισχυρισμός πως έτσι παίρνουμε τον καλύτερο χρόνο. Θα εξετάσουμε τις συνθήκες αλλά και την επιλογή ναδír σημείων κατά τις οποίες η αλλαγή σειράς αντικειμενικών δεν οδηγεί σε απώλεια λύσεων, προκειμένου να μπορέσουμε να προτείνουμε έναν αποτελεσματικό τρόπο εκτίμησης του συντελεστή nadir_cutoff. Θα μελετηθεί αν μειώνεται ο αριθμός

των επαναλήψεων και αν ναι σε τι ποσοστό και αν επηρεάζεται ο αριθμός των Pareto λύσεων που βρίσκει το πρόγραμμα.

3.2 Ανάλυση AUGMECON-Py - Εκτιμητής nadir cutoff c_i

Προκειμένου να επιτύχουμε τον καλύτερο χρόνο στον αλγόριθμο της AUGMECON-Py η σειρά των αντικειμενικών παίζει σημαντικό ρόλο όσον αφορά ποιος βρόγχος επανάληψης θα διασχιστεί πρώτος από το πρόγραμμα (Forouli et al. 2022). Υπάρχει η περίπτωση ένας βρόγχος που θα περιέχει πάρα πολλά στοιχεία να διασχιστεί εξολοκλήρου και να αφαιρούνται ορισμένες επαναλήψεις από το πλέγμα ενός σημαντικά μικρότερου βρόγχου λόγω της καινοτομίας με τον συντελεστή b που αναφέρθηκε στην ενότητα 2.7.2. Θα είναι προτιμότερη η σειρά αντικειμενικών που χρειάζονται λιγότερες συνολικά επαναλήψεις.

Επίσης σημαντικό ρόλο έχει η επιλογή nadir cutoff όπου ορίζει το ποσοστό με το οποίο πολλαπλασιάζεται η ελάχιστη τιμή από το payoff table για κάθε αντικειμενική συνάρτηση για να προσεγγίσουμε το πραγματικό σημείο ναδír. Προφανώς, όπως αναφέρθηκε και προηγουμένως, μικρότερο nadir cutoff θα οδηγήσει σε μικρότερη τιμή της προσέγγισης του πραγματικού σημείου ναδír. Όσο μικρότερη αυτή η τιμή, τόσο μεγαλύτερο και το πλέγμα (grid) των επαναλήψεων. Παρόλο που μεγαλύτερο πλέγμα σημαίνει μεγαλύτερη πιθανότητα να περιλαμβάνεται στην προσέγγιση το πραγματικό σημείο ναδír, αυτό αυξάνει σημαντικά τον χρόνο εκτέλεσης. Το να μην συμπεριληφθεί το σημείο ναδír θα οδηγήσει σε πιθανή απώλεια λύσεων. Θα γίνουν δοκιμές από πολύ χαμηλές τιμές προσέγγισης (π.χ. $0.1 \cdot$ ελάχιστη τιμή αντικειμενικής) μέχρι και $(1 \cdot$ ελάχιστη τιμή αντικειμενικής). Θα σχολιαστεί ο χρόνος και οι λύσεις σε παραδείγματα με διάφορου πλήθους και είδους τιμών.

3.2.1 Εμπειρική Ανάδειξη Προκλήσεων Εύρεσης Κατάλληλων Σημείων Ναδír και Σειράς Αντικειμενικών

Αντλώντας από δεδομένα που παρουσιάζονται στη βιβλιογραφία (Mavrotas et al, 2009), μελετάται ένα ενεργειακό πρόβλημα με μία ανάλυση σχετική με το ζητούμενο. Όπως αναφέρθηκε στην ενότητα 2.8 υπάρχει το πρόβλημα της ενσωμάτωσης ανανεώσιμων πηγών ενέργειας όπου πρέπει να συγκριθούν διαφορετικές ενεργειακές πολιτικές. Ουσιαστικά αφορά την επιλογή του κατάλληλου συνδυασμού τεχνολογιών για την αξιοποίηση επιδοτήσεων διαφόρων μεγεθών. Οι τεχνολογίες αυτές αφορούν την επένδυση σε πηγές ενέργειας με χαμηλές εκπομπές άνθρακα. Πιο συγκεκριμένα, εξετάζονται εκατό (100) ανεξάρτητες επιλογές επιδοτήσεων με οκτώ (8) διαφορετικού είδους τεχνολογίες. Οι τεχνολογίες αποτελούν τις μεταβλητές απόφασης (decision variables) του πολυκριτηριακού προβλήματος. Οι τρεις (3) αντικειμενικές συναρτήσεις του προβλήματος σχετίζονται είτε με οικονομικά κριτήρια όπως κόστη επένδυσης, είτε κριτήρια βιωσιμότητας όπως η μείωση των

ρύπων, είτε με κοινωνικοοικονομικά κριτήρια όπως η δημιουργία θέσεων εργασίας. Ως περιορισμός θεωρείται ο προϋπολογισμός της επένδυσης.

Στο πρώτο πείραμα με την AUGMECON-Py που έχουμε, θέλουμε να μεγιστοποιήσουμε τρεις αντικειμενικές συναρτήσεις (F1, F2, F3). Τα δεδομένα που χρησιμοποιήθηκαν σε αυτό αντλούνται από το παράδειγμα των Mavrotas et al, 2009. Αφού τρέξουμε ξεχωριστά τον αλγόριθμο της AUGMECON για κάθε αντικειμενική συνάρτηση βλέπουμε ότι η σειρά εύρους των αντικειμενικών κατά φθίνοντα μεγέθη είναι $F2 > F3 > F1$. Επομένως ο αλγόριθμος ορίζει εκ προεπιλογής αυτή την σειρά στα loops. Τροποποιώντας την σειρά αυτή στον κώδικα, έγιναν κατά σειρά οι δοκιμές με διαφορετικές σειρές αντικειμενικών και πήραμε τα ακόλουθα αποτελέσματα.

Βάζουμε αρχικά πολύ μικρές τιμή στον συντελεστή (0.1) για να περιλαμβάνονται όλες οι λύσεις και κρατώντας τον σταθερό καταγράφουμε τον χρόνο εκτέλεσης του αλγορίθμου και τον αριθμό των κατά Pareto βέλτιστων λύσεων (Πίνακας 3.1).

Πίνακας 3.1: Εναλλαγή σειράς αντικειμενικών με nadir cutoff 0.1

Εξωτερική obj προς max	Εξωτερικό loop (constraint)	Εσωτερικό loop (constraint)	Total Time (AUGMECON-R)	Number of Pareto Solutions
F1	F2	F3	2m 11s 779ms	31
F1	F3	F2	1m 8s 533ms	31
F2	F1	F3	2m 10s 533ms	31
F2	F3	F1	11s 402ms	31
F3	F1	F2	1m 7s 872ms	31
F3	F2	F1	12s 511ms	31

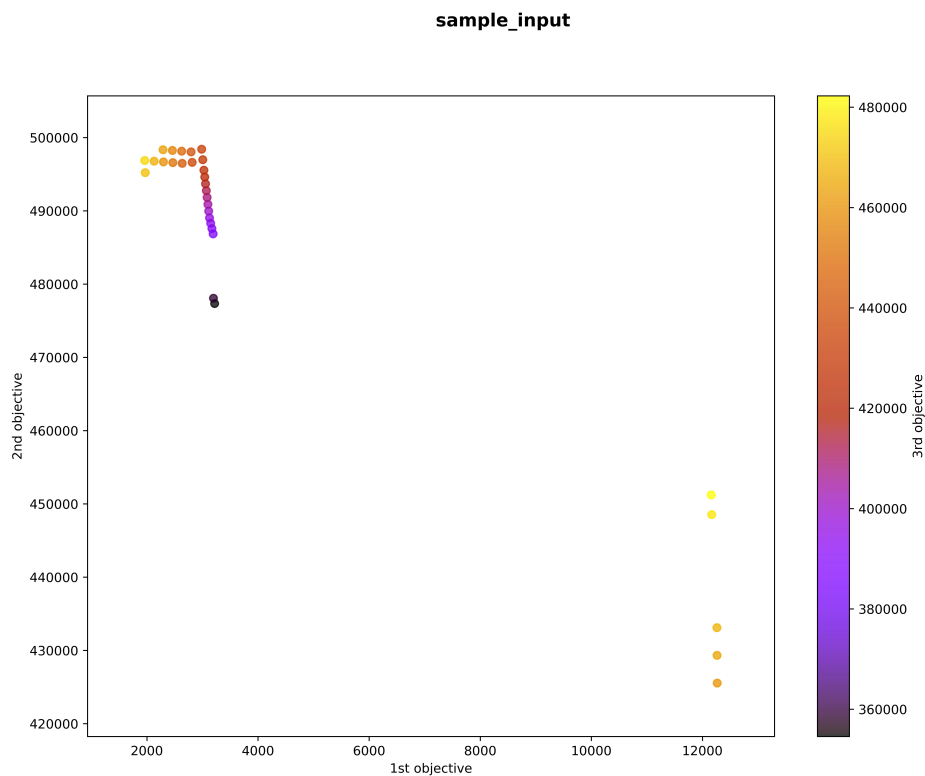
Παρατηρούμε ότι η ακολουθώντας την φθίνουσα σειρά F2, F3, F1 πετυχαίνουμε τον καλύτερο χρόνο και βρίσκουμε όλες τις λύσεις. Επίσης παρατηρούμε πως δεν σημαίνει απαραίτητα πως η αντίστροφη σειρά F1, F3, F2 θα δίνει τον χειρότερο χρόνο. Στην εκτέλεση αυτή του προγράμματος, περισσότερος χρόνος χρειάζεται για την σειρά F1, F2, F3 και F2, F1, F3 ενώ πολύ καλό χρόνο (σχεδόν όσο καλό και στην βέλτιστη περίπτωση) έχουμε βάζοντας τις αντικειμενικές στη σειρά F3, F2, F1.

Ίδια σειρά σε χρόνους παρατηρείται αν γίνουν οι δοκιμές με nadir cutoff 0.4 (Πίνακας 3.2) όπου πάλι έχουν βρεθεί και οι 31 λύσεις του Pareto front. Τέλος παρατηρείται ότι με nadir cutoff 0.1 και 0.4 η AUGMECON-Py βρίσκει και τις 31 λύσεις του μετώπου Pareto.

Πίνακας 3.2: Εναλλαγή σειράς αντικειμενικών με nadir cutoff 0.4

Εξωτερική obj προς max	Εξωτερικό loop (constraint)	Εσωτερικό loop (constraint)	Total Time (AUGMECON-2)	Number of Pareto Solutions
F1	F2	F3	1m 35s 529ms	31
F1	F3	F2	58s 142ms	31
F2	F1	F3	1m 36s 145ms	31
F2	F3	F1	11s 204ms	31
F3	F1	F2	53s 9ms	31
F3	F2	F1	13s 215ms	31

Οι λύσεις αυτές αναπαρίστανται στο ακόλουθο Γράφημα 3.1:



Γράφημα 3.1: Αναπαράσταση Pareto λύσεων του πρώτου πειράματος για nadir cutoff 0.4

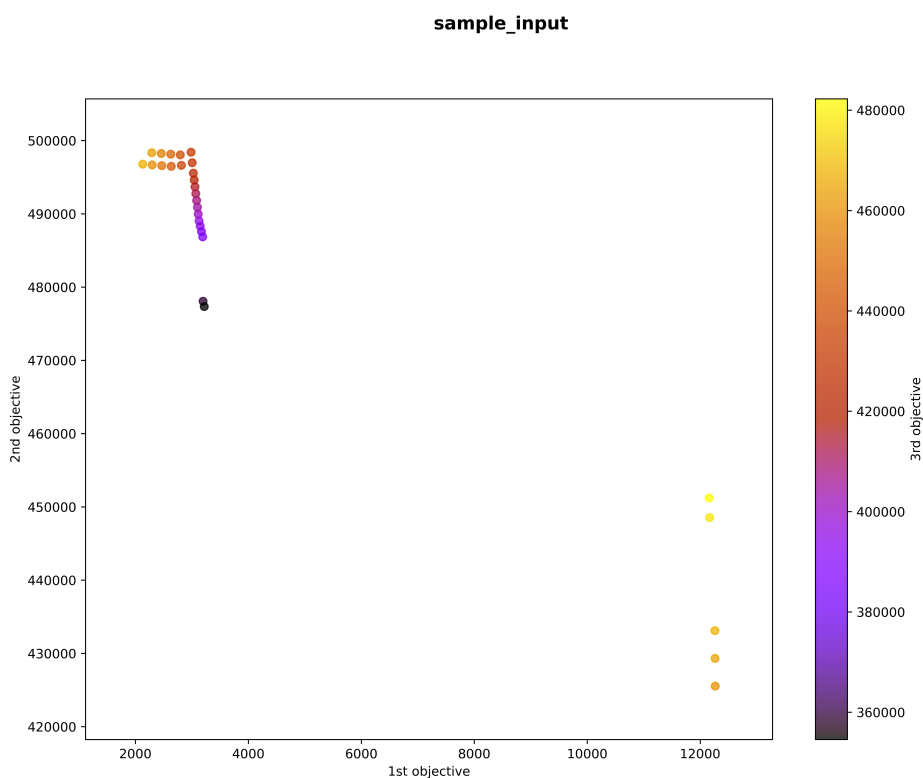
Με τα ίδια δεδομένα προβαίνουμε σε σταδιακή αύξηση του nadir cutoff για να δούμε αν από κάποιο σημείο και έπειτα δεν μας δίνει το πρόγραμμα όλες τις λύσεις. Όταν φτάνουμε σε nadir cutoff 0.66 παρατηρούμε τα ακόλουθα αποτελέσματα (Πίνακας 3.3):

Πίνακας 3.3: Εναλλαγή σειράς αντικειμενικών με nadir cutoff 0.66

Εξωτερική obj προς max	Εξωτερικό loop (constraint)	Εσωτερικό loop (constraint)	Total Time (AUGMECON-Py)	Number of Pareto Solutions
F1	F2	F3	1m 33s 263ms	31
F1	F3	F2	45s 492ms	31
F2	F1	F3	1m 0s 110ms	29
F2	F3	F1	10s 766ms	29
F3	F1	F2	48s 190ms	29
F3	F2	F1	12s 41ms	29

Παρατηρούμε στον Πίνακα 3.3 πως παρότι η σειρά των χρόνων συνεχίζει να είναι η ίδια, στις τέσσερις από τις έξι περιπτώσεις χάνουμε δύο κατά Pareto βέλτιστες λύσεις.

Οι λύσεις παρουσιάζονται στο Γράφημα 3.2:



Γράφημα 3.2: Αναπαράσταση Pareto λύσεων πρώτου πειράματος για nadir cutoff 0.66

Η προεπιλεγμένη τιμή της AUGMECON-Py για το nadir_cutoff ήταν 0.7 η οποία είναι πολύ κοντά στο όριο απώλειας λύσεων, επομένως είναι επιτακτική ανάγκη να εξεταστούν τρόποι ώστε να αποφευχθεί η μείωση του nadir_cutoff πάρα πολύ.

Οι λύσεις που απουσιάζουν είναι οι δύο κοντινότερες ως προς τον άξονα της πρώτης αντικειμενικής όπως παρατηρούμε συγκρίνοντας το Γράφημα 3.1 με το Γράφημα 3.2. Αυτό παρατηρούμε ότι συμβαίνει λόγω των σημείων ναδύρ.

Σε κάθε περίπτωση, το πρόγραμμα σε κάθε εκτέλεση δίνει τις τιμές ναδύρ που χρησιμοποίησε και το payoff table με τις τιμές των αντικειμενικών συναρτήσεων.

Για $nadir_cutoff = 0.4$ έχουμε για σειρά default (F2, F3, F1):

Πίνακας 3.4: Payoff Table για $nadir_cutoff = 0.4$

Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)	Αντικειμενική 1 (F1)
498400	432684	2986
451213	482264	12158
425529	460820	12266

Πίνακας 3.5: Μέγιστες Τιμές

Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)	Αντικειμενική 1 (F1)
498400	482264	12266

Πίνακας 3.6: Εκτίμηση Σημείων Ναδύρ με Συντελεστή $nadir_cutoff = 0.4$

Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)	Αντικειμενική 1 (F1)
170212	173074	1194

Πίνακας 3.7: Εκτίμηση Σημείων Ναδύρ με Συντελεστή $nadir_cutoff = 0.66$

Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)	Αντικειμενική 1 (F1)
280849	285571	1971

Πίνακας 3.8: Εκτίμηση Σημείων Ναδύρ με Συντελεστή $nadir_cutoff = 0.7$

Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)	Αντικειμενική 1 (F1)
297870	302879	2090

Στις παραπάνω περιπτώσεις, το μόνο που αλλάζει είναι τα σημεία ναδύρ. Οι μέγιστες τιμές παραμένουν ίδιες.

Στα αποτελέσματα βλέπουμε ότι οι δύο λύσεις που “χάνονται” έχουν σαν τιμή για την αντικειμενική συνάρτηση 1 (F1) τιμές 1961 και 1970 αντίστοιχα. Επομένως δεν περιλαμβάνονται στο grid στις περιπτώσεις (2) και (3) επομένως για αυτόν τον λόγο δεν εμφανίζονται ως λύσεις. Γενικότερα οποιοσδήποτε συντελεστής cutoff μεγαλύτερος του $2986/1961 \approx 0.65673$ δεν θα οδηγεί σε όλες τις λύσεις με αυτή τη σειρά των αντικειμενικών συναρτήσεων.

Δοκιμάζουμε το ίδιο πείραμα με σειρά αντικειμενικών F1, F2, F3:

Πίνακας 3.9: Payoff Table με Σειρά Αντικειμενικών F1, F2, F3 (nadir_cutoff = 0.4)

Αντικειμενική 1 (F1)	Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)
12266	425529	460820
2986	498400	432684
12158	451213	482264

Παρατηρούμε ότι είναι το ίδιο Payoff Table με τον Πίνακα 3.4 απλά με άλλη σειρά γραμμών.

Πίνακας 3.10: Μέγιστες Τιμές

Αντικειμενική 1 (F1)	Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)
12266	498400	482264

Πίνακας 3.11: Εκτίμηση Σημείων Ναδέρ με Συντελεστή nadir_cutoff = 0.4

Αντικειμενική 1 (F1)	Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)
1194	170212	173074

Σημείωση: Ίδια αποτελέσματα με τον Πίνακα 3.6

Πίνακας 3.12: Εκτίμηση Σημείων Ναδέρ με Συντελεστή nadir_cutoff = 0.66

Αντικειμενική 1 (F1)	Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)
1971	280849	285571

Σημείωση: Ίδια αποτελέσματα με τον Πίνακα 3.7

Πίνακας 3.13: Εκτίμηση Σημείων Ναδέρ με Συντελεστή nadir_cutoff = 0.7

Αντικειμενική 1 (F1)	Αντικειμενική 2 (F2)	Αντικειμενική 3 (F3)
2090	297870	302879

Σημείωση: Ίδια αποτελέσματα με τον Πίνακα 3.8

Ωστόσο σε αυτήν την περίπτωση οι λύσεις που έχουν στην πρώτη αντικειμενική συνάρτηση (F1) τιμές 1961 και 1970 εμφανίζονται ως Pareto λύσεις έπειτα από την εκτέλεση του προγράμματος και στις τρεις περιπτώσεις (4), (5), (6). Αυτό συμβαίνει και στην περίπτωση όπου έχουμε την σειρά F1, F3, F2 δηλαδή στις δύο περιπτώσεις όπου η F1 βρίσκεται στην θέση την συνάντησης μεγιστοποίησης και οι F2, F3 λειτουργούν ως constraints. Έτσι, επειδή οι αντίστοιχες τιμές των F2, F3 είναι εντός του πεδίου ορισμού παίρνουμε και τις τιμές για την F1 που δεν θα παίρναμε αν ήταν η F1 στη θέση των περιορισμών (2η ή 3η θέση στη σειρά).

Αντίστοιχα για τη δεύτερη αντικειμενική συνάρτηση (F2), βλέπουμε από τις λύσεις ότι η μικρότερη τιμή στο μέτωπο Pareto είναι 425529. Εδώ συμπίπτει με την ελάχιστη τιμή από το payoff table επομένως μέχρι και για συντελεστή 1 ($425529/425529 = 1$) δεν θα παραλειφθούν όταν είναι σε θέση constraint η F2.

Για την αντικειμενική 3 η ελάχιστη τιμή είναι 354568 στο Pareto front. Άρα έχει περιθώριο για συντελεστή μέχρι και $354568/432684 \approx 0.81946$. Έπειτα θα χάνονται λύσεις όταν βρίσκεται η F3 σε θέση περιορισμών (2η και 3η θέση).

Γενικά από το Payoff Table και από τα αποτελέσματα με πολύ χαμηλό συντελεστή έχουμε τις ελάχιστες τιμές των αντικειμενικών και τις ακραίες τιμές του μετώπου Pareto.

Επομένως για την Αντικειμενική Συνάρτηση F_i :

c_i = ακραία ελάχιστη τιμή κατά Pareto/ελάχιστη τιμή από payoff table

$$F_1 : c_1 = 1961/2986 \approx 0.65673$$

$$F_2 : c_2 = 425529/425529 = 1$$

$$F_3 : c_3 = 354568/432684 \approx 0.81946$$

Ιδανικός συντελεστής $c_{\min}$ είναι ο μικρότερος εκ των συντελεστών που αντιστοιχούν σε συναρτήσεις F_i που ανήκουν στους περιορισμούς του προβλήματος. Επομένως στην περίπτωση όπου έχουμε στη θέση των περιορισμών τις συναρτήσεις F3, F1 στους περιορισμούς του προβλήματος λόγω του ότι $\text{range}(F2) > \text{range}(F3) > \text{range}(F1)$, τότε ο συντελεστής $c_{\min}$ θα είναι ο μικρότερος εκ των c_1 και c_3 .

Άρα $c_{\min} = \min \{0.65673, 0.81946\} = 0.65673$

Η χρονική διαφορά ανάμεσα από την εκτέλεση του προγράμματος με `nadir_cutoff = 0.65` έναντι `0.7` μπορεί σε αυτό το απλό πείραμα να φαίνεται αμελητέα. Συγκεκριμένα, ο αλγόριθμος της AUGMECON-Py εκτελείται για μία επανάληψη Monte Carlo σε 11s 430ms για `nadir_cutoff = 0.65` και σε 10s 856ms για `nadir_cutoff = 0.7`, δηλαδή 574ms διαφορά ή αλλιώς 10% αύξηση του χρόνου. Εάν οι επαναλήψεις Monte Carlo ήταν 100 τότε η διαφορά θα ήταν 57sec 4ms. Η συγκεκριμένη περίπτωση δεν είναι από τις πιο εκτενείς υπολογιστικά ωστόσο στα πειράματα στις επόμενες ενότητες δεν θα ισχύει το ίδιο.

3.3 Βελτίωση μεθόδου AUGMECON-Py για την Εύρεση Κατάλληλων Σημείων Ναδέρ

Με βάση τις παραπάνω εμπειρικές παρατηρήσεις προκύπτουν τα ακόλουθα συμπεράσματα τα οποία θα μας βοηθήσουν να αναπτύξουμε μια μεθοδολογία για την αποδοτικότερη εκτίμηση των σημείων ναδέρ. Όταν μία συνάρτηση F_i βρίσκεται στη θέση των περιορισμών (2η και 3η θέση στο προηγούμενο πείραμά μας), ο συντελεστής `nadir_cutoff` θα πρέπει να είναι τουλάχιστον μικρότερος από το c_i .

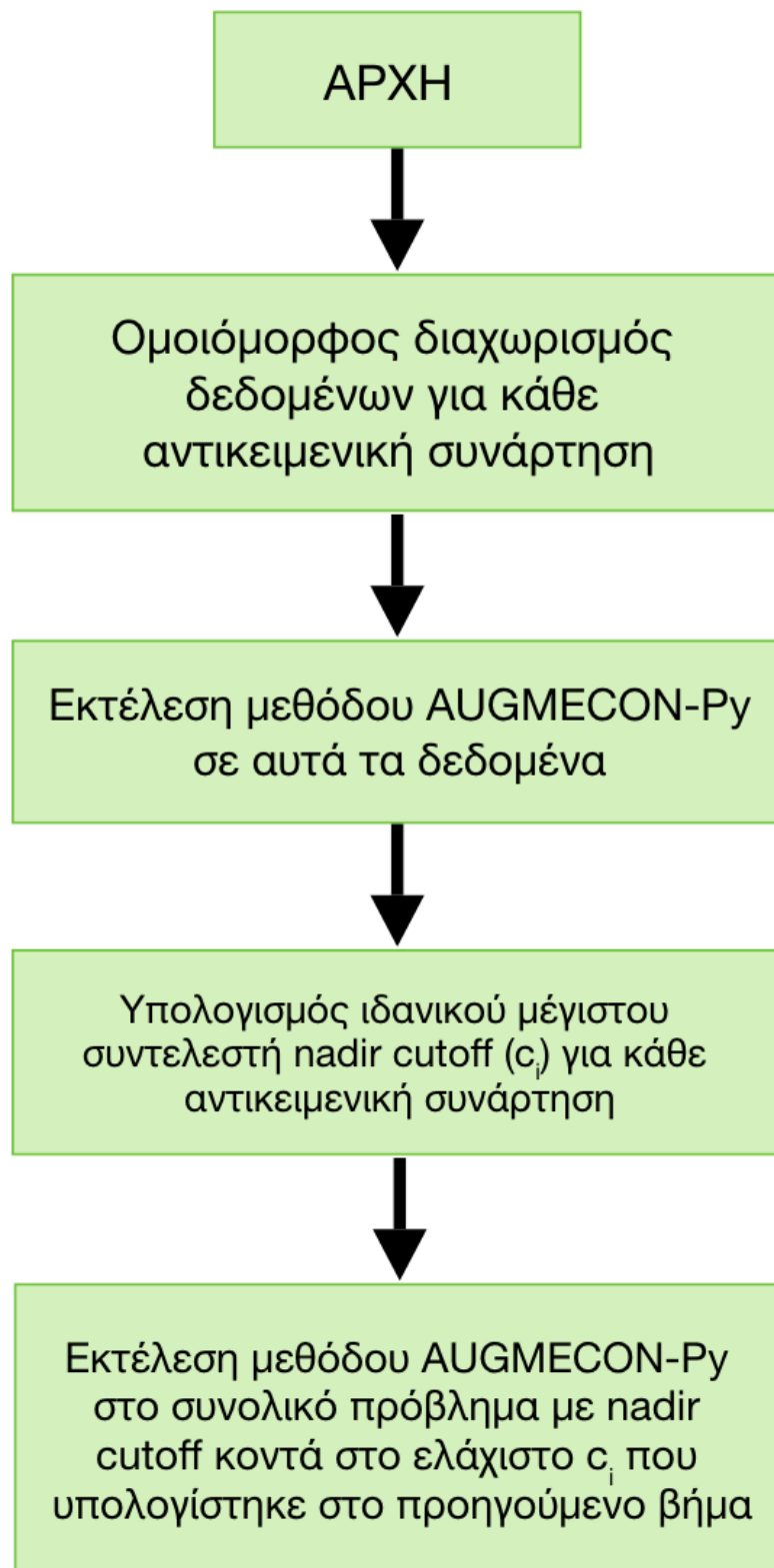
Γενικά, αυτά τα αποτελέσματα ελήφθησαν αφού είχε πρώτα βρεθεί το σύνολο των λύσεων Pareto βάζοντας στο πρόγραμμα πολύ χαμηλό συντελεστή. Ασφαλώς σε άλλα παραδείγματα με πολύ περισσότερα δεδομένα αυτό δεν θα μπορεί να είναι εφικτό. Το παράδειγμα αυτό λειτούργησε βοηθητικά για την κατανόηση της ιδέας γιατί μπορούμε να παίρνουμε διαφορετικά αποτελέσματα, δηλαδή να χάνουμε βέλτιστες λύσεις, με την αλλαγή της σειράς των αντικειμενικών συναντήσεων. Το θετικό στοιχείο της AUGMECON-Py και γενικότερα της AUGMECON-R είναι πως χάρη σε όλες τις επαναλήψεις που γλιτώνουμε συγκριτικά με προηγούμενες μεθόδους όπως η AUGMECON2, μπορεί να μειωθεί πολύ περισσότερο ο συντελεστής εκτίμησης των σημείων ναδέρ έτσι ώστε να μην χάνουμε λύσεις αλλά και να τερματίζει ο αλγόριθμος σε λογικά χρονικά όρια.

Το μεγαλύτερο πρόβλημα σχετικά με τον χρόνο εκτέλεσης παραμένει ο όγκος των δεδομένων. Προκειμένου να περιλαμβάνονται σίγουρα όλες οι λύσεις ένας πολύ μικρός συντελεστής για τα `nadir points` (π.χ. 0.1) θα εξασφάλιζε κάτι τέτοιο. Μία ιδανική περίπτωση θα ήταν αν εξετάζοντας ορισμένο μέρος των δεδομένων θα μπορούσε να υπολογιστεί με σχετική ακρίβεια ελάχιστος συντελεστής $c_{\min}$ και έπειτα με αυτόν ως δεδομένο να τρέξει το πρόγραμμα για όλα τα δεδομένα. Για παράδειγμα, μία ιδέα είναι η επιλογή του 5% ή 10% των δεδομένων για τον υπολογισμό του συντελεστή $c_{\min}$.

Επομένως, η κεντρική ιδέα βελτίωσης είναι να τροποποιηθεί ο υπάρχων αλγόριθμος της AUGMECON-Py έτσι ώστε να γίνεται μία αρχική εκτίμηση c_i του συντελεστή nadir cutoff με τον ακόλουθο τρόπο:

1. Ένα στάδιο στην αρχή του αλγορίθμου όπου το σύνολο των δεδομένων να διαχωρίζεται σε μικρότερα. Για παράδειγμα να επιλέγεται το 5% - 10% των λύσεων όσο πιο ομοιόμορφα γίνεται.
2. Εφαρμόζεται σε αυτά τα δεδομένα η μέθοδος AUGMECON-Py με πολύ μικρό συντελεστή nadir cutoff (π.χ. 0.1) και λαμβάνεται πληροφορία από το payoff table αυτού του προβλήματος σχετικά με την με την ελάχιστη τιμή κάθε αντικειμενικής του προβλήματος.
3. Από το output του προβλήματος λαμβάνεται η πληροφορία για τις ελάχιστες κατά pareto λύσεις ως προς κάθε αντικειμενική συνάρτηση.
4. Υπολογίζεται η εκτίμηση του συντελεστή c_i με χρήση όσων έγιναν γνωστά από τα βήματα 2 και 3 για κάθε αντικειμενική συνάρτηση F_i : c_i = ακραία ελάχιστη τιμή κατά Pareto/ελάχιστη τιμή από payoff table.
5. Εφαρμόζεται η μέθοδος AUGMECON-Py στο σύνολο των δεδομένων με συντελεστή nadir cutoff κοντά στην εκτίμηση c_i που βρέθηκε στο 4.

Η ιδέα αυτή θα εφαρμοστεί πειραματικά στα ολοκληρωμένα δεδομένα που αντλούνται από τη βιβλιογραφία (Van De Ven et al, 2022) τα οποία χαρακτηρίζονται από υπολογιστική πολυπλοκότητα για να αναδειχθούν τα πλεονεκτήματα της νέας αυτής μεθοδολογίας.



Γράφημα 3.3: Βελτίωση μεθόδου AUGMECON-Py

3.4 Ενσωμάτωση της Μεθοδολογίας στο Εργαλείο της AUGMECON-Py

Προκειμένου να υλοποιηθεί η ιδέα που αναφέρθηκε στην προηγούμενη ενότητα θα προστεθούν δύο ακόμη συναρτήσεις στο πρόγραμμα της AUGMECON-Py.

Η πρώτη συνάρτηση ονομάζεται “process_excel” και δέχεται δύο παραμέτρους. Η πρώτη είναι το αρχείο “sample_input.xlsx” που περιέχει τα δεδομένα σε μορφή xlsx spreadsheet. Διαθέτει ένα φύλλο εργασίας για κάθε αντικειμενική συνάρτηση και για κάθε περιορισμό. Κάθε αντικειμενική συνάρτηση αποτελείται από έναν αριθμό από στήλες ή αλλιώς levels. Στις περιπτώσεις που θα μελετήσουμε θα είναι subsidy_levels και θα είναι συνολικά 100. Θα δημιουργηθεί ένα νέο αρχείο xlsx spreadsheet όπου θα περιέχει μόνο ορισμένες από τις στήλες του αρχικού αρχείου. Θα ήταν ιδανικό να μην κρατηθούν απλά τα πρώτα k levels (πρώτες k στήλες) καθώς μπορεί για παράδειγμα να αυξάνεται γραμμικά η τιμή από το ένα level στο επόμενο και να χαθεί σημαντική πληροφορία από τα μεγαλύτερα levels. Επομένως, η δεύτερη παράμετρος θα είναι ένας ακέραιος αριθμός n ο οποίος θα υποδηλώνει το ανά πόσα levels θα επιλέγεται η επόμενη στήλη-level. Οι στήλες που θα επιλέγονται θα αποθηκεύονται σε ένα νέο αρχείο “new_sample_input.xlsx” που θα αποθηκευτεί στο νέο αρχείο που θα δοθεί ως παράμετρος στον αλγόριθμο βελτιστοποίησης AUGMECON-Py.

Η δεύτερη συνάρτηση θα δέχεται ως input το output που έδωσε ο αλγόριθμος σε μορφή xlsx spreadsheet. Από τις στήλες που αντιστοιχούν στις τιμές κάθε αντικειμενικής συνάρτησης σε κάθε περίπτωση, βρίσκει για κάθε αντικειμενική συνάρτηση την ελάχιστη τιμή. Έπειτα, υπολογίζει για κάθε αντικειμενική συνάρτηση τον συντελεστή c_i όπως διατυπώθηκε στο Κεφάλαιο 3 (c_i = ακραία ελάχιστη τιμή κατά Pareto/ελάχιστη τιμή από payoff table). Έπειτα, προτίθενται οι γραμμές κώδικα που υπολογίζουν τον c_{min} , δηλαδή τον ελάχιστο από τους συντελεστές που ανήκουν στις αντικειμενικές συναρτήσεις που χρησιμοποιούνται ως περιορισμοί.

Τέλος, εκτελείται το πρόγραμμα με τον συνολικό όγκο δεδομένων ως είσοδο (το αρχείο “sample_input.xlsx”) και στην θέση της παραμέτρου nadir_cutoff δίνεται το C_{min} .

3.5 Πιθανές Μελλοντικές Βελτιώσεις της AUGMECON-Py ως Πρόγραμμα Η/Υ

Προκειμένου να δοκιμαστεί το εκτελέσιμο πρόγραμμα σε Python και να βγει το πόρισμα ότι αντιπροσωπεύει την θεωρητική προσέγγιση, έπρεπε να δοκιμαστούν αρκετά ξεχωριστά οι μέθοδοί του, ειδικότερα εκείνες που σχετίζονται με την

διαχείριση των αντικειμενικών συναρτήσεων και την εύρεση των σημείων του μετώπου Pareto. Όσο πιο έμπειρος είναι ο προγραμματιστής και όσο πιο πολύ χρόνο αφιερώσει σε αυτό το κομμάτι τόσο πιο μεγάλη είναι η πιθανότητα να βρεθούν καλύτερες υλοποιήσεις μεθόδων ή να εντοπιστούν προγραμματιστικά λάθη. Στο μέλλον θα είναι πολύ χρήσιμο αν ακόμα περισσότεροι προγραμματιστές κάνουν τέτοιες δοκιμές.

Μερικοί από τους τρόπους που θα μπορούσε να βελτιωθεί ακόμα περισσότερο το πρόγραμμα είναι οι ακόλουθοι:

1. Καλύτερη εφαρμογή του θεωρητικού αλγορίθμου σε κώδικα
 - Ο προγραμματιστής πρέπει να μπορεί να αποτυπώσει πλήρως τον αρχικό αλγόριθμο στο πρόγραμμα αλλά και να χρησιμοποιήσει χαρακτηριστικά της γλώσσας προγραμματισμού προς όφελος της επίδοσης, τροποποιώντας τον αλγόριθμο. Για παράδειγμα η AUGMECON-Py αποτελεί τροποποίηση του AUGMECON-R εκμεταλλευόμενη στοιχεία της γλώσσας προγραμματισμού Python
 - Η γλώσσα προγραμματισμού Python χαρακτηρίζεται από μεγάλη ασφάλεια στον τομέα της μνήμης. Ωστόσο μία γλώσσα όπως η C++, λόγω τις διαφορετικής διαχείρισης της μνήμης, μάλλον θα μπορούσε να βοηθήσει σημαντικά τον χρόνο εκτέλεσης. Θα υπήρχε όμως μεγαλύτερος κίνδυνος να γίνουν σφάλματα σχετικά με τη μνήμη για αυτό και η δυσκολία υλοποίησης στην C++ θα είναι μεγαλύτερη.
2. Δημιουργία του βέλτιστου κώδικα
 - Πολύ συχνά βελτιώνοντας τον κώδικα χρησιμοποιώντας άλλες βιβλιοθήκες, άλλον solver, ή αλλάζοντας τις συναρτήσεις του προγράμματος μειώνεται η πολυπλοκότητα και ο χρόνος εκτέλεσης. Επίσης, άμα γίνει χρήση του παράλληλου προγραμματισμού (Parallel Computing) βελτιώνεται συχνά η απόδοση
3. Διαχείριση Σφαλμάτων και Προσεγγίσεων
 - Ένας ακόμα παράγοντας αποκλίσεων διαφορετικών εφαρμογών του ίδιου αλγορίθμου είναι η προσέγγιση αριθμητικών τιμών πολύ μεγάλων ή μικρών που δεν μπορούν να αποθηκευτούν αυτούσιες στη μνήμη του υπολογιστή. Η ακρίβεια και το robustness κάθε προγράμματος μπορούν να αλλάζουν.
4. Διαχείριση των πολλών Δεδομένων
 - Ο τομέας διαχείρισης δεδομένων αναπτύσσεται διαρκώς. Ένας προγραμματιστής θα μπορούσε να δημιουργήσει καλύτερες μεθόδους επεξεργασίας, ταξινόμησης και μεταφοράς των δεδομένων που θα είναι πιο εύκολο αυτά να αποτελέσουν είσοδο ενός αλγορίθμου βελτιστοποίησης.

- Στην AUGMECON-Py, η είσοδος των δεδομένων γινόταν μέσω ενός κατάλληλα διαμορφωμένου φύλλου Excel το οποίο για να ανοίξει και να διαβαστεί από το πρόγραμμα το επιβαρύνει χρονικά περισσότερο πιθανόν από έναν άλλον τρόπο όπως ένα αρχείο CSV. Το CSV αρχείο είναι πιο συμπαγές και ελαφρύ από το Excel που σημαίνει ότι χρειάζεται λιγότερο αποθηκευτικό χώρο και μεταφέρεται πιο γρήγορα. Επιπλέον, λόγω της μορφής του είναι λιγότερο πιθανόν να αλλοιωθεί ωστόσο είναι πιο εύκολο να γίνουν μετατροπές σε αυτό αν κρίνεται απαραίτητο.

5. Βελτιώσεις ως προς τη διεπαφή χρήστη (User Interface)

- Η χρήση της AUGMECON-Py που παρουσιάζεται στο paper (Forouli et al. 2022) μπορεί να γίνει ακόμα πιο φιλική στον χρήστη. Για παράδειγμα αντί να γίνεται χρήση της μόνο διαμέσου του κώδικα μπορεί να δημιουργηθεί μία εφαρμογή για τον υπολογιστή όπου χρήστες λιγότερο έμπειροι σε προγραμματισμό να μπορούν να περνούν ως είσοδο αρχεία δεδομένων και να προσδιορίζουν τις παραμέτρους κατάλληλα.
- Ένας νέος χρήστης ακόμα και αν δεν γνωρίζει τόσο καλά προγραμματισμό μπορεί να έχει ιδιαίτερα σημαντικές γνώσεις σε τομείς όπως τα μαθηματικά, τα οικονομικά, η ενέργεια ή το περιβάλλον. Επομένως παρουσιάσει ιδιαίτερο ενδιαφέρον η ερμηνεία των αποτελεσμάτων που θα δώσει ένας τέτοιος χρήστης.
- Παρόλο που μπορεί μία τέτοια λειτουργία διεπαφής χρήστη (UI) να είναι σε κάποιες περιπτώσεις πιο χρονοβόρα τα οφέλη θα είναι πολλά. Για παράδειγμα, η επίλυση πολλών προβλημάτων θα οδηγήσει σε ένα μεγάλο σύνολο δεδομένων (data set) που πιθανόν να περιλαμβάνει προβλήματα που θα μοιάζουν σε κάποιο βαθμό. Έτσι, αν ο αποφασίζων έχει αυτές τις πληροφορίες θα μπορεί να συγκρίνει τα αποτελέσματα του με τα υπόλοιπα, να κρίνει κατά πόσο είναι όμοια τα προβλήματα και σε τι βαθμό, και επομένως να διευκολύνεται η τελική του απόφαση.
- Ένα ακόμα προτέρημα της βελτίωσης της διεπαφής του χρήστη και επομένως η ακόμα μεγαλύτερη χρήση του αλγορίθμου είναι ότι έτσι αυξάνεται η πιθανότητα να εντοπιστούν τυχόν σφάλματα που μπορούν να διορθωθούν σε επόμενες εκδόσεις του προγράμματος.

Συμπεραίνοντας, η μεγαλύτερη ενασχόληση χρηστών με εμπειρία στον προγραμματισμό είναι καθοριστική για την βελτίωση ως προς τον χρόνο του αλγορίθμου. Όπως υποστηρίζεται σε αυτή την εργασία, η AUGMECON-Py είναι πολύ σημαντικό και χρήσιμο εργαλείο στην Πολυκριτηριακή Ανάλυση Αποφάσεων και η ενθαρρύνεται η χρήση της.

Κεφάλαιο 4.

Εφαρμογή της Μεθοδολογίας σε
Ενεργειακά Προβλήματα Εύρεσης
Βέλτιστων Χαρτοφυλακίων
Χρηματοδότησης Τεχνολογιών Χαμηλών
Εκπομπών

4.1 Εφαρμογή της Νέας Μεθοδολογίας σε ένα Υπολογιστικά Πολύπλοκο Πρόβλημα Ενεργειακής Βελτιστοποίησης

4.1.1: Πειραματική Μελέτη Ενεργειακού Προβλήματος με το 10% των Αρχικών Δεδομένων

Σε αυτό το κεφάλαιο θα εφαρμοστεί πειραματικά ή νέα μεθοδολογία έτσι ώστε να εξεταστεί μία διαφορετική μορφή και ένα μεγαλύτερο μέγεθος δεδομένων δεδομένων και να παρατηρηθούν οι διαφορές στις τιμές του συντελεστή c_i .

Το συγκεκριμένο πείραμα αφορά παρόμοια θεματική με αυτή που συζητήθηκε στην ενότητα 3.2.1 με τη διαφορά ότι υπάρχει αρκετά μεγαλύτερος όγκος δεδομένων. Οι τεχνολογίες που εξετάζονται είναι: τα φωτοβολταϊκά, η συγκεντρωμένη ηλιακή ενέργεια, η αιολική ενέργεια χερσαίου ανέμου, η αιολική ενέργεια υπεράκτιου ανέμου, η γεωθερμική ενέργεια, η πυρηνική ενέργεια, η ενέργεια βιομάζας και η ενέργεια βιοκαυσίμων. Ως αντικειμενικές συναρτήσεις έχουμε την εκπομπή ρύπων, τις θέσεις εργασίας το έτος 2025 και τις θέσεις εργασίας το 2030. Πάλι υπάρχει ως περιορισμός του πολυκριτηριακού προβλήματος ο συνολικός προϋπολογισμός της επένδυσης. Η συγκεκριμένη εφαρμογή της AUGMECON-Py έχει μεγαλύτερη πολυπλοκότητα γιατί αφορά την Ευρώπη και όπως είχε αποδειχτεί στο paper “COVID-19 recovery packages can benefit climate targets and clean energy jobs, but scale of impacts and optimal investment portfolios differ among major economies” (Van De Ven et al, 2022), ο ανταγωνισμός των τεχνολογιών οδηγεί σε πολύπλοκα μέτωπα Pareto, και ότι εξαιτίας αυτού στην αρχική ανάλυση είχε διπλασιαστεί ο αριθμός των επιπέδων στα δεδομένα.

Στην αρχική φάση του πειράματος επιλέξαμε στο πρόγραμμα $n=10$ έτσι ώστε από τα 100 levels να κρατιέται 1 level ανά 10 levels στο νέο αρχείο. Επειδή ο συνολικός αριθμός των levels είναι 100, εντέλει επιλέχθηκαν $100/10 = 10$ levels (10% των συνολικών).

Αρχικά τρέξαμε τον αλγόριθμο με πολύ χαμηλό nadir cutoff για να βρούμε όλες τις λύσεις. Βρέθηκε επίσης το Payoff Table και από εκεί οι μέγιστες και ελάχιστες τιμές για κάθε αντικειμενική συνάρτηση.

Πίνακας 4.1: Payoff Table για nadir_cutoff = 0.4

F1	F2	F3
76040	1071963	834836

F1	F2	F3
73457	919382	910875
43488	1422164	614684

Τα εύρη (ranges) των αντικειμενικών υπολογίζονται με τη διαφορά της μέγιστης με την ελάχιστη τιμή τους.

Πίνακας 4.2: Μέγιστες Τιμές

F1	F2	F3
32552	502782	296191

Όπως συνέβη και στην προηγούμενη περίπτωση, έχουμε ότι $\text{range}(F2) > \text{range}(F3) > \text{range}(F1)$ άρα η βέλτιστη σειρά θα είναι και πάλι F2, F3, F1.

Από τις Pareto λύσεις που πήραμε βλέπουμε ότι οι ελάχιστες τιμές που εμφανίζονται για κάθε αντικειμενική συνάρτηση είναι:

- Για την F1 η ελάχιστη τιμή της (41516) παρατηρείται στην λύση:

$$(F1, F2, F3) = (41516, 1391791, 691613)$$

- Για την F2 η ελάχιστη τιμή της (919382) παρατηρείται στην λύση:

$$(F1, F2, F3) = (73457, 919382, 910875)$$

- Για την F3 η ελάχιστη τιμή της (614684) παρατηρείται στην λύση:

$$(F1, F2, F3) = (43488, 1422164, 614684)$$

Άρα παίρνουμε του λόγους c_i για το μέγιστο nadir cutoff όταν μία αντικειμενική βρίσκεται στη θέση των περιορισμών (2η και 3η θέση).

$$F_1 : c_1 = 41516/43488 \approx 0.95465$$

$$F_2 : c_2 = 919382/919382 = 1$$

$$F_3 : c_3 = 614684/614684 = 1$$

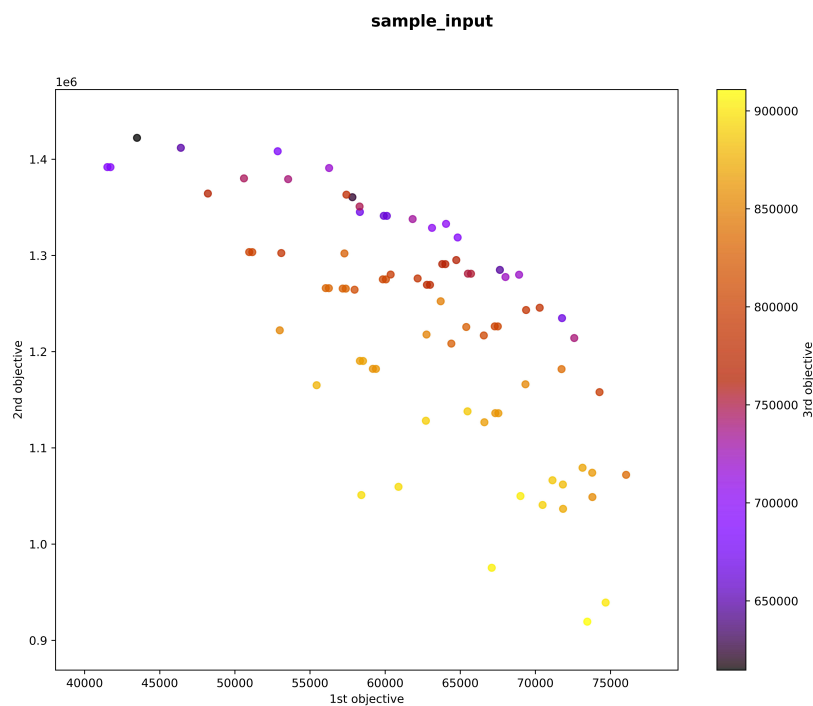
Επομένως μόνο για την περίπτωση της 1ης αντικειμενικής χρειάζεται να το επαληθεύσουμε αφού για τις F2, F3 το σημείο nadir συμπίπτει με την ελάχιστη τιμή του payoff table. Πράγματι, για nadir cutoff 0.96 έχουμε:

Πίνακας 4.3: Δοκιμές για Nadir Cutoff 0.96

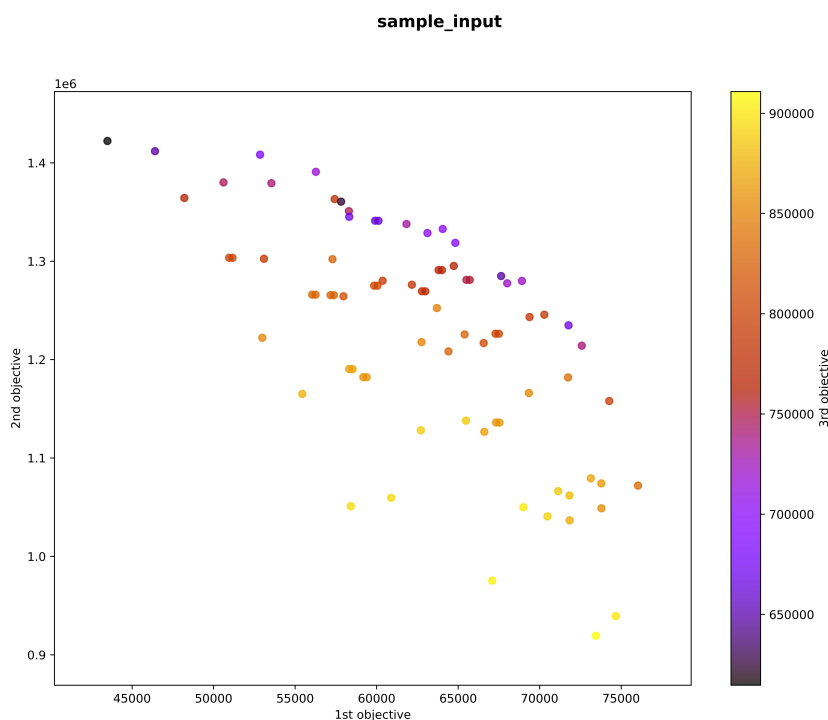
Εξωτερική (max)	Εξωτερικό loop (constraint)	Εσωτερικό loop (constraint)	Total time (AUGMECON-R)	Num of Pareto Solutions
F2	F3	F1	14s 822ms	79
F3	F2	F1	15s 278ms	79

Παρατήρηση: χάνονται δύο λύσεις από τις 81 συνολικά

Όπως φαίνεται και στα γραφήματα (Γράφημα 4.1, Γράφημα 4.2):



Γράφημα 4.1: Αναπαράσταση Pareto λύσεων πρώτου πειράματος για nadir cutoff 0.1



Γράφημα 4.2: Αναπαράσταση Pareto λύσεων πρώτου πειράματος για nadir cutoff 0.96

Παρατηρούμε ότι στο 2ο γράφημα (Γράφημα 4.2) απουσιάζουν οι δύο αριστερότερες λύσεις ως προς τον άξονα της πρώτης αντικειμενικής συνάρτησης καθώς σε αυτήν την περίπτωση είναι εκτός του grid.

Εντέλει βρίσκουμε ότι:

$$c_{min} = \min\{c_1, c_3\} = c_1 = 0.95465$$

Ο χρόνος που χρειάστηκε για την εύρεση του c_{min} ήταν περίπου 32sec. Με χρήση της πληροφορίας για nadir_cutoff = 0.95 το πρόβλημα με το σύνολο των δεδομένων χρειάστηκε περίπου 2mins 28sec. Συνολικά και για τα 2 στάδια χρειάστηκαν περίπου 3mins. Σε αυτήν την περίπτωση υπάρχει μεγαλύτερη σιγουριά για την πληρότητα του μετώπου pareto καθώς ο συντελεστής nadir_cutoff υπολογίστηκε στο πρώτο στάδιο με βάση τα υπάρχοντα δεδομένα αντί να εκτιμηθεί. Βρέθηκαν όλες οι λύσεις μετά το δεύτερο στάδιο.

Στην περίπτωση που εκτελεστεί το πρόγραμμα με nadir_cutoff = 0.7 απευθείας θα χρειαστεί περίπου 2mins 48 sec και βρέθηκαν όλες οι λύσεις προφανώς καθώς $0.7 < 0.95$. Το nadir_cutoff = 0.7 ήταν μία καλή εκτίμηση καθώς ο αποφασίζων πέτυχε ταυτόχρονα και γρηγορότερο χρόνο αλλά βρήκε και τις ίδιες λύσεις με την προηγούμενη περίπτωση. Ωστόσο δεν μπορεί να είναι σίγουρος για την πληρότητα των λύσεων του μετώπου Pareto επομένως πιθανόν μειώσει τον συντελεστή nadir_cutoff για να παρατηρήσει αν υπάρχουν επιπλέον λύσεις.

Στην περίπτωση που εκτελεστεί το πρόγραμμα με $\text{nadir_cutoff} = 0.5$ απευθείας θα χρειαστεί περίπου 3 mins. Ο αποφασίζων ίσως είναι πιο σίγουρος για την πληρότητα των λύσεων συγκριτικά με την προηγούμενη περίπτωση όμως ο χρόνος είναι μεγαλύτερος από εκείνον της μεθόδου με τα δύο στάδια που προτάθηκε. Επομένως και για κάθε μικρότερη τιμή του nadir_cutoff ο χρόνος θα μεγαλώνει όλο και περισσότερο, για παράδειγμα με $\text{nadir_cutoff} = 0.3$ ο χρόνος είναι 3mins 2sec.

Στον παρακάτω Πίνακα 4.4 καταγράφονται οι χρόνοι για τις παραπάνω περιπτώσεις:

Πίνακας 4.4: Σύγκριση χρόνου εκτέλεσης 2-Stage AUGMECON-Py

Στάδιο 1: c_{min} με 10% των δεδομένων	Στάδιο 2: AUGMEC ON-Py ($\text{nadir_cutoff} = 0.95$)	Συνολικός χρόνος σταδίων 1 και 2:	AUGMEC ON-Py ($\text{nadir_cutoff} = 0.7$)	AUGMEC ON-Py ($\text{nadir_cutoff} = 0.5$)	AUGMEC ON-Py ($\text{nadir_cutoff} = 0.3$)	AUGMEC ON-Py ($\text{nadir_cutoff} = 0.1$)
31s 88ms	2m 28s 440ms	2m 59s 528ms	2m 48s 418ms	2m 59s 739ms	3m 2s 354ms	3m 24s 712ms

Παρατηρείται ότι το χρονικό κέρδος στην περίπτωση όπου γίνεται μία επανάληψη Monte Carlo δεν φαίνεται να είναι ιδιαίτερα σημαντικό (μόλις 2s 826ms γρηγορότερο από την περίπτωση όπου εκτελεστεί η μέθοδος με $\text{nadir_cutoff} = 0.3$). Ωστόσο θα πρέπει να τονιστεί ότι συνήθως γίνεται ανάλυση Monte Carlo. Επομένως με 100 Monte Carlo επαναλήψεις οι συνολικοί χρόνοι φαίνονται στον ακόλουθο πίνακα:

Πίνακας 4.5: Σύγκριση χρόνου εκτέλεσης 2-Stage AUGMECON-Py με Monte Carlo

Στάδιο 1: c_{min} με 10% των δεδομένων	Στάδιο 2: AUGMEC ON-Py ($\text{nadir_cutoff} = 0.95$)	Συνολικός χρόνος σταδίων 1 και 2:	AUGMEC ON-Py ($\text{nadir_cutoff} = 0.7$)	AUGMEC ON-Py ($\text{nadir_cutoff} = 0.5$)	AUGMEC ON-Py ($\text{nadir_cutoff} = 0.3$)	AUGMEC ON-Py ($\text{nadir_cutoff} = 0.1$)
32s	4h 7m 24s	4h 7m 56s	4h 40m 42s	4h 59m 34s	5h 3m 55s	5h 41m 11s

Αριθμός Monte Carlo επαναλήψεων: 100

Παρατηρείται σημαντικό χρονικό κέρδος με την χρήση της μεθόδου 2-Stage AUGMECON-Py. Στον Πίνακα 4.6 φαίνεται το ποσοστό της αύξησης του χρόνου συγκριτικά με την 2-Stage AUGMECON-Py μέθοδο αναλόγως της τιμής του nadir_cutoff .

Πίνακας 4.6: Ποσοστό Αύξησης Χρόνου με την Χρήση Μόνο της AUGMECON-Py

AUGMECON-Py ($\text{nadir_cutoff} = 0.7$)	AUGMECON-Py ($\text{nadir_cutoff} = 0.5$)	AUGMECON-Py ($\text{nadir_cutoff} = 0.3$)	AUGMECON-Py ($\text{nadir_cutoff} = 0.1$)
13,2%	20,8%	22,6%	37,6%

4.1.2: Πειραματική Μελέτη Ενεργειακού Προβλήματος με το 20% των Αρχικών Δεδομένων

Θα επαναληφθεί το παραπάνω πείραμα με περισσότερα δεδομένα στο πρώτο στάδιο προκειμένου να εξεταστεί θα πόσο διαφέρει ο συντελεστής c_{min} . Εάν γίνει το πρώτο στάδιο με περισσότερα δεδομένα ανά αντικειμενική συνάρτηση (20% των δεδομένων με $n=5$) παίρνουμε τα ακόλουθα αποτελέσματα:

Πίνακας 4.7: Payoff Table για το 20% των Δεδομένων ($nadir_cutoff = 0.1$)

F1	F2	F3
43380	1433293	632535
70196	974833	913690
76261	1023595	865528

Πίνακας 4.8: Ελάχιστες τιμές αντικειμενικής από Pareto Λύσεις

F1	F2	F3
43380	868925	596986

Σημείωση: Έγινε πείραμα με πολύ χαμηλό $nadir_cutoff$

Πίνακας 4.9: Συντελεστές c_i

c_1 (Αντικειμενική F1)	c_2 (Αντικειμενική F2)	c_3 (Αντικειμενική F3)
1	0,891357801797846	0,943799157358881

Παρατηρούμε ότι γενικά οι c_i είναι αρκετά μεγάλοι και εδώ, γεγονός που βοηθάει πολύ στο να μην χάνουμε λύσεις και να έχουμε καλό χρόνο εκτέλεσης. Επειδή $range(F2) > range(F3) > range(F1)$ η συνάρτηση F2 θα είναι η αντικειμενική συνάρτηση του προβλήματος και οι συναρτήσεις F1, F3 θα είναι σε θέση περιορισμών. Επομένως $c_{min} = \min\{c_1, c_3\}$.

Άρα $c_{min} \approx 0.944$

Στο πείραμα στην υποενότητα 4.1.1 ο συντελεστής c_{min} είχε βρεθεί $c_{min} = 0.95465 \approx 0.955$. Η μείωση αυτή σε $c_{min} \approx 0.944$ οφείλεται στο ότι με περισσότερα δεδομένα, η ελάχιστη Pareto λύση είναι μικρότερη από εκείνη που είχε βρεθεί στην προηγούμενη περίπτωση, επομένως μίκρυνε και ο συντελεστής c_{min} . Ωστόσο, οι δύο αυτοί συντελεστές έχουν πολύ μικρή διαφορά και σε περιπτώσεις όπου τα δεδομένα θα έχουν μία σχέση μεταξύ τους δεν θα συμβαίνουν ακραίες

μεταβολές με την προσθήκη επιπλέον δεδομένων στο πρώτο στάδιο. Στην επόμενη ενότητα θα μελετηθεί η περίπτωση με τυχαία δεδομένα.

Όσον αφορά τον χρόνο εκτέλεσης, με $\text{nadir_cutoff} = 0.1$ ο αλγόριθμος ολοκληρώθηκε σε 1min 26sec έναντι των 32sec που είχαν χρειαστεί στο προηγούμενο πείραμα (Πίνακας 4.10). Αυτή η αύξηση περίπου ενός λεπτού για να επιτευχθεί η σιγουριά του εκτιμητή πως ο συντελεστής θα πρέπει να είναι μεγαλύτερος από 0.944 έναντι 0.955 δεν χρειάζεται, ακόμα και αν σε πολλές επαναλήψεις Monte Carlo δεν έχει σημαντική διαφορά. Ένας εκτιμητής θα μπορούσε με το 10% των δεδομένων στο πρώτο στάδιο να πάρει την απόφαση να χρησιμοποιήσει nadir_cutoff 0.95 ή ακόμα και 0.9 στο δεύτερο στάδιο και να έχει διαθέτει αρκετά μεγάλη σιγουριά για την πληρότητα του μετώπου Pareto.

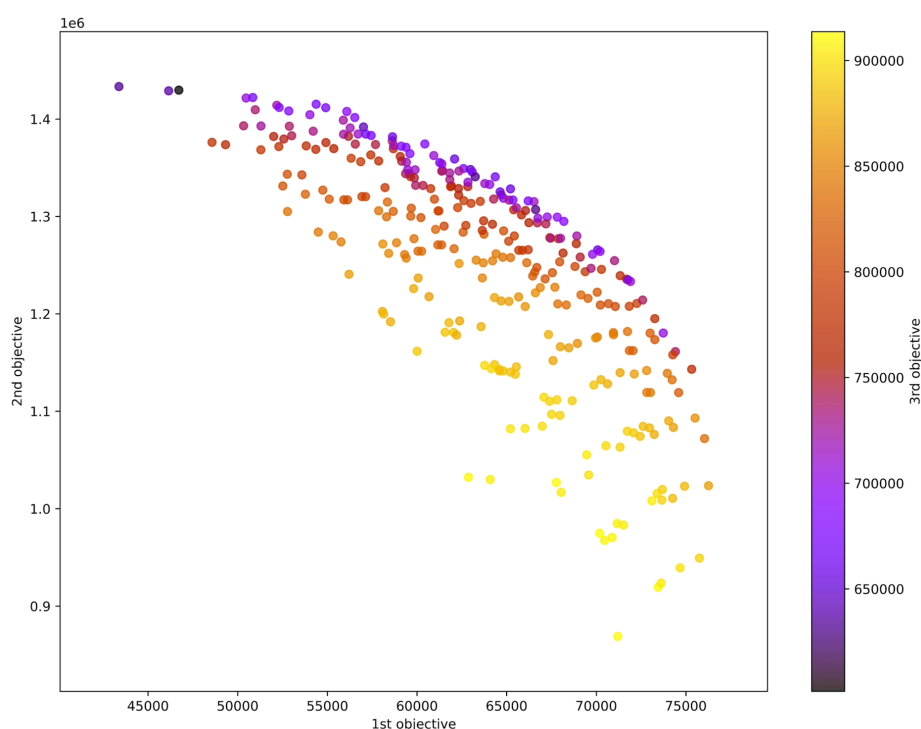
Πίνακας 4.10: Σύγκριση χρόνων σταδίου 1 ($\text{nadir_cutoff} = 0.1$)

Στάδιο 1 (χρήση 10% των δεδομένων)	Στάδιο 1 (χρήση 20% των δεδομένων)
31s 88ms	1m 25s 932ms

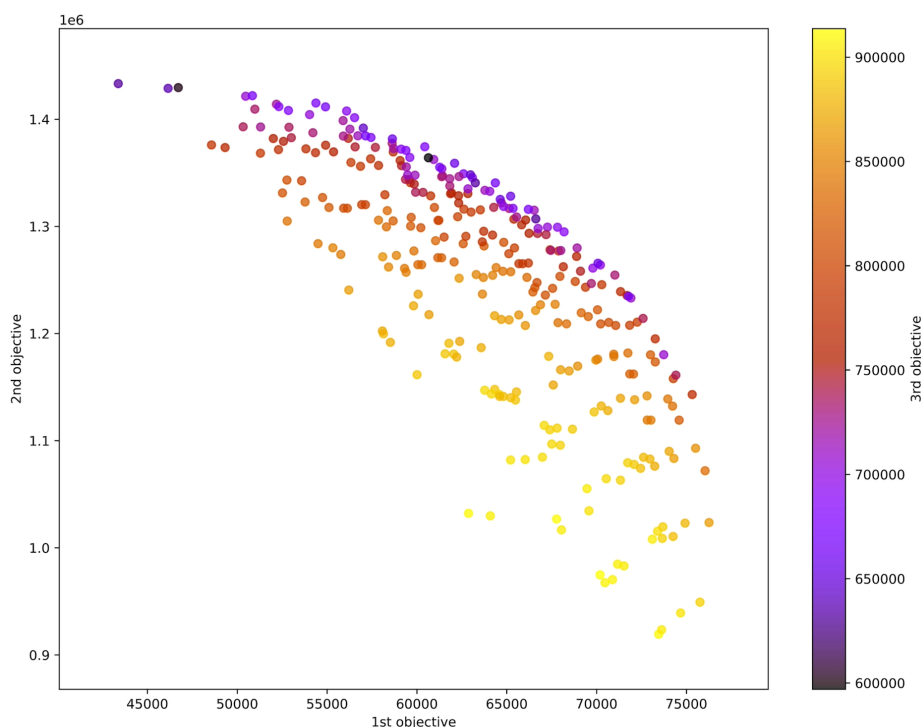
Παρατηρείται από τον Πίνακα 4.10 μία αύξηση ενός περίπου λεπτού όταν αναλύθηκε 20% αντί για 10% των δεδομένων.

Στα παρακάτω γραφήματα (Γράφημα 4.3, Γράφημα 4.4) αναπαρίστανται οι λύσεις του πρώτου σταδίου όταν το nadir_cutoff είναι 0.1 και όταν είναι 0.95 (μεγαλύτερο του $c_{\min}$).

Μέτωπο Pareto για πολύ μικρό nadir_cutoff



Γράφημα 4.3: Αναπαράσταση Pareto λύσεων δεύτερου πειράματος για nadir cutoff 0.1



Γράφημα 4.4: Αναπαράσταση Pareto λύσεων δεύτερου πειράματος για nadir cutoff 0.95

Παρατηρούμε ότι στο Γράφημα 4.4 έχουμε 311 λύσεις αντί για 312 (απουσία της χαμηλότερης ως προς το άξονα της 2ης αντικειμενικής από το Γράφημα 4.3). Επομένως και με το 10% των δεδομένων η η ακρίβεια στην εύρεση του $c_{\min}$ είναι ικανοποιητική.

4.2 Εφαρμογή της Μεθοδολογίας στο Ενεργειακό Πρόβλημα με Αβεβαιότητα ως προς τη Μορφή των Δεδομένων

Τα πειράματα που διεκπεραιώθηκαν στις προηγούμενες δύο ενότητες άντλησαν δεδομένα από τον πραγματικό κόσμο και είχαν κάποια σχέση μεταξύ τους όπως ήταν καταγεγραμμένα (γραμμική). Επομένως η επιλογή ενός μέρους εξ αυτών μπορούσε να δημιουργήσει μία πολύ καλή εικόνα και για τα υπόλοιπα δεδομένα. Σε αυτή την ενότητα, προχωράμε σε ένα πείραμα με τυχαιοποιημένα δεδομένα. Τα δεδομένα προηγουμένως είχαν επίσης συγκεκριμένη μορφή με τη μία αντικειμενική (F1) να έχει σημαντικά μικρότερο εύρος από τις άλλες 2 επομένως όταν έμπαινε στο εσωτερικό loop πετυχαίναμε πολύ καλούς χρόνους. Ωστόσο τώρα τα δεδομένα δεν αντιπροσωπεύουν κάποιο πρόβλημα του πραγματικού κόσμου αλλά δημιουργήθηκαν με εύρη παρόμοια μεταξύ τους και με τυχαίες τιμές με εύρη όμως κοντά σε πιθανές

πραγματικές τιμές. Επομένως σε αυτή της περίπτωση δεν υπάρχει η γραμμικότητα μεταξύ των levels στα δεδομένα. Τα δεδομένα με αυτήν την μορφή θα αναδεικνύουν την ακραία περίπτωση όπου η επιλογή μέρους των δεδομένων μπορεί να μην αποτυπώνει με τόσο μεγάλη ακρίβεια την τιμή του $c_{\min}$. Εάν τύχει στο πρώτο στάδιο της 2-Stage AUGMECON-Py όλες οι τιμές των δεδομένων που επιλέχθηκαν να είναι σχετικά μεγαλύτερες από αυτές που δεν επιλέχθηκαν τότε ο συντελεστής $c_{\min}$ να μην είναι τόσο ακριβής όσο στο προηγούμενο πείραμα. Επομένως, μία αρκετά λανθασμένη εκτίμηση του $c_{\min}$ μπορεί να οδηγήσει σε σημαντικές απώλειες βέλτιστων κατά Pareto λύσεων.

Εκτελούμε τον αλγόριθμο AUGMECON-Py και παρατηρούμε τα ακόλουθα:

Πίνακας 4.11: Payoff Table Πειράματος με Τυχαίοποιημένα Δεδομένα

F1	F2	F3
666336	1012620	684568
1018024	527746	737099
642415	544439	1035217

Παρατηρούμε ότι:

$$\text{range}(F1) = 1,018,024 - 642,415 = 375,609$$

$$\text{range}(F2) = 1,012,620 - 527,746 = 484,874$$

$$\text{range}(F3) = 1,035,217 - 684,568 = 350,649$$

Επομένως $\text{range}(F2) > \text{range}(F1) > \text{range}(F3)$ άρα η συνάρτηση F2 θα είναι η αντικειμενική συνάρτηση και οι συναρτήσεις F1, F3 θα μουν στη θέση των περιορισμών. Αρχικά εκτελείται το πρόγραμμα με το σύνολο των δεδομένων με μικρό $\text{nadir_cutoff}=0.1$ για να βρεθούν όλες οι λύσεις του προβλήματος. Εντέλει βρέθηκαν 141 λύσεις και οι ελάχιστες τιμές για κάθε αντικειμενική συνάρτηση φαίνονται στον παρακάτω πίνακα.

Πίνακας 3.12: Ελάχιστες τιμές αντικειμενικής από Pareto Λύσεις

Min F ₁	Min F ₂	Min F ₃
536098	318014	601459

Σημείωση: Έγινε πείραμα με πολύ χαμηλό $\text{nadir_cutoff}=0.1$

Πίνακας 4.13: Συντελεστές c_i

c_1 (Αντικειμενική F_1)	c_2 (Αντικειμενική F_2)	c_3 (Αντικειμενική F_3)
0,834504175649697	0,60258912431359	0,878596428696638

Επομένως: $c_{min} = \min\{c_1, c_3\} \approx 0.8345$ είναι ο ιδανικός c_{min} που θέλουμε να προσεγγίσουμε

Όπως και στις προηγούμενες εφαρμογές απώλεια λύσεων αν αυξήσουμε το nadir cutoff.

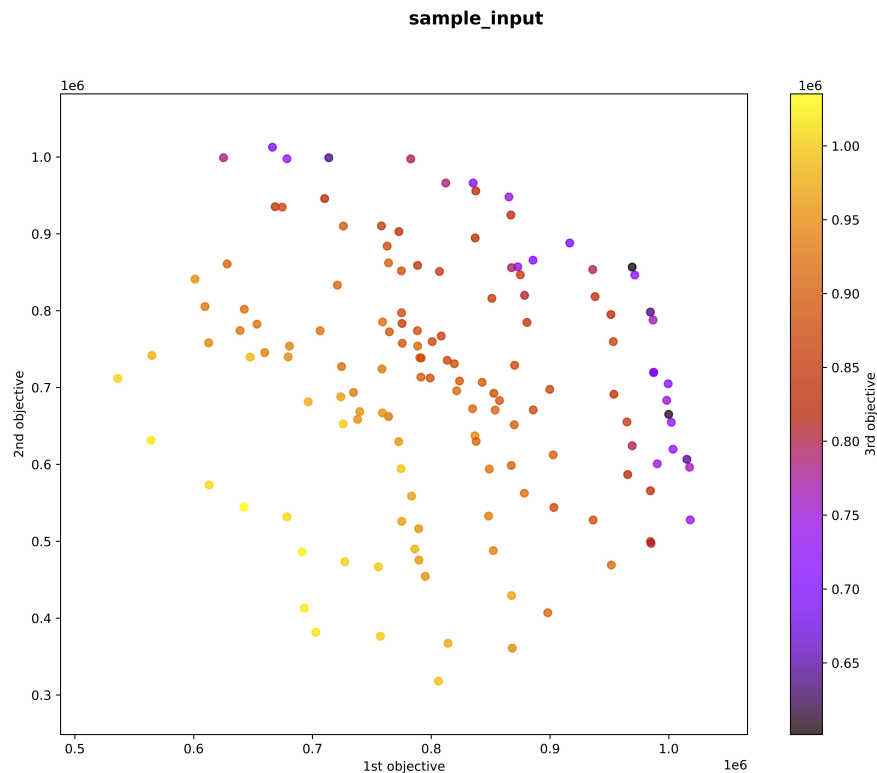
Πίνακας 4.14: Αποτελέσματα με nadir cutoff 0.1

Εξωτερική (max)	Εξωτερικό loop (constraint)	Εσωτερικό loop (constraint)	Total time (AUGMECON- D)	Number of Pareto Solutions
F2	F3	F1	4m 48s 308ms	141

Πίνακας 4.15: Αποτελέσματα με nadir cutoff 0.84

Εξωτερική (max)	Εξωτερικό loop (constraint)	Εσωτερικό loop (constraint)	Total time (AUGMECON- D)	Number of Pareto Solutions
F2	F3	F1	2m 17s 891ms	140

Παρατηρούμε την τυχαιότητα των δεδομένων και το ακανόνιστο μέτωπο Pareto στο Γράφημα 4.5 σε σύγκριση με τα γραφήματα των άλλων πειραμάτων.



Γράφημα 4.5: Αναπαράσταση Pareto λύσεων πειράματος τυχαιοποιημένων δεδομένων για πολύ μικρό nadir_cutoff = 0.1

Κάνοντας την ίδια διαδικασία με το πείραμα στην ενότητα 4.1 διαλέγοντας το 50% των δεδομένων, παρατηρείται πως η σειρά των αντικειμενικών αλλάζει καθώς αλλάζουν τα εύρη των αντικειμενικών συναρτήσεων.

Πίνακας 4.16: Payoff Table από το 50% των δεδομένων

F ₁	F ₂	F ₃
601159	841078	950857
668070	953381	590055
875584	778175	820051

Παρατηρούμε ότι:

$$\text{range}(F_1) = 875,584 - 601,159 = 274,425$$

$$\text{range}(F_2) = 953,381 - 778,175 = 175,206$$

$$\text{range}(F_3) = 950,857 - 590,055 = 360,802$$

Πίνακας 4.17: Ελάχιστες τιμές αντικειμενικής από Pareto Λύσεις

Min F ₁	Min F ₂	Min F ₃
601159	714321	590055

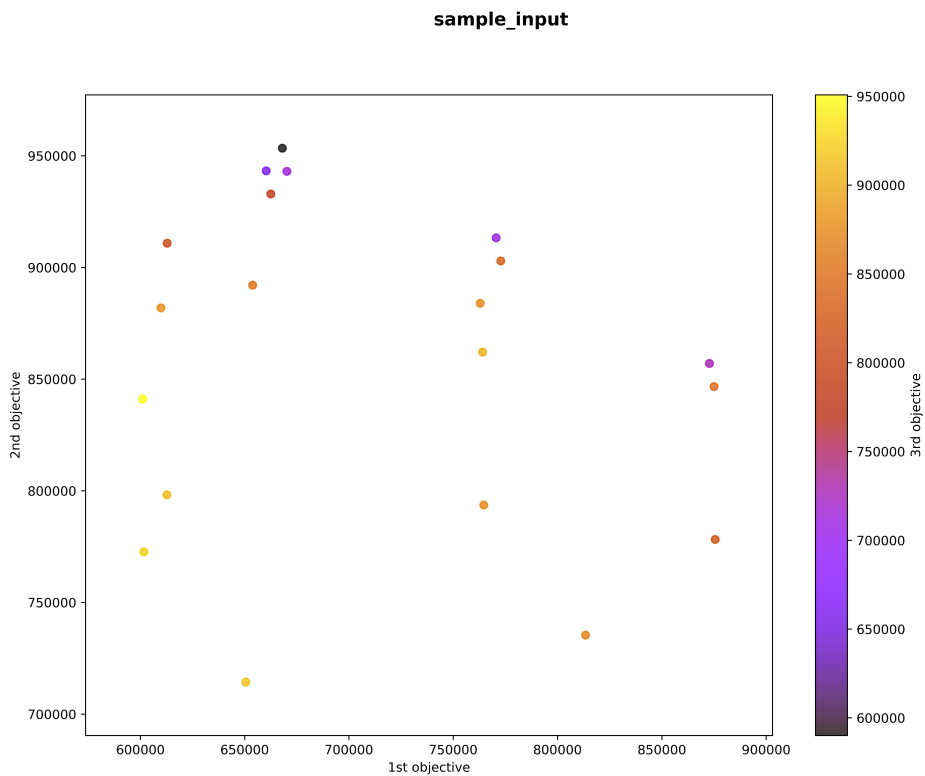
Επομένως:

$$c_1 = 601159/601159 = 1$$

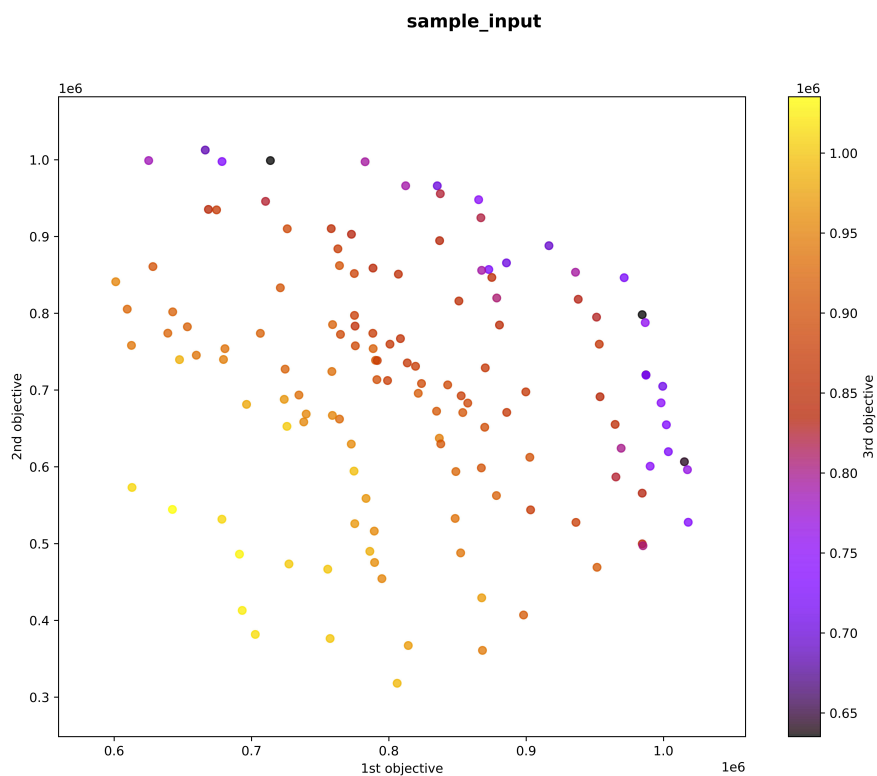
$$c_2 = 714321/778175 \approx 0.918$$

$$c_3 = 590055/590055 = 1$$

Παρατηρούμε ότι $c_{\min} = 0.918$ που είναι σημαντικά μεγαλύτερος από τον ιδανικό $c_{\min} = 0.8345$ που δεν θα έλειπε καμία λύση. Στη συνέχεια, γίνεται εφαρμογή της μεθόδου 2-Stage AUGMECON-Py όπου αναμένεται να μην βρεθούν ακριβώς όλες οι λύσεις του μετώπου Pareto λόγω της τυχαίας μορφής των δεδομένων. Με $\text{nadir_cutoff} = 0.9$ βρίσκονται οι 136 Pareto λύσεις εκ των 141. Στα παρακάτω γραφήματα απεικονίζονται αυτές οι λύσεις.



Γράφημα 4.6: Οι λύσεις που έδωσαν τα κάτω όρια για τον υπολογισμό του $c_{\min}$ στο πρώτο στάδιο με το 50% των δεδομένων



Γράφημα 4.7: Οι 136 κατά Pareto βέλτιστες λύσεις μετά το δεύτερο στάδιο με $nadir_cutoff=0.9$

Στη συνέχεια καταγράφεται ο χρόνος των παραπάνω δύο προσεγγίσεων στον παρακάτω πίνακα (Πίνακας 4.18):

Πίνακας 4.18: Σύγκριση 2-Stage AUGMECON-Py

Χρόνος σταδίου 1 (50% δεδομένων) nadir_cutoff = 0.1	Χρόνος σταδίου 2 nadir_cutoff = 0.9	Συνολικός Χρόνος 2- Stage AUGMECON- Py	Συνολικός χρόνος AUGMECON-Py nadir_cutoff=0.1
2m 8s 990ms	2m 6s 149ms	4m 15s 139ms	4m 37s 206ms

Ωστόσο, αν πάλι γίνει ανάλυση Monte Carlo τότε οι χρόνοι θα είναι οι ακόλουθοι (Πίνακας 4.19):

Πίνακας 4.19: Σύγκριση 2-Stage AUGMECON-Py σε Monte Carlo Ανάλυση

Συνολικός Χρόνος 2-Stage AUGMECON-Py	Συνολικός χρόνος AUGMECON-Py nadir_cutoff=0.1
3h 32m 23s 904ms	7h 42m 0s 588ms

Αριθμός Monte Carlo επαναλήψεων: 100

Στον Πίνακα 4.19 παρατηρείται πολύ σημαντική διαφορά στον χρόνο (118% αύξηση). Ωστόσο υπάρχει απώλεια λύσεων. Επομένως το ιδανικό θα ήταν στην συγκεκριμένη περίπτωση να εκτελεστεί μία φορά το πρόγραμμα με το σύνολο των δεδομένων με χαμηλό nadir_cutoff (π.χ. 0.1) και αφότου βρεθεί με ακρίβεια ο συντελεστής c_{min} , να εκτελεστεί το πρόγραμμα με την Monte Carlo ανάλυση. Στη συνέχεια καταγράφεται ο χρόνος σε αυτή την περίπτωση (Πίνακας 4.20):

Πίνακας 4.20: Σύγκριση χρόνου εκτέλεσης 2-Stage AUGMECON-Py

Στάδιο 1: c_{min} με 100% των δεδομένων	Στάδιο 2: AUGMEC ON-Py (nadir_cuto ff = 0.83)	Συνολικός χρόνος σταδίων 1 και 2:	AUGMEC ON-Py (nadir_cut off = 0.7)	AUGMEC ON-Py (nadir_cut off = 0.5)	AUGMEC ON-Py (nadir_cuto ff = 0.3)	AUGMEC ON-Py (nadir_cuto ff = 0.1)
4m 48s 308ms	2m 17s 891ms	7m 6s 199ms	2m 48s 418ms	2m 59s 739ms	3m 2s 354ms	3m 24s 712ms

Ωστόσο, αν πάλι γίνει ανάλυση Monte Carlo τότε οι χρόνοι θα είναι οι ακόλουθοι (Πίνακας 4.21):

Πίνακας 4.21: Σύγκριση χρόνου εκτέλεσης 2-Stage AUGMECON-Py με Monte Carlo

2-Stage AUGMECON-Py με το 100% των δεδομένων στο Στάδιο 1:	AUGMECON-Py (nadir_cutoff = 0.7)	AUGMECON-Py (nadir_cutoff = 0.1)
3h 54m 37s 408ms	4h 40m 41s 800ms	7h 42m 0s 588ms

Αριθμός Monte Carlo επαναλήψεων: 100

Επομένως ακόμα και στην περίπτωση των σχετικά τυχαίων δεδομένων υπάρχει πολύ μεγάλο όφελος με την χρήση της 2-Stage AUGMECON-Py όταν γίνεται ανάλυση Monte Carlo.

Το κέρδος του χρόνου σε ποσοστό παρουσιάζεται στον ακόλουθο Πίνακα 4.22:

Πίνακας 4.22: Ποσοστά Αύξησης Χρόνου Εκτέλεσης 2-Stage AUGMECON-Py με Monte Carlo

AUGMECON-Py (nadir_cutoff = 0.7)	AUGMECON-Py (nadir_cutoff = 0.1)
19,6%	96,9%

Αριθμός Monte Carlo επαναλήψεων: 100

Συμπεραίνουμε ότι η χρήση της βελτιωμένης μεθόδου 2-Stage AUGMECON-Py παρουσιάζει σημαντικό χρονικό κέρδος στις περιπτώσεις όπου θα γίνει Monte Carlo ανάλυση δηλαδή στην πλειονότητα των πειραμάτων του πραγματικού κόσμου. Ακόμα και αν δεν είναι επιθυμητή μία τέτοια ανάλυση, η μέθοδος αυτή δίνει μεγάλη βεβαιότητα για την πληρότητα των λύσεων σε αρκετά ικανοποιητικό χρόνο.

Κεφάλαιο 5.

Συμπεράσματα και Προοπτικές

5.1 Συνολικά Συμπεράσματα

Η παρούσα διπλωματική εργασία παρουσιάζει την εύρεση μίας νέας μεθοδολογίας εκτίμησης των ελαχίστων τιμών των αντικειμενικών συναρτήσεων ενός πολυκριτηριακού προβλήματος για την αποδοτικότερη εκτέλεση του αλγορίθμου AUGMECON-Py. Καθώς η ακριβής εκτίμηση των σημείων ναδύρ παραμένει μία σύνθετη και χρονοβόρα υπόθεση, η εκτίμησή τους παρουσιάζεται ως η καλύτερη εναλλακτική. Επιπλέον, αναφέρθηκε ότι στην AUGMECON-Py, εκφράζονται όλες οι αντικειμενικές συναρτήσεις εκτός από μία ως περιορισμοί. Πειραματικά εξετάστηκε ποια από αυτές θα πρέπει να είναι η αντικειμενική συνάρτηση του προβλήματος και με ποια σειρά θα μπου οι υπόλοιπες στους περιορισμούς προκειμένου να έχουμε πληρότητα στις λύσεις και καλό χρόνο εκτέλεσης.

Με την λεξικογραφική βελτιστοποίηση για κάθε αντικειμενική συνάρτηση, κατασκευάζεται ένας πίνακας τιμών (payoff table) με μόνο βέλτιστες κατά Pareto λύσεις και αφαιρώντας την ελάχιστη από την μέγιστη τιμή κάθε αντικειμενικής συνάρτησης, υπολογίζεται ένα εύρος. Φάνηκε πειραματικά ότι ο καλύτερος χρόνος εκτέλεσης επιτυγχάνεται όταν ανατεθούν σύμφωνα με φθίνοντα εύρη οι αντικειμενικές συναρτήσεις του πολυκριτηριακού προβλήματος στην αντικειμενική συνάρτηση του προβλήματος της AUGMECON-Py και στη συνέχεια από τον εξωτερικότερο στον εσωτερικότερο βρόγχο επαναλήψεων ως περιορισμοί. Ο χρόνος που διασχίζεται ο χώρος των λύσεων κάθε αντικειμενικής συνάρτησης αλλάζει αναλόγως εάν εκείνη βρίσκεται σε εσωτερικότερο ή εξωτερικότερο βρόγχο του αλγορίθμου λόγω της ικανότητας της μεθόδου να γλιτώνει επαναλήψεις με τη βοήθεια των συντελεστών b διασχίζοντας το πλέγμα (grid) κάθε αντικειμενικής.

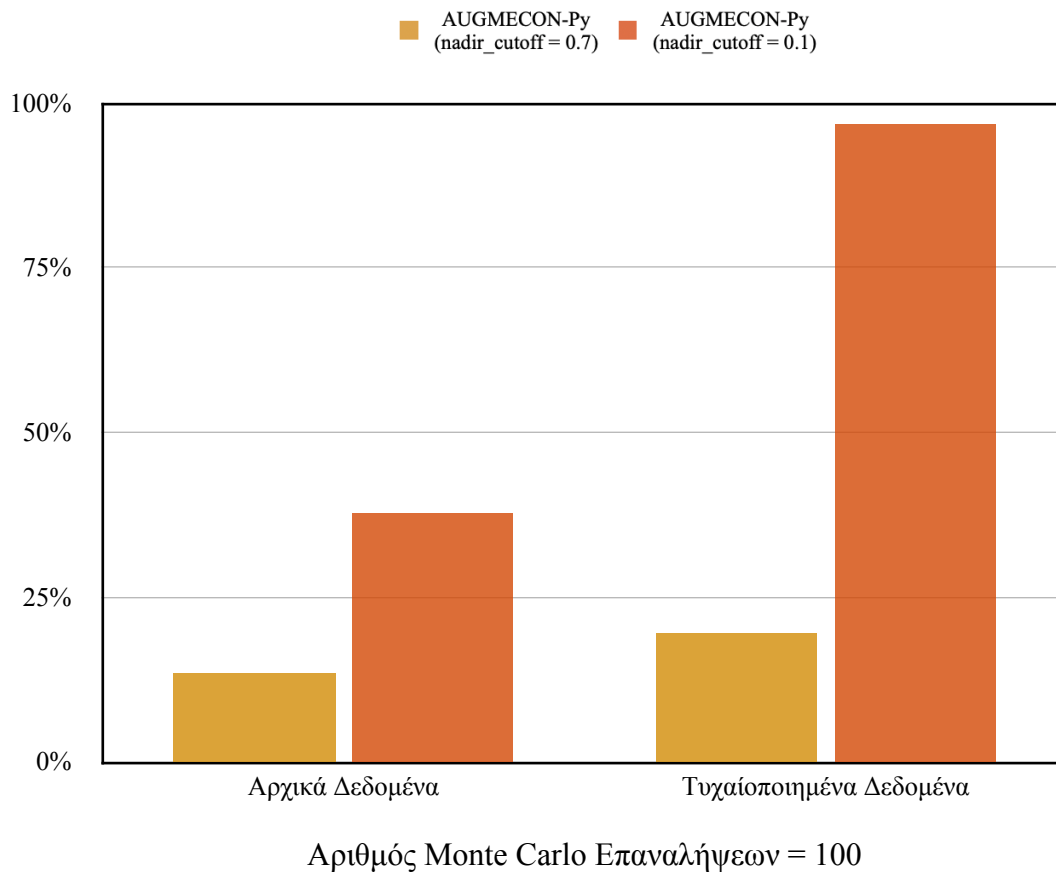
Το ζήτημα της πληρότητας των λύσεων μελετήθηκε σε πρώτη φάση μέσω πειραμάτων με διαφορετική σειρά στις αντικειμενικές συναρτήσεις. Έπειτα από τα πειράματα του αλγορίθμου της AUGMECON-Py που πραγματοποιήθηκαν προκύπτει ότι η θεωρητική ανάλυση επιβεβαιώνεται. Η σειρά των αντικειμενικών συναρτήσεων σε ένα πολυκριτηριακό πρόβλημα βελτιστοποίησης μπορεί να επηρεάσει το σύνολο των λύσεων. Όπως παρατηρήθηκε, αλλάζει δραστικά ο χρόνος που χρειάζεται ο αλγόριθμος για να συγκλίνει στις λύσεις. Ακόμα, αλλάζει το σύνολο λύσεων ανάλογα με τον προσδιορισμό των σημείων ναδύρ. Ο τρόπος που διασχίζεται ο χώρος των λύσεων αλλάζει καθώς δημιουργείται διαφορετικό πλέγμα (grid) σε κάθε περίπτωση. Είναι σημαντικό ότι για τον αλγόριθμο της AUGMECON-Py, οι λύσεις που χάνονται από την υπερεκτίμηση των σημείων ναδύρ είναι βέλτιστες κατά Pareto και όχι ασθενώς κυριαρχούμενες. Επειδή στον αλγόριθμο τελικά μία μόνο αντικειμενική συνάρτηση θα παραμείνει προς βελτιστοποίηση ενώ οι υπόλοιπες θα τροποποιηθούν κατάλληλα και θα θεωρηθούν ως επιπλέον περιορισμοί, λύσεις που βρίσκονται εκτός του πεδίου ορισμού των συναντήσεων-περιορισμών θα παραλειφθούν. Επομένως μπορεί και κάποιος σκόπιμα να ορίσει την σειρά με τρόπο κατάλληλο ώστε να μην χάνονται συγκεκριμένες λύσεις ακόμα και αν αυτή η ενέργεια έχει χρονικό αντίκτυπο. Τέτοιες επιλογές ενδείκνυται να γίνονται σε συνεννόηση με τον αποφασίζοντα ανάλογα τον σκοπό της βελτιστοποίησης.

Σε δεύτερη φάση, μελετήθηκε η μεταβολή του συντελεστή nadir-cutoff. Βρέθηκε ότι μετά από μία συγκεκριμένη τιμή του συντελεστή υπήρχε απώλεια λύσεων. Αυτή η τιμή μπορούσε να υπολογιστεί με ακρίβεια εάν γινόταν ανάλυση όλων των λύσεων των αντικειμενικών συναρτήσεων με πολύ μικρό nadir_cutoff και ασφαλώς με σημαντική χρονική επιβάρυνση. Η δυσκολία που παρουσιάστηκε ήταν η εκτίμηση ενός συντελεστή που θα είναι αρκετά μικρός ώστε να περιλαμβάνονται όλες οι λύσεις αλλά όχι τόσο ώστε να αυξάνεται σημαντικά ο χρόνος εκτέλεσης του προγράμματος.

Στην παρούσα εργασία βρέθηκε ένας τρόπος αποτελεσματικής εκτίμησης του συντελεστή nadir_cutoff έπειτα από την μελέτη μέρους των δεδομένων του προβλήματος και εφαρμόστηκε σε ενεργειακά προβλήματα εύρεσης βέλτιστων χαρτοφυλακίων χρηματοδότησης τεχνολογιών χαμηλών εκπομπών με πραγματικά δεδομένα. Η επέκταση της μεθόδου ονομάζεται εδώ 2-Stage AUGMECON-Py. Η επέκταση της μεθόδου AUGMECON-Py σε μία βελτιωμένη εκδοχή (2-Stage AUGMECON-Py) αποδείχθηκε αρκετά αποτελεσματική, ιδιαίτερα σε περιπτώσεις όπου είναι επιθυμητή η ανάλυση Monte Carlo των λύσεων ή που τα δεδομένα έχουν μία ορισμένη σχέση μεταξύ τους (π.χ. γραμμική). Τα δεδομένα συνήθως στα ενεργειακά προβλήματα διαθέτουν τέτοιου είδους σχέσεις και έτσι θα πραγματοποιούνται μειωμένοι χρόνοι εκτέλεσης της 2-Stage AUGMECON-Py με μεγάλη ακρίβεια ως προς την πληρότητα των λύσεων.

Οι νέες προγραμματιστικές συναρτήσεις που δημιουργήθηκαν για την εφαρμογή αυτής της ιδέας είναι απλές και εύκολα εφαρμόσιμες οπότε προτίθενται συμπληρωματικά στο σύνολο του κώδικα της AUGMECON-Py. Ένας επιδέξιος και έμπειρος προγραμματιστής μπορεί να επηρεάσει δραστικά την απόδοση μίας μεθόδου βελτιστοποίησης και ενός εργαλείου λήψεων αποφάσεων όπως η AUGMECON-Py με ακόμα περισσότερους τρόπους.

Έπειτα από την πειραματική μελέτη που πραγματοποιήθηκε σε ενεργειακό πρόβλημα με πραγματικά δεδομένα αλλά και με τυχαία δεδομένα αναδείχτηκε το χρονικό όφελος από τη χρήση της 2-Stage AUGMECON-Py. Στο παρακάτω γράφημα (Γράφημα 5.1) φαίνεται το χρονικό μειονέκτημα σε ποσοστό όταν χρησιμοποιείται η AUGMECON-Py έναντι της 2-Stage AUGMECON-Py σε πείραμα με ανάλυση Monte-Carlo 100 επαναλήψεων. Σε μία μέση περίπτωση όπου ο χρήστης θα χρησιμοποιούσε έναν δείκτη nadir_cutoff = 0.7 τότε θα χρειαζόταν 13,2% περισσότερο χρόνο με την AUGMECON-Py. Ωστόσο, αν επιθυμούσε την σιγουριά για την πληρότητα των λύσεων και έτρεχε το πρόγραμμα με χαμηλό nadir_cutoff = 0.1 τότε θα χρειαζόταν 37,6% περισσότερο χρόνο με την AUGMECON-Py. Στην αντίστοιχη περίπτωση με το πρόβλημα με τυχαία δεδομένα, εάν γινόταν η εκτέλεση του προγράμματος με έναν πιθανό συντελεστή nadir_cutoff = 0.7 θα χρειαζόταν 19,6% περισσότερος χρόνος με την AUGMECON - Py ενώ με nadir_cutoff = 0.1 για την βεβαιότητα στην πληρότητα λύσεων θα χρειαζόταν 96,9% περισσότερος χρόνος.



Γράφημα 5.1: Χρονικό Μειονέκτημα (%) με την Χρήση της AUGMECON-Py έναντι της 2-Stage AUGMECON-Py

Συμπεραίνοντας, υπάρχει πολύ σημαντικό όφελος από την 2-Stage AUGMECON-Py τόσο όσον αφορά τον χρόνο όσο και την σιγουριά για την πληρότητα του συνόλου των κατα Pareto βέλτιστων λύσεων.

5.2 Γενικά Συμπεράσματα για τη Βελτιστοποίηση

Πέρα από τα υπολογιστικά αποτελέσματα που βρέθηκαν και σχολιάστηκαν στο Κεφάλαιο 3 υπάρχουν ορισμένες ενδιαφέρουσες παρατηρήσεις που μπορούν να γίνουν. Γενικά η επιστήμη της βελτιστοποίησης χρειάζεται πέρα από καλή θεωρητική ανάλυση και καλή πρακτική γνώση. Εάν ο χρήστης που θα τρέξει για παράδειγμα το πρόγραμμα της AUGMECON-Py έχει μεγάλη εμπειρία σε προβλήματα που σχετίζονται με ανανεώσιμες πηγές ενέργειας τότε θα μπορεί να ορίσει πολύ καλύτερα την ανοχή σε λύσεις που αυτός επιθυμεί να έχει. Ακόμα και μία παρατήρηση μέρους των λύσεων όπως αυτές προέκυπταν με την αλλαγή των συντελεστών θα μπορεί να φανεί ιδιαίτερα χρήσιμη για τον αποφασίζοντα. Ορισμένες φορές η μορφή των δεδομένων είναι τέτοια όπου μπορεί κανείς να κρίνει για παράδειγμα με τι συντελεστή nadir cutoff θα τερματίσει το πρόγραμμα σε σχετικά γρήγορο χρόνο και δεν θα χαθεί σημαντική πληροφορία.

Η εμπειρία του παρατηρητή μπορεί να είναι κρίσιμη ακόμα και για τον τρόπο που θα συλλεχθούν τα δεδομένα και πως θα διαμορφωθούν έπειτα για να μπορούν να αναγνωριστούν σωστά και να χρησιμοποιηθούν αποδοτικά από ένα πρόγραμμα υπολογιστή. Ακόμα, μπορεί να γνωρίζει ποια κριτήρια είναι πιο σημαντικά ή για ποιες αντικειμενικές συναρτήσεις είναι σημαντικό να γνωρίζουμε το σύνολο των λύσεων που τις περιλαμβάνουν και επομένως να διαλέξει χειροκίνητα στο πρόγραμμα της AUGMECON-Py μία συγκεκριμένη σειρά των αντικειμενικών.

Εντέλει, η εμπειρία, η αντίληψη και η φαντασία στην συγκεκριμένη περίπτωση είναι εξίσου σημαντικές με την καλή θεωρητική γνώση. Θα μπορούσε κανείς να συνοψίσει τα παραπάνω με τις ακόλουθες φράσεις:

“Η βελτιστοποίηση δεν είναι μόνο μαθηματικά αλλά και τέχνη”

“Optimization is not only mathematics but also art”

“Οι καλύτεροι θεωρητικοί είναι οι καλύτεροι στην εφαρμογή της θεωρίας”

“Best theoreticians are best practitioners”

5.3 Προοπτικές Εργασίας

Η παρούσα εργασία ασχολήθηκε με την εφαρμογή της μεθόδου AUGMECON-Py και επεκτάσεις της για την μελέτη ενεργειακών πολιτικών σχετικά με τις ανανεώσιμες πηγές διαφόρων μεγεθών δεδομένων και σε πείραμα με τυχαία δεδομένα. Στο μέλλον είναι σημαντικό να γίνουν ακόμα περισσότερες εφαρμογές προκειμένου για να αποτελέσουν κίνητρο για επιπλέον βελτιώσεις. Μία από αυτές είναι η μελέτη ενός case study με που να συγκρίνει ενεργειακές πολιτικές διαφόρων κρατών λαμβάνοντας υπόψιν χαρακτηριστικά τους όπως ο πληθυσμός και το Α.Ε.Π. ανά κάτοικο σε σχέση με τον χρόνο. Λόγω της συνθετότητας του προβλήματος, η νέα μέθοδος 2-Stage AUGMECON-Py η οποία εισήχθη στην παρούσα εργασία ενδείκνυται να βρίσκει τις Pareto λύσεις του προβλήματος σε αρκετά καλό χρόνο.

Ορισμένες ακόμα ενδιαφέρουσες προεκτάσεις τις εργασίας είναι οι ακόλουθες.

- 1) Εύρεση του ιδανικού ποσοστού των δεδομένων που είναι αρκετό να αναλυθούν στο πρώτο στάδιο της νέας μεθόδου 2-Stage AUGMECON-Py προκειμένου να βρεθεί μία ικανοποιητική προσέγγιση του συντελεστή $c_{\min}$ που να παρέχει τον καλύτερο χρόνο εκτέλεσης και την μέγιστη δυνατή πληρότητα λύσεων στο στάδιο 2 της μεθόδου.
- 2) Εύρεση της κατάλληλης στρατηγικής για την αντιμετώπιση προβλημάτων που έχουν ακανόνιστα μέτωπα Pareto όπως για παράδειγμα η περίπτωση που μελετήθηκε με τα τυχαιοποιημένα δεδομένα στο ενεργειακό πρόβλημα (Κεφάλαιο 4.2).

- 3) Εφαρμογή της 2-Stage AUGMECON-Py που εισήχθη στην παρούσα εργασία σε ένα άλλο ήδη υπάρχον σύνθετο πρόβλημα πολυστοχικού προγραμματισμού με πραγματικά δεδομένα, καταγραφή του χρονικού οφέλους της νέας μεθόδου και σύγκριση του αριθμού των Pareto λύσεων.

Βιβλιογραφία

- Alves, M. J., & Costa, J. P. (2009). An exact method for computing the nadir values in multiple objective linear programming. *European Journal of Operational Research*, 198(2), 637–646. <https://doi.org/10.1016/j.ejor.2008.10.003>
- Bertsimas, D., & Brown, D. B. (2009). Constructing uncertainty sets for robust linear optimization. *Operations Research*, 57(6), 1483–1495. <https://doi.org/10.1287/opre.1080.0646>
- Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, 52(1), 35–53. <https://doi.org/10.1287/opre.1030.0065>
- Bertsimas, D., & Tsitsiklis, J. (1997). *Introduction to linear optimization*. <https://ci.nii.ac.jp/ncid/BA33943626>
- Deb, K., & Chaudhuri, S. Estimating Nadir Objective Vector Quickly Using Evolutionary Approaches.
- Dessouky, M., Ghiassi, M., & Davis, W. (1986). Estimates of the minimum nondominated criterion values in multiple-criteria decision-making. *Engineering Costs and Production Economics*, 10(1), 95–104. [https://doi.org/10.1016/0167-188x\(86\)90030-3](https://doi.org/10.1016/0167-188x(86)90030-3)
- Esther, B. P., & Kumar, K. S. (2016). A survey on residential Demand Side Management architecture, approaches, optimization models and methods. *Renewable & Sustainable Energy Reviews*, 59, 342–351. <https://doi.org/10.1016/j.rser.2015.12.282>
- Facility location. (2002). In *Springer eBooks*. <https://doi.org/10.1007/978-3-642-56082-8>
- Forouli, A., Doukas, H., Nikas, A., Sampedro, J., & Van De Ven, D. (2019). Identifying optimal technological portfolios for European power generation towards climate change mitigation: A robust portfolio analysis approach. *Utilities Policy*, 57, 33–42. <https://doi.org/10.1016/j.jup.2019.01.006>
- Forouli, A., Gkonis, N., Nikas, A., Siskos, E., Doukas, H., & Tourkolias, C. (2019). Energy efficiency promotion in Greece in light of risk: Evaluating policies as portfolio assets. *Energy*, 170, 818–831. <https://doi.org/10.1016/j.energy.2018.12.180>
- Forouli, A., Pagonis, A., Nikas, A., Koasidis, K., Xexakis, G., Koutsellis, T., Petkidis, C., & Doukas, H. (2022). AUGMECON-PY: A Python Framework for Multi-Objective Linear Optimisation under Uncertainty. *Social Science Research Network*. <https://doi.org/10.2139/ssrn.4133131>

- Fu, Z., Zhang, Y., He, S., Wang, H., Jiang, X., & Wang, S. (2022). Multi-objective programming for economy–energy–environment system and policy mix with dual constraints of carbon emission and water consumption based on
- Goodarzi, E., Ziaei, M., & Hosseini pour, E. Z. (2014). Introduction to Optimization Analysis in Hydrosystem Engineering. In *Topics in safety, risk, reliability and quality*. <https://doi.org/10.1007/978-3-319-04400-2>
- Hwang, C., Paidy, S., Yoon, K., & Masud, A. (1980). Mathematical programming with multiple objectives: A tutorial. *Computers & Operations Research*, 7(1–2), 5–31. [https://doi.org/10.1016/0305-0548\(80\)90011-8](https://doi.org/10.1016/0305-0548(80)90011-8)
- Jorge, J. M. (2009). An algorithm for optimizing a linear function over an integer efficient set. *European Journal of Operational Research*, 195(1), 98–103. <https://doi.org/10.1016/j.ejor.2008.02.005>
- Isermann, H., & Steuer, R. E. (1988). Computational experience concerning payoff tables and minimum criterion values over the efficient set. *European Journal of Operational Research*, 33(1), 91–97. [https://doi.org/10.1016/0377-2217\(88\)90257-3](https://doi.org/10.1016/0377-2217(88)90257-3)
- Jeff D Arena. 'Design Considerations for a Reconfigurable Delta Robot.' M.S. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, May 2014.
- Marler, R. T., & Arora, J. S. (2009). The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41(6), 853–862. <https://doi.org/10.1007/s00158-009-0460-7>
- Mavrotas, G. (2009). Effective implementation of the ϵ -constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, 213(2), 455–465. <https://doi.org/10.1016/j.amc.2009.03.037>
- Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ϵ -constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18), 9652–9669. <https://doi.org/10.1016/j.amc.2013.03.002>
- Messac, A. (2015). *Optimization in Practice with MATLAB®: For Engineering Students and Professionals*. <http://ci.nii.ac.jp/ncid/BB18672746>
- Metev B., Vassilev V., (2003). A method for nadir point estimation in MOLP problems, *Cyberne. Inf. Technol.* 3, 15–24.
- Miettinen, K. (1998). Nonlinear multiobjective optimization. In *International series in management science/operations research/International series in operations*

- Mishra, A. R., Rani, P., & Prajapati, R. S. (2021). Multi-criteria weighted aggregated sum product assessment method for sustainable biomass crop selection problem using single-valued neutrosophic sets. *Applied Soft Computing*, 113, 108038. <https://doi.org/10.1016/j.asoc.2021.108038>
- Multiobjective programming and planning. (1978). In *Mathematics in Science and Engineering/Mathematics in science and engineering*. [https://doi.org/10.1016/s0076-5392\(08\)x6053-9](https://doi.org/10.1016/s0076-5392(08)x6053-9)
- Nature-Inspired optimization algorithms. (2021b). In *Elsevier eBooks*. <https://doi.org/10.1016/c2019-0-03762-4>
- Nikas, A., Fountoulakis, A., Forouli, A., & Doukas, H. (2020). A robust augmented ε -constraint method (AUGMECON-R) for finding exact solutions of multi-objective linear programming problems. *Operational Research*, 22(2), 1291–1332. <https://doi.org/10.1007/s12351-020-00574-6>
- Nojavan, S., Majidi, M., Najafi-Ghalelou, A., Ghahramani, M., & Zare, K. (2017). A cost-emission model for fuel cell/PV/battery hybrid energy system in the presence of demand response program: ε -constraint method and fuzzy satisfying approach. *Energy Conversion and Management*, 138, 383–392. <https://doi.org/10.1016/j.enconman.2017.02.003>
- On a bicriterion formulation of the problems of integrated system identification and system optimization. (1971). *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3), 296–297. <https://doi.org/10.1109/tsmc.1971.4308298>
- Osiadacz, A. J. (1989). Multiple criteria optimization; theory, computation, and application, Ralph E. Steuer, Wiley Series in Probability and Mathematical Statistics - Applied, Wiley, 1986, No. of pages 546, Price f5 1.40, \$77.10. *Optimal Control Applications & Methods/Optimal Control Applications and Methods*, 10(1), 89–90. <https://doi.org/10.1002/oca.4660100109>
- Reeves, G. R., & Reid, R. C. (1988). Minimum values over the efficient set in multiple objective decision making. *European Journal of Operational Research*, 36(3), 334–338. [https://doi.org/10.1016/0377-2217\(88\)90125-7](https://doi.org/10.1016/0377-2217(88)90125-7)
- Rehman, S., & A, S. (2017). Multi-Criteria Wind Turbine Selection using Weighted Sum Approach. *International Journal of Advanced Computer Science and Applications/International Journal of Advanced Computer Science & Applications*, 8(6). <https://doi.org/10.14569/ijacsa.2017.080616>
- Shojaei, A. H., Ghadimi, A. A., Miveh, M. R., Mohammadi, F., & Jurado, F. (2020b). Multi-Objective Optimal Reactive Power Planning under Load Demand and

Wind Power Generation Uncertainties Using ϵ -Constraint Method. *Applied Sciences*, 10(8), 2859. <https://doi.org/10.3390/app10082859>

- Steuer, R. E. (1997). Non-Fully Resolved Questions about the Efficient/Nondominated Set. In *Springer eBooks* (pp. 585–589). https://doi.org/10.1007/978-3-642-60667-0_57
- Van Dao, D., Adeli, H., Ly, H., Le, L. M., Le, V. M., Le, T., & Pham, B. T. (2020). A Sensitivity and Robustness analysis of GPR and ANN for High-Performance Concrete Compressive Strength Prediction using a Monte Carlo simulation. *Sustainability*, 12(3), 830. <https://doi.org/10.3390/su12030830>
- Van De Ven, D., Nikas, A., Koasidis, K., Forouli, A., Casseti, G., Chiodi, A., Gargiulo, M., Giarola, S., Köberle, A. C., Koutsellis, T., Mittal, S., Perdana, S., Vielle, M., Xexakis, G., Doukas, H., & Gambhir, A. (2022). COVID-19 recovery packages can benefit climate targets and clean energy jobs, but scale of impacts and optimal investment portfolios differ among major economies. *One Earth*, 5(9), 1042–1054. <https://doi.org/10.1016/j.oneear.2022.08.008>
- Xing, X., Yan, Y., Zhang, H., Long, Y., Wang, Y., & Liang, Y. (2019). Optimal design of distributed energy systems for industrial parks under gas shortage based on augmented ϵ -constraint method. *Journal of Cleaner Production*, 218, 782–795. <https://doi.org/10.1016/j.jclepro.2019.02.052>
- Yang, Y., Zhang, L., Yuan, Y., Sun, J., Che, Z., Qiu, Z., Du, T., Na, H., & Che, N. S. (2023). Multi-objective optimization on energy consumption, CO2 emission and production cost for iron and steel industry. *Journal of Environmental Management*, 347, 119102. <https://doi.org/10.1016/j.jenvman.2023.119102>
- Yu, H., & Solvang, W. (2016). An Improved Multi-Objective Programming with Augmented ϵ -Constraint Method for Hazardous Waste Location-Routing Problems. *International Journal of Environmental Research and Public Health/International Journal of Environmental Research and Public Health*, 13(6), 548. <https://doi.org/10.3390/ijerph13060548>
- Zerrahn, A. (2017). Wind power and externalities. *Ecological Economics*, 141, 245–260. <https://doi.org/10.1016/j.ecolecon.2017.02.016>

Παράρτημα

Κώδικας για την Επιλογή Μέρους των Δεδομένων

```
import pandas as pd

def process_excel(input_file, n):
    # Load the Excel file
    xls = pd.ExcelFile(input_file)

    # Dictionary to store dataframes for each sheet
    sheet_dict = {}

    # Process each sheet
    for sheet_name in xls.sheet_names:
        df = pd.read_excel(input_file, sheet_name=sheet_name)

        # Select columns starting from the second column and keep every n-th column
        selected_columns = [df.columns[0]] + df.columns[1::n].tolist()
        df_new = df[selected_columns]

        # Store the processed dataframe in the dictionary
        sheet_dict[sheet_name] = df_new

    # Create a new file name
    new_file = input_file.replace('.xlsx', '_new.xlsx')

    # Save the processed dataframes to a new Excel file
    with pd.ExcelWriter(new_file, engine='openpyxl') as writer:
        for sheet_name, df_new in sheet_dict.items():
            df_new.to_excel(writer, sheet_name=sheet_name, index=False)

    print(f"Processed file saved as {new_file}")

# Example usage
input_file = '/Users/panagiotispapavassilopoulos/Desktop/PythonVenv/Augmecon-Py_improved/input/sample_input.xlsx' # Replace with your input file
n = 5 # Replace with the desired n-th column interval
process_excel(input_file, n)
```

Κώδικας για την Εύρεση της Ελάχιστης Τιμής Pareto λύσης για κάθε Αντικειμενική Συνάρτηση

```
import pandas as pd

def find_min_values(input_file):
    # Load the Excel file
    df = pd.read_excel(input_file)

    # Columns of interest
    columns_of_interest = ["first_objective", "second_objective", "third_objective"]

    # Initialize a list to store the minimum values
    min_values = []

    # Find the minimum values for the specified columns
    for col in columns_of_interest:
        if col in df.columns:
            min_values.append(df[col].min())
        else:
            print(f"Column '{col}' not found in the Excel file.")
            min_values.append(None) # Append None if the column is not found

    # Return the minimum values in the specified order
    return min_values

# Example usage
# Replace with your input file
input_file = '/Users/panagiotispapavassilopoulos/Desktop/PythonVenv/Augmecon-Py/output/out_sample_input_2024-05-28 20-21-00.961897.xlsx'

min_values = find_min_values(input_file)
print(f"Minimum values in the order of 'first_objective', 'second_objective', 'third_objective': {min_values}")
```