

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

Ανάπτυξη Εργαλειοθήκης Μεθόδων Πολυκριτήριας
Ανάλυσης σε γλώσσα Python

Νικόλαος Βαβουλιώτης



*Επιβλέπων Καθηγητής:
Νικόλαος Ματσατσίνης*

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

Νικόλαος Βαβουλιώτης

Ανάπτυξη Εργαλειοθήκης Μεθόδων Πολυκριτήριας
Ανάλυσης σε γλώσσα Python

Επιβλέπων Καθηγητής: Νικόλαος Ματσατσίνης

Εξεταστική Επιτροπή

Ν. Ματσατσίνης
(επιβλέπων)

Καθηγητής
Ευφυή Συστημάτων
Υποστήριξης
Αποφάσεων

Ε. Γρηγορούδης

Καθηγητής
Διαχείρισης
Διαδικασιών
Ποιότητας

Σ. Τσαφάρakis

Καθηγητής
Επιστημονικού
Μάρκετινγκ

**Αφιερωμένη σε όλους τους ανθρώπους που μέσα σε αυτό το ταξίδι βρίσκονταν
δίπλα μου, με στήριζαν και με βοηθούσαν μέρα με την μέρα να γίνομαι
καλύτερος άνθρωπος ...**

Περίληψη

Εμπνευσμένη από την κουλτούρα του ανοιχτού κώδικα, η παρούσα διπλωματική εργασία εστιάζει στην δημιουργία μίας βιβλιοθήκης που θα δίνει την δυνατότητα στον χρήστη να δομήσει και να επιλύσει ένα πρόβλημα πολυκριτήριας ανάλυσης μέσα σε λίγες εντολές υπό την μορφή κώδικα. Ο χρήστης θα μπορεί να εξάγει τα αποτελέσματα και να παράγει γραφήματα μέσα από την βιβλιοθήκη. Λόγω των διαφορών που υπάρχουν στις εισόδους και εξόδους των μεθόδων, η βιβλιοθήκη θα είναι χωρισμένη σε επιμέρους τμήματα βάση των απαιτήσεων στα δεδομένα εισόδου σε κάθε μέθοδο. Η διπλωματική εργασία επικεντρώνεται στη δημιουργία μιας εργαλειοθήκης για την οικογένεια των μεθόδων UTA, την ανάπτυξη των διαδικασιών για τη διαμόρφωση των απαιτούμενων εισόδων κάθε μεθόδου, λαμβάνοντας υπόψη τις πληροφορίες που θα μπορούν να δίνουν οι αποφασίζοντες καθώς και τις απαιτήσεις του προς επίλυση προβλήματος. Η ανάπτυξη της εργαλειοθήκης έχει πραγματοποιηθεί σε γλώσσα Python.

Abstract

Inspired by the open source culture this thesis focuses on the development of a library for multi criteria methods. The library will empower the user to construct and solve multi criteria problems in a few lines of code. The user will be able to extract the results to perform analysis and visualization. Due to differences in structure, the library will be grouped by methods with common data set forms. This thesis centers on the development of the tools for the UTA family methods such as the structure of the entrance for the various methods, considering the information needed and demands of the problem. The library is developed in Python programming language.

Πίνακας περιεχομένων

Περίληψη.....	iii
Abstract.....	iv
Κεφάλαιο 1: Εισαγωγή και Σκοπός της εργασίας.....	1
1.1 Γενικά.....	1
1.2 Σκοπός της εργασίας.....	1
1.3 Δομή της εργασίας.....	2
Κεφάλαιο 2: Πολυκριτήρια ανάλυση.....	3
2.1 Εισαγωγή στην Πολυκριτήρια Ανάλυση.....	3
2.2 Πολυστοχικός Προγραμματισμός.....	3
2.3 Θεωρία Πολυκριτήριας Χρησιμότητας.....	4
2.4 Θεωρία των σχέσεων υπεροχής.....	4
2.5 Μέθοδοι Ανάλυσης Παλινδρόμησης.....	5
2.6 Κατηγορίες προβλημάτων πολυκριτήριας ανάλυσης.....	6
2.7 Υφιστάμενη κατάσταση.....	6
Κεφάλαιο 3: Η οικογένεια μεθόδων UTA.....	7
3.1 Εισαγωγή.....	7
3.2 Η μέθοδος UTA.....	7
3.3 Η μέθοδος UTASTAR.....	11
3.4 Η μέθοδος UTADIS.....	17
3.5 Η μέθοδος UTA II.....	23
3.6 Η μέθοδος stochastic UTA.....	25
Κεφάλαιο 4: Η εργαλειοθήκη MultiCrit.....	28
4.1 Εισαγωγή.....	28
4.2 Εισαγωγή δεδομένων στο σύστημα, φίλτρα και αυτοματισμοί.....	28
4.3 Εφαρμογή της UTA.....	29
4.4 Εφαρμογή της UTASTAR.....	35
4.5 Εφαρμογή της UTADIS.....	39
4.6 Η συνάρτηση UTAIL.....	44
4.7 Η συνάρτηση stochastic_UTA.....	47
4.8 Γραφήματα.....	50
4.9 τ του Kendall.....	52
Κεφάλαιο 5: Συμπεράσματα και προοπτικές.....	53
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	54

Κεφάλαιο 1: Εισαγωγή και Σκοπός της εργασίας

1.1 Γενικά

Ο Homo Economicus, όντας δημιούργημα του σύγχρονου αστικού πολιτισμού, πρέπει να ανταποκριθεί σε ένα περιβάλλον όπου η λήψη των σωστών αποφάσεων αποτελούν το κλειδί για να θριαμβεύσει. Στο σύγχρονο αυτό περιβάλλον ο άνθρωπος καλείται να διαχειριστεί ένα χρηματικό ποσό, να επεξεργαστεί έναν μεγάλο όγκο πληροφοριών και να επιλέξει ανάμεσα σε πληθώρα επιλογών τα προϊόντα και τις υπηρεσίες που θα του ικανοποιήσουν τις ανάγκες και θα βελτιώσουν το βιωτικό του επίπεδο. Προκειμένου να λάβει οποιαδήποτε απόφαση αναλύει, αξιολογεί και συγκρίνει τις εναλλακτικές του επιλογές, ωστόσο η διαδικασία αυτή δεν είναι πάντοτε ορθολογική αλλά επηρεάζεται από τα συναισθήματα και τις συγκινήσεις του ατόμου, με αποτέλεσμα να αποκλίνει από τον πρότερο της στόχο.

Οι οικονομικοί οργανισμοί που αλληλεπιδρούν σε αυτό το σύγχρονο αστικό περιβάλλον, αποτελούνται από ανθρώπους οι οποίοι λαμβάνουν αποφάσεις για την διοίκηση και την διαχείριση τους. Αυτές οι αποφάσεις είναι ζωτικής σημασίας για την επιβίωση των οικονομικών οργανισμών και οφείλουν να έχουν ορθολογικό και μεθοδικό χαρακτήρα. Τον σκοπό αυτό επιτελεί ο κλάδος της επιχειρησιακής έρευνας με την ανάπτυξη μεθόδων και εργαλείων που στοχεύουν στην μαθηματική μοντελοποίηση των προβλημάτων που καλείται να αντιμετωπίσει ο εκάστοτε αποφασίζων.

Η επιχειρησιακή έρευνα είναι ο κλάδος των μαθηματικών που ασχολείται με την μαθηματική μοντελοποίηση προβλημάτων και την βελτιστοποίηση της λύσης. Κάποιοι από τους υποκλάδους της επιχειρησιακής έρευνας είναι ο Μαθηματικός Προγραμματισμός, η Πολυκριτήρια Ανάλυση, η Θεωρία Παιγνίων, η Θεωρία γράφων κ.ά. Οι μεθοδολογίες της εφαρμόζονται κατά κόρον στα αντικείμενα της διοίκησης, των οικονομικών και της μηχανικής εξασφαλίζοντας την καλύτερη αξιοποίηση πόρων και την καλύτερη λειτουργία των συστημάτων.

1.2 Σκοπός της εργασίας

Προκειμένου να δοθεί η δυνατότητα σε αναλυτές να διεκπεραιώσουν αναλύσεις πολυκριτήριων προβλημάτων χωρίς την χρήση κάποιου εξειδικευμένου λογισμικού, αλλά και σε προγραμματιστές λογισμικού να ενσωματώσουν μεθοδολογίες πολυκριτήριας ανάλυσης σε εφαρμογές τους, στη διπλωματική αυτή αναπτύχθηκε μιας εργαλειοθήκη μεθοδολογιών πολυκριτήριας ανάλυσης. Πιο συγκεκριμένα οι μεθοδολογίες αυτές ανήκουν την οικογένεια των μεθόδων UTA, οι οποίες είναι η UTA, η UTASTAR και η UTADIS. Με την χρήση της γλώσσας προγραμματισμού

Python δημιουργείται ένα εύχρηστο περιβάλλον για την εφαρμογή αυτών των μεθόδων.

1.3 Δομή της εργασίας

Η διάρθρωση της παρούσας εργασίας εκτελείται ως εξής, στο Κεφάλαιο 2 γίνεται μια αναφορά στην πολυκριτήρια ανάλυση, στις διαφορετικές κατηγορίες των προβλημάτων της, τον πολυκριτήριο γραμμικό προγραμματισμό και την υφιστάμενη κατάσταση. Στο Κεφάλαιο 3, αναλύονται εκτενώς οι πολυκριτήριες μέθοδοι της εργαλειοθήκης καθώς και το μαθηματικό τους μοντέλο. Ακόμα, στο επόμενο κεφάλαιο παρουσιάζονται οι εντολές της εργαλειοθήκης, η μορφή των δεδομένων που δέχεται ως είσοδο το σύστημα και η μορφή των αποτελεσμάτων. Τέλος, παρατίθενται τα συμπεράσματα της εργασίας και προτάσεις για μελλοντικές ιδέες και επεκτάσεις της εργαλειοθήκης .

Κεφάλαιο 2: Πολυκριτήρια ανάλυση

2.1 Εισαγωγή στην Πολυκριτήρια Ανάλυση

Η αδυναμία των υπαρχόντων μοντέλων να αντιμετωπίσουν τα πολυδιάστατα πραγματικά προβλήματα, με χρήση ενός μόνο κριτηρίου, οδήγησε στην ανάπτυξη της Πολυκριτηριακής Λήψης Αποφάσεων (Multi-Criteria Decision Making). Τα πολυκριτήρια προβλήματα, λόγω της παρουσίας πολλαπλών και αντικρουόμενων μεταξύ τους κριτηρίων αξιολόγησης των εναλλακτικών αποφάσεων, είναι προβλήματα με χαμηλό βαθμό δόμησης. Λόγω της αδυναμίας δημιουργίας ενός μεμονωμένου μοντέλου για την επίλυση κάθε είδους προβλήματος, έχουν αναπτυχθεί μέθοδοι επηρεασμένες από διάφορα θεωρητικά ρεύματα με κυρίαρχα τον Πολυκριτήριο ή πολυστοχικό μαθηματικό προγραμματισμό, την θεωρία πολυκριτήριας χρησιμότητας, την θεωρία των σχέσεων υπεροχής και των μεθόδων ανάλυσης παλινδρόμησης.

2.2 Πολυστοχικός Προγραμματισμός

Ο Πολυστοχικός Προγραμματισμός αποτελεί γενίκευση του μαθηματικού προγραμματισμού και χαρακτηρίζεται από την ύπαρξη πολλαπλών αντικειμενικών συναρτήσεων προς βελτιστοποίηση. Οι διάφορες αντικειμενικές συναρτήσεις σε ένα πολυκριτήριο σύστημα συνήθως έχουν ανταγωνιστική σχέση. Τα κριτήρια των αντικειμενικών συναρτήσεων αντικρούονται μεταξύ τους και ως εκ τούτου η τελική λύση δεν είναι βέλτιστη. Ο στόχος του πολυστοχικού προγραμματισμού όμως δεν είναι η εύρεση της βέλτιστης αλλά κάποιας ενδιάμεσης λύσης που ικανοποιεί τον αποφασίζοντα (κατά Pareto βέλτιστη) προσφέροντας μία ισορροπία ανάμεσα στα κριτήρια (Σίσκος, 2000).

Η εύρεση της λύσης επιτυγχάνεται με την επαναληπτική επίλυση του προβλήματος για κάθε αντικειμενική συνάρτηση προς βελτιστοποίηση. Υπάρχουν διάφορα θεωρητικά ρεύματα για την δόμηση των αντικειμενικών συναρτήσεων. Η γενική μορφή της αντικειμενικής συνάρτησης και των περιορισμών παρουσιάζονται στις σχέσεις 2.1. και 2.2. αντίστοιχα όπου k το πλήθος των συναρτήσεων και X η επιτρεπτή περιοχή των λύσεων.

,

$$\min/\max(f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})) \quad (2.1)$$

$$Y.π. \ x \in X = \{x \in R^k / Ax \leq b, \ x \geq 0\} \quad (2.2)$$

2.3 Θεωρία Πολυκριτήριας Χρησιμότητας

Η θεωρία πολυκριτήριας χρησιμότητας εστιάζει στην δημιουργία μίας συνάρτησης χρησιμότητας U προς βελτιστοποίηση. Αυτές οι μέθοδοι απαιτούν από τον αποφασίζοντα να βαθμολογήσει τις εναλλακτικές ως προς τα κριτήρια g_1, g_2, \dots, g_n καθώς και να ορίσει βάρη w_1, w_2, \dots, w_n τα οποία σταθμίζουν κάθε κριτήριο. Το γινόμενο των αξιών και των βαρών αθροίζεται προκειμένου να παραχθεί μία ολική εκτίμηση των εναλλακτικών από ένα συγκεκριμένο μοντέλο χρησιμότητας (Ματσατσίνης, 2010). Η θεωρία της πολυκριτήριας χρησιμότητας θεμελιώνεται πάνω σε δύο βασικές παραδοχές:

- Της αποδοχής ότι οι εναλλακτικές επιλογές είναι δυνατόν να συγκριθούν μεταξύ τους και δεν υπάρχει η περίπτωση δύο από αυτές να μην μπορούν να συγκριθούν μεταξύ τους.
- Της μεταβατικότητας των προτιμήσεων μεταξύ των εναλλακτικών επιλογών.

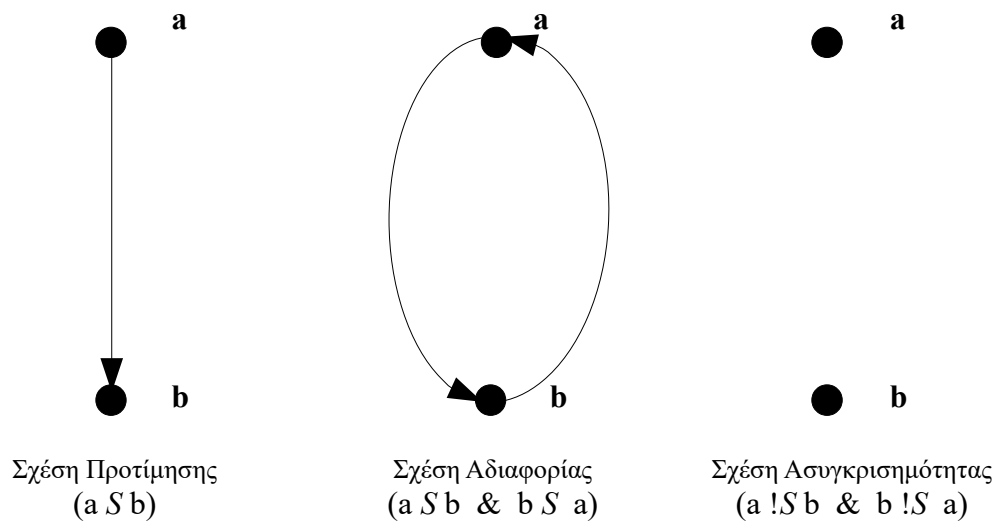
Αν και υπάρχουν διάφορες τεχνικές για τον υπολογισμό της συνάρτησης ολικής χρησιμότητας όπως η τεχνική του σταθερού σημείου και η τεχνική του σημείου μέσης αξίας, η γενική μορφή των μεθόδων ακολουθεί τα εξής βήματα:

1. Καθορισμός των εναλλακτικών και των σχετικών κριτηρίων.
2. Αξιολόγηση των εναλλακτικών ως προς κάθε κριτήριο.
3. Ορισμός σχετικών βαρών σε κάθε κριτήριο.
4. Άθροισμα των εξής γινομένων, βαρών-κριτηρίων και αξίας-κριτηρίων για τον υπολογισμό μίας ολικής χρησιμότητας για κάθε εναλλακτική.
5. Εκτέλεση ανάλυσης ευαισθησίας και πρόσδοση συστάσεων.

2.4 Θεωρία των σχέσεων υπεροχής

Σε αντίθεση με άλλες μεθόδους που δημιουργούν μία συνάρτηση χρησιμότητας οι μέθοδοι σχέσεων υπεροχής (outranking methods) χτίζουν σχέσεις προτιμήσεων μεταξύ των εναλλακτικών με την μορφή δυαδικών σχέσεων S στο σύνολο των εναλλακτικών X . Οι σχέσεις που αναπτύσσονται είναι προτίμησης (απλή υπεροχή), αδιαφορίας (διπλή υπεροχή) και ασυγκρισιμότητας (ανυπαρξία υπεροχής). Στην περίπτωση της προτίμησης $x S y$ η εναλλακτική x είναι τουλάχιστον εξίσου καλή με την εναλλακτική y , στην περίπτωση αδιαφορίας $x S y \& y S x$ η εναλλακτική x είναι τουλάχιστον εξίσου καλή με την y αλλά και η εναλλακτική y είναι τουλάχιστον

εξίσου καλή με την x και στην περίπτωση της ασυγκρισιμότητας $x \nmid S y$ & $y \nmid S x$ δεν υπάρχει σύγκριση μεταξύ των εναλλακτικών.



Σχήμα 2.1. Διμελείς σχέσης προτιμήσεων (πηγή: Ματσατσίνης, 2010)

Κυριότεροι εκφραστές αυτής της φιλοσοφίας αποτελούν οι μέθοδοι της οικογένειας ELECTRE(ELimination Et Choice Translating Reality, Roy, 1968), στην οποία υπάρχουν παραλλαγές που ανταποκρίνονται σε ποικίλους τύπους προβληματικών, καθώς και οι μέθοδοι PROMETHEE (Preference Ranking Organization METHod Enrichment Evaluations) των Brans και Vinckle (1985) .

2.5 Μέθοδοι Ανάλυσης Παλινδρόμησης

Η ανάλυση παλινδρόμησης είναι ευρέως χρησιμοποιημένη στατιστική τεχνική μοντελοποίησης που ερευνά τη συσχέτιση μεταξύ μίας εξαρτώμενης μεταβλητής και μιας ή περισσότερων ανεξάρτητων μεταβλητών. Στην πολυκριτήρια ανάλυση η τεχνική αυτή χρησιμοποιείται ως μία προσπάθεια προσέγγισης της συλλογιστικής του αποφασίζοντα. Συνθέτοντας και αναλύοντας τα κριτήρια και τις εναλλακτικές επιλογές που έχει ορίσει ο αποφασίζων, αναπτύσσεται ένα μοντέλο χρησιμότητας, το οποίο αναπαριστά βέλτιστα τις προτιμήσεις του (Greco *et al.*, 2010; Corrente *et al.*, 2013).

Σε αντίθεση με την λογική που διέπει τα περισσότερα μοντέλα πολυκριτήριας ανάλυσης, ότι η απόφαση καθορίζεται μέσα από την σύνθεση των κριτηρίων, εδώ

γίνεται δεκτή η λογική που πηγάζει μέσα από την κυβερνητική και σύμφωνα με την οποία η απόφαση και τα κριτήρια επιδέχονται προοδευτική επεξεργασία αλληλοδομούμενα μέσα στον χρόνο.

2.6 Κατηγορίες προβλημάτων πολυκριτήριας ανάλυσης

Τα προβλήματα πολυκριτήριας ανάλυσης χωρίζονται σε κατηγορίες προβλημάτων που ονομάζονται προβληματικές. Τα είδη των προβληματικών που μπορεί να τεθούν ορίζονται από τον Roy (1985) ως ακολούθως:

Προβληματική α: Στην προβληματική α κατατάσσονται τα προβλήματα που αφορούν την επιλογή μίας και μόνο εναλλακτικής από ένα σύνολο εναλλακτικών A.

Προβληματική β: Σε αυτή την κατηγορία κατατάσσονται τα προβλήματα ομαδοποίησης – ταξινόμησης των εναλλακτικών του συνόλου A σε συγκεκριμένες ομάδες με συγκεκριμένες ιδιότητες.

Προβληματική γ: Αφορά την κατάταξη όλων των εναλλακτικών ενεργειών του συνόλου A, από την πλέον προτιμώμενη προς την λιγότερο προτιμητέα.

Προβληματική δ: Σε αυτή την κατηγορία κατατάσσονται τα προβλήματα στα οποία γίνεται απλή περιγραφή όλων των εναλλακτικών των επιλογών του συνόλου A , καθώς και των συνεπειών τους βάσει των επιδόσεων τους στα επιμέρους κριτήρια.

2.7 Υφιστάμενη κατάσταση

Για ανάλυση δεδομένων με τεχνικές πολυκριτήριας ανάλυσης σε προγραμματιστικό περιβάλλον έχει αναπτυχθεί στην R το πακέτο MCDA (Meyer et al., 2021) . Το πακέτο αυτό περιέχει διάφορες μεθόδους πολυκριτήριας ανάλυσης όπως ELECTREIII, την AHP, καθώς και τις UTA, UTASTA και UTADIS. Δίνεται η δυνατότητα γραφικής αναπαράστασης των αποτελεσμάτων. Τα άτομα που συντηρούν το πακέτο είναι καθηγητές και ερευνητές του πανεπιστημίου IMT Atlantique της Γαλλίας με πολυετή εμπειρία στον χώρο της πολυκριτήριας ανάλυσης.

Στην γλώσσα προγραμματισμού Python αντίστοιχα έχει αναπτυχθεί η εργαλειοθήκη Scikit-Criteria (Cabral, Luczywo and Zanazzi, 2016) η οποία χρησιμοποιεί ένα σύστημα Διεπαφής Προγραμματισμού Εφαρμογών (API) για την εισαγωγή των δεδομένων και την επεξεργασία τους καθώς και γραφική αναπαράσταση των αποτελεσμάτων. Η εργαλειοθήκη δίνει την δυνατότητα εφαρμογής των μεθόδων ELECTRE, MOORA, TOPSIS, SIMUS και AHP. Για την επίλυση των γραμμικών προβλημάτων χρησιμοποιείται η εργαλειοθήκη PuLP της Python.

Κεφάλαιο 3: Η οικογένεια μεθόδων UTA

3.1 Εισαγωγή

Οι μέθοδοι της οικογένειας UTA είναι μία σειρά μοντέλων που βασίζονται στην μονότονη παλινδρόμηση για την ανάλυση των προτιμήσεων του αποφασίζοντα. Ανήκουν στην κατηγορία των μεθόδων ανάλυσης παλινδρόμησης και σκοπός τους είναι η δημιουργία αθροιστικών συναρτήσεων χρησιμότητας μέσα από τεχνικές γραμμικού προγραμματισμού. Οι συναρτήσεις αυτές έχουν στόχο να προσεγγίσουν την συλλογιστική των προτιμήσεων του αποφασίζοντα.

3.2 Η μέθοδος UTA

Η αρχική μέθοδος UTA (Jacquet-Lagrése and Siskos, 1982) έχει ως στόχο, δεδομένου ενός συνόλου πολυκριτηριακών εκτιμήσεων και μίας διάταξης των εναλλακτικών επιλογών, τον υπολογισμό των μερικών χρησιμοτήτων έτσι ώστε η υπολογιζόμενη προδιάταξη από το μοντέλο να είναι όσον το δυνατό πιο συνεπής με την αρχική προδιάταξη του αποφασίζοντα (Siskos, Grigoroudis and Matsatsinis, 2016).

Από ένα σετ εναλλακτικών A_R , οι οποίες έχουν μεταξύ τους είτε σχέση προτίμησης \mathbf{P} είτε σχέση αδιαφορίας \mathbf{I} , η μέθοδος προσπαθεί να δημιουργήσει μία ή περισσότερες συναρτήσεις χρησιμότητας που θα ανταποκρίνονται στις προτιμήσεις του αποφασίζοντα. Αξιοποιώντας ειδικές τεχνικές γραμμικού προγραμματισμού γίνεται εκτίμηση των συναρτήσεων ώστε να είναι συνεπής με την κατάταξη των προτιμήσεων. Η UTA χρησιμοποιεί μία μη σταθμισμένη συνάρτηση χρησιμότητας σε αντίθεση με παραλλαγές της (σχέση 1):

$$U'[g(a)] = \sum_{i=1}^n u_i[g_i(a)] + \sigma(a) \quad \text{για κάθε } a \in A \quad (1)$$

όπου $\sigma(a)$ είναι ένα πιθανό σφάλμα όσον αφορά την χρησιμότητα:

$$U'[g(a)] = \sum_{i=1}^n u_i[g_i(a)] \quad (2)$$

Με περιορισμούς κανονικοποίησης:

$$\begin{cases} \sum_{i=1}^n u_i(g_i^*) = 1 \\ u_i(g_i^*) = 0 \quad \text{Για κάθε } i \end{cases} \quad (3)$$

Τα δεδομένα που χρειάζεται ο αλγόριθμος για την υλοποίηση του είναι τα εξής:

- Τον πίνακα με τις τιμές των κριτηρίων για κάθε εναλλακτική.
- Την κατάταξη των εναλλακτικών R .
- Την μέγιστη και την ελάχιστη τιμή που μπορεί να λάβει ένα κριτήριο $[g_i^*, g_i^*]$
- Το πλήθος των διακριτών διαστημάτων που πρέπει να χωριστούν οι τιμές του εκάστοτε κριτηρίου.
- Την μονοτονία του κάθε κριτηρίου, δηλαδή αν ευνοεί θετικά η αύξηση της τιμής του κριτηρίου ή η μείωση του.
- Τις ελάχιστες διαφορές των εναλλακτικών δ και τις ελάχιστες διαφορές των διακριτών τιμών κάθε κριτηρίου s .

Βήμα 1

Αρχικά, δημιουργούνται τα διακριτά διαστήματα κάθε κριτηρίου ξεκινώντας από την χειρότερη τιμή g_* και τελειώνοντας στην καλύτερη g^* . Το πλήθος των τιμών καθορίζεται στην αρχή του αλγορίθμου.

Παραδείγματος χάρι, έστω ένα κριτήριο με αύξουσα μονοτονία με χειρότερη τιμή $g_i^*=1$ και καλύτερη τιμή $g_i^*=5$ και με διαχωρισμό τεσσάρων διαστημάτων. Οι διακριτές τιμές που θα λάβει το κριτήριο θα είναι $g=[1, 2, 3, 4, 5]$.

Βήμα 2

Στην συνέχεια ορίζονται οι μερικές χρησιμότητες $u_i(g_i^j)$ που θα υπολογιστούν στο γραμμικό πρόβλημα. Σε περίπτωση που μία τιμή του κριτηρίου δεν συμπίπτει ακριβώς με μία από τις τιμές του διαστήματος γίνεται γραμμική παρεμβολή και η μερική χρησιμότητα ορίζεται ως μέρος των διακριτών τιμών που την περιβάλουν.

Ουσιαστικά αν η μεταβλητή $g_i(\alpha)$ είναι ενδιάμεσα στο διάστημα $[g_i^j, g_i^{j+1}]$ η μερική χρησιμότητα $u_i[g_i(\alpha)]$ θα αντικατασταθεί με το άθροισμα των γινομένων $au(g_i^j)+bu(g_i^{j+1})$ όπου τα a και b προκύπτουν από την σχέση :

$$u_i[g_i(\alpha)] = u_i(g_i^j) + \frac{g_i(\alpha) - g_i^j}{g_i^{j+1} - g_i^j} [u_i(g_i^{j+1}) - u_i(g_i^j)] \quad (4)$$

Λύνοντας την σχέση βρίσκουμε:

$$a = \frac{g_i^{j+1} - g_i(\alpha)}{g_i^{j+1} - g_i^j}$$

$$b = \frac{g_i(\alpha) - g_i^j}{g_i^{j+1} - g_i^j}$$

π.χ.: όταν $[g_i^j, g_i^{j+1}] = [3, 4]$ και $g_i(\alpha) = 3,8$ τότε γίνεται αντικατάσταση $u_i[g_i(\alpha)] = 0,2 u(g_i^j) + 0,8 u(g_i^{j+1})$

Βήμα 3

Οι μερικές χρησιμότητες του κάθε κριτηρίου $u_1(\alpha), u_2(\alpha), u_3(\alpha), \dots$ της εναλλακτικής αθροίζονται και δημιουργείται η εκτιμώμενη χρησιμότητα της εναλλακτικής στην οποία στο τέλος, προστίθεται ένα σφάλμα $\sigma(\alpha)$ για την πιθανή απόκλιση που μπορεί να έχει με την πραγματικότητα.

$$U'[g(\alpha)] = \sum_{i=1}^n u_i[g_i(\alpha)] + \sigma(\alpha) \quad \text{για κάθε } \alpha \in A \quad (5)$$

Αφού εκτελεστεί αυτή η διαδικασία για κάθε εναλλακτική δημιουργούνται οι περιορισμοί του γραμμικού προβλήματος που αφορούν την εξασφάλιση της κυριαρχίας των εναλλακτικών καλύτερης προτίμησης στις εναλλακτικές χειρότερης προτίμησης. Ξεκινώντας από την καλύτερη εναλλακτική και πηγαίνοντας στην αμέσως λιγότερο καλή, δημιουργείται ένας περιορισμός της μορφής:

$$U'[g(a)] - U'[g(b)] \geq \delta \quad \text{όπου } a P b \quad (6)$$

Σε περίπτωση που δύο εναλλακτικές βρίσκονται στο ίδιο επίπεδο προτίμησης τότε ο περιορισμός έχει τη μορφή:

$$U'[g(a)] - U'[g(b)] = 0 \quad \text{όπου } a I b \quad (7)$$

Βήμα 4

Το επόμενο σετ περιορισμών που δημιουργείται εξασφαλίζει ότι σε κάθε κριτήριο οι μερικές χρησιμότητες αυξάνονται όσο βελτιώνεται η τιμή του κριτηρίου. Για κάθε ζεύγος τιμών του εκάστοτε κριτηρίου θα δημιουργηθεί ένας περιορισμός ο οποίος θα κατοχυρώνει ότι η μερική χρησιμότητα της καλύτερης τιμής θα είναι μεγαλύτερη κατά τουλάχιστον s σε σχέση με την αμέσως χειρότερη:

$$u_i(g_i^{j+1}) - u_i(g_i^j) \geq s_i \quad \text{με } i=1, 2, \dots, n \text{ και } j=1, 2, \dots, \alpha_i - 1 \quad (8)$$

Παραδείγματος χάριν ένα κριτήριο g_i με τιμές $[2, 4, 6, 8]$ οι περιορισμοί που θα δημιουργηθούν είναι:

1. $u_i(4) - u_i(2) \geq s_i$
2. $u_i(6) - u_i(4) \geq s_i$
3. $u_i(8) - u_i(6) \geq s_i$

Βήμα 5

Οι τελευταίοι περιορισμοί που δημιουργούνται αφορούν το εύρος τιμών των μερικών χρησιμοτήτων. Ο περιορισμός που θέτει το άνω όριο διατυπώνεται ως εξής:

$$\sum_{i=1}^n u_i(g_i^*) = 1 \quad (9)$$

και δηλώνει ότι το άθροισμα των καλύτερων τιμών του κάθε κριτηρίου πρέπει να αθροίζει στη μονάδα. Οι περιορισμοί που ικανοποιούν το κάτω όριο των τιμών κάθε κριτηρίου είναι:

$$u_i(g_{i^*}) = 0 \quad \text{για κάθε } i \quad (10)$$

και αφού εξασφαλιστεί ότι $u_i(g_i^j) \geq 0$ και $\sigma(\alpha) \geq 0$ για κάθε i, j και $\alpha \in R$ τότε έχουν ολοκληρωθεί οι περιορισμοί.

Βήμα 6

Η αντικειμενική συνάρτηση κάτω από την οποία θα επιλύεται το πρόβλημα, αποτελείται από το άθροισμα των σφαλμάτων $\sigma(\alpha)$ κάθε εναλλακτικής.

$$F = \sum_{\alpha \in R} \sigma(\alpha) \quad (11)$$

Έτσι γίνεται η επίλυση του εξής γραμμικού προβλήματος:

$$\min: F = \sum_{\alpha \in R} \sigma(\alpha)$$

υ.π.

$$\sum_{i=1}^n \{u_i[g_i(a)] - u_i[g_i(b)]\} + \sigma(\alpha) - \sigma(b) \geq \delta \quad \text{εάν } a P^* b$$

$$\sum_{i=1}^n \{u_i[g_i(a)] - u_i[g_i(b)]\} + \sigma(\alpha) - \sigma(b) = 0 \quad \text{εάν } a I^* b$$

$$u_i(g_i^{j+1}) - u_i(g_i^j) \geq s_i \quad \text{για κάθε } i \text{ και } j$$

$$\sum_{i=1}^n u_i(g_i^*) = 1$$

$$u_i(g_{i^*}) = 0, \quad u_i(g_i^j) \geq 0 \quad \text{και } \sigma(\alpha) \geq 0 \quad \text{για κάθε } i, j \text{ και } \alpha \in R$$

Βήμα 7

Αφού επιλυθεί το αρχικό πρόβλημα ξεκινάει η δεύτερη φάση του αλγορίθμου όπου διερευνάται μια εναλλακτική λύση γύρω από το βέλτιστο σημείο. Το γραμμικό πρόβλημα αναδομείται και η αντικειμενική συνάρτηση του αρχικού προβλήματος πλέον γίνεται περιορισμός των καινούργιων καθώς προστίθεται ένα κατώφλι. Το άθροισμα των σφαλμάτων πρέπει να είναι τουλάχιστον ίσο με την βέλτιστη λύση του αρχικού γραμμικού συν μία αποδεκτή προκαθορισμένη απώλεια ως ποσοστό της βέλτιστης λύσης:

$$\sum_{\alpha \in R} \sigma(\alpha) \leq F^* + k(F^*) \quad (12)$$

Σε αυτή την φάση οι χρησιμότητες υπολογίζονται από συναρτήσεις ελαχιστοποίησης και μεγιστοποίησης όπου μεταβλητές προς βελτιστοποίηση δεν αποτελούν πλέον τα σφάλματα αλλά οι μερικές χρησιμότητες των καλύτερων τιμών του κάθε κριτηρίου:

$$[min] \ u_i(g_i^*) \text{ και } [max] \ u_i(g_i^*) \text{ με } i=1,2,\dots,n \quad (13)$$

Οπότε τα γραμμικά προβλήματα που θα επιλυθούν κατά την μετά-βελτιστοποίηση θα είναι:

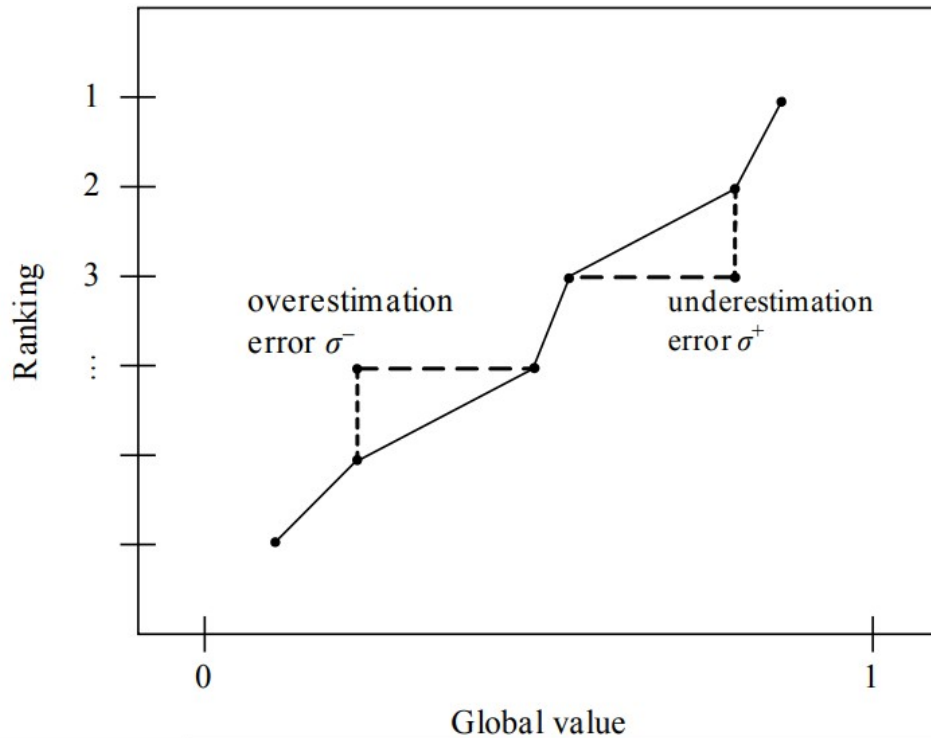
$$\begin{aligned} & [min] \ u_i(g_i^*) \text{ και } [max] \ u_i(g_i^*) \quad \text{με } i=1,2,\dots,n \\ & \text{υ.π.} \\ & \sum_{i=1}^n \{u_i[g_i(a)] - u_i[g_i(b)]\} + \sigma(\alpha) - \sigma(b) \geq \delta \quad \text{εάν } a P^* b \\ & \sum_{i=1}^n \{u_i[g_i(a)] - u_i[g_i(b)]\} + \sigma(\alpha) - \sigma(b) = 0 \quad \text{εάν } a I^* b \\ & u_i(g_i^{j+1}) - u_i(g_i^j) \geq s_i \quad \text{για κάθε } i \text{ και } j \\ & \sum_{i=1}^n u_i(g_i^*) = 1 \\ & \sum_{\alpha \in R} \sigma(\alpha) \leq F^* + k(F^*) \\ & u_i(g_i^*) = 0, \quad u_i(g_i^j) \geq 0 \text{ και } \sigma(\alpha) \geq 0 \text{ για κάθε } i, j \text{ και } \alpha \in R \end{aligned}$$

Τέλος, η λύση που θα προκύψει θα είναι η μέση τιμή των λύσεων όλων των γραμμικών προβλημάτων.

3.3 Η μέθοδος UTASTAR

Η μέθοδος UTASTAR(Siskos and Yannakopoulos 1985) αποτελεί μια βελτιωμένη έκδοση της μεθόδου UTA που παρουσιάστηκε προηγουμένως. Στο αρχικό μοντέλο που παρήγαγε η UTA, για κάθε δράση $a \in A_R$ δημιουργούταν ένα σφάλμα $\sigma(a)$ προς ελαχιστοποίηση. Καθώς πολλές φορές, η παρούσα συνάρτηση σφαλμάτων δεν είναι επαρκής για να ελαχιστοποιήσει πλήρως την διασπορά των σημείων γύρω από

την μονότονη καμπύλη προτείνεται μέσω της μεθόδου UTASTAR, η χρήση μίας διπλής συνάρτησης σφάλματος υποεκτίμησης $\sigma^+(a)$ και υπερεκτίμησης $\sigma^-(a)$ που επιτρέπει την καλύτερη σταθεροποίηση της θέσης των σημείων γύρω από την καμπύλη, όπως φαίνεται στο σχήμα 3.1(Siskos, Grigoroudis and Matsatsinis, 2016).



Σχήμα 3.1: Ordinal regression curve (ranking versus global value)(Siskos, Grigoroudis, Matsatsinis, Figueira, 2005)

Έτσι με την εισαγωγή των σφαλμάτων $\sigma^+(a)$ και $\sigma^-(a)$ η συνάρτηση χρησιμότητας παίρνει την μορφή:

$$u(g(a)) + \sigma^+(a) + \sigma^-(a) \quad (1)$$

Στην περίπτωση της UTASTAR οι μεταβλητές του γραμμικού προβλήματος δεν είναι οι αρχικές χρησιμότητες που εκφράζουν τα κριτήρια όπως ορίζονται στην UTA. Η μεταβλητή προς βελτιστοποίηση w πλέον προκύπτει ως η διαφορά των χρησιμοτήτων των διακριτών τιμών του κάθε κριτηρίου, όπως παρουσιάζεται στην σχέση 2.

$$w_{ij} = u_i(g_i^{j+1}) - u_i(g_i^j) \geq 0 \quad \forall i \text{ και } j \quad (2)$$

Το γραμμικό πρόβλημα σε αυτή την περίπτωση παίρνει την γενική μορφή:

$$\min: F = \sum_{a \in A} \{ \sigma^+(a) + \sigma^-(a) \}$$

υ.π.

$$\Delta(a, b) \geq \delta \text{ όταν } a > b$$

$$\Delta(a, b) = 0 \text{ όταν } a \sim b$$

$$\sum_i \sum_j w_{ij} = 1$$

$$w_{ij} \geq 0, \sigma^+(a) \geq 0, \sigma^-(a) \geq 0 \quad \forall a \in A, \forall i \text{ και } j$$

όπου: δ μία μικρή θετική τιμή

Προκειμένου να γίνει εκτέλεση της μεθόδου τα δεδομένα που θα πρέπει να εισαχθούν είναι:

- Ο πίνακας με τις τιμές των κριτηρίων για κάθε εναλλακτική.
- Η κατάταξη των εναλλακτικών R .
- Η μέγιστη και η ελάχιστη τιμή που μπορεί να λάβει ένα κριτήριο $[g_i^*, g_i^*]$
- Το πλήθος των διακριτών διαστημάτων που πρέπει να χωριστούν οι τιμές του εκάστοτε κριτηρίου.
- Η μονοτονία του κάθε κριτηρίου, δηλαδή αν ευνοεί θετικά η αύξηση της τιμής του κριτηρίου ή η μείωση του.
- Οι ελάχιστες διαφορές μεταξύ των εναλλακτικών δ .

Βήμα 1

Αντίστοιχα με την UTA αρχικά δημιουργούνται τα διακριτά διαστήματα κάθε κριτηρίου ξεκινώντας από την χειρότερη τιμή g^* και τελειώνοντας στην καλύτερη g^* . Το πλήθος των τιμών καθορίζεται στην αρχή του αλγορίθμου.

Παραδείγματος χάρη έστω ένα κριτήριο με αύξουσα μονοτονία με χειρότερη τιμή $g_i^*=1$ και καλύτερη τιμή $g_i^*=5$ και με διαχωρισμό τεσσάρων διαστημάτων. Οι διακριτές τιμές που θα λάβει το κριτήριο θα είναι $g=[1, 2, 3, 4, 5]$.

Βήμα 2

Στην συνέχεια ορίζονται οι μερικές χρησιμότητες $u_i(g_i^j)$. Σε περίπτωση που μία τιμή του κριτηρίου δεν συμπίπτει ακριβώς με μία από τις τιμές του διαστήματος γίνεται γραμμική παρεμβολή και η μερική χρησιμότητα ορίζεται ως μέρος των διακριτών τιμών που την περιβάλλουν.

Ουσιαστικά, αν η μεταβλητή $g_i(\alpha)$ είναι ενδιάμεσα στο διάστημα $[g_i^j, g_i^{j+1}]$ η μερική χρησιμότητα $u_i[g_i(\alpha)]$ θα αντικατασταθεί με το άθροισμα των γινομένων $au(g_i^j)+bu(g_i^{j+1})$ όπου τα a και b προκύπτουν από την σχέση :

$$u_i[g_i(\alpha)] = u_i(g_i^j) + \frac{g_i(\alpha) - g_i^j}{g_i^{j+1} - g_i^j} [u_i(g_i^{j+1}) - u_i(g_i^j)] \quad (3)$$

Λύνοντας την σχέση βρίσκουμε:

$$a = \frac{g_i^{j+1} - g_i(\alpha)}{g_i^{j+1} - g_i^j}$$

$$b = \frac{g_i(\alpha) - g_i^j}{g_i^{j+1} - g_i^j}$$

π.χ.: όταν $[g_i^j, g_i^{j+1}] = [3, 4]$ και $g_i(\alpha) = 3,8$ τότε γίνεται αντικατάσταση $u_i[g_i(\alpha)] = 0,2 u(g_i^j) + 0,8 u(g_i^{j+1})$

Βήμα 3

Στο σημείο αυτό γίνεται διαφοροποίηση της μεθόδου UTASTAR καθώς οι χρησιμότητες $u_i[g(a)]$ γίνονται συναρτήσεις των w_{ij} . Το κάθε w_{ij} αναπαριστά την διαφορά της χρησιμότητας $u_i(g_i^{j+1}) - u_i(g_i^j)$ έτσι η χρησιμότητα $u_i[g(a)]$ θα αντικατασταθεί από όλες τις χρησιμότητες w_{ij} που ορίζονται πριν την διακριτή τιμή g_i^j .

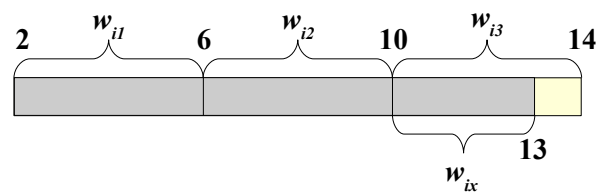
Έστω ένα διάστημα διακριτών τιμών $g_i = [2, 6, 10, 14]$ με τιμή του συγκεκριμένου κριτηρίου για αυτή την εναλλακτική $g(a) = 13$, η χρησιμότητα που ορίζεται ως $u_1(13)$ θα αντικατασταθεί από το άθροισμα $w_{i1} + w_{i2} + w_{ix}$ όπου τα w_i ορίζονται ως:

$$w_{i1} = u_i(6) - u_i(2)$$

$$w_{i2} = u_i(10) - u_i(6)$$

$$w_{ix} = u_i(13) - u_i(10) = 0,75(u_i(14) - u_i(10)) = 0,75 w_{i3}$$

Στην περίπτωση του παραδείγματος εφόσον το $g(a) = 13$ δεν θα λάβει ολόκληρη την χρησιμότητα w_{i3} , αλλά το ποσοστό που του αντιστοιχεί δηλαδή $\frac{13-10}{14-10} = 0,75$. Το ποσοστό αυτό προκύπτει μέσω γραμμικής παλινδρόμησης ανάμεσα στην πραγματική τιμή και στις διακριτές τιμές που την περιβάλλουν. Μία γραφική αναπαράσταση των w για αυτό το κριτήριο θα μπορούσε να είναι η εξής:



Αφού κάθε εναλλακτική μοντελοποιηθεί σε αθροιστικές χρησιμότητες w προστίθενται τα σφάλματα υπερεκτίμησης και υποεκτίμησης $\sigma^+(\alpha) - \sigma^-(\alpha)$.

Βήμα 4

Ξεκινώντας την δημιουργία του γραμμικού προγραμματισμού το πρώτο σετ περιορισμών που δημιουργείται αφορά την κατοχύρωση της προτίμησης κάθε εναλλακτικής υψηλότερης θέσης στην κατάταξη από της αμέσως χειρότερης. Αρχίζοντας από την καλύτερη εναλλακτική και πηγαίνοντας στην αμέσως λιγότερο καλή, δημιουργούνται οι περιορισμοί της μορφής:

$$\Delta(a, b) = u(g(a)) - u(g(b)) + \sigma^+(\alpha) - \sigma^-(\alpha) - \sigma^+(b) + \sigma^-(b) \geq \delta \quad (4)$$

$$\Delta(a, b) = u(g(a)) - u(g(b)) + \sigma^+(\alpha) - \sigma^-(\alpha) - \sigma^+(b) + \sigma^-(b) = 0 \quad (5)$$

Όπου $u(g(a))$ και $u(g(b))$ αθροίσματα w . Οι περιορισμοί αυτοί εξασφαλίζουν ότι το άθροισμα των χρησιμοτήτων της εναλλακτικής a θα έχει μεγαλύτερη τιμή από αυτό της εναλλακτικής b όταν $a P b$ (σχέση 4) και ότι το άθροισμα θα έχει την ίδια τιμή όταν $a I b$ (σχέση 5).

Παράδειγμα: Για ένα σετ εναλλακτικών με τρία κριτήρια $g_1=[4, 6, 8]$, $g_2=[10, 15, 20]$ και $g_3=[10, 7, 4, 1]$ και δύο εναλλακτικές με τιμές $a_1=(7, 16, 1)$ και $a_2=(6, 19, 4)$ όπου $a_1 P a_2$. Οι συναρτήσεις χρησιμότητας αυτών των μεταβλητών θα μοντελοποιηθούν ως εξής:

$$u(g(a_1)) = w_{11} + 0,5 w_{12} + w_{21} + 0,2 w_{22} + w_{31} + w_{32} + w_{33} + \sigma^+(\alpha_1) - \sigma^-(\alpha_1)$$

$$u(g(a_2)) = w_{11} + w_{21} + 0,8 w_{22} + w_{31} + w_{32} + \sigma^+(\alpha_2) - \sigma^-(\alpha_2)$$

Οπότε ο περιορισμός που θα σχηματιστεί ως διαφορά $\Delta(\alpha_1, \alpha_2)$ έχει την μορφή:

$$\Delta(\alpha_1, \alpha_2) = 0,5 W_{12} - 0,6 W_{22} + W_{33} + \sigma^+(\alpha_1) - \sigma^-(\alpha_1) - \sigma^+(\alpha_2) + \sigma^-(\alpha_2) \geq \delta$$

Βήμα 5

Ο επόμενος περιορισμός που δημιουργείται δεσμεύει ότι το άθροισμα όλων των w για κάθε κριτήριο ισούται με την μονάδα.

$$\sum_i \sum_j w_{ij} = 1 \quad (6)$$

Βήμα 6

Τέλος, σχηματίζεται η συνάρτηση χρησιμότητας προς ελαχιστοποίηση, η οποία αποτελείται από το άθροισμα των σφαλμάτων κάθε εναλλακτικής.

$$\min: F = \sum_{\alpha \in A} \{ \sigma^+(\alpha) + \sigma^-(\alpha) \} \quad (7)$$

Το γραμμικό πρόβλημα είναι έτοιμο προς επίλυση.

$$\min: F = \sum_{a \in A} \{ \sigma^+(a) + \sigma^-(a) \}$$

υ.π.

$$\Delta(a, b) \geq \delta \text{ όταν } a > b$$

$$\Delta(a, b) = 0 \text{ όταν } a \sim b$$

$$\sum_i \sum_j w_{ij} = 1$$

$$w_{ij} \geq 0, \sigma^+(a) \geq 0, \sigma^-(a) \geq 0 \quad \forall a \in A, \forall i \text{ και } j$$

Βήμα 7

Αφού επιλυθεί το αρχικό πρόβλημα ξεκινάει η δεύτερη φάση του αλγορίθμου όπου διερευνάται μια εναλλακτική λύση γύρω από το βέλτιστο σημείο. Το γραμμικό πρόβλημα αναδομείται και η αντικειμενική συνάρτηση του αρχικού προβλήματος πλέον γίνεται περιορισμός των καινούργιων καθώς προστίθεται ένα κατώφλι. Το άθροισμα των σφαλμάτων πρέπει να είναι τουλάχιστον ίσο με την βέλτιστη λύση του αρχικού γραμμικού συν μία αποδεκτή προκαθορισμένη απώλεια:

$$\sum_{a \in R} \{ \sigma^+(\alpha) + \sigma^-(\alpha) \} \leq F^* + \varepsilon \quad (8)$$

Η νέες αντικειμενικές συναρτήσεις κάτω από τις οποίες θα γίνει η μετά-βελτιστοποίηση σχηματίζονται από τις χρησιμότητες του κάθε κριτηρίου. Για κάθε κριτήριο δημιουργείται μία συνάρτηση προς μεγιστοποίηση.

$$[max] \quad \sum_j w_{ij} \quad \mu\epsilon \quad i=1,2,\dots,n \quad (9)$$

Οπότε τα γραμμικά προβλήματα που θα επιλυθούν κατά την μετά-βελτιστοποίηση θα είναι:

$$[max] \quad \sum_j w_{ij} \quad \mu\epsilon \quad i=1,2,\dots,n$$

υ.π.

$$\Delta(a, b) \geq \delta \text{ όταν } a > b$$

$$\Delta(a, b) = 0 \text{ όταν } a \sim b$$

$$\sum_i \sum_j w_{ij} = 1$$

$$\sum_{a \in R} \{ \sigma^+(\alpha) + \sigma^-(\alpha) \} \leq F^* + \varepsilon$$

$$w_{ij} \geq 0, \sigma^+(a) \geq 0, \sigma^-(a) \geq 0 \quad \forall a \in A, \forall i \text{ και } j$$

Τέλος, η λύση που θα προκύψει θα είναι η μέση τιμή των λύσεων όλων των γραμμικών προβλημάτων.

3.4 Η μέθοδος UTADIS

Η μέθοδος UTADIS (Devaud et al., 1980; Siskos and Jacquet-Lagrange, 1982; Jacquet-Lagrange, 1995; Doumpos and Zopounidis, 2002) ανήκει στην κατηγορία των μεθόδων ανάλυσης παλινδρόμησης βασισμένη στην λογική της UTA (Ματσατσίνης, 2010a) . Σε αντίθεση με τις δύο προηγούμενες μεθόδους σκοπός αυτής της μεθόδου είναι η δημιουργία μιας προσθετικής συνάρτησης χρησιμότητας η οποία θα ταξινομεί ένα σύνολο εναλλακτικών επιλογών σε ένα αριθμό προκαθορισμένων διαταγμένων ομοιογενών ομάδων (κλάσεων) με το ελάχιστο σφάλμα ταξινόμησης.

Σύμφωνα με τη μέθοδο αυτή, αρχικά χρησιμοποιείται ένα σύνολο αναφοράς αποτελούμενο από εναλλακτικές λύσεις οι οποίες αξιολογούνται από τον αποφασίζοντα και ταξινομούνται σε ομάδες ανάλογα με τις προτιμήσεις του, τις εμπειρίες του και το σκοπό της ανάλυσης. Στη συνέχεια με τη χρήση τεχνικών γραμμικού προγραμματισμού αναπτύσσεται μία συνάρτηση προσθετικής χρησιμότητας και γίνεται ο υπολογισμός των κατάλληλων ορίων χρησιμότητας έτσι ώστε να επιτευχθεί η ταξινόμηση των εναλλακτικών λύσεων στις ομάδες στις οποίες ανήκουν με το ελάχιστο σφάλμα ταξινόμησης.

Σε ένα σύνολο αναφοράς A το οποίο περιλαμβάνει n εναλλακτικές a_1, a_2, \dots, a_n οι οποίες αξιολογούνται βάση m κριτηρίων g_1, g_2, \dots, g_m . Σκοπός είναι η ταξινόμηση των εναλλακτικών λύσεων σε q διατεταγμένες ομάδες C_1, C_2, \dots, C_q έτσι ώστε η ομάδα C_k προτιμάται της ομάδας C_{k+1} . Η μορφή της προσθετικής συνάρτησης χρησιμότητας η οποία αναπτύσσεται μέσω της μεθόδου UTADIS είναι η ακόλουθη:

$$U(a) = \sum_{i=1}^m u_i[g_i(a)] \quad (1)$$

όπου $U(a)$ είναι η ολική χρησιμότητα μίας εναλλακτικής ενέργειας $a \in A$ και $u_i[g_i(a)]$ είναι η μερική χρησιμότητα της εναλλακτικής a στο κριτήριο αξιολόγησης g_i . Η ταξινόμηση των εναλλακτικών στις προκαθορισμένες ομάδες επιτυγχάνεται συγκρίνοντας την ολική χρησιμότητα με τα όρια χρησιμότητας u_1, u_2, \dots, u_{q-1} που διαχωρίζουν τις ομάδες ως εξής:

$$\begin{aligned} U(a) &\geq u_1 && \Leftrightarrow && a \in C_1 \\ u_1 &\geq U(a) &\geq u_2 && \Leftrightarrow && a \in C_2 \\ \dots &\dots &\dots &\dots &\dots &\dots \\ u_{k-1} &\geq U(a) &\geq u_k && \Leftrightarrow && a \in C_k \\ \dots &\dots &\dots &\dots &\dots &\dots \\ U(a) &\leq u_{q-1} && \Leftrightarrow && a \in C_q \end{aligned}$$

Το όριο u_k διαχωρίζει τις ομάδες C_k και C_{k-1}

Με την χρήση της μεθόδου UTADIS επιδιώκεται ο υπολογισμός κατάλληλων προσθετικών συναρτήσεων ολικής χρησιμότητας U οι οποίες θα προσεγγίζουν όσο το δυνατόν καλύτερα τις εκτιμήσεις του αποφασίζοντα. Σε περιπτώσεις αποκλίσεων κάποιων ταξινομήσεων για κάποια/ες εναλλακτική/ες a_j , τότε θα εμφανίζεται ένα σφάλμα σ_j . Το σφάλμα αυτό μπορεί να είναι:

- σ_j^+ : Σφάλμα υποεκτίμησης της εναλλακτικής a_j με την τοποθέτηση της σε χειρότερη ομάδα από αυτή που πραγματικά ανήκει.
- σ_j^- : Σφάλμα υπερεκτίμησης της εναλλακτικής a_j με την τοποθέτηση της σε καλύτερη ομάδα από αυτή που πραγματικά ανήκει.

Η εκτίμηση τόσο της προσθετικής συνάρτησης χρησιμότητας (μέσω της οποίας υπολογίζονται οι ολικές χρησιμότητες των εναλλακτικών) όσο και των ορίων χρησιμότητων επιτυγχάνονται μέσω της επίλυσης του ακόλουθου γραμμικού προγράμματος:

$$\begin{aligned}
 \min: \quad & F = \sum_{a \in C_1} \sigma^+(a) + \dots + \sum_{a \in C_j} [\sigma^+(a) + \sigma^-(a)] + \dots + \sum_{a \in C_q} \sigma^+(a) \\
 \text{υ.π.:} \quad & \sum_{i=1}^m u_i [g_i(a)] - u_1 + \sigma^+ \geq 0 \quad \forall a \in C_1 \\
 & \left. \begin{aligned} \sum_{i=1}^m u_i [g_i(a)] - u_{k-1} - \sigma^-(a) &\leq -\delta \\ \sum_{i=1}^m u_i [g_i(a)] - u_k + \sigma^+(a) &\geq 0 \end{aligned} \right\} \quad \forall a \in C_k (k=2,3,4,\dots,q-1) \\
 & \sum_{i=1}^m u_i [g_i(a)] - u_{q-1} - \sigma^-(a) \leq -\delta \quad \forall a \in C_q \\
 & \sum_{i=1}^m \sum_{l=1}^{\alpha_i-1} w_{il} = 1 \\
 & u_{k-1} - u_k \geq s \quad k=2,3,\dots,q-1 \\
 & w_{il} \geq 0, \quad \sigma^+(a) \geq 0, \quad \sigma^-(a) \geq 0 \quad \text{για } i=1,2,\dots,n; \quad l=1,2,\dots,\alpha_i-1
 \end{aligned}$$

Για την υλοποίηση της μεθόδου απαιτείται ο καθορισμός των παρακάτω δεδομένων:

- Τον πίνακα με τις τιμές των κριτηρίων για κάθε εναλλακτική.
- Την ομάδα C_i που ανήκει κάθε εναλλακτική.
- Τον καθορισμό των ομάδων και την σειρά προτίμησης.
- Την μέγιστη και την ελάχιστη τιμή που μπορεί να λάβει ένα κριτήριο $[g_i^*, g_i^-]$
- Το πλήθος των διακριτών διαστημάτων που πρέπει να χωριστούν οι τιμές του εκάστοτε κριτηρίου.

- Την μονοτονία του κάθε κριτηρίου, δηλαδή αν ευνοεί θετικά η αύξηση της τιμής του κριτηρίου ή η μείωση του.
- Τις ελάχιστες διαφορές των ομάδων δ και τις ελάχιστες διαφορές των διακριτών τιμών κάθε ομάδας s .

Βήμα 1

Αρχικά δημιουργούνται τα διακριτά διαστήματα κάθε κριτηρίου ξεκινώντας από την χειρότερη τιμή g^* και τελειώνοντας στην καλύτερη g^* . Το πλήθος των τιμών καθορίζεται στην αρχή του αλγορίθμου. Παραδείγματος χάρη έστω ένα κριτήριο με αύξουσα μονοτονία με χειρότερη τιμή $g_i^*=1$ και καλύτερη τιμή $g_i^*=5$ και με διαχωρισμό τεσσάρων διαστημάτων. Οι διακριτές τιμές που θα λάβει το κριτήριο θα είναι $g=[1, 2, 3, 4, 5]$.

Βήμα 2

Όπως και στις προηγούμενες μεθόδους ορίζεται για κάθε διακριτή τιμή του κάθε κριτηρίου οι μερικές χρησιμότητες $u_i(g_i^j)$. Σε περίπτωση που μία τιμή του κριτηρίου δεν συμπίπτει ακριβώς με μία από τις τιμές του διαστήματος γίνεται γραμμική παρεμβολή και η μερική χρησιμότητα ορίζεται ως μέρος των διακριτών τιμών που την περιβάλλουν.

Ουσιαστικά, αν η μεταβλητή $g_i(\alpha)$ είναι ενδιάμεσα στο διάστημα $[g_i^j, g_i^{j+1}]$ η μερική χρησιμότητα $u_i[g_i(\alpha)]$ θα αντικατασταθεί με το άθροισμα των γινομένων $au(g_i^j)+bu(g_i^{j+1})$ όπου τα a και b προκύπτουν από την σχέση :

$$u_i[g_i(\alpha)] = u_i(g_i^j) + \frac{g_i(\alpha) - g_i^j}{g_i^{j+1} - g_i^j} [u_i(g_i^{j+1}) - u_i(g_i^j)] \quad (3)$$

Λύνοντας την σχέση βρίσκουμε:

$$a = \frac{g_i^{j+1} - g_i(\alpha)}{g_i^{j+1} - g_i^j}$$

$$b = \frac{g_i(\alpha) - g_i^j}{g_i^{j+1} - g_i^j}$$

π.χ.: όταν $[g_i^j, g_i^{j+1}] = [3, 4]$ και $g_i(\alpha) = 3,8$ τότε γίνεται αντικατάσταση $u_i[g_i(\alpha)] = 0,2u(g_i^j) + 0,8u(g_i^{j+1})$

Βήμα 3

Στην συνέχεια γίνεται η μετατροπή των χρησιμοτήτων $u_i(g_i^j)$ σε χρησιμότητες w_{ij} αντίστοιχα με την UTASTAR. Το κάθε w_{ij} αναπαριστά την διαφορά της χρησιμότητας $u_i(g_i^{j+1}) - u_i(g_i^j)$ έτσι η χρησιμότητα $u_i[g(a)]$ θα αντικατασταθεί από όλες τις

χρησιμότητες w_{ij} που ορίζονται πριν την διακριτή τιμή g_i^j .

Έστω ένα διάστημα διακριτών τιμών $g_i=[2, 6, 10, 14]$ με τιμή του συγκεκριμένου κριτηρίου για αυτή την εναλλακτική $g(a)=13$, η χρησιμότητα που ορίζεται ως $u_1(13)$ θα αντικατασταθεί από το άθροισμα $w_{i1}+w_{i2}+w_{ix}$ όπου τα w_i ορίζονται ως:

$$w_{i1}=u_i(6)-u_i(2)$$

$$w_{i2}=u_i(10)-u_i(6)$$

$$w_{ix}=u_i(13)-u_i(10)=0,75(u_i(14)-u_i(10))=0,75w_{i3}$$

Στην περίπτωση του παραδείγματος εφόσον το $g(a)=13$ δεν θα λάβει ολόκληρη την χρησιμότητα w_{i3} , αλλά το ποσοστό που του αντιστοιχεί δηλαδή $\frac{13-10}{14-10}=0,75$. Το ποσοστό αυτό προκύπτει μέσω γραμμικής παλινδρόμησης ανάμεσα στην πραγματική τιμή και στις διακριτές τιμές που την περιβάλλουν.

Βήμα 4

Στην συνέχεια δημιουργούνται οι περιορισμοί του γραμμικού προβλήματος. Για κάθε εναλλακτική ανάλογα με την ομάδα που βρίσκονται οι περιορισμοί έχουν ως εξής:

$$\left. \begin{array}{l} \sum_{i=1}^m u_i[g_i(a)]-u_1+\sigma^+ \geq 0 \quad \forall a \in C_1 \\ \sum_{i=1}^m u_i[g_i(a)]-u_{k-1}-\sigma^-(a) \leq -\delta \\ \sum_{i=1}^m u_i[g_i(a)]-u_k+\sigma^+(a) \geq 0 \end{array} \right\} \quad \forall a \in C_k (k=2,3,4,\dots,q-1)$$

$$\sum_{i=1}^m u_i[g_i(a)]-u_{q-1}-\sigma^-(a) \leq -\delta \quad \forall a \in C_q$$

Όπου η ομάδα **1** είναι η καλύτερη ομάδα ενώ η ομάδα **q** είναι χειρότερη. Οι χρησιμότητες $u_1, \dots, u_k, \dots, u_{q-1}$ καθορίζουν τα όρια για την ταυτοποίηση των ομάδων.

Βήμα 5

Έπειτα δημιουργείτε περιορισμός που εξασφαλίζει ότι οι χρησιμότητες όλων των κριτηρίων αθροίζουν στην μονάδα.

$$\sum_{i=1}^m \sum_{l=1}^{\alpha_{i-1}} w_{il} = 1$$

Βήμα 6

Το επόμενο σεντ περιορισμών εξασφαλίζει την κυριαρχία των καλύτερων ομάδων στις χειρότερες ομάδες. Το όριο που καθορίζει σε ποια ομάδα ανήκει μία εναλλακτική πρέπει να απέχει τουλάχιστον μία ποσότητα s από το όριο της προηγούμενης ομάδας,

$$u_{k-1} - u_k \geq s \quad k=2,3,\dots,q-1$$

Βήμα 7

Σε αυτό το βήμα αναπτύσσεται η αντικειμενική συνάρτηση του προβλήματος η οποία αποτελείται από το άθροισμα των σφαλμάτων όλων των εναλλακτικών. Για κάθε εναλλακτική που ανήκει στην καλύτερη ομάδα υπάρχει ένα σφάλμα υποεκτίμησης σ^+ , για κάθε εναλλακτική που ανήκει στην χειρότερη ομάδα υπάρχει ένα σφάλμα υπερεκτίμησης σ^- , και για όλες τις υπόλοιπες εναλλακτικές που ανήκουν σε μία ενδιάμεση ομάδα όταν είναι διαθέσιμη υπάρχουν ένα σφάλμα σ^+ και ένα σ^- . Υπό τους περιορισμούς που δημιουργήθηκαν σε προηγούμενα βήματα λύνεται το εξής γραμμικό πρόβλημα:

$$\min: F = \sum_{\alpha \in C_1} \sigma^+(\alpha) + \dots + \sum_{\alpha \in C_j} [\sigma^+(\alpha) + \sigma^-(\alpha)] + \dots + \sum_{\alpha \in C_q} \sigma^+(\alpha)$$

υ.π.:

$$\left. \begin{aligned} \sum_{i=1}^m u_i [g_i(a)] - u_1 + \sigma^+ &\geq 0 \quad \forall \alpha \in C_1 \\ \sum_{i=1}^m u_i [g_i(a)] - u_{k-1} - \sigma^-(a) &\leq -\delta \\ \sum_{i=1}^m u_i [g_i(a)] - u_k + \sigma^+(a) &\geq 0 \end{aligned} \right\} \quad \forall \alpha \in C_k (k=2,3,4,\dots,q-1)$$
$$\sum_{i=1}^m u_i [g_i(a)] - u_{q-1} - \sigma^-(a) \leq -\delta \quad \forall \alpha \in C_q$$
$$\sum_{i=1}^m \sum_{l=1}^{\alpha_i-1} w_{il} = 1$$
$$u_{k-1} - u_k \geq s \quad k=2,3,\dots,q-1$$
$$w_{il} \geq 0, \sigma^+(\alpha) \geq 0, \sigma^-(\alpha) \geq 0 \quad \text{για } i=1,2,\dots,n; l=1,2,\dots,\alpha_i-1$$

Βήμα 8

Αφού λυθεί το αρχικό πρόβλημα η μέθοδος περνάει στην φάση της μετά-βελτιστοποίησης. Η αρχική αντικειμενική συνάρτηση γίνεται περιορισμός, ο οποίος δίνει την δυνατότητα να ερευνηθεί μία λύση η οποία θα προσεγγίζει τα αρχικά σφάλματα αλλά θα βελτιστοποιεί τις χρησιμότητες w_{ij} . Για κάθε κριτήριο δημιουργούνται δύο αντικειμενικές συναρτήσεις μία ελαχιστοποίησης και μία μεγιστοποίησης.

$$\min: \sum_{j=1}^{a_i-1} w_{ij} + \sum_{k=1}^{q-1} u_k \quad \text{για } \forall i$$

$$\max: \sum_{j=1}^{a_i-1} w_{ij} + \sum_{k=1}^{q-1} u_k \quad \text{για } \forall i$$

υ.π.:

$$\sum_{i=1}^m u_i [g_i(a)] - u_1 + \sigma^+ \geq 0 \quad \forall \alpha \in C_1$$

$$\left. \begin{aligned} \sum_{i=1}^m u_i [g_i(a)] - u_{k-1} - \sigma^-(a) &\leq -\delta \\ \sum_{i=1}^m u_i [g_i(a)] - u_k + \sigma^+(a) &\geq 0 \end{aligned} \right\} \quad \forall \alpha \in C_k (k=2,3,4,\dots,q-1)$$

$$\sum_{i=1}^m u_i [g_i(a)] - u_{q-1} - \sigma^-(a) \leq -\delta \quad \forall \alpha \in C_q$$

$$\sum_{i=1}^m \sum_{l=1}^{a_i-1} w_{il} = 1$$

$$\sum_{a \in C_1} \sigma^+(\alpha) + \dots + \sum_{a \in C_j} [\sigma^+(\alpha) + \sigma^-(\alpha)] + \dots + \sum_{a \in C_q} \sigma^+(\alpha) \leq F^* + \varepsilon$$

$$u_{k-1} - u_k \geq s \quad k=2,3,\dots,q-1$$

$$w_{il} \geq 0, \quad \sigma^+(\alpha) \geq 0, \quad \sigma^-(\alpha) \geq 0 \quad \text{για } i=1,2,\dots,n; \quad l=1,2,\dots,a_i-1$$

Αφού λυθεί το πρόβλημα και υπολογιστούν οι μερικές χρησιμότητες δίνεται η δυνατότητα να γίνει πρόβλεψη των ομάδων που ανήκουν σε μη ταυτοποιημένες εναλλακτικές. Προκειμένου να εξεταστεί η ακρίβεια των αποτελεσμάτων καλό θα ήταν να εξεταστούν τα σφάλματα που υπολόγισε η μέθοδος.

3.5 Η μέθοδος UTA II

Έχοντας ως δεδομένα το σύνολο των κριτηρίων απόφασης και τη διάταξη προτιμήσεων των εναλλακτικών από τον αποφασίζοντα η UTA II (Siskos 1980), προσπαθεί να αποδώσει τα βάρη των κριτηρίων που προσεγγίζουν καλύτερα του αποφασίζοντα σχηματίζοντας μία αθροιστική συνάρτηση με τη δομή που περιγράφεται παρακάτω:

Ορίζονται :

- A_i :το στοιχείο (ανάθεση) i στο διάνυσμα των προτιμήσεων των εναλλακτικών
- g_j :το στοιχείο (κριτήριο) j στο διάνυσμα των κριτηρίων
- $g_j(A_i)$:την επίδοση της εναλλακτικής A_i κριτήριο g_j
- g_j^* :το χειρότερο επίπεδο στο κριτήριο g_j
- g_j^* :το καλύτερο επίπεδο στο κριτήριο g_j
- $u_j(g_j)$:μία αύξουσα συνάρτηση χρησιμότητας που αντιστοιχεί τις τιμές του g_j στο $[0, 1]$
- p_j :τα ζητούμενα βάρη των κριτηρίων
- $U(g)$:η ολική συνάρτηση χρησιμότητας

Έτσι για ένα σύνολο χρησιμοτήτων θα έχουμε:

$$U(g) = \sum_{j=1}^n p_j u_j(g_j) \quad (1)$$

$$u_j(g_{j^*}) = 0 \quad (2)$$

$$u_j(g_j^*) = 1 \quad (3)$$

$$\sum_{j=1}^n p_j = 1 \quad (4)$$

Έστω $A_R = \{A_1, A_2, \dots, A_k\}$ το σύνολο των εναλλακτικών επιλογών που κατέταξε ο αποφασίζων σε σειρά προτίμησης. Η εναλλακτική A_1 είναι στην κορυφή των προτιμήσεων του αποφασίζοντα ενώ η A_k στο τέλος. Τα 4 βήματα του αλγορίθμου που ακολουθεί η UTA II για τον προσδιορισμό των βαρών είναι:

Βήμα 1

Έκφραση των συναρτήσεων ολικής χρησιμότητας των επιλεγμένων εναλλακτικών επιλογών σύμφωνα με την σχέση (1), με τους όρους δηλαδή των βαρών των κριτηρίων.

$$U[g(A_i)] = \sum_{j=1}^n p_j u_j[g_j(A_i)] \quad \forall i=1,2,\dots,k \quad (5)$$

Η συνάρτηση χρησιμότητας u σε ντετερμινιστικό περιβάλλον είναι μία αύξουσα συνάρτηση που αντιστοιχεί τις επιδόσεις των κριτηρίων αξιολόγησης στο διάστημα $[0, 1]$. Αν υπάρχει προτίμηση μεταξύ δύο εναλλακτικών σε κάποιο κριτήριο g_i τότε $u(g_i(A_j)) > u(g_i(A_{j+1}))$ ενώ για την περίπτωση αδιαφορίας μεταξύ δύο εναλλακτικών η συνάρτηση χρησιμότητας εκφράζεται με την ισότητα $u(g_i(A_j)) = u(g_i(A_{j+1}))$.

Βήμα 2

Για κάθε εναλλακτική A_i εισάγονται δύο σφάλματα ένα σφάλμα σ^+ και ένα σ^- και γράφεται για κάθε ζεύγος σφαλμάτων η αναλυτική έκφραση διαφοράς:

$$\Delta(A_i, A_{i+1}) = U[g(A_i)] - \sigma_i^+ + \sigma_i^- - [U[g(A_{i+1})] - \sigma_{i+1}^+ + \sigma_{i+1}^-] \quad (6)$$

Βήμα 3

Λύση του γραμμικού προβλήματος:

$$\min z = \sum_{i=1}^k (\sigma_i^+ + \sigma_i^-)$$

Υπό τους περιορισμούς:

$$\begin{aligned} \Delta(A_i, A_{i+1}) &\geq \delta && \text{εάν } A_i > A_{i+1} \\ \Delta(A_i, A_{i+1}) &= \delta && \text{εάν } A_i \sim A_{i+1} \\ \sum_{j=1}^n p_j &= 1 \\ p_j &\geq 0, \quad j=1,2,\dots,n \\ \sigma_i^+ &\geq 0, \quad \sigma_i^- \geq 0, \quad i=1,2,\dots,k \end{aligned}$$

όπου δ ένας πολύ μικρός αριθμός

Βήμα 4

Στο τελευταίο βήμα γίνεται ανάλυση ευστάθειας. Ο αλγόριθμος αναζητά άλλες βέλτιστες ή σχεδόν βέλτιστες λύσεις του γραμμικού προβλήματος που περιγράφεται στο προηγούμενο βήμα. Στην περίπτωση που οι λύσεις που βρέθηκαν στο προηγούμενο βήμα της μεθόδου, δεν είναι μοναδικές η UTA υπολογίζει τα μέσα βάρη από το σύνολο των βέλτιστων ή σχεδόν βέλτιστων λύσεων που μεγιστοποιούν την αντικειμενική συνάρτηση $z_j = p_j$ για κάθε $j=1,2,\dots,n$ στο πολύεδρο που ορίζουν οι παραπάνω περιορισμοί αλλά και ο ακόλουθος επιπλέον περιορισμός:

$$\sum_{i=1}^k (\sigma_i^+ + \sigma_i^-) \leq z^* + \varepsilon$$

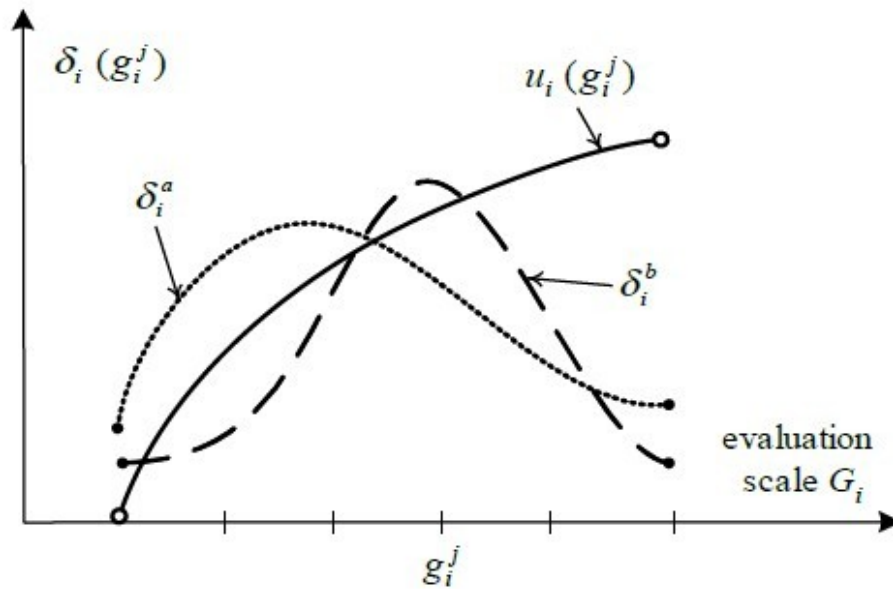
Όπου z^* η βέλτιστη λύση του γραμμικού προβλήματος του 3ου βήματος και ε ένα πολύ μικρό ποσοστό του z^* .

3.6 Η μέθοδος stochastic UTA

Στο πλαίσιο των μεθόδων πολυκριτήριας ανάλυσης υπό καθεστώς αβεβαιότητας αναπτύχθηκε η stochastic UTA (Siskos, Grigoroudis and Matsatsinis, 2016), στην οποία το συνθετικό μοντέλο που πηγάζει από τις αναφερόμενες κατατάξεις είναι μία προσθετική συνάρτηση χρησιμότητας της μορφής:

$$u(\delta^a) = \sum_{i=1}^n \sum_{j=1}^{a_i} \delta_i^a(g_i^j) u_i(g_i^j)$$

όπου δ_i^a είναι η συνάρτηση κατανομής της αξιολόγησης της εναλλακτικής a πάνω στο κριτήριο i , $\delta_i^a(g_i^j)$ η πιθανότητα η αξιολόγηση της εναλλακτικής a πάνω στο κριτήριο i να είναι g_i^j , $u_i(g_i^j)$ η μερική χρησιμότητα της εκτίμησης g_i^j , δ^a είναι το διάνυσμα των συναρτήσεων κατανομής της αξιολόγησης της εναλλακτικής a και, $u(\delta^a)$ η συνολική χρησιμότητα της εναλλακτικής a .



Σχήμα 3.2: Εξέλιξη χρησιμότητας σε συνάρτηση με την κατανομή

$$\sum_{j=1}^{a_i} \delta_i^\alpha(g_i^j) = 1$$

Η αθροιστική συνάρτηση χρησιμότητας έχει τις ίδιες ιδιότητες με την συνάρτηση αξίας :

$$\begin{cases} u(\delta^\alpha) > u(\delta^\beta) & \alpha > \beta \\ u(\delta^\alpha) = u(\delta^\beta) & \alpha = \beta \end{cases}$$

Τα βήματα της μεθόδου είναι τα εξής:

Βήμα 1

Έκφραση των εκτιμώμενων χρησιμοτήτων της κάθε εναλλακτικής $u(\delta^{a_k})$, $k=1,2,\dots,m$ σε μεταβλητές w :

$$w_{ij} = u_i(g_i^{j+1}) - u_i(g_i^j) \geq 0$$

Βήμα 2

Εισαγωγή του σφάλματος υποεκτίμησης και υπερεκτίμησης σ^+ και ένα σ^- δομώντας την παρακάτω έκφραση:

$$\Delta(\alpha_k, \alpha_{k+1}) = u(\delta^{a_k}) - \sigma_i^+ + \sigma_i^- - [u(\delta^{a_{k+1}}) - \sigma_{i+1}^+ + \sigma_{i+1}^-]$$

Βήμα 3

Επίλυση του γραμμικού προβλήματος:

$$\min z = \sum_{i=1}^k (\sigma_i^+ + \sigma_i^-)$$

Υπό τους περιορισμούς;

$$\Delta(A_i, A_{i+1}) \geq \delta \quad \text{εάν } A_i > A_i$$

$$\Delta(A_i, A_{i+1}) = \delta \quad \text{εάν } A_i \sim A_i$$

$$\sum_i \sum_j w_{ij} = 1$$

$$w_{il} \geq 0, \sigma^+(\alpha) \geq 0, \sigma^-(\alpha) \geq 0 \quad \text{για } i=1,2,\dots,n; \quad l=1,2,\dots,a_i-1$$

Βήμα 4

Στο τελευταίο βήμα γίνεται ανάλυση ευστάθειας. Ο αλγόριθμος αναζητά άλλες βέλτιστες ή σχεδόν βέλτιστες λύσεις του γραμμικού προβλήματος που περιγράφεται στο προηγούμενο βήμα. Οι τεχνικές που μπορούν να χρησιμοποιηθούν είναι οι ίδιες με τις παραπάνω μεθόδους.

Κεφάλαιο 4: Η εργαλειοθήκη MultiCrit

4.1 Εισαγωγή

Ακολουθώντας την κουλτούρα του ανοιχτού κώδικα, κύριος στόχος αυτής της διπλωματικής είναι η ανάπτυξη ενός εργαλείου το οποίο θα μπορεί με την χρήση κώδικα σε Python να πραγματοποιήσει εφαρμογή μεθόδων πολυκριτήριας ανάλυσης. Αυτή η εργαλειοθήκη μπορεί είτε να ενσωματωθεί σε συστήματα και εφαρμογές είτε να χρησιμοποιηθεί σαν αυτόνομο εργαλείο για την ανάλυση αποφάσεων. Με αυτόν τον τρόπο δίνεται η δυνατότητα σε ανθρώπους του τομέα να αξιοποιούν το έτοιμο λογισμικό ως δομικό στοιχείο και να εστιάσουν την παραγωγικότητα τους και την προσοχή τους σε άλλες διαδικασίες που απαιτεί η εργασία τους. Η MultiCrit, πέρα από μεθόδους πολυκριτήριας ανάλυσης, περιέχει συστήματα ελέγχου των δεδομένων και τεχνικές αυτόματης εισαγωγής δεδομένων για την διασφάλιση της ορθότητας των αποτελεσμάτων και την ευκολότερη χρήση των μεθόδων.

4.2 Εισαγωγή δεδομένων στο σύστημα, φίλτρα και αυτοματισμοί

Ένα μεγάλο πρόβλημα που αντιμετωπίζουν οι πολυκριτήριες μέθοδοι είναι η ορθότητα των δεδομένων που εισάγονται στο σύστημα. Προκειμένου να αποφευχθούν υπολογιστικά λάθη κατά την διάρκεια της εισαγωγής δεδομένων το πρόγραμμα περνάει τα δεδομένα μέσα από ένα σύστημα φίλτρων. Τα φίλτρα αυτά αποσκοπούν στον εντοπισμό σφαλμάτων που επιτρέπουν την διεξαγωγή και ολοκλήρωση της μεθόδου αλλά επιστρέφουν λανθασμένα αποτελέσματα που δεν εντοπίζονται εύκολα. Οι έλεγχοι που εκτελούνται είναι οι εξής :

- Έλεγχος δεδομένων πολυκριτήριου πίνακα και πίνακα κατατάξεων ή ομάδων για κενές ή μη αριθμητικές τιμές.
- Έλεγχος συνέπειας του πολυκριτήριου πίνακα με τις χειρότερες και τις καλύτερες τιμές.
- Έλεγχος των κατατάξεων για παράληψη αριθμού κατάταξης, και συμμετρία ως προς τις εναλλακτικές.
- Έλεγχος της μονοτονία των κριτηρίων.

Το σύστημα δεν εστιάζει μόνο στην σωστή διεκπεραίωση των υπολογισμών αλλά και στην βελτίωση της παραγωγικότητας του χρήστη. Μέσα από μία σειρά αυτόματων

διαδικασιών δίνεται η δυνατότητα να συνεπάγει πληροφορίες που θέλει να χρησιμοποιήσει ο χρήστης, τις οποίες υπό κανονικές συνθήκες θα τις περνούσε στο σύστημα χειροκίνητα. Οι αυτοματισμοί αυτοί είναι οι εξής:

- Σε περίπτωση που ο χρήστης δεν εισάγει μονοτονία το σύστημα λαμβάνει υπόψιν ότι όλα τα κριτήρια έχουν αύξουσα μονοτονία.
- Σε περίπτωση που ο χρήστης δεν εισάγει καλύτερη και χειρότερη τιμή το σύστημα λαμβάνει υπόψιν ότι η χειρότερες και οι καλύτερες τιμές είναι αυτές που περιέχει μέσα ο πολυκριτήριοι πίνακας, λαμβάνοντας υπόψιν βεβαίως την μονοτονία του κριτηρίου.
- Σε περίπτωση που ο χρήστης δεν εισάγει κατάταξη στις μεθόδους UTA και UTASTAR το σύστημα υποθέτει ότι η σειρά με την οποία βρίσκονται οι εναλλακτικές στον πίνακα είναι και η σειρά κατάταξη τους.
- Σε περίπτωση που ο χρήστης δεν εισάγει αριθμό διαστημάτων τότε το σύστημα υποθέτει ότι όλα τα κριτήρια χωρίζονται σε 3 διαστήματα.

4.3 Εφαρμογή της UTA

Για την υλοποίηση της κλασικής μεθόδου UTA ο χρήστης χρειάζεται αν καλέσει από την εργαλειοθήκη την εντολή **UTA(data, alternatives, criteria, ranking, monotonicity, bestvalues, worstvalues, intervals, method, epsilon, delta, sigma)**, όπως φαίνεται και στην Εικόνα 4.1. Όπως αναφέρθηκε παραπάνω (4.2) το σύστημα διαθέτει αυτοματισμούς για την ευκολότερη εισαγωγή των δεδομένων καθώς και κάποιες προκαθορισμένες τιμές, σύμφωνα με την βιβλιογραφία, σε μεταβλητές όπως η epsilon και η sigma. Όλες οι μεταβλητές της εντολής δίνουν την δυνατότητα παραμετροποίησης. Για την αποδοχή από την συνάρτηση τα δεδομένα χρειάζεται να έχουν την εξής μορφή:

data:	Η μεταβλητή αυτή αφορά τα δεδομένα του πολυκριτηρίου πίνακα, τα οποία εισάγονται είτε σε μορφή pandas data frame είτε σε numpy array. Στην περίπτωση που εισαχθούν σε μορφή pandas data frame θα πρέπει να περιέχεται και μία στήλη με ετικέτα “Ranking” ή “Rank” που θα περιγράφει τις κατατάξεις.
alternatives:	Δέχεται σε μορφή λίστας τα ονόματα των εναλλακτικών. Σε περίπτωση που δεν εισαχθεί το σύστημα ονομάζει τις εναλλακτικές ως alternative1, alternative2,...
criteria:	Δέχεται σε μορφή λίστας τα ονόματα των κριτηρίων. Σε περίπτωση που δεν εισαχθεί το σύστημα ονομάζει τα κριτήρια criteria1, criteria2...
ranking	Δέχεται σε μορφή λίστας ή πίνακα numpy την κατάταξη των εναλλακτικών. Σε περίπτωση pandas dataframe δεν χρειάζεται η εισαγωγή αυτής της μεταβλητής.
monotonicity:	Δέχεται σε μορφή λίστας ή πίνακα numpy την μονοτονία των εναλλακτικών τιμών με αποδεκτές τιμές 1 για αύξουσα μονοτονία και 0 για φθίνουσα.
bestvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις καλύτερες τιμές που μπορεί να λάβει

κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την καλύτερη τιμή την τιμή του πολυκριτηρίου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.

- worstvalues:** Δέχεται σε μορφή λίστας ή πίνακα numpy τις χειρότερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την χειρότερη τιμή την τιμή του πολυκριτηρίου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
- intervals:** Δέχεται σε μορφή λίστας ή πίνακα numpy τα διαστήματα στα οποία θα χωριστεί το κάθε κριτήριο. Αν δεν δοθεί τιμή χωρίζει όλα τα κριτήρια σε 3 διαστήματα
- method:** Δέχεται σαν είσοδο την μέθοδο με την οποία θα επιλυθεί το γραμμικό πρόβλημα. Οι αποδεκτές τιμές είναι αυτές που δέχεται η linprog της scipy, δηλαδή οι τιμές “simplex” και “interior-point”.
- epsilon:** Η μεταβλητή epsilon καθορίζει το αποδεκτό σφάλμα που ο χρήστης είναι διατεθειμένος να αποδεχτεί κατά την διάρκεια της μετά-βελτιστοποίησης . Προκαθορισμένη τιμή είναι 0,005.
- delta:** Η μεταβλητή delta ορίζει την ελάχιστη απόσταση μεταξύ της κατάταξης μίας εναλλακτικής με την αμέσως χειρότερη της κατά την δόμηση του γραμμικού προβλήματος. Προκαθορισμένη τιμή είναι 0,05.
- sigma:** Η μεταβλητή sigma καθορίζει την ελάχιστη διαφορά που πρέπει να έχουν οι χρησιμότητες ενός κριτηρίου όσο αυξάνεται η τιμή σου στο διακριτό διάστημα. Προκαθορισμένη τιμή είναι 0.

```
UTA(data, alternatives=None, criteria=None, ranking=None, monotonicity=None,  
bestvalues=None, worstvalues=None, intervals=None, method='interior-point',  
epsilon=0.005, delta=0.05, sigma=0):
```

Εικόνα 4.1.1: Ορίσματα της συνάρτησης UTA

Για την καλύτερη κατανόηση της μεθόδου και των αποτελεσμάτων που εξάγονται δίνεται στην Εικόνα 4.2 το παράδειγμα των δεδομένων για το οποία θα υπολογιστούν τα αποτελέσματα είναι το παράδειγμα από το βιβλίο του Ματσατσίνη όπου ως εναλλακτικές ο χρήστης έχει να επιλέξει ανάμεσα σε διάφορα μεταφορικά μέσα για την μετακίνηση του τα οποία τα κατατάσσει ως προς την τιμή στην πρώτη θέση του πίνακα, την χρονική διάρκεια στην δεύτερη θέση του πίνακα και την άνεση στην τρίτη θέση.

```

df = np.array([[3, 10, 1], [4, 20, 2], [2, 20, 0], [6, 40, 0], [30, 30, 3]])
ranking = [1, 2, 2, 3, 4]
alternatives = ['a', 'b', 'c', 'd', 'f']
criteria = ['1st', '2nd', '3rd']
bestvalues = [2, 10, 3]
worstvalues = [30, 40, 0]
monotonicity = [0, 0, 1]
intervals = [2, 3, 3]

```

Εικόνα 4.1.2: Παράδειγμα κατανόησης

Ως αποτέλεσμα η συνάρτηση επιστρέφει ένα αντικείμενο που περιέχει μέσα τις χρησιμότητες των εναλλακτικών, τις χρησιμότητες για όλες τις διακριτές τιμές των κριτηρίων, τις διακριτές τιμές που έχουν οριστεί για κάθε κριτήριο, τα ονόματα των εναλλακτικών και των κριτηρίων, τα σφάλματα που προέκυψαν από κάθε εναλλακτική, τις τιμές των χρησιμοτήτων της μετά-βελτιστοποίησης, τις κατατάξεις και τον δείκτη ASI. Εφόσον τα αποτελέσματα ορίζονται ως αντικείμενο μπορούν να κληθούν ως γνωρίσματα του. Τα γνωρίσματα είναι τα εξής:

- .g: Επιστρέφει σε μορφή λίστας τις διακριτές τιμές στις οποίες χωρίστηκαν τα κριτήρια.
- .crit_names: Επιστρέφει σε μορφή λίστας τα ονόματα των κριτηρίων.
- .alternatives: Επιστρέφει σε μορφή λίστας τα ονόματα των εναλλακτικών.
- .crit_utilities: Επιστρέφει τις χρησιμότητες των κριτηρίων σε μορφή πίνακα numpy.
- .all_utilities: Επιστρέφει σε μορφή numpy array όλες τις χρησιμότητες που υπολογίστηκαν κατά την διάρκεια της μετά-βελτιστοποίησης.
- .alts_utilities: Επιστρέφει τις χρησιμότητες των εναλλακτικών σε μορφή πίνακα numpy.
- .errors: Επιστρέφει τα σφάλματα σε μορφή πίνακα numpy.
- .ranking: Επιστρέφει σε μορφή λίστας τις κατατάξεις
- .ASI: Επιστρέφει την τιμή του δείκτη ASI(average stability index) (Matsatsinis, Grigoroudis and Siskos, 2018)

Τρέχοντας την μέθοδο UTA για το σετ δεδομένων της Εικόνας 4.2 και χρησιμοποιώντας την εντολή print γίνονται ορατά όλα τα γνωρίσματα και το περιεχόμενό τους .

```

g: [[30.0, 16.0, 2.0], [40.0, 30.0, 20.0, 10.0], [0.0, 1.0, 2.0, 3.0]]
criteria: ['Price', 'Time', 'Comfort']
crit_names: ['u1.1', 'u1.2', 'u1.3', 'u2.1', 'u2.2', 'u2.3', 'u2.4', 'u3.1', 'u3.2', 'u3.3', 'u3.4']
crit_utilities: array([0.22214982, 0.42964439, 0.09870077], [0.27803748, 0.47165484, 0.0267005], [0.02631294, 0.02696983, 0.00000000e+00,
1.41041065e-10, 5.99415027e-11, 3.24999987e-02, 0.00000000e+00,
2.50000003e-02, 2.50000002e-02, 2.50000001e-02],
[0.00000000e+00, 4.49999992e-02, 4.49999994e-02, 0.00000000e+00,
9.78601568e-12, 5.08373108e-01, 9.55000001e-01, 0.00000000e+00,
4.84253257e-11, 2.01777689e-11, 6.27248751e-13],
[0.00000000e+00, 3.58750000e-01, 5.33750000e-01, 0.00000000e+00,
2.89855031e-13, 1.33068246e-13, 3.24999994e-02, 0.00000000e+00,
2.50000000e-02, 2.50000000e-02, 4.33750000e-01],
[0.00000000e+00, 4.49999979e-02, 4.49999993e-02, 0.00000000e+00,
1.76610545e-10, 4.98743537e-01, 9.55000002e-01, 0.00000000e+00,
4.72610543e-10, 2.73045118e-10, 1.01238287e-10],
[0.00000000e+00, 7.45962800e-02, 8.66545367e-01, 0.00000000e+00,
5.78575528e-13, 4.36340305e-13, 1.42397183e-13, 0.00000000e+00,
1.07877663e-01, 1.11818991e-01, 1.33454633e-01],
[0.00000000e+00, 3.77378605e-01, 3.97133652e-01, 0.00000000e+00,
1.60203026e-01, 3.25782295e-01, 6.02866348e-01, 0.00000000e+00,
4.12566128e-10, 2.90778316e-10, 2.05626593e-10]))
alternatives: ['a', 'b', 'c', 'd', 'f']
alts_utilities: array([0.91378236, 0.69311487, 0.69380466, 0.41633559, 0.12540128])
errors: array([1.88631176e-03, 7.19987298e-04, 3.01968224e-05, 1.79161687e-03,
5.71888665e-04])
ranking: [1, 2, 2, 3, 4]
ASI: 0.6190734627900436

```

Εικόνα 4.1.3: Έξοδος της μεθόδου UTA

Παρακάτω φαίνεται ένα παράδειγμα για την εφαρμογή της συνάρτησης με την χρήση των pandas dataframes όπου τα δεδομένα του πολυκριτήριου πίνακα εισάγονται από αρχείο excel. Στην παρακάτω εικόνα φαίνεται η δομή που πρέπει να έχει ο πολυκριτήριος πίνακας. Η στήλη με τις κατατάξεις θα πρέπει αυστηρά να έχει την ονομασία 'Rank' ή 'Ranking'.

	A	B	C	D	E	F
1	transport	Price	Time	Rank	Comfort	
2	A	3	10	1	1	
3	B	4	20	2	2	
4	C	2	20	2	0	
5	D	6	40	3	0	
6	E	30	30	4	3	
7						

Εικόνα 4.1.4: Πολυκριτήριος Πίνακας

Τα δεδομένα των καλύτερων τιμών, των χειρότερων τιμών, της μονοτονίας και των διαστημάτων θα εισαχθούν μέσα στον κώδικα.

```
1 from MultiCrit.UtaMethods import UTA, UTASTAR, UTADIS
2 import pandas as pd
3
4 data = pd.read_excel("utastardata.xlsx")
5 bestvalues = [2, 10, 3]
6 worstvalues = [30, 40, 0]
7 monotonicity = [0, 0, 1]
8 intervals = [2, 3, 3]
9
10 res = UTA(data, bestvalues=bestvalues, worstvalues=worstvalues,
11           intervals=intervals, monotonicity=monotonicity)
12
13 print(res)
```

Εικόνα 4.1.5: Κλήση της μεθόδου UTA

Τα αποτελέσματα με τα γνωρίσματα της μεθόδου παρουσιάζονται στην επόμενη εικόνα. Ο κάθε χρήστης μπορεί να δει μέσα από την εκτύπωση της οριζόμενης μεταβλητής, τις ονομασίες των γνωρισμάτων προκειμένου να εξάγει μεμονωμένα ότι πληροφορία χρειάζεται.


```

g: [[30.0, 16.0, 2.0], [40.0, 30.0, 20.0, 10.0], [0.0, 1.0, 2.0, 3.0]]
criteria: ['Price', 'Time', 'Comfort']
crit_names: ['u1.1', 'u1.2', 'u1.3', 'u2.1', 'u2.2', 'u2.3', 'u2.4', 'u3.1', 'u3.2', 'u3.3', 'u3.4']
crit_utilities: array([0.22214982, 0.42964439, 0.027803748, 0.47165484, 0.0267005, 0.0267005, 0.0267005, 0.0267005, 0.0267005, 0.0267005, 0.0267005, 0.0267005])
all_utilities: array([[0.00000000e+00, 7.67500001e-01, 9.42500002e-01, 0.00000000e+00, 1.41041065e-10, 5.99415027e-11, 3.24999998e-02, 0.00000000e+00, 2.50000003e-02, 2.50000002e-02, 2.50000001e-02], [0.00000000e+00, 4.49999992e-02, 4.49999994e-02, 0.00000000e+00, 9.78601568e-12, 5.08373108e-01, 9.55000001e-01, 0.00000000e+00, 4.84253257e-11, 2.01777689e-11, 6.27248751e-13], [0.00000000e+00, 3.58750000e-01, 5.33750000e-01, 0.00000000e+00, 2.89855031e-13, 1.33068246e-13, 3.24999994e-02, 0.00000000e+00, 2.50000000e-02, 2.50000000e-02, 4.33750000e-01], [0.00000000e+00, 4.49999979e-02, 4.49999993e-02, 0.00000000e+00, 1.76610545e-10, 4.98743537e-01, 9.55000002e-01, 0.00000000e+00, 4.72610543e-10, 2.73045118e-10, 1.01238287e-10], [0.00000000e+00, 7.45962800e-02, 8.66545367e-01, 0.00000000e+00, 5.78575528e-13, 4.36340305e-13, 1.42397183e-13, 0.00000000e+00, 1.07877663e-01, 1.11818991e-01, 1.33454633e-01], [0.00000000e+00, 3.77378605e-01, 3.97133652e-01, 0.00000000e+00, 1.60203026e-01, 3.25782295e-01, 6.02866348e-01, 0.00000000e+00, 4.12566128e-10, 2.90778316e-10, 2.05626593e-10]])
alternatives: ['a', 'b', 'c', 'd', 'f']
alts_utilities: array([0.91378236, 0.69311487, 0.69380466, 0.41633559, 0.12540128])
errors: array([1.88631176e-03, 7.19987298e-04, 3.01968224e-05, 1.79161687e-03, 5.71888665e-04])
ranking: [1, 2, 2, 3, 4]
ASI: 0.6190734627900436

```

Εικόνα 4.1.6: Γνωρίσματα της UTA

Καλώντας το εκάστοτε γνώρισμα ο χρήστης έχει πρόσβαση στην πληροφορία που περιέχει. Για παράδειγμα, αν ο χρήστης χρειαστεί τις χρησιμότητες των εναλλακτικών απλά χρειάζεται να καλέσει **res.alts_utilities** και θα εξαχθεί ο πίνακας με τις τιμές.

```
res.alts_utilities
```

```
array([0.91378236, 0.69311487, 0.69380466, 0.41633559, 0.12540128])
```

Εικόνα 4.1.7: Κλήση των χρησιμοτήτων των εναλλακτικών

4.4 Εφαρμογή της UTASTAR

Για την εφαρμογή της μεθόδου UTASTAR ο χρήστης χρειάζεται να καλέσει από την εργαλειοθήκη την εντολή **UTASTAR(data, alternatives, criteria, ranking, monotonicity, bestvalues, worstvalues, intervals, method, epsilon, delta)** όπως φαίνεται στην Εικόνα 4.4. Οι αυτοματισμοί και τα φίλτρα που εφαρμόζονται στην UTA εφαρμόζονται και στην UTASTAR. Για την αποδοχή των δεδομένων από την συνάρτηση, τα δεδομένα πρέπει να έχουν την εξής μορφή:

data:	Η μεταβλητή αυτή αφορά τα δεδομένα του πολυκριτήριου πίνακα, τα οποία εισάγονται είτε σε μορφή pandas data frame είτε σε numpy array. Στην περίπτωση που εισαχθούν σε μορφή pandas data frame θα πρέπει να περιέχεται και μία στήλη με ετικέτα “Ranking” ή “Rank” που θα περιγράφει τις κατατάξεις.
alternatives:	Δέχεται σε μορφή λίστας τα ονόματα των εναλλακτικών. Σε περίπτωση που δεν εισαχθεί το σύστημα ονομάζει τις εναλλακτικές ως alternative1, alternative2,...
criteria:	Δέχεται σε μορφή λίστας τα ονόματα των κριτηρίων. Σε περίπτωση που δεν εισαχθεί το σύστημα ονομάζει τα κριτήρια criteria1, criteria2...
ranking	Δέχεται σε μορφή λίστας ή πίνακα numpy την κατάταξη των εναλλακτικών. Σε περίπτωση pandas dataframe δεν χρειάζεται η εισαγωγή αυτής της μεταβλητής.
monotonicity:	Δέχεται σε μορφή λίστας ή πίνακα numpy την μονοτονία των εναλλακτικών τιμών με αποδεκτές τιμές 1 για αύξουσα μονοτονία και 0 για φθίνουσα.
bestvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις καλύτερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την καλύτερη τιμή την τιμή του πολυκριτήριου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
worstvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις χειρότερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την χειρότερη τιμή την τιμή του πολυκριτήριου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
intervals:	Δέχεται σε μορφή λίστας ή πίνακα numpy τα διαστήματα στα οποία θα χωριστεί το κάθε κριτήριο. Αν δεν δοθεί τιμή χωρίζει όλα τα κριτήρια σε 3 διαστήματα
method:	Δέχεται σαν είσοδο την μέθοδο με την οποία θα επιλυθεί το γραμμικό πρόβλημα. Οι αποδεκτές τιμές είναι αυτές που δέχεται η linprog της scipy, δηλαδή οι τιμές “simplex” και “interior-point”.
epsilon:	Η μεταβλητή epsilon καθορίζει το αποδεκτό σφάλμα που ο χρήστης είναι διατεθειμένος να αποδεχτεί κατά την διάρκεια της μετά-βελτιστοποίησης . Προκαθορισμένη τιμή είναι 0,005.
delta:	Η μεταβλητή delta ορίζει την ελάχιστη απόσταση μεταξύ της κατάταξης μίας εναλλακτικής με την αμέσως χειρότερη της κατά την δόμηση του γραμμικού προβλήματος. Προκαθορισμένη τιμή είναι 0,05.

```
UTASTAR(data, alternatives=None, criteria=None, ranking=None, monotonicity=None,
         bestvalues=None, worstvalues=None, intervals=None, epsilon=0.005, delta=0.05,
         method='interior-point')
```

Εικόνα 4.2.1: Ορίσματα της συνάρτησης UTASTAR

Αντίστοιχα με την UTA η συνάρτηση UTASTAR επιστρέφει ως αποτέλεσμα ένα αντικείμενο. Το αντικείμενο αυτό επιστρέφει ως γνωρίσματα τις διακριτές τιμές που χωρίστηκαν τα κριτήρια, τις χρησιμότητες των κριτηρίων, τα ονόματα των κριτηρίων, τις χρησιμότητες w του κάθε κριτηρίου, τα ονόματα των χρησιμοτήτων w , όλες τις χρησιμότητες που υπολογίστηκαν κατά την μετά-βελτιστοποίηση, τις χρησιμότητες των εναλλακτικών, τα ονόματα των εναλλακτικών, τα σφάλματα που προέκυψαν από κάθε εναλλακτική και τον δείκτη ASI. Τα γνωρίσματα που χρειάζεται να καλέσει ο χρήστης είναι τα εξής:

.g:	Επιστρέφει σε μορφή λίστας τις διακριτές τιμές στις οποίες χωρίστηκαν τα κριτήρια.
.crit_names:	Επιστρέφει σε μορφή λίστας τα ονόματα των κριτηρίων.
.alternatives:	Επιστρέφει σε μορφή λίστας τα ονόματα των εναλλακτικών.
.crit_utilities:	Επιστρέφει τις χρησιμότητες των κριτηρίων σε μορφή πίνακα numpy.
.alts_utilities	Επιστρέφει τις χρησιμότητες των εναλλακτικών σε μορφή πίνακα numpy.
.w	Επιστρέφει τις χρησιμότητες w του κάθε κριτηρίου σε μορφή πίνακα numpy.
.w_names	Επιστρέφει σε μορφή λίστας τα ονόματα των εναλλακτικών.
.all_w	Επιστρέφει σε μορφή numpy array όλες τις χρησιμότητες που υπολογίστηκαν κατά την διάρκεια της μετά-βελτιστοποίησης.
.errors:	Επιστρέφει τα σφάλματα υποεκτίμησης και υπερεκτίμησης σε μορφή πίνακα numpy.
.ranking	Επιστρέφει σε μορφή λίστας τις κατατάξεις
.ASI:	Επιστρέφει την τιμή του δείκτη ASI (average stability index)

```

g: [[30.0, 16.0, 2.0], [40.0, 30.0, 20.0, 10.0], [0.0, 1.0, 2.0, 3.0]]
criteria: ['Price', 'Time', 'Comfort']
crit_names: ['u1.1', 'u1.2', 'u1.3', 'u2.1', 'u2.2', 'u2.3', 'u2.4', 'u3.1', 'u3.2', 'u3.3', 'u3.4']
crit_utilities: array([0.00000000e+00, 3.89866222e-01, 5.06532889e-01, 0.00000000e+00,
3.79202718e-11, 1.58029077e-01, 3.40550445e-01, 0.00000000e+00,
1.73707610e-02, 1.73707611e-02, 1.52916667e-01])
w_names: ['w1.1', 'w1.2', 'w2.1', 'w2.2', 'w2.3', 'w3.1', 'w3.2', 'w3.3']
w: array([[3.89866222e-01, 1.16666667e-01, 3.79202718e-11, 1.58029077e-01,
1.82521368e-01, 1.73707610e-02, 2.14793444e-11, 1.35545906e-01]])
all_w: array([[7.67499999e-01, 1.75000002e-01, 2.27638581e-11, 2.64382570e-11,
3.25000001e-02, 2.49999999e-02, 2.26315351e-11, 1.88461131e-11],
[4.49999995e-02, 1.03529698e-10, 8.38201316e-11, 4.74087230e-01,
4.80912772e-01, 3.65134677e-11, 3.62910068e-11, 2.56997836e-11],
[3.57098670e-01, 1.74999999e-01, 7.17682559e-12, 1.11079489e-12,
3.41513314e-02, 2.71122832e-02, 5.51549122e-12, 4.06637717e-01]])
alternatives: ['a', 'b', 'c', 'd', 'f']
alts_utilities: array([0.85612076, 0.66526606, 0.66456197, 0.47319956, 0.15291667])
errors: array([2.07879524e-03, 1.62765289e-10, 2.39634478e-10, 7.04094408e-04,
2.75134561e-10, 2.87414816e-10, 9.13746074e-04, 5.79570893e-11,
3.02487570e-11, 1.30336426e-03])
ranking: [1, 2, 2, 3, 4]
ASI: 0.8494739974117704

```

Εικόνα 4.2.2: Αποτελέσματα της UTASTAR

Τρέχοντας την μέθοδο UTASTAR για το παράδειγμα του προηγούμενου κεφαλαίου, τα αποτελέσματα παρουσιάζονται στην Εικόνα 4.2.2. Τα σφάλματα που προκύπτουν από το γνώρισμα error ακολουθούν την εξής αλληλουχία, στην πρώτη θέση βρίσκεται το σφάλμα υποεκτίμησης της πρώτης εναλλακτικής, και στην δεύτερη θέση το σφάλματα υπερεκτίμησης αντίστοιχα, στην τρίτη, και στην τέταρτη θέση το σφάλμα υποεκτίμησης και υπερεκτίμησης της δεύτερης εναλλακτικής και ούτω καθεξής.

Για καλύτερη κατανόηση παρουσιάζεται το παρακάτω παράδειγμα που έχει ως βάση το αρχείο excel της προηγούμενης μεθόδου. Στην επόμενη εικόνα φαίνονται τα δεδομένα που χρειάζονται για εισαγωγή και την εφαρμογή της μεθόδου.

```

1  from MultiCrit.UtaMethods import UTA, UTASTAR, UTADIS
2  import pandas as pd
3
4  data = pd.read_excel("utastardata.xlsx")
5  bestvalues = [2, 10, 3]
6  worstvalues = [30, 40, 0]
7  monotonicity = [0, 0, 1]
8  intervals = [2, 3, 3]
9
10 res = UTASTAR(data, bestvalues=bestvalues, worstvalues=worstvalues,
11                intervals=intervals, monotonicity=monotonicity)
12
13 print(res)

```

Εικόνα 4.2.3: Εισαγωγή δεδομένων μέσω excel και εφαρμογή της UTASTAR

Τα αποτελέσματα με τα γνωρίσματα της μεθόδου είναι τα ίδια με την εικόνα 4.2.3. Ο κάθε χρήστης μπορεί να δει μέσα από την εκτύπωση της οριζόμενης μεταβλητής, τις ονομασίες των γνωρισμάτων προκειμένου να εξάγει μεμονωμένα ότι πληροφορία

χρειάζεται. Για παράδειγμα αν ο χρήστης θέλει να εξάγει τα δεδομένα των χρησιμοτήτων w θα καλέσει το γνώρισμα `res.w`.

```
res.w
```

```
array([[3.89866222e-01, 1.16666667e-01, 3.79202718e-11, 1.58029077e-01,  
       1.82521368e-01, 1.73707610e-02, 2.14793444e-11, 1.35545906e-01]])
```

Εικόνα 4.2.4: Εξαγωγή γνώρισματος w

4.5 Εφαρμογή της UTADIS

Σε αντίθεση με τις προηγούμενες πολυκριτήριας μεθόδους, η UTADIS ορίζεται από την εργαλειοθήκη ως κλάση και υλοποιείται μέσω μεθόδων που περιέχονται μέσα στην κλάση. Για τον ορισμό της καθορίζεται μια μεταβλητή **res = UTADIS()**. Στην συνέχεια ορίζεται η μέθοδος της κλάσης, **fit(data, groups, pref_order, monotonicity=None, bestvalues=None, worstvalues=None, intervals=None, epsilon=0.005, delta=0.05, sigma=0.1, method='interior-point')**. Σκοπός της μεθόδου **fit** είναι να δημιουργηθεί το μοντέλο με το οποίο θα μπορούν να γίνουν προβλέψεις σε δεδομένα με άγνωστες ομάδες. Στην εικόνα 4.5.1 παρουσιάζεται ο τρόπος με τον οποίο καλείται η μέθοδος. Όπως και στις προηγούμενες πολυκριτήριες μεθόδους τα δεδομένα φιλτράρονται πριν εφαρμοστούν στο σύστημα. Για την αποδοχή από την μέθοδο, τα δεδομένα πρέπει να έχουν την εξής μορφή.

data:	Η μεταβλητή αυτή αφορά τα δεδομένα του πολυκριτήριου πίνακα, τα οποία εισάγονται είτε σε μορφή pandas data frame είτε σε numpy array.
groups	Δέχεται σε μορφή λίστας τις ομάδες για κάθε εναλλακτική. Η εισαγωγή της μεταβλητής είναι υποχρεωτική.
pref_order	Δέχεται σε μορφή λίστας τις προκαθορισμένες ομάδες με αύξουσα σειρά ξεκινώντας από την θέση 0 της λίστας, μέχρι την τελευταία. Η εισαγωγή της μεταβλητής είναι υποχρεωτική.
monotonicity:	Δέχεται σε μορφή λίστας ή πίνακα numpy την μονοτονία των εναλλακτικών τιμών με αποδεκτές τιμές 1 για αύξουσα μονοτονία και 0 για φθίνουσα.
bestvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις καλύτερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την καλύτερη τιμή την τιμή του πολυκριτήριου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
worstvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις χειρότερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την χειρότερη τιμή την τιμή του πολυκριτήριου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
intervals:	Δέχεται σε μορφή λίστας ή πίνακα numpy τα διαστήματα στα οποία θα χωριστεί το κάθε κριτήριο. Αν δεν δοθεί τιμή χωρίζει όλα τα κριτήρια σε 3 διαστήματα
method:	Δέχεται σαν είσοδο την μέθοδο με την οποία θα επιλυθεί το γραμμικό πρόβλημα. Οι αποδεκτές τιμές είναι αυτές που δέχεται η linprog της scipy, δηλαδή οι τιμές “simplex” και “interior-point”.
epsilon:	Η μεταβλητή epsilon καθορίζει το αποδεκτό σφάλμα που ο χρήστης είναι διατεθειμένος να αποδεχτεί κατά την διάρκεια της μετά-βελτιστοποίησης . Προκαθορισμένη τιμή είναι 0,005.
delta:	Η μεταβλητή delta ορίζει την ελάχιστη απόσταση μεταξύ της κατάταξης μίας εναλλακτικής με την αμέσως χειρότερη της κατά την δόμηση του γραμμικού προβλήματος. Προκαθορισμένη τιμή είναι 0,05.
sigma	Η μεταβλητή sigma είναι ένας μικρός θετικός αριθμός που δηλώνει την προτίμηση μεταξύ των ορίων που διαχωρίζουν τις κλάσεις. Προκαθορισμένη τιμή είναι 0,1.

```
res = UTADIS()
res.fit(data=df, groups=groups, pref_order=pref_order,
        bestvalues=bestvalues, worstvalues=worstvalues,
        intervals=intervals, epsilon=0.005, delta=0.05,
        sigma=0.1, method='interior-point')
```

Εικόνα 4.5.1: Ορισμός κλάσης UTADIS και μεθόδου fit

Για την καλύτερη κατανόηση των αποτελεσμάτων δίνεται ένα παράδειγμα από δεδομένα καταστημάτων που πάρθηκε από έρευνα που πραγματοποιήθηκε στο Πανεπιστήμιο Πειραιά από τους (Ζυγομήτρος and Σίσκος, 2007) . Για ένα σετ καταστημάτων, ο αποφασίζον λαμβάνοντας υπόψιν τον τζίρο του καταστήματος, τις εγκαταστάσεις του και το μάνατζμεντ κατηγοριοποιεί τις επιχειρήσεις σε καταστήματα που θέλει να συνεργαστεί 'YES', σε καταστήματα που δεν είναι σίγουρος αν θέλει να συνεργαστεί και θέλει να εξετάσει σε μία δεύτερη χρονική στιγμή πιο αναλυτικά '?' και σε καταστήματα που είναι σίγουρος ότι δεν θέλει να συνεργαστεί 'NO'.

Στην Εικόνα 4.5.2 παρουσιάζονται τα δεδομένα που θα εισαχθούν στο πρόγραμμα με την μορφή που ορίζονται στην Python. Στην πρώτη θέση του πίνακα df βρίσκεται το κριτήριο του τζίρου, στην δεύτερη βρίσκεται το κριτήριο των εγκαταστάσεων και στην τρίτη το κριτήριο του μάνατζμεντ.

```
df = np.array([[180, 2, 3],
               [200, 2, 2],
               [40, 3, 3],
               [80, 3, 3],
               [100, 2, 1],
               [70, 1, 2],
               [120, 1, 1]])
pref_order = ['NO', '?', 'YES']
groups = ['YES', 'YES', '?', 'YES', '?', 'NO', 'NO']
bestvalues = [200, 3, 3]
worstvalues = [0, 1, 1]
intervals = [4, 2, 2]
```

Εικόνα 4.5.2: Είσοδος δεδομένων

Ο χρήστης έχει την δυνατότητα να εξετάσει τα στοιχεία που απαρτίζουν το μοντέλο ορίζοντας την εντολή σε μία μεταβλητή $x = \text{res.fit}(\dots)$.

- .g: Επιστρέφει σε μορφή λίστας τις διακριτές τιμές στις οποίες χωρίστηκαν τα κριτήρια.
- .w_values Επιστρέφει τις χρησιμότητες w του κάθε κριτηρίου σε μορφή πίνακα numpy.
- .transpositions Επιστρέφει τα όρια των ομάδων.

- .errors: Επιστρέφει τα σφάλματα υποεκτίμησης και υπερεκτίμησης σε μορφή πίνακα numpy .
- .crit_utilities: Επιστρέφει τις χρησιμότητες των κριτηρίων σε μορφή πίνακα numpy.
- .ASI: Επιστρέφει την τιμή του δείκτη ASI (average stability index)

Ανάλογα την ομάδα που βρίσκεται μία εναλλακτική δημιουργείται είτε ένα σφάλμα υποεκτίμησης , είτε ένα σφάλμα υπερεκτίμησης είτε και τα δύο. Στην εικόνα 4.5.3 παρουσιάζονται τα αποτελέσματα του παραδείγματος.

```

g: [[0, 50.0, 100.0, 150.0, 200.0], [1, 2.0, 3.0], [1, 2.0, 3.0]]
w_values: array([0.07916667, 0.21747115, 0.01684524, 0.06333333, 0.25340476,
0.07755357, 0.15921245, 0.13301282])
all_w: array([[0.475, 0., 0., 0.38, 0.05,
0., 0.095, 0.00357143, 0.84142857,
0.03357143, 0.04642857, 0., 0.03357143, 0.04642857, 0., 0.03357143,
0., 0.38653846, 0., 0., 0.045, 0.045, 0.23192308, 0.33653846],
[0., 0.075, 0., 0., 0.45, 0.45, 0., 0.075, 0., 0., 0.45, 0.45,
0., 0.445, 0.03, 0.03, 0.03, 0.38653846, 0., 0., 0.045, 0.045,
0., 0.23192308, 0.33653846],
[0., 0.38175, 0.0975, 0., 0.089, 0.089, 0.43175, 0., 0., 0., 0., 0.]])
transitionpoints: array([0.74318361, 0.42004258])
errors: array([4.16666667e-04, 4.16666667e-04, 1.66666667e-03, 0.00000000e+00,
8.33333333e-04, 0.00000000e+00, 1.66666667e-03, 0.00000000e+00,
2.01829294e-17])
crit_utilities: array([0., 0.07916667, 0.29663782, 0.31348306, 0.37681639,
0., 0.25340476, 0.33095833, 0., 0.15921245,
0.29222527])
criteria: ['criterion1', 'criterion2', 'criterion3']
ASI: 0.6979149588898582

```

Εικόνα 4.5.3: Παρουσίαση των γνωρισμάτων της UTADIS

Όπως παρουσιάζονται τα σφάλματα στο γνώρισμα errors (Εικόνα 4.5.3), το σφάλμα στην θέση 0 του πίνακα είναι το σφάλμα υπερεκτίμησης της πρώτης εναλλακτικής καθώς ανήκει στην καλύτερη ομάδα, το σφάλμα στην θέση 1 είναι επίσης σφάλμα υπερεκτίμησης της δεύτερης εναλλακτικής, αφού ανήκει και αυτή στην καλύτερη ομάδα, αλλά τα σφάλματα στην θέση 2 και 3 ανήκουν και τα δύο στην τρίτη εναλλακτική ως σφάλμα υπερεκτίμησης και υποεκτίμησης αντίστοιχα από την στιγμή που η εναλλακτική ανήκει στην ενδιάμεση ομάδα.

Για την κατηγοριοποίηση δεδομένων άγνωστης ομάδας ο χρήστης χρειάζεται να καλέσει την μέθοδο της κλάσης **predict(data)** όπως φαίνεται στην Εικόνα 4.5.4. Η συγκεκριμένη μέθοδος επιστρέφει μία λίστα με τις ομάδες που προέβλεψε ότι ανήκουν οι εναλλακτικές.

```

data = np.array([[40, 3, 2],
[130, 1, 3],
[80, 2, 3],
[30, 3, 3],
[70, 2, 1],
[150, 1, 1],
[60, 2, 2],
[180, 1, 2]])
y = res.predict(data)

```

Εικόνα 4.5.4: Εφαρμογή της μεθόδου predict

Με την εντολή `predict` ο χρήστης έχει πρόσβαση μόνο στις ομάδες που πρόβλεψε τον μοντέλο. Σε περίπτωση που θέλει να εντοπίσει την υπολογιζόμενη χρησιμότητα του μοντέλου για κάθε εναλλακτική τότε καλεί την εντολή `altsestimation()` αφού έχει καλέσει την `predict()`. Στην **Εικόνα 4.5.5** φαίνονται η πρόβλεψη των ομάδων που έγινε από την μέθοδο της κλάσης `predict()`, τις υπολογισμένες χρησιμότητες από την `altsestimation()` και τα σύνορα των ομάδων για το παραπάνω παράδειγμα.

```
['?', '?', 'YES', '?', 'NO', 'NO', '?', '?']
[0.55350412 0.59897024 0.7552794 0.67068361 0.41955989 0.31348306
 0.53527811 0.51069551]
```

Εικόνα 4.5.5: Αναπαράσταση αποτελεσμάτων

Ως ένα παράδειγμα εισαγωγής δεδομένων από το excel παρουσιάζονται τα δεδομένα στις Εικόνες 4.5.6 και 4.5.7.

	A	B	C	D
1	Sales	Infrastructures	Management	
2	180	2	3	
3	200	2	2	
4	40	3	3	
5	80	3	3	
6	100	2	1	
7	70	1	2	
8	120	1	1	
9				

Εικόνα 4.5.6: Δομή πολυκριτήριου πίνακα μέσα από το excel

	A	B
1	Classes	
2	YES	
3	YES	
4	?	
5	YES	
6	?	
7	NO	
8	NO	
9		

Εικόνα 4.5.7: Δομή ομάδων μέσα από το excel

Τα υπόλοιπα δεδομένα εισάγονται μέσα από τον κώδικα και ο χρήστης με τις οδηγίες που δόθηκαν παραπάνω εφαρμόζει την μέθοδο UTADIS όπως φαίνεται στην Εικόνα 4.5.7.

```
1 from MultiCrit.UtaMethods import UTA, UTASTAR, UTADIS
2 import pandas as pd
3
4 data = pd.read_excel("utadisdata.xlsx",0)
5 groups = pd.read_excel("utadisdata.xlsx",1)
6 groups = groups.iloc[:,0].tolist()
7
8 bestvalues = [200, 3, 3]
9 worstvalues = [0, 1, 1]
10 intervals = [4, 2, 2]
11 monotonicity = [1, 1, 1]
12 pref_order = ['NO', '?', 'YES']
13 print(data)
14 print(groups)
15 res = UTADIS()
16
17 x = res.fit(data=data,pref_order=pref_order, groups=groups, bestvalues=bestvalues
18           , worstvalues=worstvalues, intervals=intervals, monotonicity=monotonicity)
19
20 print(x)
```

Εικόνα 4.5.8: Χρήση της fit για την δημιουργία του μοντέλου

Και για την εισαγωγή δεδομένων στο σύστημα ο χρήστης θα καλέσει αντίστοιχα τα δεδομένα με την ίδια από ένα αρχείο excel και θα εφαρμόσει την μέθοδο **predict(...)** για να γίνει πρόβλεψη της άγνωστης ομάδας, και της μεθόδου **altsestimation()** για τον υπολογισμό των χρησιμότητων.

```
21 unknowndata = pd.read_excel('utadisdata.xlsx',2)
22
23 y = res.predict(unknowndata)
24 d = res.altsestimation()
25 print(y)
26 print(d)
```

Εικόνα 4.5.9: Πρόβλεψη δεδομένων με άγνωστες ομάδες

Τα αποτελέσματα βρίσκονται στην εικόνα 4.5.5 καθώς είναι τα ίδια δεδομένα.

4.6 Η συνάρτηση UTAPI

Για την εφαρμογή της μεθόδου UTAPI ο χρήστης χρειάζεται να καλέσει από την εργαλειοθήκη την εντολή **UTAPI(data, alternatives, criteria, ranking, monotonicity, bestvalues, worstvalues, intervals, method, epsilon, delta, sigma, utilities)** όπως φαίνεται στην Εικόνα 4.4. Οι αυτοματισμοί και τα φίλτρα που εφαρμόζονται στην UTA εφαρμόζονται και στην UTAPI. Για την αποδοχή των δεδομένων από την συνάρτηση, τα δεδομένα πρέπει να έχουν την εξής μορφή:

data:	Η μεταβλητή αυτή αφορά τα δεδομένα του πολυκριτηρίου πίνακα, τα οποία εισάγονται είτε σε μορφή pandas data frame είτε σε numpy array. Στην περίπτωση που εισαχθούν σε μορφή pandas data frame θα πρέπει να περιέχεται και μία στήλη με ετικέτα “Ranking” ή “Rank” που θα περιγράφει τις κατατάξεις.
alternatives:	Δέχεται σε μορφή λίστας τα ονόματα των εναλλακτικών. Σε περίπτωση που δεν εισαχθεί το σύστημα ονομάζει τις εναλλακτικές ως alternative1, alternative2,...
criteria:	Δέχεται σε μορφή λίστας τα ονόματα των κριτηρίων. Σε περίπτωση που δεν εισαχθεί το σύστημα ονομάζει τα κριτήρια criteria1, criteria2...
ranking	Δέχεται σε μορφή λίστας ή πίνακα numpy την κατάταξη των εναλλακτικών. Σε περίπτωση pandas dataframe δεν χρειάζεται η εισαγωγή αυτής της μεταβλητής.
monotonicity:	Δέχεται σε μορφή λίστας ή πίνακα numpy την μονοτονία των εναλλακτικών τιμών με αποδεκτές τιμές 1 για αύξουσα μονοτονία και 0 για φθίνουσα.
bestvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις καλύτερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την καλύτερη τιμή την τιμή του πολυκριτηρίου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
worstvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις χειρότερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την χειρότερη τιμή την τιμή του πολυκριτηρίου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
intervals:	Δέχεται σε μορφή λίστας ή πίνακα numpy τα διαστήματα στα οποία θα χωριστεί το κάθε κριτήριο. Αν δεν δοθεί τιμή χωρίζει όλα τα κριτήρια σε 3 διαστήματα
method:	Δέχεται σαν είσοδο την μέθοδο με την οποία θα επιλυθεί το γραμμικό πρόβλημα. Οι αποδεκτές τιμές είναι αυτές που δέχεται η linprog της scipy, δηλαδή οι τιμές “simplex” και “interior-point”.
epsilon:	Η μεταβλητή epsilon καθορίζει το αποδεκτό σφάλμα που ο χρήστης είναι διατεθειμένος να αποδεχτεί κατά την διάρκεια της μετά-βελτιστοποίησης . Προκαθορισμένη τιμή είναι 0,005.
delta:	Η μεταβλητή delta ορίζει την ελάχιστη απόσταση μεταξύ της κατάταξης μίας εναλλακτικής με την αμέσως χειρότερη της κατά την δόμηση του γραμμικού προβλήματος. Προκαθορισμένη τιμή είναι 0,05.
sigma:	Η μεταβλητή sigma καθορίζει την ελάχιστη διαφορά που πρέπει να έχουν οι χρησιμότητες ενός κριτηρίου όσο αυξάνεται η τιμή σου στο διακριτό διάστημα. Προκαθορισμένη τιμή είναι 0.
utilities	Δέχεται σαν είσοδο την μέθοδο που θα υπολογιστούν οι αρχικές χρησιμότητες. Σαν επιλογή δέχεται την “UTA” και “UTASTAR”

Αντίστοιχα με την UTA η συνάρτηση UTAPI επιστρέφει ως αποτέλεσμα ένα αντικείμενο. Το αντικείμενο αυτό επιστρέφει ως γνωρίσματα τις διακριτές τιμές που χωρίστηκαν τα κριτήρια, τις χρησιμότητες των κριτηρίων, τα ονόματα των κριτηρίων, τα βάρη p του κάθε κριτηρίου, τα, τις χρησιμότητες των εναλλακτικών, τα ονόματα των εναλλακτικών, τα σφάλματα που προέκυψαν από κάθε εναλλακτική, τις λύσεις της μετά-βελτιστοποίησης και τον δείκτη ASI. Τα γνωρίσματα που χρειάζεται να καλέσει ο χρήστης είναι τα εξής:

.g:	Επιστρέφει σε μορφή λίστας τις διακριτές τιμές στις οποίες χωρίστηκαν τα κριτήρια.
.crit_names:	Επιστρέφει σε μορφή λίστας τα ονόματα των κριτηρίων.
.alternatives:	Επιστρέφει σε μορφή λίστας τα ονόματα των εναλλακτικών.
.crit_utilities:	Επιστρέφει τις χρησιμότητες των κριτηρίων σε μορφή πίνακα numpy.
.alts_utilities	Επιστρέφει τις χρησιμότητες των εναλλακτικών σε μορφή πίνακα numpy.
.all_p	Επιστρέφει σε μορφή numpy array όλα τα βάρη που υπολογίστηκαν κατά την διάρκεια της μετά-βελτιστοποίησης.
.p	Επιστρέφει σε μορφή numpy array τα βάρη των χρησιμοτήτων
.errors:	Επιστρέφει τα σφάλματα υποεκτίμησης και υπερεκτίμησης σε μορφή πίνακα numpy.
.ranking	Επιστρέφει σε μορφή λίστας τις κατατάξεις
.ASI:	Επιστρέφει την τιμή του δείκτη ASI (average stability index)

Χρησιμοποιώντας το παράδειγμα δεδομένων από την UTASTAR καλείται η UTAPI με την επιλογή UTASTAR για τον υπολογισμό των αρχικών χρησιμοτήτων.

```
res = UTAPI(df, bestvalues=bestvalues, worstvalues=worstvalues,  
            monotonicity=monotonicity, intervals=intervals,  
            method='interior-point', epsilon=0.005, delta=0.05,  
            utilities="UTASTAR")  
print(res)
```

Εικόνα 4.6.1: Η συνάρτηση της UTAPI

Και το αποτέλεσμα που προκύπτει είναι το εξής:

```
g: [[30.0, 16.0, 2.0], [40.0, 30.0, 20.0, 10.0], [0.0, 1.0, 2.0, 3.0]]
criteria: ['criterion1', 'criterion2', 'criterion3']
crit_names: ['u1.1', 'u1.2', 'u1.3', 'u2.1', 'u2.2', 'u2.3', 'u2.4', 'u3.1', 'u3.2', 'u3.3', 'u3.4']
crit_utilities: array([0.          , 0.27780838, 0.4733407 , 0.          , 0.0370847 ,
0.24917593, 0.39914993, 0.          , 0.01924711, 0.03167432,
0.12750937])
p: array([0.37469441, 0.32110908, 0.30419652])
all_p: array([[6.66550771e-01, 2.97682549e-02, 3.03680975e-01, 1.26521432e-10,
8.64860560e-11, 4.36316193e-10, 5.74250007e-11, 4.71457029e-11,
9.00000115e-03, 9.42580369e-11, 9.71609714e-11],
[8.92693540e-02, 9.10730674e-01, 4.85217872e-09, 8.92380208e-09,
1.40642037e-08, 1.24100752e-03, 6.87045574e-09, 1.65729831e-09,
1.25257871e-03, 2.39918836e-03, 3.42391755e-09],
[3.68263115e-01, 2.28283039e-02, 6.08908581e-01, 7.41420844e-13,
2.75506710e-13, 2.08796557e-13, 9.00000031e-03, 1.63443570e-13,
1.47208619e-13, 2.74319884e-13, 1.65143518e-13]])
alternatives: ['alternative1', 'alternative2', 'alternative3', 'alternative4', 'alternative5']
alts_utilities: array([0.30615048, 0.25653958, 0.25737077, 0.1564253 , 0.05069614])
errors: array([3.01702165e-09, 4.71698843e-09, 4.13669318e-04, 3.00000241e-03,
5.68202487e-10, 3.41752662e-03, 7.99729484e-04, 1.17374789e-09,
1.85208838e-09, 1.36911603e-03])
tau: 0.9486832980505137
ASI: 0.8842044465709346
```

Εικόνα 4.6.2: Τα αποτελέσματα της UTAII

Χρησιμοποιώντας τις λέξεις κλειδιά που αναφέρθηκαν πιο πάνω ο χρήστης έχει πρόσβαση στην επιθυμητή πληροφορία.

4.7 Η συνάρτηση stochastic_UTA

Για την εφαρμογή της μεθόδου stochastic_UTA ο χρήστης χρειάζεται να καλέσει από την εργαλειοθήκη την εντολή **stochastic_UTA(data, alternatives, criteria, ranking, monotonicity, bestvalues, worstvalues, distributions, metrics, intervals, method, epsilon, delta, sigma)** όπως φαίνεται στην Εικόνα 4.4. Οι αυτοματισμοί και τα φίλτρα που εφαρμόζονται στην UTA εφαρμόζονται και στην stochastic_UTA. Για την αποδοχή των δεδομένων από την συνάρτηση, τα δεδομένα πρέπει να έχουν την εξής μορφή:

data:	Η μεταβλητή αυτή αφορά τα δεδομένα του πολυκριτήριου πίνακα, τα οποία εισάγονται είτε σε μορφή pandas data frame είτε σε numpy array. Στην περίπτωση που εισαχθούν σε μορφή pandas data frame θα πρέπει να περιέχεται και μία στήλη με ετικέτα “Ranking” ή “Rank” που θα περιγράφει τις κατατάξεις.
alternatives:	Δέχεται σε μορφή λίστας τα ονόματα των εναλλακτικών. Σε περίπτωση που δεν εισαχθεί το σύστημα ονομάζει τις εναλλακτικές ως alternative1, alternative2,...
criteria:	Δέχεται σε μορφή λίστας τα ονόματα των κριτηρίων. Σε περίπτωση που δεν εισαχθεί το σύστημα ονομάζει τα κριτήρια criteria1, criteria2...
ranking	Δέχεται σε μορφή λίστας ή πίνακα numpy την κατάταξη των εναλλακτικών. Σε περίπτωση pandas dataframe δεν χρειάζεται η εισαγωγή αυτής της μεταβλητής.
monotonicity:	Δέχεται σε μορφή λίστας ή πίνακα numpy την μονοτονία των εναλλακτικών τιμών με αποδεκτές τιμές 1 για αύξουσα μονοτονία και 0 για φθίνουσα.
bestvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις καλύτερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την καλύτερη τιμή την τιμή του πολυκριτήριου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
worstvalues:	Δέχεται σε μορφή λίστας ή πίνακα numpy τις χειρότερες τιμές που μπορεί να λάβει κάθε κριτήριο. Σε περίπτωση που δεν συμπληρωθεί ο πίνακας η συνάρτηση λαμβάνει σαν την χειρότερη τιμή την τιμή του πολυκριτήριου πίνακα, έχοντας υπόψιν την μονοτονία του κριτηρίου.
distributions	Δέχεται σε μορφή λίστας την κατανομή που ακολουθεί το κάθε κριτήριο, για την κανονική κατανομή χρησιμοποιείται η λέξη κλειδί “normal”, για την ομοιόμορφη κατανομή “uniform” και αν το κριτήριο είναι αιτιοκρατικό η λέξη κλειδί είναι “del”.
metrics	Δέχεται σε μορφή numpy array τα χαρακτηριστικά της κατανομής για την κανονική κατανομή. Ο πίνακας θα πρέπει να περιέχει 2 στήλες, με την πρώτη στήλη να δέχεται την μέση τιμή και την δεύτερη να δέχεται την τυπική απόκλιση. Οι γραμμές αναπαριστούν τα κριτήρια. Σε περίπτωση ομοιόμορφης κατανομής δεν έχει σημασία τα στοιχεία που εισάγονται.
intervals:	Δέχεται σε μορφή λίστας ή πίνακα numpy τα διαστήματα στα οποία θα χωριστεί το κάθε κριτήριο. Αν δεν δοθεί τιμή χωρίζει όλα τα κριτήρια σε 3 διαστήματα
method:	Δέχεται σαν είσοδο την μέθοδο με την οποία θα επιλυθεί το γραμμικό πρόβλημα. Οι αποδεκτές τιμές είναι αυτές που δέχεται η linprog της scipy, δηλαδή οι τιμές “simplex” και “interior-point”.
epsilon:	Η μεταβλητή epsilon καθορίζει το αποδεκτό σφάλμα που ο χρήστης είναι διατεθειμένος να αποδεχτεί κατά την διάρκεια της μετά-βελτιστοποίησης. Προκαθορισμένη τιμή είναι 0,005.
delta:	Η μεταβλητή delta ορίζει την ελάχιστη απόσταση μεταξύ της κατάταξης μίας εναλλακτικής με την αμέσως χειρότερη της κατά την δόμηση του γραμμικού προβλήματος. Προκαθορισμένη τιμή είναι 0,05.

Η συνάρτηση επιστρέφει ως αντικείμενο τις διακριτές τιμές g , τα ονόματα των κριτηρίων, τις χρησιμότητες των κριτηρίων, τις χρησιμότητες σε μορφή w , τις πιθανότητες των τιμών του πολυκριτήριου πίνακα, όλες τις τιμές της μετά-βελτιστοποίησης, τα ονόματα των εναλλακτικών, τις χρησιμότητες των εναλλακτικών, τα σφάλματα, τις κατατάξεις και τον δείκτη ASI.

.g:	Επιστρέφει σε μορφή λίστας τις διακριτές τιμές στις οποίες χωρίστηκαν τα κριτήρια.
.crit_names:	Επιστρέφει σε μορφή λίστας τα ονόματα των κριτηρίων.
.alternatives:	Επιστρέφει σε μορφή λίστας τα ονόματα των εναλλακτικών.
.crit_utilities:	Επιστρέφει τις χρησιμότητες των κριτηρίων σε μορφή πίνακα numpy.
.alts_utilities	Επιστρέφει τις χρησιμότητες των εναλλακτικών σε μορφή πίνακα numpy.
.prob_array	Ο πίνακας με τις τιμές των πιθανοτήτων για κάθε τιμή του πολυκριτήριου πίνακα
.w	Επιστρέφει τις χρησιμότητες w του κάθε κριτηρίου σε μορφή πίνακα numpy.
.w_names	Επιστρέφει σε μορφή λίστας τα ονόματα των εναλλακτικών.
.all_w	Επιστρέφει σε μορφή numpy array όλες τις χρησιμότητες που υπολογίστηκαν κατά την διάρκεια της μετά-βελτιστοποίησης.
.errors:	Επιστρέφει τα σφάλματα υποεκτίμησης και υπερεκτίμησης σε μορφή πίνακα numpy.
.ranking	Επιστρέφει σε μορφή λίστας τις κατατάξεις
.ASI:	Επιστρέφει την τιμή του δείκτη ASI (average stability index)

Ως παράδειγμα θα χρησιμοποιηθεί το παράδειγμα της UTASTAR και θα προστεθούν οι πληροφορίες των κατανομών του κάθε κριτηρίου και των στοιχείων των κατανομών.

```
df = np.array([[3, 10, 1], [4, 20, 2], [2, 20, 0], [6, 40, 0], [30, 30, 3]])
ranking = [1, 2, 2, 3, 4]
alternatives = ['a', "b", "c", 'd', 'f']
criteria = ['1st', '2nd', '3rd']
bestvalues = [2, 10, 3]
worstvalues = [30, 40, 0]
monotonicity = [0, 0, 1]
intervals = [2, 3, 3]
distributions = ['normal', 'uniform', 'normal']
metrics = np.array([[14, 11.83], [0, 0], [1.2, 0.8]])
```

Εικόνα 4.7.1: Παράδειγμα των δεδομένων της stochastic UTA

Για την υλοποίηση της μεθόδου χρησιμοποιείται η συνάρτηση που αναφέρθηκε παραπάνω:

```
res = stochastic_UTA(df, ranking=ranking, alternatives=alternatives, criteria=criteria,
                    bestvalues=bestvalues, worstvalues=worstvalues, monotonicity=monotonicity,
                    distributions=distributions, metrics=metrics, intervals=intervals, method='simplex')
```

Εικόνα 4.7.2: Κάλεσμα συνάρτησης stochastic_UTA

Και το αποτέλεσμα που προκύπτει είναι το εξής:

```
g: [[30.0, 16.0, 2.0], [40.0, 30.0, 20.0, 10.0], [0.0, 1.0, 2.0, 3.0]]
criteria: ['1st', '2nd', '3rd']
crit_names: ['u1.1', 'u1.2', 'u1.3', 'u2.1', 'u2.2', 'u2.3', 'u2.4', 'u3.1', 'u3.2', 'u3.3', 'u3.4']
crit_utilities: array([[0.022, 0.34464564, 0.02362454, 0.02362454, 0.14190181],
                      [0.51345256, 0.51345256, 0.022, 0.32264564, 0.02362454, 0.11827726]])
w_names: ['w1.1', 'w1.2', 'w2.1', 'w2.2', 'w2.3', 'w3.1', 'w3.2', 'w3.3']
w: array([[0.51345256, 0.022, 0.32264564, 0.02362454, 0.11827726]])
prob_array: array([[0.82377288, 1.0, 0.40129367],
                  [0.8010307, 0.66666667, 0.84134475],
                  [0.84479694, 0.66666667, 0.0668072],
                  [0.75055803, 0.0668072],
                  [0.08810872, 0.33333333, 0.98777553]])
all_w: array([[0.911031, 0.04752059, 0.04144841, 0.06366797, 0.066, 0.87033203, 0.5656587, 0.05008429, 0.02942522, 0.35483179]])
alternatives: ['a', 'b', 'c', 'd', 'f']
alts_utilities: array([[0.88172274, 0.5590771, 0.53545256, 0.51345256, 0.14190181],
                      [0.00092884, 0.00166667, 0.0024045]])
errors: array([0.0, 0.0, 0.0, 0.0, 0.0])
ranking: [1, 2, 2, 3, 4]
ASI: 0.8286800508056085
```

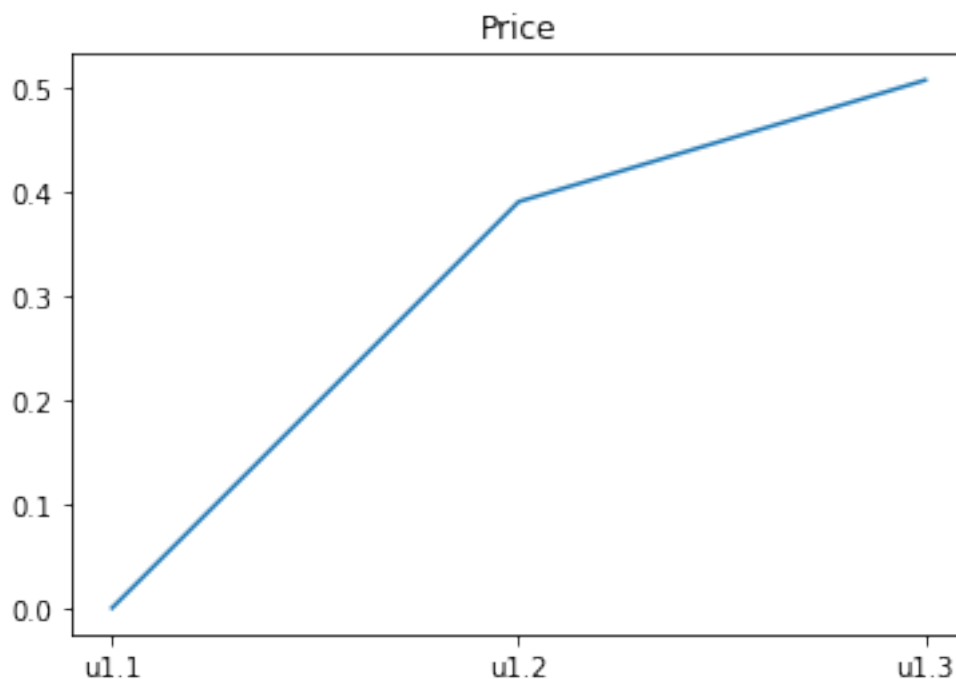
Εικόνα 4.7.3: Αποτελέσματα stochastic_UTA

Χρησιμοποιώντας τις λέξεις κλειδιά που αναφέρθηκαν πιο πάνω ο χρήστης έχει πρόσβαση στην επιθυμητή πληροφορία.

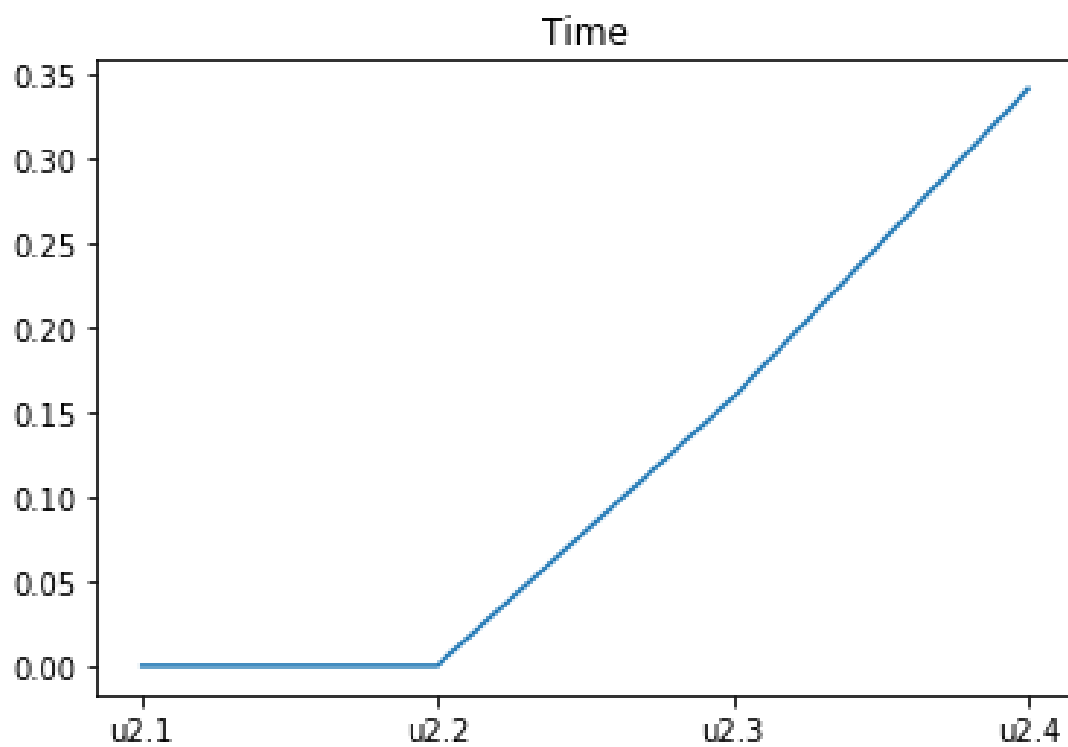
4.8 Γραφήματα

Η εργαλειοθήκη δίνει την δυνατότητα δημιουργίας γραφημάτων με την χρήση της `matplotlib` στο παρασκήνιο. Με την χρήση των αντίστοιχων εντολών ο χρήστης μπορεί να οπτικοποιήσει τα αποτελέσματα είτε των χρησιμότητων των κριτηρίων είτε των χρησιμότητων των εναλλακτικών.

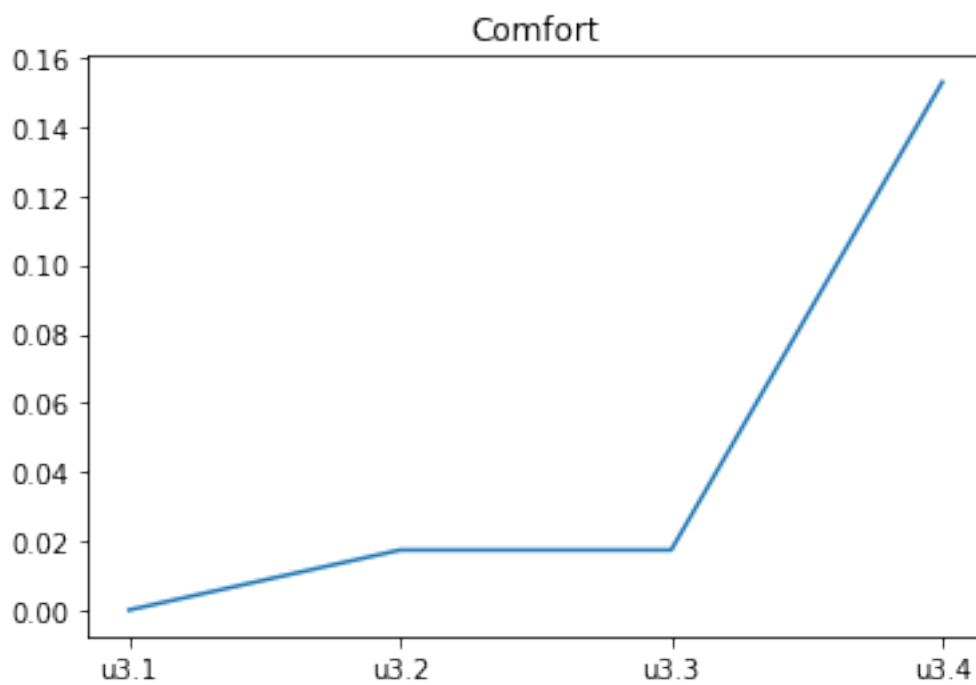
Για την οπτική αποτύπωση των κριτηρίων ο χρήστης χρειάζεται να καλέσει την εντολή `criteria_plot(res)`. Σαν είσοδο εισάγεται το αντικείμενο που έχει δημιουργήσει η μέθοδος πολυκριτήριας ανάλυσης και σαν έξοδο δημιουργούνται τα διαγράμματα των χρησιμότητων του κάθε κριτηρίου. Παρακάτω παρουσιάζονται τα γραφήματα των κριτηρίων του παραδείγματος της UTASTAR.



Γράφημα 4.8.1: Κριτήριο Τιμή

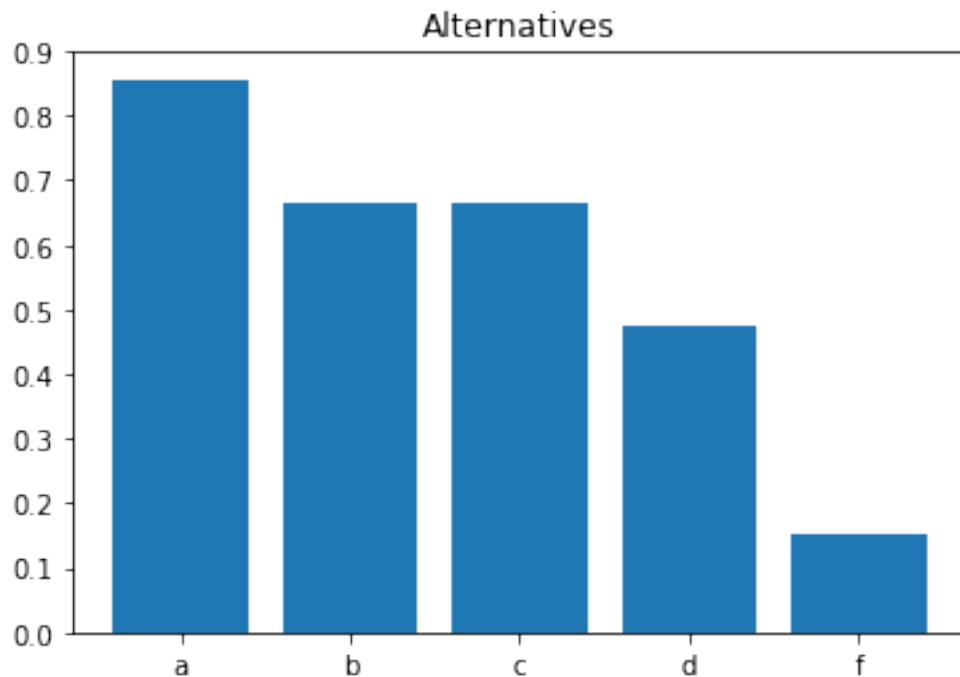


Γράφημα 4.8.2: Κριτήριο Χρονικής διάρκειας



Γράφημα 4.8.3: Κριτήριο Άνεσης

Η εντολή για την δημιουργία γραφήματος των εναλλακτικών της μεθόδου είναι η **alternatives_plot(res)** η οποία δέχεται σαν είσοδο το αντικείμενο που δημιούργησε η πολυκριτήρια μέθοδος και παρουσιάζει τις εναλλακτικές σε μορφή ραβδογράμματος. Παρακάτω παρουσιάζεται σαν παράδειγμα το αποτέλεσμα του παραδείγματος της UTASTAR.



Γράφημα 4.8.4: Αποτελέσματα UTASTAR

4.9 τ του Kendall

Για την αξιολόγηση των αποτελεσμάτων ως προς τις κατατάξεις ο χρήστης μπορεί να χρησιμοποιήσει το τ του Kendall καλώντας την συνάρτηση **ktaumetric(res)**. Το αποτέλεσμα που λαμβάνει είναι μία τιμή στο διάστημα $[-1,1]$.

```
ktaumetric(res)
```

```
0.9999999999999999
```

Εικόνα 4.9.1: ktaumetric

Κεφάλαιο 5: Συμπεράσματα και προοπτικές

Η MultiCrit καταφέρνει να επιλύσει επιτυχώς προβλήματα πολυκριτήριας ανάλυσης δίνοντας την δυνατότητα να ρυθμιστούν αρκετοί παράμετροι καθώς και να παρέχει αποτελέσματα στον χρήστη για αξιοποίηση, παρ' όλα αυτά υπάρχουν αρκετά περιθώρια βελτίωσης. Αρχικά θα ήταν εξαιρετικά πρακτικό να προστεθούν μέτρα απόδοσης μέσα στα αποτελέσματα των μεθόδων, όπως το τ του Kendall δίνοντας άμεσα περισσότερη πληροφορία στον χρήστη. Σημαντική αναβάθμιση θα πρέπει να λάβουν οι συναρτήσεις δημιουργίας γραφημάτων αποκτώντας δομή εργαλείου και όχι μίας απλής συνάρτησης. Κυριότερος στόχος από όλους όμως είναι ο εμπλουτισμός της εργαλειοθήκης με επιπλέον μεθόδους δομημένες σε κατηγορίες ανάλογα με τα χαρακτηριστικά τους, παραδείγματος χάριν η προσθήκη κατηγορίας μεθόδων ELECTREmethods που αντίστοιχα με την UTAmethods θα περιέχει διάφορες παραλλαγές της ELECTRE. Συνοψίζοντας το εργαλείο αυτό μπορεί να εξελιχθεί σε μία πληθώρα κατευθύνσεων αρκεί να υπάρχει καλή θέληση, φαντασία και διάθεση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Cabral, J. B., Luczywo, N. A. and Zanazzi, J. . (2016) ‘Scikit-Criteria: colección de métodos de análisis multi-criterio integrado al stack científico de Python’, *Procedimientos de XIV Simposio Argentino de Investigación Operativa (SIO 2016)-JAIIO 45 (Tres de Febrero, 2016).*, pp. 59–66.
- Corrente, S. *et al.* (2013) *Robust ordinal regression in preference learning and ranking*, *Machine Learning*. doi: 10.1007/s10994-013-5365-4.
- Greco, S. *et al.* (2010) ‘Robust ordinal regression’, *International Series in Operations Research and Management Science*, 142, pp. 241–283. doi: 10.1007/978-1-4419-5904-1_9.
- Matsatsinis, N. F., Grigoroudis, E. and Siskos, E. (2018) ‘Disaggregation approach to value elicitation’, *International Series in Operations Research and Management Science*, 261, pp. 313–348. doi: 10.1007/978-3-319-65052-4_13.
- Meyer, P. *et al.* (2021) *Package ‘MCDA’*. doi: 10.1007/s40070-017-0064-1>.Imports.
- Siskos, Y., Grigoroudis, E. and Matsatsinis, N. F. (2016) *UTA methods*, *International Series in Operations Research and Management Science*. doi: 10.1007/978-1-4939-3094-4_9.
- Ζυγομήτρος, Α. and Σίσκος, Ι. (2007) ‘Άσκηση : « Πολυκριτήρια Μέθοδος UTADIS » Υπεύθυνος Καθηγητής : Πανεπιστήμιο Πειραιά - Τμήμα Πληροφορικής’, pp. 2006–2007.
- Ματσατσίνης, Ν. (2010a) ‘4.5.6.4 Η μέθοδος UTADIS’, in *ΣΥΣΤΗΜΑΤΑ ΥΠΟΣΤΗΡΙΞΗΣ ΑΠΟΦΑΣΕΩΝ*, pp. 213–215.
- Ματσατσίνης, Ν. (2010b) *ΣΥΣΤΗΜΑΤΑ ΥΠΟΣΤΗΡΙΞΗΣ ΑΠΟΦΑΣΕΩΝ*. 1st edn. Chania: ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ.
- Σίσκος, Ι. (2000) *ΓΡΑΜΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ*. 2nd edn. Athens: ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ.