

ΣΥΣΤΗΜΑΤΑ ΛΗΨΗΣ ΑΠΟΦΑΣΕΩΝ

Εργασία Προσομοίωσης Συστημάτων Λήψης Αποφάσεων σε Octave

Αρβι Χότζα it22121

Γεώργιος Γλαρός it22022

A Μέρος

α) Ορισμός του προβλήματος

Μια ομάδα ποδοσφαίρου πρέπει να αποφασίσει ποιον παίκτη να αποκτήσει κατά την επόμενη μεταγραφική περίοδο. Το πρόβλημα λήψης απόφασης αφορά την επιλογή μεταξύ διαφόρων διαθέσιμων παικτών που ανταποκρίνονται στις ανάγκες της ομάδας.

β) Κριτήρια και παράγοντες

Τα κριτήρια για την αξιολόγηση των παικτών θα είναι:

- Απόδοση σε προηγούμενες σεζόν (Performance in past seasons)
- Δεξιότητες και ικανότητες (Skills and abilities)
- Ηλικία και φυσική κατάσταση (Age and physical condition)
- Κόστος μεταγραφής (Transfer cost)

Η μέθοδος που επιλέξαμε είναι η Fuzzy AHP (Fuzzy Analytical Hierarchy Process), διότι είναι ιδανική για την επίλυση του προβλήματος της επιλογής ποδοσφαιριστή λόγω της ικανότητάς της να αντιμετωπίζει την αβεβαιότητα και την ασάφεια στις αξιολογήσεις, κάτι που είναι κρίσιμο όταν οι αποφάσεις βασίζονται σε υποκειμενικά στοιχεία όπως οι απόψεις των προπονητών και των σκάουτερς. Η Fuzzy AHP παρέχει ένα ισχυρό πλαίσιο για την ιεράρχηση κριτηρίων και υποκριτηρίων, επιτρέποντας την ακριβή στάθμιση παραγόντων όπως η απόδοση, οι δεξιότητες, η ηλικία και το κόστος μεταγραφής. Επιπλέον, μπορεί να συνδυάσει αποτελεσματικά ποσοτικά δεδομένα, όπως τα στατιστικά απόδοσης, με ποιοτικές αξιολογήσεις, χρησιμοποιώντας ασαφείς αριθμούς για την αντιστοίχιση ποιοτικών δεδομένων σε ποσοτικές τιμές. Η συστηματική και δομημένη διαδικασία της Fuzzy AHP επιτρέπει τη συλλογή, σύνθεση και σύγκριση των απόψεων πολλών εμπειρογνομόνων με έναν τυποποιημένο τρόπο, καθιστώντας την την πιο κατάλληλη επιλογή για την αξιολόγηση των εναλλακτικών παικτών και τη λήψη τεκμηριωμένων αποφάσεων σε αυτό το σενάριο.

γ) Εναλλακτικές

Οι διαθέσιμες εναλλακτικές είναι:

- Παίκτης Α: Cristiano Ronaldo
- Παίκτης Β: Lionel Messi
- Παίκτης Γ: Kylian Mbappe

Επίπεδο διοίκησης:

Το πρόβλημα αυτό αφορά τη διοίκηση της ομάδας και τους προπονητές που είναι υπεύθυνοι για την απόδοση και το τμήμα σκάουτινγκ της ομάδας.

Βασικοί ρόλοι

- Διοικητικό Συμβούλιο: Λαμβάνει την τελική απόφαση.
- Προπονητής: Παρέχει τεχνική αξιολόγηση των παικτών.
- Οικονομικός Διευθυντής: Αξιολογεί το οικονομικό κόστος της μεταγραφής.
- Σκάουτερς: Συλλέγουν και αναλύουν δεδομένα απόδοσης των παικτών.

B Μέρος

Ερώτημα 1

Για να προσομοιώσουμε το σενάριο στο Octave με τη συμμετοχή 15 ειδικών, έχουμε ακολουθήσει τα παρακάτω βήματα:

1. Ορισμός Κριτηρίων και Εναλλακτικών: Επιλέξαμε τα κριτήρια και τις εναλλακτικές όπως περιγράφεται στο Μέρος A:

1. Κριτήρια: Απόδοση σε προηγούμενες σεζόν, Δεξιότητες και ικανότητες, Ηλικία και φυσική κατάσταση, Κόστος μεταγραφής.
2. Εναλλακτικές: Cristiano Ronaldo, Lionel Messi, Kylian Mbappe (βάλαμε πραγματικούς παίκτες για να είναι πιο ενδιαφέρον η εργασία).

2. Δημιουργία Πινάκων Σύγκρισης: Δημιουργήσαμε 15 πίνακες σύγκρισης από τους ειδικούς. Για κάθε πίνακα, διασφάλισαμε τη συνέπεια χρησιμοποιώντας τον δείκτη CR (Consistency Ratio). Για να βρούμε τον δείκτη CR υλοποιήσαμε τις παρακάτω εξισώσεις σε octave:

$$\lambda_{\max} = \frac{\sum (a_{ij} \cdot w_j)}{w_i} \quad \text{όπου,}$$

w είναι τα κανονικοποιημένα βάρη

λ_{\max} είναι το μέγιστο eigenvalue του matrix A

a_{ij} είναι η αριθμητική σύγκριση μεταξύ των τιμών i και j.

Έπειτα υπολογίσαμε το CI

$$CI = \frac{\lambda_{\max} - n}{n - 1} \quad \text{όπου,}$$

n είναι ο αριθμός των κριτηρίων

Τέλος,

$$CR = \frac{CI}{RI} \quad \text{όπου,}$$

RI είναι ο τυχαίος δείκτης για το συγκεκριμένο μέγεθος πίνακα.

RI = [0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45]

Το βρήκαμε από το παρακάτω link

<https://hal.science/hal-02557320/document#:~:text=In%20the%20next%20step%2C%20in.b%20extracted%20from%20Table%203.>

Παράδειγμα Pair-wise comparison matrix

	Απόδοση	Δεξιότητες	Ηλικία	Κόστος
Απόδοση	1	5	4	7
Δεξιότητες	1/5	1	1/2	3
Ηλικία	1/4	2	1	3
Κόστος	1/7	1/3	1/3	1

Κώδικας

```
% Function to check consistency and return a consistent matrix
function consistent_matrix = ensure_consistency(matrix, RI)
    n = size(matrix, 1);
    [lambda_max, CI] = calculate_consistency(matrix);
    CR = CI / RI(n);
    while CR > 0.1
        matrix = randi([1, 9], n, n); % Replace with new random matrix
        for i = 1:n
            for j = 1:n
                if i == j
                    matrix(i, j) = 1;
                elseif i < j
                    matrix(i, j) = randi([1, 9]);
                    matrix(j, i) = 1 / matrix(i, j);
                end
            end
        end
        [lambda_max, CI] = calculate_consistency(matrix);
        CR = CI / RI(n);
    end
    consistent_matrix = matrix;
end
```

3. Υπολογισμός Μέσου Όρου Πινάκων Σύγκρισης: Υπολογίσαμε τον μέσο όρο των πινάκων σύγκρισης από όλους τους ειδικούς:

Κώδικας

```
% Calculate the average comparison matrix for criteria
average_comparison = zeros(num_criteria, num_criteria);
for k = 1:num_experts
    average_comparison += expert_comparisons{k};
end
average_comparison /= num_experts;
```

4. Fuzzification: Εφαρμόσαμε τη fuzzification των τιμών χρησιμοποιώντας την triangular membership function:

Κώδικας

```
% Fuzzification with triangular membership function
function fuzzy_value = fuzzify(value)
    if value == 1
        fuzzy_value = [1, 1, 1];
    elseif value == 9
        fuzzy_value = [9, 9, 9];
    elseif value < 1
        x = 1 / value;
        fuzzy_value = [1 / (x + 1), value, 1 / (x - 1)];
    else
        fuzzy_value = [max((value - 1), 0), value, value + 1];
    end
end
```

5. Υπολογισμός Fuzzy Geometric Means και Fuzzy Weights: Υπολογίσαμε τους fuzzy geometric means για κάθε κριτήριο και στη συνέχεια τα fuzzy weights:

Χρησιμοποιήσαμε αυτή την εξίσωση για να βρούμε το Fuzzy Geometric Means

$$r_i = \left(\prod_{j=1}^n l_{ij} \right)^{\frac{1}{n}}, \left(\prod_{j=1}^n m_{ij} \right)^{\frac{1}{n}}, \left(\prod_{j=1}^n u_{ij} \right)^{\frac{1}{n}} \quad \text{όπου,}$$

l_{ij} οι κάτω τιμές των τριγωνικών αριθμών της σειράς i

m_{ij} οι μεσαίες τιμές των τριγωνικών αριθμών της σειράς i

u_{ij} οι άνω τιμές των τριγωνικών αριθμών της σειράς i

n το πλήθος των κριτηρίων

Κώδικας

```
% Function to calculate fuzzy geometric mean for a row
function r_i = fuzzy_geometric_mean(row)
    n = length(row);
    l_values = zeros(1, n);
    m_values = zeros(1, n);
    u_values = zeros(1, n);

    for i = 1:n
        l_values(i) = row{i}(1);
        m_values(i) = row{i}(2);
        u_values(i) = row{i}(3);
    end

    l_mean = prod(l_values) ^ (1 / n);
    m_mean = prod(m_values) ^ (1 / n);
    u_mean = prod(u_values) ^ (1 / n);

    r_i = [l_mean, m_mean, u_mean];
end
```

Για να βρούμε τα fuzzy weights χρησιμοποιήσαμε την παρακάτω εξίσωση

$$w_i = \left(\frac{r_i^l}{\sum_{j=1}^n r_j^l}, \frac{r_i^m}{\sum_{j=1}^n r_j^m}, \frac{r_i^u}{\sum_{j=1}^n r_j^u} \right) \text{ όπου,}$$

r_i^l η κάτω τιμή του Fuzzy Geometric Mean για το κριτήριο i

r_i^m η μεσαία τιμή του Fuzzy Geometric Mean για το κριτήριο i

r_i^u η άνω τιμή του Fuzzy Geometric Mean για το κριτήριο i

n το πλήθος των κριτηρίων

Κώδικας

```
% Calculate fuzzy geometric mean for each row
sum_l_mean = 0;
sum_m_mean = 0;
sum_u_mean = 0;
fuzzy_geometric_means = cell(size(fuzzy_matrix, 1), 1);

for i = 1:size(fuzzy_matrix, 1)
    fuzzy_geometric_means{i} = fuzzy_geometric_mean(fuzzy_matrix(i, :));
    sum_l_mean += fuzzy_geometric_means{i}(1);
    sum_m_mean += fuzzy_geometric_means{i}(2);
    sum_u_mean += fuzzy_geometric_means{i}(3);
end

% Display the fuzzy geometric means
disp('Fuzzy Geometric Means:');
disp(fuzzy_geometric_means);

% Calculate fuzzy weights for each criterion
fuzzy_weights = cell(size(fuzzy_geometric_means));
for i = 1:size(fuzzy_geometric_means, 1)
    fuzzy_weights{i} = [
        fuzzy_geometric_means{i}(1) / sum_l_mean,
        fuzzy_geometric_means{i}(2) / sum_m_mean,
        fuzzy_geometric_means{i}(3) / sum_u_mean
    ];
end
```

6. Defuzzification και Κανονικοποίηση: Υπολογίσαμε τα defuzzified weights και τα κανονικοποιήσαμε:

Για το Defuzzification κάναμε το εξής:

$$d_i = \frac{l_i + m_i + u_i}{3} \text{ όπου,}$$

l_i η κάτω τιμή του τριγωνικού αριθμού για το κριτήριο i

m_i η μεσαία τιμή του τριγωνικού αριθμού για το κριτήριο i

u_i η άνω τιμή του τριγωνικού αριθμού για το κριτήριο i

Και για την κανονικοποίηση των βαρών κάναμε το εξής:

$$w_i^{\text{normalized}} = \frac{w_i}{\sum_{j=1}^n w_j} \text{ όπου,}$$

$w_i^{\text{normalized}}$ το κανονικοποιημένο βάρος του κριτηρίου i

n το πλήθος των κριτηρίων

w_i το βάρος του κριτηρίου i

Κώδικας

```
% Function to defuzzify fuzzy weights using Center of Gravity method
function crisp_value = defuzzify(fuzzy_value)
    crisp_value = (fuzzy_value(1) + fuzzy_value(2) + fuzzy_value(3)) / 3;
end

% Calculate defuzzified weights
defuzzified_weights = zeros(size(fuzzy_weights));
for i = 1:length(fuzzy_weights)
    defuzzified_weights(i) = defuzzify(fuzzy_weights{i});
end

disp('Defuzzified Weights:');
disp(defuzzified_weights);

% Function to normalize the defuzzified weights
function normalized_weights = normalize_weights(weights)
    total = sum(weights); % Calculate the sum of all weights
    normalized_weights = weights / total; % Divide each weight by the total sum
end

% Normalize the defuzzified weights
normalized_weights = normalize_weights(defuzzified_weights);

disp('Normalized Weights:');
disp(normalized_weights);

% Ensure normalized_weights is a column vector
normalized_weights = normalized_weights(:);
```

7. Υπολογισμός Βαθμολογιών Παικτών και Επιλογή Καλύτερου Παίκτη: Ορίσαμε τις βαθμολογίες των παικτών για κάθε κριτήριο και υπολογίσαμε τους τελικούς βαθμούς για κάθε παίκτη, τις συγκρίναμε μεταξύ τους και εμφανίσαμε την καλύτερη επιλογή. Επίσης, δίνουμε την δυνατότητα ο χρήστης να εισάγει τις δικές του βαθμολογίες για τον κάθε παίκτη με ένα if statement.

Ερώτημα 2

Σε περίπτωση που κάποιος ειδικός δεν έχει δώσει τιμή σε κάποιο στοιχείο ενός ή περισσότερων πινάκων συγκρίσεων ανά ζεύγη, εφαρμόσαμε την εξής προσέγγιση:

Αντιμετώπιση Ελλειπόντων Τιμών: Όταν ένας ειδικός δεν δίνει τιμή για κάποιο στοιχείο, θέτουμε την τιμή σε **NaN**. Στη συνέχεια, υπολογίζουμε τη μέση τιμή των αντίστοιχων στοιχείων από τους άλλους ειδικούς και χρησιμοποιούμε αυτή τη μέση τιμή για να αντικαταστήσουμε την ελλείπουσα τιμή.

Κώδικας

```
% Handle missing values by taking the average from other experts
for i = 1:num_criteria
    for j = 1:num_criteria
        if i ~= j
            values = [];
            for k = 1:num_experts
                if ~isnan(expert_comparisons{k}(i, j))
                    values = [values, expert_comparisons{k}(i, j)];
                end
            end
            avg_value = mean(values);
            for k = 1:num_experts
                if isnan(expert_comparisons{k}(i, j))
                    expert_comparisons{k}(i, j) = avg_value;
                    expert_comparisons{k}(j, i) = 1 / avg_value;
                end
            end
        end
    end
end
end
```

Γ Μέρος

Στο μέρος Γ της εργασίας, πραγματοποιήσαμε ανάλυση ευαισθησίας με προσομοίωση Monte Carlo για το παράδειγμα υλοποίησης του μέρους Β, χρησιμοποιώντας την μεθοδολογία Fuzzy AHP. Η ανάλυση ευαισθησίας είχε στόχο να εξετάσει την σταθερότητα των αποτελεσμάτων σε σχέση με τις μεταβολές των κριτηρίων βαρύτητας. Η προσομοίωση πραγματοποιήθηκε για $N=10^4$ επαναλήψεις για κάθε μία από τις περιπτώσεις για διάφορα perturbation strengths s από 0.2 έως 0.6, με βήμα 0.1.

Μέθοδος Υλοποίησης

1. Φόρτωση Αρχικών Αποτελεσμάτων:
 - Τα αρχικά αποτελέσματα φορτώθηκαν από το αρχείο fuzzy_AHP_results.mat, το οποίο περιείχε τις αρχικές προτεραιότητες και βαθμολογίες των παικτών.
2. Καθορισμός των Perturbation Strengths:
 - Ορίσαμε τις τιμές perturbation strengths s σε 0.2, 0.3, 0.4, 0.5, 0.6.

3. Υπολογισμός Αρχικών Κατατάξεων:

- Υπολογίζουμε τις αρχικές κατατάξεις των παικτών βάσει των αρχικών βαθμολογιών τους.

4. Συνάρτηση Διαταραχής Βαρύτητων (Perturbation of Weights):

- Δημιουργούμε μια συνάρτηση που διαταράσσει τις βαρύτητες των κριτηρίων προσθέτοντας τυχαία κανονική διαταραχή σε κάθε βαρύτητα. Η διαταραχή ακολουθεί κανονική κατανομή με μέσο όρο την αρχική βαρύτητα και απόκλιση s . Οι νέες βαρύτητες κανονικοποιούνται ώστε να διασφαλιστεί ότι το άθροισμα είναι 1.

Κώδικας

```
% Function to perturb the weights
function perturbed_weights = perturb_weights(weights, s)
    perturbed_weights = weights + s * (2 * rand(size(weights)) - 1);
    perturbed_weights = max(0, perturbed_weights); % Ensure weights are non-negative
    perturbed_weights = perturbed_weights / sum(perturbed_weights); % Normalize weights
end
```

5. Monte Carlo Προσομοίωση:

- Για κάθε τιμή s , πραγματοποιούμε 10^4 επαναλήψεις. Σε κάθε επανάληψη:
 - Διαταράσσουμε τις βαρύτητες των κριτηρίων χρησιμοποιώντας τη συνάρτηση `perturb_weights`.
 - Υπολογίζουμε τις νέες βαθμολογίες των παικτών βάσει των διαταραγμένων βαρύτητων.
 - Υπολογίζουμε τις νέες κατατάξεις των παικτών και συγκρίνουμε με τις αρχικές.
 - Καταγράφουμε αν υπάρχει αναστροφή κατάταξης.

Κώδικας

```
% Monte Carlo simulation
for idx = 1:length(s_values)
    s = s_values(idx);
    rank_reversals = 0;

    for iter = 1:N
        % Perturb the weights
        perturbed_weights = perturb_weights(normalized_weights, s);

        % Calculate new weighted scores
        perturbed_scores = player_scores * perturbed_weights;

        % Determine new ranks
        [~, new_ranks] = sort(perturbed_scores, 'descend');

        % Check for rank reversal
        if ~isequal(new_ranks, initial_ranks)
            rank_reversals += 1;
        end
    end

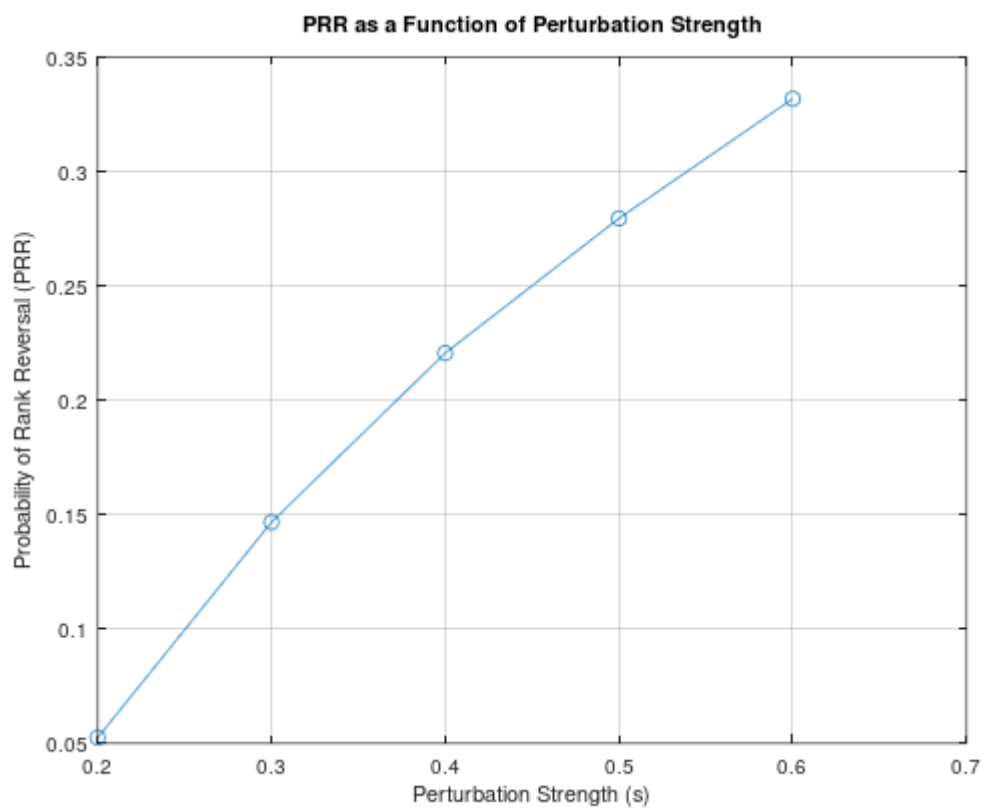
    % Calculate PRR for current s
    PRR(idx) = rank_reversals / N;

    % Store final ranks for the current s
    final_ranks(:, idx) = new_ranks;
end
```


6. Παρουσίαση αποτελεσμάτων:

- Παρουσιάζουμε τις αρχικές και τελικές κατατάξεις των παικτών για κάθε τιμή sss.
- Υπολογίζουμε και παρουσιάζουμε την πιθανότητα αναστροφής κατάταξης (PRR) για κάθε sss.
- Δημιουργούμε γράφημα που δείχνει το PRR σε σχέση με το sss.

Αποτελέσματα



Initial Ranks:

2
1
3

Final Ranks for each s:

s = 0.2:

2
1
3

s = 0.3:

2
1
3

s = 0.4:

2

1

3

s = 0.5:

1

2

3

s = 0.6:

1

2

3

Probability of Rank Reversal (PRR) for each s:

s = 0.2: PRR = 0.0525

s = 0.3: PRR = 0.1467

s = 0.4: PRR = 0.2206

s = 0.5: PRR = 0.2795

s = 0.6: PRR = 0.3318

Σημείωση: 1 = Cristiano Ronaldo, 2 = Lionel Messi, 3 = Kylian Mbappe

Σημείωση: Για αυτά τα αποτελέσματα η είσοδος είναι ο πίνακας που είναι "hardcoded".

Παρατηρήσεις

Αύξηση του PRR με την Αύξηση του s:

Όπως φαίνεται στο γράφημα, η πιθανότητα αναστροφής κατάταξης (PRR) αυξάνεται σχεδόν γραμμικά με την αύξηση του s. Αυτό είναι αναμενόμενο, καθώς μεγαλύτερες τιμές s προκαλούν μεγαλύτερες διαταραχές στις βαρύτητες των κριτηρίων, οδηγώντας σε υψηλότερη πιθανότητα αλλαγής στις κατατάξεις των παικτών.

Για s=0.5, το PRR είναι 27.95%, και για s=0.6 αυξάνεται σε 33.18%. Αυτό δείχνει μια σημαντική αύξηση της ευαισθησίας στις κατατάξεις για μεγαλύτερες διαταραχές.

Ευαισθησία των Κατατάξεων:

Για τις μεγαλύτερες τιμές s=0.5 και s=0.6, παρατηρείται αναστροφή στην κατάταξη των δύο πρώτων παικτών. Ο Cristiano Ronaldo ανεβαίνει στην πρώτη θέση, ενώ ο Lionel Messi κατεβαίνει στη δεύτερη.

Αυτή η αναστροφή δείχνει ότι για μεγάλες διαταραχές στις βαρύτητες των κριτηρίων, οι κατατάξεις γίνονται πιο ασταθείς και μπορεί να αλλάξουν σημαντικά.

Σταθερότητα Αρχικών Κατατάξεων:

Για τις τιμές s=0.2, s=0.3 και s=0.4, οι κατατάξεις παραμένουν αμετάβλητες. Αυτό υποδηλώνει ότι οι αρχικές κατατάξεις είναι σχετικά σταθερές και δεν επηρεάζονται σημαντικά από μικρές διαταραχές στις βαρύτητες των κριτηρίων.

Η πιθανότητα αναστροφής κατάταξης (PRR) είναι αρκετά χαμηλή για αυτές τις τιμές s, με 5.25% για s=0.2, 14.67% για s=0.3 και 22.06% για s=0.4. Αυτό δείχνει ότι αν και υπάρχει κάποια ευαισθησία, οι αρχικές κατατάξεις παραμένουν κυρίως αμετάβλητες.

Συμπεράσματα

Η ανάλυση ευαισθησίας δείχνει ότι οι κατατάξεις που προκύπτουν από τη μέθοδο Fuzzy AHP είναι γενικά σταθερές για μικρές διαταραχές στις βαρύτητες των κριτηρίων. Ωστόσο, για μεγαλύτερες διαταραχές, οι κατατάξεις γίνονται πιο ασταθείς, με σημαντική αύξηση της πιθανότητας αναστροφής κατάταξης. Αυτό υποδηλώνει ότι η μεθοδολογία είναι ευαίσθητη σε μεγάλες αλλαγές στις βαρύτητες και ενδέχεται να απαιτεί προσεκτική διαχείριση και εκτίμηση των κριτηρίων για την αποφυγή σημαντικών μεταβολών στις κατατάξεις των εναλλακτικών.

Σχολιασμός εργασίας

Συνεργασία μελών:

Για την συνεργασίας μεταξύ μας χρησιμοποιήσαμε το εργαλείο git

GitHub repository: <https://github.com/ArviHoxha7/DecisionMakingSystem>

Χρήσιμα links

Τα παρακάτω 4 youtube video μας βοήθησαν σημαντικά στην διεκπεραίωση της εργασίας.

AHP: <https://www.youtube.com/watch?v=J4T70o8gilk&t=509s>

Consistency in AHP: <https://www.youtube.com/watch?v=VTZft7SpV0g&t=63s>

FAHP: <https://www.youtube.com/watch?v=5k3Wz1AfVWs>

Sensitivity Analysis: <https://www.youtube.com/watch?v=9Q7cdky52vA&t=1s>