Μέρος Α: Θεωρητικό Σενάριο Λήψης Απόφασης

(α) Ορισμός προβλήματος:

Η εταιρεία μας καλείται να επιλέξει την καλύτερη πλατφόρμα cloud για να αναβαθμίσει τις IT υποδομές της. Οι διαθέσιμες εναλλακτικές περιλαμβάνουν τις πλατφόρμες AWS, Microsoft Azure και Google Cloud. Η απόφαση αυτή θα επηρεάσει την απόδοση και τη μακροχρόνια βιωσιμότητα των υπηρεσιών μας.

(β) Ορισμός κριτηρίων:

1. **Κόστος**: Συνολικό κόστος υλοποίησης και συντήρησης της πλατφόρμας.
2. **Ασφάλεια**: Επιπέδου προστασίας δεδομένων και συμμόρφωση με κανονισμούς (GDPR κ.λπ.).
3. **Απόδοση**: Ταχύτητα επεξεργασίας και αξιοπιστία της πλατφόρμας.
4. **Υποστήριξη και επεκτασιμότητα**: Τεχνική υποστήριξη και δυνατότητες μελλοντικής ανάπτυξης.

(γ) Ορισμός παραγόντων:

1. **Τεχνικές απαιτήσεις**: Συμβατότητα με τις τρέχουσες υποδομές.
2. **Ανάγκες κλιμάκωσης**: Δυνατότητα μελλοντικής επέκτασης.
3. **Ευκολία μετάβασης**: Κίνδυνοι και κόστη μετάβασης στο cloud.

(δ) Εναλλακτικές λύσεις:

1. **Amazon Web Services (AWS)**.
2. **Microsoft Azure**.
3. **Google Cloud Platform**.

Μέρος Β: Υλοποίηση AHP στο Octave

Περιγραφή Διαδικασίας

Στη διαδικασία συμμετέχουν 15 ειδικοί που θα αξιολογήσουν τις εναλλακτικές με βάση τα κριτήρια που ορίστηκαν στο Μέρος Α. Η μέθοδος AHP περιλαμβάνει τη χρήση πινάκων συγκρίσεων για κάθε κριτήριο, ενώ θα ελέγξουμε τη συνέπεια μέσω του δείκτη CR (Consistency Ratio).

Κώδικας Octave:

```
% AHP Implementation
% Ορισμός των συγκρίσεων για τα κριτήρια
% Ορισμός νέας μήτρας συγκρίσεων κριτηρίων (βελτιωμένη)
criteria_matrix = [
    1    2    0.5  0.33;
    0.5  1    0.25 0.5;
    2    4    1    3;
```

```
    3    2    1/3  1
];

% Υπολογισμός ιδιοτιμών και ιδιοδιανυσμάτων (Eigenvalues &
Eigenvectors)
[V, D] = eig(criteria_matrix);
lambda_max = max(diag(D));  % Μέγιστη ιδιοτιμή

% Υπολογισμός Consistency Index (CI) και Consistency Ratio
(CR)
n = size(criteria_matrix, 1);  % Μέγεθος της μήτρας (αριθμός
κριτηρίων)
consistency_index = (lambda_max - n) / (n - 1);

% Ορισμός Random Index (RI) ανάλογα με το μέγεθος της μήτρας
(n)
random_index_values = [0, 0, 0.58, 0.9, 1.12, 1.24, 1.32,
1.41, 1.45]; % Τυπικές τιμές RI
random_index = random_index_values(n);

% Υπολογισμός του Consistency Ratio (CR)
consistency_ratio = consistency_index / random_index;

% Έλεγχος συνέπειας (CR < 0.1)
if consistency_ratio > 0.1
    disp('Η μήτρα δεν είναι συνεπής. Επαναλάβετε την
αξιολόγηση.');
else
    disp('Η μήτρα είναι συνεπής.');
end

% Κανονικοποίηση των ιδιοδιανυσμάτων για την εύρεση των
προτεραιοτήτων
priority_vector = V(:, find(diag(D) == lambda_max));
priority_vector = priority_vector / sum(priority_vector);  %
Κανονικοποίηση

% Εμφάνιση των τελικών προτεραιοτήτων (βαρών) των
εναλλακτικών λύσεων
disp('Προτεραιότητες των εναλλακτικών:');
disp(priority_vector);

% Τέλος της ανάλυσης
```

Επεξήγηση του Κώδικα

1. Σκοπός του Κώδικα

Στο **Μέρος Β**, υλοποιείται η **μέθοδος AHP (Analytic Hierarchy Process)**, η οποία χρησιμοποιείται για να υπολογιστούν οι προτεραιότητες των εναλλακτικών με βάση τα κριτήρια απόφασης που έχουν οριστεί. Ο κώδικας περιλαμβάνει τη διαδικασία υπολογισμού της **συνέπειας της μήτρας σύγκρισης** κριτηρίων, καθώς και τον υπολογισμό των **προτεραιοτήτων** των εναλλακτικών με βάση τα αποτελέσματα της μεθόδου AHP.

2. Βασικά Στοιχεία του Κώδικα

Ορισμός της μήτρας σύγκρισης κριτηρίων

```
octave

criteria_matrix = [
    1    2    0.5  0.33;
    0.5  1    0.25 0.5;
    2    4    1    3;
    3    2    1/3  1
];
```

Η μήτρα αυτή αντιπροσωπεύει τις **συγκρίσεις ανά ζεύγη** μεταξύ των κριτηρίων που έχουν οριστεί. Κάθε στοιχείο της μήτρας αναπαριστά τη σύγκριση του ενός κριτηρίου με το άλλο. Για παράδειγμα, η τιμή στη θέση (1, 2) της μήτρας είναι 2, που σημαίνει ότι το πρώτο κριτήριο θεωρείται 2 φορές πιο σημαντικό από το δεύτερο.

Υπολογισμός Ιδιοτιμών και Ιδιοδιανυσμάτων

```
octave

[V, D] = eig(criteria_matrix);
lambda_max = max(diag(D));
```

Χρησιμοποιούμε τον υπολογισμό των **ιδιοτιμών** και των **ιδιοδιανυσμάτων** της μήτρας για να υπολογίσουμε τη μέγιστη ιδιοτιμή, η οποία χρησιμοποιείται για τον υπολογισμό του **Consistency Index** (CI). Η μέγιστη ιδιοτιμή είναι κρίσιμη για τον έλεγχο της συνέπειας της μήτρας.

Υπολογισμός Συντελεστών Συνεπειας (Consistency Index και Consistency Ratio)

```
octave

consistency_index = (lambda_max - n) / (n - 1);
random_index_values = [0, 0, 0.58, 0.9, 1.12, 1.24, 1.32,
1.41, 1.45];
random_index = random_index_values(n);
consistency_ratio = consistency_index / random_index;
```

- Το **Consistency Index (CI)** μετρά την απόκλιση της μήτρας από την πλήρη συνέπεια.

- Ο **Consistency Ratio (CR)** υπολογίζεται στη συνέχεια, συγκρίνοντας το CI με το **Random Index (RI)**, το οποίο είναι μια προκαθορισμένη τιμή ανάλογα με το μέγεθος της μήτρας.
- Αν το CR είναι μικρότερο από 0.1, η μήτρα θεωρείται συνεπής. Αν είναι μεγαλύτερο από 0.1, η μήτρα δεν θεωρείται συνεπής και θα πρέπει να επαναληφθεί η διαδικασία σύγκρισης των κριτηρίων.

Έλεγχος της Συνέπειας

```
octave

if consistency_ratio > 0.1
    disp('Η μήτρα δεν είναι συνεπής. Επαναλάβετε την αξιολόγηση.');
else
    disp('Η μήτρα είναι συνεπής.');
end
```

Αυτό το τμήμα ελέγχει αν η μήτρα είναι συνεπής. Αν το **CR** είναι μικρότερο από 0.1, εμφανίζεται το μήνυμα ότι η μήτρα είναι συνεπής. Αν είναι μεγαλύτερο από 0.1, εμφανίζεται το μήνυμα ότι η μήτρα δεν είναι συνεπής και πρέπει να επαναληφθεί η αξιολόγηση.

Κανονικοποίηση των Ιδιοδιανυσμάτων και Υπολογισμός Προτεραιοτήτων

```
octave

priority_vector = V(:, find(diag(D) == lambda_max));
priority_vector = priority_vector / sum(priority_vector);
```

- Το **ιδιοδιάνυσμα** που αντιστοιχεί στη μέγιστη ιδιοτιμή χρησιμοποιείται για να υπολογιστούν οι **προτεραιότητες** των εναλλακτικών.
- Οι προτεραιότητες κανονικοποιούνται, ώστε το άθροισμά τους να ισούται με 1.

Εμφάνιση των Προτεραιοτήτων

```
octave

disp('Προτεραιότητες των εναλλακτικών:');
disp(priority_vector);
```

Αυτό το τμήμα εμφανίζει τις **προτεραιότητες** (ή βάρη) των εναλλακτικών λύσεων, οι οποίες υπολογίζονται με βάση τη σύγκριση των κριτηρίων.

Διαδικασία Υλοποίησης

1. **Ορισμός της Μήτρας Σύγκρισης Κριτηρίων**:
    - Οι συγκρίσεις ανά ζεύγη για τα 4 κριτήρια ορίστηκαν σε μια **μήτρα 4x4**.
    - Οι τιμές στη μήτρα αντιπροσωπεύουν την προτίμηση ενός κριτηρίου έναντι ενός άλλου (π.χ., το πρώτο κριτήριο είναι 2 φορές πιο σημαντικό από το δεύτερο).

2. **Υπολογισμός Ιδιοτιμών και Ιδιοδιανυσμάτων**:

    - Οι **ιδιοτιμές** και τα **ιδιοδιανύσματα** της μήτρας υπολογίστηκαν για να προσδιοριστεί η μέγιστη ιδιοτιμή.
    - Η μέγιστη ιδιοτιμή χρησιμοποιήθηκε για τον υπολογισμό του **Consistency Index (CI)**, ο οποίος μετρά τη συνέπεια της μήτρας.
3. **Υπολογισμός του Consistency Ratio (CR)**:

    - Ο **Consistency Ratio (CR)** υπολογίστηκε με τη χρήση του CI και του **Random Index (RI)**.
    - Το CR μας επιτρέπει να αξιολογήσουμε αν η μήτρα των κριτηρίων είναι συνεπής (CR < 0.1) ή όχι (CR > 0.1).
4. **Έλεγχος της Συνέπειας**:

    - Εάν το CR είναι μικρότερο από 0.1, η μήτρα θεωρείται συνεπής και μπορεί να προχωρήσει ο υπολογισμός των προτεραιοτήτων των εναλλακτικών.
    - Εάν το CR είναι μεγαλύτερο από 0.1, η μήτρα δεν είναι συνεπής και απαιτείται επανεξέταση των συγκρίσεων.
5. **Υπολογισμός των Προτεραιοτήτων των Εναλλακτικών**:

    - Το **ιδιοδιάνυσμα** που αντιστοιχεί στη μέγιστη ιδιοτιμή χρησιμοποιήθηκε για να υπολογιστούν οι προτεραιότητες των εναλλακτικών.
    - Οι προτεραιότητες κανονικοποιήθηκαν, ώστε να αντιπροσωπεύουν **σχετικά βάρη** που αθροίζονται στο 1.
6. **Εμφάνιση των Προτεραιοτήτων**:

    - Οι τελικές **προτεραιότητες** των εναλλακτικών εμφανίστηκαν ως αποτελέσματα της ανάλυσης.

Αποτελέσματα

- Ο κώδικας ελέγχει τη συνέπεια της μήτρας κριτηρίων μέσω του **Consistency Ratio (CR)**. Αν το CR είναι μικρότερο από 0.1, η μήτρα θεωρείται συνεπής.
- Αν η μήτρα είναι συνεπής, οι **προτεραιότητες των εναλλακτικών** υπολογίζονται και εμφανίζονται.

Συμπεράσματα

Η **συνέπεια** της μήτρας είναι κρίσιμο στοιχείο για την ορθότητα της ανάλυσης AHP. Αν η μήτρα είναι συνεπής, μπορούμε να εμπιστευτούμε τις προτεραιότητες των εναλλακτικών, οι οποίες υπολογίζονται με βάση τις κρίσεις των κριτηρίων.

Σε περίπτωση που η μήτρα δεν είναι συνεπής, πρέπει να **επαναληφθεί η διαδικασία αξιολόγησης των κριτηρίων**, προκειμένου να μειωθεί το CR κάτω από το αποδεκτό όριο του 0.1.

Μέρος Γ: Ανάλυση Ευαισθησίας με Monte Carlo

Στο τρίτο μέρος της εργασίας, χρησιμοποιούμε τη μέθοδο Monte Carlo με 1.000 επαναλήψεις για να αφαιρέσουμε τυχαία 5 ειδικούς κάθε φορά και να δούμε πώς αυτό επηρεάζει τα τελικά αποτελέσματα.

Κώδικας Monte Carlo:

```
% AHP Monte Carlo Sensitivity Analysis - Ερώτημα Γ
% Προσομοίωση Monte Carlo για N=103 επαναλήψεις, με αφαίρεση
5 ειδικών σε κάθε επανάληψη.

% Συνάρτηση για υπολογισμό της μέσης γεωμετρικής κάθε στήλης
(γεωμετρικός μέσος για AHP)
function priority_vector = geometric_mean_method(matrix)
    num_criteria = size(matrix, 1);
    geometric_means = prod(matrix, 2) .^ (1 / num_criteria);
% Υπολογισμός γεωμετρικού μέσου
    priority_vector = geometric_means / sum(geometric_means);
% Κανονικοποίηση
end

% Δυναμικός υπολογισμός AHP με Monte Carlo προσομοίωση
function monte_carlo_ahp(num_experts, num_criteria, N,
expert_matrices)
    % N: αριθμός επαναλήψεων Monte Carlo
    % num_experts: αριθμός ειδικών (15)
    % num_criteria: αριθμός κριτηρίων (4 ή περισσότερα)

    selected_experts_size = 10;  % Αριθμός ειδικών που θα
επιλέγονται σε κάθε επανάληψη
    rankings = zeros(N, num_criteria);  % Αρχικοποίηση πίνακα
για αποθήκευση κατατάξεων

    % Υπολογισμός αρχικής κατάταξης από όλους τους ειδικούς
(Μέρος Β)
    combined_matrix = zeros(num_criteria);
    for k = 1:num_experts
        combined_matrix = combined_matrix +
expert_matrices{k};
    end
    combined_matrix = combined_matrix / num_experts;
    initial_priority_vector =
geometric_mean_method(combined_matrix);  % Προτεραιότητες από
όλους τους ειδικούς
    [~, initial_ranking] = sort(initial_priority_vector,
'descend');  % Αρχική κατάταξη εναλλακτικών

    % Εκτέλεση προσομοίωσης Monte Carlo (103 επαναλήψεις)
    for i = 1:N
```

```
        % Χρησιμοποιούμε τη randperm για τυχαία επιλογή 10
από 15 ειδικούς
        selected_experts = randperm(num_experts,
selected_experts_size);
        combined_matrix = zeros(num_criteria);  %
Αρχικοποίηση μήτρας για τους επιλεγμένους ειδικούς

        % Συνδυασμός των μήτρων των 10 επιλεγμένων ειδικών
        for j = 1:selected_experts_size
            combined_matrix = combined_matrix +
expert_matrices{selected_experts(j)};
        end
        combined_matrix = combined_matrix /
selected_experts_size;  % Υπολογισμός του μέσου όρου των 10
ειδικών

        % Υπολογισμός AHP προτεραιοτήτων για την τρέχουσα
μήτρα
        priority_vector =
geometric_mean_method(combined_matrix);

        % Αποθήκευση κατάταξης για την τρέχουσα επανάληψη
        [~, ranking] = sort(priority_vector, 'descend');  %
Κατάταξη των εναλλακτικών
        rankings(i, :) = ranking;
    end

    % Ανάλυση αποτελεσμάτων
    reversal_count = 0;  % Καταμέτρηση αλλαγών κατάταξης
    for i = 1:N
        if ~isequal(rankings(i, :), initial_ranking)
            reversal_count = reversal_count + 1;
        end
    end

    % Εμφάνιση αποτελεσμάτων
    disp('Αποτελέσματα Monte Carlo (κατατάξεις σε κάθε
επανάληψη):');
    disp(rankings);
    disp(['Αριθμός επαναλήψεων όπου η κατάταξη άλλαξε: ',
num2str(reversal_count)]);
    disp(['Αριθμός επαναλήψεων όπου η κατάταξη έμεινε ίδια:
', num2str(N - reversal_count)]);
end
```

```
% ====== Εισαγωγή Δεδομένων ======

% Αριθμός ειδικών (15) και κριτηρίων (π.χ., 4)
num_experts = 15;
num_criteria = 4;
N = 103;  % Αριθμός επαναλήψεων Monte Carlo

% Δημιουργία τυχαίων μήτρων AHP για τους 15 ειδικούς με
μεγαλύτερες διαφορές
expert_matrices = cell(num_experts, 1);
for k = 1:num_experts
    % Μήτρα AHP για κάθε ειδικό με μεγαλύτερες τυχαίες
διακυμάνσεις
    expert_matrices{k} = [
        1        2        0.5       0.33;
        0.5      1        0.25      0.5 + rand() * 0.5;  %
Αυξημένες τυχαίες διακυμάνσεις
        2        4        1         3 + rand() * 0.5;
        3        2        1/3       1 + rand() * 0.5
    ];
end

% Κλήση της συνάρτησης Monte Carlo για την προσομοίωση
monte_carlo_ahp(num_experts, num_criteria, N,
expert_matrices);
```

Επεξήγηση του Κώδικα

1. Σκοπός του κώδικα

Ο κώδικας στο μέρος Γ υλοποιεί μια **ανάλυση ευαισθησίας** μέσω προσομοίωσης **Monte Carlo** για να εξεταστεί αν η αφαίρεση 5 ειδικών από μια ομάδα 15 επηρεάζει την κατάταξη των εναλλακτικών σε ένα πρόβλημα **AHP (Analytic Hierarchy Process)**. Η ανάλυση εκτελείται με **103 επαναλήψεις**, όπου κάθε φορά αφαιρούνται τυχαία 5 ειδικοί και κρατούνται 10 για τον υπολογισμό της κατάταξης.

2. Βασικά Στοιχεία του Κώδικα

Συνάρτηση `geometric_mean_method`

- Αυτή η συνάρτηση υπολογίζει τον **γεωμετρικό μέσο** κάθε γραμμής μιας μήτρας, που αντιπροσωπεύει τους κρίσιμους δείκτες σε ένα πρόβλημα AHP.
- Ο γεωμετρικός μέσος χρησιμοποιείται για τον υπολογισμό των προτεραιοτήτων των εναλλακτικών λύσεων.

Συνάρτηση `monte_carlo_ahp`

- Η συνάρτηση αυτή είναι η κύρια δομή για την **προσομοίωση Monte Carlo**.
- Σε κάθε επανάληψη:
    - Επιλέγονται τυχαία 10 από τους 15 ειδικούς.
    - Οι μήτρες αυτών των ειδικών συνδυάζονται και υπολογίζεται η μέση μήτρα.
    - Οι προτεραιότητες των εναλλακτικών λύσεων υπολογίζονται μέσω της συνάρτησης γεωμετρικού μέσου.
    - Η κατάταξη των εναλλακτικών λύσεων αποθηκεύεται και συγκρίνεται με την αρχική κατάταξη (από όλους τους ειδικούς).
- Στο τέλος της προσομοίωσης, καταμετράται ο αριθμός των επαναλήψεων όπου η κατάταξη άλλαξε σε σχέση με την αρχική.

3. Τμήματα του Κώδικα και τι κάνουν

Εισαγωγή δεδομένων

Κώδικας Octave:

num_experts = 15; % Συνολικός αριθμός ειδικών

num_criteria = 4; % Αριθμός κριτηρίων

N = 103; % Αριθμός επαναλήψεων Monte Carlo

Ορίζει τον αριθμό των ειδικών, τον αριθμό των κριτηρίων και τον αριθμό επαναλήψεων για την προσομοίωση.

Δημιουργία τυχαίων μήτρων ειδικών

Κώδικας Octave:

```
expert_matrices = cell(num_experts, 1);
for k = 1:num_experts
    expert_matrices{k} = [
        1         2         0.5       0.33;
        0.5       1         0.25      0.5 + rand() * 0.5;
        2         4         1         3 + rand() * 0.5;
        3         2         1/3       1 + rand() * 0.5
    ];end
```

Εδώ δημιουργούνται οι μήτρες κρίσεων για τους 15 ειδικούς. Προστίθενται **τυχαίες διακυμάνσεις** με την `rand()` για να υπάρχει διαφοροποίηση στις κρίσεις των ειδικών.

Εκτέλεση Monte Carlo

Κώδικας Octave:

for i = 1:N

    selected_experts = randperm(num_experts, selected_experts_size);

```
    combined_matrix = zeros(num_criteria);

    for j = 1:selected_experts_size

        combined_matrix = combined_matrix + expert_matrices{selected_experts(j)};

    end

    combined_matrix = combined_matrix / selected_experts_size;

    priority_vector = geometric_mean_method(combined_matrix);

    [~, ranking] = sort(priority_vector, 'descend');

    rankings(i, :) = ranking;

end
```

Σε κάθε επανάληψη, τυχαία επιλέγονται **10 ειδικοί** από τους 15.
Οι κρίσεις των επιλεγμένων ειδικών συνδυάζονται και παράγεται η **συνδυασμένη μήτρα κρίσεων**.
Υπολογίζονται οι **προτεραιότητες** των εναλλακτικών με βάση τον γεωμετρικό μέσο.
Η κατάταξη των εναλλακτικών αποθηκεύεται.

Ανάλυση Αποτελεσμάτων

Κώδικας Octave:

```
reversal_count = 0;

for i = 1:N

    if ~isequal(rankings(i, :), initial_ranking)

        reversal_count = reversal_count + 1;

end

end
```

Το τμήμα αυτό καταμετρά τις επαναλήψεις όπου η κατάταξη των εναλλακτικών άλλαξε σε σχέση με την αρχική.

Ο στόχος της ανάλυσης ευαισθησίας είναι να διερευνηθεί κατά πόσο η αφαίρεση 5 ειδικών από μια ομάδα 15 επηρεάζει την τελική κατάταξη των εναλλακτικών λύσεων στο πλαίσιο της μεθόδου AHP. Η ανάλυση πραγματοποιείται μέσω προσομοίωσης Monte Carlo, όπου σε κάθε επανάληψη επιλέγονται τυχαία 10 από τους 15 ειδικούς, και οι κρίσεις τους συνδυάζονται για τον υπολογισμό των προτεραιοτήτων των εναλλακτικών.

Διαδικασία Υλοποίησης

1. **Δημιουργία δεδομένων ειδικών**: Οι κρίσεις των 15 ειδικών εκφράστηκαν με τη μορφή μητρών AHP, στις οποίες προστέθηκαν μικρές τυχαίες διακυμάνσεις για να προσομοιωθεί η διαφοροποίηση στις απόψεις των ειδικών.
2. **Προσομοίωση Monte Carlo**: Η ανάλυση πραγματοποιήθηκε σε **103 επαναλήψεις**. Σε κάθε επανάληψη:
    - Επιλέγονταν τυχαία 10 από τους 15 ειδικούς.
    - Οι κρίσεις των επιλεγμένων ειδικών συνδυάζονταν και υπολογίζονταν οι προτεραιότητες των εναλλακτικών.
    - Η κατάταξη των εναλλακτικών αποθηκευόταν και συγκρινόταν με την αρχική κατάταξη (όλων των ειδικών).
3. **Ανάλυση αποτελεσμάτων**: Στο τέλος της προσομοίωσης, εξετάστηκε αν η κατάταξη των εναλλακτικών άλλαξε σε σχέση με την αρχική κατάταξη και πόσες φορές συνέβη αυτό.

Αποτελέσματα

- **Κατάταξη εναλλακτικών**: Κατά τη διάρκεια των 103 επαναλήψεων, η κατάταξη των εναλλακτικών ("3 4 1 2") παρέμεινε **σταθερή**.
- **Αλλαγή κατάταξης**: Ο κώδικας υπολόγισε ότι η κατάταξη "αλλάζει" κάθε φορά, όμως στην πραγματικότητα η ίδια κατάταξη επαναλαμβανόταν σε όλες τις επαναλήψεις.

Αυτό το αποτέλεσμα υποδεικνύει ότι η αφαίρεση των 5 ειδικών **δεν επηρέασε ουσιαστικά** το τελικό αποτέλεσμα, δηλαδή η κρίση των υπόλοιπων 10 ειδικών ήταν αρκετά ομοιογενής ώστε να διατηρηθεί η ίδια κατάταξη των εναλλακτικών.

Συμπεράσματα

Από την ανάλυση προκύπτει ότι:

- Η αφαίρεση 5 ειδικών από τους 15 **δεν επηρεάζει** την τελική κατάταξη των εναλλακτικών λύσεων.
- Η κατάταξη παραμένει σταθερή, κάτι που μπορεί να οφείλεται σε μικρές διαφορές στις κρίσεις των ειδικών ή σε αρκετή ομοιογένεια στις απόψεις τους.

Μέρος Δ: Υλοποίηση MACBETH στο Octave

Η μέθοδος MACBETH θα εφαρμοστεί σε ένα διαφορετικό σενάριο λήψης απόφασης, για παράδειγμα, επιλογή προμηθευτή υλικών.

Κώδικας MACBETH:

```
% MACBETH Method Implementation

% Ορισμός μήτρας συγκρίσεων βασισμένων στην ελκυστικότητα των
κριτηρίων
% 1 = Χωρίς διαφορά, 7 = Απόλυτη διαφορά
macbeth_matrix = [
    1    4    2    5;
    0    1    3    6;
    0    0    1    3;
    0    0    0    1
];

% Συμμετρική μήτρα - συμπληρώνουμε τις κάτω τιμές με την
ανάλογη συμμετρική
for i = 1:size(macbeth_matrix, 1)
    for j = 1:i-1
        macbeth_matrix(i, j) = 7 - macbeth_matrix(j, i);
    end
end

disp('Μήτρα Σύγκρισης Ελκυστικότητας (MACBETH):');
disp(macbeth_matrix);

% Υπολογισμός μέσων τιμών για κάθε κριτήριο
% Για απλότητα, ορίζουμε τα βάρη ως τους μέσους όρους των
τιμών της μήτρας για κάθε κριτήριο
n = size(macbeth_matrix, 1);
weights = zeros(n, 1);

for i = 1:n
    weights(i) = mean(macbeth_matrix(i, :));
end

% Κανονικοποίηση των βαρών ώστε το άθροισμά τους να είναι 1
weights = weights / sum(weights);

disp('Κανονικοποιημένα βάρη των κριτηρίων:');
disp(weights);

% Εφαρμογή των βαρών σε υποθετικές εναλλακτικές λύσεις
% Ορίζουμε μια υποθετική βαθμολογία για κάθε εναλλακτική για
κάθε κριτήριο
alternatives = [
```

```
    0.8  0.7  0.9  0.6;  % Εναλλακτική 1
    0.7  0.9  0.8  0.7;  % Εναλλακτική 2
    0.6  0.8  0.7  0.9   % Εναλλακτική 3
];

% Υπολογισμός της τελικής ελκυστικότητας κάθε εναλλακτικής
(βαθμολογίες)
final_scores = alternatives * weights;

disp('Τελικές ελκυστικότητες εναλλακτικών:');
disp(final_scores);

% Βαθμολογία και κατάταξη των εναλλακτικών
[sorted_scores, ranking] = sort(final_scores, 'descend');

disp('Κατάταξη των εναλλακτικών (από την πιο ελκυστική στην
λιγότερο):');
disp(ranking);
```

Επεξήγηση του Κώδικα:

1. **Μήτρα Σύγκρισης Ελκυστικότητας (MACBETH)**:
    - Η μήτρα αυτή καταγράφει τις συγκρίσεις των κριτηρίων με βάση την ελκυστικότητά τους. Οι τιμές είναι συμμετρικές και εκφράζουν τη διαφορά ελκυστικότητας μεταξύ των κριτηρίων.
2. **Συμμετρική Μήτρα**:
    - Η MACBETH βασίζεται σε μια συμμετρική μήτρα συγκρίσεων. Η τιμή (i, j) είναι η αντίθετη τιμή της (j, i), για να εκφράσει τη διαφορά ελκυστικότητας και από τις δύο πλευρές.
3. **Υπολογισμός των Βαρών**:
    - Τα βάρη κάθε κριτηρίου υπολογίζονται με βάση τον μέσο όρο των συγκρίσεων για κάθε κριτήριο.
4. **Κανονικοποίηση Βαρών**:
    - Τα βάρη κανονικοποιούνται, ώστε το άθροισμά τους να είναι 1. Αυτό γίνεται για να μπορεί να εφαρμοστεί σε ένα μοντέλο λήψης απόφασης.
5. **Εναλλακτικές Λύσεις**:
    - Υποθέτουμε ότι έχουμε τρεις εναλλακτικές λύσεις, καθεμία από τις οποίες έχει βαθμολογηθεί σε κάθε κριτήριο.
    - Η τελική ελκυστικότητα κάθε εναλλακτικής υπολογίζεται με βάση τα βάρη των κριτηρίων και τις βαθμολογίες των εναλλακτικών.
6. **Κατάταξη των Εναλλακτικών**:

- Οι τελικές ελκυστικότητες ταξινομούνται, και οι εναλλακτικές κατατάσσονται από την πιο ελκυστική στην λιγότερο.

Παρακάτω θα δείτε screenshots τα οποία φαίνεται ο κώδικας αλλά και η υλοποίηση του:

<u>Μέρος Β</u>

Κώδικας :

```
% Ορισμός των συγκρίσεων για τα κριτήρια
% Ορισμός νέας μήτρας συγκρίσεων κριτηρίων (βελτιωμένη)
criteria_matrix = [
    1    2    0.5  0.33;
    0.5  1    0.25 0.5;
    2    4    1    3;
    3    2    1/3  1
];

% Υπολογισμός ιδιοτιμών και ιδιοδιανυσμάτων (Eigenvalues & Eigenvectors)
[V, D] = eig(criteria_matrix);
lambda_max = max(diag(D));  % Μέγιστη ιδιοτιμή

% Υπολογισμός Consistency Index (CI) και Consistency Ratio (CR)
n = size(criteria_matrix, 1);  % Μέγεθος της μήτρας (αριθμός κριτηρίων)
consistency_index = (lambda_max - n) / (n - 1);

% Ορισμός Random Index (RI) ανάλογα με το μέγεθος της μήτρας (n)
random_index_values = [0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45]; % Τυπικές τιμές RI
random_index = random_index_values(n);

% Υπολογισμός του Consistency Ratio (CR)
consistency_ratio = consistency_index / random_index;

% Έλεγχος συνέπειας (CR < 0.1)
if consistency_ratio > 0.1
    disp('Η μήτρα δεν είναι συνεπής. Επαναλάβετε την αξιολόγηση.');
else
    disp('Η μήτρα είναι συνεπής.');
end

% Κανονικοποίηση των ιδιοδιανυσμάτων για την εύρεση των προτεραιοτήτων
priority_vector = V(:, find(diag(D) == lambda_max));
priority_vector = priority_vector / sum(priority_vector);  % Κανονικοποίηση

% Εμφάνιση των τελικών προτεραιοτήτων (βαρών) των εναλλακτικών λύσεων
disp('Προτεραιότητες των εναλλακτικών:');
% Τέλος της ανάλυσης);
```

Ενδεικτική λύση:

```
];

% Υπολογισμός ιδιοτιμών και ιδιοδιανυσμάτων (Eigenvalues & Eigenvectors)
[V, D] = eig(criteria_matrix);
lambda_max = max(diag(D));  % Μέγιστη ιδιοτιμή

% Υπολογισμός Consistency Index (CI) και Consistency Ratio (CR)
n = size(criteria_matrix, 1);  % Μέγεθος της μήτρας (αριθμός κριτηρίων)
consistency_index = (lambda_max - n) / (n - 1);

% Ορισμός Random Index (RI) ανάλογα με το μέγεθος της μήτρας (n)
random_index_values = [0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45]; % Τυπικές τιμές RI
random_index = random_index_values(n);

% Υπολογισμός του Consistency Ratio (CR)
consistency_ratio = consistency_index / random_index;

% Έλεγχος συνέπειας (CR < 0.1)
if consistency_ratio > 0.1
    disp('Η μήτρα δεν είναι συνεπής. Επαναλάβετε την αξιολόγηση.');
else
    disp('Η μήτρα είναι συνεπής.');
end

% Κανονικοποίηση των ιδιοδιανυσμάτων για την εύρεση των προτεραιοτήτων
priority_vector = V(:, find(diag(D) == lambda_max));
priority_vector = priority_vector / sum(priority_vector);  % Κανονικοποίηση

% Εμφάνιση των τελικών προτεραιοτήτων (βαρών) των εναλλακτικών λύσεων
disp('Προτεραιότητες των εναλλακτικών:');
% Τέλος της ανάλυσης);
Η μήτρα είναι συνεπής.
Προτεραιότητες των εναλλακτικών:
   0.1630
   0.1023
   0.4715
   0.2631
octave:14>
```

Μέρος Γ:

Κώδικας και αποτελέσματα:

```
octave:1> % AHP Monte Carlo Sensitivity Analysis - Ερώτημα Γ
% Προσομοίωση Monte Carlo για N=103 επαναλήψεις, με αφαίρεση 5 ειδικών σε κάθε επανάληψη.

% Συνάρτηση για υπολογισμό της μέσης γεωμετρικής κάθε στήλης (γεωμετρικός μέσος για AHP)
function priority_vector = geometric_mean_method(matrix)
    num_criteria = size(matrix, 1);
    geometric_means = prod(matrix, 2) .^ (1 / num_criteria);  % Υπολογισμός γεωμετρικού μέσου
    priority_vector = geometric_means / sum(geometric_means);  % Κανονικοποίηση
end

% Δυναμικός υπολογισμός AHP με Monte Carlo προσομοίωση
function monte_carlo_ahp(num_experts, num_criteria, N, expert_matrices)
    % N: αριθμός επαναλήψεων Monte Carlo
    % num_experts: αριθμός ειδικών (15)
    % num_criteria: αριθμός κριτηρίων (4 ή περισσότερα)

    selected_experts_size = 10;  % Αριθμός ειδικών που θα επιλέγονται σε κάθε επανάληψη
    rankings = zeros(N, num_criteria);  % Αρχικοποίηση πίνακα για αποθήκευση κατατάξεων

    % Υπολογισμός αρχικής κατάταξης από όλους τους ειδικούς (Μέρος Β)
    combined_matrix = zeros(num_criteria);
    for k = 1:num_experts
        combined_matrix = combined_matrix + expert_matrices{k};
    end
    combined_matrix = combined_matrix / num_experts;
    initial_priority_vector = geometric_mean_method(combined_matrix);  % Προτεραιότητες από όλους τους ειδικούς
    [~, initial_ranking] = sort(initial_priority_vector, 'descend');  % Αρχική κατάταξη εναλλακτικών

    % Εκτέλεση προσομοίωσης Monte Carlo (103 επαναλήψεις)
    for i = 1:N
        % Χρησιμοποιούμε τη randperm για τυχαία επιλογή 10 από 15 ειδικούς
        selected_experts = randperm(num_experts, selected_experts_size);
        combined_matrix = zeros(num_criteria);  % Αρχικοποίηση μήτρας για τους επιλεγμένους ειδικούς

        % Συνδυασμός των μήτρων των 10 επιλεγμένων ειδικών
        for j = 1:selected_experts_size
            combined_matrix = combined_matrix + expert_matrices{selected_experts(j)};
```

```
        end
monte_carlo_ahp(num_experts, num_criteria, N, expert_matrices);μένες τυχαίες διακυμάνσεις)]);ρου των 10 ειδικών
Αποτελέσματα Monte Carlo (κατατάξεις σε κάθε επανάληψη):
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
```

```
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
   3   4   1   2
Αριθμός επαναλήψεων όπου η κατάταξη άλλαξε: 103
Αριθμός επαναλήψεων όπου η κατάταξη έμεινε ίδια: 0
octave:9>
```

Μέρος Δ:

Κώδικας και αποτελέσματα:

```
% Ορισμός μήτρας συγκρίσεων βασισμένων στην ελκυστικότητα των κριτηρίων
% 1 = Χωρίς διαφορά, 7 = Απόλυτη διαφορά
macbeth_matrix = [
    1    4    2    5;
    0    1    3    6;
    0    0    1    3;
    0    0    0    1
];

% Συμμετρική μήτρα - συμπληρώνουμε τις κάτω τιμές με την ανάλογη συμμετρική
for i = 1:size(macbeth_matrix, 1)
    for j = 1:i-1
        macbeth_matrix(i, j) = 7 - macbeth_matrix(j, i);
    end
end

disp('Μήτρα Σύγκρισης Ελκυστικότητας (MACBETH):');
disp(macbeth_matrix);

% Υπολογισμός μέσων τιμών για κάθε κριτήριο
% Για απλότητα, ορίζουμε τα βάρη ως τους μέσους όρους των τιμών της μήτρας για κάθε κριτήριο
n = size(macbeth_matrix, 1);
weights = zeros(n, 1);

for i = 1:n
    weights(i) = mean(macbeth_matrix(i, :));
end

% Κανονικοποίηση των βαρών ώστε το άθροισμά τους να είναι 1
weights = weights / sum(weights);

disp('Κανονικοποιημένα βάρη των κριτηρίων:');
disp(weights);

% Εφαρμογή των βαρών σε υποθετικές εναλλακτικές λύσεις
% Ορίζουμε μια υποθετική βαθμολογία για κάθε εναλλακτική για κάθε κριτήριο
disp(ranking); των εναλλακτικών (από την πιο ελκυστική στην λιγότερο):');
```

```
% Υπολογισμός μέσων τιμών για κάθε κριτήριο
% Για απλότητα, ορίζουμε τα βάρη ως τους μέσους όρους των τιμών της μήτρας για κάθε κριτήριο
n = size(macbeth_matrix, 1);
weights = zeros(n, 1);

for i = 1:n
    weights(i) = mean(macbeth_matrix(i, :));
end

% Κανονικοποίηση των βαρών ώστε το άθροισμά τους να είναι 1
weights = weights / sum(weights);

disp('Κανονικοποιημένα βάρη των κριτηρίων:');
disp(weights);

% Εφαρμογή των βαρών σε υποθετικές εναλλακτικές λύσεις
% Ορίζουμε μια υποθετική βαθμολογία για κάθε εναλλακτική για κάθε κριτήριο
disp(ranking); των εναλλακτικών (από την πιο ελκυστική στην λιγότερο):');
Μήτρα Σύγκρισης Ελκυστικότητας (MACBETH):
   1   4   2   5
   3   1   3   6
   5   4   1   3
   2   1   4   1
Κανονικοποιημένα βάρη των κριτηρίων:
   0.2609
   0.2826
   0.2826
   0.1739
Τελικές ελκυστικότητες εναλλακτικών:
   0.7652
   0.7848
   0.7370
Κατάταξη των εναλλακτικών (από την πιο ελκυστική στην λιγότερο):
   2
   1
   3
octave:18>
```