

| Systemy Wspomagania Decyzji | | |
|---|---|--------------------------------------|
| <small>Temat</small> Porównanie Metod Wielokryterialnej Analizy Decyzji Opartych o Punkty Odniesienia | | |
| <small>Wydział</small> <i>EAIIB</i> | <small>Kierunek</small> <i>Automatyka i Robotyka</i> | <small>Rok</small> <i>III</i> |

Spis treści:

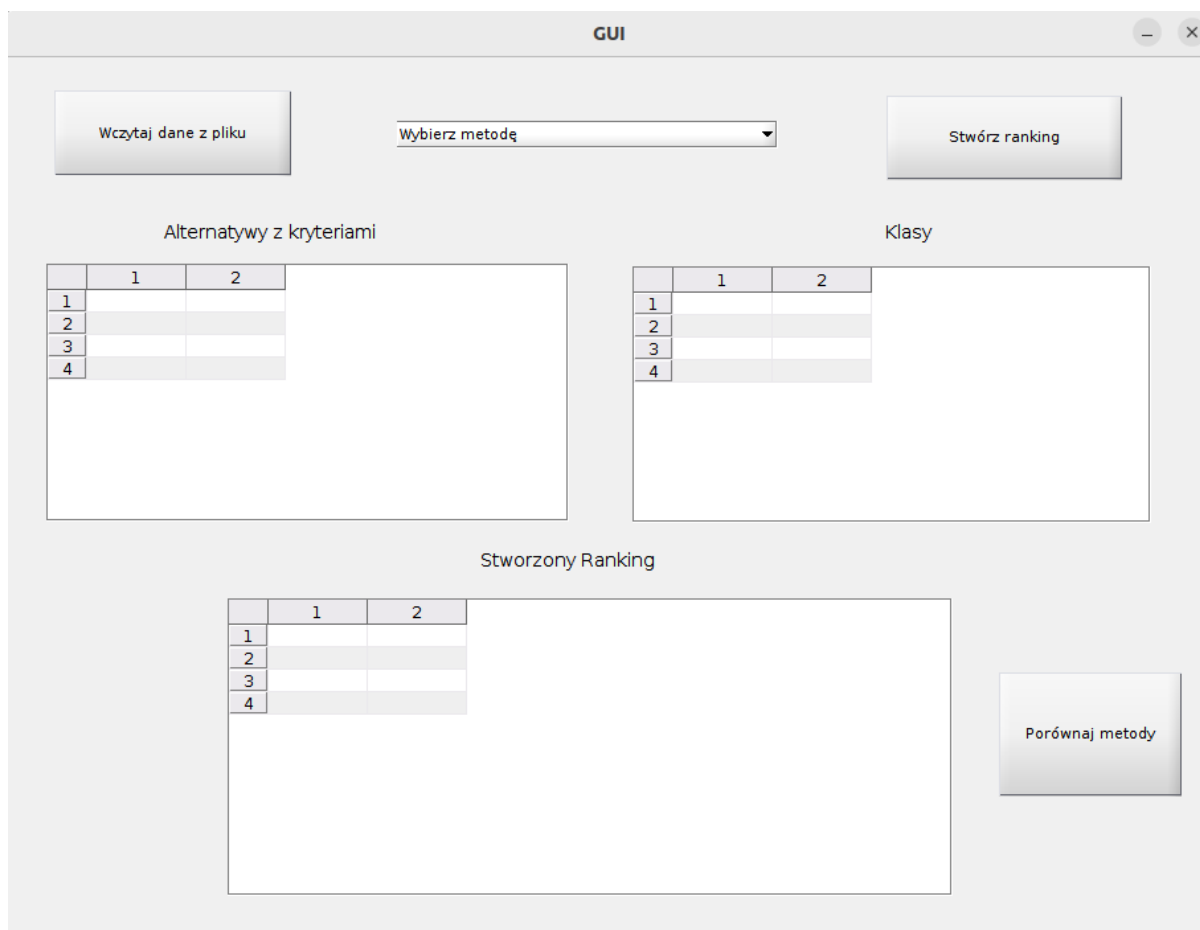
1. Wstęp
2. Opis programu
3. Opis Algorytmów
4. Testowanie
5. Podsumowanie i wnioski

Wstęp:

Celem niniejszego sprawozdania jest przedstawienie wyników przeprowadzonych ćwiczeń laboratoryjnych dotyczących metod optymalizacji wielokryterialnej w kontekście zastosowań w systemach wspomagania decyzji (SWD). W ramach zajęć skupiono się na implementacji, testowaniu oraz analizie skuteczności wybranych metod optymalizacyjnych, takich jak TOPSIS, RSM czy SP-CS, z zastosowaniem dostarczonych danych decyzyjnych oraz odpowiednich narzędzi programistycznych.

Opis programu:

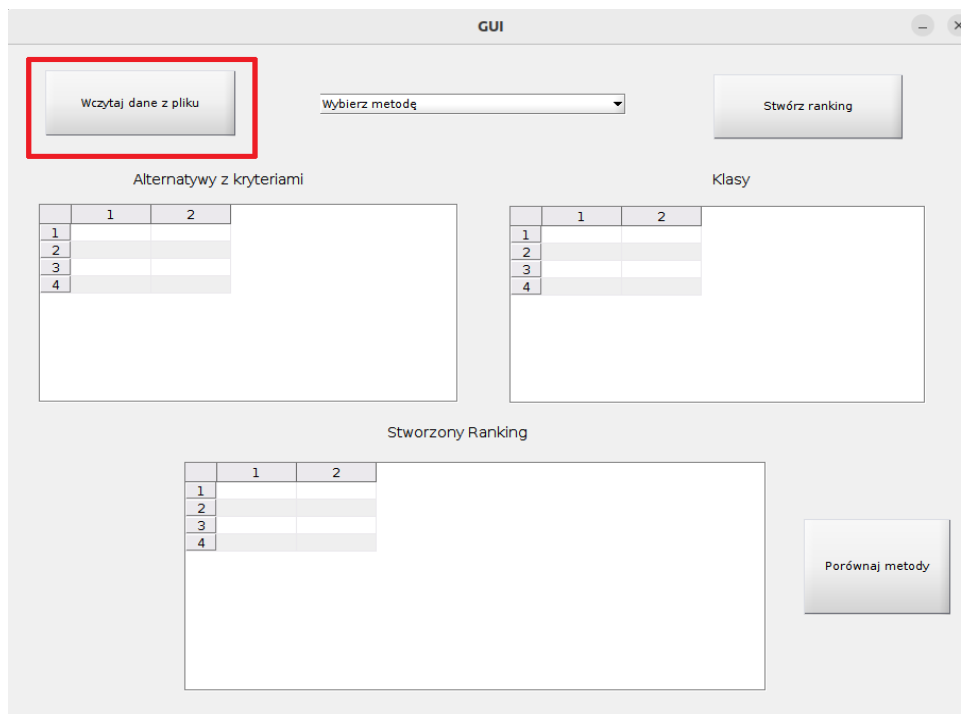
Program składa się z intuicyjnego graficznego interfejsu użytkownika (GUI) oraz pięciu zaawansowanych algorytmów, które implementują różne podejścia do wielokryterialnej analizy decyzji. Interfejs graficzny odgrywa kluczową rolę, umożliwiając użytkownikowi wygodną i efektywną obsługę programu. Dzięki niemu użytkownik ma łatwy dostęp do wszystkich funkcji, takich jak importowanie danych w formacie Excel (.xlsx), wybór jednego z pięciu dostępnych algorytmów do tworzenia rankingu oraz prezentacja wyników w czytelnej i uporządkowanej formie. Program zapewnia kompleksowe wsparcie w procesie podejmowania decyzji, łącząc zaawansowane możliwości analityczne z prostotą obsługi.



Rys.1 Panel graficzny użytkownika

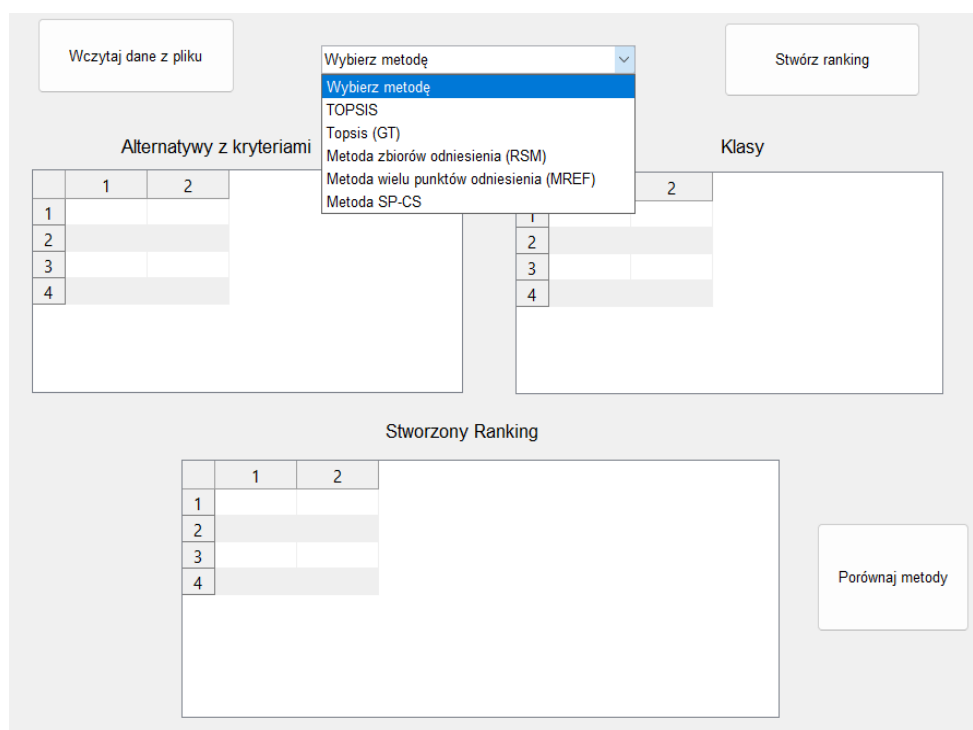
Interfejs użytkownika programu został podzielony na kilka głównych sekcji, co zapewnia przejrzystość oraz intuicyjność obsługi. Taki układ pozwala użytkownikowi w łatwy sposób korzystać z dostępnych funkcji i przeprowadzać analizy wielokryterialne. Poniżej przedstawiono krótkie opisy poszczególnych sekcji interfejsu.

Sekcja Importu Danych umożliwia użytkownikowi wczytanie danych wejściowych z pliku w formacie Excel (.xlsx). Dzięki temu użytkownik może w prosty sposób zaimportować dane, takie jak alternatywy oraz kryteria, które będą podstawą analizy. Funkcja ta eliminuje konieczność ręcznego wprowadzania dużej ilości danych, co znacząco przyspiesza pracę i redukuje ryzyko błędów. Dodatkowo pozwala na łatwe wprowadzenie danych w przejrzystej i ustrukturyzowanej formie, co ułatwia dalsze etapy analizy.



Rys.2 Sekcja Importu Danych

Wybór Metody i Generowanie Rankingu pozwala na wskazanie jednego z pięciu dostępnych algorytmów analizy wielokryterialnej oraz wygenerowanie wynikowego rankingu. W tej części interfejsu użytkownik korzysta z rozwijanego menu, aby wybrać metodę analizy najlepiej dopasowaną do charakteru problemu. Po dokonaniu wyboru wystarczy kliknąć przycisk „Stwórz ranking”, aby uruchomić algorytm i uzyskać uszeregowanie alternatyw na podstawie wybranej metody. Ta sekcja upraszcza proces podejmowania decyzji i umożliwia szybkie uzyskanie wyników.



Rys. 3 Sekcja wyboru metody

Tabela Alternatyw z Kryteriami stanowi wizualizację danych wejściowych. W tej tabeli użytkownik może zobaczyć alternatywy oraz odpowiadające im wartości kryteriów, co pozwala na lepsze zrozumienie danych i ich wpływu na analizę. Tabela ta pełni funkcję przeglądowną, umożliwiając użytkownikowi upewnienie się, że dane zostały załadowane poprawnie i zgodnie z założeniami analizy. Przejrzysty układ ułatwia interpretację i porównanie poszczególnych alternatyw.

The screenshot shows a software interface titled 'GUI'. At the top, there are three main controls: a button 'Wczytaj dane z pliku', a dropdown menu 'Wybierz metodę', and a button 'Stwórz ranking'. Below these, the interface is divided into three main sections. The first section, 'Alternatywy z kryteriami', is highlighted with a red border and contains a table with 4 rows and 2 columns. The second section, 'Klasy', contains a similar table. The third section, 'Stworzony Ranking', contains a larger table with 4 rows and 2 columns. A button 'Porównaj metody' is located at the bottom right.

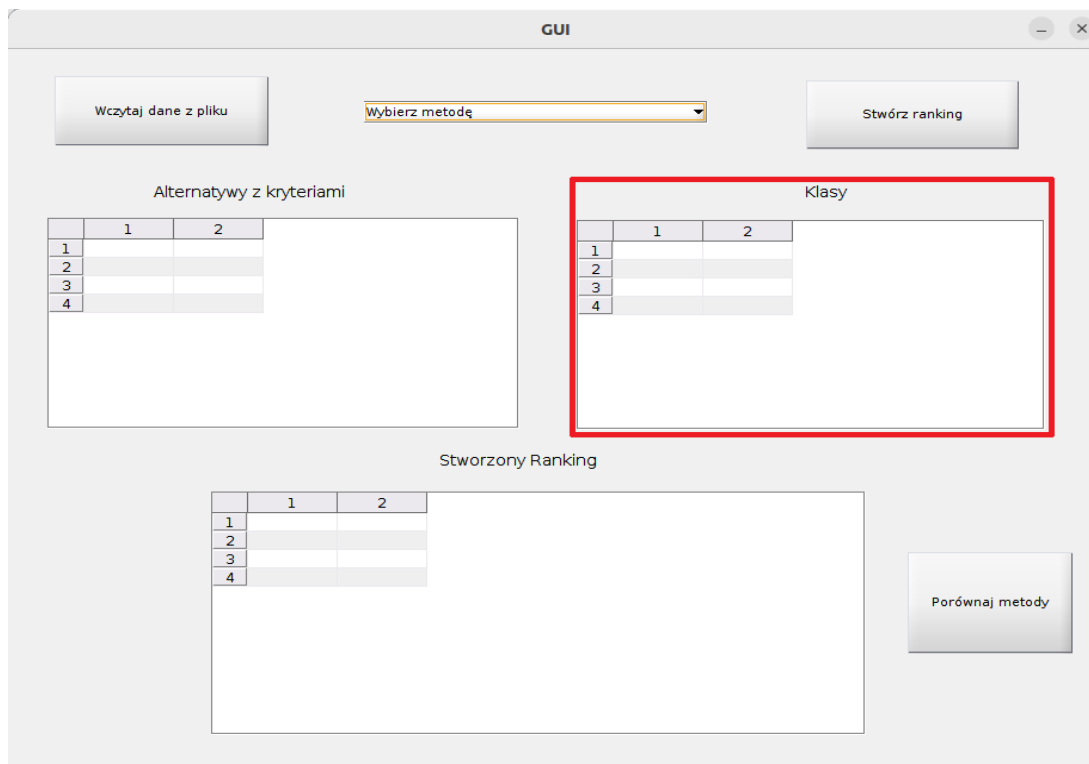
| | 1 | 2 |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |

| | 1 | 2 |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |

| | 1 | 2 |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |

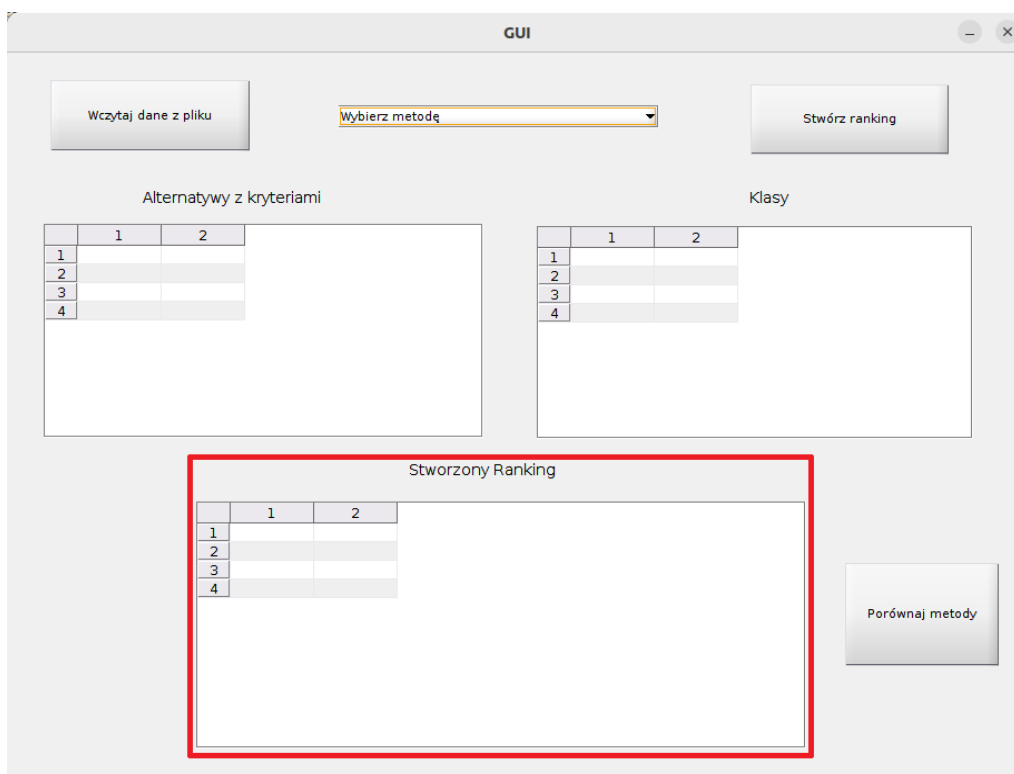
Rys.4 Sekcja Alternatyw z Kryteriami

Sekcja Klasy, która umożliwia przyporządkowanie alternatyw do odpowiednich kategorii. W tej części użytkownik może przeglądać grupowanie alternatyw na podstawie określonych kryteriów. Funkcja ta jest przydatna w analizach, gdzie istotne jest wyróżnienie określonych klas lub grup, co pozwala na bardziej szczegółowe wnioskowanie.



Rys.5 Sekcja Klas

W **Sekcji Wyników** prezentowany jest stworzony ranking. W tej tabeli użytkownik widzi uporządkowane alternatywy wraz z ich pozycjami wynikającymi z wybranej metody analizy. Dzięki temu możliwe jest szybkie porównanie alternatyw i podjęcie decyzji na podstawie uzyskanych wyników. Przejrzysty układ tabeli ułatwia interpretację oraz umożliwia eksport wyników w celu dalszego przetwarzania lub prezentacji.



Rys.6 Sekcja Wyników

Przycisk „**Porównaj metody**” w aplikacji służy do analizy porównawczej wyników uzyskiwanych przy zastosowaniu różnych metod decyzyjnych na tych samych danych wejściowych. Po jego naciśnięciu następuje pobranie danych z tabeli wejściowej „Alternatywy z kryteriami”. Dane te reprezentują różne alternatywy oceniane według określonych kryteriów. Następnie aplikacja wykonuje obliczenia przy użyciu metod decyzyjnych, takich jak Topsis (GT), RSM, MREF, SP-CS oraz klasyczny TOPSIS. Każda z tych metod przetwarza dane zgodnie z przypisanymi wagami i typami kryteriów. Wynikiem działania każdej metody jest ranking alternatyw, który jest następnie zapisywany w tabeli wynikowej i prezentowany użytkownikowi.

Wczytaj dane z pliku

Wybierz metodę

Stwórz ranking

Alternatywy z kryteriami

| | 1 | 2 |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |

Klasy

| | 1 | 2 |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |

Stworzony Ranking

| | 1 | 2 |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |

Porównaj metody

Rys.7 Przycisk porównaj metody

Opis Algorytmów:

Algorytm TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution)

Algorytm TOPSIS to popularna metoda analizy wielokryterialnej, która umożliwia porównanie alternatyw na podstawie ich odległości od punktu idealnego oraz punktu anty-idealnego. Idea algorytmu polega na założeniu, że optymalne rozwiązanie powinno być jak najbliżej punktu idealnego (reprezentującego najlepsze możliwe wartości dla wszystkich kryteriów) oraz jak najdalej od punktu anty-idealnego (reprezentującego najgorsze możliwe wartości).

Proces TOPSIS składa się z kilku etapów: normalizacji macierzy decyzyjnej, wyznaczenia punktów idealnego i anty-idealnego, obliczenia odległości każdej alternatywy od tych punktów oraz wyliczenia współczynnika bliskości do punktu idealnego. Alternatywy są następnie rangowane według wartości tego współczynnika.

TOPSIS jest intuicyjną metodą o stosunkowo niskiej złożoności obliczeniowej, co sprawia, że jest szeroko stosowana w praktyce. Główną zaletą algorytmu jest jego zdolność do uwzględnienia zarówno pozytywnych, jak i negatywnych aspektów alternatyw, co prowadzi do bardziej zbilansowanych wyników. Jednakże metoda ta zakłada liniowość preferencji i może być wrażliwa na sposób normalizacji danych, co w niektórych przypadkach może wpływać na rezultaty

```
Editor - /home/pawsooon/Documents/MATLAB/SWD_projekt/TOPSIS.m
rsm.m  x  Topsis_GT.m  x  MREF.m  x  TOPSIS.m  x  GUL.m  x  +
1  function ranking = TOPSIS(data_table, weights, criteria_types)
2      % data_table - macierz numeryczna alternatyw
3      % weights - wektor wag
4      % criteria_types - kryteria: 1 (maksymalizowane), -1 (minimalizowane)
5
6      [num_alternatives, num_criteria] = size(data_table);
7
8      % Normalizacja danych
9      normalized_matrix = zeros(num_alternatives, num_criteria);
10     for j = 1:num_criteria
11         normalized_matrix(:, j) = data_table(:, j) / sqrt(sum(data_table(:, j).^2));
12     end
13
14     % Uwzględnienie wag
15     weighted_matrix = normalized_matrix .* weights;
16
17     % Wyznaczenie punktów idealnego i anty-idealnego
18     ideal_solution = zeros(1, num_criteria);
19     anti_ideal_solution = zeros(1, num_criteria);
20     for j = 1:num_criteria
21         if criteria_types(j) == 1
22             ideal_solution(j) = max(weighted_matrix(:, j));
23             anti_ideal_solution(j) = min(weighted_matrix(:, j));
24         else
25             ideal_solution(j) = min(weighted_matrix(:, j));
26             anti_ideal_solution(j) = max(weighted_matrix(:, j));
27         end
28     end
29
30     % Wyznaczenie odległości od punktu idealnego i anty-idealnego
31     distances_to_ideal = sqrt(sum((weighted_matrix - ideal_solution).^2, 2));
32     distances_to_anti_ideal = sqrt(sum((weighted_matrix - anti_ideal_solution).^2, 2));
33
34     % Obliczenie współczynnika bliskości
35     closeness_coefficient = distances_to_anti_ideal ./ (distances_to_ideal + distances_to_anti_ideal);
36
37     % Ranking alternatyw (od najlepszej do najgorszej)
38     [~, sorted_indices] = sort(closeness_coefficient, 'descend');
39     ranking = sorted_indices;
40 end
```

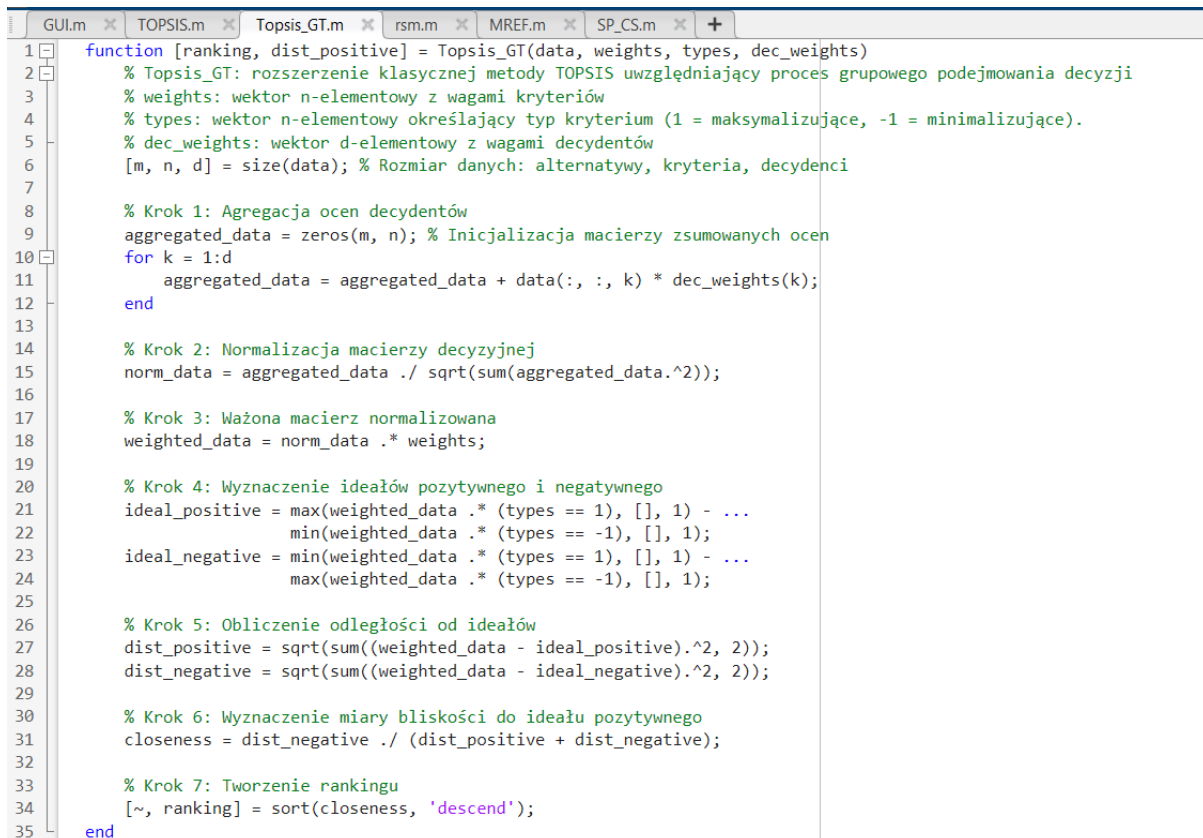
Rys.7 Algorytm TOPSIS

Algorytm TOPSIS GT (TOPSIS with Group Technique)

TOPSIS GT jest rozszerzeniem klasycznego algorytmu TOPSIS, w którym uwzględniono grupowe podejmowanie decyzji. Metoda ta zakłada, że decyzje są podejmowane przez wielu ekspertów, z których każdy może mieć własne preferencje oraz przypisane wagi do kryteriów. W TOPSIS GT wagi kryteriów oraz oceny alternatyw są agregowane w celu uzyskania jednej, wspólnej macierzy decyzyjnej, co umożliwia ocenę alternatyw z uwzględnieniem opinii wielu decydentów.

Proces analizy przebiega podobnie jak w klasycznym TOPSIS: najpierw normalizowana jest macierz decyzyjna, a następnie wyznaczane są punkty idealny i anty-idealny. Kolejno obliczane są odległości alternatyw od tych punktów oraz współczynniki bliskości. Jednakże w TOPSIS GT dodatkowo uwzględnia się metody agregacji preferencji grupowych, takie jak średnia ważona czy mediany ocen.

TOPSIS GT pozwala na bardziej realistyczne modelowanie procesów decyzyjnych, w których uczestniczy wielu interesariuszy. Zaletą tej metody jest jej elastyczność oraz zdolność do uwzględnienia różnorodności opinii, jednak może ona być bardziej czasochłonna i wrażliwa na sposób agregacji wag oraz ocen.



```
1 function [ranking, dist_positive] = Topsis_GT(data, weights, types, dec_weights)
2 % Topsis_GT: rozszerzenie klasycznej metody TOPSIS uwzględniający proces grupowego podejmowania decyzji
3 % weights: wektor n-elementowy z wagami kryteriów
4 % types: wektor n-elementowy określający typ kryterium (1 = maksymalizujące, -1 = minimalizujące).
5 % dec_weights: wektor d-elementowy z wagami decydentów
6 [m, n, d] = size(data); % Rozmiar danych: alternatywy, kryteria, decydenci
7
8 % Krok 1: Agregacja ocen decydentów
9 aggregated_data = zeros(m, n); % Inicjalizacja macierzy zsumowanych ocen
10 for k = 1:d
11     aggregated_data = aggregated_data + data(:, :, k) * dec_weights(k);
12 end
13
14 % Krok 2: Normalizacja macierzy decyzyjnej
15 norm_data = aggregated_data ./ sqrt(sum(aggregated_data.^2));
16
17 % Krok 3: Ważona macierz normalizowana
18 weighted_data = norm_data .* weights;
19
20 % Krok 4: Wyznaczenie ideałów pozytywnego i negatywnego
21 ideal_positive = max(weighted_data .* (types == 1), [], 1) - ...
22     min(weighted_data .* (types == -1), [], 1);
23 ideal_negative = min(weighted_data .* (types == 1), [], 1) - ...
24     max(weighted_data .* (types == -1), [], 1);
25
26 % Krok 5: Obliczenie odległości od ideałów
27 dist_positive = sqrt(sum((weighted_data - ideal_positive).^2, 2));
28 dist_negative = sqrt(sum((weighted_data - ideal_negative).^2, 2));
29
30 % Krok 6: Wyznaczenie miary bliskości do ideału pozytywnego
31 closeness = dist_negative ./ (dist_positive + dist_negative);
32
33 % Krok 7: Tworzenie rankingu
34 [~, ranking] = sort(closeness, 'descend');
35 end
```

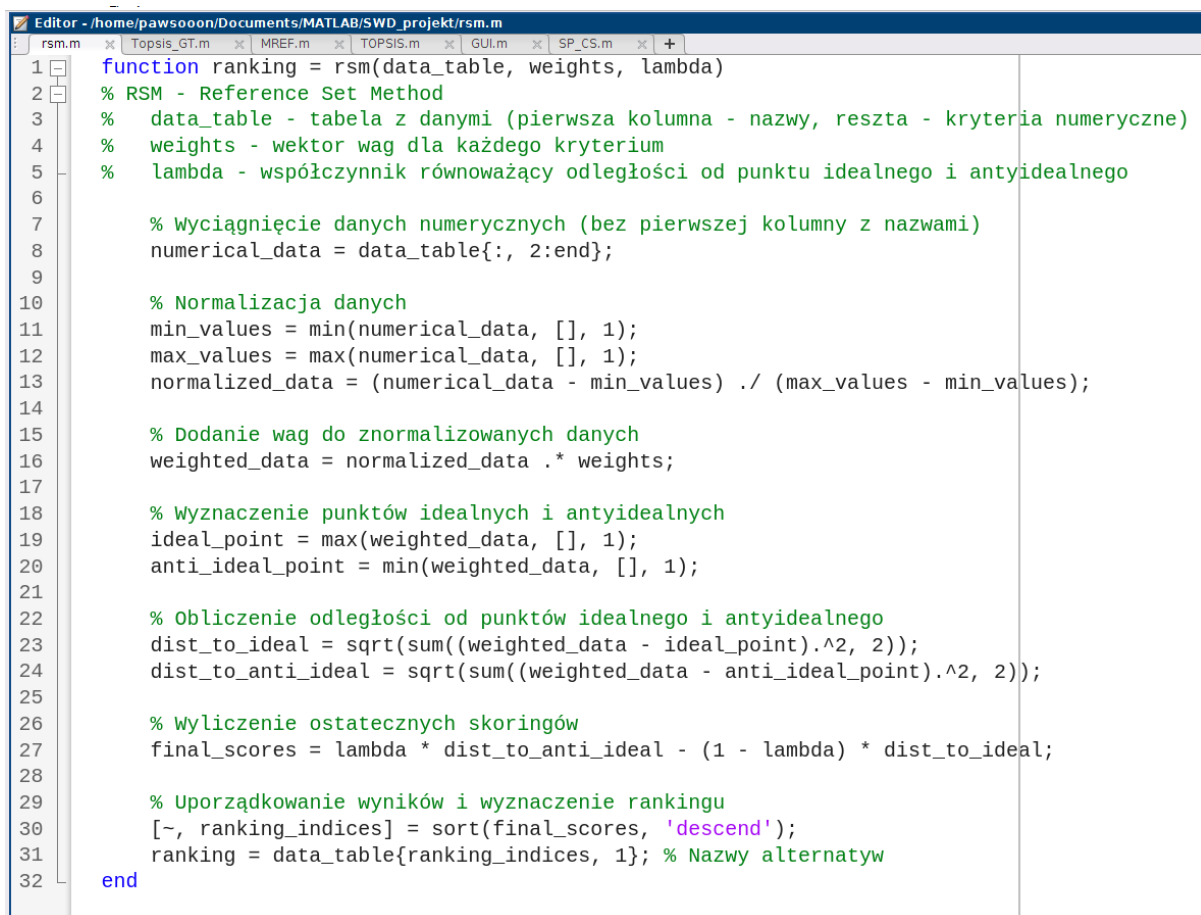
Rys.8 Algorytm TOPSIS GT

Algorytm RSM (Reference Set Method)

Algorytm RSM opiera się na koncepcji zbioru referencyjnego, który definiuje zbiór punktów reprezentujących preferowane kombinacje wartości kryteriów. Metoda ta zakłada, że punkty referencyjne są określone przez decydenta lub wyznaczone automatycznie na podstawie danych wejściowych. Głównym celem algorytmu jest zidentyfikowanie alternatyw znajdujących się najbliżej zbioru referencyjnego.

RSM działa w kilku etapach: wyznaczenia zbioru referencyjnego, normalizacji danych, obliczenia odległości alternatyw od punktów referencyjnych oraz rankingu alternatyw na podstawie minimalnych odległości. Proces ten pozwala na uwzględnienie indywidualnych preferencji decydenta, co czyni algorytm bardziej elastycznym.

Główną zaletą RSM jest zdolność do dostosowania analizy do specyficznych preferencji decydenta, co umożliwia bardziej precyzyjne podejmowanie decyzji. Jednak metoda ta może wymagać dodatkowego nakładu pracy na określenie zbioru referencyjnego oraz może być mniej skuteczna w sytuacjach, gdy preferencje są niejasne lub trudne do zdefiniowania.

The image shows a screenshot of a MATLAB script editor window titled 'Editor - /home/pawsooon/Documents/MATLAB/SWD_projekt/rsm.m'. The script defines a function 'ranking = rsm(data_table, weights, lambda)'. The comments in the script describe the steps: 1. RSM - Reference Set Method. 2. data_table - table with data (first column - names, rest - numerical criteria). 3. weights - vector of weights for each criterion. 4. lambda - coefficient balancing distances from ideal and anti-ideal points. 5. Extraction of numerical data (excluding the first column with names). 6. Normalization of data. 7. Addition of weights to normalized data. 8. Determination of ideal and anti-ideal points. 9. Calculation of distances from ideal and anti-ideal points. 10. Calculation of final scores. 11. Sorting of results and determination of ranking. The script uses MATLAB syntax with comments in Polish. The function 'rsm' takes three inputs: 'data_table', 'weights', and 'lambda'. It returns 'ranking'. The script is 32 lines long.

```
1 function ranking = rsm(data_table, weights, lambda)
2 % RSM - Reference Set Method
3 % data_table - tabela z danymi (pierwsza kolumna - nazwy, reszta - kryteria numeryczne)
4 % weights - wektor wag dla każdego kryterium
5 % lambda - współczynnik równoważący odległości od punktu idealnego i antyidealnego
6
7 % Wyciągnięcie danych numerycznych (bez pierwszej kolumny z nazwami)
8 numerical_data = data_table(:, 2:end);
9
10 % Normalizacja danych
11 min_values = min(numerical_data, [], 1);
12 max_values = max(numerical_data, [], 1);
13 normalized_data = (numerical_data - min_values) ./ (max_values - min_values);
14
15 % Dodanie wag do znormalizowanych danych
16 weighted_data = normalized_data .* weights;
17
18 % Wyznaczenie punktów idealnych i antyidealnych
19 ideal_point = max(weighted_data, [], 1);
20 anti_ideal_point = min(weighted_data, [], 1);
21
22 % Obliczenie odległości od punktów idealnego i antyidealnego
23 dist_to_ideal = sqrt(sum((weighted_data - ideal_point).^2, 2));
24 dist_to_anti_ideal = sqrt(sum((weighted_data - anti_ideal_point).^2, 2));
25
26 % Wyliczenie ostatecznych skoringów
27 final_scores = lambda * dist_to_anti_ideal - (1 - lambda) * dist_to_ideal;
28
29 % Uporządkowanie wyników i wyznaczenie rankingu
30 [~, ranking_indices] = sort(final_scores, 'descend');
31 ranking = data_table{ranking_indices, 1}; % Nazwy alternatyw
32 end
```

Rys.9 Algorytm RSM

Algorytm MREF (Multi-Criteria Reference Point Approach)

MREF jest zaawansowaną metodą wielokryterialnej analizy decyzji, która rozbudowuje koncepcję punktów referencyjnych, wprowadzając wiele punktów odniesienia reprezentujących różne aspekty preferencji decydenta. Każdy z punktów referencyjnych może odpowiadać innym scenariuszom decyzyjnym lub alternatywnym podejściom do oceny.

W MREF obliczane są odległości alternatyw od każdego z punktów referencyjnych, a następnie wyniki są agregowane w celu uzyskania globalnego rankingu alternatyw.

Zaletą MREF jest jego wszechstronność oraz zdolność do uwzględnienia różnorodnych preferencji i scenariuszy decyzyjnych. Metoda ta pozwala na bardziej kompleksową analizę, jednak może być bardziej czasochłonna i wymagać większych zasobów obliczeniowych w porównaniu z prostszymi metodami.

```
Editor - /home/pawsooon/Documents/MATLAB/SWD_projekt/MREF.m
1 function ranking = MREF(data, weights, types)
2 % MREF - metoda wielokryterialnej optymalizacji z wykorzystaniem miary Czebyszewa
3 % data - macierz alternatyw (wiersze to alternatywy, kolumny to kryteria)
4 % weights - wagi dla kryteriów (wektor kolumnowy)
5 % types - typy kryteriów (1 = maksymalizacja, -1 = minimalizacja)
6 % ranking - ranking alternatyw od najlepszej do najgorszej
7
8 % Sprawdzanie poprawności danych wejściowych
9 if size(data, 2) ~= length(weights) || length(weights) ~= length(types)
10 error('Rozmiar wag i typów kryteriów musi być zgodny z liczbą kolumn w macierzy danych.');
```

```
11 end
12
13 % Punkty referencyjne (domyślnie: minimum, maksimum, mediany w każdej kolumnie)
14 references = [min(data); max(data); median(data)];
15
16 % Normalizacja kryteriów
17 [normalized, normalized_ref] = normalize_criteria(data, types, references);
18
19 % Obliczanie odległości Czebyszewa
20 distances = Czebyszew(normalized, normalized_ref);
21
22 % Minimalna odległość względem dowolnego punktu referencyjnego
23 min_distances = min(distances, [], 2);
24
25 % Ranking alternatyw (od najlepszej do najgorszej)
26 [~, ranking_indices] = sort(min_distances);
27
28 % Zwracanie wynikowego rankingu
29 ranking = ranking_indices;
30 end
31
32 function [normalized, normalized_ref] = normalize_criteria(alternatives, criteria, references)
33 % Normalizacja kryteriów dla alternatyw i punktów referencyjnych
34 % alternatives - macierz alternatyw
35 % criteria - typy kryteriów (-1 - min, 1 - max)
36 % references - punkty referencyjne
37
38 normalized = zeros(size(alternatives));
39 normalized_ref = zeros(size(references));
40
41 for i = 1:length(criteria)
42 if criteria(i) == -1 % Minimalizacja
43 normalized(:, i) = (alternatives(:, i) - min(alternatives(:, i))) / (max(alternatives(:, i)) - min(alternatives(:, i)));
44 normalized_ref(:, i) = (references(:, i) - min(alternatives(:, i))) / (max(alternatives(:, i)) - min(alternatives(:, i)));
45 elseif criteria(i) == 1 % Maksymalizacja
46 normalized(:, i) = (max(alternatives(:, i)) - alternatives(:, i)) / (max(alternatives(:, i)) - min(alternatives(:, i)));
47 normalized_ref(:, i) = (max(alternatives(:, i)) - references(:, i)) / (max(alternatives(:, i)) - min(alternatives(:, i)));
48 else
49 error('Nieprawidłowy typ kryterium. Użyj 0 (minimalizacja) lub 1 (maksymalizacja).');
```

```
50 end
51 end
52 end
53
54 function distances = Czebyszew(normalized, normalized_ref)
55 % Obliczanie odległości Czebyszewa
56 % normalized - znormalizowane alternatywy
57 % normalized_ref - znormalizowane punkty referencyjne
58
59 num_alternatives = size(normalized, 1);
60 num_references = size(normalized_ref, 1);
61 distances = zeros(num_alternatives, num_references);
62
63 for i = 1:num_alternatives
64 for j = 1:num_references
65 distances(i, j) = max(abs(normalized(i, :) - normalized_ref(j, :)));
66 end
67 end
68 end
69
```

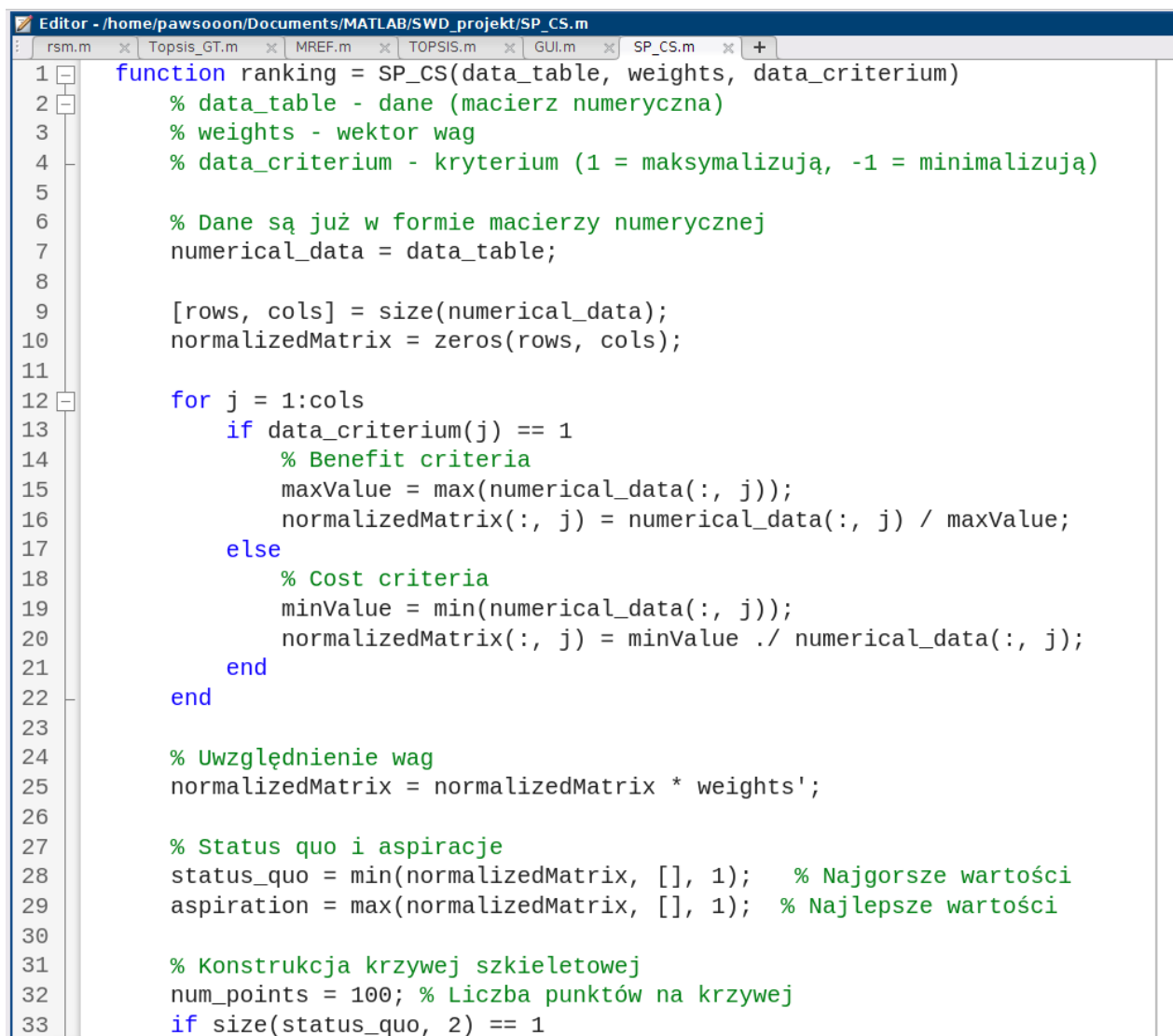
Rys.10 Algorytm MREF

Algorytm SP-CS (Set Partitioning with Constraint Satisfaction)

SP-CS jest zaawansowanym algorytmem wielokryterialnej analizy decyzyjnej, który łączy metody podziału zbiorów (set partitioning) z technikami satysfakcji ograniczeń (constraint satisfaction). Głównym celem algorytmu jest znalezienie optymalnego podziału alternatyw na zbiory niedominowane przy jednoczesnym spełnieniu określonych ograniczeń.

Proces SP-CS obejmuje identyfikację ograniczeń decyzyjnych, modelowanie problemu jako zadania optymalizacyjnego oraz iteracyjne rozwiązywanie problemu przy użyciu zaawansowanych metod matematycznych, takich jak programowanie liniowe czy algorytmy heurystyczne. Algorytm pozwala na uwzględnienie złożonych zależności między kryteriami oraz preferencji decydenta.

SP-CS wyróżnia się zdolnością do rozwiązywania bardzo skomplikowanych problemów decyzyjnych, co czyni go szczególnie przydatnym w sytuacjach wymagających uwzględnienia wielu ograniczeń. Jego główną wadą jest jednak wysoka złożoność obliczeniowa, co może ograniczać jego zastosowanie w przypadku bardzo dużych zbiorów danych.



```
Editor - /home/pawsooon/Documents/MATLAB/SWD_projekt/SP_CS.m
rsm.m  Topsis_GT.m  MREF.m  TOPSIS.m  GUI.m  SP_CS.m  +
1  function ranking = SP_CS(data_table, weights, data_criterium)
2      % data_table - dane (macierz numeryczna)
3      % weights - wektor wag
4      % data_criterium - kryterium (1 = maksymalizują, -1 = minimalizują)
5
6      % Dane są już w formie macierzy numerycznej
7      numerical_data = data_table;
8
9      [rows, cols] = size(numerical_data);
10     normalizedMatrix = zeros(rows, cols);
11
12     for j = 1:cols
13         if data_criterium(j) == 1
14             % Benefit criteria
15             maxValue = max(numerical_data(:, j));
16             normalizedMatrix(:, j) = numerical_data(:, j) / maxValue;
17         else
18             % Cost criteria
19             minValue = min(numerical_data(:, j));
20             normalizedMatrix(:, j) = minValue ./ numerical_data(:, j);
21         end
22     end
23
24     % Uwzględnienie wag
25     normalizedMatrix = normalizedMatrix * weights';
26
27     % Status quo i aspiracje
28     status_quo = min(normalizedMatrix, [], 1); % Najgorsze wartości
29     aspiration = max(normalizedMatrix, [], 1); % Najlepsze wartości
30
31     % Konstrukcja krzywej szkieletowej
32     num_points = 100; % Liczba punktów na krzywej
33     if size(status_quo, 2) == 1
```

```

34         % Jeśli mamy tylko jedno kryterium
35         gamma = linspace(status_quo(1), aspiration(1), num_points)';
36     else
37         % Dla wielu kryteriów
38         gamma = [linspace(status_quo(1), aspiration(1), num_points)', ...
39                 linspace(status_quo(2), aspiration(2), num_points)'];
40     end
41
42     % Obliczanie funkcji skoringowej
43     scores = zeros(rows, 1);
44     for i = 1:rows
45         point = normalizedMatrix(i, :);
46         distances = max(abs(gamma - point), [], 2); % Metryka Czebyszewa
47         [min_distance, idx] = min(distances);
48         t = idx / num_points; % Parametr krzywej
49         scores(i) = min_distance + t; % Skoring
50     end
51
52     % Ranking alternatyw
53     [~, ranking] = sort(scores, 'descend');
54 end

```

Rys.11 Algorytm SP-CS

Zbiory Danych

Dla aplikacji stworzono zbiory danych, które odwzorowują realne problemy decyzyjne, umożliwiając analizę wielokryterialną w różnych kontekstach. Dane te zostały zaprojektowane w sposób, który odzwierciedla typowe wyzwania podejmowania decyzji w codziennych sytuacjach. Dzięki temu aplikacja może być testowana na przykładach zbliżonych do rzeczywistych, co pozwala na ocenę jej skuteczności i wszechstronności w rozwiązywaniu problemów praktycznych

Plik **data_cars** zawiera informacje o samochodach, które mogą być analizowane pod kątem różnych kryteriów decyzyjnych. Znajduje się tam lista modeli samochodów wraz z ich cechami, takimi jak cena, spalanie, liczba koni mechanicznych, rok produkcji oraz przebieg. Te dane stanowią alternatywy, które mogą być oceniane i porównywane.

Plik **data_footbalers** to zbiór danych, który dotyczy piłkarzy i ich atrybutów związanych z grą. Pierwszy arkusz zawiera informacje o graczach, takie jak ich tempo, strzały, podania, drybling, obrona, siła fizyczna oraz wartość rynkowa. Są to dane wejściowe, które mogą być oceniane w procesie analizy.

Zbiór danych **data_phones** zawiera informacje o telefonach i ich specyfikacjach. Dla każdego modelu telefonu podano następujące cechy: cena, pojemność baterii, wielkość ekranu (w calach), ilość pamięci RAM oraz rozdzielczość aparatu (w megapikselach).

Zbiór danych **data_gym** zawiera informacje o siłowniach, które są oceniane na podstawie kilku kryteriów. Dla każdej siłowni podano jej odległość (w kilometrach), wielkość (w metrach kwadratowych), gęstość ruchu (liczba osób na 5 m²) oraz cenę miesięcznego abonamentu (w złotych). Dane te mogą służyć do analizy porównawczej siłowni w celu wyboru najodpowiedniejszej oferty.

Badanie i Porównywanie Metod:

Po zaimplementowaniu algorytmów oraz interfejsu użytkownika przystąpiono do głównej części zadania, którą było porównanie metod.

Zbiór danych dotyczących samochodów, poniżej przedstawiono wyniki dla metod **MREF** oraz **TOPSIS**:

Wczytaj dane z pliku

TOPSIS

Stwórz ranking

Alternatywy z kryteriami

| | 1 | 2 | 3 | 4 |
|---|-------------|-------|---|-----|
| 4 | Kia Ceed | 40000 | 8 | 150 |
| 5 | Honda Civic | 39000 | 8 | 150 |
| 6 | Mazda 3 | 45000 | 7 | 140 |
| 7 | BMW 320i | 47000 | 9 | 180 |
| 8 | Volkswag... | 44000 | 7 | 150 |
| 9 | Opel Astra | 38000 | 7 | 130 |

Klasy

| | 1 | 2 | 3 | 4 | 5 |
|---|--------------|-------|---|-----|----|
| 1 | Toyota Co... | 40000 | 7 | 150 | 20 |
| 2 | Ford Focus | 38000 | 6 | 150 | 20 |
| 3 | Volkswag... | 44000 | 7 | 160 | 20 |

Stworzony Ranking

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|--------------|-------|---|-----|------|-------|
| 1 | Mazda 3 | 45000 | 7 | 140 | 2020 | 37000 |
| 2 | Opel Astra | 38000 | 7 | 130 | 2020 | 40000 |
| 3 | Kia Ceed | 40000 | 8 | 150 | 2022 | 41000 |
| 4 | Audi A3 | 41000 | 6 | 140 | 2019 | 49000 |
| 5 | Honda Civic | 39000 | 8 | 150 | 2021 | 55000 |
| 6 | Volkswag... | 37000 | 7 | 130 | 2020 | 67000 |
| 7 | BMW 320i | 47000 | 9 | 180 | 2019 | 67000 |
| 8 | Toyota Co... | 42000 | 7 | 140 | 2019 | 74000 |
| 9 | Skoda Oc... | 44000 | 6 | 140 | 2018 | 87000 |

Porównaj metody

Rys.12 Wyniki wyboru samochodu metodą TOPSIS

Wczytaj dane z pliku

Metoda wielu punktów odniesienia (MREF)

Stwórz ranking

Alternatywy z kryteriami

| | 1 | 2 | 3 | 4 |
|---|-------------|-------|---|-----|
| 4 | Kia Ceed | 40000 | 8 | 150 |
| 5 | Honda Civic | 39000 | 8 | 150 |
| 6 | Mazda 3 | 45000 | 7 | 140 |
| 7 | BMW 320i | 47000 | 9 | 180 |
| 8 | Volkswag... | 44000 | 7 | 150 |
| 9 | Opel Astra | 38000 | 7 | 130 |

Klasy

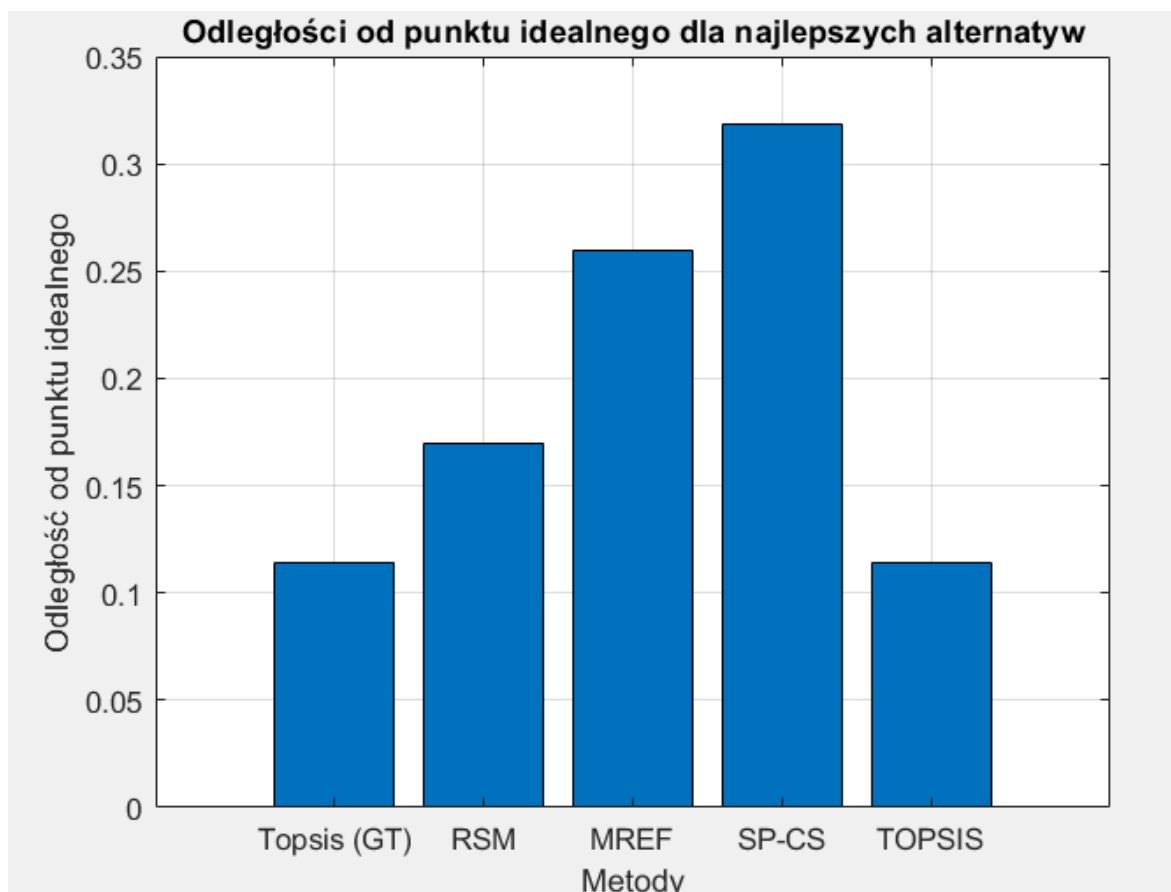
| | 1 | 2 | 3 | 4 | 5 |
|---|--------------|-------|---|-----|----|
| 1 | Toyota Co... | 40000 | 7 | 150 | 20 |
| 2 | Ford Focus | 38000 | 6 | 150 | 20 |
| 3 | Volkswag... | 44000 | 7 | 160 | 20 |

Stworzony Ranking

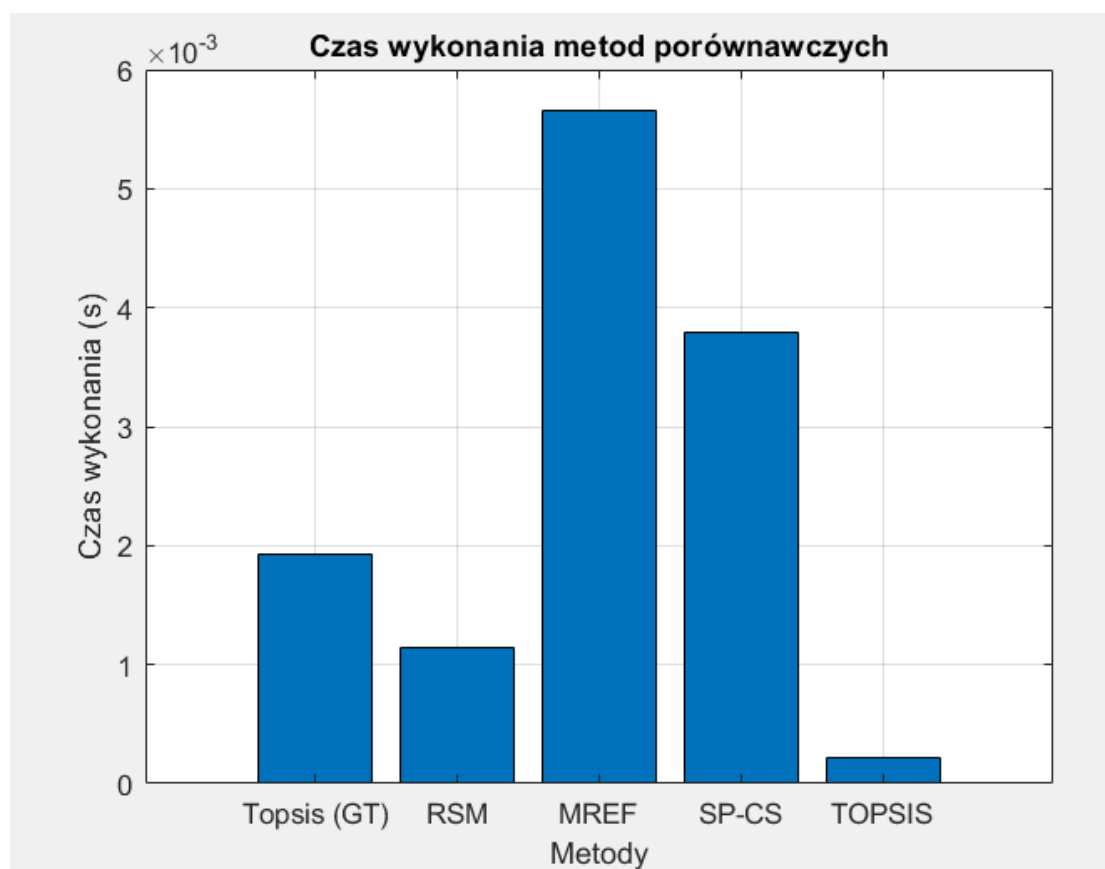
| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|--------------|-------|---|-----|------|-------|
| 1 | Toyota Co... | 42000 | 7 | 140 | 2019 | 74000 |
| 2 | Audi A3 | 41000 | 6 | 140 | 2019 | 49000 |
| 3 | Honda Civic | 39000 | 8 | 150 | 2021 | 55000 |
| 4 | Opel Astra | 38000 | 7 | 130 | 2020 | 40000 |
| 5 | Volkswag... | 44000 | 7 | 150 | 2018 | 82000 |
| 6 | Volkswag... | 37000 | 7 | 130 | 2020 | 67000 |
| 7 | Mazda 3 | 45000 | 7 | 140 | 2020 | 37000 |
| 8 | Skoda Oc... | 44000 | 6 | 140 | 2018 | 87000 |
| 9 | Kia Ceed | 40000 | 8 | 150 | 2022 | 41000 |

Porównaj metody

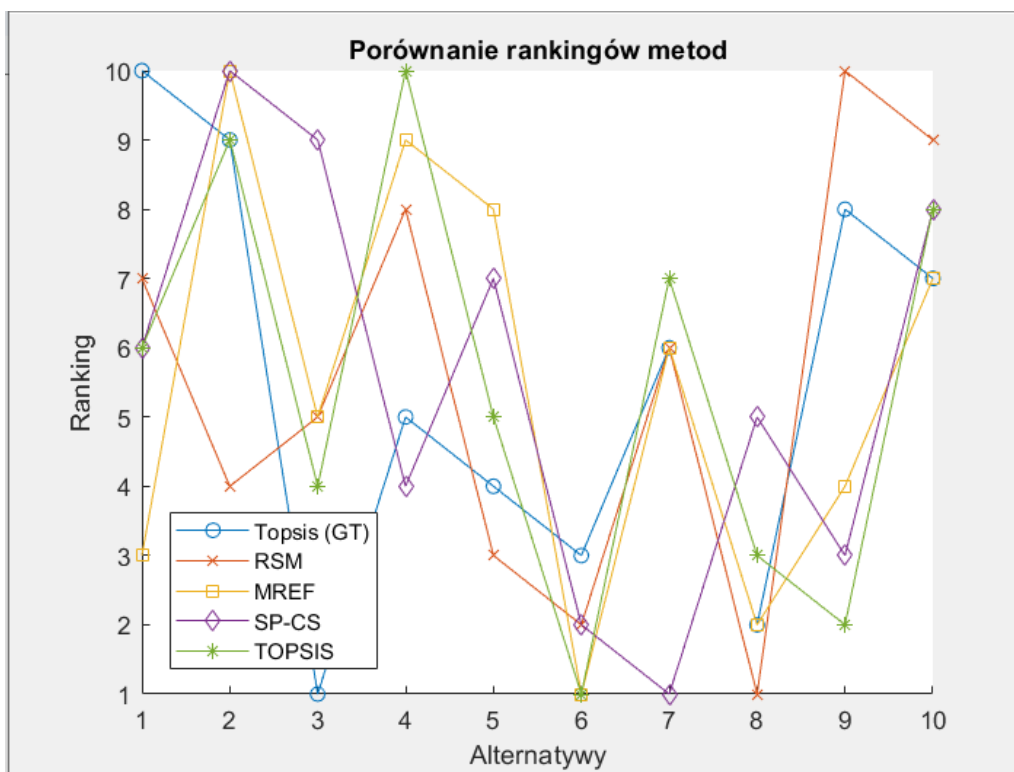
Rys.13 Wyniki wyboru samochodu metodą MREF



Rys.14 Wybór samochodu odległość od najlepszych alternatyw



Rys.15 Wybór samochodu czas wykonywania metod



Rys.16 Wybór samochodu rankingi metod

Wykresy przedstawiają wyniki działania programu, który implementuje różne algorytmy wielokryterialnej analizy decyzyjnej, takie jak TOPSIS, MREF, SP-CS, czy Topsis (GT). Pierwszy wykres pokazuje odległości od punktu idealnego dla najlepszej alternatywy wyznaczonej przez każdą z metod. Najmniejszą odległość osiągnęły algorytmy Topsis (GT) i klasyczny TOPSIS, co świadczy o ich wysokiej precyzji w identyfikacji najlepszej alternatywy w odniesieniu do punktu wzorcowego. Z kolei metoda SP-CS osiągnęła najwyższą odległość, co może wskazywać na jej większe rozbieżności w optymalizacji.

Drugi wykres porównuje czas wykonania poszczególnych algorytmów. Najbardziej czasochłonny okazał się algorytm MREF, co sugeruje jego większą złożoność obliczeniową. Metody SP-CS również wymagały stosunkowo dużo czasu. Z drugiej strony, Topsis (GT) i klasyczny TOPSIS charakteryzowały się najszybszym działaniem, co czyni je bardziej efektywnymi pod względem wydajności.

Trzeci wykres prezentuje porównanie rankingów alternatyw wygenerowanych przez różne algorytmy. Widać, że choć niektóre metody, jak Topsis (GT) i klasyczny TOPSIS, generują podobne wyniki, to inne, takie jak SP-CS czy MREF, wprowadzają większe rozbieżności w swoich rankingach. Świadczy to o różnicach w sposobie przetwarzania kryteriów i wag przez te algorytmy.

Analiza tych wykresów pozwala stwierdzić, że Topsis (GT) i klasyczny TOPSIS wyróżniają się zarówno pod względem szybkości działania, jak i dokładności, co czyni je bardzo praktycznym wyborem w zastosowaniach wymagających szybkich i precyzyjnych decyzji. MREF, mimo większej złożoności i czasu obliczeń, może być przydatny w bardziej specyficznych

przypadkach, gdzie dokładność ma nadrzędne znaczenie. SP-CS, choć bardziej czasochłonny, może wprowadzać ciekawe perspektywy przy ocenie alternatyw, ale jego zastosowanie może być ograniczone przez większe odległości od punktu idealnego. Każdy z algorytmów oferuje różne zalety w zależności od priorytetów i kontekstu decyzji.

Poniżej przedstawiono wyniki dla metod **RSM oraz TOPSIS**. Zbiór danych dotyczących piłkarzy. Dokonano również analizy porównawczej ze wszystkimi pozostałymi metodami.

Wczytaj dane z pliku

TOPSIS

Stwórz ranking

Alternatywy z kryteriami

Klasy

1
2
3
4

| | | | | |
|---|----------|----|----|----|
| 1 | Mbippi | 97 | 90 | 80 |
| 2 | Henry | 93 | 90 | 82 |
| 3 | Vinicius | 95 | 84 | 81 |
| 4 | Haaland | 88 | 92 | 70 |
| 5 | Kane | 65 | 93 | 84 |
| 6 | Cruyff | 90 | 91 | 90 |

1
2
3
4
5

| | | | | |
|---|----|----|----|----|
| 1 | 88 | 92 | 70 | 81 |
| 2 | 65 | 93 | 84 | 83 |
| 3 | 91 | 85 | 80 | 90 |
| 4 | 97 | 90 | 80 | 92 |
| 5 | 93 | 90 | 82 | 89 |
| 6 | 95 | 84 | 81 | 91 |

Stworzony Ranking

Porównaj metody

1
2
3
4
5
6

| | | | | | | |
|---|-----------|----|----|----|----|----|
| 1 | Smolarek | 91 | 85 | 80 | 90 | 41 |
| 2 | Haaland | 88 | 92 | 70 | 81 | 45 |
| 3 | Vinicius | 95 | 84 | 81 | 91 | 29 |
| 4 | G.Muller | 87 | 92 | 76 | 85 | 44 |
| 5 | C.Ronaldo | 84 | 90 | 80 | 86 | 36 |
| 6 | Best | 91 | 89 | 82 | 92 | 56 |
| 7 | Henry | 93 | 90 | 82 | 89 | 51 |
| 8 | Messi | 86 | 89 | 92 | 95 | 38 |

Rys.17 Wybór piłkarza do drużyny metodą TOPSIS

Wczytaj dane z pliku

Metoda zbiorów odniesienia (RSM)

Stwórz ranking

Alternatywy z kryteriami

Klasy

1
2
3
4

| | | | | |
|---|----------|----|----|----|
| 1 | Mbippi | 97 | 90 | 80 |
| 2 | Henry | 93 | 90 | 82 |
| 3 | Vinicius | 95 | 84 | 81 |
| 4 | Haaland | 88 | 92 | 70 |
| 5 | Kane | 65 | 93 | 84 |
| 6 | Cruyff | 90 | 91 | 90 |

1
2
3
4
5

| | | | | |
|---|----|----|----|----|
| 1 | 88 | 92 | 70 | 81 |
| 2 | 65 | 93 | 84 | 83 |
| 3 | 91 | 85 | 80 | 90 |
| 4 | 97 | 90 | 80 | 92 |
| 5 | 93 | 90 | 82 | 89 |
| 6 | 95 | 84 | 81 | 91 |

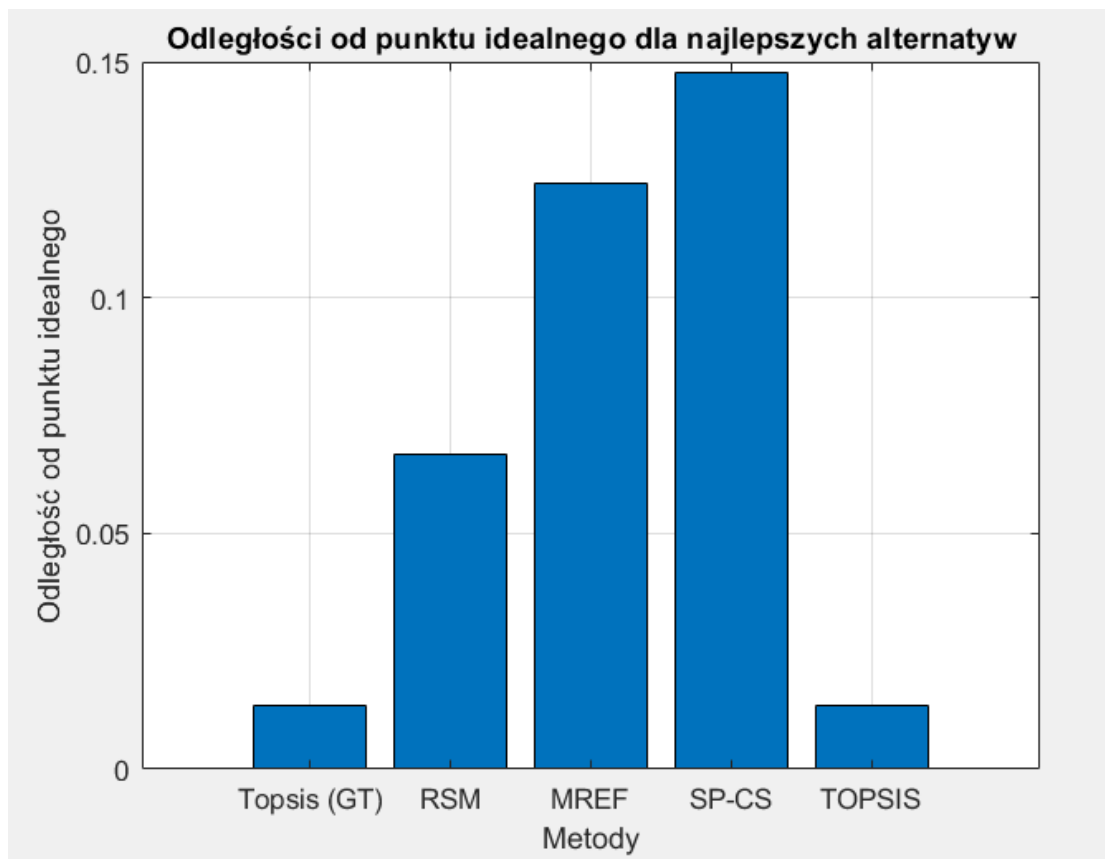
Stworzony Ranking

Porównaj metody

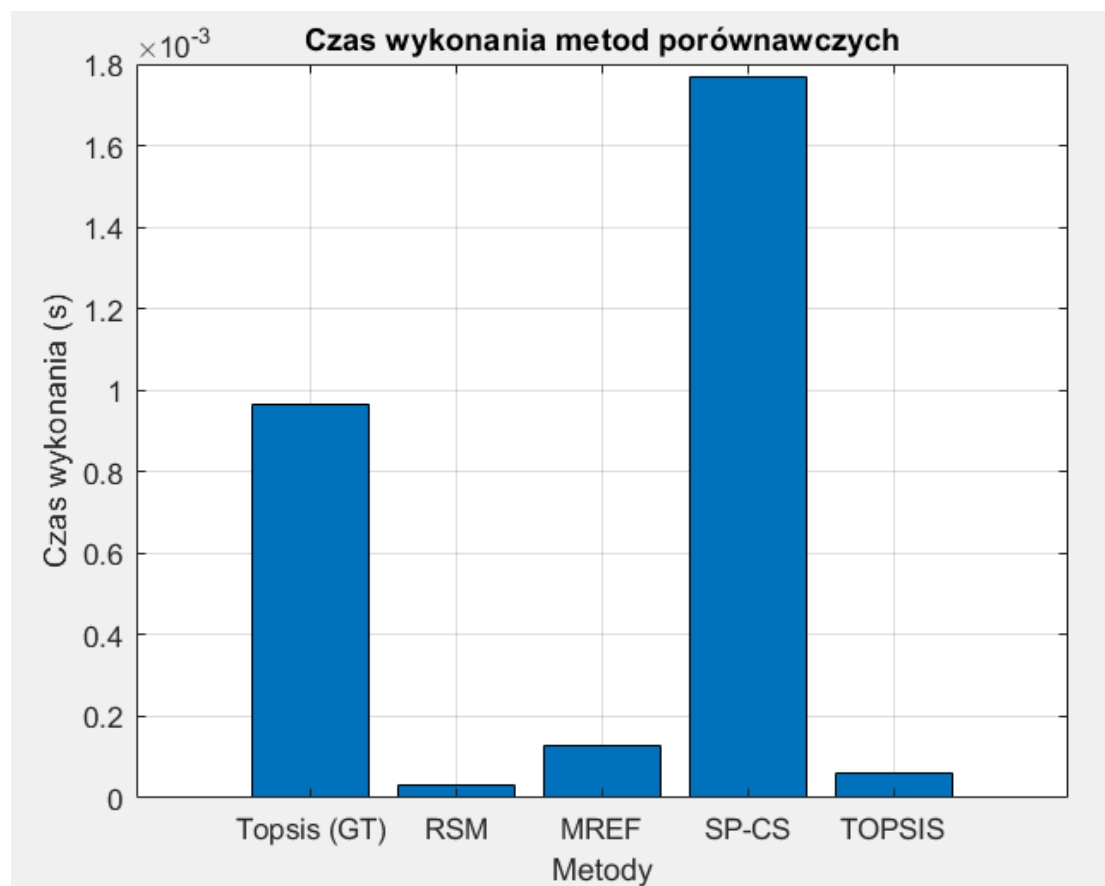
1
2
3
4
5
6

| | | | | | | |
|---|------------|----|----|----|----|----|
| 1 | Ronaldinho | 92 | 90 | 91 | 96 | 40 |
| 2 | Ronaldo | 94 | 94 | 79 | 94 | 43 |
| 3 | Mbippi | 97 | 90 | 80 | 92 | 36 |
| 4 | Pele | 93 | 94 | 91 | 94 | 58 |
| 5 | Zidane | 83 | 90 | 94 | 94 | 73 |
| 6 | Henry | 93 | 90 | 82 | 89 | 51 |
| 7 | Smolarek | 91 | 85 | 80 | 90 | 41 |
| 8 | Cruyff | 90 | 91 | 90 | 93 | 42 |

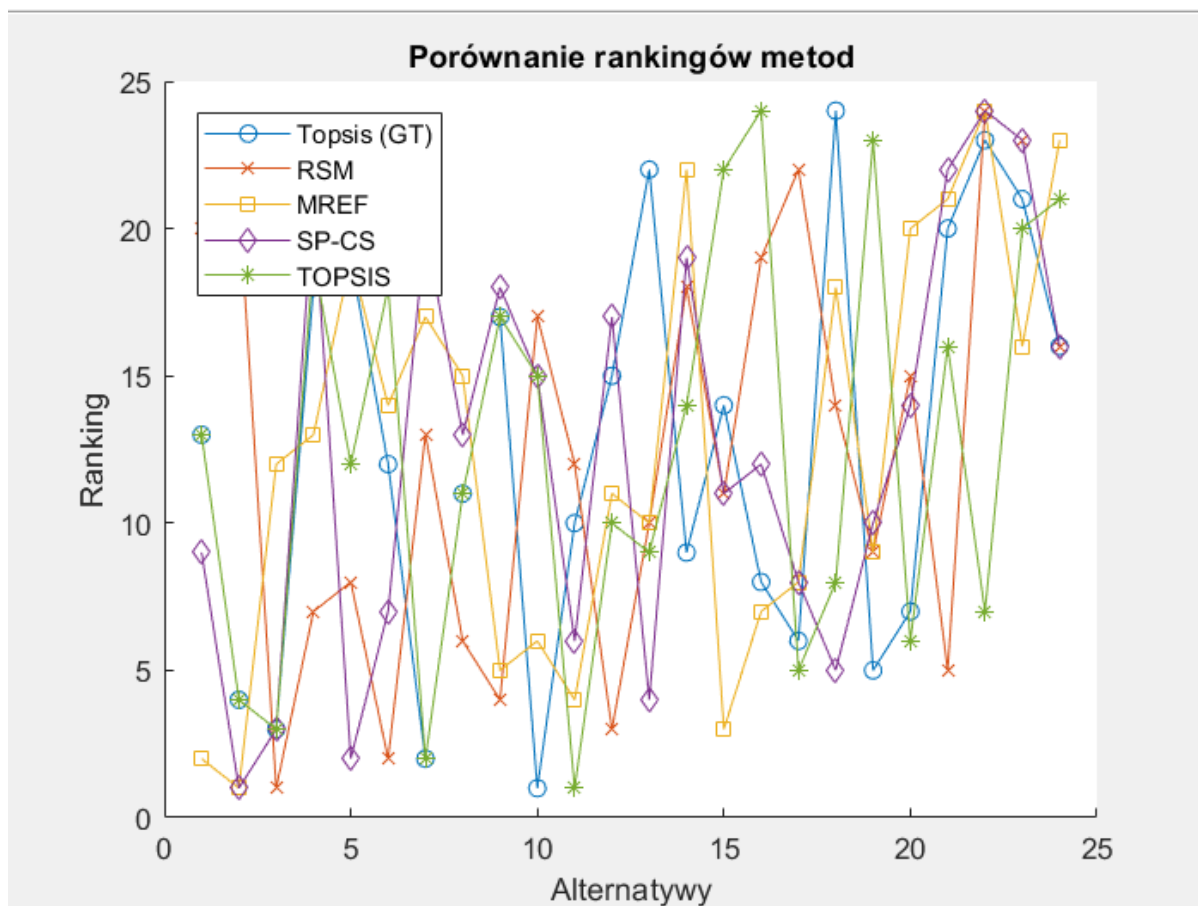
Rys.18 Wybór piłkarza do drużyny metodą RSM



Rys.19 Wybór piłkarza odległość od najlepszych alternatyw



Rys.20 Wybór piłkarza czas wykonywania metod



Rys.21 Wybór piłkarza rankingi metod

Porównanie algorytmów na zbiorze danych piłkarzy, w szczególności TOPSIS i RSM, wykazuje różnice zarówno w wynikowych rankingach, jak i podejściu do oceny alternatyw.

Metoda TOPSIS klasyfikuje piłkarzy na podstawie odległości od punktu idealnego i anty-idealnego, dzięki czemu uzyskane wyniki odzwierciedlają kompromis między różnymi kryteriami. W rankingu TOPSIS, na pierwszych miejscach pojawiają się gracze tacy jak Smolarek, Haaland i Vinicius. Jest to spójne z ich wysokimi wynikami w kluczowych atrybutach, takich jak tempo i strzały, które mają duże znaczenie dla ogólnej oceny. TOPSIS charakteryzuje się szybkim czasem działania i jasnym sposobem ustalania hierarchii, co czyni go praktycznym rozwiązaniem w analizach.

Z kolei metoda RSM opiera się na odniesieniach do zbioru wzorcowych punktów, co pozwala na bardziej złożoną ocenę alternatyw. W rankingu RSM na czołowych miejscach znajdują się Ronaldinho, Ronaldo i Mbipipi, co wskazuje na uwzględnienie dodatkowych aspektów, które mogłyby być pominięte w prostszym podejściu TOPSIS. RSM uwzględnia różnorodne kryteria w bardziej dynamiczny sposób, co prowadzi do nieco innych wyników, choć kosztem większej złożoności obliczeń i dłuższego czasu wykonania.

Pozostałe metody, takie jak MREF i SP-CS, również generują różne rankingi, ale w porównaniu do TOPSIS i RSM wydają się mniej intuicyjne w interpretacji wyników. MREF klasyfikuje piłkarzy z większym naciskiem na odległości od wielu punktów odniesienia, co

może prowadzić do bardziej niestabilnych wyników, podczas gdy SP-CS jest bardziej złożony, lecz często generuje rankingi mniej zgodne z oczekiwaniami użytkownika.

TOPSIS okazuje się bardziej przejrzystą i szybszą metodą, szczególnie w sytuacjach, gdzie priorytetem jest szybkość i prostota. RSM, choć bardziej czasochłonny, może dostarczyć bardziej zróżnicowaną ocenę, szczególnie jeśli dane alternatywy mają nietypowe rozkłady cech. Wybór odpowiedniej metody powinien być uzależniony od specyfiki problemu i dostępnego czasu na analizę.

Wyniki dla metod **TOPSIS_GT** oraz **TOPSIS**. Zbiór danych dotyczący telefonów. Dokonano również analizy porównawczej ze wszystkimi pozostałymi metodami.

Wczytaj dane z pliku

TOPSIS

Stwórz ranking

Alternatywy z kryteriami

| | 1 | 2 | 3 | 4 |
|---|-----------|------|------|--------|
| 1 | Telefon A | 2000 | 5000 | 6.5000 |
| 2 | Telefon B | 2500 | 4500 | 6.1000 |
| 3 | Telefon C | 1500 | 4000 | 5.8000 |
| 4 | Telefon D | 3000 | 6000 | 6.7000 |
| 5 | Telefon E | 1800 | 4200 | 5.9000 |
| 6 | Telefon F | 2700 | 4800 | 6.3000 |

Klasy

| | 1 | 2 | 3 | 4 | 5 |
|---|---------|------|------|--------|---|
| 1 | Punkt 1 | 2100 | 4800 | 6.4000 | |
| 2 | Punkt 2 | 2400 | 4600 | | 6 |
| 3 | Punkt 3 | 1600 | 4100 | 5.9000 | |
| 4 | Punkt 4 | 3100 | 5900 | 6.8000 | |
| 5 | Punkt 5 | 2200 | 5200 | 6.3000 | |
| 6 | Punkt 6 | 2300 | 4300 | 6.2000 | |

Stworzony Ranking

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|-----------|------|------|--------|---|-----|
| 1 | Telefon A | 2000 | 5000 | 6.5000 | 6 | 48 |
| 2 | Telefon D | 3000 | 6000 | 6.7000 | 8 | 108 |
| 3 | Telefon H | 3200 | 6200 | 6.8000 | 8 | 120 |
| 4 | Telefon J | 2100 | 4600 | 6.2000 | 5 | 48 |
| 5 | Telefon I | 2300 | 4700 | 6.4000 | 6 | 50 |
| 6 | Telefon C | 1500 | 4000 | 5.8000 | 3 | 8 |
| 7 | Telefon G | 1600 | 4100 | 6 | 3 | 10 |
| 8 | Telefon E | 1800 | 4200 | 5.9000 | 4 | 16 |
| 9 | Telefon F | 2700 | 4800 | 6.3000 | 5 | 64 |

Porównaj metody

Rys.22 Wybór telefonu metodą TOPSIS

Wczytaj dane z pliku

Topsis (GT) ▾

Stwórz ranking

Alternatywy z kryteriami

| | 1 | 2 | 3 | 4 |
|---|-----------|------|------|--------|
| 1 | Telefon A | 2000 | 5000 | 6.5000 |
| 2 | Telefon B | 2500 | 4500 | 6.1000 |
| 3 | Telefon C | 1500 | 4000 | 5.8000 |
| 4 | Telefon D | 3000 | 6000 | 6.7000 |
| 5 | Telefon E | 1800 | 4200 | 5.9000 |
| 6 | Telefon F | 2700 | 4800 | 6.3000 |

Klasy

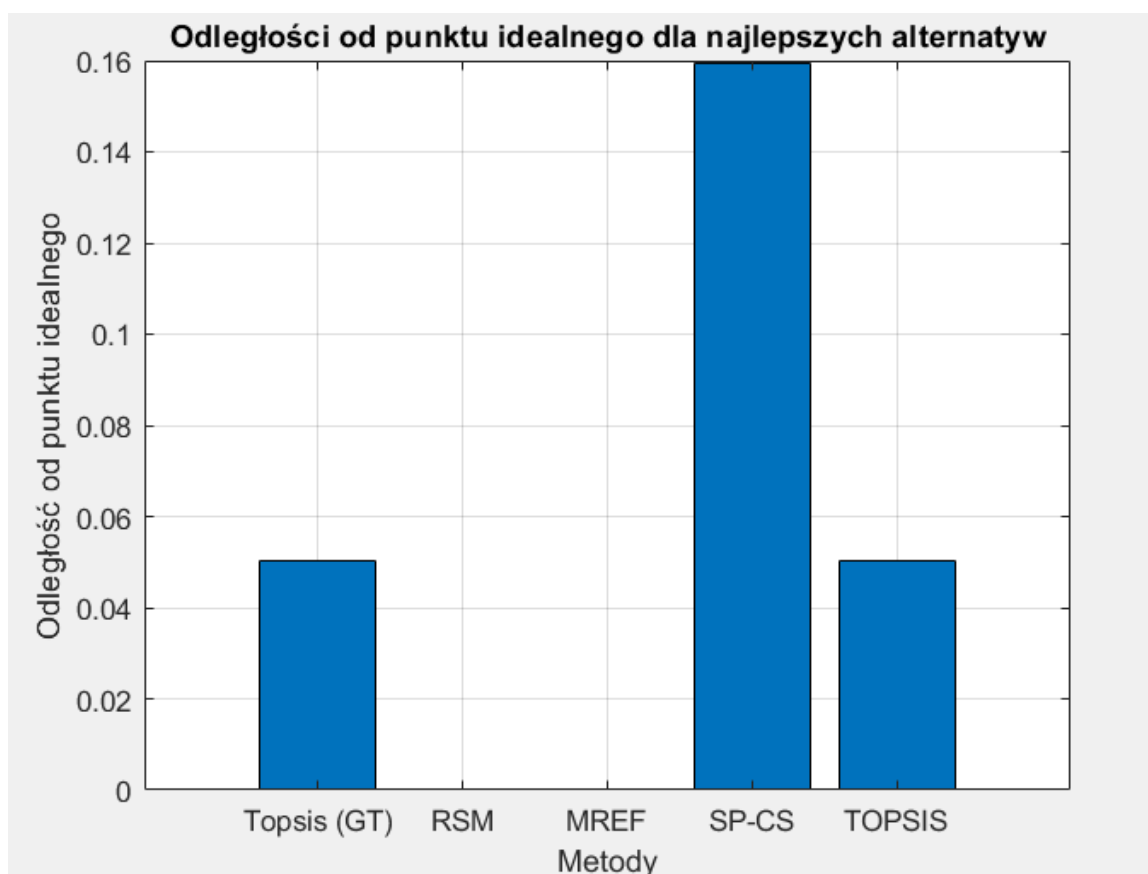
| | 1 | 2 | 3 | 4 | 5 |
|---|---------|------|------|--------|---|
| 1 | Punkt 1 | 2100 | 4800 | 6.4000 | |
| 2 | Punkt 2 | 2400 | 4600 | 6 | |
| 3 | Punkt 3 | 1600 | 4100 | 5.9000 | |
| 4 | Punkt 4 | 3100 | 5900 | 6.8000 | |
| 5 | Punkt 5 | 2200 | 5200 | 6.3000 | |
| 6 | Punkt 6 | 2300 | 4300 | 6.2000 | |

Stworzony Ranking

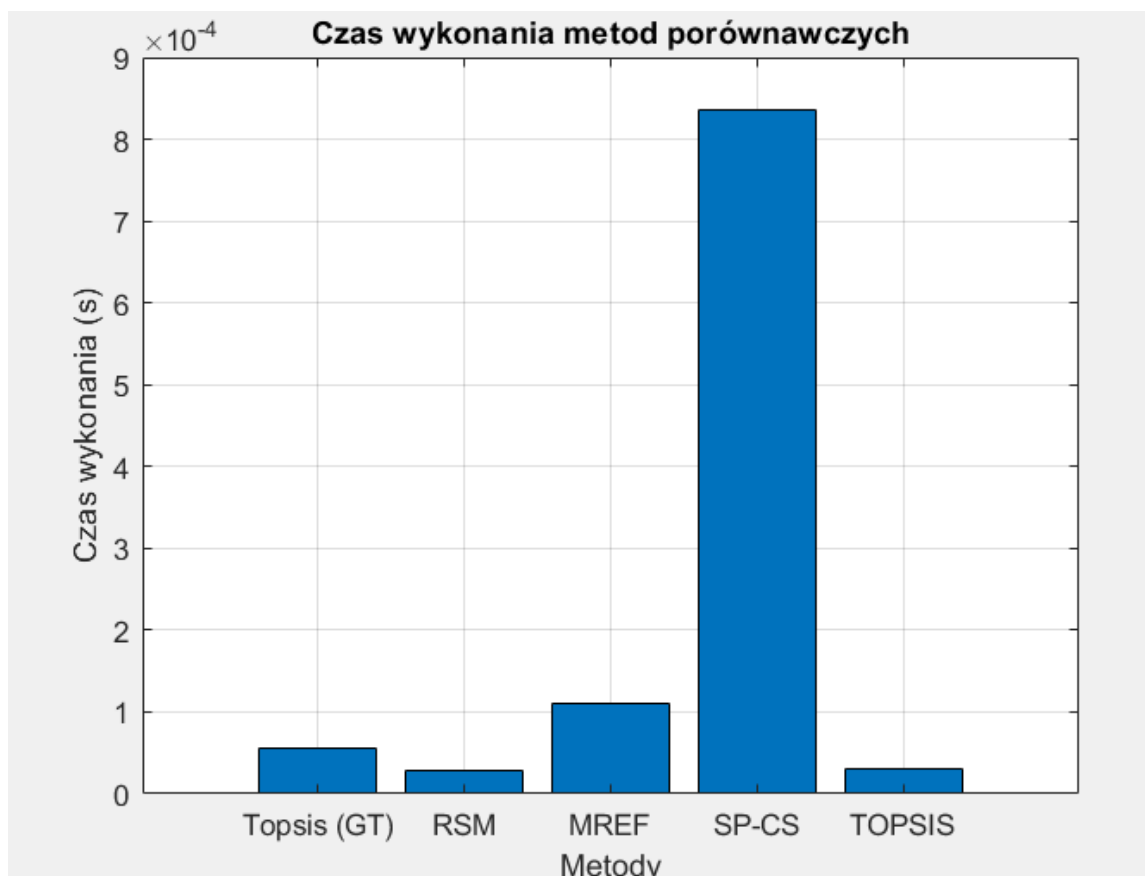
| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|-----------|------|------|--------|---|-----|
| 1 | Telefon A | 2000 | 5000 | 6.5000 | 6 | 48 |
| 2 | Telefon D | 3000 | 6000 | 6.7000 | 8 | 108 |
| 3 | Telefon H | 3200 | 6200 | 6.8000 | 8 | 120 |
| 4 | Telefon J | 2100 | 4600 | 6.2000 | 5 | 48 |
| 5 | Telefon I | 2300 | 4700 | 6.4000 | 6 | 50 |
| 6 | Telefon C | 1500 | 4000 | 5.8000 | 3 | 8 |
| 7 | Telefon G | 1600 | 4100 | 6 | 3 | 10 |
| 8 | Telefon E | 1800 | 4200 | 5.9000 | 4 | 16 |
| 9 | Telefon F | 2700 | 4800 | 6.3000 | 5 | 64 |

Porównaj metody

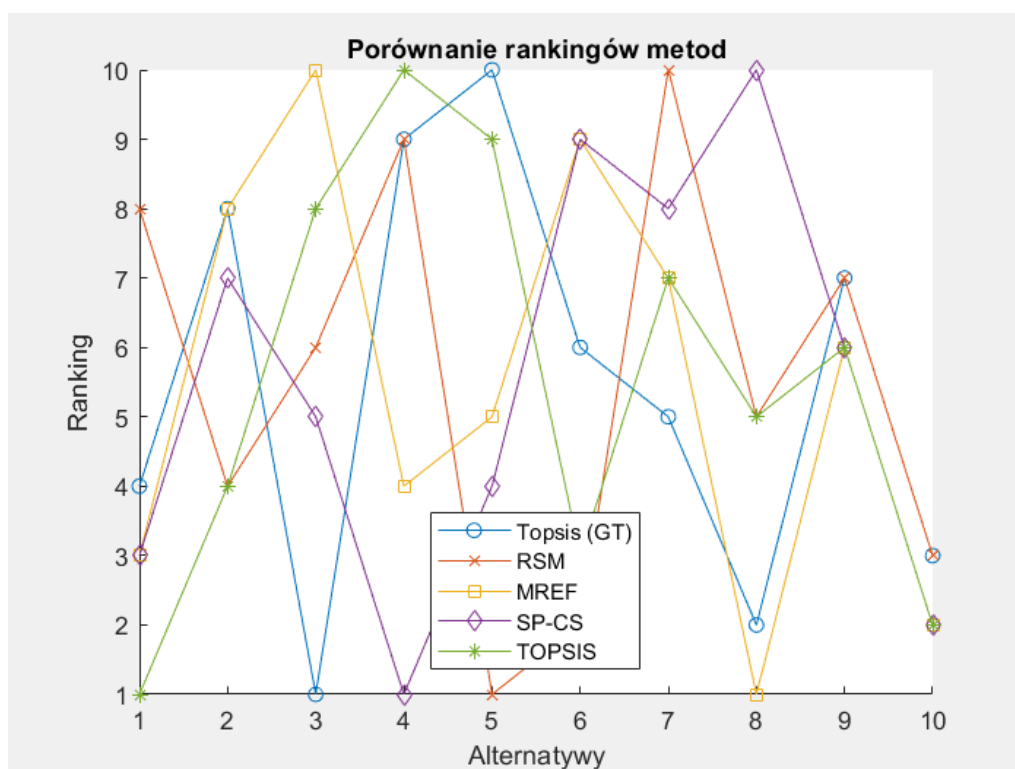
Rys.23 Wybór telefonu metodą TOPSIS_GT



Rys.24 Wybór telefonu, odległość od najlepszych alternatyw



Rys.25 Wybór telefonu czas wykonywania metod



Rys.26 Wybór telefonu rankingi metod

W zbiorze danych dotyczących telefonów zastosowano różne metody wielokryterialnej analizy decyzyjnej, w tym TOPSIS i Topsis (GT). Wyniki dla obu tych metod wskazują na podobne podejście do klasyfikacji, jednak różnice w ich działaniu są zauważalne zarówno w rankingu alternatyw, jak i w uzyskanych metrykach.

Metoda **TOPSIS** klasyfikuje telefony, wykorzystując standardowy algorytm oceny odległości od punktu idealnego i anty-idealnego. W wynikowym rankingu TOPSIS na pierwszym miejscu znalazł się Telefon A, a kolejne miejsca zajmują Telefony D, H i J. Algorytm preferuje alternatywy, które znajdują się najbliżej punktu idealnego na podstawie wszystkich kryteriów, takich jak cena, bateria, ekran czy RAM. Ranking wskazuje na wyraźny kompromis pomiędzy kryteriami minimalizacji i maksymalizacji, co czyni metodę przejrzystą w interpretacji.

Z kolei **Topsis (GT)** działa w oparciu o dodatkowe modyfikacje względem klasycznego TOPSIS, co prowadzi do subtelnych różnic w wynikach. Choć Telefon A również znajduje się na szczycie rankingu, kolejność kolejnych alternatyw różni się w porównaniu do klasycznego TOPSIS. Na przykład, Telefon H znajduje się wyżej w Topsis (GT) niż w klasycznym TOPSIS. Może to wynikać z bardziej zaawansowanej normalizacji danych lub sposobu oceny odległości w Topsis (GT). W praktyce metoda ta może być bardziej adaptacyjna, ale również wymaga większej uwagi w interpretacji.

Wykresy przedstawiają również inne aspekty porównawcze metod. TOPSIS i Topsis (GT) osiągają podobne wyniki w zakresie odległości od punktu idealnego, co świadczy o ich skuteczności w optymalnym odwzorowywaniu decyzji. Wykres czasu wykonania pokazuje, że obie metody są bardzo szybkie i znacznie przewyższają w tym aspekcie bardziej złożone podejścia, takie jak SP-CS czy MREF. W porównaniu z innymi metodami, TOPSIS i Topsis (GT) pozostają wydajnymi i wszechstronnymi narzędziami do analizy wielokryterialnej.

Wyniki dla metod **SP_CS** oraz **TOPSIS**. Zbiór danych dotyczący siłowni. Dokonano również analizy porównawczej ze wszystkimi pozostałymi metodami.

Wczytaj dane z pliku

TOPSIS

Stwórz ranking

Alternatywy z kryteriami

Klasy

1

Siłownia A

0.8000

80

1

70

2

Siłownia B

1.7000

110

2

120

3

Siłownia C

2.2000

90

2

100

4

Siłownia D

1.3000

180

1

150

5

Siłownia E

3.5000

150

3

180

1

Siłownia A

0.8000

80

1

70

2

Siłownia B

1.7000

110

2

120

3

Siłownia C

2.2000

90

2

100

4

Siłownia D

1.3000

180

1

150

5

Siłownia E

3.5000

150

3

180

Stworzony Ranking

1

Siłownia D

1.3000

180

1

150

2

Siłownia A

0.8000

80

1

70

3

Siłownia B

1.7000

110

2

120

4

Siłownia C

2.2000

90

2

100

5

Siłownia E

3.5000

150

3

180

Porównaj metody

Rys.27 Wybór siłowni metodą TOPSIS

Wczytaj dane z pliku

Metoda SP-CS

Stwórz ranking

Alternatywy z kryteriami

Klasy

1

Siłownia A

0.8000

80

1

70

2

Siłownia B

1.7000

110

2

120

3

Siłownia C

2.2000

90

2

100

4

Siłownia D

1.3000

180

1

150

5

Siłownia E

3.5000

150

3

180

1

Siłownia A

0.8000

80

1

70

2

Siłownia B

1.7000

110

2

120

3

Siłownia C

2.2000

90

2

100

4

Siłownia D

1.3000

180

1

150

5

Siłownia E

3.5000

150

3

180

Stworzony Ranking

1

Siłownia A

0.8000

80

1

70

2

Siłownia D

1.3000

180

1

150

3

Siłownia E

3.5000

150

3

180

4

Siłownia C

2.2000

90

2

100

5

Siłownia B

1.7000

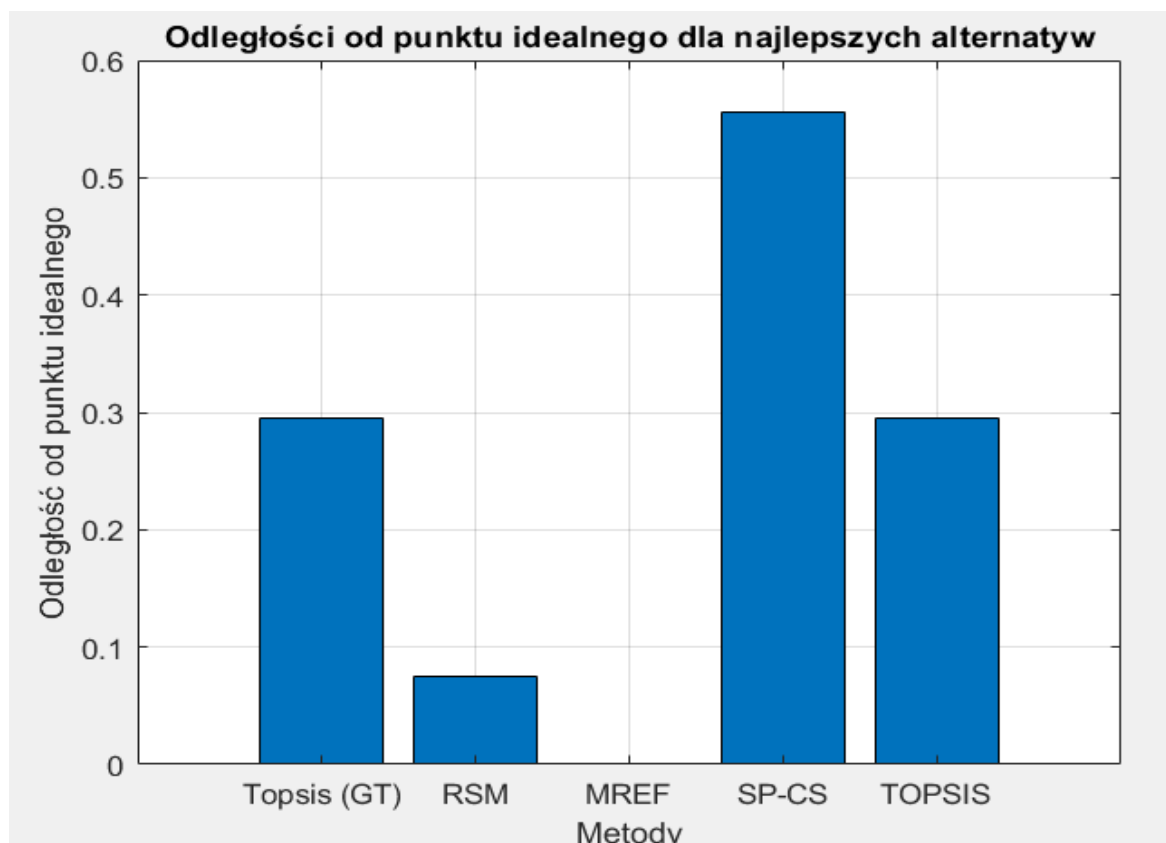
110

2

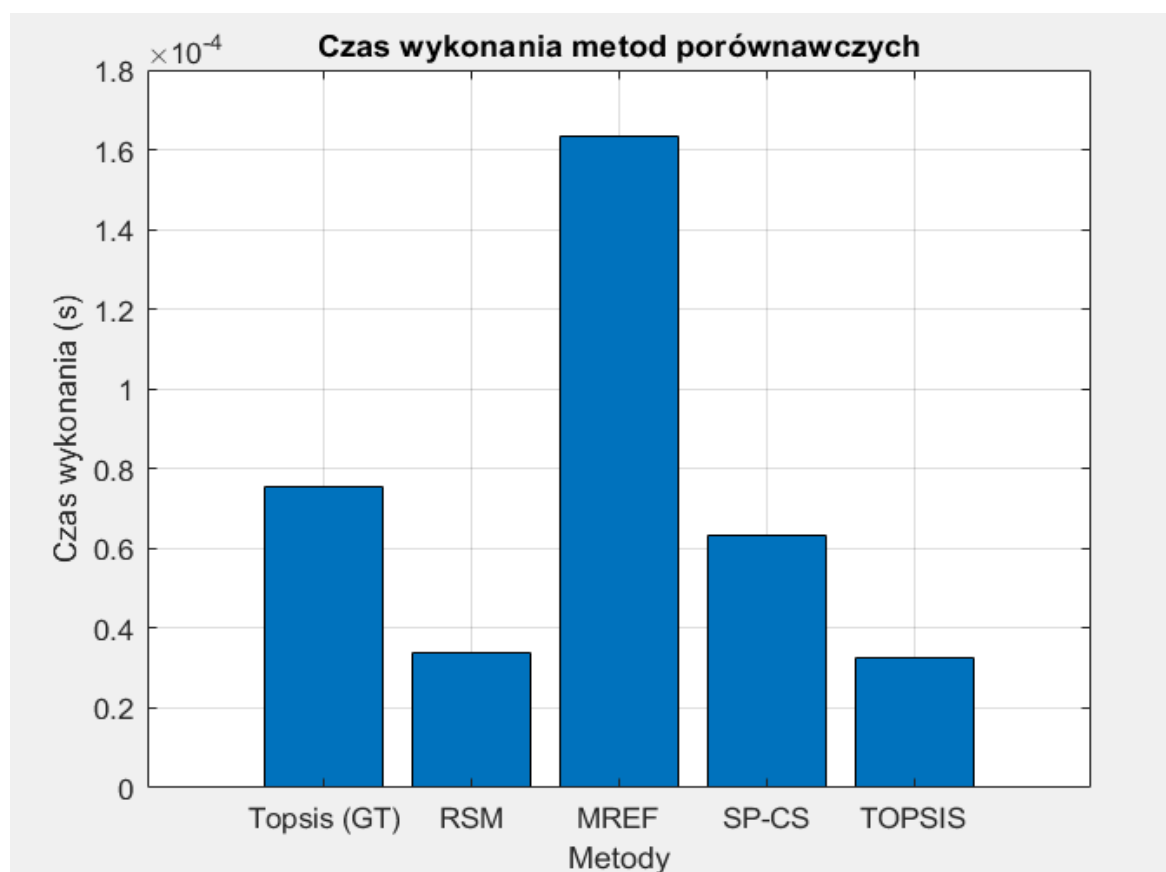
120

Porównaj metody

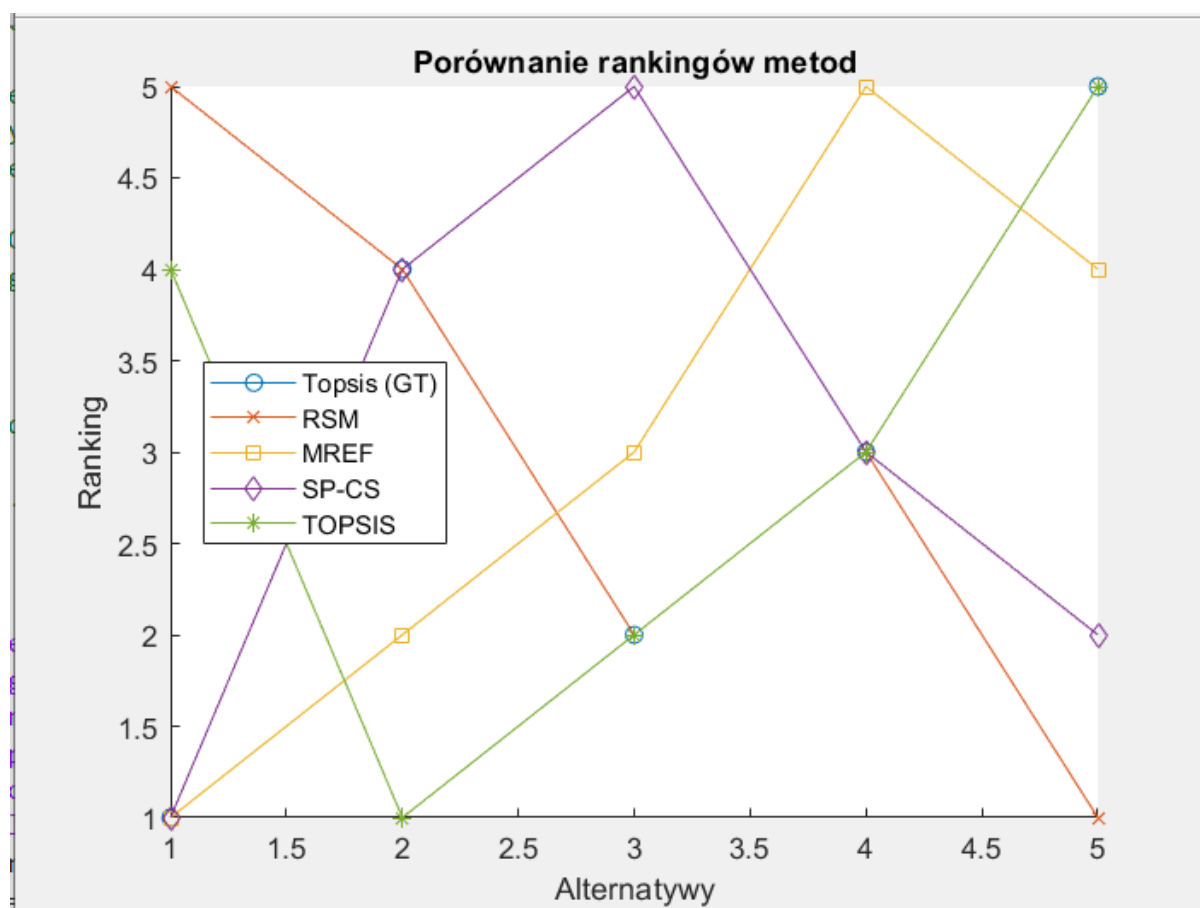
Rys.28 Wybór siłowni metodą SP-CP



Rys.29 Wybór siłowni, odległość od najlepszych alternatyw



Rys.30 Wybór siłowni czas wykonywania metod



Rys.31 Wybór siłowni rankingi metod

SP-CS generuje ranking, który na pierwszym miejscu plasuje Siłownię A, co wynika z większego nacisku na minimalizację odległości od punktu wzorcowego przy określonych kryteriach. TOPSIS, natomiast, umieszcza Siłownię D na pierwszym miejscu, co świadczy o jej lepszym balansie między maksymalizacją i minimalizacją kryteriów, takich jak wielkość czy cena.

Obie metody różnią się także w interpretacji punktu idealnego. SP-CS bardziej skupia się na konkretnych punktach odniesienia, co może prowadzić do wyższych odległości od punktu idealnego, jak pokazuje wykres odległości. TOPSIS natomiast generuje bardziej umiarkowane odległości, co świadczy o większej stabilności w podejmowaniu decyzji.

Pozostałe metody, takie jak Topsis (GT) i RSM, wprowadzają różnice w kolejności rankingów, przy czym RSM plasuje Siłownię C wyżej, co może wynikać z większego uwzględnienia specyfiki danych. MREF z kolei jest bardziej czasochłonny, ale w przypadku tego zbioru danych nie oferuje znaczących korzyści w jakości rankingów.

Pod względem czasu wykonania SP-CS jest bardziej czasochłonny od TOPSIS, co czyni tę drugą metodę bardziej wydajną w zastosowaniach praktycznych. Jednak wybór odpowiedniej metody zależy od priorytetów użytkownika – SP-CS może być przydatny, gdy specyficzne punkty odniesienia mają kluczowe znaczenie, natomiast

TOPSIS oferuje lepszy kompromis między prostotą, szybkością i jakością wyników. Pozostałe metody, takie jak Topsis (GT), również stanowią dobrą alternatywę, jeśli wymagane jest bardziej elastyczne podejście do danych.

Wnioski:

Przeprowadzone analizy porównawcze metod wielokryterialnej analizy decyzyjnej wykazały, że każda z metod ma swoje unikalne zalety i wady, a wybór odpowiedniej techniki zależy od specyfiki problemu oraz priorytetów decydenta. Metody TOPSIS i SP-CS różnią się w podejściu do oceny alternatyw. TOPSIS wykazuje większą stabilność w wynikach, generując bardziej zrównoważony ranking alternatyw dzięki efektywnemu kompromisowi pomiędzy kryteriami minimalizacji i maksymalizacji. Z kolei SP-CS bardziej skupia się na specyficznych punktach odniesienia, co czyni go bardziej elastycznym, ale również bardziej czasochłonnym.

Porównania wskazują, że TOPSIS i jego wariant TOPSIS_GT cechują się najwyższą szybkością obliczeń, co czyni je praktycznymi w zastosowaniach wymagających natychmiastowych decyzji. SP-CS, mimo swojej złożoności, może być przydatny w sytuacjach wymagających precyzyjnego uwzględnienia określonych ograniczeń. Metody takie jak MREF i RSM oferują większą elastyczność w modelowaniu preferencji decydenta, ale ich złożoność obliczeniowa i czas wykonania mogą ograniczać ich praktyczność w przypadku dużych zbiorów danych.

Wyniki analiz rankingowych pokazały, że metody klasyczne, takie jak TOPSIS, dobrze radzą sobie z wyważonymi kryteriami, natomiast SP-CS i RSM są bardziej odpowiednie, gdy priorytety decyzyjne wymagają szczególnej precyzji w uwzględnieniu punktów referencyjnych. Ogólnie rzecz biorąc, program umożliwia kompleksowe testowanie i porównywanie różnych metod analizy decyzyjnej, oferując użytkownikom elastyczne i intuicyjne narzędzie wspomagania decyzji w szerokim zakresie problemów praktycznych.