

Porównanie metod TOPSIS, RSM, Zasady Bezpieczeństwa i UTA

Optymalizacja Wielokryterialna – ćwiczenie 4

Autorzy:

Julia Nowak

Adam Złocki

Jakub Szczypek

Spis treści

Cel zadania	2
Uruchamianie programu.....	2
Fuzzy TOPSIS	4
Wariant ciągły (Fuzzy Topsis – przykłady obliczeń $U \subset R4, N = 4$).....	6
Wariant dyskretny (przykłady obliczeń $N = 3, N = 4$)	7
RSM	8
Wariant ciągły (przykłady obliczeń $U \subset R4, N = 4$).....	9
Wariant dyskretny (przykłady obliczeń $N = 3, N = 4$)	10
Metoda SP-CS (Safety Principle and Compromise Selection)	11
Wariant ciągły (przykłady obliczeń $U \subset R4, N = 4$).....	12
Wariant dyskretny (przykłady obliczeń $N = 3, N = 4$)	15
UTA_BIS.....	19
Wariant ciągły (przykłady obliczeń $U \subset R4, N = 4$).....	20
Wariant dyskretny (przykłady obliczeń $N = 3, N = 4$)	21
Wnioski	22

Cel zadania

Celem zadania było zapoznanie się z metodami wielokryterialnego wyboru (stosowanymi w przypadku zarówno dyskretnych, jak i ciągłych zbiorów decyzji) oraz rankingowania (dla dyskretnych zbiorów decyzji) opartymi o wykorzystanie punktów odniesienia i struktur metrycznych w przestrzeni wartości kryteriów.

Naszym głównym zadaniem była implementacja następujących metod:

- Fuzzy TOPSIS (Technique for Order Preference by Similarity to Ideal Solution),
- RSM (Reference Set Method),
- Utilité Additif (UTA, w wariancie sortującym UTA DIS),
- Safety Principle Based Ranking and Compromise Selection (SP-CS).

Otrzymane wyniki należało zwizualizować, a także przedstawić wyniki porównania metod oraz analizę wrażliwości rozwiązań na zmiany parametrów zastosowanych metod.

Uruchamianie programu

Aby uruchomić program należy pobrać kod z repozytorium lub pobrać paczkę „zip”. Repozytorium znajduje się pod linkiem https://github.com/Julnowak/Optymalizacja_wielokryterialna, jednak o dostęp należy poprosić autorów kodu. Aby program działał prawidłowo należy po otwarciu w edytorze Pythona zainstalować wszystkie biblioteki zawarte w pliku „requirements.txt” poprzez komendę: „pip install requirements.txt”. Gdy wszystkie biblioteki będą już zainstalowane, należy przejść do folderu „OW_GUI” i uruchomić plik „mainwindow.py”. Podjęcie tej akcji powinno skutkować pojawieniem się okna aplikacji. W razie problemów, prosimy o kontakt z autorami.

Aby rozpocząć pracę z programem należy wczytać plik Excel z kryteriami. Plik powinien być zgodny z obranym przez nas szablonem. Przykładowe dane wejściowe zostały przedstawione w folderze lab4 i odpowiednio opisane. Następnie należy wybrać interesujące nas parametry i nacisnąć „Stwórz ranking”. Wyniki są wyświetlane w zakładce wyników.

Ustawienia

Wyniki

Wczytaj dane z pliku

FUZZY TOPSIS

Stwórz ranking

Alternatywy z kryteriami

Klasy

Nr alternatywy	Nazwa alternatywy	Kryterium 1	Kryterium 2	Kryterium 3	Kryterium 4
1 1	Alternatywa1	2.0904743905067136	6	5.84352384819894	0.08489314461367614
2 2	Alternatywa2	2.136745192695085	8	4.073621060980281	0.05227531498922888
3 3	Alternatywa3	2.8824240651985216	10	5.88626684906991	0.7059703629445129
4 4	Alternatywa4	2.2189291096008237	10	5.380122269572196	0.7922222905956389
5 5	Alternatywa5	1.7437031225579984	10	5.459053818482277	0.43703281968858065
6 6	Alternatywa6	1.2748077141920466	3	3.213631370987292	0.6009147270661678
7 7	Alternatywa7	2.410480038121751	5	3.476611931926714	0.3148133065710934
8 8	Alternatywa8	1.3360660987678945	3	4.281372542974584	0.33008193847725686
9 9	Alternatywa9	1.5776556741089196	6	3.8811550966847803	0.8044078737604274
10 10	Alternatywa10	1.6334118026517506	7	5.782259172965704	0.06645932791911968
11 11	Alternatywa11	1.9956380428207177	4	3.0706892263676386	0.10974799095650767
12 12	Alternatywa12	2.2495940231932865	2	3.331834879332713	0.6459771553282687
13 13	Alternatywa13	1.7433594604359761	6	5.6502685512848725	0.2818649737537682
14 14	Alternatywa14	1.12885174325053	9	4.33478005841616	0.4602838957192489

Nr klasy	x	y	z	a
1 1	1.0	5	3.2	0.2
2 2	1.7	8	3.9	0.5

Parametry specyficzne dla algorytmów:

Variant:

Ciągły

Miara:

Czebyszewa

Granice:

0

10

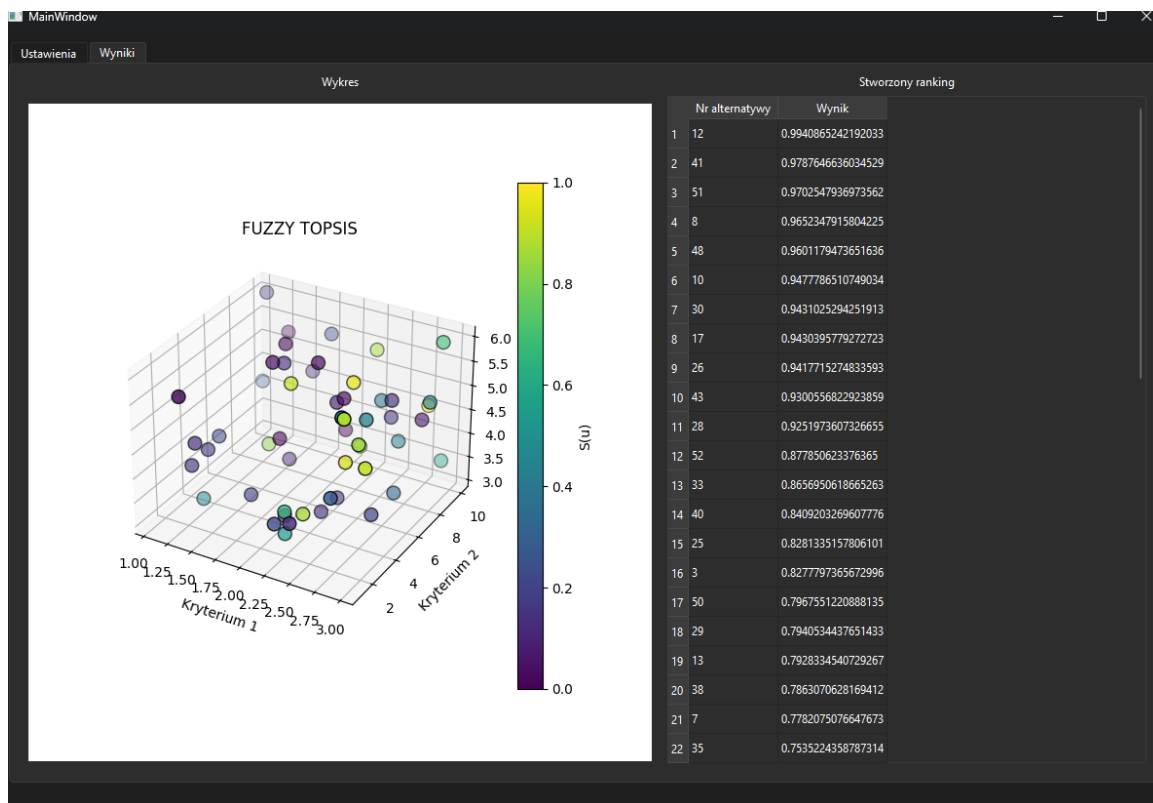
Liczba próbek:

5

Typ:

Maksymalizacja

Rys. 1 Zrzut ekranu przedstawiający zakładkę "ustawienia" programu



Rys. 2 Zrzut ekranu przedstawiający zakładkę "wyniki" programu

Fuzzy TOPSIS

Metoda **Fuzzy TOPSIS** (Technika Porządkowania Preferencji poprzez Podobieństwo do Rozwiązania Idealnego) to rozszerzenie klasycznej metody TOPSIS, która wykorzystuje logikę rozmytą do radzenia sobie z niepewnością i nieprecyzyjnością w ocenianiu alternatyw. Dzięki zastosowaniu zbiorów rozmytych, metoda ta pozwala na bardziej elastyczną ocenę kryteriów, szczególnie gdy dane są nieprecyzyjne lub oparte na subiektywnych opiniach.

Etapy metody Fuzzy TOPSIS:

1. Zdefiniowanie macierzy decyzyjnej i wag kryteriów

- Macierz decyzyjna X - zawiera wartości rozmyte reprezentujące oceny alternatyw w odniesieniu do kryteriów

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

- Wagi kryteriów

$$w = [w_1, w_2, \dots, w_n]$$

2. Normalizacja macierzy decyzyjnej:

- Dla przypadku maksymalizacji

$$r_{ij} = \frac{x_{ij}}{\max x_{ij}}$$

- Dla przypadku minimalizacji

$$r_{ij} = \frac{\min x_{ij}}{x_{ij}}$$

3. Wyznaczenie macierzy wag znormalizowanych:

$$v_{ij} = w_j * r_{ij}$$

4. Określenie rozwiązań idealnych (FPIS) i anty-idealnych (FNIS):

- Idealnych (FPIS)

$$v^* = (v_1^*, \dots, v_N^*) = (\max v_{ij} \mid j \in \text{kryteria maksymalizowane}, \min v_{ij} \mid j \in \text{kryteria minimalizowane}),$$

- Anty-idealnych (FNIS)

$$v^- = (v_1^-, \dots, v_N^-) = (\max v_{ij} \mid j \in \text{kryteria minimalizowane}, \min v_{ij} \mid j \in \text{kryteria maksymalizowane}),$$

5. Obliczanie odległości od rozwiązań:

- Idealnych (FPIS)

$$d_i^* = \sqrt{\sum_{j=1}^N (v_{ij} - v_j^*)^2}$$

- Anty-idealnych (FNIS)

$$d_i^- = \sqrt{\sum_{j=1}^N (v_{ij} - v_j^-)^2}$$

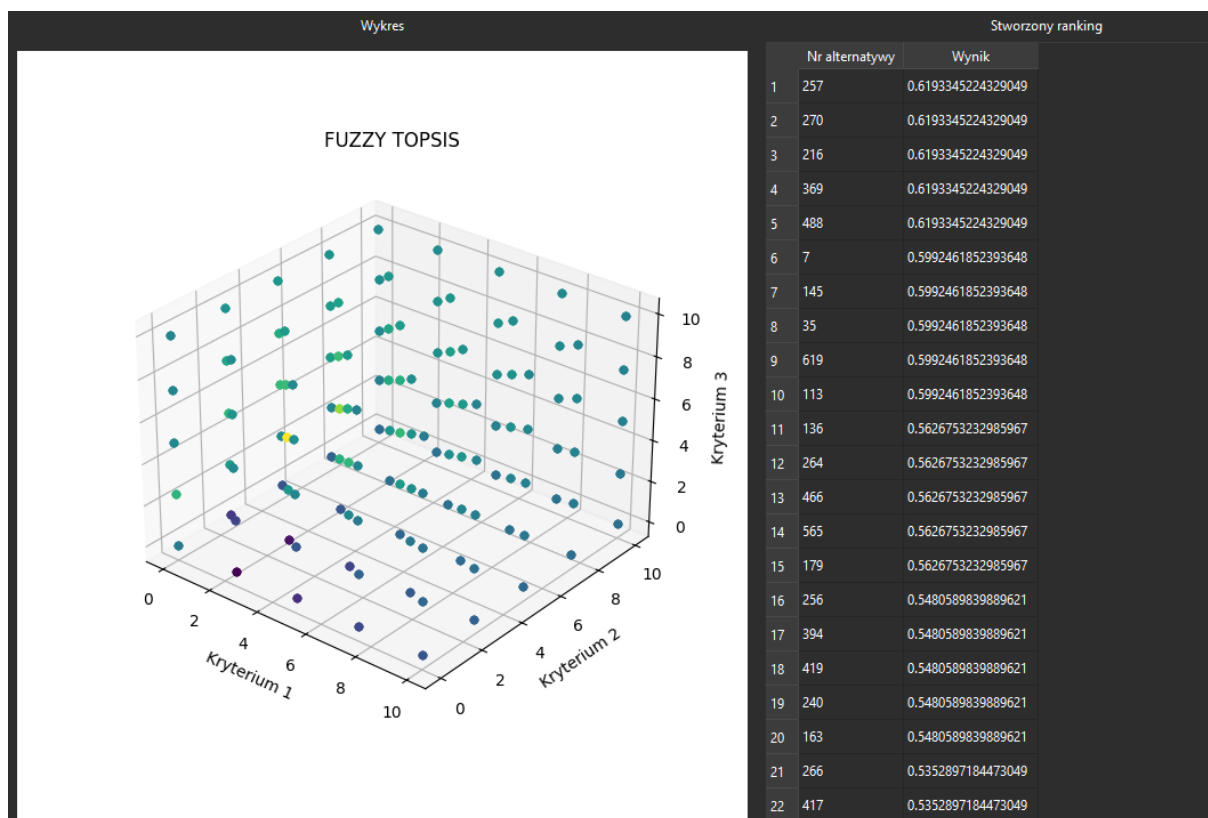
6. Obliczanie współczynnika bliskości, określającego jak blisko dana alternatywa jest do rozwiązania idealnego

$$c_i^* = \frac{d_i^-}{d_i^- + d_i^*}$$

Gdzie przyjmujemy, że $c^* \in [0, 1]$.

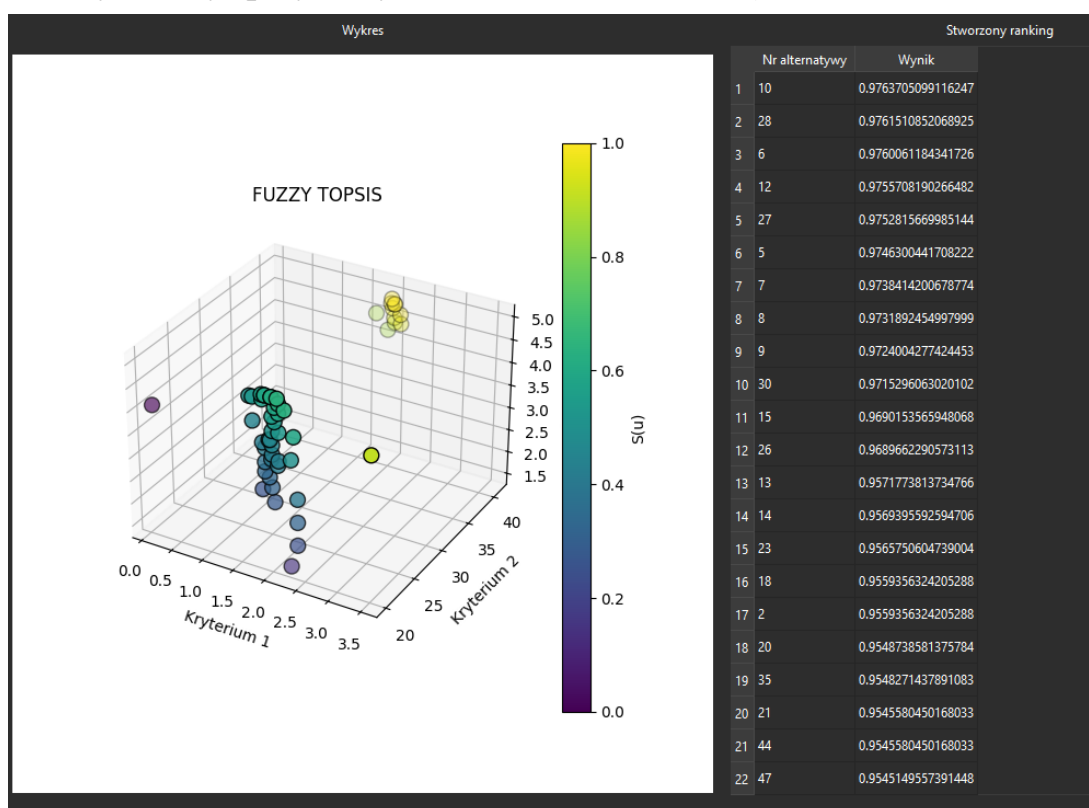
- Sortowanie alternatyw według wartości współczynnika bliskości, od najwyższej do najniższej.

Wariant ciągły (Fuzzy Topsis – przykłady obliczeń $U \subset R^4, N = 4$)

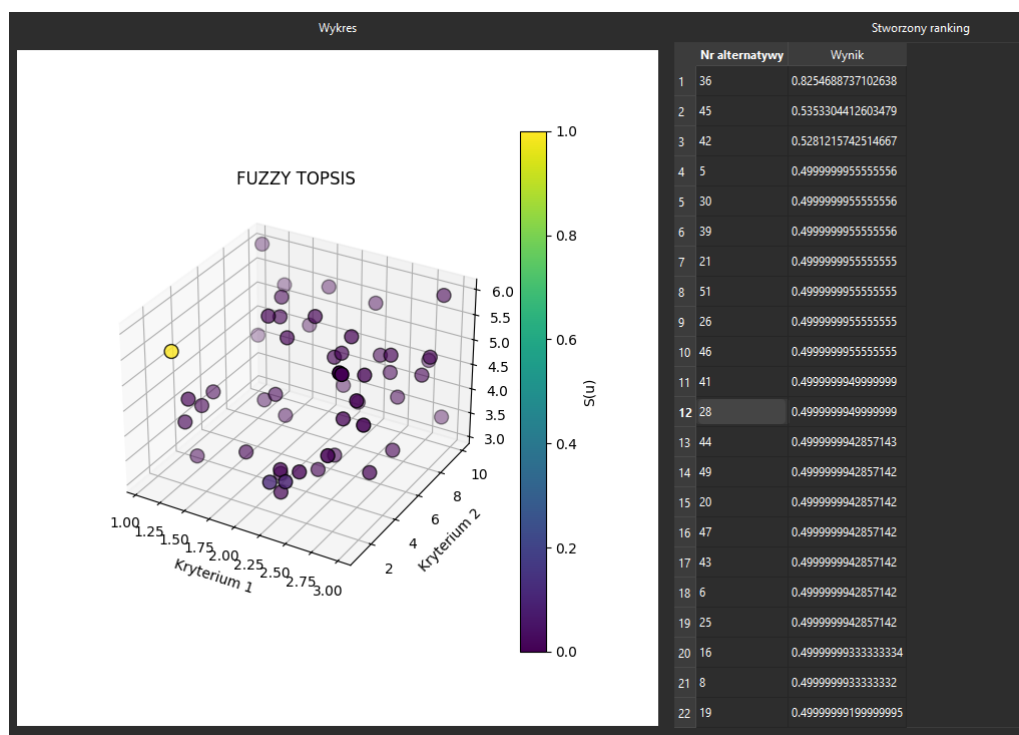


Rys. 3 Maksymalizacja wszystkich kryteriów dla $N=4$, miara czebysewa dla optymalizacji względem podanego punktu idealnego

Wariant dyskretny (przykłady obliczeń $N = 3, N = 4$)



Rys. 4 Maksymalizacja wszystkich kryteriów dla $N=3$, miara euklidesowa



Rys. 5 Minimalizacja wszystkich kryteriów dla $N=4$, miara czebyszewa

RSM

Rozszerzona metoda zbiorów odniesienia (ang. Reference Set Method, RSM) to technika analizy porównawczej, która znajduje zastosowanie w ocenie efektywności systemów, procesów lub decyzji. W RSM zamiast analizy tylko jednego zbioru odniesienia uwzględniany jest zbiór docelowy R^+ i niekorzystny R^- . Każda z analizowanych alternatyw jest porównywana z wszystkimi elementami tych zbiorów, a efekty tych porównań składają się na wartość funkcji scoringowej. Dzięki temu możliwe jest bardziej elastyczne i dokładne odzwierciedlenie specyficznych warunków, w jakich funkcjonuje oceniana jednostka. Warto pamiętać, że rezultaty zależą przede wszystkim od wybranych punktów odniesienia.

Kluczowe elementy metody:

- **Zbiór alternatyw (U):** zbiór analizowanych możliwości.
- **Punkty odniesienia docelowe (R^+):** zbiór punktów, do których algorytm wybierze najbardziej zbliżone punkty.
- **Punkty odniesienia niekorzystne (R^-):** zbiór punktów, od których algorytm wybierze najbardziej oddalone punkty.
- **Funkcja scoringowa ($S(u)$):** wyrażona jako różnica między odległością od najbliższego punktu niekorzystnego, a odległością od najbliższego punktu docelowego:

$$S(u) = d^-(u) - d^+(u)$$

Kroki algorytmu:

1. Zdefiniowanie zbiorów odniesienia R^+ i R^- :

$$R^+ = \{r_1^+, r_2^+, r_3^+, \dots\}$$

$$R^- = \{r_1^-, r_2^-, r_3^-, \dots\}$$

2. Obliczenie odległości dla każdej alternatywy u :

$$d^+(u) = \min_{r^+ \in R^+} d(u, r^+)$$

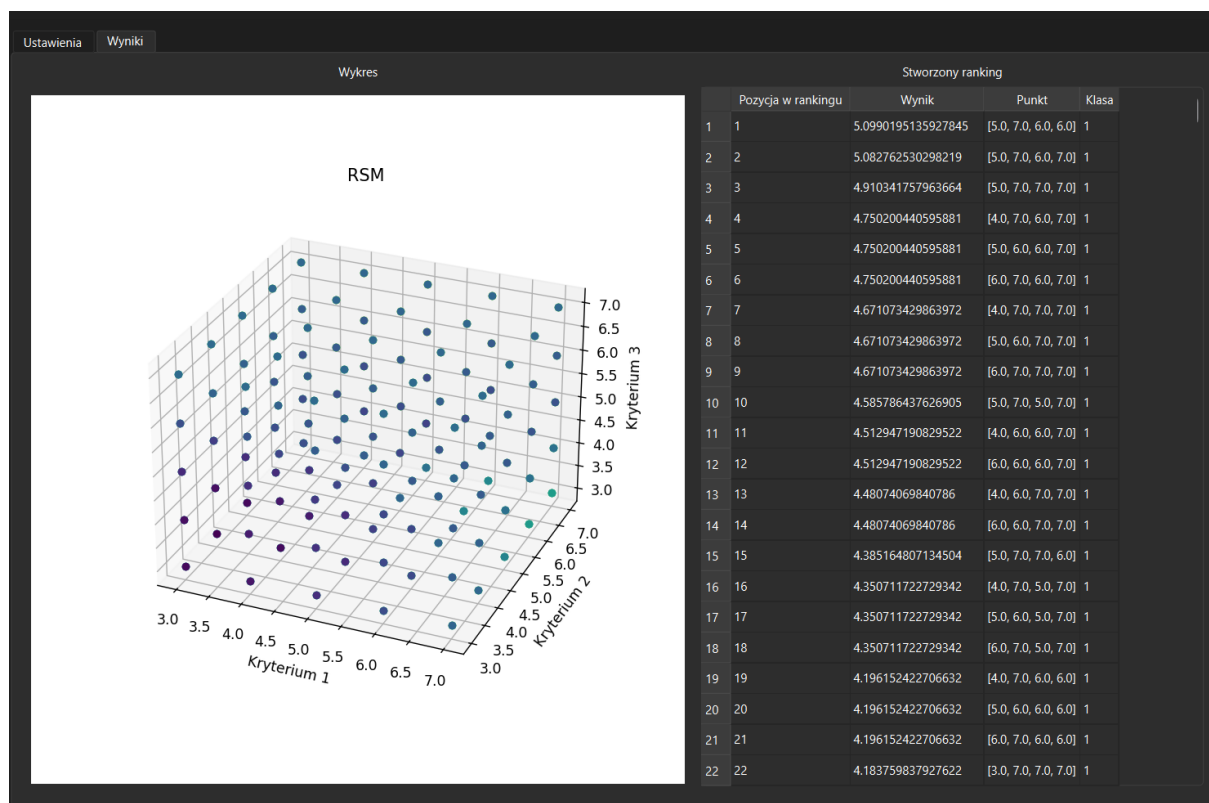
$$d^-(u) = \min_{r^- \in R^-} d(u, r^-)$$

3. Obliczenie wartości funkcji scoringowej $S(u)$ dla każdej alternatywy u :

$$S(u) = d^-(u) - d^+(u)$$

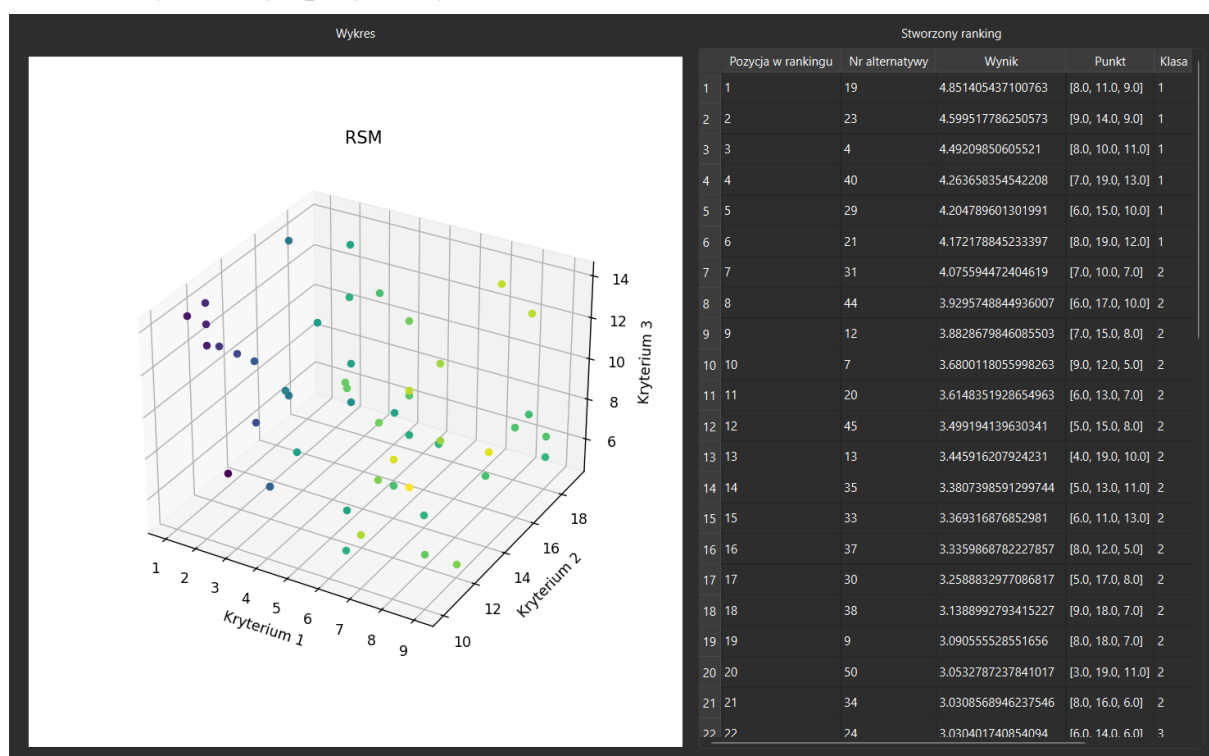
4. Utworzenie rankingu alternatyw, gdzie najlepszą alternatywą jest ta, z największą wartością funkcji scoringowej.

Wariant ciągły (przykłady obliczeń $U \subset R^4, N = 4$)

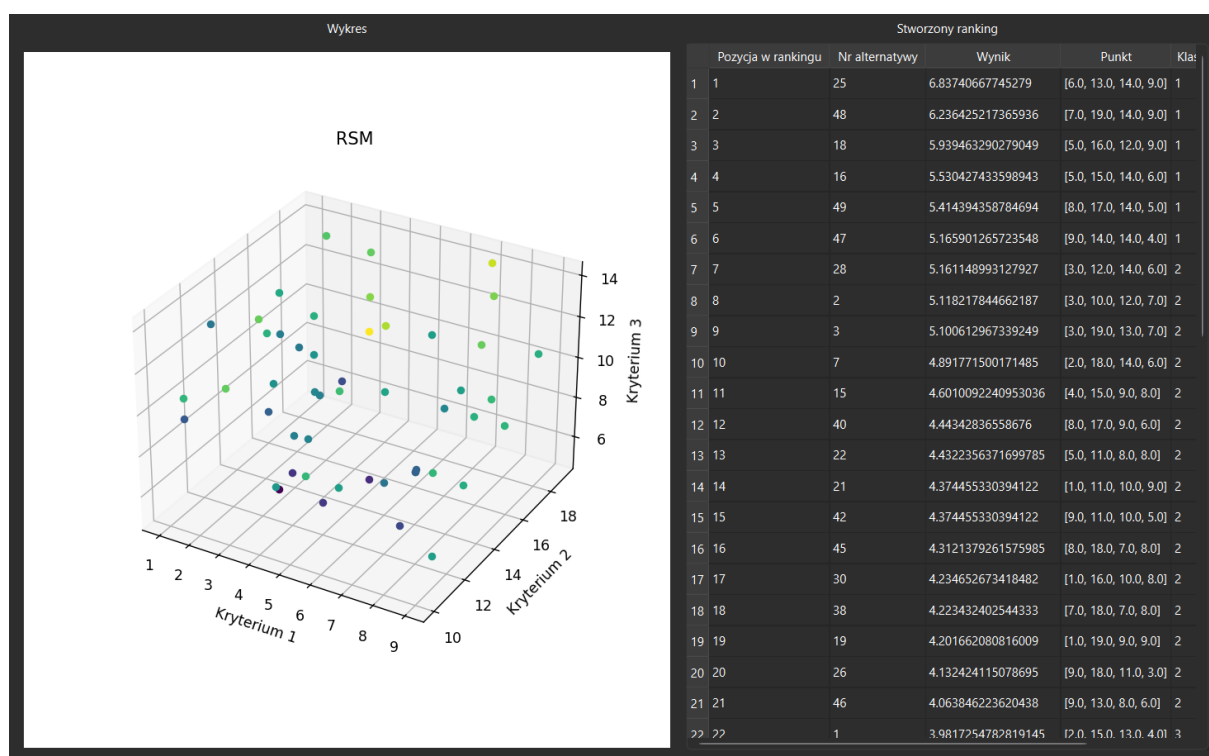


Rys. 6 Minimalizacja wszystkich kryteriów dla $N=4$ dla wariantu ciągłego

Wariant dyskretny (przykłady obliczeń $N = 3, N = 4$)



Rys. 7 Minimalizacja wszystkich kryteriów dla $N=3$ dla wariantu dyskretnego



Rys. 8 Minimalizacja wszystkich kryteriów dla $N=4$ dla wariantu dyskretnego

Metoda SP-CS (Safety Principle and Compromise Selection)

Metoda SP-CS to narzędzie stosowane w optymalizacji wielokryterialnej, które służy do wyboru rozwiązania kompromisowego między punktem status quo a punktem aspiracji. Opiera się na minimalizacji funkcji scoringowej $S(u)$, która uwzględnia zarówno odległość alternatywy od krzywej szkieletowej łączącej punkty odniesienia, jak i parametr t , reprezentujący stopień zbliżenia do punktu aspiracji.

Kluczowe elementy metody:

- **Punkt status quo (A_0):** reprezentuje obecny stan lub punkt odniesienia, od którego zaczynamy analizę.
- **Punkt aspiracji (A_1):** reprezentuje pożądany stan lub cel, do którego dążymy.
- **Krzywa szkieletowa ($\gamma(t)$):** linia łącząca punkty A_0 i A_1 , gdzie $t \in [0,1]$ określa pozycję na krzywej.
- **Funkcja scoringowa ($S(u)$)** wyrażona jako suma parametru t i odległości d alternatywy u od krzywej $\gamma(t)$:

$$S(u) = t^* + d(u, \gamma(t^*))$$

gdzie t^* minimalizuje $S(u)$ dla danej alternatywy u .

Kroki algorytmu:

1. Skonstruowanie krzywej $\gamma(t)$, która reprezentuje ciągłe przejście od punktu status quo do punktu aspiracji.

$$\gamma(t) = (1 - t)A_0 + tA_1, t \in [0,1]$$

2. Obliczenie odległości alternatyw od krzywej $\gamma(t)$:

$$d(u, \gamma(t)) = \sqrt{\sum_{k=1}^N (u_k - \gamma_k(t))^2}$$

Dla każdej alternatywy u i dla każdej wartości t obliczamy odległość $d(u, \gamma(t))$, która może zostać obliczona np. przy pomocy metryki euklidesowej.

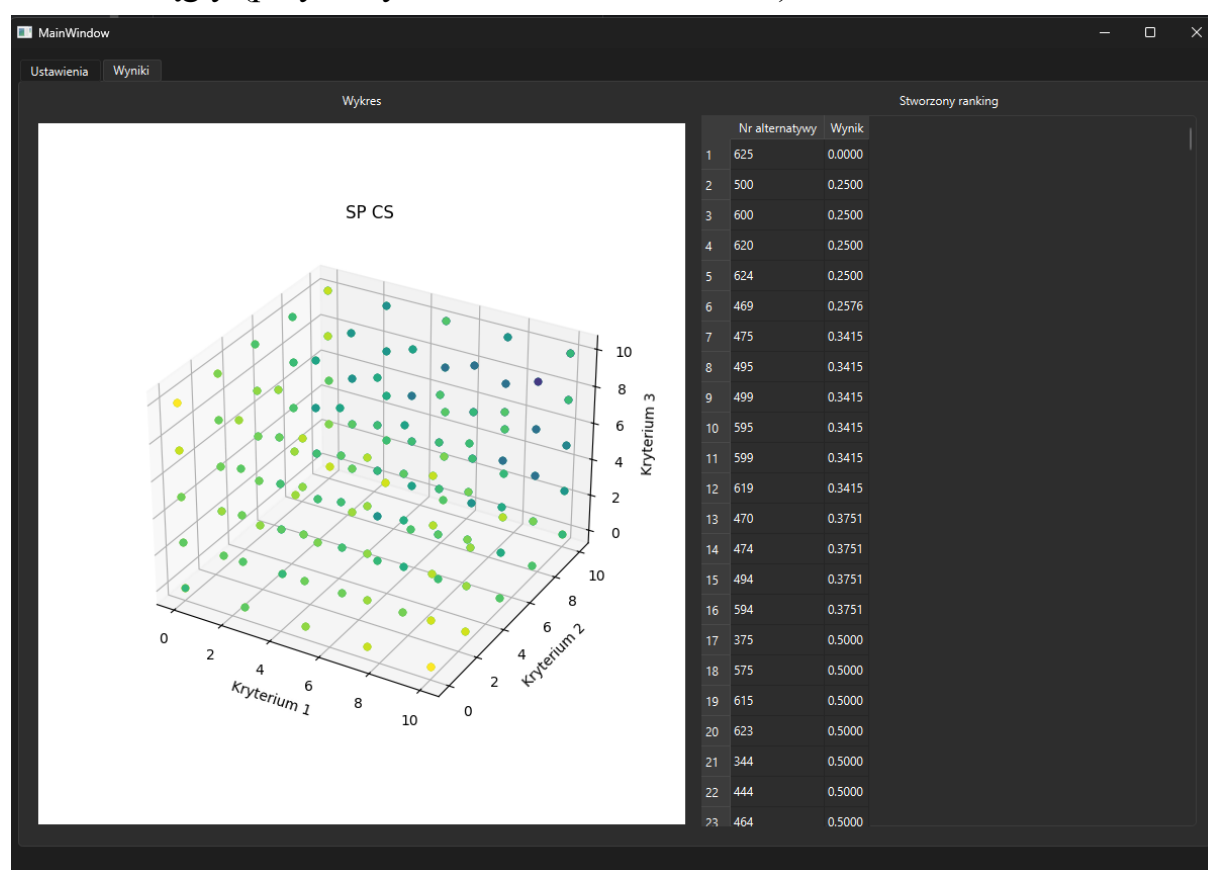
3. Minimalizacja funkcji scoringowej $S(u)$:

Dla każdej alternatywy u znajdujemy wartość t^* , która minimalizuje $S(u)$. Jest to równoznaczne ze znalezieniem punktu na krzywej $\gamma(t)$, który daje najlepszy możliwy kompromis między zbliżeniem do aspiracji, a odległością od krzywej.

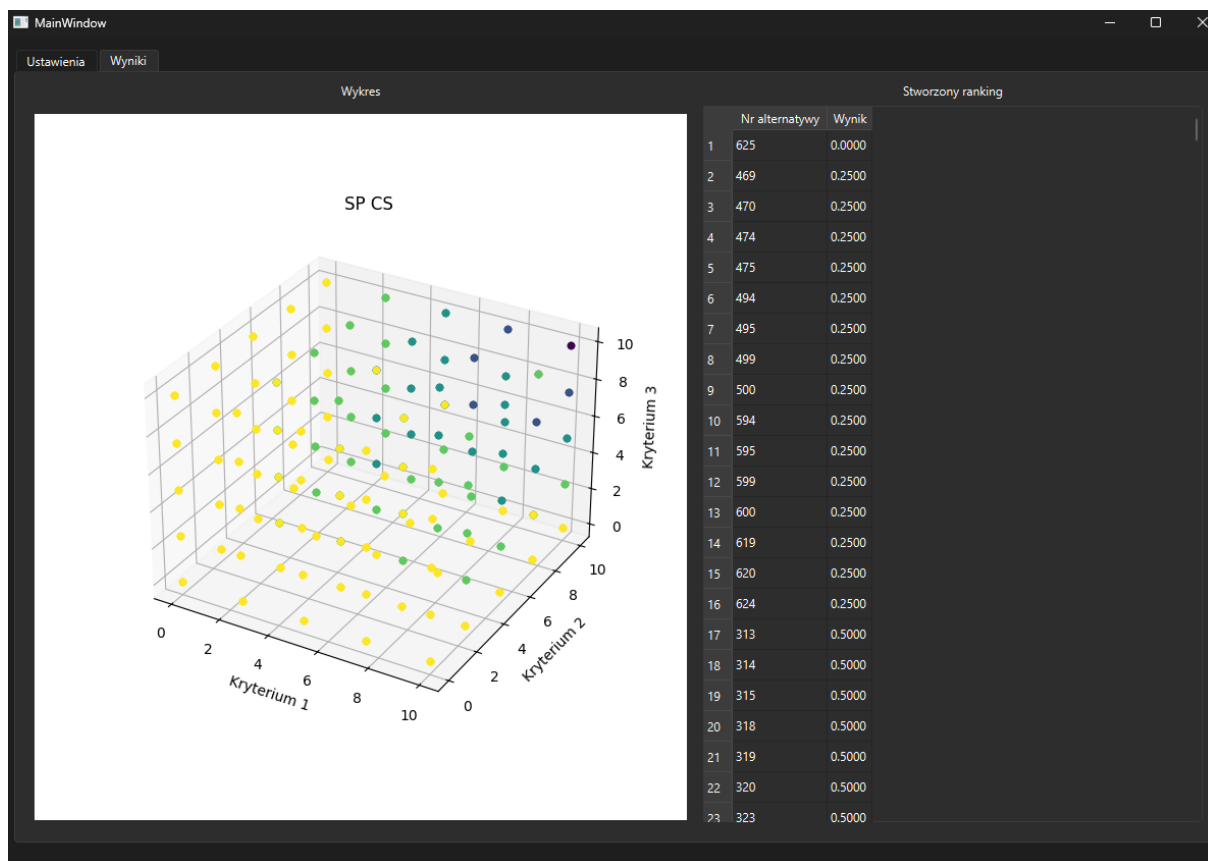
4. Utworzenie rankingu alternatyw i posortowanie według rosnących wartości $S(u)$.
Najlepsza alternatywa to ta z najmniejszym $S(u)$.

Zarówno w wariancie ciągłym, jak i dyskretnym, metoda SP-CS po odpowiedniej normalizacji zapewnia poprawne odwzorowanie preferencji decydenta. Kryteria maksymalizowane i minimalizowane są traktowane zgodnie ze swoją naturą: w pierwszym przypadku większe wartości kryterium są premiowane, a w drugim mniejsze. Ostateczny wynik to ranking rozwiązań lub regionów w przestrzeni decyzyjnej, które najlepiej spełniają założone cele, wyrażone przez punkty status quo i aspiracji, oraz minimalizują wartość $S(u)$. Dzięki temu decydent otrzymuje klarowne wskazówki, które alternatywy są najbardziej atrakcyjne z punktu widzenia wielokryterialnego kompromisu.

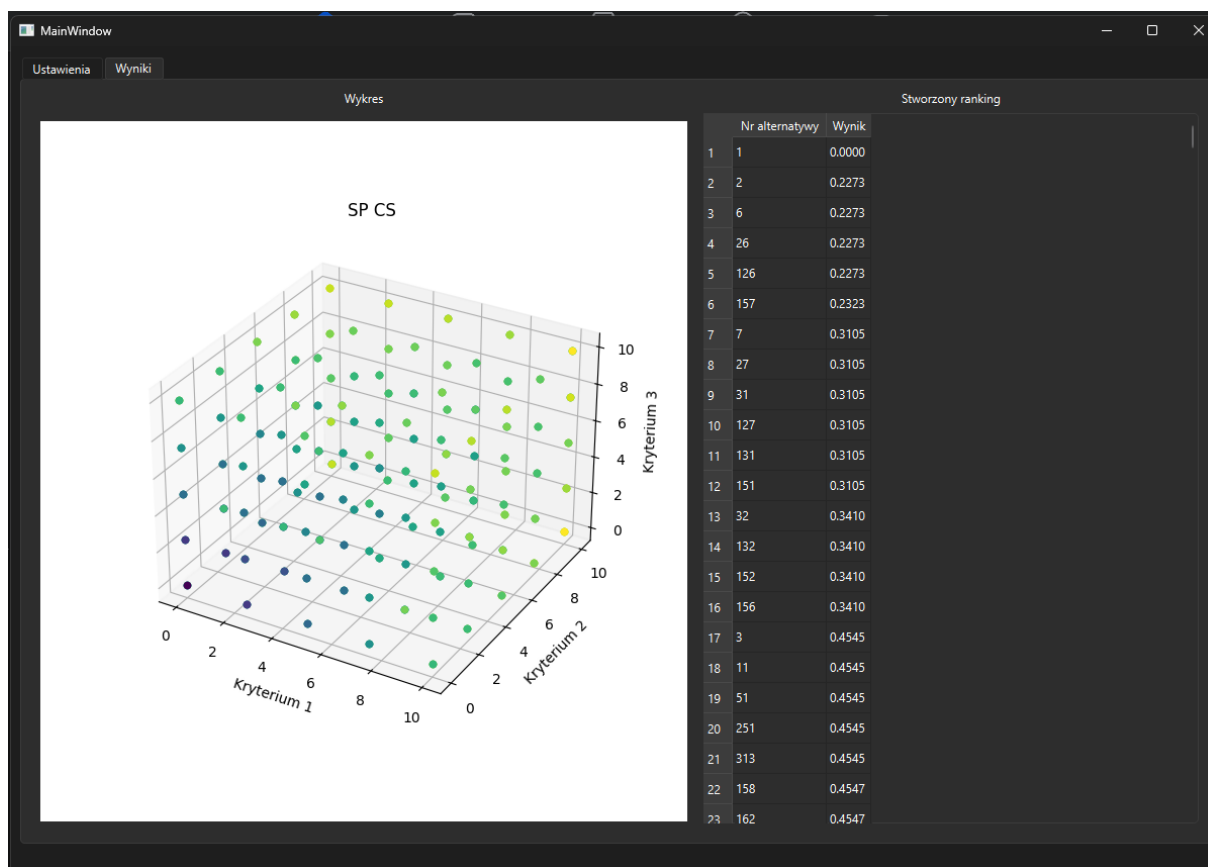
Wariant ciągły (przykłady obliczeń $U \subset R^4, N = 4$)



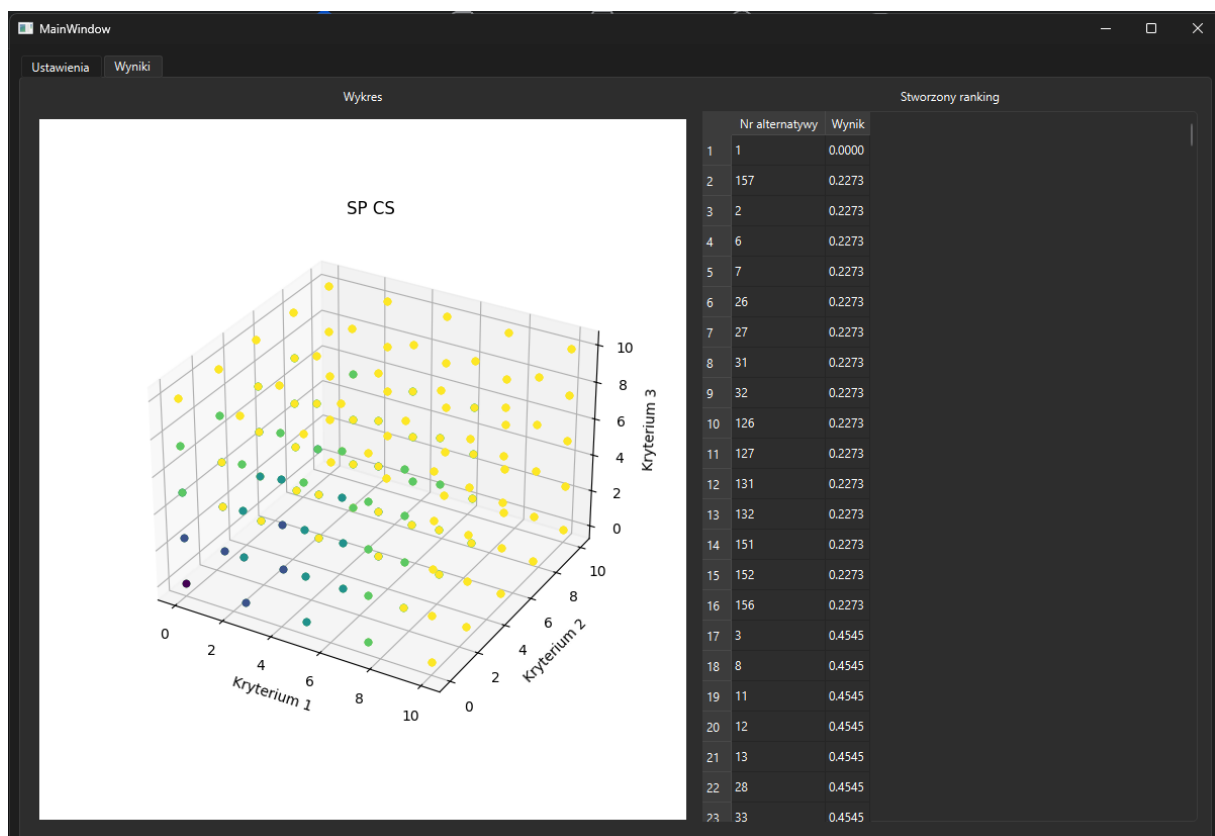
Rys. 9 Wizualizacja działania algorytmu SP-CS dla wariantu ciągłego, $N=4$, miara Euklidesowa, minimalizacja



Rys. 10 Wizualizacja działania algorytmu SP-CS dla wariantu ciągłego, $N=4$, miara Chebysheva, minimalizacja

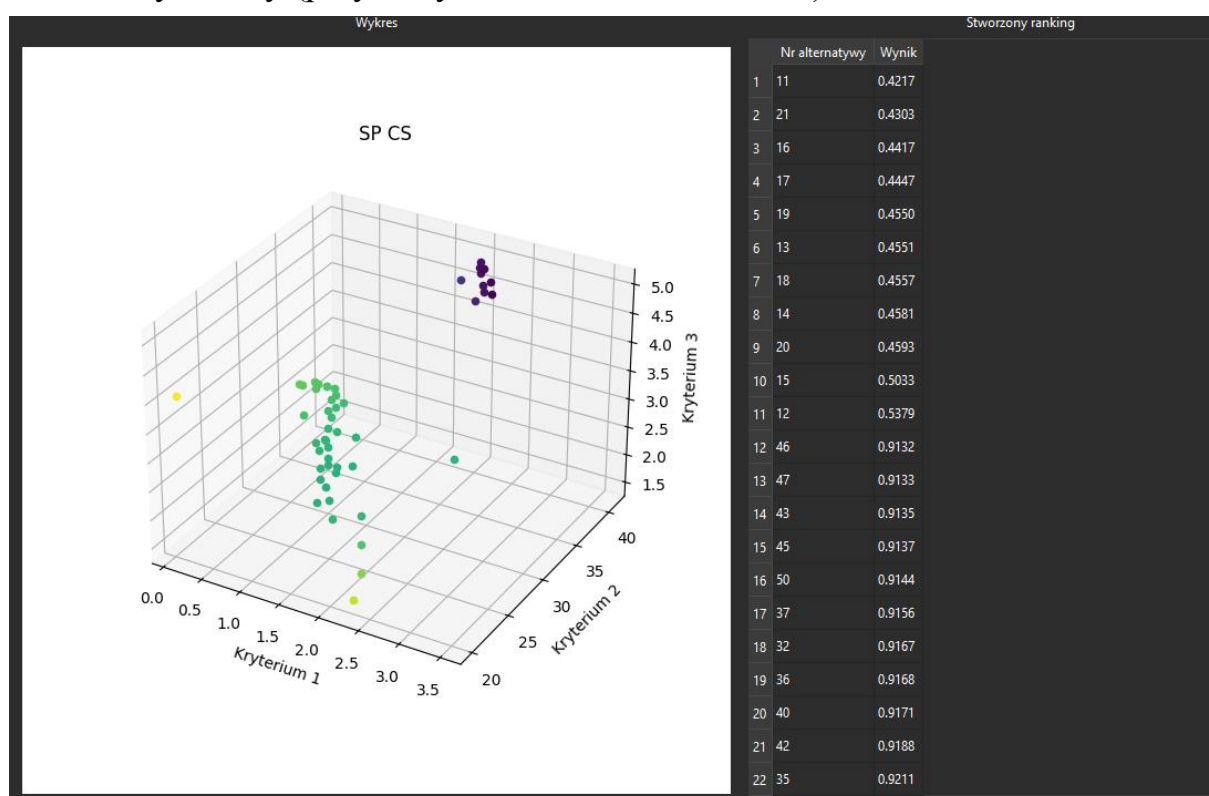


Rys. 11 Wizualizacja działania algorytmu SP-CS dla wariantu ciągłego, $N=4$, miara Euklidesowa, maksymalizacja

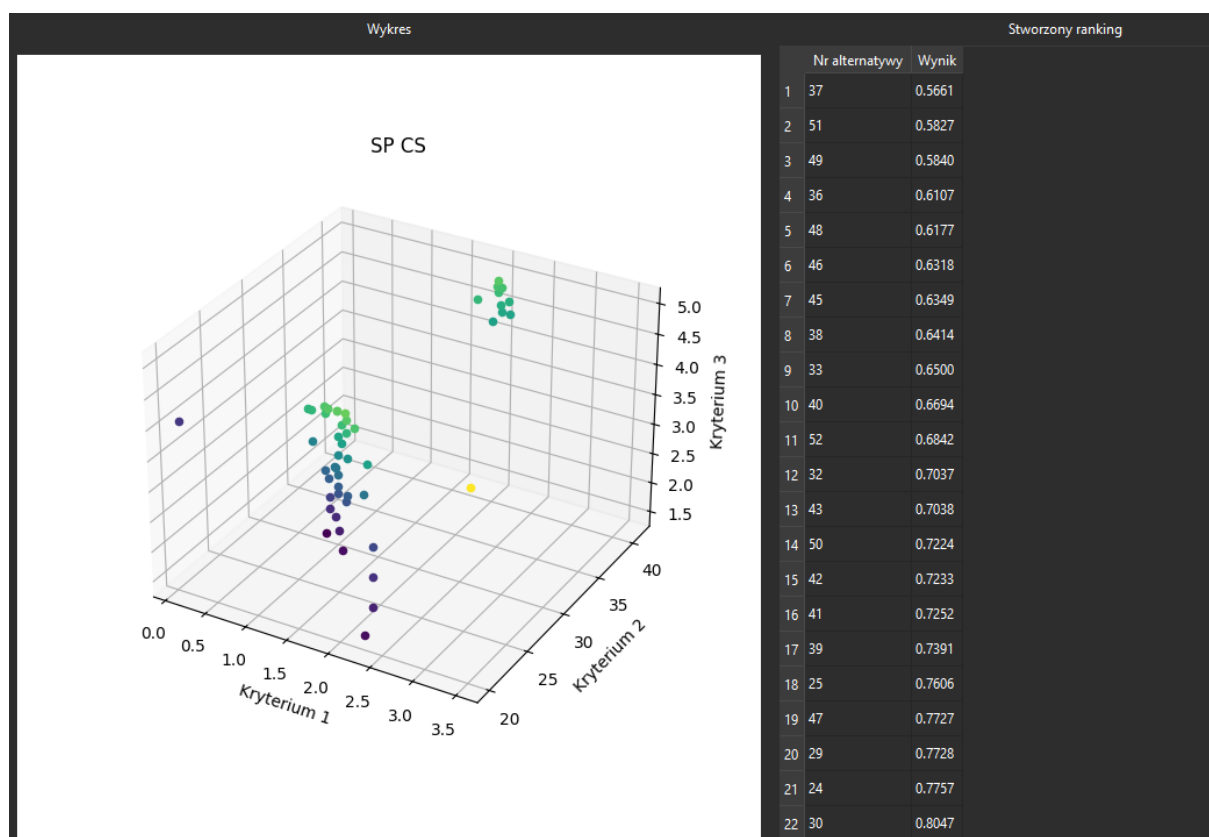


Rys. 12 Wizualizacja działania algorytmu SP-CS dla wariantu ciągłego, $N=4$, miara Chebysheva, maksymalizacja

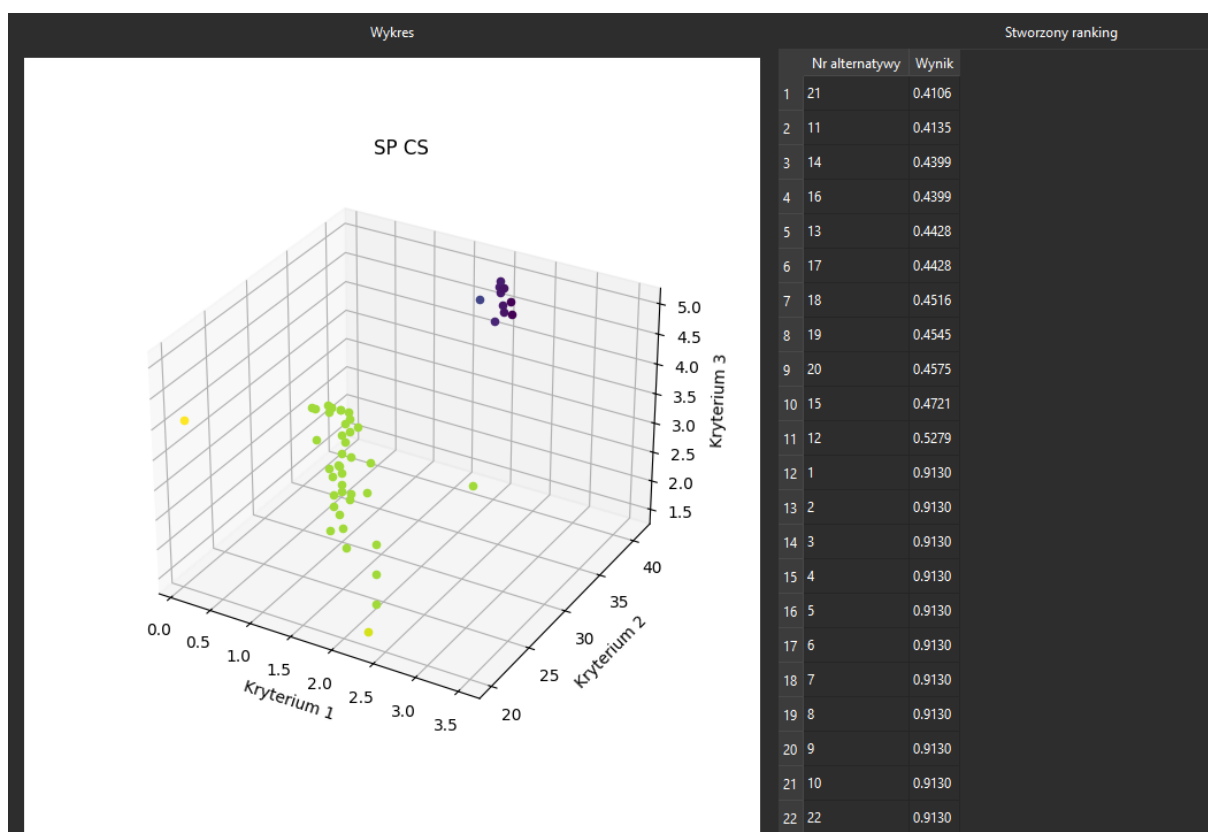
Wariant dyskretny (przykłady obliczeń $N = 3, N = 4$)



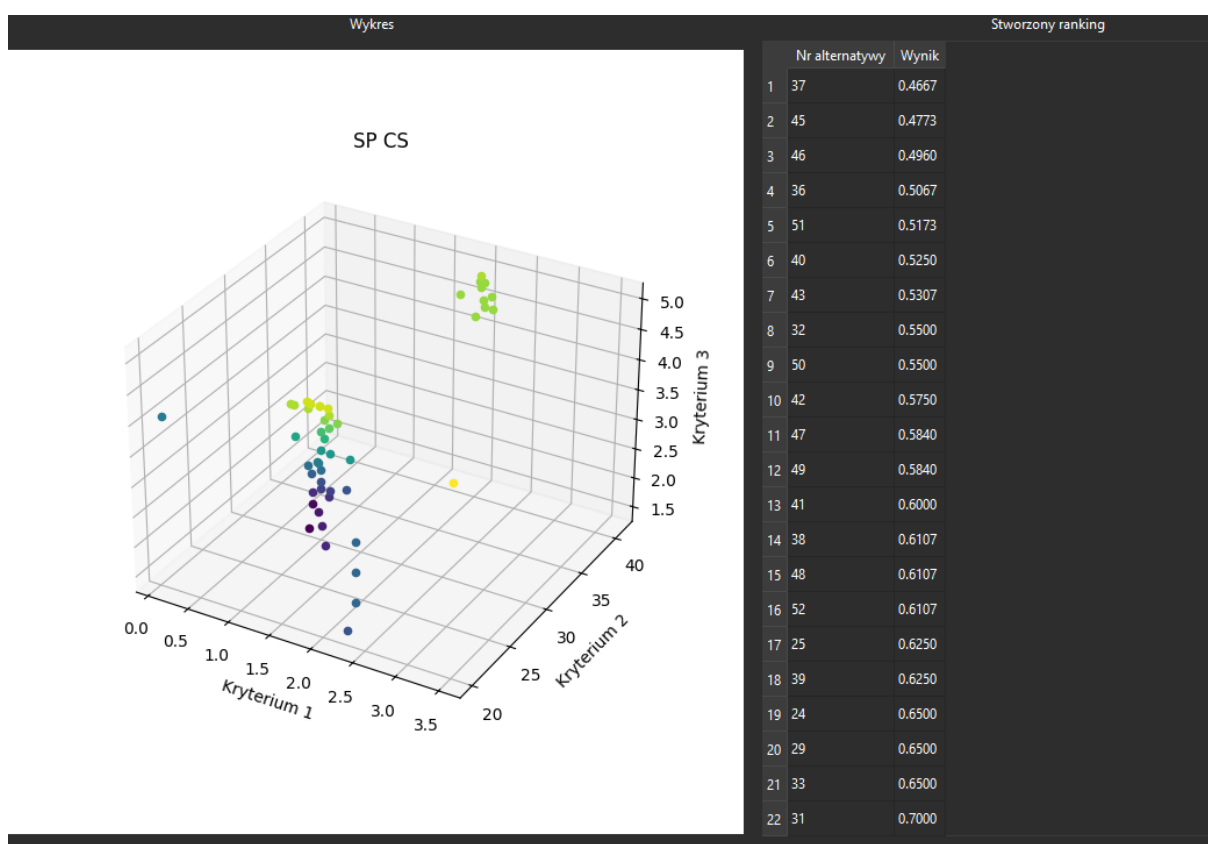
Rys. 13 Wizualizacja działania algorytmu SP-CS dla wariantu dyskretnego, $N=3$, miara Euklidesowa, minimalizacja



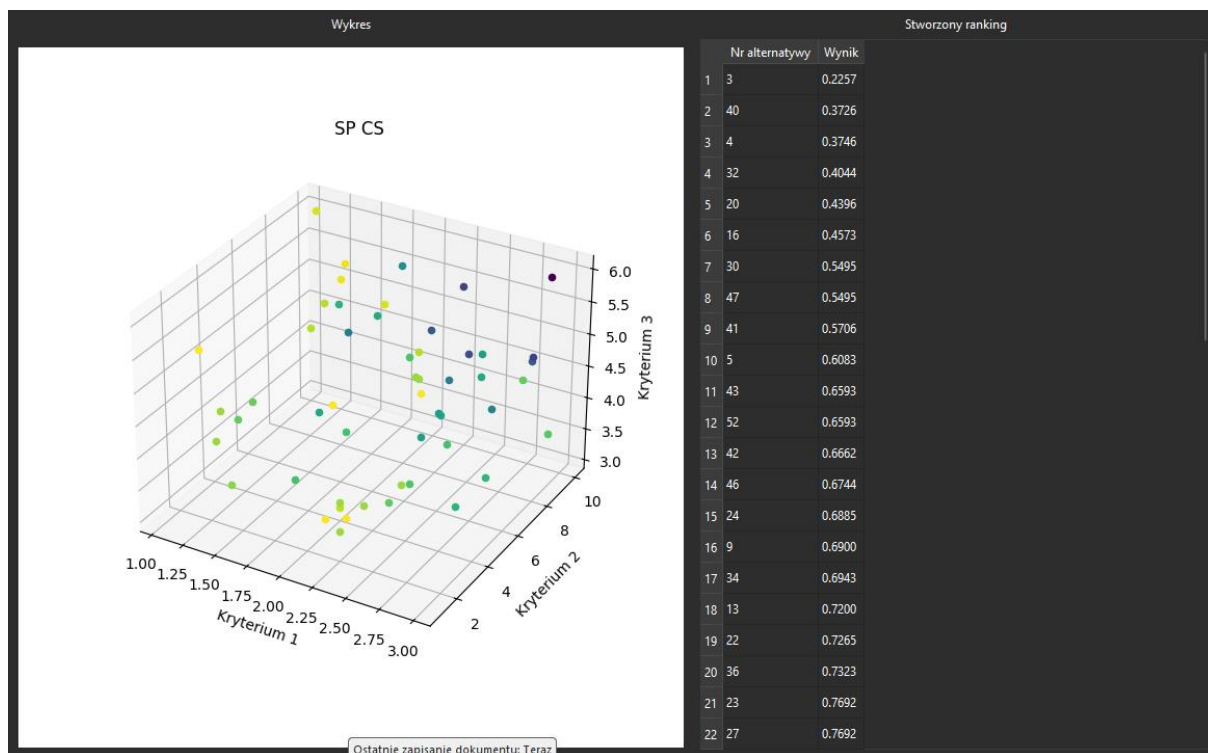
Rys. 14 Wizualizacja działania algorytmu SP-CS dla wariantu dyskretnego, $N=3$, miara Euklidesowa, maksymalizacja



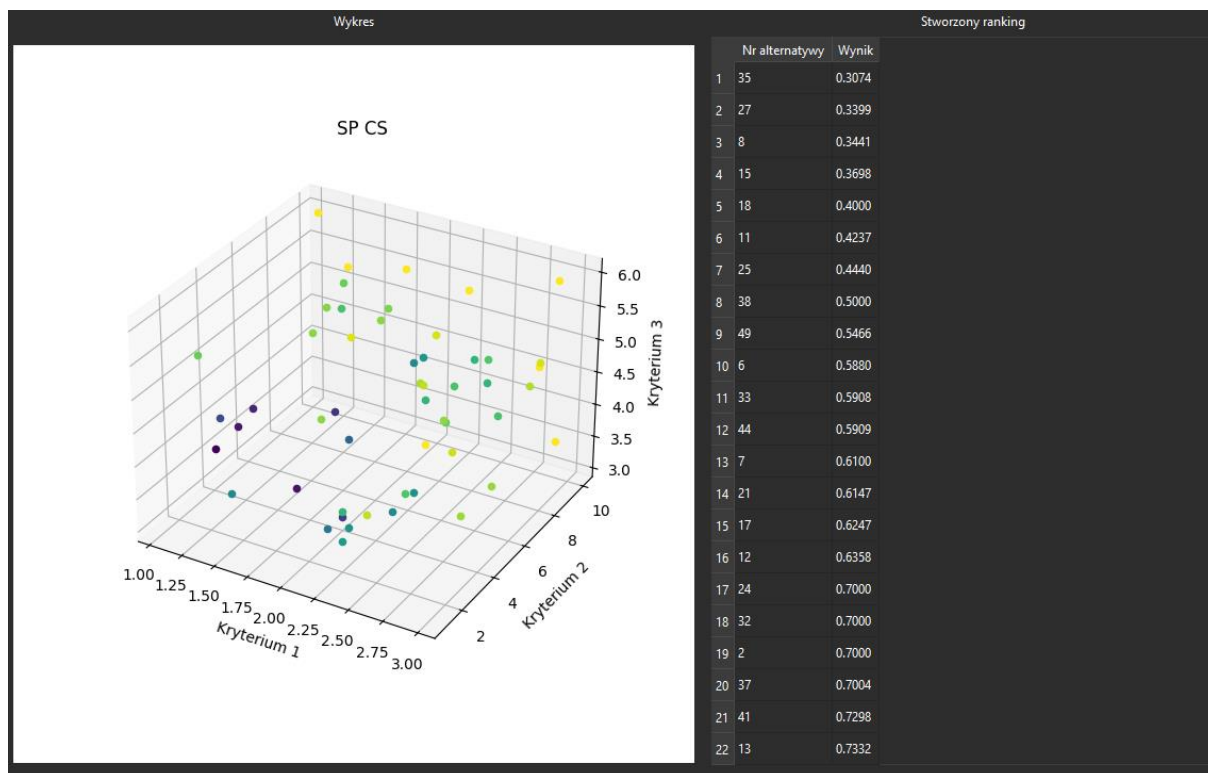
Rys. 15 Wizualizacja działania algorytmu SP-CS dla wariantu dyskretnego, $N=3$, miara Chebysheva, minimalizacja



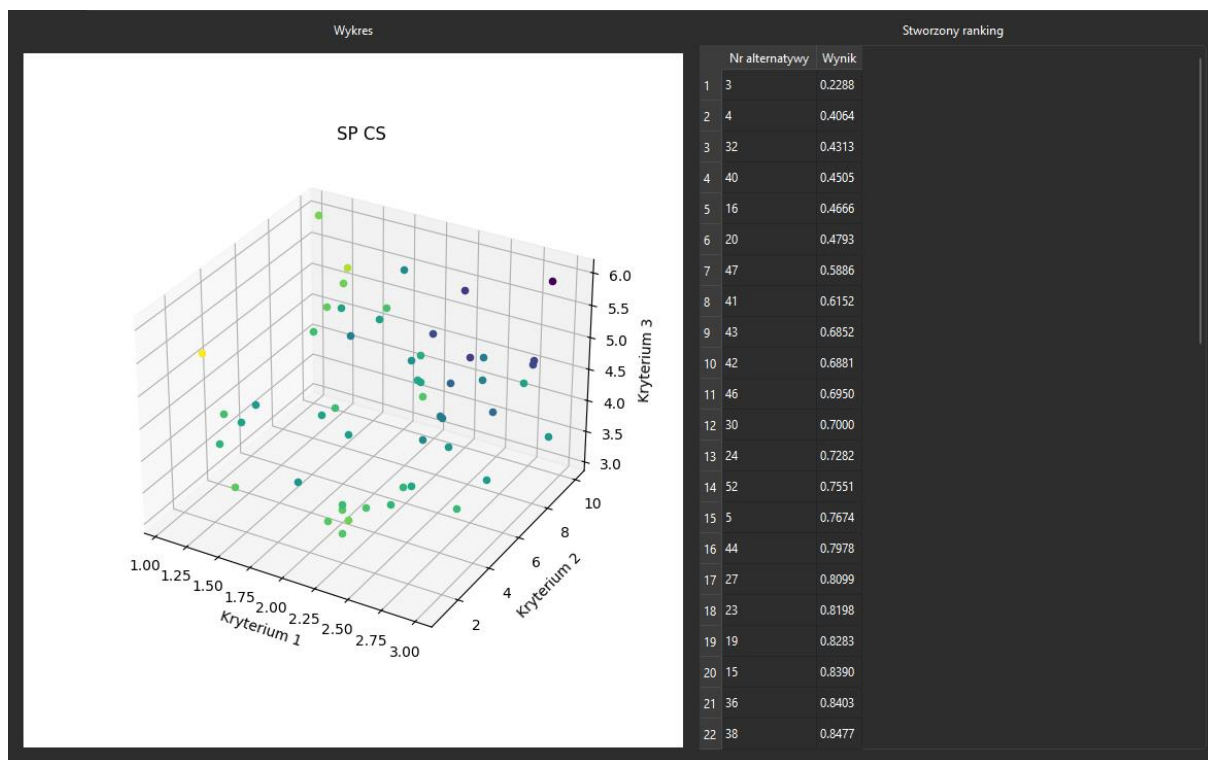
Rys. 16 Wizualizacja działania algorytmu SP-CS dla wariantu dyskretnego, $N=3$, miara Chebysheva, maksymalizacja



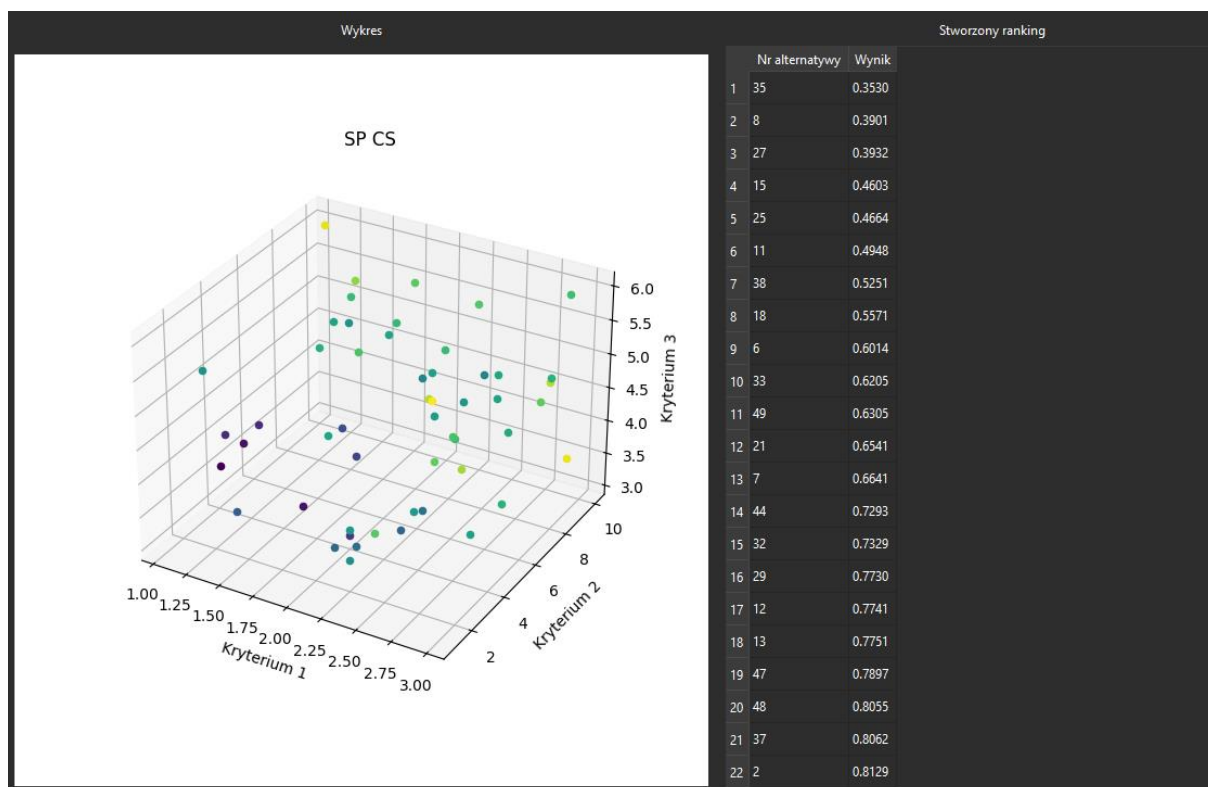
Rys. 17 Wizualizacja działania algorytmu SP-CS dla wariantu dyskretnego, $N=4$, miara Chebysheva, minimalizacja



Rys. 18 Wizualizacja działania algorytmu SP-CS dla wariantu dyskretnego, $N=4$, miara Chebysheva, maksymalizacja



Rys. 19 Wizualizacja działania algorytmu SP-CS dla wariantu dyskretnego, $N=4$, miara Euklidesowa, minimalizacja



Rys. 20 Wizualizacja działania algorytmu SP-CS dla wariantu dyskretnego, $N=4$, miara Euklidesowa, maksymalizacja

UTA DIS

Metoda UTA DIS (UTilité Additive DIScriminant) to rozszerzenie metody UTA (UTilités Additives), skoncentrowane na problemach klasyfikacji. Głównym celem jest przypisanie alternatyw do zdefiniowanych klas uporządkowanych według preferencji, na podstawie wartości funkcji użyteczności. Jest to metoda sortowania, która łączy preferencje decydenta z analizą wielokryterialną i wykorzystuje podejście regresji ordinalnej w celu modelowania preferencji. W metodzie UTA DIS nie są wykorzystywane kryteria odległościowe.

Kluczowe cechy UTA-DIS

1. Klasyfikacja do zdefiniowanych klas:

Alternatywy są przypisywane do predefiniowanych, uporządkowanych klas C_1, C_2, \dots, C_p gdzie klasa C_{h+1} jest bardziej preferowana niż C_h . Klasy mogą być definiowane semantycznie (np. "niska wydajność", "średnia wydajność", "wysoka wydajność").

2. Addytywna funkcja użyteczności:

Funkcja użyteczności $U(a)$ dla każdej alternatywy a jest sumą funkcji cząstkowych dla poszczególnych kryteriów

$$U(a) = \sum_{j=1}^m u_j(g_j(a))$$

gdzie u_j to funkcja cząstkowa dla kryterium g_j , a $g_j(a)$ to wartość a na j -tym kryterium.

3. Regresja ordinalna:

Funkcje cząstkowe u_j są wyznaczane w taki sposób, aby zapewnić zgodność z przykładami przypisania alternatyw do klas podanymi przez decydenta.

4. Modelowanie preferencji:

Metoda pozwala na użycie nie tylko pojedynczych przypisań, ale także zakresów klas, co zwiększa elastyczność w przypadkach niepewności lub braku precyzji w preferencjach.

Etapy algorytmu UTA-DIS

1. Określenie alternatyw A , wartości kryteriów $g_j(a)$ oraz przykładów przypisani alternatyw do klas

2. Formułowanie problemu optymalizacji

Wyznaczanie funkcji użyteczności $U(a)$ poprzez rozwiązanie problemu programowania liniowego, które minimalizuje błędy między wartościami funkcji Ua klasami przypisanymi w przykładach.

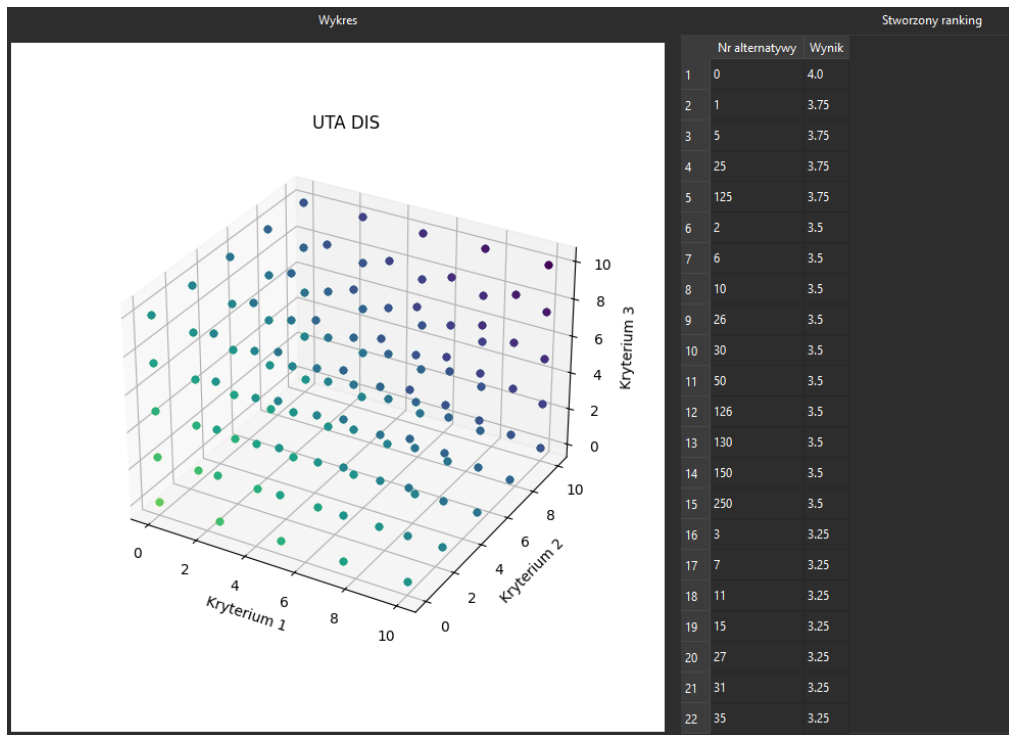
3. Sortowanie alternatyw

Każda alternatywa jest przypisywana do klasy C_h , dla której $U(a)$ spełnia określone progi.

4. Ocena wyników

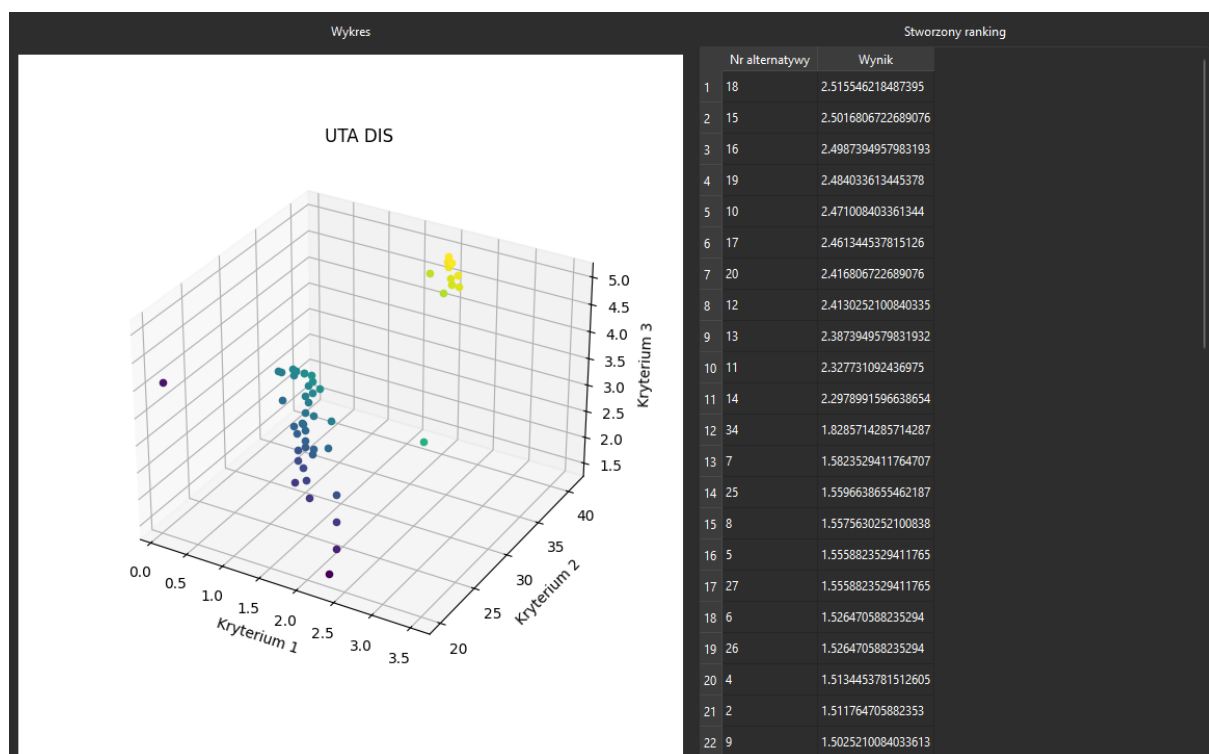
Wynikowa klasyfikacja jest analizowana pod kątem spójności z preferencjami decydenta oraz stabilności w obliczu zmienności danych.

Wariant ciągły (przykłady obliczeń $U \subset R^4, N = 4$)

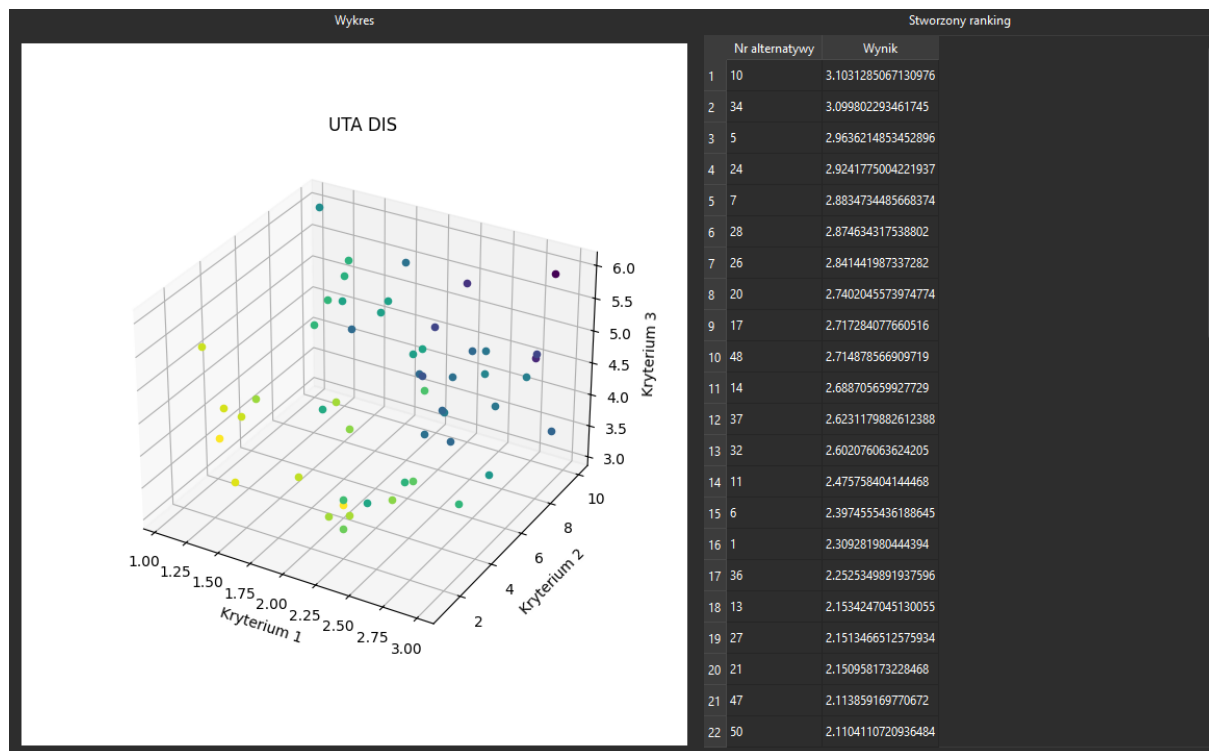


Rys. 21 Minimalizacja dla $N=4$

Wariant dyskretny (przykłady obliczeń $N = 3, N = 4$)



Rys. 22 Maksymalizacja wszystkich kryteriów dla $N=3$



Rys. 23 Minimalizacja wszystkich kryteriów dla $N=4$

Wnioski

W ramach ćwiczenia udało nam się zaimplementować wszystkie z wymaganych metod optymalizacji, zawartych w ćwiczeniu. Utworzono również funkcjonalne GUI, które naśladuje zachowanie tego, wskazanego w konspekcie, a także znacznie upraszcza interpretację wyników eksperymentów.

Metody wykorzystane w zadaniu niekiedy różniły się podejściem czy też parametrami niezbędnymi do ich realizacji, co znacznie utrudniało kwestię związaną z implementacją GUI. Zastosowano więc pewne uproszczenia, które implementują niektóre z parametrów funkcji w sposób statyczny wewnątrz kodu, co nie jest parametryzowane przez użytkownika. Gdybyśmy chcieli nadal rozwijać projekt, byłoby to możliwe do wprowadzenia.

Dobór odpowiednich parametrów, takich jak chociażby wagi jest na tyle istotny, że wpływa na sposób działania algorytmów w sposób znaczący. Ich dobór okazał się problematyczny, co mogło przełożyć się na wyniki naszych funkcji.

Wykorzystane przez nas metody dawały zazwyczaj podobne wyniki. Wymagało to czasami od nas dokładniejszego doboru parametrów, jednak ostatecznie otrzymywaliśmy zadowalający efekt, wskazujący ranking zgodnie z przyjętymi zasadami.

Wizualizacje wyników dla większych zbiorów decyzyjnych ($N=4$) okazały się niewystarczające do pełnej analizy, co wymagało odniesienia się do tabelarycznych wyników.

Jeśli chodzi o Fuzzy TOPSIS, największą trudnością okazała się implementacja liczb rozmytych i operacji na nich przeprowadzanych. Sam algorytm ma raczej prostą budowę i idealnie pokrywa bazowe elementy tworzenia rankingu.

Algorytm RSM wprowadził dodatkowo punkty referencyjne, których ustalenie okazało się na tyle problematyczne, że otrzymane wykresy nie dawały zbyt dobrych rezultatów. Ze względu na problem doboru punktów, powinien być rozważany we współpracy z algorytmem wyszukującym punkty optymalne. Dodatkowo, funkcja skoringowa zawsze jest maksymalizowana, co znacznie utrudnia maksymalizację. Wymaga to wprowadzenia zmian w algorytmie.

Algorytm SP-CS (Safety Principle and Compromise Selection) przysporzył najwięcej trudności z uwagi na unikalne podejście do problemu optymalizacji. Wyniki tej metody były rozrzucone z powodu:

- **Wrażliwości na dobór punktów odniesienia:** Zarówno punkt status quo, jak i punkt aspiracji mają kluczowy wpływ na wyniki. Nieoptymalny wybór tych punktów powoduje niespójności.
- **Wpływu metryki odległości:** Warianty metryki (np. euklidesowa, Czebyszewa) różnie wpływają na ranking, co może prowadzić do istotnych różnic między wynikami.
- **Minimalizacja funkcji scoringowej:** Proces ten wymaga starannego dostosowania parametrów, takich jak położenie punktów na krzywej aspiracji, co w niektórych przypadkach może prowadzić do niestabilnych wyników.

Algorytm UTA DIS, podobnie jak Fuzzy TOPSIS okazał się dość prostym i bazowym algorytmem, który jednak przysporzył problemów podczas implementacji, ze względu na małą dostępność danych o tym algorytmie w instrukcji do ćwiczenia.