

Systemy Wspomagania Decyzji			
Temat: Ocena wielokryterialna platform Cloudowych IIoT oraz AIoT			
Wydział <i>EAIIB</i>		Kierunek <i>Automatyka i Robotyka</i>	Rok <i>III</i>
Zespół <i>Zespół III</i>		Grupa <i>Grupa 1, wtorek 16:45</i>	Data <i>29.01.2025 r.</i>
L.p.	Skład grupy:		
1.	<i>Jakub Janik</i>		
2.	<i>Paweł Jerzyna</i>		

Spis treści:

1. Wstęp
2. Opis zagadnienia
3. Opis programu
4. Opis Algorytmów
5. Badanie Metod wielokryterialnej optymalizacji dyskretnej
6. Podsumowanie i wnioski

1. Wstęp

Współczesne rolnictwo przechodzi dynamiczne zmiany pod wpływem rozwoju nowych technologii, w tym systemów opartych na chmurze obliczeniowej (cloud computing), Internetu rzeczy dla przemysłu (IIoT – Industrial Internet of Things). Wprowadzenie zaawansowanych rozwiązań technologicznych pozwala na efektywniejsze zarządzanie procesami produkcyjnymi, poprawę kontroli środowiska uprawowego oraz optymalizację pracy autonomicznych systemów robotycznych.

Zasadniczym celem niniejszego sprawozdania jest analiza i porównanie wybranych platform cloudowych wspierających działania AIoT w kontekście ich zastosowań w robotyzowanym rolnictwie szklarniowym. Problem wyboru odpowiedniego oprogramowania zostanie przeanalizowany przy użyciu metod wielokryterialnej optymalizacji dyskretnej oraz liczby rozmytej, co umożliwi ocenę systemów pod względem ich efektywności, niezawodności, integracji z autonomicznymi robotami, kosztów oraz możliwości adaptacji do specyficznych warunków

uprawowych. Jednym z kluczowych wyzwań w tym obszarze jest pogodzenie niższych kosztów i łatwej dostępności uniwersalnych systemów cloudowych oferowanych przez globalnych dostawców, takich jak Siemens czy Microsoft, z funkcjonalnościami dedykowanych rozwiązań, takich jak SercoVision czy CloudVision, które są ściśle dostosowane do potrzeb rolnictwa. W ramach analizy podjęta zostanie próba odpowiedzi na pytanie, czy uniwersalne platformy są w stanie zastąpić rozwiązania dedykowane i w jakim zakresie.

2. Opis zagadnienia problemowego

W ramach niniejszego opracowania porównane zostaną platformy cloudowe z uwzględnieniem ich funkcjonalności, kosztów oraz możliwości integracji z systemami AIoT i IIoT. Główny nacisk zostanie położony na platformy SercoVision i CloudVision, które stanowią dedykowane rozwiązania dla rolnictwa, a także na dwie dodatkowo wybrane platformy: Agro, oraz Demeter. Podstawowe informacje dotyczące omawianych platform przedstawiono w tabeli 1. Głównym źródłem, które posłużyło do pozyskania danych był raport Gartnera^[1]. Ceny zostały oszacowane za pomocą narzędzi sztucznej inteligencji.

Platforma	Opis	Cena (szacunkowa)	Funkcjonalności kluczowe
SercoVision	Dedykowana platforma do zarządzania szklarniami, oferuje wsparcie dla robotów mobilnych i monitorowanie warunków środowiskowych.	5 000-10 000 PLN/miesiąc	Integracja z systemami VPN, monitoring parametrów szklarni, kontrola na poziomie lokalnym.
CloudVision	Nowsze rozwiązanie, niż SercoVision. Rozwiązanie chmurowe dedykowane do upraw kontrolowanych, oferujące moduły AI do optymalizacji procesów.	20 000-40 000 PLN/miesiąc	Moduły sztucznej inteligencji, możliwość modelowania upraw, wsparcie robotów RaaS.
Oracle IoT Cloud	Uniwersalna platforma IoT wspierająca szeroką gamę urządzeń IoT, integruje się z systemami AI.	15 000-25 000 PLN/miesiąc	Analiza danych IoT, integracja z AI, szerokie wsparcie techniczne.
Microsoft Azure IoT	Wszechstronna platforma IoT oferująca zaawansowaną analitykę i integrację z chmurą Azure.	15 000-30 000 PLN/miesiąc	Analityka IoT, integracja z AI, wysoka skalowalność.
PTC ThingWorx	Platforma przemysłowa IoT, oferująca zaawansowane zarządzanie produkcją i analizę danych w czasie rzeczywistym.	8 000-15 000 PLN/miesiąc	Wsparcie dla automatyzacji, moduły analizy defektów produkcyjnych, integracja z systemami IoT.
IBM Watson IoT	Rozbudowane narzędzie IoT, wspierające analizy dużych zbiorów danych i rozwój modeli AI.	15 000-25 000 PLN/miesiąc	Integracja z AI, analiza predykcyjna, zarządzanie dużymi środowiskami IoT.
Siemens	Przemysłowa platforma IIoT	20 000-40	Optymalizacja operacji,

MindSphere	od Siemens, dedykowana do zarządzania dużymi infrastrukturami.	000 PLN/miesiąc	monitorowanie maszyn, integracja z urządzeniami przemysłowymi.
Software AG Cumulocity IoT	Platforma IoT, umożliwiająca zarządzanie flotą urządzeń w czasie rzeczywistym.	8 000-12 000 PLN/miesiąc	Obsługa urządzeń IoT, łatwa konfiguracja, integracja z systemami operacyjnymi.
Altizon Datonis IoT	Narzędzie wspierające analizy w produkcji i zarządzaniu procesami przemysłowymi.	15 000-30 000 PLN/miesiąc	Automatyzacja produkcji, monitorowanie procesów, wsparcie analityczne.

Hitachi Lumada	Platforma IoT, wspierająca optymalizację procesów przemysłowych i rolniczych.	15 000-25 000 PLN/miesiąc	Wsparcie analityczne, integracja z chmurą, zarządzanie danymi w czasie rzeczywistym.
OVHcloud	Uniwersalna platforma chmurowa o niskich kosztach utrzymania i elastycznej strukturze.	5 000-12 000 PLN/miesiąc	Hosting danych IoT, analiza podstawowa, integracja z różnymi systemami IoT.
Atos Codex IoT	Narzędzie IoT wspierające analizę danych i integrację z systemami AI.	3 000-7 000 PLN/miesiąc	Integracja z AI, analiza danych w czasie rzeczywistym, modularność.
iLens	Zaawansowana platforma IoT wspierająca analitykę predykcyjną i modele AI, zorientowana na zarządzanie dużymi środowiskami przemysłowymi oraz optymalizację procesów operacyjnych.	20 000-35 000 PLN/miesiąc	Analityka predykcyjna, modele AI, zarządzanie danymi w czasie rzeczywistym.
Accenture Connected Platforms as a Service	Platforma oferująca podstawową funkcjonalność IoT, z naciskiem na integrację danych z różnych źródeł oraz zarządzanie prostymi procesami operacyjnymi.	5 000-10 000 PLN/miesiąc	Agregacja danych z różnych źródeł, monitoring podstawowych parametrów, niskie koszty utrzymania.
Everyware IoT	Rozwiązanie IoT dedykowane do podstawowego zarządzania urządzeniami i monitoringiem danych w czasie rzeczywistym, z ograniczonym wsparciem zaawansowanej analityki.	3 000-7 000 PLN/miesiąc	Zarządzanie urządzeniami IoT, łatwa konfiguracja, podstawowa analityka danych.
SAP Leonardo IoT	Podstawowa platforma IoT umożliwiająca analizy danych środowiskowych i integrację z procesami operacyjnymi, dedykowana dla średnich i małych przedsiębiorstw.	3 000-8 000 PLN/miesiąc	Analiza danych środowiskowych, integracja z systemami ERP, podstawowa modularność.
SAP Industry Cloud	Wszechstronna platforma chmurowa z funkcjonalnościami IoT, dedykowana dla przedsiębiorstw przemysłowych, wspierająca zaawansowaną analitykę oraz optymalizację procesów produkcyjnych.	15 000-30 000 PLN/miesiąc	Zaawansowana analityka IoT, optymalizacja procesów produkcyjnych, skalowalność.

Davra IoT Platform	Rozwiązanie IoT wspierające analizy operacyjne i zarządzanie flotą urządzeń.	15 000-30 000 PLN/miesiąc	Analiza predykcyjna, monitorowanie urządzeń, wsparcie dla IoT w rolnictwie.
Predix Platform	Platforma IoT od GE, dedykowana do analizy procesów w dużych przedsiębiorstwach.	5 000-10 000 PLN/miesiąc	Optymalizacja produkcji, zarządzanie urządzeniami, analiza predykcyjna.
PI System	Rozwiązanie do integracji danych z różnych źródeł IoT, wspierające analizy.	10 000-20 000 PLN/miesiąc	Agregacja danych, integracja z systemami AI, monitoring procesów.
Demeter	Platforma wspierająca zarządzanie uprawami oraz monitorowanie warunków środowiskowych w czasie rzeczywistym.	3 000-8 000 PLN/miesiąc	Moduły monitoringu w czasie rzeczywistym, integracja z AI i IoT.
Agro	Dedykowane narzędzie dla rolnictwa, wspierające automatyzację upraw i zarządzanie danymi środowiskowymi.	2 000-7 000 PLN/miesiąc	Analiza danych środowiskowych, integracja z systemami robotów mobilnych.

Tabela 1. Zestawienie wybranych platform cloudowych do porównania

SercoVision to zaawansowana platforma przeznaczona do zarządzania środowiskami szklarniowymi, integrująca się z systemami monitoringu parametrów atmosferycznych i umożliwiającą precyzyjne sterowanie warunkami upraw. Architektura SercoVision obejmuje moduł akwizycji danych, który wykorzystuje sieci czujników środowiskowych do pomiaru zmiennych takich jak temperatura, wilgotność, stężenie CO₂ oraz poziom natężenia oświetlenia. Dane te są przetwarzane w czasie rzeczywistym przez system analizy statystycznej i modele regresyjne, takie jak regresja liniowa oraz XGBoost, umożliwiając predykcję zmian warunków i zalecenie działań korekcyjnych.

Platforma obsługuje lokalne zarządzanie parametrami szklarni za pomocą interfejsu przeglądarkowego oraz aplikacji mobilnej. SercoVision implementuje protokoły komunikacyjne zgodne z IIoT i AIoT, co pozwala na integrację z autonomicznymi robotami mobilnymi i innymi systemami automatyzacji. Zastosowanie tych technologii umożliwia automatyczne wdrażanie strategii regulacji mikroklimatu oraz optymalizację zużycia zasobów. Platforma obsługuje standardowe interfejsy API, umożliwiające połączenie z zewnętrznymi systemami zarządzania produkcją oraz analityką big data.

W porównaniu do SercoVision, CloudVision charakteryzuje się rozproszoną architekturą opartą na chmurze, co eliminuje konieczność lokalnej infrastruktury obliczeniowej. System ten integruje algorytmy uczenia maszynowego i sztucznej inteligencji w celu analizy dużych zbiorów danych środowiskowych. Dzięki wykorzystaniu metod przetwarzania

strumieniowego, CloudVision umożliwia przewidywanie zmian w czasie rzeczywistym oraz dynamiczne dostosowywanie parametrów uprawowych na podstawie historycznych i bieżących danych sensorycznych.

Dostępność CloudVision jest niezależna od lokalizacji użytkownika, co pozwala na zdalne zarządzanie uprawami za pomocą przeglądarki internetowej lub aplikacji mobilnej. Platforma wykorzystuje modele symulacyjne do predykcji wpływu warunków środowiskowych na rozwój roślin i implementuje systemy automatycznej kontroli, obejmujące nawadnianie, nawożenie oraz procesy zbiorcze. Kompatybilność z wieloma standardami IoT umożliwia płynną integrację z istniejącymi systemami automatyzacji szklarni.

Dodatkową funkcjonalnością CloudVision jest moduł analizy kosztowej, który monitoruje zużycie zasobów, takich jak energia elektryczna i woda, wspomagając optymalizację operacyjną gospodarstw szklarniowych. System ten zapewnia kompleksową analizę ekonomiczną procesów uprawowych, co umożliwia precyzyjne planowanie budżetu i maksymalizację efektywności kosztowej. Dzięki zastosowaniu nowoczesnych technologii obliczeniowych oraz zaawansowanych algorytmów predykcyjnych, CloudVision stanowi bardziej elastyczne i skalowalne rozwiązanie w porównaniu do lokalnie działającego SercoVision.

Do efektywnej analizy platform IoT w kontekście agrokulturalnym wykorzystano 5 kryteriów porównawczych. Kryteria porównania stanowią kluczowy element analizy, pozwalając na systematyczną ocenę platform cloudowych w kontekście ich użyteczności, efektywności oraz dostosowania do specyficznych wymagań użytkowników. Oto szczegółowy opis poszczególnych kryteriów:

1. Ocena i negocjowanie kontraktu

Kryterium to obejmuje analizę procesu wyboru i negocjacji warunków współpracy z dostawcą platformy. W szczególności oceniana jest:

- Przejrzystość ofert i umów,
- Elastyczność dostosowania warunków kontraktu do potrzeb klienta,
- Dostępność testowych wersji platformy lub okresu próbnego, który pozwala na ocenę funkcjonalności przed podpisaniem umowy. Istotnym aspektem jest także szybkość odpowiedzi dostawcy na zapytania oraz ogólna prostota procesu wyboru.

2. Integracja i wdrożenie

To kryterium dotyczy łatwości, z jaką platforma może zostać zintegrowana z istniejącą infrastrukturą technologiczną użytkownika. Analizowane elementy obejmują:

- Wymagania sprzętowe i zgodność platformy z obecnymi systemami użytkownika,
- Czas wdrożenia systemu oraz ilość zasobów niezbędnych do pełnego uruchomienia,
- Dostępność szczegółowej dokumentacji technicznej, narzędzi do konfiguracji i szkoleń dla użytkowników. Oceniana jest również intuicyjność procesu implementacji, co wpływa na szybkie rozpoczęcie korzystania z systemu

3. Obsługa i wsparcie techniczne

W tym kryterium oceniana jest jakość usług serwisowych i wsparcia technicznego oferowanych przez dostawcę platformy. Kluczowe elementy obejmują:

- Szybkość reakcji na zgłoszenia techniczne i dostępność wsparcia 24/7,
- Zakres usług dodatkowych, takich jak regularne aktualizacje systemu, wdrażanie poprawek oraz oferowanie szkoleń dla personelu użytkownika,
- Kompetencje techniczne i profesjonalizm zespołu wsparcia.

4. Specyficzne możliwości dla rolnictwa szklarniowego

To kryterium odnosi się do funkcjonalności platform dedykowanych rolnictwu szklarniowemu (Smart Horticulture). Analiza obejmuje:

- Dostępność zaawansowanych modułów do monitorowania środowiska (np. wilgotności, temperatury, oświetlenia),
- Możliwość modelowania i symulacji procesów uprawowych w czasie rzeczywistym,
- Integrację platformy z autonomicznymi robotami mobilnymi i ich efektywne zarządzanie. Oceniana jest także adaptacja do specyficznych potrzeb gospodarstw szklarniowych, takich jak zmienne warunki klimatyczne.

5. Całkowite koszty użytkowania

Kryterium obejmuje analizę wszystkich kosztów związanych z wdrożeniem i eksploatacją platformy. Uwzględniane są:

- Opłaty licencyjne, koszty serwisowania oraz ewentualne opłaty za szkolenia użytkowników,
- Koszty modernizacji i aktualizacji systemu w trakcie użytkowania,

- Możliwości skalowania platformy w miarę wzrostu wymagań użytkownika.
Porównanie kosztów dla różnych platform pozwoli na ocenę ich opłacalności w dłuższej perspektywie czasowej.

Do oceny platform w badaniu wykorzystano pięciostopniową skalę Likerta, która została zmodyfikowana tak, aby uwzględniała liczby rozmyte trójkątne. Liczby te pozwalają na ujęcie niepewności związanej z ocenami oraz różnic w poziomie pewności, z jaką użytkownicy przypisują konkretne wartości do platform.

Każdy poziom skali Likerta został przypisany do trójkątnej liczby rozmytej, która określa: Minimalną wartość (dolny zakres), Wartość centralną (najbardziej prawdopodobna wartość) oraz Maksymalną wartość (górny zakres).

Skala ocen została przedstawiona w tabeli 2:

Term shortcut	Likert scale Q terms	Crisp numerical value (v)	Fuzzy triangular quantification	
			No self-assessment of uncertainty	With uncertainty self-assessment u
1	2	3	4	5
L	Low score	$v_{min} = 1$	(1, 1, 2)	$(v_{min}, v_{min}, \min(v_{min} + u \cdot d_i, v_{max}))$ $(\max(v_{min}, v - u \cdot d_i), v, \min(v + u \cdot d_i, v_{max}))$
BL	Above Low	2	(1, 2, 3)	
A	Average	3	(2, 3, 4)	
AA	Above Average	4	(3, 4, 5)	
H	High score	$v_{max} = 5$	(4, 5, 5)	$(\max(v_{min}, v_{max} - u \cdot d_i), v_{max}, v_{max})$

Tabela 2. Skala Likerta wraz z określeniem przedziałów niepewności i liczbami rozmytymi

Taki system pozwala na wprowadzenie informacji o poziomie pewności użytkownika co do przypisanej oceny. Jeśli użytkownik czuje się mniej pewny, poziom niepewności w liczbie rozmytej zwiększa się, co prowadzi do poszerzenia jej zakresu.

Na przykład, jeśli użytkownik przypisze ocenę „AA” (Above Average) z pewnością ocenioną na poziomie „A” (Average), trójkątna liczba rozmyta zostanie skonstruowana jako:

- Dolna granica: 3,5 (uwzględnienie niższej pewności),
- Wartość centralna: 4,
- Górna granica: 4,5 (poszerzenie przedziału w związku ze średnią pewnością).

Co istotne dla liczb skrajnych zastosowano inne rozmycie niż dla liczb środkowych. Parametry dobrano tak by pola wyznaczone przez poszczególne wierzchołki dla danych wartości zadanych niepewności

pozostały niezmiennie. Dokładny sposób tworzenia liczb trójkątnych przedstawiono w tabeli 3:

Niepewność\ Ocena	L	BL	A	AA	H
H	(1.00, 1.00, 1.00)	(2.00, 2.00, 2.00)	(3.00, 3.00, 3.00)	(4.00, 4.00, 4.00)	(5.00, 5.00, 5.00)
AA	(1.12, 1.00, 1.62)	(1.75, 2.00, 2.25)	(2.75, 3.00, 3.25)	(3.75, 4.00, 4.25)	(4.88, 5.00, 4.38)
A	(1.25, 1.00, 1.75)	(1.50, 2.00, 2.50)	(2.50, 3.00, 3.50)	(3.50, 4.00, 4.50)	(4.75, 5.00, 4.25)
BL	(1.50, 1.00, 2.50)	(1.25, 2.00, 2.75)	(2.25, 3.00, 3.75)	(3.25, 4.00, 4.75)	(4.50, 5.00, 3.50)
L	(2.00, 1.00, 4.00)	(1.00, 2.00, 3.00)	(2.00, 3.00, 4.00)	(3.00, 4.00, 5.00)	(4.00, 5.00, 2.00)

Tabela 3. Wartości liczb rozmytych w przyjętym systemie oceniania

Jeśli użytkownik ocenia platformę CloudVision jako „AA” (Above Average) z pewnością „A” (Average), odpowiadająca liczba rozmyta uwzględnia niepewność użytkownika co do jego wyboru, dając bardziej realistyczny obraz oceny. Tak skonstruowane liczby rozmyte mogą być później wykorzystane w metodach takich jak TOPSIS, aby uszeregować platformy według ich jakości. System ten pozwala na większą precyzję oceny oraz uwzględnia subiektywną niepewność użytkownika, co czyni go bardziej elastycznym w stosunku do tradycyjnych metod oceniania.

No.	Reference point class	Q-scale vector	De-fuzzified Likert scale vector	Expert certainty U -assessment	Reference point interpretation
1	2		3	4	
1	R_1		(1, 1, 2, 2)	4	Target solution proposed by Expert 1.
2	R_1		(1, 2, 2, 3)	5	Target solution proposed by Expert 2.
3	R_2		(2, 3, 3, 4)	5	Assessment of currently used SercoVision system
4	R_2		(2, 3, 3, 3)	4	Assessment of a system offered by in-house software department
5	R_3		(1, 1, 2, 1)	3	Parameters of software that was used at another similar covered crop and failed
6	R_3		(1, 2, 2, 2)	3	As above, but a different crop
7	R_4		(5, 5, 5, 5)	5	$\mathbf{x}^*(A)$ - fuzzy ideal point of A

Tabela 4. Zdefiniowane punkty referencyjne

W celu określenia, jak analizowane platformy plasują się dla naszego zestawu danych, wykorzystano wyniki ocen trzech niezależnych ekspertów. Każdy z nich przeprowadził ocenę platform zgodnie z przyjętymi wcześniej kryteriami, uwzględniającymi m.in. integrację z autonomicznymi robotami szklarniowymi, zaawansowane narzędzia do monitorowania środowiska uprawowego, wsparcie dla analizy predykcyjnej

oraz całkowite koszty użytkowania. Oceny te były następnie przetwarzane przez pięć różnych metod wielokryterialnej analizy decyzyjnej: MREF, RSM, TOPSIS, UTA STAR oraz VIKOR. Dokładne oceny znajdują się w arkuszach programu excel załączonych do projektu.

3. Opis Programu

Program składa się z intuicyjnego graficznego interfejsu użytkownika (GUI) oraz trzech 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 rankingów oraz prezentacja wyników w czytelnej i uporządkowanej formie. Program zapewnia kompleksowe wsparcie w procesie podejmowania decyzji, łącząc zaawansowane możliwości analityczne z prostotą obsługi. Dane podane w skali Likerta są automatycznie przerabiane na trójkątne liczby rozmyte.

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

GUI

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 Spearman

Porównaj metody tau-Kendall

Rys 1. Główny interfejs użytkownika

Sekcja Importu Danych umożliwia użytkownikowi wczytanie danych wejściowych z pliku w formacie Excel (.xlsx). Dzięki temu użytkownik może w prosty sposób zaimportować dane, 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.

GUI

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 Spearman

Porównaj metody tau-Kendall

Rys 2. Pole wczytaj dane z pliku

Wybór Metody i Generowanie Rankingu pozwala na wskazanie jednego z trzech 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.

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 Spearman

Porównaj metody tau-Kendall

Rys 3. Pole 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.

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 Spearman

Porównaj metody tau-Kendall

Rys 4. Tabela alternatyw z kryteriami

W **Sekcji Rankingu** 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 dalsze przetwarzanie lub prezentację danych.

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 Spearman

Porównaj metody tau-Kendall

Rys 5. Sekcja przedstawiająca stworzony ranking

Przyciski do **Porównania metod** w aplikacji służą 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. Aplikacja wykonuje obliczenia przy użyciu metod decyzyjnych: Topsis (GT), RSM, MREF. 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. Ponadto tworzony jest wykres przedstawiający odległość uzyskanych wyników od punktu idealnego oraz macierze korelacji Tau-Kendalla i Spearmana.

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 Spearman

Porównaj metody tau-Kendall

Rys 6. Przycisk Porównaj metody

4. Opis algorytmów

Zmodyfikowano algorytmy zaimplementowane w ćwiczeniu 5 w sposób taki sposób, aby utworzyć możliwość działania na liczbach rozmytych.

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. Poniżej przedstawiono zmodyfikowany algorytm TOPSIS GT

```
function [ranking, dist_positive] = Topsis_Fuzzy(matrix_of_matrices, weights, types, dec_weights)
    % Topsis_Fuzzy: Modified TOPSIS for triangular fuzzy numbers
    % matrix_of_matrices: 3D matrix containing m x n x d fuzzy values
    % weights: vector of criteria weights (n-element)
    % types: vector of criteria types (1 for maximizing, -1 for minimizing)
    % dec_weights: weights of decision-makers (d-element vector)

    [m, n, d] = size(matrix_of_matrices); % m: alternatives, n: criteria, d: decision-makers

    % Step 1: Aggregate decision-maker evaluations using weights
    aggregated_data = zeros(m, n, 3); % Fuzzy values have three components (low, mid, high)
    for k = 1:d
        aggregated_data = aggregated_data + matrix_of_matrices(:, :, k) * dec_weights(k);
    end
    aggregated_data = aggregated_data / sum(dec_weights); % Normalize by total weight

    % Step 2: Normalize fuzzy data
    normalized_data = zeros(m, n, 3);
    for j = 1:n
        min_val = min(aggregated_data(:, j, 1)); % Min of low values
        max_val = max(aggregated_data(:, j, 3)); % Max of high values

        for i = 1:m
            % Normalize each triangular component
            normalized_data(i, j, 1) = (aggregated_data(i, j, 1) - min_val) / (max_val - min_val);
            normalized_data(i, j, 2) = (aggregated_data(i, j, 2) - min_val) / (max_val - min_val);
            normalized_data(i, j, 3) = (aggregated_data(i, j, 3) - min_val) / (max_val - min_val);
        end
    end

    % Step 3: Calculate weighted normalized fuzzy data
    weighted_data = zeros(m, n, 3);
    for j = 1:n
        for i = 1:m
            weighted_data(i, j, :) = normalized_data(i, j, :) * weights(j);
        end
    end

    % Step 4: Determine fuzzy positive and negative ideal solutions
    fp = zeros(n, 3); % Fuzzy positive ideal
    fn = zeros(n, 3); % Fuzzy negative ideal

    for j = 1:n
        if types(j) == 1 % Maximizing criterion
            fp(j, :) = [max(weighted_data(:, j, 1)), max(weighted_data(:, j, 2)), max(weighted_data(:, j, 3))];
            fn(j, :) = [min(weighted_data(:, j, 1)), min(weighted_data(:, j, 2)), min(weighted_data(:, j, 3))];
        else % Minimizing criterion
            fp(j, :) = [min(weighted_data(:, j, 1)), min(weighted_data(:, j, 2)), min(weighted_data(:, j, 3))];
            fn(j, :) = [max(weighted_data(:, j, 1)), max(weighted_data(:, j, 2)), max(weighted_data(:, j, 3))];
        end
    end

    % Step 5: Calculate distances to ideal solutions
    dist_positive = zeros(m, 1);
    dist_negative = zeros(m, 1);

    for i = 1:m
        for j = 1:n
            % Distance to positive ideal
            dist_positive(i) = dist_positive(i) + sum((squeeze(weighted_data(i, j, :))' - fp(j, :)).^2);
            % Distance to negative ideal
            dist_negative(i) = dist_negative(i) + sum((squeeze(weighted_data(i, j, :))' - fn(j, :)).^2);
        end
    end
    dist_positive = sqrt(dist_positive);
    dist_negative = sqrt(dist_negative);

    % Step 6: Calculate closeness coefficient
    closeness = dist_negative ./ (dist_positive + dist_negative);

    % Step 7: Create ranking
    [~, ranking] = sort(closeness, 'descend');
end
```

Rys7. 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. Poniżej przedstawiono zmodyfikowany algorytm RSM

```

function [ranking, dist_to_ideal, final_scores] = fuzzy_rsm(data, weights, lambda)
% FUZZY_RSM - Reference Set Method for Fuzzy Numbers
% data - 3D matrix with fuzzy numbers (NxMx3), where each entry is a triangular fuzzy number
%         represented by [left, middle, right].
% weights - vector of weights for each criterion
% lambda - balancing coefficient for distances to ideal and anti-ideal points

[N, M, ~] = size(data);

% Extract fuzzy components
left = data(:, :, 1);
middle = data(:, :, 2);
right = data(:, :, 3);

% Normalization for each fuzzy component
min_left = min(left, [], 1);
max_left = max(left, [], 1);
min_middle = min(middle, [], 1);
max_middle = max(middle, [], 1);
min_right = min(right, [], 1);
max_right = max(right, [], 1);

norm_left = (left - min_left) ./ (max_left - min_left);
norm_middle = (middle - min_middle) ./ (max_middle - min_middle);
norm_right = (right - min_right) ./ (max_right - min_right);

% Weighted fuzzy components
weighted_left = norm_left .* weights;
weighted_middle = norm_middle .* weights;
weighted_right = norm_right .* weights;

% Fuzzy ideal and anti-ideal points
ideal_point = [max(weighted_left, [], 1); max(weighted_middle, [], 1); max(weighted_right, [], 1)];
anti_ideal_point = [min(weighted_left, [], 1); min(weighted_middle, [], 1); min(weighted_right, [], 1)];

% Calculate distances to ideal and anti-ideal points using centroid method
centroid_data = (norm_left + norm_middle + norm_right) / 3;
centroid_ideal = (ideal_point(1, :) + ideal_point(2, :) + ideal_point(3, :)) / 3;
centroid_anti_ideal = (anti_ideal_point(1, :) + anti_ideal_point(2, :) + anti_ideal_point(3, :)) / 3;

dist_to_ideal = sqrt(sum((centroid_data - centroid_ideal).^2, 2));
dist_to_anti_ideal = sqrt(sum((centroid_data - centroid_anti_ideal).^2, 2));

% Final scores using lambda
final_scores = lambda * dist_to_anti_ideal - (1 - lambda) * dist_to_ideal;

% Ranking based on scores
[~, ranking_indices] = sort(final_scores, 'descend');
ranking = ranking_indices;
end

```

Rys 8. 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. Poniżej przedstawiono zmodyfikowany algorytm MREF

```

function [ranking, min_distances] = MREF(data, weights, types)
% MREF - metoda wielokryterialnej optymalizacji z miarą Czebyszewa dla wartości rozmytych
% data - macierz alternatyw (wiersze: alternatywy, kolumny: kryteria, 3. wymiar: [dolna, modalna, górna])
% weights - wagi dla kryteriów (wektor kolumnowy)
% types - typy kryteriów (1 = maksymalizują, -1 = minimalizują)
% ranking - ranking alternatyw od najlepszej do najgorszej
% min_distances - minimalne odległości dla każdej alternatywy

% Sprawdzenie poprawności danych wejściowych
if size(data, 2) ~= length(weights) || length(weights) ~= length(types)
    error('Rozmiar wag i typów kryteriów musi być zgodny z liczbą kolumn w macierzy danych.');
```

end

```

% Punkty referencyjne (domyślnie: minimum, maksimum, mediany w każdej kolumnie dla wartości rozmytych)
references = compute_fuzzy_references(data);

% Normalizacja kryteriów
[normalized, normalized_ref] = normalize_fuzzy_criteria(data, types, references);

% Obliczanie odległości Czebyszewa dla wartości rozmytych
distances = Czebyszew_Fuzzy(normalized, normalized_ref);

% Minimalna odległość względem dowolnego punktu referencyjnego
min_distances = min(distances, [], 2);

% Ranking alternatyw (od najlepszej do najgorszej)
[~, ranking_indices] = sort(min_distances);

% Zwracanie wynikowego rankingu i odległości
ranking = ranking_indices;
end

function references = compute_fuzzy_references(data)
% Obliczanie punktów referencyjnych dla wartości rozmytych
num_criteria = size(data, 2);
references = zeros(3, num_criteria, 3); % [min, max, median]

for i = 1:num_criteria
    criterion_data = squeeze(data(:, i, :));
    references(1, i, :) = min(criterion_data, [], 1); % Minimum
    references(2, i, :) = max(criterion_data, [], 1); % Maksimum
    references(3, i, :) = median(criterion_data, 1); % Mediana
end
end

function [normalized, normalized_ref] = normalize_fuzzy_criteria(data, types, references)
[num_alternatives, num_criteria, dim3] = size(data);
if dim3 == 1
    % Klasyczna normalizacja (2D)
    normalized = zeros(size(data));
    normalized_ref = zeros(size(references));
    for i = 1:num_criteria
        if types(i) == -1 % Minimalizacja
            min_val = min(data(:, i));
            max_val = max(data(:, i));
            normalized(:, i) = (data(:, i) - min_val) / (max_val - min_val);
            normalized_ref(:, i) = (references(:, i) - min_val) / (max_val - min_val);
        elseif types(i) == 1 % Maksymalizacja
            min_val = min(data(:, i));
            max_val = max(data(:, i));
            normalized(:, i) = (max_val - data(:, i)) / (max_val - min_val);
            normalized_ref(:, i) = (max_val - references(:, i)) / (max_val - min_val);
        end
    end
else
    % Normalizacja dla danych fuzzy (3D)
    normalized = zeros(size(data));
    normalized_ref = zeros(size(references));
    for i = 1:num_criteria
        for j = 1:3 % Dla każdej składowej [dolna, modalna, górna]
            if types(i) == -1 % Minimalizacja
                min_val = min(data(:, i, j));
                max_val = max(data(:, i, j));
                normalized(:, i, j) = (data(:, i, j) - min_val) / (max_val - min_val);
                normalized_ref(:, i, j) = (references(:, i, j) - min_val) / (max_val - min_val);
            elseif types(i) == 1 % Maksymalizacja
                min_val = min(data(:, i, j));
                max_val = max(data(:, i, j));
                normalized(:, i, j) = (max_val - data(:, i, j)) / (max_val - min_val);
                normalized_ref(:, i, j) = (max_val - references(:, i, j)) / (max_val - min_val);
            end
        end
    end
end
end

function distances = Czebyszew_Fuzzy(normalized, normalized_ref)
% Obliczanie odległości Czebyszewa dla wartości rozmytych
num_alternatives = size(normalized, 1);
num_references = size(normalized_ref, 1);
distances = zeros(num_alternatives, num_references);

for i = 1:num_alternatives
    for j = 1:num_references
        % Obliczanie odległości Czebyszewa dla każdej z trzech składowych
        chebyshev_distances = abs(normalized(i, :, :) - normalized_ref(j, :, :));
        chebyshev_distances = max(chebyshev_distances, [], 3); % Maksymalna odległość dla każdej składowej
        distances(i, j) = max(chebyshev_distances); % Maksymalna wartość odległości dla całej alternatywy
    end
end
end

```

Rys 8. Algorytm MREF

5. Porównanie metod optymalizacji

Platformy IIOT zostały poddane niezależnej ocenie przez członków projektu, szczególną uwagę poświęcono zastosowaniu przez przyszłych użytkowników autonomicznych robotów szklarniowych. W ten sposób powstały 3 niezależne zbiory danych wejściowych

Porównanie metod nie jest jednak zadaniem prostym. Każda z analizowanych technik, takich jak TOPSIS GT, RSM czy MREF, wprowadza odmienne podejście do klasyfikacji alternatyw, uwzględniając specyfikę danych i preferencji użytkowników. W kontekście agrokultury ich wykorzystanie umożliwia ocenę systemów zarówno pod kątem ich efektywności operacyjnej, jak i potencjału integracyjnego z autonomicznymi systemami w zakresie robotyki mobilnej czy zdolności do adaptacji w specyficznych warunkach środowiskowych. Jednocześnie metody te różnią się między sobą w zakresie podejścia do niepewności, modelowania preferencji decydentów oraz sposobu obliczania odległości od punktów odniesienia.

W celu lepszego zrozumienia potencjału każdej metody, kluczowe jest porównanie ich wyników na podstawie wykresów, które ilustrują różnice w rankingu poszczególnych platform technologicznych. Analiza graficzna umożliwia identyfikację rozbieżności w podejmowanych decyzjach oraz wskazuje, które z narzędzi mogą najlepiej odpowiadać na potrzeby użytkowników w zautomatyzowanym rolnictwie. Dzięki temu możliwe staje się nie tylko zrozumienie zalet i ograniczeń poszczególnych metod, ale również wyciągnięcie wniosków dotyczących ich praktycznego zastosowania w procesach decyzyjnych związanych z nowoczesnym zarządzaniem środowiskami szklarniowymi.

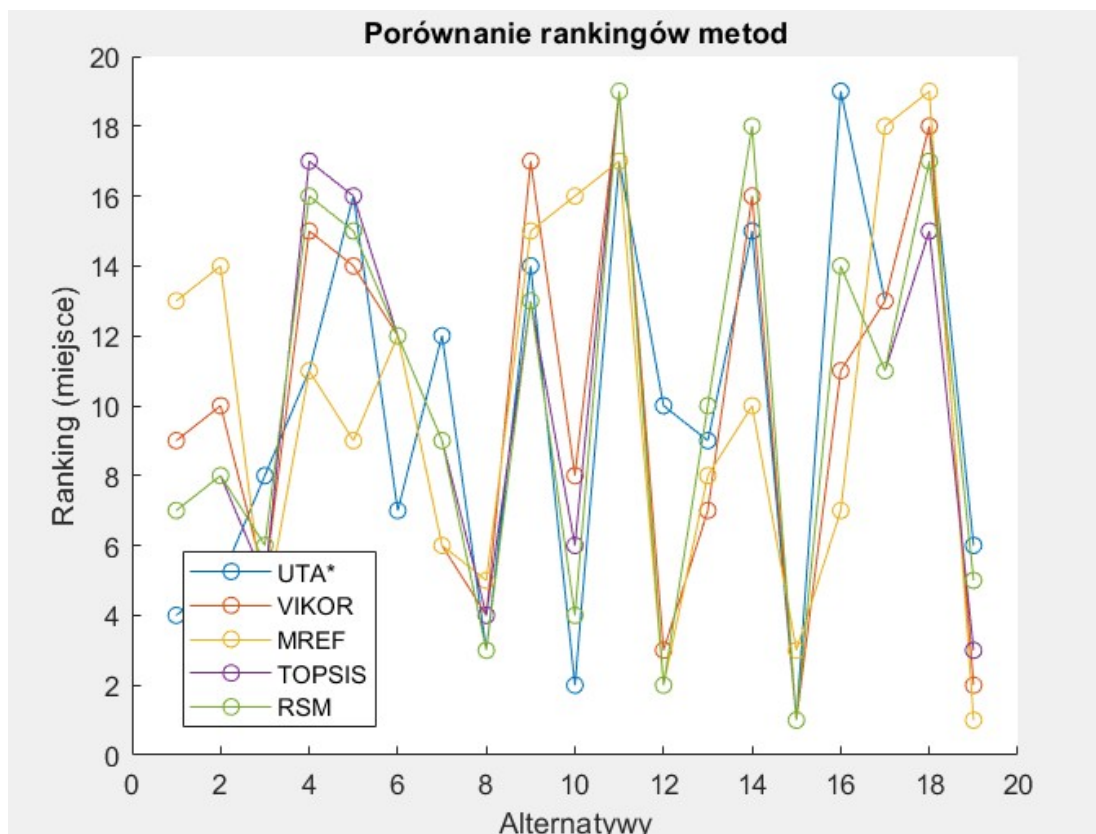
Jak wynika z powyższych tabel na szczególną uwagę zasługuje platforma Davra, która we wszystkich rankigach plasuje się na lokatach topowych.

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

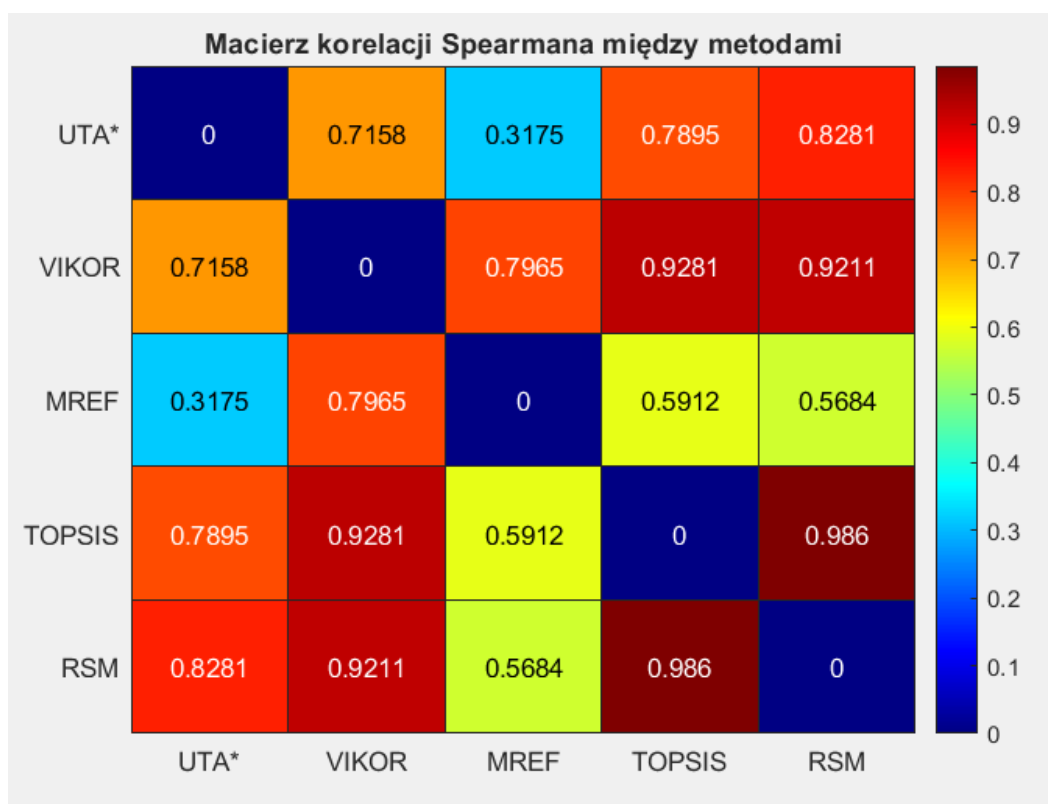
Dodatkowo do badania korelacji między danymi zaimplementowano macierz współczynnika Tau-Kendalla oraz macierz korelacji Spearmana.

Pliki w projekcie z racji na budowę funkcji i organizację nazwano numerami. Liczby odpowiadające nazwiskom są zawarte w dalszej części pracy.

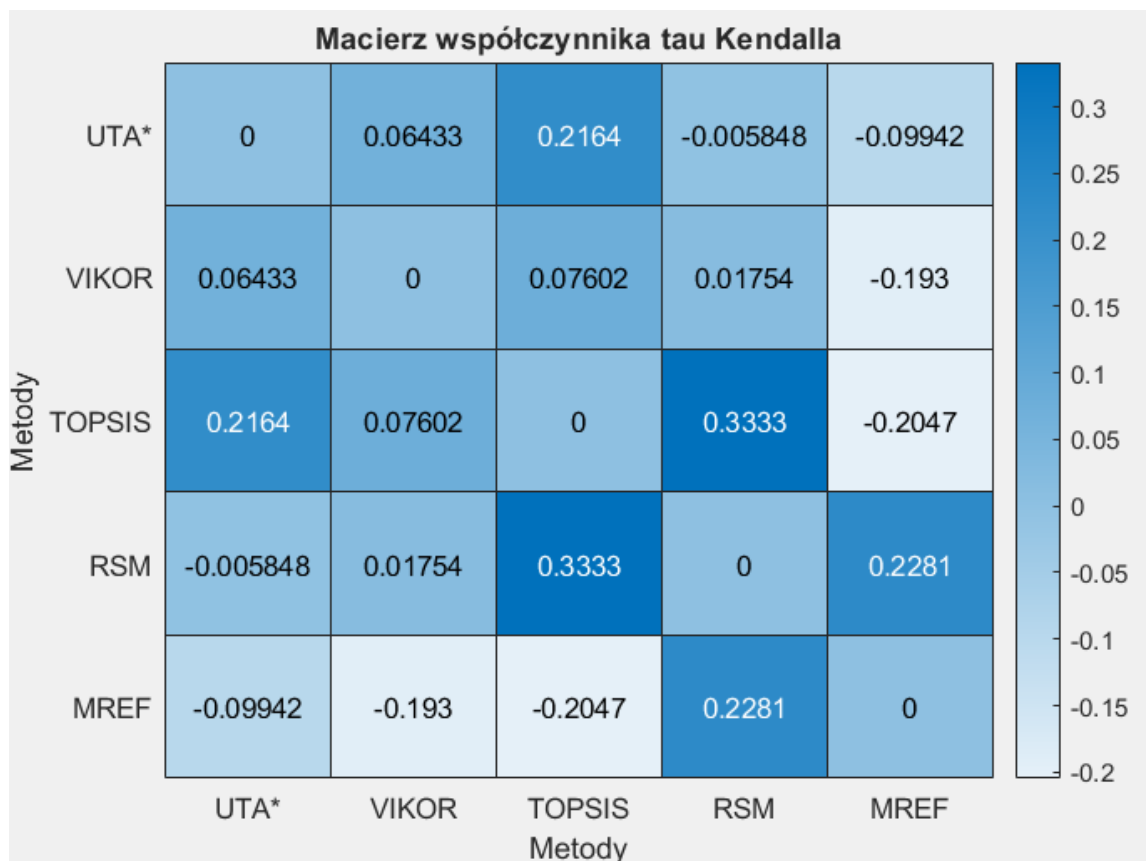
Dane z pliku ekspert_1.xlsx:



Rys 10. Rankingi metod dla 1 zbioru danych

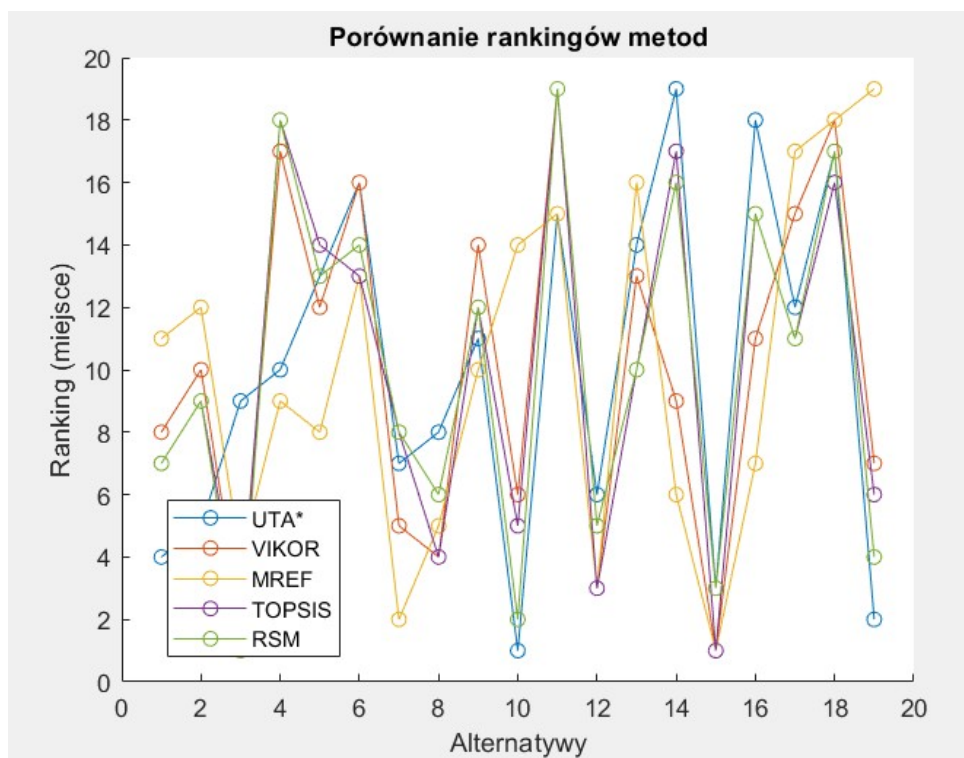


Rys 11. Korelacja Spearmana

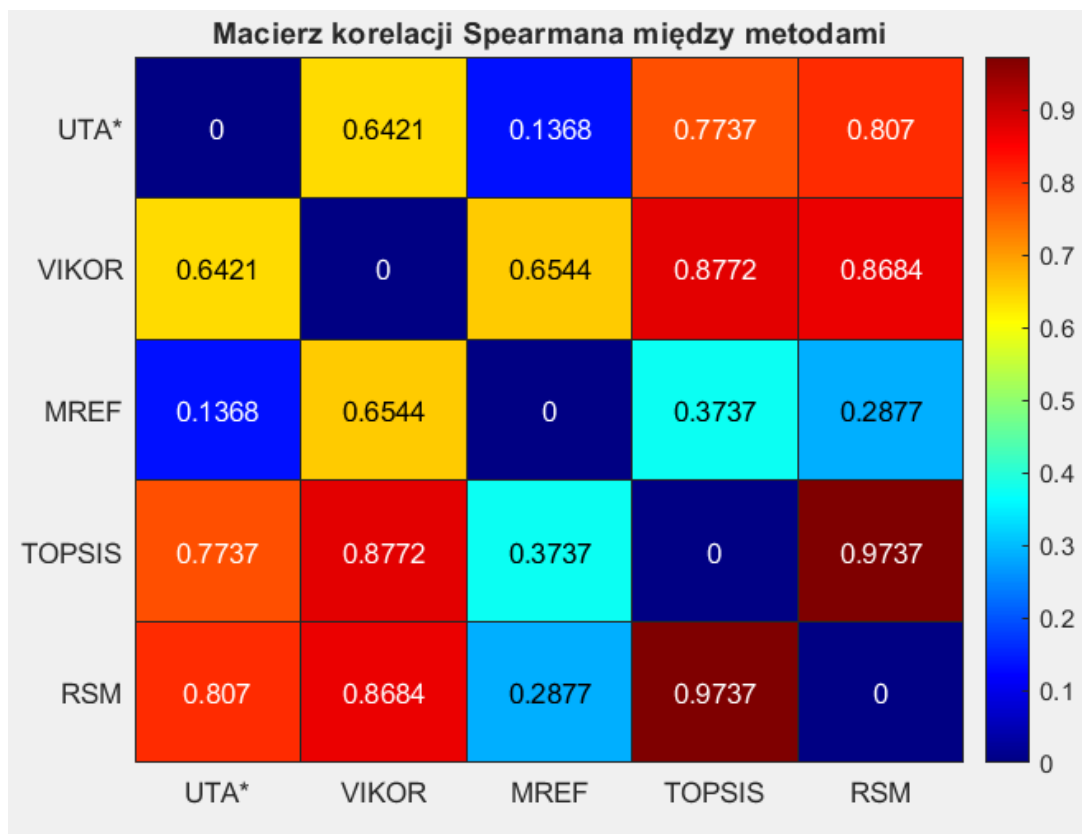


Rys 12. Porównanie Tau-Kendalla

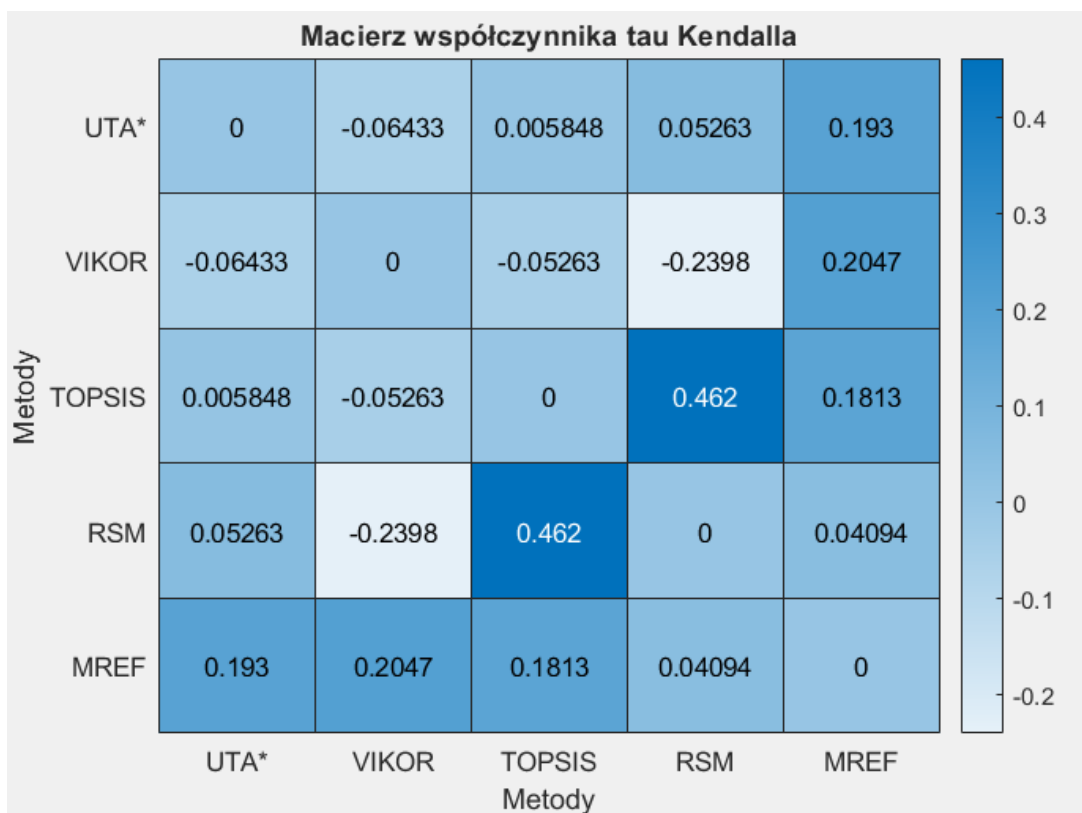
Dane z pliku ekspert_Jerzyna.xlsx (ekspert 2):



Rys 14. Rankingi metod dla 2 zbioru danych

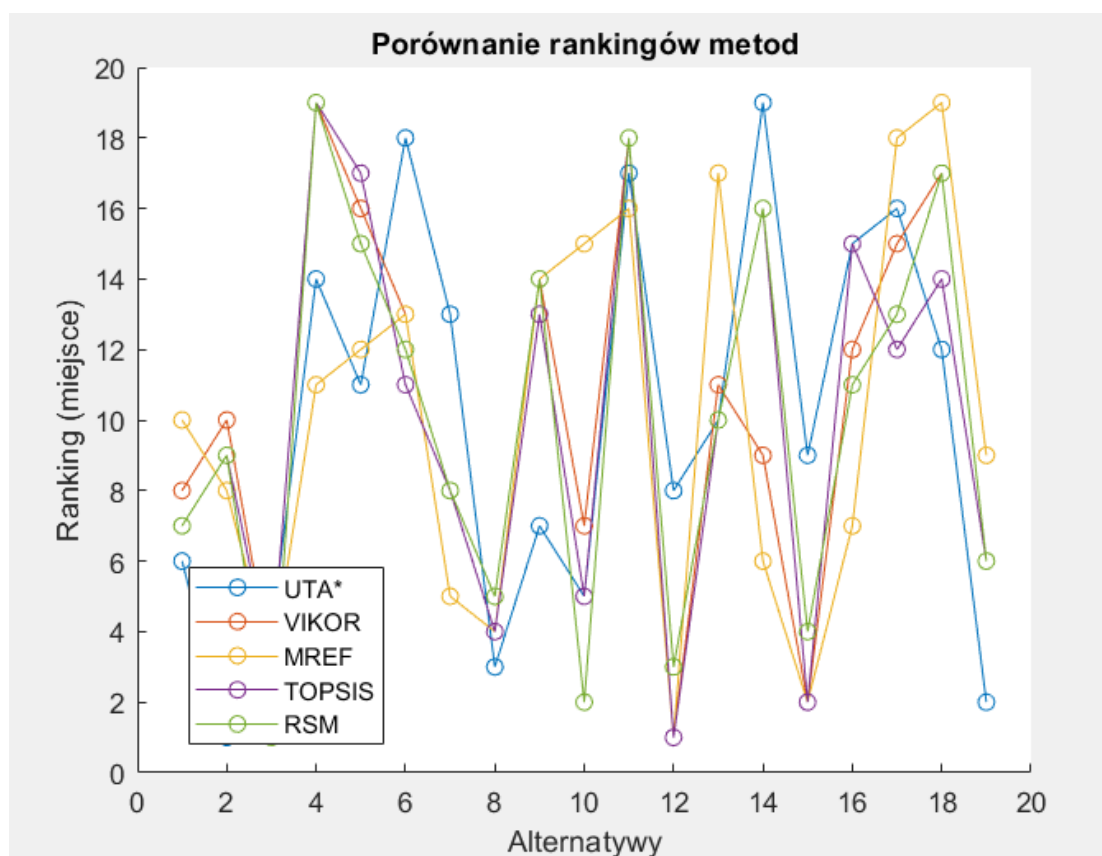


Rys 15. Korelacja Spearmana

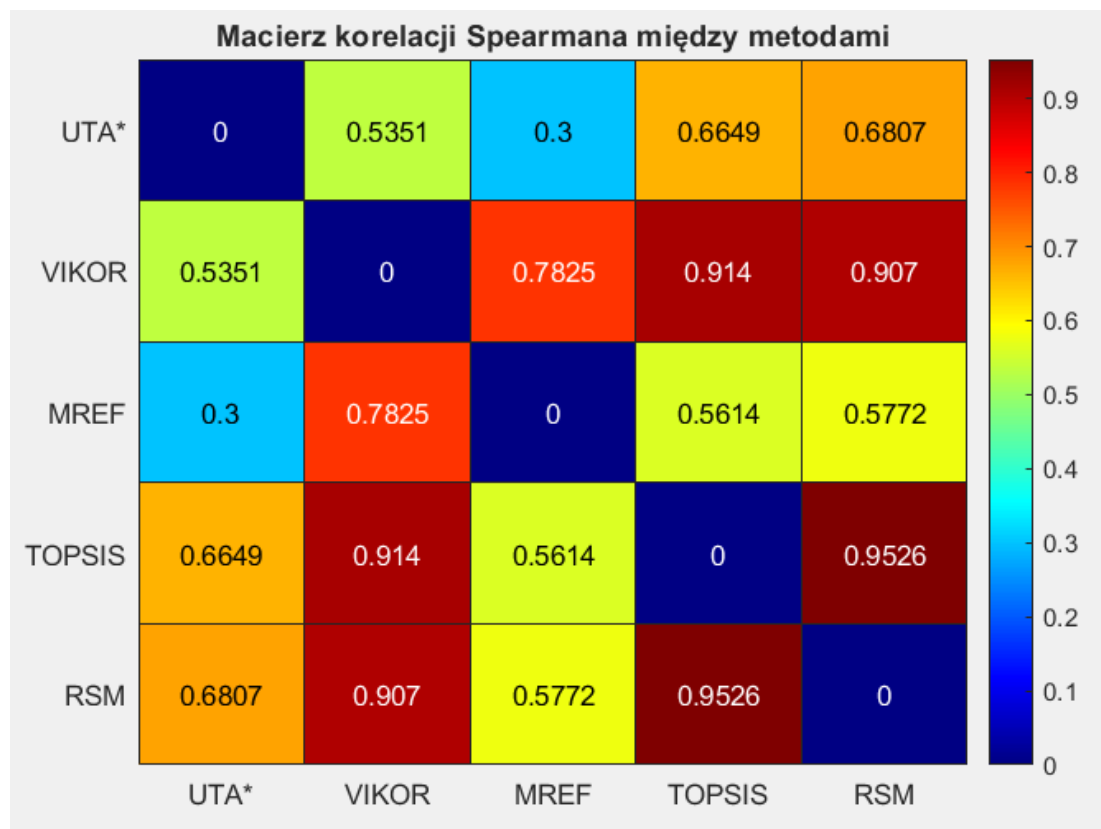


Rys 16. Porównanie Tau-Kendalla

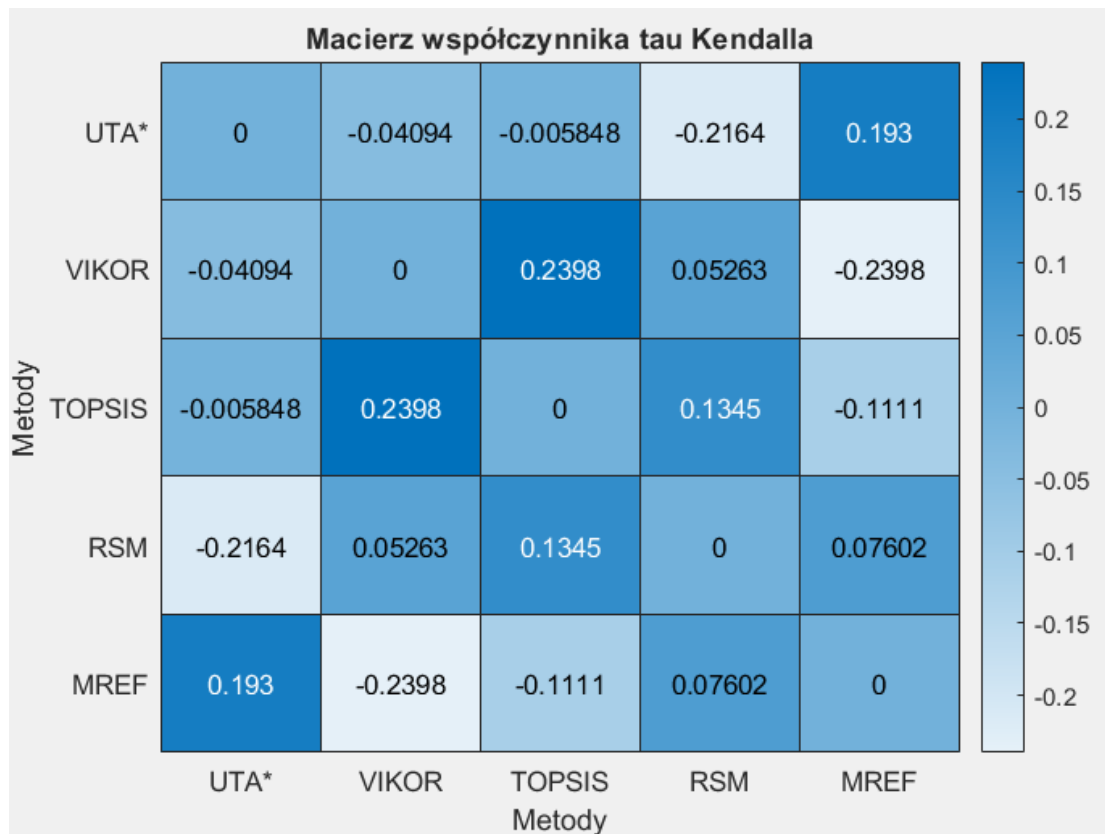
Dane z pliku ekspert_Janik.xlsx (ekspert 3).:



Rys 18. Rankingi metod dla 3 zbioru danych

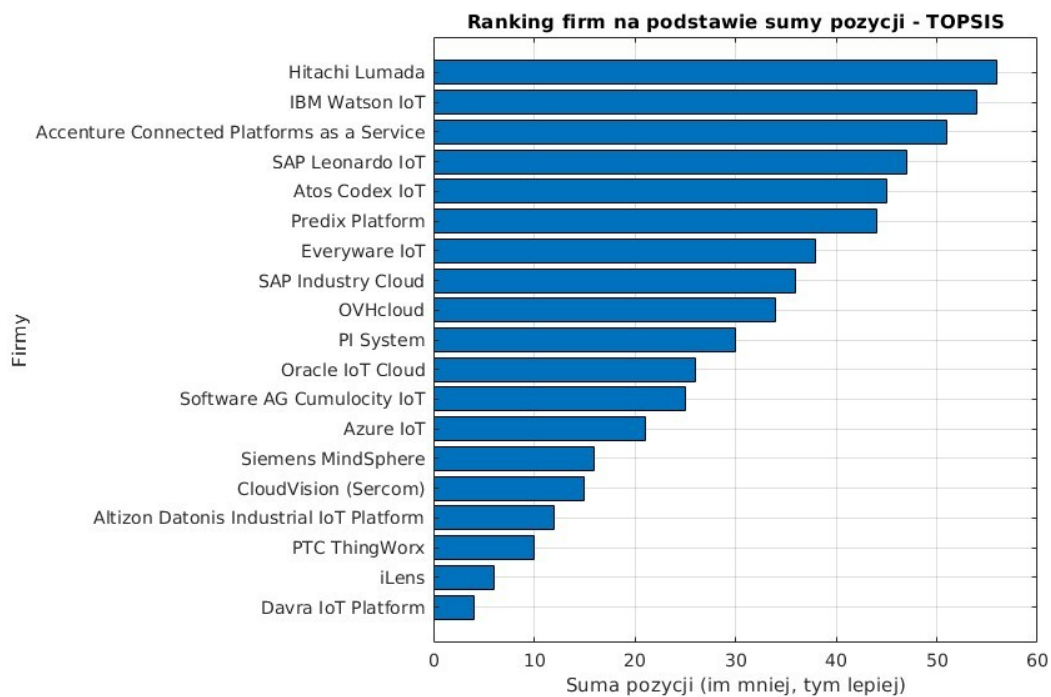


Rys 19. Korelacja Spearmana

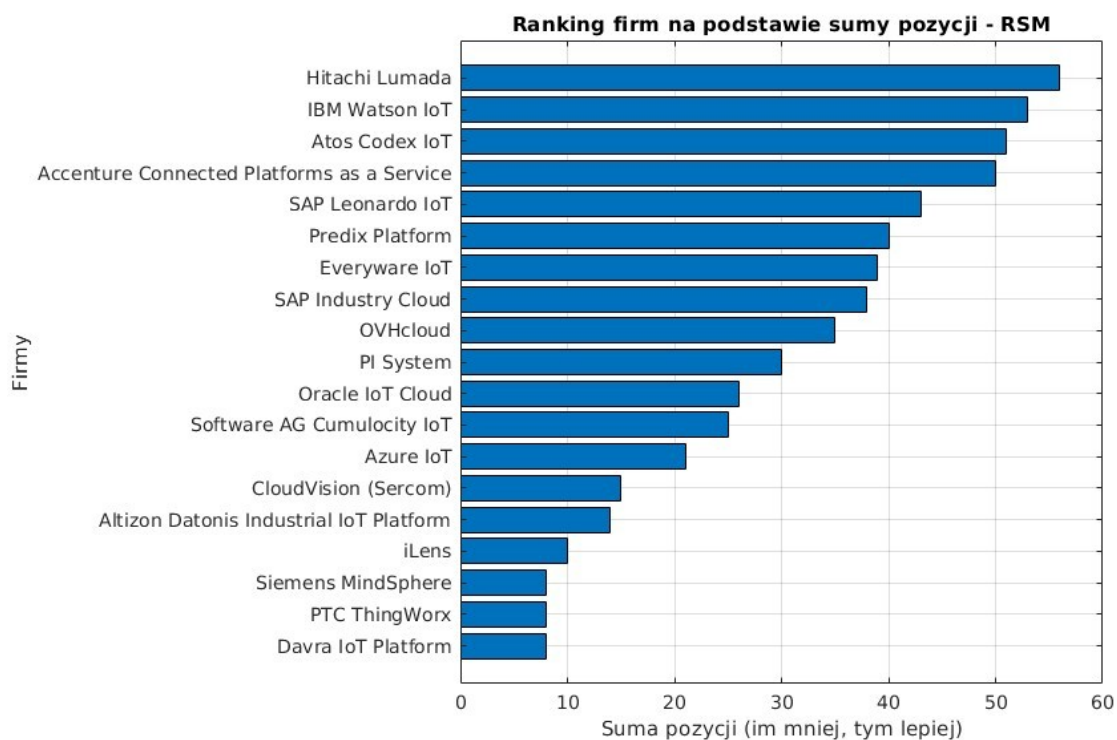


Rys 20. Porównanie Tau-Kendalla

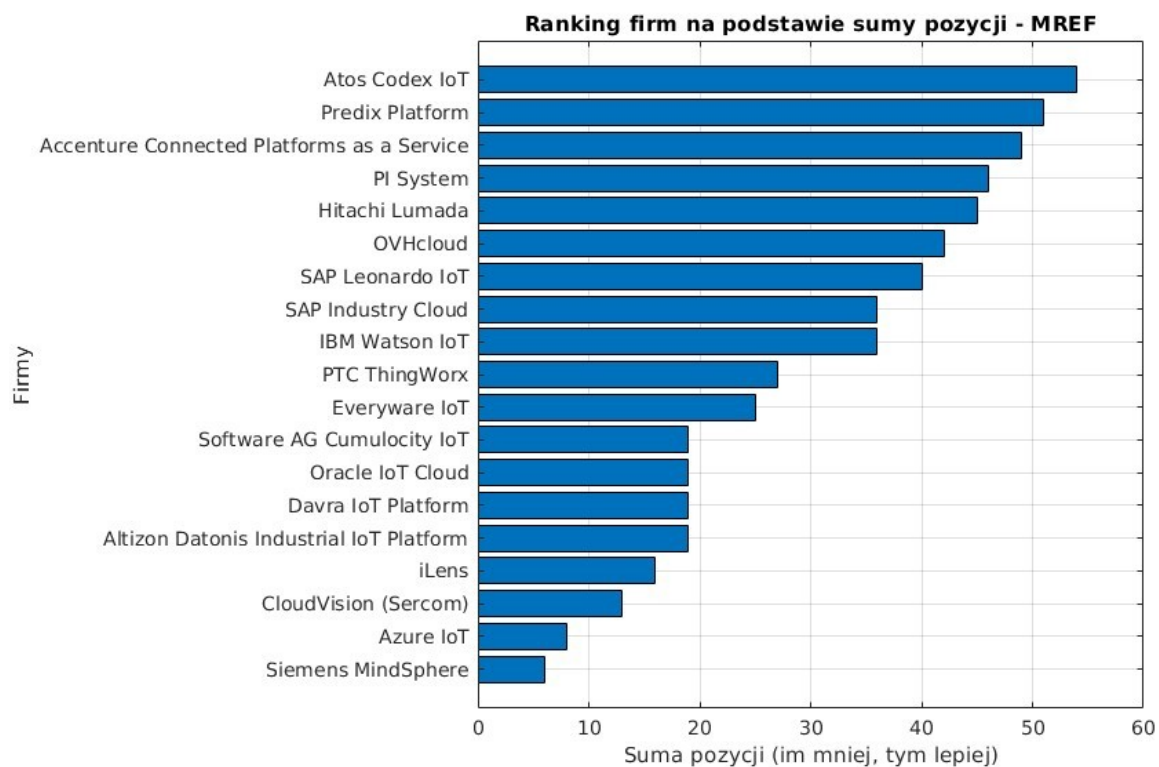
Po przeprowadzeniu wstępnej analizy przystąpiono do właściwego procesu analitycznego porównywania metod sumując miejsca jakie dane platformy zajęły za pomocą odpowiedniego algorytmu. Posłużyła do tego funkcja `sumuj_pozioamy_rankingow`, która została zaprojektowana do agregacji pozycji firm z wielu rankingów, co umożliwia uzyskanie zbiorczej oceny każdej platformy na podstawie różnych zestawień. Jej podstawowym celem jest obliczenie sumy miejsc zajmowanych przez każdą firmę w przekazanych do funkcji rankingach, co pozwala na szybkie porównanie wyników z różnych podejść lub opinii ekspertów. Argumenty wejściowe przekazywane do funkcji to struktury danych zawierające rankingi w postaci tablic komórkowych, w których pierwszy element w każdym wierszu oznacza nazwę firmy, a pozycja w tabeli odpowiada miejscu w rankingu. Wyniki algorytmów zostały zaprezentowane poniżej w formie rankingów, w których suma miejsc przyznanych danej platformie przez trzech ekspertów pozwalała na określenie jej pozycji końcowej w danej metodzie. Im mniejsza suma pozycji, tym lepsze miejsce zajmowała platforma w końcowym rankingu.



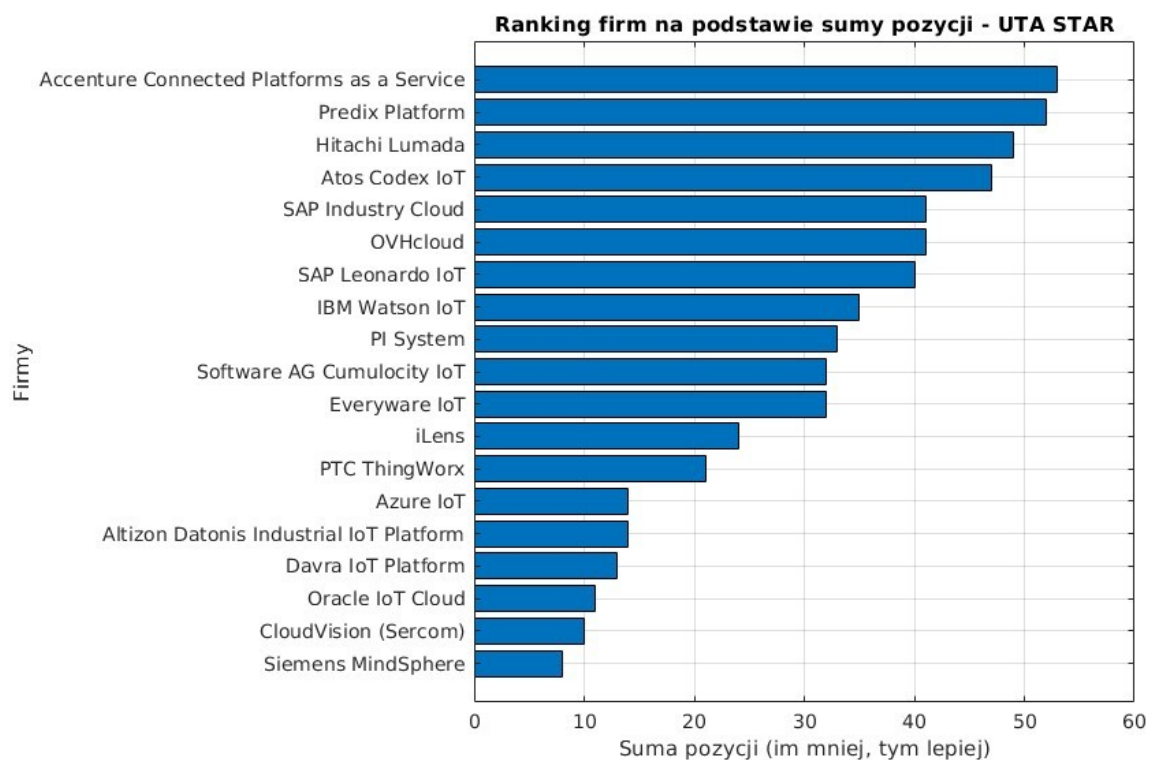
Rys 21. Ranking metody TOPSIS GT



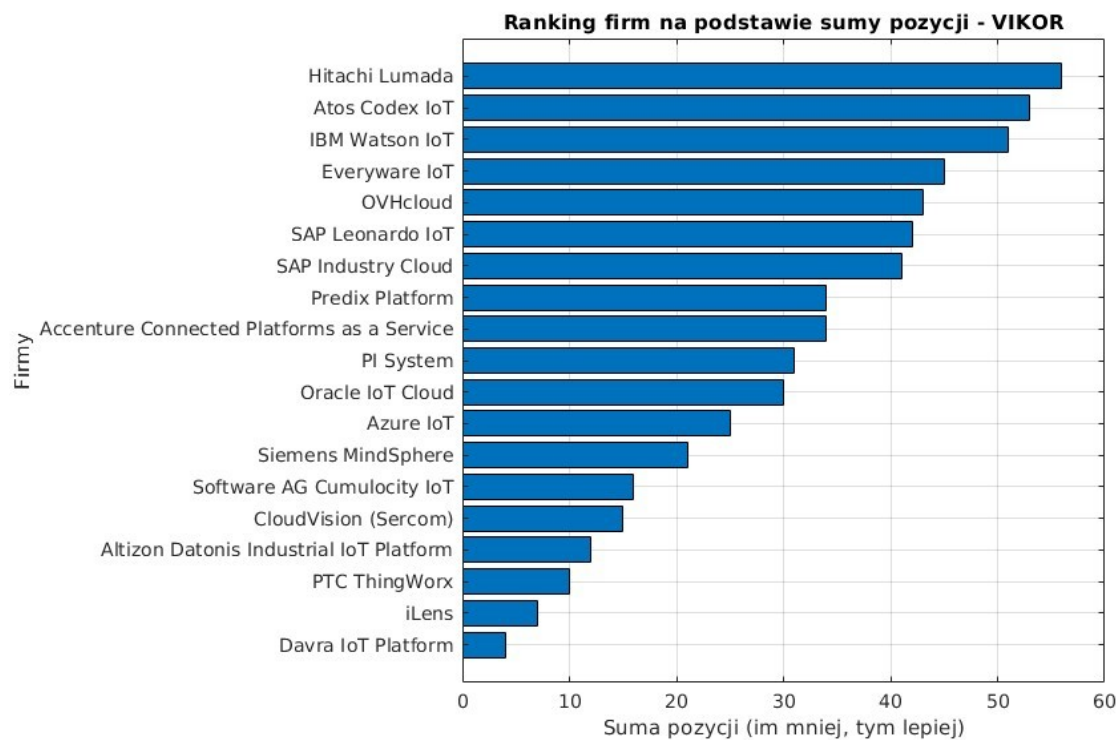
Rys 22. Ranking metody RSM



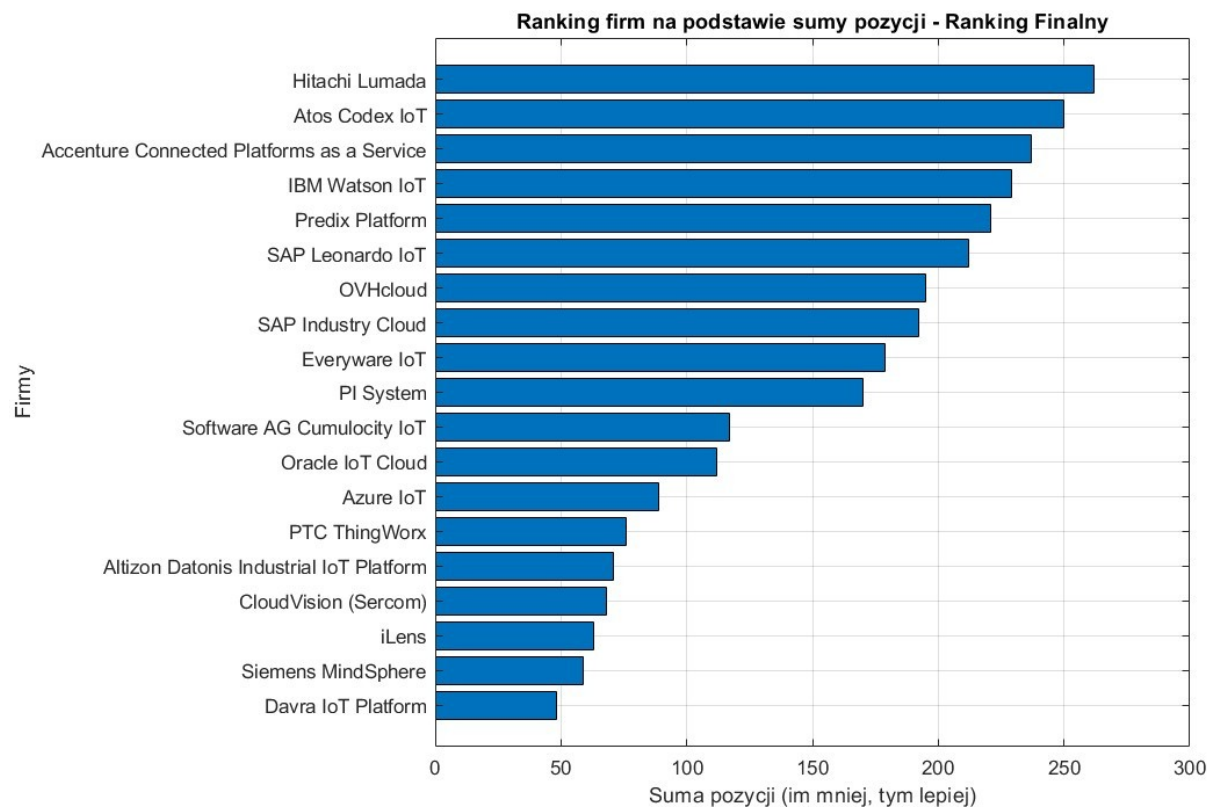
Rys 23 Ranking metody MREF



Rys 24. Ranking metody UTA STAR



Rys 25. Ranking metody VIKOR



Rys 26. Sumaryczny ranking wszystkich metod

Poniżej dokonano szczegółowej interpretacji uzyskanych danych:

Wykresy pokazują, że wg każdej z metod. Platformy takie jak Davra IoT Platform, Siemens MindSphere oraz konsekwentnie zajmują czołowe pozycje. Wynika to z ich zaawansowanej integracji z autonomicznymi systemami oraz wsparcia dla złożonych analiz predykcyjnych.

Ostateczna analiza pozycji platform względem opinii ekspertów pozwala na sformułowanie kilku istotnych wniosków:

- Davra IoT Platform oraz CloudVision (Sercom) stanowią najbardziej rekomendowane rozwiązania dla szklarni zautomatyzowanych, zarówno pod względem integracji z robotami, jak i efektywności zarządzania środowiskiem uprawowym.
- Siemens MindSphere oraz Azure IoT wyróżniają się wszechstronnością i będą dobrym wyborem dla gospodarstw nastawionych na rozwój i integrację z innymi systemami.
- Platformy ogólnego zastosowania, takie jak Atos Codex IoT, Predix Platform czy Accenture Connected Platforms as a Service, uzyskały niższe oceny w kontekście rolnictwa szklarniowego, co wynika z braku wyspecjalizowanych modułów dedykowanych tej branży.

Analiza wyników wykazała, że platformy dedykowane rolnictwu (np. CloudVision, Agro, Demeter) oraz te ściśle przemysłowe (np. Siemens MindSphere, Davra IoT Platform) są najbardziej odpowiednie dla gospodarstw szklarniowych. Ich funkcjonalność, taka jak integracja z systemami robotów mobilnych, monitorowanie środowiska oraz analiza predykcyjna, pozwala na optymalizację kosztów i poprawę plonów. Platformy uniwersalne, takie jak OVHcloud czy Everyware IoT, słabiej wypadają w kontekście rolnictwa szklarniowego, ze względu na brak funkcji dedykowanych zarządzaniu uprawami.

Warto zwrócić uwagę na to, że każda z zastosowanych metod charakteryzowała się nieco innym podejściem do oceny i agregacji danych, co w niektórych przypadkach prowadziło do różnic w uzyskanych rankingach. Na przykład:

- MREF oraz TOPSIS kładły większy nacisk na bliskość do rozwiązania idealnego, co sprzyjało platformom uniwersalnym i stabilnym.
- RSM oraz UTA STAR w większym stopniu podkreślały indywidualne preferencje ekspertów, przez co platformy mniej dedykowane rolnictwu (np. Atos Codex IoT, Predix Platform) mogły zyskać nieco wyższe lokaty.

- VIKOR z kolei uwzględniał równowagę między kryteriami, co w pewnych przypadkach spowodowało wypromowanie platform o średnich, ale stabilnych wynikach we wszystkich aspektach.

6. Podsumowanie i wnioski

Projekt umożliwił szczegółowe zbadanie potencjału platform cloudowych, które wspierają systemy IIoT i AIIoT w nowoczesnym, zautomatyzowanym rolnictwie szklarniowym. Przeprowadzone analizy oraz zastosowanie metod wielokryterialnej optymalizacji wykazały, że platformy dedykowane rolnictwu, takie jak CloudVision i Agro, oferują kluczowe funkcjonalności wspierające efektywne zarządzanie zasobami szklarniowymi. Ich możliwości, takie jak monitorowanie środowiska, modelowanie procesów uprawowych czy integracja z robotami mobilnymi, szczególnie dobrze wpisują się w potrzeby sektora szklarniowego.

Zastosowanie zaawansowanych algorytmów, takich jak TOPSIS GT, RSM i MREF, pozwoliło na precyzyjne określenie, które rozwiązania technologiczne najlepiej odpowiadają zidentyfikowanym kryteriom i scenariuszom użytkowania. Te metody podkreśliły znaczenie wielokryterialnej analizy decyzji jako narzędzia wspierającego wybór technologii w dynamicznie zmieniających się warunkach sektora rolnego.

Platforma CloudVision wyróżnia się na tle innych zaawansowanymi funkcjami analitycznymi i możliwością integracji z systemami autonomicznymi, co czyni ją szczególnie użyteczną w środowiskach szklarniowych. Z kolei Davra IoT Platform, pomimo bardziej uniwersalnego charakteru, wykazuje wyjątkową wszechstronność i niezawodność, co pozwala na jej adaptację zarówno w rolnictwie, jak i w innych sektorach przemysłowych. Siemens MindSphere i Azure IoT również zajmują czołowe pozycje dzięki swojej elastyczności, skalowalności oraz zaawansowanym możliwościom analizy danych, które są szczególnie cenione w bardziej kompleksowych operacjach. Platformy dodane do projektu początkowego nie odegrały istotnej roli w analizie porównawczej.

Podsumowując, projekt wykazał, że wdrożenie odpowiednio dobranych platform technologicznych ma potencjał, by zrewolucjonizować zarządzanie produkcją szklarniową. Technologie te mogą znacząco zwiększyć efektywność operacyjną, obniżyć koszty produkcji i jednocześnie ograniczyć negatywny wpływ na środowisko, przyczyniając się tym samym do rozwoju nowoczesnego, zrównoważonego rolnictwa. Projekt pozwolił na szczegółową analizę platform cloudowych wspierających systemy IIoT i AIIoT w kontekście zautomatyzowanego rolnictwa szklarniowego.

Bibliografia:

- [1] <https://www.gartner.com/reviews/market/global-industrial-iiot-platforms>
- [2] <https://webbylab.com/blog/best-iiot-cloud-platforms>
- [3] <https://iiot-analytics.com/our-coverage/iiot-platforms-software>
- [4] <https://www.31west.net/blog/iiot-platforms-comparison-aws-azure-google-ibm-cisco>
- [5] <https://euristiq.com/best-iiot-cloud-platforms>
- [6] <https://iiot-analytics.com/iiot-cloud>
- [7] <https://www.intuz.com/blog/top-iiot-development-platforms-and-tools>
- [8] A.M.J. Skulimowski, Selection of cloud software for robotized smart agriculture based on uncertainty analysis, fuzzy triangular number ranking, and multicriteria decision making methods
- [9] A.M.J. Skulimowski
https://www.researchgate.net/publication/221274898_Freedom_of_Choice_and_Creativity_in_Multicriteria_Decision_Making
- [10] A.M.J. Skulimowski
https://www.researchgate.net/publication/283495003_The_Art_of_Anticipatory_Decision_Making
- [11] <https://www.arista.com/en/cg-cv/cv-accessing-help-center-documentation>
- [12] <https://www.arista.com/en/cg-cv/cv-introduction-to-cloudvision>

Linki do dodatkowych platform IIoT:

- Agro

- [13] <https://risdaagro.com.my/>

- Demeter

- [14] <https://h2020-demeter.eu/tag/iiot/>

Jakub Janik	Funkcja stworz fuzzy macierz, RSM, sprawozdanie, TOPSIS
Paweł Jerzyna	GUI, MREF, prezentacja, modyfikacje algorytmów pod GUI

Ustalenie podziału zadań było zadaniem trudnym, ponieważ znakomita większość pracy była wykonywana wspólnie w jednym czasie.

