ISWD - Lab 2

PROMETHEE

Rozważmy następujący problem decyzyjny. Chcemy stworzyć ranking 8 wariantów, które są ocenione na trzech kryteriach: trzy z nich są typu zysk (g_1, g_2, g_3) i jedno typu koszt (g_4).

Tabela 1: Zbiór wariantów oceniony na czterech kryteriach.

	g_1	g_2	g_3	g_4
a_1	5	4	70	12
a_2	4	1	54	11
a_3	5	0	71	17
a_4	5	2	60	15
a_5	2	0	81	11
a_6	8	5	55	13
a_7	3	2	97	19
a_8	2	5	73	14

Dla ludzi jest dość trudno ocenić jednocześnie różne aspekty wielu opcji czy wariantów. Jednakże porównanie pojedynczych par wariantów jest o wiele łatwiejsza i mniej problematyczne. Ponadto łatwiej jest podać zalety i wady każdego z wariantów, porównując je względem jakiejś naszej wewnętrznej skali istotności i ocenić, który wariant jest bardziej preferowany.

Jak już było wspomniane na poprzednich zajęciach, jednym z typów relacji przy porównywaniu dwóch wariantów jest dominacja. Jednakże jej obecność jest dość rzadka. To znaczy, że ostateczny ranking nie wynika jedynie z ocen wariantów, a zależy głównie od informacji preferencyjnej otrzymanej od decydenta i każde rozwiązanie Pareto optymalne może być na szczycie rankingu.

Zadanie 1
Sprawdź czy istnieje relacja dominacji pomiędzy jakąkolwiek parą wariantów.

Relacja dominacji nie bierze pod uwagę pewnych elementów rozważanego problemu. Po pierwsze, niektóre różnice pomiędzy wariantami są nieistotne z punktów widzenia decydenta, tj. para wariantów jest nierozróżnialna dla decydenta pomimo różnic na poszczególnych kryteriach. Kolejną sprawą jest kwestia kompensacji słabych ocen na niektórych kryteriach silnymi ocenami na innych.

Podstawowe założenia

- Porównując parę wariantów, powinno się wziąć pod uwagę różnicę w ich ocenach, a nie absolutne wartości;
- Wyeliminowany efekt skalowania - w metodzie występują wagi, ale nie mnożymy ich bezpośrednio przez oceny, ale wyniki porównania ocen par wariantów na poszczególnych kryteriach;
- Wyjaśnienie zaproponowanej rekomendacji - wyniki są proste, zrozumiałe; PROMETHEE jest jedną z bardziej przyjaznych do zrozumienia metod, także dla ludzi bez zaawansowanego wykształcenia informatycznego, matematycznego i ekonomicznego - stąd ogromna jej popularność;
- Parametry techniczne powinny mieć praktyczne (ekonomiczne) uzasadnienie - odwołujemy się do wag kryteriów oraz parametrów odnoszących się do różnic w ocenach wariantów na poszczególnych kryteriach.

Informacja preferencyjna

Informacja preferencyjna w podstawowej wersji metod PROMETHEE ma charakter bezpośredni, tj. odwołuje się do wartości parametrów wykorzystywanych przez metodę. Oprócz tego należy wybrać typ funkcji, która wykorzystywana jest do określenie stopnia preferencji dla wszystkich par wariantów na każdym kryterium.

- Wagi kryteriów w_j - współczynniki ważności poszczególnych kryteriów; im wyższa waga, tym bardziej istotne dane kryterium i tym większa jego moc w przeforsowaniu decyzji, którą popiera. Wag ma interpretację współczynnika istotności.
- Wybór kształtu funkcji preferencji dla każdego kryterium (spośród 6 predefniniowanych), zgodnie z którą warianty będą porównywane; funkcja tłumaczy różnice w ocenach na stopień (intensywność) preferencji. Dopuszczalne jest definiowanie innych kształtów funkcji, ale w oryginalnej propozycji typu kształtów funkcji preferencji, z których się wybiera, są ściśle określone.
- Parametry opisujące funkcje preferencji:
  - Próg nierozróżnialności q_j - maksymalna różnica, przy której dwa warianty są nierozrożnialne i nie ma wtedy preferencji; równoważnie = minimalna różnica, powyżej której występuje choć częściowa preferencja;
  - Próg preferencji p_j - minimalna różnica, przy której jeden wariant jest silnie preferowany nad drugi; poniżej tej różnicy stopień preferencji jest mniejszy niż 1;

Schemat działania metod PROMETHEE

Metody PROMETHEE składają się z czterech następujących kroków:

- Krok I: Wykorzystanie funkcji preferencji na poszczególnych kryteriach do porównania par wariantów względem tych funkcji.
- Krok II: Obliczenie całkowitych stopni preferencji wypełnienie macierzy preferencji.
- Krok III: Obliczenie przepływów (dodatnich, ujemnych i całkowitych).
- Krok IV: Wypracowanie rankingu (zgodnie z założeniami metody PROMETHEE II lub PROMETHEE I) na podstawie przepływów

Obliczanie stopnia intensywności preferencji

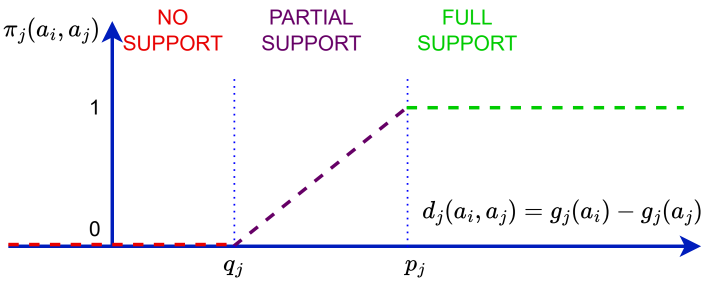
Będziemy porównywać oceny uzyskane przez parę wariantów a_i oraz a_k na każdym z kryteriów j jako: $d_j(a_i, a_k) = g_j(a_i) - g_j(a_k)$. Dla takiej różnicy możemy łatwo stwierdzić czy jest ona na wybranym kryterium pomijalnie mała i może być zignorowana, czy przeciwnie jest wystarczająco duża, tak że istnieje preferencja.

Wyrazimy naszą preferencję na tym kryterium w skali [0,1] jako cząstkowy stopnień intensywności preferencji:

$$\pi_j(a_i, a_k) = F_j(d_j(a_i, a_k)) \tag{1}$$

gdzie F_j jest funkcją preferencji decydenta. Ta funkcja może przyjmować różnorakie kształty, najbardziej popularny pokazany jest na Rysunku 1. Uwzględnia on następujące obszary:

- wariant a_k jest lepszy niż wariant a_i na kryterium j ($d_j(a_i, a_k) < 0$) - oznacza to, że nie ma żadnych przesłanek, aby wariant a_i był na tym kryterium bardziej preferowany nad a_k;
- wariant a_k jest nieznacznie gorszy niż a_i ($d_j(a_i, a_k) < q_j$)- tutaj pomimo tego, że wariant a_i ma lepszą ocenę niż a_k, to ta różnica jest zbyt mała, abyśmy mogli uznawać ją za wystarczającą do zaistnienia preferencji;
- wariant a_i jest znacząco lepszy niż wariant a_k o co najmniej próg preferencji p_j ($d_j(a_i, a_k) > p_j$) - w tym obszarze możemy w stwierdzić, iż istnieje pełna preferencja a_i nad a_k na kryterium j



Rysunek 1: Stopień intensywności preferencji dla kryterium typu zysk.

- wariant a_i jest lepszy niż wariant a_k bardziej niż próg nierozróżnialności q_j, ale mniej niż próg preferencji p_j - oznacza to, iż istnieje tylko częściowa preferencja a_i nad a_k

Globalna preferencja wariantu jest wynikiem sumy cząstkowej preferencji na wszystkich kryteriach. Decydent może podjąć decyzję, że nie wszystkie kryteria są równie istotne i ustalić wektor wag określający istotność kryteriów. Częściowa preferencja może być agregowana do **stopnia preferencji**, który może być interpretowany jako poziom intensywności relacji przewyższania lub jako moc koalicji kryteriów wspierających preferencję jednego wariantu nad drugim.:

$$\pi(a_i, a_k) = \frac{\sum_j w_j \pi_j(a_i, a_k)}{\sum_j w_j} \tag{2}$$

Tabela 2: Wartości parametrów

	min value	max value	criterion type	q_j	p_j	w_j
g_1	0	10	gain	0	5	0.09
g_2	0	5	gain	1	1	0.29
g_3	30	100	gain	6	23	0.25
g_4	10	20	cost	2	4	0.37

Zadanie 2
Oblicz stopnie preferencji dla $\pi(a_1, a_2)$ i $\pi(a_1, a_3)$.

Tabela 3: Stopnie preferencji $\pi(a_i, a_j)$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a_1	0.00			0.53	0.34	0.13	0.70	0.05
a_2	0.00	0.00	0.37	0.37	0.04	0.00	0.39	0.22
a_3	0.00	0.18	0.00	0.07	0.05	0.15	0.04	0.05
a_4	0.00	0.02	0.29	0.00	0.34	0.00	0.41	0.05
a_5	0.07	0.25	0.43	0.59	0.00	0.25	0.37	0.21
a_6	0.05	0.36	0.71	0.34	0.38	0.00	0.75	0.09
a_7	0.25	0.25	0.54	0.25	0.46	0.25	0.00	0.27
a_8	0.00	0.48	0.48	0.39	0.29	0.18	0.66	0.00

Obliczenie przepływów

Po obliczeniu stopni preferencji dla wszystkich par wariantów otrzymujemy macierz stopni preferencji. Istnieje kilka różnych metod eksploatujących tę macierz w celu wyznaczenia ostatecznego rankingu wariantów, my skorzystamy z metody **Net Flow Score**, która bazuje na obliczaniu następujących przepływów:

1. **przepływ dodatni** = siła = jak bardzo wariant jest preferowany nad wszystkie pozostałe warianty; dla wariantu a obliczany jako sumaryczny stopień preferencji wariantu a nad innymi (czasem jako

średni stopni preferencji wariantu a_i nad innymi):
$\phi^+(a_i) = \sum_{a_j} \pi(a_i, a_j)$

2. **przepływ ujemny** = słabość = jak bardzo wszystkie pozostałe warianty są preferowane nad dany wariant; dla wariantu a obliczany jako sumaryczny stopień preferencji innych wariantów nad wariantem a (czasem jako suma średni preferencji innych wariantów nad wariantem a_i):
$\phi^-(a_i) = \sum_{a_j} \pi(a_j, a_i)$.

Zadanie 3
Oblicz dodatni, ujemny i całkowity przepływ dla wariantu a_1.

Tabela 4: Wartości przepływów

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
ϕ^+		1.39	0.54	1.11	2.17	2.68	2.27	2.48
ϕ^-		2.	3.48	2.54	1.9	0.96	3.32	0.94
ϕ		-0.61	-2.94	-1.43	0.27	1.72	-1.05	1.54

PROMETHEE I

Głównym celem metody PROMETHEE I jest stworzenie częściowego rankingu wariantów poprzez wzięcie pod uwagę przepływów dodatnich i ujemnych. Biorą pod uwagę porównania parami, w parze wariantów istnieje preferencja, jeżeli oba: dodatni i ujemny przepływ są zgodne co do tej relacji. Nierozróżnialność zachodzi wówczas, gdy oba przepływy wariantów są identyczne. Jeżeli jednak nie ma spójności w przepływach, wówczas pomiędzy wariantami zachodzi nieporównywalność.

$$aP^Ib \iff \begin{cases} \phi^+(a_i) > \phi^+(a_k) \text{ and } \phi^-(a_i) < \phi^-(a_k) & \text{, or} \\ \phi^+(a_i) = \phi^+(a_k) \text{ and } \phi^-(a_i) < \phi^-(a_k) & \text{, or} \\ \phi^+(a_i) > \phi^+(a_k) \text{ and } \phi^-(a_i) = \phi^-(a_k) \end{cases} \tag{3}$$

$$aI^Ib \iff \phi^+(a_i) = \phi^+(a_k) \text{ and } \phi^-(a_i) = \phi^-(a_k) \tag{4}$$

$$aR^Ib \iff \begin{cases} \phi^+(a_i) > \phi^+(a_k) \text{ and } \phi^-(a_i) > \phi^-(a_k) & \text{, or} \\ \phi^+(a_i) < \phi^+(a_k) \text{ and } \phi^-(a_i) < \phi^-(a_k) \end{cases} \tag{5}$$

Zadanie 4: Narysuj ranking z wykorzystaniem PROMETHEE I.

PROMETHEE II

Ta metoda jest wykorzystywana wówczas kiedy chcemy uniknąć nieporównywalności występujących w PROMETHEE I i znaleźć całkowity ranking wariantów. Aby go uzyskać, bierzemy pod uwagę **przepływ całkowity** $\phi(a_i)$ = tym wyższy, im wyższa siła i mniejsza; słabość liczony jako różnica przepływu dodatniego i ujemnego.

$$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i) \tag{6}$$

$$a_iP^{II}a_k \iff \phi(a_i) > \phi(a_k) \tag{7}$$

$$a_iI^{II}a_k \iff \phi(a_i) = \phi(a_k) \tag{8}$$

W metodach opartych na porównaniach parami może wystąpić **rank reversal problem**. To znaczy, że pomimo tego, iż w bezpośrednim porównaniu wariantów wariant a_i jest bardziej preferowany niż wariant a_k, to w rankingu końcowym wariant a_k będzie się znajdował wyżej niż wariant a_i. Jest to związane z faktem, że ostateczny ranking nie zależy jedynie od wartości ocen wariantów na kryteriach, ale również ich porównań z innymi wariantami.

PROMETHEE V

Ta metoda jest wykorzystywana dla problemów wyboru z dodatkowymi ograniczeniami. Pamiętajmy jednak, że metody PROMETHEE I i II również mogą być wykorzystane do problemów wyboru, gdzie wybieramy top k elementów jak podzbiór najlepszych wariantów. Jednakże bardzo często w problemach wyboru występuje sytuacja, w której musimy jednocześnie wybrać pewien podzbiór najlepszych wariantów, jak i uwzględnić dodatkowe ograniczenia. Całkowity przepływ wariantu z PROMETHEE II może być wykorzystany jako całkowita ocena wariantu, więc naszym celem będzie maksymalizacja sumy tych ocen dla wybranych wariantów. W celu zamodelowania wybrania wariantu wprowadzamy zmienne binarne x_i, które będą mówiły o tym, czy dany wariant został wybrany do końcowego podzbioru, czy nie.

$$\begin{cases} x_i = 1 \iff \text{wariant } a_i \text{ został wybrany} \\ x_i = 0 \iff \text{wariant } a_i \text{ nie został wybrany} \end{cases} \tag{9}$$

Zgodnie z powyższym możemy zdefiniować problem programowania matematycznego:

$$\max \sum_{i}^{n} \phi(a_i) \cdot x_i \tag{10}$$

przy ograniczeniach:

$$\sum_{i}^{n} \lambda_{p,i} \cdot x_i \{\leqslant, =\geqslant\} \beta_p \text{ for } p = 1, 2...P \tag{11}$$

$$x_i \in \{0, 1\} \text{ for } i = 1, 2...n \tag{12}$$

gdzie $\lambda_{p,i}$ oraz β_p są parametrami definiowanymi przez decydenta.
Rozważmy następujący przykład:

Tabela 5: Przykład dla PROMETHEE V - dane wejściowe

	$\phi(a_i)$	$\lambda_{1,i}$	$\lambda_{2,i}$
a_1	2.50	30.0	90.0
a_2	-0.61	135.0	105.0
a_3	-2.94	120.0	75.0
a_4	-1.43	60.0	135.0
a_5	0.27	120.0	30.0
a_6	1.72	60.0	30.0
a_7	-1.05	120.0	90.0
a_8	1.54	75.0	60.0

Z dodatkowymi ograniczeniami:

- ograniczenie 1: suma parametrów $\lambda_{1,i}$ wariantów wybranych musi być nie mniejsza niż 160
  $\sum_i^n \lambda_{1,i} x_i \geqslant 160$
- ograniczenie 2: suma parametrów $\lambda_{2,i}$ wariantów wybranych musi być nie większa niż 175
  $\sum_i^n \lambda_{2,i} x_i \leqslant 175$
- ograniczenie 3: wybieramy podzbiór jedynie 2 warianty
  $\sum_i^n x_i = 2$

Bez użycia solvera możemy sprawdzić wszystkie możliwe pary wariantów i obliczyć odpowiednie wartości dla nich:

Zadanie 5:
Znajdź optymalny zbiór wariantów, który powinien zostać wybrany.

Tabela 6: Przykład - ograniczenia dla PROMETHEE V

a_i	a_j	$\sum_i^n \phi(a_i)x_i$	$\sum_i^n \lambda_{1,i}x_i$	$\sum_i^n \lambda_{2,i}x_i$	ograniczenie 1	ograniczenie 2
a_1	a_2	1.89	165	195	T	
a_1	a_3	-0.44	150	165		T
a_1	a_4	1.07	90	225		
a_1	a_5	2.77	150	120		T
a_1	a_6	4.22	90	120		T
a_1	a_7	1.45	150	180		
a_1	a_8	4.04	105	150		T
a_2	a_3	-3.55	255	180	T	
a_2	a_4	-2.04	195	240	T	
a_2	a_5	-0.34	255	135	T	T
a_2	a_6	1.11	195	135	T	T
a_2	a_7	-1.66	255	195	T	
a_2	a_8	0.93	210	165	T	T
a_3	a_4	-4.37	180	210	T	
a_3	a_5	-2.67	240	105	T	T
a_3	a_6	-1.22	180	105	T	T
a_3	a_7	-3.99	240	165	T	T
a_3	a_8	-1.40	195	135	T	T
a_4	a_5	-1.16	180	165	T	T
a_4	a_6	0.29	120	165		T
a_4	a_7	-2.48	180	225	T	
a_4	a_8	0.11	135	195		
a_5	a_6	1.99	180	60	T	T
a_5	a_7	-0.78	240	120	T	T
a_5	a_8	1.81	195	90	T	T
a_6	a_7	0.67	180	120	T	T
a_6	a_8	3.26	135	90		T
a_7	a_8	0.49	195	150	T	T

Język Python i biblioteka PuLP

Biblioteka PuLP jest darmową biblioteką do rozwiązywania problemów opisanych językiem programowania liniowego bądź całkowitoliczbowego. Do wykreślenia uzyskanych punktów Pareto optymalnych można wykorzystać bibliotekę matplotlib. Przykładowy kod programu wykorzystującego bibliotekę PuLP znajduje się poniżej:

```
from pulp import *
import numpy as np

# Utworzenie instancji problemu
model = LpProblem(name="jakis-problem", sense=LpMaximize)

# Utworzenie dwoch zmiennych decyzyjnych
x1 = LpVariable(name="x1", lowBound=0, cat='Continuous')
x2 = LpVariable(name="x2", lowBound=0, cat='Continuous')

# Ograniczenia problemu
model += (1 * x1 + 1*x2 >= 1, "#1 constraint")
model += (2 * x1 + 1*x2 <= 6, "#2 constraint")
model += (-1 * x1 + 1*x2 == 1, "#3 constraint")

# Funkcja celu
obj_func = 4*x1 + 2 * x2
model += obj_func

# Uruchomienie solvera
status = model.solve()

# Wypisanie statusu
print(f"status: {model.status}, {LpStatus[model.status]}")
# WYNIK: status: 1, Optimal
```

```
# Wypisanie realizacji funkcji celu
print(f"objective: {model.objective.value()}")
# WYNIK: objective: 12.000000199999999

# Wypisanie wartosci zmiennych decyzyjnych
print(x1.value())
print(x2.value())
# WYNIK:
#   1.6666667
#   2.6666667


### Inny sposob -- macierzowy -- na utworzenie listy zmiennych
### oraz ograniczen

var_names = ["First", "Second"]
x = LpVariable.dicts("x", var_names, 0)
x
# WYNIK: {'First': x_First, 'Second': x_Second}

const_names = ["GE", "LE", "EQ"]
sense = [1, -1, 0] # GE, LE, EQ
coefs = [[1,1],[2,1],[-1,1]] # Matrix coefs
rhs = [1, 6, 1]

for c, s, r, cn in zip(coefs, sense, rhs, const_names):
    expr = lpSum([x[var_names[i]] * c[i] for i in range(2)])
    model += LpConstraint(e=expr, sense = s, name = cn, rhs = r)

obj_coefs = [4,2]
model += lpSum([x[var_names[i]] * obj_coefs[i] for i in range(2)])
```