

## Technologie programistyczne

Platforma Diviz - Implementacja modułowego systemu wspomagania wielokryterialnego  
porządkowania z modelem preferencji w postaci addytywnej funkcji wartości.

25 stycznia 2014

Autor: **Paweł Rychły** inf94362 ISWD pawelrychly@gmail.com

Zajęcia wtorkowe, 15:15.

## 1 Cel projektu

Celem projektu jest implementacja zbioru modułów działających na platformie Diviz. Moduły te składać się będą na system wspomagania wielokryterialnego porządkowania wariantów decyzyjnych w oparciu o metodykę odpornej regresji porządkowej z zastosowaniem addytywnej funkcji wartości. W ogólności moim zadaniem jest opracowanie metody wymiany różnorodnych typów informacji preferencyjnej pomiędzy użytkownikiem a metodą wspomagania wielokryterialnego porządkowania.

## 2 Szczegółowy opis modułów

**Moduł realizujący analizę rankingów ekstremalnych** Rozszerzenie metody wspomagania porządkowania, które dla każdego wariantu zwraca jego możliwie najlepszą i najgorszą pozycję w rankingu.

**Moduł realizujący wyszukiwanie reduktów dla relacji koniecznej słabej preferencji**

Moduł ten, dla każdej znanej koniecznej relacji słabej preferencji pomiędzy dwoma wariantami wyszukuje, takiego minimalnego zbioru porównań parami, będącego podzbiorem ograniczeń zadanych przez użytkownika, który gwarantuje zajście danej relacji.

**Moduł realizujący wyszukiwanie reduktów w oparciu o rankingi ekstremalne** Celem tego modułu jest znalezienie, dla każdego wariantu, takiego minimalnego zbioru porównań parami, będącego podzbiorem ograniczeń zadanych przez użytkownika, który gwarantowałby, że analizowany wariant zajmowałby pozycję w rankingu zgodną z jego aktualnymi rankingami ekstremalnymi.

**Moduły realizujące analizę post factum** Grupa czterech modułów, których celem jest znalezienie modyfikacji funkcji oceny wariantu lub samej wartości wariantu w taki sposób aby spełnione były zadane przez użytkownika ograniczenia. Moduły różnią się pomiędzy sobą rodzajem warunków zadawanych przez użytkownika oraz tym, czy modyfikowana będzie funkcja oceny czy sam wariant. Możliwe będą dwa typy ograniczeń zadawanych przez użytkownika. W przypadku pierwszego rodzaju, moduły będą dążyć do osiągnięcia przez pewien wariant zadanej przez użytkownika pozycji w rankingu. Drugi rodzaj ograniczenia będzie dotyczył relacji przewyższania pomiędzy dwoma wariantami.

## 3 Idea rozwiązania

Zbiór siedmiu modułów, umożliwiających uruchamianie opisanych algorytmów za pomocą platformy Decision Deck. Programy te powinny operować na danych zapisanych w postaci plików xml zgodnych ze standardem XMCDa.

## 4 Technologie i narzędzia

1. język programowania: R
2. środowisko programistyczne: RStudio
3. system kontroli wersji git
4. XMCDa, diviz

5. biblioteki `ror`, `RXMCDA`

## 5 Opis implementacji

System składa się z siedmiu niezależnych programów. Każdy z nich zaimplementowany jest zgodnie ze standardem XMCD. Dzięki temu funkcjonalność modułów może być udostępniona w postaci usług platformy Decision Deck. Każdy moduł opisany jest przy pomocy następujących plików:

1. plik zawierający kod źródłowy programu
2. plik `description-wsDD.xml` zawierający opis parametrów algorytmu.
3. folder zawierający zbiory testów algorytmu.

## 6 Instrukcja obsługi

Poniższy opis zakłada, że użytkownik posiada zainstalowane środowisko języka R. Dodatkowo aby uruchomić opisywane moduły, konieczna jest instalacja pakietów `ror` (polecenie: `install.packages("ror")`) oraz `RXMCDA` (`install.packages("RXMCDA")`). Docelowo praca z każdym z modułów powinna być możliwa za pomocą programu `Diviz`. Aktualnie możliwe jest uruchomienie algorytmów z poziomu konsoli. Obsługa programów w obu przypadkach jest bardzo podobna. Typowa interakcja pomiędzy użytkownikiem a modulem może zostać opisana w następujący sposób:

### 6.1 Przygotowanie danych wejściowych w postaci zbioru plików xml (XMCD)

Przykładowe pliki z danymi wejściowymi można znaleźć w folderach zawierających testy algorytmów.

```
<xmcda:XMCD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xmcda="http://www.decision-deck.org/2012/XMCD-2.2.0"
  xsi:schemaLocation="http://www.decision-deck.org/2012/XMCD-2.2.0 http://www.decision-deck.org/xmcda/_downloads/XMCD-2.2.0.xsd">
  <methodParameters>
    <parameter name="strict">
      <value>
        <integer>1</integer>
      </value>
    </parameter>
    <parameter name="is-possible-comprehensive-modifying">
      <value>
        <integer>1</integer>
      </value>
    </parameter>
    <parameter name="is-improvement">
      <value>
        <integer>1</integer>
      </value>
    </parameter>
  </methodParameters>
</xmcda:XMCD>
```

Rysunek 1: Przykład pliku xml (XMCD)

## 6.2 Uruchomienie algorytmu wraz z podaniem lokalizacji folderu z danymi oraz miejsca zapisu wyników.

Uruchomienie algorytmu z poziomu konsoli możliwe jest za pomocą polecenia:

```
$ R --slave --vanilla --args folder_z_danymi folder_z_wynikami  
< nazwa_algorytmu.R (Linux)  
  
$ R.exe --slave --vanilla --args folder_z_danymi folder_z_wynikami  
\\< nazwa_algorytmu.R (Windows)
```

## 6.3 Realizacja obliczeń i zapisanie wyników w postaci pliku xml (XMCDA)

# 7 Opis przeprowadzonych testów.

### 7.1 Przykład testowy

W trakcie testowania implementacji metod, wykorzystałem niewielki przykład testujący składający się z 6 wariantów opisanych przy pomocy 4 kryteriów typu zysk.

	g1	g2	g3	g4
a1	1	1	1	1
a2	3	7	3	2
a3	1	10	5	5
a4	5	5	5	5
a5	1	5	5	5
a6	10	10	10	10

Tabela 1: Zastosowany przykład

Jak można zauważyć wariant 6 dominuje wszystkie pozostałe przykłady. Wariant 1 jest zdominowany przez wszystkie inne warianty.

### 7.2 Analiza rankingów ekstremalnych - funkcja `extremeRankingAnalysis(...)`

Funkcja `extremeRankingAnalysis`, dla każdego wariantu, znajduje jego możliwie najlepszą i najgorszą pozycję w rankingach.

#### 7.2.1 ERA - Test 1

**Informacja preferencyjna** : a5 silnie preferowane nad a4.

**Oczekiwany wynik** : Jak można zauważyć warianty a5 i a4 różnią się tylko na 1 atrybucie. Ponadto wariant a4 jest na nim lepszy od wariantu a5. W tym przypadku rozwiązanie nie powinno istnieć, ponieważ ograniczenia są sprzeczne.

**Wynik** : Funkcja zwróciła informację o błędzie: "Model infeasible"

### 7.2.2 ERA - Test 2

**Informacja preferencyjna** : a2 silnie preferowane nad a3

**Oczekiwany wynik** : Jak można zauważyć, gdyby usunąć atrybut 1 wariant 3 dominował by wariant 2. Ponieważ wariant 2 jest silnie preferowany nad wariant 3, znaczenie atrybutu 1 musi być duże. Z tego powodu, można spodziewać się, że pozycja wariantu 4 będzie możliwie lepsza od pozycji wariantu 3. ( Przykład 4 przewyższa wariant 3 na atrybucie mającym duże znaczenie. ) Ponieważ wariant 1 jest zdominowany przez wszystkie inne, a wariant 6 dominuje pozostałe przykłady, ich pozycje w rankingu zawsze będą równe odpowiednio: 6 i 1.

**Wynik** : Funkcja zwróciła następujący wynik:

	najgorsza pozycja	najlepsza pozycja
a1	6	6
a2	3	2
a3	4	3
a4	4	2
a5	5	5
a6	1	1

Tabela 2: Najgorsze i najlepsze możliwe pozycje wariantów w rankingu

## 7.3 Znajdowanie reduktów preferencyjnych dla zadanych relacji koniecznych - funkcja preferentialReductsForNecessaryRelations.

Funkcja znajduje redukty preferencyjne ( minimalne podzbiory porównań wprowadzonych przez użytkownika ) indukujące dla pary atrybutów a i b, zależność a jest koniecznie słabo preferowane nad b. Funkcja znajduje takie redukty dla każdej koniecznej słabej preferencji wynikającej z informacji preferencyjnej.

### 7.3.1 PR - Test 1

**Informacja preferencyjna** : a2 silnie preferowane nad a3, a4 silnie preferowane nad a3, a6 silnie preferowany nad a5

**Oczekiwany wynik** : Jak można zauważyć podane ograniczenia są nadmiarowe. Zarówno ograniczenie 2 jak i 3 wynikają z analizowanego przykładu oraz ograniczenia 1. Oczekiwany wynik powinien zawierać redukty o liczności co najwyżej 1.

**Wynik** : Funkcja zwróciła następujące redukty:

$a2 \geq^N a1$  {zbiór pusty}  
 $a2 \geq^N a3$  {a2 silnie preferowane nad a3}  
 $a2 \geq^N a5$  {a2 silnie preferowane nad a3}  
 $a3 \geq^N a1$  {zbiór pusty}  
 $a3 \geq^N a5$  {zbiór pusty}  
 $a4 \geq^N a1$  {zbiór pusty}  
 $a4 \geq^N a3$  {a4 silnie preferowane nad a3}

$a3 \succeq^N a5$  {zbiór pusty}  
 $a5 \succeq^N a1$  {zbiór pusty}  
 $a6 \succeq^N a1$  {zbiór pusty}  
 $a6 \succeq^N a2$  {zbiór pusty}  
 $a6 \succeq^N a3$  {zbiór pusty}  
 $a6 \succeq^N a4$  {zbiór pusty}  
 $a6 \succeq^N a5$  {zbiór pusty}

Wynik ten jest zgodny z przewidywaniami. Do tego aby przykład a2 był słabo koniecznie preferowany nad a5 wystarcza ograniczenie o silnej preferencji a2 nad a3, ponieważ powoduje ono wzrost znaczenia atrybutu pierwszego.

#### 7.4 Znajdowanie reduktów preferencyjnych dla zadanych maksymalnych i minimalnych pozycji w rankingu - funkcja preferentialReductsForRanks.

Funkcja, dla każdego wariantu, znajduje redukty preferencyjne takie, że jego najlepsza pozycja w rankingu nie będzie lepsza od  $P^*$ , oraz jego najgorsza pozycja w rankingu nie będzie gorsza od  $P_*$ .  $P^*$  oraz  $P_*$  to odpowiednio najlepsza i najgorsza możliwa pozycja w rankingu. Wartości te, są wynikiem wykonania funkcji EkstremeRankingAnalysis.

##### 7.4.1 PR - Test 1

**Informacja preferencyjna** : a3 silnie preferowane nad a2, a4 silnie preferowane nad a2, a5 silnie preferowany nad a2

	najgorsza pozycja	najlepsza pozycja
a1	6	6
a2	5	5
a3	3	2
a4	3	2
a5	4	4
a6	1	1

Tabela 3: Najgorsze i najlepsze możliwe pozycje wariantów w rankingu

#### Rankingi ekstremalne

**Oczekiwany wynik** : Zarówno pozycja wariantu pierwszego jak i ostatniego wynika bezpośrednio z wartości jakie przyjmują ich oceny na wszystkich atrybutach. Wariant 1 jest zdominowany przez wszystkie pozostałe, przez co zawsze będzie on zajmował ostatnią pozycję. Podobnie ostatni wariant zawsze będzie zajmował pozycję 1. Ich redukty preferencyjne powinny być puste. Wariant 2 zajmuje zawsze 5 pozycję w rankingu. Wynika to z wprowadzenia ograniczenia "a5 silnie przewyższa a2". Sprawia ono, że atrybuty 3 i 4 zyskują na znaczeniu. Tym samym wariant ten jest przewyższany przez wszystkie warianty z wyjątkiem przykładu pierwszego. Ograniczenie "a3 silnie przewyższa a2", pomimo podobnych cech nie powinno być reduktem wariantu 2. Wynika to z faktu, że wpływa ono pozytywnie na ważność atrybutu 2, która może spowodować przewyższanie wariantu 4 przez wariant 2. Zarówno wariant 3 jak i 4 mogą przyjmować pozycję od 3

do 2. Najgorsza możliwa pozycja wynika częściowo z faktu, że oba warianty dominują wariant 5. Aby zapewnić, że nie zajmą one pozycji 4, konieczne jest wprowadzenie ograniczenia zapewniającego, że wariant drugi zajmie gorszą pozycję od dwóch wyżej wymienionych.

**Wynik** : Funkcja zwróciła następujące redukty:

- 1 - [6:6 ] {zbiór pusty}
- 2 - [5:5 ] {a5 silnie preferowane nad a2}
- 3 - [3:2 ] {a3 silnie preferowane nad a2}, { a5 silnie preferowany nad a2 }
- 4 - [3:2 ] {a4 silnie preferowane nad a2}, { a5 silnie preferowany nad a2 }
- 5 - [4:4 ] {a5 silnie preferowane nad a2}
- 6 - [1:1 ] {zbiór pusty}

Wynik ten jest zgodny z przewidywaniami.

## 7.5 Analiza post factum dla zadanej relacji przewyższania - funkcje postFactumPreferenceRelatedEvalModifying

Funkcja postFactumPreferenceRelatedEvalModifying znajduje informację o tym jak możliwie mało należy poprawić oceny wariantu i-tego, aby możliwie/koniecznie preferowany był on nad wariant j-ty.

### 7.5.1 Analiza post factum - possible improvement - Test 1

**Informacja preferencyjna** : a3 silnie preferowane nad a2

**Cel analizy** Jak bardzo należy poprawić ocenę wariantu 5 na drugim kryterium aby możliwie przewyższał on wariant 3.

**Oczekiwany wynik** : Wariant 3 różni się od wariantu 2 jedynie na kryterium 2. Aby wariant ten był nie gorszy od wariantu 3, jego ocena na tym kryterium powinna być dwukrotnie większa.

**Wynik** : Funkcja zwróciła wynik 2, co jest zgodne z naszymi oczekiwaniami.

### 7.5.2 Analiza post factum - necessary improvement - Test 2

**Informacja preferencyjna** : a3 silnie preferowane nad a2

**Cel analizy** Jak bardzo należy poprawić ocenę wariantu 5 na drugim kryterium aby koniecznie przewyższał on wariant 3.

**Oczekiwany wynik** : Wariant 3 posiada maksymalną ocenę na atrybucie 2. Na wszystkich innych atrybutach oceny wariantów są takie same. Powiększenie oceny wariantu 5 na kryterium drugim, może jedynie spowodować, że a5 będzie możliwie przewyższał a3. Relacja konieczna nie jest osiągalna.

**Wynik** : Funkcja zwróciła komunikat o braku rozwiązania.

### 7.5.3 Analiza post factum - necessary improvement - Test 3

**Informacja preferencyjna** : a2 silnie preferowane nad a3

**Cel analizy** Jak bardzo należy poprawić ocenę wariantu 5 na pierwszym kryterium aby koniecznie przewyższał on wariant 4.

**Oczekiwany wynik** : Wprowadzona informacja preferencyjna powoduje silny wzrost znaczenia atrybutu pierwszego względem pozostałych. Ocena wariantu 4 na kryterium 5 jest pięć razy lepsza od oceny wariantu 5 na tym atrybucie. Z tego powodu oczekiwanym wynikiem jest liczba zbliżona do 5.

**Wynik** : Funkcja zwróciła wartość 5 co jest zgodne z oczekiwaniami.

## 7.6 Analiza post factum dla zadanej pozycji w rankingu - funkcje postFactumRankRelatedEvalModifying

Funkcja postFactumRankRelatedEvalModifying znajduje informację o tym jak możliwie mało należy poprawić oceny wariantu i-tego, aby możliwie/koniecznie zajmował on co najmniej k-tą pozycję w rankingu.

### 7.6.1 Analiza post factum - possible Improvement - Test 1

**Informacja preferencyjna** : a3 silnie preferowane nad a2

**Cel analizy** Jak bardzo należy poprawić oceny wariantu 2 aby możliwie zajmował on przynajmniej 2 pozycję.

**Oczekiwany wynik** : Jak można zauważyć, aby wariant 1 osiągnął pozycję drugą w rankingu, dla przynajmniej jednej funkcji, musi on przewyższać wszystkie przykłady z wyjątkiem a6. Na przynajmniej jednym kryterium wariant musi być nie gorszy od tych wariantów. Taki warunek będzie spełniony, jeśli pięciokrotnie powiększymy oceny tego wariantu na wszystkich kryteriach.

**Wynik** : Funkcja zwróciła wynik 5, co jest zgodne z naszymi oczekiwaniami.

### 7.6.2 Analiza post factum - necessary improvement - Test 1

**Informacja preferencyjna** : a3 silnie preferowane nad a2

**Cel analizy** Jak bardzo należy poprawić ocenę wariantu 5 aby koniecznie (dla wszystkich funkcji) zajmował on co najmniej pozycję 2.

**Oczekiwany wynik** : Aby dla wszystkich możliwych funkcji użyteczności, wariant pierwszy zajmował co najmniej drugą pozycję, musiałby on zawsze przewyższać wszystkie przykłady z wyjątkiem ostatniego. Aby osiągnąć ten cel, konieczne było by 10 krotne powiększenie ocen na jego atrybutach.

**Wynik** : Funkcja zwróciła wartość 10, co jest zgodne z oczekiwaniami.



## 7.7 Analiza post factum dla zadanej relacji preferencji - funkcja modyfikująca użyteczność: postFactumPreferenceRelatedUtilityModifying

**7.7.1 Test 1 - Poszukiwanie minimalnej wartości o jaką należy zwiększyć użyteczność wariantu a1 by możliwie zachodziła relacja: a1 preferowany nad a4**

**informacja preferencyjna** :  $a2 > a3$   $a4 > a5$

**Liczba punktów charakterystycznych na atrybutach** : [c1: 2,c2: 2,c3: 2,c4: 2]

**Parametry** : strict = 1; is-possible-comprehensive-modifying = 1; is-improvement = 1;

**Cel** : a1 preferowany nad a4.

**Oczekiwany wynik** : Suma użyteczności a1 powiększona o znaną wartość powinna być równa sumie użyteczności a4.

**Wynik** : Funkcja zwróciła wartość 0.44444. Funkcje użyteczności powiązane z każdym z atrybutów są liniowymi funkcjami, które rozpoczynają się i kończą w punktach charakterystycznych. Poniżej wypisano wartości funkcji użyteczności odpowiadające punktom charakterystycznym w postaci (wartość punktu charakterystycznego, użyteczność w tym punkcie).

**atrybut 1** (1, 0), (10, 0.500275)

**atrybut 2** (1, 0), (10, 0.0001)

**atrybut 3** (1, 0), (10, 0.499525)

**atrybut 4** (1, 0), (10, 0.0001)

Znając funkcje użyteczności można obliczyć wartości użyteczności wariantów a1 i a4:  $U(a1) = 0$

$U(a4) = \frac{5-1}{10-1} * 0.500275 + \frac{4}{9} * 0.0001 + \frac{4}{9} * 0.499525 + \frac{4}{9} * 0.0001 = \frac{4}{9} * 1 \approx 0.44444$

Jak widać  $U(a1) + 0.44444 = U(a2)$ . Wynik jest więc zgodny z oczekiwaniami.

**7.7.2 Test 2 - Poszukiwanie maksymalnej wartości o jaką można zmniejszyć użyteczność wariantu by wciąż zachodziła relacja: a3 preferowane nad a2**

**informacja preferencyjna** :  $a2 > a5$

**Liczba punktów charakterystycznych na atrybutach** : [c1: 2,c2: 2,c3: 2,c4: 2]

**Parametry** : strict = 1; is-possible-comprehensive-modifying = 1; is-improvement = 0;

**Cel** : a3 preferowany nad a2.

**Oczekiwany wynik** : Aby a3 był ciągle możliwie preferowany nad a2, wartość Umiss powinna spełniać równanie:  $U(a3) - U_{miss} = U(a2)$

**Wynik** Wynik 0.3332667 co jest zgodne z oczekiwaniami. Punkty charakterystyczne:

**atrybut 1** (1, 0), (10, 0.0001)

**atrybut 2** (1, 0), (10, 0.60006)

**atrybut 3** (1, 0), (10, 0.0001)

**atrybut 4** (1, 0), (10, 0.39974)

$$U(a2) = \frac{3-1}{10-1} * 0.0001 + \frac{6}{9} * 0.60006 + \frac{2}{9} * 0.0001 + \frac{1}{9} * 0.39974 \approx 0.00002 + 0.40004 + 0.00002 + 0.04441 \approx 0.44449$$

$$U(a3) = \frac{1-1}{10-1} * 0.0001 + \frac{9}{9} * 0.60006 + \frac{4}{9} * 0.0001 + \frac{4}{9} * 0.39974 \approx 0 + 0.60006 + 0.00004 + 0.17766 \approx 0.777764$$

A więc  $U(a2) = U(a3) - 0.3332667$

**7.7.3 Test 3 - O ile należy poprawić a3 aby: a3 było preferowane nad a2 dla wszystkich możliwych funkcji.**

**informacja preferencyjna :**  $a2 > a5$

**Liczba punktów charakterystycznych na atrybutach :** [c1: 2,c2: 2,c3: 2,c4: 2]

**Parametry :** strict = 1; is-possible-comprehensive-modifying = 0; is-improvement = 1;

**Cel :** a3 koniecznie preferowany nad a2.

**Oczekiwany wynik :** Aby a3 był koniecznie preferowany nad a2, wartość Umiss powinna spełniać równanie:  $U(a3) + U_{miss} = U(a2)$

**Wynik** Wynik 0.2220667 co jest zgodne z oczekiwaniami. Punkty charakterystyczne:

**atrybut 1** (1, 0), (10, 0.9997)

**atrybut 2** (1, 0), (10, 0.0001)

**atrybut 3** (1, 0), (10, 0.0001)

**atrybut 4** (1, 0), (10, 0.0001)

$$U(a2) = \frac{3-1}{10-1} * 0.9997 + \frac{6}{9} * 0.0001 + \frac{2}{9} * 0.0001 + \frac{1}{9} * 0.0001 \approx 0.22215 + 0.00006 + 0.00002 + 0.00001 \approx 0.22287$$

$$U(a3) = \frac{1-1}{10-1} * 0.9997 + \frac{9}{9} * 0.0001 + \frac{4}{9} * 0.0001 + \frac{4}{9} * 0.0001 \approx 0 + 0.0001 + 0.00004 + 0.00004 \approx 0.00018$$

A więc  $U(a2) = U(a3) + 0.2220667$

**7.7.4 Test 4 - Poszukiwanie minimalnej wartości o jaką należy zmniejszyć użyteczność wariantu by możliwie zaszła relacja przeciwna do zadanej: a4 preferowane nad a1**

**informacja preferencyjna :**  $a2 > a3$   $a4 > a5$

**Liczba punktów charakterystycznych na atrybutach :** [c1: 2,c2: 2,c3: 2,c4: 2]

**Parametry :** strict = 1; is-possible-comprehensive-modifying = 0; is-improvement = 0;

**Cel :** a4 preferowany nad a1.

**Oczekiwany wynik :** Zgodnie z wynikiem testu 1, aby a1 był możliwie preferowany nad a4, konieczne jest dodanie do jego użyteczności wartości 0.44444. W aktualnym teście badamy jak bardzo można zmniejszyć wartość użyteczności a4 aby możliwie a1 przewyższało a4. Wynik powinien być taki sam jak w przypadku poprzedniego testu.

**Wynik** Wynik 0.44444 co jest zgodne z oczekiwaniami.

**7.7.5 Test 5 - Poszukiwanie minimalnej wartości o jaką należy zmniejszyć użyteczność wariantu by możliwie zaszła relacja przeciwna do zadanej: a2 preferowany nad a1**

**informacja preferencyjna** :  $a2 > a3$   $a4 > a5$

**Liczba punktów charakterystycznych na atrybutach** : [c1: 2,c2: 2,c3: 2,c4: 2]

**Parametry** : strict = 1; is-possible-comprehensive-modifying = 0; is-improvement = 0;

**Cel** : a2 preferowany nad a1.

**Oczekiwany wynik** : Poszukujemy liczby Umiss o jaką trzeba zmniejszyć użyteczność wariantu a2, tak aby możliwie a1 przewyższał a2.  $U(a1) = U(a2) - U_{miss}$

**Wynik** Wynik: 0.1778622

Wynik jest zgodny z oczekiwaniami: Punkty charakterystyczne:

**atrybut 1** (1, 0), (10, 0.60016)

**atrybut 2** (1, 0), (10, 0.0001)

**atrybut 3** (1, 0), (10, 0.0001)

**atrybut 4** (1, 0), (10, 0.39964)

$U(a1) = 0$

$U(a2) = \frac{3-1}{10-1} * 0.60016 + \frac{6}{9} * 0.0001 + \frac{2}{9} * 0.0001 + \frac{1}{9} * 0.39964 \approx 0.13337 + 0.00006 + 0.00002 + 0.04440 \approx 0.177854$

A więc  $U(a1) = U(a2) - 0.1778622$

## 7.8 Analiza post factum dla zadanej pozycji w rankingu - funkcja modyfikująca użyteczność: postFactumRankRelatedUtility-Modifying

### 7.8.1 Przypomnienie przykładu testowego.

	c1	c2	c3	c4
a1	1	1	1	1
a2	3	7	3	2
a3	1	10	5	5
a4	5	5	5	5
a5	1	5	5	5
a6	10	10	10	10

Tabela 4: Zastosowany przykład

**7.8.2 Test 1 - Poszukiwanie minimalnej wartości o jaką należy zwiększyć użyteczność wariantu a1, aby zajmował on conajmniej 4 pozycję w rankingu dla przynajmniej jednej kompatybilnej funkcji wartości.**

**informacja preferencyjna** :  $a2 > a3$

**Liczba punktów charakterystycznych na atrybutach** : brak punktów charakterystycznych

**Parametry** : strict = 1; is-possible-comprehensive-modifying = 1; is-improvement = 1;

**Cel** : a1 znajduje się na conajmniej 4 pozycji w rankingu dla przynajmniej jednej kompatybilnej funkcji wartości.

**Oczekiwany wynik** : Suma użyteczności a1 powiększona o znaną wartość powinna być nie gorsza od użyteczności przynajmniej dwóch innych wariantów.

	najgorsza pozycja	najlepsza pozycja
a1	6	6
a2	3	2
a3	4	3
a4	4	2
a5	5	5
a6	1	1

Tabela 5: Najgorsze i najlepsze możliwe pozycje wariantów w rankingu przed dodaniem znalezionej użyteczności

**Wynik** : Funkcja zwróciła wartość 0.0007. Użyteczności dla poszczególnych wartości atrybutów:

**atrybut 1** (1, 0), (3, 0.9989), (5, 0.9990), (10, 0.9991)

**atrybut 2** (1, 0), (5, 0.0001), (7, 0.0002), (10, 0.0003)

**atrybut 3** (1, 0), (3, 0.0001), (5, 0.0002), (10, 0.0003)

**atrybut 4** (1, 0), (2, 0.0001), (5, 0.0002), (10, 0.0003)

Znając funkcje użyteczności można obliczyć wartości użyteczności wszystkich wariantów:

	c1	c2	c3	c4	suma użyteczności—
a1	0 (1)	0 (1)	0 (1)	0 (1)	0
a2	0.9989 (3)	0.0002 (7)	0.0001 (3)	0.0001 (2)	0.9993
a3	0 (1)	0.0003 (10)	0.0002 (5)	0.0002 (5)	0.0007
a4	0.9990 (5)	0.0001 (5)	0.0002 (5)	0.0002 (5)	0.9995
a5	0 (1)	0.0001 (5)	0.0002 (5)	0.0002 (5)	0.0005
a6	0.9991 (10)	0.0003 (10)	0.0003 (10)	0.0003 (10)	1

Tabela 6: Wartości użyteczności na poszczególnych atrybutach oraz ich suma. W nawiasie podane są wartości ocen.

Jak można zauważyć użyteczność a1 po dodaniu znalezionej wartości, będzie nie gorsza od użyteczności wariantów a3 ( 0.0007) oraz a5 (0.0005). Wariant ten, będzie zajmował więc pozycję w rankingu nie gorszą od 4. Znalazona wartość jest więc poprawna.

**7.8.3 Test 2 - Poszukiwanie maksymalnej wartości o jaką można zmniejszyć użyteczność wariantu a4, aby zajmował on conajmniej 4 pozycję w rankingu dla przynajmniej jednej kompatybilnej funkcji wartości.**

**informacja preferencyjna** :  $a2 > a3$

**Liczba punktów charakterystycznych na atrybutach** : brak punktów charakterystycznych

**Parametry** : strict = 1; is-possible-comprehensive-modifying = 1; is-improvement = 0;

**Cel** : a1 znajduje się na conajmniej 4 pozycji w rankingu dla przynajmniej jednej kompatybilnej funkcji wartości.

**Oczekiwany wynik** : Suma użyteczności a4 pomniejszona o znaną wartość powinna być nie gorsza od użyteczności przynajmniej dwóch innych wariantów.

	najgorsza pozycja	najlepsza pozycja
a1	6	6
a2	3	2
a3	4	3
a4	4	2
a5	5	5
a6	1	1

Tabela 7: Najgorsze i najlepsze możliwe pozycje wariantów w rankingu przed pomniejszeniem a4 o znaną użyteczność

**Wynik** : Funkcja zwróciła wartość 0.999. Użyteczności dla poszczególnych wartości atrybutów:

**atrybut 1** (1, 0), (3, 0.9989), (5, 0.9990), (10, 0.9991)

**atrybut 2** (1, 0), (5, 0.0001), (7, 0.0002), (10, 0.0003)

**atrybut 3** (1, 0), (3, 0.0001), (5, 0.0002), (10, 0.0003)

**atrybut 4** (1, 0), (2, 0.0001), (5, 0.0002), (10, 0.0003)

Znając funkcje użyteczności można obliczyć wartości użyteczności wszystkich wariantów:

	c1	c2	c3	c4	suma użyteczności—
a1	0 (1)	0 (1)	0 (1)	0 (1)	0
a2	0.9989 (3)	0.0002 (7)	0.0001 (3)	0.0001 (2)	0.9993
a3	0 (1)	0.0003 (10)	0.0002 (5)	0.0002 (5)	0.0007
a4	0.9990 (5)	0.0001 (5)	0.0002 (5)	0.0002 (5)	0.9995
a5	0 (1)	0.0001 (5)	0.0002 (5)	0.0002 (5)	0.0005
a6	0.9991 (10)	0.0003 (10)	0.0003 (10)	0.0003 (10)	1

Tabela 8: Wartości użyteczności na poszczególnych atrybutach oraz ich suma. W nawiasie podane są wartości ocen.

Jak można zauważyć użyteczność a4, która obecnie wynosi 0.9995, po odjęciu znalezionej wartości, będzie nie gorsza od użyteczności wariantów a1 (0.0000) oraz a5 (0.0005). Wariant ten, będzie zajmował więc pozycję w rankingu nie gorszą od 4. Znalazona wartość jest więc poprawna.

**7.8.4 Test 3 - Poszukiwanie minimalnej wartości o jaką należy powiększyć użyteczność wariantu a4, aby zajmował on co najmniej 2 pozycję w rankingu dla wszystkich kompatybilnych funkcji wartości.**

**informacja preferencyjna :**  $a2 > a3$

**Liczba punktów charakterystycznych na atrybutach :** [c1: 3, c2:3, c3:3, c4:3]

**Parametry :** strict = 1; is-possible-comprehensive-modifying = 0; is-improvement = 1;

**Cel :** a1 znajduje się na co najmniej 2 pozycji w rankingu dla wszystkich kompatybilnych funkcji wartości.

**Oczekiwany wynik :** Suma użyteczności a4 powiększona o znaną wartość powinna być nie lepsza od użyteczności przynajmniej dwóch innych wariantów.

**Wynik :** Funkcja zwróciła wartość 0.1105194. Użyteczności dla punktów charakterystycznych:

**atrybut 1** (1, 0), (5.5, 0.0006), (10, 0.0007)

**atrybut 2** (1, 0), (5.5, 0.9987), (10, 0.9988)

**atrybut 3** (1, 0), (5.5, 0.0001), (10, 0.0002)

**atrybut 4** (1, 0), (5.5, 0.0001), (10, 0.0002)

Znając funkcje użyteczności można obliczyć wartości użyteczności wszystkich wariantów:

Kryterium pierwsze:  $u_1(1) = 0$

$u_1(3) = \frac{2}{4.5} * 0.0006 \approx 0.00026$

$u_1(5) = \frac{4}{4.5} * 0.0006 \approx 0.00053$

$u_1(10) = 0.0007$

$u_2(1) = 0$

$u_2(5) = \frac{4}{4.5} * 0.9987 \approx 0.88773$

$u_2(7) = 0.9988 + \frac{7-5.5}{10-5.5} * 0.0001 \approx 0.99883$

$u_2(10) = 0.9989$

$u_3(1) = 0$

$u_3(3) = \frac{2}{4.5} * 0.0001 \approx 0.00004$

$u_3(5) = \frac{4}{4.5} * 0.0001 \approx 0.00008$

$u_3(10) = 0.0002$

$u_4(1) = 0$

$u_4(2) = \frac{1}{4.5} * 0.0001 \approx 0.00002$

$u_4(5) = \frac{4}{4.5} * 0.0001 \approx 0.00008$

$u_4(10) = 0.0002$

Jak można zauważyć użyteczność a4, która obecnie wynosi 0.8884, po dodaniu znalezionej wartości 0.1105194 będzie wynosić około: 0.9990, tym samym będzie nie lepsza od użyteczności wariantów a2 ( 0.9992), a3 (0.9990) oraz a6 (1). Wariant ten, będzie zajmował więc pozycję w rankingu nie gorszą od 2. Znalaziona wartość jest więc poprawna.

	c1	c2	c3	c4	suma użyteczności—
a1	0 (1)	0 (1)	0 (1)	0 (1)	0
a2	0.00026 (3)	0.99883 (7)	0.00004(3)	0.00002 (2)	0.9992
a3	0 (1)	0.9989 (10)	0.00008 (5)	0.00008 (5)	0.9990
a4	0.00053 (5)	0.88773 (5)	0.00008 (5)	0.00008 (5)	0.88842
a5	0 (1)	0.88773 (5)	0.00008 (5)	0.00008 (5)	0.8879
a6	0.0007 (10)	0.9989 (10)	0.0002 (10)	0.0002(10)	1

Tabela 9: Wartości użyteczności na poszczególnych atrybutach oraz ich suma. W nawiasie podane są wartości ocen.

**7.8.5 Test 4 - Poszukiwanie maksymalnej wartości o jaką można pomniejszyć użyteczność wariantu a4, aby nadal zajmował on co najmniej 4 pozycję w rankingu dla wszystkich kompatybilnych funkcji wartości.**

**informacja preferencyjna :**  $a2 > a3$

**Liczba punktów charakterystycznych na atrybutach :** [c1: 3, c2:3, c3:3, c4:3]

**Parametry :** strict = 1; is-possible-comprehensive-modifying = 0; is-improvement = 0;

**Cel :** a1 znajduje się na co najmniej 4 pozycji w rankingu dla wszystkich kompatybilnych funkcji wartości.

**Oczekiwany wynik :** Suma użyteczności a4 pomniejszona o znaną wartość powinna być nie lepsza od użyteczności przynajmniej czterech innych wariantów.

**Wynik :** Funkcja zwróciła wartość 0.0006555556. Użyteczności dla punktów charakterystycznych:

**atrybut 1** (1, 0), (5.5, 0.0006), (10, 0.9994)

**atrybut 2** (1, 0), (5.5, 0.0001), (10, 0.0002)

**atrybut 3** (1, 0), (5.5, 0.0001), (10, 0.0002)

**atrybut 4** (1, 0), (5.5, 0.0001), (10, 0.0002)

Znając funkcje użyteczności można obliczyć wartości użyteczności wszystkich wariantów:

Kryterium pierwsze:  $u_1(1) = 0$

$u_1(3) = \frac{2}{4.5} * 0.0006 \approx 0.00026$

$u_1(5) = \frac{4}{4.5} * 0.0006 \approx 0.00053$

$u_1(10) = 0.9994$

$u_2(1) = 0$

$u_2(5) = \frac{4}{4.5} * 0.0001 \approx 0.00009$

$u_2(7) = 0.0001 + \frac{7-5.5}{10-5.5} * 0.0001 \approx 0.00013$

$u_2(10) = 0.0002$

$u_3(1) = 0$

$u_3(3) = \frac{2}{4.5} * 0.0001 \approx 0.00004$

$u_3(5) = \frac{4}{4.5} * 0.0001 \approx 0.00008$

$u_3(10) = 0.0002$

$$\begin{aligned}
u_4(1) &= 0 \\
u_4(2) &= \frac{1}{4.5} * 0.0001 \approx 0.00002 \\
u_4(5) &= \frac{4}{4.5} * 0.0001 \approx 0.00008 \\
u_4(10) &= 0.0002
\end{aligned}$$

	c1	c2	c3	c4	suma użyteczności—
a1	0 (1)	0 (1)	0 (1)	0 (1)	0
a2	0.00026 (3)	0.00013 (7)	0.00004(3)	0.00002 (2)	0.00045
a3	0 (1)	0.0002 (10)	0.00008 (5)	0.00008 (5)	0.00036
a4	0.00053 (5)	0.00009 (5)	0.00008 (5)	0.00008 (5)	0.00078
a5	0 (1)	0.00009 (5)	0.00008 (5)	0.00008 (5)	0.00025
a6	0.9994 (10)	0.0002 (10)	0.0002 (10)	0.0002(10)	1

Tabela 10: Wartości użyteczności na poszczególnych atrybutach oraz ich suma. W nawiasie podane są wartości ocen.

Jak można zauważyć użyteczność a4, która obecnie wynosi 0.00078, po usunięciu znalezionej wartości 0.0006555556 będzie wynosić około: 0.00012, tym samym będzie nie lepsza od użyteczności wariantów a2 ( 0.00045), a3 (0.00036), a5 (0.00025) oraz a6 (1). Wariant ten, będzie zajmował więc pozycję w rankingu nie gorszą od 4. Znalazona wartość jest więc poprawna.