

Metoda AHP in reševanje v jeziku R

red. prof. dr. Robert Leskovar

9. november 2022

Povzetek

Spominčica obravnava večkriterijsko metodo za podpora odločanju z nazivom **analiziranje hierarhije procesov**. Kratek naziv za metodo, ki jo je v 70-tih letih 20.-tega stoletja razvil profesor Thomas L. Saaty, je AHP. AHP omogoča modeliranje in analiziranje kompleksnih odločitev. Racionalno komponento predstavlja matrična algebra, ocenjevanje pa temelji na izsledkih psihologije. Predstavljena so pravila za definiranje matrik odnosov med kriteriji in alternativami. V jeziku R so prikazani trije primeri izračuna a) lastnih vektorjev, b) lastnih vrednosti, c) konsistentnosti matrike odnosov (IC) in razmerja konsistentnosti matrike odnosov (CR). V drugem primeru je poleg navedenega razvita funkcija **fprim_matrika**, ki vrne normaliziran lastni vektor, stolpec, v katerem se nahaja lastni vektor matrike, IC in CR. V tretjem primeru je izračunana še funkcija koristi pri izbiri najprimernejšega projektnega sodelavca. Z metodo AHP je možno modelirati tako kvalitativne kot kvantitativne kriterije izbire. Čeprav ni teoretične omejitve pri številu kriterijev in njihovi strukturi (širina in globina odločitvenega drevesa), je pri konkretnih aplikacijah zaradi porabe časa (stroški človeškega napora) smiselno uporabiti do deset kriterijev v eni veji odločitvenega drevesa ter do treh nivojev globine hierarhije. Metoda AHP je uporabna pri samostojnem (individualnem) in skupinskem odločanju v vrsti situacij v poslovnem, osebnem in družbenem življenju.

Kazalo

1	Uvod	3
2	Matrika odnosov	3
3	Funkcija koristi	3
4	Izračun uteži in parcialnih dosežkov alternativ	4
5	Primer 1 v jeziku R - izračun lastnega vektorja	5
6	Primer 2 v jeziku R - izračun funkcije koristi	7
7	Primer 3 v R - večkriterijska izbira novega projektne- glavca	10
	Literatura	20

Generirano 9. november 2022 09:42:41 CET

1 Uvod

Bolje je imeti nekaj od nečesa kot
pa nič od ničesar

Max Bunker

Profesor Thomas L. Saaty je kreator ene najpomembnejših sodobnih metod za večkriterijsko odločanje. Metoda AHP je osnova za številne izvedenke kot je npr. mehka metoda AHP (fuzzy AHP method), neodločna mehka lingvistična metoda AHP (hesitant fuzzy linguistic analytic hierarchical priorities, HFL AHP) in podobne.

Odločitveno situacijo oz. odločitveni problem predstavlja: a) hierarhija kriterijev (sodil), b) prioritete (uteži), ki so podane z matrikami odnosov med kriteriji in c) parcialni dosežki alternativ, ki so prav tako podani z matrikami odnosov za vsak kriterij. Najprimernejša alternativa doseže najvišjo vrednost funkcije koristi, ta pa je vsota zmnožkov uteži in dosežkov.

2 Matrika odnosov

Matrika odnosov popisuje povezave med pari kriterijev in tudi alternativ. V tem besedilu bomo matriko odnosov označili z A , posamezne elemente pa a_i^j , pri čemer i označuje vrstico, j pa stolpec. Značilnosti matrike odnosov A :

1. matrika je kvadratna; dimenzija n je enaka številu kriterijev (ali alternativ) v posameznem korenskem kriteriju odločitvenega drevesa; za vsak korenski element definiramo novo matriko odnosov
2. vrednost elementov v glavni diagonali je ena (1), pri čemer glavna diagonala poteka od zgornjega, skrajno levega elementa do spodnjega, skrajno desnega elementa
3. pri ostalih elementih velja :

$$a_i^j = \frac{1}{a_j^i} \quad (1)$$

Relacija velja tudi v obratni smeri, kar pomeni, da za rešitev problema določimo $\frac{n \times (n-1)}{2}$ elementov matrike. Primer: matriki z dimenzijo 3×3 določimo 3 elemente; v glavni diagonali imamo 3 enke, določimo 3 elemente nad glavno diagonalo, elemente pod glavno diagonalo pa izračunamo kot recipročne vrednosti elementov, preslikanih preko glavne diagonale.

3 Funkcija koristi

Funkcija koristi je vsota produktov pomembnosti posameznega elementarnega kriterija (uteži) in dosežka določene alternative pri tem kriteriju. Določa vrednost, dosežek alternative in služi za rangiranje alternativ. V nadaljevanju bomo funkcijo koristi $i - te$ alternative označili kot Q_i , pomembnost $i - tega$ kriterija

kot w_i in dosežek i -te alternative pri j -tem kriteriju kot q_i^j . Formula funkcije koristi je:

$$Q_i = \sum_{i=1}^n w_i \times q_i^j \quad (2)$$

Pomen oznak:

- Q_i funkcija koristi i -te alternative
- n število vrstic (tudi stolpev) matrike odnosov
- w_i pomembnost i -tega kriterija
- q_i^j dosežek i -te alternative pri j -tem kriteriju

Večja vrednost funkcije koristi običajno pomeni prednost, razen v primeru, ko problem formuliramo kot iskanje minimuma.

4 Izračun uteži in parcialnih dosežkov alternativ

Za izračun uteži kriterijev in parcialnih dosežkov alternativ je potrebno poiskati lastne vektorje in lastne vrednosti matrike odnosov. Neskladnosti pri odločanju o pomembnosti kriterijev in parcialnih dosežkov alternativ pri posameznih kriterijih ugotovimo z indeksom konsistentnosti in razmerjem konsistentosti glede na velikost matrike.

Za izračun pomembnosti i -tega kriterija in tudi za izračun dosežka i -te alternative pri j -tem kriteriju rešimo matrično enačbo:

$$A \times w = \lambda_{max} \times w \quad (3)$$

Pomen oznak:

- A matrika odnosov
- w lastni vektor matrike A
- λ_{max} največja realna lastna vrednost matrike A

Matriko odnosov tvorimo za primerjavo: a) kriterijev in b) dosežkov alternativ pri definiranih kriterijih. Izračun lastnih vektorjev in lastnih vrednosti vrne več rešitev. Uporabimo lastni vektor w , ki ima naslednje lastnosti:

- vsi realni deli elementov lastnega vektorja so enako predznačeni (vsi samo pozitivno ali vsi samo negativno)
- imaginarni del kateregakoli elementa lastnega vektorja je $0i$.

Med več lastnimi vrednostmi matrike izberemo največjo realno vrednost. Slednjo bomo uporabili pri izračunu indeksa konsistentnosti (CI, consistency index) in razmerja konsistentnosti (CR, consistency ratio).

Indeks konsistentnosti označuje skladnost med pari ocen v matriki odnosov. Je brezdimenzijska vrednost, v idealnem primeru pa je 0. Indeks konsistentnosti

izračunamo zaradi preverjanja pravilnosti oz. še boljše skladnosti ocen v matriki odnosov. Formula za indeks konsistentnosti je razmerje:

$$IC = (\lambda_{max} - n)/(n - 1) \quad (4)$$

Pomen oznak:

- λ_{max} največja realna lastna vrednost matrike A
- n število vrstic (tudi stolpcev) matrike odnosov

Saaty je definirala konstante za naključne matrike odnosov, ki imajo dimezije od 1 do 15. V tabeli 1 so podane konstante za matrike do dimenzije do 12. Za

Tabela 1: Konstante za naključne matrike odnosov velikosti 1 do 12.

1	2	3	4	5	6	7	8	9	10	11	12
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48

velikosti matrik (13, 14 in 15) so pripadajoče konstante (1.56, 1.57 in 1.59). Razmerje konsistentnosti (CR, consistency ratio) določa formula:

$$CR = IC/konstanta \quad (5)$$

Včasih pri velikosti matrike 4 in 5 pri praktični uporabi vrednost CR aproksimiramo kar s IC, ker je konstanta zelo blizu 1. Kadar je vrednost CR manjša od 0.1, smatramo, da je matrika odnosov skladna (konsistentna), sicer moramo popraviti vrednosti. Ocene v matriki odnosov so lahko neustrezne zaradi več vzrokov. Najpogostejši vzrok je tipkarska napaka. Lahko pa so ocene neustrezne ker odločevalec namerno preferira določeno alternativo. Metoda AHP zelo dobro zazna kadar je ocenjevalec (odločevalec) pristranski, saj mora biti res sistematično pristran. Slednjega seveda tudi metoda AHP ne more zaznati.

5 Primer 1 v jeziku R - izračun lastnega vektorja

V tem primeru se bomo osredotočili zgolj na matriko odnosov med kriteriji, izračun lastnih vektorjev, lastnih vrednosti, določitev primerne realnega lastnega vektorja, največje lastne vrednosti, indeksa konsistentnosti (IC) in razmerja konsistentnosti. Alternativ tu ne bomo posebej obravnavali, ker je postopek računanja popolnoma enak kot pri kriterijih. Najprej definiramo matriko odnosov A (glej tabelo 2). Privzeta sintaksa oblikovanja matrike v jeziku R je stolpčna orientiranost. To pomeni, da najprej zapišemo vse vrednosti prvega stolpca, nato drugega in dalje do zadnjega.

```
library(xtable) #za generiranje tabele v latex-u
A <-matrix(c(1,1,1/2,1,1,1/2,2,2,1),nrow = 3, ncol = 3)
print(xtable(A))
```

Izračunamo lastni vektor in lastno vrednost:

	1	2	3
1	1.00	1.00	2.00
2	1.00	1.00	2.00
3	0.50	0.50	1.00

```
rezultat <- eigen(A)
```

Spremenljivka ali bolje objekt *rezultat* vsebuje lastne vektorje v spremenljivki *vectors* in lastne vrednosti v spremenljivki *values*. Zaradi krajšega zapisa lastne vektorje in lastne vrednosti shranimo ločeno v posebnih delovnih spremenljivkah: *vektorji* in *vrednosti*:

```
vektorji <- rezultat$vectors
vrednosti <- as.matrix(rezultat$values)
print(vektorji)

##           [,1]      [,2]      [,3]
## [1,] 0.6666667 -0.8728716  0.0000000
## [2,] 0.6666667  0.4364358 -0.8944272
## [3,] 0.3333333  0.2182179  0.4472136

print(vrednosti)

##           [,1]
## [1,] 3.000000e+00
## [2,] -2.220446e-16
## [3,] 0.000000e+00

#pripravimo tabelo za Latex
print(xtable(vektorji))

print(xtable(vrednosti))
```

Lastni vektorji matrike A so prikazani tudi v tabeli 3, lastne vrednosti pa v tabeli 4

	1	2	3
1	0.67	-0.87	0.00
2	0.67	0.44	-0.89
3	0.33	0.22	0.45

Izberemo lastni vektor, ki ima samo realne vrednosti in so le te enako predznačene. V prikazanem primeru pogojem ustreza prvi vektor (to je prvi stolpec v matriki rezultata, tabela 3). Zato ta vektor normaliziramo za nadaljnjo uporabo (glej tabelo 5).

Tabela 4: Lastne vrednosti matrice A .

	x
1	3.00
2	-0.00
3	0.00

```
sum(vektorji[,1])

## [1] 1.666667

normalizirani_vektor <- vektorji[,1]/sum(vektorji[,1])
print(xtable(as.matrix(normalizirani_vektor)))
```

Tabela 5: Normaliziran prvi lastni vektor matrice A .

	x
1	0.40
2	0.40
3	0.20

Izračunali smo normalizirane uteži kriterijev. Sledi še preverjanje konsistentnosti. Največja lastna vrednost matrice je 3 (glej tabelo 4). Indeks konsistentnosti (IC) je zato 0, prav tako tudi razmerje konsistentnosti (CR).

```
konst <- c(0.00, 0.00, 0.58, 0.90, 1.12, 1.24, 1.32, 1.41,
          1.45, 1.49, 1.51, 1.48, 1.56, 1.57, 1.59)
ikon <- (max(vrednosti) - nrow(A))/(nrow(A) - 1)
rkon <- ikon/konst[nrow(A)]
```

Izračunano razmerje konsistentnosti (CR) je 0. Ker je to < 0.1 , lahko normalizirani lastni vektor uporabimo v izračunu funkcije koristi.

6 Primer 2 v jeziku R - izračun funkcije koristi

V tem primeru prikazujemo programsko kodo brez podrobne razlage. Ta primer obravnava tako kriterije kot tudi alternative ter izračun funkcije koristi. V jeziku R smo sprogramirali smo funkcijo `fprim_vektor` za obdelavo matrice odnosov. Ta funkcija sprejme matriko odnosov B kot vhodni parameter, vrne pa sestavljeni vektor: a) normalizirani vektor, b) stolpec, v katerem je zapisan lastni vektor iz funkcije *eigen*, c) indeks konsistentnosti (IC) in d) razmerje konsistentnosti (CR). Če je v kateremkoli od zadnjih treh elementov vrednost nič, funkcija ni izračunala rešitve. Matrika B predstavlja matriko odnosov med kriteriji.

```
fprim_vektor <- function(matrika)
## funkcija fprim_vektor
## vhodni parameter 1: matrika odnosov
```

```

### izhodni parameter : vektor, ki vsebuje
###                               normalizirani vektor,
###                               +1=stolpec v seznamu lastnih vektorjev
###                               +2=IC,
###                               +3=CR
### RL(27.1.2022):odprava napake pri matrikah z dimenzijo > 3
### RL(30.1.2022):odprava napake pri matrikah z dimenzijo 2
### RL(30.1.2022):za matriko z dimenzijo > 15 je konstanta 1.59
###
{
  konst <- c(0.00, 0.00, 0.58, 0.90, 1.12, 1.24, 1.32, 1.41,
            1.45, 1.49, 1.51, 1.48, 1.56, 1.57, 1.59)
  toleranca <- 1.0e-12
  dm <- nrow(matrika)
  lvlv <- eigen(matrika)
  vektorji <- lvlv$vectors
  vrednosti <- as.matrix(lvlv$values)
  rezultat <- replicate((dm + 3), 0)
  for(i in 1:ncol(vektorji)) {
    if (Re(vektorji[i,1]) > 0) {predznak <- TRUE}
    else {predznak <- FALSE}
    spr_predznak <- FALSE
    spr_imaginar <- FALSE
    for(j in 1:nrow(vektorji)) {
      if (Re(vektorji[j,i]) >= 0) {predznakn <- TRUE}
      else {predznakn <- FALSE}
      if (xor(predznak,predznakn)) {spr_predznak <- TRUE}
      if (Im(vektorji[j,i]) != 0) {spr_imaginar <- TRUE}
    }
    if (!spr_predznak & !spr_imaginar) {rezultat[dm + 1] <- i}
  }
  if (rezultat[dm + 1] > 0) {
    vsota <- sum(vektorji[,rezultat[dm + 1]])
    normalizirani_vektor <- Re(vektorji[,rezultat[dm + 1]]/vsota)
    for(k in 1:dm) { rezultat[k] <- normalizirani_vektor[k]}
    rezultat[dm + 2] <- (max(Re(vrednosti)) - dm)/(dm - 1)
    if(dm > 15) {faktor <- konst[15]} else {faktor <- konst[dm]}
    if(dm > 2) {rezultat[dm + 3] <- rezultat[dm + 2]/faktor}
    else {rezultat[dm + 3] <- rezultat[dm + 2]}
    if(rezultat[dm + 2] < toleranca) {rezultat[dm + 2] <- 0}
    if(rezultat[dm + 3] < toleranca) {rezultat[dm + 3] <- 0}
    print(paste("*IC: ", rezultat[dm + 2], "*CR:",
               rezultat[dm + 3]))
  }
  else {print(paste("*IC in CR ni izračunan."))}
  return (rezultat)
}

B <-matrix(c(1,3,9,5,1/3,1,1,1,1/9,1,1,1/3,1/5,1,3,1),

```

```

      nrow = 4, ncol = 4)
komplet <- fprim_vektor(B)

## [1] "*IC: 0.062448878528329 *CR: 0.0693876428092545"

if (komplet[nrow(B)+1] > 0) {
  print("Normalizirani vektor")
  for(i in 1:nrow(B)) {print(komplet[i])}
  print(paste("IC: ", komplet[nrow(B)+2], "CR:",
             komplet[nrow(B)+3]))
}

## [1] "Normalizirani vektor"
## [1] 0.05759632
## [1] 0.2663244
## [1] 0.4546261
## [1] 0.2214532
## [1] "IC: 0.062448878528329 CR: 0.0693876428092545"

```

Za izračun funkcije koristimo še dosežke alternativ za 4 lastnosti. Če bi imeli tri alternative, bi tvorili 4 matrice odnosov, dimenzija vsake od teh pa bi bila 3×3 . Za vsako od teh matrik bi izračunali lastne vektorje, lastne vrednosti, indeks konsistentnosti in razmerje konsistentnosti. Če bi bile matrice primerno konsistentne, bi izračunali še 3 normalizirane vektorje, ki podajajo dosežke posameznih alternativ. Vsak od teh normaliziranih vektorjev bi imel 4 elemente. Zaradi obsega te razlage predpostavimo, da smo dosežke alternativ izračunali in smo jih zbrali v tabeli 6, v programski kodi pa smo to matriko imenovali C . Funkcija koristi je vsota zmnožkov uteži in posameznih dosež-

Tabela 6: Dosežki treh alternativ pri štirih kriterijih.

	k1	k2	k3	k4
A-1	0.22	0.7	0.33	0.12
A-2	0.38	0.2	0.50	0.35
A-3	0.40	0.1	0.27	0.53

kov alternativ. V matriki C to torej dosežki treh alternativ (vrstice) pri štirih lastnostih (stolpci). Funkcijo koristi za posamezne alternative smo označili s Q_1, Q_2, Q_3 . Programska koda prikazuje tudi izpis vrednosti funkcije koristi pri posamezni alternativni in na koncu še sporočilo o najboljši alternativni.

```

C <-matrix(c(0.22, 0.38, 0.40, 0.7, 0.2, 0.1, 0.33, 0.50, 0.27,
            0.12, 0.35, 0.55), nrow = 3, ncol = 4)
s <- ncol(C)
ime_izbire <-c('Izbira 1','Izbira 2','Izbira 3')
ime_kriterija <-c('Lastnost 1','Lastnost 2','Lastnost 3','Lastnost 4')
Q1 <- sum(komplet[1:4]*t(C[1,1:s]))
Q2 <- sum(komplet[1:4]*t(C[2,1:s]))
Q3 <- sum(komplet[1:4]*t(C[3,1:s]))
Q <- cbind(Q1,Q2,Q3)

```

```

for(i in 1:ncol(Q)) {
  print(paste("Alternativa", i, " funkcija koristi Q (",Q[i],")"))
}

## [1] "Alternativa 1 funkcija koristi Q ( 0.375699269589175 )"
## [1] "Alternativa 2 funkcija koristi Q ( 0.379973138479101 )"
## [1] "Alternativa 3 funkcija koristi Q ( 0.294219263090465 )"

print(paste("Alternativa", which.max(Q)))

## [1] "Alternativa 2"

print(paste0("ima največjo vrednost funkcije koristi (",max(Q),")"))

## [1] "ima največjo vrednost funkcije koristi (0.379973138479101)"

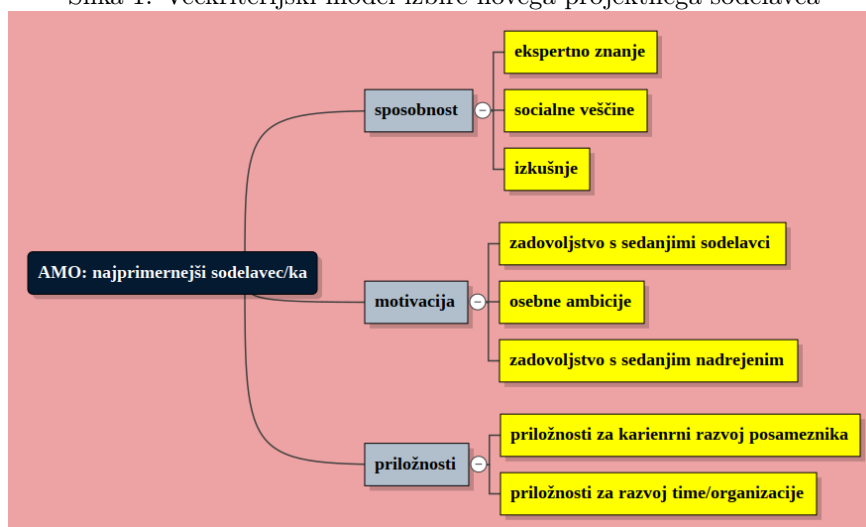
```

Izpis kaže, da je največjo vrednost dosegla alternativa 2. Ima pa le majhno prednost pred alternativo 1. Ali je ta razlika res pomembna in odločujoča določa kontekst odločitvenega problema in seveda človek - odločevalec.

7 Primer 3 v R - večkriterijska izbira novega projektnega sodelavca

Primer prikazuje reševanje večkriterijskega problema izbire najprimernejšega projektnega sodelavca ali sodelavke. Vodja tima uporabi večkriterijski model AMO (Ability, Motivation, Opportunity), ki je prikazan na sliki 1. Model ima tri agregatne kriterije, vsak od njih pa še dva ali tri elementarne kriterije. Skupno je osem elementarnih kriterijev. Prijavili so se štiri kandidati z imeni: Ana, Borut, Cene in Darko. Vodja tima kandidate medsebojno primerja pri vseh osmih kriterijih.

Slika 1: Večkriterijski model izbire novega projektnega sodelavca



Najprej definiramo funkcijo `fprim_vektor` kot v prejšnjem primeru.

```
fprim_vektor <- function(matrika)
  ### funkcija fprim_vektor
  ### vhodni parameter 1: matrika odnosov
  ### izhodni parameter : vektor, ki vsebuje
  ###                               normalizirani vektor,
  ###                               +1=izbrani stolpec v seznamu lastnih vektorjev
  ###                               +2=IC,
  ###                               +3=CR
  ### RL(27.1.2022):odprava napake pri matrikah z dimenzijo > 3
  ### RL(30.1.2022):odprava napake pri matrikah z dimenzijo 2
  ### RL(30.1.2022):za matriko z dimenzijo > 15 je konstanta 1.59
  ###
  {
    konst <- c(0.00, 0.00, 0.58, 0.90, 1.12, 1.24, 1.32, 1.41,
              1.45, 1.49, 1.51, 1.48, 1.56, 1.57, 1.59)
    toleranca <- 1.0e-12
    dm <- nrow(matrika)
    lvlv <- eigen(matrika)
    vektorji <- lvlv$vectors
    vrednosti <- as.matrix(lvlv$values)
    rezultat <- replicate((dm + 3), 0)
    for(i in 1:ncol(vektorji)) {
      if (Re(vektorji[i,1]) > 0) {predznak <- TRUE}
      else {predznak <- FALSE}
      spr_predznak <- FALSE
      spr_imaginar <- FALSE
      for(j in 1:nrow(vektorji)) {
        if (Re(vektorji[j,i]) >= 0) {predznakn <- TRUE}
        else {predznakn <- FALSE}
        if (xor(predznak,predznakn)) {spr_predznak <- TRUE}
        if (Im(vektorji[j,i]) != 0) {spr_imaginar <- TRUE}
      }
      if (!spr_predznak & !spr_imaginar) {rezultat[dm + 1] <- i}
    }
    if (rezultat[dm + 1] > 0) {
      vsota <- sum(vektorji[,rezultat[dm + 1]])
      normalizirani_vektor <- Re(vektorji[,rezultat[dm + 1]]/vsota)
      for(k in 1:dm) { rezultat[k] <- normalizirani_vektor[k]}
      rezultat[dm + 2] <- (max(Re(vrednosti)) - dm)/(dm - 1)
      if(dm > 15) {faktor <- konst[15]} else {faktor <- konst[dm]}
      if(dm > 2) {rezultat[dm + 3] <- rezultat[dm + 2]/faktor}
      else {rezultat[dm + 3] <- rezultat[dm + 2]}
      if(rezultat[dm + 2] < toleranca) {rezultat[dm + 2] <- 0}
      if(rezultat[dm + 3] < toleranca) {rezultat[dm + 3] <- 0}
      print(paste("*IC: ", rezultat[dm + 2], "*CR:",
                  rezultat[dm + 3]))
    }
    else {print(paste("*IC in CR ni izračunan."))}
  }
}
```

```

return (rezultat)
}

```

Pri pripravi podatkov so v tej spominčici uporabljene ocene, ki jih je v okviru izdelave portfelja nalog pri predmetu Metode in tehnike sistemske analize predstavil študent Miha Dolenc. Nekaj matrik je zaradi previsoke nekonsistentnosti ($CR > 0.1$) popravljenih tako, da je zagotovljena konsistentnost. Pri tem je potrebno poudariti, da so popravljene matrike nastale zgolj z logičnim sklepanjem, vendar pa zaradi nepoznavanja situacije morda napačno prikazujejo razmerja. Nekonsistentnost je bila zmanjšana z najmanjšim možnim številom popravkov. Z večjim številom popravkov, ki bi bili kontekstno (vsebinsko) utemeljeni, bi prav tako zmanjšali nekonsistentnost. Obstaja možnost, da bi bili izračunani drugačni lastni vektorji - uteži kriterijev ali ocene dosežkov alternativ pri elementarnih kriterijih. Popravki originalnih matrik študenta Miha Dolenca so komentirani v izvorni kodi.

```

### Priprava dvonivojskega odločitvenega drevesa - model AMO
# 1. nivo kriterijev: najprimernejši sodelavec:
#           sposobnost,motivacija,priložnosti
## Popravl sem originalno matriko MD:
## a1<-matrix(c(1,3,2,1/3,1,2,1/2,1/2,1),
##           nrow=3, ncol=3,byrow=TRUE )
## ker je razmerje CR > 0.1.
a1<-matrix(c(1,3,2,1/3,1,1/1.2,1/2,1.2,1),
           nrow=3, ncol=3,byrow=TRUE )
a1_ime <- c("sposobnost","motivacija","priložnosti")
komplet_a1 <- fprim_vektor(a1)

## [1] "*IC: 0.00276755586924926 *CR: 0.00477164805042976"

print("Normalizirani vektor za 1. nivo kriterijev")

## [1] "Normalizirani vektor za 1. nivo kriterijev"

if (komplet_a1[nrow(a1)+1] > 0) {
  for(i in 1:nrow(a1)) {
    print(paste(a1_ime[i],":=", komplet_a1[i]))}
  print(paste("IC: ", komplet_a1[nrow(a1)+2], "CR:",
             komplet_a1[nrow(a1)+3]))
}

## [1] "sposobnost := 0.548476752572933"
## [1] "motivacija := 0.196942890403194"
## [1] "priložnosti := 0.254580357023873"
## [1] "IC: 0.00276755586924926 CR: 0.00477164805042976"

# 2. nivo kriterijev: sposobnost:
#   ekspertno znanje, socialne veščine, izkušnje
b1<-matrix(c(1,2,2,1/2,1,1,1/2,1,1), nrow=3, ncol=3,byrow=TRUE )
## popravl sem vašo matriko MD:
## b1<-matrix(c(1,2,2,1/2,1,3,1/2,1/3,1),

```

```

## nrow=3, ncol=3,byrow=TRUE )
## ker je razmerje CR > 0.1.
b1_ime <- c("ekspertno znanje", "socialne veščine", "izkušnje")
komplet_b1 <- fprim_vektor(b1)

## [1] "*IC: 0 *CR: 0"

print(
  paste("Normalizirani vektor za 2. nivo kriterijev: ", a1_ime[1]))

## [1] "Normalizirani vektor za 2. nivo kriterijev: sposobnost"

if (komplet_b1[nrow(b1)+1] > 0) {
  for(i in 1:nrow(b1)) {
    print(paste(b1_ime[i], ":", komplet_b1[i]))
    print(paste("IC: ", komplet_b1[nrow(b1)+2], "CR:",
      komplet_b1[nrow(b1)+3]))
  }

## [1] "ekspertno znanje := 0.5"
## [1] "socialne veščine := 0.25"
## [1] "izkušnje := 0.25"
## [1] "IC: 0 CR: 0"

# 2. nivo kriterijev: motivacija:
# zadovoljstvo s sedanjimi sodelavci, osebne ambicije,
# zadovoljstvo s sedanjim nadrejenim)
b2<-matrix(c(1,3,2,1/3,1,1/1.2,1/2,1.2,1),
  nrow=3, ncol=3,byrow=TRUE )
## popravil sem originalno matriko MD:
# b2<-matrix(c(1,3,2,1/3,1,2,1/2,1/2,1), nrow=3, ncol=3,byrow=TRUE)
## ker je razmerje CR > 0.1.
b2_ime <- c("zadovoljstvo s sedanjimi sodelavci",
  "osebne ambicije",
  "zadovoljstvo s sedanjim nadrejenim")
komplet_b2 <- fprim_vektor(b2)

## [1] "*IC: 0.00276755586924926 *CR: 0.00477164805042976"

print(
  paste("Normalizirani vektor za 2. nivo kriterijev: ", a1_ime[2]))

## [1] "Normalizirani vektor za 2. nivo kriterijev: motivacija"

if (komplet_b2[nrow(b2)+1] > 0) {
  for(i in 1:nrow(b2)) {
    print(paste(b2_ime[i], ":", komplet_b2[i]))
    print(paste("IC: ", komplet_b2[nrow(b2)+2], "CR:",
      komplet_b2[nrow(b2)+3]))
  }
}

```

```

## [1] "zadovoljstvo s sedanjimi sodelavci := 0.548476752572933"
## [1] "osebne ambicije := 0.196942890403194"
## [1] "zadovoljstvo s sedanjim nadrejenim := 0.254580357023873"
## [1] "IC: 0.00276755586924926 CR: 0.00477164805042976"

# 2. nivo kriterijev: priložnosti:
# za karierni razvoj posameznika, za razvoj tima/organizacije
b3<-matrix(c(1,2,1/2,1), nrow=2, ncol=2, byrow=TRUE )
b3_ime <- c("za karierni razvoj posameznika",
           "za razvoj tima/organizacije")
komplet_b3 <- fprim_vektor(b3)

## [1] "*IC: 0 *CR: 0"

print(
  paste("Normalizirani vektor za 2. nivo kriterijev: ", a1_ime[3]))

## [1] "Normalizirani vektor za 2. nivo kriterijev: priložnosti"

if (komplet_b3[nrow(b3)+1] > 0) {
  for(i in 1:nrow(b3)) {
    print(paste(b3_ime[i], ":", komplet_b3[i]))
    print(paste("IC: ", komplet_b3[nrow(b3)+2], "CR:",
               komplet_b3[nrow(b3)+3]))
  }
}

## [1] "za karierni razvoj posameznika := 0.666666666666667"
## [1] "za razvoj tima/organizacije := 0.333333333333333"
## [1] "IC: 0 CR: 0"

# zdaj izračunam uteži elementarnih kriterijev
# Uteži agregatnih kriterijev so shranjeni v objektu komplet_a1
# (komplet_a1[1],komplet_a1[2] in komplet_a1[3]).
# Uteži elementarnih kriterijev so shranjeni v objektih
# (komplet_b1,komplet_b2 in komplet_b3). Pri teh treh objektih
# moram upoštevati le prve tri, dve in tri vrednosti. Zato so
# napisane tri for zanke. Lahko pa bi do enake rešitve prišli
# z osmimi neposrednimi prireditvami npr.
# komplet_normaliziranih_utezi[1] <- komplet_a1[1]*komplet_b1[1]
# ...
# komplet_normaliziranih_utezi[8] <- komplet_a1[1]*komplet_b3[2]
#
komplet_normaliziranih_utezi <-numeric(8)
for(i in 1:3) {
  komplet_normaliziranih_utezi[i] <- komplet_a1[1]*komplet_b1[i]}
for(i in 4:6) {
  komplet_normaliziranih_utezi[i] <- komplet_a1[2]*komplet_b2[i-3]}
for(i in 7:8) {
  komplet_normaliziranih_utezi[i] <- komplet_a1[3]*komplet_b3[i-6]}
# preverim ali je njihova vsota 1
sum(komplet_normaliziranih_utezi)

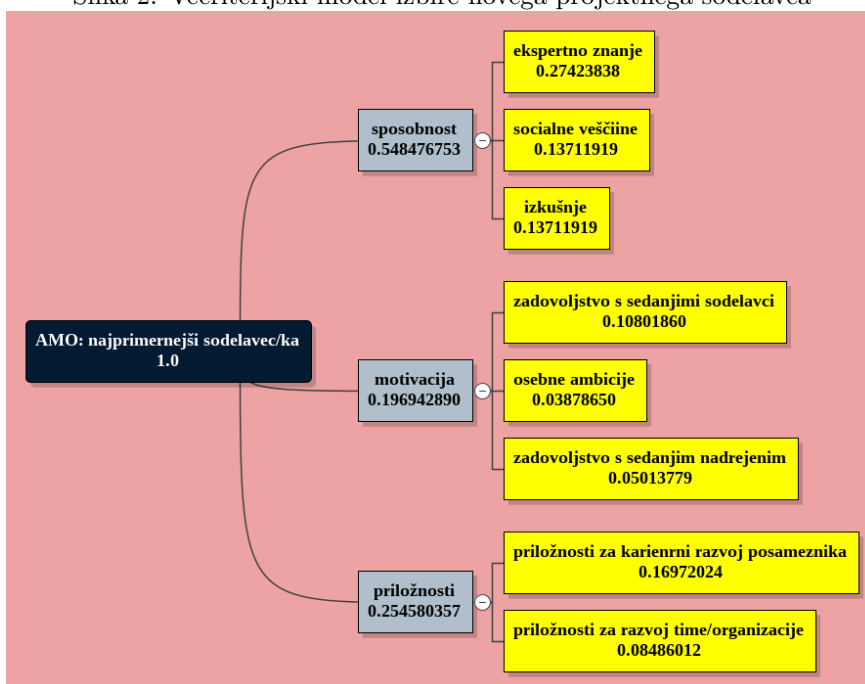
```

```
## [1] 1

# poimenujem še elementarne vektorje
b_ime <-c(b1_ime,b2_ime,b3_ime)
```

Na sliki 2 so prikazane izračunane vrednosti kriterijev, tako agregatnih kot elementarnih. Sledi ocenjevanje kandidatov (Ana, Borut, Cene in Darko) pri osmih elementarnih kriterijih.

Slika 2: Večriterijski model izbire novega projektnega sodelavca



```
#####
## Ocenjevanje kandidatov - 8 elementarnih kriterijev
#####
s_ime<-c("Ana","Borut","Cene","Darko")
s1<-matrix(c(1, 2, 2, 2,
             1/2,1, 1, 1,
             1/2,1 ,1, 1,
             1/2,1, 1, 1), nrow=4, ncol=4,byrow=TRUE )
## popravi sem originalno matriko DM:
## s1<-matrix(c(1,2,2,2,1,1,4,2,1/2,1/4,1,2,1/2,1/2,1/2,1),
##           nrow=3, ncol=3,byrow=TRUE )
## ker je razmerje CR > 0.1.
komplet_s1 <- fprim_vektor(s1)

## [1] "*IC: 0 *CR: 0"
```

```

print(paste("Normalizirani vektor za alternative pri kriteriju: ",
           b_ime[1]))

## [1] "Normalizirani vektor za alternative pri kriteriju:  ekspertno znanje"

if (komplet_s1[nrow(s1)+1] > 0) {
  for(i in 1:nrow(s1)) {
    print(paste(s_ime[i],":=", komplet_s1[i]))}
    print(paste("IC: ", komplet_s1[nrow(s1)+2], "CR:",
               komplet_s1[nrow(s1)+3]))
  }

## [1] "Ana := 0.4"
## [1] "Borut := 0.2"
## [1] "Cene := 0.2"
## [1] "Darko := 0.2"
## [1] "IC: 0 CR: 0"

s2<-matrix(c(1, 1, 3, 2,
            1, 1, 2, 4,
            1/3, 1/2, 1, 2,
            1/2, 1/4, 1/2, 1), nrow=4, ncol=4,byrow=TRUE )
komplet_s2 <- fprim_vektor(s2)

## [1] "*IC: 0.0393140904814591 *CR: 0.0436823227571768"

print(paste("Normalizirani vektor za alternative pri kriteriju: ",
           b_ime[2]))

## [1] "Normalizirani vektor za alternative pri kriteriju:  socialne veščine"

if (komplet_s2[nrow(s2)+1] > 0) {
  for(i in 1:nrow(s2)) {
    print(paste(s_ime[i],":=", komplet_s2[i]))}
    print(paste("IC: ", komplet_s2[nrow(s2)+2], "CR:",
               komplet_s2[nrow(s2)+3]))
  }

## [1] "Ana := 0.352232199017773"
## [1] "Borut := 0.366079887560441"
## [1] "Cene := 0.168783948277308"
## [1] "Darko := 0.112903965144478"
## [1] "IC: 0.0393140904814591 CR: 0.0436823227571768"

s3<-matrix(c(1, 2, 4, 5,
            1/2,1, 2, 2,
            1/4,1/2,1, 3,
            1/5,1/2,1/3,1),
           nrow=4, ncol=4,byrow=TRUE )
komplet_s3 <- fprim_vektor(s3)

## [1] "*IC: 0.0430133469625057 *CR: 0.0477926077361175"

```

```

print(paste("Normalizirani vektor za alternative pri kriteriju: ",
           b_ime[3]))

## [1] "Normalizirani vektor za alternative pri kriteriju: izkušnje"

if (komplet_s3[nrow(s3)+1] > 0) {
  for(i in 1:nrow(s3)) {
    print(paste(s_ime[i],":=", komplet_s3[i]))}
    print(paste("IC: ", komplet_s3[nrow(s3)+2], "CR:",
               komplet_s3[nrow(s3)+3]))
  }

## [1] "Ana := 0.505579917761567"
## [1] "Borut := 0.242078461751061"
## [1] "Cene := 0.16388521939442"
## [1] "Darko := 0.0884564010929514"
## [1] "IC: 0.0430133469625057 CR: 0.0477926077361175"

s4<-matrix(c(1,2,3,2,1/2,1,6,3,1/3,1/6,1,1,1/2,1/3,1,1),
           nrow=4, ncol=4,byrow=TRUE )
komplet_s4 <- fprim_vektor(s4)

## [1] "*IC: 0.0808855187855748 *CR: 0.0898727986506387"

print(paste("Normalizirani vektor za alternative pri kriteriju: ",
           b_ime[4]))

## [1] "Normalizirani vektor za alternative pri kriteriju: zadovoljstvo s sedanjimi sodel

if (komplet_s4[nrow(s4)+1] > 0) {
  for(i in 1:nrow(s4)) {
    print(paste(s_ime[i],":=", komplet_s4[i]))}
    print(paste("IC: ", komplet_s4[nrow(s4)+2], "CR:",
               komplet_s4[nrow(s4)+3]))
  }

## [1] "Ana := 0.4006354090904"
## [1] "Borut := 0.368367700360149"
## [1] "Cene := 0.100393854176444"
## [1] "Darko := 0.130603036373007"
## [1] "IC: 0.0808855187855748 CR: 0.0898727986506387"

s5<-matrix(c(1,1,1,4,1,1,2,2,1,1/2,1,2,1/4,1/2,1/2,1),
           nrow=4, ncol=4,byrow=TRUE )
komplet_s5 <- fprim_vektor(s5)

## [1] "*IC: 0.0404401145198801 *CR: 0.0449334605776445"

print(paste("Normalizirani vektor za alternative pri kriteriju: ",
           b_ime[5]))

## [1] "Normalizirani vektor za alternative pri kriteriju: osebne ambicije"

```

```

if (komplet_s5[nrow(s5)+1] > 0) {
  for(i in 1:nrow(s5)) {
    print(paste(s_ime[i],":=", komplet_s5[i]))}
    print(paste("IC: ", komplet_s5[nrow(s5)+2], "CR:",
      komplet_s5[nrow(s5)+3]))
  }

## [1] "Ana := 0.326726896772236"
## [1] "Borut := 0.326726896772236"
## [1] "Cene := 0.231030804303685"
## [1] "Darko := 0.115515402151843"
## [1] "IC: 0.0404401145198801 CR: 0.0449334605776445"

s6<-matrix(c(1,2,1,3,1/2,1,1,1,1,1,1,3,1/3,1,1/3,1),
  nrow=4, ncol=4,byrow=TRUE )
komplet_s6 <- fprim_vektor(s6)

## [1] "*IC: 0.0393140904814583 *CR: 0.0436823227571758"

print(paste("Normalizirani vektor za alternative pri kriteriju: ",
  b_ime[6]))

## [1] "Normalizirani vektor za alternative pri kriteriju: zadovoljstvo s sedanjim nadrej

if (komplet_s6[nrow(s6)+1] > 0) {
  for(i in 1:nrow(s6)) {
    print(paste(s_ime[i],":=", komplet_s6[i]))}
    print(paste("IC: ", komplet_s6[nrow(s6)+2], "CR:",
      komplet_s6[nrow(s6)+3]))
  }

## [1] "Ana := 0.356912028328312"
## [1] "Borut := 0.199503521827586"
## [1] "Cene := 0.30846464643784"
## [1] "Darko := 0.135119803406262"
## [1] "IC: 0.0393140904814583 CR: 0.0436823227571758"

s7<-matrix(c(1,2,4,2,1/2,1,2,1,1/4,1/2,1,1,1/2,1,1,1),
  nrow=4, ncol=4,byrow=TRUE )
komplet_s7 <- fprim_vektor(s7)

## [1] "*IC: 0.0202156758513805 *CR: 0.0224618620570895"

print(paste("Normalizirani vektor za alternative pri kriteriju: ",
  b_ime[7]))

## [1] "Normalizirani vektor za alternative pri kriteriju: za karierni razvoj posameznika

if (komplet_s7[nrow(s7)+1] > 0) {
  for(i in 1:nrow(s7)) {
    print(paste(s_ime[i],":=", komplet_s7[i]))}
    print(paste("IC: ", komplet_s7[nrow(s7)+2], "CR:",
      komplet_s6[nrow(s7)+3]))
  }

```

```

## [1] "Ana := 0.448847800046356"
## [1] "Borut := 0.224423900023178"
## [1] "Cene := 0.135730142564403"
## [1] "Darko := 0.190998157366063"
## [1] "IC: 0.0202156758513805 CR: 0.0436823227571758"

s8<-matrix(c(1,1,1,1,1,1,3,4,1,1/3,1,1,1,1/4,1,1),
           nrow=4, ncol=4,byrow=TRUE )
komplet_s8 <- fprim_vektor(s8)

## [1] "*IC: 0.0690515197732437 *CR: 0.0767239108591597"

print(paste("Normalizirani vektor za alternative pri kriteriju: ",
           b_ime[8]))

## [1] "Normalizirani vektor za alternative pri kriteriju: za razvoj tima/organizacije"

if (komplet_s8[nrow(s8)+1] > 0) {
  for(i in 1:nrow(s8)) {
    print(paste(s_ime[i],":=", komplet_s8[i]))
    print(paste("IC: ", komplet_s8[nrow(s8)+2], "CR:",
               komplet_s8[nrow(s8)+3]))
  }
}

## [1] "Ana := 0.237690340561601"
## [1] "Borut := 0.432596236127018"
## [1] "Cene := 0.169141042767803"
## [1] "Darko := 0.160572380543578"
## [1] "IC: 0.0690515197732437 CR: 0.0767239108591597"

```

V zaključku pripravim matriko, ki vsebuje ocene kandidatov pri osmih elementarnih kriterijih. Ocene kandidatov pri posameznem elementarnem kriteriju so normalizirane.

```

#####
#### pripravim matriko komplet_normaliziranih_ocen - ocene kandidatov
#### za vsak primer inicializiram matriko ocen z nič
komplet_normaliziranih_ocen <-matrix(numeric(32),nrow=4, ncol=8)
#### napolnim matriko komplet_normaliziranih_ocen
#### dobim matriko 4 (kandidati) x 8 (elementarni kriteriji)
komplet_normaliziranih_ocen <-matrix(
  c(komplet_s1[1:4], komplet_s2[1:4],
    komplet_s3[1:4], komplet_s4[1:4],
    komplet_s5[1:4], komplet_s6[1:4],
    komplet_s7[1:4], komplet_s8[1:4]), nrow=4, ncol=8)
#####

```

Izračun kriterijske funkcije je sedaj preprost. Seštejem vsoto produktov uteži in dosežkov posameznega kandidata pri vseh osmih elementarnih kriterijih.

```
#####
#### Izračunam funkcijo koristi Q[i]:
#### množim vektor normaliziranih kriterijev
#### z matriko komplet_normaliziranih_ocen, pri čemer mora biti
#### slednji transponiran, da lahko izvedem matrično operacijo
#### množenja.
Q <-matrix(numeric(4),nrow=1, ncol=4)
colnames(Q) <- s_ime
Q[1] <- sum(komplet_normaliziranih_utezi*t(komplet_normaliziranih_ocen[1,]))
Q[2] <- sum(komplet_normaliziranih_utezi*t(komplet_normaliziranih_ocen[2,]))
Q[3] <- sum(komplet_normaliziranih_utezi*t(komplet_normaliziranih_ocen[3,]))
Q[4] <- sum(komplet_normaliziranih_utezi*t(komplet_normaliziranih_ocen[4,]))
Q

##           Ana      Borut      Cene      Darko
## [1,] 0.3975103 0.2755031 0.1731235 0.1538631
```

Iz izpisa je razvidno, da je najprimernejša kandidatka Ana.

Literatura

- [1] Coyle Geoff. Practical Strategy. Open Access Material. AHP Web access: 30.11.2021 <https://training.fws.gov/courses/references/tutorials/geospatial/CSP7306/Readings/AHP-Technique.pdf>
- [2] Saaty L.Thomas. (1980) The Analytic Hierarchy Process, McGraw Hill International