SVEUČILIŠTE U ZAGREBU
EKONOMSKI FAKULTET ZAGREB

POSLIJEDIPLOMSKI STUDIJ OPERACIJSKA ISTRAŽIVANJA

MODELI VIŠEKRITERIJSKOG ODLUČIVANJA I HEURISTIKE ZA NJIHOVO RJEŠAVANJE

MAGISTARSKI RAD

SILVIJA VLAH

ZAGREB, 2008.

Sadržaj

1 Uvod

Donošenje odluke je problem koji se pojavljuje u svakoj djelatnosti. U kontekstu višekriterijske optimizacije, problem odlučivanja se najčešće promatra kao problem u kojem se donositelj odluke mora opredijeliti za jednu od alternativa uzimajući u obzir sve relevantne faktore, odnosno kriterije. Kako su kriteriji u pravilu konfliktni, odabir donositelja odluke neće biti optimalno rješenje u tradicionalnom smislu, već će riječ biti o zadovoljavajućem rješenju od kojeg u danoj situaciji ne postoji bolje. Vrednovanje mogućih odluka podložno je subjektivnom dojmu donositelja odluke te ovisi i o težinama, odnosno važnostima kriterija.

Jedan od primjera višekriterijskog odlučivanja je i odabir ponuda pristiglih na javni natječaj Hrvatskog fonda za privatizaciju (HFP). Naime, Hrvatski fond za privatizaciju je utemeljen kako bi proveo i dovršio privatizaciju nekadašnjih društvenih poduzeća, u kojima danas državne institucije još imaju dionice i udjele. Kako bi se omogućio transparentno i korektno provođenje privatizacije, razvijene su procedure javnih poziva ili natječaja u kojima se, između ostalog, navode i kriteriji ocjene ponuda.

1.1 Područje i predmet istraživanja

Metode višekriterijske optimizacije koriste se u problemima odlučivanja uz postojanje više kriterija. Poteškoće koje se javljaju pri višekriterijskom odlučivanju proizlaze iz činjenice da su kriteriji odlučivanja u pravilu konfliktni, te konačna odluka predstavlja kompromis između postavljenih kriterija. Pri odlučivanju se mora voditi računa i o važnostima kriterija. Tako se javlja i problem međusobne usporedbe važnosti kriterija.

U odabiru najuspješnije ponude u spomenutom procesu privatizacije, ključni kriterij nije samo ponuđena cijena nego se također uzimaju u obzir i drugi kriteriji kao što su daljnje investicije, te preuzimanje obaveza nad zaposlenicima. Međutim, konačna odluka uvelike ovisi o važnostima tj. težinama pojedinih kriterija. Dakle, da bi se odabrala najuspješnija ponuda u procesu privatizacije, donositelj odluke mora odrediti važnost svakog kriterija.

1.2 Svrha i ciljevi istraživanja

Cilj ovog istraživanja je osmišljavanje heuristika za donošenje odluke uz postojanje više kriterija. Pojavom računala, razvijali su se i različiti sustavi za potporu odlučivanju, no takvi sustavi često zahtijevaju velike količine informacija od donositelja odluke. Tako, neki zahtijevaju usporedbu kriterija po parovima, dok neki čak zahtijevaju konkretne ocjene važnosti kriterija unutar nekog raspona ocjena.

Heuristike koje su kreirane u ovom radu implementirat će se u vidu programskog sustava za potporu odlučivanju koji omogućava jednostavno korištenje što je posljedica manje količine informacija koje se očekuju od donositelja odluke. Heuristika će uz matematičke metode uključivati i logičko razmišljanje donositelja odluke kroz grupiranje mogućih izbora obzirom na sličnosti u razinama zadovoljenja kriterija. Naime, alternative se svrstavaju u grupe obzirom na razinu zadovoljenja kriterija. Ako je alternativa svrstana u najbolju grupu alternativa se smatra uspješnom u zadovoljenju kriterija. Tada će samo nijanse odlučiti o konačnom odabiru alternative.

Pored efikasne heuristike, važan je i način na koji se rezultati prezentiraju donositelju odluke. U tu svrhu potrebno je razviti prikladnu programsku potporu koja će rezultate prikazati u obliku koji mogu interpretirati i donositelji odluka koji nisu stručnjaci u području operacijskih istraživanja. Također, važno je ne opterećivati donositelja odluke prevelikim zahtjevima na ulazne podatke.

Kreirana programska potpora upotrijebit će se za valorizaciju ponuda zaprimljenih na natječaj Hrvatskog fonda za privatizaciju te odabir jedne od ponuđenih ponuda. Pri tome se od donositelja odluke očekuje vrlo jednostavno vrednovanje važnosti kriterija. Naime, potrebno je specificirati koji su kriteriji najvažniji, koji su najmanje važni i koji su negdje između po važnosti.

1.3 Struktura rada

Rad se sastoji od trinaest poglavlja. U prvom, uvodnom poglavlju iznose se osnovne postavke za izradu ovog rada, govori se o području i predmetu istraživanja, te ciljevima koji se žele ovim radom ostvariti.

Drugo poglavlje opisuje problem višekriterijskog odlučivanja, te navodi neke od osnovnih pristupa rješavanju ovakvih problema.

U trećem poglavlju definiraju se osnovni pojmovi višekriterijske optimizacije te metode rješavanja pripadajućih problema, dok se u petom poglavlju opisuju problemi višekriterijske kombinatorijalne optimizacije.

Peto poglavlje navodi nešto složenije,interaktivne metode za rješavanje problema višekriterijske optimizacije, dok se u šestom poglavlju opisuju algoritmi i heuristike koji će biti korišteni kao komponente heuristika koje će u ovome radu biti kreirane.

U sedmom poglavlju opisuje se problem odabira ponuda za privatizaciju zaprimljenih na natječaj HFP, te se u osmom poglavlju formulira odgovarajući matematički model.

Postupak rješavanja problema valorizacije i odabira ponuda za privatizaciju dat je u devetom poglavlju gdje se i detaljno opisuju dvije heuristike koje se predlažu u ovom radu. Također, prikazana je i implementacija te opis funkcionalnosti programskog sustava za potporu odlučivanju koji se temelji na dvije kreirane heuristike.

Deseto poglavlje daje usporedbu sustava za potporu odlučivanju sa programskim paketom ExpertChoice koji se temelji na AHP metodi.

U jedanaestom poglavlju iznose se zaključci, dok se u završnim poglavljima daje popis literature korištene u izradi ovog rada, te pregled priloga ovom radu.

U prilogu su dati i osnovni matematički alati koji su potrebni za istraživanja u okviru ovoga rada.

2 Problem višekriterijskog odlučivanja

Donošenje odluke je problem koji se pojavljuje u svakoj djelatnosti. U kontekstu višekriterijske optimizacije, problem odlučivanja se najčešće promatra kao problem u kojem se donositelj odluke (eng. decision maker – DM) mora opredijeliti za jednu od alternativa koje su poznate ili ih tek treba generirati uzimajući u obzir sve relevantne faktore, odnosno kriterije. Kako su kriteriji u pravilu konfliktni, odabir donositelja odluke neće biti optimalno rješenje u tradicionalnom smislu, već će riječ biti o zadovoljavajućem rješenju od kojeg u danoj situaciji ne postoji bolje. U procesu donošenja odluka pojavljuje se mnoštvo problema kao što su:

- postojanje više kriterija
- kriteriji su obično konfliktni
- vrednovanje mogućih odluka je podložno subjektivnom dojmu donositelja odluke
- donositelj odluke često ne može lako usporediti dvije moguće odluke, odnosno odlučiti koje rješenje zapravo preferira
- skup mogućih odluka kao i važnosti kriterija su rijetko fiksni, te se mogu mijenjati u realnom vremenu

U procesu donošenja odluka moguće je nekoliko pristupa, pa se mogu promatrati sljedeći problemi [13]:

- odabir jedne alternative
  (na primjer, odabir ponuda za privatizaciju, odabir projekta u koji će se investirati)
- svrstavanje alternativa u određene grupe obzirom na kriterije
  (na primjer, predviđanje poslovnog neuspjeha – klasifikacija tvrtki na one koje dobro posluju i one koje ne)
- rangiranje alternativa od najbolje do najlošije
  (na primjer, rangiranje dionica po financijskim pokazateljima poslovanja te burzovnim kretanjima)
- opis alternativa obzirom na razinu zadovoljenja kriterija
  (na primjer, opis financijskih karakteristika skupa poduzeća – bonitet se ocjenjuje kao dobar ako je vrijednost odgovarajućeg kriterija prešla neki zadani prag)

Elementarni čimbenici problema višekriterijskog odlučivanja su:

- identifikacija skupa alternativa
- identifikacija skupa kriterija
- određivanje sustava mjerenja važnosti kriterija

Skup alternativa može biti skup neprekidnih ili diskretnih vrijednosti. Na primjer, kod problema odabira ponuda za privatizaciju riječ je o diskretnom skupu alternativa. Napomenimo da prema svojstvima dopustivih (mogućih) rješenja ili alternativa možemo razlikovati dvije skupine problema. Višeatributna analiza odluke (višeatributno odlučivanje) rješava probleme kod kojih je skup dopustivih rješenja diskretan, unaprijed određen i konačan. Višekriterijska optimizacija rješava probleme u kojima dopustiva rješenja ne moraju biti unaprijed poznata, samim tim im ni broj ne mora biti poznat niti konačan i određena su funkcijama ograničenja. Takvi problemi mogu se zvati i neprekidnima jer se rješenja moraju generirati prije no što ih je moguće vrednovati.

Kriteriji na osnovu kojih se odluka donosi ovise od područja i konkretne situacije, odabrani su od strane stručnjaka iz područja unutar kojeg se donosi odluka, te moraju biti mjerljivi.

Težine, odnosno važnosti kriterija mogu biti zadavane direktno ili indirektno. Kod direktnog pristupa, analitičar ili sustav za potporu odlučivanju dobiva informacije o važnosti kriterija direktno od donositelja odluke, zadavanjem konkretnih vrijednosti težinama, ili usporedbom kriterija po parovima i slično. Indirektan pristup ne zahtijeva nikakvo konkretno ocjenjivanje važnosti kriterija. Umjesto toga, donositelju odluke se nude fiktivni problemi u kojima on odabire rješenja, te se na osnovu njegovih odabira u pojedinim situacijama zaključuje o njegovim preferencijama, odnosno o težinama kriterija.

Iako postoji mnoštvo različitih pristupa rješavanju problema višeatributne analize odluke, ovdje će biti riječi samo o metodama višekriterijske optimizacije obzirom da se one mogu primijeniti kako na problem s unaprijed poznatim i konačnim skupom mogućih rješenja, tako i na probleme u kojima je taj skup potrebno tek generirati.

2.1 Pristupi rješavanju problema višekriterijskog odlučivanja

"Multiple-criteria decision aid" (MCDA) je engleski naziv za znanstveno područje koje se bavi razvitkom metodologije i metoda kojima se donositeljima odluke pomaže pri odlučivanju u kompleksnim situacijama koje podrazumijevaju postojanje više konfliktnih ciljeva, odnosno kriterija. Dok računala nisu bila u tako raširenoj uporabi kao danas, MCDA metodama koristili su se analitičari koji su pomagali u procesu donošenja odluka. U novije doba, računala su omogućila programsku potporu odlučivanju, odnosno sustave za potporu odlučivanju (eng. decision suport system – DSS) u kojima su implementirane ove metode višekriterijske analize.

Analitičar ili DSS sustav daje donositelju odluke mogućnost za odabir rješenja uz pomoć odgovarajućeg algoritma ili metode, a na temelju informacija danih od strane donositelja odluke. Nakon što se prikupe svi relevantni podaci od donositelja odluke, sustav daje prijedlog rješenja. Međutim, nužno je donositelju odluke prepustiti konačni odabir, odnosno dati mu mogućnost intervencije.

Obzirom na trenutak u kojem donositelj odluke može intervenirati postoje tri kategorije metoda [7]:

- a priori metode
  Metode koje omogućavaju donositelju odluke da intervenira prije samog procesa. U njima, proces ne može početi bez da donositelj odluke ne osigura neke potrebne podatke, kao na primjer, težine kriterija.
- a posteriori metode
  To su metode koje omogućavaju donositelju odluke da posreduje nakon pronalaženja rješenja. One pomažu donositelju odluke da iz skupa zadovoljavajućih rješenja odabere ono koje njemu najbolje odgovara.
- interaktivne metode
  To su metode koje omogućavaju donositelju odluke da posreduje tijekom pronalaženja rješenja. U njima je proces iterativan. Svaka iteracija osigurava donositelju odluke neko rješenje koje ne mora nužno biti i najbolje rješenje. Tada donositelj odluke daje potrebnim parametrima neke nove vrijednosti i tako usmjerava proces dalje.

Obzirom na metodologije koje se koriste, metode višekriterijske optimizacije možemo podijeliti na dvije grupe. Postoje egzaktne metode koje će pronaći najkvalitetnije rješenje ako ono postoji. Takve su metode višekriterijskog programiranja. Pitanje „kvalitete“ rješenja bit će razjašnjeno u idućem poglavlju. S druge strane, postoje i metode koje pronalaze približna rješenja (kao što su aproksimacijski algoritmi, heuristike i metaheuristike). Interaktivne metode mogu kombinirati sve navedene pristupe rješavanja problema višekriterijske optimizacije.

3 Osnovni pojmovi višekriterijske optimizacije

3.1 Problem višekriterijske optimizacije

Neka je $S \subseteq \mathbf{R}^n$, $f_{\mathrm{i}} : \mathbf{R}^n \to \mathbf{R}, i = 1,\ldots,k$. S nazivamo skupom mogućih rješenja ukoliko sve točke iz tog skupa zadovoljavaju ograničenja problema. Funkcije $f_i, i = 1,\ldots,k$ nazivamo *kriterijima* ili *funkcijama cilja*. Problem višekriterijske optimizacije odnosi se na rješavanje problema optimiziranja (minimiziranja ili maksimiziranja) dvije ili više funkcija cilja na nekom skupu mogućih rješenja.

Dakle, problem višekriterijske optimizacije (višekriterijskog programiranja) je:

$$\begin{array}{l} \max z_1 = f_1(x) \\ \ldots \\ \max z_k = f_k(x) \\ \quad \text{uz uvjet } x \in S \end{array} \qquad \text{(VP)}$$

Problem se lako može prevesti u problem minimuma:

$$\begin{array}{l} \min(-f_1(x)) \\ \ldots \\ \min(-f_k(x)) \\ \quad \text{uz uvjet } x \in S \end{array}$$

Stoga, radi jasnoće i jednostavnosti postupka pretpostavljamo da sve funkcije cilja maksimiziramo.

Skup $Z = f(S) = \{(f_1(x),\ldots,f_k(x)) \mid x \in S\}$ svih vrijednosti funkcije $f = (f_1(x),\ldots,f_k(x))$ na skupu mogućih rješenja S nazivamo *kriterijskim skupom*, koji je podskup *prostora kriterija* tj. $Z \subseteq \mathbf{R}^k$. Elemente skupa Z nazivamo *kriterijskim vektorima (vektorima funkcija cilja)*, te ih označavamo sa $f(S)$ ili $z = (z_1, z_2,\ldots,z_k)$, gdje su $z_{\mathrm{i}} = f_i(x)$ za svaki $i = 1,\ldots,k$, *kriterijske vrijednosti*.

3.2 Karakterizacija rješenja

Neka je dat problem maksimuma

$$\max_{x \in S} z_i = f_i(x), \qquad (\mathrm{P}_i), \quad i = 1,\ldots,k .$$

Pretpostavljamo da postoji optimalno rješenje x_i^* problema (P_i) za svako $i = 1,\ldots,k$, za koje vrijedi $z_i^* = f_i(x_i^*)$.

Definicija. Neka je x_i^* optimalno rješenje problema $(\mathrm{P_i})$, $i = 1,\ldots,k$. Vektor $z^{id} = \left(z_i^{id}\right) \in \mathbf{R}^k$ takav da vrijedi $z_i^{id} = f_i(x_i^*), i = 1,\ldots,k$ se naziva *idealnom točkom* ili *idealnim kriterijskim vektorom* vektorske funkcije $f(x) = \left(f_1(x), f_2(x),\ldots, f_k(x)\right)$.

Kažemo da problem (VP) ima optimalno rješenje ako je $x_i^* = x^*, \forall i = 1,\ldots,k$. U tom slučaju je $f_i(x^*) \geq f_i(x)$ za svaki $i = 1,\ldots,k$, $x \in S$ ili kraće $f(x^*) \geq f(x)$ za svaki $x \in S$. U svakom drugom slučaju, optimalno rješenje u klasičnom smislu te riječi ne postoji. Stoga ćemo uvesti pojam skupa *efikasnih rješenja* ili *Pareto optimalnih rješenja* koje je 1951. godine uveo T.C.Koopmans [11].

Budući da kriterijski skup nije potpuno uređen skup jer je podskup skupa $\mathbf{R}^k$, on općenito neće imati maksimalni element koji je nužno jedinstven, već može imati više maksimalnih elemenata. Ipak, neke vektore ciljeva, odnosno kriterija možemo izdvojiti za promatranje. To su vektori kojima se nijedna komponenta ne može poboljšati, a da se ne pokvari bar jedna od ostalih komponenti. Tu definiciju objavio je Edgeworth 1881. godine, a ona se obično zove Pareto optimalnost po francusko-talijanskom ekonomistu i sociologu Vilfredu Paretu, koji ju je dalje proširio.

Slijedi formalnija definicija Pareto *efikasnih rješenja* ili *Pareto optimalnih rješenja*.

Definicija. $x \in S$ je *slabo Pareto optimalno rješenje*, zovemo ga još *slabo efikasno rješenje*, ako ne postoji $y \in S$ tako da je $f_i(y) > f_i(x)$ za svako $i = 1,\ldots,k$.

Za točku z iz kriterijskog skupa koja odgovara slabom efikasnom rješenju x tj. $z_i = f_i(x), i = 1,\ldots,k$ kažemo da je *slabo nedominirana* ili *slabo neinferiorna*.

Označimo sa WE (od eng. *weakly efficient*) skup slabih Pareto optimalnih rješenja od (VP). Skup WE definira u dvodimenzionalnom kriterijskom prostoru krivulju koju još zovemo slaba efikasna krivulja.

Definicija. $x \in S$ je *Pareto optimalno rješenje*, zovemo ga još *efikasno rješenje*, ako ne postoji $y \in S$ tako da je $f_i(y) \geq f_i(x)$ za svako $i = 1,\ldots,k$, sa barem jednom strogom nejednakošću tj. da je $f_i(y) > f_i(x)$ za barem jedan *i*.

Pareto optimalno rješenje ili efikasno rješenje naziva se i *nedominirano* ili *neinferiorno*. Znači, za točku $z^{id} = \left(z_i^{id}\right)$ iz kriterijskog skupa koja odgovara efikasnom rješenju x tj. $z_i = f_i(x), i = 1, \ldots, k$ kažemo da je *nedominirana* ili *neinferiorna*.

Reći ćemo da rešenje y dominira nad rješenjem x je $f_i(y) \geq f_i(x)$ za svako $i = 1, \ldots, k$, sa barem jednom strogom nejednakošću tj. da je $f_i(y) > f_i(x)$ za barem jedan i.

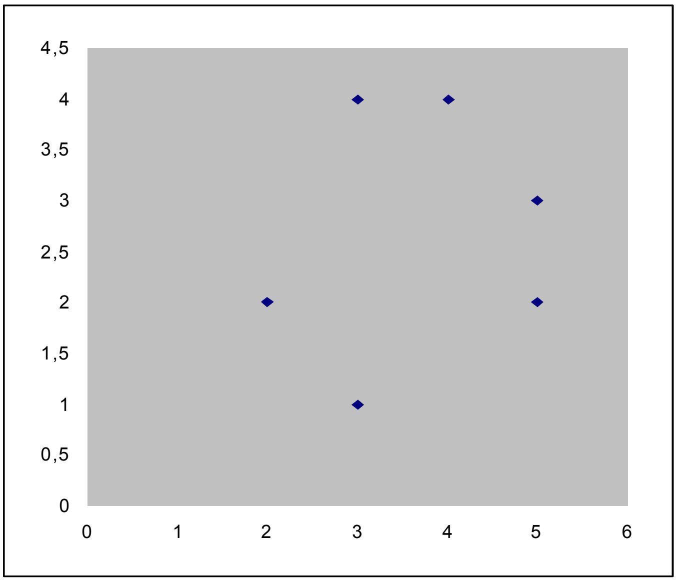
Prema definiciji, neko rješenje x je Pareto optimalno rješenje ako ne postoji neko drugo rješenje y koje je bar po jednom kriteriju bolje od x (u smislu da je vrijednost kriterija u danoj točki veća), a po ostalima nije gore od njega. Dakle, može se reći i da je rješenje x Pareto optimalno ili efikasno rješenje ako ne postoji neko drugo rješenje y koje dominira nad rješenjem x.

Označimo sa E (od eng. *efficient*) skup Pareto optimalnih rješenja od od (VP). Skup E definira u dvodimenzionalnom kriterijskom prostoru krivulju koju zovemo efikasna krivulja.

Za skup slabo efikasnih rješenja WE i skup efikasnih rješenja E vrijedi relacija: E $\subseteq$ WE.

Navedene definicije su valjane samo ako svaki kriterij ima maksimalnu vrijednost.

Promotrimo na sljedećoj slici primjer skupova slabo efikasnih i efikasnih rješenja:



Pretpostavimo da se vrijednosti kriterija žele maksimizirati te da je dan sljedeći kriterijski skup:

Z = {(3,1),(5,2),(5,3),(4,4),(3,4),(2,2)}

Tada je skup slabo efikasnih rješenja:

WE = {(5,2),(5,3),(4,4),(4,3)}

Skup efikasnih rješenja je:

E = {(5,3),(4,4)}

Slika 3.1 Slabo efikasna i efikasna rješenja

U literaturi se uz efikasno rješenje može pronaći i termin strogo efikasno rješenje. Efikasno rješenje $x \in S$ je striktno efikasno rješenje ukoliko ne postoji $y \in S, y \neq x$ za koje vrijedi $f_i(y) = f_i(x)$ za svako $i = 1, \ldots, k$.

Potrebno je definirati i pravu Pareto optimalnost. Ideja pravo Pareto optimalnih rješenja je to što neograničeni kompromisi između kriterija nisu dopušteni. Intuitivno, možemo zamisliti da se Pareto optimalno rješenje s vrlo visokim ili vrlo niskim kompromisima među kriterijima u osnovi se, za donositelja odluke, ne razlikuje od slabo Pareto optimalnog rješenja. Dakle, rješenje je pravo Pareto optimalno ako postoji najmanje jedan par kriterija za koji je konačno smanjenje u jednom kriteriju moguće samo na trošak nekog razumnog povećanja u drugom kriteriju.

Postoji nekoliko definicija prave Pareto optimalnosti. Navedimo jednu od njih:

Definicija. $x \in S$ je *pravo Pareto optimalno rješenje* ili *pravo efikasno rješenje* ako je Pareto optimalno i ako postoji realni broj M > 0, takav da za svaku funkciju $f_i(y), i = 1,\ldots,k$ i svaki $y \in S$ koji zadovoljavaju $f_i(x) > f_i(y)$ postoji barem jedna funkcija f_j, takva da je $f_j(y) > f_j(x)$ i

$$\frac{f_i(x) - f_i(y)}{f_j(y) - f_j(x)} \geq M .$$

Točka z iz kriterijskog skupa je pravo Pareto optimalna ili prava efikasna ako je njezin odgovarajući vektor odluke pravo Pareto optimalan.

Primjer pravih efikasnih rješenja bit će dan kasnije kada se prava efikasnost definira pomoću konveksnog konusa.

Označimo sa PRE (od eng. *proper efficient*) skup pravih efikasnih rješenja.

Vrijedi relacija: PRE $\subseteq$ E.

3.3 Karakterizacija rješenja pomoću konveksnog konusa

Optimalnost u višekriterijskom kontekstu, odnosno efikasnost moguće je definirati pomoću konveksnog konusa kojega ćemo zvati *konus uređaja*.

Definicija. Skup $C \subseteq \mathbf{R}^k$ je *konus* ako $\forall z \in C, \lambda \in \mathbf{R}, \lambda \geq 0 \Rightarrow \lambda z \in C$. Ako je $C \cap (-C) = \{0\}$, gdje je $-C = \{-x \,|\, x \in C\}$, tada kažemo da je C *konus sa vrhom*.

Pomoću konveksnog konusa sa vrhom možemo generirati parcijalni uređaj.

Standardni uređaj se definira pomoću konveksnog konusa sa vrhom $C = \mathbf{R}_+^k = \{(z_1,\ldots,z_k) : z_i \geq 0, i = 1,\ldots,k\}$:

$$z \in \mathbf{R}^k,\ \ z \geq 0 \Leftrightarrow z \in C$$

U daljem tekstu, slovom C bit će označeni gore navedeni konveksni konus tj. $C = \mathbf{R}_+^k = \{(z_1,\ldots,z_k) : z_i \geq 0,\, i = 1,\ldots,k\}$.

Nadalje, za $z, a \in \mathbf{R}^k$, te vrijedi:

$$z \leq 0 \Leftrightarrow z \in -C$$
$$z \geq a \Leftrightarrow z \in a + C$$
$$z \leq a \Leftrightarrow z \in a - C$$

Također, za $z, a \in \mathbf{R}^k$ vrijedi:

$$z > 0 \Leftrightarrow z \in C \setminus \{0\}$$
$$z > a \Leftrightarrow z \in a + C \setminus \{0\}$$
$$z >> 0 \Leftrightarrow z \in \operatorname{int} C, \qquad \text{gdje je } \operatorname{int} C = \{(z_1, z_2, \ldots, z_n) : z_1 > 0, \ldots, z_n > 0\}.$$

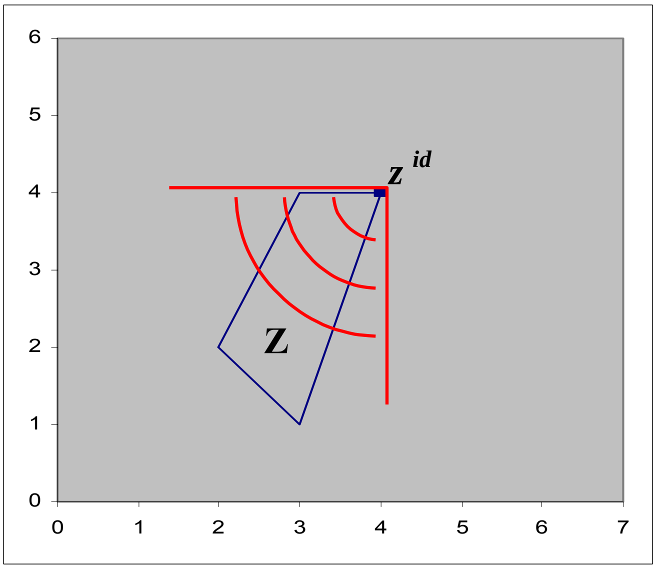
Definicija. $z^* = f(x^*)$ je idealna točka ako je $Z = f(S) \subseteq z^* - C$, odnosno, x^* je idealno rješenje ako je $Z = f(S) \subseteq f(x^*) - C$.

Definicija. $z^* = f(x^*)$ je efikasna točka ako je $(z^* + C) \cap Z = \{z^*\}$, odnosno, x^*je efikasno rješenje ako je $(f(x^*) + C) \cap f(S) = \{f(x^*)\}$. Skup svih efikasnih točaka iz kriterijskog skupa Z, uz uređaj generiran konusom C označava se sa $\mathrm{E}(Z|K)$.

Definicija. $z^* = f(x^*)$ je slaba efikasna točka ako je $z^* \in \mathrm{E}(Z|K)$, gdje je konus $K = \operatorname{int} C \cup \{0\}$, odnosno, x^* je slabo efikasno rješenje ako je $f(x^*) \in \mathrm{E}(f(S)|K)$.

Definicija. $z^* = f(x^*)$ je prava efikasna točka ako postoji konus $K \subset \mathbf{R}^n$ i vrijedi $C \setminus \{0\} \subseteq \operatorname{int} K$ takav da je $z^* \in \mathrm{E}(Z|K)$, odnosno, x^* je pravo efikasno rješenje ako postoji konus $K \subset \mathbf{R}^n$ i vrijedi $C \setminus \{0\} \subseteq \operatorname{int} K$ takav da je $f(x^*) \in \mathrm{E}(Z|K)$.

Idealnu točku identificiranu pomoću konveksnog konusa C može se vidjeti na sljedećoj slici:

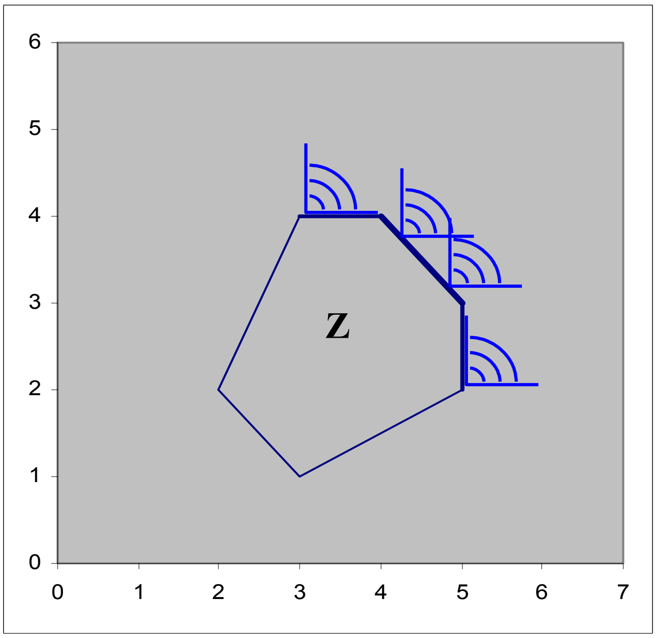


Pretpostavimo da se vrijednosti kriterija žele maksimizirati te da je dan sljedeći kriterijski skup Z (na slici).

z^{id} je idealna točka jer je $Z \subseteq z^{id} - C$.

Slika 3.2 Idealna točka

Na sljedećoj slici identificirana su slabo efikasna i efikasna rješenja pomoću konveksnog konusa *C*:



Pretpostavimo da se vrijednosti kriterija žele maksimizirati te da je dan sljedeći kriterijski skup Z (na slici).

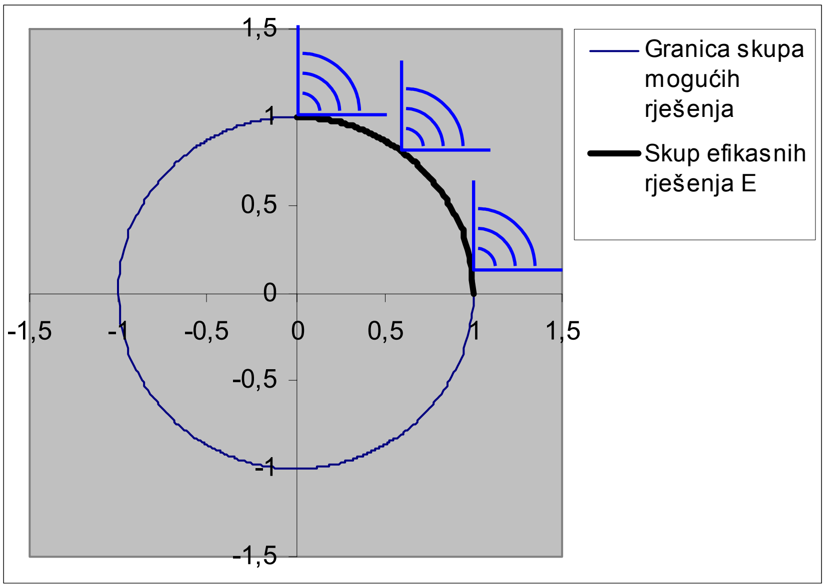
Tada je skup slabo efikasnih rješenja WE čini dužina od točke (3,4) do točke (4,4), te dužina od točke (5,3) do točke (5,2).

Skup efikasnih rješenja je dužina od točke (4,4) do točke (5,3).

Slika 3.3 Slabo efikasna i efikasna rješenja

Naveli smo da vrijedi PRE $\subseteq$ E $\subseteq$ WE. Kod problema linearnog programiranja i problema cjelobrojnog programiranja skup efikasnih rješenja podudara se sa skupom pravih efikasnih rješenja, dakle vrijedi da je E = PRE.

Zato ćemo prava efikasna rješenja ilustrirati na primjeru nelinearne funkcije. Promotrimo sljedeću sliku:

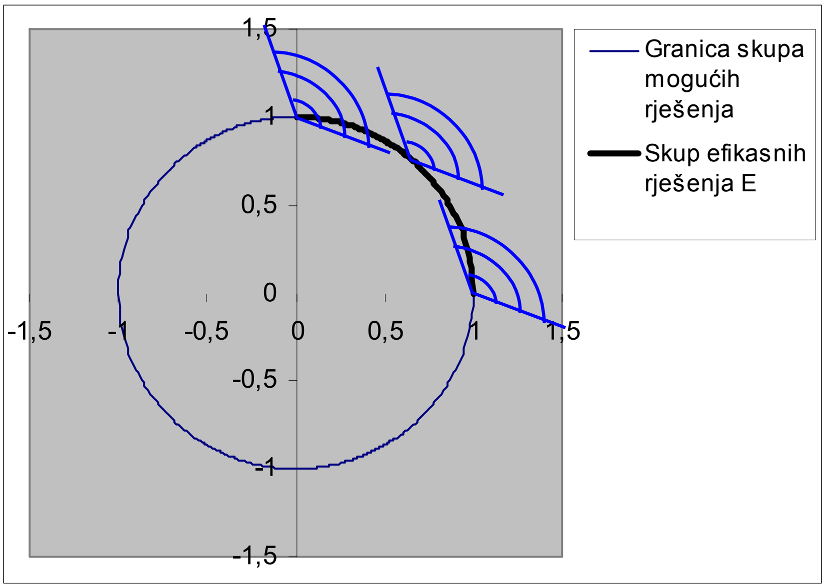


Pretpostavimo da se vrijednosti kriterija žele maksimizirati, a kriterijski je skup dan sa:

$$Z = \{(z_1, z_2) : z_1^2 + z_2^2 \leq 1\}$$

Skup efikasnih rješenja E je luk (dio kružnice) od točke (1,0) do točke (0,1).

Slika 3.4 Efikasna rješenja



Uz pretpostavke kao i na prethodnoj slici, te uzimajući u obzir definiciju pravih efikasnih točaka uz pomoć konveksnog konusa, ovdje prave efikasne točke predstavljaju točke skupa $E \backslash \{(1,0),(0,1)\}$.

Slika 3.5 Efikasna i prava efikasna rješenja

3.4 Skalarna reprezentacija problema višekriterijskog programiranja

Rješenje problema višekriterijskog programiranja moguće je odrediti pomoću različitih metoda za određivanje Pareto optimalnih rješenja. Većina se zapravo svodi na skalarizaciju problema višekriterijske optimizacije tj. na sažimanje svih funkcija cilja iz originalnog problema u jednu funkciju cilja, što se može postići na različite načine.

3.4.1 Težinska suma kriterija

Umjesto originalnog problema višekriterijske optimizacije rješava se problem sa jednom funkcijom cilja koja se dobiva konveksnom kombinacijom, odnosno težinskom sumom funkcija cilja, odnosno kriterija iz originalnog problema.

Dakle,

$$\begin{array}{l} \max z_1 = f_1(x) \\ \dots \\ \max z_k = f_k(x) \\ \quad \text{uz uvjet } x \in S \end{array} \quad \text{(VP)} \qquad \longrightarrow \qquad \begin{array}{c} \max \sum_{i=1}^{k} \lambda_i f_i(x) \\ \text{uz uvjet } x \in S \end{array} \quad (\mathrm{P}_\lambda),$$

gdje parametri $\lambda_i \geq 0$ predstavljaju pondere ili težine kriterija tj. označavaju važnost koja je dodijeljena pojedinom kriteriju tj. funkciji cilja.

Navedimo relevantne teoreme na kojima se temelji ova metoda.

Teorem 3.1. Neka je $x^* \in S$ optimalno rješenje problema (P_λ). Tada vrijedi:

- ako je $\lambda_i > 0, i = 1, \dots, k$, tada je x^* efikasno rješenje problema (VP)
- ako je $\lambda_i \geq 0, i = 1, \dots, k$, tada je x^* slabo efikasno rješenje problema (VP)
- ako je $\lambda_i \geq 0, i = 1, \dots, k$ i x^* je jedinstveno optimalno rješenje problema (P_λ), tada je x^* efikasno rješenje problema (VP)

Dokaz.

- Neka je $x^* \in S$ optimalno rješenje problema (P_λ). Tada postoje $\lambda_i > 0, i = 1, \dots, k$, takvi da je $\sum_{i=1}^{k} \lambda_i f_i(x^*) \geq \sum_{i=1}^{k} \lambda_i f_i(x)$ za svaki $x \in S$.

  Pretpostavimo da x^* nije efikasno rješenje problema (VP). Tada postoji $\hat{x} \in S$ takav da je $f_{\mathrm{i}}(\hat{x}) \geq f_i(x^*)$ za svako $i = 1, \dots, k$, te $f_{\mathrm{j}}(\hat{x}) > f_j(x^*)$ za barem jedno $j \in \{1, \dots, k\}$. Tada vrijedi da je $\lambda_{\mathrm{i}} f_{\mathrm{i}}(\hat{x}) \geq \lambda_{\mathrm{i}} f_i(x^*)$ za svako $i = 1, \dots, k$, te stroga nejednakost $\lambda_{\mathrm{j}} f_{\mathrm{j}}(\hat{x}) > \lambda_{\mathrm{j}} f_j(x^*)$ za barem jedno $j \in \{1, \dots, k\}$. Zbog te stroge nejednakosti i činjenice da je $\lambda_i > 0, i = 1, \dots, k$, vrijedi da je

$\sum_{i=1}^{k} \lambda_i f_i(\hat{x}) > \sum_{i=1}^{k} \lambda_i f_i(x^*)$ što je u kontradikciji sa $\sum_{i=1}^{k} \lambda_i f_i(x^*) \geq \sum_{i=1}^{k} \lambda_i f_i(x)$, čime je dokaz završen. □

- Neka je $x^* \in S$ optimalno rješenje problema (P$_\lambda$). Tada postoje $\lambda_i \geq 0, i = 1,\ldots,k$, takvi da je $\sum_{i=1}^{k} \lambda_i f_i(x^*) \geq \sum_{i=1}^{k} \lambda_i f_i(x)$ za svaki $x \in S$.

  Pretpostavimo da x^* nije slabo efikasno rješenje problema (VP). Tada postoji $\hat{x} \in S$ takav da je $f_i(\hat{x}) > f_i(x^*)$ za svako $i = 1,\ldots,k$. Tada vrijedi da je $\lambda_i f_i(\hat{x}) > \lambda_i f_i(x^*)$ za svako $i = 1,\ldots,k$, te stroga vrijedi i da je $\sum_{i=1}^{k} \lambda_i f_i(\hat{x}) > \sum_{i=1}^{k} \lambda_i f_i(x^*)$ što je u kontradikciji sa $\sum_{i=1}^{k} \lambda_i f_i(x^*) \geq \sum_{i=1}^{k} \lambda_i f_i(x)$, čime je dokaz završen. □

- Neka je $x^* \in S$ jedinstveno optimalno rješenje problema (P$_\lambda$). Tada postoje $\lambda_i \geq 0, i = 1,\ldots,k$, takvi da je $\sum_{i=1}^{k} \lambda_i f_i(x^*) > \sum_{i=1}^{k} \lambda_i f_i(x)$ za svaki $x \in S$.

  Pretpostavimo da x^* nije efikasno rješenje problema (VP). Tada postoji $\hat{x} \in S$ takav da je $f_i(\hat{x}) \geq f_i(x^*)$ za svako $i = 1,\ldots,k$, te $f_j(\hat{x}) > f_j(x^*)$ za barem jedno $j \in \{1,\ldots,k\}$. Usprkos te stroge nejednakosti, zbog mogućnosti da je $\lambda_j = 0$ vrijedi da je $\lambda_i f_i(\hat{x}) \geq \lambda_i f_i(x^*)$ za svako $i = 1,\ldots,k$, te stoga vrijedi i da je $\sum_{i=1}^{k} \lambda_i f_i(\hat{x}) \geq \sum_{i=1}^{k} \lambda_i f_i(x^*)$ što je u kontradikciji sa $\sum_{i=1}^{k} \lambda_i f_i(x^*) > \sum_{i=1}^{k} \lambda_i f_i(x)$, čime je dokaz završen. □

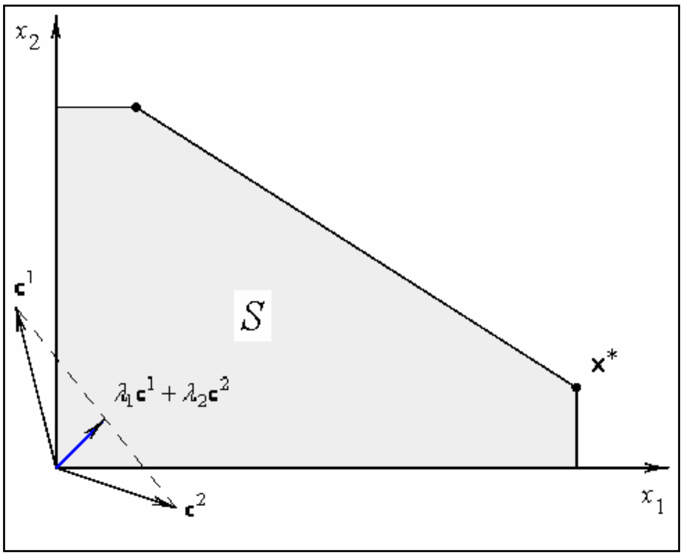
Teorem 3.2. Neka je $x^* \in S$ optimalno rješenje problema (P$_\lambda$), pri čemu je S konveksan skup i $f_i, i = 1,\ldots,k$ konkavne funkcije. Tada vrijedi:

- ako je x^* pravo efikasno rješenje problema (VP), tada postoje $\lambda_i > 0, i = 1,\ldots,k$ takvi da je x^* optimalno rješenje problema (P$_\lambda$)
- ako je x^* slabo efikasno rješenje problema (VP), tada postoje $\lambda_i \geq 0, i = 1,\ldots,k$ takvi da je x^* optimalno rješenje problema (P$_\lambda$)

Dokaz: Vidi u [5].

Dakle, Pareto optimalno rješenje problema višekriterijske optimizacije u kojem je S konveksan skup i $f_i, i=1,\dots,k$ konkavne funkcije može se pronaći rješavanjem navedenog skalarnog problema, te se mijenjanjem težinskih koeficijenata mogu dobiti sva efikasna rješenja.

Promotrimo jedan takav primjer sa dvije linearne funkcije cilja koje se želi maksimizirati:



Slika 3.6 Težinska suma kriterija

Na slici je prikazan smjer rasta funkcija cilja $f_1(x)=\mathbf{c}^1x$, $f_2(x)=\mathbf{c}^2x$, a težine obje funkcije iznose 0.5.

Vidimo optimalno rješenje odgovarajućeg problema (P_{λ}), x^* koje je ujedno i jedno efikasno rješenje problema višekriterijske optimizacije (VP).

Međutim, za probleme u kojima ne vrijedi pretpostavka da je S konveksan skup i $f_i, i=1,\dots,k$ konkavne funkcije, nije uvijek moguće dobiti sva efikasna rješenja jer mogu postojati takozvana *nepodržana efikasna rješenja*, odnosno *nepodržane nedominirane točke*. Nedominirana točka kriterijskog skupa (slika efikasnog rješenja) je nepodržana ako je dominirana nekom konveksnom kombinacijom druge dvije nedominirane točke, odnosno, ako postoji konveksna kombinacija druge dvije nedominirane točke koja je bolja (po svim kriterijima) od navedene nepodržane nedominirane točke.

Problem nepodržanih efikasnih rješenja tj. nedominiranih točaka javlja se tako i u problemima višekriterijskog cjelobrojnog, te mješovitog cjelobrojnog programiranja, no postoje metode kojima se ovaj problem zaobilazi o čemu će više riječi biti kasnije.

Kod samog korištenja ove metode javlja se problem određivanja težina. Tako je u slučaju a priori metoda potrebno definirati način određivanja odgovarajućih težina kriterija koje se koriste u funkciji cilja. U slučaju interaktivnih metoda, moguće je iterativno mijenjati težine prema smjernicama donositelja odluke. U slučaju a posteriori metoda, moramo pratiti parametarsku analizu pomoću spomenutih težina.

3.4.2 Pristup ε-ograničenja

U pristupu ε-ograničenja maksimiziramo jedan kriterij uzimajući u obzir da je preostalih k-1 kriterija ograničeno odozdo.

Dakle,

$$\begin{array}{l} \max z_1 = f_1(x) \\ \dots \\ \max z_k = f_k(x) \\ \quad \text{uz uvjet } x \in S \end{array} \quad \text{(VP)} \quad \longrightarrow \quad \begin{array}{l} \max f_j(x) \\ \text{uz uvjet } f_i(x) \geq \varepsilon_i,\, i \neq j,\, i = 1,\dots,k \quad (\mathrm{P}\varepsilon_j), \\ \qquad x \in S \end{array}$$

Navedimo relevantne teoreme koje karakteriziraju rješenja dobivena ovom metodom.

Teorem 3.3. Neka je $x^* \in S$ optimalno rješenje problema (Pε_j) za neki $j \in \{1,\dots,k\}$. Tada je x^* slabo efikasno rješenje problema (VP).

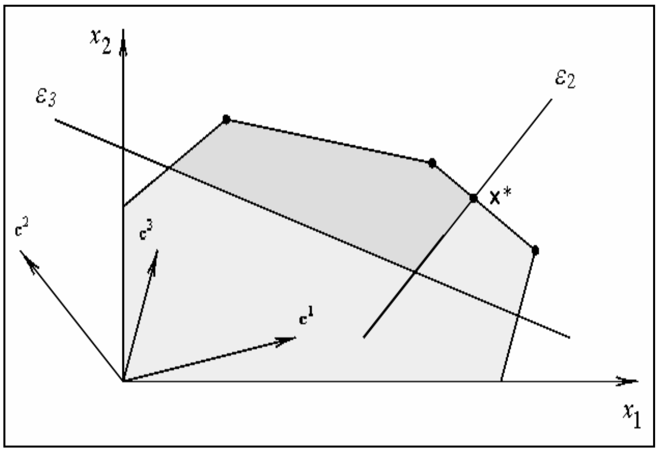
Teorem 3.4. Neka je $x^* \in S$ jedinstveno optimalno rješenje problema (Pε_j) za neki $j \in \{1,\dots,k\}$. Tada je x^* efikasno rješenje problema (VP).

Teorem 3.5. $x^* \in S$ je efikasno rješenje problema (VP) ako i samo ako postoje realni parametri $\varepsilon_1,\dots,\varepsilon_k$ tako da je x^* optimalno rješenje problema (Pε_j) za svaki $j \in \{1,\dots,k\}$.

Dokaze ovdje navedenih teorema može se vidjeti u [5].

Ovaj pristup se često primjenjuje u interaktivnom algoritmu jer donositelj odluke može modificirati ograničenja, te analizirati utjecaj tih modifikacija na konačno rješenje. Tako se pri svakoj iteraciji pronalazi optimalno rješenje za jedan jednokriterijski problem.

Promotrimo ovu metodu na primjeru sa tri linearne funkcije cilja koje se želi maksimizirati:



Slika 3.7 Pristup ε-ograničenja

Funkcija $f_2(x) = \mathbf{c}^2 x$ je ograničena odozdo sa ε_2, funkcija $f_3(x) = \mathbf{c}^3 x$ sa ε_3, a funkciju $f_1(x) = \mathbf{c}^1 x$ se maksimizira.

Vidi se dobiveno optimalno rješenje x^*, tj. efikasno rješenje odgovarajućeg problema (VP).

3.4.3 Kompromisno programiranje

Kompromisno programiranje (metoda globalnog kriterija) bazira se na minimiziranju udaljenosti od referentne točke, obično idealne točke. Također, i ovdje se mogu uključiti i težine kao mjere važnosti kriterija. Za mjerenje udaljenosti između točaka mogu se koristiti različite metrike.

Općenito govoreći, referentna točka ili referentni kriterijski vektor je svaki vektor koji se smatra ciljem koji se treba dostići. Tako metode za rješavanje problema višekriterijske optimizacije koriste i utopijsku točku, te nadir točku.

Definicija. z^{ut} je *utopijska točka* ili *utopijski kriterijski vektor* ako i samo ako z^{ut} dominira nad z^{id}; $z^{ut} \geq z^{id}$ sa barem jednom strogom nejednakošću. Ova točka ne mora odgovarati mogućem rješenju.

Definicija. z^{na} je *nadir točka* ili *nadir kriterijski vektor* ako je $z_i^{na} = \min_{x \in \mathrm{E}} f_i(x), i = 1, \ldots, k$, gdje je E skup efikasnih rješenja problema. Općenito, nadir kriterijski vektor ne mora odgovarati niti jednom mogućem rješenju.

Ukoliko se kao referentna točka koristi idealna točka, općenito se problem (VP) svodi na problem:

$$\min_{x \in S} \left\| z^{id} - f(x) \right\|$$

Za mjerenje udaljenosti može se koristiti L_p metrika:

$$\left\| z^1 - z^2 \right\|_p = \left(\sum_{i=1}^{k} \left| z_i^1 - z_i^2 \right|^p \right)^{\frac{1}{p}}, p \in [1, +\infty\rangle .$$

Dakle,

$$\begin{array}{l} \max z_1 = f_1(x) \\ \ldots \\ \max z_k = f_k(x) \\ \quad \text{uz uvjet } x \in S \end{array} \quad (\mathrm{VP}) \quad \longrightarrow \quad \begin{array}{l} \min \left(\sum_{i=1}^{k} \left| z_i^{id} - f_i(x) \right|^p \right)^{\frac{1}{p}} \quad (\mathrm{PL_p}) \\ \text{uz uvjet } x \in S \end{array}$$

Eksponent $1/p$ se može i izostaviti iz problema ($\mathrm{PL_p}$) jer su takvi problemi ekvivalentni za $1 \leq p < +\infty$ obzirom da je funkcija $x^{1/p} = \sqrt[p]{x}$ rastuća funkcija.

Ako je p = +∞ , metrika se također zove Čebiševljeva metrika, te je takav (PL_p) Čebiševljev problem oblika

$$\min_{x \in S} \max_{i=1,\ldots,k} \left| z_i^{id} - f_i(x) \right| \quad \text{(Pč)}.$$

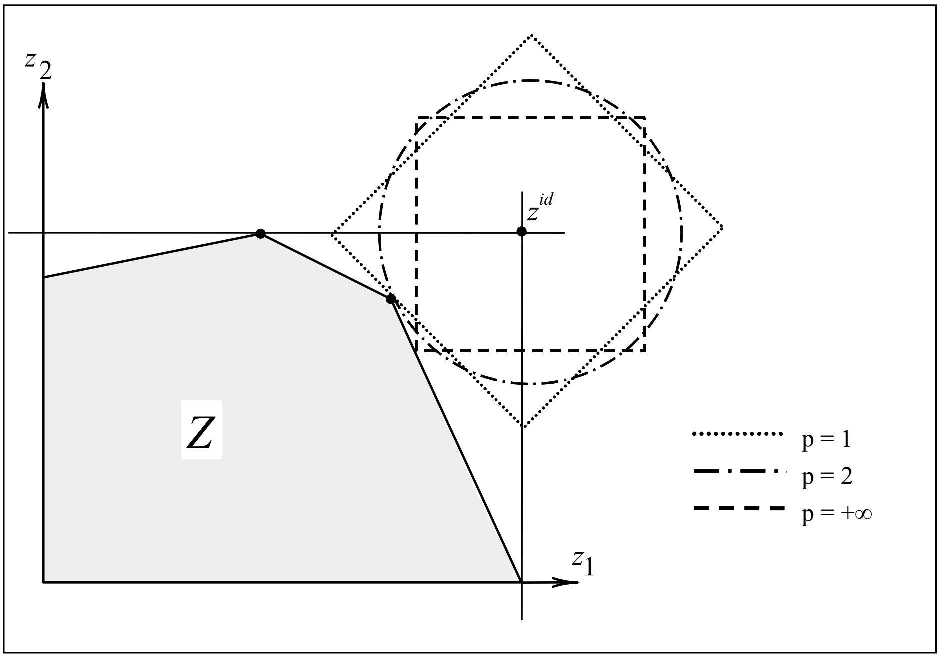
Kako je $z_i^{id} \geq f_i(x)$ za svaki $i = 1,\ldots,k$, gore navedene probleme možemo zapisati kao:

$$\min_{x \in S} \sum_{i=1}^{k} \left(z_i^{id} - f_i(x) \right)^p \quad (\text{PL}_\text{p}),$$

te posebno za p = +∞

$$\min_{x \in S} \max_{i=1,\ldots,k} \left\{ z_i^{id} - f_i(x) \right\} \quad \text{(Pč)}.$$

Na sljedećoj slici ilustrirana je ova metoda uz primjenu najčešćih metrika, p = 1, p = 2, te p = +∞, odnosno Čebiševljevom metrikom.



Slika 3.8 Kompromisno programiranje

Na prethodnoj slici kružnica predstavlja točke koje su jednako udaljene od idealne točke z^{id} ukoliko se kao mjera udaljenosti koristi metrika L_2. Minimiziranje udaljenosti od idealne točke bi u tom slučaju značilo pronalazak kružnice sa centrom u idealnoj točki sa najmanjim polumjerom za koji presjek te kružnice i kriterijskog skupa *Z* nije prazan skup. Slično se primjenjuju i ostale metrike.

Navedimo relevantne teoreme vezane za kompromisno programiranje.

Teorem 3.6. Neka je $x^* \in S$ optimalno rješenje problema (PL$_p$) za $1 \le p < \infty$. Tada je x^* efikasno rješenje problema (VP).

Dokaz. Neka je $x^* \in S$ optimalno rješenje problema (PL$_p$) za $1 \le p < \infty$ i neka x^* nije efikasno rješenje problema (VP). Tada postoji $\hat{x} \in S$ takav da je $f_i(\hat{x}) \ge f_i(x^*)$ za svako $i = 1, \ldots, k$, te $f_j(\hat{x}) > f_j(x^*)$ za barem jedno $j \in \{1, \ldots, k\}$. Stoga je $0 \le z_i^{id} - f_i(\hat{x}) \le z_i^{id} - f_i(x^*)$ za svako $i = 1, \ldots, k$, te $z_j^{id} - f_j(\hat{x}) < z_j^{id} - f_j(x^*)$ za barem jedno $j \in \{1, \ldots, k\}$. Tada vrijedi da je

$$\left(\sum_{i=1}^{k} \left| z_i^{id} - f_i(\hat{x}) \right|^p \right)^{\frac{1}{p}} < \left(\sum_{i=1}^{k} \left| z_i^{id} - f_i(x^*) \right|^p \right)^{\frac{1}{p}}, \; 1 \le p < \infty,$$

što je u kontradikciji sa tvrdnjom da je $x^* \in S$ optimalno rješenje problema (PL$_p$), te je dokaz završen. □

Teorem 3.7. Neka je $x^* \in S$ optimalno rješenje problema (Pč). Tada je x^* slabo efikasno rješenje problema (VP).

Dokaz. Neka je $x^* \in S$ optimalno rješenje problema (Pč) za $1 \le p < \infty$. Tada je $\max_{i=1,\ldots,k} \{ z_i^{id} - f_i(x^*) \} \le \max_{i=1,\ldots,k} \{ z_i^{id} - f_i(x) \}$ za svaki $x \in S$. □

Pretpostavimo da x^* nije slabo efikasno rješenje problema (VP). Tada postoji $\hat{x} \in S$ takav da je $f_i(\hat{x}) > f_i(x^*)$ za svako $i = 1, \ldots, k$. Stoga je $0 \le z_i^{id} - f_i(\hat{x}) < z_i^{id} - f_i(x^*)$ za svako $i = 1, \ldots, k$, pa vrijedi da je $\max_{i=1,\ldots,k} \{ z_i^{id} - f_i(\hat{x}) \} < \max_{i=1,\ldots,k} \{ z_i^{id} - f_i(x^*) \}$ za svako $i = 1, \ldots, k$, što je u kontradikciji sa tvrdnjom da je $x^* \in S$ optimalno rješenje problema (Pč) tj. da je $\max_{i=1,\ldots,k} \{ z_i^{id} - f_i(x^*) \} \le \max_{i=1,\ldots,k} \{ z_i^{id} - f_i(x) \}$ za svaki $x \in S$. □

Teorem 3.8. Neka je $x^* \in S$ jedinstveno optimalno rješenje problema (Pč). Tada je x^* efikasno rješenje problema (VP).

Dokaz. Neka je $x^* \in S$ jedinstveno optimalno rješenje problema (Pč). Tada je $\max_{i=1,\ldots,k} \{ z_i^{id} - f_i(x^*) \} < \max_{i=1,\ldots,k} \{ z_i^{id} - f_i(x) \}$ za svaki $x \in S$.

Pretpostavimo da x^* nije efikasno rješenje problema (VP). Tada postoji $\hat{x} \in S$ takav da je $f_i(\hat{x}) \geq f_i(x^*)$ za svako $i = 1,\ldots,k$, te $f_j(\hat{x}) > f_j(x^*)$ za barem jedno $j \in \{1,\ldots,k\}$. Stoga je $0 \leq z_i^{id} - f_i(\hat{x}) \leq z_i^{id} - f_i(x^*)$ za svako $i = 1,\ldots,k$, te $z_j^{id} - f_j(\hat{x}) < z_j^{id} - f_j(x^*)$ za barem jedno $j \in \{1,\ldots,k\}$. Tada vrijedi da je $\max\{z_i^{id} - f_i(\hat{x})\} \leq \max\{z_i^{id} - f_i(x^*)\}$ za svako $i = 1,\ldots,k$, te $\max\{z_j^{id} - f_j(\hat{x})\} < \max\{z_j^{id} - f_j(x^*)\}$ za barem jedno $j \in \{1,\ldots,k\}$, što je u kontradikciji sa tvrdnjom da je $x^* \in S$ jedinstveno optimalno rješenje problema (Pč) tj. da je $\max_{i=1,\ldots,k}\{z_i^{id} - f_i(x^*)\} < \max_{i=1,\ldots,k}\{z_i^{id} - f_i(x)\}$ za svaki $x \in S$. □

Teorem 3.9. Ako problem (Pč) ima optimalno rješenje, tada je barem jedno od tih optimalnih rješenja efikasno rješenje problema (VP).

Dokaz. Vidi u [5].

3.4.4 Kompromisno programiranje sa težinama

Generalizacija kompromisnog programiranja koristi težine kriterija. Naime, metrika se modificira na način da se udaljenostima pojedinačnih kriterija pridruže težine. Odgovarajuće metrike tada nazivamo težinskim metrikama.

Težinsku metriku definiramo na sljedeći način:

$$\left\|z^1 - z^2\right\|_p^{\lambda} = \left(\sum_{i=1}^{k} \lambda_i \left|z_i^1 - z_i^2\right|^p\right)^{\frac{1}{p}}, p \in [1,\infty\rangle,\ \lambda_i \geq 0, i = 1,\ldots,k .$$

Dakle,

$$\begin{array}{l} \max z_1 = f_1(x) \\ \ldots \\ \max z_k = f_k(x) \\ \quad \text{uz uvjet } x \in S \end{array} \quad (\text{VP}) \quad \longrightarrow \quad \begin{array}{l} \min\left(\sum_{i=1}^{k} \lambda_i \left|z_i^{id} - f_i(x)\right|^p\right)^{\frac{1}{p}} \\ \text{uz uvjet } x \in S \end{array} \quad (\text{P}\lambda\text{L}_\text{p})$$

Kako je $z_i^{id} \geq f_i(x)$ za svako $i = 1,\ldots,k$, gore navedene probleme možemo zapisati kao:

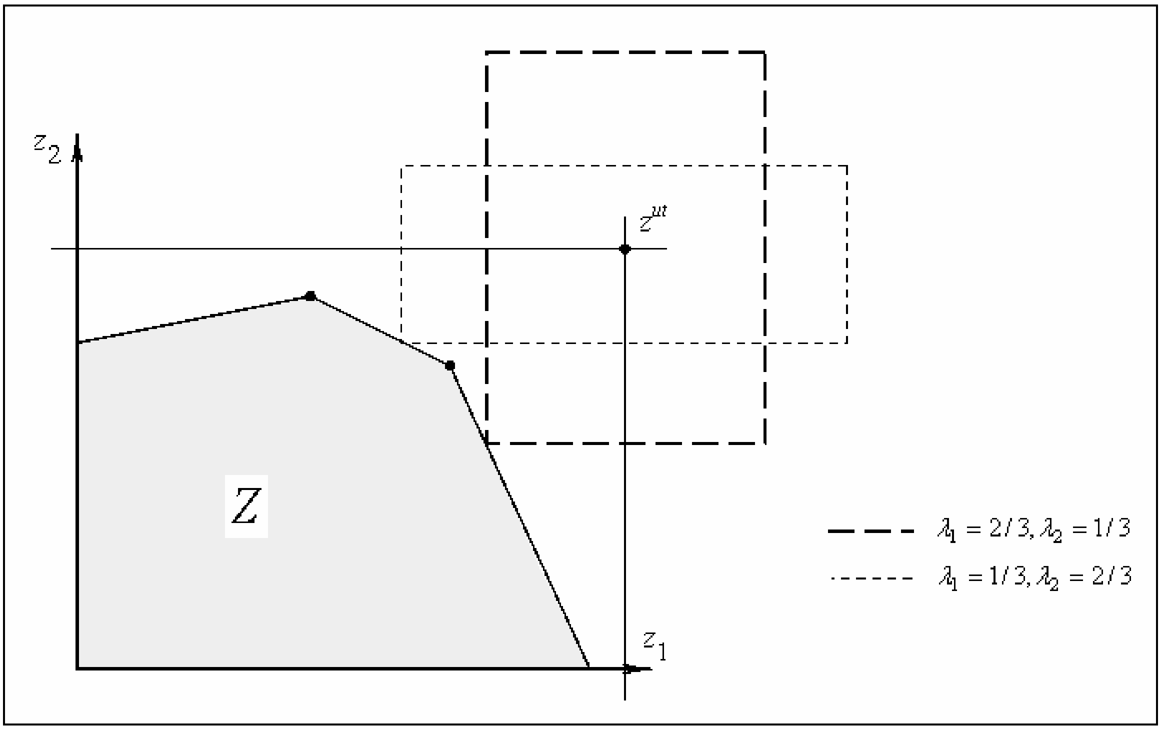
$$\min_{x \in S} \sum_{i=1}^{k} \lambda_i \left(z_i^{id} - f_i(x)\right)^p \quad (\text{P}\lambda\text{L}_\text{p}),$$

te posebno za p = $+\infty$

$$\min_{x\in S} \max_{i=1,\dots,k} \left\{ \lambda_i \left(z_i^{id} - f_i(x) \right) \right\} \quad (\text{P}\lambda\text{č}).$$

Varirajući težine kriterija mogu se dobiti različita efikasna rješenja za $1 \le p < \infty$, te slabo efikasna za $p = +\infty$. $\lambda_1 = 1/3, \lambda_2 = 2/3 \quad \lambda_1 = 2/3, \lambda_2 = 1/3$

Tako na sljedećoj slici možemo vidjeti dva različita rješenja dobivena za različite vrijednosti težina uz Čebiševljevu metriku, dakle $p = +\infty$, te korištenje utopijske točke:



Slika 3.9 Kompromisno programiranje sa težinama

I ovdje vrijede teoremi slični onima kod kompromisnog programiranja bez težina kriterija. Navedimo najrelevantnije.

Teorem 3.10. Neka je $x^* \in S$ optimalno rješenje problema ($\text{P}\lambda\text{L}_p$) za $1 \le p < +\infty$ i $\lambda_i \ge 0$, za svaki $i = 1,2,\dots,k$. Tada je x^* slabo efikasno rješenje problema (VP).

Teorem 3.11. Neka je $x^* \in S$ optimalno rješenje problema ($\text{P}\lambda\text{L}_p$) za $1 \le p < +\infty$. Tada je x^* efikasno rješenje problema (VP) ako vrijedi barem jedna od sljedećih tvrdnji:

- x^* je jedinstveno optimalno rješenje problema ($\text{P}\lambda\text{L}_p$)
- $\lambda_i > 0$, za svaki $i = 1,2,\dots,k$

Teorem 3.12. Neka su $\lambda_i > 0$, za svaki $i = 1,2,\dots,k$. Tada vrijedi:

- Ako je $x^* \in S$ optimalno rješenje problema (Pλč), tada je x^* slabo efikasno rješenje problema (VP).
- Ako je $x^* \in S$ jedinstveno optimalno rješenje problema (Pλč), tada je x^* efikasno rješenje problema (VP).
- Ako problem (Pλč) ima optimalno rješenje, tada je barem jedno od tih optimalnih rješenja efikasno rješenje problema (VP).

Dokaze gore navedenih teorema može se vidjeti u [5].

Teorem 3.13. (Choo i Atkins 1983.) $x^* \in S$ je slabo efikasno rješenje problema (VP) ako i samo ako postoje $\lambda_i > 0,\ \ i = 1,2,\dots,k$ takvi da je x^* optimalno rješenje problema (Pλč) uz upotrebu utopijske točke kao referentne.

Dokaz. „$\Leftarrow$“ Neka je $x^* \in S$ optimalno rješenje problema (Pč) uz upotrebu utopijske točke kao referentne. Tada je $\max\limits_{i=1,\dots,k} \left\{ z_i^{ut} - f_i\left(x^*\right) \right\} \leq \max\limits_{i=1,\dots,k} \left\{ z_i^{ut} - f_i(x) \right\}$ za svaki $x \in S$.

Pretpostavimo da x^* nije slabo efikasno rješenje problema (VP). Tada postoji $\hat{x} \in S$ takav da je $f_i(\hat{x}) > f_i\left(x^*\right)$ za svako $i = 1,\dots,k$, te je $0 \leq z_i^{ut} - f_i(\hat{x}) < z_i^{ut} - f_i\left(x^*\right)$ za svako $i = 1,\dots,k$. Tada vrijedi da je $\max\left\{ z_i^{ut} - f_i(\hat{x}) \right\} < \max\left\{ z_i^{ut} - f_i\left(x^*\right) \right\}$ za svako $i = 1,\dots,k$, što je u kontradikciji sa tvrdnjom da je $x^* \in S$ optimalno rješenje problema (Pč) tj. da je $\max\limits_{i=1,\dots,k} \left\{ z_i^{ut} - f_i\left(x^*\right) \right\} \leq \max\limits_{i=1,\dots,k} \left\{ z_i^{ut} - f_i(x) \right\}$ za svaki $x \in S$.

„$\Rightarrow$“ Neka je x^* slabo efikasno rješenje problema (VP). Definirajmo težine na sljedeći način: $\lambda_i = 1/\left(z_i^{ut} - f_i\left(x^*\right)\right)$. Tada su $\lambda_i > 0,\ \ i = 1,2,\dots,k$.

Pretpostavimo da x^* nije optimalno rješenje problema (Pč). Tada postoji $x \in S$ takav da vrijedi:

$$\max_{i=1,\dots,k} \lambda_i \left(z_i^{ut} - f_i(x)\right) > \max_{i=1,\dots,k} \frac{1}{z_i^{ut} - f_i\left(x^*\right)} \left(z_i^{ut} - f_i\left(x^*\right)\right) = 1,$$

te stoga i $\lambda_i\left(z_i^{ut} - f_i(x)\right) < 1$ za svako $i = 1,\ldots,k$. Dijeljenjem sa λ_i dobivamo $z_i^{ut} - f_i(x) < z_i^{ut} - f_i\left(x^*\right)$ za svako $i = 1,\ldots,k$, te $f_i(x) > f_i\left(x^*\right)$ za svako $i = 1,\ldots,k$, što je u kontradikciji sa tvrdnjom da je x^* slabo efikasno rješenje problema (VP). □

Ovaj pristup se može koristiti u kombinaciji s a priori i a posteriori algoritmima, tako da se najprije definira način određivanja utopijske točke (ukoliko se koristi utopijska točka), a nakon toga se za svaku tako odabranu utopijsku točku prati parametarska analiza težina kako bi se postigla sva Pareto optimalna rješenja.

Utopijsku točku se može postaviti na različite načine. Na primjer, ukoliko je $z^{id} = (z_i^{id}) \in Z^k$ idealna točka, za utopijsku točku može se uzeti točka $z^{ut} = (z_i^{id} + \varepsilon) \in Z^k$, gdje je $\varepsilon \in R, \varepsilon > 0$ proizvoljan pozitivan broj.

Napomena. Čebiševljev problem (Pč) se može zapisati i u alternativnom obliku:

$$\begin{array}{l} \min \max\limits_{i=1,\ldots,k} \left\{z_i^{id} - f_i(x)\right\} \\ \text{uz uvjet } x \in S \end{array} \quad \longrightarrow \quad \begin{array}{l} \min \ \alpha \\ \text{uz uvjet } z_i^{id} - f_i(x) \le \alpha, \ i = 1,\ldots,k \\ \qquad x \in S, \ \alpha \in R \end{array}$$

Analogno, uz korištenje težina:

$$\begin{array}{l} \min \max\limits_{i=1,\ldots,k} \left\{\lambda_i\left(z_i^{id} - f_i(x)\right)\right\} \\ \text{uz uvjet } x \in S \end{array} \quad \longrightarrow \quad \begin{array}{l} \min \ \alpha \\ \text{uz uvjet } \lambda_i\left(z_i^{id} - f_i(x)\right) \le \alpha, \ i = 1,\ldots,k \\ \qquad x \in S, \ \alpha \in R \end{array}$$

Napomena. Umjesto izraza $z_i^{id} - f_i(x)$ u prethodno navedenim problemima mogu se koristiti i izrazi kao što su

$$\frac{z_i^{id} - f_i(x)}{z_i^{id}} \text{ ili } \frac{z_i^{id} - f_i(x)}{z_i^{id} - z_i^{na}},$$

koji su tada normirane vrijednosti. Naime, vrijednosti kriterija je uobičajeno svesti na interval vrijednosti od 0 do 1 da se ne bi dogodilo da utjecaji nekih kriterija budu zanemareni zbog drugih koji imaju veći raspon vrijednosti, o čemu će biti više riječi kasnije.

3.4.5 Uvećana težinska Čebiševljeva metrika

Kako je rješavajući Čebiševljev problem moguće dobiti ne samo efikasna, već i slabo efikasna rješenja, uvest ćemo uvećanu težinsku Čebiševljevu metriku. Njezinim korištenjem izbjegava se navedeni problem dobivanja slabo efikasnih rješenja, te se na taj način dobivaju samo efikasna rješenja.

Uvećanu Čebiševljevu metriku definiramo na sljedeći način:

$$\left\| z^1 - z^2 \right\| = \max_{i=1,\dots,k} \left(\lambda_i \left| z_i^1 - z_i^2 \right| \right) + \rho \sum_{i=1}^{k} \left| z_i^1 - z_i^2 \right|,$$

gdje je $\lambda_i > 0,\ i = 1,\dots,k,$ a $\rho > 0$ mala vrijednost.

Dakle,

$$\begin{array}{l} \max z_1 = f_1(x) \\ \dots \\ \max z_k = f_k(x) \\ \quad \text{uz uvjet } x \in S \end{array} \quad (\mathrm{VP}) \quad \longrightarrow \quad \begin{array}{l} \min \left[\max\limits_{i=1,\dots,k} \left\{ \lambda_i \left| z_i^{id} - f_i(x) \right| \right\} + \rho \sum\limits_{i=1}^{k} \left| z_i^{id} - f_i(x) \right| \right] \\ \text{uz uvjet } x \in S \end{array} \quad (\mathrm{P\lambda č}\rho)$$

Navedimo teorem koji karakterizira rješenja ovih problema:

Teorem 3.14.(Steuer i Choo 1983.) $x^* \in S$ je efikasno rješenje problema (VP) ako i samo ako postoji utopijska točka z^{ut}, te težine $\lambda_i > 0,\ i = 1,\dots,k,$ i mali realni broj $\rho > 0$ tako da je x^* optimalno rješenje problema (Pλčρ).

Dokaz. Vidi u [5].

Ekvivalentno, možemo koristiti i sljedeći problem:

$$\begin{array}{l} \min \max\limits_{i=1,\dots,k} \left\{ \lambda_i \left| z_i^{id} - f_i(x) \right| \right\} - \rho \sum\limits_{i=1}^{k} f_i(x) \\ \text{uz uvjet } x \in S \end{array}.$$

Dobiveni se problem može alternativno zapisati na sljedeći način:

$$\begin{array}{l} \min \max\limits_{i=1,\dots,k} \left\{ \lambda_i \left| z_i^{id} - f_i(x) \right| \right\} - \rho \sum\limits_{i=1}^{k} f_i(x) \\ \text{uz uvjet } x \in S \end{array} \quad \longrightarrow \quad \begin{array}{l} \min \left(\alpha - \rho \sum\limits_{i=1}^{k} f_i(x) \right) \\ \text{uz uvjet } \lambda_i \left(z_i^{id} - f_i(x) \right) \le \alpha,\ i = 1,\dots,k. \\ \qquad x \in S,\ \alpha \ge 0 \end{array}$$

Kao referentnu točku moguće je koristiti i neku drugu točku osim idealne i utopijske, međutim u tom slučaju se treba dozvoliti da varijabla α poprimi i negativne vrijednosti. Tako će se zapravo minimizirati udaljenosti od točki koje su „poželjne“ ili „nedostižne“ i od kojih sigurno ne postoje bolje, kao što su utopijska i idealna točka, dok će se udaljenost od „dostižnih“ točki, odnosno onih od kojih postoje bolje, maksimizirati. Ovakav pristup spada u generalizirano kompromisno programiranje koje će biti pojašnjeno kasnije.

Iako se ovom metodom mogu dobiti sva efikasna rješenja problema višekriterijskog linearnog problema sa neprekidnim tj. realnim varijablama kao i u slučaju problema sa konačnim diskretnim skupom mogućih rješenja, u slučaju problema sa skupom cjelobrojnih vrijednosti kao skupom mogućih rješenja, te nelinearnih problema, moguće je postojanje nepodržanih efikasnih rješenja. Modifikacija ovdje opisane metode daje sva efikasna rješenja višekriterijskog problema optimizacije čak i u navedenim slučajevima beskonačnog broja cjelobrojnih varijabli, te nelinearnih problema. Međutim, najprije je potrebno uvesti metodu leksikografskog uređaja, te metodu leksikografskog uređaja sa ciljevima.

3.4.6 Ciljno programiranje

Naziv ciljno programiranje predložili su Charnes i Cooper, 1961. Svrha ciljnog programiranja je naći moguće rješenje što bliže nekom zadanom cilju. Znači, kao i kod kompromisnog programiranja, traži se točka koja minimizira udaljenost od zadane referentne točke, koja u ovom slučaju ne mora biti niti idealna točka, niti utopijska točka, te se udaljenost definira pomoću L_1 metrike.

Dakle, zadani cilj tj. referentna točka, označimo je sa $g = (g_1, g_2, \ldots, g_k)$, može biti i podbačen i premašen. Označimo sa d_i^+ premašaj, a sa d_i^- podbačaj cilja *i*. Preciznije, za rješenje $x \in S$ definiramo premašaj cilja kriterija *i* sa $d_i^+ = f_i(x) - g_i \geq 0$, a podbačaj sa $d_i^- = g_i - f_i(x) \geq 0$.

Općenito možemo formulirati problem sa:

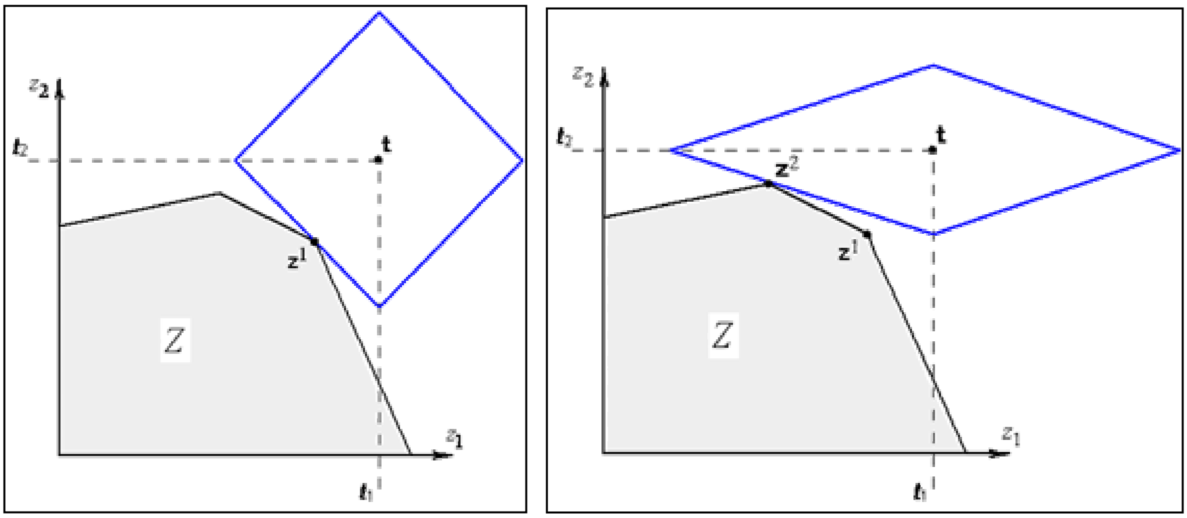
$$\min \sum_{i=1}^{k} \left(d_i^+ + d_i^-\right)$$
$$f_i(x) - d_i^+ + d_i^- = g_i, \quad i = 1, \ldots, k$$
$$x \in S, \quad d_i^+, d_i^- \geq 0, i = 1, \ldots, k$$

Funkcija cilja $\sum_{i=1}^{k}\left(d_i^+ + d_i^-\right)$ je strogo rastuća u varijablama d_i^+ i d_i^-, $i = 1,\dots,k$. Očito je da rješenje problema mora zadovoljavati $d_i^+ \cdot d_i^- = 0,\ i = 1,\dots,k.$

I kod ciljnog programiranja moguće je uvesti težine kriterija, odnosno, u ovom slučaju težine odstupanja od cilja po kriteriju, $\lambda_i > 0, i = 1,\dots,k$. Tako se može promatrati i problem težinskog ciljnog programiranja:

$$
\begin{aligned}
&\min \sum_{i=1}^{k}\left(\lambda_i d_i^+ + \lambda_i d_i^-\right)\\
&f_i(x) - d_i^+ + d_i^- = g_i, \quad i = 1,\dots,k\\
&x \in S, \quad d_i^+, d_i^- \geq 0, i = 1,\dots,k
\end{aligned}
$$

Na sljedećim slikama ilustrirano je ciljno programiranje sa dva kriterija koja se maksimiziraju uz korištenje točke t kao referentne točke. Na prvoj slici koristi se L_1 metrika bez težina, odnosno sa podjednakom važnošću kriterija, dok se na drugoj slici otprilike dvostruka važnost daje drugom kriteriju u odnosu na prvi. Tako se dobivaju i različita efikasna rješenja.



Slika 3.10 Ciljno programiranje

3.4.7 Generalizacija kompromisnog programiranja

Skalarizacija problema višekriterijskog programiranja može se izvršiti i uvođenjem takozvanih funkcija postignuća (eng. achievement function). Funkcijom postignuća naziva se funkciju $s_{\mathrm{R}} : \mathbf{R}^k \to \mathbf{R}$ koja se koristi za definiranje udaljenosti od referentne točke umjesto L_p metrike. Referentna točka može biti i točka koja nije ni idealna, niti utopijska, no udaljenosti od točki koje su „poželjne“ ili „nedostižne“ i od kojih sigurno ne postoje bolje, kao što su utopijska i idealna točka se minimiziraju, dok se udaljenost od „dostižnih“ točki, odnosno onih od kojih postoje bolje, maksimizira.

Tada se problem višekriterijskog programiranja može skalarizirati na sljedeći način:

$$\begin{array}{l} \max z_1 = f_1(x) \\ \ldots \\ \max z_k = f_k(x) \\ \quad \text{uz uvjet } x \in S \end{array} \quad (\text{VP}) \quad \longrightarrow \quad \begin{array}{l} \min \quad s_{\mathrm{R}}(f(x)) \\ \quad \text{uz uvjet } x \in S \end{array} \quad (\text{PR})$$

Definicija. Za funkciju postignuća $s_{\mathrm{R}} : \mathbf{R}^k \to \mathbf{R}$ kažemo da je:

- rastuća ako za svako $y^1, y^2 \in \mathbf{R}^k$ za koje je $y_i^1 \le y_i^2, i = 1,\ldots,k$ vrijedi $s_{\mathrm{R}}(y^1) \le s_{\mathrm{R}}(y^2)$
- striktno rastuća ako za svako $y^1, y^2 \in \mathbf{R}^k$ za koje je $y_i^1 < y_i^2, i = 1,\ldots,k$ vrijedi $s_{\mathrm{R}}(y^1) < s_{\mathrm{R}}(y^2)$
- strogo rastuća ako za svako $y^1, y^2 \in \mathbf{R}^k$ za koje je $y_i^1 \le y_i^2, i = 1,\ldots,k$ i $y_i^1 < y_i^2$ za barem jedan i, vrijedi $s_{\mathrm{R}}(y^1) < s_{\mathrm{R}}(y^2)$

Primjer striktno rastuće funkcije dostignuća uz referentnu točku $z^R = (z_1^R,\ldots,z_k^R)$:

$$s_{\mathrm{R}}(f(x)) = \max_{i=1,\ldots,k} \left\{ \lambda_i \left(z_i^R - f_i(x) \right) \right\}, \quad \lambda_i > 0, i = 1,\ldots,k .$$

Primjeri strogo rastućih funkcija dostignuća su

$$s_{\mathrm{R}}(f(x)) = \max_{i=1,\ldots,k} \left\{ \lambda_i \left(z_i^R - f_i(x) \right) \right\} + \rho_1 \sum_{i=1}^{k} \lambda_i \left(z_i^R - f_i(x) \right),$$

$$s_{\mathrm{R}}(f(x)) = -\left\| z^R - f(x) \right\|^2 + \rho_2 \left\| \left(z^R - f(x) \right)_+ \right\|^2,$$

gdje je $\lambda_i > 0, i = 1,\ldots,k$, $\rho_1 > 0$ mali broj, $\rho_2 > 1$ parametar kazne, a $\left(z^R - f(x) \right)_+$ čije komponente imaju vrijednosti $\max\left\{ 0, \mathrm{z}_{\mathrm{i}}^{\mathrm{R}} - f_i(x) \right\}$.

Teorem 3.15. (Wierbicki 1986)

- Neka je funkcija postignuća s_R rastuća. Ako je $x^* \in S$ jedinstveno optimalno rješenje problema (PR), tada je x^* efikasno rješenje problema (VP).
- Neka je funkcija postignuća s_R striktno rastuća. Ako je $x^* \in S$ optimalno rješenje problema (PR), tada je x^* slabo efikasno rješenje problema (VP).
- Neka je funkcija postignuća s_R strogo rastuća. Ako je $x^* \in S$ optimalno rješenje problema (PR), tada je x^* efikasno rješenje problema (VP).

Dokaz. Vidi u [5].

3.5 Ostale reprezentacije problema višekriterijskog programiranja

3.5.1 Leksikografska metoda

U skup kriterija može se uvesti leksikografski uređaj, odnosno hijerarhija po prioritetima donositelja odluke. Naime, utvrđuje se koji je kriterij najvažniji tj. najvećeg prioriteta, koji slijedi iza njega tj. koji je sljedeći po prioritetu itd. Dakle, najprije se nađu rješenja koja maksimiziraju prvi kriterij po važnosti. Zatim, ukoliko takvih rješenja ima više od jednog, među njima se traže rješenja koja maksimiziraju drugi kriterij po važnosti itd. Na taj se način vrši maksimiziranje skalarnih funkcija u nizu, te se taj postupak naziva još i sekvencijalno maksimiziranje. Umjesto parcijalnog uređenja kriterijskog skupa *Z(S)*, ovdje imamo potpuno uređenje tog skupa pomoću leksikografskog uređaja.

Preciznije, u kriterijski skup uvodimo leksikografski uređaj kako slijedi.

Definicija. Leksikografski uređaj definiramo na sljedeći način:

$$\begin{aligned}
&\forall x, y \in \mathbf{R}^k, \quad x \geq y \quad \Leftrightarrow \\
&x_1 > y_1 \\
&\text{ili} \quad x_1 = y_1 \ \text{i} \ x_2 > y_2 \\
&\cdots \\
&\text{ili} \quad x_1 = y_1, \ldots, x_{k-1} = y_{k-1} \ \text{i} \ x_k > y_k \\
&\text{ili} \quad x_1 = y_1, \ldots, x_{k-1} = y_{k-1}, x_k = y_k .
\end{aligned}$$

U leksikografskoj metodi se kriterijski vektori uspoređuju koristeći leksikografski uređaj. Problem sekvencijalnog maksimiziranja funkcija ciljeva možemo zapisati na sljedeći način:

$$\underset{x\in S}{Lex\max}\{(z_1,\dots,z_k)\} = \underset{x\in S}{Lex\max}\{(f_1(x),\dots,f_k(x))\}.$$

Metoda leksikografskog uređaja može se formalizirati na sljedeći način:

$$S^1 = \left\{x^0 \in S : f_1\left(x^0\right) = \max_{x\in S} f_1(x)\right\} \qquad (1)$$

$$S^2 = \left\{x^0 \in S^1 : f_2\left(x^0\right) = \max_{x\in S^1} f_2(x)\right\} \qquad (2)$$

$$\dots$$

$$S^k = \left\{x^0 \in S^{k-1} : f_k\left(x^0\right) = \max_{x\in S^{k-1}} f_k(x)\right\} \qquad (k)$$

Izračunavanje optimalnog rješenja $x^0 \in S$ problema sekvencijalnog maksimiziranja funkcija ciljeva ekvivalentno je određivanju rješenja $x^0 \in S$ tako da vrijedi (1),(2),…,(*k*).

Teorem 3.16. Ako je $x^* \in S$ rješenje problema sekvencijalnog maksimiziranja, tada je $x^* \in S$ efikasno rješenje problema (VP).

Dokaz. Vidi u [5].

3.5.2 Metoda leksikografskog uređaja sa ciljevima

Leksikografski uređaj se također može upotrijebiti za minimiziranje udaljenosti od idealne ili utopijske točke.

Definicija. Promatramo problem maksimuma *k* kriterija, $z_1,\dots,z_k$, pri čemu je $z_i = f(x), x \in S, i = 1,\dots,k$. Dana je referentna točka $z^{ref} = \left(z_1^{ref},\dots,z_k^{ref}\right)$. Određivanje optimalnog rješenja $x^* \in S$ problema $Lex_{obj}\min(Z) = \underset{x\in S}{Lex\min}\left\|z^{ref} - f(x)\right\|$ ekvivalentno je nalaženju rješenja $x^* \in S$ na sljedeći način:

$$S^1 = \left\{x^0 \in S : \left\|z_1^{ref} - f_1\left(x^0\right)\right\| = \min_{x\in S}\left\|z_1^{ref} - f_1(x)\right\|\right\} \qquad (1)$$

$$S^2 = \left\{x^0 \in S^1 : \left\|z_2^{ref} - f_2\left(x^0\right)\right\| = \min_{x\in S^1}\left\|z_2^{ref} - f_2(x)\right\|\right\} \qquad (2)$$

$$\dots$$

$$S^k = \left\{x^0 \in S^{k-1} : \left\|z_k^{ref} - f_k\left(x^0\right)\right\| = \min_{x\in S^{k-1}}\left\|z_k^{ref} - f_k(x)\right\|\right\} \qquad (k)$$

Dakle, ukoliko se kao referentna točka koristi idealna ili utopijska točka, tada su problemi $Lex \max(Z)$ i $Lex_{obj} \min(Z)$ ekvivalentni.

Također, i ovdje se mogu uključiti i težine tj. ponderi kao mjere važnosti kriterija. Tako se, uz težine kriterija i uporabu Čebiševljeve metrike kao mjere udaljenosti i idealne točke kao referentne točke, dobiva se sljedeći zapis problema:

$$Lex \min_{x \in S} \max_{i=1,\ldots,k} \left\{ \lambda_i \left(z_i^{id} - f_i(x) \right) \right\}.$$

3.5.3 Dvofazna uvećana leksikografska težinska Čebiševljeva metoda

Dvofazna uvećana leksikografska Čebiševljeva metoda koristi uvećanu Čebiševljevu metriku sa nenegativnim težinama za minimiziranje udaljenosti od idealne ili referentne točke koristeći leksikografski uređaj. Dakle, problem se zapisuje kao:

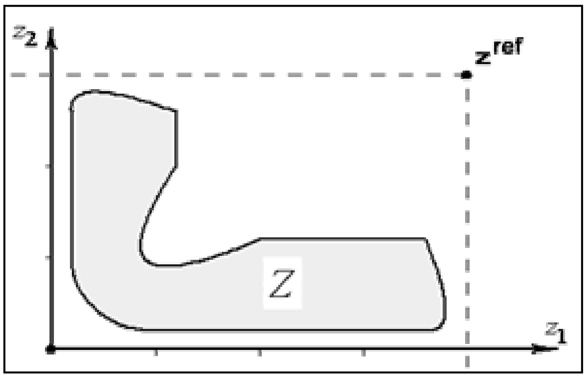
$$\begin{array}{lcl} Lex \min \max\limits_{i=1,\ldots,k} \left\{ \lambda_i \left(z_i^{id} - f_i(x) \right) \right\} - \rho \sum\limits_{i=1}^{k} f_i(x) & & Lex \min \left\{ \alpha, -\sum\limits_{i=1}^{k} f_i(x) \right\} \\ \text{uz uvjet } x \in S & \longrightarrow & \text{uz uvjet } \lambda_i \left(z_i^{id} - f_i(x) \right) \le \alpha,\ i = 1,\ldots,k \quad (\mathrm{P\lambda Lex}) \\ & & x \in S,\ \alpha \ge 0 \end{array}$$

Modifikacija metode sastoji se u razdvajanju pronalaženja rješenja na dvije faze, te se na taj način izbjegava postavljanje parametra ρ.

U prvoj fazi pronalazi se rješenje koje minimizira udaljenost od idealne točke. Moguće je u ovoj fazi dobiti i slabo efikasna rješenja. Ukoliko rješenje nije jedinstveno, u drugoj fazi se na tako dobivenom skupu rješenja minimizira vrijednost $-\sum_{i=1}^{k} f_i(x)$ tj. maksimizira $\sum_{i=1}^{k} f_i(x)$ te se time odbacuju slabo efikasna rješenja.

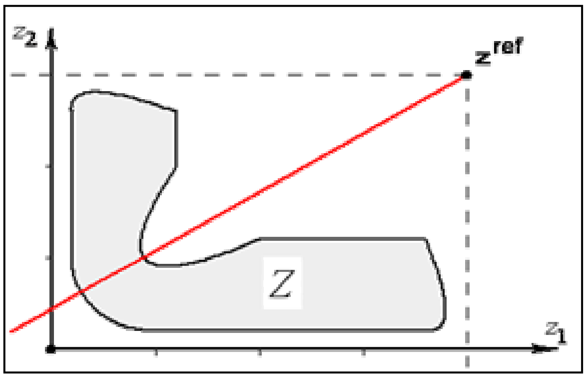
Ovom metodom pronalaze se efikasna rješenja, a moguće je naći i nepodržana rješenja u slučaju cjelobrojnih varijabli ili nelinearne prirode problema.

Ova metoda je predočena na sljedećim slikama:



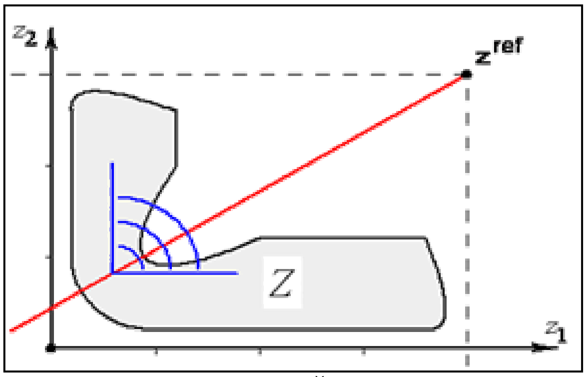
Slika 3.11 Dvofazna Čebiševljeva metoda

Na slici je prikazan kriterijski skup te referentna točka, ovdje konkretno utopijska točka za višekriterijski problem sa dva kriterija – dakle bikriterijski problem. Pretpostavlja se da se oba kriterija treba maksimizirati.



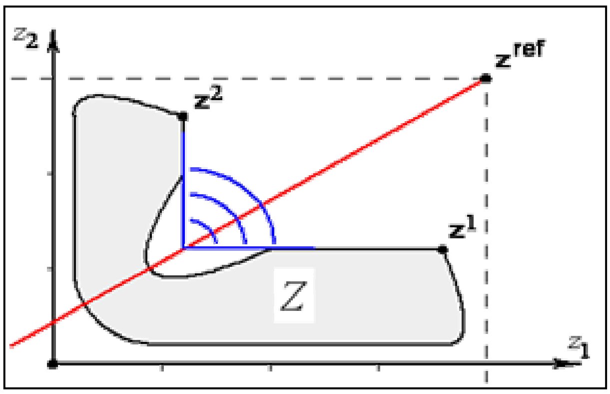
Slika 3.12 Dvofazna Čebiševljeva metoda

Zraka na slici predstavlja smjer kretanja prema referentnoj točki koji je dobiven na osnovu danih težina λ_1 i λ_2. Smjer je dat vektorom $-\left(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}\right)$.



Slika 3.13 Dvofazna Čebiševljeva metoda

Minimiziranje varijable α predstavljeno je konusom koji se približava referentnoj točki, dakle minimizira se udaljenost od referentne točke, u ovom slučaju utopijske točke.



Slika 3.14 Dvofazna Čebiševljeva metoda

Na ovoj slici prikazana su dobivena rješenja tj. efikasne točke. Točke z^1 i z^2 su obje strogo efikasne odnosno strogo nedominirane.

Navedimo teoreme koji karakteriziraju rješenja ovakve reprezentacije problema višekriterijske optimizacije.

Teorem 3.17. (Steuer i Choo) $x^* \in S$ je efikasno rješenje problema (VP) ako i samo ako postoje težine $\lambda_i > 0, i = 1,\ldots,k$ tako da je x^* optimalno rješenje problema (PλLex).

Teorem 3.18. (Steuer) $x^* \in S$ je efikasno rješenje problema (VP) ako i samo ako postoje težine $\lambda_i > 0, i = 1,\ldots,k$ i utopijska točka z^{ut} tako da je x^* optimalno rješenje problema (PλLex) uz korištenje točke z^{ut} kao referentne točke.

Dokaze ovdje navedenih teorema može se vidjeti u [5].

Postoje interaktivne metode koje se temelje na ovoj metodi, no o tome će više riječi biti kasnije.

4 Višekriterijska kombinatorijalna optimizacija

Višekriterijska kombinatorijalna optimizacija bavi se višekriterijskim problemima u kojima je skup mogućih rješenja konačan skup, te problemima u kojima su varijable odlučivanja cjelobrojne.

U okviru ove vrste problema razlikujemo probleme višekriterijskog cjelobrojnog programiranja i probleme višekriterijskog 0-1 programiranja. Kod problema višekriterijskog cjelobrojnog programiranja sve su varijable cjelobrojne te ćemo takve probleme označavati sa (VCP). Varijable problema višekriterijskog 0-1 programiranja mogu poprimati isključivo binarne vrijednosti, te ćemo takve probleme označavati sa (VBP). Možemo spomenuti i postojanje problema višekriterijskog mješovitog cjelobrojnog programiranja u kojima jedan dio varijabli poprima realne vrijednosti, a drugi cjelobrojne.

Ako pretpostavimo da je skup mogućih rješenja ograničen, tada je skup mogućih rješenja problema cjelobrojnog programiranja, pa tako i problema 0-1 programiranja konačan. Stoga se kaže da se višekriterijska cjelobrojna optimizacija bavi višekriterijskim problemima sa konačnim skupom mogućih rješenja. U literaturi se ovi problemi mogu naći pod engleskim nazivom „multiobjective combinatorial optimization“, skraćeno MOCO.

Pokazat će se da se kombinatorijalni problemi višekriterijske optimizacije razlikuju od općeg problema višekriterijske optimizacije, odnosno problema sa realnim varijablama. Naime, kao što je prije spomenuto, kod višekriterijskih problema sa cjelobrojnim varijablama javlja se problem nepodržanih efikasnih rješenja čije je pojavljivanje implicirano nepostojanjem pretpostavke konveksnosti na kriterijskom skupu. U ovom poglavlju, bit će dana definicija nepodržanih rješenja problema višekriterijske kombinatorijalne optimizacije. Također, rješavanje problema višekriterijske kombinatorijalne optimizacije najčešće zahtijeva dulje vrijeme rješavanja, o čemu će također ovdje biti riječi..

Promatrat ćemo višekriterijski cjelobrojni program, dakle program

$$\begin{aligned} &\max z = \left(f_1(x),\ldots,f_k(x)\right) \\ &\text{uz uvjet } g_j(x) \le 0, \quad j = 1,\ldots,m \qquad \text{(VCP)} \\ &\qquad x \in \mathbf{Z}^n \end{aligned}$$

u kojem su funkcije cilja $f_i, i = 1,\ldots,k$ te funkcije ograničenja $g_j, j = 1,\ldots,m$ linearne funkcije, , a varijable su vektori čije komponente mogu poprimiti cjelobrojne vrijednosti.

Također, možemo promatrat i višekriterijski 0-1 program, dakle program

$$\max z = \left(f_1(x), \ldots, f_k(x)\right)$$
$$\text{uz uvjet } g_j(x) \leq 0, \quad j = 1, \ldots, m \qquad \text{(VBP)}$$
$$x \in \{0,1\}^n$$

u kojem su funkcije cilja $f_i, i = 1, \ldots, k$ te funkcije ograničenja $g_j, j = 1, \ldots, m$ također linearne, a varijable su vektori čije komponente mogu poprimiti vrijednosti 0 ili 1.

Skup mogućih rješenja za navedene probleme označit ćemo sa *S*.

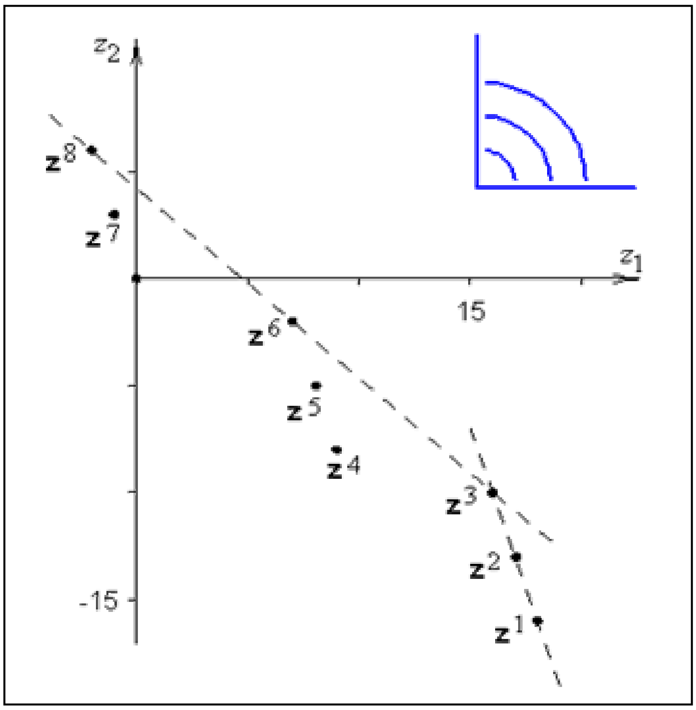
Definirajmo pojam nepodržanih rješenja problema višekriterijske kombinatorijalne optimizacije (VCP).

Definicija. Neka je x^* efikasno rješenje problema (VCP). Ako postoje $\lambda_i > 0, i = 1, \ldots, k$ takvi da je x^* optimalno rješenje problema $\max_{x \in S} \sum_{i=1}^{k} \lambda_i f_i(x)$ tada x^* nazivamo *podržanim efikasnim rješenjem*, a $f\left(x^*\right)$ *podržanom nedominiranom točkom*. Inače x^* nazivamo *nepodržanim efikasnim rješenjem*, a $f\left(x^*\right)$ *nepodržanom nedominiranom točkom*.

Dakle, neka će efikasna rješenja možda biti nemoguće dobiti bez obzira na kombinaciju vrijednosti težina kriterija tj. $\lambda_i > 0, i = 1, \ldots, k$. To su upravo nepodržana efikasna rješenja, sa odgovarajućim nepodržanim nedominiranim točkama.

Po definiciji je nedominirana točka kriterijskog skupa (slika efikasnog rješenja) nepodržana ako je dominirana nekom konveksnom kombinacijom druge dvije nedominirane točke iz kriterijskog skupa.

Na sljedećim slikama mogu se vidjeti nepodržane nedominirane točke u slučaju kada je riječ o diskretnom kriterijskom skupu (varijable su cjelobrojne) koji stoga nije konveksan, te postoji mogućnost nepodržanih rješenja.



Nedominirane točke su:

$$E = \{z^1, z^2, z^3, z^4, z^5, z^6, z^7, z^8\}$$

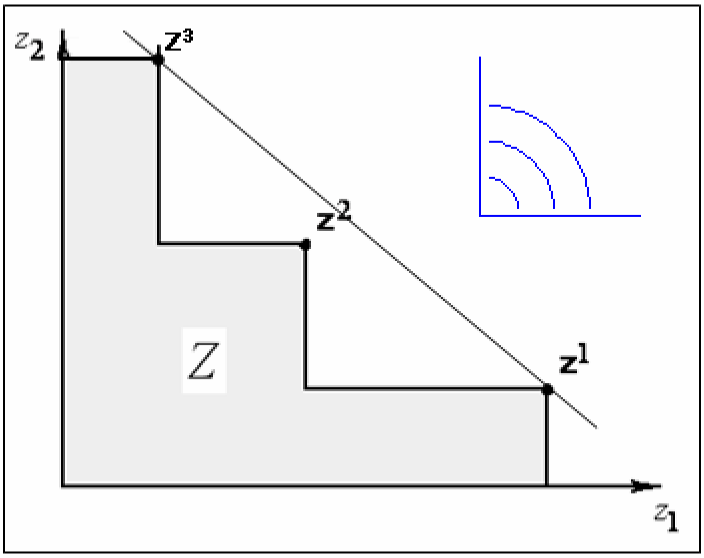
Podržane nedominirane točke su:

$$E^p = \{z^1, z^2, z^3, z^6, z^8\}$$

Nepodržane nedominirane točke su:

$$E^{np} = \{z^4, z^5, z^7\}$$

Slika 4.1 Podržane nedominirane točke



Nedominirane točke su:

$$E = \{z^1, z^2, z^3\}$$

Podržane nedominirane točke su

$$E^p = \{z^1, z^3\}$$

Nepodržane nedominirane točke su

$$E^{np} = \{z^2\}$$

Slika 4.2 Podržane nedominirane točke

Rješavanje višekriterijskih cjelobrojnih problema pomoću metoda koje se svode na rješavanje skalarnih i neskalarnih reprezentacija problema većinom neće rezultirati pronalaskom nepodržanih rješenja. Iznimke su metoda ε-ograničenja, dvofazna leksikografska Čebiševljeva metoda, te kompromisno programiranje.

Osim problema nepodržanih rješenja, postoji i problem povećanja složenosti algoritama za rješavanje problema cjelobrojnog programiranja, pa tako i problema višekriterijskog cjelobrojnog programiranja.

Tako sa metodama koje pronalaze i nepodržana efikasna rješenja postoji problem dugog vremena rješavanja tj. dugog vremena izvršavanja algoritama za izvršavanje uslijed povećane složenosti algoritama.

Na primjer, problem rješavanja višekriterijskog cjelobrojnog programa metodom ε-ograničenja je NP-težak problem.

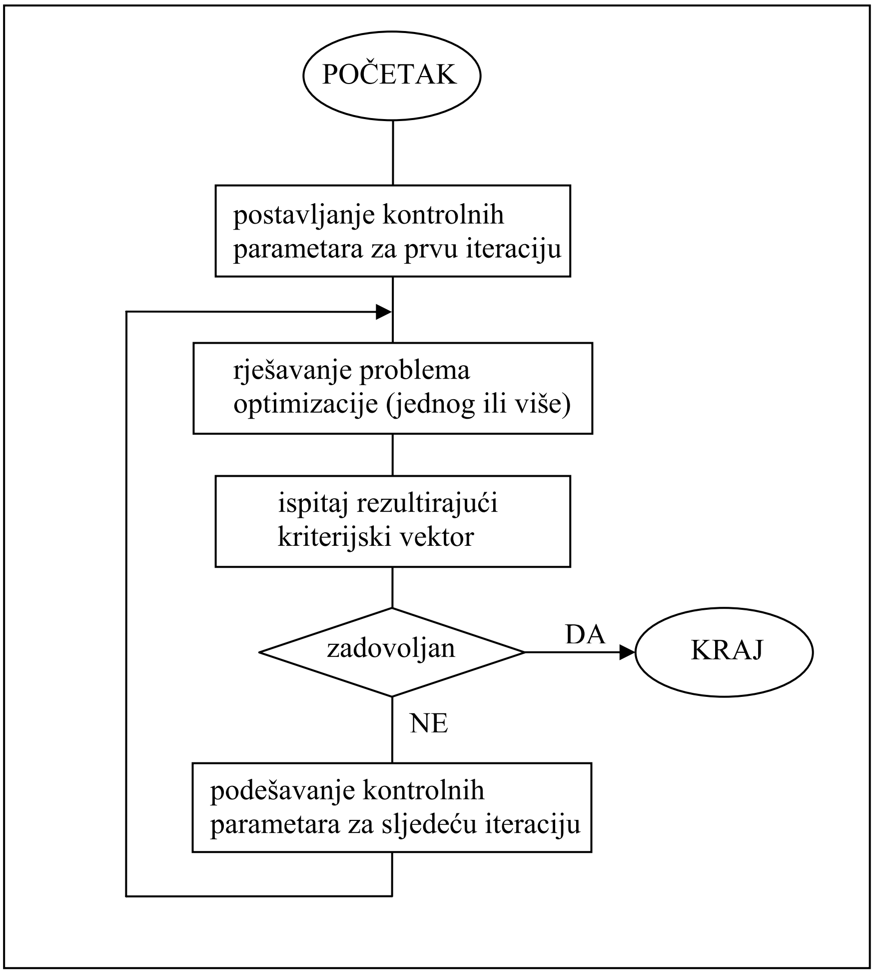
Znači, prije navedene metode imaju velike nedostatke kod cjelobrojnih problema. Većinom ili ne pronalaze nepodržana efikasna rješenja ili imaju jako velika vremena izvršavanja, pa su neupotrebljivi u mnogo situacija.

Vremena izvršavanja mogu smanjiti rješavanjem problema približnim metodama koje ne garantiraju efikasnost rješenja, no može se dokazati da daju „dovoljno dobra“ rješenja u smislu bliskosti efikasnim rješenjima. Također, interaktivne metode mogu skratiti vremena izvršavanja jer se kroz interakciju sa donositeljem odluke veliki dio mogućih rješenja može odbaciti čime se smanjuje skup pretraživanja. O tome će biti više riječi u narednom poglavlju

Iznimka od problema sa navedenim problemima su problemi višekriterijske cjelobrojne optimizacije u kojima je skup mogućih rješenja eksplicitno zadan u smislu da su moguća rješenja eksplicitno navedena umjesto da su predočena ograničenjima. Takav je i problem odlučivanja o izboru ponude za privatizaciju nekog poduzeća u kojem su ponuđači unaprijed zadani i fiksni.

5 Interaktivne metode višekriterijske optimizacije

Opći algoritam interaktivnih metoda prikazan je na sljedećoj slici.



Slika 5.1 Opći interaktivni algoritam

Dakle, interaktivni program je program koji uključuje interakciju algoritma sa donositeljem odluke. Tako se postavljanje kontrolnih parametara za iteracije, kao što je pokazano na slici, može vršiti obzirom na utjecaj donositelja odluke. Na primjer, donositelj može mijenjati težine kriterija ili na neki drugi način utjecati na izvršavanje interaktivnog programa.

Dok se metode koje nisu interaktivne najčešće koriste za generiranje cijelog skupa ili podskupa nedominiranih točki tj. efikasnih rješenja, interaktivne metode karakterizira pronalaženje rješenja u iteracijama u kojima se izmjenjuju izračuni u svrhu pronalaženja rješenja, te intervencije donositelja odluke. Metode koje generiraju skup rješenja se često kombiniraju interaktivnim pristupom na način da generirani skup dalje koristi u interakciji sa donositeljem odluke za pronalazak odgovarajućeg rješenja. Također, moguća je interakcija sa donositeljem odluke tijekom samog procesa generiranja skupa efikasnih rješenja. Prednosti interaktivnog pristupa je manje vrijeme izvršavanja, te omogućavanje većeg utjecaja donositelja odluke na konačnu odluku o efikasnom rješenju kojim će najviše biti zadovoljan.

Postoje i metode kojima uopće nije cilj pronaći cijeli skup efikasnih rješenja. Umjesto toga, te se metode navode od strane donositelja odluke, pa se samo u tom smjeru traže efikasna rješenja. Takve metode u daljem tekstu bit će nazvane navođenim metodama.

Postoje dvije različite metodologije i unutar samog interaktivnog pristupa metodama za višekriterijsku optimizaciju.

Postoje metode kojima je cilj konvergencija prema efikasnom rješenju koje najbolje odgovara preferencijama donositelja odluke, no one su poprilično nefleksibilne te zahtijevaju konzistenciju odluka iz iteracije u iteraciju, dakle ne dozvoljavaju kontradikcije u iskazima donositelja odluke u iteracijama. Tako ovaj način podrazumijeva postojanje stanovite implicitne funkcije korisnosti za donositelja odluke, te se nazivaju metode „implicitne funkcije korisnosti“ [1].

Kao potpuna suprotnost gore opisanom principu, postoje metode u kojima donositelj odluke tijekom iteracija uvijek ima mogućnost opozvati svoju odluku i promijeniti mišljenje glede usporedbe različitih efikasnih rješenja. Znači da metode ne moraju nužno biti konvergentne, u smislu da konvergiraju nekom određenom rješenju. Stoga se može reći da ovakve interaktivne metode zapravo daju konstruktivnu proceduru koja donositelja odluke vodi do najbolje moguće odluke. Zato se ovakve metode nazivaju i metode „otvorene komunikacije“ [1].

5.1 Metode generiranja skupa efikasnih rješenja

Metode generiranja skupa efikasnih rješenja generiraju cijeli skup efikasnih rješenja ili njegov podskup. Često je krajnji cilj dobiti neki reprezentativni podskup efikasnih rješenja, na neki način dobro distribuiranih rješenja, iz kojeg bi donositelj odluke na kraju na različite načine mogao doći do konačne odluke. Ove metode također mogu biti interaktivne i u samom procesu generiranja skupa efikasnih rješenja na način da se donositelju odluke omogućuje nekakva intervencija i u okviru pojedine iteracije unutar postupka generiranja skupa efikasnih rješenja.

Jedna skupina ovih metoda svodi se na konstruktivan proces kojim se u svakom koraku skupu do tada pronađenih efikasnih rješenja dodaju nova. Ovakav pristup ima, na primjer, imaju Klein i Hannan (1982), te Sylva i Crema (2004).

Druga skupina ovakvih metoda radi i sa rješenjima koja nisu efikasna, no ona bivaju odbačena, te na kraju procesa ostaju samo efikasna rješenja. Jedna od takvih metoda je i metoda Deckro i Winkofsky (1983), te Kiziltan i Yucaoglu (1983).

Međutim, ovakve metode ranijeg datuma kao što su gore navedene, uglavnom se ne nose dobro sa problemima velikih dimenzija, posebice onima sa cjelobrojnim varijablama. Naime, vremena izvršavanja za takve probleme su takva da se metode teško u praksi mogu koristiti u takvim situacijama.

Od početka 80-tih godina, počelo se raditi na smanjivanju vremena izvršavanja metoda generiranja na način da se donositelj odluke uključi u sam proces generiranja, te metode postaju interaktivne, a sve u cilju smanjenja skupa pretraživanja, te time i smanjivanja vremena izvršavanja.

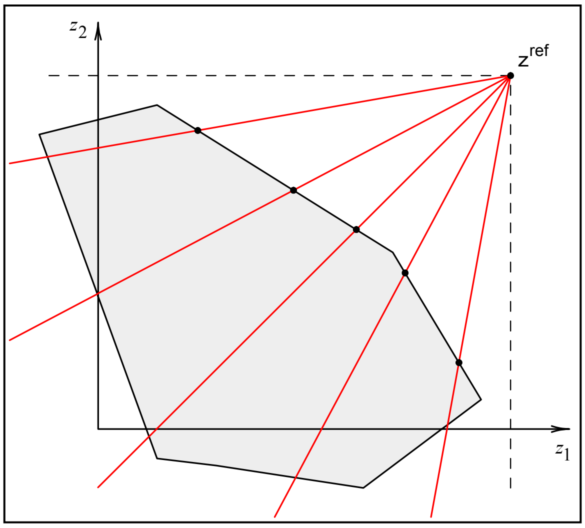
U posljednje vrijeme pojavio se novi pristup u pokušajima smanjenja vremena izvršavanja metoda generiranja koristeći heuristike i metaheuristike. Ove metode ne garantiraju pronalazak optimalnog rješenja u jednokriterijskim problemima, čak niti ocjenu kvalitete dobivenog rješenja, no kroz mnoga testiranja i usporedbama sa rješenjima dobivenih egzaktnih metoda, dakle onih koje sigurno pronalaze optimalno rešenje ukoliko ono postoji, pokazalo se da daju dovoljno kvalitetna rješenja. Uzevši u obzir puno manje vrijeme izvršavanja primjenom heuristika i metaheuristika u odnosu na egzaktne metode, heuristike i metaeuristike se sve više koriste za rješavanje jednokriterijskih problema, a posljednjih godina i višekriterijskih problema. Njihovom se primjenom uvelike smanjuje vrijeme izvršavanja generirajućih metoda za višekriterijske probleme.

Tako Caballero et al. (2006) prilagođavaju metaheuristiku u literaturi poznatu pod engleskim nazivom „tabu search" za rješavanje problema višekriterijskog cjelobrojnog programiranja generiranjem aproksimacije skupa efikasnih rješenja iz kojeg se potom metodama kompromisnog programiranja dolazi do rješenja kojeg će se ponuditi donositelju odluke.

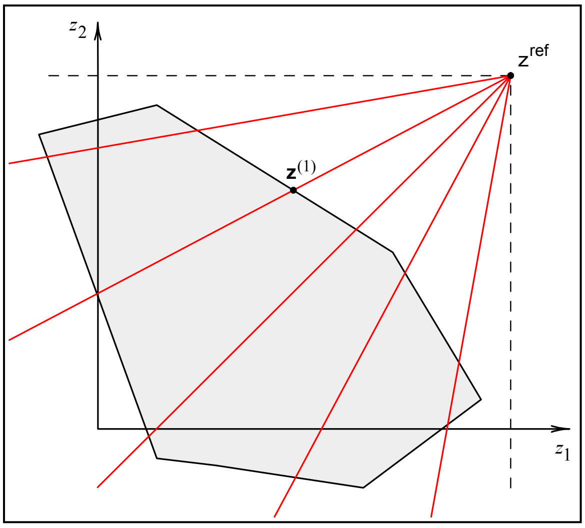
5.2 Navođene metode

Ove metode koriste smjernice dane od donositelja odluke u svojoj potrazi za odgovarajućim efikasnim rješenjem. Dakle, ovim metodama nije cilj pronaći cijeli skup efikasnih rješenja, niti njegovu aproksimaciju. Stoga ovakve metode općenito imaju manje vrijeme izvršavanja nego metode generiranja.

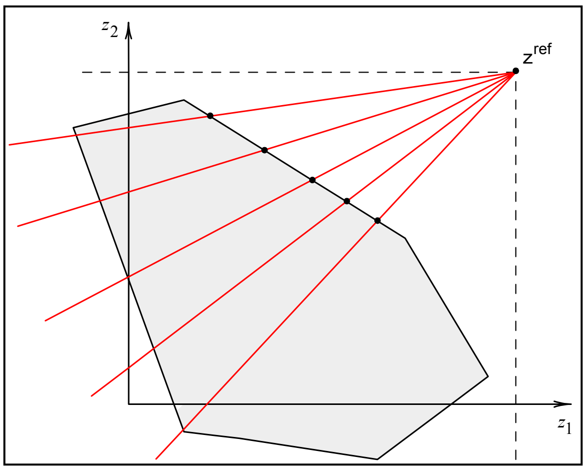
Steuer i Choo (1983) predlažu metodu u kojem donositelj odluke u svakoj iteraciji odabire preferirano rješenje iz manjeg skupa efikasnih rješenja koja su mu predložena. Rješenja koja mu se predlažu na početku su rješenja koja se dobivaju rješavanjem određenog broja uvećanih težinskih Čebiševljevih problema (Pλčρ) koristeći različite težine, obično ravnomjerno distribuirane. Donositelj odabire jedno od ponuđenih rješenja, te se postupak ponavlja, no težine su sada raspoređene bliže onim vrijednostima koje su imale pri rješavanju uvećanih težinskih Čebiševljevih problema kojim je ta točka dobivena. Tako se dobiva točka koja je relativno blizu onoj dobivenoj u prethodnoj iteraciji. Takav postupak se iterativno ponavlja. Očito ova metoda podrazumijeva postojanje implicitne funkcije korisnosti. Prve dvije iteracije ove metode mogu se ilustrirati sljedećim slikama:



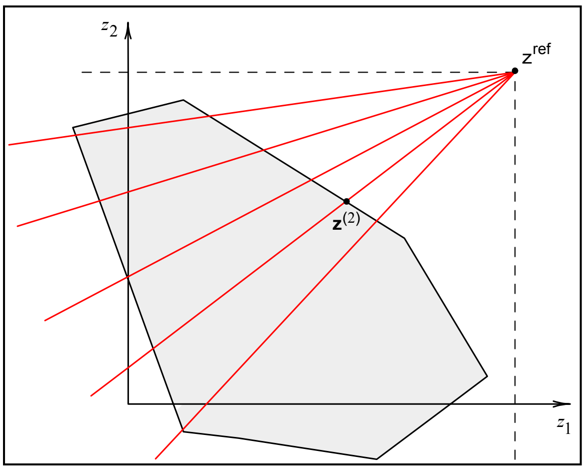
Slika 5.2 Steuer i Choo – iteracija 1



Slika 5.3 Steuer i Choo – iteracija 1



Slika 5.4 Steuer i Choo – iteracija 2



Slika 5.5 Steuer i Choo – iteracija 2

Na slici prikazan je korak gore opisane metode u kojem se donositelju odluke predlaže pet rješenja koja se dobivaju rješavanjem pet uvećanih težinskih Čebiševljevih problema (Pλčρ) koristeći različite, ravnomjerno distribuirane težine, što je prikazano na slici 5.2. Od predloženih rješenja, donositelj odluke odlučuje se za jedno kao što je to prikazano na slici 5.3 gdje je odabrana točka z^1. Zatim se donositelju odluke predlaže pet novih rješenja, ovaj put iz okoline točke z^1. Rješenja se ponovno dobivaju rješavanjem pet uvećanih težinskih Čebiševljevih problema (Pλčρ) koristeći različite, ravnomjerno distribuirane težine što je prikazano na slici 5.4. Tada se donositelj odluke ponovno odlučuje za jedno od predloženih rješenja kao što je to prikazano na slici 5.5 gdje je odabrana točka z^2. Takav postupak se iterativno ponavlja sve dok donositelj odluke ne bude zadovoljan izborom. Tako u ovom postupku donositelj odluke navodi algoritam prema odgovarajućem efikasnom rješenju, te u svakom koraku algoritma „profinjuje“ svoju potragu.

Steuer, Silverman i Whisman (1993) predlažu metodu koja kombinira gore opisanu metodu sa metodom koju je predložio Wierzbicki (1982), a koja koristi takozvani aspiracijski vektor (vektor željenih razina). U metodi Wierbizbickog donositelj odluke u svakoj iteraciji na osnovu ponuđenih odluka navodi svoje željeno rješenje, odnosno željene vrijednosti kriterija, koje se naziva aspiracijskim vektorom, umjesto da bira odluku od ponuđenih odluka. Ova metoda također podrazumijeva postojanje implicitne funkcije korisnosti. Steuer i koautori smatraju da je najbolje kombinirati gore navedene metode jer je njihova metoda korisnija u početnim iteracijama zbog disperzije težina, dok je metodu Wierbizbickog bolje koristiti u kasnijim iteracijama gdje donositelj odluke zapravo odlučuje o konačnom rješenju, pa želi profiniti pretragu.

Karaivanova, Narula i Vassilev (1993) predlažu adaptaciju metode autora Steuer i Choo na način da se za rješavanje uvećanih težinskih Čebiševljevih problema (Pλčρ) koristi heuristika. Dakle, dobivena rješenja mogu biti efikasna, no moguće je dobiti i rješenje koje nije efikasno, ali je blizu nekog efikasnog rješenja.

Vassilev i Narula (1993) predlažu interaktivnu metodu „otvorene komunikacije“ gdje donositelj odluke u svakoj iteraciji ima mogućnost opozvati svoju odluku i promijeniti mišljenje glede usporedbe različitih efikasnih rješenja. Dakle, njegove odluke ne moraju biti niti konzistentne. U svakoj iteraciji maksimizira se najmanja udaljenost od prethodnog rješenja da bi se odmaklo što dalje od njega, no uz dodatna ograničenja koja nastaju specifikacijom kriterija čiju vrijednost u trenutačnom rješenju donositelj odluke želi poboljšati, zatim onih za koje dopušta pogoršanje, te onih za koje bi želio zadržati jednaku vrijednost kriterija.

Sličan princip ima i metoda autora Alves i Climaco (2002) koja od donositelja odluke u svakoj iteraciji traži da direktno specificira točku aspiracije, ili da navede kriterij čiju bi vrijednost želio poboljšati u odnosu na prethodno rješenje.

Primjećuje se da kod interaktivnih pristupa često traži da donositelj usporedi sam neki manji broj efikasnih rješenja, te eventualno odabere jedno. Polazi se od pretpostavke da donositelj odluke može samostalno usporediti i više od dva rješenja, no usporedbe i odabir između više od dva efikasna rješenja mogu biti prilično teške i zahtjevne za donositelja odluke, posebice kod problema sa velikim brojem kriterija (Jaszkiewicz i Slowinski, 1995). Stoga se neke metode ograničavaju na dvije mogućnosti koje se nude donositelju odluke, u svakoj iteraciji ili samo u konačnoj odluci.

Međutim, razvijaju se i metode koje na neki način grupiraju efikasna rješenja, te olakšavaju zadatak donositelja odluke da odabere preferirano rješenje čak i iz uzorka većeg od dva efikasna rješenja.

Tako Vasillev (2003) u svakoj iteraciji algoritma kombinira upite donositelju odluke o kriterijima čije bi vrijednosti želio poboljšati u trenutačnom rješenju, te odabir preferiranog rješenja iz predočenih efikasnih rješenja koja su bliska po kakvoći obzirom na lokalne preferencije donositelja odluke.

Steuer u svom programu ADBASE (2006.) za generiranje skupa efikasnih rješenja primjenjuje metodu koja se bazira na tome da se donositelju u svakoj iteraciji predočuje određen broj efikasnih rješenja koja bi trebao usporediti, a ta efikasna rješenja dobivaju se grupiranjem efikasnih rješenja oko odabranog rješenja u prethodnoj iteraciji.

6 Heuristike

Prva heuristika koja će biti kreirana koristi kompromisno programiranje u kojem se minimizira udaljenost od idealne točke, dok druga heuristika koja će biti kreirana koristi kompromisno programiranje sa težinama.

Obje heuristike koriste metodu grupiranja koju su predložili Shi i Malik (2000) u radu [15]. Metoda je osmišljena za segmentaciju slike, dakle, podjelu slike u grupe piksela koji su blizu jedan drugog ili onih koji su slične boje. No, metoda se može poopćiti i koristiti za grupiranje točaka (u ovom slučaju alternativa) po sličnosti. Dakle, sve točke koje su na neki način slične, trebale bi pripasti istoj grupi. Broj grupa može, ali i ne mora unaprijed biti poznat. Takve grupe nastale grupiranjem u kojem broj grupa nije unaprijed poznat poznate su u literaturi i pod nazivom engleskim nazivom „cluster“. Sličan problem je i problem klasifikacije, u kojem se točke svrstavaju u unaprijed poznate grupe sa donjim i gornjim ogradama na vrijednosti kriterija za ulazak u svaku grupu.

Osnovna ideja i razlog primjene grupiranja u višekriterijskom odlučivanju je grupiranje mogućih izbora, dakle alternativa, i to samo onih koje predstavljaju efikasna rješenja, obzirom na sličnosti u razinama zadovoljenja kriterija kako bi se za konačni odabir mogle uzeti u razmatranje samo one alternative koje su svrstane u najbolju grupu alternativa. Svaka alternativa koja se nalazi u najboljoj grupi se smatra uspješnom u zadovoljenju kriterija, te su u takvoj grupi razlike među alternativama male, te tada konačna odluka ovisi o donositelju odluke. Na taj način metoda kao potpora odlučivanju dobiva na fleksibilnosti i bliža je razmišljanju donositelja odluke.

Postoje različite metode za gore opisana grupiranja, no one često daju neželjene rezultate i svrstavaju u zasebne grupe točke koje možda ne bi trebalo. Vrlo često takve metode favoriziraju grupe u kojima se nalazi samo jedna točka. U metodi koja će ovdje biti opisana, ta pojava se izbjegava korištenjem drugačijeg kriterija podjele točaka.

Općenito, problem grupiranja u dvije grupe može se opisati kao podjela skupa točaka S na dva disjunktna skupa. Pri tome se u svakoj grupi (skupu) trebaju nalaziti točke koje su na neki način slične. To znači da će se rez (u smislu razdvajanja početnog skupa na dvije grupe) raditi tamo gdje su razlike među točkama najveće, odnosno sličnosti najmanje. Dakle, pri kreiranju grupa, minimizira se rez koji ukazuje na sličnosti elemenata. To se može shvatiti kao rezanje skupa točaka na dva dijela pri čemu se traži rez koji će raskinuti veze između točaka koje su najmanje slične.

Funkcija koja mjeri kolika će bit vrijednost tako raskinutih veza između točaka može se definirati na različite načine.

Najjednostavnije je možda tu mjeru definirati kao zbroj svih veza (sličnosti) između parova točaka koje će biti prekinute. Preciznije, rez između grupa točaka *A* i *B* može se definirati kao suma sličnosti svakog para gdje je jedan element iz skupa A, a drugi element iz skupa B, znači,

$$rez(A,B) = \sum_{a \in A, b \in B} w(a,b),$$

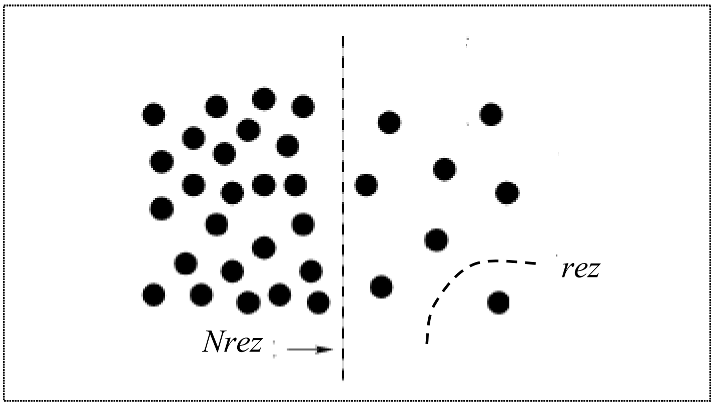
gdje je $w(a,b)$ intenzitet sličnosti elemenata *a* i *b*. Međutim, tada se javlja gore spomenuti problem favoriziranja manjih (jednočlanih) skupova tj. grupa.

Algoritam grupiranja koji predlažu Shi i Malik minimizira rez koji je definiran sa

$$Nrez(A,B) = \frac{rez(a,B)}{sl(A,S)} + \frac{rez(a,B)}{sl(B,S)},$$

gdje je $sl(A,S)$ zapravo suma sličnosti svih elemenata grupe *A* sa elementima cijelog početnog skupa tj. $sl(A,S) = \sum_{a \in A, s \in S} w(a,s)$, te analogno, $sl(B,S) = \sum_{b \in A, s \in S} w(b,s)$. Tako definirani rez navedeni autori nazivaju *normaliziranim rezom*.

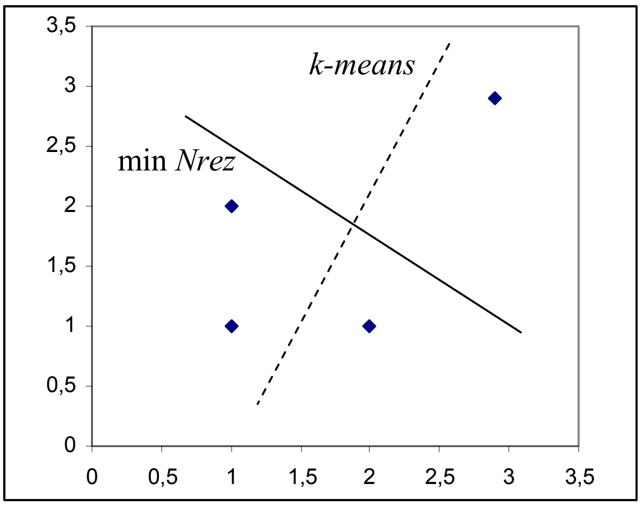
Na sljedećoj slici vidi se grupiranje skupa točaka na dvije grupe dobivenog minimiziranjem reza definiranog sa *rez*, te *Nrez*.



Slika 6.1 **Grupiranje uz *rez* i *Nrez***

Na slici je prikazano grupiranje koje se dobiva ukoliko se skup točaka podijeli na dva disjunktna skupa minimizirajući funkciju *rez*, te *Nrez*. Vidi se da je u prvom slučaju dobivena jedna grupa koja sadrži samo jednu točku, a ostale su točke u drugoj grupi, što ilustrira prije opisani problem favoriziranja jednočlanih skupova. Minimiziranje funkcije *Nrez* dobiva se naočigled bolje grupiranje.

U literaturi se često mogu pronaći radovi u kojima se koristi metoda *k-srednjih vrijednosti* za grupiranje (od eng. *k-means*). Za primjenu metode potrebno je unaprijed zadati broj grupa, kao i početno grupiranje ili centre oko kojih će se napraviti početno grupiranje. Ti centri se nazivaju centroidima, a predstavljaju umjetne točke koje predstavlja „srednju“ lokaciju određene grupe primjera. Kroz iteracije metode dolazi se do „najboljeg“ grupiranja u smislu minimiziranja udaljenosti između elemenata unutar svake grupe. Međutim, metoda je prilično nestabilna jer se za različita početna rješenja dobivaju različita grupiranja. Na primjer, u radu [8] navodi se da je metoda *k-means* primijenjena na isti početni skup točaka 10000 puta sa različitim početnim rješenjima, te je na taj način dobiveno čak 9874 različitih grupiranja. Naime za grupiranje skupa od *n* elemenata na *k* grupa postoji više od (n-k)k / k! različitih početnih rješenja. Također, *k-means* metoda ne funkcionira dobro na tzv. ne-globularnim podacima. Globularni podaci su oni kod kojih su grupe na stanovit način vidljive „golim okom“. Preciznije to su skupovi točaka koji se sastoje od koncentriranih točaka u konveksnim skupovima. Navedeni nedostatak *k-means* metode može se ilustrirati na slici, pa je tako na sljedećoj slici prikazano grupiranje na dvije grupe koristeći *k-means* metodu (jedan od rezultata koji daje - najčešći) i metodu minimiziranja normaliziranog reza *Nrez* koja i ovdje daje bolje grupiranje:



Slika 6.2 **Grupiranje uz *Nrez* i *k-means***

Kako je problem minimiziranja normaliziranog reza NP-težak problem [12], pronalazak točnog rješenja bi zahtijevao veliku količinu vremena za rješavanje te bi takva metoda bila neupotrebljiva u praksi. Stoga autori Shi i Malik predlažu heuristiku koja pronalazi približno rješenje ovog problema, te ima puno kraće vrijeme izvršavanja što ju čini primjenjivom u praksi. Dokazavši da se problem može riješiti rješavanjem generaliziranog problema svojstvenih vrijednosti matrice sličnosti između točaka, predložili su sljedeći postupak za grupiranje:

1. Za dani skup od n točaka, odrediti numeričku mjeru sličnosti i definirati matricu sličnosti između točaka $\mathbf{W}_{n \times n}$ tako da element u i-tom retku i j-tom stupcu matrice predstavlja sličnost i-tog i j-tog elementa početnog skupa, dakle $\mathbf{W} = (w_{ij}), i = 1, \ldots, n, j = 1, \ldots, n.$
2. Riješiti sustav $(\mathbf{D} - \mathbf{W})x = \lambda \mathbf{D}x$ za nekoliko najmanjih generaliziranih svojstvenih vrijednosti λ, pri čemu je $\mathbf{D}$ dijagonalna matrica sa elementima danim sa $\mathbf{D}_{ii} = \sum_j w_{ij}$ što predstavlja sumu sličnosti element i sa svim ostalim elementima.
3. Pomoću svojstvenog vektora koji odgovara drugoj najmanjoj svojstvenoj vrijednosti podijeliti početni skup na dva skupa tj. izvršiti grupiranje.
4. Odlučiti treba li dobivene grupe dalje dijeliti, te ukoliko je potrebno grupe dalje dijeliti na isti način, dakle, rekurzivno.Postupak se završava kada *Nrez* postane veći od neke unaprijed date vrijednosti.

U trećem koraku opisanog postupka, potrebno je podijeliti točke na dvije grupe koristeći generalizirani svojstveni vektor koji odgovara drugoj najmanjoj generaliziranoj svojstvenoj vrijednosti. U idealnom slučaju, komponente generaliziranog svojstvenog vektora bit će cijeli brojevi 1 ili -1, pa bi se u tom slučaju točke podijelile u grupe na sljedeći način:

- $x_i > 0 \quad \Leftrightarrow \quad i$ -ta točka pripada prvoj grupi
- $x_i < 0 \quad \Leftrightarrow \quad i$ -ta točka pripada drugoj grupi

Ukoliko su komponente generaliziranog svojstvenog vektora realni brojevi, tada se odabire točka razgraničenja r koja će dati najmanju vrijednost za *Nrez*, te se točke dijele u grupe ponovno na sličan način:

- $x_i > r \quad \Leftrightarrow \quad i$ -ta točka pripada prvoj grupi
- $x_i < r \quad \Leftrightarrow \quad i$ -ta točka pripada drugoj grupi

Kako autori ovu heuristiku primjenjuju na segmentaciju slike, mjeru sličnosti definiraju tako da ona bude obrnuto proporcionalna udaljenosti između piksela.

Također, kod ove metode grupiranja može se unaprijed zadati željeni broj grupa, ali i ne mora.

Za potrebe heuristike za višekriterijsko odlučivanje, navedena heuristika za grupiranje mora se modificirati tako da se definira drugačija mjera sličnosti u što je potrebno uključiti više kriterija te njihove težine.

Ideja prve heuristike koja se predlaže u ovom radu je najprije grupirati alternative obzirom na najvažnije kriterije koji su međusobno podjednako važni, zatim, najbolju grupu od ovako dobivenih grupa alternativa dalje grupirati po kriterijima koji su sljedeći po važnosti, ali međusobno podjednako važni, i tako dalje. To znači da će mjera sličnosti korištena ovdje biti razlika u udaljenostima od idealne točke koja se opisuju u kompromisnom programiranju. Na ovaj način se izbjegava korištenje konkretnih vrijednosti težina kriterija, no rješenje će biti rezultat leksikografskog uređaja obzirom na grupe kriterija. Preciznije, odabrat će se efikasna rješenja koja su dovoljno blizu idealnoj točki obzirom na najvažnije kriterije, zatim će među tako dobivena rješenjima odabrati ona koja su blizu idealne točke obzirom na sljedeću grupu kriterija i tako dalje.

Druga heuristika se također bazira na gore opisanoj metodi grupiranja, no ona istovremeno grupira obzirom na sve dane kriterije. Koristi se ideja opisana u člancima [19] i [20] o računanju vjerojatnosti da alternativa pripada nekoj grupi. U navedenim člancima riječ je o problemu klasifikacije u kojem se alternative svrstavaju u unaprijed poznate grupe sa donjim i gornjim ogradama vrijednosti kriterija za ulazak u svaku grupu. Ideja se koristi u drugoj heuristici koja se predlaže u ovome radu te će se prilagoditi potrebama rješavanja problema višekriterijskog odlučivanja. Naime, kako bi se svi kriteriji uzeli u obzir istovremeno, potrebno im je dodijeliti težine, no to se mora napraviti tako da težine odgovaraju poretku kriterija po važnosti. Dakle, što je veća važnost kriterija, pridružena mu težina mora biti veća. Tada se postupak grupiranja svih efikasnih rješenja ponavlja određen broj puta pri čemu se svaki put koriste različite težine kriterija koje odgovaraju poretku kriterija po važnosti. Postupak određivanja težina kriterija koje odgovaraju poretku kriterija bit će također opisan. Pri tome je mjera sličnosti razlika u težinskim udaljenostima od idealne točke kakve se koriste u kompromisnom programiranju sa težinama. Broj ulazaka pojedine alternative u najbolju grupu u odnosu na ukupan broj ponavljanja postupka grupiranja ukazuje na vjerojatnost da alternativa zaista treba pripadati najboljoj grupi alternativa.

Dakle, kao komponente heuristika predloženih u ovome radu koristit će se skalarizacije problema višekriterijskog odlučivanja metodama kompromisnog programiranja i kompromisnog programiranja sa težinama, zatim gore opisana heuristika za grupiranje, te metoda računanja vjerojatnosti da pojedina alternativa pripada najboljoj grupi alternativa o čemu će više riječi biti kasnije.

7 Pretpostavke problema valorizacije ponuda zaprimljenih na natječaj Hrvatskog fonda za privatizaciju

7.1 Opis problema

Metode višekriterijskog odlučivanja mogu se primijeniti na odabir ponuda pristiglih na javni natječaj Hrvatskog fonda za privatizaciju (HFP). Kao što je u uvodu navedeno, Hrvatski fond za privatizaciju je utemeljen kako bi proveo i dovršio privatizaciju nekadašnjih društvenih poduzeća, u kojima danas državne institucije još imaju dionice i udjele. Budućnost ovlasti HFP nad odabirom ponuda za privatizaciju je još neizvjesna uslijed previranja koja su se dogodila u 2007. godini, no ovdje će proces odabira biti opisan kako je to navedeno na web stranicama HFP (www.hfp.hr). Međutim, pretpostavke nužne za primjenu metoda višekriterijskog odlučivanja na odabir ponuda za privatizaciju koje će ovdje biti navedene moraju biti poštivane neovisno o ustanovi koja je ovlaštena za isto.

Dakle, kako bi se omogućio transparentno i korektno provođenje privatizacije, razvijene su procedure javnih poziva ili natječaja u kojima se, između ostalog, navode i kriteriji ocjene ponuda. Utvrđene i fiksne procedure natječaja su nužne za garanciju ravnopravnog i transparentnog procesa sklapanja poslova. Na web stranicama HFP objavljeni su koraci u postupku provedbe privatizacije modelom javnog prikupljanja ponuda.

Privatizacija modelom javnog prikupljanja ponuda uključuje slijedeće korake:

A. HFP objavljuje javni Poziv na kupnju dionica u medijima, na svojoj web stranici (www.hfp.hr), i kroz e-mail različitim kanalima uključujući veleposlanstva, trgovačku komoru, zainteresirane stranke i potencijalne investitore.

B. Zainteresirane i kvalificirane osobe koje se natječu mogu kupiti Natječajnu Dokumentaciju i izvršiti «due diligence». Dokumentacija sadrži:

   - Profil društva
   - Uvjete i termine
   - Upute o natjecanju
   - Obrazac za prijavu
   - Obrazac za garancije (garancija ponude i garancija o plaćanju kupovne cijene)
   - Standardne ugovore o prodaji i transferu udjela
   - Kriteriji ocjene ponuda
   - Procedure za posjete društvu

C. HFP mora dobiti ponude najkasnije do krajnjeg roka navedenog u natječaju. Javno otvaranje zapečaćenih ponuda se odvija na datum krajnjeg roka za dostavu ponuda uz prisutstvo javnog bilježnika koji javno bilježi sve ponude. Osobe koje su poslale ponude ili njihovi ovlašteni zastupnici mogu nazočiti javnom otvaranju ponuda.

D. Izvršni tim HFP-a, Upravni Odbor i Vlada Republike Hrvatske ocjenjuju ponude na osnovu unaprijed određenih kriterija.

E. Potpisivanje ugovora.

U odabiru najuspješnije ponude, ključni kriterij nije samo ponuđena cijena nego se također uzimaju u obzir i prijedlozi kupca za razvitak i buduće investicije u društvo kao i da preuzme obveze prema zaposlenicima i vjerovnicima.

Dakle, kriteriji odabira ponuda za privatizaciju su poznati u svakom natječaju iako mogu varirati iz natječaja u natječaj. O kriterijima prema kojima će se ponude valorizirati u svakom natječaju odlučuju stručnjaci HFP-a.

7.2 Mogući kriteriji izbora ponuda za privatizaciju

Kriteriji za vrednovanje ponuda svakako moraju bit mjerljivi. Postoje kriteriji koji su po prirodi kvantitativni, kao što su ponuđena cijena ili planirano ulaganje koji predstavljaju iznose u kunama. Vrijednosti kriterija kao što je, na primjer, visina podmirenja obveza prema državnim vjerovnicima, mogu se izraziti i kao postotak od ukupnog iznosa duga prema državnim vjerovnicima.

Problemi se mogu javiti kod kriterija koji po svojoj prirodi nisu kvantitativni, kao što je bonitet. Bonitet za svakog ponuditelja ocjenjuje strana agencija koju angažira HFP.

Naravno, za svaki kriterij potrebno je znati preferiraju li se veće vrijednosti ili manje. Međutim, osim kriterija koji služe kao mjerilo usporedbe vrijednosti svakog ponuditelja, mogu postojati i eliminacijski kriteriji. Eliminacijski kriteriji su kriteriji kod kojih postoji donji ili gornji prag na vrijednost kriterija koji ako se ne zadovolje eliminiraju takvog ponuditelja iz natječaja. Primjerice, može biti zahtijevana minimalna cijena koja se mora ponuditi ili minimalno ulaganje i slično. U slučaju da ponuditelj ne zadovolji uvjet koji je propisan u konkretnom natječaju, njegova ponuda se eliminira iz razmatranja.

Moguća je i kombinacija kriterija kao mjerila usporedbe i eliminacijskog kriterija, u smislu da se ponude koje zadovolje eliminacijski kriterij i dalje po njemu vrednuju, uz sve ostale kriterije.

Sljedeća pretpostavka koja mora biti zadovoljena da bi se primjenjivale metode višekriterijske optimizacije jest da je donositelj odluke u stanju usporediti važnosti kriterija. Upravo ovdje se javlja veliki problem jer se od donositelja odluke ne može očekivati velika preciznost u ocjenama kriterija, ali konačni ishod uvelike ovisi upravo o važnostima tj. težinama kriterija. Uz male razlike u ocjenama kriterija moguće je dobiti bitno različite efikasne odluke.

8 Formuliranje problema valorizacije ponuda zaprimljenih na natječaj Hrvatskog fonda za privatizaciju

Problem valorizacije ponuda zaprimljenih na natječaj Hrvatskog fonda za privatizaciju može se formulirati kao problem kompromisnog programiranja.

Razmatra se situacija u kojoj se evaluira N ponuda za privatizaciju na osnovu M kriterija. Neka je a_{ij} vrijednost kriterija j, u ponudi za privatizaciju i, $i = 1,\dots,N$, $j = 1,\dots,M$. Na primjer, ukoliko je prvi kriterij ponuđena cijena, tada bi a_{11} predstavljao cijenu koju je ponudio prvi ponuđač, a_{21} cijena koju je ponudio drugi ponuđač i tako dalje.

Kompromisno programiranje minimizira udaljenost od idealnog rješenja, dakle, možemo reći da se traži rješenje koje će biti što bliže idealnom. Idealno rješenje se dobiva na način da se pronađe najbolja ponuđena vrijednost obzirom na svaki kriterij. Na primjer, u problemu odabira ponuda za privatizaciju, idealna ponuda će imati ponuđenu cijenu jednaku najvećoj ponuđenoj cijeni svih ponuđača, imat će planirano ulaganje jednako najvećem planiranom ulaganju svih ponuđača i tako dalje. Međutim, u većini slučajeva, idealna ponuda je nedostižna i predstavlja samo referentnu ponudu kojoj bi konačno odabrana ponuda trebala biti najbliža.

Neka je $a_j^{id}, j = 1,\dots,M$ najbolja vrijednost j-tog kriterija od svih ponuda za privatizaciju. Na primjer, ukoliko je prvi kriterij ponuđena cijena, a_1^{id} je najveća ponuđena cijena u konkretnom natječaju za privatizaciju. Slično, ukoliko je drugi kriterij planirano ulaganje, a_2^{id} je najveće planirano ulaganje od svih ponuda zaprimljenih u konkretnom natječaju za privatizaciju. U slučaju da se kod nekog kriterija preferiraju manje vrijednosti, u idealnoj ponudi će se nalaziti najmanja vrijednost za taj kriterij od svih ponuđača. Dakle, a^{id} predstavlja idealnu ponudu.

Tada se problem odabira između N ponuda za privatizaciju obzirom na M kriterija može formulirati sljedeći skalarizaciju problema višekriterijskog odlučivanja na sljedeći način:

$$\min\left(\sum_{j=1}^{M}\left(a_i^{id} - \sum_{i=1}^{N} a_{ij} X_i\right)^2\right)$$

$$\sum_{i=1}^{N} X_i = 1$$

$$X_i \in \{0,1\}, i = 1,\dots,N,$$

gdje varijabla X_i označava koja je ponuda odabrana, preciznije

$$X_i = \begin{cases} 1 \text{ - ako je odabrana } i \text{ - ta ponuda} \\ 0 \text{ - ako } i \text{ - ta ponuda nije odabrana.} \end{cases}$$

Obzirom da će u natječaju za privatizaciju biti odabrana samo jedna ponuda, u matematičku formulaciju problema uključeno je ograničenje $\sum_{i=1}^{N} X_i = 1$.

Gore navedena formulacija problema odnosi se na slučaj gdje su svi kriteriji podjednako važni, odnosno, kriteriji imaju jednake težine. Da bi se problem formulirao uz pretpostavku različitih težina kriterija, težine će biti označene sa $w_j, j = 1,\ldots,M$. Što je težina veća, to će kriterij imati veću važnost. Na primjer, ukoliko je prvi kriterij ponuđena cijena, a drugi kriterij planirano ulaganje, tada bi se veća važnost ponuđenoj cijeni u odnosu na planirano ulaganje dala na način da se težine postave tako da je $w_1 > w_2$. Također, za težine se najčešće koriste brojevi između 0 i 1, pri čemu je suma svih težina jednaka 1. Uz dodijeljene težine kriterijima, problem odabira ponude za privatizaciju može se formulirati na sljedeći način:

$$\begin{aligned} &\min\left(\sum_{j=1}^{M} w_j \left(a_i^{id} - \sum_{i=1}^{N} a_{ij} X_i \right)^2 \right) \\ &\sum_{i=1}^{N} X_i = 1 \\ &X_i \in \{0,1\}, i = 1,\ldots,N. \end{aligned}$$

U model se mogu uključili i eliminacijski kriteriji kod kojih postoji donji ili donji prag na vrijednost kriterija koji ako se ne zadovolje eliminiraju takvog ponuditelja iz natječaja. Neka su $d \in \{1,\ldots,M\}$ kriteriji kod kojih postoji donji prag, tj. minimalna vrijednost $\underline{a}_d$ uz koju se ponuda uzima u daljnje razmatranje, te neka su $g \in \{1,\ldots,M\}$ kriteriji kod kojih postoji gornji prag, tj. maksimalna vrijednost $\overline{a}_g$ uz koju se ponuda uzima u dalje razmatranje. Tada se problem može formulirati na sljedeći način:

$$\min\left(\sum_{j=1}^{M} w_j \left(a_i^{id} - \sum_{i=1}^{N} a_{ij} X_i\right)^2\right)$$

$$\sum_{i=1}^{N} \left(\overline{a}_g - a_{ig}\right) X_i \geq 0$$

$$\sum_{i=1}^{N} \left(a_{id} - \underline{a}_d\right) X_i \geq 0$$

$$\sum_{i=1}^{N} X_i = 1$$

$$X_i \in \{0,1\},\ i = 1,\ldots,N.$$

Međutim, da bi se primijenio ovakav model, težine je potrebno unaprijed zadati što bi za donositelja odluke bio pretežak zadatak. Naime, donositelj odluke će vjerojatno moći ocijeniti koji su kriteriji važniji, a koji manje važni, no previše je za očekivati da bi on težinama mogao dati konkretne vrijednosti. Na primjer, teško je procijeniti hoće li se nekom kriteriju dati težina 0.1 ili možda 0.11, a male razlike u težinama mogu rezultirati različitim rješenjima problema, odnosno različitim odabirom ponude za privatizaciju.

Stoga će se u heuristici koja će biti opisana pretpostavljati da je dovoljno da donositelj odluke kriterije rangira po važnosti.

9 Rješavanje problema valorizacije ponuda zaprimljenih na natječaj Hrvatskog fonda za privatizaciju

Predlažu se dvije heuristike koje će rješavati problem odabira ponude za privatizaciju. Rad heuristike bit će ilustriran na primjeru sa fiktivnim ponudama i manjim brojem kriterija radi lakšeg opisivanja pojedinih koraka heuristika.

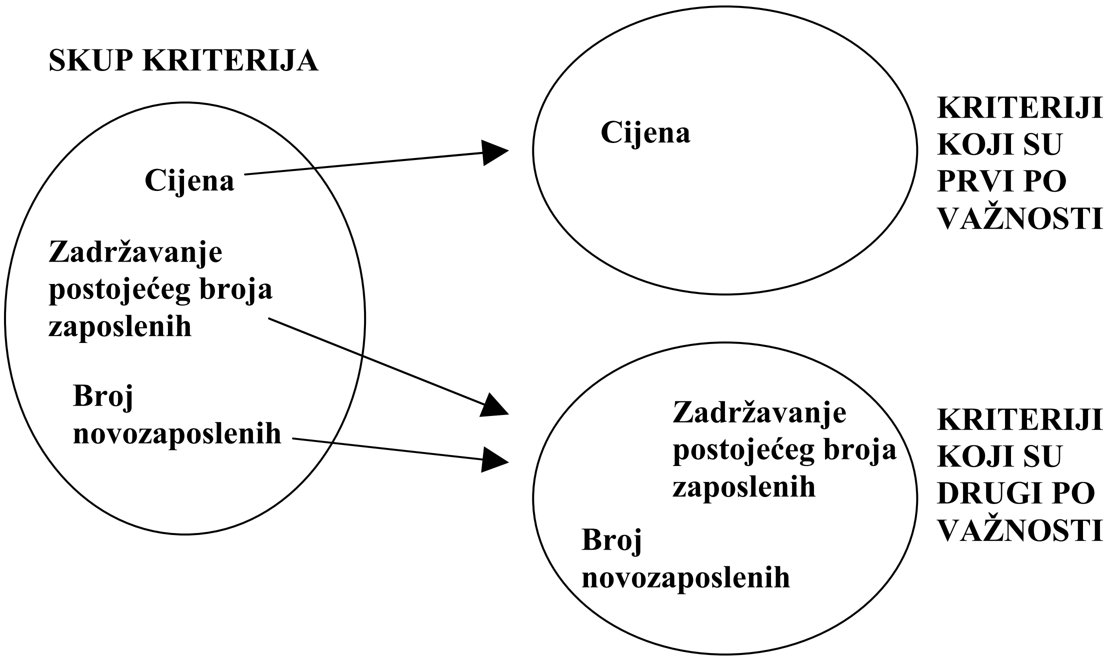
Promatrat će se sljedeći primjer fiktivnog natječaja za privatizaciju u kojem se odabir ponude vrši obzirom na tri kriterija, a na natječaj je zaprimljeno osam ponuda:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih (u god.)	Broj novozaposlenih (u periodu od 3 god.)
A	5000000	3	400
B	4000000	4	500
C	6000000	4	400
D	10000000	3	350
E	11000000	3	300
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0

Tabela 9.1 Ponude zaprimljene na fiktivni natječaj za privatizaciju

Cijena je kriterij koji ponuditelj navodi u kunama. Zadržavanje postojećeg broja zaposlenika ima definiranu eliminacijsku vrijednost na razini od jedne godine: svaki ponuđač mora zadržati sve zaposlene u roku jedne godine. Ponuđač može steći prednost nad ostalim ponuđačima ukoliko ponudi zadržavanje zaposlenika u dužem vremenskom periodu od minimalnog. Broj novozaposlenih se kao dio socijalne komponente može prikazati vrijednostima u apsolutnom broju zaposlenih u razdoblju od iduće tri godine.

Osim što je potrebno navesti po kojim kriterijima se ponude vrednuju, potrebno je da donositelj odluke rangira kriterije po važnosti. Dovoljno je da iskaže koji kriteriji su najvažniji, koji manje važni, te koji su negdje između po važnosti. Taj proces se može ilustrirati na sljedeći način:



Slika 9.1 Grupiranje kriterija po važnosti

Dakle, na prethodnoj slici prikazan je slučaj kada je cijena najvažniji kriterij, dok su kriteriji zadržavanja postojećeg broja zaposlenih, te broj novozaposlenih drugi po važnosti – znači, ne važniji od cijene, ali su međusobno podjednako važni.

9.1 Prilagodba ulaznih podataka

Kako su ulazni podaci tj. vrijednosti kriterija najčešće izraženi u različitim mjernim jedinicama, potrebno je na neki način, prilagoditi podatke. Naime, vrijednosti kriterija je uobičajeno normalizirati svođenjem na interval vrijednosti od 0 do 1 jer su različiti kriteriji mjereni u različitim jedinicama, pa bi, ako bi se vrijednosti kriterija uspoređivale direktno, utjecaji nekih kriterija bili zanemareni zbog drugih koji imaju veći raspon vrijednosti.

Na primjer, u gore navedenom fiktivnom natječaju postoji kriterij cijene koji se navodi u kunama, a s druge je strane kriterij broja novozaposlenih čije su vrijednosti apsolutni brojevi novozaposlenih ljudi. Cijena će često dostići milijunske iznose, dok će broj novozaposlenih teško biti sedmeroznamenkasti broj. Osim toga, vrijednosti ta dva kriterija nisu usporedive, a ne može se ni reći da je podjednaka promjena ako se cijena poveća za jednu jedinicu – jednu kunu ili ako se broj novozaposlenih poveća za jednog čovjeka. Stoga se ulazne podatke ne može direktno koristiti u evaluaciji ponuda, nego se numerički podaci moraju na neki način transformirati, točnije normalizirati kako bi se vrijednosti kriterija lakše mogli usporediti svodeći ih na interval vrijednosti od 0 do 1.

Normalizacija vrijednosti kriterija olakšava računske probleme svojstvene različitim mjernim jedinicama u matrici odluke. Postoje različiti načini normalizacije.

9.1.1 Vektorska normalizacija

Svaku vrijednost kriterija a_{ij} treba podijeliti s normom *j*-tog kriterija tj. normom vektora vrijednosti svih ponuda po *j*-tom kriteriju, dakle vektora $a_j = (a_{1j}, a_{2j}, \ldots, a_{Nj})$. Norma *j*-tog vektora računa se po formuli:

$$\|a_j\| = \sqrt{\sum_{i=1}^{N} a_{ij}^{2}} .$$

Dakle, normalizirane vrijednosti ponuda po kriteriju *j* mogu se tada izračunati kao:

$$a_{ij}^{vnorm} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{N} a_{ij}^{2}}} .$$

Prednost te normalizacije je da su svi kriteriji mjereni u jedinicama bez dimenzije, pa je olakšana usporedba između različitih kriterija.

Nedostatak je činjenica da ta procedura ne vodi do mjernih skala s jednakom duljinom. Minimalne i maksimalne vrijednosti skale nisu jednake za sve kriterije, tako da su izravne usporedbe još uvijek komplicirane.

Na prije navedenom primjeru izbora ponude za privatizaciju na fiktivnom natječaju vektorska bi normalizacija rezultirala sljedećim vrijednostima kriterija:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
A	0,175373835	0,363803438	0,431682605
B	0,140299068	0,48507125	0,539603257
C	0,210448602	0,48507125	0,431682605
D	0,35074767	0,363803438	0,37772228
E	0,385822437	0,363803438	0,323761954
F	0,455971971	0,242535625	0,215841303
G	0,459479448	0,242535625	0,205049238
H	0,462986925	0,121267813	0

Tabela 9.2 Vektorska normalizacija podataka

9.1.2 Linearna transformacija

Kod linearne transformacije je potrebno podijeliti vrijednosti ponuda po pojedinom kriteriju sa maksimalnom vrijednošću tog kriterija (za kriterije kod kojih se preferira veća vrijednost). Time se dobivaju relativne vrijednosti u odnosu na najbolju moguću ponudu, tj. najbolja ponuda po tom kriteriju dobiva ocjenu 1, a ostale su zapravo izražene u postotnom obliku.

Transformirana vrijednost od a_{ij} jednaka je:

$$a_{ij}^{lnorm} = \frac{a_{ij}}{a_j^{\max}},$$

gdje je $a_j^{\max} = \max\limits_{i=1,\dots,N} a_{ij}$.

Očito je da je $0 \le a_{ij}^{lnorm} \le 1$, i ponuda je po j-tom kriteriju bolja što je $a_{ij}^{\ln orm}$ bliži 1. Prednost te transformacije je da su svi podaci transformirani na linearan, proporcionalan način, tako da relativni poredak veličina ostaje isti.

U slučaju kriterija kod kojih se preferiraju manje vrijednosti a_{ij}^{lnorm} se računa drugačije, tj.

$$a_{ij}^{lnorm} = 1 - \frac{a_{ij}}{a_j^{\max}}.$$

Kada postoje i kriteriji kod kojih se preferiraju veće vrijednosti kao i oni kojih se preferiraju manje vrijednosti, nije zgodno koristiti prije navedene načine linearne transformacije vrijednosti kriterija jer su im baze različite. Tada je bolje kriterije kod kojih se preferiraju manje vrijednosti izraziti kao recipročne vrijednosti podataka $1/a_{ij}$. Tada relacija transformacije za takve kriterije postaje:

$$a_{ij}^{lnorm} = \frac{\dfrac{1}{a_{ij}}}{\max\limits_{i=1,\dots,N}\left(\dfrac{1}{a_{ij}}\right)} = \frac{\min\limits_{i=1,\dots,N}\left(a_{ij}\right)}{a_{ij}} = \frac{a_j^{\min}}{a_{ij}}.$$

Na primjeru fiktivnog natječaja linearna bi transformacija rezultirala sljedećim vrijednostima kriterija:

Ponude \ Kriteriji	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
A	0,378787879	0,75	0,8
B	0,303030303	1	1
C	0,454545455	1	0,8
D	0,757575758	0,75	0,7
E	0,833333333	0,75	0,6
F	0,984848485	0,5	0,4
G	0,992424242	0,5	0,38
H	1	0,25	0

Tabela 9.3 Linearna transformacija podataka

Nešto složeniji oblik slične transformacije bio bi

$$a_{ij}^{lnorm} = \frac{a_{ij} - a_j^{\min}}{a_j^{\max} - a_j^{\min}},$$

za kriterije kod kojih se preferiraju veće vrijednosti, te

$$a_{ij}^{lnorm} = \frac{a_j^{\max} - a_{ij}}{a_j^{\max} - a_j^{\min}}$$

za kriterije kod kojih se preferiraju manje vrijednosti.

U nazivniku tih izraza je razlika između najveće i najmanje vrijednosti ponude po *j*-tom kriteriju tako da tada kriteriji poprimaju vrijednosti od 0 do 1. Najlošija ponuda po *j*-tom kriteriju tada ima $a_{ij}^{lnorm} = 0$, dok najbolja ima $a_{ij}^{lnorm} = 1$.Nedostatak takvog pristupa je da takva transformacija skale ne vodi do proporcionalnih promjena u rezultatima.

Transformirani podaci za fiktivni natječaj bi tada izgledali ovako:

Ponude \ Kriteriji	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
A	0,108695652	0,666666667	0,8
B	0	1	1
C	0,217391304	1	0,8
D	0,652173913	0,666666667	0,7
E	0,760869565	0,666666667	0,6
F	0,97826087	0,333333333	0,4
G	0,989130435	0,333333333	0,38
H	1	0	0

Tabela 9.4 Linearna transformacija – drugi način

9.1.3 Transformacija pomoću sume (postotna transformacija)

Jedna od najjednostavnijih transformacija vrijednosti, ali također veoma često korištena, provodi se jednostavno na način da se vrijednosti ponuda po pojedinom kriteriju podijele sa sumom vrijednosti svih ponuda po tom kriteriju, tj.

$$a_{ij}^{snorm} = \frac{a_{ij}}{\sum_{i=1}^{N} a_{ij}} .$$

Na taj način dobiva se postotni udio svake vrijednosti a_{ij} u ukupnoj sumi vrijednosti koje ponude imaju po tom kriteriju.

Na prijašnjem primjeru to izgleda ovako:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
A	0,066401062	0,136363636	0,170940171
B	0,05312085	0,181818182	0,213675214
C	0,079681275	0,181818182	0,170940171
D	0,132802125	0,136363636	0,14957265
E	0,146082337	0,136363636	0,128205128
F	0,172642762	0,090909091	0,085470085
G	0,173970784	0,090909091	0,081196581
H	0,175298805	0,045454545	0

Tabela 9.5 Postotna transformacija podataka

Naravno da suma ocjena u svakom stupcu sada mora biti jednaka jedinici (uz toleranciju zaokruživanja).

9.2 Opis heuristike

Prije nego se pristupi analizi ponuda i odabiru jedne od ponuda za privatizaciju, najprije se vrši eliminacija ponuda koje ne zadovoljavaju eliminacijske kriterije. Na primjer, ukoliko je u prije spomenutom fiktivnom natječaju donji prag na prvi kriterij tj. cijenu bio $\underline{a}_1$= 5.000.000,00 kuna, tada ponuda B ne zadovoljava taj eliminacijski kriterij jer je $a_{21} = 4000000 < 5000000 = \underline{a}_1$. Stoga se ponuda B eliminira iz daljnjeg razmatranja, a ponude koje ostaju za daljnje razmatranje su sljedeće:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
A	5000000	3	400
C	6000000	4	400
D	10000000	3	350
E	11000000	3	300
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0

Tabela 9.6 Eliminacija nezadovoljavajućih ponuda

Nadalje, ponude koje nisu efikasne, u smislu efikasnosti koja se definira u okviru teorije višekriterijske optimizacije, potrebno je također eliminirati iz daljnjeg razmatranja. U fiktivnom natječaju za privatizaciju, ponuda A nije efikasna jer postoji druga ponuda koja je bar po jednom kriteriju bolja od ponude A, a po ostalima nije gora od njega. Na primjer, ponuda C je po prva dva kriterija bolja od ponude A, a po trećem kriteriju je ponuda C jednaka ponudi A. Stoga, ponuda A nije efikasna te ju se eliminira iz daljnjeg razmatranja.

Nakon faze eliminiranja po eliminacijskim kriterijima i eliminiranja ponuda koje nisu efikasne, ostaje sljedeći skup ponuda za dalje razmatranje:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
C	6000000	4	400
D	10000000	3	350
E	11000000	3	300
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0

Tabela 9.7 Eliminacija neefikasnih ponuda

Slijedi evaluacija ponuda korištenjem jedne od dvije heuristike koje će ovdje biti opisane.

9.2.1 Heuristika 1

Prva heuristika koja se predlaže u ovom radu bazira se na sukcesivnom grupiranju ponuda za privatizaciju. Grupiranje se vrši algoritmom opisanim u poglavlju 7 koji su koji su predložili Shi i Malik [15] s time da se kao mjera sličnosti ponuda koristi razlika udaljenosti ponuda od idealne točke. Grupiranje se vrši najprije obzirom na prvu grupu kriterija, a zatim se najbolja grupa od tako dobivenih grupa dalje grupira po sljedećoj grupi kriterija i tako dalje. Najboljom grupom smatrat ćemo onu grupu kojoj pripada ponuda koja ima najmanju udaljenost od idealne točke. Kada se iskoriste sve grupe kriterija, heuristika završava svoj rad, te predlaže ponude od kojih bi se donositelj odluke trebao odlučiti za jednu.

Podaci o ponudama se najprije filtriraju tako da se eliminiraju ponude koje ne zadovoljavaju eliminacijske kriterije, kao i ponude koje nisu efikasne. Zatim se podaci o ponudama normaliziraju, te se zatim izračuna udaljenost od idealne točke za svaku ponudu.

Kako je unutar svake iteracije heuristike potrebno pronaći ponudu koja ima najmanju udaljenost od idealne točke, koristi se model opisan u prethodnom poglavlju. Dakle, u svakoj iteraciji se rješava problem težinskog kompromisnog programiranja. Međutim, pošto je broj ponuda konačan i dovoljno mali, ovdje se taj problem može riješiti računanjem težinske udaljenosti od idealne točke za svaku ponudu i odabirom one čija je tako izračunata udaljenost najmanja. Osim toga, kako je za potrebe algoritma grupiranja također potrebno izračunati težinske udaljenosti od idealne točke za svaku ponudu, traženje rješenja gore spomenutog problema kompromisnog programiranja ne povećava vrijeme izvršavanja heuristike.

Sljedeća tablica daje prikaz idealne točke izračunate na temelju originalnih podataka o ponudama tj. vrijednosti $a_{ij}, i = 1,\ldots,6, j = 1,\ldots,3$:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
C	6000000	4	400
D	10000000	3	350
E	11000000	3	300
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0
Idealna ponuda	13200000	4	400

Tabela 9.8 Idealna ponuda

Međutim, heuristika koristi vrijednosti kriterija normalizirane linearnom transformacijom, te idealnu ponudu obzirom na tako transformirane vrijednosti tj. vrijednosti $a_{ij}^{lnorm}, i = 1,\ldots,6, j = 1,\ldots,3$:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
C	0,454545455	1	1
D	0,757575758	0,75	0,875
E	0,833333333	0,75	0,75
F	0,984848485	0,5	0,5
G	0,992424242	0,5	0,475
H	1	0,25	0
Idealna ponuda	1	1	1

Tabela 9.9 Idealna ponuda – normalizirana

U ovom fiktivnom natječaju za privatizaciju bilo je pretpostavljeno da je cijena najvažniji kriterij, a zadržavanje postojećeg broja zaposlenih i broj novozaposlenih su sljedeći kriteriji po važnosti i međusobno su podjednako važni.

Stoga se ponude najprije grupiraju obzirom na prvi kriterij pa je ovdje mjera sličnosti koja se koristi u algoritmu grupiranja razlika udaljenosti od idealne ponude, ali samo obzirom na prvu grupu kriterija, dakle, u ovom slučaju cijenu. Znači, promatraju se samo sljedeći podaci:

Kriteriji / Ponude	Cijena
C	0,454545455
D	0,757575758
E	0,833333333
F	0,984848485
G	0,992424242
H	1
Idealna ponuda	1

Tabela 9.10 Kriterij cijene

Tada su udaljenosti ponuda od idealne ponude računate korištenjem L_2 metrike tj.

$$dist(id,i) = \left\| a^{id} - a^{i} \right\|_2 = \left(\sum_{j=1}^{1} \left| a_j^{id} - a_{ij} \right|^2 \right)^{\frac{1}{2}} = \sqrt{\sum_{j=1}^{1} \left(a_j^{id} - a_{ij} \right)^2}, i = 1,\ldots,6,$$

pri čemu pretpostavljamo da se radi sa već normaliziranim vrijednostima kriterija ponuda.

U ovom slučaju formula se svodi na $\left\| a^{id} - a^{i} \right\|_2 = a_1^{id} - a_{i1} = 1 - a_{i1}, i = 1,\ldots,6$, te su tako izračunate udaljenosti ponuda od idealne točke date u sljedećoj tablici:

Ponude	Udaljenost od idealne ponude
C	0,545454545
D	0,242424242
E	0,166666667
F	0,015151515
G	0,007575758
H	0

Tabela 9.11 Udaljenost od idealne ponude uz kriterij cijene

Razlike udaljenosti ponuda od idealne točke se koriste za definiranje mjere sličnosti ponuda. Međutim, kako je sličnost ponuda veća što su im razlike udaljenosti od idealne točke manje, mjera sličnosti ponuda za potrebe algoritma grupiranja definira se kao:

$$w_{ik} = \begin{cases} 1 - \left| dist(id,i) - dist(id,k) \right|, & i \neq k \\ 0, \quad i = k. \end{cases}$$

Iako bi w_{ik} u slučaju da je $i = k$ trebao biti jednak 1 jer je ponuda sama sebi najviše slična, radi potrebe algoritma, ta vrijednost se postavlja na 0.

Matrica sličnosti ponuda tada izgleda ovako:

Sličnosti	C	D	E	F	G	H
C	0	0,69697	0,621212	0,469697	0,462121	0,454545
D	0,69697	0	0,924242	0,772727	0,765152	0,757576
E	0,621212	0,924242	0	0,848485	0,840909	0,833333
F	0,469697	0,772727	0,848485	0	0,992424	0,984848
G	0,462121	0,765152	0,840909	0,992424	0	0,992424
H	0,454545	0,757576	0,833333	0,984848	0,992424	0

Tabela 9.12 Matrica međusobnih sličnosti ponuda

Može se primijetiti da je sličnost ponuda C i D jednaka 0.69697, dok je sličnost ponuda E i D jednaka 0.924242, što znači da je ponuda D sličnija ponudi E nego ponudi C. Stoga bi se ponuda D prije našla u grupi sa ponudom E nego sa ponudom C.

Algoritam za grupiranje zahtijeva i računanje sume sličnosti ponude i sa svim ostalim ponudama što je u algoritmu za grupiranje koji se ovdje koristi bilo označeno sa $\mathbf{D}_{ii} = \sum_j w_{ij}$ i činilo matricu $\mathbf{D}$. U sljedećoj tablici prikazani su izračunati elementi matrice $\mathbf{D}$:

Sličnosti sa svim ostalim ponudama	C	D	E	F	G	H
C	2,704545	0	0	0	0	0
D	0	3,916667	0	0	0	0
E	0	0	4,068182	0	0	0
F	0	0	0	4,068182	0	0
G	0	0	0	0	4,05303	0
H	0	0	0	0	0	4,022727

Tabela 9.13 Sume sličnosti ponuda

Sada je potrebno riješiti sustav $(\mathbf{D}-\mathbf{W})x = \lambda \mathbf{D}x$ za nekoliko najmanjih generaliziranih svojstvenih vrijednosti, te se vrši podjela skupa ponuda tj. grupiranje kako je to opisano u algoritmu grupiranja u poglavlju 7.

Od svih grupa koje se dobiju ovim grupiranjem odabire se najbolja tj. ona u kojoj je ponuda koja ima najmanju udaljenost od idealne točke. U ovom slučaju, najbolja grupa dobivena grupiranjem po prvoj grupi kriterija (cijena) jest grupa ili skup ponuda {F,G,H} tj. grupa:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0

Tabela 9.14 Grupiranje ponuda po kriteriju cijene

Dakle, eliminirane su ponude C, D i E, te se sada vrši grupiranje preostalih ponuda obzirom na drugu grupu kriterija. U ovom fiktivnom natječaju kriteriji koji su sljedeći po važnosti, ali su međusobno podjednako važni, su kriteriji zadržavanja postojećeg broja zaposlenih, te broj novozaposlenih.

Heuristika sada treba grupirati preostale ponude obzirom na drugu grupu kriterija jer se one smatraju sličnima po prvoj grupi kriterija. Stoga su podaci koji se dalje razmatraju sljedeći:

Kriteriji / Ponude	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
F	2	200
G	2	190
H	1	0

Tabela 9.15 Ponude preostale nakon prvog grupiranja

Tada se udaljenosti ponuda od idealne ponude ponovno računaju korištenjem L_2 metrike, međutim sada se promatra drugi i treći kriterij istovremeno tj.

$$dist(id,i) = \left\| a^{id} - a^{i} \right\|_2 = \left(\sum_{j=2}^{3} \left| a_j^{id} - a_{ij} \right|^2 \right)^{\frac{1}{2}} = \sqrt{\sum_{j=2}^{3} \left(a_j^{id} - a_{ij} \right)^2}, i = 1,\ldots,6,$$

pri čemu pretpostavljamo da se radi već normaliziranim vrijednostima kriterija ponuda.

Gore navedena formula sada se svodi na

$$dist(id,i) == \sqrt{\left(a_2^{id} - a_{i2}\right)^2 + \left(a_3^{id} - a_{i3}\right)^2}, i = 1,\ldots,6,$$

te se ovako izračunate udaljenosti ponuda od idealne točke koriste za izračunavanje sličnosti ponuda na način koji je prije opisan.

Matrica sličnosti ponuda tada izgleda ovako:

Sličnosti	F	G	H
F	0	0,982549	0,542893
G	0,982549	0	0,560344
H	0,542893	0,560344	0

Tabela 9.16 Matrica međusobnih sličnosti ponuda

U sljedećoj tablici prikazane su izračunati elementi matrice **D** potrebne za algoritam:

Sličnosti sa svim ostalim ponudama	F	G	H
F	1,525442	0	0
G	0	1,542893	0
H	0	0	1,103238

Tabela 9.17 Sume sličnosti ponuda

Od svih grupa koje se dobiju ovim grupiranjem odabire se najbolja tj. ona u kojoj je ponuda koja ima najmanju udaljenost od idealne točke, ovaj put obzirom na drugu grupu kriterija. Najbolju grupu ponuda čine ponude F i G, a ponuda H je stoga eliminirana.

Heuristika je završila sa radom jer nema više grupa kriterija po kojima bi se dalje vršilo grupiranje ponuda.

Cjelokupni postupak koji je heuristika izvršila može se ilustrirati na sljedeći način:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
A	5000000	3	400
B	4000000	4	500
C	6000000	4	400
D	10000000	3	350
E	11000000	3	300
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0

Eliminacija ponuda koje ne zadovoljavaju eliminacijske kriterije

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
A	5000000	3	400
C	6000000	4	400
D	10000000	3	350
E	11000000	3	300
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0

Eliminacija ponuda koje nisu efikasne

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
C	6000000	4	400
D	10000000	3	350
E	11000000	3	300
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0

Eliminacija ponuda obzirom na kriterij cijene

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0

Eliminacija ponuda obzirom na kriterije zadržavanja postojećeg broja zaposlenih i broja novozaposlenih

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
F	13000000	2	200
G	13100000	2	190

Heuristika svoj rad završava prijedlogom ponuda od kojih bi se donositelj odluke trebao odlučiti za jednu. Moguće je da će se prijedlog sastojati od samo jedne ponude, no moguće je da se predlaže i više ponuda koje se smatraju podjednako dobrima i na donositelju odluke je da napravi konačni izbor.

Tako u ovom fiktivnom natječaju heuristika predlaže donositelju odluke da se odluči za jednu od ponuda F i G i ostavlja donositelju odluke konačan izbor ponude.

Cijela procedura izvršavanja opisane heuristike može se sažeti na sljedeći način:

Heuristika1

G ← Grupe kriterija za evaluaciju ponuda

S ← Skup ponuda pristiglih na natječaj

S ← eliminiraj_donji_gornji_prag(S)

S ← eliminiraj_neefikasne(S)

S ← transformiraj_podatke(S)

za svaku grupu kriterija *j* iz G **radi**

    P ← grupiraj(S, *j*-ta grupa kriterija iz G)

    S ← najbolja_grupa_ponuda(P)

predloži donositelju odluke ponudu/e iz S

Slika 9.2 Sažetak heuristike 1

9.2.2 Heuristika 2

Druga heuristika koja se predlaže u ovom radu se također bazira se na grupiranju ponuda za privatizaciju, pri čemu se grupiranje vrši algoritmom opisanim u poglavlju 7 sa mjerom sličnosti ponuda definiranom kao razlika udaljenosti ponuda od idealne točke. Međutim, grupiranje se vrši obzirom sve kriterije istovremeno, ali tako da se pazi na poredak kriterija po važnosti.

Da bi se grupiranje vršilo obzirom na sve kriterije istovremeno poštujući poredak kriterija po važnosti, kriterijima moraju biti pridodane težine koje će odgovarati poretku kriterija. Kako se od donositelja odluke ne traži da specificira težine, heuristika vrši simulacije pri kojima svaki put grupira ponude obzirom na drugačije postavljene težine koje sama heuristika slučajno određuje poštujući pri tome poredak kriterija po važnosti. U svakom takvom grupiranju određuje se najbolja dobivena grupa ponuda, te se po završetku svih simulacija broji koliko je puta određena ponuda bila smještena u najbolju grupu. Taj se broj u odnosu na ukupan broj simulacija smatra vjerojatnošću da promatrana ponuda pripada grupi najboljih ponuda.

Ovaj proces simuliranja ponašanja nekog modela sa različitim vrijednostima ulaznih parametara poznat je pod nazivom Monte Carlo simulacija [9]. Metode Monte Carlo simulacije omogućavaju aproksimativno rješavanje mnoštva matematičkih problema pomoću implementacije statističkog uzorkovanja na računalu. Osnovni je princip izračunavanje vrijednosti zadane funkcije slučajnih varijabli. Pri tome je potrebno za svaku slučajnu varijablu generirati niz uzoraka koji se podvrgava zadanoj razdiobi, a potom se za svaki skup uzoraka izračunava vrijednost funkcije čije se ponašanje simulira. Kao rezultat se dobiva razdioba funkcije slučajnih varijabli.

Varijable koje se generiraju pri ovoj heuristici su težine kriterija za odabir ponude, a funkcije čija se ponašanja simuliraju jesu težinske udaljenosti od idealne ponude koje se dalje koriste u postupku grupiranja ponuda, te odabiru najbolje grupe. Broj slučajnih odabira u postupku simuliranja bit će nazvan brojem ponavljanja, te mora biti dovoljno velik da bi se dobila dovoljno precizna simulacija.

Kao i u prvoj heuristici koja je opisana, i ovdje se podaci o ponudama najprije filtriraju tako da se eliminiraju ponude koje ne zadovoljavaju eliminacijske kriterije, kao i ponude koje nisu efikasne. Zatim se podaci o ponudama normaliziraju, te se zatim izračuna udaljenost od idealne točke za svaku ponudu. Međutim, za razliku od prve heuristike gdje je mjera sličnosti bila razlika u udaljenostima od idealne točke kakve se koriste u kompromisnom programiranju, u drugoj heuristici se moraju uzeti u obzir i težine kriterija. Stoga će ovdje mjera sličnosti biti razlika u težinskim udaljenostima od idealne točke kakve se koriste u kompromisnom programiranju sa težinama.

Rad druge heuristike će također biti prikazan na prijašnjem primjeru fiktivnog natječaja za privatizaciju u kojem su nakon prvih eliminacija ostale za daljnje razmatranje sljedeće ponude:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
C	6000000	4	400
D	10000000	3	350
E	11000000	3	300
F	13000000	2	200
G	13100000	2	190
H	13200000	1	0
Idealna ponuda	13200000	4	400

Tabela 9.18 Ponude preostale nakon početnih eliminacija

Transformirani podaci su kao i prije:

Kriteriji / Ponude	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih
C	0,454545455	1	1
D	0,757575758	0,75	0,875
E	0,833333333	0,75	0,75
F	0,984848485	0,5	0,5
G	0,992424242	0,5	0,475
H	1	0,25	0
Idealna ponuda	1	1	1

Tabela 9.19 Normalizirani podaci o ponudama

Za potrebe prvog ponavljanja tj. prvog grupiranja obzirom na sve kriterije, heuristika izračunava težine kriterija na slučajan način, no poštujući poredak kriterija po važnosti. Kako je cijena bila najvažniji kriterij, a zadržavanje postojećeg broja zaposlenih i broj novozaposlenih druga grupa kriterija po važnosti, primjer težina kriterija koje bi heuristika mogla u ovom slučaju predložiti predstavljaju težine:

$$w_1 = 0.9, \quad w_2 = 0.05, \quad w_3 = 0.05.$$

Dakle, cijena ne smije imati manju težinu od druga dva kriterija, a druga dva kriterija su podjednako važni pa imaju jednake težine. Suma težina svih kriterija treba biti jednaka jedan.

Uz ovako postavljene težine, heuristika računa težinsku udaljenost ponuda od idealne ponude na sljedeći način:

$$wdist(id,i) = \left\| a^{id} - a^{i} \right\|_2^w = \left(\sum_{j=1}^{3} w_j \left| a_j^{id} - a_{ij} \right|^2 \right)^{\frac{1}{2}} = \sqrt{\sum_{j=1}^{3} w_j \left(a_j^{id} - a_{ij} \right)^2}, i = 1, \ldots, 6,$$

te su tako izračunate udaljenosti ponuda od idealne točke date u sljedećoj tablici:

Ponude	Udaljenost od idealne ponude
C	0,517463617
D	0,238325013
E	0,176776695
F	0,158765902
G	0,162274160
H	0,279508497

Tabela 9.20 Težinska udaljenost od idealne ponude

Vidi se da je uz ovakve težine ponuda F najbliža idealnoj ponudi.

Razlike udaljenosti ponuda od idealne točke se i ovdje koriste za definiranje mjere sličnosti ponuda. Također, mjeru sličnosti ponuda za potrebe algoritma grupiranja se definira slično kao i prije tj.:

$$w_{ik} = \begin{cases} 1 - \left| wdist(id,i) - wdist(id,k) \right|, & i \neq k \\ 0, & i = k. \end{cases}$$

Matrica sličnosti ponuda tada izgleda ovako:

Sličnosti	C	D	E	F	G	H
C	0	0,720861	0,659313	0,641302	0,644811	0,762045
D	0,720861	0	0,938452	0,920441	0,923949	0,958817
E	0,659313	0,938452	0	0,981989	0,985497	0,897268
F	0,641302	0,920441	0,981989	0	0,996492	0,879257
G	0,644811	0,923949	0,985497	0,996492	0	0,882766
H	0,762045	0,958817	0,897268	0,879257	0,882766	0

Tabela 9.21 Matrica međusobnih sličnosti ponuda

Od svih grupa koje se dobiju ovim grupiranjem odabire se najbolja tj. ona u kojoj je ponuda koja ima najmanju težinsku udaljenost od idealne točke obzirom na sve kriterije istovremeno.

Najbolju grupu ponuda sa ovako definiranim težinama čine ponude E, F i G.

Nakon završenog prvog grupiranja, heuristika ponovno razmatra ponude koje je razmatrala i u prvom grupiranju, te izračunava nove težine kriterija na slučajan način, no opet poštujući poredak kriterija po važnosti. Dakle, radi se drugo ponavljanje unutar Monte Carlo simulacije. Primjer težina kriterija koje bi heuristika mogla sada predložiti predstavljaju težine:

$$w_1 = 0.6, \quad w_2 = 0.2, \quad w_3 = 0.2.$$

Uz ovako postavljene težine, heuristika ponovno izračunava težinsku udaljenost ponuda od idealne ponude, te su tako izračunate udaljenosti ponuda od idealne točke date u sljedećoj tablici:

Ponude	Udaljenost od idealne ponude
C	0,422507274
D	0,225580824
E	0,204124145
F	0,316445479
G	0,324282956
H	0,559016994

Tabela 9.22 Težinska udaljenost od idealne ponude

Vidi se da je uz ovakve težine ponuda E najbliža idealnoj ponudi, dok je u prvom grupiranju sa drugačijim težinama to bila ponuda F. Objašnjenje leži u težinama. Naime, u oba slučaja su težine postavljene tako da se poštuje poredak kriterija po važnosti, no u drugom slučaju je manja težina dana cijeni nego u prvom slučaju. Stoga se mijenja i ponuda koja je najbliža idealnoj obzirom na težinsku udaljenost, a time se mijenjaju i sličnosti između ponuda.

Matrica sličnosti ponuda sa ovdje postavljenim težinama sada izgleda ovako:

Sličnosti	C	D	E	F	G	H
C	0	0,803074	0,781617	0,893938	0,901776	0,86349
D	0,803074	0	0,978543	0,909135	0,901298	0,666564
E	0,781617	0,978543	0	0,887679	0,879841	0,645107
F	0,893938	0,909135	0,887679	0	0,992163	0,757428
G	0,901776	0,901298	0,879841	0,992163	0	0,765266
H	0,86349	0,666564	0,645107	0,757428	0,765266	0

Tabela 9.23 Matrica međusobnih sličnosti ponuda

Od svih grupa koje se dobiju ovim grupiranjem odabire se najbolja tj. ona u kojoj je ponuda koja ima najmanju težinsku udaljenost od idealne točke obzirom na sve kriterije istovremeno.

Najbolju grupu ponuda ovaj put činit će ponude D i E.

Sa prvom postavkom težina najbolju grupa ponuda činile su ponude E, F i G. Dakle, od ovdje prikazane dvije simulacije sa različitim postavljanjem težina, ponude C i H nisu niti jednom bile svrstane u najbolju grupu ponuda, ponude D, F i G su po jedan put bile u najboljoj grupi, dok je ponuda E u oba grupiranja bila svrstana u najbolju grupu.

Na primjer, u prvom prikazanom grupiranju, sličnost ponuda E i F iznosila je 0.981989, dok je u drugom slučaju iznosila 0.887679. Kako se sa težinama mijenjaju težinske udaljenosti od idealne točke, tako se mijenja i sličnost ponuda što objašnjava različito dobivene najbolje grupe uz različito postavljene težine kriterija.

Međutim, ovaj postupak se mora ponoviti relativno veliki broj puta da bi se moglo na neki način govoriti o vjerojatnosti da se određena ponuda nalazi u najboljoj grupi ponuda.

Dalje će biti prikazan rezultat ponavljanja ovog postupka 500 puta, pri čemu će biti dani podaci o broju ulaska u najbolju grupu za svaku ponudu. Pri tome, u svakoj iteraciji, dakle ponavljanju grupiranja, heuristika će na slučajan način odabrati težine koje će poštivati zadani poredak važnosti kriterija. Vjerojatnost da se ponuda zaista treba nalaziti u najboljoj grupi ponuda dobit će se kao kvocijent broja ponavljanja u kojima je ponuda pripala najboljoj grupi, i ukupnog broja ponavljanja. U sljedećoj tablici prikazani su brojevi ulaska svake ponude u najbolju grupu od svih 500 iteracija, te vjerojatnosti da se ponuda zaista treba nalaziti u najboljoj grupi ponuda tj. u grupi ponuda koje bi donositelj odluka trebao razmotriti pri konačnoj odluci:

Ponude	Broj ulazaka u najbolju grupu (maks. 500)	Vjerojatnosti pripadnosti grupi najboljih ponuda	Postotak ulazaka u najbolju grupu
C	0	0	0%
D	410	0.82	82%
E	460	0.92	92%
F	350	0.70	70%
G	350	0.70	70%
H	10	0.02	2%

Tabela 9.24 Vjerojatnosti pripadnosti ponuda najboljoj grupi ponuda

Heuristika svoj rad završava s ovako izračunatim vjerojatnostima pripadnosti najboljoj grupi koje donositelju odluke daju smjernice pri konačnom odabiru. Naime, donositelj odluke bi najprije trebao razmotriti ponude sa najvećim vjerojatnostima pripadnosti najboljoj grupi ponuda od kojih bi se donositelj odluke trebao odlučiti za jednu.

Dakle, najprije je generiran niz vektora težina kriterija koje su generirane uz uniformnu razdiobu u okviru ograničenja koja su rezultirala grupiranjem kriterija po težini od strane donositelja odluke. Tako generirane težine koristile su se u izračunavanju udaljenosti ponuda od idealne točke, zatim, grupiranju ponuda, te odabiru najbolje grupe ponuda. Kao rezultat se dobila raspodjela pripadnosti ponuda najboljoj grupi kao funkcija slučajnih težina kriterija.

Napravljeno je 500 ponavljanja unutar Monte Carlo simulacije, te je moguće ispitati analizu osjetljivosti na broj ponavljanja što će biti prikazano u sljedećem poglavlju.

Također, potrebno je ispitati koliko bi se rezultati heuristike promijenili ukoliko bi se koristila normalna distribucija za generiranje slučajnih težina kriterija.

Cijela procedura izvršavanja opisane heuristike može se sažeti na sljedeći način:

Heuristika2

G ← Grupe kriterija za evaluaciju ponuda

S ← Skup ponuda pristiglih na natječaj

S ← eliminiraj_donji_gornji_prag(S)

S ← eliminiraj_neefikasne(S)

S ← transformiraj_podatke(S)

za svaku ponudu *i*

  br_ulaska_u_najgrupu(*i*) = 0

  vj_pripadnosti_ najgrupi(*i*) = 0

dok je $k \leq$ *maks_iter* **radi**

  w ← slučajne_težine_poštujući_poredak_kriterija

  P ← grupiraj(S, *w*)

  za svaku ponudu *i*

    ako $i \in$ najbolja_grupa_ponuda(P) **onda**

      br_ulaska_u_najgrupu(*i*) = br_ulaska_u_najgrupu(*i*) + 1

vj_pripadnosti_ najgrupi(*i*) = br_ulaska_u_najgrupu(*i*) / *maks_iter*

Slika 9.3 Sažetak heuristike 2

Rečeno je da se slučajne težine generiraju od strane heuristike poštujući poredak kriterija. Za generiranje težina osmišljena su dva algoritma.

Prvi algoritam kriterijima dodjeljuje težine tako da važniji kriteriji imaju veće težine od manje važnih, kao i da kriteriji koji su jednako važni imaju jednake težine. Na kraju se težine transformiraju tako da suma težina bude jednaka jedan. Algoritam se može sažeti na sljedeći način:

<u>TežineKriterija1</u>

maks_težina = 1 / ukupan_broj_kriterija

težina_grupe_kriterija(1) = slučajno_između(0,max_težina)

za svaku grupu kriterija *k* > 1 **radi**

    maks_težina = težina_grupe_kriterija(*k*-1)

    težina_grupe_kriterija(*k*) = slučajno_između(0,max_težina)

za svaku grupu kriterija *k* > 1 **radi**

    za svaki kriterij *j* iz grupe kriterija *k* **radi**

        težina_kriterija(*j*) = težina_grupe_kriterija(*k*) / suma svih umnožaka težina_grupe_kriterija(*t*)*broj_kriterija_u_grupi(*t*)

Slika 9.4 Algoritam postavljanja težina kriterija 1

Za razliku od gore opisanog algoritma, pretpostavimo da donositelj želi ne samo da je svaki kriterij u prvoj grupi kriterija važniji od svakog kriterija u drugoj grupi, nego da je svaki kriterij u prvoj grupi kriterija važniji od svih kriterija u drugoj grupi kriterija zajedno. Na primjeru kriterija koji su se koristili u opisu ovdje kreiranih heuristika, to bi značilo da cijena ne samo da je važnija od zadržavanja postojećeg broja zaposlenih i broja novozaposlenih radnika, nego da je cijena važnija od ta druga dva kriterija zajedno. Pod tom pretpostavkom, drugi algoritam za dodjelu težina kriterijima dodjeljuje težine tako da važniji kriteriji imaju veće težine od cijele grupe manje važnih kriterija, kao i da kriteriji koji su jednako važni imaju jednake težine. Na primjer, u primjeru fiktivnog natječaja za privatizaciju, cijena će imati veću težinu od zbroja težina svih kriterija iz grupe sljedeće po važnosti. Dakle, ukoliko bi se cijeni dodijelila težina 0.8, tada druga dva kriterija ne smiju imati zbroj težina veći od

0.8, dakle kriterij zadržavanja broja zaposlenih ne smije imati težinu veću od 0.4, što vrijedi i za kriterij broja novozaposlenih, no ta dva kriterija i tako trebaju imati jednaku težinu.

Na kraju se težine i ovdje transformiraju tako da suma težina bude jednaka jedan. Algoritam je vrlo sličan prvom algoritmu za dodjelu težina kriterija, te se ovdje neće navoditi.

Kako je već rečeno, oba navedena algoritma koriste uniformnu razdiobu za slučajno generiranje težina, te će u sljedećem poglavlju dati prikaz rezultata dobivenih uz korištenje normalne razdiobe pri generiranju težina.

Također, bit će dane usporedbe rezultata ovdje opisane heuristike uz korištenje prvog i drugog algoritma za slučajno generiranje težina kriterija za evaluaciju ponuda.

Iz rezultata heuristika za navedeni primjer odabira ponuda za privatizaciju, vidljivo je da heuristike ne daju iste rezultate. Ova pojava bit će razjašnjena ukoliko se rezultati obje heuristike usporede koristeći tabelarni prikaz svih ponuda na fiktivnom natječaju za privatizaciju uz ishode heuristika.

Podaci se mogu vidjeti u sljedećoj tablici:

Ponude	Kriteriji			Heuristika1	Heuristika2
	Cijena	Zadržavanje postojećeg broja zaposlenih	Broj novozaposlenih	+ predložena – nije predl.	% pripadnosti najboljoj grupi
A	5000000	3	400	eliminirana	eliminirana
B	4000000	4	500	eliminirana	eliminirana
C	6000000	4	400	–	0%
D	10000000	3	350	–	82%
E	11000000	3	300	–	92%
F	13000000	2	200	+	70%
G	13100000	2	190	+	70%
H	13200000	1	0	–	2%

Tabela 9.25 Usporedba rezultata dvaju heuristika

Ponude A i B bile su eliminirane iz daljnjeg razmatranja odmah na početku obje heuristike, B jer nije zadovoljavala donji prag na cijenu, a ponuda A jer nije efikasna. Prva heuristika je zatim najprije odbacila ponude C, D i E obzirom na cijenu, a zatim i ponudu H obzirom na preostala dva kriterija. Donositelju odluke je predložila ponude F i G.

Međutim, iako po drugoj heuristici ponude F i G imaju dosta veliku vjerojatnost pripadnosti najboljoj grupi ponuda, ta heuristika ipak prednost daje ponudi D, a naročito ponudi E. Razlog je u tome što prva heuristika grupira ponude sukcesivno, prateći leksikografski uređaj grupa kriterija, dok druga heuristika traži više kompromisno rješenje, no također poštujući poredak kriterija po važnosti.

Tako druga heuristika daje prednost ponudi E koja ima manju cijenu od ponuda F i G, no znatno je bolja u odnosu na ponude F i G obzirom na druga dva kriterija.

Dakle, prva heuristika je na stanoviti način stroža nego druga heuristika, pa bi bilo poželjno da donositelj odluke možda razmotri najprije prijedlog rješenja po prvoj heuristici, a da zatim promotri i sugestije koje daje druga heuristika.

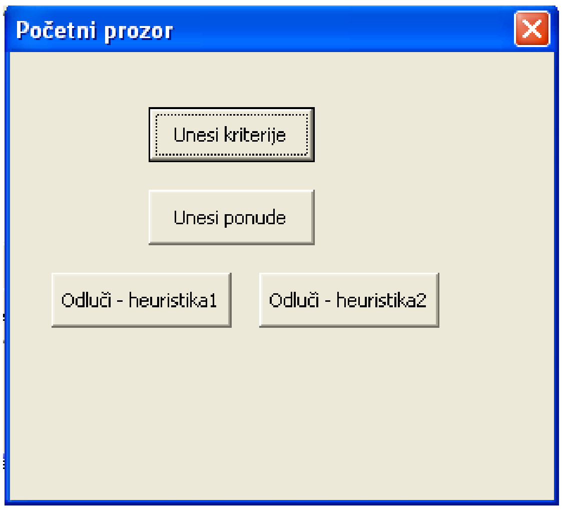
9.3 Implementacija u vidu programskog sustava za potporu odlučivanju

Obje heuristike kreirane u ovome radu implementirane su u vidu programskog sustava za potporu odlučivanju. Implementacija je vršena pomoću programskog jezika Microsoft Excel Visual Basic. U ovome poglavlju bit će dana funkcionalna specifikacija programskog sustava za potporu odlučivanju kreiranog na temelju dvije heuristike koje se u ovom radu predlažu.

9.3.1 Početni prozor programskog sustava za evaluaciju ponuda za privatizaciju

Za prikaz funkcionalnosti koristit će se već korišteni primjer ponuda zaprimljenih na natječaj za privatizaciju.

Početni prozor prikazuje gumbe za unos kriterija, unos ponuda te pomoć pri odabiru ponude uz obje heuristike koje su kreirane u ovom radu:

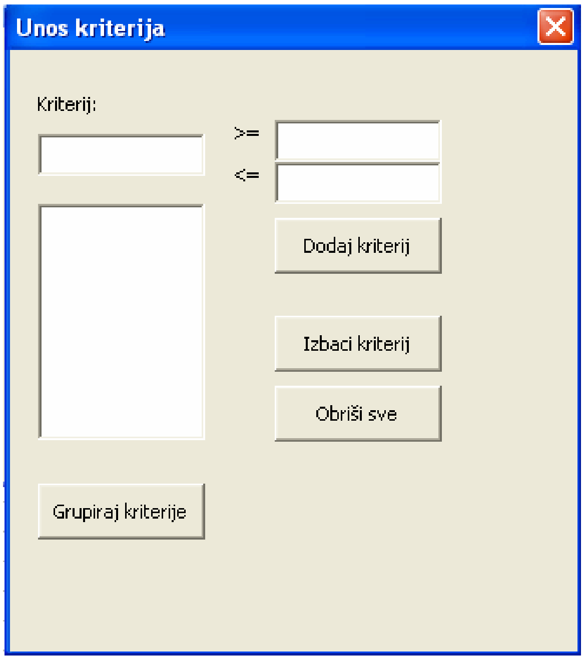


Slika 9.5 DSS – početni prozor

Korisnik najprije odabire opciju za unos kriterija pritiskom na odgovarajući gumb.

9.3.2 Unos kriterija

Korisnik tj. donositelj odluke najprije unosi kriterij po kriterij, pri čemu za svaki kriterij može unijeti donji i gornji prag za eliminaciju ponuda po tom kriteriju. Prozor za unos kriterija prikazan je na sljedećoj slici:

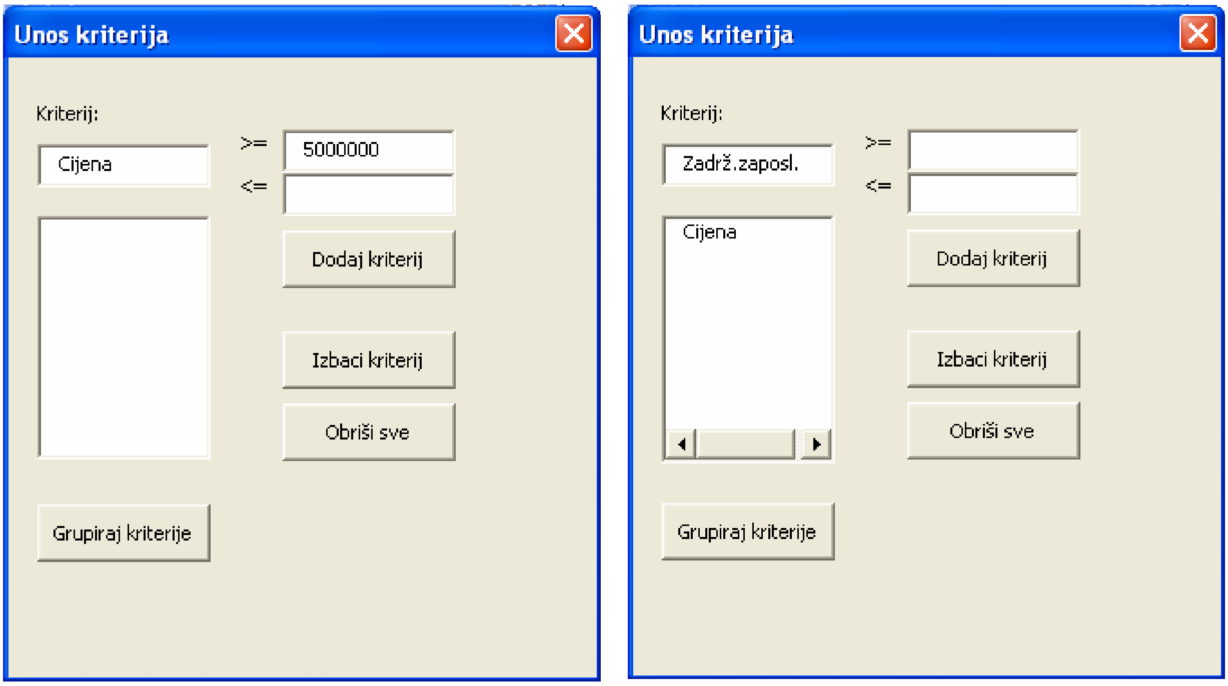


Slika 9.6 DSS – unos kriterija

Sa „>=" je označeno polje gdje se unosi donji prag, dok je sa „<=" označeno polje za unos gornjeg praga na vrijednosti kriterija koji se trenutno unosi. Ukoliko se ništa ne upiše za donji i gornji prag, smatra se da nema donjih i gornjih ograničenja za kriterij.

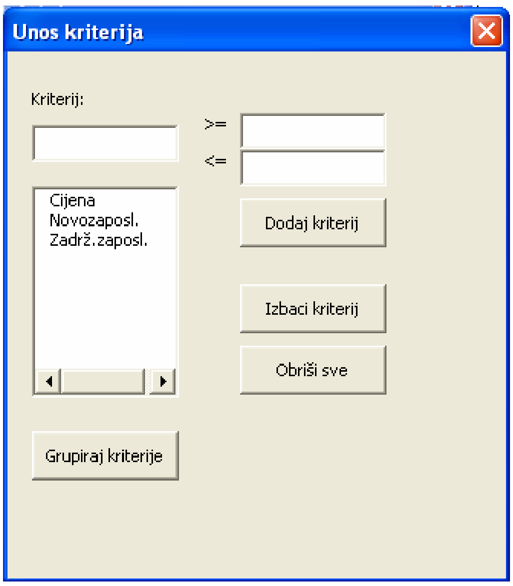
Pritiskom na „Dodaj kriterij" dodaje kriterij u listu kriterija po kojima će se vršiti evaluacija ponuda, dok pritiskom na „Izbaci kriterij" briše trenutno odabrani kriterij sa liste kriterija. Također, postoji i opcija „Obriši sve" kojom se može isprazniti cijela lista unesenih kriterija.

U primjeru fiktivnog natječaja za privatizaciju koji se ovdje koristi, polja na ovome prozoru bi se popunjavala na sljedeći način:



Slika 9.7 DSS – unos kriterija

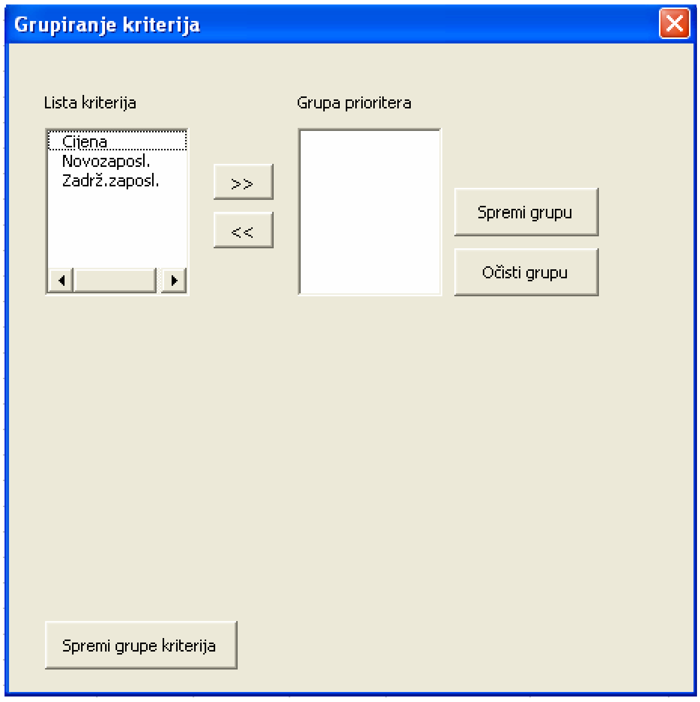
Tako je korisnik najprije unio cijenu kao kriterij pri čemu je postavio donji prag na cijenu od 5.000.000,00 kn, što znači da cijena ne smije biti manja od tog iznosa. Zatim je unio ostale kriterije po kojima će se ponude vrednovati kao što je to prikazano na slici dolje, te pristupa grupiranju kriterija obzirom na njihovu važnost pritiskom na gumb „Grupiraj kriterije“.



Slika 9.8 DSS – Unos kriterija

9.3.3 Grupiranje kriterija po važnosti

Na sljedećoj slici prikazan je prozor za grupiranje kriterija po važnosti.

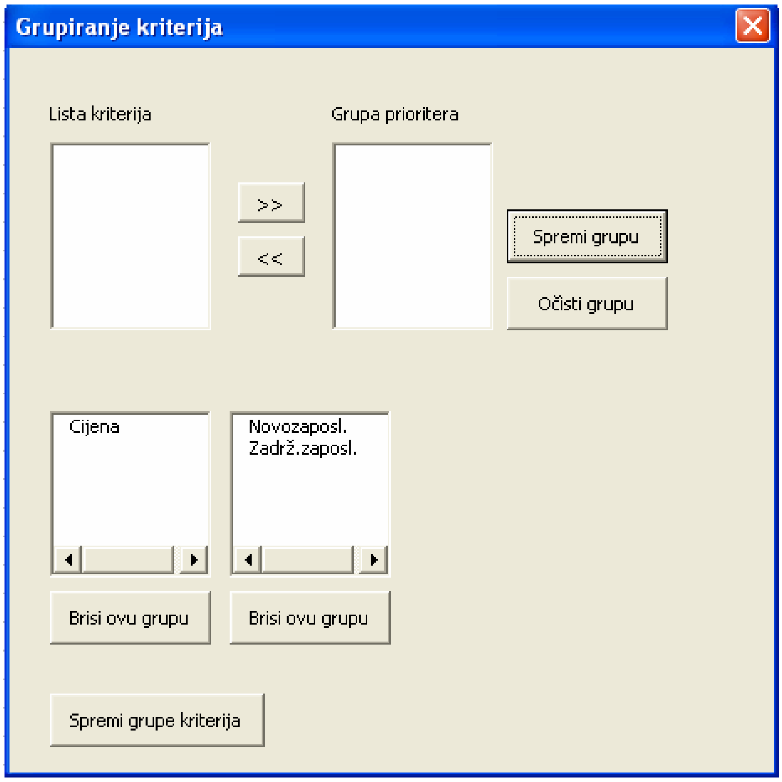


Slika 9.9 DSS – grupiranje kriterija po važnosti

Grupa kriterija se kreira prebacivanjem željenih kriterija sa lijevog na desno polje popisa, nakon čega se grupa sprema pritiskom na „Spremi grupu“ nakon čega se može kreirati nova grupa ukoliko ima još kriterija na početnom popisu sa lijeve strane. Pritiskom na „Očisti grupu“ svi kriteriji iz trenutne grupe se vraćaju na početnu listu kriterija i grupa se može nanovo kreirati.

Prva grupa kriterija koja se kreira smatra se grupom kriterija sa najvećom važnošću, slijedi grupa sa kriterijima koji su sljedeći po važnosti i tako dalje. Kriteriji koji su u istoj grupi smatraju se podjednako važnima.

Na prijašnjem primjeru fiktivnog natječaja za privatizaciju kriteriji su bili grupirani na sljedeći način:

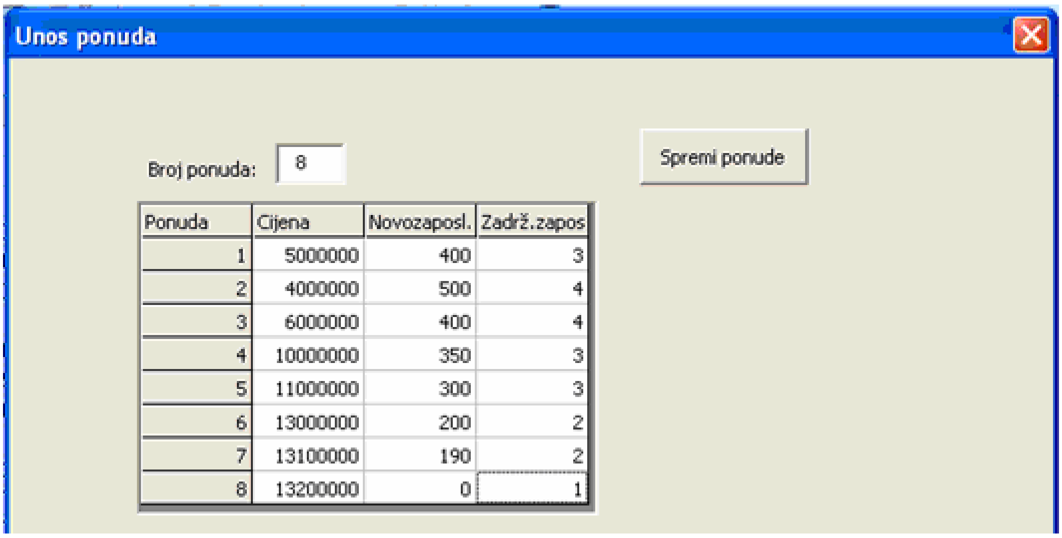


Slika 9.10 DSS – grupiranje kriterija po važnosti

Nakon što su svi kriteriji smješteni u grupe, kreirano grupiranje koje čini poredak kriterija po važnosti se sprema pritiskom na gumb „Spremi grupe kriterija" što korisnika vraća na početni prozor programa gdje se tada odabire opcija za unos ponuda za privatizaciju.

9.3.4 Unos ponuda

Ponude se unose tako da se najprije specificira broj ponuda koji će biti unesen, a zatim se jednostavno unose vrijednosti kriterija svake ponude kao što je to prikazano na sljedećoj slici:



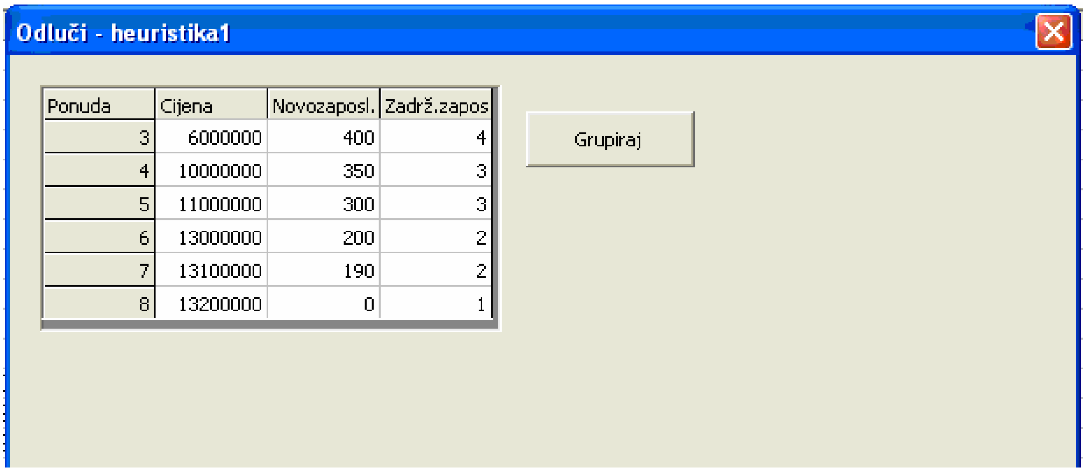
Ponuda	Cijena	Novozaposl.	Zadrž.zapos
1	5000000	400	3
2	4000000	500	4
3	6000000	400	4
4	10000000	350	3
5	11000000	300	3
6	13000000	200	2
7	13100000	190	2
8	13200000	0	1

Slika 9.11 DSS – unos ponuda

Kada se ponude spreme, može se pristupiti evaluaciji ponuda pomoću dvije heuristike koje su kreirane u ovome radu i zatim implementirane u ovome programskom sustavu.

9.3.5 Evaluacija ponuda pomoću heuristike 1

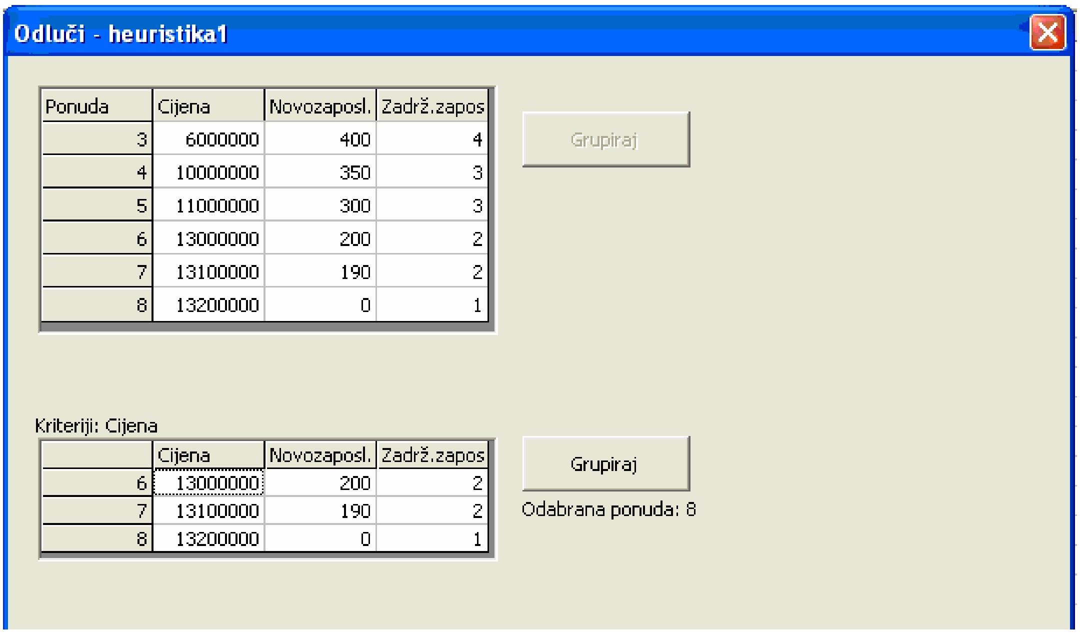
Pri evaluaciji ponuda pomoću prve heuristike koja je bila opisana najprije se prikazuje popis svih ponuda koje nisu eliminirane zbog nezadovoljavanja eliminacijskih kriterija ili zbog neefikasnosti.



Ponuda	Cijena	Novozaposl.	Zadrž.zapos
3	6000000	400	4
4	10000000	350	3
5	11000000	300	3
6	13000000	200	2
7	13100000	190	2
8	13200000	0	1

Slika 9.12 DSS – evaluacija ponuda pomoću heuristike 1

Pritiskom na gumb sa natpisom „Grupiraj" vrši se grupiranje obzirom na prvu grupu kriterija, te se prikazuje grupa ponuda koje su svrstane u najbolju grupu i ostaju za daljnje razmatranje:



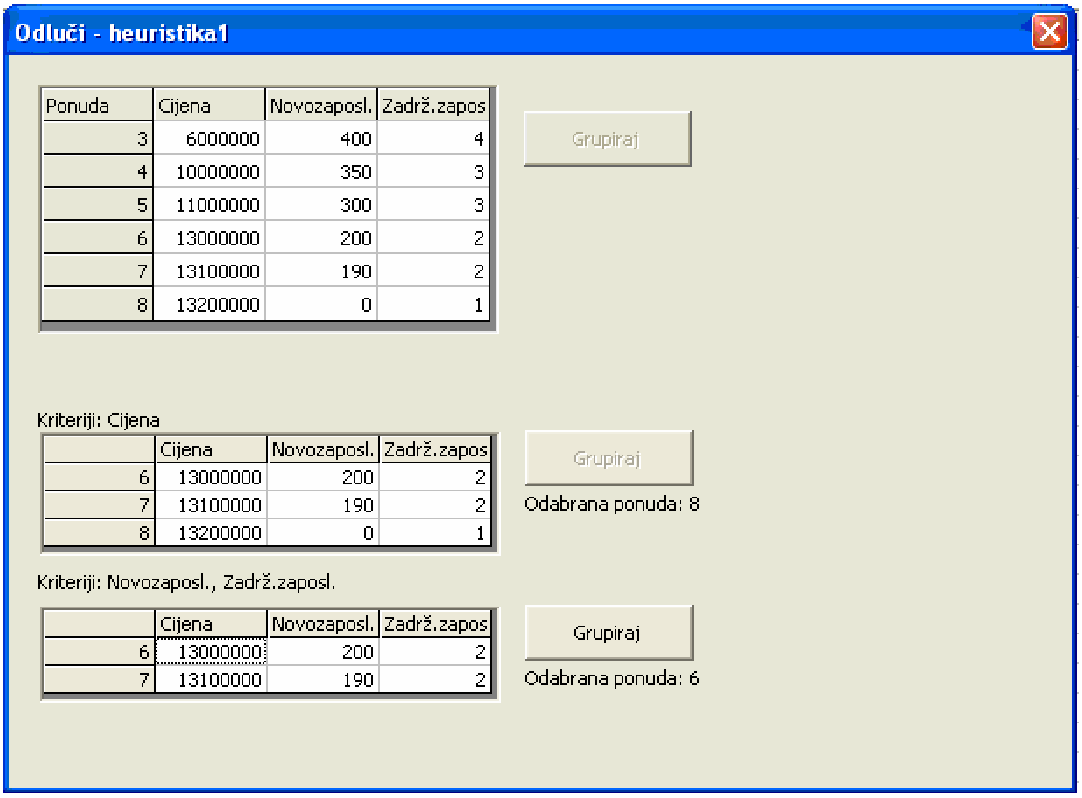
Ponuda	Cijena	Novozaposl.	Zadrž.zapos
3	6000000	400	4
4	10000000	350	3
5	11000000	300	3
6	13000000	200	2
7	13100000	190	2
8	13200000	0	1

	Cijena	Novozaposl.	Zadrž.zapos
6	13000000	200	2
7	13100000	190	2
8	13200000	0	1

Slika 9.13 DSS – evaluacija ponuda pomoću heuristike 1

Pritiskom na novi gumb „Grupiraj“ vrši se grupiranje obzirom na drugu grupu kriterija, te se prikazuje grupa ponuda koje su svrstane u sada najbolju grupu i ostaju za daljnje razmatranje i tako dalje. Također, navedena je ponuda koja je najbliža idealnoj točki obzirom na kriterije koji se u tekućem grupiranju promatraju.

U primjeru koji se ovdje prikazuje je druga grupa kriterija ujedno i posljednja, te se time postupak završava.

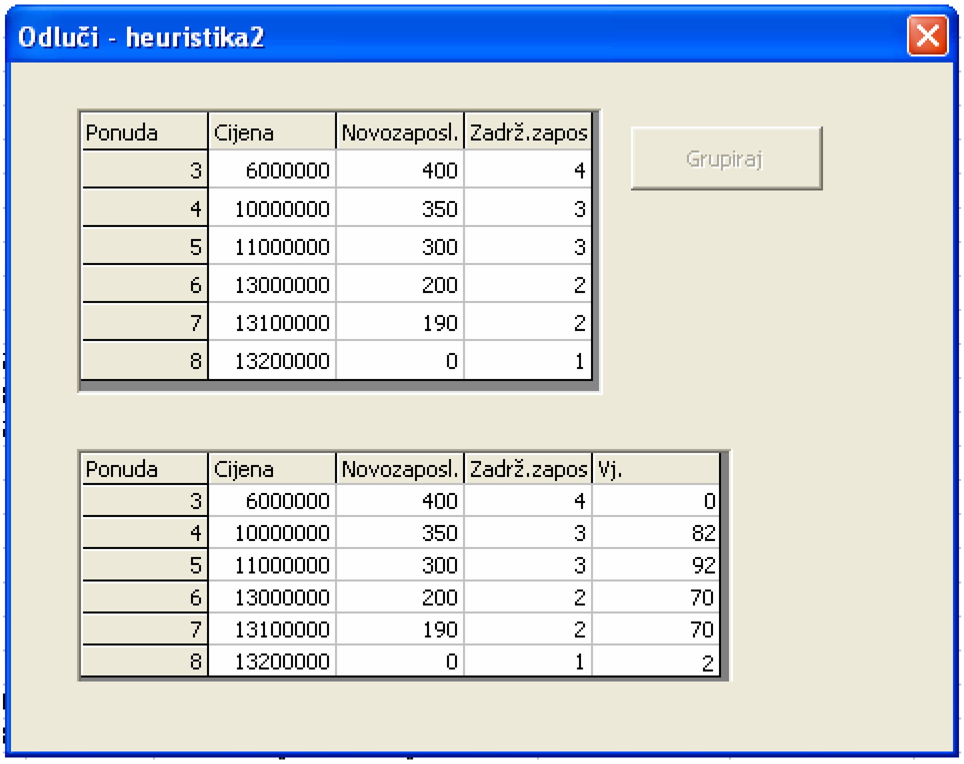


Slika 9.14 DSS – evaluacija ponuda pomoću heuristike 1

9.3.6 Evaluacija ponuda pomoću heuristike 2

Pri evaluaciji ponuda pomoću druge heuristike koja je bila opisana, također se prikazuje popis svih ponuda koje nisu eliminirane zbog nezadovoljavanja eliminacijskih kriterija ili zbog neefikasnosti.

Zatim se pritiskom na gumb „Grupiraj“ izračunavaju i prikazuju vjerojatnosti pripadnosti ponude najboljoj grupi ponuda obzirom na sve kriterije istovremeno na način kako je to bilo objašnjeno prilikom opisivanja druge heuristike koja se kreirala u ovom radu.



Ponuda	Cijena	Novozaposl.	Zadrž.zapos
3	6000000	400	4
4	10000000	350	3
5	11000000	300	3
6	13000000	200	2
7	13100000	190	2
8	13200000	0	1

Ponuda	Cijena	Novozaposl.	Zadrž.zapos	Vj.
3	6000000	400	4	0
4	10000000	350	3	82
5	11000000	300	3	92
6	13000000	200	2	70
7	13100000	190	2	70
8	13200000	0	1	2

Slika 9.15 DSS – evaluacija ponuda pomoću heuristike 2

Donositelj odluke koristeći ovaj sustav za potporu odlučivanju može brzo i jednostavno analizirati ponude pristigle na natječaj za privatizaciju razmatrajući rezultate dobivene uporabom jedne i druge heuristike. Pri tome ne treba precizno definirati težine kriterije, već je dovoljno poredati tj. grupirati kriterije obzirom na njihove važnosti.

Rezultati izvršavanja ovog programa na podacima fiktivnog natječaja za privatizaciju bili su korišteni prilikom opisivanja rada heuristika u prethodnim poglavljima.

9.4 Analiza osjetljivosti rješenja na promjenu ulaznih podataka

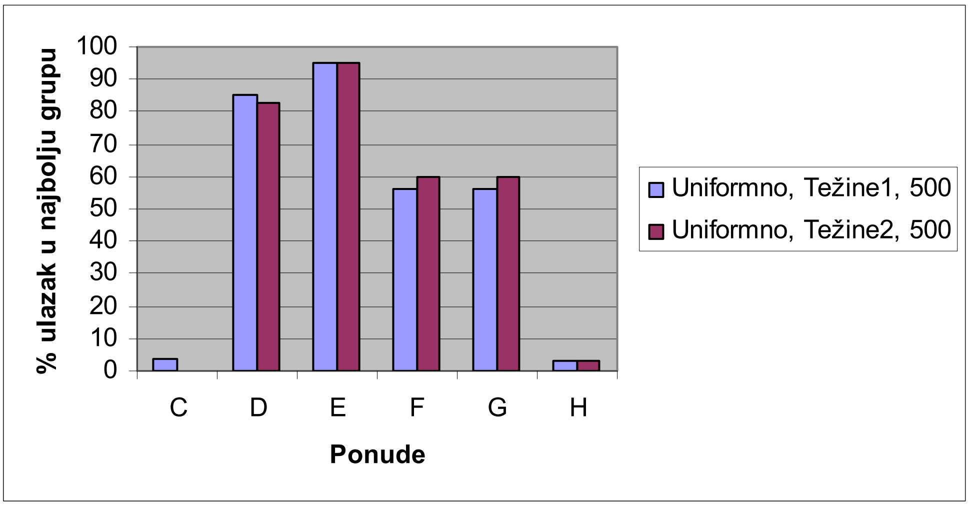
Općenito, analiza osjetljivosti promatra promjene rješenja nastale uslijed promjene ulaznih parametara. Analiza vezana za ovdje opisani model i heuristike može se podijeliti na dva segmenta. Prvi se odnosi na utjecaj promjena u vrijednostima pojedinih kriterija, te važnosti kriterija po grupama koje unosi korisnik programskog sustava kreiranog u ovome radu. Drugo, može se promatrati utjecaj načina generiranja težina kriterija unutar heuristike.

Utjecaj promjene vrijednosti kriterija pojedinih ponuda na evaluaciju ponuda za privatizaciju korisnik može razmotriti jednostavnim povratkom na prozor sustava za potporu odlučivanju na kojem se unose vrijednosti kriterija i promjenom željenih vrijednosti. Slično, utjecaj promjena u grupiranju kriterija po važnosti donositelj odluke može razmotriti povratkom na prozor za grupiranje kriterija i odabirom drugačijeg grupiranja kriterija.

Nadalje, mogu se usporediti dva navedena algoritma za slučajno generiranje težina kriterija za evaluaciju ponuda koji se koriste u drugoj heuristici kreiranoj u ovome radu. Također, može se razmotriti i utjecaj razdiobe koja se koristi pri slučajnom generiranju težina. Broj ponavljanja grupiranja ponuda na osnovu kojeg se zaključuje o vjerojatnosti pripadnosti najboljoj grupi ponuda utječe na konačnu evaluaciju ponuda, te se stoga mogu promatrati i promjene u evaluaciji ponuda uslijed promjene broja ponavljanja.

Na svim grafovima koji će ovdje biti prikazani oznake su kako slijedi: razdioba, algoritam za generiranje težina, broj ponavljanja. Tablice korištene u prikazu grafova su priložene uz rad.

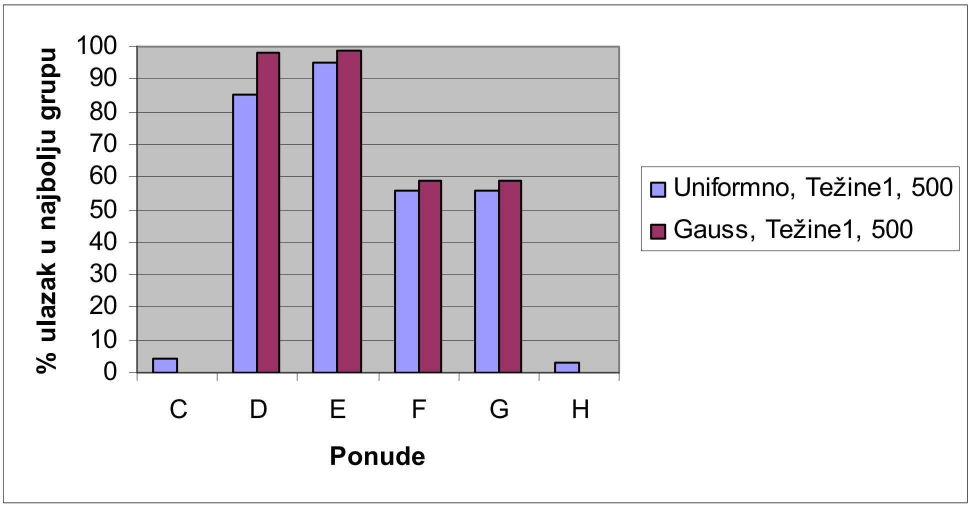
Na sljedećem grafu prikazana je usporedba evaluacija ponuda korištenjem druge heuristike pri čemu se u jednom slučaju koristi prvi algoritam za slučajno generiranje težina kriterija, dok se u drugom slučaju koristi drugi algoritam za generiranje težina kriterija.



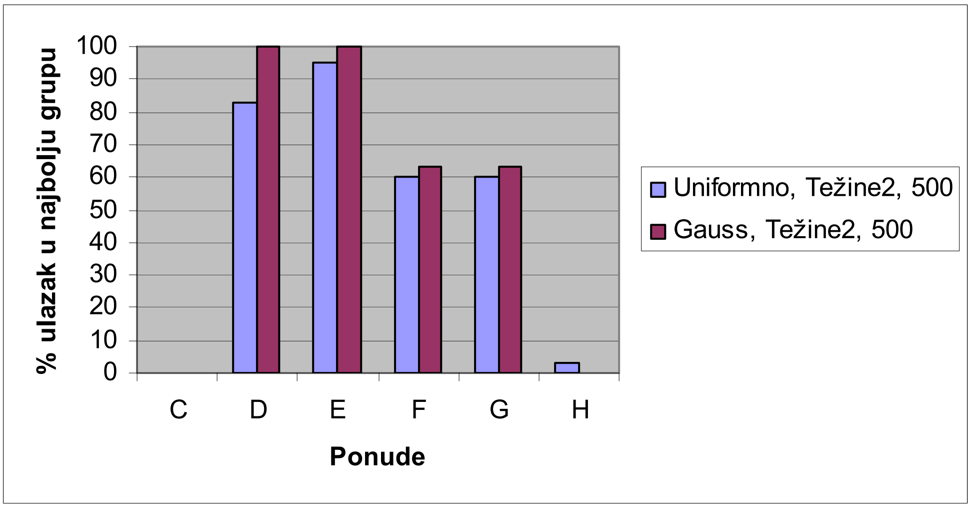
Slika 9.16 Usporedba evaluacija uz korištenje algoritama Težine1 i Težine2

Ukoliko se pri slučajnom generiranju težina koristi uniforma raspodjela, mogu se primijetiti razlike u vjerojatnostima ulaska u najbolju grupu ponuda nastale uslijed razlika u generiranju težina različitim algoritmima. Naime, prvi algoritam za generiranje težina kriterijima dodjeljuju težine tako da kriteriji koji su manje važnosti imaju manje težine što je također slučaj i u drugom algoritmu, no drugi algoritam za dodjelu težina kriterijima dodjeljuje težine tako da važniji kriteriji imaju veće težine od cijele grupe manje važnih kriterija.

Usporedba evaluacija ponuda obzirom na distribuciju vjerojatnosti (uniformna i normalna) koja se koristi pri slučajnom generiranju težina za oba korištena algoritma za generiranje težina može se vidjeti na sljedećim grafovima:



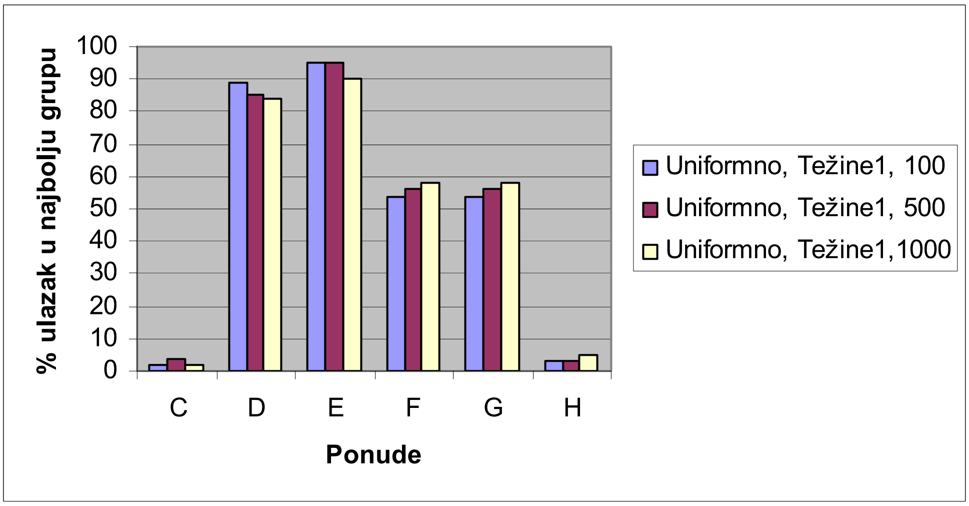
Slika 9.17 **Usporedba korištenja uniformne i Gaussove distribucije, Težine1**



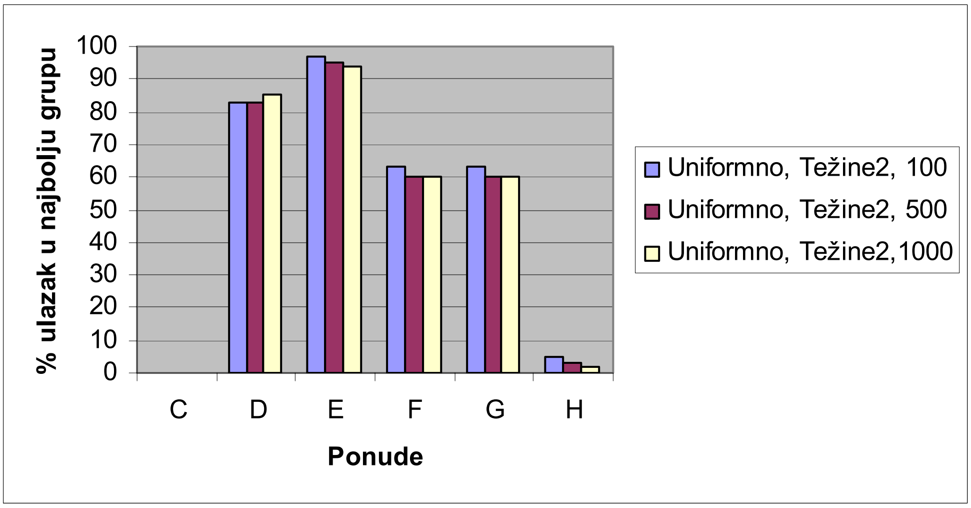
Slika 9.18 **Usporedba korištenja uniformne i Gaussove distribucije, Težine2**

Kako je i za očekivati, normalna ili Gaussova razdioba daje prednost srednjim slučajevima obzirom na ekstremne slučajeve što je vidljivo i iz prethodnih grafova. Na primjer, uz korištenje uniformne razdiobe za slučajno generiranje težine iz intervala od 0.5 do 1, svaki broj ima podjednaku vjerojatnost da bude odabran, dok će u slučaju Gaussove distribucije vjerojatnosti, brojevi oko sredine intervala, dakle 0.75, biti odabrani sa većom vjerojatnošću od brojeva bliže granicama intervala. Tako, u tom slučaju, broj 0.76 ima veću vjerojatnost da bude odabran za težinu kriterija nego broj 0.55.

Promjene u evaluaciji ponuda uslijed promjene broja ponavljanja mogu se vidjeti na sljedećim grafovima uz uniformnu razdiobu težina:



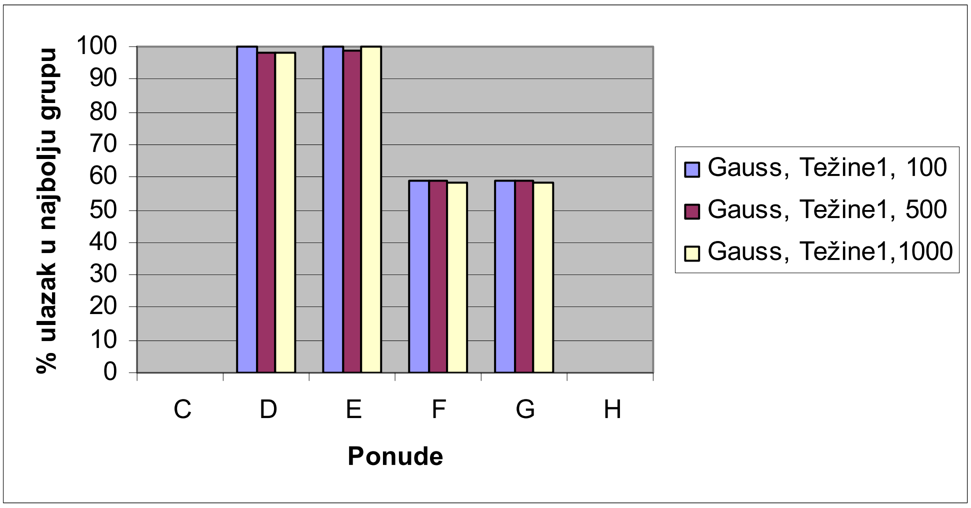
Slika 9.19 **Usporedba uz različit broj ponavljanja i Težine1, uniformna distribucija**



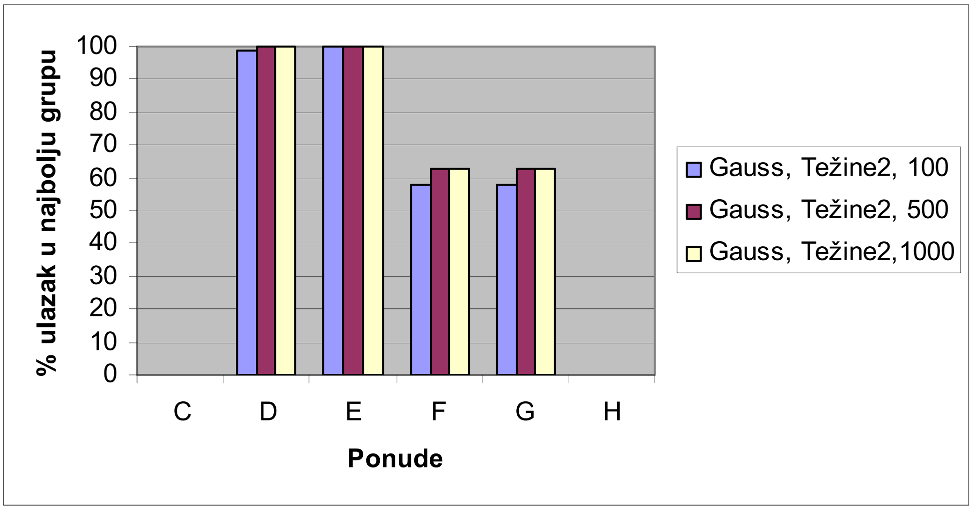
Slika 9.20 **Usporedba uz različit broj ponavljanja i Težine2, uniformna distribucija**

Iako razlike u vjerojatnostima uz korištenje različitog broja ponavljanja unutar Monte Carlo simulacija postoje, ovdje one nisu toliko značajne u smislu da bi donositelj odluke drugačije razmatrao odluku uz različit broj ponavljanja. Međutim, preporučljivo je koristiti što veći broj ponavljanja jer se na taj način povećava preciznost evaluacije.

Uz normalnu razdiobu težina promjene u evaluaciji ponuda uslijed promjene broja ponavljanja su slične te se vide na sljedećim grafovima:



Slika 9.21 Usporedba uz različit broj ponavljanja i Težine1, Gaussova distribucija



Slika 9.22 Usporedba uz različit broj ponavljanja i Težine2, Gaussova distribucija

Može se primijetiti ono da je program koji koristi Gaussovu razdiobu u ovom slučaju bio manje osjetljiv na broj ponavljanja, što je i očekivano. No, postavlja se pitanje koju je distribuciju bolje koristiti. Naime, možda bi se ipak podjednaka vjerojatnost trebala dati svim težinama koje odgovaraju poretku kriterija po važnosti.

Usprkos činjenici da male razlike u evaluaciji ponuda korištenjem druge heuristike uz promjene gore navedenih parametara definiranih unutar heuristike postoje, donositelj odluke dobiva dovoljno preciznu informaciju o kakvoći pojedine ponude. Naime, razlike u evaluacijama nisu se pokazale dovoljno velikima da bi bitno utjecale na vjerojatnosti pripadnosti najboljoj grupi, te ne bi značajno utjecale na sud donositelja odluke o kakvoći ponuda.

Prva heuristika kreirana u ovome radu ne koristi algoritme za slučajno generiranje težina, te nije podložna utjecajima gore navedenih parametara. Međutim, kako je već rečeno, ona je puno rigidnija i preporuča se korištenje u kombinaciji sa drugom, fleksibilnijom heuristikom.

10 Usporedba rješenja dobivenih kreiranim programskim sustavom za potporu odlučivanju sa rješenjima dobivenim AHP metodom i programom ExpertChoice

AHP metoda (Analitički Hijerarhijski Proces) je često korištena metoda za rješavanje problema višekriterijskog odlučivanja. Program za potporu odlučivanju ExpertChoice se bazira na AHP metodi te će ovdje biti dana usporedba programa kreiranog u ovome radu sa softverskim paketom ExpertChoice čija se demo verzija može skinuti sa www.expertchoice.com. AHP metodu osmislio je Thomas Saaty [14] u cilju pomaganja donositeljima odluke pri odlučivanju, pri čemu se važnosti kriterija i vrijednosti alternativa uspoređuju po parovima.

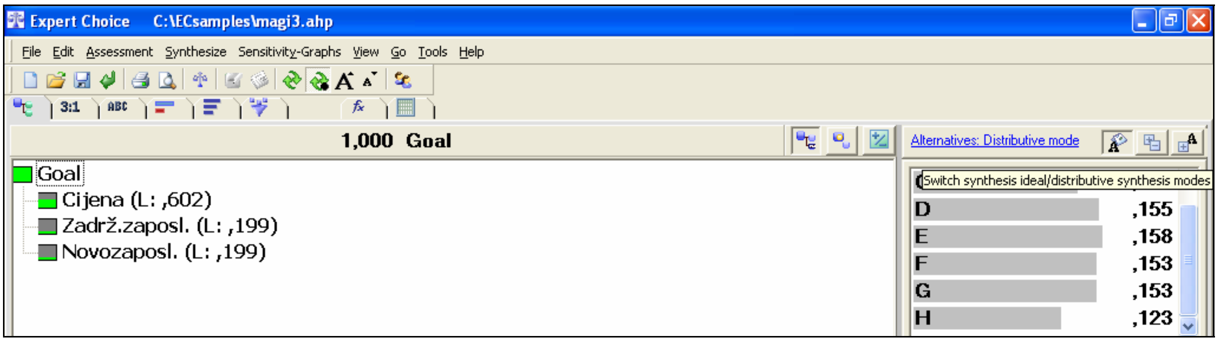
Međutim, da bi se program ExpertChoice mogao usporediti sa programom kreiranim u ovom radu, bilo je potrebno izbjeći usporedbe alternativa po parovima. Stoga su umjesto toga, korišteni direktni načini unosa podataka u program, te su se upisivale normalizirane vrijednosti alternativa po kriterijima. Ponovno se koristi prije navedeni fiktivni natječaj za privatizaciju.

U programu ExpertChoice se zahtijeva postavljanje konkretnih težina kriterija, iako se to radi usporedbom kriterija po parovima. Kako druga heuristika opisana u ovom radu radi više simulacija uz različite težine koje prate poredak kriterija i daje vjerojatnosti pripadnosti najboljoj grupi, nije moguće usporediti krajnji ishod izvršavanja te heuristike sa rezultatom izvršavanja programa ExpertChoice.

Međutim, moguće je usporediti rezultat jednog ponavljanja druge heuristike opisane u ovome radu sa rezultatima izvršavanja programa ExpertChoice jer se u tom programu zahtijeva postavljanje težina kriterija, iako se to radi usporedbom kriterija po parovima. Naravno, uslijed razlika u metodama, za očekivati je da će se rezultati razlikovati.

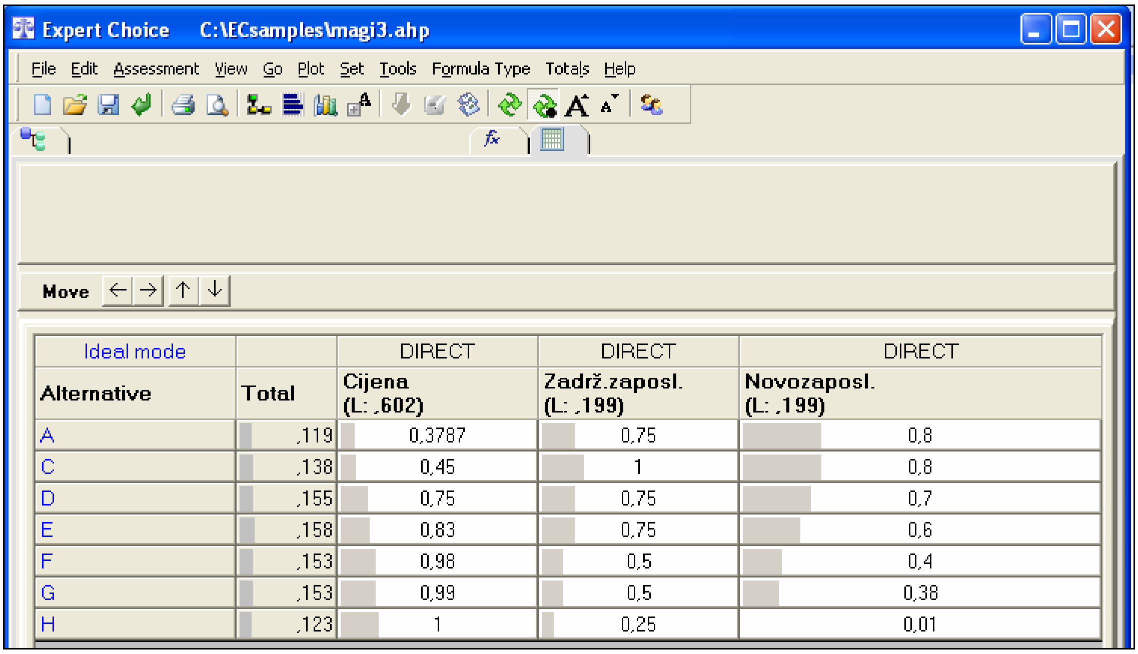
U slučaju kada su težine u jednom ponavljanju druge heuristike za fiktivni natječaj za privatizaciju imale vrijednosti $w_1 = 0.6$, $w_2 = 0.2$, $w_3 = 0.2$, ponuda koja je bila najbliža idealnoj ponudi bila je ponuda E. Najbolju grupu nastalu grupiranjem ponuda u tom ponavljanju tj. iteraciji činile su ponude E i D.

Na sljedećoj slici prikazan je prozor programa ExpertChoice u kojem se vidljivi uneseni kriteriji i njihove težine za gore navedeni slučaj.



Slika 10.1 ExpertChoice – unos podataka

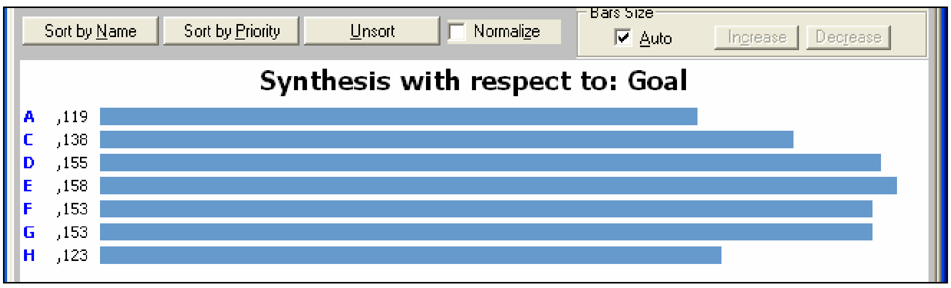
Unos normaliziranih vrijednosti kriterija u program ExpertChoice za fiktivne ponude prikazan je na sljedećoj slici:



Ideal mode		DIRECT	DIRECT	DIRECT
Alternative	Total	Cijena (L: ,602)	Zadrž.zaposl. (L: ,199)	Novozaposl. (L: ,199)
A	,119	0,3787	0,75	0,8
C	,138	0,45	1	0,8
D	,155	0,75	0,75	0,7
E	,158	0,83	0,75	0,6
F	,153	0,98	0,5	0,4
G	,153	0,99	0,5	0,38
H	,123	1	0,25	0,01

Slika 10.2 ExpertChoice – unos normaliziranih vrijednosti kriterija

Ocjena kakvoće ponuda može se vidjeti na sljedećoj slici:

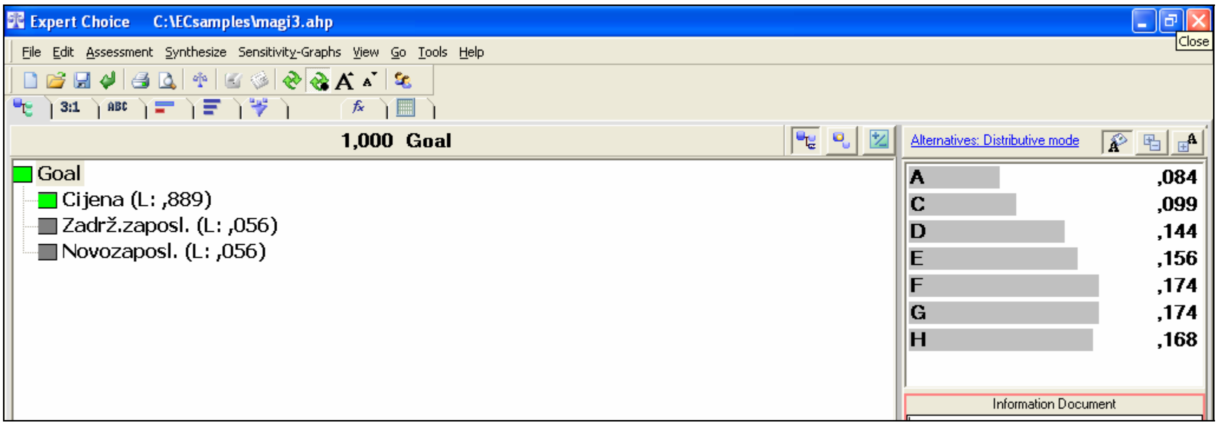


Slika 10.3 ExpertChoice – ocjena kakvoće ponuda

Vidi se da su i ovdje ponude D i E dobile najveće ocjene, kao i u heuristici koja je ovdje kreirana. Međutim, treba imati na umu da je to tek jedno izvršavanje heuristike uz fiksirane težine.

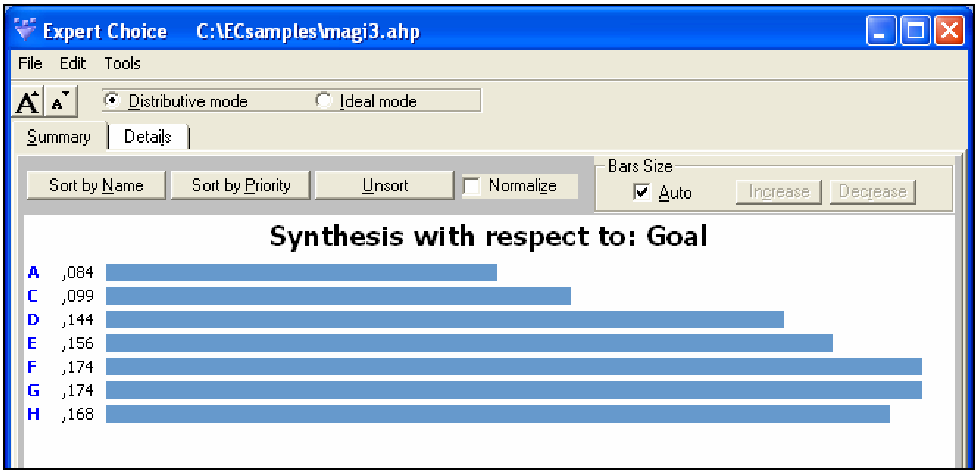
Kada težine kriterija u izvršavanju heuristike poprimaju vrijednosti $w_1 = 0.9$, $w_2 = 0.05$, i $w_3 = 0.05$, ponuda koja bi bila najbliža idealnoj ponudi bila bi ponuda F. Ponude E, F i D bi u tom slučaju činile najbolju grupu nastalu grupiranjem u tom izvršavanju.

Na sljedećoj slici prikazan je prozor programa ExpertChoice u kojem se vidljivi uneseni kriteriji i njihove težine za ovako postavljene težine kriterija.



Slika 10.4 ExpertChoice – unos podataka

Ocjena kakvoće ponuda u ovom slučaju može se vidjeti na sljedećoj slici:



Slika 10.5 ExpertChoice – ocjena kakvoće ponuda

Prve tri ponude po visini ocjene su ovdje ponude E, F i H. Dakle, u ovome slučaju postoje razliku u rezultatima heuristike i programa ExpertChoice. Razlika nestaje uslijed korištenja kompromisnog programiranja u heuristici, za razliku od programa ExpertChoice koji ga ne koristi.

Može se zaključiti da je prednost programa ExpertChoice upravo činjenica da se kriteriji i vrijednosti alternativa po kriterijima mogu uspoređivati čime se donositelju odluke pruža više slobode. No, s druge strane, to je ujedno i mana jer zahtijeva previše vremena i previše preciznih informacija od donositelja odluke jer je upitno u kojoj mjeri on zapravo može reći da je neki kriterij duplo važniji od drugog ili možda 2.1 puta važniji od drugog.

Stoga je prednost programa kreiranog u ovome radu upravo to što ne zahtijeva previše informacija od donositelja odluke. Umjesto toga, pomoću druge heuristike se radi veliki broj simulacija sa različitim težinama kriterija koje ipak prate važnosti kriterija zadane od strane donositelja odluke njihovim grupiranjem. Ukoliko donositelj odluke želi rigidniji pristup, u smislu da uređaj među kriterijima smatra gotovo leksikografskim, koristiti prvu heuristiku kreiranu u ovome radu.

11 Zaključak

U današnje su vrijeme sustavi sve složeniji jer se temelje na velikom broju podataka, velikom broju opcija za djelovanje, a vrijeme u kojem se odluka mora donijeti sve je kraće. Zato treba djelovati efikasno, a to znači na osnovu raspoloživih podataka u kratkom roku predložiti dobro rješenje problema i donijeti efikasnu odluku. Uz metode optimizacije koje operacijska istraživanja koriste, kao i uz modernu računalnu programsku potporu to je djelovanje kvalitetnije.

U problemima odlučivanja uz postojanje više kriterija koriste se metode višekriterijske optimizacije koriste koje se mogu primijeniti i na problem odabira najuspješnije ponude u procesu privatizacije. U odabiru najuspješnije ponude, ključni kriterij nije samo ponuđena cijena nego se uzimaju u obzir i drugi kriteriji kao što su daljnje investicije, te preuzimanje obaveza nad zaposlenicima. Kako konačna odluka uvelike ovisi o važnostima tj. težinama pojedinih kriterija donositelj odluke mora odrediti važnost svakog kriterija.

Postojeća programska rješenja uglavnom su dio nekog globalnog rješenja i nisu prilagođena specifičnostima koje se u praksi javljaju. Da bi se povećala efikasnost zainteresiranih donositelja odluke, potrebno je razvijati metode i postupke koji će u rješavanju uzimati u obzir i te specifičnosti, te prilikom izrade takvih metoda treba voditi računa i o jednostavnosti korištenja programskih rješenja u koja su takve metode inkorporirane.

U ovome radu osmišljene su dvije heuristike za višekriterijsko odlučivanje koje su potom implementirane u vidu sustava za programsku potporu odlučivanju. Pokazana je primjena ovih heuristika na problem valorizacije i odabira najuspješnije ponude u procesu privatizacije. Pri tome programski sustav u kojem su heuristike implementirane zahtijeva od donositelja odluke minimalne informacije o važnostima kriterija. Time se omogućava lakše donošenje odluke jer se od donositelja odluke očekuje samo da kriterije rangira i grupira po važnosti, što također čini sustav za potporu odlučivanju jednostavnim za korištenje. Obzirom da se unutar heuristika ispituju različite mogućnosti pri evaluaciji ponuda uz generiranje različitih težina kriterija unutar heuristike, doprinosi se kvalitetnijem i bržem donošenju odluke.

11.1 Smjernice za buduća istraživanja

U hrvatskom se gospodarstvu kvantitativne metode optimizacije ne primjenjuju u dovoljnoj mjeri. Zato treba razvijati metode rješavanja na konkretnim problemima iz prakse i znanstvenim pristupom doprinositi njihovom rješavanju. Tako će razvijene heuristike uvelike i u kratkom roku povećati efikasnost donošenja odluka. Također, rješavanjem konkretnih problema i postizavanjem boljih rezultata od postojećih, povećava se povjerenje čimbenika iz prakse, koji su zapravo donositelji odluka, u korištenje takvih metoda.

Kako se problem višekriterijskog odlučivanja javlja u raznim djelatnostima, ovdje kreirane heuristike bi se mogle prilagoditi i primijeniti i na nekim drugim, sličnim problemima i na taj način dobiti programska podrška za njihovo rješavanje u vrlo kratkom roku.

Planira se i dodavanje novih specifičnosti, ciljeva i zahtjeva u programski kod, a radi dodatne prilagodbe specifičnostima koje se u praksi javljaju, te u svrhu poopćenja i mogućnosti korištenja podrške i kod nekih drugih, sličnih problema.

12 Literatura

[1] Alves M.J. i Climaco J. (2007). A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research*, 180, str. 99-115.

[2] Brauers W.K. (1995). PRIVATA: A model for privatization with multiple non-transitive objectives. *Public Choice*, 85, str. 353-370

[3] Caballero, R et al. (2007). Solving a multiobjective location routing problem with a metaheuristic based on tabu search: Application to a real case in Andalusia. *European Journal of Operational Research*, 127(3), str. 1751-1763.

[4] Choo, E. i Atkins, D. (1983). Proper efficiency in nonconvex multicriteria programming. *Mathematics of Operations Research*, 8(3), str. 467-470.

[5] Ehrgott, M. (2005). Multicriteria Optimization. 2nd ed., Berlin: Springer.

[6] Ehrgott, M. i Gandibleux X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4), str. 425-460.

[7] Evans, G.W. (1984). An Overview of Techniques for Solving Multiobjective Mathematical Programs. *Management Science*, 30(11), str. 1268-1282.

[8] Falkenauer E. i Marchand A. (2001). Using k-means? Consider arrayminer. *Proceedings of the 2001 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences* (METMBS'2001)

[9] Gamerman, D. (1997). Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference. Boca Raton, FL: CRC Press.

[10] Karaivanova, J.N., Narula, S.C. i Vassilev, V. (1993). An interactive procedure for multiple objective integer linear programming problems. *European Journal of Operational Research* 68, str. 344–351.

[11] Koopmans, T.C., ed. (1951). Activity Analysis of Production and Allocation. New York: John Wiley & Sons, Inc.

[12] Lewis H. i Papadimitriou C. (1998). Elements of the Theory of Computation, 2nd ed., New Jersey: Prentice Hall.

[13] Roy, B. (1996). Multicriteria Methodology for Decision Aiding. Dordrecht: Kluwer Academic Publishers.

[14] Saaty, T. L. (1980). The Analytic Hierarchy Process. New York: McGraw Hill Company.

[15] Shi J. i Malik J. (2000). Normalized Cuts and Image Segmentation, *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 22(8), str. 888-905.

[16] Steuer, R. (2000). ADBASE: A multiple objective linear programming solver. Technical report. Terry College of Business, University of Georgia, Athens, GA, 30602 USA.

[17] Steuer, R. i Choo, E. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26, str. 326-344.

[18] Steuer, R.E., Silverman, J. i Whisman A.W. (1993). A combined Tchebycheff/aspiration criterion vector interactive multiobjective programming procedure. *Management Science*, 39 (10), str. 1255–1260.

[19] Tervonen T. et al. (2007), A stochastic method for robustness analysis in sorting problems. *European Journal of Operational Research*. U objavljivanju.

[20] Tervonen T. i Lahdelma R. (2007). Implementing stochastic multicriteria acceptability analysis. *European Journal of Operational Research*, 178, str. 500–513.

[21] V. Vassilev, K. Genova, M. Vassileva i S. Narula (2003). Classification-Based Method of Linear Multicriteria Optimization. *International Journal on Information Theories and Aplications,* 10(3), str. 266-270.

[22] Wierzbicki, A.P. (1982). A mathematical basis for satisficing decision making. *Mathematical Modelling,* 3, str. 391–405.

13 Prilozi

13.1 Potrebni matematički pojmovi i alati

13.1.1 Relacija poretka

U skupu realnih brojeva za svaka dva broja možemo reći jesu li jednaki ili je jedan veći od drugoga. Međutim, u prostoru $\mathbf{R}^k$ mogu postojati parovi vektora koji su međusobno neusporedivi. Na primjer, možemo reći da je (1,1) manje od (3,3) jer je svaka komponenta prvog vektora manja od odgovarajuće komponente drugog, ali ne znamo kako usporediti (1,3) i (3,1) jer je prva komponenta prvog vektora manja od odgovarajuće iz drugog, no druga komponenta prvog vektora je veća od odgovarajuće komponente drugog vektora. Da bismo ipak mogli uvesti uređaj u skupu $\mathbf{R}^k$, potrebno je definirati neke osnovne pojmove.

Definicija. *Relaciju poretka* „≤“ definiramo na sljedeći način:

$$\forall z^1, z^2 \in \mathbf{R}^k : z^1 \le z^2 \Leftrightarrow z_i^1 \le z_i^2,\ i = 1,2,\ldots,k$$
$$z^1 = z^2 \Leftrightarrow z_i^1 = z_i^2,\ i = 1,2,\ldots,k$$

Podsjetimo se još nekih definicija.

Definicija. *Relacija parcijalnog uređaja* „≤“ na skupu $Z \subseteq \mathbf{R}^k$ je binarna relacija na skupu Z koja je:

1) refleksivna

   $z \le z$

2) antisimetrična

   $z^1 \le z^2, z^2 \le z^1 \Rightarrow z^1 = z^2$

3) tranzitivna

   $z^1 \le z^2, z^2 \le z^3 \Rightarrow z^1 \le z^3$

Definicija. Neka je „≤“ relacija parcijalnog uređaja. Ako su dodatno, svaka dva elementa skupa Z u relaciji, odnosno za svaki $z^1, z^2 \in \mathbf{R}^k$ vrijedi $z^1 \le z^2$ ili $z^2 \le z^1$, tada kažemo da je ≤ *relacija totalnog* ili *potpunog uređaja*, a Z je *potpuno uređen skup*.

Skup koji nije potpuno uređen općenito neće imati maksimalni (minimalni) element koji je nužno jedinstven, već može imati više maksimalnih (minimalnih) elemenata. Podsjetimo se definicije maksimalnog elementa parcijalno uređenog skupa. Maksimalni element z^M skupa Z ima svojstvo

$$z^M \in Z, z \in Z \; z \geq z^M \Rightarrow z = z^M$$

ili

$$z_i^M \geq z_i \text{ za svaki } i = 1, \ldots, k \, .$$

13.1.2 Rastuća i strogo rastuća funkcija

Definicija. Za funkciju $f : \mathbf{I} \to \mathbf{R}$, gdje je $\mathbf{I} \subseteq \mathbf{R}$, kažemo da je rastuća na intervalu $\mathbf{I}$ ako $\forall x_1, x_2 \in \mathbf{I},\; x_1 \leq x_2 \Rightarrow f(x_1) \leq f(x_2)$.

Definicija. Za funkciju $f : \mathbf{I} \to \mathbf{R}$, gdje je $\mathbf{I} \subseteq \mathbf{R}$, kažemo da je strogo rastuća na intervalu $\mathbf{I}$ ako $\forall x_1, x_2 \in \mathbf{I},\; x_1 < x_2 \Rightarrow f(x_1) < f(x_2)$.

13.1.3 Konveksnost

Definicija. Za skup $S \subseteq \mathbf{R}^k$ kažemo da je *konveksan* skup ako vrijedi da je $\forall x^1, x^2 \in S, \alpha \in \mathbf{R}, 0 \leq \alpha \leq 1 \Rightarrow \alpha x^1 + (1-\alpha)x^2 \in S$.

Definicija. Za funkciju $f : S \to \mathbf{R}$, gdje je $S \subseteq \mathbf{R}^k$ konveksan skup, kažemo da je *konveksna* ako $\forall x^1, x^2 \in S, \alpha \in \mathbf{R}, 0 \leq \alpha \leq 1 \Rightarrow f\left((1-\alpha)x^1 + \alpha x^2\right) \leq \alpha f\left(x^1\right) + (1-\alpha) f\left(x^2\right)$

13.1.4 Množenje skupa skalarom

Neka je $S \subseteq \mathbf{R}^k$, $\alpha \in R$. Množenje skupa S skalarom α definira se na sljedeći način:

$$\alpha S = \{\alpha s : s \in S\}.$$

Specijalno: $-S = \{-s : s \in S\}$.

13.1.5 Zbrajanje skupova

Neka su $S_1, S_2 \subseteq \mathbf{R}^k$. Zbroj skupova S_1 i S_2 definira se na sljedeći način:

$$S_1 + S_2 = \{s_1 + s_2 : s_1, s_2 \in S\}.$$

Ako je $S_1 = \{s\}$, može se pisati i $s + S_2$ umjesto $S_1 + S_2$.

Dakle, ako je $S_1 = \{s\}$, $S_1 + S_2 = s + S_2 = \{s + s_2 : s_2 \in S\}$.

13.1.6 Složenost algoritama

Složenost algoritama mjeri vrijeme potrebno za rješavanje nekog problema koristeći određeni algoritam tj. težinu problema koji se rješava. Navest ćemo osnovne pojmove vezane za vrijeme izvršavanja algoritama.

Algoritme možemo podijeliti u dvije skupine obzirom na način dobivanja rješenja.

Definicija. *Deterministički algoritam* je algoritam koji će pri svakom izvršavanju u bilo kojim uvjetima od istog unosa doći do istog izlaza slijedeći svaki put identičan niz naredbi.

Definicija. *Nedeterministički algoritam* je algoritam koji može odabrati jedno rješenje te provjeriti zadovoljava li to rješenje uvjete problema koji algoritam rješava. Nedeterministički algoritam završava neuspješno onda i samo onda kada ne postoje izbori koji vode do uspjeha, dakle kada rješenje problema uopće ne postoji.

Složenost algoritma mjeri se brojem operacija potrebnih da se nađe rješenje problema odlučivanja gdje problemom odlučivanja smatramo svaki problem u kojem je potrebno odgovoriti je li nešto rješenje problema ili nije.

Definirajmo preciznije problem odlučivanja, te algoritam odlučivanja.

Definicija. *Problem odlučivanja* je problem koji daje kao izlaz 0 ili 1 (laž ili istina). *Algoritam odlučivanja* je algoritam koji rješava problem odlučivanja. *Problem optimizacije* je problem koji zahtijeva određivanje optimalne vrijednosti (minimalne ili maksimalne) zadane funkcije cilja. *Algoritam optimizacije* je algoritam koji rješava problem optimizacije.

Algoritmi odlučivanja su u bliskoj vezi s algoritmima optimizacije. Naime, svaki algoritam optimizacije ima svoj pripadni algoritam odlučivanja. Ako se problem odlučivanja ne može riješiti determinističkim algoritmom u polinomijalnom vremenu, tada se ni odgovarajući optimizacijski problem ne može riješiti determinističkim algoritmom u polinomijalnom vremenu. Slično, ako se problem odlučivanja može riješiti determinističkim algoritmom u polinomijalnom vremenu, tada se i odgovarajući optimizacijski problem može riješiti determinističkim algoritmom u polinomijalnom vremenu. Dakle, dovoljno je promatrati složenost algoritama odlučivanja.

Definicija. Kažemo da algoritam ima *složenost* $O(g(n))$ ako postoji konstanta c takva da je broj operacija algoritma najviše $cg(n)$ za sve moguće ulazne podatke problema odlučivanja, gdje je g neka funkcija, a n veličina ulaznih podataka.

Na primjer, ako neki algoritam ima složenost $O(g(n))$, gdje je $g(n)$ eksponencijalna funkcija veličine ulaznih podataka, kažemo da je algoritam eksponencijalan.

Definicija. Kažemo da je problem odlučivanja rješiv u polinomijalnom vremenu tj. da pripada klasi problema P ako postoji deterministički algoritam koji rješava problem u $O(p(n))$ operacija gdje je *p* polinom stupnja *n*. Za algoritam odlučivanja kažemo da je polinomijalni ako postoji polinom *p* takav da je vrijeme izvršavanja algoritma $O(p(n))$.

Definicija. Kažemo da problem odlučivanja pripada klasi problema NP ako postoji nedeterministički algoritam koji ga rješava u polinomijalnom vremenu.

Da bismo rekli kada je neki problem NP-težak, potrebno je uvesti problem zadovoljivosti. No, za to je potrebno je definirati pojmove koji će se koristiti.

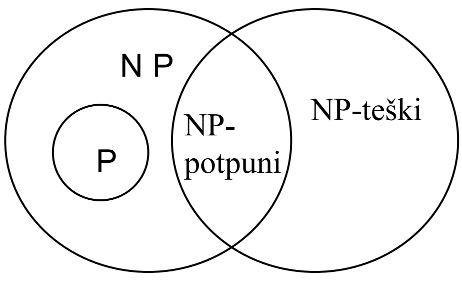
Propozicijski račun. Neka su $x_1, x_2, \ldots$ Booleovske varijable (varijable koje mogu poprimiti samo vrijednosti „istina“ ili „laž“. Neka je $\overline{x}_i$ negacija od x_i. Literal je ili varijabla ili negacija varijable. Formula u propozicijskom računu je izraz konstruiran od literala i operacija $\wedge$ (logičko „i“) i $\vee$ (logičko „ili“). Primjer: $(x_1 \wedge x_2) \vee (x_3 \wedge \overline{x}_4)$.

Problem zadovoljivosti glasi: treba odrediti mogu li se varijablama formule u propozicijskom računu pridružiti vrijednosti istina/laž tako da cijela formula bude istinita.

Definicija. Neka su L_1 i L_2 problemi. L_1 *se svodi* na L_2 (pišemo $L_1 \propto L_2$) ako postoji polinomijalni deterministički algoritam za L_1 koji poziva algoritam za L_2, s time da se svaki taj poziv broji kao jedan osnovni korak.

Definicija. Problem L je NP-težak ako se problem zadovoljivosti svodi na L. Problem L je NP-potpun ako je L NP-težak i L $\in$ NP.

Prema uobičajenom vjerovanju, odnos skupova P, NP NP-teških i NP-potpunih problema izgledao kao na slici:



Slika 13.1 Složenost algoritama

Dakle, vrijedi $\mathsf{P} \subset \mathsf{NP}$. Poznati neriješeni problem glasi: je li $\mathsf{P} = \mathsf{NP}$? Vjeruje se da je $\mathsf{P} \neq \mathsf{NP}$, no za to još nema dokaza. Međutim, postoje problemi u NP takvi da ako se pokaže da su oni u P, nužno će cijeli NP biti u P.

Također, optimizacijski problem može biti NP-težak, a da nije NP-potpun. Dakle, postoje NP-teški problemi koji nisu NP-potpuni.

13.1.7 Svojstvene vrijednosti i svojstveni vektori

Definicija. Skalar λ je *svojstvena vrijednost* matrice $\mathbf{A}_{n\times n}$ ako postoji vektor $x \in \mathbf{R}^n$ sa barem jednom komponentom različitom od nule takav da je $\mathbf{A}x = \lambda x$, a svaki takav vektor x naziva se *svojstvenim vektorom* matrice $\mathbf{A}$ pridružen svojstvenoj vrijednosti λ.

Definicija. Neka su $\mathbf{A}$ i $\mathbf{B}$ matrice formata $n \times n$ .Skalar λ naziva se generalizirana svojstvena vrijednost, a vektor $x \in \mathbf{R}^n$ sa barem jednom komponentom različitom od nule odgovarajućim generaliziranim svojstvenim vektorom para matrica $(\mathbf{A}, \mathbf{B})$ ako je $\mathbf{A}x = \lambda \mathbf{B}x$.

13.2 Popis slika

13.3 Popis tablica

Tablice korištene u prikazu grafova u 10. poglavlju:

Uniform, težine1

	10	100	200	300	500	700	1000
1	0	2	2	2	4	2	2
2	100	89	90	82	85	84	84
3	100	95	94	92	95	92	90
4	60	54	58	56	56	54	58
5	60	54	58	56	56	54	58
6	0	3	2	4	3	3	5

Uniform, težine2

	10	**100**	**200**	**300**	**500**	**700**	**1000**
1	0	0	0	0	0	0	0
2	90	83	83	81	83	84	85
3	100	97	86	95	95	94	94
4	60	63	59	61	60	59	60
5	60	63	59	61	60	59	60
6	0	5	3	2	3	3	2

Gauss, težine1

	10	**100**	**200**	**300**	**500**	**700**	**1000**
1	0	0	0	0	0	0	0
2	100	100	98	97	98	99	98
3	100	100	100	99	99	100	100
4	70	59	60	59	59	59	58
5	70	59	60	59	59	59	58
6	0	0	0	0	0	0	0

Gauss, težine2

	10	**100**	**200**	**300**	**500**	**700**	**1000**
1	0	0	0	0	0	0	0
2	100	99	100	100	100	100	100
3	100	100	100	100	100	100	100
4	60	58	60	62	63	63	63
5	60	58	60	62	63	63	63
6	0	0	0	0	0	0	0

Sažetak

Metode višekriterijske optimizacije koriste se u problemima odlučivanja uz postojanje više kriterija. Poteškoće koje se javljaju pri višekriterijskom odlučivanju proizlaze iz činjenice da su kriteriji odlučivanja u pravilu konfliktni, te konačna odluka predstavlja kompromis između postavljenih kriterija. Kako se pri odlučivanju mora voditi računa i o važnostima kriterija, javlja se i problem međusobne usporedbe važnosti kriterija.

U radu su osmišljene heuristike za donošenje odluke uz postojanje više kriterija Heuristike uz matematičke metode uključuju i logičko razmišljanje donositelja odluke kroz grupiranje mogućih izbora obzirom na sličnosti u razinama zadovoljenja kriterija.

Jedan od primjera višekriterijskog odlučivanja je i odabir ponuda pristiglih na javni natječaj Hrvatskog fonda za privatizaciju (HFP). Stoga se prikazuje implementacija kreiranih heuristika u vidu programske potpore za valorizaciju ponuda zaprimljenih na natječaj Hrvatskog fonda za privatizaciju te odabir jedne od ponuđenih ponuda. Kreirana programska potpora odlučivanju omogućava jednostavno korištenje što je posljedica manje količine informacija koje se očekuju od donositelja odluke.

Ključne riječi

Višekriterijsko odlučivanje, kompromisno programiranje, odabir ponuda za privatizaciju, grupiranje.

Summary

Methods of multiple criteria optimization are used in decision making problems in presence of multiple criteria. Difficulties that arise in multiple criteria decision making are consequences of the fact that the criteria are almost always in conflict with each other. Thus the final decision will be the result of compromise between the conflicting criteria. There is also a problem of comparing the importance of criteria which leads to difficulties in assigning the weights to criteria.

This paper constructs heuristics for multiple criteria decision making. The heuristics that are constructed include also a kind of logical thinking of the decision maker by grouping the alternatives according to their similarities in satisfying the level of each criterion.

The problem of evaluating takeover bids in process of privatization is an example of multiple criteria decision making problem. Thus, this paper presents the application of the heuristics created here to the case of evaluation and selection of takeover bids subjected to a tender of Croatian Privatization Fund.

The heuristics created here are implemented in a decision support system which can contribute to easier and quicker decision making process which is a consequence of small amount of information it requires from the decision maker to make the final decision.

Key words

Multiple criteria decision making, compromise programming, selection of takeover bids in privatization, grouping.

Životopis

Rođena sam 27.02.1979. godine u Somboru, Srbija. Diplomirala sam na Matematičkom odjelu Prirodoslovno-matematičkog fakulteta u Zagrebu završivši smjer „Računarstvo“ 2002. godine.

Neposredno nakon diplomiranja zaposlila sam se u Plivi d.d. na radnom mjestu projektant-programer. Od 2004. godine radim kao asistent na Katedri za matematiku Ekonomskog fakulteta Zagreb. 2005. godine sam upisala znanstveni poslijediplomski studij „Operacijska istraživanja“.

Sudjelovala sam na konferenciji KOI 2006 s radom " A VNS-Lagrangean heuristics for scheduling of patients in hospitals" uz koautoricu dr.sc. Zrinku Lukač. Sa koauorima dr.sc. Kristinom Šorić, dr.sc. Višnjom Vojvodić Rosenzweig i dr.sc. Darkom Tipurić objavila sam rad „Decision Support System for Evaluating Takeover Bids in Privatization“ u zborniku radova međunarodne konferencije „Contemporary Challenges of Theory and Practice in Economics“ u Beogradu.

Aktivno se koristim engleskim i pasivno španjolskim jezikom.