

Izrada simulacijskog modela interaktivne metode višekriterijalnog optimiranja

Nadih, Branimir

Undergraduate thesis / Završni rad

2010

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:835410>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-05-04**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

ZAVRŠNI RAD

Voditelj rada:

dr. sc. Dragutin Lisjak

Branimir Nadih

Zagreb, 2010

IZJAVA

Izjavljujem da sam ovaj završni rad radio samostalno na Fakultetu strojarstva i brodogradnje u Zagrebu znanjem stečenim tijekom studija.

B.N.

ZAHVALA

Zahvaljujem se prvenstveno mentoru prof.dr.sc Dragutinu Lisjaku na pomoći oko rada i stručnom vodstvu.

Zahvaljujem se i svim ostalim profesorima i kolegama koji su mi pomagali tijekom izrade ovog rada.

Također se zahvaljujem Slavenu Biliću i cijeloj Hrvatskoj nogometnoj reprezentaciji što se nisu kvalificirali na Svjetsko nogometno prvenstvo 2010.g. što mi je u velikoj mjeri omogućilo da se u potpunosti posvetim izradi ovoga rada.

B.N.

SADRŽAJ

SAŽETAK	4
POPIS SLIKA	5
POPIS TABLICA	5
1. UVOD	6
2. RAZVOJ MATEMATIČKOG PROGRAMIRANJA.....	9
2.1 Povijesni razvoj	9
2.2 Moderne metode optimizacije.....	10
3. PRIMJENA OPTIMIZACIJE U TEHNICI	11
4. POSTAVLJANJE OPTIMIZACIJSKOG PROBLEMA	12
5. VIŠEKRITERIJALNO ODLUČIVANJE	13
6. VIŠEKRITERIJALNO PROGRAMIRANJE	14
6.1 Osnovni pojmovi	14
6.1.1 Definicija.....	15
6.1.2 Kriterijalni skup.....	16
6.1.3 Marginalno rješenje.....	16
6.1.4 Ideal, idealna točka, idealna vrijednost funkcije cilja	16
6.1.5 Savršeno rješenje	17
7. METODE RJEŠAVANJA PROBLEMA VIŠEKRITERIJALNOG PROGRAMIRANJA	19
7.1 Sekvencijalna optimizacija	19
7.2 Agregiranje funkcija cilja u jednu funkciju cilja	21
7.3 Geometrijska metoda	21
7.4 Multikriterijalna simplex metoda	22
7.5 Interaktivni pristup	22
8. STEP (STEM) METODA	24
8.1 Algoritam STEP metode	24

8.2 Primjer problema rješenog STEP metodom	28
8.2.1 Određivanje skupa mogućih rješenja.....	29
8.2.2 Određivanje marginalnih rješenja	30
8.2.3 Određivanje idealnog rješenja	31
8.2.4 Određivanje efikasnog rješenja.....	32
8.2.5 Određivanje kompromisnog rješenja	33
8.2.6 Grafički prikaz kompromisnog rješenja	38
8.3 Simulacija STEP metode programskim jezikom Matlab	39
9. ZAKLJUČAK	44
10. PRILOG	45
10.1 M-file sa kodom simulacijskog modela.....	45
11. LITERATURA	72

SAŽETAK

U ovom radu je opisan problem optimizacije i višekriterijalnog optimiranja. Uz kratku povjest razvoja samog matematičkog programiranja i modernih metoda optimizacije, pokušao sam prikazati gdje se sve danas primjenjuju razne metode optimizacije u tehnici. Prikazao sam kako se matematički postavlja sam problem optimizacije i objasnio sve bitne pojmove za višekriterijalno programiranje i odlučivanje. Od raznih metoda unutar problema višekriterijalnog programiranja, detaljno sam razradio algoritam interaktivne *STEP* metode. Rješavanje problema *STEP* metodom pokazao sam na dva primjera od kojih je jedan riješen ručno, a drugi riješen računalnim simulacijskim modelom kojeg sam programirao u programskom jeziku Matlab.

POPIS SLIKA

Slika 1. Minimum funkcije $f(x)$ je jednak maksimumu funkcije $-f(x)$	8
Slika 2. Optimalno rješenje $cf(x)$ ili $c+f(x)$ je isto kao i $f(x)$	8
Slika 3. Grafički prikaz skupa mogućih rješenja i ekstremnih točaka	30
Slika 4. Prikaz idealne točke	32
Slika 5. Prikaz optimalnih funkcija cilja za svaku od točaka	33
Slika 6. Grafički prikaz 1. kompromisnog rješenja	39
Slika 7. Definicija problema simulacijskim modelom	40
Slika 8. Unos podataka u simulacijski model	41
Slika 9. Prikaz unešenih podataka u simulacijskom modelu	42
Slika 10. Grafičko rješenje simulacijskog modela	43
Slika 11. Prikaz prve <i>payoff</i> matrice (simulacijski model)	43
Slika 12. Prikaz prvog kompromisnog rješenja (simulacijski model)	44

POPIS TABLICA

Tablica 1. Metode operacijskog istraživanja	9
Tablica 2. Payoff tablica (matrica).....	26
Tablica 3. Payoff matrica (primjer).....	35

1. UVOD

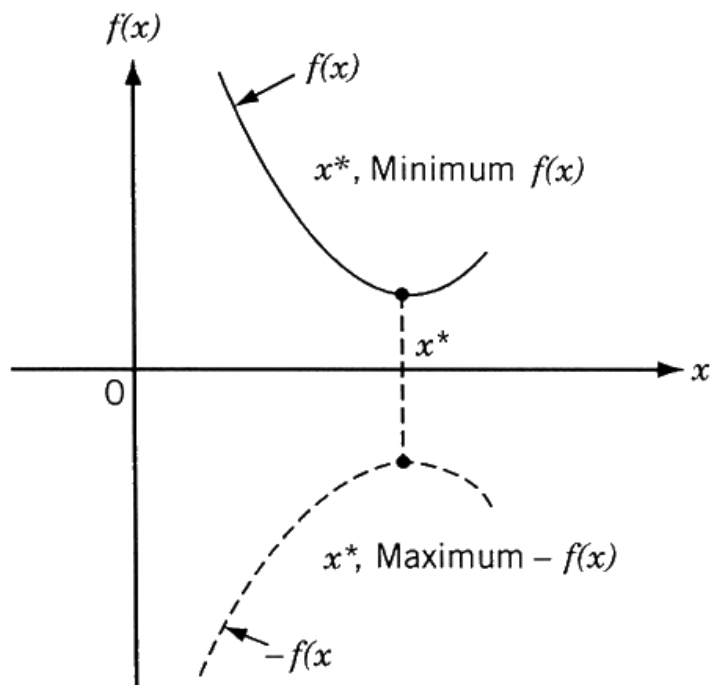
Optimizacija je postupak dobivanja najboljeg rezultata u datim okolnostima. U brojnim fazama projektiranja, izgradnje i održavanja bilo kojeg tehničkog sustava, inženjeri moraju donijeti mnoge tehnološke i rukovodilačke odluke. Krajnji cilj svih takvih odluka je ili minimizirati potreban rad i troškove ili maksimizirati željenu dobit. Kako se uloženo vrijeme ili željena dobit u svakoj situaciji mogu prikazati kao funkcija određenih varijabli odluka, optimizacija se može definirati kao proces pronalaženja uvjeta koji daju maksimalnu ili minimalnu vrijednost funkcije.

Iz slike 1. vidljivo je da ako točka x^* odgovara minimalnoj vrijednosti funkcije $f(x)$, tada ta ista točka odgovara maksimalnoj vrijednosti negativne funkcije $-f(x)$. Stoga se, bez gubitka općenitosti, optimizacija može svesti na čistu minimizaciju jer se maksimum funkcije može naći traženjem minimuma negativne funkcije.

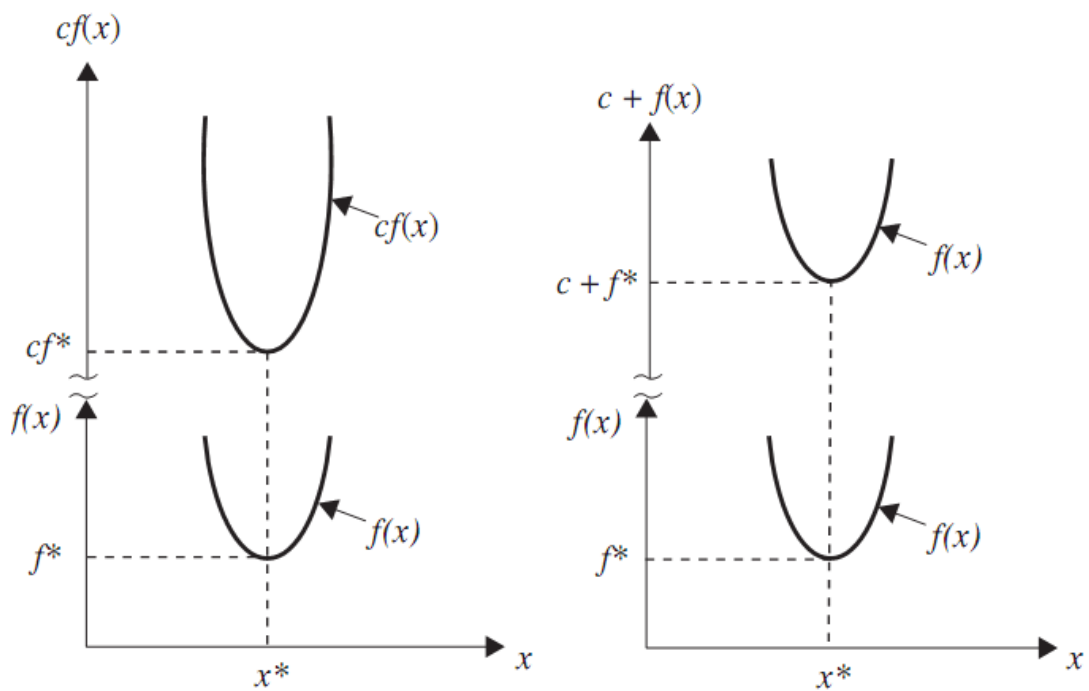
Korištenje sljedećih operacija na funkcijama neće promijeniti optimalno rješenje x^* (slika 2.):

1. Množenje (ili dijeljenje) funkcije $f(x)$ sa pozitivnom konstantom c .
2. Zbrajanje (ili oduzimanje) pozitivne konstante c sa (ili od) funkcije $f(x)$.

Ne postoji samo jedna metoda za efikasno rješavanje problema optimizacije. Postoji veliki broj metoda za rješavanje različitih oblika optimizacijskih problema. Metode koje traže optimalno rješenje su poznate i kao tehnike matematičkog programiranja i uglavnom se izučavaju unutar operacijskih istraživanja. Operacijsko istraživanje je dio matematike koji se bavi primjenom znanstvenih metoda i tehnika na problemima odlučivanja kako bi se ustanovilo najbolje ili optimalno rješenje. Počeci operacijskih istraživanja datiraju iz prvog dijela 2. svjetskog rata. Za vrijeme rata, Britanska vojska se suočila sa problemom raspodjele malog broja ograničenih resursa (kao što su borbeni avioni, radari i podmornice) na različite operacije (raspodjela na brojne ciljeve i destinacije). Kako nije bilo dostupne sistematske metode koje bi riješile ovaj problem, vojska je pozvala tim matematičara kako bi razvili metodu koja na znanstveni način rješava ovaj problem. Metode koje su oni razvili su bile ključne za dobivanje zračne bitke za Britaniju. Te metode, kao što je linearno programiranje, koje su nastale kao rezultat vojnih istraživanja naknadno su postale poznate kao metode operacijskih istraživanja.



Slika 1. Minimum funkcije $f(x)$ je jednak maksimumu funkcije $-f(x)$



Slika 2. Optimalno rješenje $cf(x)$ ili $c+f(x)$ je isto kao i $f(x)$

U tablici 1. prikazane su različite metode operacijskih istraživanja

Tablica 1. Metode operacijskog istraživanja

Matematičko programiranje ili optimizacijske tehnike	Tehnike stohastičkih procesa	Statističke metode
Računske metode Računanje varijanti Nelinearno programiranje Geometrijsko programiranje Kvadratno programiranje Linearno programiranje Dinamičko programiranje Cijelobrojno programiranje Stohastičko programiranje Odvojeno programiranje Višekriterijalno programiranje Mrežne metode: CPM i PERT Teorija igre <i>Moderne ili nekonvencionalne tehnike optimizacije</i> Genetski algoritmi Simulated annealing "Kolonija mrava" optimizacija Optimizacija roj čestica Neuronske mreže "Fuzzy" optimizacija	Teorija statističke odluke Markov proces Teorija čekanja u redu Teorija obnove Simulacijske metode Teorija pouzdanosti	Regresijska analiza Klaster analiza (prepoznavanje uzorka) Projektiranje eksperimenta Analiza razlika (faktor analiza)

Tehnike matematičkog programiranja (optimiranja) korisne su pri pronalasku minimuma funkcije sa više varijabli, sa određenim ograničenjima. Tehnike stohastičkih procesa primjenjuju se za analizu problema opisanih s više nasumičnih varijabli, za poznatu razdiobu vjerojatnosti. Statističke metode omogućavaju analizu eksperimentalnih podataka i izradu empirijskih modela da bi se dobila što bolja prezentacija rezultata dobivenih mjerenjem. Ovaj rad bavi se sa optimizacijskim tehnikama i to ponajviše sa višekriterijalnim linearnim optimiranjem.

2. RAZVOJ MATEMATIČKOG PROGRAMIRANJA

2.1 Povijesni razvoj

Počeci optimizacijskih metoda datiraju iz vremena Newton-a, Lagrange-a, i Cauchy-a [1]. Razvoj diferencijalnih računskih metoda optimizacija razvili su Newton i Leibnitz. Osnove varijantnog računa, koji se bavi minimizacijom funkcija, su postavili Bernoulli, Euler, Lagrange i Weirstrass. Metode optimizacije sa ograničenjima, koje uključuju i dodatak nepoznatih množitelja, postale su poznate po svojem izumitelju, Lagrange-u. Cauchy je napravio prvu primjenu metode "najstrmijeg pada" (*steepest descent*) da bi riješio probleme minimizacije bez ograničenja. Usprkos tim ranim doprinosima, jako malo napretka se dogodilo do sredine dvadesetog stoljeća, kada su brza digitalna računala omogućila implementaciju optimizacijskih procesa i potaknula istraživanja novih metoda. Usljedio je veliki napredak koji je popratila i brojna literatura o optimizacijskim tehnikama. Napredak je rezultirao stvaranjem nekoliko novih područja optimizacijskih teorija.

Glavnina razvoja područja numeričkih metoda optimizacije bez ograničenja odvijala se u Velikoj Britaniji šezdesetih godina.

Razvoj *simplex* metode, koju je izumio Dantzig 1947.g., za probleme linearnog programiranja, te objava Bellman-ovog principa optimalnosti 1957.g. za probleme dinamičkog programiranja usmjerili su razvoj metoda za optimizaciju koje uključuju ograničenja. Rad Kuhn-a i Tucker-a 1951.g. o nužnim i dovoljnim uvjetima za optimalno rješenje problema programiranja bio je temelj za veliki broj kasnijih istraživanja u nelinearnom programiranju. Značajan je doprinos Zoutendijk-a i Rosen-a u području nelinearnog programiranja ranih šezdesetih godina. Bez obzira na to što ne postoji niti jedna tehnika za rješavanje svih problema nelinearnog programiranja, radovi Carroll-a, Fiacco-a i McCormick-a omogućili su rješavanje složenih problema koristeći poznate metode optimizacije bez ograničenja.

Geometrijsko programiranje razvili su šezdesetih godina Duffin, Zener i Peterson. Gomory je napravio početne korake u cijelobrojnom programiranju, dio optimizacije koji se najbrže razvija. Razlog tome je činjenica što većina problema iz stvarnog života spada u ovu kategoriju problema. Dantzig, Charnes i Cooper razvili su tehnike stohastičkog programiranja i riješili probleme tako da su pretpostavili da su parametri projektiranja neovisni i normalno distribuirani.

Želja za optimizacijom više od jednog kriterija i cilja uz zadovoljavanje fizičkih ograničenja dovela je do razvoja metoda višekriterijalnog programiranja. Ciljno programiranje su prvobitno predstavili Charnes i Cooper za linearne probleme 1961.g. Osnove teorije igre je postavio Von Neumann 1928.g. i od tad se tehnika koristi u rješavanju više matematičkih, ekonomskih i vojnih problema, a tek se u zadnjih par godina počela koristiti pri rješavanju problema projektiranja u tehnici.

2.2 Moderne metode optimizacije

Moderne metode optimizacije [1], ili netradicionalne optimizacijske metode, pojavile su se kao moćne metode za rješavanje složenih tehničkih optimizacijskih problema zadnjih godina. One uključuju genetičke algoritme, 'simulated annealing', koloniju mrava, optimizaciju rojem čestica te optimizaciju na bazi neuronskih mreža i fuzzy logike.

Genetički algoritmi predstavljaju računalne čiji se optimizacijski algoritmi temelje na zakonima genetike prirode te prirodne selekcije. Prvi ih je predstavio 1975.g. John Holland.

Metoda 'Simulated annealing' (simulirano žarenje) bazira se na teoriji procesa hlađenja metala. Metodu su prvobitno razvili Kirkpatrick, Gelatt i Vecchi.

Algoritam optimizacijom roja čestica oponaša ponašanje organizama koji žive u grupama kao što su kolonije ili rojevi insekata (npr. mrava, termita, pčela i osa), jata ptica i riba. Algoritam su prvobitno predstavili 1995 Kennedy i Eberhart.

Optimizacija kolonijom mrava bazira se na oponašanju kolonija mrava koje uvijek pronalaze najkraći put od mravinjaka do hrane. Metodu je prvi razvio Marco Dorigo 1992.g.

Metoda neuronskih mreža se temelji se oponašanju rada ljudskog mozga čija je primjena došla do izražaja sa sve većim povećanjem brzine računalnih procesora. Metodu su prvi koristili Hopfield i Tank 1985.g.

"Fuzzy" optimizacija razvila se za rješavanje problema kod kojih su ograničenja nejasna tj. neprecizno postavljena. "Fuzzy" metodu jednokriterijalnog i višekriterijalnog optimiranja prvi je prezentirao Rao 1986.g.

3. PRIMJENA OPTIMIZACIJE U TEHNICI

Optimizacija, u širem značenju, može se primjeniti za rješavanje bilo kojeg problema u tehnici. Neki od primjera primjene metoda optimizacije su:

1. Dobivanje minimalne težine pri projektiranju aviona i svemirskih letjelica,
2. Pronalazak optimalne putanje svemirskih letjelica,
3. Postizanje minimalnih troškova pri projektiranju držača, temelja, mostova, tornjeva, dimnjaka, brana,
4. Za projektiranje minimalne težine građevina za potrese, vjetrove, i druge oblike opterećenja i nasumičnih naprezanja,
5. Za projektiranje maksimalne učinkovitosti sustava za dovod vode i navodnjavanje,
6. Za kontrolu zastoja, čekanja i dobivanje minimalnog vremena proizvodnje,
7. Za optimalno projektiranje alatnih strojeva i ostalih strojnih komponenata,
8. Za određivanje uvjeta obrade na stroju sa ciljem minimiziranja troškova proizvodnje,
9. Projektiranje opreme za rukovanje materijalom: konvejera, kamiona, dizalica uz minimalne troškove,
10. Pri projektiranju pumpa, turbina i opreme za prijenos topline kako bi postigli maksimalnu učinkovitost,
11. Za optimalno projektiranje generatora, motora i transformatora,
12. Optimalno projektiranje električnih mreža,
13. Za dobivanje najkraće ruta za trgovca koji mora posjetiti par gradova,
14. Optimalno planiranje procesa, kontroliranje i zakazivanje rokova,
15. Analiza i izrada empiriskog modela za eksperimentalne podatke kako bi se dobila najbolja prezentacija dobivenih rezultata,
16. Optimalno projektiranje kemijskih postrojenja i opreme,
17. Za projektiranje cijevovoda u procesnoj industriji,
18. Pronalazak savršene lokacije za tvornicu,
19. Planiranje održavanja i izmjene opreme uz najmanje troškove,
20. Planiranje najbolje strategije za maksimalni profit,
21. Preraspodjela resursa na različite aktivnosti kako bi se maksimizirala dobit .

4. MATEMATIČKI OPIS PROBLEMA OPTIMIZACIJE

Problem optimizacije matematički se opisuje na sljedeći način:

$$\text{Pronađi } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ koji minimizira } f(\mathbf{X})$$

gdje $f(\mathbf{X})$ ovisi o sljedećim ograničenjima:

$$\begin{aligned} g_j(\mathbf{X}) &\leq 0, & j &= 1, 2, \dots, m \\ l_j(\mathbf{X}) &= 0, & j &= 1, 2, \dots, p \end{aligned} \tag{1}$$

gdje je:

\mathbf{X} n -dimenzijski vektor zvan vektor varijabli odluke,

$f(\mathbf{X})$ se zove funkcija cilja,

$g_j(\mathbf{X})$ i $l_j(\mathbf{X})$ su ograničenja jednakosti, odnosno nejednakosti.

Broj varijabli n i broj ograničenja m i p ne moraju biti jednaki.

$$\text{Pronađi } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ koji minimizira } f(\mathbf{X}) \tag{2}$$

Problem opisan izrazom (1) je problem sa ograničenjima, a problem opisan izrazom (2) je problem bez ograničenja.

5. VIŠEKRITERIJALNO ODLUČIVANJE

Postoje mnogi razlozi rastućeg interesa za višekriterijalno odlučivanje. Prvi i najvažniji je spoznaja da je većina problema odlučivanja sama po sebi višekriterijalna. Čak i mnogi problemi koji se klasično razmatraju kao jednokriterijalni mogu se promatrati kao višekriterijalni.

Drugi važan razlog pojačanog interesa za višekriterijalno odlučivanje je izuzetan razvoj mogućnosti, brzine, kapaciteta i fleksibilnosti računala. Naime, algoritmi za rješavanje višekriterijalnih problema uglavnom zahtijevaju više memorije i vremena zauzeća procesora nego jednokriterijalni. Pored toga, većina višekriterijalnih algoritama zahtijeva interaktivni pristup, tj. komunikaciju između donosioca odluke i algoritma metode optimizacije. Taj interaktivni pristup zahtijeva brze odgovore od računala, a i prilagodljivost stroja i programa pomoću kojeg se izvodi algoritam.

U višekriterijalnom odlučivanju važnu ulogu igra donosilac odluke. Donosilac odluke je osoba čije je zadatak odabir najboljeg kompromisnog rješenja zadanog višekriterijalnog problema. Uloga donosioca odluke nije toliko bitna u rješavanju jednokriterijalnih problema. Naime kad je jednokriterijalni problem jednom formuliran, dobivanje njegovog rješenja je relativno jednostavno.

Kod višekriterijalnog odlučivanja donosilac odluke je neizostavan te ga se ne može se nadomjestiti bez obzira na kvalitetu metode tj. algoritma. Nijedna metoda ne može sama za sebe odrediti najbolje rješenje u pojedinoj situaciji. U najboljem slučaju, metoda može koristiti donosiocu odluke da ojača osnovu na temelju kojoj se odluka donosi i da poboljša kvalitetu procesa odlučivanja.

Višekriterijalno odlučivanje najčešće se razdvaja na dva različita područja [2]:

1. Višekriterijalna optimizacija ili višekriterijalno programiranje – problemi gdje je broj alternativa (mogućih rješenja) velik.
2. Višeatributna analiza odlučivanja – tu se najčešće svrstavaju problemi s manjim (konačnim) brojem alternativa.

6. VIŠEKRITERIJALNO PROGRAMIRANJE

6.1 Osnovni pojmovi

Problem višekriterijalnog programiranja je problem izbora vrijednosti za svaku od varijabli odluke x_1, x_2, \dots, x_n , gdje je X vektor varijabli odluke, tj.

$$X = [x_1, x_2, \dots, x_n], \quad (3)$$

u namjeri da se optimira r ($r \geq 2$) kriterijalnih funkcija $f_1(X), f_2(X), \dots, f_r(X)$.

Traži se dakle optimum funkcije:

$$f(X) = [f_1(X), f_2(X), \dots, f_r(X)], \quad (4)$$

te se iz tog razloga veoma često govori o **vektorskoj optimizaciji**.

Pretpostavimo da donosilac odluke želi maksimizirati svaku od tih funkcija istovremeno (minimizacija se svodi na maksimizaciju jednostavnim množenjem funkcije cilja s -1). U problemu, pored toga, postoje različita ograničenja na varijable odluke, tj. postoji tzv. skup mogućih rješenja S , odnosno naš vektor odluke X mora biti element skupa mogućih rješenja, tj. $X \in S$.

Pretpostavimo, nadalje, da je $S \subseteq R^n$ i neka postoji r funkcija cilja:

$$f_k : S \rightarrow R, k=1, 2, \dots, r.$$

Problem maksimizacije svih r funkcija cilja istovremeno, odnosno problem višekriterijalnog programiranja može se matematički definirati ovako:

$$\text{Max } f(X) = [f_1(X), f_2(X), \dots, f_r(X)] \quad (5)$$

$$X \in S$$

Ako su sve funkcije iz tog problema, uključujući i one koje definiraju skup S , linearne radi se o problemu višekriterijalnog linearnog programiranja (**VLP**).

$$\begin{aligned}
& \text{Max} \left[\sum_{j=1}^n c_{1j} x_j, \sum_{j=1}^n c_{2j} x_j, \dots, \sum_{j=1}^n c_{rj} x_j \right] \\
& \sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1, 2, \dots, m \\
& x_j \geq 0, j = 1, 2, \dots, n
\end{aligned} \tag{6}$$

ili u matričnom obliku

$$\begin{aligned}
& \text{Max} \quad CX \\
& AX \leq B \\
& X \geq 0
\end{aligned} \tag{7}$$

gdje je C kriterijalna matrica, odnosno matrica koeficijenta iz svih funkcija cilja, odnosno

$$CX = (C^1X, C^2X, \dots, C^rX)^T \text{ (oznaka za transpoziciju)}$$

gdje su C^1, C^2, \dots, C^r reci matrice $C(r, n)$, A je matrica reda (m, n) , a B je vektor m -tog reda.

Uvedimo nadalje neke osnovne pojmove koje koristimo u višekriterijalnom programiranju.

6.1.1 Definicija

Za proizvoljne vektore $X, Y \in R^n$ kažemo da je $X \leq Y$ ako i samo ako je $x_i \leq y_i$ za svaki $i = 1, 2, \dots, n$.

Time smo u vektorskom prostoru R^n (skup vektora iz prostora možemo razmatrati kao skup jednostupčanih matrica tipa $n \times 1$) uveli relaciju parcijalnog uređaja. Takav uređaj nije potpuni uređaj, tj. pri takvom uređenju vektora postoje vektori koji su međusobno neusporedivi. Takvi su npr. vektori:

$$X = \begin{bmatrix} 3 \\ 5 \end{bmatrix} \text{ i } Y = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

6.1.2 Kriterijalni skup

Svakom elementu skupa mogućih rješenja, tj. svakom vektoru $X \in S$, pridružen je kriterijalni vektor $f(X)$

$$f(X) = [f_1(X), f_2(X), \dots, f_r(X)]$$

Skup

$$f(S) = \{f(X) : X \in S\} \quad (8)$$

zovemo kriterijalni skup.

Drugim riječima, svakom elementu skupa S , tj. svakom mogućem rješenju, pridružili smo vektor čije su komponente vrijednosti funkcija cilja za to moguće rješenje. Na taj način smo dobili skup točaka u jednom drugom vektorskom prostoru, čija je dimenzija jednaka broju funkcija cilja (r). Takav skup točaka (vektora) zovemo kriterijalni skup.

6.1.3 Marginalno rješenje

Označimo maksimu svake komponente vektora $f(X)$ na skupu mogućih rješenja sa

$$\begin{aligned} \max_{X \in S} f_k(X) &= f_k(X^{k*}) = f_k^* \end{aligned} \quad (9)$$

Vektor X^{k*} zvat ćemo marginalno rješenje za funkciju cilja f_k , odnosno taj vektor je optimalno rješenje jednokriterijalnog problema linearnog programiranja čija je funkcija cilja upravo f_k .

Takvih optimalnih rješenja za k -tu funkciju cilja očito ima onoliko koliko ima funkcija cilja, odnosno u svakom problemu višekriterijalnog programiranja postoji r marginalnih rješenja. Naravno, neka od tih marginalnih rješenja mogu se i ponavljati, tj. može se desiti da neko rješenje bude optimalno rješenje za dva ili više jednokriterijalnih problema.

6.1.4 Ideal, idealna točka ili idealna vrijednost funkcije cilja

Vektor $f^* = [f_1^*, f_2^*, \dots, f_r^*]$ čija je k -ta komponenta optimalna vrijednost k -te funkcije cilja naziva se ideal, idealna točka ili idealna vrijednost vektorske funkcije cilja $f(X)$.

6.1.5 Savršeno rješenje

Za rješenje X^S višekriterijalnog problema kaže se da je savršeno rješenje ako za njega sve funkcije cilja istovremeno postižu maksimalne vrijednosti, tj.

X^S je savršeno rješenje ako je $X^S \in S$ (moguće rješenje) i ako je

$$f_i(X^S) \geq f_i(X) \text{ za } i = 1, 2, \dots, r. \text{ i za svaki } X \in S.$$

Ako takvo rješenje postoji unutar skupa S tada je vrijednost funkcije cilja za takvo rješenje upravo idealna točka. Međutim, uobičajeno je da su barem dva od r kriterija međusobno konflikttni. Iz tih razloga savršeno rješenje uglavnom ne postoji pa ga često nazivaju i utopijskim rješenjem (utopia solution).

6.1.6 Efikasno rješenje (nedominirano rješenje, Pareto optimalno rješenje)

Efikasno rješenje X^E problema višekriterijalnog programiranja je moguće rješenje, $X^E \in S$, za koje ne postoji neko drugo moguće rješenje $X \in S$, takvo da je

$$f_i(X^E) \leq f_i(X) \text{ za sve } i = 1, 2, \dots, r, \text{ i}$$

$$f_i(X^E) < f_i(X) \text{ za neki } i = 1, 2, \dots, r$$

tj. ne postoji drugo moguće rješenje koje dominira nad X^E .

Očito je da je u drugoj nejednakosti umjesto znaka " $<$ " potpuno ekvivalentno mogao stajati i znak " \neq ".

Drugim riječima efikasno rješenje problema višekriterijalnog programiranja je ono moguće rješenje za koje ne postoji neko drugo moguće rješenje koje je za barem po jednom kriteriju bolje, a po ostalima nije gore od njega.

Može se također reći da je efikasno rješenje ono moguće rješenje za koje je porast vrijednosti bilo kojeg kriterija (funkcije cilja) moguće postići samo po cijeni smanjenja jednog od ostalih kriterija (trade-off, razmjena, Pareto optimalnost).

Efikasno rješenje može se promatrati i preko skupa svih vrijednosti vektorske funkcije f na skupu mogućih rješenja, tj. na kriterijalnom skupu:

$$f(S) = \{f(X) : X \in S\}.$$

Skup $f(S)$ nije potpuno uređen skup i općenito nema najvećeg elementa. Najveći element bila bi idealna točka, ali ona uglavnom ne pripada skupu S (jer uglavnom ne postoji savršeno rješenje). Međutim, taj skup može imati više maksimalnih elemenata f^0 , tj. onih od kojih nema "boljih" ili "većih".

Dakle, $f^0 = (f_1^0, f_2^0, \dots, f_r^0)$ je maksimalni element kriterijalnog skupa ako ima svojstvo:

$$f \in f(S) \text{ i } f \geq f^0 \Rightarrow f = f^0$$

tj. od njega nema "većeg" elementa, gdje pojam "veći" znači da su sve komponente jednog vektora veće od komponenti drugog vektora.

Neka je f^0 jedna od maksimalnih vrijednosti vektorske funkcije $f(X)$ na skupu S , tada se točka $X^0 \in S$ u kojoj se postiže taj maksimum naziva efikasnim ili nedominiranim rješenjem problema višekriterijalnog programiranja.

7. METODE RJEŠAVANJA PROBLEMA VIŠEKRITERIJALNOG PROGRAMIRANJA

7.1 Sekvencijalna optimizacija

Neka od efikasnih rješenja problema višekriterijalnog programiranja mogu se dobiti tzv. sekvencijalnom optimizacijom ili optimizacijom u nizu. Radi se o tome da se skup kriterija hijerarhizira po važnosti, tj. utvrdi što je za donosioca odluke najvažniji kriterij, koji je drugi po važnosti itd. Pretpostavimo da je uveden takav hijerarhijski poredak i da je on upravo f_1, f_2, \dots, f_r . Dakle, najvažnija funkcija cilja za donosioca odluke je f_1 , zatim f_2 itd. Tada se prvo određuju rješenja X iz S u kojima funkcija f_1 postiže svoj maksimum. Ako ih ima više među njima se traže ona koja maksimiziraju f_2 itd.

Očito je, također, da neka druga hijerarhizacija funkcija cilja, odnosno drugo uređenje skupa $f(S)$ generira neka druga efikasna rješenja razmatranog problema višekriterijalnog programiranja. U većini slučajeva postoji mnogo efikasnih rješenja. Moguće je čak i da cijeli skup mogućih rješenja S bude efikasan. Jedno od tih efikasnih rješenja preferirat će donosilac odluke. Međutim, da bi odredili koje je to rješenje treba dobiti daljnje informacije od donosioca odluke koje sadrže njegove preferencije

Jedan od načina da se to postigne je tzv. **funkcija korisnosti** (utility function, value function). To je funkcija $v(f(X))$ koja pridružuje realan broj svakom $X \in S$, odnosno definira se novi problem koji više kriterija sadrži u implicitnom obliku.

$$\begin{array}{l} \text{Max } v(f(X)) \\ X \in S \end{array} \quad (10)$$

Time je ustvari višekriterijalni problem preveden u jednokriterijalni. Međutim, takvu funkciju je veoma teško korektno odrediti. Tako dobiveno rješenje naziva se **najbolje kompromisno rješenje** u smislu da je ono kompromis među različitim kriterijima zadanog problema. To rješenje je naravno jedno od efikasnih rješenja.

Glavni problemi koji se javljaju kod rješavanja problema višekriterijalnog programiranja sastoje se upravo u tome na koji način, kada i kako modelirati informacije o preferencijama donosioca odluke. Važnu ulogu u donošenju konačne odluke igra i idealna točka koja se često koristi kao polazna točka za usporedbu. Iako je ona uglavnom nedostižna donosilac odluke sigurno će preferirati ono moguće i efikasno rješenje koje je najbliže uočenom idealu.

Kompromisno rješenje se često bira kao ona točka skupa S čija je slika u kriterijalnom skupu "najbliža" idealnoj točki f^* .

Mjera bliskosti vektora

$$f(X) = [f_1(X), f_2(X), \dots, f_r(X)]$$

i ideala

$$f^* = (f_1^*, f_2^*, \dots, f_r^*)$$

je jedna od L_p metrika ($1 \leq p \leq \infty$), tj.

$$\|f(X) - f^*\|_p = \left[\sum_{k=1}^r |f_k(X) - f_k^*|^p \right]^{1/p} \quad (11)$$

Najpoznatija od tih metrika je tzv. Euklidova metrika L_2 :

$$\|f(X) - f^*\|_2 = \sqrt{\sum_{k=1}^r |f_k(X) - f_k^*|^2} \quad (12)$$

Traženje kompromisnog rješenja svodi se dakle na to da se od svih točaka i skupa S odredi ona čija je slika u kriterijalnom skupu "najbliža" idealnoj točki, tj. traži se rješenje problema:

$$\text{Min} \left\{ \Lambda(f(X) - f^*), X \in S, p \geq 1 \right\} \quad (13)$$

gdje je Λ vektor težina (pondera, važnosti) pojedinog kriterija.

Npr. za L_2 traži se

$$\text{Min} \sqrt{\lambda_1 (f_1(X) - f_1^*)^2 + \dots + \lambda_r (f_r(X) - f_r^*)^2} \quad (14)$$

$$X \in S$$

Za pondere vrijedi $\lambda_k > 0$, $k = 1, 2, \dots, r$, a najčešće se uzima i $\lambda_1 + \lambda_2 + \dots + \lambda_r = 1$.

7.2 Agregiranje funkcija cilja u jednu funkciju cilja

Drugi način određivanja najboljeg kompromisnog rješenja je svođenje višekriterijalnog na jednokriterijalno programiranje, tj. agregiranje funkcija cilja u jednu funkciju cilja. Može se reći da ustvari određujemo funkciju korisnosti za naš problem.

$$\text{Max } f(X) = f_1(X) + f_2(X) + \dots + f_r(X)$$

i/ili

$$\text{Max } f(X) = \lambda_1 f_1(X) + \lambda_2 f_2(X) + \dots + \lambda_r f_r(X) \quad (15)$$

Prvo se odrede marginalna rješenja, tj. optimalna rješenja po svakoj funkciji cilja posebno.

Idealna vrijednost naše vektorske funkcije cilja je vektor čije su komponente optimalne vrijednosti svake funkcije cilja posebno. Taj vektor obično ne postoji unutar skupa mogućih rješenja. Za optimalno rješenje dobivamo uvijek neko efikasno rješenje, jer ponderirana suma funkcija cilja za optimalno rješenje uvijek će dati jedno od efikasnih rješenja problema višekriterijalnog programiranja.

Ako smo na neki način odredili više (ili sva) efikasna rješenja tada najbolje kompromisno rješenje možemo odabrati i kao ono čiji vektor vrijednosti funkcije cilja ima minimalnu udaljenost od idealne točke.

7.3 Geometrijska metoda

Skup mogućih rješenja nekog problema možemo prikazati i grafički u 2D prikazu, ako je problem zadan sa dvije varijable.

Ucrtamo u koordinatni sustav pravce (poluravnine) linearnih ograničenja. Os x predstavlja vrijednosti varijable x_1 , a os y vrijednosti varijable x_2 . Pravci (poluravnine) ograničenja će u koordinatnom sustavu, da li međusobno ili sa koordinatnim osima, zatvarati prostor. Skup svih točaka unutar tog prostora (skupa S) je skup mogućih rješenja. Točke sjecišta tih pravaca, ili rubne točke tog prostora, uzimam kao ekstremne točke našeg višekriterijalnog problema.

Za te točke prvo odredimo marginalna rješenja, tj. optimalna rješenja po svakoj funkciji cilja za sve ekstremne točke. Nakon toga odredimo efikasna rješenja danog problema. Ako spojimo efikasna rješenja sa funkcijama cilja dobiti ćemo nove pravce (odnosno poluravnine) koje ucrtavamo na graf. Time dobivamo novi skup mogućih rješenja. Da bi odredili da li su i ostale točke iz skupa S efikasne dovoljno je da te pravce paralelno translatiramo po skupu S .

Dakle naši problemi imaju beskonačno mnogo efikasnih rješenja od kojih je onda potrebno odabrati nama najbolje.

7.4 Multikriterijalna simplex metoda

Možemo reći da se osnovni problem višekriterijalnog programiranja sastoji iz dva koraka :

1. Odrediti skup svih efikasnih točaka
2. Iz skupa svih efikasnih točaka izdvojiti najbolje kompromisno rješenje

Jedna od metoda koja nam služi za određivanje efikasnih ekstremnih točaka problema višekriterijalnog linearnog programiranja je tzv. multikriterijalna simplex metoda. Ta metoda određuje samo ekstremne efikasne točke, međutim sve ostale efikasne točke mogu se dobiti kao konveksne kombinacije nekog od podskupova ekstremnih efikasnih točaka. M.Zeleny je razvio algoritam za određivanje svih efikasnih točaka, ako su poznate ekstremne [2].

7.5 Interaktivni pristup

Donosiocu odluke je vrlo teško unaprijed reći što je za njega najbolje rješenje s obzirom na sve kriterije. Jedino što u početku zna je da najbolje rješenje treba tražiti u skupu efikasnih rješenja. Međutim, kada se i odredi taj skup može biti veoma teško iz njega odabrati najbolje kompromisno rješenje. Iz tih razloga prevladava mišljenje da rješenje problema višekriterijalnog programiranja treba tražiti interaktivnim pristupom.

Interaktivne metode zasnivaju se na komunikaciji između donosioca odluke i računala, odnosno analitičara ili operacionog istraživača sa računalnim softverom. Možemo reći da se one zasnivaju na progresivnom definiranju preferencija uz istovremeno istraživanje skupa mogućih rješenja.

U svakoj iteraciji interaktivne metode odvija se dijalog između donosioca odluke i računala. Počinje s fazom računanja kroz koju se određuje jedno kompromisno rješenje i ono se predlaže donosiocu odluke. On uspoređuje to rješenje s idealnom točkom i ako nije zadovoljan reagira i daje dodatne informacije o svojim preferencijama. Tada se na temelju tih novih podataka traži novo kompromisno rješenje i postupak se nastavlja dok se ne nađe zadovoljavajuće rješenje.

Najpoznatije interaktivne metode višekriterijalnog programiranja su metoda pomjerajućeg ideala (displaced ideal) koju je razvio već prije spomenuti M. Zeleny, metoda Zionts – Walleniusa, GDF metoda (Geoffrion – Dyer – Feinberg), te grupa tzv. POP i STEP metoda. U okviru ovog rada zadržat ćemo se na ovim posljednjim. Njih su razvili R. Benayoun, J. Tergyn, J. Montgolfier, O. Larittchev i Ph. Vincke.

U početnoj fazi svih tih metoda optimizira se posebno svaka funkcija cilja, tj. rješavaju se problemi tipa:

$$\text{Max } C^k X$$

$$AX \leq A_o \quad k = 1, 2, \dots, r$$

$$X \geq 0$$

Ako postoji savršeno rješenje problem je riješen. U protivnom rješava se pomoćni problem linearnog programiranja koji određuje ono moguće rješenje čija je slika u prostoru funkcija cilja na neki način "najbliža" idealnoj točki. To rješenje je jedno od efikasnih rješenja i ono se nudi donosiocu odluke kao prvo rješenje.

U drugoj fazi postavljaju se pitanja koja pomažu i donosiocu odluke i istraživaču da dobiju rješenje koje će u potpunosti (koliko je to moguće) zadovoljiti donosioca odluke. Ta pitanja su:

- da li kompromisno rješenje ima neke nezadovoljavajuće vrijednosti za funkcije cilja (donosiocu odluke su na raspolaganju njihove maksimalne vrijednosti iz prve faze)
- koju bi funkciju cilja želio povećati s tim da dozvoli da se neka druga smanji
- želi li proširiti ili oslabiti neka ograničenja, itd.

Uvažavajući njegove odgovore rješava se novi problem i donosi novo kompromisno rješenje sve dok se ne dobije zadovoljavajuće, odnosno najbolje kompromisno rješenje.

8. STEP (STEM) METODA

8.1 Algoritam STEP metode

1971.g. R. Benayoun je razvio *STEP* metodu kao interaktivnu tehniku koja određuje najbolje kompromisno rješenje u ne više od p iteracija, gdje je p broj funkcija cilja. Metoda se bazira na geometrijskom pojmu najbolje, odnosno najbliže točke idealnom rješenju, sa izmjenama u kriterijima potaknuti reakcijom donosioca odluke na dobivena rješenja.

Neka je zadan problem višekriterijalnog linearnog programiranja:

$$\text{Max } f_1(X) = C^1 X$$

$$\text{Max } f_2(X) = C^2 X$$

.....

$$\text{Max } f_r(X) = C^r X$$

$$X \in S$$

gdje je $S = \{X : X \geq 0, AX \leq B\}$, a C^k vektor redak koeficijenata iz k -te funkcije cilja.

Pojedinačno se optimizira svaka funkcija cilja, tj. posebno se rješavaju jednokriterijalni problemi:

$$\text{Max } C^k X$$

$$AX \leq A_0 \quad k = 1, 2, \dots, r$$

$$X \geq 0$$

Neka je X^{k*} vektor koji maksimizira k -tu funkciju cilja (optimalno rješenje za k -ti problem). Vrijednost k -te funkcije cilja za to rješenje (tj. optimalna vrijednost funkcije cilja k -tog problema) je:

$$C^k X^{k*} = f_k^*.$$

Ako postoji moguće rješenje \bar{X} takvo da je $C^k \bar{X} = f_k^*$ za svaki $k = 1, 2, \dots, r$, tada je \bar{X} savršeno rješenje našeg problema VLP, a vektor $f^* = [f_1^*, f_2^*, \dots, f_r^*]$ je idealna točka u prostoru funkcija cilja.

Metoda počinje konstruiranjem payoff matrice ili tablice (matrica isplativosti). Payoff tablica se izrađuje kako bi se vidjele težinske vrijednosti naših rješenja i njihova udaljenost od idealnog rješenja.

Tablica se popunjava na sljedeći način. $f_j(X^{k*})$ je vrijednost j -te funkcije cilja za ono rješenje koje maksimizira k -tu funkciju cilja. U j -tom retku nalaze se vrijednosti j -te funkcije cilja za marginalna rješenja X^{k*} ($k = 1, 2, \dots, r$).

Tablica 2. Payoff tablica (matrica)

	X^{1*}	X^{2*}	X^{k*}	X^{r*}
f_1	$f_1(X^{1*})$	$f_1(X^{2*})$	$f_1(X^{k*})$	$f_1(X^{r*})$
f_2	$f_2(X^{1*})$	$f_2(X^{2*})$	$f_2(X^{k*})$	$f_2(X^{r*})$
.....
f_k	$f_k(X^{1*})$	$f_k(X^{2*})$	$f_k(X^{k*})$	$f_k(X^{r*})$
.....
f_j	$f_j(X^{1*})$	$f_j(X^{2*})$	$f_j(X^{k*})$	$f_j(X^{r*})$

U k -tom stupcu te matrice prikazane su vrijednosti svih funkcija cilja za vektor (rješenje) X^{k*} koji maksimizira k -tu funkciju cilja. Vrijednosti $f_i(X^{i*})$ na glavnoj dijagonali formiraju idealnu točku u prostoru funkcije cilja.

Označimo nadalje sa f_k^{\min} minimalnu vrijednost u k -tom retku tabele, tj.

$$f_k^{\min} = \min \{ C^k X^{1*}, C^k X^{2*}, \dots, C^k X^{r*} \}$$

i formiramo sljedeći izraz:

$$\alpha_k = \begin{cases} \frac{f_k^* - f_k^{\min}}{f_k^*} \cdot \frac{1}{\sqrt{\sum_{j=1}^n (c_{kj}^2)}}, f_k^* > 0 \\ \frac{f_k^{\min} - f_k^*}{f_k^{\min}} \cdot \frac{1}{\sqrt{\sum_{j=1}^n (c_{kj}^2)}}, f_k^* \leq 0 \end{cases} \quad (16)$$

te pomoću tih izraza formirajmo pondere(težine, važnosti) Π_k za svaki pojedini kriterij:

$$\Pi_k = \frac{\alpha_k}{\sum_{j=1}^n \alpha_j} \quad (17)$$

Očito je da je suma tih pondera Π_k jednaka 1, i ti ponderi se pridružuju k -toj funkciji cilja. Veći ponder (važnost) ima ona funkcija cilja koja se više rasipa (varira) na skupu marginalnih rješenja, tj. vektora X^{k*} .

Tada se rješava pridruženi problem linearnog programiranja:

$$\begin{aligned} & \text{Min } \lambda \\ & [f_k^* - C^k X] \cdot \Pi_k \leq \lambda, k = 1, 2, \dots, r \\ & X \in S \\ & \lambda \geq 0 \end{aligned} \quad (19)$$

Taj problem kao optimalno rješenje ima opet jedno od efikasnih rješenja početnog problema višekriterijalnog linearnog programiranja.

Očito je dakle da se tim problemom linearnog programiranja traži takav vektor X da razlika $f_k^* - C^k X$ bude što je moguće manja (jer se minimizira λ).

S druge strane izraz

$$[f_k^* - C^k X] \cdot \frac{1}{\sqrt{\sum_{j=1}^n (c_{kj}^2)}}$$

predstavlja udaljenost točke X od hiperravnine određene k -tom funkcijom cilja.

Tada se rješava pomoćni problem linearnog programiranja gdje se početnim ograničenjima pridodaju ova nova. Rješenje tog problema je također jedno od efikasnih rješenja sa odgovarajućim vrijednostima funkcija cilja.

To je prvo kompromisno rješenje koje se nudi donosiocu odluke. Vrijednosti funkcija cilja tog rješenja donosilac odluke uspoređuje sa idealnom točkom, dakle uspoređuje vrijednost funkcije cilja f^h (dobiveno od rješenja X^h u h -toj iteraciji) sa idealnom točkom f^* .

Ako su vrijednosti nekih funkcija cilja zadovoljavajuće, a neke ne, donosilac odluke dozvoljava smanjenje funkcija cilja f_p za neku veličinu Δf_p . Ta veličina je dakle prihvatljiva veličina smanjenja p -te funkcije cilja.

Nakon toga formira se novi skup mogućih rješenja i u njemu se traži novo kompromisno rješenje. Taj novi skup mogućih rješenja (u $h+1$ iteraciji) sastoji se od svih ograničenja koji su tvorili skup S i od nekih dodatnih ograničenja. Prvo dodatno ograničenje smanjuje vrijednost p -te funkcije cilja, tj. dozvoljava da vrijednost p -te funkcije cilja bude smanjena za Δf_p . Ostala dodatna ograničenja odnose se na funkcije s kojima donosilac odluke nije zadovoljan i one moraju povećati svoje vrijednosti ili barem ostati jednake.

Taj novi skup rješenja je S^{h+1} (skup mogućih rješenja u $h+1$ iteraciji) i izgleda ovako:

$$S^{h+1} = \begin{cases} S^h (\text{ili } S^1) \\ f_p(X) \geq f_p(X^h) - \Delta f_p \\ f_k(X) \geq f_k(X^h), k \neq p, k = 1, \dots, r \end{cases}$$

Rješavanjem ponovo pomoćnog problema linearnog programiranja, ali sad sa novim skupom mogućih rješenja, dobiva se novo kompromisno rješenje.

Postupak se nastavlja sve dok donosilac odluke nije zadovoljan sa obje funkcije cilja, tj. do najboljeg kompromisnog rješenja.

8.2 Primjer problema riješenog *STEP* metodom

U zadatku je potrebno, za zadane funkcije cilja i linearna ograničenja:

- a) odrediti skup mogućih rješenja
- b) odrediti marginalna rješenja,
- c) odrediti idealno rješenje
- d) odrediti efikasna rješenja
 - 1.) odrediti pravce funkcije cilja
 - 2.) odrediti efikasne točke
- e) odrediti kompromisno rješenje (interaktivnom step metodom)
- f) nacrtati crtež.

$$\left. \begin{array}{l} \text{Max } f_1(X) = x_1 - 2x_2 \\ \text{Max } f_2(X) = x_1 + 3x_2 \\ \text{Max } f_3(X) = -x_1 \end{array} \right\} \text{ funkcije cilja}$$

$$\left. \begin{array}{l} 3x_1 - x_2 \geq 2 \\ 1x_1 + 2x_2 \leq 10 \\ -2x_1 + 3x_2 \geq 1 \\ x_1, x_2 \geq 0 \end{array} \right\} \text{ linearna ograničenja}$$

8.2.1 Određivanje skupa mogućih rješenja

Prvo u koordinatni sustav ucrtamo pravce linija ograničenja i odredimo skup mogućih rješenja. To se radi tako da se, npr. ograničenju $1x_1 + 2x_2 \leq 10$ znak nejednakosti zamjeni sa znakom jednakosti i pronađu se točke x_1, x_2 .

$$1x_1 + 2x_2 = 10$$

$$3x_1 - x_2 = 2$$

$$-2x_1 + 3x_2 = 1$$

$$x_1 = 10$$

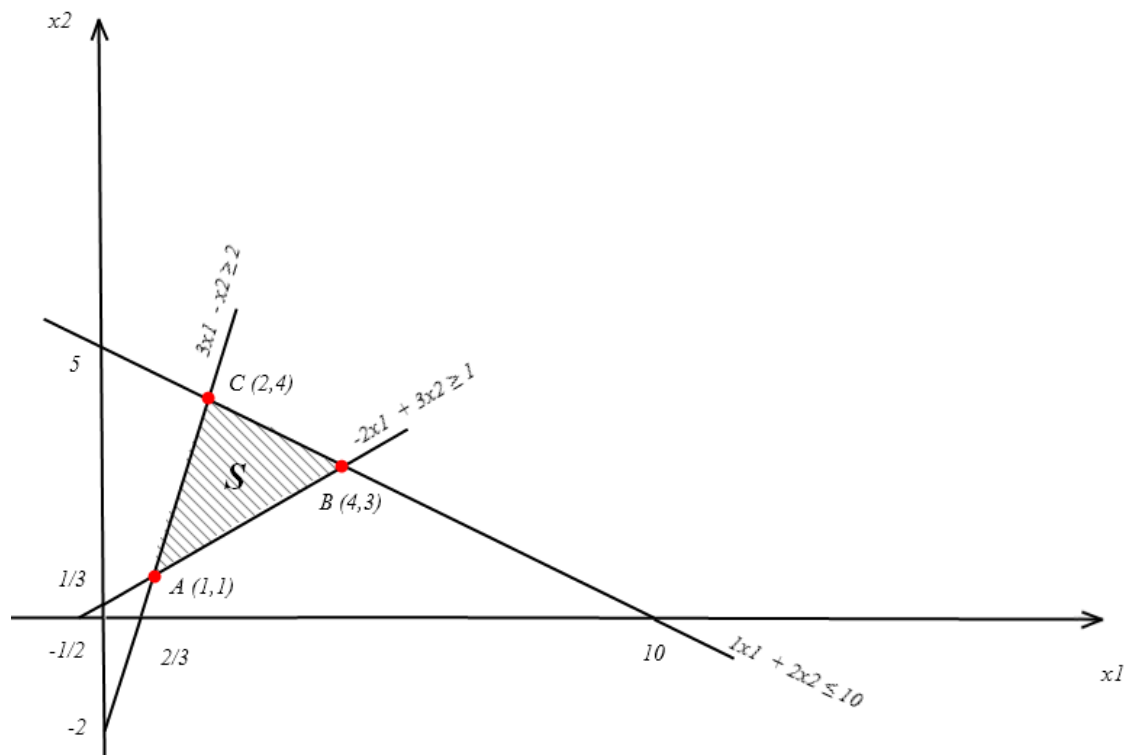
$$x_2 = -2$$

$$x_1 = -1/2$$

$$2x_2 = 10 \Rightarrow x_2 = 5$$

$$x_1 = 2/3$$

$$x_2 = 1/3$$



Slika 3. Grafički prikaz skupa mogućih rješenja i ekstremnih točaka

Skup mogućih rješenja prikazan je na slici 8.1 i vidi se da sadrži tri ekstremne točke. Treba, nadalje, odrediti marginallna rješenja.

8.2.2 Određivanje marginalnih rješenja

Marginalna rješenja dobivamo računajući vrijednosti funkcije cilja za sve ekstremne točke.

$$\begin{array}{ll} z_1 = x_1 - 2x_2 & A(1,1) \Rightarrow \mathbf{z_1=-1} ; z_2=4 ; \mathbf{z_3=-1} \\ z_2 = x_1 + 3x_2 \Rightarrow & B(4,3) \Rightarrow z_1=-2 ; z_2=13 ; z_3=-4 \\ z_3 = -x_1 & C(2,4) \Rightarrow z_1=-6 ; \mathbf{z_2=14} ; z_3=-2 \end{array}$$

Za prvu funkciju cilja optimalno rješenje je $X^{1*} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, odnosno $x_1^*=1$ i $x_2^*=1$, sa optimalnom vrijednošću prve funkcije cilja $f_1^* = -1$.

Za drugu funkciju cilja optimalno rješenje je $X^{2*} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$, odnosno $x_1^*=2$ i $x_2^*=4$, sa optimalnom vrijednošću prve funkcije cilja $f_2^* = 14$.

Za treću funkciju cilja optimalno rješenje je $X^{3*} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, odnosno $x_1^*=1$ i $x_2^*=1$, sa optimalnom vrijednošću prve funkcije cilja $f_3^* = -1$.

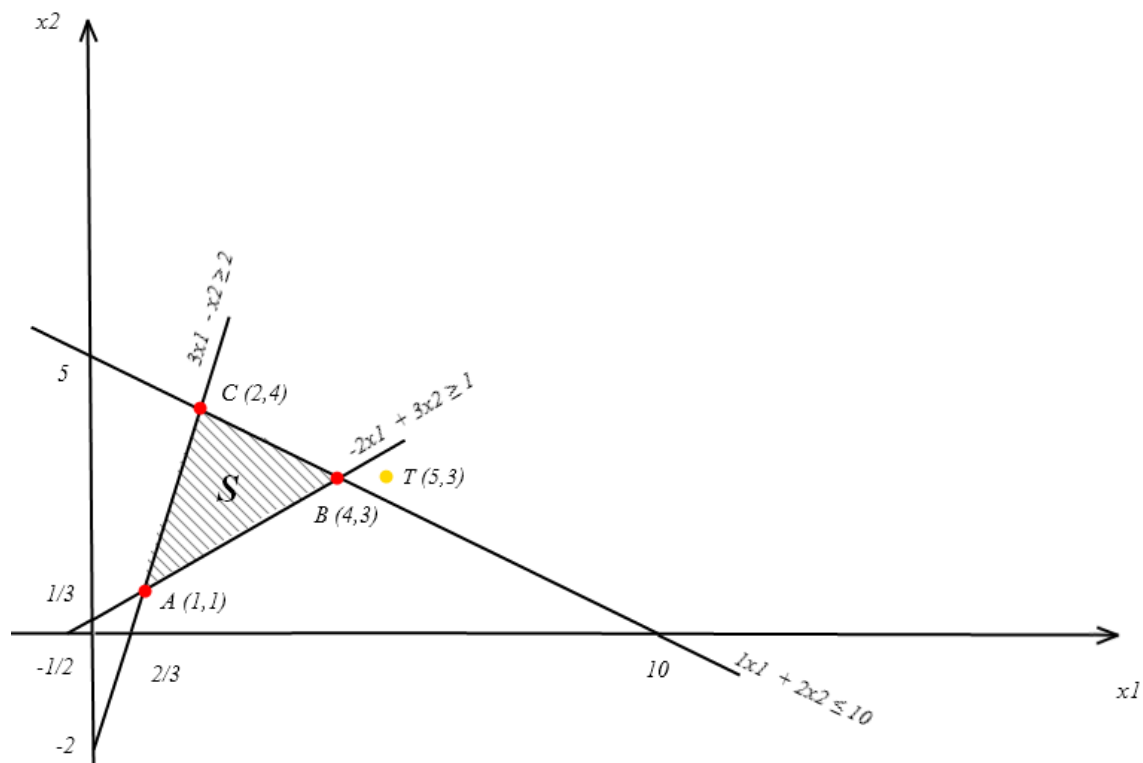
Dakle idealna vrijednost vektorske funkcije cilja (ideal) je vektor $f^* = \begin{bmatrix} -1 \\ 14 \\ -1 \end{bmatrix}$.

8.2.3 Određivanje idealnog rješenja

Idealno rješenje dobivamo računajući vrijednosti x_1 i x_2 , za maksimalne vrijednosti pojedinih funkcija cilja.

$$x_1 - 2x_2 = -1 \Rightarrow x_1 = 2x_2 - 1 \Rightarrow x_1 = 5$$

$$x_1 + 3x_2 = 14 \Rightarrow 2x_2 - 1 + 3x_2 = 14 \Rightarrow x_2 = 3$$



Slika 4. Prikaz idealne točke

Idealno rješenje je točka **T(5,3)**. Ako sada pogledamo na grafički prikaz na slici 8.2, vidimo da nam se idealno rješenje ne nalazi u skupu mogućih rješenja.

8.2.4 Određivanje efikasnog rješenja

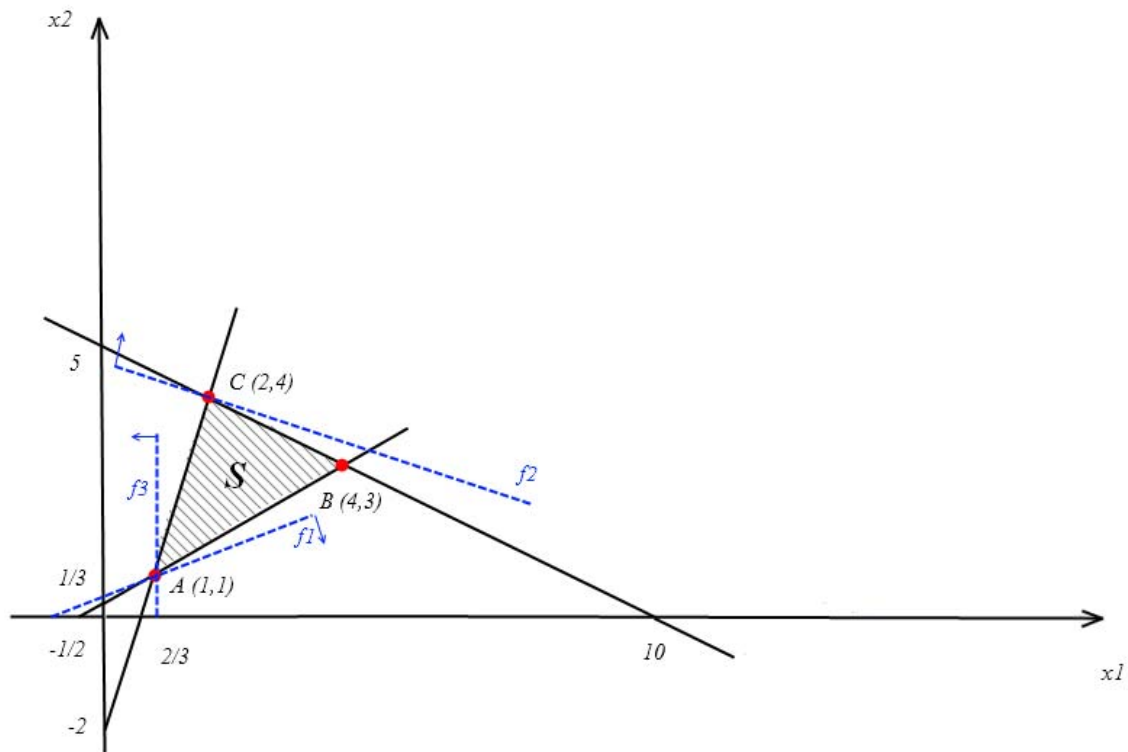
8.2.4.1 Određivanje pravaca funkcije cilja

Na crtež ucrtavamo pravce funkcije cilja u za svaku od marginalnih točaka.

$$x_1 - 2x_2 = -1 \Rightarrow x_1 = -1, \quad x_2 = 1/2$$

$$-x_1 = -1 \Rightarrow x_1 = 1, \quad x_2 = 0$$

$$x_1 + 3x_2 = 14 \Rightarrow x_1 = 14, \quad x_2 = 14/3$$



Slika 5. Prikaz optimalnih funkcija cilja za svaku od točaka

8.2.4.2 Određivanje efikasnih točaka

Efikasno rješenje X^E problema višekriterijalnog programiranja je moguće rješenje, $X^E \in S$, za koje ne postoji neko drugo moguće rješenje $X \in S$, takvo da je

$$f_i(X^E) \leq f_i(X) \text{ za sve } i = 1, 2, \dots, r, \text{ i}$$

$$f_i(X^E) \neq f_i(X) \text{ za neki } i = 1, 2, \dots, r$$

tj. ne postoji drugo moguće rješenje koje dominira nad X^E .

$$\left. \begin{array}{lcl} z_1(1,1) = -1 & = & z_1(1,1) = -1 \\ z_2(1,1) = 4 & = & z_2(1,1) = 4 \\ z_3(1,1) = -1 & = & z_3(1,1) = -1 \end{array} \right\} A(1,1) \quad \text{nije efikasna}$$

$$\left. \begin{array}{lcl} z_1(2,4) = -6 & \leq & z_1(1,1) = -1 \\ z_2(2,4) = 14 & \neq & z_2(1,1) = 4 \\ z_3(2,4) = -2 & \leq & z_3(1,1) = -1 \end{array} \right\} C(2,4) \quad \text{je efikasna}$$

$$\left. \begin{array}{lcl} z_1(4,3) = -2 & \geq & z_1(2,4) = -6 \\ z_2(4,3) = 13 & \neq & z_2(2,4) = 14 \\ z_3(4,3) = -4 & \geq & z_3(2,4) = -2 \end{array} \right\} B(4,3) \quad \text{je efikasna}$$

Drugim riječima efikasno rješenje problema višekriterijalnog programiranja je ono moguće rješenje za koje ne postoji neko drugo moguće rješenje koje je za barem po jednom kriteriju bolje, a po ostalima nije gore od njega.

8.2.5 Određivanje kompromisnog rješenja

Uvođenje **payoff matrice** :

Tablica se popunjava na sljedeći način. $f_j(X^{k*})$ je vrijednost j -te funkcije cilja za ono rješenje koje maksimizira k -tu funkciju cilja. U j -tom retku nalaze se vrijednosti j -te funkcije cilja za marginalna rješenja X^{k*} ($k = 1, 2, \dots, r$).

U našem primjeru nije potrebno raditi payoff matricu za točku B(2,4) jer ona nema niti jedan maksimum funkcije cilja.

Tablica 3. Payoff matrica

	$X^{1*} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$X^{1*} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$
f_1	-1	-6
f_2	4	14
f_3	-1	-2

Označimo sa f_k^{\min} minimalnu vrijednost u svakom retku payoff matrice

$$f_k^{\min} = \begin{bmatrix} -6 \\ 4 \\ -2 \end{bmatrix} \quad f_k^* = \begin{bmatrix} -1 \\ 14 \\ -1 \end{bmatrix}$$

i izračunamo parametre α_k na temelju ovih formula:

$$\alpha_k = \begin{cases} \frac{f_k^* - f_k^{\min}}{f_k^*} \cdot \frac{1}{\sqrt{\sum_{j=1}^n (c_{kj}^2)}}, f_k^* > 0 \\ \frac{f_k^{\min} - f_k^*}{f_k^{\min}} \cdot \frac{1}{\sqrt{\sum_{j=1}^n (c_{kj}^2)}}, f_k^* \leq 0 \end{cases}$$

Za naš primjer proračun izgleda ovako:

$$\frac{f_1^{\min} - f_1^*}{f_1^{\min}} \cdot \frac{1}{\sqrt{\sum_{j=1}^n (c_{1j}^2)}} = \frac{-6 - (-1)}{-6} \cdot \frac{1}{\sqrt{(1)^2 + (2)^2}} = 0,3727$$

$$\frac{f_2^* - f_2^{\min}}{f_2^*} \cdot \frac{1}{\sqrt{\sum_{j=1}^n (c_{2j}^2)}} = \frac{14 - 4}{14} \cdot \frac{1}{\sqrt{(1)^2 + (3)^2}} = 0,2259$$

$$\frac{f_3^{\min} - f_3^*}{f_3^{\min}} \cdot \frac{1}{\sqrt{\sum_{j=1}^n (c_{3j}^2)}} = \frac{-2 - (-1)}{-2} \cdot \frac{1}{\sqrt{(-1)^2 + (0)^2}} = 0,5$$

Zatim sa dobivenim podacima računamo parametre Π_k , na temelju ove formule:

$$\Pi_k = \frac{\alpha_k}{\sum_{j=1}^n \alpha_j}$$

Za naš primjer proračun izgleda ovako:

$$\Pi_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3} = \frac{0.3727}{0.3727 + 0.2259 + 0.5} = 0.3392$$

$$\Pi_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3} = \frac{0.2259}{0.3727 + 0.2259 + 0.5} = 0.2056$$

$$\Pi_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} = \frac{0.5}{0.3727 + 0.2259 + 0.5} = 0.4551$$

Pri čemu je :

$$\Pi_1 + \Pi_2 + \Pi_3 = 1$$

Nakon toga rješavamo pridruženi problem linearnog programiranja, po ovim formulama:

$$\text{Min } \lambda$$

$$[f_k^* - C^k X] \cdot \Pi_k \leq \lambda, k = 1, 2, \dots, r$$

$$X \in S$$

$$\lambda \geq 0$$

Koji kao optimalno rješenje ima opet jedno od efikasnih rješenja početnog problema višekriterijalnog linearnog programiranja, a on u našem primjeru izgleda ovako:

$$\text{Min } \lambda$$

$$[-1 - (x_1 - 2x_2)] \cdot 0.3392 \leq \lambda$$

$$[14 - (x_1 + 3x_2)] \cdot 0.2056 \leq \lambda$$

$$[-1 - (-x_1)] \cdot 0.4551 \leq \lambda$$

$$X \in S, \lambda \geq 0$$

tj.

$$\lambda = x_3$$

$$\text{Min } f(x) = x_3$$

$$-0.3392x_1 + 0.6784x_2 - x_3 \leq 0.3392$$

$$-0.2056x_1 - 0.6168x_2 - x_3 \leq -2.8784$$

$$0.4551x_1 - x_3 \leq 0.4551$$

$$-3x_1 + x_2 \leq -2$$

$$1x_1 + 2x_2 \leq 10$$

$$2x_1 - 3x_2 \leq -1$$

Rješenje tog problema (koje se dobije simplex metodom) je također jedno od efikasnih rješenja

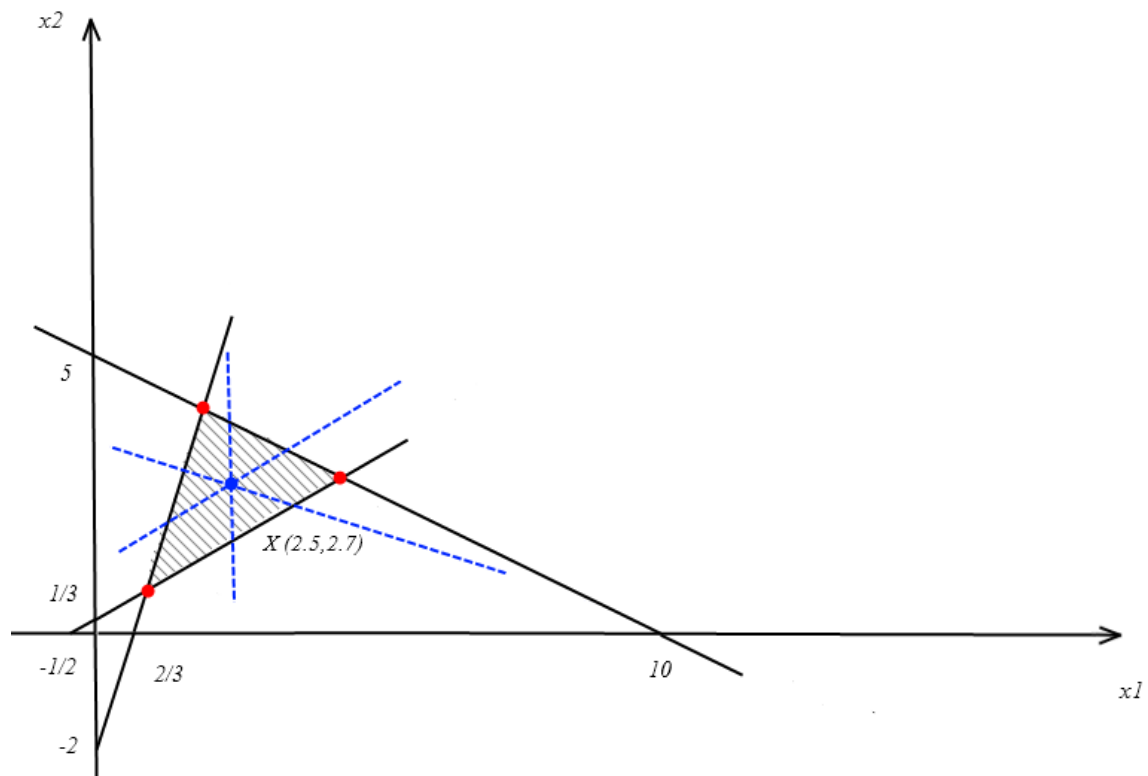
$$X^1 = \begin{bmatrix} 2.4868 \\ 2.7408 \\ 0.6766 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 = \lambda \end{matrix}$$

Odgovarajuće vrijednosti funkcija cilja su:

$$f^1 = \begin{bmatrix} f_1(X^1) \\ f_2(X^1) \\ f_3(X^1) \end{bmatrix} = \begin{bmatrix} -2.9948 \\ 10.7092 \\ -2.4868 \end{bmatrix}$$

To je prvo kompromisno rješenje koje se nudi donosiocu odluke. Vrijednosti funkcija cilja tog rješenja uspoređuju se sa idealnom točkom. Ukoliko nismo zadovoljni ide se u drugu iteraciju.

8.2.6 Grafički prikaz kompromisnog rješenja



Slika 6. Grafički prikaz 1. kompromisnog rješenja i njegovih pravaca funkcije cilja

8.3 Simulacija STEP metode programskim jezikom Matlab

Pomoću programskog jezika *Matlab* razvijen je simulacijski model *STEP* metode kojom se rješavaju problemi višekriterijalnog linearnog programiranja.

U nastavku je kao primjer prikazan način rešavanja problema višekriterijalnog linearnog programiranja za prethodno opisani problem:

$$\left. \begin{array}{l} \text{Max } f_1(X) = x_1 - 2x_2 \\ \text{Max } f_2(X) = x_1 + 3x_2 \\ \text{Max } f_3(X) = -x_1 \end{array} \right\} \text{ funkcije cilja}$$
$$\left. \begin{array}{l} 3x_1 - x_2 \geq 2 \\ 1x_1 + 2x_2 \leq 10 \\ -2x_1 + 3x_2 \geq 1 \end{array} \right\} \text{ linearna ograničenja}$$
$$x_1, x_2 \geq 0 \} \text{ broj varijabli}$$

1. Otvara se prvi ekran u kojem se definira problem (Slika 7.):

STEP metoda : Unos podataka

File Edit View Insert Tools Desktop Window Help

DEFINICIJA PROBLEMA

Broj funkcija cilja : 3

Broj varijabli : 2

Broj linearnih ograničenja : 3

Broj nelinearnih nejednadžbi : 0

Broj nelinearnih jednadžbi : 0

Nastavi

Slika 7. Definicija problema simulacijskim modelom

2. Klikom na tipku *Nastavi* program za upisane vrijednosti izgenerira niz linija za upis potrebnih podataka za svaku od funkcija cilja, varijabli i linearnih ograničenja (Slika 8.).

STEP metoda : Unos podataka

File Edit View Insert Tools Desktop Window Help

DEFINICIJA PROBLEMA

Broj funkcija cilja : 3

Broj varijabli : 2

Broj linearnih ograničenja : 3

Broj nelinearnih nejednadžbi : 0

Broj nelinearnih jednadžbi : 0

Nastavi

FUNKCIJE CILJA

Min		Izraz	Najveća vrijednost	Najmanja vrijednost
Min	f1			
Min	f2			
Min	f3			

VARIJABLE

	Donja granica	<=	Početna vrijednost	<=	Gornja granica
x1					
x2					

LINEARNA OGRANIČENJA

0	*	x1	+	0	*	x2	<=	0
0	*	x1	+	0	*	x2	<=	0
0	*	x1	+	0	*	x2	<=	0

NEMA NELINEARNIH NEJEDNADŽBI

NEMA NELINEARNIH JEDNADŽBI

Slika 8. Unos podataka u simulacijski model

Potrebno je upisati podatke u polja za svaku od linija. Izraz za funkciju cilja potrebno je upisati na način da se prvo upisuje koeficijent ispred varijable x1, zatim operator puta "*", te zatim "x1". Istu proceduru potrebno je ponoviti za svaku od varijabli. Ako varijabla nema koeficijenta, ispred iste se upisuje nula.

Podatke za donju i gornju granicu i početnu vrijednost varijable nije potrebno pisati ukoliko ne postoje ograničenja.

STEP metoda : Unos podataka

File Edit View Insert Tools Desktop Window Help

Broj nelinearnih jednadžbi: 0

Nastavi

FUNKCIJE CILJA

Min						
Min	f1	$1 \cdot x_1 - 2 \cdot x_2$	Najveća vrijednost	Najmanja vrijednost		
Min	f2	$1 \cdot x_1 + 3 \cdot x_2$	Najveća vrijednost	Najmanja vrijednost		
Min	f3	$-1 \cdot x_1 + 0 \cdot x_2$	Najveća vrijednost	Najmanja vrijednost		

VARIJABLE

x1	Donja granica	<=	Pocetna vrijednost	<=	Gornja granica	
x2	Donja granica	<=	Pocetna vrijednost	<=	Gornja granica	

LINEARNA OGRANIČENJA

-3	*	x1	+	1	*	x2	<=	-2
1	*	x1	+	2	*	x2	<=	10
2	*	x1	+	-3	*	x2	<=	-1

NEMA NELINEARNIH NEJEDNADŽBI

NEMA NELINEARNIH JEDNADŽBI

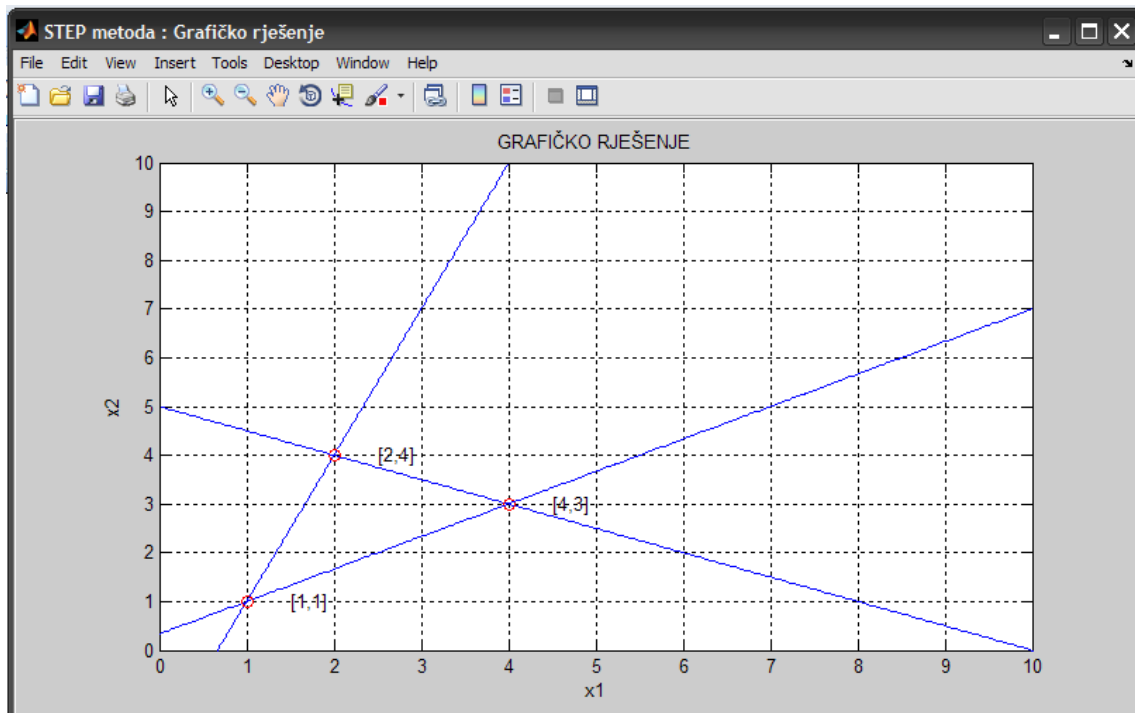
IZBOR SOLVERA

STEP METODA

Riješi

Slika 9. Prikaz unešenih podataka u simulacijskom modelu

- Nakon unosa svih podataka potrebno je kliknuti na gumb "Riješi" te se dobiva grafički prikaz rješenja (Slika 10.) i ekran sa prikazom prve *payoff* matrice (Slika 11.).



Slika 10. Grafičko rješenje simulacijskog modela

STEP metoda : Analitičko rješenje

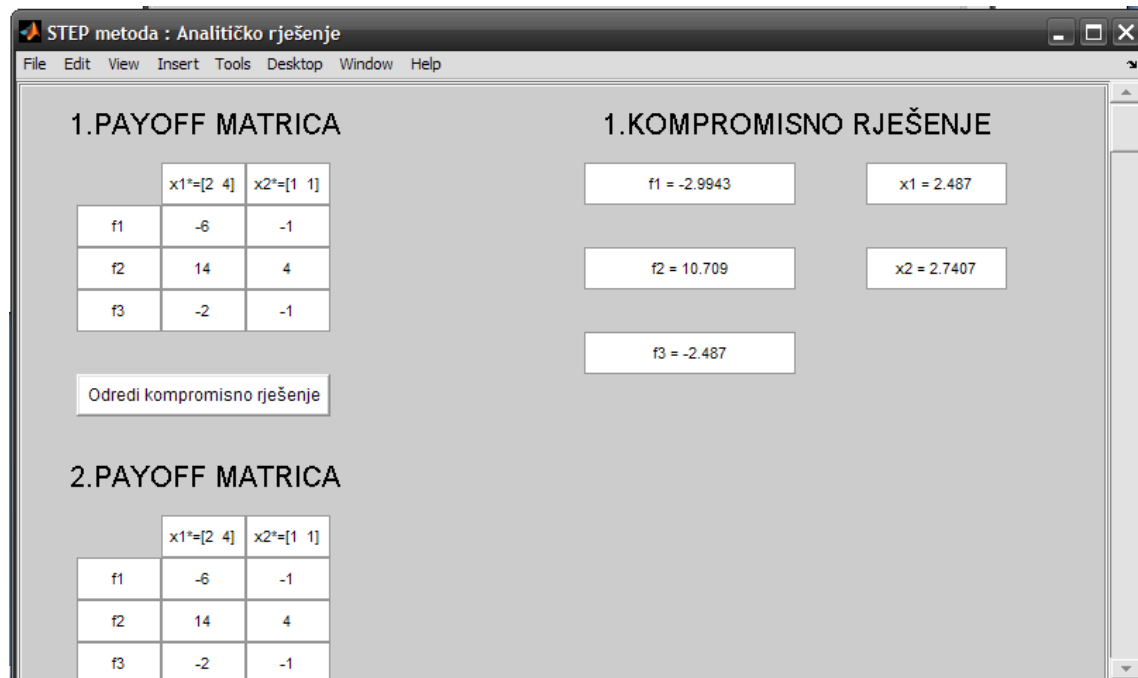
1. PAYOFF MATRICA

	$x1^*=[2 \ 4]$	$x2^*=[1 \ 1]$
f1	-6	-1
f2	14	4
f3	-2	-1

Odredi kompromisno rješenje

Slika 11. Prikaz prve *payoff* matrice

4. Klikom na tipku *Odredi kompromisno rješenje* program za vrijednosti iz *payoff* matrice izgenerira prvo kompromisno rješenje i drugu *payoff* matricu (Slika 12.) .



Slika 12. Prikaz prvog kompromisnog rješenja

Ukoliko nismo u potpunosti zadovoljni sa prvim kompromisnim rješenjem intervencijom na željenim vrijednostima u drugoj *payoff* matrici i pritiskom na tipku *Odredi kompromisno rješenje* (ispod druge *payoff* matrice), izgenerirati će se drugo kompromisno rješenje.

9. ZAKLUČAK

Tendencija današnjeg globalnog tržišta ne ostavlja prostora dugotrajnom pronalaženju rješenja zadanih problema, već se isti moraju u što kraćem roku realizirati. Brojnim tehnologijama i pomagalima ljudi nastoje doskočiti toj problematici. Operativni problemi u poslovanju karakteristično ovise o velikom broju varijabli na koje je djelomice moguće utjecati. Donošenju ispravne odluke, moderne svjetske organizacije, pristupaju sa višekriterijalnim optimiranjem. Kod problema višekriterijalnog optimiranja, potrebno je naći uvjete koji daju minimalnu ili maksimalnu vrijednost naše funkcije cilja (vrijeme, dobit, troškovi). Programski paket *Matlab* u sebi ima naredbe za rješavanje simplex metode, ali nema *STEP* metodu. Iz tog razloga smo napravili ovaj simulacijski model kako bi iskoristili tu simplex metodu i matematičke mogućnosti programa za rješavanje *STEP* metode, te sve to ukomponirali u grafičko sučelje. Model je interaktivan što znači da obzirom na unesene podatke program generira potrebne linije i onda prema upisanim podacima u tim linijama rješava problem, crta grafičko rješenje i dopušta utjecanje na željeno kompromisno rješenje. Model još nije do kraja usavršen. Potrebno je riješiti problem crtanja grafičkog rješenja kada imamo više od dvije varijable i usavršiti algoritam pronalaska ekstremnih točaka. Postoje neki paketi koji rješavaju razne metode optimiranja, ali nema paketa koji ih ima sve u jednom sa mogućnost izbora željenog modela. Ovaj simulacijski model je postavio izgled grafičkog sučelja, algoritam za prvu metodu i ostavio mogućnost implementacije drugih modela, kako bi se dobio jedan sveobuhvatan alat za višekriterijalno optimiranje.

10. PRILOG

10.1 M-file sa kodom simulacijskog modela

```
function GUI

clc
clear all
screensize = get(0,'ScreenSize');
assignin('base','screensize',screensize);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%***** kreiranje početnog ekrana
w_unos_podataka=figure('Name','STEP metoda : Unos
podataka','NumberTitle','off','Position',[screensize(1),screensize(2)/2,screensize(3)/2,screensize(4)]);

%***** paneli za slider
panel_g = uipanel('Parent',w_unos_podataka,'BackgroundColor',[0.8 0.8 0.8],'Position',[0 0 0.97 1]);
panel = uipanel('Parent',panel_g,'BackgroundColor',[0.8 0.8 0.8],'Position',[0 -1 1 2]);
set(gca,'Parent',panel,'visible','off');
assignin('base','panel',panel);

%***** kreiranje slidera
s = uicontrol('Style','Slider','Parent',w_unos_podataka,'Units','normalized','Position',[0.97 0.035 0.03
0.965],'Value',1,'Callback',{@slider_callback,panel});

%***** Definicija naslov kontrola
naziv_naslava={'Naslov_'};
default_text={'DEFINICIJA PROBLEMA'};
x_koordinata=280;
x_inkrement=0;
y_koordinata=1910;
y_inkrement=40;
sirina=240;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

%***** kreiranje naslova
kreiraj_naslov(naziv_naslava,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_
teksta,velicina_fonta,poravnavanje_teksta);

%***** Definicija tekst kontrola
naziv_texta={'Tekst_','Tekst_','Tekst_','Tekst_','Tekst_'};
default_text={'Broj funkcija cilja :','Broj varijabli :','Broj linearnih ograničenja :','Broj nelinearnih nejednadžbi
:','Broj nelinearnih jednadžbi :'};
x_koordinata=260;
x_inkrement=0;
y_koordinata=1870;
y_inkrement=40;
sirina=210;
visina=30;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[12.0];
poravnavanje_teksta='left';

%***** kreiranje početnog teksta
kreiraj_text(naziv_texta,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teks
ta,velicina_fonta,poravnavanje_teksta);

%***** Definicija edit tekst kontrola
naziv_edit_text_kontrola={'Edit_','Edit_','Edit_','Edit_','Edit_'};
default_text={'',' ',' ',' ',' '};
x_koordinata=495;
x_inkrement=0;
y_koordinata=1870;
y_inkrement=40;
sirina=40;
visina=30;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';
odziv_funkcije={@pokupi_broj_funkcija_cilja,@pokupi_broj_varijabli,@pokupi_broj_linearnih_ogranicenja,@pokupi_broj_nel
inearnih_nejednadzbi,@pokupi_broj_nelinearnih_jednadzbi};

%***** kreiranje početnih edit text kontrola
kreiraj_edit_text_kontrolu(naziv_edit_text_kontrola,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,vis
ina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

%***** Definicija pushbutton kontrola
naziv_pushbutton_kontrola={'Button_'};
default_text={'Nastavi'};
x_koordinata=260;
x_inkrement=0;
y_koordinata=1650;
```

```

y_inkrement=0;
sirina=275;
visina=40;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[10,0];
poravnavanje_teksta='center';
odziv_funkcije=@pritisak_gumba);

%***** kreiranje prvog pushbuttona

kreiraj_pushbutton_kontrolu(naziv_pushbutton_kontrole,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,v
isina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

%=====
%**** FUNKCIJE ZA CALLBACK(ODZIV)*****
%=====

%**** FUNKCIJA ZA odziv slidera
function slider_callback(s,eventdata,arg1)

    val = get(s,'Value');
    set(arg1,'Position',[0 -val 1 2]);

%**** FUNKCIJA ZA odziv slidera kod analitičkog rješenja
function slider_callback_2(s_2,eventdata,arg1)

    val = get(s_2,'Value');
    set(arg1,'Position',[0 -val 1 2]);

%=====

%**** FUNKCIJA ZA SKUPLJANJE UPISANIH POČETNIH BROJEVA

%**** funkcija za skupljanje upisanih početnih brojeva broja funkcija cilja
function [broj_funkcija_cilja] = pokupi_broj_funkcija_cilja(varargin)

    broj_funkcija_cilja=str2double(get(gcbo,'String'));
    assignin('base','broj_funkcija_cilja', broj_funkcija_cilja);

%**** funkcija za skupljanje upisanih početnih brojeva broja varijabli
function [broj_varijabli] = pokupi_broj_varijabli(varargin)

    broj_varijabli=str2double(get(gcbo,'String'));
    assignin('base','broj_varijabli', broj_varijabli);

%**** funkcija za skupljanje upisanih početnih brojeva broja linearnih ogracenjenja
function [broj_linearnih_ogracenjenja] = pokupi_broj_linearnih_ogracenjenja(varargin)

    broj_linearnih_ogracenjenja=str2double(get(gcbo,'String'));
    assignin('base','broj_linearnih_ogracenjenja', broj_linearnih_ogracenjenja);

%**** funkcija za skupljanje upisanih početnih brojeva broja nelinearnih nejednadzbi
function [broj_nelinearnih_nejednadzbi] = pokupi_broj_nelinearnih_nejednadzbi(varargin)

    broj_nelinearnih_nejednadzbi=str2double(get(gcbo,'String'));
    assignin('base','broj_nelinearnih_nejednadzbi', broj_nelinearnih_nejednadzbi);

%**** funkcija za skupljanje upisanih početnih brojeva broja nelinearnih jednadzbi
function [broj_nelinearnih_jednadzbi] = pokupi_broj_nelinearnih_jednadzbi(varargin)

    broj_nelinearnih_jednadzbi=str2double(get(gcbo,'String'));
    assignin('base','broj_nelinearnih_jednadzbi', broj_nelinearnih_jednadzbi);

%=====
%***Poziv funkcije za zaokruživanje broja na 4 decimale
function z = round2(x,y)

    z = round(x/y)*y;

%=====

%===== SIMPLEX METODA LINEARNOG PROGRAMIRANJA
function [X,Fx]=simplex_metoda(C1,A1,B1)

% Start with the default options
options = optimset;
% Modify options setting
options = optimset(options,'Display','final');
options = optimset(options,'Diagnostics','on');
options = optimset(options,'LargeScale','off');
options = optimset(options,'Simplex','on');

%[x]=linprog(f,[],[],A,b);
[X,fval,exitflag,output]=linprog(C1,[],[],A1,B1,[],[],[],options);
% [X,fval,exitflag,output]=linprog(C,A,B,Aeq,Beq,lb,ub,x0,options);
Fx=C1*X;

%=====

%**** FUNKCIJA ZA callback(odziv) pushbutton kontrole

function pritisak_gumba(varargin)

%****FUNKCIJE CILJA*****

```



```

el=evalin('base','broj_funkcija_cilja');

%***** Definicija naslova funkcije cilja
naziv_naslova={'Naslov_fc'};
default_text={'FUNKCIJE CILJA','NEMA FUNKCIJA CILJA'};
x_koordinata=280;
x_inkrement=0;
y_koordinata=1610;
y_inkrement=0;
sirina=240;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

%***** kreiranje naslova funkcije cilja
if el==0

kreiraj_naslov(naziv_naslova,default_text(2),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
else

kreiraj_naslov(naziv_naslova,default_text(1),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
end

%***** Definicija popupmenu kontrola funkcije cilja
naziv_popupmenu_kontrola={};
elementi_popupmenu_kontrola={};
x_koordinata=50;
x_inkrement=0;
sirina=45;
visina=20;
odziv_funkcije='';

%***** Definicija naziv funkcije kontrola funkcije cilja
naziv_edit_text_kontrola_nf_fc={};
default_text_nf_fc={};
x1_koordinata=105;
x1_inkrement=0;
sirina1=80;
visina1=20;
odziv_funkcije1='';

%***** Definicija izraz kontrola funkcije cilja
naziv_edit_text_kontrola_iz_fc={};
default_text_iz_fc={};
x2_koordinata=195;
x2_inkrement=0;
sirina2=380;
visina2=20;
odziv_funkcije2='';

%***** Definicija najveća vrijednost kontrola funkcije cilja
naziv_edit_text_kontrola_nv_fc={};
default_text_nv_fc={};
x3_koordinata=585;
x3_inkrement=0;
sirina3=100;
visina3=20;
odziv_funkcije3='';

%***** Definicija najmanja vrijednost kontrola funkcije cilja
naziv_edit_text_kontrola_nm_fc={};
default_text_nm_fc={};
x4_koordinata=695;
x4_inkrement=0;
sirina4=100;
visina4=20;
odziv_funkcije4='';

%***** zajednicke postavke svim elementima linije funkcije cilja
y_koordinata=1570;
y_inkrement=40;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';

%***** Izvrtiti petlju za kreiranje funkcija cilja ovisno o unesenom broju
for i=1:el

    naziv_popupmenu_kontrola(i)='min_max_';
    elementi_popupmenu_kontrola(i)={'Min','Max'};

    naziv_edit_text_kontrola_nf_fc(i)='naz_funk_cilja_';
    default_text_nf_fc(i)='f';

    naziv_edit_text_kontrola_iz_fc(i)='izraz_funk_cilja_';
    default_text_iz_fc(i)='Izraz';

    naziv_edit_text_kontrola_nv_fc(i)='nv_funk_cilja_';
    default_text_nv_fc(i)='Najveća vrijednost';

    naziv_edit_text_kontrola_nm_fc(i)='nm_funk_cilja_';
    default_text_nm_fc(i)='Najmanja vrijednost';

end

```

```

%**** kreiranje elemenata linije funkcije cilja

kreiraj_popupmenu_kontrolu(naziv_popupmenu_kontrola,elementi_popupmenu_kontrola,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,odziv_funkcije);

kreiraj_edit_text_kontrolu_nf_fc(naziv_edit_text_kontrola_nf_fc,default_text_nf_fc,x1_koordinata,x1_inkrement,y_koordinata,y_inkrement,sirina1,visina1,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije1);

kreiraj_edit_text_kontrolu_iz_fc(naziv_edit_text_kontrola_iz_fc,default_text_iz_fc,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirina2,visina2,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije2);

kreiraj_edit_text_kontrolu_nv_fc(naziv_edit_text_kontrola_nv_fc,default_text_nv_fc,x3_koordinata,x3_inkrement,y_koordinata,y_inkrement,sirina3,visina3,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije3);

kreiraj_edit_text_kontrolu_nm_fc(naziv_edit_text_kontrola_nm_fc,default_text_nm_fc,x4_koordinata,x4_inkrement,y_koordinata,y_inkrement,sirina4,visina4,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije4);

%=====

%****VARIABLE*****

e2=evalin('base','broj_varijabli');
y_koordinata=evalin('base','y_koordinata');

%**** Definicija naslova varijabli
naziv_naslova={'Naslov_var'};
default_text={'VARIJABLE','NEMA VARIJABLI'};
x_koordinata=280;
x_inkrement=0;
y_koordinata=y_koordinata-20;
y_inkrement=0;
sirina=240;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

%**** kreiranje naslova varijabli
if e2==0

kreiraj_naslov(naziv_naslova,default_text(2),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
else

kreiraj_naslov(naziv_naslova,default_text(1),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
end

%**** Definicija naziv varijabla kontrola varijable
naziv_edit_text_kontrola_naziv_var={};
default_text_naziv_var={};
x_koordinata=180;
x_inkrement=0;
odziv_funkcije='';

%**** Definicija donja granica kontrola varijable
naziv_edit_text_kontrola_dg_var={};
default_text_dg_var={};
x1_koordinata=270;
x1_inkrement=0;
odziv_funkcije1='';

%**** Definicija tekst kontrola varijabli
naziv_texta_var={};
default_text_var={};
x2_koordinata=350;
x2_inkrement=0;
sirina2=40;
visina2=20;
velicina_fonta2=[12.0];

%**** Definicija pocetna vrijednost kontrola varijable
naziv_edit_text_kontrola_pv_var={};
default_text_pv_var={};
x3_koordinata=390;
x3_inkrement=0;
odziv_funkcije3='';
sirina3=100;

%**** Definicija tekst kontrola varijabli
naziv_texta_var1={};
default_text_var1={};
x4_koordinata=490;

%**** Definicija donja granica kontrola varijable
naziv_edit_text_kontrola_gg_var={};
default_text_gg_var={};
x5_koordinata=530;
x5_inkrement=0;
odziv_funkcije5='';

%**** zajednicke postavke svim elementima linije varijabli
y_inkrement=40;
y_koordinata=y_koordinata-y_inkrement;
sirina=80;
visina=20;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';

```

```

%**** Izvrtiti petlju za kreiranje varijabli ovisno o unesenom broju
for i=1:e2

    naziv_edit_text_kontrole_naziv_var{i}='naziv_var_';
    default_text_naziv_var{i}='x';

    naziv_edit_text_kontrole_dg_var{i}='dg_var_';
    default_text_dg_var{i}='Donja granica';

    naziv_texta_var{i}='manje_jed_var_';
    default_text_var{i}='<=';

    naziv_edit_text_kontrole_pv_var{i}='pv_var_';
    default_text_pv_var{i}='Pocetna vrijednost';

    naziv_texta_varl{i}='manje_jed_varl_';
    default_text_varl{i}='<=';

    naziv_edit_text_kontrole_gg_var{i}='gg_var_';
    default_text_gg_var{i}='Gornja granica';

end

%**** kreiranje elemenata linije varijabli

kreiraj_edit_text_kontrolu_naziv_var(naziv_edit_text_kontrole_naziv_var,default_text_naziv_var,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

kreiraj_edit_text_kontrolu_dg_var(naziv_edit_text_kontrole_dg_var,default_text_dg_var,x1_koordinata,x1_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije1);

kreiraj_text_var(naziv_texta_var,default_text_var,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirina2,visina2,boja_pozadine,boja_teksta,velicina_fonta2,poravnavanje_teksta);

kreiraj_edit_text_kontrolu_pv_var(naziv_edit_text_kontrole_pv_var,default_text_pv_var,x3_koordinata,x3_inkrement,y_koordinata,y_inkrement,sirina3,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije3);

kreiraj_text_varl(naziv_texta_varl,default_text_varl,x4_koordinata,x4_inkrement,y_koordinata,y_inkrement,sirina2,visina2,boja_pozadine,boja_teksta,velicina_fonta2,poravnavanje_teksta);

kreiraj_edit_text_kontrolu_gg_var(naziv_edit_text_kontrole_gg_var,default_text_gg_var,x5_koordinata,x5_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije5);

%=====

%****LINEARNA OGRANICENJA*****

e3=evalin('base','broj_linearnih_ogranicenja');
y_koordinata=evalin('base','y_koordinata');

%**** Definicija naslova linearnih ogranicenja
naziv_naslova={'Naslov_lin_ogr'};
default_text={'LINEARNA OGRANICENJA','NEMA LINEARNIH OGRANICENJA'};
x_koordinata=240;
x_inkrement=0;
y_koordinata=y_koordinata-20;
y_inkrement=0;
sirina=320;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

%**** kreiranje naslova linearnih ogranicenja
if e3==0

kreiraj_naslov(naziv_naslova,default_text(2),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
else

kreiraj_naslov(naziv_naslova,default_text(1),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
end

%**** Definicija A kontrola linearnih ogranicenja
naziv_edit_text_kontrole_A_lin_ogr={};
default_text_A_lin_ogr={};
x_koordinata=50;
x_inkrement=160;
odziv_funkcije='';

%**** Definicija tekst kontrola linearnih ogranicenja
naziv_texta_lin_ogr={};
default_text_lin_ogr={};
x1_koordinata=130;
x1_inkrement=160;
sirina1=20;
velicina_fonta1=[15.0];

%**** Definicija X kontrola linearnih ogranicenja
naziv_edit_text_kontrole_x_lin_ogr={};
default_text_x_lin_ogr={};
x2_koordinata=150;
x2_inkrement=160;
sirina2=40;
odziv_funkcije2='';

%**** Definicija tekst kontrola linearnih ogranicenja
naziv_texta_lin_ogr1={};

```

```

default_text_lin_ogr1={};
x3_koordinata=190;
x3_inkrement=160;
velicina_fonta3=[12.0];

%***** Definicija popupmenu kontrola linearnih ogranicenja
naziv_popupmenu_kontrola_lin_ogr={};
elementi_popupmenu_kontrola_lin_ogr={};
x4_koordinata=0;
x4_inkrement=0;
sirina4=40;
velicina_fonta4=[7.4];
odziv_funkcije4='';

%***** Definicija jednako kontrola linearnih ogranicenja
naziv_edit_text_kontrola_jednako_lin_ogr={};
default_text_jednako_lin_ogr={};
x5_koordinata=0;
x5_inkrement=0;
odziv_funkcije5='';

%***** zajednicke postavke svim elementima linije varijabli
y_inkrement=40;
y_koordinata=y_koordinata-y_inkrement;
sirina=80;
visina=20;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';

for z=1:e2

    naziv_edit_text_kontrola_A_lin_ogr{z}='A_';
    default_text_A_lin_ogr{z}='0';

    naziv_texta_lin_ogr{z}='puta_';
    default_text_lin_ogr{z}='';

    naziv_edit_text_kontrola_x_lin_ogr{z}='x_';
    default_text_x_lin_ogr{z}='x';

end

for q=1:e2-1

    naziv_texta_lin_ogr1{q}='plus_';
    default_text_lin_ogr1{q}='+';

end

for i=1:e3

    naziv_popupmenu_kontrola_lin_ogr{i}='manje_jednako_lin_ogr_';
    elementi_popupmenu_kontrola_lin_ogr{i}={'<=', '='};

    naziv_edit_text_kontrola_jednako_lin_ogr{i}='jednako_lin_ogr_';
    default_text_jednako_lin_ogr{i}='0';

end

%***** kreiranje elemenata linije linearnih ogranicenja

kreiraj_edit_text_kontrolu_A_lin_ogr(naziv_edit_text_kontrola_A_lin_ogr,default_text_A_lin_ogr,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

kreiraj_text_lin_ogr(naziv_texta_lin_ogr,default_text_lin_ogr,x1_koordinata,x1_inkrement,y_koordinata,y_inkrement,sirina1,visina,boja_pozadine,boja_teksta,velicina_fonta1,poravnavanje_teksta);

kreiraj_edit_text_kontrolu_x_lin_ogr(naziv_edit_text_kontrola_x_lin_ogr,default_text_x_lin_ogr,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirina2,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije2);

kreiraj_text_lin_ogr1(naziv_texta_lin_ogr1,default_text_lin_ogr1,x3_koordinata,x3_inkrement,y_koordinata,y_inkrement,sirina1,visina,boja_pozadine,boja_teksta,velicina_fonta3,poravnavanje_teksta);

kreiraj_popupmenu_kontrolu_lin_ogr(naziv_popupmenu_kontrola_lin_ogr,elementi_popupmenu_kontrola_lin_ogr,x4_koordinata,x4_inkrement,y_koordinata,y_inkrement,sirina4,visina,boja_pozadine,boja_teksta,velicina_fonta4,odziv_funkcije4);

kreiraj_edit_text_kontrolu_jednako_lin_ogr(naziv_edit_text_kontrola_jednako_lin_ogr,default_text_jednako_lin_ogr,x5_koordinata,x5_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije5);

%=====

%*****NELINEARNE NEJEDNADŽBE*****

e4=evalin('base','broj_nelinearnih_nejednadzbi');
y_koordinata=evalin('base','y_koordinata');

%***** Definicija naslova nelinearnih nejednadzbi
naziv_naslova={'Naslov_nelin_nejed'};
default_text={'NELINEARNE NEJEDNADŽBE','NEMA NELINEARNIH NEJEDNADŽBI'};
x_koordinata=235;
x_inkrement=0;
y_koordinata=y_koordinata-20;
y_inkrement=0;
sirina=330;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];

```

```

poravnavanje_teksta='center';

%***** kreiranje naslova nelinearnih nejednadžbi
if e4==0

kreiraj_naslov(naziv_naslova,default_text(2),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
else

kreiraj_naslov(naziv_naslova,default_text(1),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
end

%***** Definicija naziv funkcije kontrola nelinearnih nejednadžbi
naziv_edit_text_naziv_kontrole_nelin_nejed={};
default_text_naziv_nelin_nejed={};
x_koordinata=105;
x_inkrement=0;
odziv_funkcije='';

%***** Definicija izraz kontrola nelinearnih nejednadžbi
naziv_edit_text_kontrole_iz_nelin_nejed={};
default_text_iz_nelin_nejed={};
x1_koordinata=215;
x1_inkrement=0;
sirinal=380;
odziv_funkcijel='';

%***** Definicija tekst kontrola nelinearnih nejednadžbi
naziv_texta_nelin_nejed={};
default_text_nelin_nejed={};
x2_koordinata=595;
x2_inkrement=0;
velicina_fonta2=[12.0];

%***** zajednicke postavke svim elementima linije nelinearnih nejednadžbi
y_inkrement=40;
y_koordinata=y_koordinata-y_inkrement;
sirina=100;
visina=20;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';

for i=1:e4

    naziv_edit_text_naziv_kontrole_nelin_nejed{i}='naziv_nelin_nejed';
    default_text_naziv_nelin_nejed{i}='Naziv nejedndžbe';

    naziv_edit_text_kontrole_iz_nelin_nejed{i}='iz_nelin_nejed';
    default_text_iz_nelin_nejed{i}='Izraz';

    naziv_texta_nelin_nejed{i}='mj_nelin_nejed';
    default_text_nelin_nejed{i}='<=0';

end

%***** kreiranje elemenata linije nelinearnih nejednadžbi

kreiraj_edit_text_naziv_kontrole_nelin_nejed(naziv_edit_text_naziv_kontrole_nelin_nejed,default_text_naziv_nelin_nejed,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

kreiraj_edit_text_kontrole_iz_nelin_nejed(naziv_edit_text_kontrole_iz_nelin_nejed,default_text_iz_nelin_nejed,x1_koordinata,x1_inkrement,y_koordinata,y_inkrement,sirinal,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcijel);

kreiraj_text_nelin_nejed(naziv_texta_nelin_nejed,default_text_nelin_nejed,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta2,poravnavanje_teksta)

%=====

%*****NELINEARNE JEDNADŽBE*****

e5=evalin('base','broj_nelinearnih_jednadzbi');
y_koordinata=evalin('base','y_koordinata');

%***** Definicija naslova nelinearnih jednadžbi
naziv_naslova={'Naslov_nelin_jed'};
default_text={'NELINEARNE JEDNADŽBE','NEMA NELINEARNIH JEDNADŽBI'};
x_koordinata=240;
x_inkrement=0;
y_koordinata=y_koordinata-20;
y_inkrement=0;
sirina=320;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

%***** kreiranje naslova nelinearnih jednadžbi
if e5==0

kreiraj_naslov(naziv_naslova,default_text(2),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
else

kreiraj_naslov(naziv_naslova,default_text(1),x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);
end

```

```

end

%***** Definicija naziv funkcije kontrola nelinearnih jednadzbi
naziv_edit_text_naziv_kontrole_nelin_jed={};
default_text_naziv_nelin_jed={};
x_koordinata=105;
x_inkrement=0;
odziv_funkcije='';

%***** Definicija izraz kontrola nelinearnih jednadzbi
naziv_edit_text_kontrole_iz_nelin_jed={};
default_text_iz_nelin_jed={};
x1_koordinata=215;
x1_inkrement=0;
sirina=380;
odziv_funkcije1='';

%***** Definicija tekst kontrola nelinearnih jednadzbi
naziv_texta_nelin_jed={};
default_text_nelin_jed={};
x2_koordinata=595;
x2_inkrement=0;
velicina_fonta2=[12.0];

%***** zajednicke postavke svim elementima linije nelinearnih jednadzbi
y_inkrement=40;
y_koordinata=y_koordinata-y_inkrement;
sirina=100;
visina=20;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';

for i=1:e5

    naziv_edit_text_naziv_kontrole_nelin_jed(i)='naziv_nelin_jed';
    default_text_naziv_nelin_jed(i)='Naziv jedndzbe';

    naziv_edit_text_kontrole_iz_nelin_jed(i)='iz_nelin_jed';
    default_text_iz_nelin_jed(i)='Izraz';

    naziv_texta_nelin_jed(i)='mj_nelin_jed';
    default_text_nelin_jed(i)='0';

end

%***** kreiranje elemenata linije nelinearnih jednadzbi
kreiraj_edit_text_naziv_kontrole_nelin_jed(naziv_edit_text_naziv_kontrole_nelin_jed,default_text_naziv_nelin_jed,x_koordinata
,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkci
je);

kreiraj_edit_text_kontrole_iz_nelin_jed(naziv_edit_text_kontrole_iz_nelin_jed,default_text_iz_nelin_jed,x1_koordinata,x1_inkr
ement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije1);

kreiraj_text_nelin_jed(naziv_texta_nelin_jed,default_text_nelin_jed,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirin
a,visina,boja_pozadine,boja_teksta,velicina_fonta2,poravnavanje_teksta);

%=====
%***** IZBOR SOLVERA*****

y_koordinata=evalin('base','y_koordinata');

%***** Definicija naslova izbora solvera
naziv_naslova={'Naslov_sol'};
default_text={'IZBOR SOLVERA'};
x_koordinata=280;
x_inkrement=0;
y_koordinata=y_koordinata-20;
y_inkrement=0;
sirina=240;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

%***** kreiranje naslova
kreiraj_naslov(naziv_naslova,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_
teksta,velicina_fonta,poravnavanje_teksta);

%***** Definicija liste solvera
naziv_liste={'Listbox_sol'};
default_text={'STEP METODA'};
x_koordinata=300;
x_inkrement=0;
y_inkrement=40;
y_koordinata=y_koordinata-100;
sirina=200;
visina=80;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[10.0];
poravnavanje_teksta='center';
odziv_funkcije='';

```

```

%**** kreiranje listu solvera

kreiraj_listbox(naziv_liste,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_t
eksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

%=====

%****GUMB ZA RIJEŠAVANJE PROBLEMA*****

y_koordinata=evalin('base','y_koordinata');

%**** Definicija pushbutton kontrola
naziv_pushbutton_kontrola={'Button_r'};
default_text={'Riješi'};
x_koordinata=260;
x_inkrement=0;
y_koordinata=y_koordinata-40;
y_inkrement=0;
sirina=275;
visina=40;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[10.0];
poravnavanje_teksta='center';
odziv_funkcije=@graficko_rijesenje);

%**** kreiranje pushbuttona za riješavanje zadatka

kreiraj_pushbutton_kontrolu(naziv_pushbutton_kontrola,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,v
isina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

%=====

%****FUNKCIJE ZA KREIRANJE*****

%=====

%****FUNKCIJE ZA KREIRANJE POČETNIH ELEMENATA (i svih naslova, pushbuttona i listboxa)

%**** FUNKCIJA za kreiranje naslova
function
kreiraj_naslov(naziv_naslova,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_
teksta,velicina_fonta,poravnavanje_teksta)

    broj_kontrola=length(naziv_naslova);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_naslova{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','text','String',default_text{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_
_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Parent',panel);

        y_koordinata=y_koordinata-y_inkrement;
        x_koordinata=x_koordinata-x_inkrement;

    end

%**** FUNKCIJA za kreiranje teksta
function
kreiraj_text(naziv_texta,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teks
ta,velicina_fonta,poravnavanje_teksta)

    broj_kontrola=length(naziv_texta);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_texta{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','text','String',default_text{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_
_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Parent',panel);

        y_koordinata=y_koordinata-y_inkrement;
        x_koordinata=x_koordinata-x_inkrement;

    end

%**** FUNKCIJA za kreiranje edit tekst kontrola
function
kreiraj_edit_text_kontrolu(naziv_edit_text_kontrola,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,vis
ina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

    broj_kontrola=length(naziv_edit_text_kontrola);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrola{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_
_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_
funkcije{i},'Parent',panel);

        y_koordinata=y_koordinata-y_inkrement;
        x_koordinata=x_koordinata-x_inkrement;

```

```

end

%***** FUNKCIJA za kreiranje pushbutton kontrole
function
kreiraj_pushbutton_kontrolu(naziv_pushbutton_kontrole,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,v
isina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije)

    broj_kontrola=length(naziv_pushbutton_kontrole);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_pushbutton_kontrole{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','pushbutton','String',default_text{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor'
',boja_pozadine','ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',
odziv_funkcije,'Parent',panel);

        y_koordinata=y_koordinata-y_inkrement;
        x_koordinata=x_koordinata-x_inkrement;

    end

%***** FUNKCIJA za kreiranje listbox kontrole
function
kreiraj_listbox(naziv_liste,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_t
eksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije)

    broj_kontrola=length(naziv_liste);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_liste{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','listbox','String',default_text{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',b
oja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odz
iv_funkcije,'Parent',panel);

        y_koordinata=y_koordinata-y_inkrement;
        x_koordinata=x_koordinata-x_inkrement;

    end

    assignin('base','y_koordinata',y_koordinata);

%=====

%*****FUNKCIJE ZA KREIRANJE ELEMENATA FUNKCIJE CILJA

%***** FUNKCIJA za kreiranje popupmenu kontrole funkcije cilja
function
kreiraj_popupmenu_kontrolu(naziv_popupmenu_kontrole,elementi_popupmenu_kontrole,x_koordinata,x_inkrement,y_koordinata,y_inkre
ment,sirina,visina,boja_pozadine,boja_teksta,odziv_funkcije)

    broj_kontrola=length(naziv_popupmenu_kontrole);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_popupmenu_kontrole{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','popupmenu','String',elementi_popupmenu_kontrole{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'B
ackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'Callback',odziv_funkcije,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x_koordinata=x_koordinata-x_inkrement;

    end

    assignin('base','y_koordinata',y_koordinata);

%***** FUNKCIJA za kreiranje edit tekst kontrole naziv funkcije funkcije cilja
function
kreiraj_edit_text_kontrolu_nf_fc(naziv_edit_text_kontrole_nf_fc,default_text_nf_fc,xl_koordinata,xl_inkrement,y_koordinata,y
_inkrement,sirinal,visinal,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcijel);

    broj_kontrola=length(naziv_edit_text_kontrole_nf_fc);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrole_nf_fc{i}, num2str(i)];
        tekst=strcat(default_text_nf_fc{i}, num2str(i));

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',tekst,'Position',[xl_koordinata,y_koordinata,sirinal,visinal],'BackgroundColor',boja_pozadi
ne,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcij
el,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        xl_koordinata=xl_koordinata-xl_inkrement;

    end
end

```



```

%**** FUNKCIJA za kreiranje edit tekst kontrole izraz funkcije cilja
function
kreiraj_edit_text_kontrolu_iz_fc(naziv_edit_text_kontrole_iz_fc,default_text_iz_fc,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirina2,visina2,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije2);

    broj_kontrola=length(naziv_edit_text_kontrole_iz_fc);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrole_iz_fc{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_iz_fc{i},'Position',[x2_koordinata,y_koordinata,sirina2,visina2],'BackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije2,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x2_koordinata=x2_koordinata-x2_inkrement;

    end

%**** FUNKCIJA za kreiranje edit tekst kontrole najveća vrijednost funkcije cilja
function
kreiraj_edit_text_kontrolu_nv_fc(naziv_edit_text_kontrole_nv_fc,default_text_nv_fc,x3_koordinata,x3_inkrement,y_koordinata,y_inkrement,sirina3,visina3,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije3);

    broj_kontrola=length(naziv_edit_text_kontrole_nv_fc);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrole_nv_fc{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_nv_fc{i},'Position',[x3_koordinata,y_koordinata,sirina3,visina3],'BackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije3,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x3_koordinata=x3_koordinata-x3_inkrement;

    end

%**** FUNKCIJA za kreiranje edit tekst kontrole najmanja vrijednost funkcije cilja
function
kreiraj_edit_text_kontrolu_nm_fc(naziv_edit_text_kontrole_nm_fc,default_text_nm_fc,x4_koordinata,x4_inkrement,y_koordinata,y_inkrement,sirina4,visina4,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije4);

    broj_kontrola=length(naziv_edit_text_kontrole_nm_fc);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrole_nm_fc{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_nm_fc{i},'Position',[x4_koordinata,y_koordinata,sirina4,visina4],'BackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije4,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x4_koordinata=x4_koordinata-x4_inkrement;

    end

%=====
%****FUNKCIJE ZA KREIRANJE ELEMENATA VARIJABLI

%**** FUNKCIJA za kreiranje edit tekst kontrole naziv varijable
function
kreiraj_edit_text_kontrolu_naziv_var(naziv_edit_text_kontrole_naziv_var,default_text_naziv_var,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

    broj_kontrola=length(naziv_edit_text_kontrole_naziv_var);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrole_naziv_var{i}, num2str(i)];
        tekst=strcat(default_text_naziv_var{i}, num2str(i));

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',tekst,'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x_koordinata=x_koordinata-x_inkrement;

    end

%**** FUNKCIJA za kreiranje edit tekst kontrole donja granica varijable

```

```

function
kreiraj_edit_text_kontrolu_dg_var(naziv_edit_text_kontrola_dg_var,default_text_dg_var,x1_koordinata,x1_inkrement,y_koordinata
,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije1);

    broj_kontrola=length(naziv_edit_text_kontrola_dg_var);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrola_dg_var{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_dg_var{i},'Position',[x1_koordinata,y_koordinata,sirina,visina],'BackgroundCol
or',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback
',odziv_funkcije1,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x1_koordinata=x1_koordinata-x1_inkrement;

    end

    %**** FUNKCIJA za kreiranje teksta varijabli
function
kreiraj_text_var(naziv_texta_var,default_text_var,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirina2,visina2,boja_po
zadine,boja_teksta,velicina_fonta2,poravnavanje_teksta)

    broj_kontrola=length(naziv_texta_var);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_texta_var{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','text','String',default_text_var{i},'Position',[x2_koordinata,y_koordinata,sirina2,visina2],'BackgroundColo
r',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta2,'HorizontalAlignment',poravnavanje_teksta,'Parent'
,panel);

        y_koordinata=y_koordinata-y_inkrement;
        x2_koordinata=x2_koordinata-x2_inkrement;

    end

    %**** FUNKCIJA za kreiranje edit tekst kontrola pocetna vrijednost varijable
function
kreiraj_edit_text_kontrolu_pv_var(naziv_edit_text_kontrola_pv_var,default_text_pv_var,x3_koordinata,x3_inkrement,y_koordinata
,y_inkrement,sirina3,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije3);

    broj_kontrola=length(naziv_edit_text_kontrola_pv_var);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrola_pv_var{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_pv_var{i},'Position',[x3_koordinata,y_koordinata,sirina3,visina],'BackgroundCol
or',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callbac
k',odziv_funkcije3,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x3_koordinata=x3_koordinata-x3_inkrement;

    end

    %**** FUNKCIJA za kreiranje teksta varijabli
function
kreiraj_text_var1(naziv_texta_var1,default_text_var1,x4_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirina2,visina2,boja
_pozadine,boja_teksta,velicina_fonta2,poravnavanje_teksta)

    broj_kontrola=length(naziv_texta_var1);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_texta_var1{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','text','String',default_text_var1{i},'Position',[x4_koordinata,y_koordinata,sirina2,visina2],'BackgroundCol
or',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta2,'HorizontalAlignment',poravnavanje_teksta,'Parent'
,panel);

        y_koordinata=y_koordinata-y_inkrement;
        x4_koordinata=x4_koordinata-x2_inkrement;

    end

    %**** FUNKCIJA za kreiranje edit tekst kontrola gornja granica varijable
function
kreiraj_edit_text_kontrolu_gg_var(naziv_edit_text_kontrola_gg_var,default_text_gg_var,x5_koordinata,x5_inkrement,y_koordinata
,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije5);

    broj_kontrola=length(naziv_edit_text_kontrola_gg_var);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrola_gg_var{i}, num2str(i)];

```

```

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_gg_var{i},'Position',[x5_koordinata,y_koordinata,sirina,visina],'BackgroundCol
or',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback
',odziv_funkcije5,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x5_koordinata=x5_koordinata-x5_inkrement;

    end

    assignin('base', 'y_koordinata', y_koordinata);
    assignin('base', 'y_koordinata_lo', y_koordinata);

%=====
%****FUNKCIJE ZA KREIRANJE ELEMENATA LINEARNIH OGRANIČENJA

%**** FUNKCIJA za kreiranje edit tekst kontrole linearnih organicenja
function
kreiraj_edit_text_kontrolu_A_lin_ogr(naziv_edit_text_kontrola_A_lin_ogr,default_text_A_lin_ogr,x_koordinata,x_inkrement,y_koo
rdinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

    broj_kontrola=length(naziv_edit_text_kontrola_A_lin_ogr);
    panel=evalin('base','panel');
    e3=evalin('base','broj_linearnih_ogranicenja');
    y=evalin('base','y_koordinata_lo');

    for i=1:broj_kontrola
        y_koordinata=y-60;

        for z=1:e3

            naziv_kontrola=[naziv_edit_text_kontrola_A_lin_ogr{i}, num2str(z), num2str(i)];

            handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_A_lin_ogr{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundC
olor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callba
ck',odziv_funkcije,'Parent',panel);

            assignin('base', naziv_kontrola , handles.(naziv_kontrola));

            y_koordinata=y_koordinata-y_inkrement;

        end

        x_koordinata=x_koordinata+x_inkrement;

    end

    assignin('base', 'y_koordinata', y_koordinata);

%**** FUNKCIJA za kreiranje teksta linearnih organicenja
function
kreiraj_text_lin_ogr(naziv_texta_lin_ogr,default_text_lin_ogr,x1_koordinata,x1_inkrement,y_koordinata,y_inkrement,sirina,vis
ina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta)

    broj_kontrola=length(naziv_texta_lin_ogr);
    panel=evalin('base','panel');
    e3=evalin('base','broj_linearnih_ogranicenja');
    y=evalin('base','y_koordinata_lo');

    for i=1:broj_kontrola
        y_koordinata=y-60;

        for z=1:e3

            naziv_kontrola=[naziv_texta_lin_ogr{i}, num2str(i) num2str(z)];

            handles.(naziv_kontrola) =
uicontrol('Style','text','String',default_text_lin_ogr{i},'Position',[x1_koordinata,y_koordinata,sirina,visina],'BackgroundC
olor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Paren
t',panel);

            y_koordinata=y_koordinata-y_inkrement;

        end

        x1_koordinata=x1_koordinata+x1_inkrement;

    end

%**** FUNKCIJA za kreiranje edit tekst kontrole linearnih organicenja
function
kreiraj_edit_text_kontrolu_x_lin_ogr(naziv_edit_text_kontrola_x_lin_ogr,default_text_x_lin_ogr,x2_koordinata,x2_inkrement,y_k
oordinata,y_inkrement,sirina2,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije2);

    broj_kontrola=length(naziv_edit_text_kontrola_x_lin_ogr);
    panel=evalin('base','panel');
    e3=evalin('base','broj_linearnih_ogranicenja');
    y=evalin('base','y_koordinata_lo');

    for i=1:broj_kontrola
        y_koordinata=y-60;
        tekst=strcat(default_text_x_lin_ogr{i}, num2str(i));

        for z=1:e3

            naziv_kontrola=[naziv_edit_text_kontrola_x_lin_ogr{i}, num2str(z), num2str(i)];

```

```

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',tekst,'Position',[x2_koordinata,y_koordinata,sirina2,visina],'BackgroundColor',boja_pozadine,
e,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije
2,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
    end

    x2_koordinata=x2_koordinata+x2_inkrement;

end

%***** FUNKCIJA za kreiranje teksta linearnih organiceja
function
kreiraj_text_lin_ogr1(naziv_texta_lin_ogr1,default_text_lin_ogr1,x3_koordinata,x3_inkrement,y_koordinata,y_inkrement,sirina1,
visina,boja_pozadine,boja_teksta,velicina_fonta3,poravnavanje_teksta)

    broj_kontrola=length(naziv_texta_lin_ogr1);
    panel=evalin('base','panel');
    e3=evalin('base','broj_linearnih_organiceja');
    y=evalin('base','y_koordinata_lo');

    for i=1:broj_kontrola

        y_koordinata=y-60;

        for z=1:e3

            naziv_kontrola=[naziv_texta_lin_ogr1{i}, num2str(z), num2str(z)];

            handles.(naziv_kontrola) =
uicontrol('Style','text','String',default_text_lin_ogr1{i},'Position',[x3_koordinata,y_koordinata,sirina1,visina],'Background
Color','ForegroundColor',boja_teksta,'FontSize',velicina_fonta3,'HorizontalAlignment',poravnavanje_teksta,'Parent',panel);

            y_koordinata=y_koordinata-y_inkrement;
        end

        x3_koordinata=x3_koordinata+x3_inkrement;

    end

    assignin('base','x_koordinata', x3_koordinata);

%***** FUNKCIJA za kreiranje popupmenu kontrole linearnih organiceja
function
kreiraj_popupmenu_kontrolu_lin_ogr(naziv_popupmenu_kontrola_lin_ogr,elementi_popupmenu_kontrola_lin_ogr,x4_koordinata,x4_inkr
ement,y_koordinata,y_inkrement,sirina4,visina,boja_pozadine,boja_teksta,velicina_fonta4,odziv_funkcije4)

    x4_koordinata=evalin('base','x_koordinata');

    broj_kontrola=length(naziv_popupmenu_kontrola_lin_ogr);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_popupmenu_kontrola_lin_ogr{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','popupmenu','String',elementi_popupmenu_kontrola_lin_ogr{i},'Position',[x4_koordinata,y_koordinata,sirina4,
visina],'BackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta4,'Callback',odziv_funkcije4,'
Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;

    end

    assignin('base','x_koordinata', x4_koordinata);

%***** FUNKCIJA za kreiranje edit tekst kontrole linearnih organiceja
function
kreiraj_edit_text_kontrolu_jednako_lin_ogr(naziv_edit_text_kontrola_jednako_lin_ogr,default_text_jednako_lin_ogr,x5_koordinat
a,x5_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funk
cije5);

    x4_koordinata=evalin('base','x_koordinata');
    x5_koordinata=x4_koordinata+40;

    broj_kontrola=length(naziv_edit_text_kontrola_jednako_lin_ogr);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrola_jednako_lin_ogr{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_jednako_lin_ogr{i},'Position',[x5_koordinata,y_koordinata,sirina,visina],'Back
groundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,
'Callback',odziv_funkcije5,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;

    end

end

%=====

```

```

%****FUNKCIJE ZA KREIRANJE ELEMENATA NELINEARNIH NEJEDNADŽBI

%**** FUNKCIJA za kreiranje naziv kontrole nelinearnih nejednadzbi
function
kreiraj_edit_text_naziv_kontrole_nelin_nejed(naziv_edit_text_naziv_kontrole_nelin_nejed,default_text_naziv_nelin_nejed,x_koor
dinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_
funkcije);

    broj_kontrola=length(naziv_edit_text_naziv_kontrole_nelin_nejed);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_naziv_kontrole_nelin_nejed{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_naziv_nelin_nejed{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'Bac
kgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta
,'Callback',odziv_funkcije,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x_koordinata=x_koordinata-x_inkrement;

    end

%**** FUNKCIJA za kreiranje naziv za kreiranje nelinearnih nejednadzbi
function
kreiraj_edit_text_kontrole_iz_nelin_nejed(naziv_edit_text_kontrole_iz_nelin_nejed,default_text_iz_nelin_nejed,x1_koordinata,x
1_inkrement,y_koordinata,y_inkrement,sirinal,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkci
jel);

    broj_kontrola=length(naziv_edit_text_kontrole_iz_nelin_nejed);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_kontrole_iz_nelin_nejed{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_iz_nelin_nejed{i},'Position',[x1_koordinata,y_koordinata,sirinal,visina],'Back
groundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta
,'Callback',odziv_funkcijel,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;
        x1_koordinata=x1_koordinata-x1_inkrement;

    end

%**** FUNKCIJA za kreiranje teksta nelinearnih nejednadzbi
function
kreiraj_text_nelin_nejed(naziv_texta_nelin_nejed,default_text_nelin_nejed,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement
,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta2,poravnavanje_teksta)

    broj_kontrola=length(naziv_texta_nelin_nejed);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_texta_nelin_nejed{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','text','String',default_text_nelin_nejed{i},'Position',[x2_koordinata,y_koordinata,sirina,visina],'Backgrou
ndColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta2,'HorizontalAlignment',poravnavanje_teksta,'Pa
rent',panel);

        y_koordinata=y_koordinata-y_inkrement;
        x2_koordinata=x2_koordinata-x2_inkrement;

    end

    assignin('base', 'y_koordinata', y_koordinata);

%=====

%****FUNKCIJE ZA KREIRANJE ELEMENATA NELINEARNIH JEDNADŽBI

%**** FUNKCIJA za kreiranje naziv kontrole nelinearnih jednadzbi
function
kreiraj_edit_text_naziv_kontrole_nelin_jed(naziv_edit_text_naziv_kontrole_nelin_jed,default_text_naziv_nelin_jed,x_koordinata
,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkci
je);

    broj_kontrola=length(naziv_edit_text_naziv_kontrole_nelin_jed);
    panel=evalin('base','panel');

    for i=1:broj_kontrola

        naziv_kontrola=[naziv_edit_text_naziv_kontrole_nelin_jed{i}, num2str(i)];

        handles.(naziv_kontrola) =
uicontrol('Style','edit','String',default_text_naziv_nelin_jed{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'Backg
roundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'
Callback',odziv_funkcije,'Parent',panel);

        assignin('base', naziv_kontrola , handles.(naziv_kontrola));

        y_koordinata=y_koordinata-y_inkrement;

```

```

        x_koordinata=x_koordinata-x_inkrement;

    end

    %**** FUNKCIJA za kreiranje naziv kontrole nelinearnih jednadzbi
    function
    kreiraj_edit_text_kontrola_iz_nelin_jed(naziv_edit_text_kontrola_iz_nelin_jed,default_text_iz_nelin_jed,x1_koordinata,x1_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcijel);

        broj_kontrola=length(naziv_edit_text_kontrola_iz_nelin_jed);
        panel=evalin('base','panel');

        for i=1:broj_kontrola

            naziv_kontrola=[naziv_edit_text_kontrola_iz_nelin_jed(i), num2str(i)];

            handles.(naziv_kontrola) =
            uicontrol('Style','edit','String',default_text_iz_nelin_jed(i),'Position',[x1_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcijel,'Parent',panel);

            assignin('base', naziv_kontrola , handles.(naziv_kontrola));

            y_koordinata=y_koordinata-y_inkrement;
            x1_koordinata=x1_koordinata-x1_inkrement;

        end

        %**** FUNKCIJA za kreiranje teksta nelinearnih nejednadzbi
        function
        kreiraj_text_nelin_jed(naziv_texta_nelin_jed,default_text_nelin_jed,x2_koordinata,x2_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta2,poravnavanje_teksta)

            broj_kontrola=length(naziv_texta_nelin_jed);
            panel=evalin('base','panel');

            for i=1:broj_kontrola

                naziv_kontrola=[naziv_texta_nelin_jed(i), num2str(i)];

                handles.(naziv_kontrola) =
                uicontrol('Style','text','String',default_text_nelin_jed(i),'Position',[x2_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta2,'HorizontalAlignment',poravnavanje_teksta,'Parent',panel);

                y_koordinata=y_koordinata-y_inkrement;
                x2_koordinata=x2_koordinata-x2_inkrement;

            end

            assignin('base', 'y_koordinata', y_koordinata);

%=====

    %****FUNKCIJE ZA KREIRANJE PAYOFF MATRICE

    %**** FUNKCIJA za kreiranje naslova payoff matrice
    function
    kreiraj_naslov_payoff(naziv_naslova_payoff,default_text_payoff,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta)

        broj_kontrola=length(naziv_naslova_payoff);
        panel_analiticki=evalin('base','panel_analiticki');

        for i=1:broj_kontrola

            naziv_kontrola=[naziv_naslova_payoff(i), num2str(i)];

            handles.(naziv_kontrola) =
            uicontrol('Style','text','String',default_text_payoff(i),'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Parent',panel_analiticki);

            y_koordinata=y_koordinata-y_inkrement;
            x_koordinata=x_koordinata+x_inkrement;

        end

        assignin('base', 'y_koordinata_nas', y_koordinata);

    %**** FUNKCIJA za kreiranje edit tekst kontrole tocke payoff matrice
    function
    kreiraj_edit_text_kontrolu_tocke_payoff(naziv_edit_text_kontrola_tocke_payoff,default_text_tocke_payoff,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);

        broj_kontrola=length(naziv_edit_text_kontrola_tocke_payoff);
        panel_analiticki=evalin('base','panel_analiticki');
        T=evalin('base','T');
        a=evalin('base','z');

        for i=1:broj_kontrola

            b=a(i);

            naziv_kontrola=[naziv_edit_text_kontrola_tocke_payoff(i), num2str(i)];

            tekst=[default_text_tocke_payoff(i),num2str(i),'*','=','[',num2str(T(b,:)),'']];

```

```

        handles.(naziv_kontrole) =
uicontrol('Style','edit','String',tekst,'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_pozadine,
'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Parent',panel_analiticki);

        assignin('base', naziv_kontrole , handles.(naziv_kontrole));

        x_koordinata=x_koordinata+x_inkrement;
        y_koordinata=y_koordinata-y_inkrement;

    end

    %***** FUNKCIJA za kreiranje edit tekst kontrola funkcije payoff matrice
    function
kreiraj_edit_text_kontrolu_funkcije_payoff(naziv_edit_text_kontrole_funkcije_payoff,default_text_funkcije_payoff,x1_koordinat
a,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);

        broj_kontrola=length(naziv_edit_text_kontrole_funkcije_payoff);
        panel_analiticki=evalin('base','panel_analiticki');

        for i=1:broj_kontrola

            naziv_kontrole=[naziv_edit_text_kontrole_funkcije_payoff{i}, num2str(i)];

            tekst=[default_text_funkcije_payoff{i},num2str(i)];

            handles.(naziv_kontrole) =
uicontrol('Style','edit','String',tekst,'Position',[x1_koordinata,y1_koordinata,sirina,visina],'BackgroundColor',boja_pozadin
e,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Parent',panel_analiticki
);

            assignin('base', naziv_kontrole , handles.(naziv_kontrole));

            x1_koordinata=x1_koordinata+x1_inkrement;
            y1_koordinata=y1_koordinata-y1_inkrement;

        end

        %***** FUNKCIJA za kreiranje prvog stupca payoff matrice
        function
kreiraj_edit_text_kontrolu_prvi_stupac_payoff(naziv_edit_text_kontrole_prvi_stupac_payoff,default_text_prvi_stupac_payoff,x_k
oordinata,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,
odziv_funkcije);

            broj_kontrola=length(naziv_edit_text_kontrole_prvi_stupac_payoff);
            panel_analiticki=evalin('base','panel_analiticki');
            fx=evalin('base','fx');
            a=evalin('base','z');
            b=a(1);

            for i=1:broj_kontrola

                naziv_kontrole=[naziv_edit_text_kontrole_prvi_stupac_payoff{i}, num2str(i)];

                tekst=num2str(fx(b,i));

                handles.(naziv_kontrole) =
uicontrol('Style','edit','String',tekst,'Position',[x_koordinata,y1_koordinata,sirina,visina],'BackgroundColor',boja_pozadine
,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije,
'Parent',panel_analiticki);

                assignin('base', naziv_kontrole , handles.(naziv_kontrole));

                x_koordinata=x_koordinata+x1_inkrement;
                y1_koordinata=y1_koordinata-y1_inkrement;

            end

            %***** FUNKCIJA za kreiranje drugog stupca payoff matrice
            function
kreiraj_edit_text_kontrolu_drugi_stupac_payoff(naziv_edit_text_kontrole_drugi_stupac_payoff,default_text_drugi_stupac_payoff,
x2_koordinata,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_tek
sta,odziv_funkcije2);

                broj_kontrola=length(naziv_edit_text_kontrole_drugi_stupac_payoff);
                panel_analiticki=evalin('base','panel_analiticki');
                fx=evalin('base','fx');
                a=evalin('base','z');
                b=a(2);

                for i=1:broj_kontrola

                    naziv_kontrole=[naziv_edit_text_kontrole_drugi_stupac_payoff{i}, num2str(i)];

                    tekst=num2str(fx(b,i));

                    handles.(naziv_kontrole) =
uicontrol('Style','edit','String',tekst,'Position',[x2_koordinata,y1_koordinata,sirina,visina],'BackgroundColor',boja_pozadin
e,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije
2,'Parent',panel_analiticki);

                    assignin('base', naziv_kontrole , handles.(naziv_kontrole));

                    x2_koordinata=x2_koordinata+x1_inkrement;
                    y1_koordinata=y1_koordinata-y1_inkrement;

                end

                assignin('base', 'y_koordinata' , y1_koordinata);

```

```

%***** FUNKCIJA za kreiranje pushbutton kontrole za dobivanje kompromisnog rješenja
function
kreiraj_pushbutton_kontrolu_payoff(naziv_pushbutton_kontrole,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,s
irina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije)

    broj_kontrola=length(naziv_pushbutton_kontrole);
    panel_analiticki=evalin('base','panel_analiticki');

    for i=1:broj_kontrola

        naziv_kontrole=[naziv_pushbutton_kontrole{i}, num2str(i)];

        handles.(naziv_kontrole) =
uicontrol('Style','pushbutton','String',default_text{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor'
',boja_pozadine','ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',
odziv_funkcije,'Parent',panel_analiticki);

        y_koordinata=y_koordinata+y_inkrement;
        x_koordinata=x_koordinata+x_inkrement;

    end

    assignin('base','y_koordinata', y_koordinata);

%***** FUNKCIJA za kreiranje naslova kompromisnog rješenja
function
kreiraj_naslov_kompromisno(naziv_naslova,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_po
zadine,boja_teksta,velicina_fonta,poravnavanje_teksta)

    broj_kontrola=length(naziv_naslova);
    panel_analiticki=evalin('base','panel_analiticki');

    for i=1:broj_kontrola

        naziv_kontrole=[naziv_naslova{i}, num2str(i)];

        handles.(naziv_kontrole) =
uicontrol('Style','text','String',default_text{i},'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja
_pozadine,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Parent',panel_an
aliticki);

        assignin('base', naziv_kontrole , handles.(naziv_kontrole));

        y_koordinata=y_koordinata+y_inkrement;
        x_koordinata=x_koordinata+x_inkrement;

    end

%***** FUNKCIJA za kreiranje edit tekst kontrola kompromisnog rješenja
function
kreiraj_edit_text_kontrolu_kompromisno(naziv_edit_text_kontrole_kompromisno,default_text_kompromisno,x_koordinata,x_inkrement
,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

    kompromisno=evalin('base','kompromisno');
    broj_kontrola=length(naziv_edit_text_kontrole_kompromisno);
    panel_analiticki=evalin('base','panel_analiticki');

    for i=1:broj_kontrola

        naziv_kontrole=[naziv_edit_text_kontrole_kompromisno{i}, num2str(i)];

        tekst=['f',num2str(i),' = ',num2str(kompromisno(i))];

        handles.(naziv_kontrole) =
uicontrol('Style','edit','String',tekst,'Position',[x_koordinata,y_koordinata,sirina,visina],'BackgroundColor',boja_pozadine,
'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije,'
Parent',panel_analiticki);

        assignin('base', naziv_kontrole , handles.(naziv_kontrole));

        x_koordinata=x_koordinata+x_inkrement;
        y_koordinata=y_koordinata+y_inkrement;

    end

%***** FUNKCIJA za kreiranje edit tekst kontrola kompromisnog rješenja
function
kreiraj_edit_text_kontrolu_kompromisno_x(naziv_edit_text_kontrole_kompromisno_x,default_text_kompromisno_x,xl_koordinata,x_in
krement,y_koordinata,y_inkrement,sirinal,visinal,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcijel)
;

    X=evalin('base','X');
    broj_kontrola=length(naziv_edit_text_kontrole_kompromisno_x);
    panel_analiticki=evalin('base','panel_analiticki');

    for i=1:broj_kontrola

        naziv_kontrole=[naziv_edit_text_kontrole_kompromisno_x{i}, num2str(i)];

        tekst=['x',num2str(i),' = ',num2str(X(i))];

        handles.(naziv_kontrole) =
uicontrol('Style','edit','String',tekst,'Position',[xl_koordinata,y_koordinata,sirinal,visinal],'BackgroundColor',boja_pozadin
e,'ForegroundColor',boja_teksta,'FontSize',velicina_fonta,'HorizontalAlignment',poravnavanje_teksta,'Callback',odziv_funkcije
l,'Parent',panel_analiticki);

        assignin('base', naziv_kontrole , handles.(naziv_kontrole));

```



```

        xl_koordinata=xl_koordinata+x_inkrement;
        y_koordinata=y_koordinata-y_inkrement;

    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%***** FUNKCIJA ZA RIJEŠAVANJE PROBLEMA
function graficko_rijesenje(varargin)

%***** pokupljanje brojeva sa workspacea
e1=evalin('base','broj_funkcija_cilja');
e2=evalin('base','broj_varijabli');
e3=evalin('base','broj_linearnih_ogranicenja');
e4=evalin('base','broj_nelinearnih_nejednadzbi');
e5=evalin('base','broj_nelinearnih_jednadzbi');

%*****kreiraj matrice A,B i C
A=[];
B=[];
C=[];

%*****pokupi podatke za sve linije funkcije cilja
for i=1:e1

    naziv_kontrola1=['min_max_', num2str(i)];
    fc1=evalin('base',naziv_kontrola1);
    vrijednost_popup_fc=char(get(fc1,'string'));
    assignin('base', naziv_kontrola1 , vrijednost_popup_fc);

    naziv_kontrola2=['naz_funk_cilja_', num2str(i)];
    fc2=evalin('base',naziv_kontrola2);
    naziv_fc=char(get(fc2,'string'));
    assignin('base', naziv_kontrola2 , naziv_fc);

    naziv_kontrola3=['izraz_funk_cilja_', num2str(i)];
    fc3=evalin('base',naziv_kontrola3);
    izraz_fc=char(get(fc3,'string'));
    assignin('base', naziv_kontrola3 , izraz_fc);

    %*****prikazi linije funkcije cilja
    display(strcat(naziv_fc,'=',izraz_fc));

    %*****napuni matricu C
    ileft=1;

    for j=1:e2

        find_x=['x',num2str(j)];
        irlight = findstr(find_x, izraz_fc);
        c = izraz_fc(ileft:irlight-2);
        C(i,j)=str2double(c);

        if j>=10
            ileft =findstr(find_x, izraz_fc)+3;
        else
            ileft =findstr(find_x, izraz_fc)+2;
        end

    end

    naziv_kontrola4=['nvv_funk_cilja_', num2str(i)];
    fc4=evalin('base',naziv_kontrola4);
    nvv_fc=str2double(get(fc4,'string'));
    assignin('base', naziv_kontrola4 , nvv_fc);

    naziv_kontrola5=['nmv_funk_cilja_', num2str(i)];
    fc5=evalin('base',naziv_kontrola5);
    nmv_fc=str2double(get(fc5,'string'));
    assignin('base', naziv_kontrola5 , nmv_fc);

end

assignin('base', 'C', C);
%*****pokupi podatke za sve linije varijabla
for i=1:e2

    naziv_kontrola1=['naziv_var_', num2str(i)];
    var1=evalin('base',naziv_kontrola1);
    naziv_var=char(get(var1,'string'));
    assignin('base', naziv_kontrola1 , naziv_var);

    naziv_kontrola2=['dg_var_', num2str(i)];
    var2=evalin('base',naziv_kontrola2);
    dg_var=str2double(get(var2,'string'));
    assignin('base', naziv_kontrola2 , dg_var);

    naziv_kontrola3=['pv_var_', num2str(i)];
    var3=evalin('base',naziv_kontrola3);
    pv_var=str2double(get(var3,'string'));
    assignin('base', naziv_kontrola3 , pv_var);

    naziv_kontrola4=['gg_var_', num2str(i)];
    var4=evalin('base',naziv_kontrola4);
    gg_var=str2double(get(var4,'string'));
    assignin('base', naziv_kontrola4 , gg_var);

end

%*****pokupi podatke za sve linije linearnih ogranicenja
for i=1:e3

```

```

naziv_kontrola1=['A_', num2str(i)];

for j=1:e2

    naziv_kontrola2=[naziv_kontrola1, num2str(j)];
    a_lo=evalin('base',naziv_kontrola2);
    vrijednost_A=str2double(get(a_lo,'string'));
    assignin('base', naziv_kontrola2 , vrijednost_A);

    %*****napuni matricu A
    A(i,j)=[vrijednost_A];

end

end
assignin('base', 'A', A);

for i=1:e3

    naziv_kontrola3=['x_', num2str(i)];

    for j=1:e2

        naziv_kontrola4=[naziv_kontrola3, num2str(j)];
        x_lo=evalin('base',naziv_kontrola4);
        naziv_x=char(get(x_lo,'string'));
        assignin('base', naziv_kontrola4 , naziv_x);

    end

end

for i=1:e3

    naziv_kontrola5=['manje_jednako_lin_ogr_', num2str(i)];
    popup_lo=evalin('base',naziv_kontrola5);
    vrijednost_popup_lo=char(get(popup_lo,'string'));
    assignin('base', naziv_kontrola5 , vrijednost_popup_lo);

    naziv_kontrola6=['jednako_lin_ogr_', num2str(i)];
    jed_lo=evalin('base',naziv_kontrola6);
    vrijednost_jednako_lo=str2double(get(jed_lo,'string'));
    assignin('base', naziv_kontrola6 , vrijednost_jednako_lo);

    %*****napuni matricu B
    b=str2double(get(jed_lo,'string'));
    B(i)=[b];

end

assignin('base', 'B', B);
%*****pokupi podatke za sve linije nelinearnih nejednadžbi
for i=1:e4

    naziv_kontrola1=['naziv_nelin_nejed', num2str(i)];
    nelin_nejed=evalin('base',naziv_kontrola1);
    naziv_nelin_nejed=char(get(nelin_nejed,'string'));
    assignin('base', naziv_kontrola1 , naziv_nelin_nejed);

    naziv_kontrola2=['iz_nelin_nejed', num2str(i)];
    nelin_nejed2=evalin('base',naziv_kontrola2);
    izraz_nelin_nejed=char(get(nelin_nejed2,'string'));
    assignin('base', naziv_kontrola2 , izraz_nelin_nejed);

end

%*****pokupi podatke za sve linije nelinearnih jednadžbi
for i=1:e5

    naziv_kontrola1=['naziv_nelin_jed', num2str(i)];
    nelin_jed=evalin('base',naziv_kontrola1);
    naziv_nelin_jed=char(get(nelin_jed,'string'));
    assignin('base', naziv_kontrola1 , naziv_nelin_jed);

    naziv_kontrola2=['iz_nelin_jed', num2str(i)];
    nelin_jed2=evalin('base',naziv_kontrola2);
    izraz_nelin_jed=char(get(nelin_jed2,'string'));
    assignin('base', naziv_kontrola2 , izraz_nelin_jed);

end

%*****prikazi matrice A, B i C
display(A);
B=B';
display(B);
display(C);

%*****pokupi ekstremne točke
for i=1:e3

j=strcat(num2str(A(i,1)), '*x1+', num2str(A(i,2)), '*x2=', num2str(B(i)));

    if A(i,1)==0
        naziv_kontrola=['jed_zatocke', num2str(i)];
        x2=strcat(num2str(B(i)), '/', num2str(A(i,2)));
        handles.(naziv_kontrola)=char(x2);

    elseif A(i,2)==0
        naziv_kontrola=['jed_zatocke', num2str(i)];
        x2=strcat('0');
        handles.(naziv_kontrola)=char(x2);
    end
end

```

```

else
    naziv_kontrole=['jed_zatocke', num2str(i)];
    x2=solve(j,'x2');
    handles.(naziv_kontrole)=char(x2);

end

if i==2
    naziv_kontrola1=['jed_zatocke', num2str(i-1)];
    jednadzba=strcat(handles.(naziv_kontrola1),'=',handles.(naziv_kontrole));
    x1=solve(jednadzba,'x1');
    jednadzba2=strcat('x2','=',handles.(naziv_kontrola1));
    x2=solve(jednadzba2,'x2');
    x1=eval(x1);
    x2=eval(x2);
    a1=x1;
    a2=x2;
    T(i-1,:)= [a1 a2];
end

if i==3
    naziv_kontrola1=['jed_zatocke', num2str(i-1)];
    jednadzba=strcat(handles.(naziv_kontrola1),'=',handles.(naziv_kontrole));
    x1=solve(jednadzba,'x1');
    jednadzba2=strcat('x2','=',handles.(naziv_kontrola1));
    x2=solve(jednadzba2,'x2');
    x1=eval(x1);
    x2=eval(x2);
    a1=x1;
    a2=x2;
    T(i-1,:)= [a1 a2];

    naziv_kontrola2=['jed_zatocke', num2str(i-2)];
    jednadzba=strcat(handles.(naziv_kontrola2),'=',handles.(naziv_kontrole));
    x1=solve(jednadzba,'x1');
    jednadzba2=strcat('x2','=',handles.(naziv_kontrola2));
    x2=solve(jednadzba2,'x2');
    x1=eval(x1);
    x2=eval(x2);
    a1=x1;
    a2=x2;
    T(i,:)= [a1 a2];
end

end

assignin('base', 'T' , T);
%****prikazi matricu ekstremnih tocaka
display(T);

%****odredi marginalna rjesenja
for i=1:e1
    for j=1:e3
        fx(i,j)=C(j,1)*T(i,1)+C(j,2)*T(i,2);
    end
end

display(fx);
assignin('base', 'fx' , fx);
%****odredi maksimalne vrijednosti marginalnih rjesenja
for i=1:e1
    max_fx(i)=max(fx(:,i));
    pos(i)=find(fx(:,i)==max_fx(i));
    z=find(pos==i);
end

broj_stupaca=length(z);
assignin('base', 'broj_stupaca' , broj_stupaca);
display(max_fx);
assignin('base', 'z' , z);

%****odredi maksimalne vrijednosti marginalnih rjesenja
for i=1:2
    j_id=strcat(num2str(C(i,1)),'*x1+',num2str(C(i,2)),'*x2=',num2str(max_fx(i)));

    if C(i,1)==0
        naziv_kontrole=['jed_zatocke', num2str(i)];
        x2=strcat(num2str(max_fx(i)),'/',num2str(C(i,2)));
        handles.(naziv_kontrole)=char(x2);

    elseif C(i,2)==0
        naziv_kontrole=['jed_zatocke', num2str(i)];
        x2=strcat('0');
        handles.(naziv_kontrole)=char(x2);

    else
        naziv_kontrole=['jed_zatocke', num2str(i)];
        x2=solve(j_id,'x2');
        handles.(naziv_kontrole)=char(x2);

    end

    if i==2
        naziv_kontrola1=['jed_zatocke', num2str(i-1)];
        jednadzba=strcat(handles.(naziv_kontrola1),'=',handles.(naziv_kontrole));
        x1=solve(jednadzba,'x1');
        jednadzba2=strcat('x2','=',handles.(naziv_kontrola1));
        x2=solve(jednadzba2,'x2');
        x1=eval(x1);
        x2=eval(x2);
        a1=x1;
        a2=x2;
    end
end

```

```

        I=[a1 a2];
        end
    end

    idealno=strcat('Idealno rješenje je:', ' I = (' ,num2str(I),')');
    disp(idealno);

%**** kreiranje prozor za graficko rješenje
w_graficko_rjesenje=figure('Name','STEP metoda : Grafičko rješenje','NumberTitle','off','Position',[850,525,800,425]);
panel_graficki = uipanel('Parent',w_graficko_rjesenje,'BackgroundColor',[0.8 0.8 0.8],'Position',[0 0 1 1]);

%**** crtanje grafickog rješenja
ya=max(B);

for i=1:el
    j=strcat(num2str(A(i,1)),'*', 'x1', '+', num2str(A(i,2)),'*', 'x2', '=', num2str(B(i)));
    ezplot(j,[0,ya])
    hold on
    a=T(i,1);
    b=T(i,2);
    plot(a,b,'or');
    t=strcat([' ', num2str(a), ', ', num2str(b), ' ]');
    text(a+0.5,b,t);
end

xlabel('x1');
ylabel('x2');
title('GRAFIČKO RJEŠENJE');
grid;

%**** kreiranje prozor za analitičko rješenje
w_analiticko_rjesenje=figure('Name','STEP metoda : Analitičko rješenje','NumberTitle','off','Position',[850,20,800,425]);
panel_a = uipanel('Parent',w_analiticko_rjesenje,'BackgroundColor',[0.8 0.8 0.8],'Position',[0 0 0.97 1]);
panel_analiticki = uipanel('Parent',panel_a,'BackgroundColor',[0.8 0.8 0.8],'Position',[0 -1 1 2]);
set(gca,'Parent',panel_analiticki,'visible','off');
assignin('base','panel_analiticki', panel_analiticki);

%**** kreiranje slidera
s_2 = uicontrol('Style','Slider','Parent',w_analiticko_rjesenje,'Units','normalized','Position',[0.97 0 0.03
1], 'Value',1, 'Callback',{@slider_callback_2,panel_analiticki});

%**** Definicija naslov kontrola payoff matrice
naziv_naslava_payoff={'Naslov payoff 1'};
default_text_payoff={'1.PAYOFF MATRICA'};
x_koordinata=10;
x_inkrement=0;
y_koordinata=795;
y_inkrement=0;
sirina=240;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

kreiraj_naslov_payoff(naziv_naslava_payoff,default_text_payoff,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visin
a,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta)

%**** Definicija tocaka payoff tablice
naziv_edit_text_kontrole_tocke_payoff={};
default_text_tocke_payoff={};
x_koordinata=100;
x_inkrement=60;
y_koordinata=755;
y_inkrement=0;

%**** Definicija funkcija payoff tablice
naziv_edit_text_kontrole_funkcije_payoff={};
default_text_funkcije_payoff={};
x1_koordinata=40;

%**** Definicija prvog stupca podataka payoff tablice
naziv_edit_text_kontrole_prvi_stupac_payoff={};
default_text_prvi_stupac_payoff={};
odziv_funkcije='';

%**** Definicija drugog stupca podataka payoff tablice
naziv_edit_text_kontrole_drugi_stupac_payoff={};
default_text_drugi_stupac_payoff={};
x2_koordinata=160;
odziv_funkcije2='';

%**** zajednicke postavke svim elementima payoff matrice
x1_inkrement=0;
y1_koordinata=725;
y1_inkrement=30;
sirina=60;
visina=30;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';

for i=1:broj_stupaca
    naziv_edit_text_kontrole_tocke_payoff{i}='tocke_payoff_';

```

```

        default_text_tocke_payoff{i}='x';
    end

    for i=1:el
        naziv_edit_text_kontrole_funkcije_payoff{i}='funkcije_payoff_';
        default_text_funkcije_payoff{i}='f';
        naziv_edit_text_kontrole_prvi_stupac_payoff{i}='prvi_stupac_payoff_';
        default_text_prvi_stupac_payoff{i}={};
        naziv_edit_text_kontrole_drugi_stupac_payoff{i}='drugi_stupac_payoff_';
        default_text_drugi_stupac_payoff{i}={};
    end

kreiraj_edit_text_kontrolu_tocke_payoff(naziv_edit_text_kontrole_tocke_payoff,default_text_tocke_payoff,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);

kreiraj_edit_text_kontrolu_funkcije_payoff(naziv_edit_text_kontrole_funkcije_payoff,default_text_funkcije_payoff,x1_koordinata,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);

kreiraj_edit_text_kontrolu_prvi_stupac_payoff(naziv_edit_text_kontrole_prvi_stupac_payoff,default_text_prvi_stupac_payoff,x_koordinata,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

kreiraj_edit_text_kontrolu_drugi_stupac_payoff(naziv_edit_text_kontrole_drugi_stupac_payoff,default_text_drugi_stupac_payoff,x2_koordinata,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije2);

y_koordinata=evalin('base','y_koordinata');

%***** Definicija pushbutton kontrola
naziv_pushbutton_kontrola={'Button_pm'};
default_text={'Odredi kompromisno rješenje'};
x_koordinata=40;
x_inkrement=0;
y_koordinata=y_koordinata-30;
y_inkrement=0;
sirina=180;
visina=30;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[9.0];
poravnavanje_teksta='center';
odziv_funkcije=@kompromisno_rjesenje);

%***** Kreiranje pushbuttona za dobivanje kompromisnog rješenja

kreiraj_pushbutton_kontrolu_payoff(naziv_pushbutton_kontrola,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

%=====

%***** FUNKCIJA ZA RIJEŠAVANJE PROBLEMA
function kompromisno_rjesenje(varargin)

%***** kupljanje brojeva sa workspacea
el=evalin('base','broj_funkcija_cilja');
e2=evalin('base','broj_varijabli');
e3=evalin('base','broj_linearnih_ogranicenja');
bs=evalin('base','broj_stupaca');
A=evalin('base','A');
B=evalin('base','B');
C=evalin('base','C');

for i=1:bs

    irlight = 7;
    ileft=5;

    if bs>10
        irlight = 8;
        ileft=6;
    end

    naziv_kontrola1=['tocke_payoff_', num2str(i)];
    fc1=evalin('base',naziv_kontrola1);
    izraz_tocka=char(get(fc1,'string'));

    for j=1:bs

        z = izraz_tocka(ileft+1:irlight-1);
        X(i,j)=[str2double(z)];
        ileft=findstr(' ', izraz_tocka);
        irlight = findstr(']', izraz_tocka);

        if bs>2
            irlight = findstr(' ', izraz_tocka);
        elseif i==bs
            irlight = findstr(']', izraz_tocka);
        end

    end

end

for i=1:el

    naziv_kontrola1=['prvi_stupac_payoff_', num2str(i)];
    fx1=evalin('base',naziv_kontrola1);
    vrijednost_fx1=str2double(get(fx1,'string'));

```

```

        %*****napuni matricu FX1
        FX1(i)=[vrijednost_fx1];

        naziv_kontrola2=['drugi_stupac_payoff_', num2str(i)];
        fx2=evalin('base',naziv_kontrola2);
        vrijednost_fx2=str2double(get(fx2,'string'));

        %*****napuni matricu FX1
        FX2(i)=[vrijednost_fx2];

    end

    %***** racunanje faktora alfa
    for i=1:e1
        FX=[FX1;FX2];
        FX=FX';

        fk(i)=max(FX(i,:));
        fk_min(i)=min(FX(i,:));

        if fk(i)>0

            for j=1:e2
                r=C(i,j)^2;
                Z(i,j)=[r];
            end
            z=Z(i,:);
            s=sum(z);

            f=fk(i)-fk_min(i);
            t=f/fk(i);
            k=sqrt(s);
            o=1/k;
            alfa(i)=t*o;

        end

        if fk(i)<0

            for j=1:2
                r=C(i,j)^2;
                Z(i,j)=[r];
            end
            z=Z(i,:);
            s=sum(z);

            f=fk_min(i)-fk(i);
            t=f/fk_min(i);
            k=sqrt(s);
            o=1./k;
            alfa(i)=t*o;

        end

    end

    %***** racunanje faktora pi
    sum_alfa=sum(alfa);

    for i=1:e1

        py(i)=alfa(i)/sum_alfa;

    end

    %***** racunanje faktora lamda
    for i=1:e1
        naziv_kontrola=['izraz_funk_cilja_', num2str(i)];
        fc=evalin('base',naziv_kontrola);
        izraz_fc=char(fc);

        jed1=strcat('b1=',num2str(fk(i)), '*',num2str(py(i)));
        B2=solve(jed1,'b1');
        r1=round2(B2,1e-4);
        R1=eval(-r1);

        %***** punjenje matrice B4
        B4(i)=[R1];

        for j=1:e2+1

            if j==e2+1
                R2=-1;
            else
                jed2=strcat('b2=',num2str(C(i,j)), '*',num2str(py(i)));
                B3=solve(jed2,'b2');
                r2=round2(B3,1e-4);
                R2=eval(-r2);
            end

            %***** punjenje matrice A2
            A2(i,j)=[R2];

            %***** punjenje matrice C1
            if j<e2+1
                c1=0;
                C1(j)=[c1];
            end
        end
    end
end

```

```

        else
            c2=1;
            C1(j)=[c2];
        end

        %***** punjenje matrice A3
        if j==e2+1
            A(i,j)=0;
        end
        A3(i,j)=[A(i,j)];
    end

end

%***** punjenje konačne matrice matrice A1
for i=1:e1+e3

    for j=1:e2+1

        if i<=e1
            A1(i,j)=[A2(i,j)];
        else
            z=i-e1;
            A1(i,j)=[A3(z,j)];
        end
    end
end

%***** punjenje konačne matrice matrice B1
for i=1:e1+e3

    if i<=e1
        B1(i)=[B4(i)];
    else
        z=i-e1;
        B1(i)=[B(z)];
    end
end

display(A1);
B1=B1';
display(B1);
display(C1);

%***Poziv funkcije SIMPLEX metode i ispis rjesenja
[X,Fx]=simplex_metoda(C1,A1,B1);
display(X);
display(Fx)

for i=1:e1

    kompromisno(i)=X(1)*C(i,1)+X(2)*C(i,2);

end

assignin('base','kompromisno', kompromisno);
assignin('base','X', X);

%***** Definicija naslov kontrola payoff matrice
naziv_naslava={'Naslov kompromisno'};
default_text={'1.KOMPROMISNO RJEŠENJE'};
x_koordinata=400;
x_inkrement=0;
y_koordinata=795;
y_inkrement=0;
sirina=300;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

kreiraj_naslov_kompromisno(naziv_naslava,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta)

%***** Definicija edit text kontrola kompromisnog rješenja
naziv_edit_text_kontrole_kompromisno={};
default_text_kompromisno={};
x_koordinata=400;
sirina=150;
odziv_funkcije='';

%***** Definicija edit text kontrola kompromisnog rješenja
naziv_edit_text_kontrole_kompromisno_x={};
default_text_kompromisno_x={};
x1_koordinata=600;
sirina=100;
odziv_funkcije1='';

%***** zajednicke postavke svim elementim kompromisnog rješenja
x_inkrement=0;
y_koordinata=755;
y_inkrement=60;
visina=30;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';

```

```

for i=1:e1

naziv_edit_text_kontrole_kompromisno{i}='kompromisno';
default_text_kompromisno{i}={};

end

for i=1:e2

naziv_edit_text_kontrole_kompromisno_x{i}='kompromisno_x';
default_text_kompromisno_x{i}={};

end

kreiraj_edit_text_kontrolu_kompromisno(naziv_edit_text_kontrole_kompromisno,default_text_kompromisno,x_koordinata,x_inkrement
,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

kreiraj_edit_text_kontrolu_kompromisno_x(naziv_edit_text_kontrole_kompromisno_x,default_text_kompromisno_x,x1_koordinata,x_in
krement,y_koordinata,y_inkrement,sirina1,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcijel)
;

y_koordinata=evalin('base','y_koordinata');

%***** Definicija naslov kontrola payoff matrice 2
naziv_naslava_payoff={'Naslov_payoff_2'};
default_text_payoff={'2.PAYOFF MATRICA'};
x_koordinata=10;
x_inkrement=0;
y_koordinata=y_koordinata-60;
y_inkrement=0;
sirina=240;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

kreiraj_naslov_payoff(naziv_naslava_payoff,default_text_payoff,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visin
a,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta)

%***** Definicija tocaka payoff tablice
naziv_edit_text_kontrole_tocke_payoff={};
default_text_tocke_payoff={};
x_koordinata=100;
x_inkrement=60;
y_koordinata=y_koordinata-40;
y_inkrement=0;

%***** Definicija funkcija payoff tablice
naziv_edit_text_kontrole_funkcije_payoff={};
default_text_funkcije_payoff={};
x1_koordinata=40;

%***** Definicija prvog stupca podataka payoff tablice
naziv_edit_text_kontrole_prvi_stupac_payoff={};
default_text_prvi_stupac_payoff={};
odziv_funkcije='';

%***** Definicija drugog stupca podataka payoff tablice
naziv_edit_text_kontrole_drugi_stupac_payoff={};
default_text_drugi_stupac_payoff={};
x2_koordinata=160;
odziv_funkcije2='';

%***** zajednicke postavke svim elementima payoff matrice
x1_inkrement=0;
y1_koordinata=y_koordinata-30;
y1_inkrement=30;
sirina=60;
visina=30;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[8.0];
poravnavanje_teksta='center';

for i=1:bs

naziv_edit_text_kontrole_tocke_payoff{i}='tocke_payoff_';
default_text_tocke_payoff{i}='x';

end

for i=1:e1

naziv_edit_text_kontrole_funkcije_payoff{i}='funkcije_payoff_';
default_text_funkcije_payoff{i}='f';
naziv_edit_text_kontrole_prvi_stupac_payoff{i}='prvi_stupac_payoff_';
default_text_prvi_stupac_payoff{i}={};
naziv_edit_text_kontrole_drugi_stupac_payoff{i}='drugi_stupac_payoff_';
default_text_drugi_stupac_payoff{i}={};

end

kreiraj_edit_text_kontrolu_tocke_payoff(naziv_edit_text_kontrole_tocke_payoff,default_text_tocke_payoff,x_koordinata,x_inkrem
ent,y_koordinata,y_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);

```



```

kreiraj_edit_text_kontrolu_funkcije_payoff(naziv_edit_text_kontrole_funkcije_payoff,default_text_funkcije_payoff,x1_koordinat
a,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta);

kreiraj_edit_text_kontrolu_prvi_stupac_payoff(naziv_edit_text_kontrole_prvi_stupac_payoff,default_text_prvi_stupac_payoff,x_k
oordinata,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,
odziv_funkcije);

kreiraj_edit_text_kontrolu_drugi_stupac_payoff(naziv_edit_text_kontrole_drugi_stupac_payoff,default_text_drugi_stupac_payoff,
x2_koordinata,x1_inkrement,y1_koordinata,y1_inkrement,sirina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_tek
sta,odziv_funkcije2);

y_koordinata=evalin('base','y_koordinata');

%***** Definicija pushbutton kontrola
naziv_pushbutton_kontrola={'Button_pm'};
default_text={'Odredi kompromisno rješenje'};
x_koordinata=40;
x_inkrement=0;
y_koordinata=y_koordinata-30;
y_inkrement=0;
sirina=180;
visina=30;
boja_pozadine='white';
boja_teksta='black';
velicina_fonta=[9.0];
poravnavanje_teksta='center';
odziv_funkcije=@kompromisno_rjesenje_2);

%***** kreiranje pushbuttona za dobivanje kompromisnog rješenja

kreiraj_pushbutton_kontrolu_payoff(naziv_pushbutton_kontrole,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,s
irina,visina,boja_pozadine,boja_teksta,velicina_fonta,poravnavanje_teksta,odziv_funkcije);

%***** FUNKCIJA ZA RIJEŠAVANJE PROBLEMA
function kompromisno_rjesenje_2(varargin)

y_koordinata=evalin('base','y_koordinata_nas');

%***** Definicija naslov kontrola payoff matrice
naziv_naslava={'Naslov kompromisno 2'};
default_text={'2.KOMPROMISNO RJEŠENJE'};
x_koordinata=400;
x_inkrement=0;
y_koordinata=y_koordinata;
y_inkrement=0;
sirina=300;
visina=30;
boja_pozadine=[0.8 0.8 0.8];
boja_teksta='black';
velicina_fonta=[15.0];
poravnavanje_teksta='center';

kreiraj_naslov_kompromisno(naziv_naslava,default_text,x_koordinata,x_inkrement,y_koordinata,y_inkrement,sirina,visina,boja_po
zadine,boja_teksta,velicina_fonta,poravnavanje_teksta)

```

11. LITERATURA

[1] S. S. Rao. Engineering Optimization Theory and Practice, Fourth Edition. John Wiley & Sons, Inc. 2009.g.

[2] M. Zeleny. Linear Multiobjective Programming. Springer-Verlag Berlin 1970.g.