FAKULTET TEHNIČKIH NAUKA

Departman za proizvodno mašinstvo

Tehnoekonomska optimizacija i preduzetništvo

Tema:

ANALITIČKE METODE OPTIMIZACIJE

Dr Dejan Lukić

Analitičke metode optimizacije

Glavno obeležje analitičkih metoda sastoji se u tome da je **matematički model optimizacije datog objekta poznat** ili da se može postaviti budući da su **zakonitosti i pojave** unutar objekta **potpuno poznate**.

Gradijentna metoda

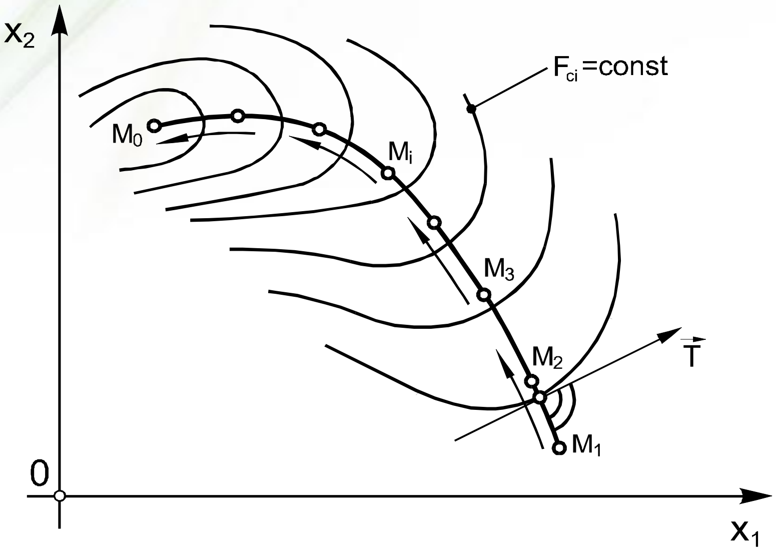
U grupu najčešće korišćenih metoda u optimizaciji raznovrsnih objekata, kao što su procesi i sistemi, spada gradijentna metoda. Razlog za to su njene osnovne osobine:

- ***Univerzalnost***, *tj. mogućnost metode da se pomoću nje optimiziraju i linearne i nelinearne funkcije optimizacije, bez ograničenja i sa ograničenjima, sa linearnim i nelinearnim ograničenjima, dakle, da se reše optimizacioni zadaci sa najopštijim modelom optimizacije.*
- ***Efikasnost*** *i relativna* ***jednostavnost procedure*** *rešavanja i najsloženijih optimizacionih zadataka.*
- ***Razvijeni algoritmi***, *odnosno procedure metode orijentisani su na upotrebi računara.*

Gradijentna metoda

U osnovi gradijentne metode sadržan je **princip pretraživanja i približavanja** pa se ova metoda može, u metodološkom smislu, označiti kao **metoda pretraživanja**. Ona inače pripada, kako je ranije naglašeno, grupi numeričkih metoda optimizacije.

Suština metode i njene procedure sastoji se u iterativnom približavanju **optimumu M_o** po **gradijentnoj trajektoriji**. Ova trajektorija je, kao što je poznato, upravna na **ekvidistantne linije** nivoa u konturnom dijagramu funkcije **F_c**. U njenim tačkama se postiže najveća promena vrednosti (najveći porast odnosno pad ili najbrži rast odnosno opadanje) funkcije optimizacije F_c.



Postepeno približavanje optimumu M_0 funkcije optimizacije F_c pomoću gradijentne metode

Sadržaj pojedinih sukcesivnih koraka u algoritmu gradijentne metode optimizacije obuhvata:

*1.**Izbor početne tačke*** $M_p=M_1(x_1)$, čije su koordinate

$$\vec{x}_1 = (x_{11}, x_{21}, x_{31}, ..., x_{k1})$$

Ukoliko su uz funkciju optimizacije ***F_c*** data određena ograničenja, tada je potrebno da se izvrši provera da li koordinate tačke M_1 **zadovoljavaju sistem datih ograničenja**.

2.Izračunavanje vrednosti F_c u tački M_1, tj.

$$F_{c1} = F_{c1}(x_{11}, x_{21}, x_{31}, ..., x_{k1})$$

Gradijentna metoda

3.Određivanje gradijenta $gradF_c = \nabla F_c = (\frac{\partial F_c}{\partial x_1}, \frac{\partial F_c}{\partial x_2}, \frac{\partial F_c}{\partial x_3}, ..., \frac{\partial F_c}{\partial x_k})$

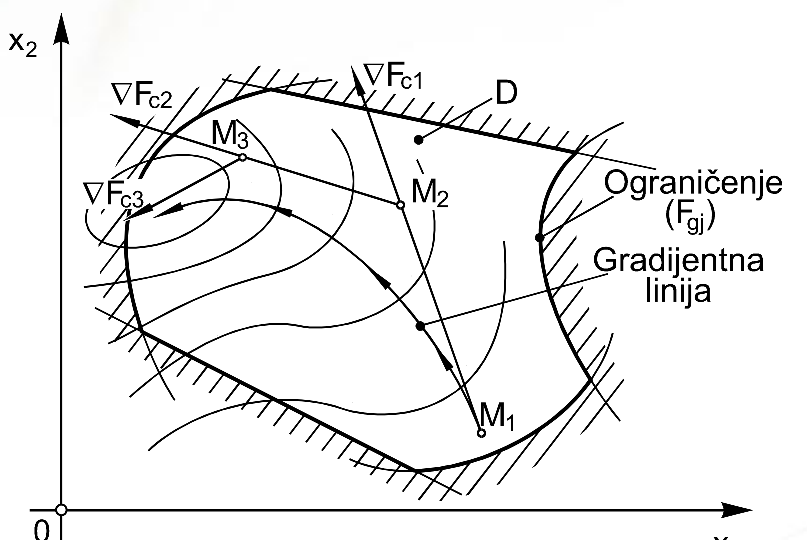
funkcije optimizacije F_c u tački M_1 $gradF_{c1} = \nabla F_{c1} = (\frac{\partial F_{c1}}{\partial x_1}, \frac{\partial F_{c1}}{\partial x_2}, \frac{\partial F_{c1}}{\partial x_3}, ..., \frac{\partial F_{c1}}{\partial x_k})$

4.Određivanje veličine koraka $\overrightarrow{\Delta x_1} = \lambda_1 gradF_{c1}$

po gradijentnoj liniji od tačke M_1=M_p do M_2. Veličina koraka zavisi od veličine promene, odnosno oblika površine F_c u okolini početne tačke M_1. Pri suviše velikom koraku može se znatno odstupiti od gradijentne linije, pa čak i **prekoračiti optimum objekta**, ali ako je korak izuzetno mali biće duža i sporija iterativna procedura približavanja optimumu.

5. Pošto je izabran korak, određuju se zatim koordinate nove tačke $M_2(x_2)$

$$\vec{x}_2 = (x_{12}, x_{22}, x_{32}, ..., x_{k2}) = \vec{x}_1 + \lambda_1 gradF_{c1}$$



Jedna praktična gradijentna procedura iterativnog približavanja optimumu objekta

6. Proverava se da li koordinate tačke M_2 zadovoljavaju sistem datih ograničenja.

7.Izračunava se vrednost funkcije optimizacije F_{c2} u tački M_2, tj.

$$F_{c2} = F_{c2}(x_{12}, x_{22}, x_{32}, ..., x_{k2})$$

8.Upoređivanje vrednosti F_{c1} i F_{c2} funkcije optimizacije u tačkama M_1 i M_2.

9.Ponavlja se opisana procedura za tačku M_1 u tački M_2.

Gradijentna metoda

10. Procedure tačaka M_1 i M_2 ponavljaju se i u svim narednim tačkama $M_{3,4,...}$ gradijentne linije sve dotle dok se ne dostigne traženi **optimum F_{co} date funkcije optimizacije F_c**.

Smatra se da je iterativna procedura gradijentnog metoda okončana, tj., neka tačka iz dopuštenog domena biće optimalna: ako modul gradijenta grad $F_c(x)$ u ovoj, optimalnoj tački **ima malu vrednost** što znači da su komponente vektora *grad$F_c(x)$* **vrlo bliske ili skoro jednake nuli**, dakle,

$$\frac{\partial F_c}{\partial x_i} \approx 0, \qquad i = \overline{1, k},$$

ili ako se ova tačka nalazi **na granici dopuštene oblasti** (u ovoj tački ne moraju tada biti komponente gradijenta gradF_c bliske nuli).

Kao što se iz izložene procedure vidi, **izbor početne tačke, izbor pravca kretanja** i **izbor koraka na pravcu kretanja** ka optimumu predstavlja tri ključna elementa gradijentnog metoda optimizacije.

Primer: Primena gradijentne metode

Odrediti minimum funkcije cilja:

$$F_c = X_1^2 + X_2^2 - 6X_1 - 4X_2 + 18$$

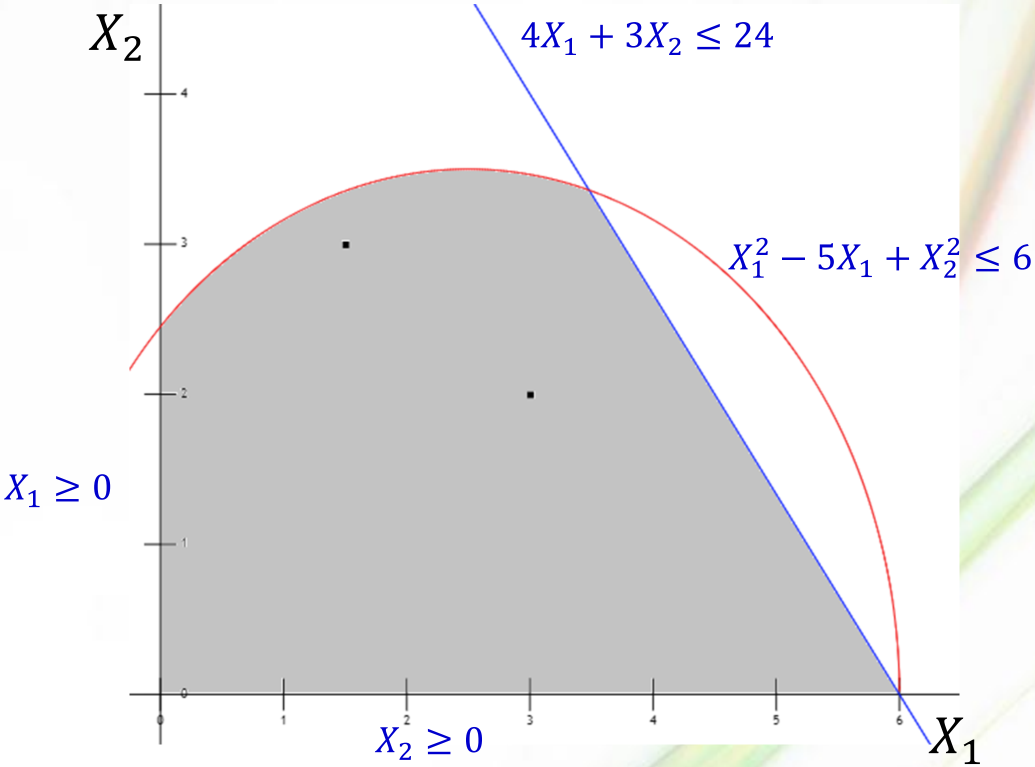
pri sledećim ograničenjima:

$$X_1^2 - 5X_1 + X_2^2 \leq 6$$

$$4X_1 + 3X_2 \leq 24$$

$$X_1 \geq 0$$

$$X_2 \geq 0$$



nule funkcija su:

$$\boldsymbol{X_1^2 - 5X_1 + X_2^2 \leq 6}$$

Nule funkcije:

$$X_1 = 0;\ X_2 = \sqrt{6} = 2{,}45$$

$$X_2 = 0;\ X_1^2 - 5X_1 \leq 6,$$

$$X_{1_1} = -1 \quad X_{1_2} = 6$$

$$\boldsymbol{4X_1 + 3X_2 \leq 24}$$

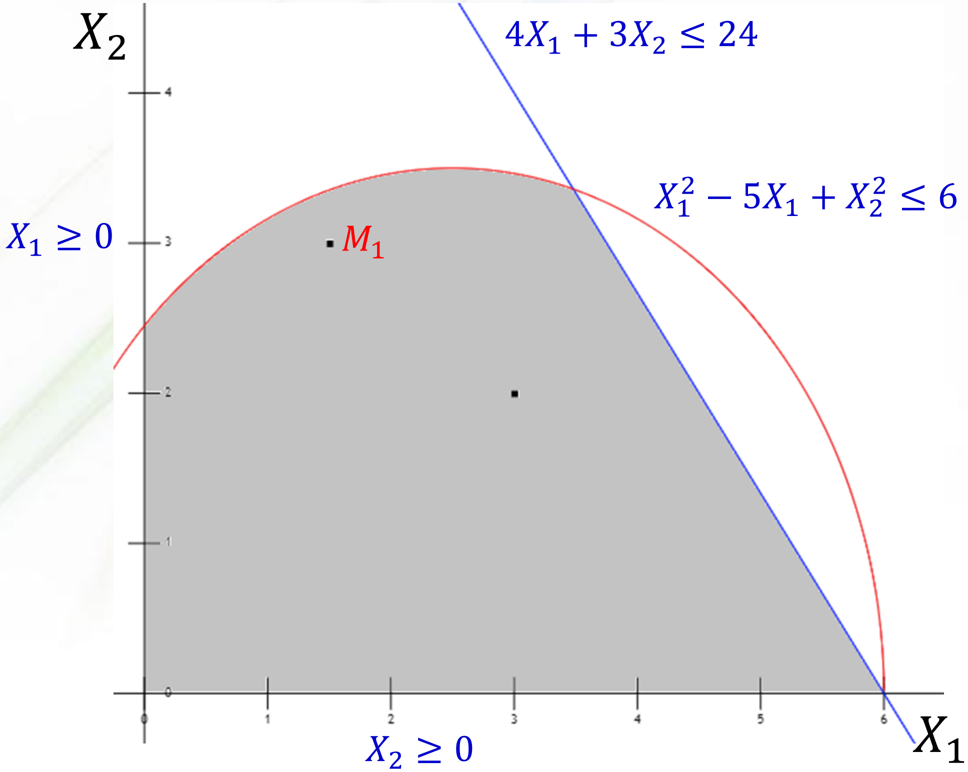
Nule funkcije:

$$X_1 = 0 \rightarrow X_2 = 8$$

$$X_2 = 0 \rightarrow X_1 = 6$$

Korak 1. Izbor početne tačke M_1

U dopuštenoj oblasti datoj na dijagramu, biramo proizvoljnu tacku $M_1(1{,}5;\ 3)$



M (X1=1,5; X2=3)

Tačka M_1 pripada dopuštenoj oblasti jer zadovoljava data ograničenja

$$(1{,}5)^2 - 5 \cdot 1{,}5 + 3^2 = 3{,}75 < 6$$

$$4 \cdot 1{,}5 + 3 \cdot 3 = 15 < 24$$

$$1{,}5 > 0$$

$$3 > 0$$

Korak 2. Izračunavanje vrednosti F_c u tački M_1

$$F_c = X_1^2 + X_2^2 - 6X_1 - 4X_2 + 18$$

$$Fc(M1) = (1{,}5)^2 + 3^2 - 6 \cdot 1{,}5 - 4 \cdot 3 + 18 = 8{,}25$$

Korak 3. Određivanje gradijenta u tački M_1

$$grad\ F_c = \nabla F_c = \left(\frac{\partial(X_1^2 + X_2^2 - 6X_1 - 4X_2 + 18)}{\partial X_1} ; \frac{\partial(X_1^2 + X_2^2 - 6X_1 - 4X_2 + 18)}{\partial X_2} \right)$$

$$grad\ F_c = (2X_1 - 6; 2X_2 - 4) = (-3; 2)$$

$$X_1 = 1{,}5 \qquad X_2 = 3$$

Ovde vidimo da u tački M_1 nije minimum jer je $\nabla F_c \neq 0$

Korak 4. Određivanje veličine koraka

$$\Delta\overrightarrow{X_1} = \lambda_1 \cdot grad\ F_{c_1}$$

$$M_2 = \overrightarrow{X_2} = \overrightarrow{X_1} - \lambda_1 \cdot grad\ F_{c_1}$$

$$M_2 = \overrightarrow{X_2} = [1{,}5 - \lambda_1(2X_1 - 6); 3 - \lambda_1(2X_2 - 4)]$$

$$M_2 = \overrightarrow{X_2} = [1{,}5 - \lambda_1(2 \cdot 1{,}5 - 6); 3 - \lambda_1(2 \cdot 3 - 4)]$$

$$M_2 = \Delta\overrightarrow{X_2} = (1{,}5 + 3\lambda_1; 3 - 2\lambda_1)$$

$$F_{c(X_2)} = F_c\left(X_1 - \lambda_1 \, grad \, F_{c_1}\right)$$

$$F_{c(X_2)} = (1{,}5 + 3\lambda_1)^2 + (3 - 2\lambda_1)^2 - 6(1{,}5 + 3\lambda_1) - 4(3 - 2\lambda_1) + 18$$

$$F_{c(X_2)} = 2{,}25 + 9\lambda_1 + 9\lambda_1^2 + 9 - 12\lambda_1 + 4\lambda_1^2 - 9 - 18\lambda_1 - 12 + 8\lambda_1 + 18$$

$$F_{c(X_2)} = 13\lambda_1^2 - 13\lambda_1 + 8,$$

Vrednost parametra λ_1 određuje se iz ekstremuma funkcije $F_{c(X_2)}$

$$\frac{\partial F_{c(X_2)}}{\partial \lambda_1} = 0 \rightarrow \frac{\partial(13\lambda_1^2 - 13\lambda_1 + 8{,}25)}{\partial \lambda_1} = 0$$

$$26\lambda_1 - 13 = 0$$

$$\lambda_1 = \frac{13}{26} = 0{,}5$$

Sada možemo odrediti koordinate tačke M_2:

$$M_2 = \overrightarrow{X_2} = (1{,}5 + 3 \cdot 0{,}5; 3 - 2 \cdot 0{,}5)$$

$$M_2 = \overrightarrow{X_2} = (3; 2)$$

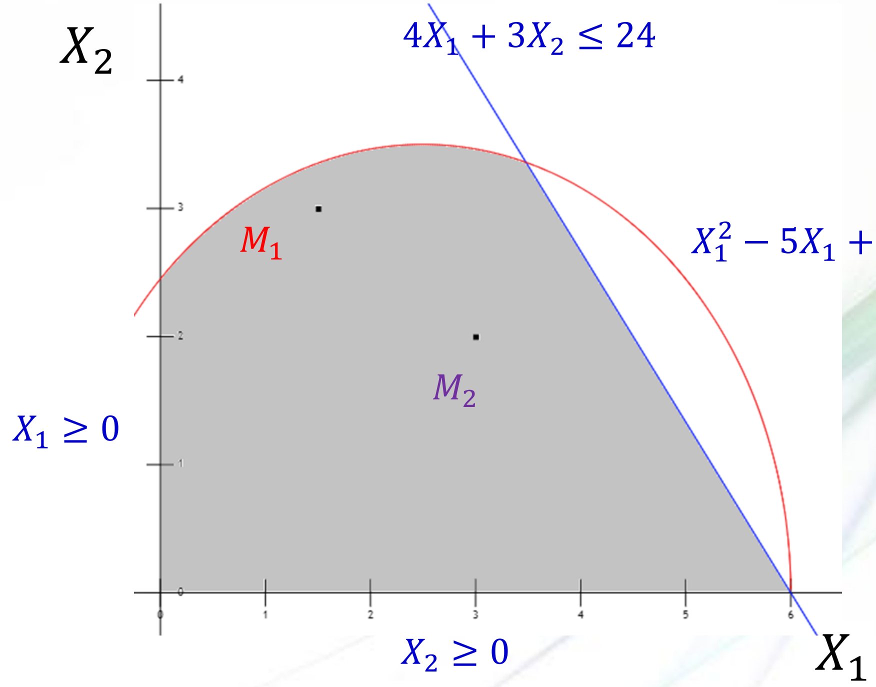
Korak 5. Provera ograničenja za tačku M_2

$$3^2 - 5 \cdot 3 + 2^2 = -2 = -2 < 6$$

$$4 \cdot 3 + 3 \cdot 2 = 18 < 24$$

$$3 > 0$$

$$2 > 0$$



Tačka M_2 pada u oblast ograničenja

Korak 6. Izračunavanje vrednosti funkcije cilja F_c u tački M_2:

$$F_c = 3^2 + 2^2 - 6 \cdot 3 - 4 \cdot 2 + 18 = 5$$

$$F_{c_2} < F_{c_1} \ \ tj \ \ 5 < 8{,}25$$

Korak 7. Određivanje gradijenta u tački M_2:

$$grad\ F_{c_2} = \left(\frac{\partial (X_1^2 + X_2^2 - 6X_1 - 4X_2 + 18)}{\partial X_1} ; \frac{\partial (X_1^2 + X_2^2 - 6X_1 - 4X_2 + 18)}{\partial X_2} \right)$$

$$grad\ F_{c_2} = (2X_1 - 6; 2X_2 - 4) = (2 \cdot 3 - 6; 2 \cdot 2 - 4) = (0; 0)$$

Dakle možemo no osnovu dobijenih rezultata zaključiti da je minimum F_c u tački M_2 jer je $\frac{\partial F_{c_2}}{\partial X_2} = 0$

Vrednost F_c u tački minimuma iznosi $\boldsymbol{F_c = 5}$ i to je najniža vrednost u dozvoljenoj radnoj oblasti.

Gradijentna metoda

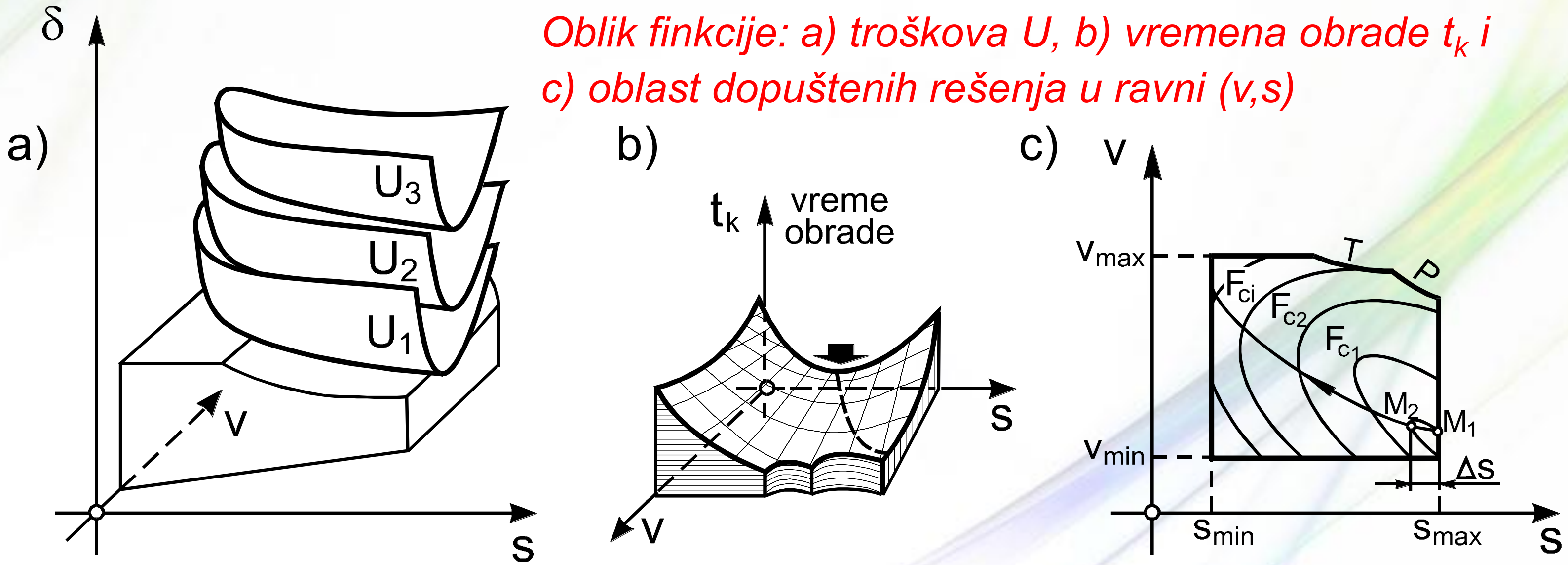
Pri unutrašnjoj optimizaciji obradnih procesa, odnosno određivanju optimalnih **režima obrade**, na bazi **troškova i vremena obrade** kao funkcija optimizacije, pogodno je primeniti gradijentni iterativni metod.

Ove funkcije optimizacije mogu se prikazati izrazom opšteg oblika:

$$F_c = F_c(v, s, \delta, c_1, c_2, \ldots, c_i, \ldots, c_k)$$

Ulazne veličine obuhvataju grupu promenljivih i konstantnih veličina. Prvu grupu čine **brzina rezanja *v***, **pomak *s*** i **dubina rezanja *δ*,** a drugu veličine $c_1, c_2, \ldots, c_i, \ldots, c_k$.

Oblik finkcije: a) troškova U, b) vremena obrade t_k i c) oblast dopuštenih rešenja u ravni (v,s)



Određivanje optimalnih režima obrade na bazi funkcije optimizacije, vrši se tako što se gradijentni metod primenjuje u svim ravnima *(v,s)* omeđenim parametrima i funkcijama ograničenja obradnog procesa, a broj ovih ravni, odnosno iteracija, određuje broj tehnoloških vrednosti dubina rezanja.

Prema tome, primena gradijentnog iterativnog metoda za rešavanje ovog optimizacionog zadatka zahteva sledeđu proceduru:

1. *Definisanje skupa tehnoloških vrednosti dubina rezanja:*

$$\delta = \{\delta_1, \delta_2, \ldots, \delta_i, \ldots, \delta_p\}$$

2. *Definiše se skup ograničenja za pomak, pri δ_i=const*

$$s_{\min} \leq s \leq s_{\max}$$

3. *Određivanje optimalne brzine rezanja v_{o1} u tački M1 na osnovu formule, slika 2c u kojoj je $s_1 = s_{max}$, pri posmatranoj dubini rezanja, δ_i=const, odnosno:*

$$\left(\frac{\partial Fc}{\partial v}\right)_{\substack{s=s_1 \\ \delta_i=const}} = 0$$

4. *Provera ograničenja za brzinu rezanja i složenih ograničenja pri δ_i=const, s=s1:*

$$v_{\min} \leq v_{o1} \leq v_{\max}$$

$$F_{gj} \leq 0$$

5. *Izračunavanje vrednosti funkcije u tački M1, gde su: $v=v_{o1}$, $s_1=s_{max}$, δ_i=const.*

$$F_c = F_c(v, s, \delta)$$

6. *Usvajanje prvog manjeg pomaka:*

$$s_2 = s_1 - \Delta s$$

7. *Procedura se nastavlja (od tačke 3-6) sve dok se ne zadovolji uslov dobijanja optimalne vrednosti funkcije cilja (min vreme, min troškovi, itd.)*

$$F_{c_{i-1}} < F_{c_i} < F_{c_{i+1}}$$

Iterativnim ponavljanjem izložene procedure za sve tehnološke vrednosti dubina rezanja odrediće se najmanja vrednost funkcije, a samim tim i optimalna tačka režima obrade posmatranog obradnog procesa, kao objekta optimizacije.

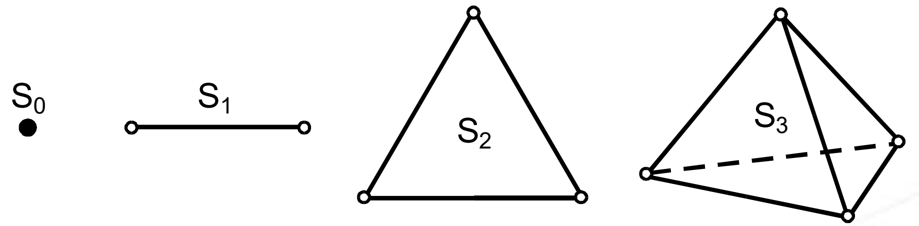
Simpleksna metoda

Slično gradijentnoj metodi, i simpleksna metoda se odlikuje:

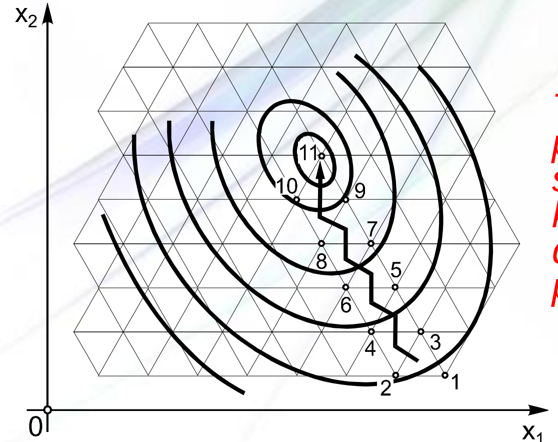
- *Univerzalnošću primene, bez obzira na oblik modela optimizacije,*
- *Jednostavnošću iterativne procedure,*
- *Mogućnošću rešavanja vrlo složenih optimizacionih zadataka,*
- *Orijentisanošću simpleksne procedure na računare, itd.*

Neki iterativni koraci u simpleksnoj proceduri nešto su lakši, jednostavniji u operativnom smislu (u odnosu na gradijentnu proceduru), jer je u simpleksnom metodu **isključena potreba za izračunavanjem parcijalnih izvoda** date funkcije optimizacije F_c, što je veoma značajno naročito kada je model optimizacije odnosno topologija funkcije optimizacije relativno složen.

Simpleksna metoda se, s obzirom na širinu primene, vrlo uspešno koristi u obe grupe optimizacionih metodologija, tj. i kao **metod eksperimentalne** (adaptivne) optimizacije i kao **analitički metod optimizacije**. Razlika je samo u tome što se vrednosti funkcije optimizacije F_c određuju u jednoj metodologiji merenjem, dakle, eksperimentalnim putem na objektu optimizacije, a u drugoj izračunavanjem iz matematičkog izraza funkcije optimizacije.



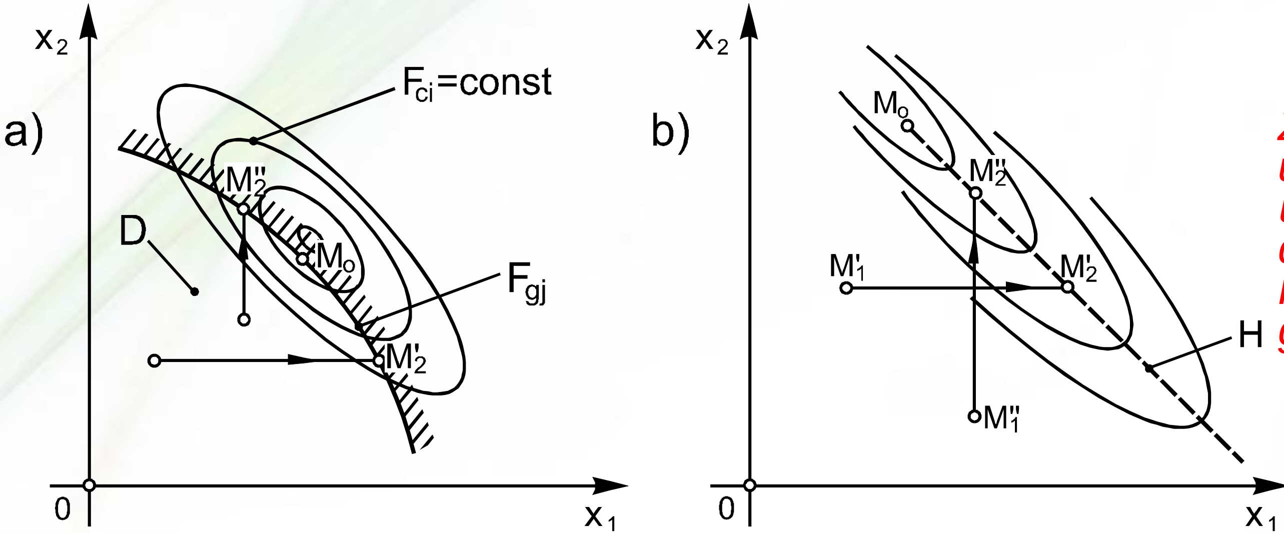
Oblici pravilnih simpleksa



Trajektorija pomeranja simpleks-planova ka optimumu dvofaktornog procesa

Gaus-Zajdelova metoda

U **Gaus-Zajdelovoj metodi** ili, kako se još naziva, **metoda uzastopnih promena** ulaznih veličina menja se, polazeći od neke polazne tačke M_p=M_1, sukcesivno, jedna za drugom, svaka nezavisno promenljiva veličina, sve dotle dok se u pravcu te promenljive veličine ne dostigne **ekstrem** date funkcije optimizacije, tj. do tačke M_2.



Zaustavljanje procedure uzastopnih promena ulaznih veličina (za slučaj dvodimenzionalne funkcije F_c) na granici F_{gj} (a) i grebenu H funkcije F_c (b)

Zatim se, počevši od tačke M_2, menja, na isti način, naredna promenljiva veličina, a posle nje i ostale, sve do tačke optimuma funkcije F_c. **Pri promeni jedne, ostale promenljive veličine zadržavaju stalnu vrednost**. Redosled uzastopnih promena nezavisnih veličina potpuno je proizvoljan.

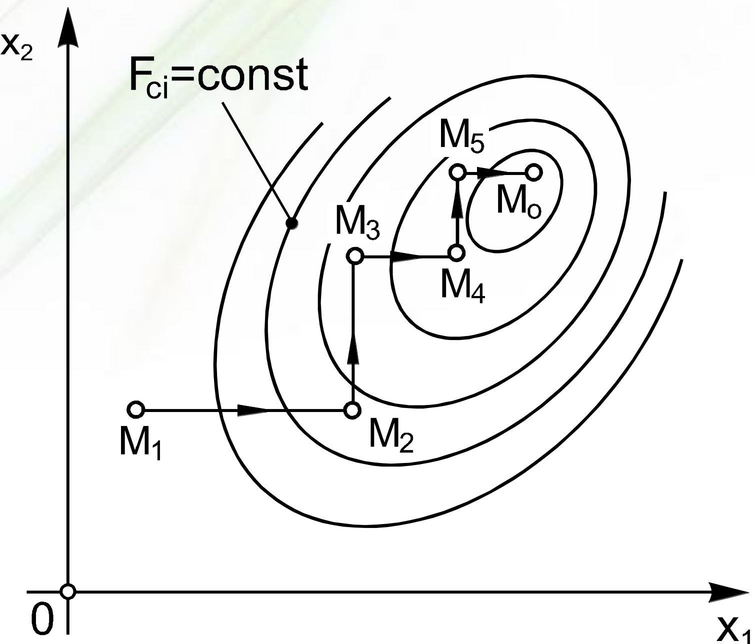
Tačka ekstrema funkcije optimizacije F_c pri promeni neke veličine x_i određuje se relativno lako pomoću poznatih postupaka iz matematičke analize.

Metoda se odlikuje krajnjom jednostavnošću iterativne procedure i odsustvom računanja parcijalnih izvoda. Mana metode je, međutim, u njenoj **relativno dugoj proceduri** i u značajnim teškoćama određivanja optimuma funkcije F_c pri nailasku na ograničenja F_{gj}, ili na oštri greben H funkcije F_c, a takođe i u otkrivanju nekog od **lokalnih ekstrema**, zavisno od položaja početne tačke, umesto globalnog ekstrema.

Metoda relaksacije

Za razliku od Gaus-Zajdelove metode, u **metodi relaksacije** procedura kretanja ka optimumu započinje iz neke početne tačke M_1, ne u pravcu proizvoljno uzete ose već u **pravcu one ose za koju je promena** (porast, opadanje) date funkcije optimizacije $\boldsymbol{F_c}$ **najveća**.

Taj osni pravac ili pravac kretanja određuje se tako što se u početnoj tački izračunavaju vrednosti **parcijalnih izvoda** funkcije F_c po svim nezavisno promenljivim veličinama.



Dalji tok procedure odvija se tako što se u narednim koracima (u tačkama $M_2, M_3, M_4, \ldots, M_o$) **ponavljaju operacije izvedene u prvom koraku** (tačka M_1). Procedura se smatra završenom (što znači da je određena tačka optimuma M_o) ako, pri kretanju iz tačke M_o po bilo kom osnom pravcu, **ne nastupa bitna promena vrednosti funkcije optimizacije** $\boldsymbol{F_c}$. Ovaj kriterijum se praktično izražava uslovom:

$$\sum_{i=1}^{k} \left(\frac{\partial F_c}{\partial x_i} \right)^2 < \delta$$

Kada $\delta \to 0$ tada su parcijalni izvodi u tački nagomilavanja, odnosno tački optimuma, kojoj inače konvergira procedura ovog metoda, jednaki nuli što predstavlja poznati potrebni uslov za **ekstrem funkcije optimizacije** $\boldsymbol{F_c}$.

Relaksaciona metoda, kao i Gaus-Zajdelova, vrlo je jednostavna, **lako se programira i automatizuje njena procedura**. Pri tome je njena procedura nešto kraća. Ima iste mane kao i Gaus-Zajdelova metoda.

Metoda skeniranja

Ova metoda naziva se još i **metoda potpunog pretraživanja**, a karakteriše se pretraživanjem vrednosti funkcije optimizacije F_c u tačkama **dopuštene oblasti**, u kojoj ili na čijoj se granici nalazi optimum funkcije F_c.

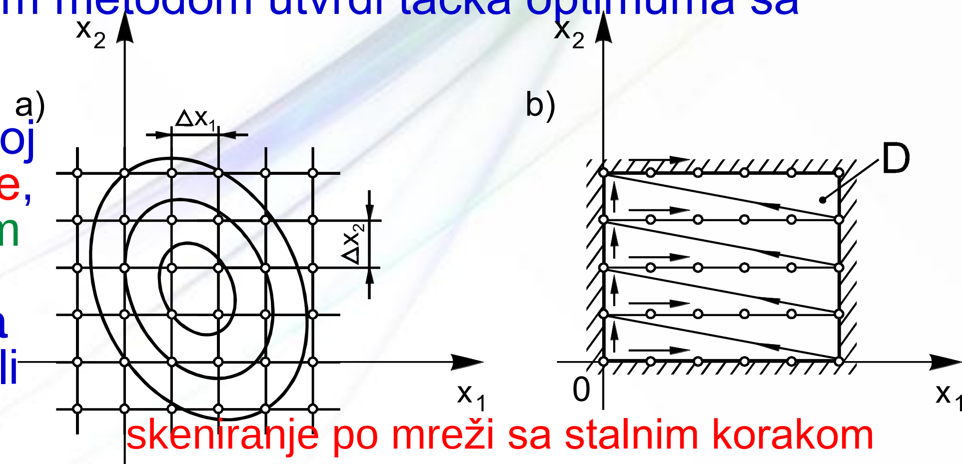
Što je **gustina tačaka**, u kojima se ispituje vrednost funkcije F_c u dopuštenoj oblasti D, **veća**, odnosno što je **korak skeniranja manji**, biće **viša tačnost pretraživanja**, veća sigurnost da se u skupu lokalnih, bezuslovnih ili uslovnih, otkrije globalni ekstrem i obratno.

Ali pri ovome treba imati u vidu to da **veliki broj tačaka**, odnosno obimni skup računskih operacija, koji raste sa smanjenjem koraka skeniranja i povećanjem broja ulaznih veličina, **umanjuje** se **praktični značaj metoda**, naročito kada je broj ulaznih varijabli relativno velik.

Zato se ova metoda praktično vrlo uspešno koristi za identifikovanje optimuma funkcije F_c, bez obzira na njen oblik i tipove funkcija ograničenja, samo u onim slučajevima kada je **relativno mala dimenzionalnost objekta** odnosno modela optimizacije, tj. kada broj ulaznih veličina nije veći od dve-tri.

Metoda se može **kombinovati sa nekom drugom metodom** i to tako što se pomoću nje približno identifikuje uža oblast globalnog ekstrema, odnosno optimuma, a zatim se drugom, na primer, gradijentnom metodom utvrdi tačka optimuma sa željenom tačnošću.

Metode skeniranja se **dele prema vrsti plana** pretraživanja optimuma u dopuštenoj oblasti. Kako plan može biti u obliku mreže, spirale, sa stalnim ili promenljivim korakom Δx_i itd., to se ove metode dele na **metode skeniranja po mreži**, **metode skeniranja po spirali**, metode skeniranja sa stalnim ili promenljivim korakom, itd.



skeniranje po mreži sa stalnim korakom

Metoda skeniranja

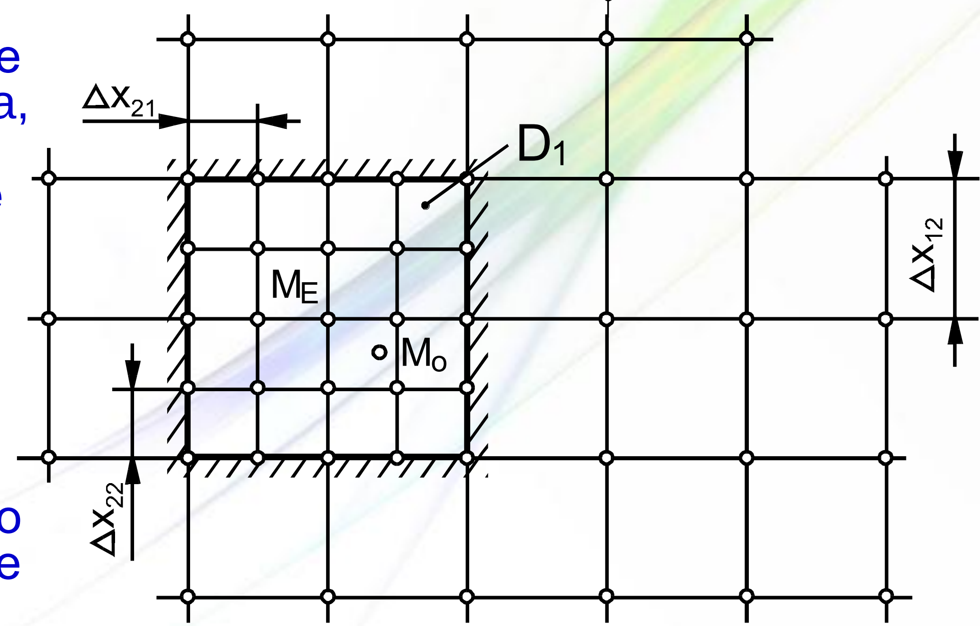
Metoda skeniranja po spirali je pogodna samo za slučaj dvodimenzionalnih funkcija optimizacije $F_c=F_c(x_1,x_2)$.

Metodologija skeniranja sa promenljivim korakom odvija se najčešće tako što se, u prvoj fazi, identifikuje **uža oblast *D***, oko optimuma primenom procedure sa **konstantnim** i **relativno velikim korakom Δx_{1i}**.

Zatim se, u drugoj fazi, pretražuje i određuje optimum sa potrebnom tačnošću korišćenjem **manjeg koraka** $\Delta x_{2i}<\Delta x_{1i}$, dakle, gušćeg rasporeda tačaka u lokalizovanoj oblasti D_1, oko optimuma M_o.

U tački M_E funkcija F_c imala je ekstremnu vrednost, najveću ili najmanju u odnosu na ostale tačke, utvrđenu u prvoj fazi.

Pored jednostavnosti procedure i određene sigurnosti identifikacije globalnog ekstrema, metoda skeniranja poseduje još jedno praktično značajno svojstvo: **isključena je potreba za izračunavanjem parcijalnih izvoda**. Uz to **sistem ograničenja ne otežava proceduru**, jer se plan tačaka pretraživanja smešta u dopuštenu oblast, pogotovo ako su funkcije ograničenja date u obliku nejednačina. Ako je, međutim, neka iz sistema funkcija ograničenja, ili ceo sistem ograničenja, data u obliku jednačine



metoda skeniranja sa promenljivim korakom

Dinamičko programiranje

Zadaci i objekti optimizacije, koji se rešavaju metodama linearnog i nelinearnog programiranja, a koji su izloženi u prethodnim tačkama, smatraju se jednoetapnim ili statičkim zadacima, jer **ne zavise od vremena**, pa se procedura određivanja optimuma, odnosno upravljanja objektom, na primer, nekim procesom, proteže na **jednu etapu**.

Ako, međutim, objekat optimizacije, na primer neki proces, **zavisi od vremena**, tj. ako se njegova optimizacija ili njegovo optimalno upravljanje izvodi **u više sukcesivnih etapa**, u više vremenskih perioda, čime se postiže optimizacija ili optimalno upravljanje procesa u celini, tada je reč o višeetapnim zadacima, odnosno procesima, koji se rešavaju **metodama dinamičkog programiranja**.

Ali metodologija dinamičkog programiranja sadrži **jednu manu**: za razliku od mnogih prethodnih metoda, u kojima su definisani i razvijeni relativno strogi i univerzalni algoritmi rešavanja optimizacionih zadataka, u metodama dinamičkog programiranja **nedostaje ova univerzalnost algoritma**, pa se pojedine grupe optimizacionih zadataka rešavaju na osnovu **posebnih**, za te grupe **razvijenih algoritama**. Inače, složeni problemi višefaktornosti rešavaju se primenom računara.

U tehnici i proizvodnoj tehnologiji postoje brojni objekti optimizacije koji se rešavaju metodama dinamičkog programiranja. Među ove objekte, odnosno optimizacione zadatke, spadaju sledeći osnovni zadaci:

- *Optimizacija proizvodnih tehnologija, odnosno optimizacija procesa obrade delova na obradnim sistemima sa stanovišta minimalnog vremena obrade i niz drugih,*
- *Optimalno planiranje proizvodnih programa,*
- *Optimalno projektovanje i optimizacija konstrukcija, mehanizama, jedinica i sistema proizvodne tehnike,*
- *Optimalna zamena i modernizacija obradnih i tehnoloških sistema, sistema upravljanja i dr.,*
- *Analiza pouzdanosti elemenata i sistema proizvodne tehnike,*
- *Optimalna raspodela resursa,*
- *Optimalno korišćenje obradnih i tehnoloških sistema i dr.*

Dinamičko programiranje

Primer iz ove grupe zadataka: neka je za obradu jedne serije delova potrebno, u i-tom mesecu, m_i obradnih sistema. Ako se u sledećem (i+1)-*om* mesecu menja obim rada, za što je potrebno m_{i+1} obradnih sistema za obradu date serije delova, potrebno je da se, pri zadatom obimu serije, odredi **optimalni broj obradnih sistema**, koji će se koristiti u svakom mesecu, kako bi se postigli **minimalni troškovi obrade** date serije:

- ***Rešavanje transportnih problema**, kao što su proizvodne linije i uopšte u transportu, na primer, određivanje najkraćeg puta na mreži i dr.,*
- ***Optimalno upravljanje zalihama**,*
- ***Rešavanje zadataka mrežnog planiranja** i niz drugih.*

Metodologija dinamičkog programiranja odlikuje se **raščlanjivanjem ili dekompozicijom nekog složenog optimizacionog zadatka**, tj. zadatka koji sadrži, uz ostalo, **veći broj promenljivih**, a koji se inače rešava ovim metodom, na **niz uzastopnih etapa**, pri čemu sada, u pojedinim etapama, figuruše **relativno manji broj varijabli**. Time se olakšava rešavanje datog optimizacionog zadatka. To je prvi, opšti korak algoritma ove metodologije.

Drugi korak se odnosi na primenu **Belmanovog principa optimalnosti** na optimizaciju ovakvih višeetapnih procesa. Prema ovom principu optimalno rešenje se karakteriše time da, bez obzira na to kakvo je neko dato rešenje, naredno rešenje mora biti optimalno u odnosu na to dato rešenje. To važi za sve etape procesa, a to znači da sva naredna rešenja moraju biti optimalna u odnosu na početno, bez obzira na to kakvo je to početno rešenje.

Dinamičko programiranje

Metod dinamičkog programiranja se koristi pri rešavanju optimizacionih zadataka šije su funkcije optimizacije **separabilnog oblika**.

Za **separabilnu funkciju optimizacije** i funkcije ograničenja karakteristično je da se izražavaju oblicima:

$$F_c = F_1(x_1) + F_2(x_2) + F_3(x_3) + \ldots + F_n(x_n)$$

$$F_{gi} = F_{gi1}(x_1) + F_{gi2}(x_2) + F_{gi3}(x_3) + \ldots + F_{gin}(x_n) \le 0$$

To dopušta da se procedure optimizacije mnogih objekata tretiraju kao višeetapni procesi, pa, imajući u vidu ovo, mnogi se optimizacioni modeli sa separabilnim funkcijama optimizacije mogu **svesti na modele sa jednim ograničenjem**:

$$F_c = g_1(x_1) + g_2(x_2) + g_3(x_3) + \ldots + g_n(x_n) = \sum_{i=1}^{n} g_j(x_j)$$

Gde su $g_j(x_j) = c_j x_j$ - **funkcije pojedinih dobiti** koje mogu imati linearni ili nelinearni oblik.

Sistemom funkcija koje su definisane metodom dinamičkog programiranja definiše se **algoritam određivanja optimalnog rešenja** u datom optimizacionom zadatku.

Metodom dinamičkog programiranja se vrlo uspešno rešavaju zadaci optimizacije **diskretnih, višeetapnih procesa** čiji su kriterijumi optimalnosti izraženi u obliku aditivnih, odnodno separabilnih funkcija, sastavljenih od parcijalnih funkcija optimizacije pojedinih etapa.