Taikomieji optimizavimo metodai

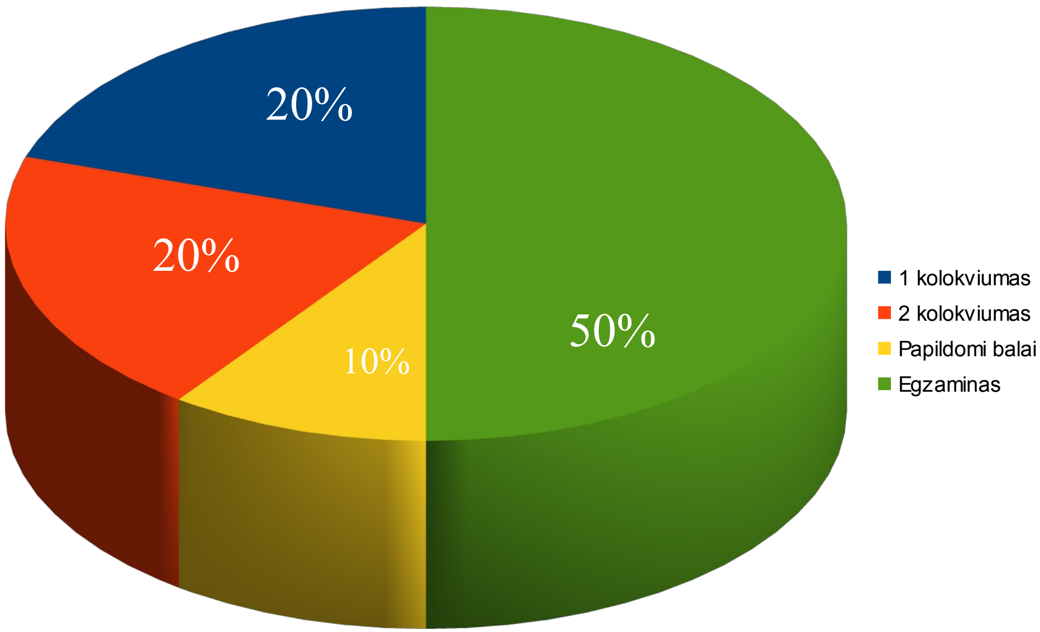
LITERATŪRA

- A. Apynis. Optimizavimo metodai. V., 2005
- G. Dzemyda, V. Šaltenis, V. Tiešis. Optimizavimo metodai, V., 2007
- V. Būda, M. Sapagovas. Skaitiniai metodai : algoritmai, uždaviniai, projektai, V., 1998.
- R. Čiegis, V. Būda, Skaičiuojamoji matematika, V.,1997
- V. Čiočys, R. Jasilionis. Matematinis programavimas. V., 1990

Pagrindinės tėmos

- Optimizavimo uždavinio matematinis formulavimas
- Vieno kintamojo funkcijų optimizavimas
- Daugelio kintamųjų funkcijų optimizavimas be apribojimų
- Daugelio kintamųjų funkcijų optimizavimas esant apribojimams
- Globalusis optimizavimas
- Tiesinio programavimo uždaviniai
- Daugiakriteriniai optimizavimo uždaviniai

Vertinimo tvarka



Papildomi balai

Praktinis uždavinys

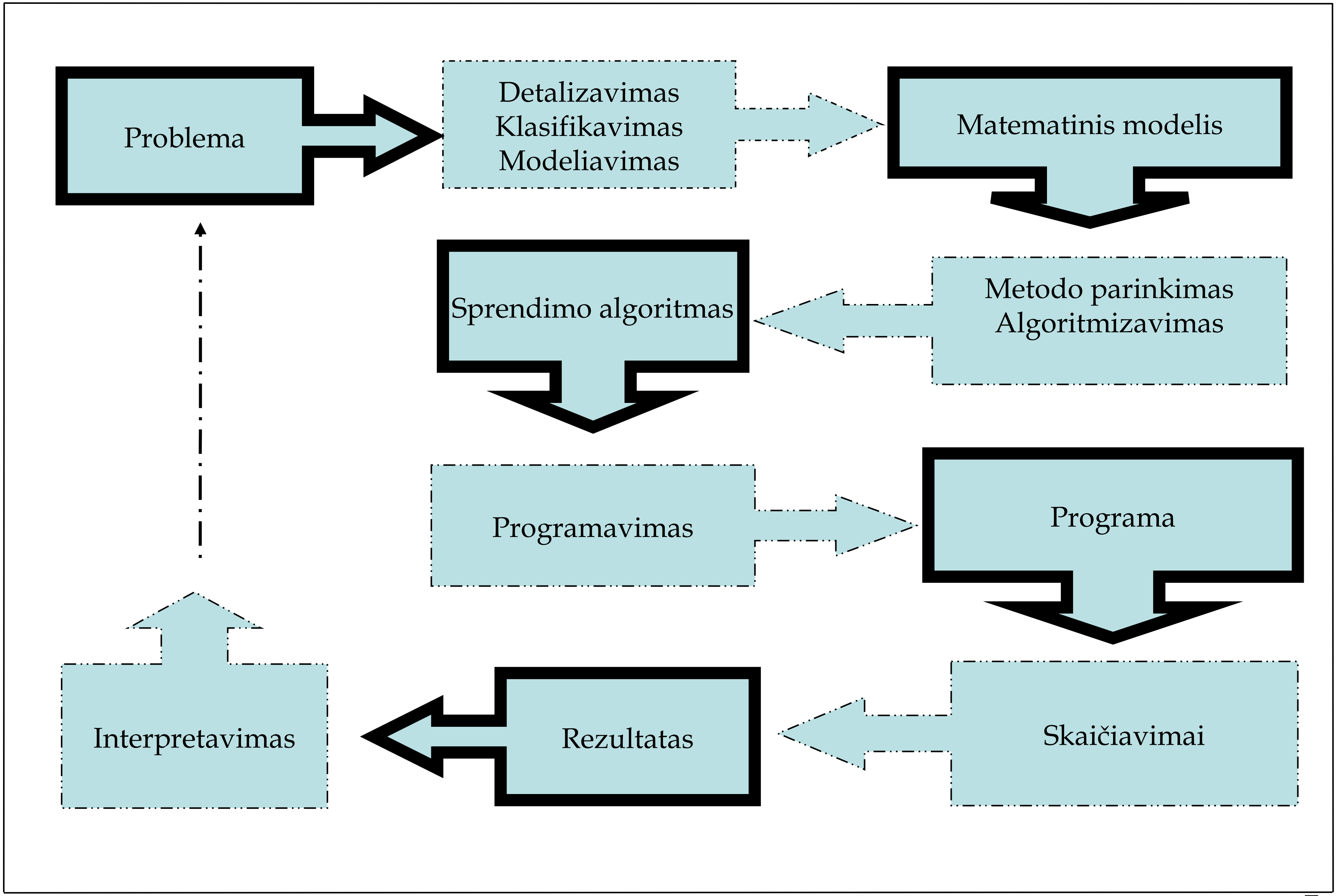
Matematinis uždavinys

Ką ir kiek gaminti, kad maksimizuoti pelną? → Tiesinio programavimo gamybos planavimo uždavinys

Kapitalo investavimas → Stochastinis modeliavimas ir optimizavimas (Vertybinių popierių kainų svyravimų aprašymas)

Projekto darbų tvarkaraščio sudarymas → Tinklinio planavimo uždavinys

Problemos sprendimo schema



Matematinio modelio sudarymo schema

- Įvedami kintamieji (nežinomieji)
- Užrašomi apribojimai kintamiesiems
- Užrašoma tikslo funkcija ir nurodoma *min* ar *max* jos reikšmė yra ieškoma
- Užrašomas uždavinio matematinis modelis

Problemos pavyzdys

	Swedbank	SEB	ŠB	
Antanas	5%	2%	5%	**50**
Leonas	25%	15%	40%	**60**
Petras	15%	30%	10%	**10**
	100	**45**	**60**	

Problemos matematinis modelis

(Tiesinio programavimo uždavinys)

$$\min(0{,}05a_1 + 0{,}02a_2 + 0{,}05a_3 + 0{,}25l_1 + 0{,}15l_2 + 0{,}4l_3 + 0{,}15p_1 + 0{,}3p_2 + 0{,}1p_3)$$

$$\begin{cases} a_1 + a_2 + a_3 = 50 \\ l_1 + l_2 + l_3 = 60 \\ p_1 + p_2 + p_3 = 10 \\ a_1 + l_1 + p_1 \le 100 \\ a_2 + l_2 + p_2 \le 45 \\ a_3 + l_3 + p_3 \le 60 \end{cases}$$

$$a_1 \ge 0, a_2 \ge 0, a_3 \ge 0, l_1 \ge 0, l_2 \ge 0, l_3 \ge 0, p_1 \ge 0, p_2 \ge 0, p_3 \ge 0.$$

Optimizavimo uždavinio matematinis formulavimas

- **Matematiškai formuluojant** optimizavimo uždavinį turi būti duoti bent du dalykai — optimizuojamų elementų aibė ir tų elementų gerumo matas.

- Matematinis uždavinio modelis būtų: $\underset{X \in D}{opt}\, f(X)$

- $f(X)$: $D \to Y$ yra gerumo matas, optimizavimo teorijoje vadinamas tikslo funkcija.
- D yra optimizuojamų objektų aibė, teorijoje vadinama leistinąja sritimi.
- *opt* yra geriausio elemento vertinimo būdas (pvz., pagal *min* ar *max* f(X) reikšmę).
- Y yra funkcijos reikšmių aibė.

- Funkcijos reikšmių aibė turi būti bent iš dalies sutvarkyta: jos elementai turi būti palyginami. Jei kuriems nirs elementams palyginimo santykis nenustatytas, jie laikomi ekvivalentiškais.

Optimizavimo uždavinio matematinis formulavimas

- Funkcijos reikšmių aibė turi būti bent iš dalies sutvarkyta – jos elementai turi būti palyginami. T.y. Aibėje Y turi būti apibrėžtas dviejų aibės elementų palyginimo santykis. Jei kuriems nors elementams palyginimo santykis nenustatytas, jie laikomi ekvivalentiškais.

- Atvaizdavimas $f(X)$: $D \rightarrow Y$ aibėje D sukuria atitinkamą palyginimo santykį kriterijaus $f(X)$ atžvilgiu. Be šio santykio negalėtume išrinkti geriausio elemento.

y_1

- Taigi formuluojant optimizavimo uždavinį šis reikalavimas yra vianintelis būtinas formalus reikalavimas. Toliau konkretinant aibes D bei Y ir funkciją $f(X)$ gaunamos įvairios optimizavimo uždavinių rūšys.

Funkcijos optimumas

- Funkcijos optimumu vadinamas toks X_{opt}, kuriam neegzistuoja X toks, kad $f(X)$ geresnis už $f(X_{opt})$. T.y. Aibėje D geresnių elementų nėra.

- Minimumo paieškos (arba minimizavimo) uždavinys yra ekvivalentus maksimizavimo uždaviniui su ta pačia tikslo funkcija, tik padauginta iš -1.

Aplinka ir atstumas

- Atstumas yra funkcija d, tenkinanti sąlygas:
- $d(U,V) \geq 0$;
- $d(U,V) = 0$ tada ir tik tada, kai $U = V$; Atstumas teigiamas tik tarp skirtingų taškų.
- $d(U,V) = d(V,U)$; Atstumas yra simetrinis;
- $d(U,V) \leq d(U,W) + d(W,V)$; Atstumas tenkina trikampio taisyklę.

$$d_1(X,Z) = \sum_{i=1}^{n} |x_i - z_i|$$

$$d_2(X,Z) = \sqrt{\sum_{i=1}^{n} (x_i - z_i)^2}$$

$$d_\infty(X,Z) = \max_{1 \leqslant i \leqslant n} |x_i - z_i|$$

Aplinka ir atstumas

- Aibė, kurioje apibrėžtas atstūmas, vadinama metrine erdve.

- Metrinės erdvės elementai vadinami taškais.

- Jei aibė D būtų nesutvarkyta, ir jei neturėtume jokių žinių apie tikslo funkciją, tada praktiškai nebūtų įmanoma sukurti racionalų optimumo paieškos algoritmą. Pvz., apskaičiavus funkcijos reikšmę viename taške nebūtų galima pasakyti apie kitas reikšmes. Liktų tik perrinkti visus D elementus – skaičiuoti jų tikslo funkcijos reikšmes ir išrinkti geriausią. Jei elementų daug ar net begalybė – užduotis beviltiška.

- Todėl nagrinėjams tam tikros leistinųjų sričių ir tikslo funkcijų klasės. Praktikoje konkretų matematinį modelį padeda suformuluoti žinios apie uždavinį.

Aplinka ir atstumas

- Optimizavimo algoritmuose svarbi lokaliojo optimumo, t.y. optimumo tam tikroje aplinkoje sąvoka.

- Taško U ε-aplinka vadinsime aibę taškų, nenutolusių nuo *U* daugiau nei per ε:

$$N_\varepsilon(U) = \{X\colon d(U,X) \le \varepsilon\}$$

- Funkcijos lokuliuoju optimumu vadinamas taškas *U*, kuris leistinojoje srityje turi aplinką tik ne iš geresnių taškų.

- Optimumas visoje leistinojoje srityje vadinamas globaluoju optimumu.

- Daugelį uždavinių galima išspręsti efektyviais lokaliojo optimumo paieškos algoritmais. Tačiau jei funkcija turi daug lokaliųjų optimumų, tai paprastai sunku rasti globalųjį.

Optimumo egzistavimas. Ekstremalių reikšmių teorema

- Ieškant optimumo gera žinuoti, kad jis egzistuoja.

- Aibės *D* ribiniu tašku vadinamas taškas *X*, jei bet kurioje norimai mažoje jo aplinkoje atsiras tašįkas $Z \neq X$. T.y. Taškas *X* nėra sankaupos taškas.

- Metrinės erdvės poaibis *D* vadinamas uždaru, jei kiekvienas jos ribinis taškas X priklauso *D*.

- Metrinės erdvės *Q* poaibis *D* vadinamas aprėžtu, jei egzistuoja metrinės erdvės taškas *Z* ir teigiamas skaliaras *r*, tokie, kad bet kuriam *X* iš D galioja $d(X,Z) \leq r$. T.y. jei *D* telpa į kokį nors hiperskritulį.

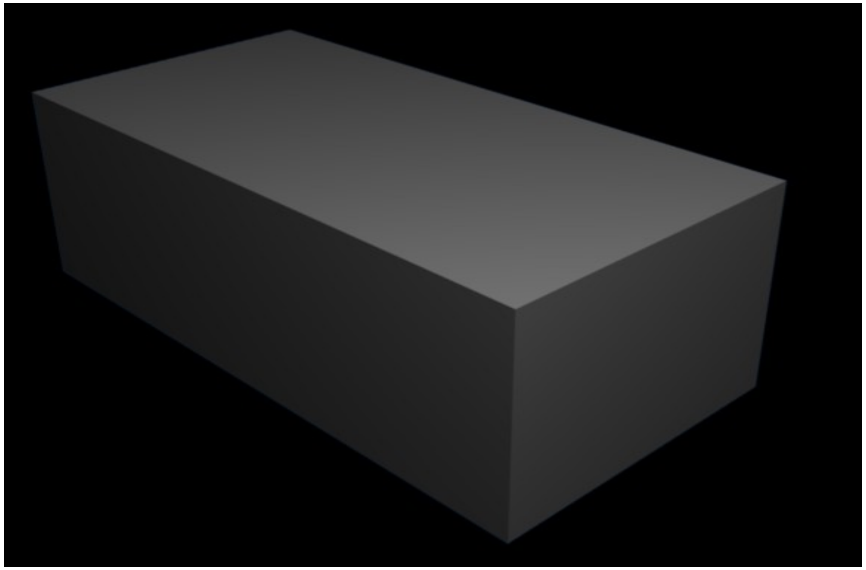
- Metrinės erdvės poaibis vadinamas kompaktu, jei jis yra uždaras ir aprėžtas.

Vėjerštraso ekstremalių reikšmių teorema

- Reali tolydi funkcija, apibrėžta kompakte, tame kompakte įgyja minimalią ir maksimalią reikšmes.

Uždavinių formulavimo pavyzdžiai

- Stačiakampio gretasienio tūris.
- Stačiakampio gretasienio sienų plotų suma lygi 1. Kokia turėtų būti stačiakampio gretasionio formos dėžė, kad jos tūris būtų maksimalus?



Stačiakampio gretasienio tūris

- Tūris:

$$f\left(X\right) = x_1\ x_2\ x_3\ .$$

- Apribojimai:

$$D = \left\{ X \in R^3 : x_i \geqslant 0,\ \ i = 1,2,3;\ \ 2x_1x_2 + 2x_1x_3 + 2x_2x_3 = 1 \right\}.$$

Netiesinė regresija

- Aproksimavimo uždavinys: eksperimentų rezultatai aproksimuojami funkcija su nežinomais parametrais.
- Parametrų reikšmės gaunamos minimizuojant skirtumus tarp eksperimento rezultatų ir funkcijos reikšmių.

- Teoriniam idealių dujų atvejui galioja lygtis $PV = RT$. Kur P – slėgis, V – molinis tūris, T – temperatūra, R – universali dujų konstanta, lygi 82,06.

- Tikrovėje yra kitaip.
- Buvo pasiūlyta keletas neidealių dujų dėsnių, vienas iš jų – Redliko-Kvongo lygtis

$$P = \frac{RT}{V - b} - \frac{a}{\sqrt{T}\,V\left(V + b\right)}.$$

Netiesinė regresija

- Lygtyje yra du nežinomi patrametrai – *a* ir *b*. P_i, V_i, ir T_i yra žinomi iš eksperimentų rezultatų.
- Parametrai *a* ir *b* randami minimizuojant eksperimentų rezultatų ir teorinio *P* skirtumų kvadratų sumą *y*:

$$\min_{a,b} = \min_{a,b}\left\{\sum_{i=1}^{n}\left[P_i - \left(\frac{RT_i}{V_i - b} - \frac{a}{\sqrt{T_i}V_i(V_i + b)}\right)\right]^2\right\}.$$

Gaisrininkų paskyrimas

- Miško gaisras plinta 2 km ruožu 32 m/min greičiu. Jį galima sustabdyti pastačius užtvarą.
- Vienas darbininkas per minutę gali pastatyti 2 m užtvaros.
- Darbininko atvežimas į darbo vietą ir atgal kainuoja 20 eu.
- Darbo užmokėstis – 6 eu/val .
- Miško 1 km^2 kaina – 20000 eu.

- Kiek reikia darbininkų, kad nuostoliai būtų minimalūs?

Gaisrininkų paskyrimas

- Minimizuosime nuostolius N nuo gesinimo pradžios: $N = M + D$.
- M – sudegusio miško kaina; D – išmokos darbininkams.

- Miško degimo trukmė yra $T = t + l$.
- l – darbininkų pristatymo į gesinimo vietą laikas, t – gaisro gesinimo trukmė.

- Per laiką T sudegs $64000T$ m^2 miško. Nuostoliai sudarys:

$$M = 64000\left(t+l\right)\frac{20000}{1000000} = 1280\left(t+l\right).$$

- Tarkime, gaisra gesins x darbininkų. Vieno darbininko darbo kaina yra

$$\left(20 + 6\frac{1}{60}t\right).$$

Gaisrininkų paskyrimas

- Tuomet x darbininkų darbas kainuos

$$D = x\left(20 + 6\frac{1}{60}t\right) = x(20 + 0{,}1\,t).$$

- Nuostolio funkcija bus

$$y = 1280(t+l) + x(20+0{,}1\,t).$$

- Yra lygybinis apribojimas – per laik1 t darbininkai turi pastatyti 2000 m užtvara:

$$2tx = 2000.$$

Gaisrininkų paskyrimas

- Apribojimą, kartu ir nežinomą kintamąjį t, galima eliminuoti, jį išreiškę per x:

$$t = \frac{2000}{2x} = \frac{1000}{x}.$$

- Iš čia išplaukia:

$$y = 1280\left(\frac{1000}{x}+l\right) + x\left(20+0{,}1\,\frac{1000}{x}\right) = \frac{1280000}{x} + 20x + 100 + 1280\,l.$$

- Gavome minimizavimo uždavinį su sveikuoju kintamuoju:

$$\min_x y = \min_x \left\{ \frac{1280000}{x} + 20x + 100 + 1280\,l \right\}.$$

Būtinos optimumo sąlygos

- Būtina optimumo sąlyga yra sąlyga, kuri visada galioja, jei taške yra optimumas.
- Lokaliojo minimumo sąlygos idėja: visomis galimomis kryptimis iš optimumo taško funkcija nemažėja.

- Konstruojant optimizavimo metodus ir įrodant tų metodų bei optimumų savybes funkcijos dažnai aproksimuojamos Teilogo daugianatiais.

- **Teiloro teorema**. Jei funkcija $f(X)$ yra n+1 kartą tolydžiai diferencijuojama taško X aplinkoje $N_\varepsilon(X) = \{Z: d(X,Z) \leq \varepsilon\}$, tada kiekviename tos aplinkos taške Z ją galima išreikšti Teiloro skleidiniu:

$$f(Z) = \sum_{i_1=0}^{n} \ldots \sum_{i_n=0}^{n} \left\{ \frac{\partial^{i_1}}{\partial x_1^{i_1}} \ldots \frac{\partial^{i_n}}{\partial x_n^{i_n}} \frac{f(X)}{i_1! \ldots i_n!} (z_1 - x_1)^{i_1} \ldots (z_n - x_n)^{i_n} \right\} + O\left(\|Z - X\|^{n+1}\right).$$

Būtinos optimumo sąlygos

- Iš teoremos matome, kad visada atsiras pakankamai maža taško X aplinka, kurioje funkcija $f(X)$ galima norimu tikslumu aproksimuoti n-ojo laipsnio daugianariu.
- Tai išplaukia iš visų Teilogo skleidinio narių tolydumo ir iš to, kad liekamasia narys yra aukštesnio laipsnio ir mažindami ε, paklaida galime pafdaryti norimai mažą.

- **Apibrėžimas**. Diferencijuojamos funkcijos $f\colon R^n \to R$ gradientą (dalinių išvestinių taške X vektorių) pažymėkime $\blacktriangledown f(X)$.

- *Gradientas?*

Būtinos optimumo sąlygos

- Vieną arba du kartus tolydžiai diferencijuojamų funkcijų pirmosios ar antrosios eilės Teiloro skleidiniai taške $Z = X + \lambda\, p$ vektorine forma atitinkamai užrašomi taip:

$$f(Z) = f(X + \lambda p) = f(X) + \lambda \nabla f(X)\, p^T + O\left(\lambda^2 \|p\|^2\right),$$

$$f(Z) = f(X) + \lambda \nabla f(X)\, p^T + \frac{\lambda^2}{2} p H(X)\, p^T + O\left(\lambda^3 \|p\|^3\right),$$

Čia λ – skaliaras, p – n-matis vektorius, T – transponavimo ženklas, H – hesianas.

Būtinos optimumo sąlygos

- Jei funkcija yra tolydžiai diferencijuojama, tai kryptinė išvestinė kryptimi p

$$\frac{d\,f(X)}{d\,p} = \frac{\lim\limits_{h \to 0} f(X + h\,p) - f(X)}{h}$$

yra lygi $\nabla f(X)p^T$, čia p yra vienetinis krypties vektorius. Tai tiesiogiai išplaukia iš Teiloro teoremos pirmosios eilės Teiloro skleidiniui.

- **Apibrėžimas**. Taške X iš D vektorius p vadinamas leistinąja kryptimi, jei egzistuoja toks $\rho>0$, kad $X+\lambda p$ priklauso D visiems $0 \leq \lambda \leq \rho$.
- T.y. leistinoji kryptis yra kryptis, nevedanti už leistinuosios srities ribų bent jau taško X aplinkoje.

Būtinos optimumo sąlygos

- **Teorema**. Jei taške X^* iš D yra lokalusis minimumas, sritis D iškila, o funkcija f tolydžiai diferencijuojama, tai kryptinės išvestinės visomis taške X^* leistinomis kryptimis p yra neneigiamos:

$$\nabla f(X) p^T \geqslant 0.$$

Būtinos optimumo sąlygos

- Kagangi sritis iškila, kiekvieną lokaliojo minimumo taško X^* aplinkos tašką X galime sujungti su tašku X^* leistinąja kryptimi p, kurios visoje atkarpoje $[X; X^*]$ tenkinama nelygybė

$$f(Z) \geq f(X^*), X \leq Z \leq X^*.$$

- Iš Teiloro teoremos turime:

$$f(Z) = f(X^* + \lambda p) = f(X^*) + \lambda \nabla f(X^*) p^T + O(\lambda^2 \|p\|^2) =$$
$$= f(X^*) + \lambda \left\{ \nabla f(X^*) p^T + O(\lambda \|p\|^2) \right\}.$$

- Kadangi išvestinės tolydžiai diferencijuojamos, nelygybė

$$f(X^* + \lambda p) - f(X^*) = \lambda \left\{ \nabla f(X^*) p^T + O(\lambda \|p\|^2) \right\} \geqslant 0.$$

galioja bet kokiems mažiems λ tik jei $\nabla f(X^*) p^T \geqslant 0.$

$\blacksquare$

Būtinos optimumo sąlygos

- Kai uždavinys neturi apribojimų, būtina optimumo sąlyga $\nabla f(X^*) p^T \geqslant 0$ turi galioti visomis kryptimis. Tai gali būti tik jei $\nabla f(X^*) = 0$.

- **Apibrėžimas**. Taškai, kuriuose gradientas yra nulinis, vadinami **stacionariaisiais**.

- Sąlyga $\nabla f(X^*) = 0$ vadinama pirmosios eilės būtina optimumo sąlyga.

- Pirmosios eilės optimumo sąlyga yra tik būtina, bet ne pakankama minimumo sąlyga ir gali būti tenkinama tiek minimumo, tiek maksimumo, tiek balno taške.

- Norėdami nustatyti optimumo tipą, remsimes antrosios eilės išvestinėmis, kurių matricų savybės pasako apie tikslo funkcijos iškilumą ar įgaubtumą.

$\blacksquare$

Būtinos optimumo sąlygos

- Tarkime, kad funkcija f yra dukart tolydžiai diferencijuojama funkcija, t. y. Turi tolydžių antųjų išvestinių matricą $H(X)$. Tada funkciją galime skleisti antrosios eilės Teiloro daugianariu:

$$f(X) = f(X^* + \lambda p) = f(X^*) + \lambda \nabla f(X^*) p^T + \frac{\lambda^2}{2} p H(X^*) p^T + O\left(\lambda^3 \|p\|^3\right).$$

- Iš nelygybės $f(X^* + \lambda p) - f(X^*) \geqslant 0$ išplaukia:

$$\nabla f(X^*) = 0.$$

Jei $\nabla f(X^*) = 0$, tai $p H(X^*) p^T \geqslant 0$.

Antrosios eilės būtinos minimumo sąlygos

- Teorema. Jei taške X^* iš D yra lokalusis minimumas, sritis D iškila, o funkcija f dukart tolydžiai diferencijuojama, tai visoms taške X^* leistinomis kryptimis p galioja būtinos minimumo sąlygos:

$$\nabla f(X^*) = 0.$$

Jei $\nabla f(X^*) = 0$, tai $p H(X^*) p^T \geqslant 0$.

- Prisiminsime, kad jei bet kuriems vektoriams X galioja $XAX^T > 0$, tai kvadratinė forma ir atitinkama kvadratinė matrica A vadinamos teigiamai apibrėžtomis.

- Jei nėra apribojimų, tai iš šios teorėmos tiesiogiai išplaukia kita teorema.

Antrosios eilės būtinos minimumo sąlygos

- Teorema. Jei taške X^* iš R^n yra lokalusis minimumas, o funkcija f dukart tolydžiai diferencijuojama, tai taške X^* galioja šios būtinos minimumo antrosios eilės sąlygos:

$$\nabla f\left(X^*\right) = 0.$$

Hesianas $H(X^*)$ yra neneigiamai apibrėžtas

Pakankamos optimumo sąlygos

- Pakankamosiomis vadinamos tos sąlygos, iš kurių išplaukia optimumo egzistavimas duotame taške.

- Teorema. Tarkime, kad funkcija f dukart tolydžiai diferencijuojama. Jei taške X^* iš R^n tenkinamos šios antrosios eilės pakankamos minimumo sąlygos:

  $$\nabla f\left(X^*\right)=0.$$

  Hesianas yra teigiamai apibrėžtas.

tai taške X^* yra lokalusis minimumas.

Pakankamos optimumo sąlygos

- **Įrodymas**. Kadangi $\nabla f\left(X^*\right)=0$, antrosios eilės Teiloro daugianarį perrašome taip:

$$f\left(X\right)=f\left(X^*+\lambda p\right)=f\left(X^*\right)+\lambda^2\left\{\frac{1}{2}pH\left(X^*\right)p^T+O\left(\lambda\|p\|^3\right)\right\}.$$

- $pH(X^*)p^T > 0$, todėl kiekvienam p atsiras toks λ_m, kad visiems λ iš intervalo $(0;\lambda_m]$ galios nelygybė $f(X) > f(X^*)$.

- Parodėme, kad galiojant teoremos sąlygoms egzistuoja taško X^* aplinka, kurioje $f(X) > f(X^*)$.

∎

Pavyzdys. Kopečių ilgis

- Kopečios turi remtis į sieną, grindis ir į $a \times b$ dydžio dėžės briauną. Optimizavimo tikslas rasti tokį kampą x, kad kopečių ilgis y būtų mažiausias.

Optimizavimo algoritmų sustojimo sąlygos

- Optimizavimo algoritmas generuoja taškų seką $\{X_n\}$ taip, kad sekos ribiniai taškai artėtų prie optimumų.
- Lokaliojo, o kartais ir globaliojo optimumo paieškos algoritmams reikia nurodyti pradinį tašką X_0 bei sustojimo sąlygų parametrus.
- Dažniausiai naudojami sustojimo sąlygos: pagal **gradiento normą**, pagal **tikslo funkcijos** reikšmes, pagal **argumento** reikšmes:

$$\left\|\nabla f\left(X_k\right)\right\| < \varepsilon .$$

Algoritmo generuojama seka gali nepasiekti pakankamai mažos gradiento normos arba funkcija gali būti nediferencijuojama. Be to, jei tikslo funkcijos paviršios turi griovių, tai algoritmas gali šokinėti apie optimuma mažais žingsneliais. Todėl tikslinga stabdyti algoritmą, kai tikslo funkcijos arba argumento reikšmės per iteracija mažai pakinta:

$$\left|f\left(X_k\right) - f\left(X_{k-1}\right)\right| < \varepsilon .$$

$$\left\|X_k - X_{k-1}\right\| < \varepsilon .$$

Optimizavimo uždavinių klasifikacija

- Klasifikacija nėra formalus dalykas. Kiekvienai klasei sukurti specifiniai sprendimo metodai pagal tos klasės savybes.
- Klasifikuojama pagal tikslo funkciją / apribojimus bei argumentus.

- **Tiesinis programavimas**: tikslo funkcija ir apribojimai yra tiesiniai.
- **Netiesinis programavimas**: tikslo funkcija ar apribojimai yra netiesiniai.
- **Iškilusis programavimas**: tikslo funkcija ir leistinoji kintamųjų sritis yra iškilos.
- **Diskretus programavimas**: kintamųjų aibė yra diskreti.
- **Sveikaskaitis programavimas**: argumentai yra sveikieji skaičiai.
- **Bulinis programavimas**: kintamieji įgyja reikšmes 0 arba 1.

Optimizavimo uždavinių klasifikacija

- Atsižvelgiant ir į tikslo funkciją ir, į apribojimus, optimizavimo uždaviniai skirstomi į **lokaliojo optimizavimo** (vienaekstremius) ir **globaliojo optimizavimo** (daugiaekstremius) uždavinius.

Optimizavimo metodų palyginimas

- Universalių optimizavimo metodų nėra!
- Kiekvienas metodas turi savo stipriąsias ir silpnąsias puses. Net ir tobuliausiam metodui galima sukurti specialų uždavinį, kurio metodas «neįkas».

- Metodų palyginimo rezultatai gali priklausyti nuo to
  - kaip metodas suprogramuotas;
  - kokios sustojimo sąlygos;
  - kokios metodo specifinių parametrų reikšmės;
  - kaip parinktas pradinis taškas;
  - koks parinktas pradinis žingsnio ilgis.

Optimizavimo metodų palyginimas

- **Pagrindiniais lyginimo kriterijais** laikytini šie:
  - tam tikros uždavinių klasės optimumo radimo sėkmė;
  - tikslo funkcijos reikšmių skaičiavimų skaičius;
  - programos atlikimo trukmė;

- Reikia atkreipti dėmesį į tai, kad programa gali «įstrigti» dar nepasiekus optimumo, kai tikslo funkcija netenkina prielaidų, kuriomis pagrįstas metodas, o programuotojai tokių avarinių atvejų nemumatė.

-

Optimizavimo metodų palyginimas

- Optimumo radimo sėkmė yra glaudžiai susijusi su sprendinio tikslumu.
- Vienos uždavinių klasės tikslumas gali būti nepriimtinas kitai uždavinių klasei.
- Pvz., lokaliesiems metodams, kuriais sprendžiami vienaekstremiai uždaviniai, keliami aukšti tikslumo reikalavimai. Tuo tarpu spręsdami globaliojo optimizavimo uždavinius su daugelio kintamųjų ir sudėtinga tikslo funkcija esame priversti tenkintis apytiksliais sprendiniais.

Optimizavimo metodų palyginimas

- Tikslumas matuojamas surasto optimalaus taško X_k atstumu tuo tikrojo optimalaus taško X*, arba TF reikšmių skirtumu:

$$\Delta X = \left\| X_k - X^* \right\|, \qquad \Delta f = \left| f(X_k) - f(X^*) \right|.$$

- Tikslumo matas parenkamas atsižvalgiant į TF pobūdį.

- Jei optimumo aplinkoje TF kinta nedaug, racionalu vartoti matą ΔX.
  - Matas ΔX paplitęs uždaviniuose, kai lokalusis optimumas yra labai toli nuo globaliojo optimumo, tuo tarpu atitinkamos TF reikšmės mažai skiriasi.

- Jei uždavinys sprendžiamas norimu tikslumu ε, metodo efektyvumo kriterijumi įprasta laikyti TF skaičiuojamų reikšmių skaičių.
  - Šis matas pasiteisina, kai apskaičiuoti TF reikšmę duotajame taške yra brangu, reikia daug skaičiavimo laiko.

Optimizavimo metodų palyginimas

- Jei metodas yra skirtas optimizuoti greitai skaičiuojamas funkcijas, tada kaip lyginimo kriterijus naudojamas skaičiavimo laikas. Taikant sudėtingus optimizavimo metodus, skaičiavimo laiką lemia pats metodas, tad skaičiavimo laikas gerai atspinti tikrąją metodo vertę.

- Naudojant skirtingų charaktarsitikų kompiuterius, reikia į tai atsižvalgti, vertinant optimizavimo metodų tyrimo rezultatus. Be to, skaičiavimo laiką dar lemia programavimo kultūra ir stilius.

- Efektyvumas priklauso ir nuo įvairiausių iš anksto parenkamų parametrų, kurių apstu daugelyje metodų. Parametrų parinkimas savaime tampa atskiru optimizavimo uždaviniu.

- Be to, labai didelė yra ir testinių uždavinių parinkimo įtaka.

Optimizavimo metodų palyginimas. Testiniai uždaviniai

- Testinių uždavinių pobūdis turėtų būti kuo artimesnis praktiniams uždaviniams, tačiau pageidautina, kad jie būtų sprendžiami sparčiai – eksperimentuojant, juos tenka spręsti daug kartų.

- Testiniuoje uždaviniuose stengiamasi koncentruoti spendimo metodui neparankias savybes ir sunkumus:
  - lokaliesiems metodams parenkamitestiniai uždaviniai su grioviais funkcijos reikšmių paviršiuje
  - globaliesiems – su daugeliu minimumų, daugelių kintamųjų.

Optimizavimo metodų palyginimas. Testiniai uždaviniai

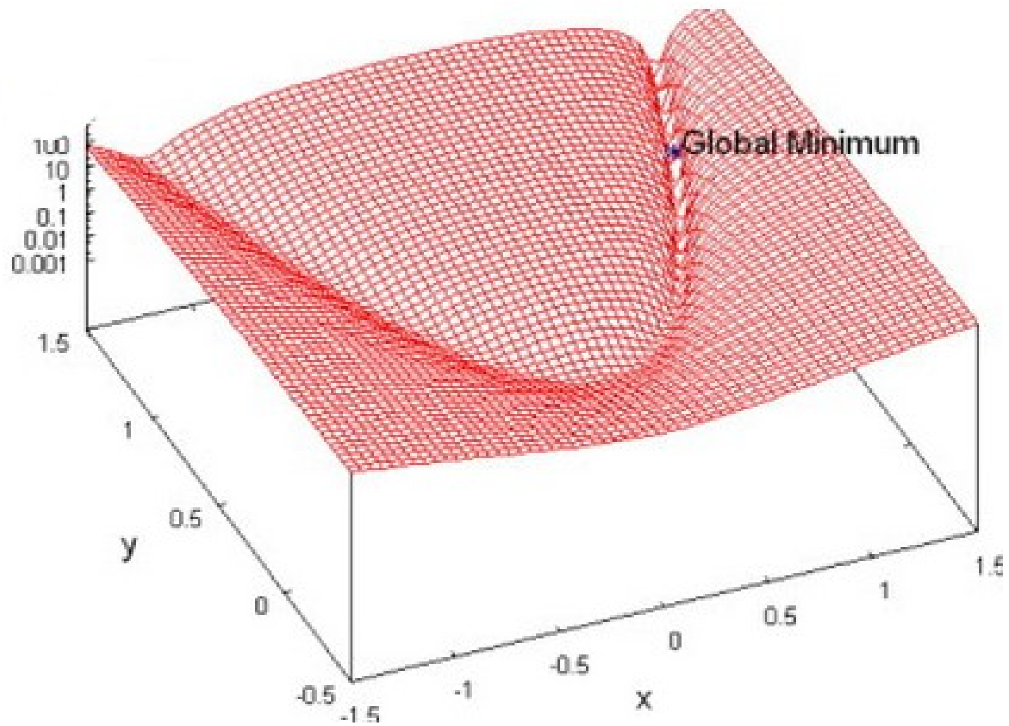
- Tiriant lokaliuosius metodus kaip testinė funkcija dažnai naudojama **Rozenbroko funkcija**, kuruos paviršius yra lenktas griovys.

$$f(X) = \sum_{i=1}^{n-1} \left[100 (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right], \quad X \in \mathbb{R}^n.$$

Optimizavimo metodų palyginimas.

- **Dviejų kintamųjų Rozenbroko funkcija**

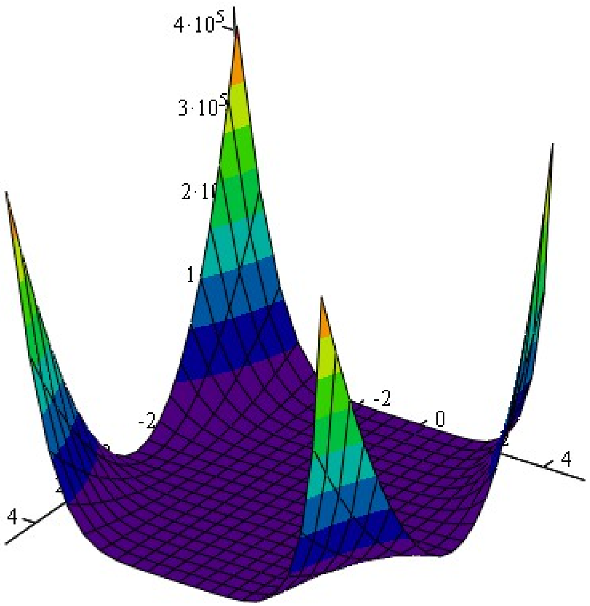
$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

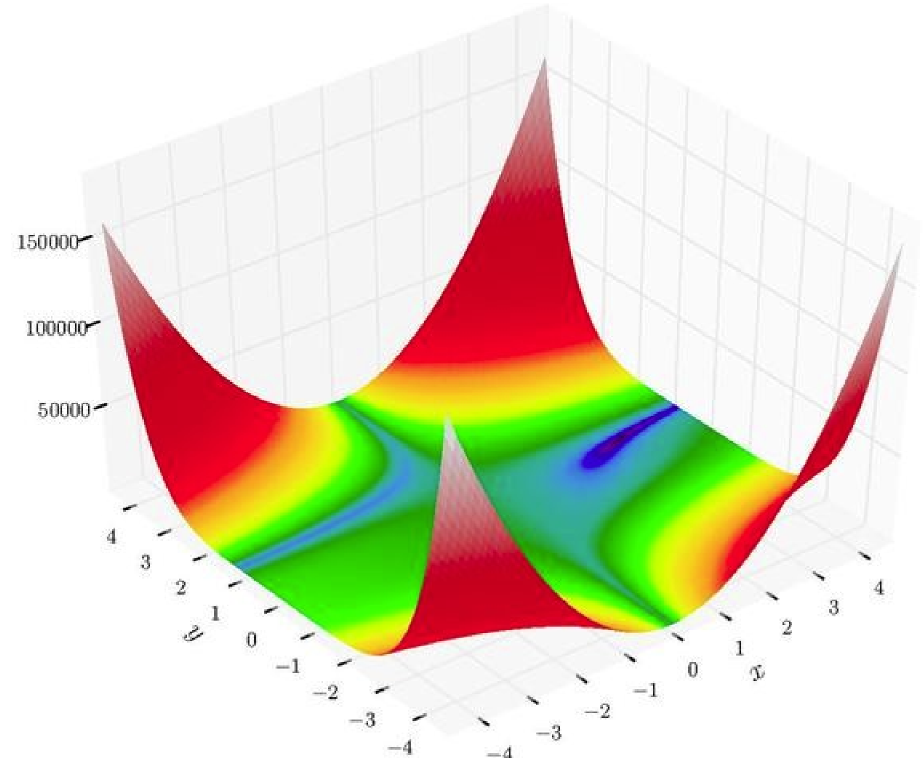


Optimizavimo metodų palyginimas.

- **Dviejų kintamųjų Bilo funkcija**

$$f(x_1, x_2) = (a - x_1 + x_1 x_2)^2 + (b - x + x_1 x_2^2)^2 + (c - x_1 + x_1 x_2^3)^2$$

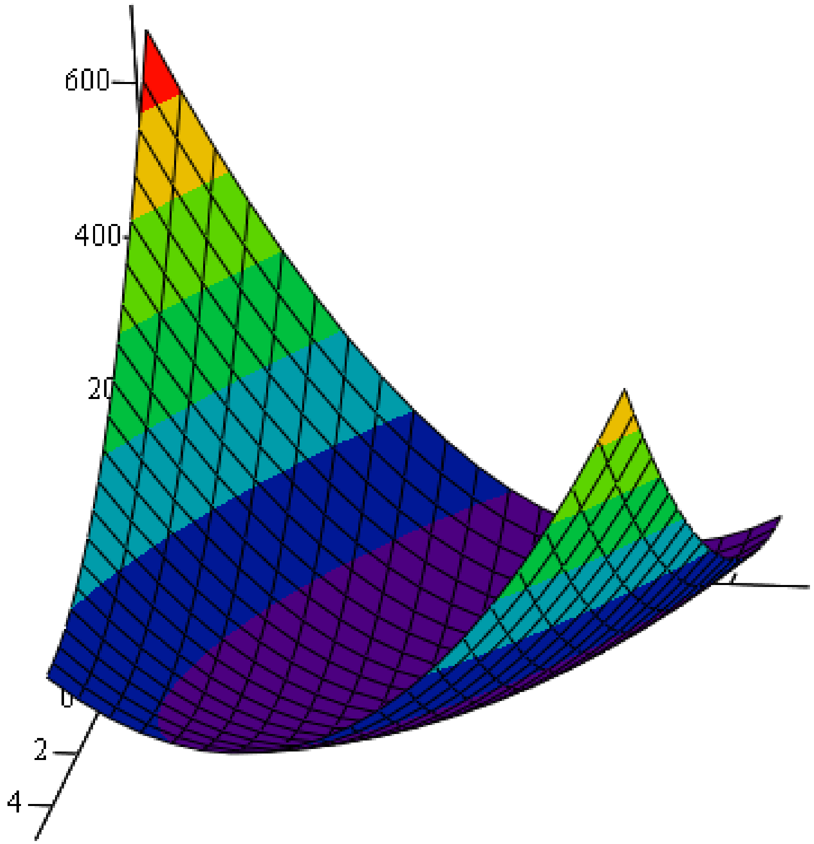




Optimizavimo metodų palyginimas.

- **Dviejų kintamųjų Buto funkcija**

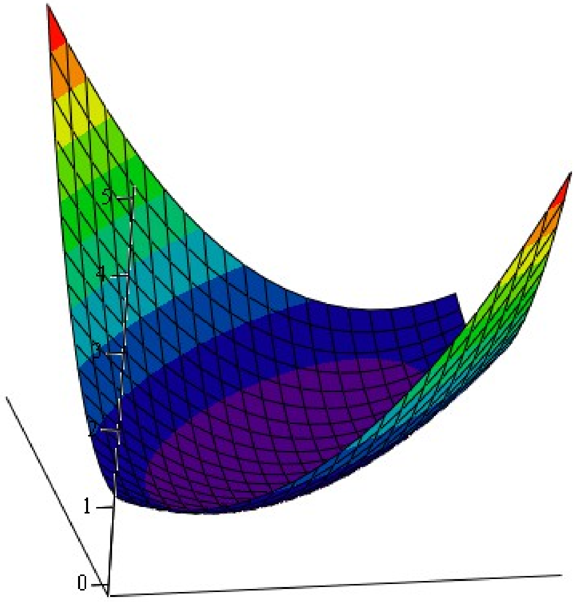
$$f(x_1, x_2) = (a_{11} x_1 + a_{12} x_2 - b_1)^2 + (a_{21} x_1 + a_{22} x_2 - b_2)^2$$

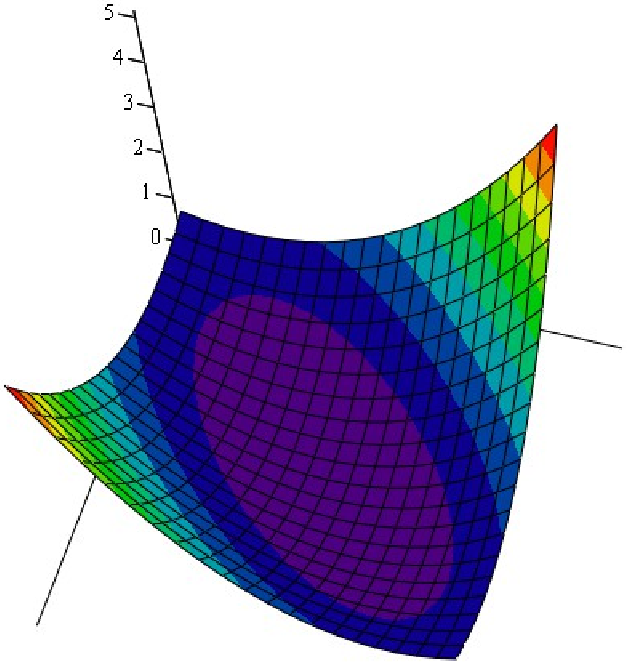


Optimizavimo metodų palyginimas.

- **Dviejų kintamųjų Matyaso funkcija**

$$f(x_1,\ x_2) = a(x_1^2 + x_2^2) - b\,x_1 x_2$$

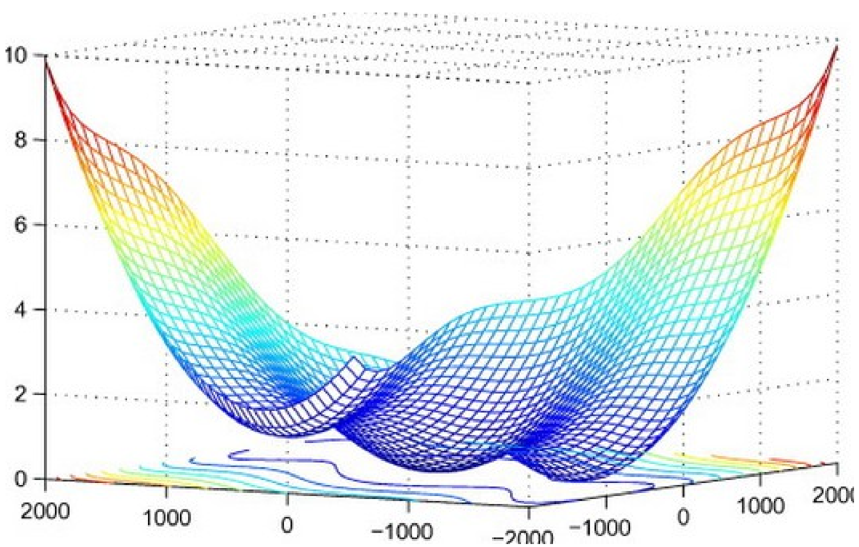


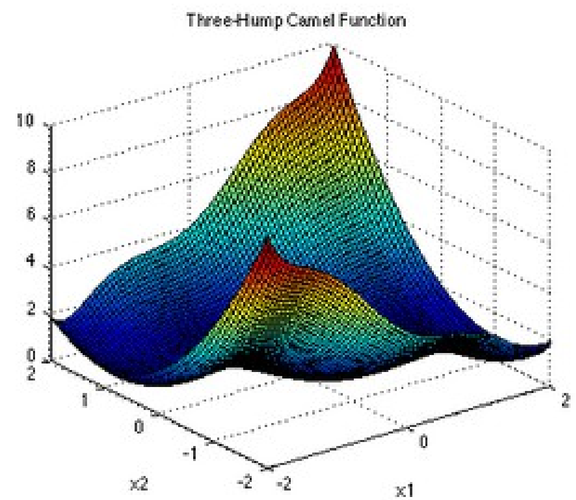


Optimizavimo metodų palyginimas.

- **Kupranugario funkcija**

$$f(x_1,\ x_2) = 2x_1^2 - 1{,}05\,x_1^4 + \frac{x_1^6}{6} + x_1 x_2 + x_2^2$$





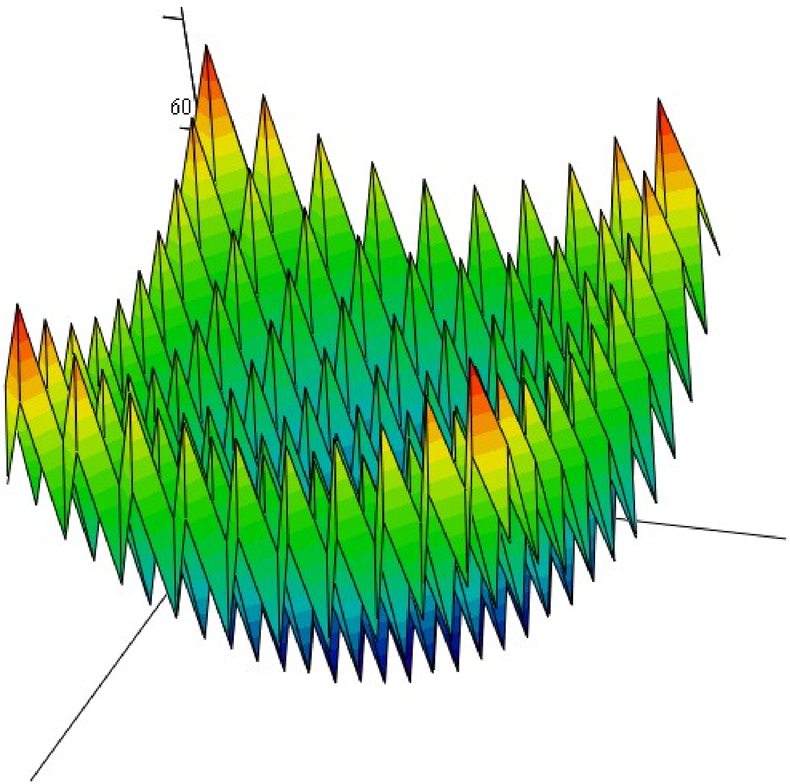
Optimizavimo metodų palyginimas. Testiniai uždaviniai

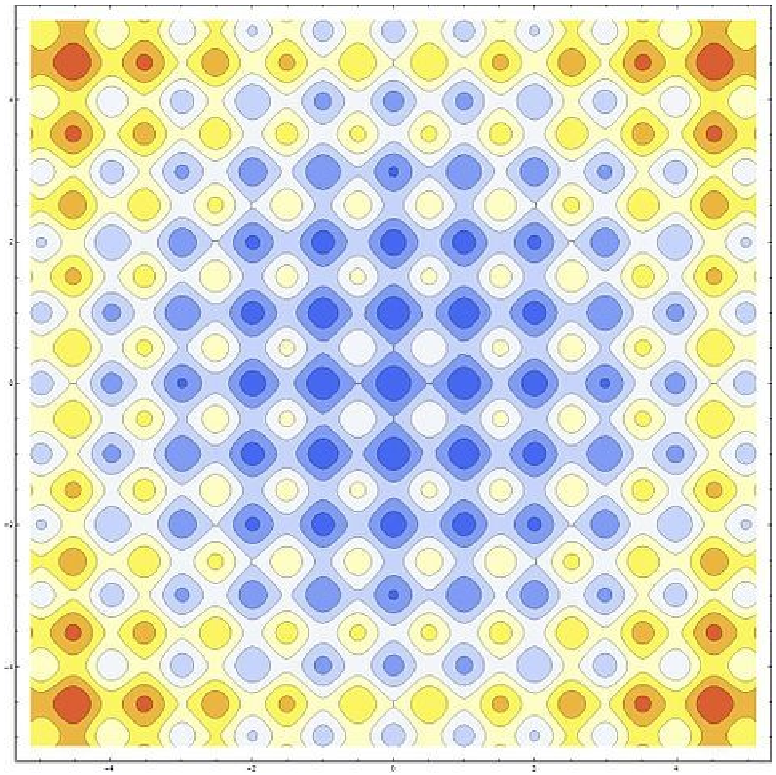
- Tiriant globaliuosius optimizavimo metodus kaip testinė funkcija dažnai naudojama daugiaekstremė **Rastrigino funkcija:**

$$f(X) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right), \quad X \in \mathbb{R}^n.$$

- Dviejų kintamųjų **Rastrigino funkcija:**

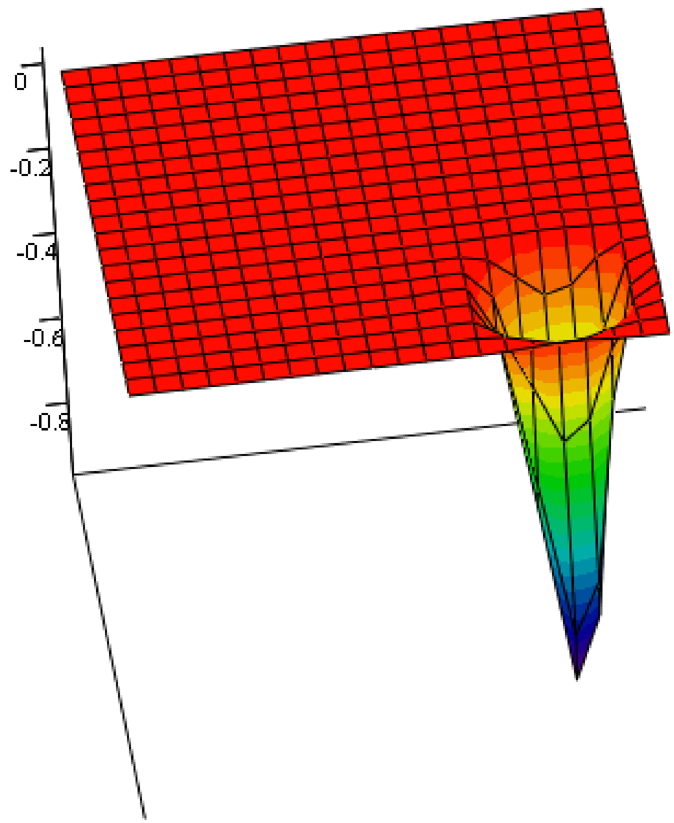
$$f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10\cos(2\pi x_1) - 10\cos(2\pi x_2).$$





- **Isomo funkcija:**

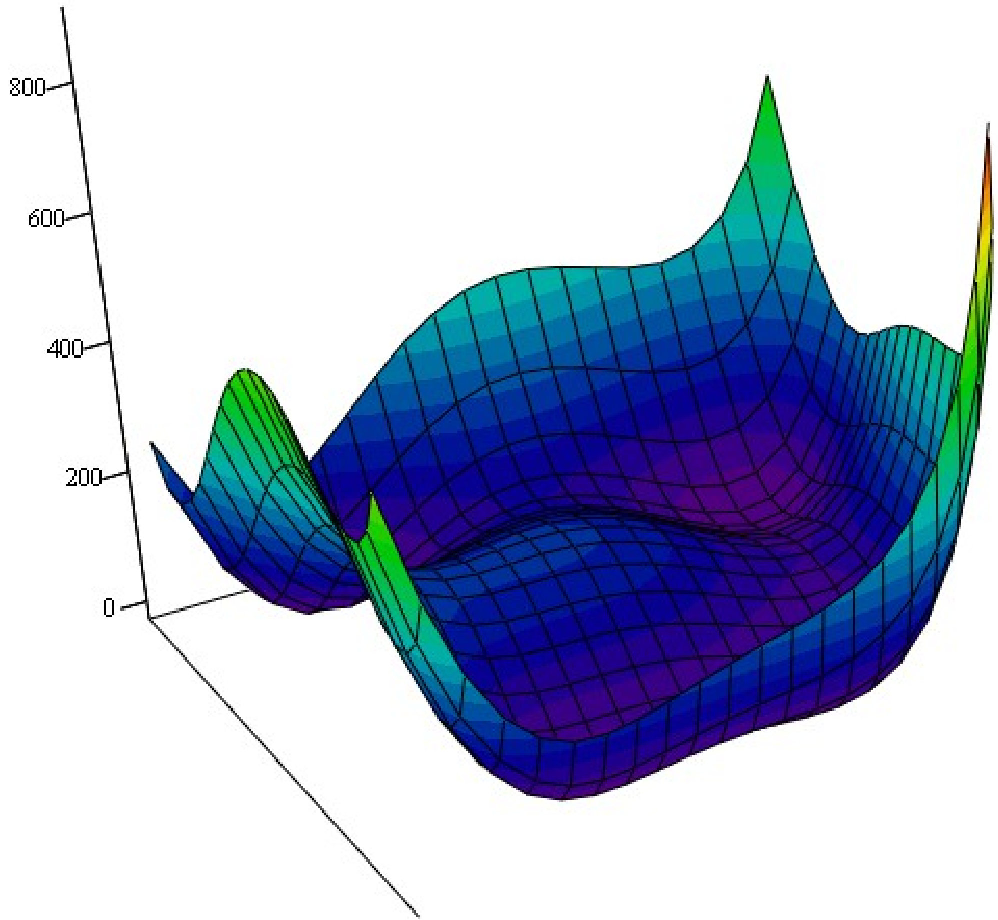
$$f(x_1, x_2) = -\cos x_1 \cos x_2 e^{-(x_1-\pi)^2-(x_2-\pi)^2}.$$



Optimizavimo metodų palyginimas. Testiniai uždaviniai globaliesiems metodams

- **Himelblau funkcija:**

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$



Daugelio kintamųjų funkcijų optimizavimo (be apribojimų) metodų palyginimas. Kursinio darbo planas (~25 psl.)

1. Įvadas (problema, problemos aktualumas, istorinė/bibliografinė apžvalga).
2. Metodas X. Metodas Y. Aprašymas. Privalumai. Trūkumai. Schema.
3. Testinė funkcija (R^2 *ir* R^n). Aprašymas, grafikas.
4. Eksperimentų serijos (testavimo plano) aprašymas: tikslumas (10^{-2}, 10^{-4}, 10^{-8}), tikslumo matai (ΔX, Δf) ir sustojimo salygos, metodų parametrų ir pradinių taškų parinkimas.
5. Skaičiavimų rezultatai (lentelės, vizalizavimas).
6. Rezultatų analizė ir išvados.
7. Literatūra.
8. Priedai.

Daugelio kintamųjų funkcijų optimizavimo metodai

- Negradientiniai metodai
  - Simplekso / Nelderio-Mido metodas (1)
  - Koordinatinės paieškos metodas (2)
  - Huko-Dživso paieškos metodas (3)
  - Pauelo jungtinių krypčių metodas (4)
- Gradientiniai nusileidimo metodai
  - Greičiausio nusileidimo metodas (5)
  - Niutono metodas (6)
  - Broideno metodas (7)
  - Devidono-Fletčerio-Pauelo (DFP) metodas (8)
  - Broideno–Fletčerio–Goldfarbo–Šano (BFGS) metodas (9)

Vieno kintamojo funkcijų optimizavimas. Uždavinių savybės

- Nagrinėsime vieno kintamojo $f(x)$: $R \to R$ tolydžių funkcijų optimizavimą.
- Vienmatė paieška – neatsiejama daugelio kintamųjų funkcijų optimizavimo metodų sudedamoji dalis.

- **Monotoniškumas**. Jei bet kuriems $X_1 < X_2$ iš srities D galioja $f(X_1) > f(X_2)$, tai funkcija yra monotoniškai mažiejanti, o jei $f(X_1) < f(X_2)$, tai funkcija yra monotoniškai didėjanti srityje D.

- **Unimodali funkcija**. Funkcija yra unimodali atkarpoje tada ir tik tada, jei ji monotoniška abipus vienintelio šiame intervale optimumo taško.

Vieno kintamojo funkcijos optimumo identifikavimas

- Tarkime, vieno kintamiojop funkcija $f(x)$ yra apibrėžta intervale (a, b) ir šiame intervale $n+1$ kartą diferencijuojama.
- Jei $a < x^* < b$ ir $a < x^* + \varepsilon < b$, tuomet pagal Teiloro teoremą taške $x^* + \varepsilon$ funkciją galime išskleisti Teiloro skleidiniu:

$$f(x^* + \varepsilon) = f(x^*) + \varepsilon f'(x^*) + \frac{\varepsilon^2}{2!} f^{(2)}(x^*) + \ldots + \frac{\varepsilon^n}{n!} f^{(n)}(x^*) + O_{n+1}(\varepsilon),$$

kur $O_{n+1}(\varepsilon)$ – skleidinio paklaida, proporcinga ε^{n+1}; mažindami ε, galime kiek norime sumažinti paklaidą.

Vieno kintamojo funkcijos optimumo identifikavimas

- Jei x^* yra lokaliojo minimumo taškas intervale (a, b), tada turi egzistuoti tam tikra taško x^* ε-aplinka, kuruios visiems taškams $x^* - \varepsilon \le x \le x^* + \varepsilon$ galioja $f(x) > f(x^*)$. Iš čia išplaukia, jog

$$f(x^* + \varepsilon) - f(x^*) = \varepsilon f'(x^*) + \frac{\varepsilon^2}{2!} f^{(2)}(x^*) + ... + \frac{\varepsilon^n}{n!} f^{(n)}(x^*) + O_{n+1}(\varepsilon) > 0$$

- Jei ε mažas, didžiausią įtaką turi pirmasis narys. ε gali būti ir teigiamas, ir neigiamas, todėl nelygybė bus tenkinama tik kai $f'(x^*) = 0$.
- Analogiškai mąstant nesunku įsitikinti, kad nelygybė bus teisinga tik kai $f^{(2)}(x^*) > 0$.

- **Teorema**.

  Būtina sąlyga, kad x^* būtų lokaliojo minimumo taškas: $f'(x^*) = 0; f^{(2)}(x^*) > 0$.

  Būtina sąlyga, kad x^* būtų lokaliojo maksimumo taškas: $f'(x^*) = 0; f^{(2)}(x^*) < 0$.

Vieno kintamojo funkcijos optimumo identifikavimas. Pakankamos ekstremumo egzistavimo sąlygos

- Jei funkcija $f(x)$ yra dukart diferencijuojama taške x^* ir $f'(x^*) = 0$; $f^{(2)}(x^*) > 0$, tai taške x^* funkcija turi lokalųjį minimumą.
- Jei funkcija $f(x)$ yra dukart diferencijuojama taške x^* ir $f'(x^*) = 0$; $f^{(2)}(x^*) < 0$, tai taške x^* funkcija turi lokalųjį maksimumą.

- Tarkime, kad $f(x)$ yra n kartų diferencijuojama taške x^* . Tegu $f^{(k)}(x^*) = 0$, kai $k = 1, \ldots n\text{-}1$, o $f^{(n)}(x^*) \neq 0$. Jei n yra lyginis, tai $f(x)$ taške x^* turi minimumą, jei $f^{(n)}(x^*) > 0$, arba maksimumą, jei $f^{(n)}(x^*) < 0$. Jei n yra nelyginis – ekstremumo nėra.

Vieno kintamojo funkcijos optimumo identifikavimas

- Kaip rasti vieno kintamojo funkcijos globalųjį minimumą paprasčiausiu būdu?
- Sprendžiamas uždavinys

$$\min_{x \in [a, b]} f(x).$$

- Randame visus kritinius taškus atkarpoje [a, b]: $\{x_k\}$.
- Apskaičiuojame $\{f(x_k)\}$, $f(a)$, $f(b)$.
- Globalųjį minimumą atitiks **min{{$\boldsymbol{f(x_k)}$}, $\boldsymbol{f(a)}$, $\boldsymbol{f(b)}$}**.

- **Pavyzdys**. Rasti funkcijų minimumus srytyje $D = \{x\text{: } -2 \leq x \leq 4\}$

$$f(x) = -x^3 + 3x^2 + 9x + 10$$
$$g(x) = x^4 + 32x - 3$$

Interpoliaciniai optimizavimo algoritmai

- Šiuose algoritmuose nauji bandymo taškai randami taikant tikslo funkcijos ekstrapoliaciją ir interpoliaciją. Funkcija aproksimuojama daugianariu, ir kaip naujas taškas imamas to daugianario minimumo taškas.

- Deviso-Sveno-Kempi algoritmas
- Pauelo algoritmas
- Kvadratinė interpoliacija su išvestinėmis

- Vieno kintamojo funkcijos minimizavimas dažnai būna pagalbinis veiskmas daugelio kintamųjų funkcijos minimizavimo algoritmuose. Pastaruosiuose dažnai naudojamos išvestinės. Todėl ir vienmačiame algoritme tikslinga panaudoti jau turimą išvestinės pagal kryptį reikšmę.

Deviso-Sveno-Kempi algoritmas (Davies-Swann-Campey)

1. Taikomas Sveno metodas minimumo taško intervalui $[X_n; X_{n+2}]$ nustatyti.
2. Prie trijų paskutinių Sveno metodo sekos taškų X_n, X_{n+1}, X_{n+2} pridedamas trečias (tarp X_{n+1} ir X_{n+2}).
3. Iš keturių vienodais atstumais išdėstytų taškų atmetamas vienas iš kraštinių, kuris yra toliau nuo X, atitinkančio mažiausią rastą funkcijos reikšmę.
4. Pagal tris atrinktus taškus, tikslo funkciją aproksimuojame parabole.
5. Sprendinį patikslinti galima sumažinus Sveno metodo žingsnį ir startuojant iš 4 žingsnyje rasto artinio. Algoritmą tikslinga stabdyti, kai tikslo funkcijos arba argumento reikšmės per iteraciją mažai pakinta.

$$\left|f(X_k) - f(X_{k-1})\right| < \varepsilon .$$

$$\left|X_k - X_{k-1}\right| < \varepsilon .$$

Pauelo algoritmas

- Atliekama kvadratinė aproksimacija pagal tris nebūtinai vienodai vienas nuo kito nutolusius taškus, kutie gaunami pradžioje padarius bandamuosius žingsnius iš pradinio taško.
- Algoritmui duodamas pradinis taškas, pradinis žingsnis ΔX ir tikslumas.

1. Pagal pradinį tašką X_1 skaičiuojame $X_2 = X_1 + \Delta X$.
2. Imame dar vieną tašką X_3 funkcijos mažėjimo kryptimi (žingsnis ΔX).
3. Pagal tris atrinktus taškus, tikslo funkciją aproskimuojame parabole. Surandame aproksimuojančios parabolės stacionarūjį tašką:

$$\hat{X} = -\frac{1}{2}\frac{\left(X_2^2 - X_3^2\right) f\left(X_1\right) + \left(X_3^2 - X_1^2\right) f\left(X_2\right) + \left(X_1^2 - X_2^2\right) f\left(X_3\right)}{\left(X_2 - X_3\right) f\left(X_1\right) + \left(X_3 - X_1\right) f\left(X_2\right) + \left(X_1 - X_2\right) f\left(X_3\right)}$$

4. Paieška baigiama, jei parabolės vuršunės taškas nuo nors iš X_1, X_2, X_3 nenutolęs daugiau nei duotas tikslumas pagal argumentą.
5. Priešingu atvėju iš taškų, negretimų minimumo taškui, išmetamas didžiausios funkcijos reikšmės taškas.
6. Toliau vykdomas 3 žingsnis.

Pauelo algoritmas

- Skaitiniai eksperimentai parodė, kad DSK ir Pauelo algoritmų junginys yra efektyvesnis už DSK.
- Pradžioje atliekami 1-5 DSK žingsniai ir Pauelo algoritmo sustojimo sąlygų tikrinimas, toliau kartojami Pauelo algoritmo 3 žingsnis ir sustojimo sąlygų tikrinimas.
- Pauelo interpoliacijos kartojimas yra efektyvesnis už DSK kartojimą, nes taikant Pauelo algoritmą interpoliaciniai taškai neturi būti vienodai nutolę, todėl panaudojami ankstesnėse iteracijose gauti taškai.

Kvadratinė interpoliacija su išvestinėmis

- Vieno kintamojo funkcijos minimizavimas dažnai būna pagalbinis veiksmas daugelio kintamųjų funkcijos minimizavimo algoritmuose.
- Daugelio kintamųjų funkcijos minimizavimo algoritmuose dažnai naudojamos išvestinės – nusileidžiama antigradiento kryptimi, ar kita kryptimi, gauta naudojant gradientą.
- Todėl ir vienmačiame algoritme tikslinga panaudoti jau turimą išvestinės pagal kryptį reikšmę.

Kvadratinė interpoliacija su išvestinėmis

- Tarkime, pradiniame taške x_1 turime funkcijos ir jos išvestinės reikšmes $f(x_1)$ ir $f'(x_1)$. Taip pat taške x_2 turime funkcijos reikšmę $f(x_2)$. Rasime funkcijos $f(x)$ aproksimacijos $y(x) = a\,x^2 + b\,x + c$ minimumo tašką x^*.
- Turime trijų lygčių sistemą su trimis nežinomaisiaias:

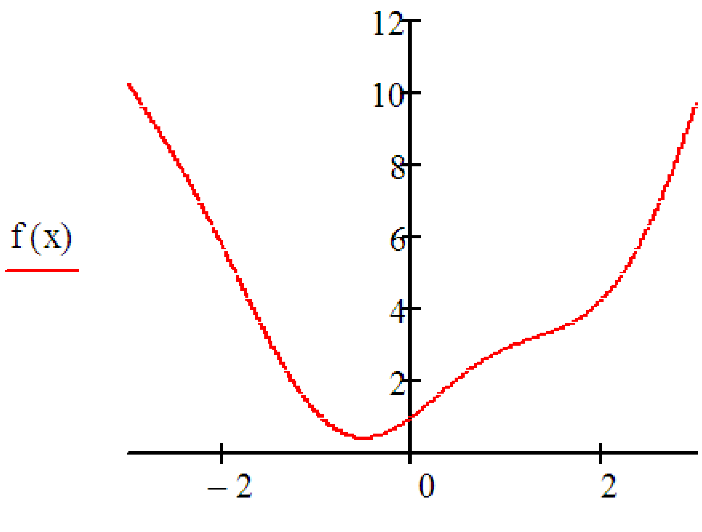
$$\begin{cases} f_1 = f(x_1) = & a\,x_1^2 + b\,x_1 + c \\ f_2 = f(x_2) = & a\,x_2^2 + b\,x_2 + c \\ f_d = f'(x_1) = & 2\,a\,x_1 + b \end{cases}$$

Kvadratinė interpoliacija su išvestinėmis

$f(x) := x^2 + \sin(2x) + 1$ $\quad \varepsilon := 0.01$

$df(x) := 2\cdot x + 2\cdot\cos(2\cdot x)$

$$h(x) := \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



$$R = \begin{pmatrix} -0.515 \\ 0.408 \\ 0.001 \end{pmatrix}$$

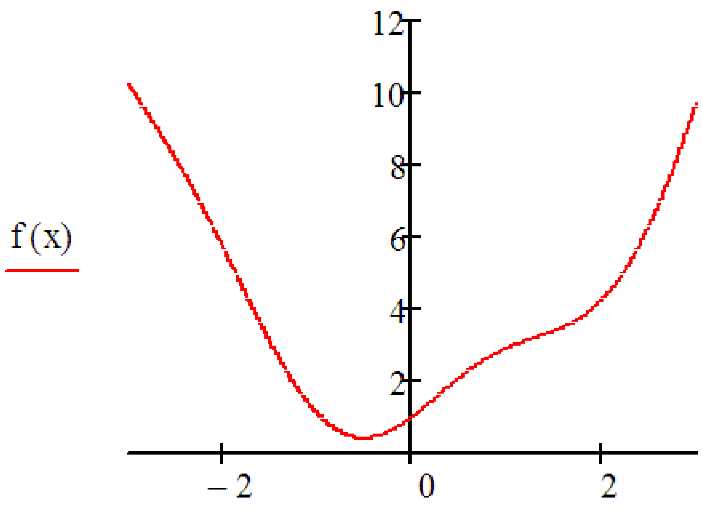
$$
R := \left|
\begin{array}{l}
x1 \leftarrow -1 \\
x2 \leftarrow 0 \\
\Delta \leftarrow |x1 - x2| \\
\text{while } \Delta > \varepsilon \\
\quad \left|
\begin{array}{l}
f1 \leftarrow f(x1) \\
f2 \leftarrow f(x2) \\
fd \leftarrow df(x1) \\
a \leftarrow \dfrac{fd\cdot(x1 - x2) - (f1 - f2)}{(x1 - x2)^2} \\
b \leftarrow \dfrac{fd\cdot(x1 - x2)^2 + 2\cdot x1\cdot[f1 - f2 - fd\cdot(x1 - x2)]}{(x1 - x2)^2} \\
xm \leftarrow \dfrac{-b}{2a} \\
x1 \leftarrow x2\cdot h(f1 - f2) + x1\cdot h(f2 - f1) \\
x2 \leftarrow xm \\
\Delta \leftarrow |x1 - x2|
\end{array}
\right. \\
\begin{pmatrix} xm \\ f(xm) \\ \Delta \end{pmatrix}
\end{array}
\right.
$$

Kvadratinė interpoliacija su išvestinėmis

$f(x) := x^2 + \sin(2x) + 1$ $\varepsilon := 0.01$

$df(x) := 2\cdot x + 2\cdot\cos(2\cdot x)$

$$h(x) := \begin{array}{|l} 1 \quad \text{if } x > 0 \\ 0 \quad \text{otherwise} \end{array}$$



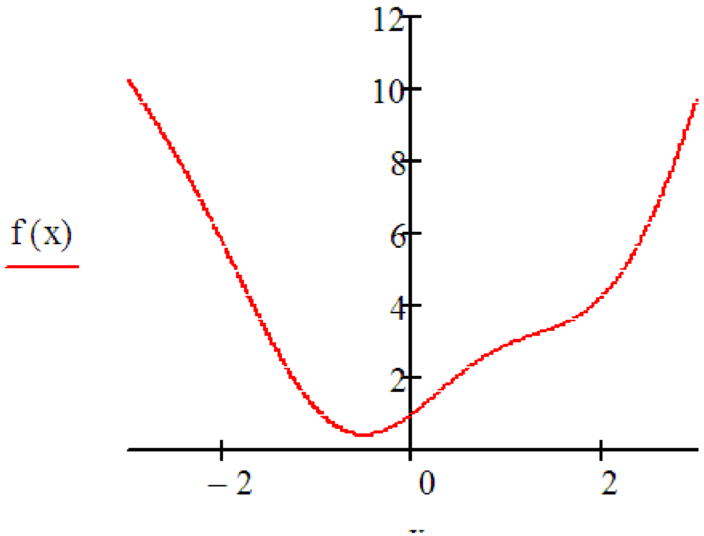
$$R := \begin{array}{|l}
x1 \leftarrow -1 \\
x2 \leftarrow 0 \\
\Delta \leftarrow x1 - x2 \\
\text{while } |\Delta| > \varepsilon \\
\quad \begin{array}{|l}
\Delta f \leftarrow f(x1) - f(x2) \\
fd \leftarrow df(x1) \\
xm \leftarrow x1 - \dfrac{\Delta}{2\left(1 - \dfrac{\Delta f}{fd\cdot\Delta}\right)} \\
x1 \leftarrow x2 \quad \text{if } \Delta f > 0 \\
x2 \leftarrow xm \\
\Delta \leftarrow x1 - x2
\end{array} \\
\begin{pmatrix} xm \\ f(xm) \\ |\Delta| \end{pmatrix}
\end{array}$$

$$R = \begin{pmatrix} -0.515 \\ 0.408 \\ 0.001 \end{pmatrix}$$

Kvadratinė interpoliacija su išvestinėmis

$f(x) := x^2 + \sin(2x) + 1 \qquad \varepsilon := 0.01$

$df(x) := 2\cdot x + 2\cdot\cos(2\cdot x)$



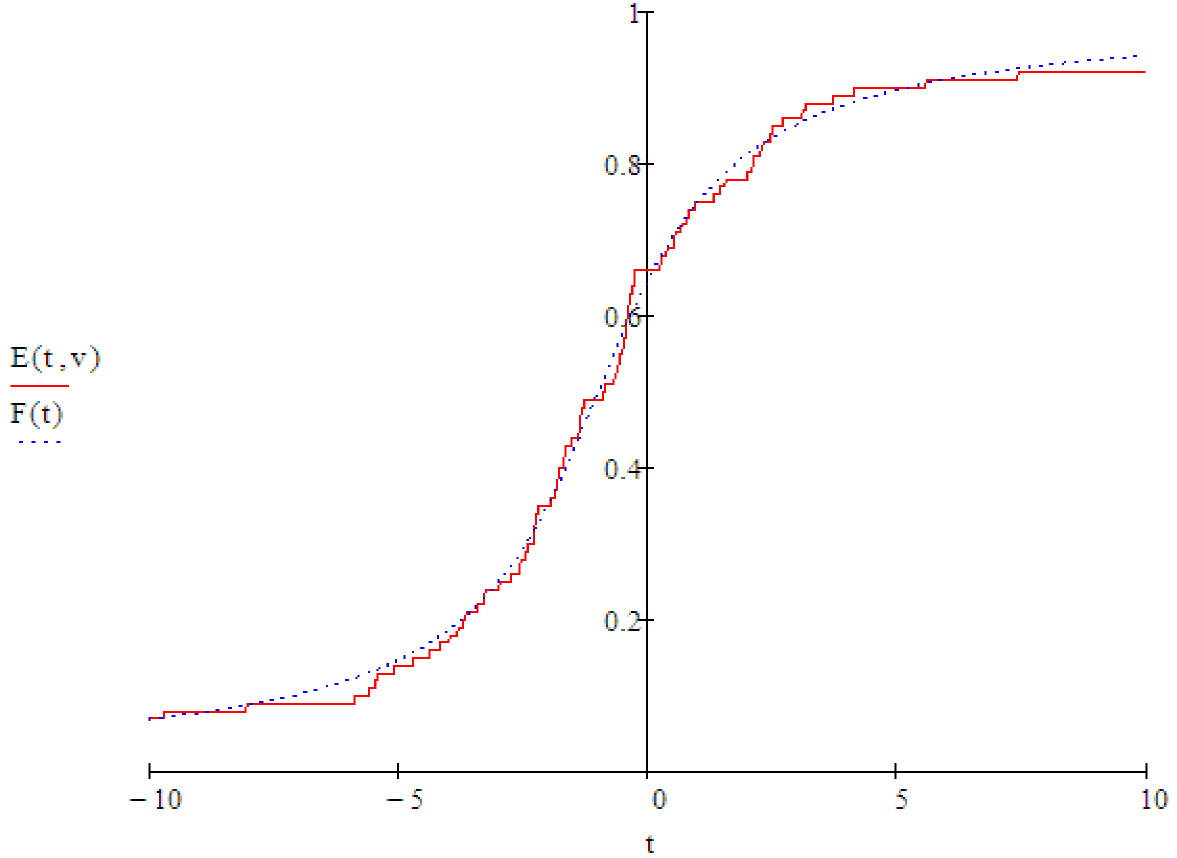
$$R2 = \begin{pmatrix} -0.515 \\ 0.408 \\ 0.001 \end{pmatrix}$$

```
R2 :=  | x1 ← −1
       | x2 ← 0
       | fd ← df(x1)
       | Δ ← x1 − x2
       | while |Δ| > ε
       |    | Δf ← f(x1) − f(x2)
       |    | xm ← x1 − Δ / (2·(1 − Δf/(fd·Δ)))
       |    | if Δf > 0
       |    |    | x1 ← x2
       |    |    | fd ← df(x1)
       |    | x2 ← xm
       |    | Δ ← x1 − x2
       | (xm, f(xm), |Δ|)
```

$$xm \leftarrow x1 - \frac{\Delta}{2\left(1 - \frac{\Delta f}{fd\cdot\Delta}\right)}$$

Laboratorinis darbas 2.

- Empiriniai duomenys $\{x_k\}$ (pvz., grąžos)
- Hipotetinis pasiskirstymo dėsnis $F(x,\Theta)$.
  - Θ – dėsnio parametrų vektorius.
-



Laboratorinis darbas 2.

- Jei imtyje $\{x_k\}$ yra n elementų ir $N(x)$ – imties elementų, mažesnių už x, skaičius, tai **empirinė pasiskirstymo funkcija**

$$\hat{F}_n(x) = \frac{1}{n} N(x) = \frac{1}{n} \sum_{k=1}^{n} H(x - x_k)$$

- Empirinę pasiskirstymo funkciją galima skaičiuoti pozicinių statistikų pagalba

$$\hat{F}_n(x) = \begin{cases} 0, & x \leqslant x_{(1)}, \\ \dfrac{k}{n}, & x_{(k)} < x \leqslant x_{(k+1)}, \\ 1, & x > x_{(n)}. \end{cases}$$

Laboratorinis darbas 2.

- Du algoritmai empirinei pasiskirstymo funkcijai skaičiuoti

```
epf(t, U) := | U ← sort(U)
             | 0 if t ≤ U_0
             | 1 if t > U_{rows(U)-1}
             | otherwise
             |   | for k ∈ 0 .. rows(U) − 2
             |   |    z ← k / rows(U)  if U_k < t ≤ U_{k+1}
             |   | z
```

```
E(t, X) := | 0 if t ≤ X_0
           | 1 if t > X_{rows(X)-1}
           | otherwise
           |   | nmin ← 0
           |   | nmax ← rows(X) − 1
           |   | while nmax − nmin > 1
           |   |   | n ← round((nmin + nmax) / 2)
           |   |   | nmax ← n if t ≤ X_n
           |   |   | nmin ← n if t > X_n
           |   | nmax / rows(X)
```

Maksimalaus tikėtinumo metodas

- Tikėtinumo funkcija

$$L(\Theta) = \prod_{k=1}^{n} f(x_k, \Theta),$$

kur $f(x,\Theta)$ – hipotetinio dėsnio tankio funkcija.

- Logaritminė suvidurkinta tikėtinumo funkcija

$$W(\Theta) = \frac{1}{n} \sum_{k=1}^{n} \ln f(x_k, \Theta).$$

- Maksimalaus tikėtinumo įverčiai

$$\hat{\Theta} = arg\, \min_{\Theta} \left[-W(\Theta) \right].$$

Daugelio kintamųjų funkcijų optimizavimas be apribojimų

- Nagrinėsime daugelio kintamųjų funkcijų f: $R^n \to R$ minimizavimo be apribojimų uždavinius:

$$\min_{X \in R^n} f(X),$$

kur X yra n kintamųjų vektorius.

- Praktiniai uždaviniai paprastai turi apribotą leistinąją sritį. Tačiau uždavinių su apribojimais sprendimo metodai arba grindžiami optimizavimo be apribojimo metodų idėjomis, arba tie metodai tiesiogiai naudojami.

Daugelio kintamųjų funkcijų optimizavimas be apribojimų

- Ekstremumus galima ieškoti, keičiant minimizavimo uždavinį sistemos grad $f(X)=0$ sprendimu. Taip randami funkcijos stacionarieji taškai kurie gali būti *min*, *max* arba balno taškais.

- Gerai, jei sistemą pavyksta išspręsti analiziškai, tačiau dažnai šią sistemą išspręsti nė kiek ne lengviau nei tiesioginį minimizavimo uždavinį (pvz., Himelblau funkcija)

Daugelio kintamųjų funkcijų optimizavimas be apribojimų

- Taigi, minimizavimo uždavinį ne visada pavyksta pakeisti lygčių sistemos grad $f(X)$= 0 sprendimu, todėl sukurta įvairių minimizavimo be apribojimų metodų.

- Nagrinėsime šiuos optimizavinmo metodus:

1. Negradientinius metodus, grindžiamus tik TF reikšmių skaičiavimu.
2. Nusileidimo metodus, kuriuose naudojamos išvestinės.
   - Gradientiniai metodai, kuriuose naudojamos TF pirmosios eilės išvestinės;
   - Antrosios eilės metodai, kuriuose naudojamos TF pirmųjų ir antrųjų išvestinių reikšmės.

- Idealaus metodo nėra. Todėl pagal situaciją ir galimybes pasirenkamas vienas iš galimų metodų, viena iš tinkamų turimų programų.

Negradientiniai metodai

- Pagrįsti šiomis prielaidomis:
  - $f(X)$ tolydi;
  - išvestinės gali ir neegzistuoti, nei jų reikšmės skaičiavimuose nenaudojamos.

- Šiais metodais randami lokalieji optimumai.

- Nagrinėsime metodus
  - Simplekso metodas (nepainioti su tiesinio programavimo simplekso metodu!);
  - Koordinatinės paieškos metodas;
  - Huko-Dživso metodas;
  - Pauelo jungtinių krypčių metodas.

Simplekso metodas

- *n*-matėje erdvėje konstruojamas simpleksas, kuris, vykstant skaičiavimams, „vartomas“. Simpleksas yra trikampio apibendrinimas – tai *n*-matis iškilas daugiakampis su *n*+1 tiesiškai nepriklausoma viršune.

- Simpleksas vadinamas taisyklingu, jei viršūnės vienodai nutolusios viena nuo kitos.

- Vykdant algoritmo skaičiavimus, konstruojamas naujas simpleksas – viena iš viršunių perkeliama į naują vietą. Naujoji viršnė yra ant tiesės, nubrėžtos per likusių viršunių svorio centrą. Ji tiek pat nutolusi nuo svorio centro, kaip iki perkelimo, tik į priešingą pusę.

Simplekso metodo algoritmas

1. Sudaromas pradinis simpleksas. Jo viršunėse apskaičiuojamos $f(X)$ reikšmės. Simplekso dydį reguliuoja parametas α.
2. Randama didžiausia $f(X)$ reikšmė ir ją atitinkanti viršūnė.
3. Ši viršūnė perstumiama tiese, nubrėžta per likulių viršūnių svorio centrą, ir tampa nauja simplekso viršūne. Taip kiekvienoje iteracijoje panaikinama simplekso viršūnė su didžiausia funkcijos reikšme.
4. 2-3 žingsniai kartojami tol, kol simpleksas neuždengs minimumo taško arba neprasidės ciklinis šokinėjimas tarp dviejų ar daugiau simpleksų.

Simplekso metodo algoritmas

- Žingsnių kartojimo ciklas gali būti nutraukiamas pagal kurią nors iš trijų sąlygų:

1) Naujoje viršūnėje funkcijos reikšmė didesnė nei kitose viršūnėse. Taip gali atsitikti kai minimumo taškas uždengtas simpleksu arba kai simpleksas yra siauro griovio dugne. Tada toliau žengiama iš viršūnės, kurio pagal funkcijos dydį yra antra.
2) Ciklinis jūdėjimas. Jei kuri nors simplekso viršunė neišmetama per daugiau nei M iteracijų, $M = [1{,}65\, n + 0{,}05\, n^2]$, būtina sumažinti simplekso dydį. Naujo simplekso bazinis taškas parenkamas ten, kur senajame buvo mažiausia $f(X)$ reikšmė. Kiekviena naujojo simplekso kraštinė sutrumpinama: $\alpha_1 = K\, \alpha_0$, $K < 1$.
3) Parenkamos sustojimo sąlygos. Skaičiavimai baigiami, kai:
   - simpleksas tampa mažas;
   - funkcijos reikšmės simplekso viršūnėse tampa panašios;
   - viršijamas leistinas funkcijos reikšmių skaičiavimų kiekis.

- Pagal šį algoritmą atliekami dviejų tipų skaičiavimai:

1. Taisyklingojo simplekso sudarymas pagal duotą bazinį tašką ir mastelį α.
2. Naujos simplekso viršunės skaičiavimai.

Simplekso sudarymas

- Duotas bazinis taškas $X_0 = (x_{01}, \ldots, x_{0n})$ ir koeficientas α.
- Kitos taisyklingo simplekso viršūnės skaičiuojamos taip:

$$X_i = \begin{cases} x_{0j} + \dfrac{\sqrt{n+1}+n-1}{n\sqrt{2}}\alpha, & j \neq i \\ x_{0j} + \dfrac{\sqrt{n+1}-1}{n\sqrt{2}}\alpha, & j = i \end{cases}, \qquad i, j = 1, \ldots, n\ ;$$

- Parinkus α=1, simplekso kraštinė būtų lygi 1.
- α parenkamas eksperimentiškai.

Naujos viršūnės skaičiavimas

- Jei atmetame X_j , n viršūnių svorio centras lygus:

$$X_c = \frac{1}{n} \sum_{\substack{i=0 \\ i \neq j}}^{n} X_i .$$

- Per taškus X_j ir X_c nubrėžtos tiesės lygtis:

$$X = X_j + \lambda \left(X_c - X_j \right).$$

- Taigi nauja simplekso viršūnė

$$X_{new} = - X_j + 2 X_c .$$

Simplekso algoritmo savybės

- Paprastas.
- Nedaug valdymo parametrų:
  - mastelio koeficientas α;
  - simplekso mažinimo daugiklis K;
  - skaičiavimo pabaigos parametrai.
- Algoritmas efektyvus net kai TF reikšmės skaičiuojamos su didelėmis paklaidomis.
- Reikia nedaug kompiuterinės atminties – tik $n(n+2)$ dydžio masyvo.

Simplekso algoritmo trūkumai

- Koeficientas α bendras visoms simplekso kraštinėms. Trūkumas šalinamas taip normuojant kintamuosius, kad suvienodėtų funkcijos kitimas visomis kryptimis.
- Lėtas, nes neįvertinami ankstesniai žingsniais gauti rezultatai.
- Gali labai sumažėti α. Pvz., kai simpleksas apeina «kalną», arba praeina siaurą griovį.

Nelderio ir Mido metodas [ND]

- Siekdami išvengti dalies simplekso metodo problemų, Nelderis ir Midas (1965 m.) pasiūlė simplekso metodo modifikaciją su netaisyklingu deformuojamu simpleksu – jis gali ir plėstis ir trauktis bet kuria kryptimi.
- Žr. R. Čiegis, V. Būda, Skaičiuojamoji matematika, V.,1997

Koordinatinės paieškos metodas

- Pradinis taškas $X_0 = (x_{01}, \ldots, x_{0n})$.
- Paieška vykdoma cikliškai keičiamomis kryptimis. Kryptis turi būti tiesiškai nepriklausomos. Paprastai naudojama erdvės R^n bazė iš ortogonalių vektorių:

$$
\begin{aligned}
e_1 &= (1;\ 0;\ 0;\ldots;\ 0;\ 0),\\
e_2 &= (0;\ 1;\ 0;\ldots;\ 0;\ 0),\\
&\ldots\\
e_n &= (0;\ 0;\ 0;\ldots;\ 0;\ 1).
\end{aligned}
$$

Koordinatinės paieškos metodo algoritmas

1. $k = 0$.
2. $i = 1$.
3. $X_{k+1} = X_k + \alpha\, e_i$, k = k + 1. Čia α yra daugiklis parenkamas išsprendus vienmačio minimizavimo uždavinį: $\alpha = \underset{\gamma}{arg\, min}\, f\left(X_k + \gamma\, e_i\right)$.
4. $i = i+1$; jei $i \leq n$, einama į 3 žingsnį.
5. Jei algoritmo sustojimo sąlygos netenkinamos, einama į 2 žingsnį.

- Algoritmo sustojimo sąlygos:
  - atliktas fiksuotas iteracijų skaičius (iteracija — tai 2-4 žingsnių skaičiavimai, kai $i=1,...,n$);
  - gretimų iteracijų TF reikšmės skiriasi mažiau nei tam tikras tikslumas ε.

- Algoritmo trūkumas — cikliškai kartojamos vienmatės paieškos: daug TF skaičiavimų.

Huko-Dživso paieškos metodas

- Metodo idėja: paieška vykdoma kryptimi $d_i = X_i - X_{i-1}$.
- Tai labai paspartina konvergavimą lyginant su koordinatine paieška, nes taip sudarytos kryptys *d* turi tendenciją išsidėstyti išilgai TF paviršiaus griovio. Svarbiausia – čia jau įvertinami ankstesnėse iteracijose gauti rezultatai.
- Metodas susideda iš dviejų dalių:
  - tiriamosios paieškos su cikliniu kintamųjų keitimu
  - greitėjančios paieškos pagal pavyzdį.

Tiriamoji paieška

- Tiriamoji paieška turi nustatyti lokalias TF savybes ir rasti krypti išilgai griovio.
- Tiriamosios paieškos taško X_0 aplinkoje algoritmas:

1. Startuojame iš bazinio taško $X_0 = (x_{01}, \dots, x_{0n})$ ir formuojame naują tašką $X_1 = (x_{11}, \dots, x_{1n})$. Priskiriame $X_1 = X_0$. Naudosime pagalbinį tašką X_p.
2. Priskiriame $X_p = X_1$. Keičiame pirmąją taško X_p koordinatę: $x_{p1} = x_{11} + \Delta_1$ ir skaičiuojame $f(X_p)$. Jei $f(X_p) < f(X_1)$, tai $X_1 = X_p$. Priešingu atvėju $x_{p1} = x_{11} - \Delta_1$ ir skaičiuojame $f(X_p)$. Jei $f(X_p) < f(X_1)$, tai $X_1 = X_p$.
3. Tą patį atliekame su antrąja, trečiąja, …, n-ąja taško X_p koordinate.

- Jei tiriamosios paieškos algoritmu gautas taškas X_1 nesutampa su tašku X_0, tai X_1 tampa nauju baziniu tašku.

Paieška pagal pavyzdį

- Žymėjimai:
  - X_k – einamasis bazinis taškas,
  - $X_{k\text{-}1}$ – ankstesnis bazinis taškas,
  - P_{k+1} – taškas, gautas judant pagal pavyzdį,
  - X_{k+1} – naujas bazinis taškas.

- Žengiama iš taško X_k išilgai tiesėsm jungiančios šį tašką su $X_{k\text{-}1}$:

$$P_{k+1} = X_k + \left(X_k - X_{k-1} \right).$$

Huko – Dživso metodo algoritmas

1. Pradžioje nustatome algoritmo parametrus:

- pradinį bazinį tašką X_0 ir prieaugius Δ_i, i=1, …, n;
- žingsnio mažinimo koeficientą $\alpha > 0$ ir paieškos pabaigos parametrą $\varepsilon > 0$.

2. Atliekame tiriamąją paiešką iš bazinio taško.
3. Jei tiriamoji paieška buvo sėkminga – eiti į 5 žingsnį, ne – į 4 žingsnį.
4. Tikriname pabaigos sąlygą – ar $\|\Delta\| < \varepsilon$. Jei ne – sumažinti prieaugius $\Delta_i = \Delta_i / \alpha$, ir eiti į 2 žingsnį.
5. Atliekame paiešką pagal pavyzdį: $P_{k+1} = X_k + \left(X_k - X_{k-1}\right)$.
6. Atliekame tiriamąją paiešką, kaip bazinį tašką naudodami P_{k+1} .Tiriamosios paieškos rezultatą pažymėkime X_{k+1}.
7. Jei $f(X_{k+1}) < f(X_k)$, tai $X_{k-1} = X_k$, $X_k = X_{k+1}$ ir eiti į 5 žingsnį. Priešingai - eiti į 4 žingsnį su baziniu tašku X_k.

Pauelo jungtinių krypčių metodas

- Metodas sukurtas su prielaida, kaf TF yra kvadratinė.
- Gali būti taikomas bet kokios iškilos tolydžios funkcijos optimizavimui, nes tokios funkcijos minimumo taške funkciją to taško aplinkoje galima gana tiksliai aproksimuoti kvadratine funkcija.

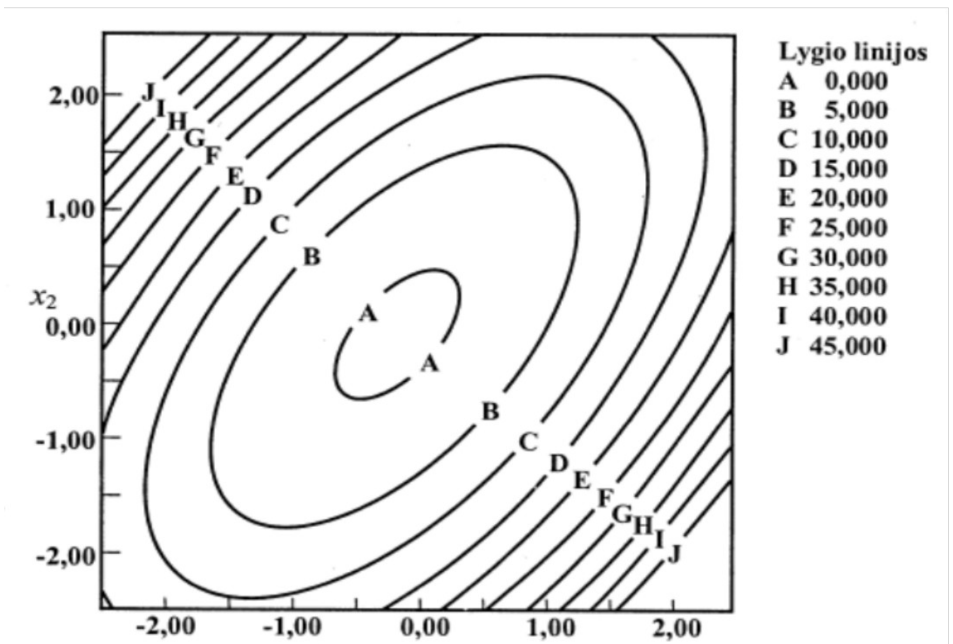
- Kvadratinė funkcija vektorinėmis operacijomis užrašoma taip:

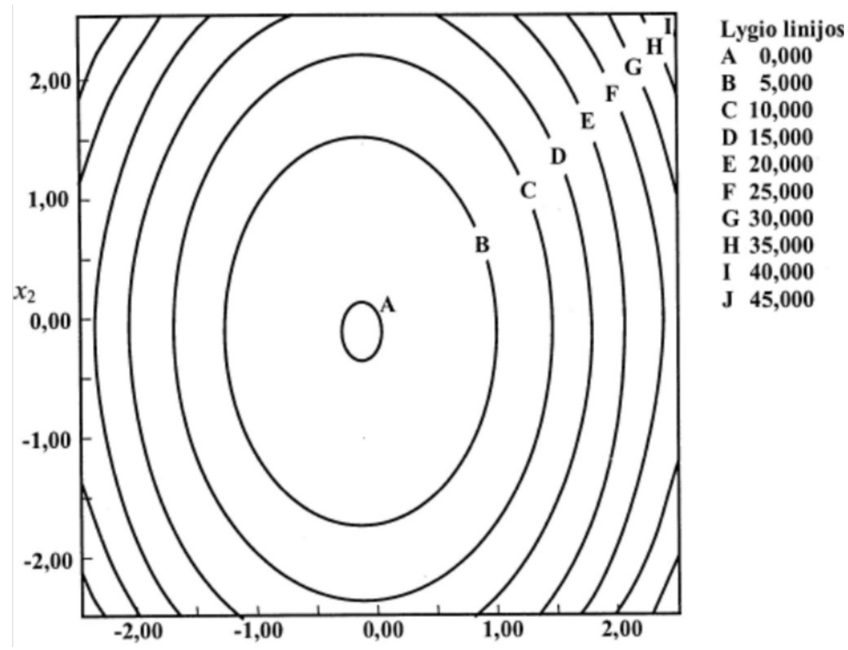
$$g(X) = a + bX^T + \frac{1}{2} XCX^T$$

$$g(X) = 8 + (1\,;\ 3)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \frac{1}{2}(x_1;\ x_2)\begin{pmatrix} 2 & -5 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Kvadratinė funkcija

- Atliekant koordinatinę paiešką, reikėtų žengti daug žingsnių «pjūklu».
- Sumažinti žingsnių skaičių galima pasukus koordinačių ašis. Šiuo atveju pradėję paiešką iš bet kurio taško, funkcijos minimumą rasime tik viena koordinatinės paieškos iteracija.





Jungtinės kryptys

- Tarkime, C yra kvadratinė simetrinė n eilės matrica.
- Kryptys s_1, s_2, ..., s_r, $r \le n$, vadinamos C jungtinėmis, jei jos yra tiesiškai nepriklausomos ir
$$s_i C s_j^T = 0, \quad \forall\, i \neq j .$$

- Nagrinėkime kvadratinę formą $Q(X) = XCX^T$. Ieškosime transformacijos R, kuri kvadratinę formą transformuotų į diagonaliąją: $X^T = RZ^T$, čia T – matrica, Z – naujų kintamųjų vektorius.
- Taigi, $Q(X) = ZR^TCRZ^T = ZDZ^T$. Čia D – diagonalinė matrica.
- Turime
$$X^T = R\,Z^T = \sum_{j=1}^{n} R^{\langle j \rangle} z_j \quad ,$$

čia $R^{<j>}$ - matricos R j-asis stulpelis.

Jungtinės kryptys

- Kitaip tariant, gauname naują koordinačių sistemą $R^{<j>}$, $j = 1, ..., n$, sudarytą iš C jungtinių krypčių, kurios matricą C transformuoja į diagonalinę:

$$R^T C R = D, \qquad \left(R^{\langle j \rangle}\right)^T C R^{\langle k \rangle} = \begin{cases} d_{kk}, & j = k, \\ 0, & j \neq k. \end{cases}$$

- Taigi gauname

$$g(X) = a + bX^T + \frac{1}{2} XCX^T = a + b\sum_{j=1}^{n} R^{\langle j \rangle} z_j + \frac{1}{2}\sum_{j=1}^{n} d_{jj} z_j^2 =$$

$$= a + b\sum_{j=1}^{n} \left(b R^{\langle j \rangle} z_j + \frac{1}{2} d_{jj} z_j^2\right) = a + \sum_{j=1}^{n} G_j(z_j).$$

Jungtinės kryptys

- Funkcija $g(Z)$ priklauso nuo vieno kintamojo funkcijų sumos. Toks optimizavimo uždavinys suvedamas į n vieno kintamojo uždavibių sprendimą:

$$\min_{z_j}\left[G_j(z_j)\right], \quad j=1,\dots,n.$$

- **Išvada**: sudarę vektorių $R^{<j>}$, $j = 1, \dots, n$ sistemą, dar vadinama jungtinių krypčių sistema, kvadratinės funkcijos optimumą rasime per n vienmačių pasieškų kryptimis $R^{<j>}$.
- Kaip rasti ***D***? Jei C žinoma – problemos nėra. Tinka Gauso-Žordano metodas. Mūsų atveju C nėra žinoma. Tačiau vis tiek problemą spręsime naudodamiesi kvadratinės funkcijos savybėmis.

Lygiagretaus poerdvio savybė

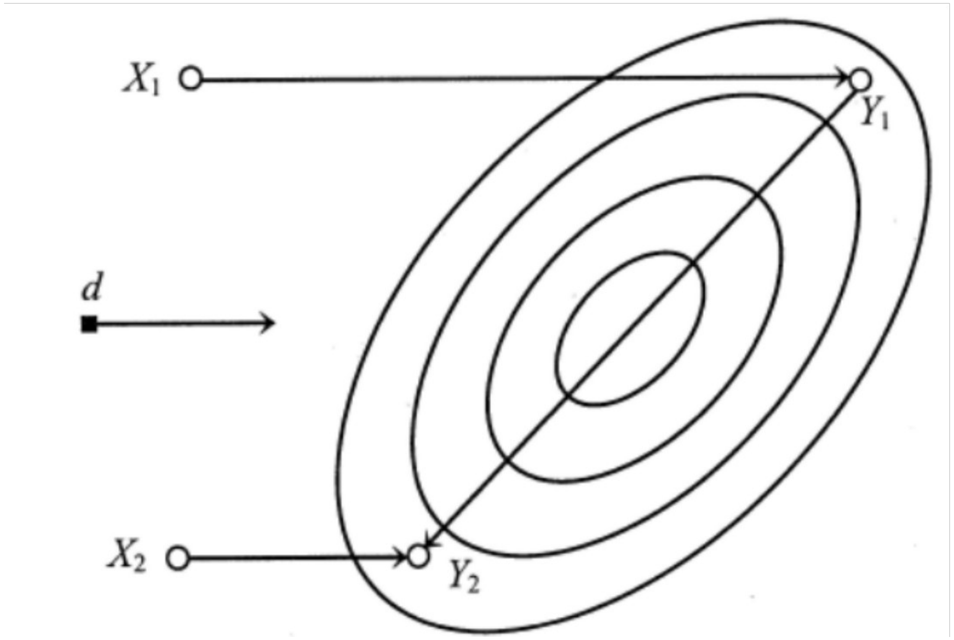
- Tarkime, turime
  - kvadratinę funkciją $g(X)$;
  - du taškus X_1 ir X_2;
  - skaliarą λ;
  - kryptį d.

$$g(X) = a + bX^T + \frac{1}{2} XCX^T$$

- Tada jei taškas Y_1 minimizuoja $g(X_1+\lambda d)$, o taškas Y_2 minimizuoja $g(X_2+\lambda d)$, tai kryptys $(Y_2 - Y_1)$ ir d yra jungtinės matricos C atžvilgiu.

Jungtinės kryptys plokštumoje

- Dviem vienmatėmis paieškomis randame Y_1 ir Y_2 . Minimumo tašką rasime judėdami kryptimi Y_2-Y_1 .



- Taigi dviejų kitamųjų atveju po dviejų vienmačių paieškų galima sudaryti jungtinių krypčių sistemą, o trečiąja paieška rasti kvadratinės funkcijos minimumą.

Lygiagretaus poerdvio savybė

- Grįžkime prie bendro pobūdžio kvadratinės funkcijos

$$g(X) = a + bX^T + \frac{1}{2} XCX^T$$

- Tiesės iš taško X_1 kryptimi d lygtis yra $X = X_1 + \lambda\, d$.
- Funkcijos $g(X)$ minimumas iš taško X_1 kryptimi d randamas ieškant tokio λ^*, kad $g'_\lambda(X) = 0$.

$$\frac{\partial g}{\partial \lambda} = \frac{\partial g}{\partial X} \frac{\partial X}{\partial \lambda} = (b + XC)\, d^{T.}$$

- Tarkime, minimumas yra taške Y_1, todėl $(b + Y_1C)d^T = 0$.
- Tarkime, minimumas tiesėje $X = X_2 + \lambda\, d$ yra taške Y_2, todėl $(b + Y_2C)d^T = 0$. Turime $(Y_2 - Y_1)\mathrm{C}d^T = 0$. Taigi kryptys $(Y_2 - Y_1)$ ir d yra jungtinės.

Lygiagretaus poerdvio savybė

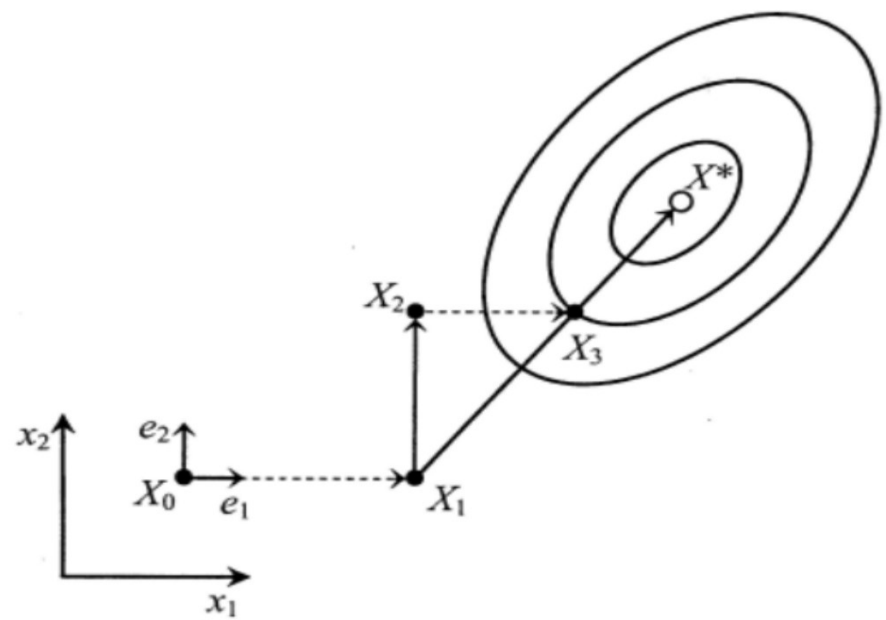
- Matome, kad jungtinės kryptys nustatomos pagal du taškus ir kryptį. Tai nepatogu – geriau turėti vieną tašką ir einant iš jo konstruoti jungtines kryptys.
- Tegul tai taškas X_0. Kaip pagrindą imsime koordinatinius vekorius

$$
\begin{gathered}
e_1 = (1;\ 0;\ 0;\ldots;\ 0;\ 0),\\
e_2 = (0;\ 1;\ 0;\ldots;\ 0;\ 0),\\
\ldots\\
e_n = (0;\ 0;\ 0;\ldots;\ 0;\ 1).
\end{gathered}
$$

Tai ne jungtinės kryptys!

Lygiagretaus poerdvio savybė

- Nagrinėsime atvejį , kai $n = 2$.
- Tegu $X_1 = X_0 + \lambda_0\, e_1$. Čia λ_0 – koeficientas, minimizuojantis $f(X_0 + \lambda_0\, e_1)$.
- Tegu $X_2 = X_1 + \lambda_1\, e_2$. Čia λ_1 – koeficientas, minimizuojantis $f(X_1 + \lambda\, e_2)$.
- Tegu $X_3 = X_2 + \lambda_2\, e_1$. Čia λ_2 – koeficientas, minimizuojantis $f(X_2 + \lambda\, e_1)$.
- Gavome: kryptys $(X_3 - X_1)$ ir e_1 yra jungtinės, nes kryptimi e_1 atlikome dvi paieškas: iš X_0 į X_1 ir iš X_2 į X_3 .



- Jei funkcija yra kvadratinė, tai atlikę paiešką kryptimis e_1 ir $X_3 - X_2$ gauname funkcijos minimumo tašką X^*.

Pauelo jungtinių krypčių metodo algoritmas

1. Pasirenkame pradinį tašką X_0 ir tiesiškai nepriklausomų krypčių sistemą $s_1 = e_1$, ... , $s_n = e_n$. Sudarome sąrašą krypčių, kuriomis minimizuosime: s_n, s_1 , ... , s_n .

2. Minimizuojame $f(X)$ iš eilės visomis $n+1$ kryptimis.

3. Randame naują jungtinę kryptį.

4. Iš krypčių sąrašo išmetame dvi pirmasias kryptis, sarašo priekyje ir gale įrašome 3 žingsniu gautą jungtinę kryptį. (*Taip minimumai kure bus nustatomi 2 žingsniu einant sąrašo pirmąja ir paskutiniąja kryptimi, bus gaunami nuosekliai minimizuojant tomis pačiomis jungtinėmis kryptimis. Todėl tuos minimumo taškus jungianti tiesė bus nauja jungtinė kryptis*). Einame į 2 žingsnį.

- *Būtina patikrinti gautų krypčių tiesinį nepriklausomumą (jei TF nekvadratinė). Jei nauja kryptis yra tiesiškai priklausoma nuo saraše esančių krypčių, ji į sąrašą neįtraukiama.*

Gradientiniai nusileidimo metodai

- Ieškant funkcijos minimumo nusileidimo metodais generuojama seka taškų $\{X_k\}$, kurių funkcijos reikšmės monotoniškai mažėja. Taškas X_{k+1} randamas iš taško X_k leidžiantis funkcijos mažėjimo kryptimi.

- Idėja. Funkcijos gradiento vektorius yra nukreiptas greičiausio funkcijos augimo kryptimi; antigradientas – nukreiptas greičiausio funkcijos mažėjimo kryptimi;

Privalumai

- Nusileidmo metodų iteracinė schema labai paprasta ir akivaizdi:

$$X_{k+1} = X_k + \alpha_k S_k,$$

čia

- S_k – nusileidimo krypties vektorius k-toje iteracijoje;
- $\alpha_k > 0$ – žingsnio daugiklis; jei krypties vektorius normuotas, α_k yra žingsnio ilgis.

Trūkumai

- Gradientas teikia nedaug informacijos apie funkcijos savybes, nes taikoma tik tiesinė funkcijos aproksimacija taško aplinkoje.

- Jei funkcijos paviršius turi griovį, algoritmo generuojama seka gali šokinėti nuo vieno griovio krašto ant kito, o funkcijos reikšmės – mažėti labai lėtai.

- Naudojamos išvestinės, kurių išraiškas galima gauti ne visada (pvz., kai funkcijos reikšmės gaunamos modeliavimo būdu).

- *Bandant išvengti šių trūkumų buvo pasiūlyta sudėtingesnių metodų (Niutono ir kintamos metrikos), kuriuose naudojamos antrosios eilės išbestinės arba jų aproksimacijos. Šiuose metoduose nusileidimo krypčių formulės grindžiamos TF aproksimavimu kvadratine funkcija. Tai didina metodo sudėtingumą, tačiau tokie algoritmai konverguoja žymiai greičiau.*

Išvestinių skaitinės formulės

- Dažnai išvestinių reikšmės gaunamos skaitiniais metodais, pvz., paprastomis skirtuminėsmis formulėmis:

$$\frac{\partial f(X)}{\partial x_i} \approx \frac{f(x_1, \dots, x_i+\Delta, \dots, x_n) - f(X)}{\Delta}$$

$$\frac{\partial f(X)}{\partial x_i} \approx \frac{f(x_1, \dots, x_i+\Delta, \dots, x_n) - f(x_1, \dots, x_i-\Delta, \dots, x_n)}{2\Delta}$$

- Vienpusei formulei reikia *n* papildomų TF reikšmių kiekvienam gradientui. Minimumo aplinkoje, gradientui artėjant prie 0, vienpusė formulė nėra tiksli. Tenka naudoti dvipuses diferencijavimo formules, kur reikia 2*n* TF reikšmių.

Greičiausio nusileidimo metodas

- Metodo idėja: eiti nusileidimi kryptimi iki pirmo lokaliojo minimumo šia kryptimi.

$$X_{k+1} = X_k - \alpha_k \nabla f\left(X_k\right),$$

$$\alpha_k = \underset{\alpha \geqslant 0}{arg\,min}\, f\left(X_k - \alpha \nabla f\left(X_k\right)\right).$$

- Tiksliai minimizuoti pagal kryptį nereikia, nes tai reikalauja daug skaičiuojamųjų resursų, o kryptis tik išimtimiais atvejais tiksliai veda link TF minimumo. Kiek paėjus antigradiento kryptimi ir radus apytikrio minimumo tašką, šiame taške skaičiuojama nauja kryptis.

Greičiausio nusileidimo metodo konvergavimas

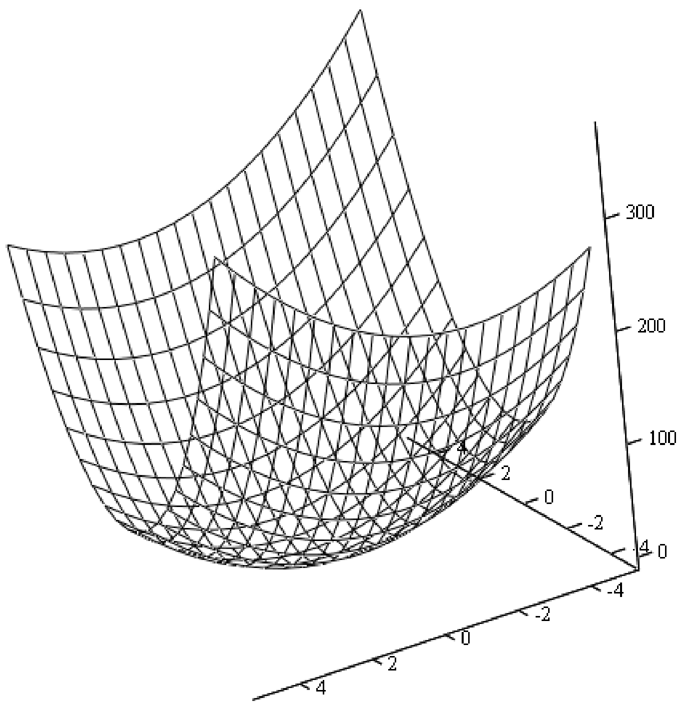
- **Teorema**. Tarkime, TF f: $R^n \rightarrow R$ yra diferencijuojama ir aprėžta iš apačios. Tada greičiausio nusileidimo metodu generuojama taškų seka su monotoniškai mažėjančiomis TF reikšmėmis ir gradiento norma artėja į nulį.

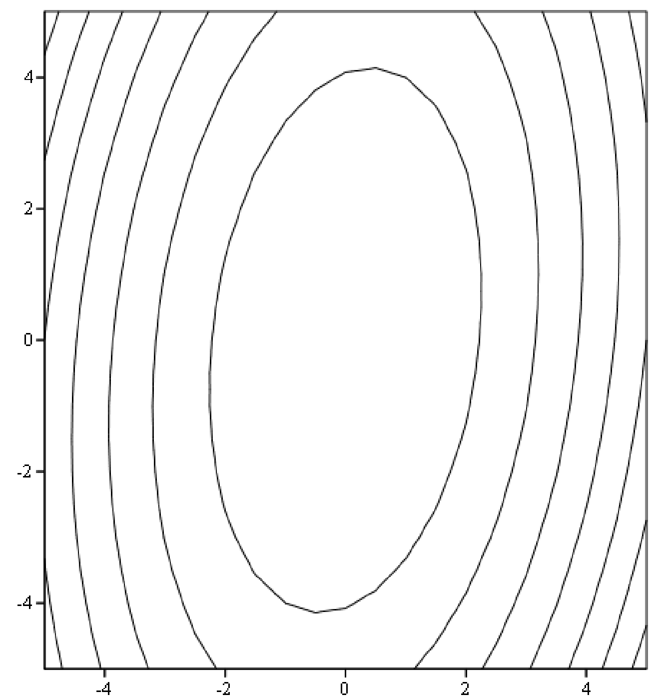
- *Griečiausio nusileidimo metodas yra paprastas ir stabilus. Bet jei TF paviršius turi griovį, tikslinga taikyti jungtinių krypčių metodą ar metodus su antromis išvestinėmis. Jie taikytini ir optimumo aplinkoje, nes joje TF gerai aproksimuojama kvadratine funkcija.*

$$f(X) = 10x_1^2 - 2x_1x_2 + 3x_2^2$$

Greičiausio nusileidimo metodas

- $$f(X) = 10x_1^2 - 2x_1x_2 + 3x_2^2$$





Niutono metodas

- Niutono metode panaudojama TF aproksimacija antros eilės Teiloro daugianariu:

$$f(X) \approx f(X_k) + \nabla f(X_k)(X - X_k)^T + \frac{1}{2}(X - X_k)H(X_k)(X - X_k)^T.$$

- Pažymėkime $\Delta X_k = X_{k+1} - X_k$. Funkcijos aproksimacijoje X pakeitę į X_{k+1}, gauname:

$$g(X_{k+1}) = f(X_k) + \nabla f(X_k)(\Delta X_k)^T + \frac{1}{2}\Delta X_k H(X_k)(\Delta X_k)^T.$$

Tai kvadratinė funkcija. Ieškosime tokio ΔX_k , kad X_{k+1} būtų funkcijos $g(X_{k+1})$ stacionarusis taškas.

Niutono metodas

- Diferencijuojame $g(X_{k+1})$ pagal vektoriaus ΔX_k komponentes ir išvestines prilyginame nuliui:

$$\nabla f(X_k) + \Delta X_k H(X_k) = 0.$$

- Išsprendę gauname kvadratinei funkcijai optimalų žingsnį:

$$\Delta X_k = -\nabla f(X_k) H^{-1}(X_k).$$

- Iš čia lengva užrašyti Niutono metodo iteracinę formulę:

$$X_{k+1} = X_k - \nabla f(X_k) H^{-1}(X_k).$$

Niutono-Rafsono metodas

- Jei TF būtų iškila kvadratinė funkcija, jos minimumą pasiektume vienu žingsniu. Bendru atveju atliekama daugiau žingsniu, todėl rastas optimalus žingsnio ilgus netenka prasmės. Tenka įvesti žingsnio ilgio parametrą:

$$X_{k+1} = X_k - \lambda_k \frac{\nabla f(X_k) H^{-1}(X_k)}{\left\| \nabla f(X_k) H^{-1}(X_k) \right\|}.$$

- Žingsnio ilgį žymesime

$$\alpha_k = \frac{\lambda_k}{\left\| \nabla f(X_k) H^{-1}(X_k) \right\|}.$$

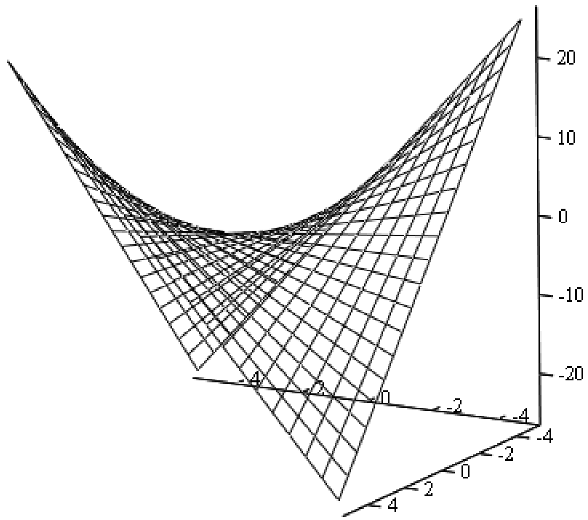
- Iš čia lengva užrašyti Niutono-Rafsono metodo iteracinę formulę:

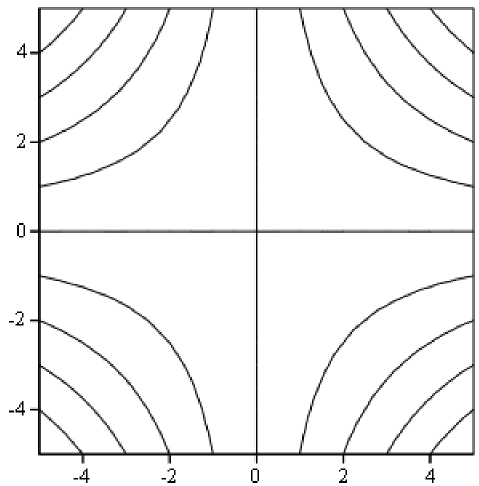
$$X_{k+1} = X_k - \alpha_k \nabla f(X_k) H^{-1}(X_k).$$

$$\alpha_k = \underset{\alpha \geqslant 0}{arg\, min}\, f(X_k - \alpha \nabla f(X_k) H^{-1}(X_k)).$$

Niutono metodo trūkumai

- Kai TF lokalioje srityje tiksliai aproksimuojama kvadratine funkcija ir kai matrica H^{-1} – teigiamai apibrėžta, galime tikėtis greito sprendimo.
- Šis reikalavimas gadina viską. Nes tik jei matrica, atvirkštinė Hesės matricai yra teigiamai apibrėžta, garantuojama, kad TF reikšmė sumažės.
- *Geometriškai tai paaiškinama paparastai. Kvadratinė aproksimacija geometriškai reiškia apvalią arba elipsinę idubą, turinčią lokalųjį minimumą tik kai matrica yra teigiamai apibrėžta. Kitais atvejais aproksimuojanti funkcija gali virsti balno paviršiumi, ir krypties vektorius užuot rodęs tiksliausią kryptį link minimumo taško, rodys kryptį į balno tašką.*





Modifikuotas Niutono metodas

- Minėtą trūkumą galima pašalinti, pakoregavus H^{-1} taip, kad ji visada būtų teigiamai apibrėžta.

- Jei invertuotant Hesės matricą, gaunamas požymis, kad atvirkštinė matrica nėra teigiamai apibrėžta, tada žengiamas paprastas greičiausio nusileidimo žingsnis.

- Kita Niutono metodo modifikacija:

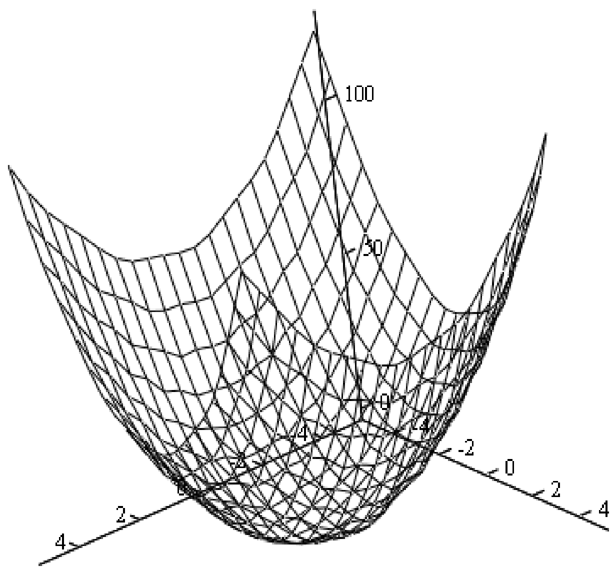
$$X_{k+1} = X_k - \alpha_k \nabla f(X_k) H^{-1}(X_0),$$

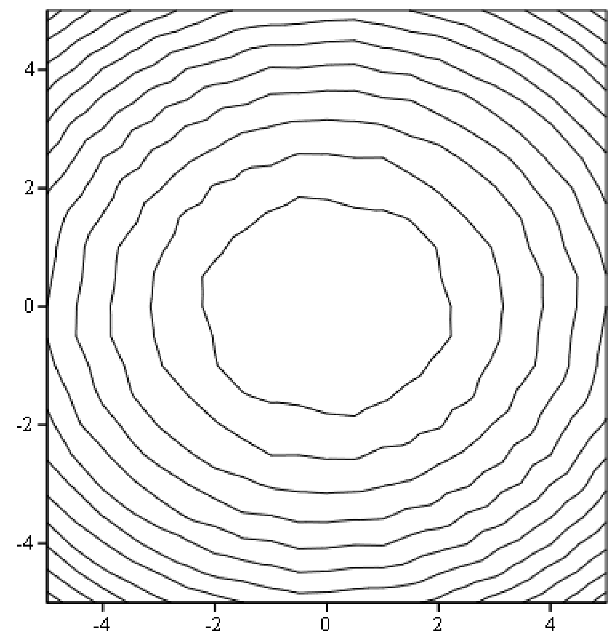
$$\alpha_k = \underset{\alpha \geqslant 0}{arg\,min}\, f\left(X_k - \alpha \nabla f(X_k) H^{-1}(X_0)\right).$$

Atvirkštinė Hesės matrica skaičiuojama tik vieną kartą ir naudojama visose iteracijose. Ši modifikacija labai sumažina skaičiavimų apimtį, tačiau iteracinis procesas konverguoja daug lečiau nei Niutono-Rafsono metodo iteracinis procesas.

Niutono metodas

$$f(x, y) := 2 \cdot x^2 + 3 \cdot y^2 + \sin(2 \cdot x \cdot y)$$





Niutono metodas

$$f(x) := 2\cdot(x_0)^2 + 3\cdot(x_1)^2 + \sin(2\cdot x_0\cdot x_1)$$

$\varepsilon := 0.01$ $\quad\quad \Delta := 10^{-3}$

$$R := \left| \begin{array}{l} x^{\langle 0\rangle} \leftarrow \begin{pmatrix} 1.5 \\ -1.1 \end{pmatrix} \\ k \leftarrow 0 \\ \text{while } \sqrt{x^{\langle k\rangle}\cdot x^{\langle k\rangle}} > \varepsilon \\ \quad \left| \begin{array}{l} x^{\langle k+1\rangle} \leftarrow x^{\langle k\rangle} - H\left(x^{\langle k\rangle}\right)^{-1}\cdot \text{grad}\left(x^{\langle k\rangle}\right) \\ k \leftarrow k+1 \end{array} \right. \\ x \end{array} \right.$$

$$R^T = \begin{pmatrix} 1.5 & -1.1 \\ -0.543 & 0.571 \\ -0.201 & 0.144 \\ -0.001 & 0.001 \end{pmatrix}$$

$$fmin := f\left(R^{\langle cols(R)-1\rangle}\right)$$

$$fmin = 0$$

Skaitinis diferencijavimas

$$\Delta := 10^{-3}$$

$$D := \begin{pmatrix} \Delta & 0 & \Delta & \Delta \\ 0 & \Delta & \Delta & -\Delta \end{pmatrix}$$

$$\mathrm{grad}(x) := \begin{pmatrix} \dfrac{f\left(x + D^{\langle 0 \rangle}\right) - f\left(x - D^{\langle 0 \rangle}\right)}{2\Delta} \\ \dfrac{f\left(x + D^{\langle 1 \rangle}\right) - f\left(x - D^{\langle 1 \rangle}\right)}{2\Delta} \end{pmatrix}$$

$$H(x) := \begin{pmatrix} \dfrac{f\left(x + D^{\langle 0 \rangle}\right) + f\left(x - D^{\langle 0 \rangle}\right) - 2 \cdot f(x)}{\Delta^2} & \dfrac{f\left(x + D^{\langle 2 \rangle}\right) - f\left(x + D^{\langle 3 \rangle}\right) - f\left(x - D^{\langle 3 \rangle}\right) + f\left(x - D^{\langle 2 \rangle}\right)}{4 \cdot \Delta^2} \\ \dfrac{f\left(x + D^{\langle 2 \rangle}\right) - f\left(x + D^{\langle 3 \rangle}\right) - f\left(x - D^{\langle 3 \rangle}\right) + f\left(x - D^{\langle 2 \rangle}\right)}{4 \cdot \Delta^2} & \dfrac{f\left(x + D^{\langle 1 \rangle}\right) + f\left(x - D^{\langle 1 \rangle}\right) - 2 \cdot f(x)}{\Delta^2} \end{pmatrix}$$

Kintamos metrikos metodai

- Kintamos metrikos metodose bandyta išvengti Niutono metodo trūkumų: nereikia kiekviename žingsnyje apversti Hesės matricos ir skaičiuoti antrųjų išvestinių.

- Pagrindinė metodų idėja: vietoj tikslios atvirkštinės Hesės matricos naudojama aproksimuojanti matrica

$$\eta(X_k) \approx H^{-1}(X_k),$$

kuriai gauti panaudotos tik pirmųjų išvestinių reikšmės.

- Naujo taško koordinatės skaičiuojamos pagal iteracinę formulę:

$$X_{k+1} = X_k - \alpha_k \nabla f(X_k)\eta(X_k).$$

$$\alpha_k = \mathop{arg\,min}_{\alpha \geqslant 0} f(X_k - \alpha \nabla f(X_k)\eta(X_k)).$$

Kintamos metrikos metodai

- $\eta(X_k)$ vadinama krypčių matrica. Jei ši matrica būtų vienetinė, tai turėtume greičiausio nusileidimo metodą. Niutono metodė ši matrica yra $H^{-1}(X_k)$.

- Prisiminkime gradiento Teiloro skleidinį:

$$\nabla f(X) = \nabla f(X_k) + (X - X_k) H(X_k).$$

- X pakeitę į X_{k+1}, ir padauginę abi lygties puses iš $H^{-1}(X_k)$, gaunome:

$$X_{k+1} - X_k = \left[\nabla f(X_{k+1}) - \nabla f(X_k)\right] H^{-1}(X_k) \quad \rightarrow \quad \Delta X = \Delta Y\, H^{-1}.$$

- Žiūrėkime į šią lygtį kaip į n tiesnių lygčių sistemą, turinčią n nežinomų parametrų, kuriais aproksimuojama H^{-1}, kai ΔX ir ΔY yra žinomi.

- *Ši aproksimavimo idėja, sudarant ir sprendžiant tokią tiesinių lygčių sistemą, konkrečiuose kintamos metrikos metoduose realizuojama įvairiai.*

Kintamos metrikos metodai

- Daugumoje metodų H^{-1} aproksimuojanti matrica η_{k+1} sudaroma iš ankstesnio žingsnio krypčių matricos $\eta_{k+1} = \eta_k + \Delta\eta_k$.

- Įvairūs kintamos metrikos metodai iš esmės skiriasi tik matricos $\Delta\eta_k$ apskaičiavimo būdu.

1. Broideno metodas
2. Devidono-Fletčerio-Pauelo (DFP) metodas
3. Broideno–Fletčerio–Goldfarbo–Šano (BFGS) metodas

Broideno metodas

- Broideno metode panaudota ši matricos $\Delta\eta_k$ perskaičiavimo formulė:

$$\Delta \eta_k = \frac{(\Delta X_k - \Delta Y_k \eta_k)^T (\Delta X_k - \Delta Y_k \eta_k)}{\Delta Y_k (\Delta X_k - \Delta Y_k \eta_k)^T}.$$

$$\Delta X_k = X_{k+1} - X_k, \quad \Delta Y_k = \nabla f(X_{k+1}) - \nabla f(X_k).$$

- Yra įrodyta, kad jei TF kvadratinė, tai Broideno metodu iš bet kokios pradinės teigiamai apibrėžtos krypčių matricos η_0 jau po n žingsnių gausime $\eta_n = H^{-1}$.

- Bendru atveju, vartojant šį metodą, galimos šitokios komplikacijos:

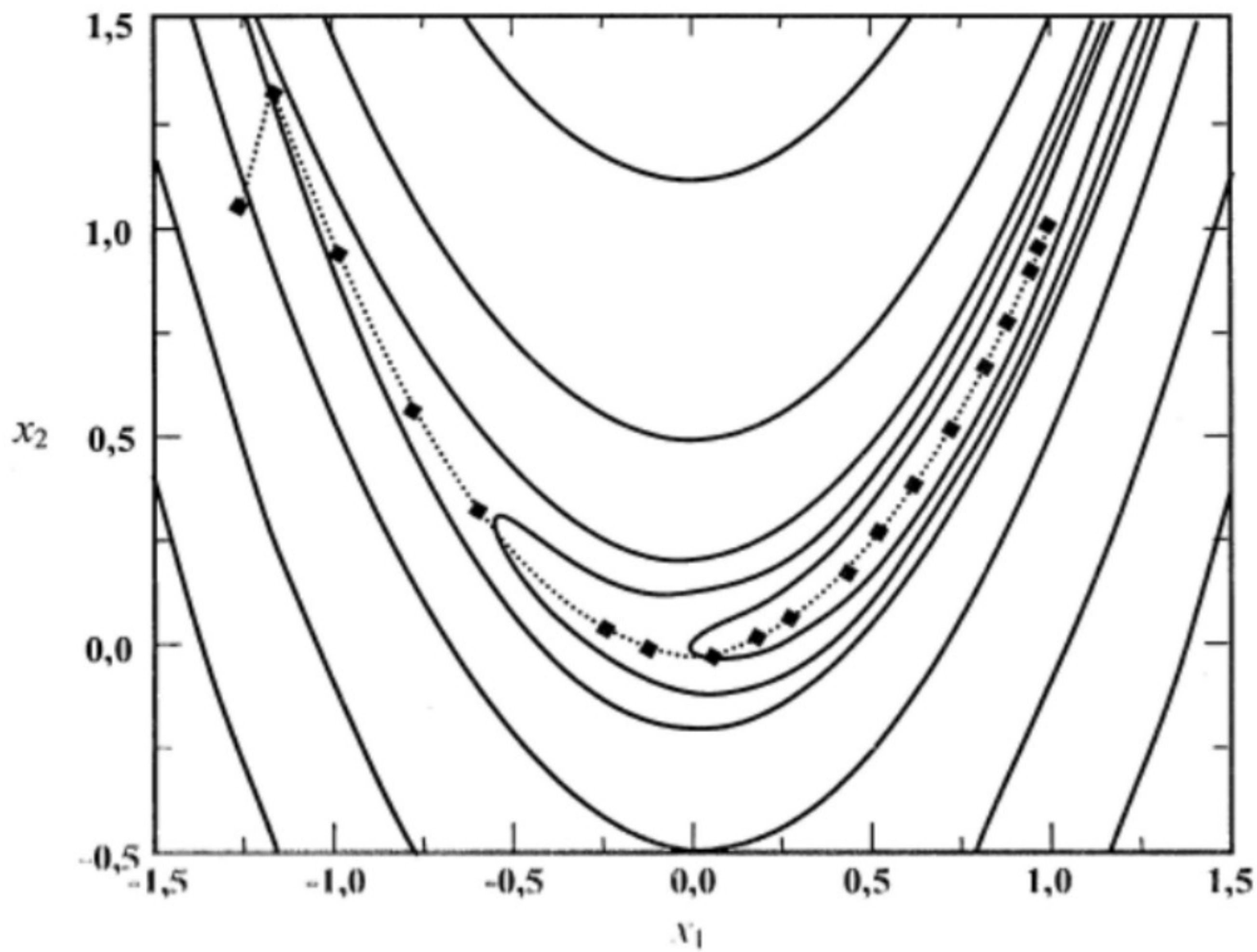
1. η_{k+1} gali nebūti teigiamai apibrėžta, tada tenka imtis jos koregavimo priemonių;
2. $\Delta\eta_{k+1}$ matricos narių reikšmės gali labai išaugti, susikaupus apvalinimo paklaidoms;
3. Atsitiktinai sutapus gretimų žingsnių ΔX_k kryptims, susidaro avarinės situacijos.

Broideno metodo algoritmas

1. Apibrėžti tikslumą ε, pradinį tašką X_0 ir pradinę teigiamai apibrėžtą krypčių matricą $\eta_k = \eta(X_0)$, $k = 0$. Pvz., $\eta_0 = E$.
2. Rasti paieškos kryptį $S_k = -\nabla f(X_k)\, \eta_k$.
3. Rasti žingsnio ilgį paieškos kryptimi: $\alpha_k = \underset{\alpha \geqslant 0}{arg\,min}\, f\left(X_k + \alpha S_k\right)$.
4. $\Delta X_k = \alpha_k S_k$; $X_{k+1} = X_k + \Delta X_k$.
5. Rasti ΔY_k : $\Delta Y_k = \nabla f\left(X_{k+1}\right) - \nabla f\left(X_k\right)$.
6. Jei $\|\Delta X_k\| > \varepsilon$ – iteracinis procesas tesiamas. Priešingu atveju: $X^* \approx X_{k+1}$.
7. Perskaičiuoti krypčių matricą $\eta_{k+1} = \eta_k + \Delta\eta_k$, padidinti iteracijų skaitiklį $k = k + 1$ ir grįžti į 2 žingsnį.

$$\Delta \eta_k = \frac{\left(\Delta X_k - \Delta Y_k \eta_k\right)^T \left(\Delta X_k - \Delta Y_k \eta_k\right)}{\Delta Y_k \left(\Delta X_k - \Delta Y_k \eta_k\right)^T}.$$

Kintamos metrikos metodu gauta traektorija Rozenbroko funkcijai



Devidono-Fletčerio-Pauelo metodas

- DFP metode (vienas iš populiariausiu, kartu su BFGS, kintamos metrikos metodas) vartojama matricos $\Delta\eta_k$ perskaičiavimo formulė:

$$\Delta \eta_k = \frac{(\Delta X_k)^T \Delta X_k}{\Delta X_k (\Delta Y_k)^T} - \frac{\eta_k (\Delta Y_k)^T \Delta Y_k (\eta_k)^T}{\Delta Y_k \eta_k (\Delta Y_k)^T}.$$

- Kaip ir taikydami Broideno metodą kvadratinei funkcijai, po n žingsnių turėsime $\eta_n = H^{-1}$.

Devidono-Fletčerio-Pauelo metodo algoritmas

1. Apibrėžti tikslumą ε, pradinį tašką X_0 ir pradinę teigiamai apibrėžtą krypčių matricą $\eta_k = \eta(X_0)$, $k = 0$.
2. Rasti paieškos kryptį $S_k = -\blacktriangledown f(X_k)\,\eta_k$.
3. Rasti žingsnio ilgį paieškos kryptimi: $\alpha_k = \underset{\alpha \geqslant 0}{arg\,min}\, f\,(X_k + \alpha\, S_k)$.
4. $\Delta X_k = \alpha_k S_k$; $X_{k+1} = X_k + \Delta X_k$.
5. Rasti ΔY_k : $\Delta\, Y_k = \nabla\, f\,(X_{k+1}) - \nabla\, f\,(X_k)$.
6. Jei $\|\Delta X_k\| > \varepsilon$ – iteracinis procesas tesiamas. Priešingu atveju: $X^* \approx X_{k+1}$.
7. Perskaičiuoti krypčių matricą $\eta_{k+1} = \eta_k + \Delta\eta_k$, padidinti iteracijų skaitiklį $k = k + 1$ ir grįžti į 2 žingsnį.

$$\Delta\,\eta_k = \frac{(\Delta\, X_k)^T \Delta\, X_k}{\Delta\, X_k (\Delta\, Y_k)^T} - \frac{\eta_k (\Delta\, Y_k)^T \Delta\, Y_k (\eta_k)^T}{\Delta\, Y_k\, \eta_k (\Delta\, Y_k)^T}.$$

Devidono-Fletčerio-Pauelo metodas

- Paprastai imama vienetinė pradinė matrica (nors gali būti ir bet kuri teigiamai apibrėžta matrica). Tuomet pirmasis žingsnis atliekamas greičiausio nusileidimo kryptimi. Tolimesniais žingsniais, matricai η_n artėjant prie H^{-1}, palaipsniui pereinama iš gradiento krypties į Niutono kryptį.

- Kadangi vis perskaičiuojant matricą η_n kaupiasi paklaidos, gali susidaryti skaičiavimo požiūriu neparankių situacijų. Norint jų išvengti, matrica periadiškai atnaujinama, t. y. padaroma vienetine.

- Taikant Niutono ir kintamos metrikos metodus labai aktualus klausimas, ar priimtina išvestines skaičiuoti ne analiziškai, o pagal skirtumines firmules. Tyrimai parodė, kad tai leistina, tačiau pageidautina, kad skaičiavimo tikslumas nebūtų mažas, nes priešingu atveju daugėja skaičiavimo žingsnių.

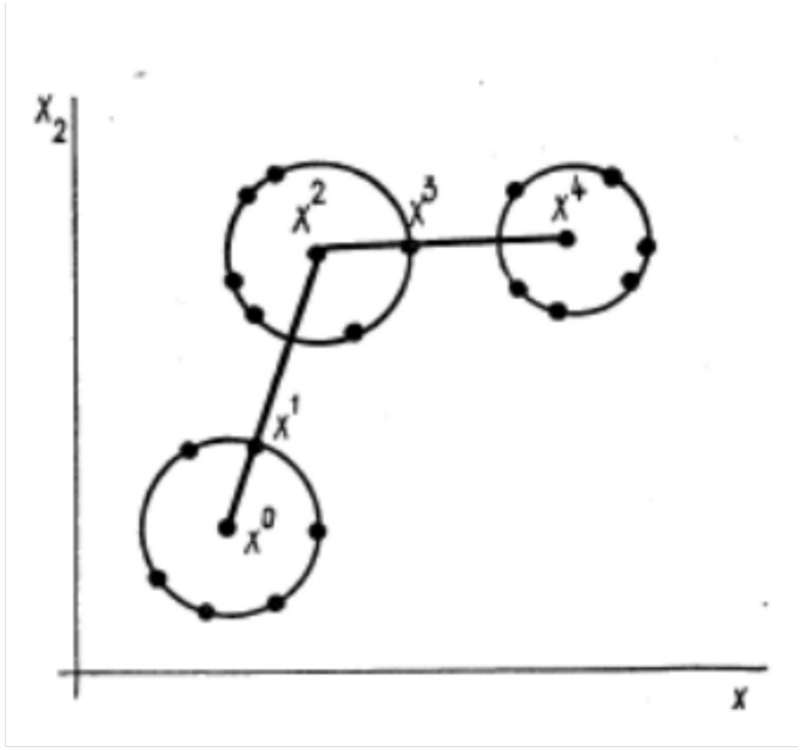
Broideno–Fletčerio–Goldfarbo–Šano metodo algoritmas

1. Apibrėžti tikslumą ε, pradinį tašką X_0 ir pradinę teigiamai apibrėžtą krypčių matricą $\eta_k = \eta(X_0)$, $k = 0$.
2. Rasti paieškos kryptį $S_k = -\nabla f(X_k)\,\eta_k$.
3. Rasti žingsnio ilgį paieškos kryptimi: $\alpha_k = \underset{\alpha \geqslant 0}{arg\,min}\, f\,(X_k + \alpha S_k)$.
4. $\Delta X_k = \alpha_k S_k$; $X_{k+1} = X_k + \Delta X_k$.
5. Rasti ΔY_k : $\Delta Y_k = \nabla f\,(X_{k+1}) - \nabla f\,(X_k)$.
6. Jei $\|\Delta X_k\| > \varepsilon$ – iteracinis procesas tesiamas. Priešingu atveju: $X^* \approx X_{k+1}$.
7. Perskaičiuoti krypčių matricą $\eta_{k+1} = \eta_k + \Delta\eta_k$, padidinti iteracijų skaitiklį $k = k + 1$ ir grįžti į 2 žingsnį.

$$\Delta\eta_k = \frac{((\Delta X_k)^T \Delta X_k)(\Delta Y_k (\Delta X_k)^T + \Delta Y_k \eta_k (\Delta Y_k)^T)}{(\Delta Y_k (\Delta X_k)^T)^2} - \frac{(\Delta X_k)^T \Delta Y_k \eta_k + \eta_k (\Delta Y_k)^T \Delta X_k}{\Delta Y_k (\Delta X_k)^T}$$

Stochastinis paieškos metodas

- Stochastinis iteracinis procesas: $X_{k+1} = X_k + Z_k$,
  čia Z_k – atsitiktinis vektorius, paimtas iš mažos taško X_k aplinkos, Jo pasiskirstymo dėsnis gali priklausyti nuo ankstesnių paieškos rezultatų,



Stochastinis paieškos metodo algoritmas

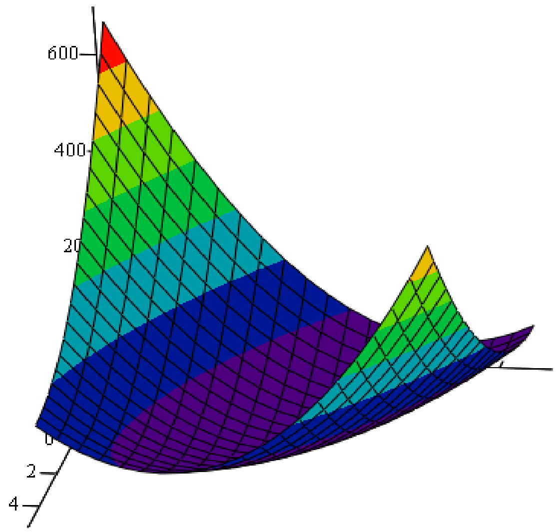
1. Apibrėžti tikslumą ε, pradinį tašką X_0, pradinį spindulį r, spindulio mažinimo koeficientą $\beta < 1$, taškų skaičių n ir periodą $2p$. Apibrėžiama sustojimo sąlyga. Iteracijų skaitiklis $i = 1$.
2. Aplinkoje $\rho(X_{i-1}, r)$ atsitiktinai parenkami n taškai $\{Y_k\}$. $X_i = arg\ min\ \{f(Y_k)\}$.
3. Vykdoma vienmatė paieška $P_{i-1} = X_i - X_{i-1}$ kryptimi: $\alpha_{i-1} = arg \min_{\alpha \geq 0} f(X_{i-1} + \alpha P_{i-1})$.
4. $X_{i+1} = X_{i-1} + \alpha_{i-1} P_{i-1}$.
5. Tikrinama sustojimo sąlyga. Jei sustojimo sąlyga nėra patenkinta, tai didinti iteracijų skaitiklį: $i = i + 2$, ir grįžti į 2 žingsnį (jei periodas yra iteracijos numerio kartotinis – sumažinti spindulį: $r = \beta\, r$).

Stochastinis paieškos metodas

- Paieška galėtų būti efektyvesnė, jei tolimesnėse iteracijose, atsižvelgdami į ankstesnius rezultatus, atsitiktinai imtume taškus ne visame apskritime, o tik jo dalyje.

- Nors stochastinė paieška nėra labai efektyvi, tačiau ji paprasta, nesunkiai realizuojama programiškai, ir **gali būti modifikuota, atsižvelgiant į sprendžiamo uždavinio specifiką**.

- **Buto funkcija** $f(x_1, x_2) = (a_{11} x_1 + a_{12} x_2 - b_1)^2 + (a_{21} x_1 + a_{22} x_2 - b_2)^2$



Daugelio kintamųjų funkcijų optimizavimas esant apribojimams

- Uždavinys formuluojamas taip:

$$\min_{X \in D} f(X),$$

kai leistinąją sritį D apibrėžia nelygybių ir lygybių sistema:

$$\begin{cases} g_i(X) \geqslant 0, & i = 1, \dots, m_1, \\ h_j(X) = 0, & j = 1, \dots, m_2. \end{cases}$$

- Nelygybinių ir lygybinių apribojimų vektorius žymėsime atitinkamai $g(X)$ ir $h(X)$.

Kintamųjų transformacijos

- Pagrindinės nelygybinių apribojimų klasės:

$$\begin{bmatrix} x_i \geqslant a\,, \\ x_i > a\,, \\ a < x_i < b\,, \\ a \leqslant x_i \leqslant b \end{bmatrix}$$

Kiekvienai klasei parinkus atitinkama kintamųjų transformaciją, optimizavimą su apribojimais galima suvesti į optimizavimą be apribojimų.

Kintamųjų transformacijos. Neneigiamumo reikalavimas

- Neneigiamumo reikalavimas $x \geq 0$ gali būti eliminuotas taikant keitinį

$$x = y^2.$$

- Analogiškai eliminuojami apribojimai

$$\begin{bmatrix} x \geqslant a & \rightarrow & x = a + y^2, \\ x \leqslant a & \rightarrow & x = a - y^2. \end{bmatrix}$$

- Tačiau nors šios transformacijos yra paprastos ir nesunkiai pritaikomos, neneigiamumo apribojimai yra eliminuojami TF ir likusių apribojimų netiesiškumo didinimo kaina, kas, savo ruožtu, mažina optimizacijos proceso efektyvumą.

Kintamųjų transformacijos. Neneigiamumo reikalavimas

- $$\min_{X \geqslant 0} f\left(X\right) = -x_1^2 + 3x_2^2 - 2x_3^2 + x_1 x_2 - x_1 x_3$$

$$\begin{cases} x_1^2 + x_2^2 + x_3^2 - 25 & = 0 \\ 8x_1 + 14x_2 - 7x_3 - 52 & = 0 \end{cases}$$

Kintamųjų transformacijos. Teigiamumo reikalavimas

- Teigiamumo reikalavimas $x > 0$ gali būti eliminuotas taikant keitinį

$$x = e^y.$$

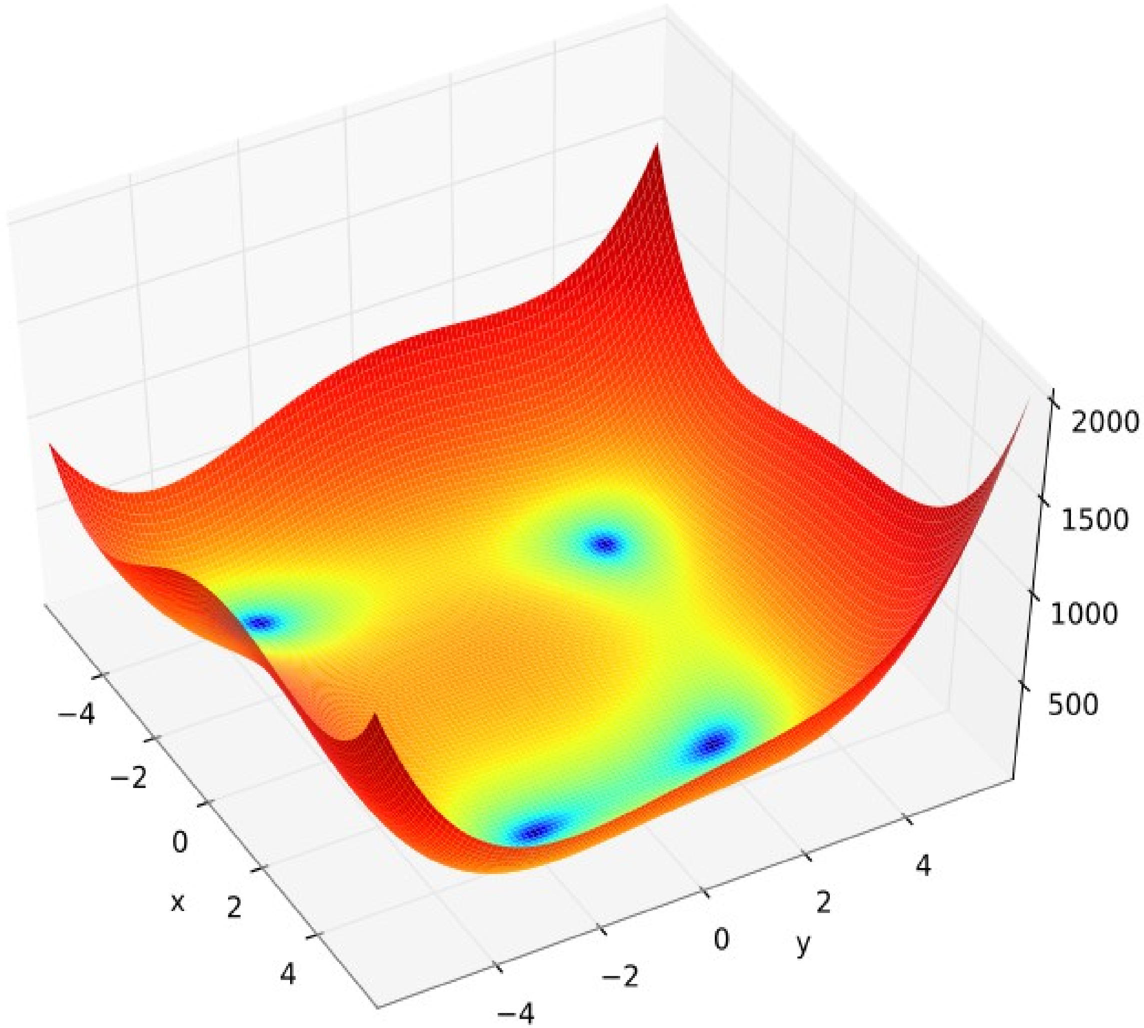
- Analogiškai eliminuojami apribojimai

$$\begin{bmatrix} x > a & \rightarrow & x = a + e^y, \\ x < a & \rightarrow & x = a - e^y. \end{bmatrix}$$

$$\min_{X > 0} f(X) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$$

Kintamųjų transformacijos. Teigiamumo reikalavimas

- $\min\limits_{X>0} f\left(X\right)=\left(x_1^2+x_2-11\right)^2+\left(x_1+x_2^2-7\right)^2$



Kintamųjų transformacijos. Griežti intervaliniai apribojimai

- Greižtų intervalinių apribojimų a < x < b eliminavimui gali būti taikomos funkcijos, 1D erdvę (- ∞; ∞) atvaizduojančias į atkarpą. [Bet kokia glodi pasiskirstymo funkcija tinka!].
- Kombinuojant tokias funkcijas su tiesine transformacija

$$u = \frac{b-a}{2} v + \frac{b+a}{2},$$

gausime atvaizdį iš begalinio intervalo (- ∞; ∞) į atvirą intervalą (a; b).

$$\begin{bmatrix} f(t) = \tanh t, & f(t) = e^{-e^{-t}} \\ f(t) = \arctan t, & f(t) = t\left(1 + |t|^{\gamma}\right)^{\frac{-1}{\gamma}} \end{bmatrix}$$

Lagranžo funkcija ir uždavinio su apribojimais stacionarieji taškai

- Tegu optimizavimo uždavinio TF ir apribojimų funkcijos yra tolydžiai diferencijuojamos. Nagrinėsime minimizavimo uždavinį

$$\min_{X \in D_1} f(X)$$

$$h(X) = 0$$

- Ieškodami TF ekstremumo, judame daugiamačiu paviršiumi $D_1 = \{X\!: h_1(X) = 0\}$.

$$\min_{x-2y+5z=0} x^2 + y^2 + z^2$$

Lagranžo funkcija ir uždavinio su apribojimais stacionarieji taškai

- Taškas X^* bus optimizavimo uždavinio stacionarusis taškas, jei gradiento $\nabla f(X^*)$ proekcija į paviršių D_1 bus nulinė. Tiksliau, jei nulinė bus proekcija į hiperplokštumą, liečiančią paviršių D_1 taške X^*.
- Taip bus, jei gradienatas $\nabla f(X^*)$ bus lygiagretus paviršiaus D_1 normalei $\nabla h(X^*)$:

$$\nabla f(X^*) = \lambda \nabla h(X^*),$$

čia λ yra skaliaras, vadinamas Lagranžo daugikliu.

- Taigi staciaonarusis taškas yra taškas, tenkinantis šias sąlygas:

$$\begin{cases} \nabla f(X) = & \lambda \nabla h(X) \\ h(X) = & 0 \end{cases}$$

Lagranžo funkcija ir uždavinio su apribojimais stacionarieji taškai

- Salygas galima užrašyti, naudojant Lagranžo funkciją $L(X,\lambda)$:

$$L(X,\lambda) = f(X) - \lambda h(X).$$

Taškai, tenkinantys sąlygą $\nabla L(X,\lambda) = 0$ yra stacionarūs, nes

$$\begin{cases} \nabla_X L(X,\lambda) = \nabla f(X) - \lambda \nabla h(X) = 0, \\ \nabla_\lambda L(X,\lambda) = h(X) = 0. \end{cases}$$

Lagranžo funkcija ir uždavinio su apribojimais stacionarieji taškai

- Daugelio lygybinių apribojimų atveju leistinoji sritis yra apribojimų paviršių sankirta. TF gradiento projekcija į ytą sankirtą stacionariame taške turi būti nulinė. TF gradientas turi būti lygiagretus kuriai nors apribojimų funkcijų gradientų tiesinei kombinacijai:

$$\nabla f(X) = \sum_{j=1}^{m_2} \lambda_j \nabla h_j(X).$$

- Lagranžo funkcija šiuo atveju yra

$$L(X,\ \Lambda) = f(X) - \sum_{j=1}^{m_2} \lambda_j h_j(X).$$

- Lagranžo daugiklių metodas susideda iš $n + m_2$ lygčių sistemos $\blacktriangledown L(X,\Lambda) = 0$ sprendimo. Sprendinių (X^*, Λ^*) komponentės yra optimizavimo uždavinio su lygybiniais apribojimais stacionarieji taškai. Dalis jų gali būti lokaliųjų minimumu taškai, jei tokių minimumų yra.

Lagranžo funkcija ir uždavinio su apribojimais stacionarieji taškai

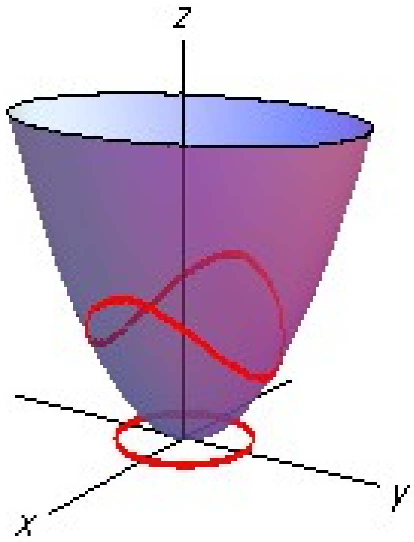
- $$\min_{2x^2+6y^2+z^2=1} x-2y+5z$$

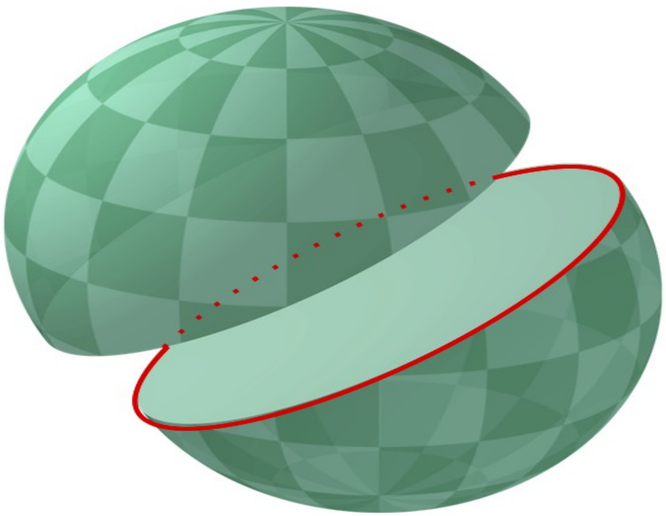
$$\min_{x^2+y^2=1} x^2+2y^2$$

$$\max_{x-2y+5z=1} \sqrt{1-(2x^2+6y^2)}$$

$$\min(4y-2z)$$
$$\begin{cases} 2x-y-z &= 2 \\ x^2+y^2 &= 1 \end{cases}$$

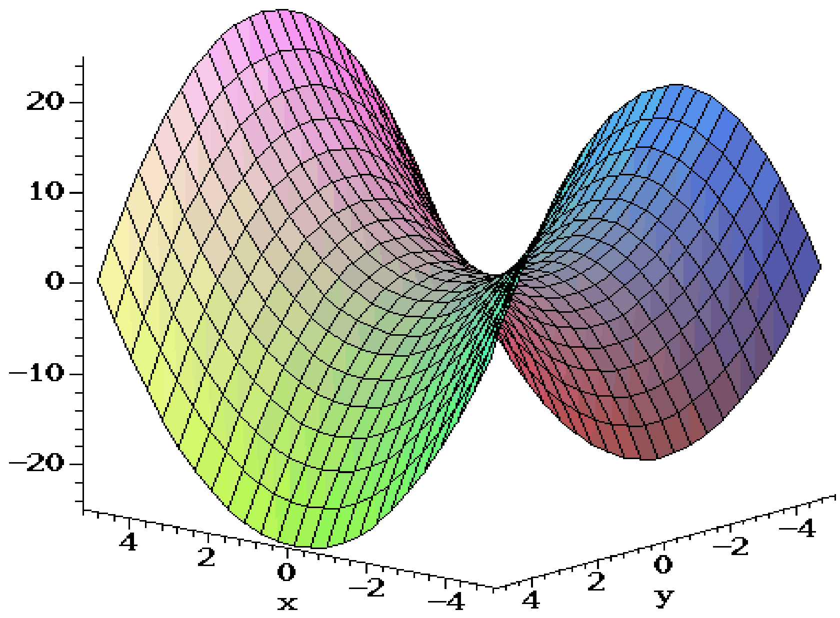
$$\left(\tfrac{2}{\sqrt{13}}, -\tfrac{3}{\sqrt{13}}, -2+\tfrac{7}{\sqrt{13}}\right)$$

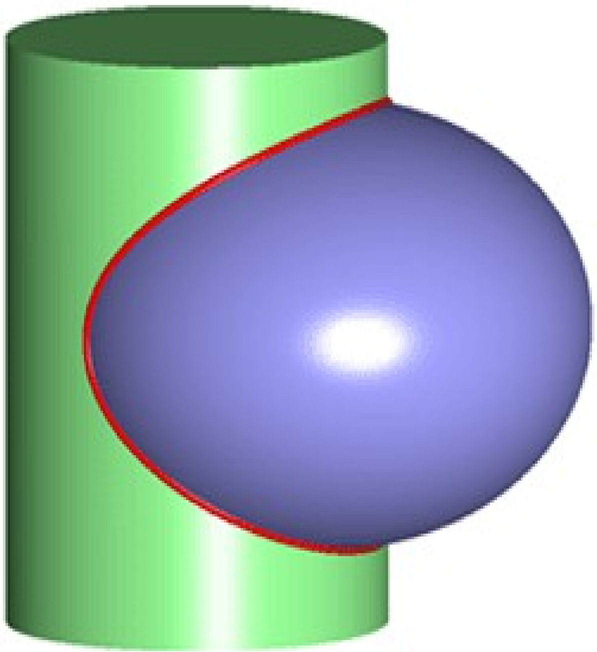




Lagranžo funkcija ir uždavinio su apribojimais stacionarieji taškai

- $$\min_{\frac{x^2}{8}+\frac{y^2}{2}=1} \; x\,y \qquad\qquad \max_{(x+1)^2+2\,y^2=1} \; \sqrt{1-10\,x^2-5\,y^2}$$





Lagranžo funkcija nelygybinių apribojimų atveju

- Tegu optimizavimo uždavinio TF ir apribojimų funkcijos yra tolydžiai diferencijuojamos. Nagrinėsime minimizavimo uždavinį

$$\min_{X \in D} f(X)$$

$$\begin{cases} g_i(X) \geqslant 0\ , & i = 1,\dots,m_1 \\ h_j(X) = 0\ , & j = 1,\dots,m_2 . \end{cases}$$

- Leistinojoje srityje D nelygybiniai apribojimai gali būti lygūs nuliui (tada jie vadinami **aktyviais**) arba didesni už nulį – tai **neaktyvūs** apribojimai.

Lagranžo funkcija nelygybinių apribojimų atveju

- Aktyvių apribojimų atveju Lagranžo funkcija formuojama panašiai kaip lygybinių apribojimų atveju, imdami tik **neneigiamus** Lagranžo daugiklius.

- Neaktyvūs apribojimai neturi jokios įtakos nei taško stacionarumo sąlygai, nei gradiento stracionariame taške reikšmei. Jei stacionariame taške visi apribojimei neaktyvūs, tai turime atvejį, ekvivalentų optimizavimui be apribojimų. Taigi neaktyvių apribojimų Lagranžo koeficientai turi būti nuliniai.

- Tai pasieksime su sąlyga

$$\mu_i g_i(X) = 0, \qquad i = 1,\dots, m_1,$$

t. y. arba apribojimas yra nenulinis (neaktyvus) ir atitinkamas koeficientas lygus nuliui, arba apribojimas nulinis, ir atitinkamas koeficientas teigiamas,

Lagranžo funkcija nelygybinių apribojimų atveju

- Turime Lagranžo funkcija:

$$L(X,\Lambda,M) = f(X) - \sum_{j=1}^{m_1} \mu_j g_j(X) - \sum_{j=1}^{m_2} \lambda_j h_j(X).$$

- Taigi nelygybinių apribojimų atveju Lagranžo daugiklių metodas suvedamas į šios sistemos sprendimą kintamųjų X, Λ, M atžvilgiu:

$$\begin{cases} \nabla L(X\,\Lambda, M) = 0, & \\ \mu_i g_i(X) = 0, & i = 1,\dots, m_1, \\ \mu_i \geqslant 0, & i = 1,\dots, m_1. \end{cases}$$

- Šios sistemos pagrindu formuluojamos Karušo-Kuno-Takerio (KKT) optimumo sąlygos.

Karušo-Kuno-Takerio optimumo sąlygos

- Formulavimas: V. Karušas (*magistro darbas*, 1939 m.)
- Įrodymas: V. Karušas (*disertacija*, 1942 m.)
- Nepriklausomai: H. Kunas ir A. Takeris (*publikacija konferencijos darb.,* 1951 m.)
- Paprastai vadinami Kuno-Takerio sąlygomis.

- **Būtinos optimumo sąlygos**. Tarkime, X^* yra optimizavimo su apribojimais uždavinio lokaliojo minimumo taškas, o TF ir apribojimų funkcijos yra tolydžiai diferencijuojamos. Tada egzistuoja konstantos $\lambda \geq 0$, $\mu_i \geq 0$, λ_i iš kurių bent viena nelygi nuliui ir

$$\begin{cases} \lambda \nabla f(X^*) - \sum_{j=1}^{m_1} \mu_j \nabla g_j(X^*) - \sum_{j=1}^{m_2} \lambda_j \nabla h_j(X^*) = 0 \\ \mu_i g_i(X^*) = 0, \quad i = 1,\dots, m_1. \end{cases}$$

Karušo-Kuno-Takerio optimumo sąlygos

- **Pakankamos optimumo sąlygos**. Tarkime, TF ir apribojimų funkcijos yra tolydžiai diferencijuojamos, funkcija f iškila, funkcijos g_i įgaubtos (iškilos į viršu, arba – tai ekvivalentu - iškilos yra $(-g_i)$), o funkcijos h_i – tiesinės. Tada jei leistinam taškui X^* egzistuoja tokios konstantos $\mu_i \geq 0$, ir λ_i , kad galiotų

$$\begin{cases} \nabla f\left(X^*\right) - \sum_{j=1}^{m_1} \mu_j \nabla g_j\left(X^*\right) - \sum_{j=1}^{m_2} \lambda_j \nabla h_j\left(X^*\right) = 0, \\ \mu_i g_i\left(X^*\right) = 0, \quad i = 1,\dots, m_1, \end{cases}$$

tai X^* yra šio optimizavimo uždavinio globaliojo minimumo taškas.

Iškila funkcija. Iškila aibė.

- **Funkcija** f: $D \to R$, apibrėžta aibėje D (aibė D yra iškila) yra vadinama **iškila**, jei bet kokiems X_1 ir X_2 iš D ir $0 \leq t \leq 1$

$$f\left(t X_1 + \left(1-t\right) X_2\right) \leqslant t f\left(X_1\right) + \left(1 - t\right) f\left(X_2\right).$$

- **Aibė** D yra **iškila**, jei bet kokiems X_1 ir X_2 iš D ir $0 \leq t \leq 1$: $t X_1 + \left(1-t\right) X_2 \in D$.

Lagranžo funkcijos (LF) minimakso taškai. Dualumas.

- Parodėme, kad optimizavimo uždavinio su apribojimais sprendiniai yra LF stacionarieji taškai. Parodysime, kad tie taškai nėra LF ekstremumai, bet balno taškai.
- Turime uždavinį:
$$\min_{X \in D} f(X)$$
$$\left\{ h_j(X) = 0 \,, \quad j = 1,\dots, m_2 . \right.$$
- Kai X tenkina apribojimus, tai $L(X, \Lambda) = f(X)$ su bet kokiomis Λ reikšmemis. Taigi
$$f(X^*) = \min_X f(X) = \min_X \max_\Lambda L(X, \Lambda).$$
- Kei apribojimai netenkinami, tei bent viena $h_j(X) \neq 0$, todėl
$$R(X) = \max_\Lambda L(X, \Lambda) = \infty .$$
- Taigi apribojimų netenkinantys taškai X nėra funkcijos $R(X)$ minimumo taškai.

Lagranžo funkcijos (LF) minimakso taškai. Dualumas.

- Taigi vietoje uždavinio su apribojimais

$$\min_{X \in D} f(X)$$

$$\left\{ h_j(X) = 0 \;, \quad j = 1, \ldots, m_2 . \right.$$

turime uždavinį be apribojimų

$$\min_X \max_\Lambda L(X, \Lambda),$$

su trūkia TF.

- Šį uždavinį galima apversti – sukeisti vietomis *min* ir *max*. Funkcija

$$G(\Lambda, M) = \inf_{X \in D} L(X, \Lambda, M)$$

vadinama optimizavimo su apribojimais uždavinio **dualiąją funkcija**, Lagranžo daugiklaiai – **dualiais kintamaisiais**, dualios funkcijos maksimizavimo uždavimys – **dualiu uždaviniu**.

Lagranžo funkcijos (LF) minimakso taškai. Dualumas.

- Dualios funkcijos maksimumas yra uždavinio

.

$$\min_{X \in D} f(X)$$

$$\begin{cases} g_i(X) \geqslant 0\ , & i = 1,\ldots,m_1 \\ h_j(X) = 0\ , & j = 1,\ldots,m_2. \end{cases}$$

minimumo įvetis iš apačios, nes bet kuriems $\Lambda \geq 0$ ir M galioja nelygybė

$$G(\Lambda, M) \leqslant f(X^*).$$

- **Iškilojo programavimo** uždaviniui galioja stipresnis teiginys – tiesioginio minimizavimo uždavinio ir dualaus uždavinio TF reikšmės sprendinių taškuose sutampa.

- *Iškilojo programavimo uždavinio TF yra iškila, nelygybinių apribojimų funkcijos įgaubtos, lygybinių apribojimų funkcijos – tiesinės*.

Sleiterio sąlyga.

- Taškas *Z* vadinamas uždavinio

.

$$\min_{X \in D} f(X)$$

$$\begin{cases} g_i(X) \geqslant 0\,, & i = 1,\dots,m_1 \\ h_j(X) = 0\,, & j = 1,\dots,m_2. \end{cases}$$

Sleiterio tašku, jei $g(Z) > 0$ ir $h(Z) = 0$.

- Jei uždavinio leistinoji sritis turi Sleiterio tašką, sakoma, kad uždavinys tenkina Sleiterio sąlygą.

- Jei uždavinys iškilojo programavimo su apribojimais uždavinys tenkina Sleiterio sąlygą ir turi sprendinį X^*, tai

$$f(X^*) = \max_{\Lambda,\, M} G(\Lambda, M).$$

Baudos funkcijų metodas.

- Nagrinėsime uždavinį

$$\min_{X \in D} f\left(X\right)$$

$$\left\{ g_j\left(X\right) \leqslant 0 \;, \quad j = 1,\dots,m \,. \right.$$

- Apibrėžkime funkciją $F(X) = f(X) + P(X)$, čia

$$P\left(X\right) = \begin{cases} 0 \,, & X \in D \,, \\ \infty \,, & X \notin D \,. \end{cases}$$

Kaip atrodo funkcijų F, f ir P grafikai?

Baudos funkcijų metodas.

- Funkciją $P(X)$ galima interpretuoti kaip begalinę baudą už leistinuosios srities pažeidimą.

- X^* yra pradinio sąlyginio optimizavimo uždavinio sprendinys tada ir tik tada, kai X^* yra nesąlyginio optimizavimo uždavimio *min* $F(X)$ sprendinys.
- Be to, $F(X^*) = f(X^*)$. Taigi sąlyginį uždavinį galima pakeisti nesąlyginiu.

- Problema: ∞.
- Taikant baudos funkcijų metodą, funkcija $P(X)$ aproksimuojama paprastesnių funkcijų $\{B_k(X)\}$ seka:

$$\lim_{k\to\infty} B_k(X) = P(X),$$

$$\lim_{k\to\infty} \left(f(X) + B_k(X) \right) = f(X) + P(X) = F(X).$$

Baudos funkcijų metodas.

- Seką $\{B_k(X)\}$ sudarome šitaip. Pasirenkame funkcija

$$B(X) = \begin{cases} 0\,, & X \in D, \\ c > 0, & X \notin D, \end{cases}$$

ir skaičių seką $\{r_k\}$, $r_k > 0$, $r_k > r_{k+1}$, $\lim\limits_{k \to \infty} r_k = 0.$

- Tada funkcijų seka $\{B(X) / r_k\}$ aproksimuoja funkciją $P(X)$, o seka $\{F(X,r_k)\}$, kai

$$F(X, r_k) = f(X) + \frac{1}{r_k} B(X), \qquad k = 1,2,\ldots,$$

aproksimuoja funkciją $F(X)$. $B(X)$ vadinama **baudos funkcija**.

- Dažniausiai naudojamos šitokios baudos funkcijos:

$$B(X) = \sum_{i=1}^{m} \left[max\left\{ g_i(X),\ 0 \right\} \right]^2, \qquad B(X) = \sum_{i=1}^{m} max\left\{ g_i(X),\ 0 \right\}.$$

Baudos funkcijų metodas.

- Tarkime, kad funkcijų seka $F(X, r_k) = f(X) + \frac{1}{r_k} B(X), \quad k = 1,2,...,$

yra sudaryta ir kiekviena sekos funkcija $F(X,r_k)$ turi minimumo tašką X_k. Minimumo taškų sekos $\{X_k\}$ savybes nusako dvi teoremos.

- Sekai $\{X_k\}$ teisingos šitokios nelygybės:

$$F(X_k, r_k) \leqslant F(X_{k+1}, r_{k+1}),$$
$$B(X_k) \geqslant B(X_{k+1}),$$
$$f(X_k) \leqslant f(X_{k+1}).$$

- Tegu sąlyginis optimizavimo uždavinys turi sprendinį X^*; TF, apribojimų funkcijos ir baudos funkcija yra tolydžios; visų funkcijų $F(X,r_k)$ minimumų taškai $\{X_k\}$ yra tam tikroje aprėžtoje uždaroje aibėje. Tada bet kuris sekos $\{X_k\}$ konverguojantis posekis konverguoja į sprendinį X^* ir

$$\lim_{k \to \infty} F(X_k, r_k) = f(X^*).$$

Baudos funkcijų metodas.

- Iš antros teoremos išplaukia, kad sprendžiant sąlyginį optimizavimo uždavinį, galima naudotis nesąlyginias metodais. Tuo tikslu pradinį uždavinį keičiame nesąlyginių uždavinių seka

$$\left\{ \min_{X \in R^n} \left(f(X) + \frac{1}{r_k} B(X) \right) \right\}.$$

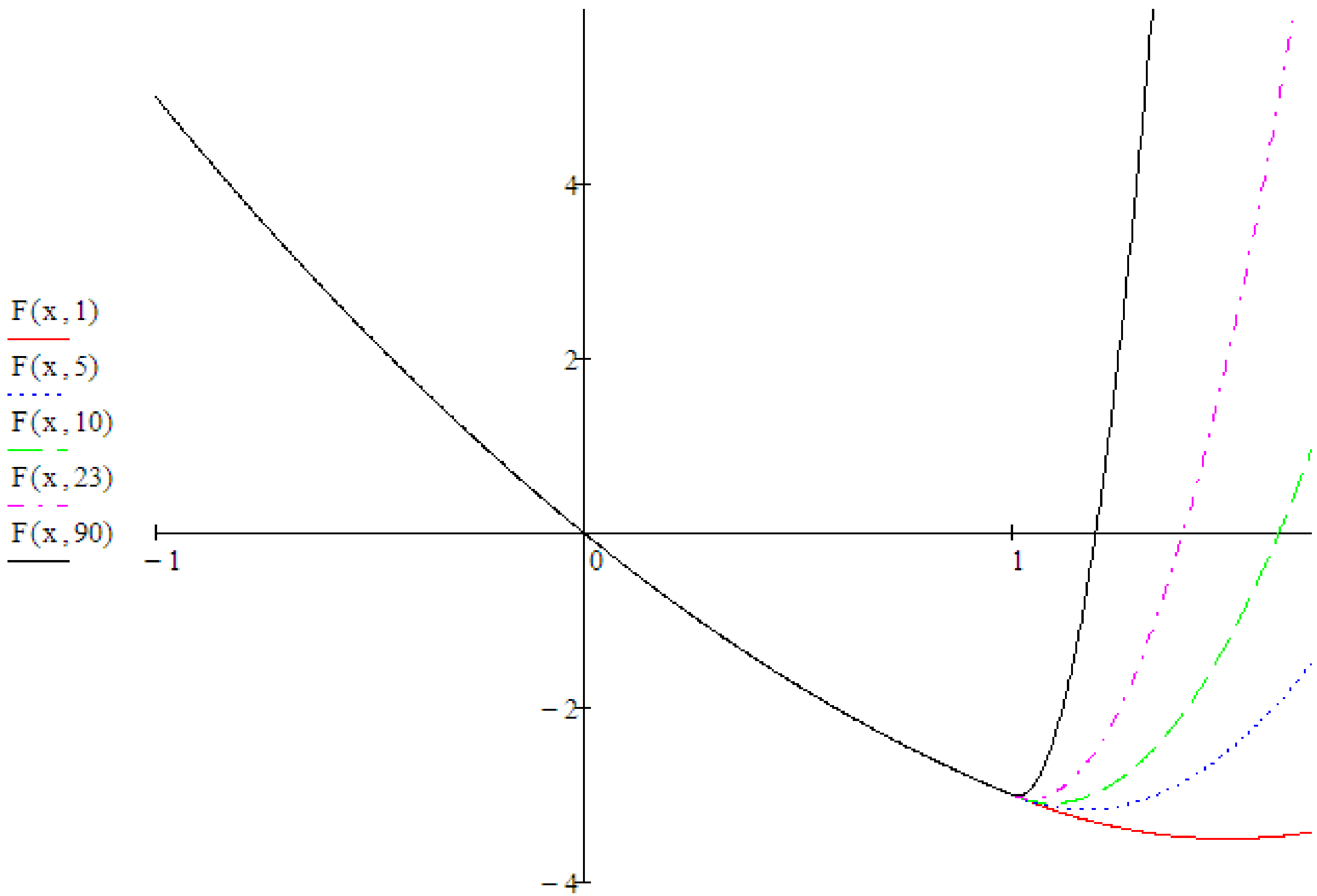
- Išsprendę sekos k-ąjį uždavinį, gauname pradinio uždavinio sprendinio artinį X_k.

- Taikant baudos funkcijų metodą, dažniausiai tenka optimizavimo uždavinius spręsti artutiniais skaičiavimo metodais. Apskaičiuoti X_{k+1} žinant X_k nėra sudėtinga, nes jei r_{k+1} nedaug skiriasi nuo r_k, tai ir X_{k+1} nedaug skiriasi nuo X_k . Todėl ieškant X_{k+1} , pradiniu artiniu galima imti $X_{k.}$

Baudos funkcijų metodas.

- $$\min_{x \leqslant 1} \left(x^2 - 4x\right)$$

  $$F(x,k) := x^2 - 4x + k \cdot \max(x-1,0)^2$$



Baudos funkcijų metodas.

- Seka $\left\{\frac{1}{r_k}B(X)\right\}$ aproksimuoja funkciją $F(X)$ iš leistinosios aibės išorės. Ir seka $\{X_k\}$ iš išorės artėja prie optimumo taško. Todėl šis metodas kartais vadinamas **išorinių baudų funkcijų metodu**.

- Taikant kitą metodą funkcija

$$P(X)=\begin{cases}0\,, & X\in D\,,\\ \infty\,, & X\notin D\,.\end{cases}$$

aproksimuojama seka funkcijų, kurios apibrėžtos tik leistinųjų vektorių aibės viduje. Toks metodas vadinamas **vidinių baudų funkcijų metodu** arba **barjerų metodu**.

Barjerų metodas.

- Kai uždavinio funkcijos už leistinosios srities ribų neapibrėžtos, taikomas barjerų metodas. Nagrinėsime uždavinį

$$\min_{X \in D} f\left(X\right)$$

$$\left\{ g_j\left(X\right) < 0 \,, \quad j = 1,\ldots,m \,, \right.$$

kur funkcijos $f(X)$ ir $g_i(X)$ yra tolydžios, aibė D – aprėžta, o jos vidinių taškų aibė – netuščia.

- Aibėje D apibrėžkime tolydžią funkciją $B(X)$, tenkinančią šias sąlygas: $B(X) > 0$, $B(X) \to \infty$, kai X artėja prie aibės D krašto.

- Imkime skaičių seką skaičių seką $\{r_k\}$, $r_k > 0$, $r_k > r_{k+1}$, $\lim\limits_{k \to \infty} r_k = 0.$

- Aibėje D seka $\{r_k B(X)\}$ aproksimuoja funkciją $P(X)$, čia

$$P\left(X\right) = \begin{cases} 0 \,, & X \in D \,, \\ \infty \,, & X \notin D \,. \end{cases}$$

Barjerų metodas.

- Aibėje D seka $\{F(X, r_k)\}$, kai

$$F(X, r_k) = f(X) + r_k B(X), \qquad X \in D, \qquad k \in N,$$

aproksimuoja funkciją $F(X)$.

- Funkcija $B(X)$ vadinama **barjerų** funkcija.

- Galima imti

$$B(X) = -\sum_{i=1}^{m} \frac{1}{g_i(X)}, \qquad B(X) = -\sum_{i=1}^{m} \ln(-g_i(X)).$$

Barjerų metodas.

- Taikant barjerų metodą pradinis uždavinys pakeičiamas uždavinių seka:

$$\min_{X \in D} F(X, r_k), \qquad k \in N.$$

- Metodo esmė išplaukia iš šių teiginių.

- **Teorema 1**. Egzistuoja toks X_k iš D, kad $\min\limits_{X \in D} F(X, r_k) = F(X_k, r_k), \quad k \in N.$

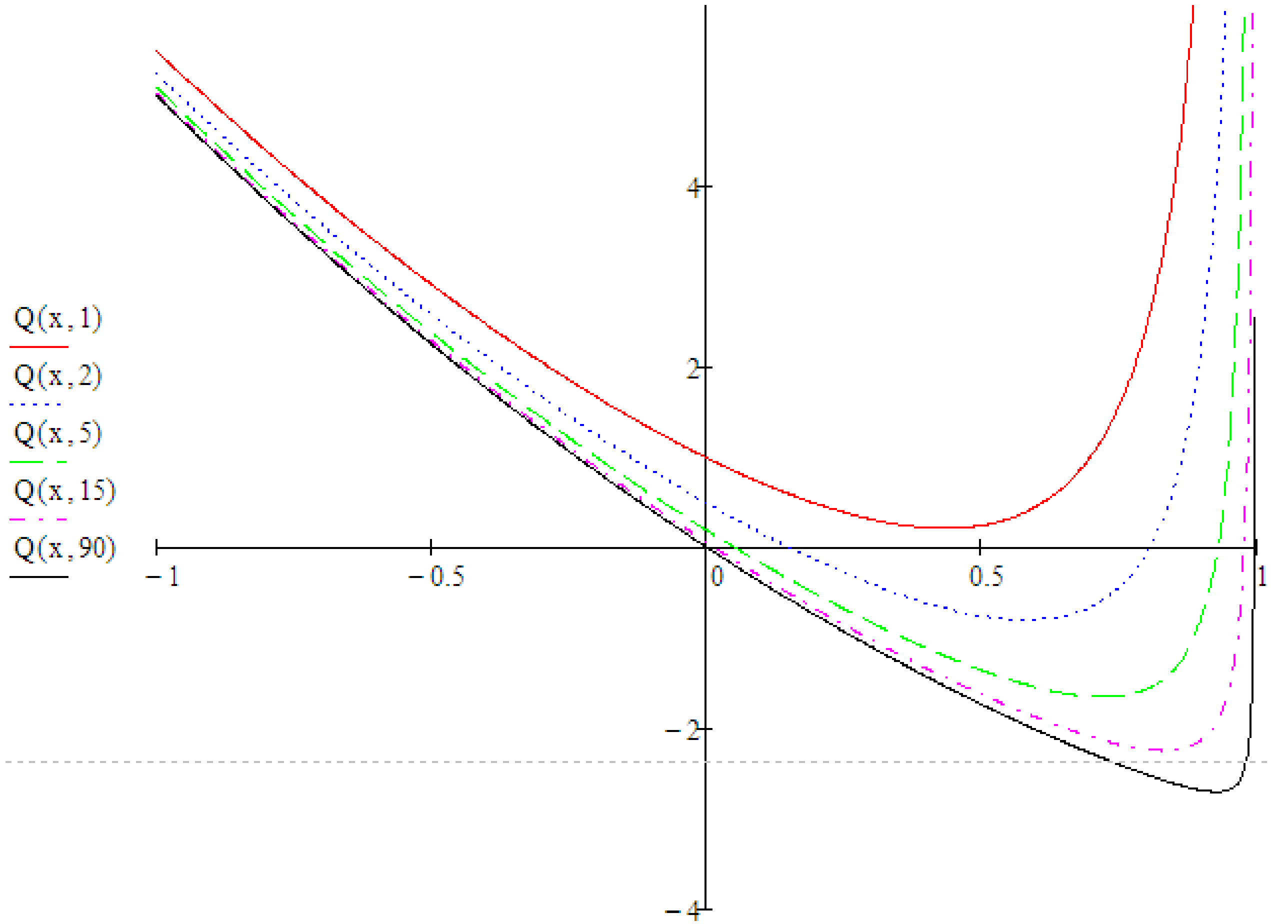
- **Teorema 2**. Seka $\{X_k\}$ turi tokias savybes:

$$F(X_k, r_k) \geqslant F(X_{k+1}, r_{k+1}),$$
$$B(X_k) \leqslant B(X_{k+1}),$$
$$f(X_k) \geqslant f(X_{k+1}).$$

Barjerų metodas.

- $$\min_{x<1}\left(x^2 - 4x\right)$$

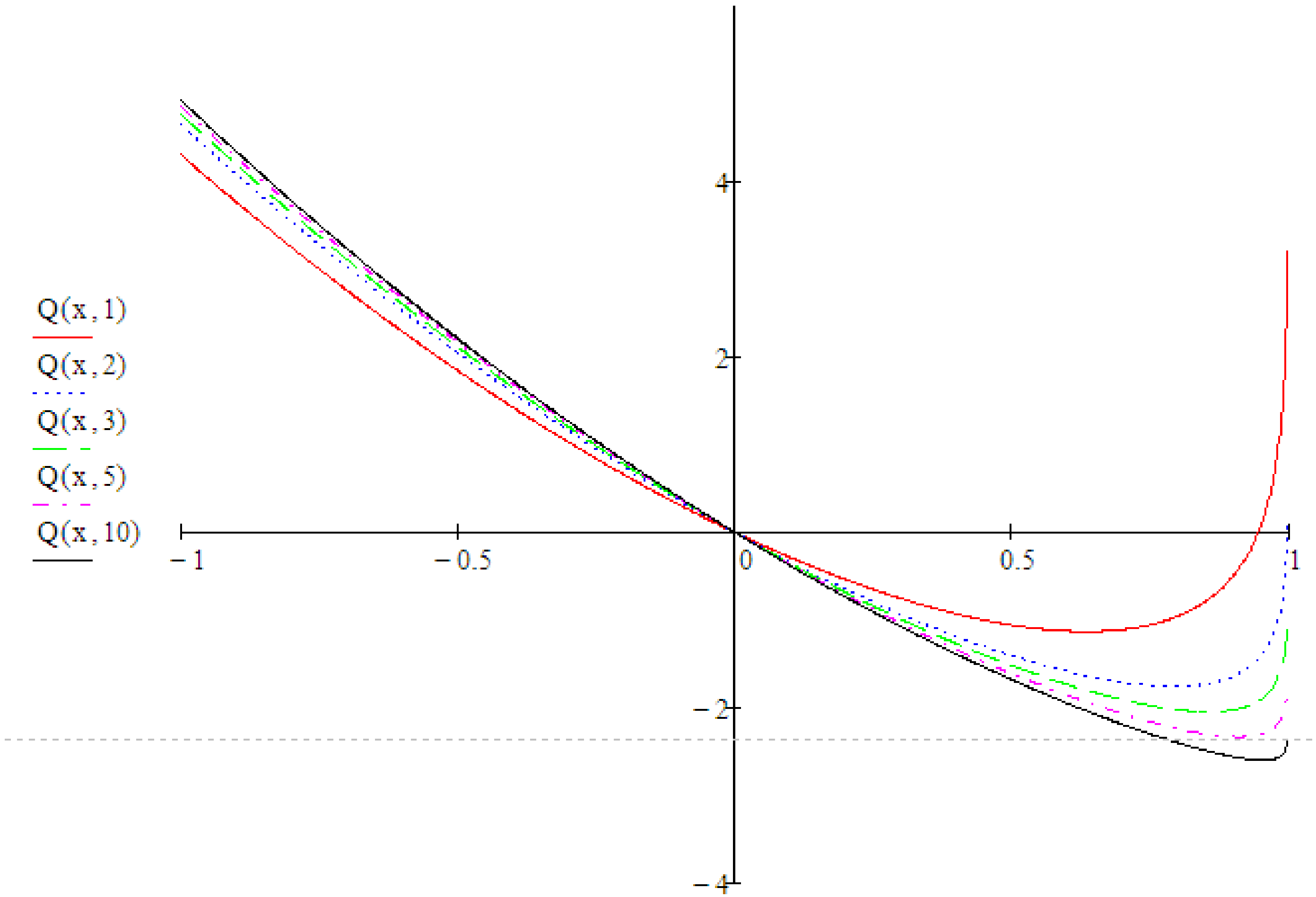
$$Q(x,k) := x^2 - 4x - \frac{1}{k}\cdot\frac{1}{(x-1)}$$



Barjerų metodas.

- $$\min_{x<1}\left(x^2 - 4x\right)$$

$$Q(x,k) := x^2 - 4x - \frac{1}{k}\cdot\ln(1-x)$$



Modifikuotos Lagranžo funkcijos metodas.

- Nagrinėtų baudų metodų trūkumas tas, kad ieškant minimumo, kuris yra ant leistinosios srities ribos, barjerų metodu gautas sprendinys tos ribos nepasiekia arba šiek tiek išeina už leistinosios srities, jei naudojamas išorinės baudos metodas. Be to, baudos sukuria griovius palei apribojimus.

- Lagranžo daugiklių metodu galima rasti tikslų sprendinį, tačiau tai iš tikrųjų įmanoma tik sprendžiant iškilojo programavimo arba nesudėtingus uždavinius. Bendru atveju surastas Lagranžo funkcijos stacionarusis taškas nebūtinai yra minimumas – jis gali būti maksimumo arba persilenkimo taškas.

Modifikuotos Lagranžo funkcijos metodas.

- Todėl naudojama **modifikuota Lagranžo funkcija** – prie Lagranžo funkcijos pridedama bauda, kuri pati ir jos išvestinės lygios nuliui stacionariame taške ir auga tolstant nuo jo.

$$ML(X,\Lambda,M,r) = f(X) - \sum_{j=1}^{m_2}\left[\lambda_j h_j(X) - \frac{1}{2} r_j h_j^2(X)\right] -$$

$$- \sum_{j=1}^{m_1}\begin{cases} \mu_j g_j(X) - \frac{1}{2} r_j g_j^2(X), & jei \quad g_j(X) < \frac{\mu_j}{r_j}, \\ \frac{\mu_j^2}{2 r_j}, & jei \quad g_j(X) \geqslant \frac{\mu_j}{r_j}. \end{cases}$$

Modifikuotos Lagranžo funkcijos metodas.

- k-oje iteracijoje naudojamus daugiklius pažymėkime λ_{jk}, μ_{jk} ir r_{jk}. Tada k-oje iteracijoje minimizuojame $ML\left(X, \Lambda_k, M_k, r_k\right)$ su ankstesnės iteracijos daugikliais ir gauname minimumo tašką X_{k+1}.

- Tada galioja lygybė

$$\nabla ML\left(X_{k+1}, \Lambda_k, M_k, r_k\right) = 0,$$

iš kurios galima gauti $\nabla f(X_{k+1})$ priklausomybę nuo Λ_k ir M_k.

- Siekdami, kad taškas X_{k+1} būtų ir Lagranžo funkcijos stacionarusis taškas, taip perskaičiuojame daugiklius λ_{jk}, μ_{jk} , kad galiotų lygybė

$$\nabla f\left(X_{k+1}\right) - \sum_{j=1}^{m_1} \mu_{j,\,k+1} \nabla g_j\left(X_{k+1}\right) - \sum_{j=1}^{m_2} \lambda_{j,\,k+1} \nabla h_j\left(X_{k+1}\right) = 0 .$$

Modifikuotos Lagranžo funkcijos metodas.

- Sulyginė iš lygites

$$\nabla ML(X_{k+1}, \Lambda_k, M_k, r_k) = 0,$$

gautą $\blacktriangledown f(X_{k+1})$ išraišką su

$$\nabla f\left(X_{k+1}\right) - \sum_{j=1}^{m_1} \mu_{j,\,k+1} \nabla g_j\left(X_{k+1}\right) - \sum_{j=1}^{m_2} \lambda_{j,\,k+1} \nabla h_j\left(X_{k+1}\right) = 0.$$

Turime daugiklių Λ ir M perskaičiavimo kitai iteracijai formules:

$$\begin{cases} \lambda_{j,\,k+1} = \lambda_{j,\,k} - r_{j,\,k} h_j\left(X_{k+1}\right), \\ \mu_{j,\,k+1} = max\left[\mu_{j,\,k} - r_{j,\,k} h_j\left(X_{k+1}\right),\, 0\right]. \end{cases}$$

Modifikuotos Lagranžo funkcijos metodas.

• Taigi kiekvienoje algoritmo k-je iteracijoje sprendžiamas minimizavimo be apribojimų uždavinys

$$X_{k+1} = \arg\min_{X} ML\left(X, \Lambda_k, M_k, r_k\right),$$

Lagranžo daugikliai Λ ir M perskaičiuojami pagal formules

$$\begin{cases} \lambda_{j,\,k+1} = \lambda_{j,\,k} - r_{j,\,k} h_j\left(X_{k+1}\right), \\ \mu_{j,\,k+1} = max\left[\mu_{j,\,k} - r_{j,\,k} h_j\left(X_{k+1}\right),\ 0\right] \end{cases}$$

ir padidinami baudos daugikliai r_{jk}. Baudos daugiklis didinamas (paprastai dešimteriopai), kai apribojimų pažeidimai lėtai mažėja.

• Jei apribojimai yra lygybiniai, tai baudos daugiklis r_{jk} didinamas, kai

$$\left|h_j\left(X_{k+1}\right)\right| > 0{,}25\left|h_j\left(X_k\right)\right|.$$

• Jei apribojimai yra lygybiniai, tai baudos daugiklis r_{jk} didinamas, kai

$$\left|min\left[g_j\left(X_{k+1}\right),\ 0\right]\right| > 0{,}25\left|min\left[g_j\left(X_k\right),\ 0\right]\right|.$$

Gradiento projekcijų (leistinųjų krypčių) metodas.

- Kai nusileidimo kryptis (pvz., antigradientas) veda už leistinosios srities ribų, tadi ji keičiama kryptimi, kuri yra kuo mažiau nutolusi nuo gautos nusileidimo krypties ir neveda už srities ribų.

- Metodas paprastai taikomas tiesiniams (hiperplokštuminiams) apribojimams ir nauja nusileidimo kryptis gaunama nusileidimo kryptį projektuojant į aktyvių apribojimų sankirtos hiperplokštumą.

- Aktyviais apribojimais vadinami tie apribojimai, kurie nagrinėjamame taške įgyja nulinę reikšmę, t. y. tame taške nusako leistinosios srities ribą.

- Kelių aktyvių apribojimų sankirta yra mažesnio skeičiaus matmenų hiperplokštuma. TF antigradiento ortogonali projekcija į tą hiperplokštumą yra nusileidimo kryptis ir sudaro mažiausią kampą iš visų leistinųjų krypčių kampų su antigradientų.

- Aprašysime projektavimo operatorių.

Gradiento projekcijų (leistinųjų krypčių) metodas.

- Tiesiniai apribojimai yra šio pavidalo:

$$\begin{cases} a_{i1}x_1 + a_{in}x_n + b_i \geqslant 0, & i = 1,\dots,m_1, \\ a_{i1}x_1 + a_{in}x_n + b_i = 0, & i = m_1+1,\dots,m_1+m_2. \end{cases}$$

- Įveskime matricą $A(X)$, kurios elementai yra aktyvių apribojimų koeficientai a_{ij}. Tai koeficientai tų apribojimų, kurie taške X lygus nuliui. Tada projektavimo operatorius yra

$$P(X) = I - A(X)^T \left(A(X)A(X)^T\right)^{-1} A(X).$$

Gradiento projekcijų metodo iteracinio proceso schema.

- $$X_{k+1} = X_k + \alpha_k S_k,$$

kur

$$\alpha_k = arg \min_{0 \leqslant \alpha \leqslant d} f\left(X_k + \alpha S_k\right),$$

$$S_k = -\frac{\nabla f\left(X_k\right)}{\left\|\nabla f\left(X_k\right)\right\|} P^T\left(X_k\right).$$

- Žingsnio daugiklio apribojimas $0 \leq \alpha \leq d$ nurodo, kiek galima nueiti kryptimi S_k, iki bus prieitas naujas aktyvus apribojimas.

Gradiento projekcijų (leistinųjų krypčių) metodas.

•

$$\min\left(x_1^2 + \left(x_2 + 3\right)^2\right)$$

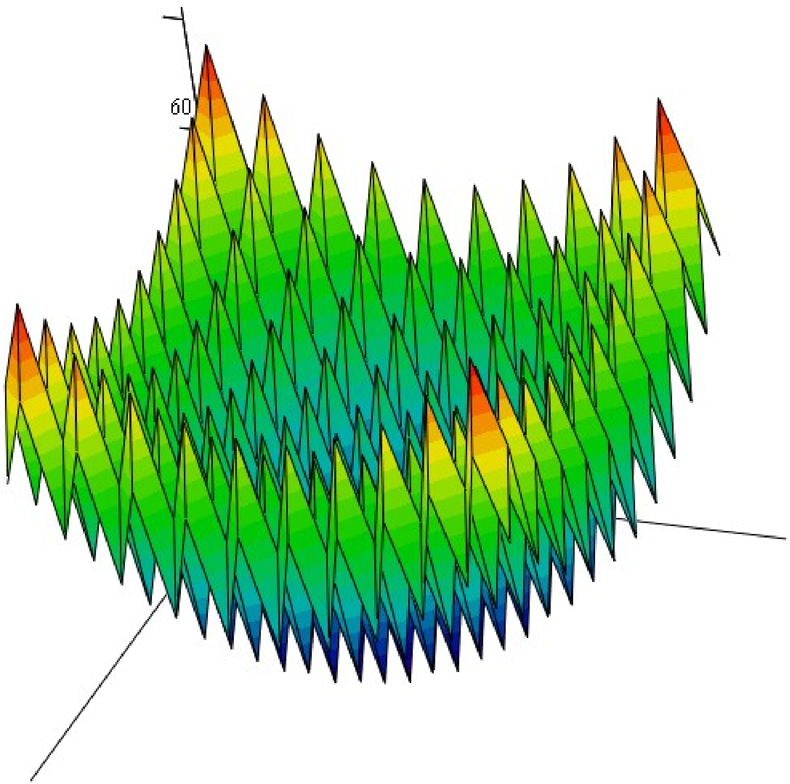
$$\begin{cases} -9x_1 - 12x_2 + 36 \geqslant 0 \\ x_1 + x_2 + 2 \geqslant 0 \\ x_1 \geqslant 0 \\ x_2 \geqslant 0 \end{cases}$$

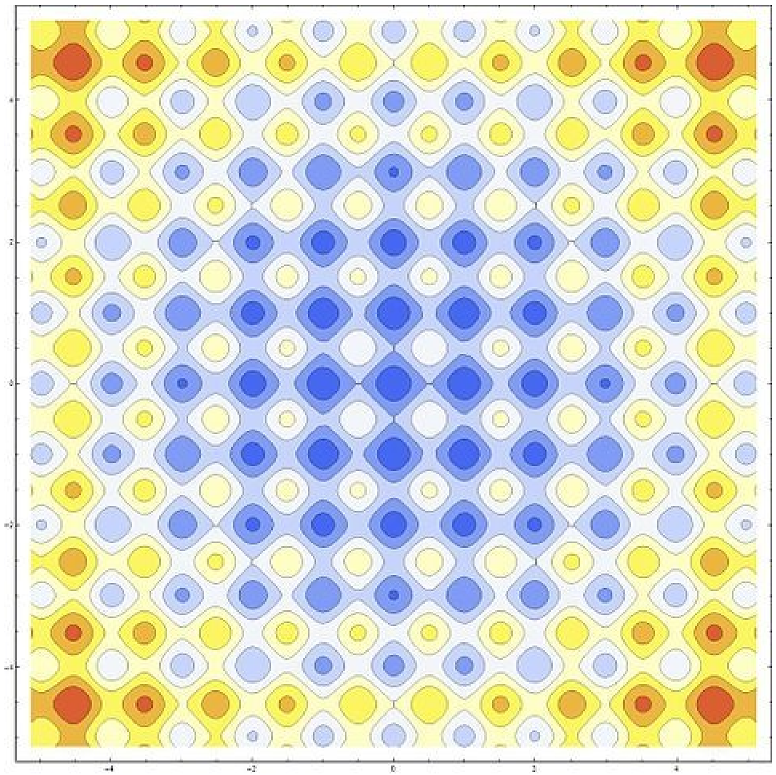
• $x_0 = (4; 0)$

Globalusis optimizavimas. Daugiaekstremiai uždaviniai

- Praktikoje susiduriama su globaliojo optimizavimo uždaviniais, kai lokialiųjų minimumų yra ne vienas. Jų skaičius gali būti didžiulis.

$$f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10\cos(2\pi x_1) - 10\cos(2\pi x_2).$$





Paprasčiausia atsitiktinė paieška

- Tai yra pasyvus metodas, kai parenkant bandymų taškus neatsižvelgiama į anksčiau atliktų bandymų rezultataus. Leistinojoje srityje parenkami N bandymų taškai $\{X_k\}$ ir juose apskaičiuojamos TF reikšmės $\{f(X_k)\}$.

- Apytiksliu sprendiniu laikoma minimali gautoji TF reikšmė ir atitinkamas minimumo taškas.

$$\min_{1 \leqslant k \leqslant N} f\left(X_k\right), \qquad X^* \sim \arg \min_{1 \leqslant k \leqslant N} f\left(X_k\right)$$

Paprasčiausia atsitiktinė paieška

- Metodo sėkmė priklauso nuo to, kaip parinkdime bandymų taškas.

- Retai kada turime apriorinių žinių apie TF pobūrį, todėl bet kuris leistinosios srities taškas laikomas vienodai perspektyviu. Tolkiomis sąlygomis bandymų taškus leistinojoje srityje tikslinga išdėstyti kuo tolugiau.

- Tolygiai išdėstyti taškus paptasti tik retais atv., pvz., kai leistinoji sritis – vienmatė atkarpa, daugiamatis kubas arba gretasienis.

- Bendru atveju tolygaus išdėstymo algoritmai sunkaiu realizuojami
  - kai matmenų skaičius $n > 1$;
  - kai bandymų skaičius $N \neq 2^n$;
  - kai bandymų skaičius N reikia didinti nepažeidžiant taškų išdėstymo tolygumo.

Paprasčiausia atsitiktinė paieška

- Taškai išdėstomi tikimybine prasme tolygiai paprasčiausiu atsitiktinės paieškos **Monte Karlo metodu**, kai bandymų taškų koordinatės – nepriklausomi a.d., tolygiai pasiskirstę daugiamatėje leistinojoje srityje.

- Kai TF yra tolydi, tokiu metodu priartėjama prie sprendinio su tikimybe 1.

- Nemažindami bendrumo laikysime, kad leistonosios srtities tūris lygus 1. Tarkime, X* yra minimumo taškas, U – norimai maža šio taško aplinka, kurios tūris lygus V.

- Tikimybė, kad per N nepriklausom bandymų į aplinką U nepateks nei vienas taškas lygi $(1\text{-}V)^N$. Tikimybė, kad nors vienas iš N taškų pateks į minimumo aplika lygi $1\text{-}(1\text{-}V)^N$. Kai $N \to \infty$, ši tikimybė artįja prie 1.

- Taigi gauta **tolydžios** funkcijos minimali reikšmė min $f(X_k)$ stochastiškai artėja prie funkcijos minimumo.

Paprasčiausia atsitiktinė paieška

- Kai leistinoji sritis yra nesudėtingo pavidalo, tada Monte Karlo metodas yra **paprastas** skaičiavimo požiūriu. Vietoje bandymų taškų koordinačių pakanka imti tolygiai pasiskirsčiusius pseudoatsitiktinius skaičius.

- Jei leistinoji sritis yra n-matis gretasienis, aprašomas nelygybėmis $a_j < x_j < b_j$, tai taško X_i koordinatės generuojamos taip: $x_{ij} = a_j + r_{ij}(b_j - a_j)$. Čia $r_{ij} \sim R(0;1)$.

LP paieška

- Nagrinėsime LP sekas, taikomas vietoje atsitiktinių skaičių Monte Karlo metoduose, taip pat sprendžiant optimizavimo uždavonius.

- Taškų pasiskirstymo **tolydumas gali būti suprantamas įvairiai**. Kartais manoma, kad daugiamačiai taškai tolygiausiai išdėstomi vadinamaisiais kubiniais tinklieliais. Tačiau optimizavimo algoritmuose naudojant taip išdėstytus taškus, rezultatas būtų blogesnis negu su atsitiktiniais taškais.

- Kokio pobūdžio tašku išdėstymo tolygumas reikalingas?

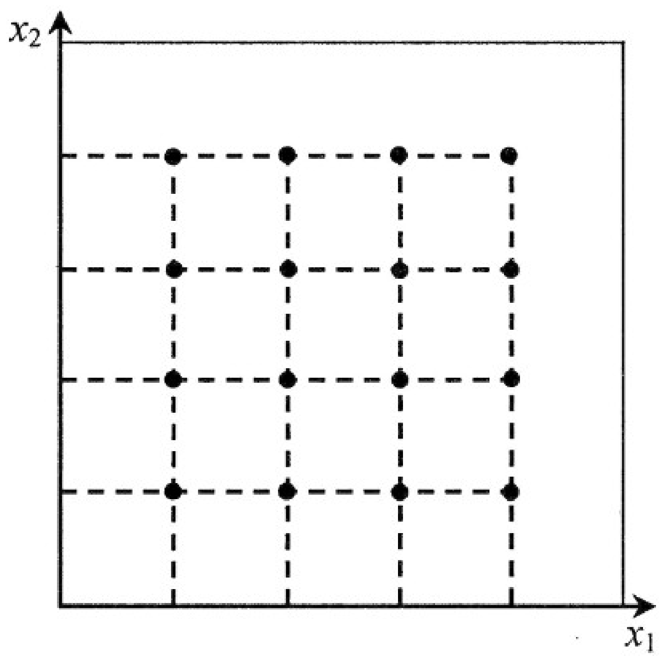
- *n* kintamųjų TF teoriškai priklauso nuo visų n kintamųjų, tačiau praktiniuose uždaviniuose jų priklausomumo laipsnis nėra vinodas. Būna esminių kintamųjų, kurių įtaka didesnė, nei kitų.

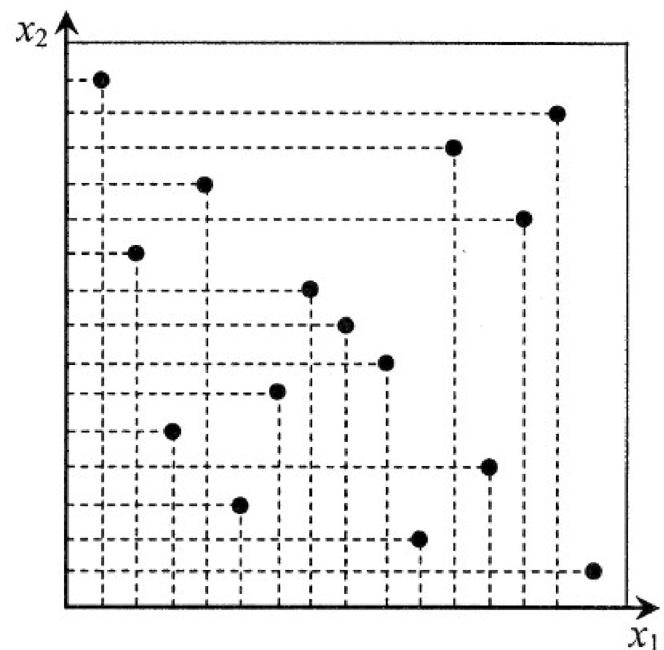
- Tarkime, kad TF iš tikrųjų priklauso nuo *k* esminių kintamųjų. Tuomet svarbus ne bandymo taškų išdėstymo *n*-mačiame kube, o taškų projekcijų į atitinkamą *k*-matę kubo hiperbriauną tolygumas.

LP paieška

- Iš anksto patikrinti, kurie kintamieji ar jų gruopės dominuoja, beveik neįmanoma. Galima manyti, kad TF labiausiai priklauso nuo bet kurio pavienio kintamojo, nuo bet kurios kintamųjų poros ir t.t.

- Kadangi iš anksto nežinome TF struktūros, turime konstroti taškų sistemas, kurių projekcijos į visas galimas įvairias matmenų skaičiaus kubo hiperbriaunas būtų išdėstytos tolygiai.





LP sekų programiniai generatoriai

- Kuriant LP sekų programinisu generatorius buvo keliami šie reikalavimai:

1) Generuojamų taškų tolygumas turi būti asimptotiškai optimalus;
2) Taškai tūri būti išdėstomi tolygiai ne tik kai $N \to \infty$, bet ir kai N mažas;
3) Generuojamų taškų koordinačių apskaičiavimo algoritmas turi būti palyginti paprastas, kad generatoriausprograminė įranga taškus generuotų maždaug tiek pat laiko kaip standartiniai pseudoatsitiktinių skaičių generatoriai.

- Pirmasis reikalavimas LP sekoms tenkinamas automatiškai. Turėta sunkumų tik su antruoju ir trečiuoju reikalavimais. Sukurtieji LP generatoriai, esant mažiems N, taškus išdėsto tolygiausiai, kai $N = 2^m - 1$.

LP sekos daugiamačiame vienetiniame kube generavimo algoritmas

- n-mačių LP sekos taškų koordinates pažymėkime $\{X_i\}$, $X_i = (x_{i1}, \ldots, x_{in})$, $i = 1,\ldots, N$.
- Taško numerį užrašykime kaip dvejetainį skaičių: $i = e_m e_{m-1} \ldots e_2 e_1$.
- LP taškų koordinatės randamos atlikus tokią operaciją:

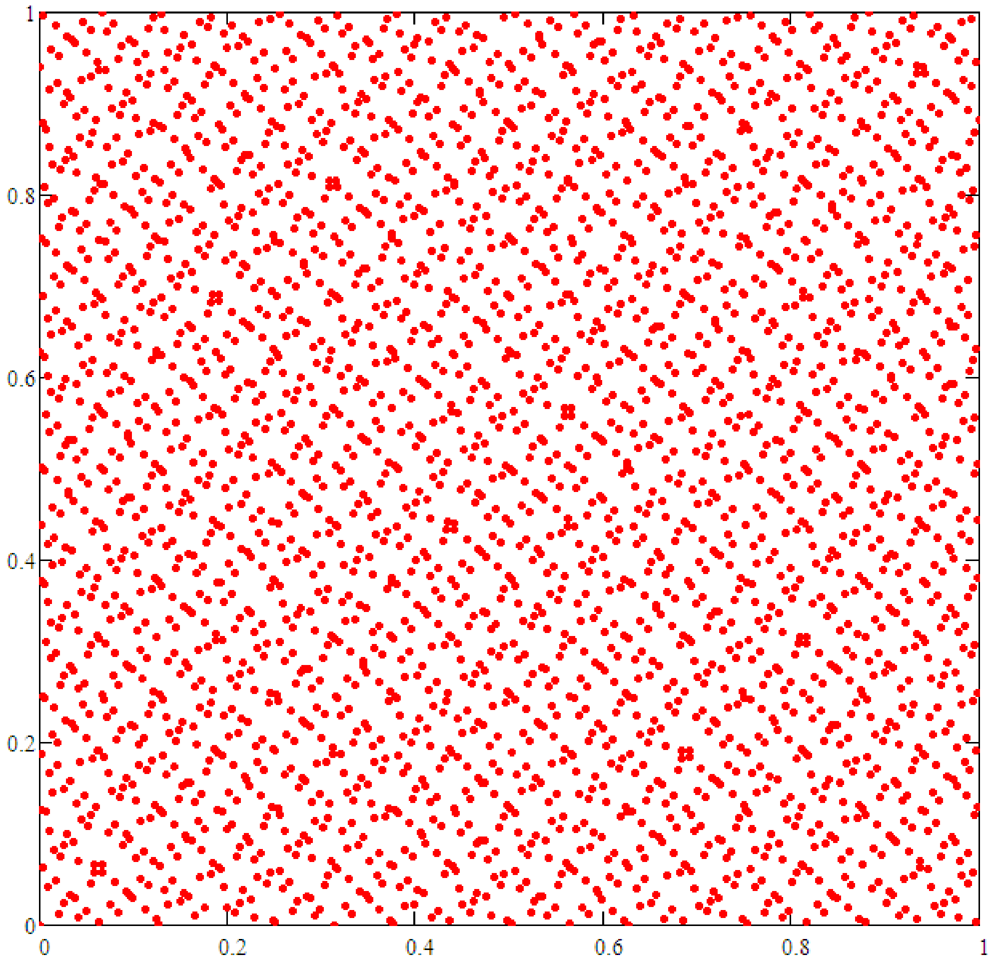
$$x_{ij} = \underset{k=1}{\overset{m}{XOR}} \left(e_k V_{jk}\right)$$

- Dydžiai V_{js} imami iš lentelių. Lentalių apimtis yra $1 \le s \le 20$, $1 \le j \le 51$, taigi pagal jas galima rasti n-mačius taškus X_i, kai $n < 51$ ir $N < 2^{21}$.
- Pirmasis taškas visada būna kubo centre, t. y. $x_{i1} = 1, \ldots, x_{in} = 0{,}5$.

V_{js} koeficientai

Dešimtainiai skaičiai					Dvejetainiai skaičiai				
$j \backslash s$	1	2	3	4	$j \backslash s$	1	2	3	4
1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	1	0,1	0,01	0,001	0,0001
2	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{5}{8}$	$\frac{15}{16}$	2	0,1	0,11	0,101	0,1111
3	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{7}{8}$	$\frac{11}{16}$	3	0,1	0,01	0,111	0,1011

LP_2 seka (N = 3000)



0.5000	0.5000
0.2500	0.7500
0.7500	0.2500
0.1250	0.6250
0.6250	0.1250
0.3750	0.3750
0.8750	0.8750
0.0625	0.9375
0.5625	0.4375
0.3125	0.1875

Pastabos

- Kai matmenų skaičius $n > 3$, teoriškai neįmanoma sukonstruoti tokių generatorių (tiksliau, parinkti tokių dydžių V_{js}), kad visos galimos pirmųjų taškų projekcijos būtų tolygiai išdėstytos.
- Generatorių autoriai stengėsi, kad bent jau pirmosios taškų X_i koordinatės būtų išdėstytos tolygiai.
- Todėl dudėjant koordinatės, o kartu ir TF kintamojo numeriui, projekcijų kokybė ir LP paieškos metodo efektyvumas mažėja. Kai j = 8-10, jo efektyvumas pasidaro artimas atsitiktinio ie6kojimo metodo efektyvumui.
- Vadinasi, LP paie6kos metodo efektyvumas priklauso nuo kintamųjų numeravimo tvarkos ir reikia stengtis, kad pirmieji būtų esminiai kintamieji, kurių įtaka didžiausia.

Pasyviųjų metodų efektyvumas

- Paieškos metodų privalumai:

1) TF gali būti bet kokia: daugiaekstremia, su trūkiais, be išvestinių. Svarbu tik, kad galėtume rasti jos reikšmę.
2) Taikant šiuos metodus, pačiam metodui sunaudojama labai negaug skaičiavimo laiko.

- Įvertinkime N – kiek bandymų reikėtų atlikti, kad išspręstume uždavinį norimu tikslumu. Tikimybė p, kad nors vienas iš N taškų pataikys į globaliojo minimumo taško aplinką, kurios tūris lygus V, lygi $1-(1-V)^N$. Taigi

$$N = \frac{\ln(1-p)}{\ln(1-V)}$$

- Formulė rodo ir trečią privalumą: efektyvumas nepriklauso nuo TF kintamųjų skaičiaus n. Ar taip yra iš tikrųjų?

Pasyviųjų metodų efektyvumas

- Norint pasiekti didelį tikslumą (mažą V): ir didelę pataikymo tikimybę p, reikia atlikti ne tiek jau daug bandymų:

V \ p	0,9	0,95	0,99
0,1	22	29	40
0,01	230	299	459
0,005	460	598	919

- Atrodytų nesunku kad ir tūkstantį kartų apskaičiuoti sudėtingoa TF reikšmes. Užtat su tikimybe 0,99 rastume globaliojo minimumo taško aplinką, kurios tūris **V** sudaro 0,005 dalį viso srities tūrio.

Pasyviųjų metodų efektyvumas

- Deja, daugiamačiu atveju metodas visiškai naefektyvus, nes tikslumą apibūdina ne minimumo aplinkos tūris V, o tos aplinkos (tarkime, ji daugiamačio rutulio formos) spindulio ilgis, t. y. tikslumas pagal koordinates.

$$V_n = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}+1\right)} R^n$$

- Kai V = 0,001 ir n = 60, paklaida lygi _, o kintamojo kitimo intervalo ilgis lygus 1.

- Paradoksas atsiranda dėl daugiamatės erdvės savybių, nes joje objektų projekcijos ir tūriai susiję kitaip nei esame įpratę.

Kartotiniai nusileidimai

- Paprasčiausi **euristiniai** (intuicija pagrįsti) **globaliojo** optimizavimo klasės metodai sudaryti apibendrinant **lokaliųjų** metodų idėjas.

- Tarkime, leistinosios srities D dalyje D_i (tegu tokių dalių yra m) TF yra unimodali ir turi minimumo tašką M_i. Tada mums jau žinomais lokaliojo optimizavimo metodais iš bet kurio pradinio srities D_i taško P_i galima rasti atitinkamą lokaliojo minimumo tašką M_i. Srities D dalis D_i yra vadinama lokaliojo minimumo taško M_i **traukos sritimi**.

- Jei turime pradinių taškų $\{P_i\}$ seką, tai globaliojo optimizavimo uždaviniui išspręsti pakanka išspręsti m lokaliojo optimizavimo uždavinių. Deja iš anksto nežinomas ne tik lokaliųjų minimumų skaičius m, bet ir apskritai ar uždavinys yra vieno, ar daugelio ekstremumų.

Kartotiniai nusileidimai

- Vienas paprasčiausių būdų – pradinius nusileidimo taškus rasti vienų iš tolygaus taškų išdėstymo metodų.

- Nusileisdami iš atsitiktinio arba LP sekos taško į artimiausią lokalųjį minimumą, tarsi patiksliname netikslų pasyvios paieškos sprendinį.

- Tokie metodai yra neracionalūs – iš daugelio pradinių taškų kartais leidžiamės į tą patį lokalųjį minimumą, o globalųjį galima ir praleisti. Todėl taikomos įvairios pradinių taškų atrinkimo taisyklės, leidžiamasi lygiagrečiai iš daugelios pradinių taškų, o vėliau blogesni iš jų atmetami.

Atsitiktinė paieška klasterizavimo būdu

- Algoritmas pagrįstas daugamatės statistikos pricedūra – klasterine analize, kuri tiria taškų daugiametėje erdvėje grupavimą į vadinamuosiuis klasterius.

- Analizuojant atstumus tarp taškų nustatomos taškų sutankėjimo sritys, klasterių centrai, ir taškai priskiriami kuriam nors klasteriui.

- Atsitiktinės paieškos klasterizavimo būdu algoritmą sudaro šie etapai:

1) Pradinių tašku parinkimas;
2) Lokalioji paieška;
3) Klasterių radimas;
4) Pradinių taškų skaičiaus mažinimas.

- Po ketvirtojo etapo, jei uždavinys dar neišspręstas, vėl grįžtama į antrąjį etapą.

Atsitiktinė paieška klasterizavimo būdu

- Per **pimąjį** etapą leistinojoje srityje tolygiai parenkamas tam tikras pradinių atsitiktinių taškų skaičius.

- Per **antrąjį** etapą vienu iš lokaliųjų metodu iš visų pradinių taškų leidžiamasi į lokaliuosius minimumus. Gautas apytikslis sprendinys yra pradinis taškas tolimiems žingsniams.

- Per **trečiąjį** etapą analizuopjami nauji taškai. Jei taškai sudaro klasterį, spėjama, kad jie yra iš to paites lokaliojo minimumo traukos srities. Taigi iš jų visų leistis toliau neracionalu.

- Per **ketvirtąjį** etapą taškų skaičius palaipsniui mažinamas, tačiau patikimumui padidinti klasteryje paliekama daugiau nei vienas tolesnio nusileidimo taškas.

- Šiuo metodu randami ne tik globalieji, bet ir lokalieji minimumai, o tai dažnai domina specialistus.

Klasterinė analizė

- Taikydami KA, nustatome objektų panašumą ir suskirstome juos į **klasterius** .
- Klasteris – **panašių** objektų grupė.
- Klasterinės analizės tikslas – suskirstyti objektus taip, kad skirtumai klastrių viduje būtų kuo mažesni, o tarp klasterių – kuo didesni.
- Skirstydami objektus į klasterius dažniausiai **nežinome**, kiek klasterių realiai egzistuoja.

Klasterinė analizės etapai

- 1. Pasirinkti klasterizuojamus objektus.
- 2. Nuspręsti, pagal kokius požymius klasterizuojame.
- 3. Pasirinkti objektų panašumo matą.
- 4. Vienu ar kitu metodu suskirstyti objektus į klasterius.
- 5. Peržiūrėti gautus rezultatus.

Objektų panašumo matai (metriniai atstumo matai)

- Daugelis metrinių atstumo matų yra metrikos.

- **Metrika** – tai skaitinė neneigiama dviejų objektų X ir Y funkcija $d(X,Y)$, tenkinanti sąlygas:

1) Simetriškumo: $d(X,Y) = d(Y,X)$.

2) Trikampio nelygybės: $d(X,Y) \leq d(X,Z) + d(Z,Y)$.

3) Netapačių objektų atskiriamumo: jei $X \neq Y$, tai $d(X,Y) \neq 0$.

4) Tapačių objektų neatskiriamumo: jei $d(X,Y) = 0$, tai $X = Y$.

Metriniai atstumo matai

Atstumas	$d(X,Y)$
Euklido	$\lVert X - Y \rVert = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$
Euklido atstumo kvadratas	$\lVert X - Y \rVert = \sum_{i=1}^{m} (x_i - y_i)^2$
Minkovskio	$\left(\sum_{i=1}^{m} \lvert x_i - y_i \rvert^{l} \right)^{\frac{1}{l}}, \quad l > 0$
Manheteno (blokinis)	$\sum_{i=1}^{m} \lvert x_i - y_i \rvert$
Čebyšovo	$\max_{i} \lvert x_i - y_i \rvert$
Mahalanobio atstumo kvadratas	$(X - Y)^T V^{-1} (X - Y),$ $v_{ij} = \frac{1}{m} \sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y})$ V — vektorių kovariacinė matrica

Klasterinės analizės metodų klasifikacija

- Skiriamos dvi pagrindinės KA metodų klasės – **hierarchiniai** ir **nehierarchiniai** metodai.

- **Hierarchinių** metodų rezultatai nusako klasterių tarpusavio hierarchiją. Taikydami hierarchinius metodus, nustatome bendrą visų klasterių tarpusavio priklausomybių struktūrą ir tik po to sprendžiame, koks klasterių skaičius yra optimalus.

- Hierarchiniai metodai skirstomi į **jungimo** ir **skaidymo** metodus. **Jungimo** metodai smulkius klasterius jungia vis į stambesnius, kol galu gale lieka vienas. **Skaidymo** metodai yra jungimo metodų priešingybė: vienintelis klasteris nuosekliai skaidomas

- **Nehierarchiniai** metodai paprastai taikomi tada, kai iš anksto žinomas (arba pasirenkamas) klasterių skaičius ir norima tiriamus objektus klasterizuoti.

Jungimo metodai

- Hierarchinių jungimo metodų strategiją galima apibrėžti taip:

1) Turime N klasterių po 1 objektą ir simetrinę atstumų matricą D.
2) Pagal atstumų matricą nustatome du klasterius, tarp kurių atstumas yra mažiausias. Tarkime, kad tai klasteriai U ir V.
3) Sujungiame klasterius U ir V. Naują klasterį pavadiname $[UV]$. Tada atstumų matricą pakeičiame taip: a) išbraukiame stulpelius ir eilutes, atitinkančius klasterius U ir V; b) pridedame eilutę ir stulpelį su atstumais tarp $[UV]$ ir likusiųjų klasterių.
4) Kartojame 2 ir 3 žingsnius (N-1) kartą. Procesą baigiame, kai visi objektai yra viename klasteryje.

- Tyrėjas pats sprendžia, kuriuo etapu objektų paskirstymas į klasterius yra optimalus.

- Dažnai optimalus klasterių skaičius randamas matematiškai – parenkamas **skaidymo kokybės funkcionalas** ir žiūrima, kuriuo hierarchinės klasterizacijos etapu jis minimalus. SKF pavyzdys gali būti objektų atstumų iki klasterių centrų sumos.

Klasterių panašumo matai ir jungimo metodai

Atstumas	$d(U,V)$
• Vienetinės jungties (artimiausio kaimyno)	$d(U,V) = \min_{X_i \in U,\ Y_j \in V} d(X_i, Y_j)$
Pilnosios jungties (tolimiausio kaimyno)	$d(U,V) = \max_{X_i \in U,\ Y_j \in V} d(X_i, Y_j)$
Vidutinės jungties	$d(U,V) = \frac{1}{n_U\ n_V} \sum_{X_i \in U} \sum_{Y_j \in V} d(X_i, Y_j)$
Centrų	$d(U,V) = d(\bar{U}, \bar{V})$
Vordo	$d(U,V) = \frac{\left\| \bar{U} - \bar{V} \right\|^2}{\frac{1}{n_U} + \frac{1}{n_V}}$

Klasterių panašumo matai ir jungimo metodai

- Jungimo metodas, kai atstumas tarp klasterių yra vienetinės jungties, vadinamas vienetinės jungties metodu ir t.t.:

- Pavyzdys. Žinomas 5 automobilių galingumas (a.j.) ir kiek jie sunaudoja benzino 100 km. Šiuos automobilius norima suskirstyti į klasterius (vienetinės jungties metodu ir Vordo metodu). Automobilių skirtumams įvertinti naudosime Euklido atstumo kvadratą.

	Automobilis	Galingumas	Degalai
1	Renault	95	8
2	Audi	92	8
3	Subaru	95	10
4	Kia	94	6
5	Opel	93	5

Klasterių panašumo matai ir jungimo metodai

- Koks metodas yra geriausias?

- Vienareikšmiško atsakymo nėra: klasterius apibudina daug charakteristikų: požymių vektorių skaida nuo klasterio centro, forma ir t.t. Neturint išankstinės informacijos apie nagrinėjamų duomenų struktūras, gautus rezultatus lyginti sunku. Todėl objektų klasterizavimui rekomenduojama taikyti keletą klasterizavimo metodų.

- Tikslumo taisyklė. Sprendžiant klasterizavimo uždavinius, visados pravartu žinoti atsakymą.

Klasterizavimo metodų savybės

- Centrų metodų klasterizavimo eigoje atstumai tarp klasterių dažnai mažėja, nors iš tiesų jungiami vis mažiau ir mažiau panašūs klasteriai (tai ne pranašumas!).

- Taikant Vordo metodą gaunama daug mažų klasterių.

- Vordo metodo rezultatams labai didelę įtaką daro objektai su požymių išskirtimis.

Genetiniai algoritmai, evoliucinės strategijos

- Šie euristiniai metodai mėgdžioja biologinę evoliuciją ir natūralią atranką.

- Naudojama adaptyvi paieškos procedūra, besiremianti kandidatų į sprendinio tašką populiacija.

- Kiekvienoje iteracijoje atmetami prasčiausi kandidatai į optimalų sprendinį. Geriau prisitaikę kandidatai «kryžminami» su kitais sukeičiant jų komponentes. Taip gaunama nauja kandidatų karta. Galimos ir «mutacijos» - nežymūs kai kurių kandidatų komponenčių pakeitimai.

- Nuoseklūs kryminimai iš mutacijos sukuria naujus sprendinius, kurie palaipsniai artėja globaliojo optimumo link.

- .Įvairiuose metoduose naudojamos įvairios kryžminimo, mutacijos ir kandidatų atmetimo taisyklės. Pradinė populiacijos aibė susideda iš N taškų, kuriose TF reikšmės žinomos. Toliau vykdomas sutraukimo procesas, artinantis taškus link globaliojo optimumo aplinkos.

Valdomos atsitiktinės paieškos algoritmas (Price, 1978)

1) Generuojame pradinę populiacijos aibę $S = \{X_i\}$, i = 1,...,N. Taškai leistinojoje srityje generuojami pagal tolygų skirstinį. Randamos reikšmės $\{f(X_i)\}$. Pageidautina, kad $N >> n$, kur n – TF kintamųjų skaičius. Populiacijos kartų skaitykliui k suteikime nulinę reikšmę, $k = 0$.
2) Randame blogiausią ir geriausią aibės S taškus; X_{min} ir X_{max}, atitinkančius f_{min} ir f_{max}. Jei sustojimo sąlyga ($f_{max} - f_{min} < \varepsilon$) patenkinta, sustojame.
3) Atsitiktinai parenkame (tai būtų selekcija) n+1 tašką iš S: $\{Y_i\}$, i = 1 ,..., n+1. Generuojame naują tašką (tai būtų mutacija): $X_{new} = 2\,C - Y_{n+1}$. Čia C yra taškų $\{Y_i\}$ svorio centras (skaičiuojamas kaip taškų $\{Y_i\}$, i = 1 ,..., n aritmetimis vidurkis). Jei X_{new} nepriklauso leistinajai sričiai, kartojame trečią žingsnį; priešingu atveju randame $f(X_{new})$. Jei $f(X_{new}) > f(X_{max})$, taip pat kartojame trečią žingsnį.
4) Keičiame aibę S ir padidinę generacijų skaitiklio reikšmę $k = k$+1 grįžtame į antrą žingsnį.

$$S = S \cup \{X_{new}\} \setminus \{X_{max}\}$$

Atkaitinimo modeliavimo metodas

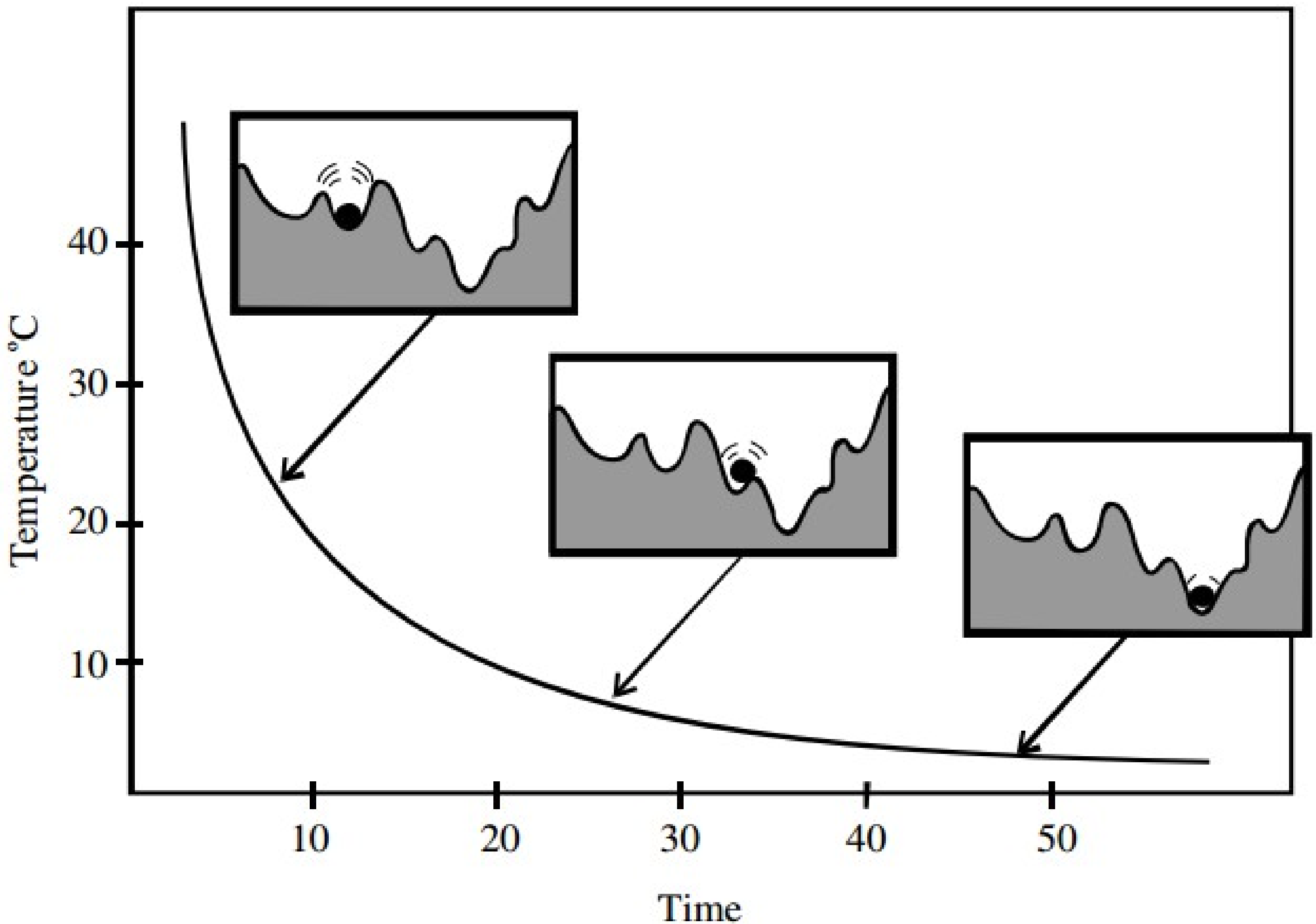
- Simulated annealing (Kirkpatrick, 1983)

- Metodas grindžiamas analogija tarp įkaitinto metalo atvėsimo proceso ir minimizavimo metodo.

- Atkaitinimas pagrįstas faktu, kad įkaitintas metalasvėsdamas ir stingdamas įgauna kristalines minimalios energijos struktūras. Jei aušinimas pakankamai lėtas, metalo atomai suformuoja struktūrą, kurios energija **globaliai minimali.**

Atkaitinimo modeliavimo metodas

•



Atkaitinimo modeliavimo metodo algoritmas

- Algoritmas susideda iš dviejų ciklų

- Vidiniame cikle generuojamas atsitiktinis taškas, kurio skirstinio tankis didžiausias aplink geriausią vidiniame cikle surastą tašką (rekordą). Jei naujame taške TF reikšmė mažiausia iš surastų, jis pakeičia ankstesnį rekordą.

- Su tikimybe, kuri mažėja «šąlant», geriausias iš vidiniame cikle surastų taškų gali būti keičiamas blogesniu (taip vykdoma globalesnė paieška).

- Išoriniame cikle mažinama temperatūra ir taškų skirstinio dispersija. Todėl paieška darosi vis lokalesnė.

Atkaitinimo modeliavimo metodo algoritmo schema.
1 žingsnis.

- Nustatomas pradinis taškas (jis kartu ir geriausias taškas, ir rekordas):

$$X_{opt} = X_0, \qquad X_{rek} = X_0.$$

- Nustatoma pradinė temperatūra .

$$T_{opt} = T_0.$$

- Nustatomas pradinis dispersijos parametras $d = d_0$.
- Nustatomas dispersijos parametro mažinimo koeficientas β: 0 < β < 1.
- Nustatomas temperatūros mažinimo koeficientas α: 0 < α < 1.
- Nustatomas leistinas nesėkmingų bandymų pagerinti rekordą skaičius, *k*.
- Nustatoma galutinė temperatūra, T_{gal}.

Atkaitinimo modeliavimo metodo algoritmo schema. 2 žingsnis.

- Atsitiktinai generuojamas naujas taškas:

$$X = X_{rek} + d\,Z\,.$$

- Čia Z – atsitiktinis taškas (vektorius su nuliniu vidurkiu) .

$$x_i^{opt} = x_i^{rek} + d\,r\,, \qquad r \sim R[-1\,;\ 1]$$

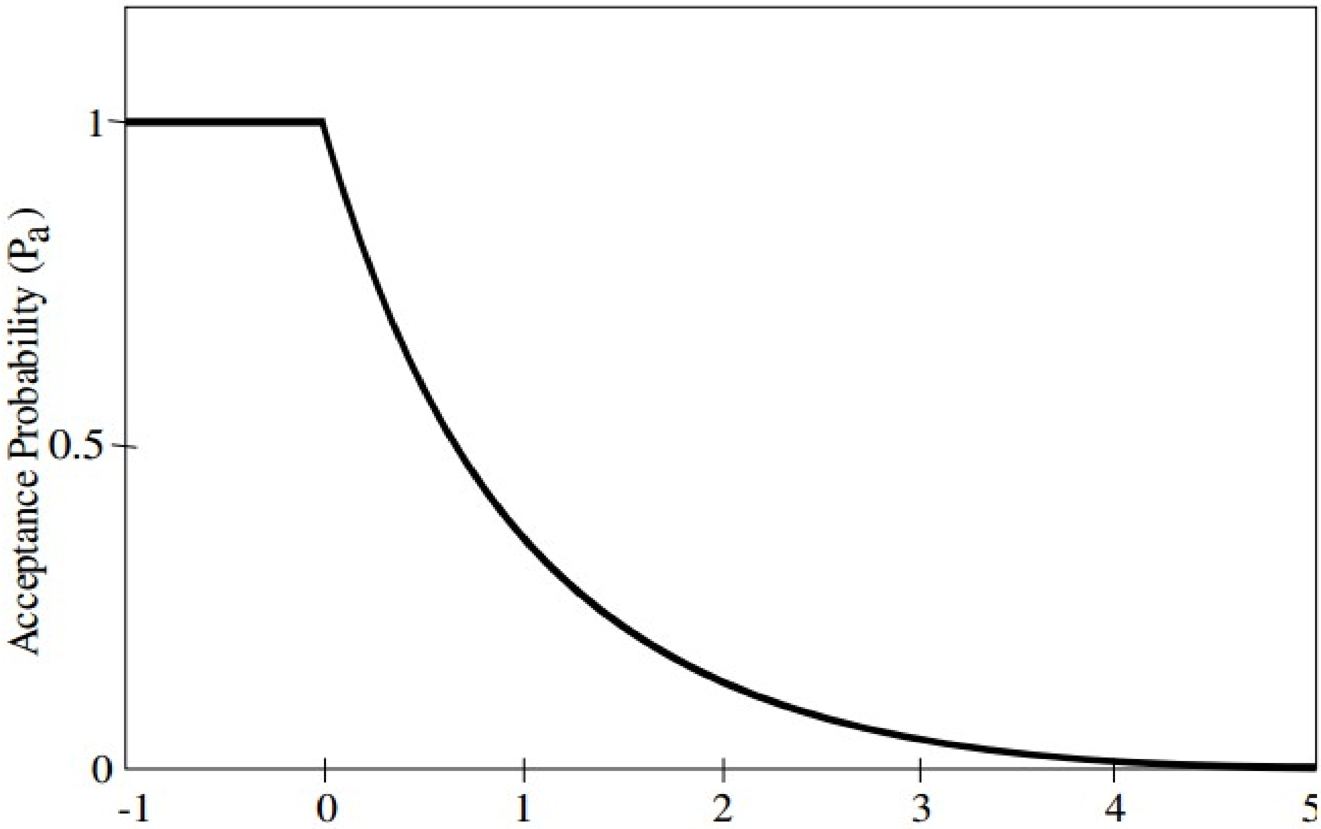
- Jei $f(X) < f(X_{opt})$, tai keičiamas geriausias taškas: $X_{opt} = X\,.$
- Jei $f(X) < f(X_{rek})$, tai eiti į 4 žingsnį; priešingu atveju su tikimybe

$$P = min\left(1,\ e^{-\frac{f(X) - f(X_{rek})}{T}}\right)$$

eiti į 4 žingsnį ir nauju rekordu laikyti blogesnį tašką: $X_{rek} = X\,.$

Atkaitinimo modeliavimo metodo algoritmo schema.
2 žingsnis. Metropolio sąlyga

•



Atkaitinimo modeliavimo metodo algoritmo schema. 3-6 žingsniai.

- **3 žingsnis**. Vidinio ciklo baigimo sąlyga: jei $f(X_{opt})$ nebuvo pagerintas per k TF skaičiavimų, eiti į 5 žingsnį.

- **4 žingsnis**. Su $X_{rek} = X$ eiti į 2 žingsnį.

- **5 žingsnis**. Jei $T < T_{gal}$, bagiti algoritmo darbą, o minimumo aproksimacija laikyti tašką X_{opt}.

- **6 žingsnis**. Nustatyti $X_{rek} = X_{opt}$, $T = \alpha T$, $d = \beta d$ ir eiti į 2 žingsnį.

Pastaba. *Atkaitinimo modeliavimo metodas retai įstringa lokaliojo minimumo taške. Šis metodas išsiskiria galimybe atsisakyti pasiekto lokalaus rekordo ir pereiti į blogesnį tašką.*

Globaliojo optimizavimo metodų testavimas. Standartinės testinės funkcijos

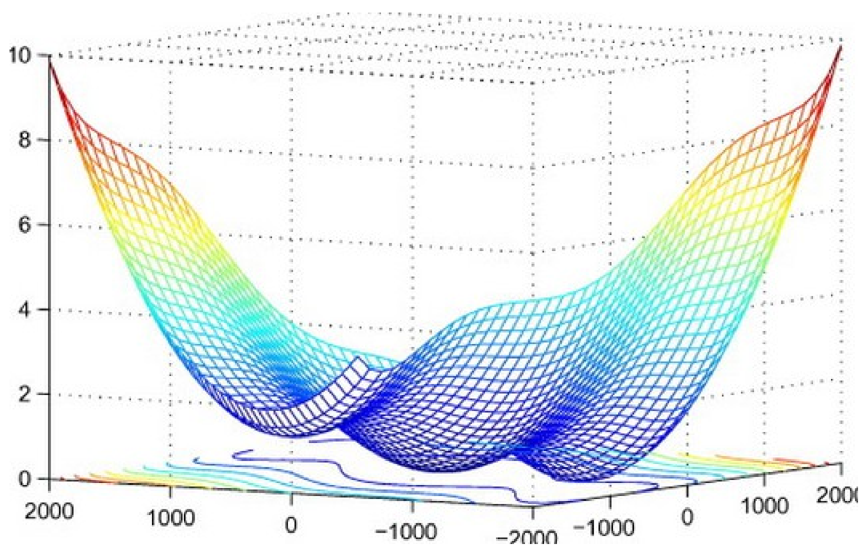
- Daugelio GO metodų prielaidų bei efektyvumo teoriškai dažniausiai neįmanoma patikrinti. Todėl jų kokybė yra vertinama eksperimentiškai – sprendžiant uždavinius.

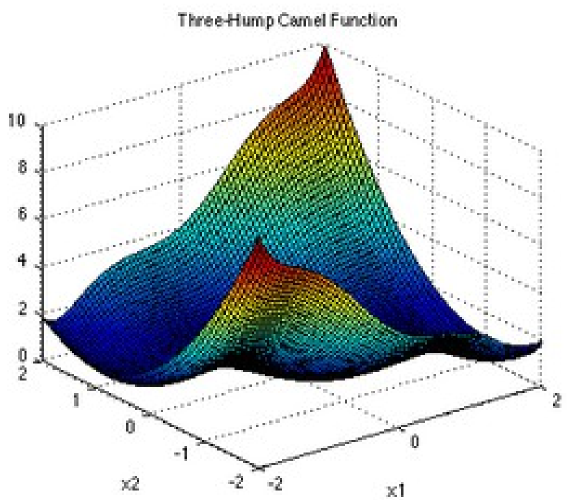
- Paprastai metodams įvertinti kuriamos specialios testinės funkcijos.
  - Kupranugario funkcija;
  - Rastrigino funkicija;
  - Himelblau funkcija;
  - Branino funkcija;
  - Branino ir Hu funkcija;
  - Goldšteino-Praiso funkcija.

Standartinės testinės funkcijos

- **Kupranugario funkcija**

$$f(x_1, x_2) = 2x_1^2 - 1{,}05x_1^4 + \frac{x_1^6}{6} + x_1 x_2 + x_2^2$$





Testiniai uždaviniai

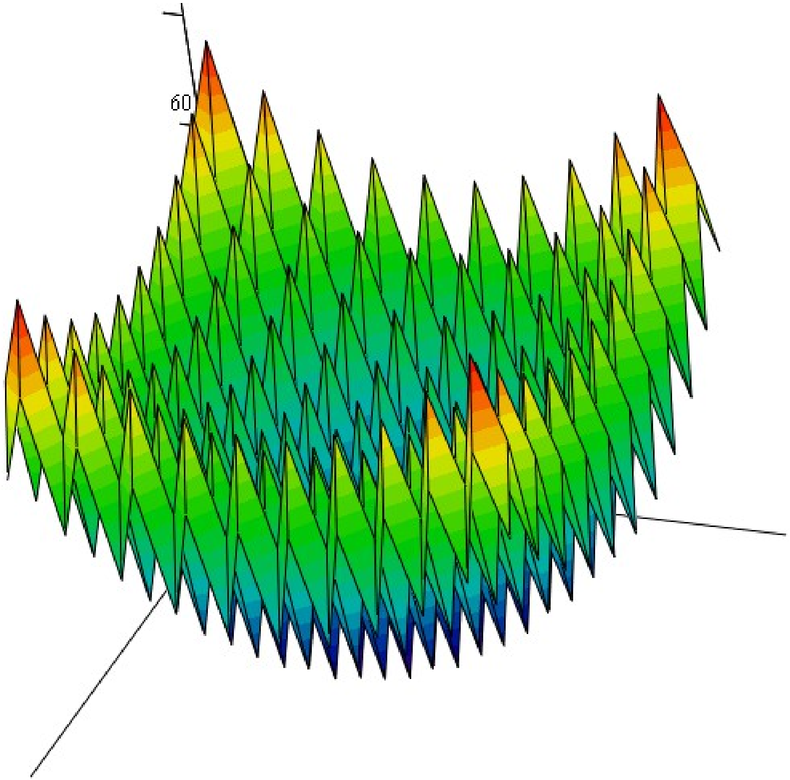
- Tiriant globaliuosius optimizavimo metodus kaip testinė funkcija dažnai naudojama daugiaekstremė **Rastrigino funkcija:**

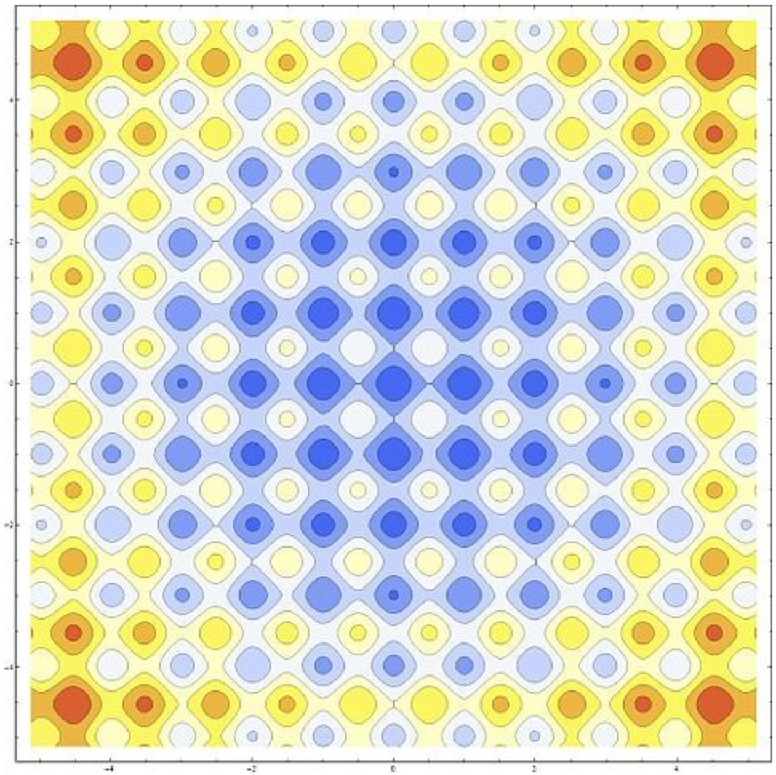
$$f(X) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right), \quad X \in \mathbb{R}^n.$$

Rastrigino funkcija

- Dviejų kintamųjų **Rastrigino funkcija:**

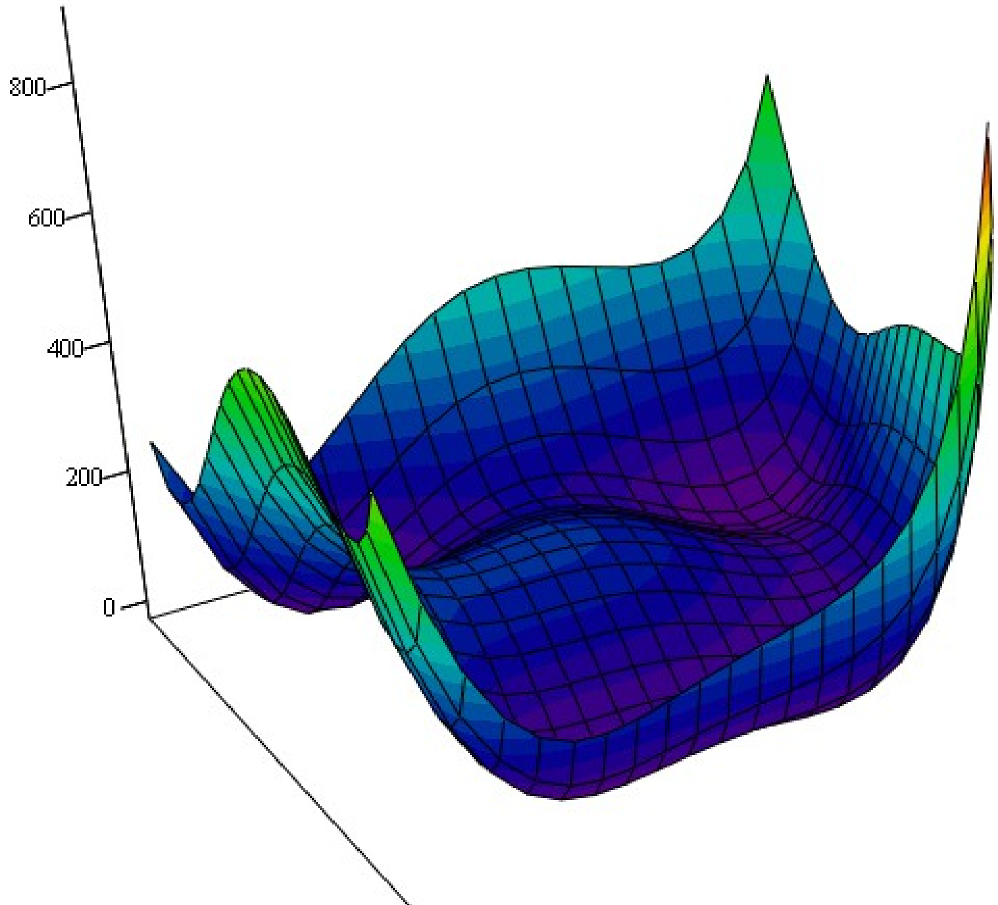
$$f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10\cos(2\pi x_1) - 10\cos(2\pi x_2).$$





- **Himelblau funkcija:**

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$



- **Branino funkcija:**

$$f_{BR}(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4,$$

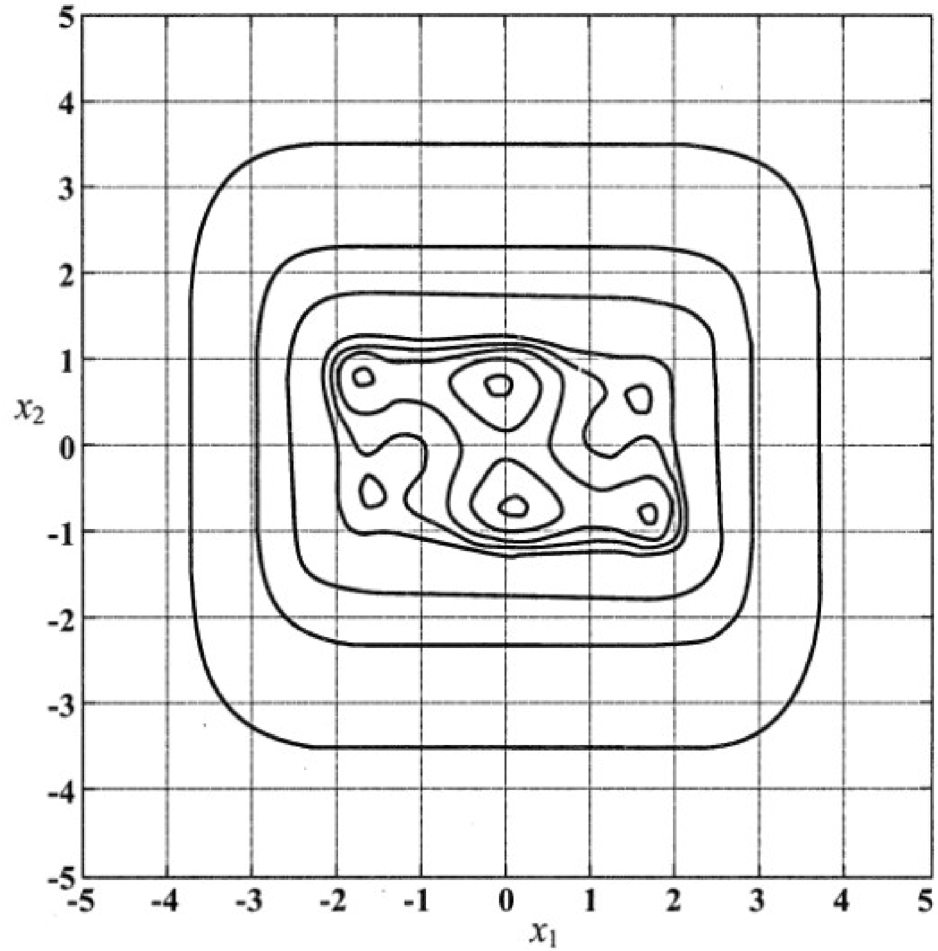
$$-5 \le x_i \le 5, \quad i = 1, 2.$$

Funkcija turi tris poras tarpusavyje lygių lokaliųjų minimumų, kurių reikšmės yra $f_1 = -1{,}0316285$; $f_2 = -0{,}2154$ ir $f_3 = 2{,}1042$.

Globalusis minimumas pasiekiamas dviejuose taškuose:

(0,08983; −0,7126) ir (−0,08983; 0,7126).

- **Branino funkcijos lygio linijos:**



- **Branino ir Hu funkcija:**

$$f_B(X) = (x_2 - \frac{5,1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10\,,$$

$$-5 \le x_1 \le 10\,, \qquad 0 \le x_2 \le 15\,.$$

Globalusis minimumas, lygus 0,398 , pasiekiamas trijuose taškuose:

(-3,142; 12,275), (3,142; 2.275) ir (9,425; 2,425).

Testiniai uždaviniai globaliesiems metodams

- **Goldšteino-Praiso funkcija:**

$$f_{GP}(X) =$$

$$= [1 + (x_1 - x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times$$

$$\times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)],$$

$$-2 \le x_i \le 2, \; i=1, 2.$$

Globalusis minimumas, lygus 3, yra taške (0; -1). Leistinojoje srityje yra iš viso keturi lokalieji minimumai.

Testiniai uždaviniai globaliesiems metodams

- Testinių funkcijų kūrimas GO metodams – sudėtingas procesas. Vizulaliai įvertinti TF savybes įmanoma tik jei matmenų skaičius $n < 3$.

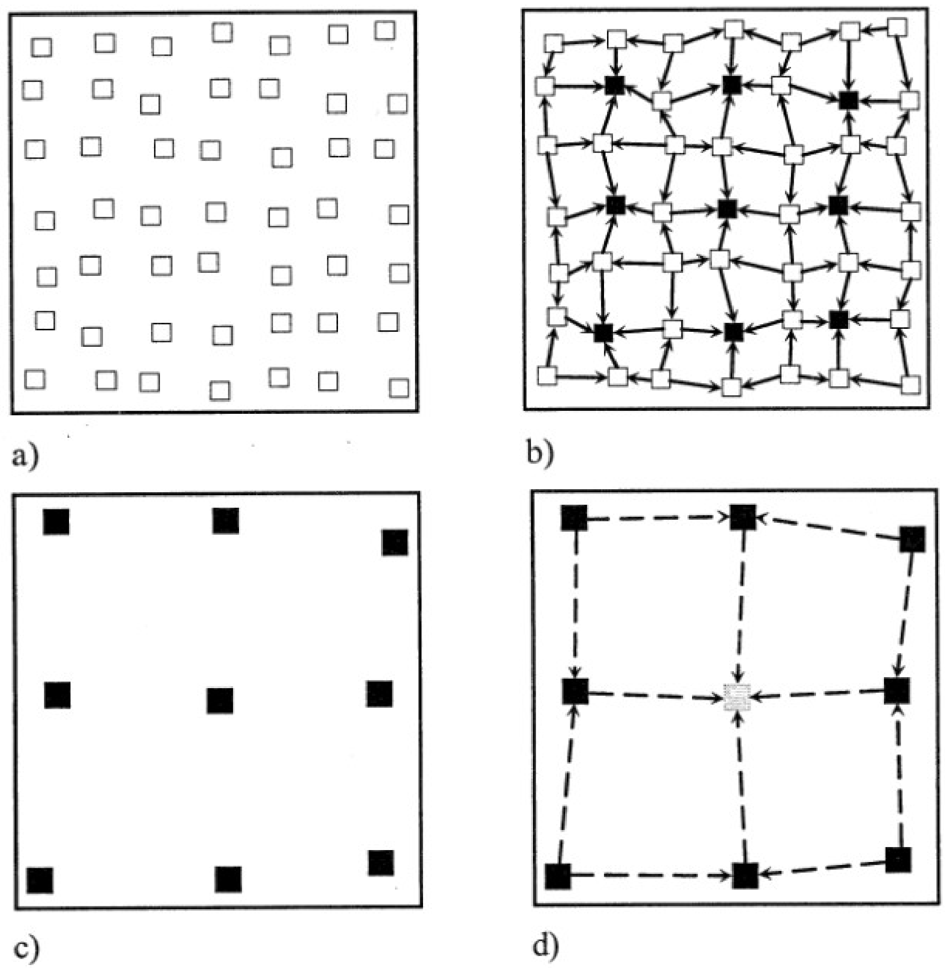
- GO uždavinio sudėtingumas priklauso ne tiek nuo lokaliųjų minimumų skaičiaus, kiek nuo jų išsidėstymo chaotiškumo.

- Galima išskirti lokalijųjų minimumų struktūras.
- Pirmojo lygio minimumai – tai visi lokalieji minimumai.
- Antrojo lygio minimumai išskiriami išanalizavus pirmojo lygio lokalijųjų minimumų kaimynystės santykius. Kaimynai yra tie lokalieji minimumai, kurių traukos zonos liečiasi. Antrojo lygio minimumai yra tie, kurių visi kaimynai didesni.
- Analogiškai iš antrojo lygio išskiriami trečio lygio lokalieji minimumai ir t.t.

- Kuo daugiau lokaliųjų minimumų aukštesniuose lygiuose, tuo sunkesnis darodi optimizavimo uždavinys. Todėl testinėse funkcijuose pageidautina galimybė keisti lokaliųjų minimimų skaičių įvairiuose lygiose.
- Taip galima nustatyti norima testinio uždavinio sunkumą.

Testiniai uždaviniai globaliesiems metodams

•



a) pirmojo lygio lokalieji minimumai;
b) pirmojo lygio kaimyninių lokaliųjų minimumų struktūra;
c) antrojo lygio lokalieji minimumai;
d) antrojo lygio kaimyninių lokaliųjų minimumų struktūra, pilkas kvadratas – vienintelis trečiojo lygio lokalusis minimumas.

Tiesiniai matematiniai modeliai

- Gamybinėje ir paslaugų teikimo srityse svarbi yra turimų išteklių panaudojimo problema. Šiai problemai sėkmingai spręsti gali būti naudojami įvairūs matematiniai metodai.
- Tuo atveju, kai tarp kintamųjų, ir ribotų išteklių bei siekiamų tikslų kriterijų nustatomos tiesinės priklausomybės, taikytini tiesinio programavimo uždavinių sprendimo metodai.

  - Gamybos planavimo uždavinys
  - Transporto uždavinys
  - Dietos uždavinys
  - Darbų paskyrimo uždavinys

Tiesiniai matematiniai modeliai

- Tiesinio programavimo uždavinys yra formuluojamas kaip dalinis bendrojo optimizavimo uždavinio atvejis, jo išraiška yra tokia:

$$z = \sum_{j=1}^{n} c_j x_j \to max \ \left(\text{arba } min\right) \quad - \ \textit{tikslo funkcija}$$

$$\left\{ \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \ \ i = 1, \ldots, m \right. \quad - \ \textit{pagrindiniai apribojimai} \ \left(\leqslant, \geqslant, =\right)$$

$$x_j \geqslant 0, \ \ j = 1, \ldots, n \quad - \ \textit{neneigiamumo reikalavimas}$$

- x_j galima interpretuoti kaip gaminamos produkcijos kiekį (gamybos planas), b_i – turimų išteklių kiekį; čia i – išteklių tipas, j – produkcijos rūšis.

Tiesinio programavimo uždavinių formos

- Atsižvelgiant į tai, kad
  1) TP uždavinio TF galio reikėti maksimizuoti arba minimizuoti,
  2) Pagrindiniuose apribojimuose gali būti lygybės arba nelygybės ($\geq$, $\leq$),
  3) Visiems arba nevisiems kintamiesiems gali būti neneigiamumo reikalavimas,
- TP uždavinius galima užrašyti įvairiomis formomis.

- $\mathbf{K}_{\mathbf{max}}$: maksimizavimo uždavinys kanonine forma.
- $\mathbf{K}_{\mathbf{min}}$: minimizavimo uždavinys kanonine forma.
- $\mathbf{S}_{\mathbf{max}}$: maksimizavimo uždavinys standartine forma.
- $\mathbf{S}_{\mathbf{min}}$: minimizavimo uždavinys standartine forma.
- $\mathbf{B}_{\mathbf{max}}$: bendros formos maksimizavimo uždavinys.
- $\mathbf{B}_{\mathbf{min}}$: bendros formos minimizavimo uždavinys.
- **BF**: bendriausias TP uždavinių pavidalas.

TP maksimizavimo uždavinys kanonine forma

- **K$_{\mathbf{max}}$**

1) TF reikia maksimizuoti;
2) Pagrindiniai apribojimai yra lygybių pavidalo;
3) Visi kintamieji yra neneigiami.

$$max\left(\sum_{j=1}^{n} c_j x_j\right) \qquad - \ tikslo\ funkcija$$

$$\left\{\begin{array}{ll} \sum\limits_{j=1}^{n} a_{ij} x_j = b_i, \ \ i = 1, \ldots, m & - \ pagrindiniai\ apribojimai \\ x_j \geqslant 0, \ \ j = 1, \ldots, n & - \ papildomi\ apribojimai \end{array}\right.$$

TP minimizavimo uždavinys kanonine forma

- **K$_{\min}$**

1) TF reikia minimizuoti;
2) Pagrindiniai apribojimai yra lygybių pavidalo;
3) Visi kintamieji yra neneigiami.

$$\min\left(\sum_{j=1}^{n} c_j x_j\right) \qquad - \ \textit{tikslo funkcija}$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, \ldots, m & - \ \textit{pagrindiniai apribojimai} \\ x_j \geqslant 0, \quad j = 1, \ldots, n & - \ \textit{papildomi apribojimai} \end{cases}$$

TP maksimizavimo uždavinys standartine forma

- **S**$_{\mathbf{max}}$

1) TF reikia maksimizuoti;
2) Pagrindiniai apribojimai yra nelygybių ≤ pavidalo;
3) Visi kintamieji yra neneigiami.

$$max\left(\sum_{j=1}^{n} c_j x_j\right) \qquad - \ tikslo\ \ funkcija$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = 1, \dots, m & - \ pagrindiniai\ \ apribojimai \\ x_j \geqslant 0, \quad j = 1, \dots, n & - \ papildomi\ \ apribojimai \end{cases}$$

TP minimizavimo uždavinys standartine forma

- $\mathbf{S}_{\mathbf{min}}$

1) TF reikia minimizuoti;
2) Pagrindiniai apribojimai yra nelygybių ≥ pavidalo;
3) Visi kintamieji yra neneigiami.

$$min\left(\sum_{j=1}^{n} c_j x_j\right) \qquad - \ tikslo \ \ funkcija$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \geqslant b_i, \quad i = 1, \dots, m & - \ pagrindiniai \ \ apribojimai \\ x_j \geqslant 0, \quad j = 1, \dots, n & - \ papildomi \ \ apribojimai \end{cases}$$

Bendros formos maksimizavimo uždavinys

- $\mathbf{B}_{\mathbf{max}}$

1) TF reikia maksimizuoti;
2) Pagrindiniai apribojimai yra lygybių ir nelygybių ≤ pavidalo;
3) Kintamieji nebūtinai yra neneigiami.

$$max\left(\sum_{j=1}^{n} c_j x_j\right)$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, & i = 1, \dots, m_1 \\ \sum_{j=1}^{n} a_{ij} x_j = b_i, & i = m_1 + 1, \dots, m \end{cases}$$

$$x_j \geqslant 0, \quad j = 1, \dots, n_1,$$

$$x_j \in R, \quad j = n_1 + 1, \dots, n.$$

- $\mathbf{S}_{\mathbf{max}}$ yra atskiras $\mathbf{B}_{\mathbf{max}}$ atvejis, kai $m_1 = m$ ir $n_1 = n$.
- $\mathrm{K}_{\mathbf{max}}$ yra atskiras $\mathbf{B}_{\mathbf{max}}$ atvejis, kai $m_1 = 0$ ir $n_1 = n$.

Bendros formos minimizavimo uždavinys

- $\mathbf{B}_{\mathbf{min}}$

1) TF reikia maksimizuoti;
2) Pagrindiniai apribojimai yra lygybių ir nelygybių ≥ pavidalo;
3) Kintamieji nebūtinai yra neneigiami.

$$min\left(\sum_{j=1}^{n} c_j x_j\right)$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \geqslant b_i, \quad i = 1, \dots, m_1 \\ \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = m_1 + 1, \dots, m \end{cases}$$

$$x_j \geqslant 0, \quad j = 1, \dots, n_1,$$

$$x_j \in R, \quad j = n_1 + 1, \dots, n.$$

- $\mathbf{S}_{\mathbf{min}}$ yra atskiras $\mathbf{B}_{\mathbf{min}}$ atvejis, kai $m_1 = m$ ir $n_1 = n$.
- $\mathrm{K}_{\mathbf{min}}$ yra atskiras $\mathbf{B}_{\mathbf{min}}$ atvejis, kai $m_1 = 0$ ir $n_1 = n$.

Bendriausias TP uždavinio pavidalas

- **B**$_{\mathbf{min}}$

1) TF reikia maksimizuoti arba minimizuoti;
2) Pagrindiniai apribojimai yra lygybių ir nelygybių pavidalo;
3) Kintamieji nebūtinai yra neneigiami.

$$\begin{Bmatrix} min \\ max \end{Bmatrix} \left(\sum_{j=1}^{n} c_j x_j \right)$$

$$\begin{cases} \sum\limits_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = 1, \ldots, m_1 \\ \sum\limits_{j=1}^{n} a_{ij} x_j \geqslant b_i, \quad i = m_1 + 1, \ldots, m_2 \\ \sum\limits_{j=1}^{n} a_{ij} x_j = b_i, \quad i = m_2 + 1, \ldots, m \end{cases}$$

$$x_j \geqslant 0, \quad j = 1, \ldots, n_1,$$
$$x_j \in R, \quad j = n_1 + 1, \ldots, n.$$

TP uždavinių formų transformacijos

- Tarp formų yra ryšiai, t. y. nuo vienos formos galima pereiti prie kitos.

1) Norint pakeisti nelygybės ženklą, reikia jos abi puses padauginti iš neigiamo skaičiaus.
2) Norint iš nelygybės $\leq$ gauti lygybę, reikia prie nelygybės kairiosios pusės pridėti **neneigiamą** papildomą kintamąjį.
3) Norint iš nelygybės $\geq$ gauti lygybę, reikia iš nelygybės kairiosios pusės atimti **neneigiamą** papildomą kintamąjį.
4) Norint iš lygybės gauti nelygybę $\leq$ arba $\geq$, reikia lygybę parašyti kaip dviejų nelygybių sistemą ir po to vieną nelygybę padauginti iš neigiamo skaičiaus.
5) Jeigi kintamajam nėra neneigiamumo reikalavimo, tai jį galima užrašyti, kaip dviejų neneigiamų kintamųjų skirtumą.
6) Minimizavimo uždavinį galima suvesti į maksimizavimo (iš atvirkščiai), pasinaudojus lygybe $\min z = -\max(-z)$.

Geometrinis TP uždavinių sprendimo metodas

- Geometriškai galima spręsti TP uždavinius dviem atvejais:

1) n =2.
2) Uždavinys duotas kanonine forma ir $n - r = 2$. Čia n – nežinomųjų skaičius, r - lygčių sistemos rangas.

- Pirmuoju atveju uždavinys gali buti duotas bet kokiame pavidale. Tarkime, BF_{max}:

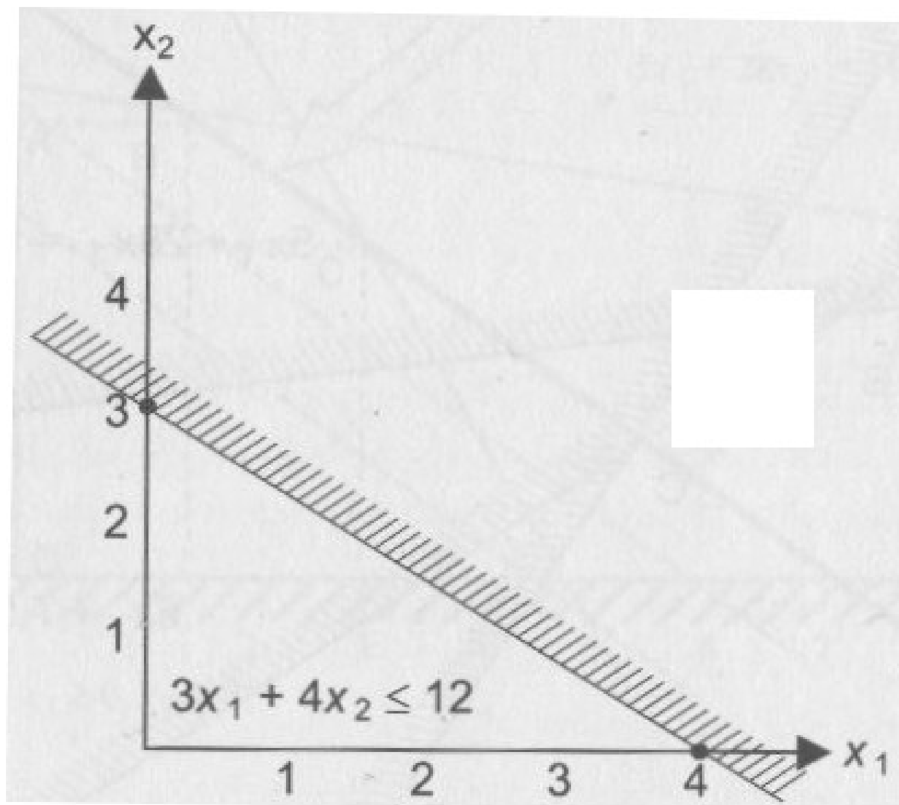
$$max\left(\sum_{j=1}^{2} c_j x_j\right)$$

$$\begin{cases} \sum_{j=1}^{2} a_{ij} x_j \leqslant b_i, \quad i = 1, \dots, m_1 \\ \sum_{j=1}^{2} a_{ij} x_j \geqslant b_i, \quad i = m_1 + 1, \dots, m_2 \\ \sum_{j=1}^{2} a_{ij} x_j = b_i, \quad i = m_2 + 1, \dots, m \end{cases}$$

$$x_j \geqslant 0, \quad j \in J \subset \{1, 2\}$$

Geometrinis TP uždavinių sprendimo metodas

- Išspresti šį uždavinį geometriškai, vadinasi, rasti tokį apribojimų srities tašką, kurio koordinatės suteiktų TF maksimalią reikšmę.
- Lygybės $a_{i1}\, x_1 + a_{i2}\, x_2 = b_i$ geometrinis vaizdas yra tiesė plokštumoje x_1Ox_2 . Pvz., $3x_1 + 4x_2 = 12$.



Geometrinis TP uždavinių sprendimo metodas

- Nelygybė $a_{i1}\, x_1 + a_{i2}\, x_2 \le b_i$ (arba $a_{i1}\, x_1 + a_{i2}\, x_2 \ge b_i$) geometriškai reiškia pusplokštumę, kurios kontūras yra tiesė $a_{i1}\, x_1 + a_{i2}\, x_2 = b_i$.

- Pusplokštumei nustatyti parenkamas bet koks taškas, nepriklausantys šiai tiesei. Jei taško koordinatės tenkina nelygybę , tau duotoji nelygybė nusako ta pusplokštumę, kurioje yra pasirinktas taškas, o priešingu atveju – kitą pusplokštumę.

- Apribojimų sistemos visos nelygybės geometriškai reiškia pusplokštumių bendrą dalį (sankirtą) – apribojimų sritį. Apribojimų sritis gali būti vieno iš keturių pavidalų:

a) begalinė neaprėžta sritis;

b) iškilus daugiakampis;

c) taškas;

d) tuščia aibė.

TP uždavinio geometrinio sprendimo etapai

1) iš apribojimų nustatoma apribojimų sritis;
2) randami srities kraštiniai taškai;
3) brėžiama atraminė tiesė $c_1 x_1 + c_2 x_2 = 0$ ir nustatoma tikslo funkcijos didėjimo kryptis, t.y. brėžiamas vektorius (c_1, c_2);
4) randamas optimalus sprendinys ir optimali tikslo funkcijos reikšmė.

- Geometriškai sprendžiant tiesinio programavimo uždavinius, galimi tokie pagrindiniai atvejai:

a) optimalus sprendinys vienintelis – atramos tiesė eina per vieną apribojimų srities kraštinį tašką;
b) optimalių sprendinių be galo daug – atraminė tiesė eina per du kraštinius taškus;
c) optimalių sprendinių nėra, atramine tiesė $\rightarrow \infty$ - apribojimų sritis neapibrėžta;
d) optimalių sprendinių nėra ir apribojimų sritis – tuščia aibė, t.y. apribojimų sistema nesuderinta.

Modifikuotas Žordano-Gauso metodas tiesinių lygčių sistemoms spręsti

- Tegul duotas TP uždavinys $\mathbf{K}_{\mathbf{max}}$ forma su pagringinių apribojimų matrica

$$\begin{cases} a_{11}x_1 & + a_{12}x_2 & + \dots & + a_{1\mathrm{n}}x_n & = & b_1 \\ a_{21}x_1 & + a_{22}x_2 & + \dots & + a_{2\mathrm{n}}x_n & = & b_2 \\ a_{31}x_1 & + a_{32}x_2 & + \dots & + a_{3\mathrm{n}}x_n & = & b_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1}x_1 & + a_{m2}x_2 & + \dots & + a_{mn}x_n & = & b_m \end{cases}$$

- Tarkime, lygčių sistemos rangas yra r.
- Bazinių kintamųjų bus r ir laisvųjų kintamųjų $n - r$.

Modofikuotas Žordano-Gauso metodas tiesinių lygčių sistemoms spręsti

- Sprendžiant sistemą MGŽ metodu, vienoje lygtyje (nebūtinai pirmoje, kaip Gauso metode) paliekamas narys su x_1 , panaikinant narius su x_1 kitose lygtyse, kitoje lygtyje paliekamas narys su x_2 , panaikinant narius su x_2 kitose lygtyse, ir t.t.
- Sistema užrašoma lentelės pavidalu. Į lentelė surašomi sistemos koeficientai, laisvieji nariai ir Σ – kontrolinis stulpelis.

$$\sigma_i = \sum_{j=1}^{n} a_{ij} + b_i, \quad i=1, \dots, m.$$

x_1	x_2	...	x_n	B	Σ
a_{11}	a_{12}	...	a_{1n}	b_1	σ_1
a_{21}	a_{22}	...	a_{2n}	b_2	σ_2
...	...	...	...	...	...
a_{m1}	a_{m2}	...	a_{mn}	b_m	σ_m

Modifikuotas Žordano-Gauso metodas tiesinių lygčių sistemoms spręsti

- Po to parenkamas nelygus nuliui koeficientas, vietoje kurio norima gauti vienetą. Jis vadinamas **pagrindiniu elementu** ir lentelėje išryškinamas. Vienetą kiekvienoje eilutėje (nebūtinai įstrižainėje) galima gauti ir dalybos būdu.
- Toliau panaudodami pagrindinį elementą ir sistemos pertvarkį, kai sudedama bet kuri lygtis ir kita lygtis, padauginta iš nelygaus nuliui skaičiaus, gauname nulius ne tik žemiau, bet ir aukščiau vieneto.
- Kiti naujos sistemos elementai, neesantys pagrindinio elemento eilutėje ir stulpelyje, gaunami pagal **stačiakampio taisyklę**: jei pagrindinis elementas a_{ij}, tai naujasis elementas a'_{ij} , naujasis laisvasis narys b'_j ir naujasis kontrolinis elementas σ'_j skaičiuojami taip:

$$a'_{lk} = \frac{a_{ij}a_{lj} - a_{lj}a_{ik}}{a_{ij}}, \quad b'_l = \frac{a_{ij}b_l - a_{lj}b_i}{a_{ij}}, \quad \sigma'_l = \frac{a_{ij}\sigma_l - a_{lj}\sigma_i}{a_{ij}}$$

Įrodymas. Įrodysime tik pirmąją lygybę (kitos įrodomos analogiškai). Nagrinėkime pertvarkomos lentelės fragmentą

…	x_j…	…	x_k	**B**	**Σ**
…	…	…	…	…	…
…	a_{ij}	…	a_{ik}	b_i	σ_i
…	…	…	…	…	…
…	a_{lj}	…	a_{lk}	b_l	σ_l
…	…	…	…	…	…

Norint vietoje a_{ij} gauti nulį, reikia i-tąją eilutę padauginti iš (- a_{lj} / a_{ij}) ir gautus rezultatus pridėti prie l-tosios eilutės. Tuomet vietoje a_{lk} bus

$$a'_{lk} = a_{ik}\left(-\frac{a_{lj}}{a_{ij}}\right) + a_{lk} = \frac{a_{ij}a_{lj} - a_{lj}a_{ik}}{a_{ij}}.$$

Modifikuotas Žordano-Gauso metodas tiesinių lygčių sistemoms spręsti

- Sprendžiant tiesinių lygčių sistemą modifikuotu Gauso ir Žordano nuo vienos lentelės (sistemos) prie kitos pereinama pagal šias taisykles:

1) Tarp koeficientų prie nežinomųjų parenkamas bet kuris nelygus nuliui elementas, vadinamas pagrindiniu (vedančiuoju) elementu, $a_{ij} \neq 0$. Eilutė, kurioje yra vedantysis elementas, vadinama vedančiąja eilute, o stulpelis – vedančiuoju stulpeliu.
2) Vedančiosios eilutės elementus dalinome iš vedančiojo elemento ir užrašome naujoje lentelėje. Vedančiojo stulpelio vietoje naujoje lentelėje rašome nulius.
3) Visus kitus naujos lentelės elementus skaičiuojame pagal stačiakampio taisyklę
4) Apskaičiuojami naujos lentelės kontrolės stulpelio Σ elementai:

$$\sigma_i' = \sum_{j=1}^{n} a_{ij}' + b_i', \quad i = 1, \dots, m.$$

Žordano-Gauso metodas tiesinių lygčių sistemoms spręsti

- Skaičiavimai bus atlikti teisingai, jeigu elementai, apskaičiuoti pagal stačiakampio taisyklę, sutampa su gautais sudedant eilutės elementus.
- Toliau naujoje lentelėje vedantį elementą parenkame iš kitos eilutės ir pagal nurodytas keturias taisykles pereiname prie sekančios lentelės ir t.t. , kol vedantysis elementas bus parinktas kiekvienoje eilutėje, arba gausime, kad sistema nesuderinta (neturi sprendinio). Sistemos nesuderinamumo požymis – tokio pavidalo eilutė:

$$0 \;\; 0 \;\; \ldots \;\; 0 \;\; b_l \;\; \sigma_l \qquad \left(b_l \neq 0\right).$$

- Čia užrašyti nuliai reiškia nežinomųjų nulinius koeficientus, o b_l reiškia lygties laisvojo nario nenulinę reikšmę. Pilnai nulinė eilutė iš lentelės pašaliname.

Žordano-Gauso metodas tiesinių lygčių sistemoms spręsti

$$\begin{cases} 2x_1 - x_2 - x_3 = 5 \\ 3x_1 - 2x_2 + x_3 = 4 \\ x_1 - x_2 - 3x_3 = 4 \end{cases}$$

x_1	x_2	x_3	B	Σ
2	-1	-1	5	5
3	-2	1	4	6
1	-1	-3	4	1
0	**1**	5	-3	3
0	1	10	-8	3
1	-1	-3	4	1
0	1	5	-3	3
0	0	**5**	-5	0
1	0	2	1	4
0	1	0	2	3
0	0	1	-1	0
1	0	0	3	4

Žordano-Gauso metodas tiesinių lygčių sistemoms spręsti

$$\begin{cases} 2x_1 + x_2 + 3x_3 = 13 \\ x_1 - x_2 - 2x_3 = -7 \\ 3x_1 + 3x_2 - x_3 = 6 \end{cases}$$

Žordano-Gauso metodas tiesinių lygčių sistemoms spręsti

$$\begin{cases} 2x_1 + x_2 + 3x_3 = 13 \\ x_1 - x_2 - 2x_3 = -7 \\ 3x_1 + 3x_2 - x_3 = 6 \end{cases}$$

x_1	x_2	x_3	B	Σ
2	1	3	13	19
1	-1	-2	-7	-9
3	3	-1	6	11
0	3	7	27	37
1	-1	-2	-7	-9
0	6	5	27	38
0	1	7/3	9	37/3
1	0	1/3	2	10/3
0	0	-9	-27	-36
0	1	0	2	3
1	0	0	1	1
0	0	1	3	4

Taigi lygčių sistemos sprendinys yra (1; 2; 3).

Neapibrėžtųjų sistemų sprendimas modifikuotu Gauso ir Žordano metodu

- Sprendžiant bendrąją tiesinių lygčių sistemą, kurios lygčių skaičius nebūtinai lygus nežinomųjų skaičiui,

$$\begin{cases} a_{11}x_1 & + a_{12}x_2 & + \ldots & + a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + a_{22}x_2 & + \ldots & + a_{2n}x_n & = & b_2 \\ a_{31}x_1 & + a_{32}x_2 & + \ldots & + a_{3n}x_n & = & b_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{m1}x_1 & + a_{m2}x_2 & + \ldots & + a_{mn}x_n & = & b_m \end{cases}$$

galima sudaryti dvi matricas: sistemos koeficientų matricą A ir išplėstinę matricą A'.

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}, \quad A' = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} & b_m \end{pmatrix}.$$

- Išplėstines matricos rangas lygus $r(A)$ arba $r(A) + 1$.

Neapibrėžtųjų sistemų sprendimas modifikuotu Gauso ir Žordano metodu

- **Kronekerio ir Kapelio teorema**. Tiesinių lygčių sistema suderinta tada ir tik tada, kada išplėstinės matricos rangas lygus sistemos matricos rangui: $r(A') = r(A) = r$.

- Šios teoremos sąlyga galima patikrinti ir neieškant matricų A ir A' rangų.

- Jei išsprendus sistemą MGŽ metodu negaunama eilutė

$$0 \quad 0 \quad \dots \quad 0 \quad b_l \quad \sigma_l \qquad (b_l \neq 0).$$

tai sistema yra **suderinta**, o matricos ir išplėstinės matricos rangas r lygus nenulinių eilučių, esančių paskutinėje lentelėje skaičiui.

- Jei $r = n$ (rangas lygus sistemos nežinomųjų skaičiui), tai lygčių sistema **turi vienintelį sprendinį**.

- Jei $r < n$, tai sistema turi be galo daug sprendinių (yra **neapibrėžtoji**).

Neapibrėžtųjų sistemų sprendimas modifikuotu Gauso ir Žordano metodu

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 6 \\ 2x_1 - x_2 + 2x_4 = 0 \\ 3x_1 + x_2 - x_3 - x_4 + 3x_5 = 2 \end{cases}$$

x_1	x_2	x_3	x_4	x_5	B	Σ
1	1	1	1	1	6	11
2	-1	0	2	0	0	3
3	**1**	-1	-1	3	2	7
-2	0	2	2	-2	4	4
5	0	-1	**1**	3	2	10
3	1	-1	-1	3	2	7
-12	0	**4**	0	-8	0	-16
5	0	-1	1	3	2	10
8	1	-2	0	6	4	17
-3	0	1	0	-2	0	-4
2	0	0	1	**1**	2	6
2	1	0	0	2	4	9

Neapibrėžtųjų sistemų sprendimas modifikuotu Gauso ir Žordano metodu

- Iš paskutinės lentelės matome, kad lygčių sistemos rangas lygus 3 ir tiek nežinomųjų galima išreikšti likusiais dviem nežinomaisiais:

$$\begin{cases} x_3 = & & 3x_1 & + 2x_5 \\ x_4 = & 2 & - 2x_1 & - x_5 \\ x_2 = & 4 & - 2x_1 & - 2x_5 \end{cases}$$

- Nežinomieji x_1 ir x_5 gali įgyti bet kurias realiąsias reikšmes, todėl jie vadinami **laisvaisiais** nežinomaisiais, o likusieji x_2 , x_3 , x_4 – **baziniais**.
- Kadangi laisvieji nežinomieji gali įgyti bet kurias reikšmes, tai tokia sistema turi be galo daug sprendinių. Tačiau iš jų visų ypač svarbūs vadinamieji **baziniai sprendiniai**, gaunami laisviesiems nežinomiesiems suteikus nulines reikšmes.

- Jei šio pavyzdžio laisvieji nežinomieji x_1 ir x_5 lygus nuliui, tai bazinių nežinomųjų reikšmės: $x_2 = 4$, $x_3 = 0$, $x_4 = 2$. Taigi pirmasis bazinis sprendinys – (0, 4, 0, 2, 0).

Neapibrėžtųjų sistemų sprendimas modifikuotu Gauso ir Žordano metodu

- Norint gauti kitą bazinį sprendinį, reikia parinkti pagrindinį elementą paskutinės lentelės pirmajame arba penktajame (t.y. laisvųjų nežinomųjų) stulpelyje.
- Parinkę vienetą penktajame stulpelyje, gausime tokią lentelę:

x_1	x_2	x_3	x_4	x_5	**B**	**Σ**
1	0	1	2	0	4	8
2	0	0	1	1	2	6
-2	1	0	-2	0	0	-3

- Iš šios lentelės matome, kad laisvieji nežinomieji - x_1 ir x_4 ; baziniai nežinomieji - x_2 , x_3 , x_5 , antrasis bazinis sprendinys – (0, 0, 4, 0, 2).
- Šiame pavyzdyje gali būti 10 laisvųjų nežinomųjų porų. Todėl daugiausia galima gauti 10 bazinių sprendinių. Bendruoju atveju, kai lygčių sistemos nežinomųjų skaičius n, o jos rangas r, bazinių sprendinių skaičius neviršija dydžio

$$C_n^r = \frac{n!}{r!(n-r)!}$$

Neneigiamieji baziniai sprendiniai

- Praktiniams lygčių sistemų taikymams svarbūs **neneigiamieji baziniai sprendiniai**, t.y. tokie baziniai sprendiniai, kurių kiekvieno nežinomo reikšmė yra neneigiama.

- Neneigiamieji baziniai sprendiniai ieškomi taip:

1) lygtys, kurių laisvieji nariai yra neigiami, dauginamos iš (-1), nes laisvųjų narių stulpelio B elementai turi būti neneigiami;
2) pagrindiniu elementu renkamas tik teigiamas skaičius. Jei j-tajame stulpelyje yra keli teigiami skaičiai a_{ij}, tai jiems sudaromi santykiai $b_i \,/\, a_{ij}$ ir pagrindiniu elementu renkamas mažiausio santykio vardiklis.

- Taip parinkus pagrindinį elementą ir pertvarkant sistemą MGŽ metodu, laisvųjų narių stulpelyje gaunami tik neneigiami skaičiai. Jei stulpelyje nėra teigiamų skaičių, tai pagrindinis elementas tokiame stulpelyje nerenkamas.

- Kitus neneigiamus bazinius sprendinius gauname ta pačia perėjimo tvarka.

Neneigiamieji baziniai sprendiniai

- Rasime tiesinių lygčių sistemos

$$\begin{cases} x_1 + 2x_2 + 3x_3 - 4x_4 + x_5 = -2 \\ 3x_1 + x_2 + 4x_3 + 2x_4 - 3x_5 = 5 \\ \quad + 2x_2 - 3x_3 - x_4 + 2x_5 = -3 \end{cases}$$

du neneigiamuosius bazinius sprendinius.

Neneigiamieji baziniai sprendiniai

- Stulpelio *B* elementai turi būti neneigiami skaičiai, todėl pirmoji lentelė atrodo taip:

x_1	x_2	x_3	x_4	x_5	B	Σ
-1	-2	-3	4	-1	2	-1
3	1	4	2	-3	5	12
0	-2	3	1	-2	3	3

- Pagrindinių elementu parenkame vienintelį antrojo stulpelio teigiama skaičių 1. Gauname tokią lentelę:

x_1	x_2	x_3	x_4	x_5	B	Σ
5	0	5	8	-7	12	23
3	1	4	2	-3	5	12
6	0	11	5	-8	13	27

Neneigiamieji baziniai sprendiniai

- Šios lentelės trečiajame stulpelyje yra trys teigiami skaičiai, todėl sudarome tris santykius:

$$\frac{12}{5} = 2{,}4\,; \quad \frac{5}{4} = 1{,}25\,; \quad \frac{13}{11} \approx 1{,}18\,.$$

- Pagrindinis elementas – mažiausio santykio vardiklis 11. Gauname naują lentelę:

x_1	x_2	x_3	x_4	x_5	**B**	**Σ**
25/11	0	0	**63/11**	-37/11	67/11	118/11
9/11	1	0	2/11	-1/11	3/11	24/11
6/11	0	1	5/11	-8/11	13/11	27/11

Neneigiamieji baziniai sprendiniai

- Šioje lentelėje pagrindiniu imame ketvirtojo stulpelio elementą 63/11, nes šis skaičius yra mažesnio santykio

$$\frac{67/11}{63/11} \approx 1{,}06\,; \quad \frac{3/11}{2/11} = 1{,}5\,; \quad \frac{13/11}{5/11} = 2{,}6\,.$$

vardiklis. Skaičiuodami užpildome lentelę:

x_1	x_2	x_3	x_4	x_5	**B**	**Σ**
25/63	0	0	1	-37/63	67/63	118/63
47/63	1	0	0	**1/63**	5/63	116/63
23/63	0	1	0	-29/63	44/63	101/11

Neneigiamieji baziniai sprendiniai

- Iš šios lentelės užrašome pirmą neneigiamąjį bazinį sprendinį:

$$x_1 = 0; \quad x_2 = \frac{5}{63}; \quad x_3 = \frac{44}{63}; \quad x_4 = \frac{67}{63}; \quad x_5 = 0.$$

- Norėdami gauti antrą neneigiamąjį bazinį sprendinį, pagrindiniu elementu parinkime 1/63. Tuomet gausime tokią lentelę:

x_1	x_2	x_3	x_4	x_5	**B**	**Σ**
28	37	0	1	0	4	70
47	63	0	0	1	5	116
22	29	1	0	0	3	55

- Antras neneigiamas sprendinys bus toks:

$$x_1 = 0; \quad x_2 = 0; \quad x_3 = 3; \quad x_4 = 4; \quad x_5 = 5.$$

- Norėdami dar gauti trečią neneigiamąjį bazinį sprendinį, turėtume pagrindiniu elementu parinkti 47 arba 63.

Neneigiamieji baziniai sprendiniai

- Rasime tiesinių lygčių sistemos

$$\begin{cases} x_1 - 2x_2 + x_4 = -3 \\ 3x_1 - x_2 - 2x_3 = 1 \\ 2x_1 + x_2 - 2x_3 - x_4 = 4 \\ x_1 + 3x_2 - 2x_3 - 2x_4 = 7 \end{cases}$$

visus bazinius sprendinius ir visus neneigiamus bazinius sprendinius.

Simplekso metodas

- Kai tiesinio programavimo uždavinio kintamųjų skaičius yra didelis, grafinio sprendimo metodą taikyti negalime.

- Tokių uždavinių sprendimui taikomas **analitinis** sprendimo metodas.

- Analitinio tiesinio programavimo uždavinio sprendimo idėja yra tokia pati kaip ir grafinio: **nuosekliai peržiūrimos leistinų sprendinių srities viršūnės**, vienoje iš kurių randamas optimalus sprendinys.

- Tiesinio programavimo uždavinių analitiniam sprendimui buvo sukurtas specialus kryptingo viršūnių peržiūrėjimo algoritmas. Pagal šį algoritmą peržiūros metu nuo vienos viršūnės iki kitos einama tokia kryptimi, kuria tikslo funkcijos reikšmė artėja prie optimalios.

- Analitinis tiesinio programavimo uždavinių sprendimo metodas vadinamas **simplekso metodu.**

Simplekso metodas

- **Simpleksas** yra geometrinė sąvoka, kuria apibūdinamas paprasčiausias tam tikro matavimų skaičiaus daugiasienis.

- Kūno simpleksu *n*-matėje erdvėje vadinama *n*+1 jo viršūnių aibė. Pavyzdžiui, plokštumoje, simpleksas bus trys trikampio viršūnės, trimatėje erdvėje – keturios keturbriaunio viršūnės, ir t.t.

- Perėjimas nuo vienos viršūnės prie kitos vadinamas iteracija.

Simplekso metodas

- Simplekso metodą galima taikyti, kai patenkintos šios 3 sąlygos:

1) Uždavinys duotas forma K_{max} .
2) Apribojimų sistemos laisvieji nariai yra neneigiami.
3) Žinomas aprobojimų sistemos neneigiamas bazinis sprendinys.

$$max\left(\sum_{j=1}^{n} c_j x_j\right) \qquad - \ tikslo\ \ funkcija$$

$$\begin{cases} \displaystyle\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, \ldots, m & - \ pagrindiniai\ \ apribojimai \\ x_j \geqslant 0, \quad j = 1, \ldots, n & - \ papildomi\ \ apribojimai \end{cases}$$

Simplekso metodas

- Laikykime, kad lygčių sistemos rangas yra r ir $x_1, \ldots, x_r$ – baziniai kintamieji.
- Tuomet iš pagrindinių apribojimų bazinius kintamuosius išreiškiame laisvaisiais:

$$\left\{ x_i = b_i' - \sum_{j=r+1}^{n} a_{ij}' x_j \ , \quad i = 1, \ldots, r \right.$$

- TF taip pat išreiškiame laisvaisiais kintamaisiais:

$$z = \sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{r} c_j \left(b_i' - \sum_{k=r+1}^{n} a_{ik}' x_k \right) + \sum_{j=r+1}^{n} c_j x_j = \gamma_0 - \sum_{j=r+1}^{n} \gamma_j x_j$$

- Koeficientai γ prie laisvųjų kintamųjų vadinami **laisvųjų kintamųjų įvertinimais**.

Simplekso metodas

- **Teorema**. Jei K_{max} TP uždavinys turi vienintelį optimalų sprendinį, tai jis yra apribojimų sistemos neneigiamas bazinis sprendinys. Jei optimalių sprędinių yra be galo daug, tai vienas iš jų yra **neneigiamas bazinis spredinys** (NBS).

- Išskant NBS GŽ metodu, nuo vieno BS prie kito pereinama netvarkingai.

- Simplekso metodo esmė – nuo vieno NBS pereinama prie geresnio NBS, tokio, kurį atitinkanti TF reikšmė yra didesnė.

Simplekso metodas

- Bazinių kintamųjų ir TF išraiškas laisvaisias kintamaisias užrašome lentelės, vadinamos simplekso lentelės pavidalu

BK	C_{BK}	B	c_1	c_2	...	c_p	...	c_r	c_{r+1}	c_{r+2}	...	c_q	...	c_n	Σc_j
			x_1	x_2	...	x_p	...	x_r	x_{r+1}	x_{r+2}	...	x_q	...	x_n	Σ
x_1	c_1	b'_1	1	0	...	0	...	0	$a'_{1,r+1}$	$a'_{1,r+2}$	...	a'_{1q}	...	a'_{1n}	σ'_1
x_2	c_2	b'_2	0	1	...	0	...	0	$a'_{2,r+1}$	$a'_{2,r+2}$	...	a'_{2q}	...	a'_{2n}	σ'_2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
x_p	c_p	b'_p	0	0	...	1	...	0	$a'_{p,r+1}$	$a'_{p,r+2}$	...	a'_{pq}	...	a'_{pn}	σ'_p
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
x_r	c_r	b'_r	0	0	...	0	...	1	$a'_{r,r+1}$	$a'_{r,r+2}$	...	a'_{rq}	...	a'_{rn}	σ'_r
m+1	-	γ_0	0	0	...	0	...	0	γ_{r+1}	γ_{r+2}	...	γ_q	...	γ_n	σ'_{m+1}

Simplekso metodas

- Lentelės paskutinėje eilutėje (m+1) atsispindi TF išraiška laisvaisias kinatamaisias.

$$z = \gamma_0 - \gamma_{r+1} x_{r+1} - \gamma_{r+2} x_{r+2} - \dots - \gamma_q x_q - \dots - \gamma_n x_n .$$

- Koeficientus γ lengva gauti iš simplekso lentelės stulpelių.
- γ_0 gauname sudauginus atitinkamus II ir III stulpelio skaičius ir sudedame.
- γ_{r+1} gavimui reikia sudauginti atitinkamus II stulpelio ir x_{r+1} stulpelio elementus, sandaugas sudėti, o po to dar atimti c_{r+1} .
- Analogiškai gaunami kiti laisvųjų kintamųjų įvertinimai.

Simplekso metodas

- Iš užpildytos lentelės matome ne tik pradinį neneigiamą bazinį įvertį (III stulpelis) ir atitinkamą TF reikšmę γ_0 , bet ir nustatome, ar turimas NBS yra optimalus, ar jį dar galima pagerinti.

- TF $z = \gamma_0 - \gamma_{r+1} x_{r+1} - \gamma_{r+2} x_{r+2} - ... - \gamma_q x_q - ... - \gamma_n x_n$.

reikšme γ_0 galima padidinti, jei vietoje kurio nors laisvojo kintamojo x_q reikšmės imtume ne nulį, bet teigiamą skaičių su sąlyga, kad $\gamma_q < 0$.

- Vadinasi, jei egzistuoja laisvojo kintamojo neigiamas įvertinimas $\gamma_q < 0$, tai galima ieškoti geresnio NBS.

Simplekso metodas

- Jeigu GŽ metodu ieškant NBS, vedančiuoju stulpeliu galėjome pasirinkti bet kurį, kuriame tik yra teigiamų elementų, tai simplekso metode, kai ieškome geresnio NBS, vedantysis stulpelis turi tenkinti dvi sąlygas:

1) $\gamma_q < 0$;
2) stulpelyje egzistuoja bent vienas teigiamas elementas.

- Tegul šitos abi sąlygos yra išpildytos, ir vedantysis elementas yra a'_{pq} . Sekančioje lentelėje kintamasis x_q bus bazinis, o x_p bus laisvasis- pasikeičia dviejų kintamųjų, nusakomų vedančiuoju elementu, vaidmenys. Pereiname prie kitos simplekso lentelės GŽ metodu.

Simplekso metodas

- Į klaisimą, kada skaičiavimus baigti, atsako **teorema**:

1) Jeigu egzistuoja bent vienas neigiamas įvertinimas γ_q ir atitinkamame stulpelyje yra bent vienas teigiamas elementas, tai turimą NBS galima pagerinti pereinant prie kitos lentelės.
2) Jeigu egzistuoja bent vienas neigiamas įvertinimas γ_q , ir atitinkamame stulpelyje nėra teigiamų elementų, tai uždavinys optimalaus sprendinio neturi ir max $z = \infty$.
3) Jeigu neegzistuoja neigiamų įvertinimų γ_q , tai turimas NBS **yra optimalus**.

Simplekso metodas

- Išnagrinėkime tiesinio programavimo uždavinio sprendimo simplekso metodu pavyzdį.
- Tarkime, reikia nustatyti, koks turi būti gaminamas 4 rūšių produkcijos P_1, P_2, P_3, P_4 kiekis, jeigu kiekvienai produkcijos rūšiai reikalingi trejopi ištekliai: darbas, žaliavos, finansai.
- Išteklių kiekis, reikalingas produkcijos vieneto pagaminimui, vadinamas sąnaudų norma. Sąnaudų normos ir pelnas, gaunamas už kiekvieną produkcijos vienetą, bei turimų išteklių kiekis nurodyti lentelėje.

Išteklių rūšis	Produkcijos rūšis				Kiekis
	P_1	P_2	P_3	P_4	
Pelnas	60	70	120	130	
Darbas	1	1	1	1	16
Žaliavos	6	5	4	3	110
Finansai	4	6	10	13	100

Simplekso metodas

- Sudarome matematinį modelį.

1) Įvedame nežinomuosius: x_j – j-osios produkcijos kiekis.
2) Užrašome apribojimus kintamiesiems:

$$\begin{cases} x_1 + x_2 + x_3 + x_4 \leqslant 16 \\ 6x_1 + 5x_2 + 4x_3 + 3x_4 \leqslant 110 \\ 4x_1 + 6x_2 + 10x_3 + 13x_4 \leqslant 100 \end{cases}$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad x_4 \geqslant 0.$$

3) Apibrėžiame tikslo funkciją. Mūsų optimalumo kriterijus – pelnas, taigi,

$$60x_1 + 70x_2 + 120x_3 + 130x_4 \to max$$

4) Užrašome uždavinio matematinį modelį.

$$max\left(60x_1 + 70x_2 + 120x_3 + 130x_4\right)$$

$$\begin{cases} x_1 + x_2 + x_3 + x_4 \leqslant 16 \\ 6x_1 + 5x_2 + 4x_3 + 3x_4 \leqslant 110 \\ 4x_1 + 6x_2 + 10x_3 + 13x_4 \leqslant 100 \end{cases}$$

$$x_1 \geqslant 0,\quad x_2 \geqslant 0,\quad x_3 \geqslant 0,\quad x_4 \geqslant 0.$$

- Šiuo atveju grafiškai spręsti uždavinio neįmanoma, todėl jis bus sprendžiamas analitiniu būdu, naudojant simplekso metodą.

Simplekso metodas

- Pirmiausia pereinama nuo nelygybių sistemos prie lygčių sistemos – įvedami papildomi kintamieji, kurie yra lygus nepanaudotos išteklių rūšies kiekiui.

$$max\left(60x_1 + 70x_2 + 120x_3 + 130x_4\right)$$

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 16 \\ 6x_1 + 5x_2 + 4x_3 + 3x_4 + x_6 = 110 \\ 4x_1 + 6x_2 + 10x_3 + 13x_4 + x_7 = 100 \end{cases}$$

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,\ x_4 \geqslant 0,\ x_5 \geqslant 0,\ x_6 \geqslant 0,\ x_7 \geqslant 0.$$

- Simplekso metodą galima taikyti, kai patenkintos šios 3 sąlygos:

1) Uždavinys duotas forma K_{max} .
2) Apribojimų sistemos laisvieji nariai yra neneigiami.
3) Žinomas aprobojimų sistemos neneigiamas bazinis sprendinys.

Simplekso metodas

- Bazinių kintamųjų ir TF išraiškas laisvaisias kintamaisias užrašome lentelės, vadinamos simplekso lentelės pavidalu

BK	C_{BK}	B	60	70	120	130	0	0	0	380
			x_1	x_2	x_3	x_4	x_5	x_6	x_7	Σ
x_5	0	16	1	1	1	1	1	0	0	21
x_6	0	110	6	5	4	3	0	1	0	131
x_7	0	100	4	6	10	13	0	0	1	134
m+1	-	0	-60	-70	-120	-130	0	0	0	-380

Simplekso metodas

- $$\max\left(x_1 - 6x_2 - 3x_3 - 4x_4\right)$$

$$\begin{cases} 5x_1 & & -\ x_3 & +\ 2x_4 & = 4 \\ 6x_1 & +\ 2x_2 & -\ 2x_3 & +\ 3x_4 & = 4 \\ 4x_1 & -\ 5x_2 & +\ x_3 & & = 5 \end{cases}$$

$$x_1 \geqslant 0,\quad x_2 \geqslant 0,\quad x_3 \geqslant 0,\quad x_4 \geqslant 0.$$

$$x_{opt} = \left(1;\ 0;\ 1;\ 0\right),\qquad z_{max} = -2.$$

$$\begin{Bmatrix} max \\ min \end{Bmatrix}\left(6x_1 + x_2 + 2x_3 + 5x_4\right)$$

$$\begin{cases} 4x_1 & & & -\ 3x_4 & \leqslant 5 \\ -5x_1 & +\ x_2 & & +\ 2x_4 & \leqslant 1 \\ x_1 & -\ 3x_2 & +\ x_3 & +\ x_4 & = 2 \end{cases}$$

$$x_1 \geqslant 0,\quad x_2 \geqslant 0,\quad x_3 \geqslant 0,\quad x_4 \geqslant 0.$$