

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

**PREKYBOS BAZĖS UŽDAVINYS**

Laboratorinis darbas 1

Atliko: DGTfm-18 gr. st. Laura Cvilikaitė

Priėmė: Lekt. dr. Aleksandr Igumenov

Vilnius, 2019

**I užduotis (Prekybos bazės uždavinys) (patikslinta)**

Prekybos bazė įsipareigojo saugoti prekes ir išduoti jas vartotojui kiekvieną dieną po tonų. Ji gali gauti prekes periodiškai kas parų vienodo dydžio partijomis po tonų. Vienos prekių partijos priėmimas ir pakrovimas į sandėlį kainuoja Lt. Vienos tonos vidutinės išdavimo išlaidos Lt. Naujos prekės vežamos paskutinę anksčiau atvežtų prekių išdavimo dieną. Reikia nustatyti optimalią vienos prekių partijos apimtį ir jos vežimo periodą , kad prekybos bazės išlaidos per parą būtų mažiausios. Taip pat ( - liekana):

1. **Sudarykite uždavinio matematinį modelį.**

– tonų skaičius, išduodamas vartotojui kiekvieną dieną;

– parų skaičius;

– tonų skaičius;

– vienos prekių partijos pakrovimo ir priemimo į sandėlį kaina;

– vienos tonos vidutinės išdavimo išlaidos;

Vienkartinės (vienos dienos) prekių išdavimo išlaidos – Lt

Vienos dienos prekių priemimo ir pakrovimo į sandėlį išlaidos - Lt

Prekybos bazės išlaidos per parą aprašomos kintamojo funkcija:

Be to, žinome, kad , todėl

Taigi, uždavinio matematinis modelis:

1. **Užrašykite tikslo funkciją.**

Kadangi reikia ieškoti mažiausių išlaidų per parą, naudojame minimumo funkciją.

1. **Matlab aplinkoje suprogramuokite intervalo dalijimo pusiau metodą (tegu a= 1, b= 30**).
2. Tikslo funkcija

function [ct] = tikslo\_funkcija(t) %tikslo funkcija, priklausanti nuo %laiko t

b = mod(20182746, 7)/2 + 2.5; %parametras b duotas sąlygoje, %įsistatome savo stud\_nr

c2 = mod(20182746, 30) + 150; %parametras c2 duotas sąlygoje,

%įsistatome savo stud\_nr

c = mod(20182746, 5) + 3; %parametras c duotas sąlygoje,

%įsistatome savo stud\_nr

ct = b\*c\*t + c2./t; %tikslo funkcijos išraiška

end

1. Intervalo dalijimo pusiau metodas

%intervalo dalijimo pusiau metodas

function [x\_min\_m, f\_min, nauja\_a, nauja\_b, iteration, L1, xpoint, xvalue] = int\_dal\_metodas(a, b)

iteration = 1; %pirma iteracija

x\_m = (a + b)/2; %x\_m - intervalo vidurio taškas, a –

%intervalo pradžia, b - intervalo pabaiga

fx\_m = tikslo\_funkcija(x\_m); %tikslo funkcijos reikšmė intervalo

%vidurio taške

x\_min\_m = x\_m; %turima minimali intervalo vidurio

%koordinatė

fx\_m1 = fx\_m; %nauja funkcijos reikšmė

fx\_m2 = fx\_m; %nauja funkcijos reikšmė

xpoint(iteration) = x\_m; %išsaugomas taškas x\_m

xvalue(iteration) = fx\_m1; %išsaugoma funkcijos reikšmė x\_m taške

nauja\_a = a; %nauja intervalo pradžia

nauja\_b = b; %nauja intervalo pabaiga

L = b - a; %randame intervalo ilgį

L1 = L; %išsaugome intervalo ilgį

while fx\_m2 >= fx\_m1 %ciklas taškams palyginti

iteration = iteration + 1; %didiname iteracijų skaičių

x1 = a + L/4; %didiname pradžios tašką ketvirtadaliu

x2 = b - L/4; %mažiname pabaigos tašką ketvirtadaliu

fx1 = tikslo\_funkcija(x1); %funkcijos reikšmė naujoje intervalo pradžioje taške x1

fx2 = tikslo\_funkcija(x2); %funkcijos reikšmė naujoje intervalo pabaigoje taške x2

%jeigu funkcijos reikšmė naujoje intervalo pradžioje yra mažesne už

%tikslo funkcijos reikšmę buvusiame vidurio taške, tuomet

if(fx1 < fx\_m)

b = x\_m; %intervalo pabaiga yra lygi buvusiam vidurio taškui

x\_m = x1; %buvęs vidurio taškas tampa nauja intervalo pradžia

else %jei funkcijos reikšmė naujoje intervalo pabaigoje yra mažesnė

%už matematinio modelio reikšmę buvusiame vidurio taške, tuomet

if(fx2 < fx\_m)

a = x\_m; %intervalo pradžia yra lygi buvusiam vidurio taškui

x\_m = x2; %buvęs vidurio taškas tampa nauja intervalo pabaiga

else %kitu atveju

a = x1; %intervalo pradžia pasistumia ketvirtadaliu viso intervalo ilgio

b = x2; %intervalo b atsitraukia ketvirtadaliu viso intervalo ilgio

end

end

fx\_m1 = tikslo\_funkcija(x\_m); %tikslo funkcijos reikšmė naujame vidurio taške

xpoint(iteration) = x\_m; %išsaugomas taškas

xvalue(iteration) = fx\_m1; %išsaugoma reikšmė

L = b - a; %išsaugomas naujas intervalo ilgis

if(fx\_m2 > fx\_m1) %jeigu tenkinama sąlygą

L1 = L; %perkeliamas taškas

nauja\_a = a; %perkeliamas taškas

nauja\_b = b; %perkeliamas taškas

x\_min\_m = x\_m; %perkeliamas taškas

fx\_m2 = fx\_m1; %perkeliamas taškas

end

end

f\_min = fx\_m2; %rasta tikslo funkcijos minimali reikšmė

end

1. Išlaidų apskaičiavimas ir grafiko braižymas

%funkcija minimalioms išlaidoms rasti

function [q, t, x\_min\_m, f\_min, islaidos, iteration, L] = min\_islaidos(a, b)

[x\_min\_m, f\_min, nauja\_a, nauja\_b, iteration, L, xpoint, xvalue] = int\_dal\_metodas(a, b);

mid\_point = (nauja\_a+nauja\_b)/2; %mid\_point - intervalo vidurio taškas, nauja\_a - intervalo pradžia, nauja\_b - intervalo pabaiga

mid\_point\_floor = floor(mid\_point); %vidurio taško apatinė riba

f\_floor = tikslo\_funkcija(mid\_point\_floor); %tikslo funkcijos reikšmė taške mid\_point\_floor

mid\_point\_ceil = ceil(mid\_point); %vidurio taško viršutinė riba

f\_ceil = tikslo\_funkcija(mid\_point\_ceil); %tikslo funkcijos reikšmė taške mid\_point\_ceil

if(f\_floor > f\_ceil) %lyginame tikslo funkcijos reikšmę su apatine reikšme

islaidos = f\_ceil; %išsaugoma viršutinė reikšmė

t = mid\_point\_ceil %išsaugoma viršutinė reikšmė

else

islaidos = f\_floor; %išsaugoma apatinė reikšmė

t = mid\_point\_floor; %išsaugoma reikšmė

end

x = a:1:b; %vektorius nuo intervalo pradžios iki pabaigos

y = tikslo\_funkcija(x); %tikslo funkcija

%aukso pjūvis ir interpoliacija

optimization = optimset('Display', 'iter', 'MaxFunEvals', 1000, 'PlotFcns', @optimplotfval , 'TolX', L);

[xfminbnd, yfminbnd] = fminbnd(@tikslo\_funkcija, a, b, optimization)

figure %brėžiame grafiką

plot(x, y, '.-', xpoint, xvalue, '.-'), legend('Kiekviename taške', 'Dalijame intervalą pusiau');

title('Išlaidų kitimas pagal vežimo periodą t');

xlabel('Prekių vežimo periodas (t)');

ylabel('Išlaidos per dieną');

b = mod(20182746, 7)/2 + 2.5; %parametras b duotas sąlygoje, įsistatome savo stud\_nr

q = t \* b; %vienos partijos dydis

f\_min %tikslo f-jos minimumas

x\_min\_m %tikslo f-jos minimumo taškas

t %optimalus prekių vežimo periodas dienomis

q %optimali prekių partijos apimtis

islaidos %optimalus sprendinys

iteration %iteracijų skaičius

L %intervalo ilgis

xfminbnd %funkcijos minimalaus taško koordinatė auksinio pjūvio ir kvadratinės interpoliacijos metodu

yfminbnd %funkcijos minimumo reikšmė auksinio pjūvio ir kvadratinės interpoliacijos metodu

end

1. **Raskite tikslo funkcijos minimumą (mažiausias prekybos bazės išlaidas) ir minimumo tašką (prekių vežimo intervalą). Rastas optimalus prekių vežimo periodas bus intervalas. Paimkite jo vidurinę reikšmę ir suapvalinkite ją iki sveikojo skaičiaus į vieną ir kitą pusę. Iš gautų dviejų reikšmių išrinkite optimalų sprendinį (kad tikslo funkcijos reikšmė būtų mažesnė). Taip pat apskaičiuokite optimalią prekių partijos apimtį.**

* Tikslo funkcijos minimumas

f\_min = 100.4667

* Tikslo funkcijos minimumo taškas

x\_min\_m = 2.8125

* Rastas optimalus prekių vežimo periodas dienomis

t = 3

* Optimalus sprendinys (kad tikslo funkcijos reikšmė būtų mažesnė).

islaidos = 100

* Optimali prekių partijos apimtis

q = 12

1. **Nustatykite, kiek iteracijų reikia atlikti, norint gauti rezultatą norimu tikslumu (tikslumas nurodomas intervalo ilgiu). Čia viena iteracija –funkcijos vienos reikšmės apskaičiavimas.**

iteration = 5

L = 3.6250

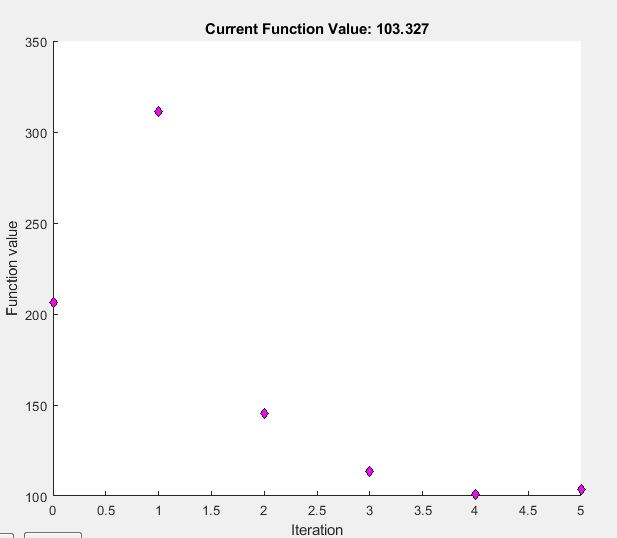
1. **Nubraižykite tikslo funkcijos grafiką, nurodykite jo pavadinimą, x ir y ašių pavadinimus.**

P. S.Turi būti suprogramuotos trys funkcijos:

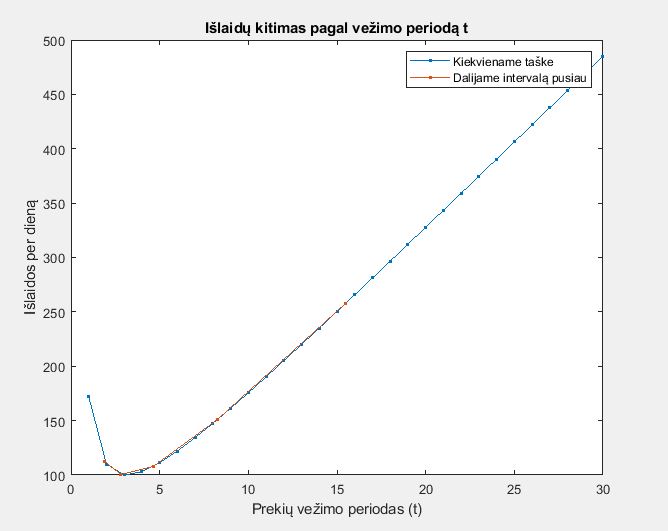
•pirmoje apskaičiuojama tikslo funkcijos reikšmė;

•antroje įgyvendintas intervalo dalijimo pusiau metodas;

•trečioje kreipiamasi į antrąją funkciją esant įvairioms tikslumo (intervalo ilgio) reikšmėmis; taip pat papildomi veiksmai, nurodyti 4 žingsnyje; grafiko braižymas.



**Pav. 1 Auksinio pjūvio ir kvadratinės interpoliacijos metodas**



**Pav. 2 Intervalo dalijimo pusiau metodas**

1. **Raskite funkcijos minimumą auksinio pjūvio metodu ir kvadratine interpoliacija naudojant Matlab funkciją fminbnd. Nustatykite, kiek kartų reikia skaičiuoti funkcijos reikšmes, kad gautume tą patį tikslumą, kaip ir intervalo dalijimo pusiau metodu. P. S.Turi būti papildyta trečia funkcija kreipiniu į funkciją fminbnd.**

Funkcijos minimalaus taško koordinatė auksinio pjūvio ir kvadratinės interpoliacijos metodu:

xfminbnd = 3.6149

Funkcijos minimumo reikšmė auksinio pjūvio ir kvadratinės interpoliacijos metodu:

yfminbnd = 100.9932

Func-count x f(x) Procedure

1 12.077 206.149 initial

2 18.923 311.012 golden

3 7.84597 145.418 golden

4 5.23104 113.519 golden

5 3.61493 100.993 golden

6 2.40659 103.327 golden

Optimization terminated:

the current x satisfies the termination criteria using OPTIONS.TolX of 3.625000e+00