САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и кибербезопасности

высшая школа компьютерных технологий и информационных систем

**РАСЧЁТНОЕ ЗАДАНИЕ**

**«Нелинейное программирование. Условная оптимизация целевой функции»**

по дисциплине «Системный анализ и принятие решений»

Выполнил:

студент гр. 5130901/20102

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вагнер А.А.

(подпись)

Преподаватель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сиднев А.Г.

(подпись)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Санкт-Петербург

2024

Исходные данные. Вариант 1.

Содержанием задания является поиск условного максимума целевой функции следующими методами:

* Лагранжа (ограничение 5);
* Била (ограничения 1, 2, 3, 4);
* проекции градиента (ограничения 1, 2, 3, 4);
* возможных направлений (ограничения 6, 7);
* штрафных функций (ограничения 1, 2, 3, 4);
* барьерных функций (ограничения 1, 2, 3, 4);
* Метод Франка-Вульфа (ограничения 1, 2, 3, 4).

1. Метод Лагранжа

При

Напишем функцию Лагранжа:

Условие стационарности L(X,V) в точке (Xi, Vi):

Решим систему уравнений:

Матрица Гесса:

Она отрицательно определена, поэтому данная точка является точкой максимума L(X,V), соответственно X\* является решением задачи условной оптимизации.

Изображение выглядит как текст, снимок экрана, диаграмма, круг

Автоматически созданное описание

Рис. – Результат работы программы

Листинг 1 – Код программы

function [] = Main()

clc;

close all;

clear all;

mas = [-6,-9,4,20,60,-1,1,1,1,1,0,4,6,2,1,9,36];

start\_x1 =0;

start\_x2 = 0;

stop\_x1 = 7;

stop\_x2 = 7;

fprintf('Решение задачи методом Лагранжа:\nУсловия стационарности в точке (X\*,V\*):\n\n');

[ Dermas ] = PartialDerivative( mas, 1 );

fprintf('\nПолучили СЛАУ. Решаем и получаем единственное решение:\n\n');

[ x1,x2,v ] = GetAnswer ( Dermas );

fprintf('\nЗначение функции:\n');

[ f ] = F(x1, x2, mas);

[ H ] =PlotGraph( mas,x1,x2,start\_x1,start\_x2,stop\_x1,stop\_x2);

hold on;

plot([start\_x1 stop\_x1],[x2 x2],':');

plot([x1 x1],[start\_x2 stop\_x2],':');

hold off;

legend('Линии равного уровня','Точка максимума');

end

function [ Dermas ] = PartialDerivative( mas, indicator )

Dermas(1,1)=mas(1)\*2;

Dermas(1,2)=mas(3);

Dermas(1,3)=mas(4);

Dermas(2,1)=mas(3);

Dermas(2,2)=mas(2)\*2;

Dermas(2,3)=mas(5);

if (indicator ==0)

fprintf ('dL/dx1 = %dx1 + ( %dx2 ) + ( %d )= 0\n',Dermas(1,1),Dermas(1,2),Dermas(1,3));

fprintf ('dL/dx2 = %dx1 + ( %dx2 ) + ( %d )= 0\n',Dermas(2,1),Dermas(2,2),Dermas(2,3));

end;

if (indicator ==1)

Dermas(3,1)=mas(10);

Dermas(3,2)=mas(11);

Dermas(3,3)=-mas(14);

Dermas(3,4)=0;

Dermas(1,4)=mas(10);

Dermas(2,4)=mas(11);

fprintf ('dL/dx1 = %dx1 + ( %dx2 ) + ( %d ) + ( %dv ) = 0\n',Dermas(1,1),Dermas(1,2),Dermas(1,3),Dermas(1,4));

fprintf ('dL/dx2 = %dx1 + ( %dx2 ) + ( %d ) + ( %dv ) = 0\n',Dermas(2,1),Dermas(2,2),Dermas(2,3),Dermas(2,4));

fprintf ('dL/dv = %dx1 + ( %dx2 ) + ( %d ) = 0\n',Dermas(3,1),Dermas(3,2),Dermas(3,3));

end;

end

function [ x1,x2,v ] = GetAnswer ( Dermas )

A=[Dermas(1,1), Dermas(1,2),Dermas(1,4);

Dermas(2,1), Dermas(2,2),Dermas(2,4);

Dermas(3,1), Dermas(3,2),Dermas(3,4);];

b = [-Dermas(1,3); -Dermas(2,3); -Dermas(3,3)];

x = inv(A)\*b;

x1 = x(1,1);

x2 = x(2,1);

v = x(3,1);

fprintf('(X\*,V\*) = (x1\*,x2\*,v\*) = (%-.4f,%-.4f,%-.4f)\n',x1,x2,v);

end

function [handler] = PlotGraph( mas,x1,x2,start\_x1,start\_x2,stop\_x1,stop\_x2)

x\_1=start\_x1:.1:stop\_x1;

x\_2=start\_x2:.1:stop\_x2;

[x\_1,x\_2]=meshgrid(x\_1,x\_2);

w=(mas(1)\*x\_1.^2 +mas(2)\*x\_2.^2 + mas(3)\*x\_1.\*x\_2 + mas(4)\*x\_1 + mas(5)\*x\_2);

handler = figure;

contour(x\_1,x\_2,w,50)

hold on;

plot (x1, x2, 'o-k','LineWidth', 2);

xlabel('x1');

ylabel('x2');

hold off

end

function [f] =F(x1, x2, mas)

f = mas(1)\*x1^2+mas(2)\*x2^2+mas(3)\*x1\*x2+mas(4)\*x1+mas(5)\*x2;

fprintf('\nf(x)=%-.4f\n\n',f);

end

1. Метод Била

Приведём к канонической форме:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 1 | -1 | 4 |
|  | -1 | -1 | 6 |
|  |  |  |  |

Данная точка является допустимой, но не оптимальной.

Найдем соотношение между приращением свободной переменной и изменениями базисных переменных и и производной:

при при

при

Так как производная обращается в ноль раньше базисных переменных, введём в задачу новую свободную переменную . Переменную х2 включим в базис.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 2 | -9 | 30 |
|  | 1 | -1 | 4 |
|  | -1 | -1 | 6 |

Пересчитаем таблицу

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | -2 | 1 | -30 |
|  | -7 | -1 | -6 |
|  | 11 | -1 | -24 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Выразим целевую функцию через новые свободные переменные:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Выберем ведущий столбец как максимальную производную в точке а ведущую строку как минимальное частное деления на отрицательные элементы столбца.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | 100 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Выразим целевую функцию через новые свободные переменные:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Решение

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Рис. Траектория решения методом Била

Листинг 2 – Код программы

function [] = Main()

clc;

close all;

clear all;

mas = [-6,-9,4,20,60,-1,1,1,1,1,0,4,6,2,1,9,36];

start\_x1 =0;

start\_x2 = 0;

stop\_x1 = 8;

stop\_x2 = 8;

x1\_bil=[0, 0, 2.18181818];

x2\_bil=[0, 3.333333, 3.818181];

[ H2 ] = PlotBil( mas,x1\_bil,x2\_bil,start\_x1,start\_x2,stop\_x1,stop\_x2 );

end

function [ handler ] = PlotBil( mas,x1\_bil,x2\_bil,start\_x1,start\_x2,stop\_x1,stop\_x2 )

x=start\_x1:.1:stop\_x1;

y=start\_x2:.1:stop\_x2;

[ H ] =PlotGraph( mas,x1\_bil,x2\_bil,start\_x1,start\_x2,stop\_x1,stop\_x2);

handler =H;

hold on;

plot (x,x\*0,':b','LineWidth', 1);

plot (0\*y,y,':b','LineWidth', 1);

if (mas(7)==0)

plot (x, (mas(12)-mas(6)\*x),':b','LineWidth', 1);

else

plot (x, (mas(12)-mas(6)\*x)/mas(7),':b','LineWidth', 1);

end;

if (mas(9)==0)

plot (x, (mas(13)-mas(8)\*x),':b','LineWidth', 1);

else

plot (x, (mas(13)-mas(8)\*x)/mas(9),':b','LineWidth', 1);

end;

legend('Линии равного уровня','Ход решения','Ограничения');

hold off;

end

function [handler] = PlotGraph( mas,x1,x2,start\_x1,start\_x2,stop\_x1,stop\_x2)

x\_1=start\_x1:.1:stop\_x1;

x\_2=start\_x2:.1:stop\_x2;

[x\_1,x\_2]=meshgrid(x\_1,x\_2);

w=(mas(1)\*x\_1.^2 +mas(2)\*x\_2.^2 + mas(3)\*x\_1.\*x\_2 + mas(4)\*x\_1 + mas(5)\*x\_2 );

handler = figure;

contour(x\_1,x\_2,w,50)

hold on;

plot (x1, x2, 'o-k','LineWidth', 2);

xlabel('x1');

ylabel('x2');

hold off

end

1. Метод проекции градиента

**Шаг 1.**

;

направление допустимо

**Шаг 2.**

Найдём матрицу проекции P

**Шаг 3.**

Проверим условие останова:

=> Найденное решение потимально

Решение:

mas = [-6,-9,4,20,60,-1,1,1,1,1,0,4,6,2,1,9,36];

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рис. Траектория поиска решения методом проекции градиента

Листинг 3 – Код программы

function [] = Main()

clc;

close all;

clear all;

mas = [-6,-9,4,20,60,-1,1,1,1,1,0,4,6,2,1,9,36];

start\_x1 =0;

start\_x2 = 0;

stop\_x1 = 8;

stop\_x2 = 8;

x1\_grad=[0, 1.5, 2.4205];

x2\_grad=[0, 4.5, 3.5795];

[ H2 ] = PlotBil( mas,x1\_grad,x2\_grad,start\_x1,start\_x2,stop\_x1,stop\_x2 );

function [ handler ] = PlotBil( mas,x1\_bil,x2\_bil,start\_x1,start\_x2,stop\_x1,stop\_x2 )

x=start\_x1:.1:stop\_x1;

y=start\_x2:.1:stop\_x2;

[ H ] =PlotGraph( mas,x1\_bil,x2\_bil,start\_x1,start\_x2,stop\_x1,stop\_x2);

handler =H;

hold on;

plot (x,x\*0,':b','LineWidth', 1);

plot (0\*y,y,':b','LineWidth', 1);

if (mas(7)==0)

plot (x, (mas(12)-mas(6)\*x),':b','LineWidth', 1);

else

plot (x, (mas(12)-mas(6)\*x)/mas(7),':b','LineWidth', 1);

end;

if (mas(9)==0)

plot (x, (mas(13)-mas(8)\*x),':b','LineWidth', 1);

else

plot (x, (mas(13)-mas(8)\*x)/mas(9),':b','LineWidth', 1);

end;

legend('Линии равного уровня','Ход решения','Ограничения');

hold off;

end

end

function [handler] = PlotGraph( mas,x1,x2,start\_x1,start\_x2,stop\_x1,stop\_x2)

x\_1=start\_x1:.1:stop\_x1;

x\_2=start\_x2:.1:stop\_x2;

[x\_1,x\_2]=meshgrid(x\_1,x\_2);

w=(mas(1)\*x\_1.^2 +mas(2)\*x\_2.^2 + mas(3)\*x\_1.\*x\_2 + mas(4)\*x\_1 + mas(5)\*x\_2 );

handler = figure;

contour(x\_1,x\_2,w,50)

hold on;

plot (x1, x2, 'o-k','LineWidth', 2);

xlabel('x1');

ylabel('x2');

hold off

end

1. Метод возможных направлений