**Дата:**08**.**12**.**2017

**Виконано:** Пазенок А.О.

**Посилання:**

**Перевірено:** Гордєєв Артем Дмитрович

**Дисципліна:** «Техніка експериментальних досліджень»

**Лабораторна робота № 7**

**Тема: «**МЕТОДИ ПАРАМЕТРИЧНОЇ ОПТИМІЗАЦІЇ ТЕХНОЛОГІЧНИХ СИСТЕМ**»**

**Мета:**

1. Дослідити графічний метод параметричної оптимізації.

2. Побудувати програму в MATLAB, яка реалізує графічний метод параметричної оптимізації.

Хід роботи:

1. Завдання.

Необхідно знайти максимальне значення функції мети  за наявності таких обмежень:



Визначення оптимального значення параметра провести графічним методом. Вхідні дані наведені в табл. 1.

Таблиця 7.1.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Коефіцієнт | Варіант | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| *А* | 3 | 2 | 1 | 3 | 5 | 3 | 4 | 3 | 2 | 3 |
| *В* | 5 | 4 | 4 | 2 | 4 | 1 | 5 | 4 | 5 | 6 |

2. **Вхідні дані.**

Створіть блок-програму для вводу вхідних даних для подальших розрахунків та побудови графіку оптимізації.

Вхідними даними для розрахунків та побудови графіку оптимізації являються:

* коефіцієнт цільової функції *А*;
* коефіцієнт цільової функції *B*;
* коефіцієнти біля змінних *x1*, *x2* в системі нерівностей обмежень.
* 3. **Вивід створених даних на екран комп’ютера.** Виведіть в «Command window» створені функцію мети та системи нерівностей обмежень (Рис. 7.2). Для цього необхідно в MatLab одним вектором описати введені дані та вивести їх за допомогою функції *fprintf*. Зазначте, що дані коефіцієнтів *А* і *В*, введених в пункті 2.1, будуть мати цифровий формат, а не буквений, який необхідний для роботи функції *fprintf*.

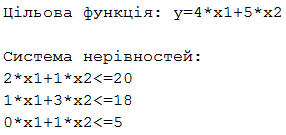


Рис. 2. Приклад візуалізації введених даних

4. **Розрахунок функції мети.**

4.1. Визначте *x1*, *x2* з кожної нерівності системи обмежень, щоб згодом побудувати необхідні лінії на графіку. Для цього необхідно використати значення, що зберігаються в змінних «sys1», «sys2», «sys3». Зазначте, що лінію на графіку можна побудувати знаючи дві точки на осях координат, відштовхуйтесь від цього, щоб зрозуміти як розрахувати *x1*, *x2*. Результат розрахунків має міститись в змінних «x1\_1» і т.д., де перша цифра означає номер змінної в певній нерівності, а друга – номер нерівності.

Виведіть результати розрахунків в «Command window» за допомогою функції *fprintf*.

4.2. Розрахуйте функцію мети в різних точках графіку. Для того щоб окреслити область допустимих значень (ОДЗ) необхідно знайти чотири точки функції мети. При чому дві точки ОДЗ будуть знаходитись на осях координат, а дві інші точки будуть точками перетину ліній, що описуються нерівностями в системі.

Для того, щоб знайти точки на осях координат достатньо умовно прирівняти одні зі змінних системи до нуля (наприклад, усі *x1* системи) та знайти мінімальне значення з усіх інших знайдених змінних (усі *x2* системи). Так як розрахунки змінних *x1*, *x2* проводились в пункті 4.1, можна скористатись результатами, що містяться в змінних «x1\_1» і т.д.

Для знаходження двох точок перетину ліній необхідно вирішити систему рівнянь відносно цих прямих. Для цього можна скористатись методом Гауса рішення матриць у математиці. Для кожної точки ОДЗ, що утворилась в разі перетину ліній, необхідно сформувати дві матриці:

- матриця, що містить значення коефіцієнтів при *x1*, *x2* кожної нерівності (в програмі це змінні – «A\_sys\_1\_2» і т.д., де перша та друга цифри означають номера систем між якими шукають розв’язки) (див. рис. 2);

- матриця, що містить значення правої частини нерівності (в програмі це змінні – «В\_sys\_1\_2» і т.д.) (див. рис. 2).

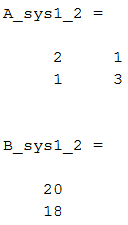


Рис. 7.2. Вигляд матриць у вікні «Command window»

Після цього необхідно перемножити дві матриці і отримати необхідні розв’язки нерівностей, тобто значення точок перетину *x1*, *x2*. Розв’язки систем рівнянь мають міститись в змінних «Y\_sys\_1\_2» і т.д.

5. **Побудова графіку оптимізації.**

Для побудови графіку оптимізації необхідно скористатись функціями *area*, *plot*, *line*. За допомогою функції *area* можна побудувати область допустимих значень, за допомогою функції *line* всі лінії графіку, а за допомогою функції *plot* точки багатокутнику ОДЗ на графіку.

Скористайтесь функцією *text*, щоб підписати всі необхідні точки цільової функції на графіку. Приклад графіку оптимізації представлено на рис. 3.



Рис. 3. Приклад графіку оптимізації

6. **Визначення максимального значення функції мети.**

Необхідно розрахувати цільові функції  для кожної точки графіку (ABCD) та знайти максимальне значення. Для цього сформуйте вектори рядки розрахованих значень *x1*, *x2* системи для кожної із точок ОДЗ, і збережіть у змінних «X\_A», «X\_B» і т.д.

Підставивши значення коефіцієнтів A, B та розрахованих значень *x1*, *x2* можна розрахувати значення цільової функції за формулою  для кожної точки ОДЗ («Y\_A», «Y\_B» і т.д.). Розрахуйте максимум серед знайдених значень («Y\_max»).

Для того, щоб знайти точку максимуму і потім відобразити кінцевий результат необхідно створити блок-програму порівняння максимуму цільової функції зі всіма знайденими точками, при якому в змінну «ansv» буде занесено текстове значення точки екстремуму цільової функції. Додатково необхідно занести значення коефіцієнтів саме для цієї точки максимуму, щоб потім представити не тільки назву точки максимуму, а й саму цільову функцію. Таку блок-програму можна створити скориставшись функціями *if* та *elseif*. Приклад виводу результату показано на рис. 3.

|  |
| --- |
| clc  clear all  close all  % --- Ввод данных  % ----- целевая функция -----  A=input('Введите коэфициент "A" функции цели, A='); % Пускай пользователь сам введет коэ-т цели А для функции цели. Используйте фразу: "'Введите коэфициент "A" функции цели, A='"  B=input('Введите коэфициент "В" функции цели, B='); % Пускай пользователь сам введет коэ-т цели В для функции цели. Используйте фразу: "'Введите коэфициент "B" функции цели, B='"  A=num2str(A);  B=num2str(B);  % ---- Визуализация введенной целевой функции ---  y\_txt=['y=',A,'\*x1+',B,'\*x2']; % создайте одним вектором всё необходимое для вывода функции в command window (см. указания)  fprintf(y\_txt)  % -------    % ----- Система уравнений ----  sys1=[2,1,20];  sys2=[1,3,18];  sys3=[0,1,5];  sys1\_txt=[num2str(sys1(1,1)),'\*x1+',num2str(sys1(1,2)),'\*x2<=',num2str(sys1(1,3)),'; ']; % создайте одним вектором всё необходимое для вывода неравенства в command window.  sys2\_txt=[num2str(sys2(1,1)),'\*x1+',num2str(sys2(1,2)),'\*x2<=',num2str(sys2(1,3)),'; ']; % создайте одним вектором всё необходимое для вывода неравенства в command window  sys3\_txt=[num2str(sys3(1,1)),'\*x1+',num2str(sys3(1,2)),'\*x2<=',num2str(sys3(1,3)),'; ']; % создайте одним вектором всё необходимое для вывода неравенства в command window  fprintf('\nСистема неравенств:\n')  fprintf(sys1\_txt);  fprintf(sys2\_txt);  fprintf(sys3\_txt);  % ---------------------------  % ---- конец ввода данных    % ---- Расчеты ----  % Одна из переменных системы приравняется нулю!    % для 1 неравенства:  x1\_1=sys1(1,3)/sys1(1,1); % Определите  x2\_1=sys1(1,3)/sys1(1,2);  % для 2 неравенства:  x1\_2=sys2(1,3)/sys2(1,1);  x2\_2=sys2(1,3)/sys2(1,2);  % для 3 неравенства:  x1\_3=sys3(1,3)/sys3(1,1);  x2\_3=sys3(1,3)/sys3(1,2);    % отображение расчетов:  fprintf('\nРасчет для построения графика:\n')  fprintf('\nРезультат расчетов для неравенства №1:%2.3f ',x1\_1,x2\_1) % фразу начните со слов "'Результат расчетов для неравенства №1:"  fprintf('\nРезультат расчетов для неравенства №2:%2.3f',x1\_2,x2\_2) % фразу начните со слов "'Результат расчетов для неравенства №2:"  fprintf('\nРезультат расчетов для неравенства №3:%2.3f',x1\_3,x2\_3) % фразу начните со слов "'Результат расчетов для неравенства №3:"  % -----------------  % ---- Расчеты для построение графика -------  % area  % решим систему неравенств  X1min=min([x1\_1,x1\_2,x1\_3]); % Находим минимальное значение на каждой из осей (точки на осях). для обоих минимумов нужно реализовать нахождение в одну строку.  X2min=min([x2\_1,x2\_2,x2\_3]);    % -- Для точек пересечения линий  % расчет точке пересечения линий для систем 1 и 2  A\_sys1\_2=[sys1(1,1),sys1(1,2);sys2(1,1),sys2(1,2)]; % матрица, содержащая коэфициенты при двух иксах для неравенств 1 и 2 системы  B\_sys1\_2=[sys1(1,3);sys2(1,3)]; % матрица, содержащая числа правой части неравенств  Y\_sys1\_2=inv(A\_sys1\_2)\*(B\_sys1\_2); % решение системы (нахождение точек пересечения)    % расчет точке пересечения линий для систем 1 и 3  A\_sys1\_3=[sys1(1,1),sys1(1,2);sys3(1,1),sys3(1,2)];  B\_sys1\_3=[sys1(1,3);sys3(1,3)];  Y\_sys1\_3=inv(A\_sys1\_3)\*(B\_sys1\_3);    % расчет точке пересечения линий для систем 2 и 3  A\_sys2\_3=[sys2(1,1),sys2(1,2);sys3(1,1),sys3(1,2)];  B\_sys2\_3=[sys2(1,3);sys3(1,3)];  Y\_sys2\_3=inv(A\_sys2\_3)\*(B\_sys2\_3);    % % ---- построение графика оптимизации  area([0,Y\_sys1\_2(2),Y\_sys2\_3(2),x2\_3],[x1\_1,Y\_sys1\_2(1),Y\_sys2\_3(1),0]) % Изучите функцию и постройте ваш вариант ОДЗ  grid on  xlim([0 10])  ylim([0 19])  hold on    line([x2\_1,0],[0,x1\_1]) % изучите функцию и постройте линии для вашего варианта  line([x2\_2,0],[0,x1\_2])  line([x2\_3,x2\_3],[0,20])    plot([0,Y\_sys1\_2(2),Y\_sys2\_3(2),x2\_3],[x1\_1,Y\_sys1\_2(1),Y\_sys2\_3(1),0],'\*') % изучите функцию и постройте все точки описывающие координаты ОДЗ в виде "о"  hold off    % Подпись точек целевой функции  text(Y\_sys1\_2(2),Y\_sys1\_2(1),'Y(B)') % т. 'Y(B)'  text(Y\_sys2\_3(2),Y\_sys2\_3(1),'Y(C)') % т. 'Y(C)'  text(x2\_3,0,'Y(D)') % т. 'Y(D)'  xlabel('X2')  ylabel('X1')    % % Точки целевой функции  % создадим точки X1 X2 для проверки целевой функции  X\_A=[0,x1\_1]; % т. 'Y(A)'  X\_B=[Y\_sys1\_2(2),Y\_sys1\_2(1)]; % т. 'Y(B)'  X\_C=[Y\_sys2\_3(2),Y\_sys2\_3(1)]; % т. 'Y(C)'  X\_D=[x2\_3,0]; % т. 'Y(D)'    % целевая функция  Y\_A=sum(X\_A.\*[A,B]) ;  Y\_B=sum(X\_B.\*[A,B]) ;  Y\_C=sum(X\_C.\*[A,B]) ;  Y\_D=sum(X\_D.\*[A,B]) ;    Ymax=max([Y\_A,Y\_B,Y\_C,Y\_D]) ; % максимальное значение целевой функции  % определение точки максимума целевой функции  if Ymax == Y\_A  ansv='A';  x1 = 0;  x2 = x1\_1;  elseif Ymax == Y\_B  ansv='B';  x1 = Y\_sys1\_2(2);  x2 = Y\_sys1\_2(1);  elseif Ymax == Y\_C  ansv='C';  x1 = Y\_sys2\_3(2);  x2 = Y\_sys2\_3(1);  elseif Ymax == Y\_D  ansv='D';  x1 = x2\_3;  x2 = 0;  end  Ansver=['Ответ: Функция цели принимает значение в т.(',ansv,'): y(',ansv,')=',num2str(A),'\*',num2str(x1),'+',num2str(B),'\*',num2str(x2),]; % Заполните строчку ответа самтостоятельно  disp(Ansver)  disp(['Остаточний варіант рівняння матеме вигляд y(',ansv,')=A\*x1 +B\*x2']) |

**Результати виконання:**

Введите коэфициент "A" функции цели, A=5

Введите коэфициент "В" функции цели, B=4

y=5\*x1+4\*x2

Система неравенств:

2\*x1+1\*x2<=20; 1\*x1+3\*x2<=18; 0\*x1+1\*x2<=5;

Расчет для построения графика:

Результат расчетов для неравенства №1:10.000

Результат расчетов для неравенства №1:20.000

Результат расчетов для неравенства №2:18.000

Результат расчетов для неравенства №2:6.000

Результат расчетов для неравенства №3:Inf

Результат расчетов для неравенства №3:5.000Ответ: Функция цели принимает значение в т.(B): y(B)=5\*3.2+4\*8.4

Остаточний варіант рівняння матеме вигляд y(B)=A\*x1 +B\*x2

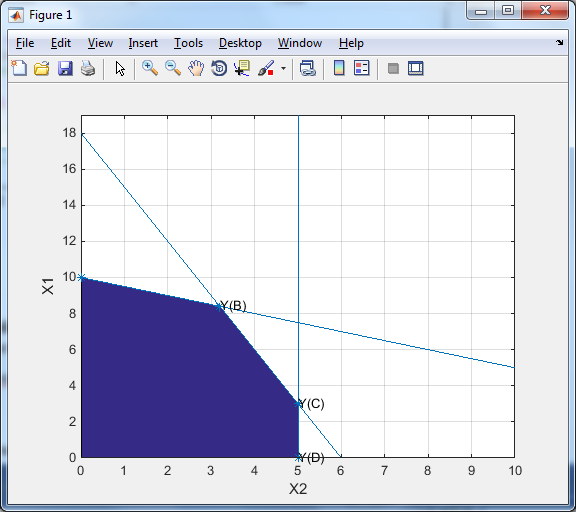


Рис.1 Результат виконання графічного методу

**Контрольні питання:**

1. Що розуміють під терміном “параметрична оптимізація”?

Параметрична оптимізація потрібна для знаходження оптимальних значень параметрів.

1. Які методи оптимізації відносяться до числових та в яких випадках вони застосовуються?

Числові методи оптимізації основані на математичному програмуванні:

* лінійне програмування (показник ефективності і обмеження лінійно залежні);
* нелінійні (відповідно нелінійно залежні);
* динамічні (коли задача більш складна).

1. З якою кількістю параметрів дозволяє розв’язувати задачі графічний метод параметричної оптимізації.

Графічний метод є найпростішим, для виконання і використовується при наявності двох параметрів

**Висновок:** В ході роботи ми розглянули методи параметричної оптимізації, та спробували застосувати графічний метод.