|  |  |
| --- | --- |
| **UNBIZ1957с** | **МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ****ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА****ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ****ЗАТВЕРДЖЕНО****на засіданні кафедри цифрової економіки** **та бізнес-аналітики****протокол № 6 від “21” січня 2020 р.****Зав. кафедри \_\_\_\_\_\_\_\_\_\_\_\_ Шевчук І.Б.** (підпис)**ПЛАНИ ЛАБОРАТОРНИХ ЗАНЯТЬ І МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ЩОДО ЇХ ПРОВЕДЕННЯ** **З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ** **Дослідження операцій** (назва навчальної дисципліни)**галузь знань:** 05 «Соціальні та поведінкові науки»  (шифр та найменування галузі знань)**спеціальність:** 051 “Економіка”  (код та найменування спеціальності)**спеціалізація:** \_\_ \_Інформаційні технології в бізнесі\_\_\_\_\_\_\_\_\_\_\_ (найменування спеціалізації)**освітній ступінь:** бакалавр  (бакалавр/магістр) **Укладач:**Мищишин О.Я. доцент, к. ф.-м.н., доцент  (ПІБ, посада, науковий ступінь, вчене звання)**ЛЬВІВ 2020** |
| ***КАФЕдра цифрової економіки та бізнес-аналітики*** |

**3. ПЛАНИ ЛАБОРАТОРНИХ ЗАНЯТЬ**

**План лабораторного заняття № 6**

**Тема № 6. Розв’язування задач нелінійного програмування.**

**Навчальний час: 2** год.

**Міжпредметні зв’язки:** Зв’язок із елементами знань і умінь таких навчальних дисциплін як „Теорія випадкових процесів” та „Інформатика”.

**Мета і завдання лабораторного заняття:** познайомити з поняттям нелінійного програмування загальної форми задачі, її властивостями та способами побудови, побудувати загальну форму для різних типів задач.

**Питання для перевірки базових знань за темою лабораторного заняття:**

1. Що таке нелінійне програмування?
2. Що таке квадратичне програмування?
3. Що таке загальна форма задачі квадратичного програмування?
4. Що таке задача квадратичного програмування?
5. Що таке критерій Сільвестрова?

**Завдання:**

1. Вивчити теоретичні основи задач квадратичного програмування. Опрацювати приклади.
2. Використовуючи схему побудови, виконати наступні завдання:

*Приклад*. *1*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 8x1-12x2→(min),*

*при обмеженнях*

*2x1+x2 ≤ 15,*

*2x1+x2 ≤ 19,*

*x1, x2 ≤ 0.*

*Приклад*. *2*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 8x1-19x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 14,*

*2x1+7x2 ≤ 29,*

*x1, x2 ≤ 0.*

*Приклад*. *3*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 5x1+8x2→(min),*

*при обмеженнях*

*2x1+5x2 ≤ 15,*

*x1+3x2 ≤ 18,*

*x1, x2 ≤ 0.*

*Приклад*. *4*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 7x1-21x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 13,*

*x1+7x2 ≤ 19,*

*x1, x2 ≤ 0.*

*Приклад*. *5*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 5x1-25x2→(min),*

*при обмеженнях*

*2x1+x2 ≤ 15,*

*5x1+7x2 ≤ 9,*

*x1, x2 ≤ 0.*

*Приклад*. *6*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 17x1-25x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 10,*

*x1+7x2 ≤ 9,*

*x1, x2 ≤ 0.*

*Приклад*. *7*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 8x1-12x2→(min),*

*при обмеженнях*

*2x1+x2 ≤ 15,*

*2x1+x2 ≤ 19,*

*x1, x2 ≤ 0.*

*Приклад*. *8*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 8x1-19x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 14,*

*2x1+7x2 ≤ 29,*

*x1, x2 ≤ 0.*

*Приклад*. *9*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 5x1+8x2→(min),*

*при обмеженнях*

*2x1+5x2 ≤ 15,*

*x1+3x2 ≤ 18,*

*x1, x2 ≤ 0.*

*Приклад*. *10*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 7x1-21x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 13,*

*x1+7x2 ≤ 19,*

*x1, x2 ≤ 0.*

*Приклад*. *11*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 5x1-25x2→(min),*

*при обмеженнях*

*2x1+x2 ≤ 15,*

*5x1+7x2 ≤ 9,*

*x1, x2 ≤ 0.*

*Приклад*. *12*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 17x1-25x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 10,*

*x1+7x2 ≤ 9,*

*x1, x2 ≤ 0.*

*Приклад*. *13*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 8x1-12x2→(min),*

*при обмеженнях*

*2x1+x2 ≤ 15,*

*2x1+x2 ≤ 19,*

*x1, x2 ≤ 0.*

*Приклад*. *14*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 8x1-19x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 14,*

*2x1+7x2 ≤ 29,*

*x1, x2 ≤ 0.*

*Приклад*. *15*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 5x1+8x2→(min),*

*при обмеженнях*

*2x1+5x2 ≤ 15,*

*x1+3x2 ≤ 18,*

*x1, x2 ≤ 0.*

*Приклад*. *16*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 7x1-21x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 13,*

*x1+7x2 ≤ 19,*

*x1, x2 ≤ 0.*

*Приклад*. *17*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 5x1-25x2→(min),*

*при обмеженнях*

*2x1+x2 ≤ 15,*

*5x1+7x2 ≤ 9,*

*x1, x2 ≤ 0.*

*Приклад*. *18*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 17x1-25x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 10,*

*x1+7x2 ≤ 9,*

*x1, x2 ≤ 0.*

*Приклад*. *19*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 8x1-12x2→(min),*

*при обмеженнях*

*2x1+x2 ≤ 15,*

*2x1+x2 ≤ 19,*

*x1, x2 ≤ 0.*

*Приклад*. *20*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 8x1-19x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 14,*

*2x1+7x2 ≤ 29,*

*x1, x2 ≤ 0.*

*Приклад*. *21*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 5x1+8x2→(min),*

*при обмеженнях*

*2x1+5x2 ≤ 15,*

*x1+3x2 ≤ 18,*

*x1, x2 ≤ 0.*

*Приклад*. *22*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 7x1-21x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 13,*

*x1+7x2 ≤ 19,*

*x1, x2 ≤ 0.*

*Приклад*. *23*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 5x1-25x2→(min),*

*при обмеженнях*

*2x1+x2 ≤ 15,*

*5x1+7x2 ≤ 9,*

*x1, x2 ≤ 0.*

*Приклад*. *24*

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 17x1-25x2→(min),*

*при обмеженнях*

*2x1+4x2 ≤ 10,*

*x1+7x2 ≤ 9,*

*x1, x2 ≤ 0.*

 **Означення 3.** Квадратичною формою функції *n* змінних *x*1, *x*2, … *x*n  називається функція вигляду



 **Означення 4.** Квадратична форма називається додатньо визначеною, якщо визначник матриці⏐А⏐>0 і додатньо напіввизначений – якщо ⏐**А**⏐≥0 при всіх ***x*** ≠ 0.

 **Теорема 2 (критерій Сільвестрова).** Для того, щоб квадратична форма була додатньо визначена, необхідно і достатньо, щоб усі головні мінори матриці **A** були додатні.

 *Δ*1 = *a*11, *Δ*2 = , *Δ*3 =, *Δ*2 = ⏐А⏐

 Достатня умова мінімума визначена теоремою.

  **Теорема 3.** Для того, щоб двічі неперервно-диференційована функція *n* змінних мала в критичній точці мінімум, достатньо, щоб ма­три­ця її других похідних (матриця Гессе) була додатньо визна­чена.

 Задача квадратичного програмування полягає у знаходженні мінімального розв’язку випуклої квадратичної функції на допустимій множині, заданій лінійними обмеженнями





де **x** = (*x*1, *x*2, … , *x*n) – матриця вектор-колонка, що має n компонент; **xТ** – матриця вектор-рядок, транспонована до **x** ; **rТ** – вектор-рядок коефі­цієнтів лінійної форми, що має *n* компонент; **D** – квадратна матриця коефіцієнтів квадратичної форми; A – матриця коефіцієнтів лінійних обмежень задачі розміру *m*×*n*; **b** – вектор-колонка вільних членів лінійних обмежень.

Оскільки квадратна матриця **D** квадратичної форми є симетри­чною, то функцію мети можна переписати у вигляді



де Q - матриця других похідних квадратичної форми по змінних *x*1, … *x*n .

***Q***= (***qij***) =

Згідно теоремі Куна-Таккера точка мінімума **x**(*x*1,*x*2,…,*x*n)цільової функції на допустимій множині, визначеної умовами, можна знайти, як розв’язок системи рівнянь з додатковими змінними *λi*, *xn+i*, *i*=1, … , *m*, *μj*, *j*=1, … , *n*:



Систему лінійних рівнянь (139) - (144) за рахунок особливостей рівнянь (141)-(142) не завжди можна розв’язати методами розв’язку лінійних систем. Розв’язок може бути отриманий симплексним методом.

Для знаходження початкового базисного розв’язку системи рівнянь (139)-(140) скористаємося методом штучного базиса, причому не потрібно включати до базисних одночасно змінні *λi* і *xn+1* з одним і тим же індексом *i* та змінними *xj*, *μj* з одним індексом *j*.

Побудуємо функцію *g(x)* = *xn+1+ xn+2+…+ xn+m .* Потрібно знайти мінімум функції *g(x)* при обмеженнях виду *a*il *x*1 + *a*i2 *x*2 + … + *x*n+m = *bi*, *i*=1, … , m , *xk*≥0, *k* = 1, … , *n*+*m*.

Змінні *xn+1+ xn+2+…+ xn+m*  можна взяти за базисні, точка (0,0, … , *b*1, *b*2, … , *b*m ) – початкові базисні розв’язки симплексного методу.

В результаті отримуємо точку **x**(0) = (*x*1(0), *x*1(0), … , *x*n(0), *x*n+1(0), … , *x*n+m(0)), яка мінімізує *g(x)* при обмеженнях.

*Приклад*.

*Розв’язати задачу квадратичного програмування*

*z(****x****) = x12 + x22 - 10x1-15x2→(min),*

*при обмеженнях*

*2x1+3x2 ≤ 13,*

*2x1+x2 ≤ 10,*

*x1, x2 ≤ 0.*

 *Розв’язок.*

Матриця Q(*x*1,*x*2) =  - матриця других похідних додатньо визна­чена *Δn* > 0, тому *z*(**x**) – випукла. Побудуємо функцію Лагранжа

F(**x**, **λ**, **μ**) = *x*12 + *x*22 - 15*x*2 + *λ*1(2*x*1+3*x*2 – 13) + *λ*2(2*x*1+*x*2 – 10) – *μ*1 *x*1– *μ*2 *x*2.

Система (139)-(144) набере вигляду

2*x*1 + 2*λ*1 + 2*λ*2 – *μ*1 = 10, (1\*\*)

2*x*2 + 3*λ*1 + 2*λ*2 – *μ*2 = 15, (2\*\*)

2*x*1 + 3*x*2 + *x*3 = 13, (3\*\*)

2*x*1 + *x*2 + *x*4 = 10, (4\*\*)

*μ*1*x*1= 0, *μ*2*x*2= 0, *λ*1*x*3 = 0, *λ*2*x*4 = 0. (5\*\*)

У рівняння (3\*\*) та (4\*\*) введемо додаткові змінні *x*3 та *x*4 , для пе­ре­тво­рення нерівностей системи обмежень у рівняння. Розв’язки задачі будемо шукати методом штучного базису, введені штучні змінні *u*1 і *u*2 виразимо з рівнянь (1\*\*) і (2\*\*) відповідно. Базисний розв’язок склада­ють змінні *x*3, *x*4, *u*1, *u*2 , проте він не є опорним. Виразимо штучну цільову функцію *W* = *u*1+ *u*2 виразимо через вільні невідомі *x*1,*x*2, *λ*1, *λ*2, *μ*1, *μ*2.

*W* = 25 - 2*x*1 - 2*x*2 - 5*λ*1 - 3*λ*2 + *μ*1 + *μ*2 .

*W* + 2*x*1 + 2*x*2 + 5*λ*1 + 3*λ*2 - *μ*1 -*μ*2 = 25.(6\*\*)

Нижче подана послідовність симплекс таблиць. Рамками обведені ключові елементи, а колами – ті елементи, які на даному етапі розв’язку не можна переводити з вільних невідомих в базисні згідно з умов (5\*\*). Перетворення здійснюється до відсутності додатних коефіцієнтів в “0”-рядку.

Таблиця n+1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №іте­рації | № ряд­ка | Базисні невідомі | Опор­ний розв’язок | *x*1 | *x*2 **↓** | *λ*1 | *λ*2 | *μ*1 | *μ*2 |
|  | 0 | *w* | 25 | 2 | 2 | 5 | 3 | -1 | -1 |
| І | 1 | *u*1 | 10 | 2 | 0 | 2 | 2 | -1 | 0 |
|  | 2 | *u*2 | 15 | 0 | 2 | 3 | 1 | 0 | -1 |
|  | 3 | *x*3**→** | 13 | 2 | 3 | 0 | 0 | 0 | 0 |
|  | 4 | *x4* | 10 | 2 | 1 | 0 | 0 | 0 | 0 |

 В “0” – стрічці таблиці n+1 є додатні коефіцієнти серед незаборонених при *x*1 і *x*2. Виберемо змінну *x*2 і переведемо її в базисну, для того знайдемо  тому змінна *x*3 стає вільною, а  *x*2 – базисною.

Перехід від одніє таблиці до наступної здійснюється за правилами для симплекс таблиць (див. тема 4).

Таблиця n+2.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №іте­рації | № ряд­ка | Базисні невідомі | Опор­ний розв’язок | *x*1 | *x*3 | *λ*1 | *λ*2 | *μ*1 | *μ*2 |
|  | 0 | *w* | 49/3 | 2/3 | 0 | 5 | 3 | -1 | -1 |
| ІI | 1 | *u*1 | 10 | 2 | 0 | 2 | 2 | -1 | 0 |
|  | 2 | *u*2 | 19/3 | -4/3 | 0 |  3  | 1 | 0 | -1 |
|  | 3 | *x*2 | 13/3 | 2/3 | 1 | 0 | 0 | 0 | 0 |
|  | 4 | *x4* | 17/3 | 4/3 | 0 | 0 | 0 | 0 | 0 |

В “0” – стрічці таблиці 2 максимальний додатній коефіцієнт серед дозволених при λ1, а  тому змінна *u*2 стає вільною, а  *λ*1 – базисною. Перейдемо до третьої ітерації.

Таблиця n+3.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №іте­рації | № ряд­ка | Базисні невідомі | Опор­ний розв’язок | *x*1 | *x*3 | *u*2 | *λ*2 | *μ*1 | *μ*2 |
|  | 0 | *w* | 52/9 | 26/9 | 0 | 0 | 4/3 | -1 | 2/3 |
| ІII | 1 | *u*1 | 52/9 | 26/9 | 0 | 0 | 4/3 | -1 | 2/3 |
|  | 2 | *λ*1 | 19/9 | -4/9 | 0 | 1 | 1/3 | 0 | -1/3 |
|  | 3 | *x*2 | 13/3 | 2/3 | 1 | 0 | 0 | 0 | 0 |
|  | 4 | *x4* | 17/3 | 4/3 | 0 | 0 | 0 | 0 | 0 |

Ключовий елемент 26/9 в таблиці n+3 змінну *x*1 переводить в базисну, а змінна *u*1 стає вільною.

У таблиці n+4 “0”- рядок заповнений нулями, функція W набуває мінімального значення.

Таблиця n+4.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №іте­рації | № ряд­ка | Базисні невідомі | Опор­ний розв’язок | *u*1 | *x*3 | *u*2 | *λ*2 | *μ*1 | *μ*2 |
|  | 0 | *w* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ІII | 1 | *x*1 | 2 | 1 | 0 | 0 | 6/13 | -9/26 | 3/13 |
|  | 2 | *λ*1 | 3 | 0 | 0 | 1 | 7/13 | 2/13 | -3/13 |
|  | 3 | *x*2 | 3 | 0 | 1 | 0 | -4/13 | 3/13 | -2/13 |
|  | 4 | *x4* | 3 | 0 | 0 | 0 | 8/13 | 3/13 | 4/13 |

Шуканий розв’язок задачі квадратичного програмування z(x1,x2)→(min) = z(2;3) = -52

**Навчальне обладнання, ТЗН, презентація тощо:** ноутбук, ПЕОМ.

**Питання і завдання студентам для контролю знань.**

1. Що таке ЗДО?
2. Які ви знаєте властивості ЗДО?
3. Що таке система обмежень?
4. Що таке цільова функція?
5. Що таке многокутник розв’язків?
6. Загальна форма задачі про розподіл ресурсів.
7. Загальна форма задачі про розкрій матеріалу.
8. Загальна форма транспортної задачі.

**Укладач: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_**Мищишин О.Я. доцент, к. ф.-м.н., доцент

 (підпис) (ПІБ, посада, науковий ступінь, вчене звання)