Методичні вказівки до виконання лабораторних робіт з курсу:

„Методи оптимізації”

для студентів 5-го курсу денної форми навчання

спеціальності 7.091401 “Системи управління і автоматики”

ЗМІСТ

ВСТУП

При підготовці до лабораторних робіт студенти вивчають методичні вказівки до їх виконання, конспект лекцій, рекомендовану літературу, а також виконують підготовчу роботу у відповідності до теми завдання.

При підготовці до виконання лабораторних робіт необхідно дати відповіді на наведені контрольні запитання. Глибоке вивчення теоретичного матеріалу, набутті навики програмування та вміння використовувати пакети прикладних програм допоможуть студентам успішно виконати лабораторні роботи.

По виконаній роботі оформлюється звіт. Звіт має бути віддрукований на аркушах формату А4. В окремих випадках допускається написання звіту від руки (чітким та розбірливим почерком, чорною пастою) або його представлення в електронному вигляді.

Лабораторна робота №1

Тема: Методи пошукової оптимізації функцій однієї змінної

Мета: дослідити використання методів звуження інтервалу для розв'язання задачі одномірної оптимізації.

Теоретичні відомості

Оптимізація – це пошук найкращого рішення. В кожній задачі оптимізації використовуються такі поняття як критерій, керовані змінні та цільова функція.

Критерій – це показник, який дозволяє визначити якість отриманого рішення задачі.

Керовані змінні – це такі параметри задачі, значення яких можна змінювати.

Цільова функція – це функція, що пов'язує керовані змінні та критерії таким чином, що дозволяє обчислити значення критерію при довільних значеннях керованих змінних.

Методи звуження інтервалу використовують таку теорему: якщо цільова функція *f(x)* унімодальна, неперервна і в точці x^* має оптимум, то для точок x_1, x_2, визначених за умовою $a < x_1 < x_2 < b$ існує два правила:

1) якщо функція $f(x_1) > f(x_2)$, то оптимум x^* знаходиться на відрізку $[x_1;b]$, $a \to x_1$;
2) якщо $f(x_1) < f(x_2)$, то оптимум x^* знаходиться на відрізку $[x_2;a]$, $b \to x_2$;
3) якщо не виконуються умови першого та другого правила, то оптимум знаходиться на відрізку $[x_1;x_2]$, $a \to x_1$, $b \to x_2$.

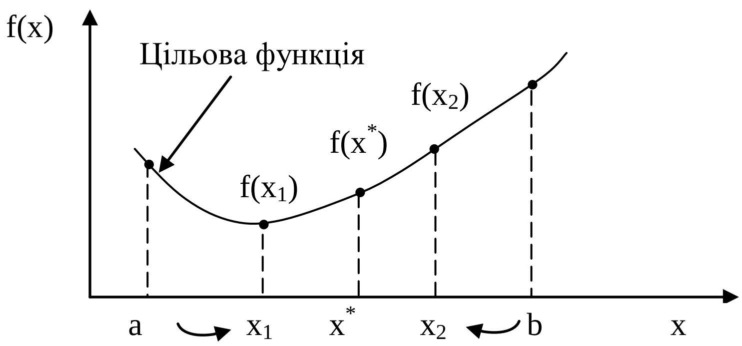


Рис. 1. Метод половинного ділення

З використанням даної теореми розроблено декілька алгоритмів пошуку оптимуму:

1) алгоритм половинного ділення;
2) метод дихотомії;
3) алгоритм методу золотого перерізу;
4) Фібоначчі.

Порядок виконання роботи

1. Скласти блок-схему алгоритму та розробити програму оптимізації функції $y = f(x)$, унімодальної на інтервалі $[a; b]$. Вихідні дані брати з таблиці 1 у відповідності до варіанту.

Таблиця 1

Варіанти завдань

Варі-ант	Цільова функція $f(x)$	Інтервал невизначеності $[a; b]$	Тип шуканого оптимуму	Точність рішення ε	Метод оптимізації
1	$y = 2^x / (0{,}5 - x)$	$[0{,}6; 3{,}6]$	max	0,01	половинного ділення
2	$y = (x-5)^2 + e^{x/2}$	$[-10; 10]$	min	0,1	дихотомії
3	$y = -e^{2+0{,}1x}$	$[-1; 1]$	max	0,01	золотого перерізу
4	$y = \sqrt{x^2 - 5x + 12}$	$[-20; 20]$	min	0,1	Фібоначчі
5	$y = \sin(x)\ln(x)$	$[13; 17]$	max	0,01	половинного ділення
6	$y = -x \cdot e^{-0{,}1x}$	$[0; 50]$	min	0,1	дихотомії
7	$y = -x^2 + 3x - 12$	$[-3; 3]$	max	0,01	золотого перерізу
8	$y = -\lvert x^3 - 50x \rvert$	$[0; 7]$	min	0,1	Фібоначчі

2. Результати пошуку оптимуму функції представити:

а) для методу половинного ділення таблицею вигляду

k	$a^{(k)}$	$b^{(k)}$	$L^{(k)}$	$x_1^{(k)}$	$x_m^{(k)}$	$x_2^{(k)}$	$y_1^{(k)}$	$y_m^{(k)}$	$y_2^{(k)}$
1									
...									
k									
$k+1$									

б) для методу дихотомії таблицею вигляду

k	$a^{(k)}$	$b^{(k)}$	$L^{(k)}$	$x_1^{(k)}$	$x_m^{(k)}$	$x_2^{(k)}$	$y_1^{(k)}$	$y_2^{(k)}$
1								
...								
k								
$k+1$								

в) для методів золотого перерізу і Фібоначчі таблицею вигляду

k	$a^{(k)}$	$b^{(k)}$	$L^{(k)}$	$x_1^{(k)}$	$x_2^{(k)}$	$y_1^{(k)}$	$y_2^{(k)}$
1							
...							
k							
$k+1$							

де k – номер кроку пошуку; L – довжина інтервалу невизначеності.

3. Достовірність отриманих результатів перевірити, використовуючи математичний пакет MathCAD.

4. Зробити висновки. Оформити звіт

Склад звіту

1. Титульний аркуш.
2. Короткі теоретичні відомості.
3. Завдання.
4. Блок-схема та лістинг програми.
5. Результати оптимізації за розробленою програмою.
6. Результати дослідження у MathCAD.
7. Висновки.

Контрольні запитання

1. Що таке цільова функція і проектні параметри ?
2. Що таке унімодальна функція ?
3. Особливості алгоритмів оптимізації одномірних функцій.
4. Суть алгоритму методу половинного ділення.
5. Як можна підвищити ефективність методу оптимізації ?
6. Суть алгоритму методу золотого перерізу та Фібоначчі ?

Література

1. Методи оптимізації складних систем. Навчальний посібник. І.В.Кузьмін, М.М.Биков, С.М.Москвіна. – Вінниця: ВДТУ, 2003.
2. Жилинскас А., Шалтянис В. Поиск оптимума. – М.:Наука.-1989. С. 22-28.
3. Банди Б. Методы оптимизации. Вводный курс. – М.:”Радио и связь”, 1988
4. Дегтярев Ю.И. Методы оптимизации: учебное пособие. – М.: Советское радио, 1980.
5. Ашманов С.А., Тимохов А.В. Теория оптимизации в задачах и упражнениях. – М.: Наука, 1991.
6. Полак Е. Чисельні методи оптимізації. – М.: Мир, 1974.
7. Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. – М.: Наука, 1986.

Лабораторна робота №2

Тема: Методи нульового порядку функції багатьох змінних

Мета: дослідити використання методів нульового порядку для розв'язання задачі багатомірної оптимізації

Теоретичні відомості

До **методів нульового порядку** відноситься пошук за сімплексом (не плутати з сімплекс методом в лінійному програмувані), метод покоординатного спуску Гауса-Зейделя, випадковий пошук та інші. Розглянемо основні ідеї означених методів.

Пошук за симплексом [4] полягає в тому, що при пошуку оптимума в просторі незалежних змінних будується регулярний сімплекс і обчислюється значення цільової функції в його вершинах. Регулярний сімплекс в n-мірном просторі представляє собою багатограник, який має n+1 рівновіддалених вершин. Наприклад, у випадку двох зміних сімплексом є рівнобічний трикутник; в трьохвимірному просторі сімплекс представляє собою тетраедр. Після побудови сімплекса визначається вершина, яка має найбільше значення цільової функції. На рис.2 це вершина з номером 1.

Далі знайдена вершина проектується через центр ваги інших вершин симплекса (точки 2 і 3) в нову вершину (точка 4). Точки 2, 3 та 4 використовуються в якості вершини нового симплекса. Таким чином, трикутник немов би перевертається через сторону з найменшим значенням цільової функції. При пошуці мінімуму використовуються наступні два правила.

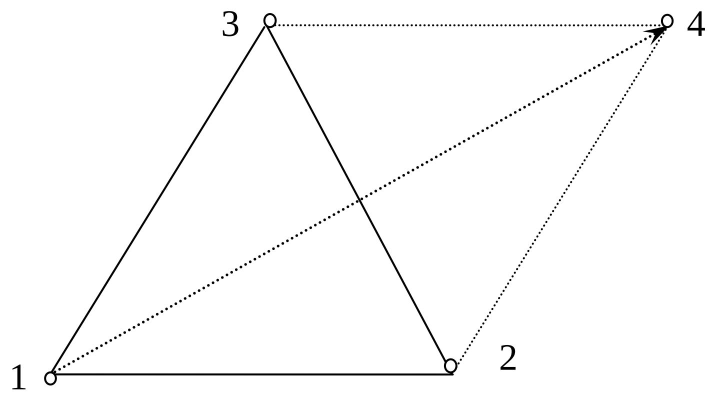


Рис. 2. Ілюстрація ідеї метода пошуку за симплексом

Правило "накриття" точки мінімуму

Якщо вершина, якій відповідає найбільше значення цільової функції, побудована на попередній ітерації, то замість неї береться вершина з меньшим значенням цільової функції.

Правило циклічного руху

Якщо деяка вершина сімплекса не виключається на протязі багатьох ітерацій, то необхідно зменшити розмір сімплекса.

Пошук завершується, коли розмір сімплекса або різниця між значеннями функції в вершинах сімплекса стають достатньо малими. Недоліком цього метода є велика кількість ітерацій; крім того він незавжди забезпечує розв'язок задачі.

Ідея метода **покоординатного спуску** полягає в тому, що спочатку робиться пробний крок в напрямі, який паралельний до однієї з координатної вісі і обчислюється значення цільової функції. Якщо значення цільової функції зменшується, то рух продовжується далі в цьому ж напрямку, а якщо функція збільшується, то ми повертаємося назад і робимо пробний крок в іншому напрямку. І так продовжується до тих пір, доки не буде знайдена оптимальна точка (рис 3). Особливість цього методу полягає в тому, що пошук оптимуму проводиться виключно паралельно координатним оосям. На початку пошуку вибирається великий крок і перевіряється значення функції по всім напрямкам. Якщо буде зростання функції по всім напрямкам, то необхідно зменшити крок.

Недоліком цього метода є обмежені можливості при пошуку оптимуму. Метод застосовують тоді, коли залежність між змінними x_1, x_2,…, x_n практично відсутні.

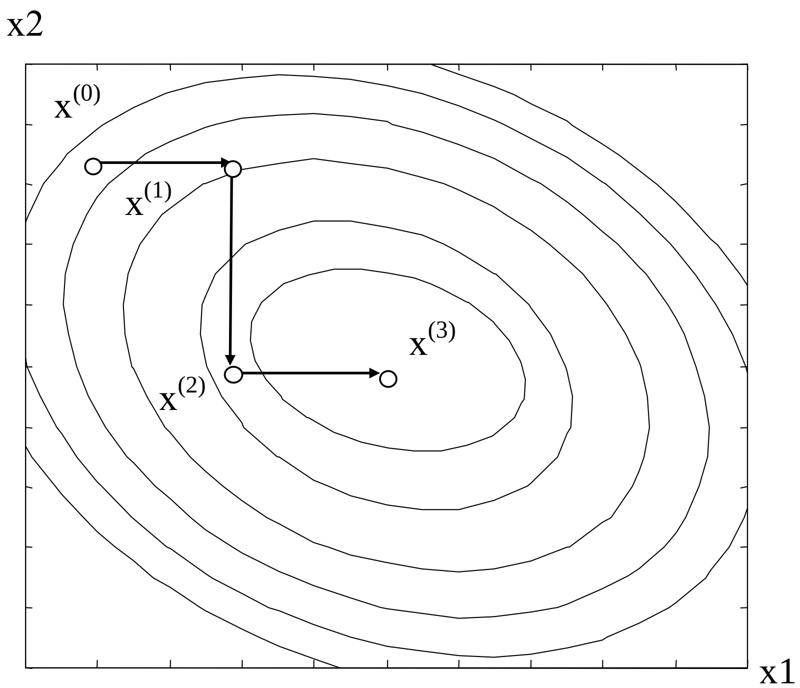


Рис. 3. Ілюстрація покоординатного спуску

Ідея **випадкового пошуку** полягає в тому, що вибір напрямку руху здійснюється випадково. Якщо цільова функція зменшується, то рух в вибраному напрямку продовжується, в протилежному випадку необхідно повернутися на один крок назад та знову випадково обрати напрямок пошуку і т.д.

Усі випадкові методи пошуку реалізуються за ітераційною формулою:

$$X^{(k+1)} = X^{(k)} + \xi^{(k)},$$

де k – номер ітерації;

ξ,$^{(k)}$ - випадкова величина.

Популярність методів випадкового пошуку пояснюється їхньою простотою та широкими можливостями для користувачів самому модифікувати методи.

Порядок виконання роботи

1. Скласти схему алгоритму та програму оптимізації функції $y = f(x_1, x_2)$ методом нульового порядку. Вихідні дані брати з таблиці 2 у відповідності до варіанту.

2. Достовірність отриманих результатів перевірити, використовуючи математичний пакет MathCAD.

3. Зробити висновки. Оформити звіт.

Таблиця 2

Варіанти завдань

Варіант	Цільова функція y	Точність рішення ε	Метод оптимізації
1	$y=(x_1-2)^2+(x_2-4)^2$	0,001	Хука-Дживса
2	$y=x_1^2+x_2^2-x_1x_2$	0,0001	Покоординатного спуску
3	$y=e^{x_1^2+x_2^2}+2x_1-3{,}5x_2$	0,001	Нелдера-Міда
4	$y=(x_1+3)^2+(x_2+5)^2$	0,01	Хука-Дживса
5	$y=e^{2x_1^2+x_2^2}+1{,}1x_1+3{,}6x_2$	0,001	Покоординатного спуску
6	$y=e^{x_1^2+0{,}8x_2^2}+1{,}2x_1+2x_2$	0,01	Нелдера-Міда

*Вибрати початкові параметри в залежності від особливостей методу й обраної функції.

Склад звіту:

1. Титульний аркуш.
2. Короткі теоретичні відомості.
3. Завдання.
4. Блок-схема та лістинг програми.
5. Результати оптимізації за розробленою програмою.
6. Результати дослідження у MathCAD.
7. Висновки.

Контрольні запитання

1. Класифікація методів безумовної оптимізації функцій багатьох змінних.
2. Особливості методів нульового порядку оптимізації багатомірних функцій.
3. Чому методи нульового порядку мають таку назву? Що мається на увазі?
4. Суть методів покоординатного спуску, Хука-Дживса та Нелдера-Міда.

Література

1. Методи оптимізації складних систем. Навчальний посібник. І.В.Кузьмін, М.М.Биков, С.М.Москвіна. – Вінниція: ВДТУ, 2003.
2. Банди Б. Методы оптимизации. Вводный курс. – М.:”Радио и связь”, 1988
3. Дегтярев Ю.И. Методы оптимизации: учебное пособие. – М.: Советское радио, 1980.
4. Ашманов С.А., Тимохов А.В. Теория оптимизации в задачах и упражнениях. – М.: Наука, 1991.
5. Полак Е. Чисельні методи оптимізації. – М.: Мир, 1974.
6. Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. – М.: Наука, 1986.

Лабораторна робота №3

Тема: Градієнтні методи оптимізації функцій багатьох змінних

Мета: дослідити використання градієнтних методів для розв'язання задачі багатомірної оптимізації.

Теоретичні відомості

Градієнтні методи відносяться до методів першого порядку. Особливість цих методів – це використання градієнта.

Градієнтом функції f(X) називається вектор, величина якого визначає швидкість зміни функції f(X), а напрямок співпадає з напрямком найбільшого зростання цієї функції. Вектор, який протилежний за напрямком градієнту називається *антиградієнтом*.

Всі методи першого порядку, базуються на ітераційній процедурі, яка реалізується за формулою:

$$X^{(k+1)} = X^{(k)} + \lambda^{(k)} \cdot \nabla f(X^{(k)}),$$

де k - номер ітерації;

$X^{(k)}$ – поточне наближення до розв'язку;

$\lambda^{(k)}$ –параметр, який характеризує довжину кроку;

$\nabla f(X^{(k)})$ – напрямок пошуку в n-мірному просторі керованих змінних.

Градієнтний метод є послідовністю кроків, кожний з яких складається з двох операцій:

1) визначення напрямку найбільшої крутизни спуска, тобто напрямок антиградієнта функції f(X).
2) переміщення в вибраному напрямку на задану відстань.

Градієнтний метод має свої недоліки. Для того, щоб рухатися завжди в напрямку антиградієнта, крок повинен бути невеликим. Тому їх буде багато. Але оскільки на кожному кроці необхідно постійно обчислювати градієнт, то наближення до оптимуму буде повільним.

Метод **найшвидшого спуску** є модифікацією градієнтного метода. Він відрізняється від градієнтного тим, що градієнт обчислюється тільки в початковій точці і рух в напрямку антіградієнта продовжується до тих пір, поки зменшується значення цільової функції f(X). Вибір кроку λ при переході від точки $X^{(k)}$ в $X^{(k+1)}$ визначається згідно умови:

$$\lambda^{(k)} = \min_{\lambda \geq 0} f(X^{(k)} + \lambda^{(k)} \cdot \nabla f(X^{(k)})),$$

тобто на кожному кроці розв'язується одномірна задача мінімізації.

Метод найшвидшого спуску має два недоліка: по-перше, методу властива поступова збіжність до точки мінімума внаслідок малого $\nabla f(X)$ в околиці цієї точки, по-друге, необхідно розв'язувати задачу одномірної оптимізації – обирати на кожному кроці оптимальне значення $\lambda^{(k)}$. Головна перевага цього метода полягає в тому, що йому властива стійкість та простота обчислень. Використовують цей метод тоді, коли цільову функцію можна добре апроксимувати лінійною залежністю.

Порядок виконання роботи:

1. Скласти схему алгоритму та розробити програму пошуку локального мінімуму функції $y = f(x_1, x_2)$ градієнтним методом. Вихідні дані брати з таблиці 3 у відповідності до варіанту.

Таблиця 3

Варіанти завдань

Варіант	Цільова функція y	Початкова точка x^0	Початкова довжина кроку λ	Точність рішення ε	Метод оптимізації
1	$y = (x_1 - 2)^2 + (x_2 - 4)^2$	$[1{,}2; 3{,}4]$	0,1	0,001	Флетчера-Рівса
2	$y = x_1^2 + x_2^2 - x_1 x_2$	$[0{,}1; 0]$	0,01	0,0001	Найшвидшого спуску
3	$y = e^{x_1^2 + x_2^2} + 2x_1 - 3{,}5x_2$	$[0; 0]$	0,1	0,001	Градієнтного спуску
4	$y = (x_1 + 3)^2 + (x_2 + 5)^2$	$[-4; -9]$	0,1	0,01	Флетчера-Рівса
5	$y = e^{2x_1^2 + x_2^2} + 1{,}1x_1 + 3{,}6x_2$	$[-1; 0{,}5]$	0,1	0,001	Найшвидшого спуску
6	$y = e^{x_1^2 + 0{,}8x_2^2} + 1{,}2x_1 + 2x_2$	$[1; -0{,}2]$	0,1	0,01	Градієнтного спуску

2. Результати пошуку оптимуму функції представити таблицею вигляду:

Крок пошуку k	Поточна точка x	Значення функції y	Поточна довжина кроку λ	Градієнт G	Норма градієнта $\|G\|$
1					
. . .					
k					
. . .					
N					

3. Достовірність отриманих результатів перевірити, використовуючи математичний пакет MathCAD.

4. Зробити висновки. Оформити звіт.

Склад звіту

1. Титульний аркуш.
2. Короткі теоретичні відомості.
3. Завдання.
4. Блок-схема та лістинг програми.
5. Результати оптимізації за розробленою програмою.
6. Результати дослідження у MathCAD.
7. Висновки.

Приклад програми пошуку оптимуму функції двох змінних у MathCAD.

1. Записуємо цільову фнкцію

$$f(x1,x2) := e^{\left(x1^2+x2^2\right)} + 2\cdot x1 - 3.5x2$$

2. Визначаємо і виводимо частинні похідні цільової функції f'x1(x1,x2) і f'x2(x1,x2)

$$fx1(x1,x2) := \frac{d}{dx1}f(x1,x2) \quad fx1(x1,x2) \rightarrow 2\cdot x1\cdot \exp\left(x1^2 + x2^2\right) + 2$$

$$fx2(x1,x2) := \frac{d}{dx2}f(x1,x2) \quad fx2(x1,x2) \rightarrow 2\cdot x2\cdot \exp\left(x1^2 + x2^2\right) - 3.5$$

3. Розвязуємо систему рівнянь f'x1(x1,x2)=0 f'x2(x1,x2)=0, використовуючи процедуру given-find

Given

$fx1(x1,x2) = 0$

$fx2(x1,x2) = 0$

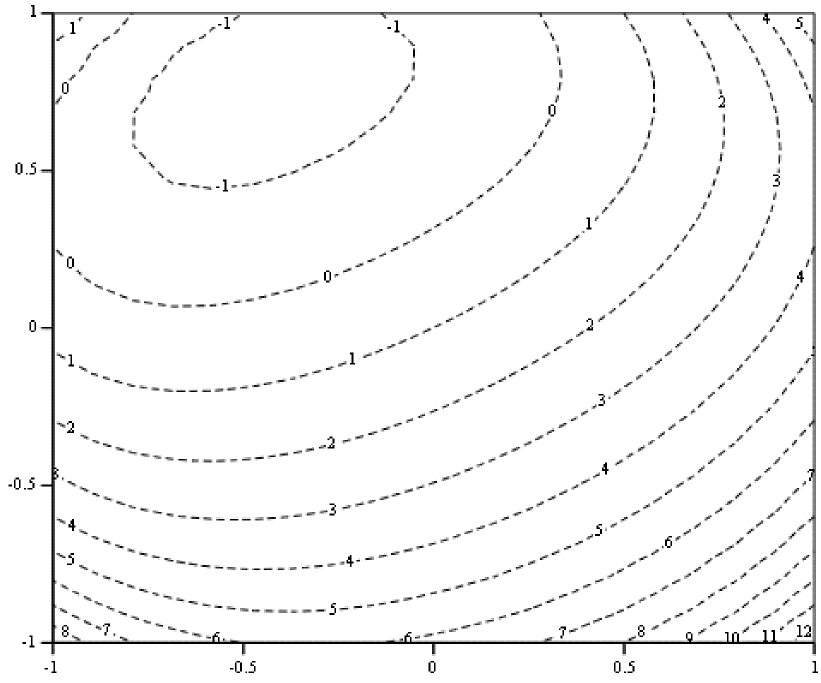
$$Find(x1,x2) \rightarrow \begin{pmatrix} -.44588749512269884770 \\ .78030311646472298348 \end{pmatrix}$$

4. Виводимо значення цільової функції в точці мінімуму x=-0.4459 y=0.7803

$f(-0.4459, 0.7803) = -1.38$

5. Будуємо лінії рівня цільової функції, використовуючи функцію CreateMesh

$Z := CreateMesh(f, -1, 1, -1, 1)$



Z

Контрольні запитання

1. Особливості алгоритмів оптимізації багатомірних функцій.
2. Що таке градієнт та антиградієнт?
3. В чому полягає суть градієнтних методів пошуку оптимуму?

4. Чим градієнтний метод відрізняється від методу найшвидшого спуску?
5. Суть методу Флетчера-Рівса.

Література

1. Методи оптимізації складних систем. Навчальний посібник. І.В.Кузьмін, М.М.Биков, С.М.Москвіна. – Вінниця: ВДТУ, 2003.
2. Банди Б. Методы оптимизации. Вводный курс. – М.:”Радио и связь”, 1988.
3. Дегтярев Ю.И. Методы оптимизации: учебное пособие. – М.: Советское радио, 1980.
4. Ашманов С.А., Тимохов А.В. Теория оптимизации в задачах и упражнениях. – М.: Наука, 1991.
5. Полак Е. Чисельні методи оптимізації. – М.: Мир, 1974.
6. Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. – М.: Наука, 1986.

Лабораторна робота №4

Тема: Дослідження залежності часу оптимізації від розмірності задачі

Мета: дослідити залежність часу оптимізації від розмірності задачі за допомогою відомих методів.

Теоретичні відомості

Збільшення кількості керованих змінних (розмірності) істотно ускладнює розв'язання задачі оптимізації. А при деякому значені починається різке зростання часу обчислення оптимуму (рис.4). Таке явище в оптимізації називається проблемою "прокляття" розмірності.

Однією з задач сучасних методів оптимізації є розробка ефективних алгоритмів, що дозволяють відсунути стіну складності. *Ефективним алгоритмом* вважається такий алгоритм, складність якого (кількість ітерацій) описується поліноміальною функцією від розмірності задачі.

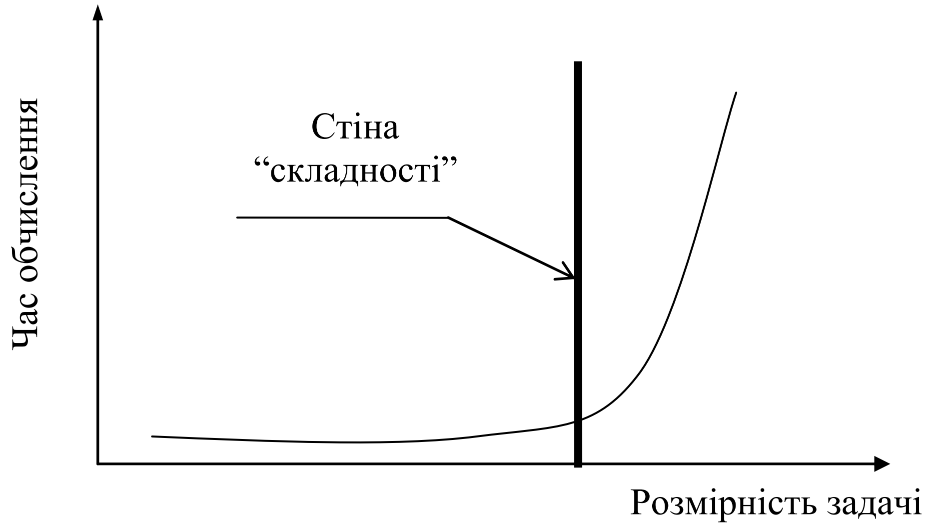


Рис. 4. Проблема "прокляття" розмірності

Порядок виконання роботи

1. Розробити програму для дослідження залежності часу розв'язання задачі безумовної оптимізації від кількості керованих змінних.

2. Вихідні дані брати з таблиці 4 у відповідності до варіанту.

Таблиця 4

Варіанти завдань

Варіант	Цільова функція $f(x)$	Початкова точка x^0
1	$y=\sum_{i=1}^{n}\frac{(x_i-n)^2}{i^2}$	$x_i^0=2i$, $i=\overline{1,n}$
2	$y=\sum_{i=1}^{n}\frac{(x_i+7i)^2}{i}$	$x_i^0=0$, $i=\overline{1,n}$

3	$y=\sum_{i=1}^{n}(x_i-i)^4$	$x_i^0=\frac{i}{2},\ i=\overline{1,n}$
4	$y=\sum_{i=1}^{n}\frac{(x_i+i)^2}{0{,}1i^2}$	$x_i^0=i,\ i=\overline{1,n}$
5	$y=\sum_{i=1}^{n}\frac{(x_i+i)^4}{i}$	$x_i^0=5i,\ i=\overline{1,n}$
6	$y=\sum_{i=1}^{n}(0{,}1x_i+2i)^2$	$x_i^0=3,\ i=\overline{1,n}$

3. Для пошуку мінімуму функції використати будь-який градієнтний метод.

4. Експеримент провести в діапазоні від $n=2$ до $n=30$.

5. Результати експерименту представити таблицею 5.

Таблиця 5

Варіанти завдань

Кількість керованих змінних n	Час пошуку оптимуму t, сек
2	
...	
30	

6. Достовірність отриманих результатів перевірити, використовуючи математичний пакет MathCAD. За допомогою методу найменших квадратів апроксимувати отримані експериментальні дані функцією $a_0+a_1e^{a_2n}$ та вивести графік цієї функції.

7. Зробити висновки. Оформити звіт.

Склад звіту

1. Титульний аркуш.
2. Короткі теоретичні відомості.
3. Завдання.
4. Блок-схема та лістинг програми.
5. Результати оптимізації за розробленою програмою.
6. Результати дослідження у MathCAD.
7. Висновки.

Контрольні запитання

1. Особливості алгоритмів оптимізації багатомірних функцій.
2. Класифікація методів безумовної оптимізації функцій багатьох змінних.
3. Як залежить час оптимізації від розмірності задачі?
4. Чи завжди при збільшенні розмірності задачі збільшується час оптимізації?
5. Які фактори впливають на час розв’язання задачі оптимізації?
6. Постановка задачі апроксимації.

Література

1. Методи оптимізації складних систем. Навчальний посібник. І.В.Кузьмін, М.М.Биков, С.М.Москвіна. – Вінниція: ВДТУ, 2003.
2. Дегтярев Ю.И. Методы оптимизации: учебное пособие. – М.: Советское радио, 1980.
3. Ашманов С.А., Тимохов А.В. Теория оптимизации в задачах и упражнениях. – М.: Наука, 1991.
4. Евдокимов А.Г. Минимизация функций и ее приложения к задачам автоматизированного управления инженерными сетями. – Х.: Вища шк., 1985. – 288 с.
5. Штовба С.Д. Методи оптимізації в середовищі Matlab. Лабораторний практикум: Навч. пос. – Вінниця, ВДТУ, 2001. – 56 с.

Приклад програми обробки результатів експерименту у MathCAD.

1. Задаємо значення змінних, необхідних для роботи з масивами

$ORIGIN := 1 \qquad N := 20 \qquad i := 2..N$

2. Вводимо значення часу оптимізації, отримані в результаті експерименту

$$T := (2\ \ 2.5\ \ 3.2\ \ 4.5\ \ 6.2\ \ 8\ \ 10.6\ \ 14\ \ 19\ \ 26\ \ 38\ \ 52\ \ 76\ \ 106\ \ 140\ \ 200\ \ 280\ \ 400\ \ 600)^T$$

3. Записуємо цільову функцію

$$g(a0, a1, a2) := \left| \begin{array}{l} z \leftarrow 0 \\ \text{for}\quad j \in 1..N-1 \\ \quad z \leftarrow z + \left[T_j - \left(a0 + a1 \cdot e^{a2 \cdot j}\right)\right]^2 \end{array} \right.$$

4. Формуємо початкове наближення

$a0 := 2 \qquad a1 := 1 \qquad a2 := 1$

5. Знаходимо мінімум функції g(a0, a1, a2) за допомогою функції Minimize

$$P := \mathrm{Minimize}(g, a0, a1, a2) \qquad P = \begin{pmatrix} 0.503 \\ 0.613 \\ 0.361 \end{pmatrix}$$

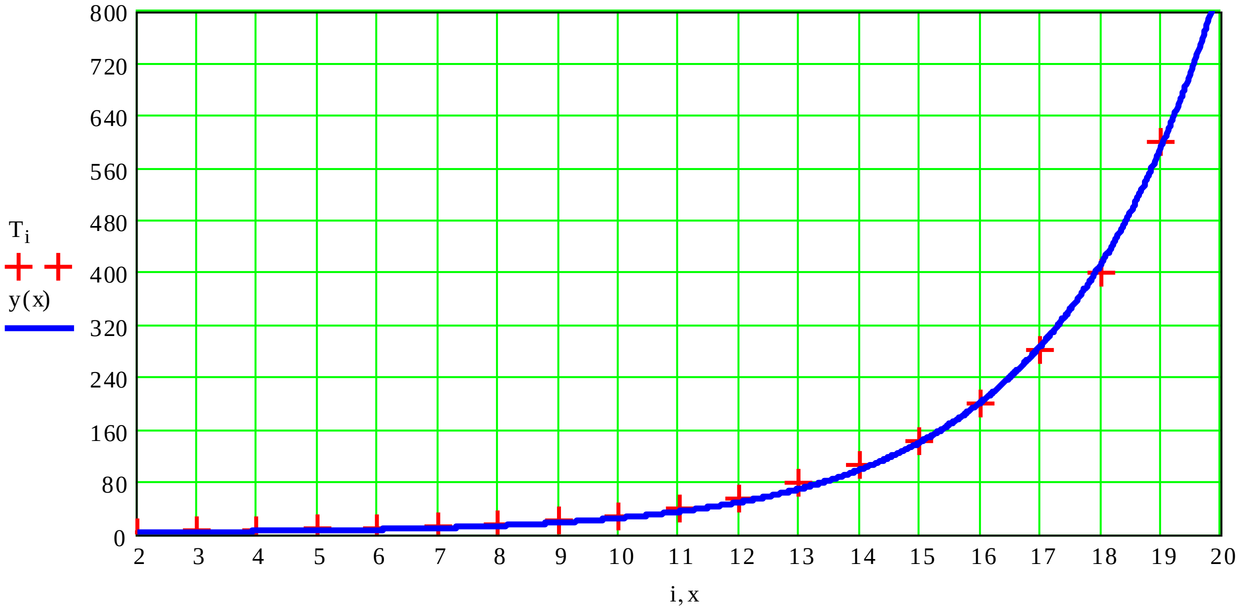
6. Визначаємо значення функції g(a0, a1, a2) в точці мінімуму

$$gmin := \left| \begin{array}{l} y \leftarrow 0 \\ \text{for}\quad j \in 1..N-1 \\ \quad y \leftarrow y + \left[T_j - \left(P_1 + P_2 \cdot e^{P_3 \cdot j}\right)\right]^2 \end{array} \right. \qquad gmin = 495.88$$

7. Задаємо отриману апроксимувальну функцію

$$y(x) := P_1 + P_2 \cdot e^{P_3 \cdot x}$$

8. Виводимо значення експериментальних даних та графік апроксимувальної функції



Лабораторна робота №5

Тема: Аналіз чутливості оптимального розв'язку задачі лінійного програмування

Мета: провести аналіз чутливості оптимального розв'язку задачі використовуючи програмні пакети MathCAD та MS Excel.

Теоретичні відомості

Аналіз чутливості – це процедура, яка дозволяє встановити залежність оптимального рішення до варіації початкових даних. Аналіз чутливості відіграє велику роль в задачах оптимізації. Аналіз чутливості потрібно проводити за двома причинами:

1. Деякі параметри задач лінійного програмування, такі, як фінансові кошти, запаси ресурсів можно регулювати. Аналіз чутливості дозволяє оцінити вплив зміни цих параметрів на оптимальне рішення. Якщо виявитися, що оптимальне рішення (наприклад, відношення прибутку до витрат) можна значно покращити за рахунок невеликих змін параметрів, то необхідно провести ці зміни.

2. В багатьох випадках оцінки параметрів отримуються шляхом статистичної обробки експерементальних даних. Тому такі оцінки не можуть бути точними. Якщо вдається визначити, які параметри в більшій степені впливають на значення цільової функції, то необхідно збільшити точність їх оцінок.

Важливу роль при аналізі чутливості виробничих задач відіграють тіньові ціни та маргінальні оцінки. Для цього використовують значення тіньових цін та маргінальні оцінки. Тіньові ціни визначають приріст максимального прибутку при використані додаткової одиниці деякого ресурсу. Значення маргінальної оцінки показує наскільки знижується максимальний прибуток при випуску одиниці цієї продукції. Більш детальна інформація про аналіз чутливості в задачах лінійного програмування наведена в [5].

Порядок виконання роботи

1. Самостійно придумати та розв'язати задачу лінійного програмування (кількість керованих змінних повинна бути не менша 4, кількість обмежень на значення керованих змінних – не менша 3).

2. Визначити зону нечутливості оптимального розв'язку задачі лінійного програмування до варіації початкових даних. Експерименти проводити у математичному пакеті MathCAD.

3. Ознайомитися з надбудовою MS Excel „Поиск решения" (див. довідкову систему і приклади розв'язку задач в Office\Samples\Solvsamp.xls) та отримати за її допомогою розв'язок задачі.

4. Зробити висновки. Оформити звіт.

Склад звіту

1. Титульний аркуш.
2. Короткі теоретичні відомості.
3. Змістовна постановка задачі. Формалізована постановка задачі.
4. Результати розв’язання задачі у MathCAD. Результати аналізу чутливості.
5. Результати розв’язання задачі у MS Excel.
6. Висновки.

Контрольні запитання

1. Постановка задачі лінійного програмування.
2. До якого класу задач оптимізації відноситься задача лінійного програмування?
3. Методи розв’язання задач лінійного програмування.
4. Основі етапи сиплекс-методу розв’язання задачі лінійного програмування.
5. З якою метою проводять аналіз чутливості оптимального розв’язку?

Література

1. Банди Б. Основы линейного программирования. – М.: Радио и связь, 1989. – 176 с.
2. Вентцель Е.С. Исследование операций. М.: Наука, 1980.
3. Пантелеев А.В. Методы оптимизации в примерах и задачах. – М.: Высш. шк., 2002. – 544 с.Ашманов С.А., Тимохов А.В. Теория оптимизации в задачах и упражнениях. – М.: Наука, 1991.
4. Штовба С.Д. Методи оптимізації в середовищі Matlab. Лабораторний практикум: Навч. пос. – Вінниця, ВДТУ, 2001. – 56 с.
5. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике. Кн.1.- М.: Мир.- 1986.- 320с.

Приклад програми обробки результатів експерименту у MathCAD.

1. Записуємо ціьову функцію

$f(x1, x2, x3, x4) := x1 + 2\cdot x2 - x3 + x4$

2. Формуємо початкове наближення (можна задати будь-які чис.

$x1 := 0 \quad x2 := 1 \quad x3 := 0 \quad x4 := 5$

3. Описуємо обмеження

Given

$x1 + 2x2 + x4 = 4$

$x1 + x2 + x3 = 8$

$x1 \geq 0 \quad x2 \geq 0 \quad x3 \geq 0 \quad x4 \geq 0$

4. Знаходимо мінімум цільової функції за допомогою функції Mir
(для пошуку максимуму використовують функцію Maximize)

$Reshenie := Minimize(f, x1, x2, x3, x4)$

5. Виводимо точку мінімуму та значення цільової функції в цій т

$Reshenie^{T} = (0 \quad 0 \quad 8 \quad 4)$

$f(0, 0, 8, 4) = -4$

Лабораторна робота № 6

Тема: Переборні методи оптимізації

Мета: дослідити використання оптимізаційних методів для комп'ютерного керування мережами.

Теоретичні відомості

Графи, в яких кожному ребру відповідає певне число називається зваженим. Інколи такі графи називають мережевими.

При описі зважених графів в алгоритмах використовується матриця відстаней. Відсутність зв'язку між вершинами в такій матриці позначається комп'ютерним аналогом нескінченності – достатньо великим числом (заздалегідь більшим за суму довжин всіх ребер).

Метод перебору є найпростішим і найнеефективнішим. Він дозволяє знайти розв'язок у будь-якому випадку (якщо розв'язок взагалі існує), але кількість варіантів, які необхідно перебрати найчастіше надто велика.

Метод гілок та границь дозволяє скоротити кількість варіантів у порівнянні з методом перебору. Застосовується в задачах, де критерій пошуку є монотонно зростаючою функцією кількості ребер.

Ідея методу: Знаходиться перший шлях, що задовольняє умові задачі. При пошуку іншого шляху з додаванням до нього чергового ребра перевіряємо, чи не став вже цей частковий шлях довшим за початковий (порівнюємо з границею). Якщо став, то далі нарощувати його немає сенсу – він вже заздалегідь гірший, і всі наступні варіанти (гілки), що мають своїм початком цей частковий шлях, не розглядаються.

У випадках, коли шуканий розв'язок задачі повинен або може містити не всі вершини графа, а певну їх підмножину, з'являється необхідність генерування підмножин множини вершин.

Алгоритм генерування підмножин грунтується на внутрішньому представленні множини в пам'яті комп'ютера. В пам'яті кожному елементу множини відповідає один біт (1 – елемент належить множині, 0 – не належить), а всій множині – послідовність бітів, довжина якої дорівнює кількості елементів множини. Тому для генерування підмножин достатньо встановлювати окремі біти внутрішнього представлення у стан 1 або 0. Це простіше всього зробити, сумістивши у пам'яті множину з цілим числом за допомогою опису absolute і перебираючи значення цілого числа (при цьому змінюються його біти, а відтак і множина).

Порядок виконання роботи

1. Скласти програму оптимізації КСК у відповідності до варіанта
   1) оптимізація кільцевої мережі по довжині кабелю. Використати метод прямого перебору.
   2) оптимізація лінійної мережі по довжині кабелю. Використати метод гілок та границь.
   3) оптимізувати розподіл технологічного обладнання на групи (5

груп з 15 одиниць) по критерію рівної завантаженості. Використати генерацію підмножин.

4) оптимізація технологічного процесу по витратам часу. Використати метод прямого перебору.
5) оптимізація складу технологічного обладнання. Використати метод генерації підмножин.

2. Оцінити ефективність алгоритму кількість циклів, обсяг пам’яті, час використання.

Склад звіту

1. Титульний аркуш.
2. Короткі теоретичні відомості
3. Завдання.
4. Блок-схема та лістинг програми.
5. Результати оптимізації за розробленою програмою.
6. Результати оптимізації
7. Висновки

Контрольні запитання

1. Як описуються множини у програмі?
2. Як описуються графи у програмі?
3. Як використовується внутрішнє представлення множин для генерації підмножин?
4. У чому сутність методу гілок та границь?

Література

1. Макс Ж. Методы и техника обработки сигналов при измерениях .В 2-х томах – М.: Мир, 1983.
2. В.М.Дубовой, Р.Н.Квєтний. Програмування комп'ютеризованих систем управління та автоматики. - Вінниця: ІЗМН, 1997. - 208 с.
3. Дубовой В.М., Квєтний Р.Н. Програмування персональних комп'ютерів систем управління. – В.: ВДТУ, 1999.

Лабораторна робота №7

Тема: Пошукові задачі оптимізації

Мета: дослідити використання оптимізаційних методів на графах.

Теоретичні відомості

Пошук в глибину (depth first search – dfs).

Ідея алгоритма:

1) Починаючи з вихідної вершини V0 побудувати шлях у графі, використовуючи всі ребра з мінімальним номером кінцевої вершини. Послідовність вершин занести у *стек*.
2) Якщо на цьому шляху не з'явиться кінцева вершина VN – вилучити останню вершину з стека і взяти ребро з більшим номером кінцевої вершини.
3) Якщо невикористаних ребер немає – вилучити вершину зі стека.

Алгоритм пошуку в глибину знаходить перший – але не найкоротший, підходящий шлях.

Пошук в ширину (bredth first search – bfs).

Ідея алгоритма:

1) Вершини при пошуку в ширину заносити в *чергу*. Елемент черги складається з двух частин: номер вершини та номер елемента черги, звідки до цієї вершини потрапили.
2) До черги заноситься початкова вершина, а у другу частину елемента – 0 (ознака початку шляху).
3) Розглядаються *всі* вершини, суміжні з початковою, і заносяться до черги.
4) Якщо серед розглянутих вершин немає кінцевої – пересунути вказівник початку черги і розглянути вершини, суміжні з тією, на яку вказує вказівник. Дійшовши таким чином до кінцевої вершини, пройти шлях у зворотному порядку, використовуючи другу частину елементів черги.

Пошук з поверненням

Головна риса пошуку з поверненням, яка відрізняє цей алгоритм від пошуку в глибину полягає в тому, що розглянуті вершини графа, шлях через які не задовольнив умові задачі, не виключаються з подальшого розгляду, а повертаються до множини вершин.

Порядок виконання роботи

1. Скласти програму оптимізації КСК у відповідності до варіанту
   1) Знайти маршрут передачі повідомлення комп'ютерною мережею, задана графом рис.5. Використати метод пошуку в ширину.
   2) Аналогічно варіанту 1. Використати метод пошуку в глибину
   3) Технологічний процес описаний мережевим графіком рис.6. Цифри над дугами – це час виконання операцій. Знайти критичний шлях. Використати метод гілок та границь.
   4) Аналогічно вар.3. Використати метод пошуку з поверненням.

5) Послідовність логічних висновків на базі знань задана графом переходів рис.7. Побудувати найкоротшу послідовність доведення висновку 6 на основі даних 1.

2. Послідовність кроків пошуку писати у текстовий файл.
3. Проаналізувати кількість пошукових кроків алгоритму.

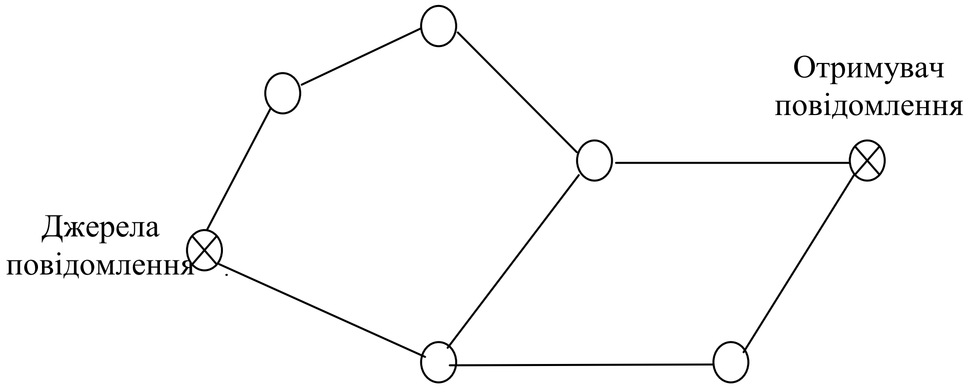


Рис. 5 Знайти маршрут передачі повідомлення комп’ютерною мережею

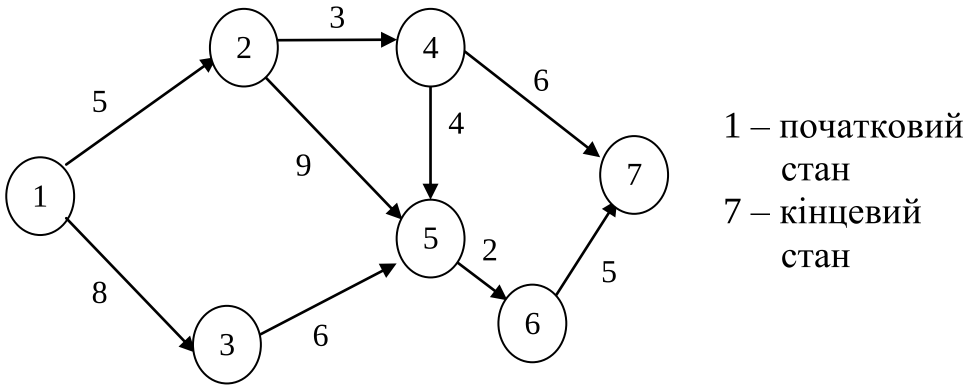


Рис. 6 Технологічний процес

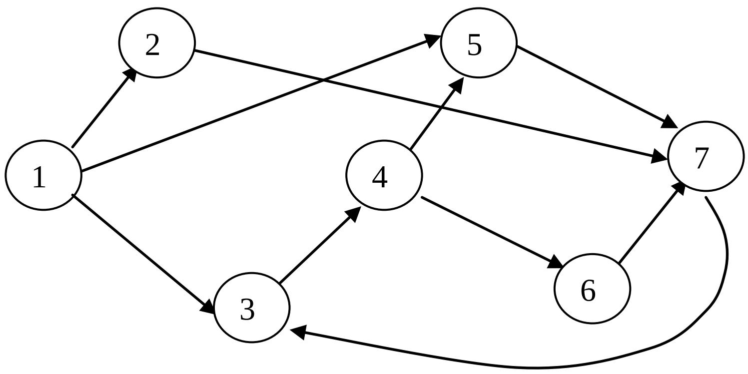


Рис. 7 Послідовність логічних висновків на базі знань

Склад звіту

1. Мета роботи
2. Короткі теоретичні відомості
3. Текст програми
4. Послідовність кроків пошуку
5. Висновки

Контрольні запитання

1. У чому сутність методу пошуку в ширину? Його геометрична інтерпретація.
2. Яка структура даних використовується у методі пошуку в ширину.
3. У чому сутність методу пошуку в глибину? Його геометрична інтерпретація.
4. Яка структура даних використовується у методі пошуку в глибину?
5. У чому відмінність пошуку з поверненням від пошуку в глибину?

Література

1. В.М.Дубовой, Р.Н.Квєтний. Програмування комп'ютеризованих систем управління та автоматики. - Вінниця: ІЗМН, 1997. - 208 с.
2. Дубовой В.М., Квєтний Р.Н. Програмування персональних комп'ютерів систем управління. – В.: ВДТУ, 1999.
3. Простое и сложное в программировании. М.: Наука, 1988

Лабораторна робота № 8

Тема: Генетичні алгоритми оптимізації

Мета: Навчитись визначати глобальні екстремуми складних функцій за допомогою генетичних алгоритмів.

Теоретичні відомості

Вперше ідея використання генетичних алгоритмів для навчання виникла в 1970 році. В другій половині 80-х років до цієї ідеї повернулись в зв'язку із навчанням нейронних мереж. Генетичний алгоритм (ГА) нагадує біологічні процеси. Найбільш важливі з них випадкові мутації в хромосомах, кроссовер і рекомбінація генетичного матеріалу та міграція генів. ГА працює наступним чином. Ініціалізується популяція і всі хромосоми порівнюють у відповідності із заданою функцією оцінки. Далі виконується функція репродукції популяції хромосом. Батьки вибираються наступним чином у відповідності із значенням оцінки (ймовірність оцінки того, що дана хромосома стане батьком, пропорційна одержаній оцінці). Репродукція відбувається індивідуально для одного батька шляхом мутації хромосоми, або для двох батьків шляхом кроссовера генів. Діти, що одержалися оцінюються і розміщуються в популяції.

В результаті описаних операцій на кожному етапі еволюції одержуються популяції із більш досконалими індивідуами.

Генетичне наслідування моделюється наступним чином (табл. 6):

Таблиця 6

Основні терміни теорії ГА

ГА	Пояснення
Хромосома	Вектор(послідовність) із нулів та одиниць. Кожна позиція(біт) називається геном.
Індивідуум Генетичний код	Набір хромосом = варіант розв’язку задачі
Кроссовер	Операція, при якій хромосоми обмінюються частинами
Мутація	Випадкова зміна однієї чи кількох позицій в хромосомі

Рис.8. Алгоритм обчислень

Кроссовер називається одноточковим, якщо обмін частинами хромосом відбувається так

1 0 1 0 1 1 1 0 0 1		1 0 1 0 1 0 0 1 1 1
	⇒	
0 1 1 1 1 0 0 1 1 1		0 1 1 1 1 1 1 0 0 1

Двохточковий кроссовер

1 0 1 0 1 1 1 0 0 1		1 0 1 1 1 0 0 0 0 1
	⇒	
0 1 1 1 1 0 0 1 1 1		0 1 1 1 1 1 1 1 1 1

Мутація полягає в зміні одного або декількох генів(бітів) з вірогідністю рівною коефіцієнту мутації. Кожний біт має однаковий шанс бути мутованим.

Батьківські пари можна вибирати такими способами:

Панміксія – обидва члени популяції, що створюють батьківські пари випадковим чином вибираються із усієї популяції, при чому будь-який член популяції може стати учасником декількох пар.

Селективний – батьками можуть стати лише ті особи, значення пристосованості яких не менше середнього значення пристосованості по популяції.

Інбрідінг – перший член пари вибирається випадковим чином, а другим з більшою ймовірністю буде максимально близький до нього член популяції.

Аутбрідінг – пари формуються аналогічно, але із максимально далеких членів популяції.

Розрізняють фенотипний і генотипний інбрідінг та аутбрідінг.

Існує два механізми відбору членів нової популяції: елітний та відбір з витісненням. При *елітному відборі* нова популяція складатиметься лише із найкращих членів репродукційної групи, що об'єднує в себе батьків, потомків і мутантів. При *відборі з витісненням* те, чи буде член репродукційної групи заноситись в нову популяцію визначається не лише величиною її пристосованості, але й тим, чи є у новій популяції член із аналогічним набором хромосом.

Порядок виконання роботи

Знайти $extr F(\vec{x}), \vec{x} \in D$, де D – обмежена область, використовуючи ГА. Завдання обрати згідно до варіанту (табл. 7).

Таблиця 7

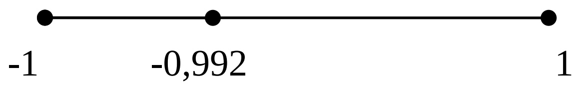
Початкові дані

№	Функція	Кросовер	Ймовірність мутації	Вибір батьків	Механізм вибору
1	$F(x_1, x_2) = \dfrac{100}{100(x_1^2 - x_2) + (1-x)^2 + 1}$, $-1{,}28 \le x_{1,2} \le 1{,}28.$	Одноточковий	0,01	Панміксія	Елітний
2	$F(x_1, x_2, ..., x_5) = \sum_{i=1}^{5} [x_i]$, $-5{,}12 \le x_{1,2,3,4,5} \le 5{,}12.$	Двохточковий	0,02	Селективний	Із витісненням
3	$F(x_1, x_2) = 0{,}002 \sum_{j=1}^{25} \dfrac{1}{j + \sum_{i=1}^{2} (x_i - a_i)^6}$, $a_{1j} = 16[(j \bmod 5) - 2], a_{2j} = 16[(j\%5) - 2].$	Одноточковий	0,015	Інбрідінг	Елітний
4	$F(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10\cos 2\pi x_1 - 10\cos 2\pi x_2$, $-5{,}12 \le x_{1,2} \le 5{,}12.$	Двохточковий	0,01	Аутбрідінг	Із витісненням
5	$F(x_1, ..., x_{10}) = \sum_{i=1}^{10} (10\cos(2\pi x_i) - x_i^2) - 100$, $-5{,}12 \le x_{1,2,...10} \le 5{,}12.$	Одноточковий	0,015	Панміксія	Елітний
6	$F(x_1, x_2, ..., x_5) = \sum_{i=1}^{5} [x_i]$, $-5{,}12 \le x_{1,2,3,4,5} \le 5{,}12.$	Одноточковий	0,01	Панміксія	Елітний
7	$F(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10\cos 2\pi x_1 - 10\cos 2\pi x_2$, $-5{,}12 \le x_{1,2} \le 5{,}12.$	Одноточковий	0,02	Панміксія	Елітний
8	$F(x_1, x_2, ..., x_5) = \sum_{i=1}^{5} [x_i]$, $-5{,}12 \le x_{1,2,3,4,5} \le 5{,}12.$	Двохточковий	0,015	Панміксія	Елітний
9	$F(x_1, ..., x_{10}) = \sum_{i=1}^{10} (10\cos(2\pi x_i) - x_i^2) - 100$, $-5{,}12 \le x_{1,2,...10} \le 5{,}12.$	Одноточковий	0,02	Панміксія	Із витісненням
10	$F(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10\cos 2\pi x_1 - 10\cos 2\pi x_2$, $-5{,}12 \le x_{1,2} \le 5{,}12.$	Одноточковий	0,01	Селективний	Елітний

Рекомендації

Наприклад, необхідно знайти $\max f(x_1, x_2)$, де $-1 \le x_1, x_2 \le 1$ і $f(x_1, x_2) = \sqrt{3}\cos x + \sin x$.

Для цього, розіб’ємо відрізок [-1;1] на 255 відрізків.



Будемо кодувати -1$\rightarrow$ 00000000, - 0,992 $\rightarrow$ 00000001,...., 1$\rightarrow$ 11111111.

Оскільки змінних дві, то хромосома складатиметься із 16 генів, перша половина яких відповідає x_1, друга –x_2. Всього таких хромосом - 65536.

Випадковим чином виберемо серед них 100 - початкову популяцію. Для фенотипів популяції: обчислимо значення $(F_1, \ldots, F_{100})$. Кожному значенню F_i співставимо ймовірність $p_i = \dfrac{F_i}{\sum_{i=1}^{100} F_i}$. Подальші кроки залежать від варіанту.

Приклад:

Склад звіту

1. Мета роботи
2. Постановка задачі
3. Короткі теоретичні відомості
4. Текст програми
5. Результати оптимізації
6. Висновки

Контрольні запитання

1. Для розв' язку яких задач використовують Г А?
2. Переваги та недоліки ГА.
3. Етапи функціонування ГА.
4. Генотипи та фенотипи.
5. Символьна модель ГА.
5. Геометрична інтерпретація символьної моделі.

Література

1. Ротштейн А.П. Интелектуальные технологиии идентификации. Винница: Вінниця–УНІВЕРСУМ, 1999. – 320 с.
2. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382 с.