

ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
„ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ”
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

Метаеврістики в розв’язанні економічних задач

Методичні рекомендації до лабораторних занять
для аспірантів зі спеціальності «Економіка»

Затверджено
Вченою радою ЗНУ
Протокол № від

ЗАПОРІЖЖЯ

20__

УДК: 330.46 (075.8)

ББК: Ув6я73

С 409

Метаеврістики в розв'язанні економічних задач: методичні рекомендації до лабораторних занять для здобувачів ступеня вищої освіти бакалавра напряму підготовки «Економічна кібернетика» / Укладачі: І.В. Козін, Н.К.Макшишко – Запоріжжя : ЗНУ, 20__ . – __ с.

У методичних рекомендаціях до лабораторних занять розглянуто основні поняття теорії метаеврістик, зокрема еволюційних алгоритмів, які можуть застосовуватися в процесі підтримки прийняття управлінських рішень. У межах лабораторних занять студенти виконують лабораторні роботи, які сприятимуть розвитку їх аналітичних та практичних здібностей.

Методичні рекомендації містять стислі теоретичні відомості, питання для самостійного контролю знань із кожної викладеної теми, термінологічний словник, тестові завдання та перелік рекомендованої літератури. Видання сприятиме формуванню необхідних теоретичних знань і практичних навичок із застосування систем підтримки прийняття рішень, оволодінню студентами методологічних та організаційно-технологічних засад побудови системи підтримки прийняття управлінських рішень в економіко-організаційних та виробничих системах за рахунок дотримання принципу подання матеріалу від простого до складного.

Методичні рекомендації призначено для аспірантів зі спеціальності «Економіка».

Рецензент _____

Відповідальний за випуск *Н.К. Максешко*, д.е.н., професор

ЗМІСТ

ВСТУП	4
Лабораторна робота № 1 Двійкова система числення. Кодування Грея...	6
Лабораторна робота № 2 Побудова еволюційної моделі.....	12
Лабораторна робота № 3 Фрагментарні моделі для пошуку субоптимальних розв'язків оптимізаційних задач	13
Лабораторна робота № 4 Метаеврістики для задачі комівояжера	21
Тести для контролю знань.....	26
Рекомендована література.....	28

ВСТУП

Дисципліна «*Метаеврістики в розв'язанні економічних задач*» відноситься до циклу дисциплін базової професійної та практичної підготовки здобувача ступеня PhD з економіки. *Предметом* курсу є методи пошуку оптимальних рішень за допомогою алгоритмів, що засновані на метаеврістичних.

«*Метаеврістики в розв'язанні економічних задач*» є однією з важливих професійно-орієнтованих дисциплін у процесі формування фахівця з економіки.

Багато прикладних задач сучасної економіки відносяться до класу задач, що важко розв'язуються. Навіть використання можливостей сучасної комп'ютерної техніки не дозволяє знаходити точні оптимальні розв'язки таких задач. На жаль навіть задачі пошуку наближених розв'язків з оцінкою точності для таких задач також є складними. До таких задач належать багато варіантів задач маршрутизації, логістики, розкрою та пакування тощо. Для пошуку оптимальних розв'язків таких задач виправдано використання алгоритмів, які не мають гарантованих оцінок точності, але достатньо прості і зрозумілі з точки зору «здорового глузду». Саме до таких алгоритмів відносяться сучасні метаеврістики.

Значний розвиток теорія метаеврістик отримала в 21 столітті. Поруч з відомими генетичним алгоритмом, алгоритмом мурашиної колонії з'явилася низка нових алгоритмів, які імітують явища як живої так і неживої природи. Сьогодні ці алгоритми займають значне місце в багатьох системах підтримки прийняття рішень і інших системах побудованих за принципами штучного інтелекту.

Мета курсу полягає у формуванні системи фундаментальних теоретичних знань і практичних навичок з основ теорії сучасних метаеврістик та можливостей їх використання в різноманітних економічних задачах .

Згідно з визначеною метою, курс має такі *завдання*:

- вивчення особливостей сучасних підходів до розв'язку оптимізаційних задач на базі метаеврістик;
- набуття практичних навичок побудови еволюційних моделей для використання відповідних алгоритмів.
- ознайомлення з основними принципами побудови метаеврістик;
- ознайомлення з методами побудови фрагментарних моделей для деяких оптимізаційних задач.

За підсумками вивчення курсу *студент повинен знати*:

- методи визначення множини Парето;
- основні принципи побудови інтегральних критеріїв;
- основні методи підтримки прийняття рішень;

Студент повинен вміти:

- використовувати генетичний алгоритм для пошуку субоптимальних розв'язків оптимізаційних задач;

- розробляти фрагментарні моделі для складних оптимізаційних задач;
- розробляти фрагментарні моделі на базі фрагментарної структури задачі;
- використовувати методи і прийоми математичного моделювання;


Лабораторні заняття з дисципліни *«Метаеврістики в розв'язанні економічних задач»* складаються з чотирьох лабораторних робіт, які логічно і змістовно відповідають навчальній та робочій програмам дисципліни. Виконання лабораторних робіт дозволить студенту оволодіти необхідними знаннями та уміннями, які передбачені даною дисципліною, а також здобути відповідні практичні навички.

Набуті студентами знання та практичні навички з дисципліни *«Метаеврістики в розв'язанні економічних задач»* будуть необхідні їм при виконанні аналітичних досліджень під час написання дисертаційної роботи.

Лабораторна робота № 1

Тема: ДВІЙКОВІ УЯВЛЕННЯ

Мета роботи: вивчення можливостей уявлення цілих та дробових чисел в двійковій формі.

 **Ключові слова:** двійкові уявлення, код.

Завдання:

1. Ознайомитись із програмою «Двійкові уявлення»
2. Виконати тренувальні розрахунки
3. Виконати 3 типи розрахунків по 8 завдань у кожному та зберегти їх у базі даних
 - А) двійкові уявлення цілих чисел
 - Б) код Грея
 - В) двійкові уявлення дробових чисел
4. Оформити звіт з лабораторної роботи, який включає висновки щодо можливостей використання засобів аналізу даних MS Excel у СПР, описання процесу виконання роботи, ілюстративний матеріал.

Теоретичні відомості

Алгоритм обчислення двійкового уявлення цілих чисел

Нехай задано ціле число x . Двійкове уявлення цілого числа є рядок $zb_n b_{n-1} \dots b_1 b_0$. Кожне число z, b_i $i = 1, 2, \dots, n$ дорівнює 0 або 1.

На початковому кроці $y_0 = |x|$ ділимо число y_0 на 2. Число b_0 - залишок від поділу y_0 на 2. Вважаємо y_1 - результат від поділу y_0 на 2.

На кроці з номером i ділимо число y_i на 2 і вважаємо число b_i - залишок від поділу y_i на 2. Вважаємо y_{i+1} - результат від поділу y_i на 2. Процес триває до тих пір, поки на черговому кроці число не звернеться в 0. Число z вибирається рівним 1, якщо число x було негативним і 0 в іншому випадку.

Зворотнє перетворення двійкового числа $zb_n b_{n-1} \dots b_1 b_0$ на десяткове число проводиться за формулою

$$x = b_n 2^n + b_{n-1} 2^{n-1} + \dots + b_1 2 + b_0$$

Якщо $z = 1$, то числу x приписуємо знак "-".

Код Грея. Алгоритм обчислення коду Грея

Коди Грея легко виходять з двійкових чисел шляхом побітової операції «Додаток по модулю 2» з тим самим числом, зрушеним праворуч на один біт. Отже, i -й біт коду Грея G_i виражається через біти двійкового коду B_i наступним чином:

$$G_i = B_i \oplus B_{i+1},$$

де \oplus – операція «Складання по модулю 2»; біти нумеруються справа ліворуч, починаючи з молодшого.

Приклад: перетворити двійкове число 11010 код Грея.

11010

01101

10111

Перетворення коду Грея на двійковий код

Зворотний алгоритм – перетворення коду Грея на двійковий код – можна виразити рекурентною формулою

$$B_i = B_{i+1} \oplus G_i,$$

причому перетворення здійснюється побітно, починаючи зі старших розрядів, і значення B_{i+1} , що використовується у формулі, обчислюється на попередньому етапі алгоритму. Якщо підставити в цю формулу вищенаведений вираз для i -го біта коду Грея, отримаємо

$$B_i = B_{i+1} \oplus G_i = B_{i+1} \oplus (B_i \oplus B_{i+1}) = B_i \oplus (B_{i+1} \oplus B_{i+1}) = B_i \oplus 0 = B_i.$$

Алгоритм обчислення двійкового уявлення десяткового дробу.

Нехай задано дробове число x . Для перетворення цього числа до двійкового дробу необхідно задати такі параметри:

Точність подання числа x у десятковій формі (число знаків після коми) та кількість знаків k у дробовій частині двійкового числа. В двійковому уявленні дробового числа є два рядки рядок $z b_n b_{n-1} \dots b_1 b_0$ і $f_1 f_2 \dots f_k$. Кожне з чисел z, b_i, f_j $i = 1, 2, \dots, n$, $j = 1, 2, \dots, k$ дорівнює 0 або 1. Послідовність $z b_n b_{n-1} \dots b_1 b_0$ є двійковим уявленням цілого числа, отриманого шляхом відкидання дробової частини. Нехай α - дробова частина числа $abs(x)$.

На черговому кроці з номером j вважаємо f_j рівним 1 або 0 залежно від умови $\alpha \geq 0.5$. Після цього вважаємо α рівним цілій частині числа 2α . Операція повторюється рівно k разів.

На початковому кроці вважаємо $y_0 = |x|$ і ділимо число y_0 на 2. Число b_0 - залишок від поділу y_0 на 2. Вважаємо y_1 - результат від поділу y_0 на 2.

На кроці з номером i ділимо число y_i на 2 і вважаємо число b_i - залишок від поділу y_i на 2 і вважаємо y_{i+1} - результат від поділу y_i на 2. Процес триває до тих пір, поки на черговому кроці число y_i не стане рівним до 0. Число z вибирається рівним 1, якщо число x було негативним і 0 інакше.

Зворотне перетворення за двома послідовностями $z b_n b_{n-1} \dots b_1 b_0$ і $f_1 f_2 \dots f_k$ відновлює десяткове дробове число x . Для цього спочатку по послідовності $z b_n b_{n-1} \dots b_1 b_0$ відновлюється ціле число y . По послідовності $f_1 f_2 \dots f_k$ отримуємо

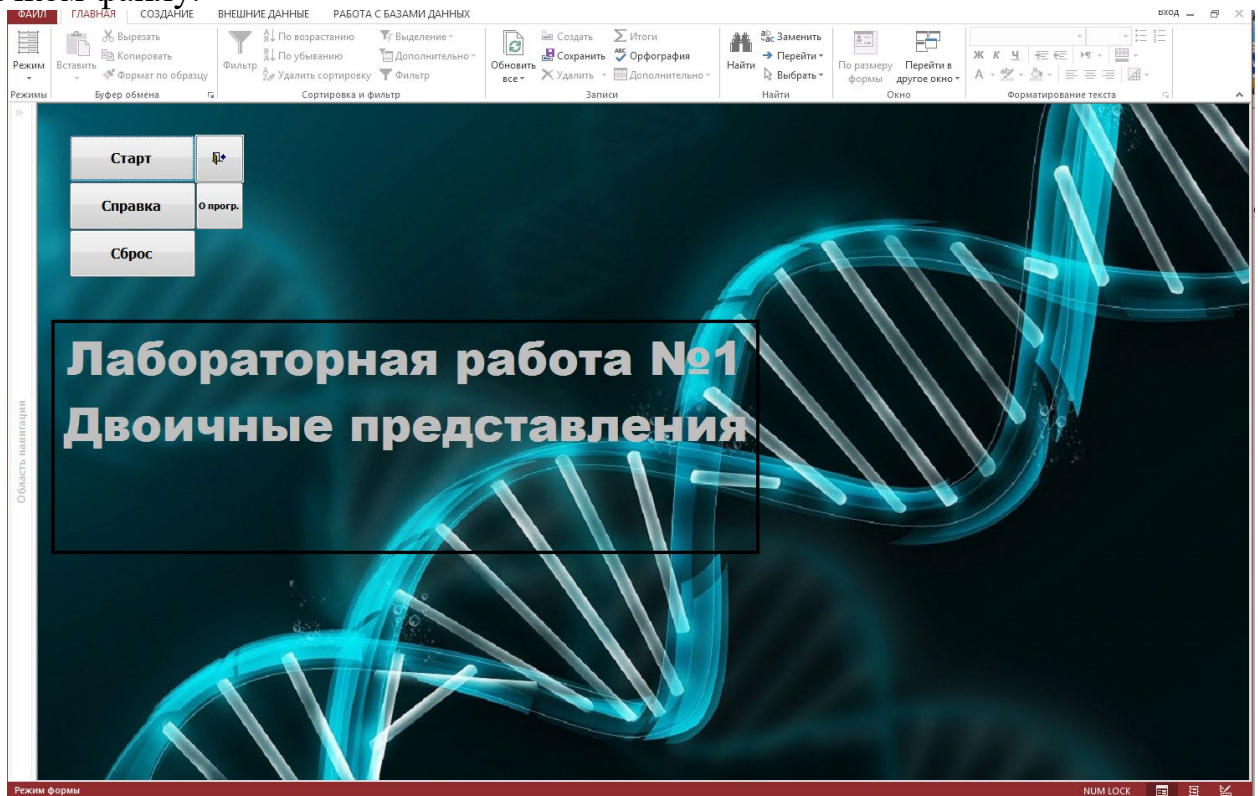
дробову частину числа за такою формулою: $r = \frac{f_1}{2} + \frac{f_2}{4} + \dots + \frac{f_k}{2^k}$. Обчислюване число x обчислюємо за формулою: $x = s * (abs(y) + r)$ та округляємо з необхідною точністю. Тут $s = -1$, якщо число $z = 1$, і $s = 1$ якщо число $z = 0$.

Слід зазначити, що для дробів перетворення на двійкове уявлення і назад не є взаємно зворотними. Тобто при послідовному застосуванні цих двох перетворень результат може відрізнятись від вихідного числа.

Хід лабораторної роботи з використанням програми «Двійкові уявлення чисел».

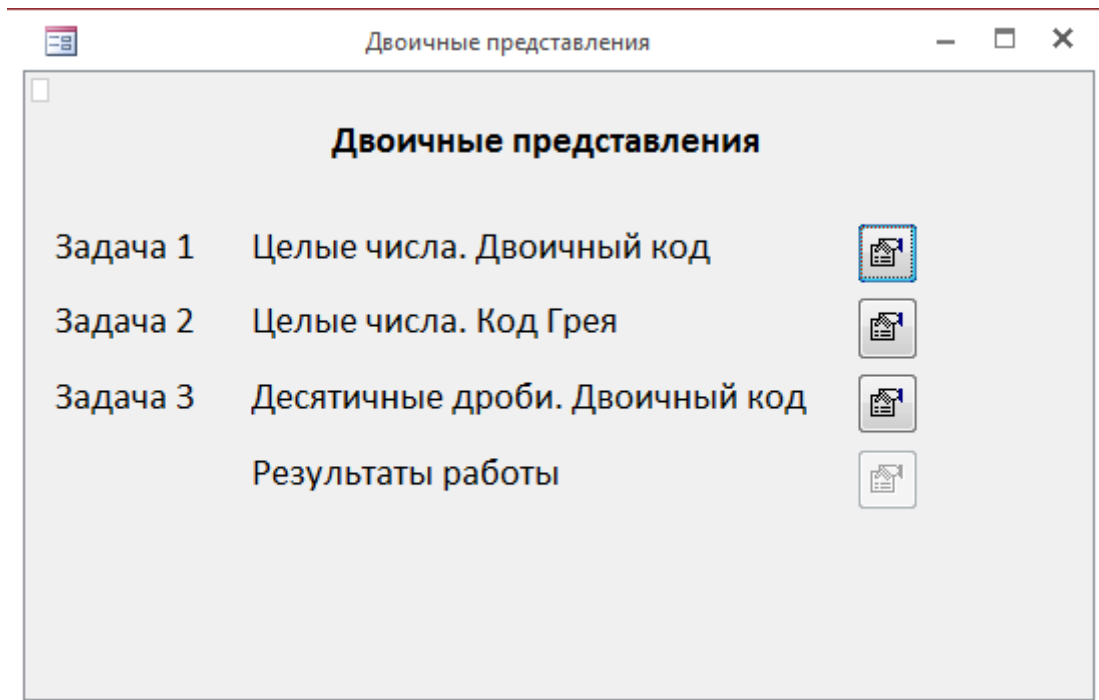
Програма передається студентам як файл Work_1_xx.mdb. До початку роботи потрібно змінити ім'я файлу, замінивши вираз xx порядковим номером студента у журналі групи.

Запуск програми здійснюється подвійним клацанням лівої кнопки миші за значком файлу.



Початкове вікно програми.

Після натискання кнопки «Старт» відбувається перехід у перше робоче вікно програми



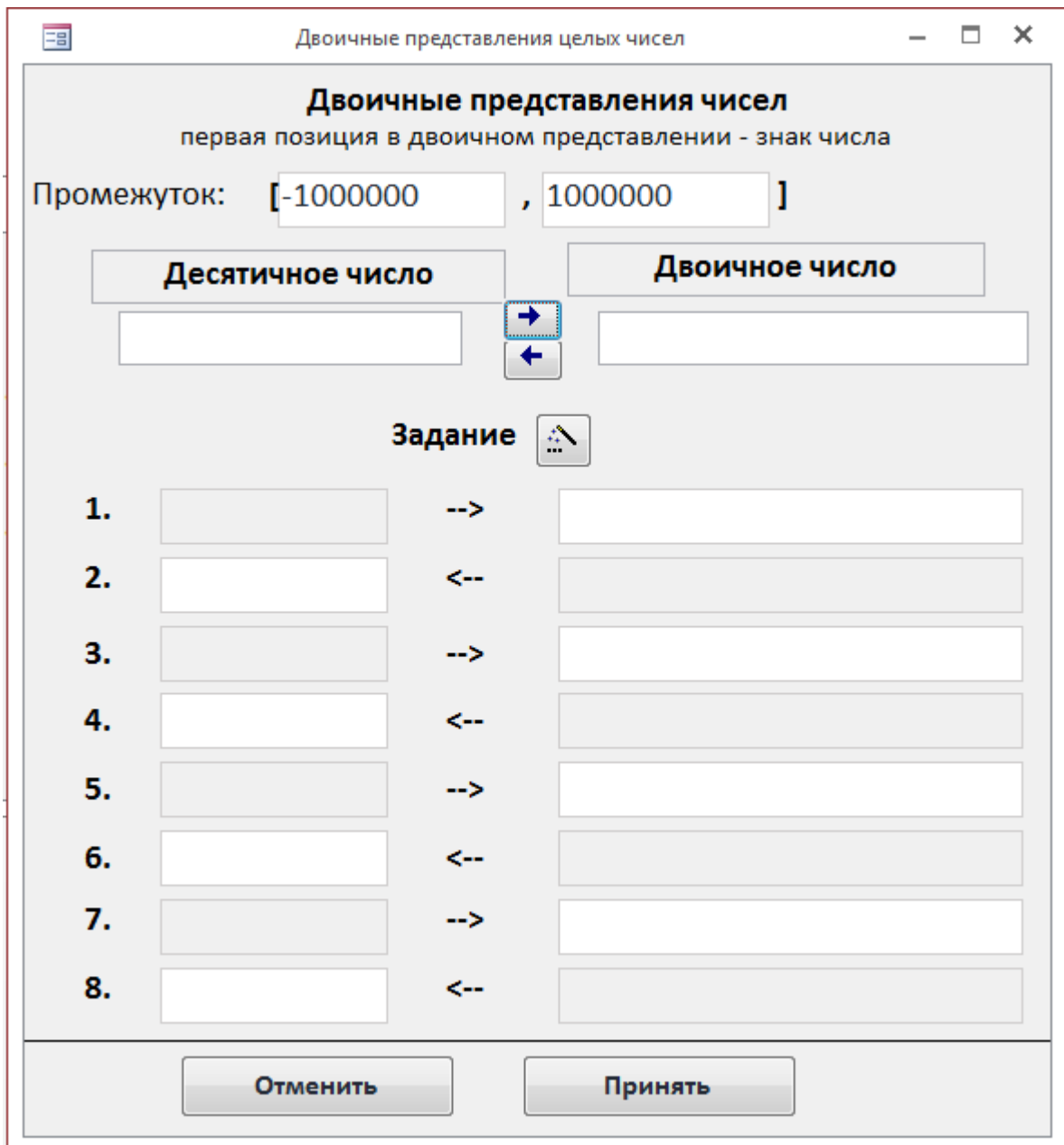
Перше робоче вікно програм

Спочатку доступні всі три завдання. Після того, як завдання виконано та результати збережені, відповідна кнопка завдання стає недоступною.

Розглянемо як приклад задачу номер 1 – Цілі числа.

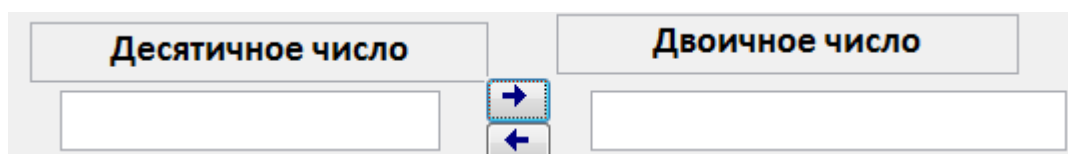
Мета завдання ознайомити студентів із перетвореннями двійкових чисел у стандартний двійковий код та зворотному перетворенню з двійкового коду на ціле число.



Вікно завдання показано на малюнку



Вікно задачі №1.

Генерація восьми завдань відбувається при натисканні кнопки. Завдання полягають у обчисленні двійкового коду заданого числа та обчисленні цілого числа за двійковим кодом (4 завдання одного типу, 4-другого). Перед генерацією завдань можна провести пробні обчислення у вікні обчислень



Однак після створення серії завдань кнопки обчислень   будуть заблоковані. Для розблокування кнопок потрібно натиснути кнопку

"скасувати", а потім знову увійти до тесту. Однак у цьому випадку завдання буде змінено.

Завдання виконуються вручну за допомогою алгоритмів перетворень. Результати оформляються у вигляді звіту та заносяться у форму у відповідні поля.

Якщо результати виконання завдань влаштовують студента, потрібно натиснути кнопку «прийняти». Після цього тестове завдання вважається виконаним і вхід в цей тест буде заблокований.


Аналогічні дії проводяться за двома завданнями - «Код Грея» і «Двійкові уявлення десяткових дробів»

Після того, як усі 3 тести будуть виконані файл Work_1_xx.mdb, передається викладачеві для перевірки результатів розрахунків.

Оцінка за роботу виставляється відповідно до критеріїв.

Тема: ПОБУДОВА ЕВОЛЮЦІЙНОЇ МОДЕЛІ.

Мета роботи: познайомитися з поняттям множини Парето та поняттям ефективного рішення багатокритеріальної задачі. Використовуючи алгоритм послідовного пошуку визначити множену Парето для багатокритеріальної задачі.

 **Ключові слова:** множина Парето, алгоритм послідовного пошуку, ефективне рішення багатокритеріальної задачі.

Завдання:

1. Познайомитися з алгоритмом послідовного пошуку для визначення множини Парето при наявності багатьох критеріїв.
2. Шляхом використання алгоритму послідовного пошуку виділити множину Парето для багатокритеріальної задачі.
3. Оформити звіт з лабораторної роботи, який включає висновки щодо використання алгоритму послідовного пошуку.



Теоретичні відомості


?

Питання для самостійного контролю знань

1. Дайте опис класичного генетичного алгоритму
2. Як виглядає узагальнена еволюційна модель?
3. Що таке турнірний відбір?
4. Наведіть приклади геометричного оператора кросоверу?

Тема: ФРАГМЕНТАРНІ МОДЕЛІ ДЛЯ ПОШУКУ СУБОПТИМАЛЬНИХ РОЗВ'ЯЗКІВ ОПТИМІЗАЦІЙНИХ ЗАДАЧ

Мета роботи: ознайомитися з принципами побудови фрагментарних моделей оптимізаційних задач та з особливостями використання комп'ютерної системи «Фрагментарні структури та метаеврістики».

 **Ключові слова:** фрагментарна модель, фрагментарний алгоритм, метаеврістика, накриваюче відображення, прямокутний розкрій.

 **Завдання:**

1. Ознайомитися з теоретичними відомостями щодо використання комп'ютерної системи «Фрагментарні структури та метаеврістики».
2. За допомогою інструментарію системи «Фрагментарні структури та метаеврістики» згенерувати низку тестових завдань для задачі прямокутного розкрою.
3. Побудувати фрагментарну модель загальної задачі прямокутного розкрою. Знайти декілька допустимих розв'язків цієї задачі.
4. Оформити звіт з лабораторної роботи, який включає висновки щодо можливостей використання комп'ютерної системи «Фрагментарні структури та метаеврістики».



Теоретичні відомості

Комп'ютерна система «Фрагментарні структури та метаеврістики» призначена для тестування та оцінки якості різних метаеврістик Система включає такі файли: файл основної програми-СУБД: EVFTester.mdb, файл документації: EVFTester.DOC, тестові бази даних: файли з розширенням .mdb, які містять серії завдань. Кількість таких файлів необмежена.

Вимоги до обладнання:

- процесор із частотою не нижче 2 мГц.
- об'єм оперативної пам'яті не менше 512 мб
- об'єм дискової пам'яті не менше 50 Мб
- операційна система: WINDOWS 2007/2010
- СУБД: ACCESS XP 2007/2010/2016



Рис 3.1 – Комп'ютерна система «Фрагментарні структури та метаеврістики»

Запуск системи здійснюється подвійним натисканням лівої кнопки миші на файлі EVFTester.mdb. У стартовому вікні (рис 3.1) системи вибирається тип завдань, які розглядатимуться. На сьогодні у переліку типів присутні такі:

- завдання ЦЛП
- завдання розміщення блоків
- завдання на графах
- розклади

Однак, цей список може поповнюватися.

Для початку роботи системи потрібно натиснути кнопку «Старт».

Для закінчення роботи використовується кнопка виходу або закриття вікна.

Головне вікно програми

Після натискання кнопки старту відкривається головне вікно програми (рис 3.2). У цьому вікні представлено таблицю з описами завдань обраного класу.

Кожен тип завдань складається з кількох класів завдань. Клас задач визначається набором параметрів, що визначає індивідуальне завдання класу. Клас буде розбитий на підкласи за деякими ознаками. Проте належність

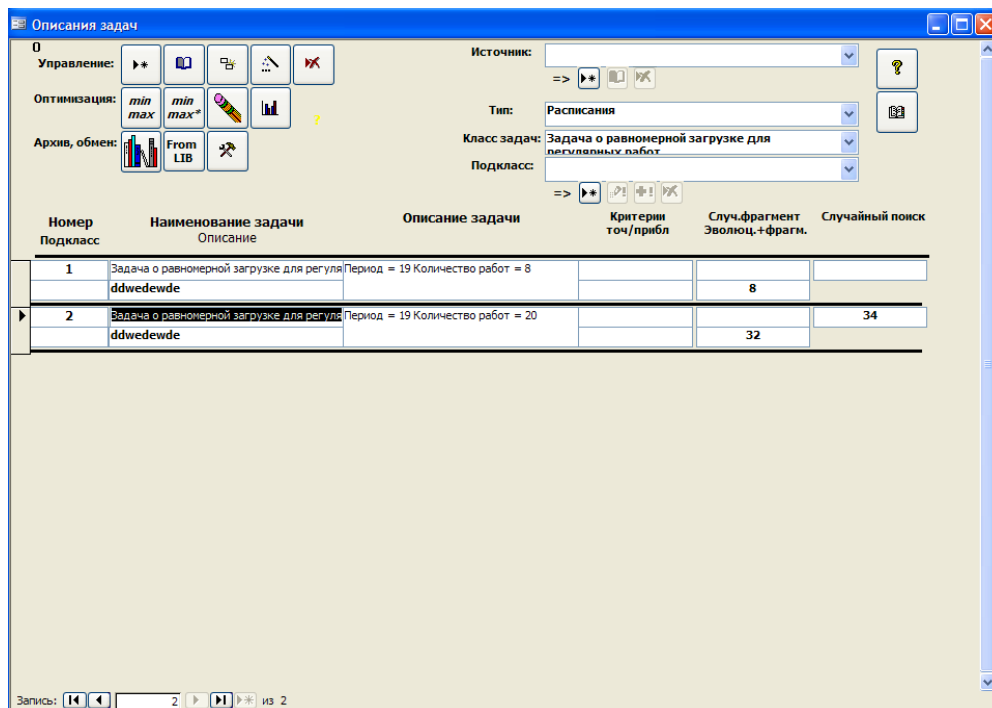



Рис 3.2 – Головне вікно системи «Фрагментарні структури та метаевристики»

завдання тому чи іншому підкласу можна змінити. Кожне завдання обов'язково входить у певний клас, але може входити у підкласи цього.

У назві форми розташовані кнопки управління, списки вибору та титульний рядок таблиці описів завдань.

Поле «Джерело» вказує базу даних, в якій знаходяться розглянуті описи завдань. Якщо ця підлога порожня, то як база використовується сама база EVFTester.mdb.

Кнопка  в заголовку форми дозволяє отримати довідку поточного вікна завдання.

Робота із джерелами. Кнопки керування джерелами. Вибір джерела зі списку. Джерело вибирається зі списку джерел за назвою зі списку джерел (рис 3.3).

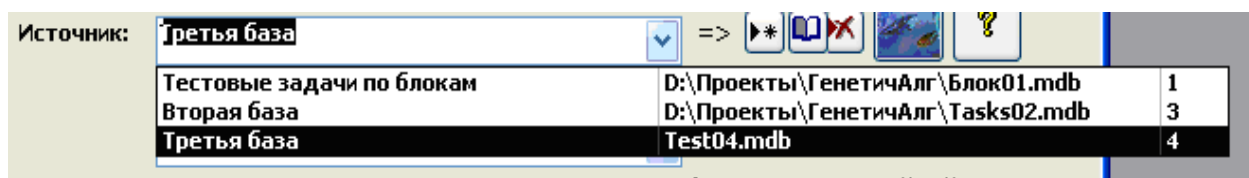



Рис. 3.3 – Вибір джерел даних

У списку вказано найменування джерела, повне ім'я файлу бази даних та номер джерела у списку. Якщо список порожній або потрібне джерело у списку відсутнє, необхідно створити новий елемент списку, натиснувши кнопку «додати» .

Подвійне клацання лівої кнопки миші на полі «Джерело» очищає це поле та автоматично підключає дані поточної бази даних програми, тобто файл EVFTester.mdb.

Вибір класу завдань провадиться зі списку класів даного типу завдань у зазначеному джерелі. Перелік описів завдань класу відображається у табличній частині форми. Подвійне клацання лівої кнопки миші очищає поле «клас». У цьому таблична частина форми зникає.

У міру розвитку системи до списку класів будуть додаватися нові класи. З кожним класом задач пов'язаний алгоритм опису задачі класу, алгоритми пошуку рішення та алгоритми візуалізації задач класу. Опис цих алгоритмів кожного конкретного класу завдань наводяться в окремому розділі документації, присвяченому цьому класу.

У табличній частині форми (рис. 3.4) виводяться рядки таблиці, що містить перелік згенерованих у системі завдань.

Номер Подкласс	Наименование задачи Описание	Описание задачи	Критерии точ./прибл	Случ.фрагмент Эволюц.+фрагм.	Случайный поиск
1	Задача директора (задача одного станка) Версия №1.	Количество работ = 11	1556	2198 1615	1636
2	Задача директора (задача одного станка) Версия №2.	Количество работ = 91	153315	214719 188431	193372
3	Задача директора (задача одного станка) Версия №3.	Количество работ = 94	134087	205646 180697	175898
4	Задача директора (задача одного станка) Версия №4.	Количество работ = 54	57499	79919 67814	68619
5	Задача директора (задача одного станка) Версия №5.	Количество работ = 100	183017	267660 232042	239875
6	Задача директора (задача одного станка)	Количество работ = 42	31057	41610	38250

Рис. 3.4 – Перелік згенерованих у системі завдань

У першій колонці вказуються порядковий номер завдання та номер підкласу. Друга колонка містить найменування задачі та її короткий опис. Повний опис задачі наводиться у третій колонці. Наступні три колонки містять останні результати роботи алгоритмів різних типів для обраного завдання. Результат роботи алгоритму це обчислене значення цільової функції. У програмі використовуються такі типи алгоритмів:

- точний алгоритм;
- відомий наближений алгоритм;
- фрагментарний алгоритм за деякого упорядкування фрагментів;
- ЕВФ-алгоритм;
- алгоритм випадкового пошуку на множині допустимих рішень.

Особливості реалізації алгоритмів для різних класів завдань описані в окремих файлах інструкції.

Щоб детальніше побачити параметри останніх розрахунків, можна скористатися кнопкою «min-max».

Перелік завдань у таблиці можна фільтрувати та сортувати за звичайними правилами роботи з таблицями СУБД ACCESS.

Для пошуку оптимальних рішень будь-який із завдань переліку потрібно перейти у вікно пошуку рішень (рис. 3.5).

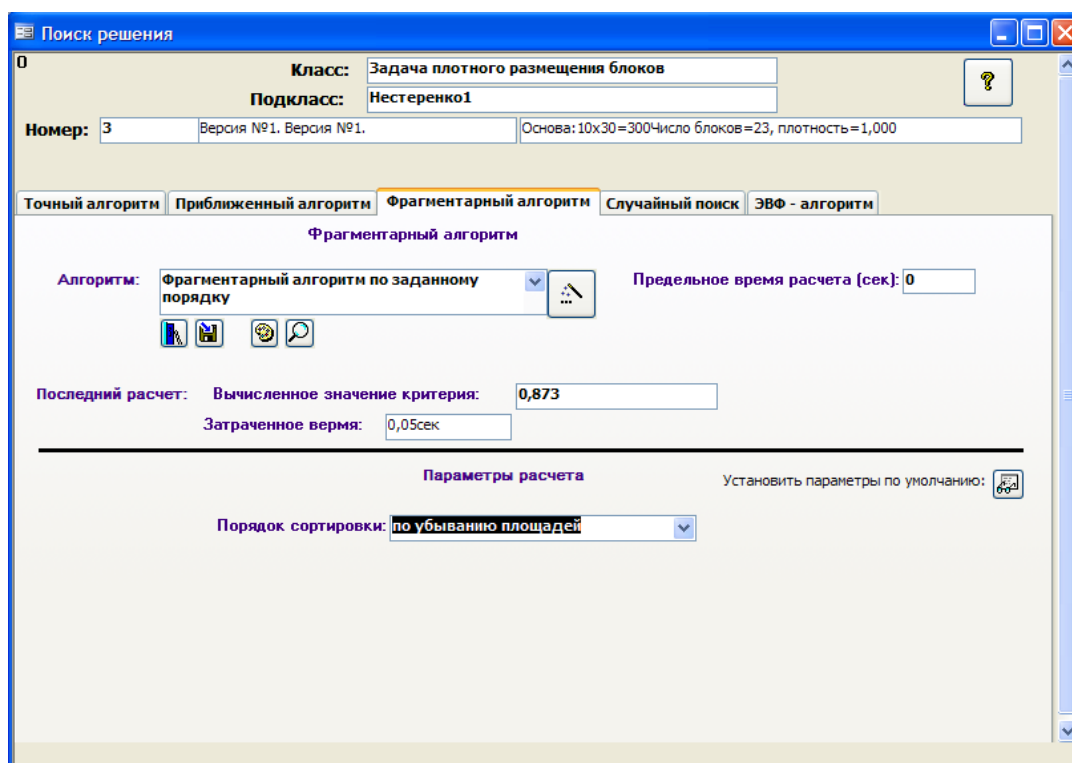


Рис. 3.5 – Вікно пошуку рішень

Область даних форми "Пошук рішення" розбита на п'ять зон, кожна з яких присвячена одному з типів алгоритмів для обраного класу завдань. Кожен тип алгоритмів має власний набір параметрів розрахунку.

Область даних форми "Пошук рішення" розбита на п'ять зон, кожна з яких присвячена одному з типів алгоритмів для обраного класу завдань. Кожен тип алгоритмів має власний набір параметрів розрахунку. Параметри розрахунку за замовчуванням всім типам надаються натисканням кнопки .

У будь-якій із зазначених зон вибирається зі списку конкретний алгоритм розв'язання задачі, встановлюються параметри розрахунку, запускається процедура розрахунку шляхом натискання кнопки розрахунку для відповідного алгоритму. У полі "обчислене значення критерію" заноситься результат розрахунку оптимального значення критерію.

Граничний час розрахунку встановлюється за секунди. Якщо граничний час розрахунку дорівнює нулю, то розрахунок проводиться без обмеження часу. В іншому випадку алгоритм зупиняється після закінчення часу розрахунку.

Якщо час розрахунку вичерпано, то для процедури точного алгоритму виконується спроба протягом такого самого проміжку часу знайти оптимальне наближене значення критерію. Якщо наближене значення знайдено, воно виводиться у полі значення критерію зі знаком «*».

Для інших типів алгоритмів після закінчення часу розрахунку у полі значення критерію вноситься значення, отримане на останньому етапі відповідного алгоритму.

Значення критерію може мати вигляд $\pm Ma + b$, де M - велике позитивне число. Наявності величини M у значенні критерію показує, що опис рішення входять неіснуючі фрагменти.

Якщо процедура пошуку оптимального рішення закінчилася невдачею, то поле значення критерію заноситься слово «ні».

Для алгоритму випадкового пошуку необхідно вказати кількість розіграшів, тобто рішень, що визначаються випадковим чином.

Для ЕВФ-алгоритму необхідно встановити такі параметри розрахунку:

- а) розмір (чисельність) популяції;
- б) кількість пар, що схрещуються, в одному поколінні;
- в) кількість поколінь;
- г) ймовірність мутації;
- д) коефіцієнт відбору.

Аналіз результатів. Для того щоб отримати порівняльні оцінки якості алгоритмів на серії завдань необхідно виконати такі дії:

– виконати розрахунки для досліджуваної серії завдань за всіма порівнюваними алгоритмами;

– досягти того, щоб у табличній частині форми «Опису завдань» були присутні всі завдання серії і тільки вони. Цього можна досягти, виділивши відповідний підклас та встановивши необхідні фільтри;

– натиснути кнопку аналізу результатів .

В результаті відкриється вікно діаграм, що відображають порівняльну якість роботи різних алгоритмів на заданій серії задач (рис. 3.6).



Рис. 3.6 – Вікно діаграм порівняння якості роботи різних алгоритмів

У заголовку вікна виділяються типи алгоритмів порівняння (типи алгоритмів, які описані цієї серії завдань, недоступні для позначки).

Розподіл перших місць за типами алгоритмів: діаграма показує кількість завдань, для яких тип алгоритмів, що розглядається, приводив до найкращих результатів серед усіх застосовуваних алгоритмів в аналізованій серії завдань. Ця діаграма застосовна лише за порівнянні наближених алгоритмів.

Порівняльний розподіл за типами алгоритмів: число показує скільки разів у серії завдань результат, отриманий за цим типом алгоритмів, був не гіршим за результати, отримані за іншими типами.

Рейтинг алгоритмів: обчислюється за правилом Борда як сума балів, набраних алгоритмом з усіх завдань вибірки. За перше місце алгоритм отримує 4 бали, за друге – 3 бали, за 3-тє два бали, за 4-тє 1 бал та за п'яте місце порівняно – 0 балів.

Порівняльний аналіз різних метаевристичних алгоритмів показав, що ефективність алгоритмів приблизно однакова з невеликою перевагою еволюційно-фрагментарного алгоритму.




Питання для самостійного контролю знань

1. Дайте визначення поняттю проект.
2. Для чого призначена система *Project Expert*?
3. Які етапи включає робота з системою *Project Expert*?

4. У чому полягає сутність аналізу проекту в системі *Project Expert*?
5. Який програмний модуль у системі *Project Expert* використовується для порівняльного аналізу варіантів розробленого проекту?

Тема: МЕТАЕВРІСТИКИ ДЛЯ ЗАДАЧІ КОМІВОЯЖЕРА

Мета роботи: ознайомитися з принципами побудови та особливостями використання метаевристик на базі фрагментарних моделей оптимізаційних задач на прикладі задачі комівояжера.

 **Ключові слова:** фрагментарна модель, еволюційно-фрагментарний алгоритм, задача комівояжера.

Завдання:

Задана матриця вартостей перевозок между пунктами споживання.

Виробництво знаходиться в пункті № 1.

Потрібно за допомогою однієї машини розвести продукцію від пункту виробництва до всіх пунктів споживання, побувавши в кожному пункті рівно один раз.

(Задача комівояжера)

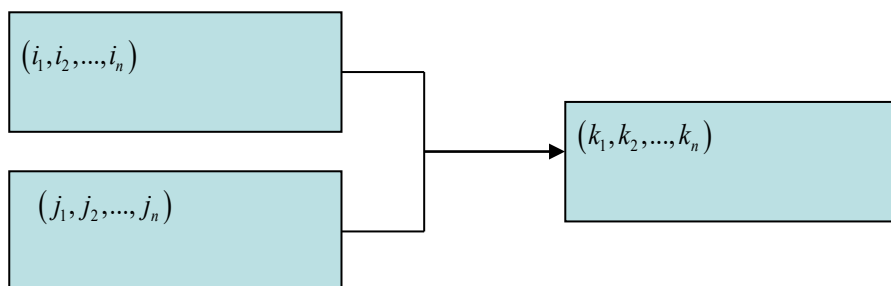
1. Показати, що задача має фрагментарну структуру.
2. За допомогою жадібного алгоритму знайти один з розв'язків задачі та розрахувати його вартість.
3. За допомогою фрагментарного алгоритму знайти будь який інший розв'язок, який відрізняється від того, що знайдено в пункті 2. Розрахувати його вартість
4. Застосувавши операцію кросоверу та мутації побудувати третій розв'язок.
Обчислити його вартість



Теоретичні відомості

Використовується геометричний оператор кросоверу на перестановках і мутація, описи яких наведено на рисунку.

Геометричний оператор кросоверу:



$$k_s \in \min \{i_s, j_s\}, \quad s = 1, 2, \dots, n$$

$$(2, 3, 4, 7, 8, 1, 6, 5) + (3, 4, 6, 2, 1, 5, 8, 7) \rightarrow (2, 3, 4, 6, 1, 5, 7, 8)$$

Оператор мутації: $M : X \rightarrow X$

$$(\dots i, \dots j, \dots) \Rightarrow (\dots j, \dots i, \dots)$$

Варіанти завдання

Варіант № 1

0	104	138	182	164	176	24	192
93	0	34	152	126	176	65	192
116	124	0	20	7	55	46	53
89	38	23	0	63	70	197	163
117	90	88	91	0	153	144	100
14	163	40	195	105	0	30	189
111	199	15	42	20	179	0	194
25	179	41	35	59	72	185	0

Варіант № 2

0	20	8	200	22	71	15	84
55	0	92	81	59	15	135	91
88	149	0	75	15	49	146	32
68	155	5	0	85	113	38	42
49	37	50	128	0	158	144	23
51	39	63	169	86	0	102	32
198	170	70	152	28	68	0	47
70	179	191	105	132	62	95	0

Варіант № 3

0	164	180	150	19	57	129	41
84	0	37	94	157	146	163	102
135	173	0	109	137	52	42	77
94	41	35	0	125	57	102	55
50	149	84	47	0	177	29	115
105	99	125	68	118	0	200	36
105	18	166	74	173	121	0	51
83	164	85	15	76	167	120	0

Варіант № 4

0	191	43	170	31	70	40	88
156	0	149	114	78	193	101	95
130	45	0	165	92	66	48	195
188	148	105	0	74	132	153	146
14	166	194	121	0	182	18	131
18	110	71	42	180	0	172	45
51	26	77	177	138	46	0	5
93	103	185	45	73	19	48	0

Варіант № 5

0	138	93	134	27	71	124	71
22	0	54	137	76	86	162	84
13	48	0	166	44	18	99	35
157	106	139	0	197	0	20	92
39	191	185	98	0	83	154	139
35	71	166	166	47	0	109	198
28	38	141	19	14	85	0	125
201	176	176	195	6	9	60	0

Варіант № 6

0	69	143	178	92	48	196	191
194	0	76	76	200	117	151	132
116	178	0	158	69	149	31	140
158	72	118	0	141	121	70	172
168	100	104	153	0	180	60	38
4	121	177	148	188	0	194	121
161	30	41	138	69	149	0	14
165	40	51	32	64	114	7	0

Варіант № 7

0	200	52	1	10	166	199	193
3	0	131	52	201	177	36	61
87	186	0	105	64	186	30	165
186	159	119	0	120	10	118	116
46	176	160	184	0	16	164	72
91	76	166	100	122	0	152	24
82	26	131	193	150	75	0	97
77	93	15	180	198	178	180	0

Варіант № 8

0	89	30	4	107	170	182	9
44	0	185	114	19	23	72	198
51	79	0	144	68	54	192	22
131	68	108	0	79	141	75	138
23	49	87	179	0	48	108	16
176	171	156	27	4	0	38	169
111	185	153	87	41	190	0	125
147	101	53	51	182	141	193	0

Варіант № 9

0	119	60	156	137	119	24	14
21	0	79	95	170	127	40	71
4	73	0	130	102	36	66	198
91	113	121	0	70	18	87	22
162	71	81	97	0	176	145	132
43	121	198	61	86	0	170	169
95	162	145	144	57	178	0	149
23	5	198	162	23	136	197	0

Варіант № 10

0	160	183	119	194	111	100	174
128	0	184	4	117	156	190	110
46	176	0	112	173	35	125	180
162	101	40	0	32	145	47	176
7	100	29	30	0	35	140	180
92	173	6	110	68	0	138	76
25	146	128	189	72	15	0	76
12	103	19	85	157	167	131	0
13	187	12	190	136	18	7	109



Питання для самостійного контролю знань

1. Описати загальну схему еволюційного алгоритму.?
2. В якому випадку оператор кросоверу називають геометричним?
3. Які особливості має метаеврістика на базі фрагментарного алгоритму.?
4. Яка трудомісткість фрагментарного алгоритму для задачі комівояжера.

ТЕСТИ ДЛЯ КОНТРОЛЮ ЗНАНЬ

- S: Що дозволяє оператор кросоверу (кросоверінгу) в моделі Холанда?
- : створити два розв'язки-нащадки;
- : створити новий допустимий розв'язок задачі;
- : створити один новий розв'язок;
- : знайти оптимальний розв'язок задачі
-
- S: Яка з перерахованих моделей є еволюційною?
- : модель Холланда
- : модель Марковіца
- : імітаційна модель
- : модель лінійного програмування
-
- S: Яка з перерахованих оптимізаційних задач не має фрагментарної структури?
- : задача управління запасами
- : задача про мінімальне остовне дерево
- : задача комівояжера
- : задача лінійного розкрою
-
- S: Яка масова задача називається складною?
- : для якої не існує алгоритм поліноміальної трудомісткості
- : для якої невідомий універсальний алгоритм
- : для якої існує алгоритм поліноміальної трудомісткості
- : яка не може бути вирішена за час порівнянне з тривалістю життя людини
-
- S: Яка метрика використовується для обчислення відстані між двома бінарними послідовностями?
- : метрика Хемінга
- : метрика Кендала
- : метрика Евкліда
- : манхеттенська метрика
-
- S: Яка метрика використовується для обчислення відстані між двома перестановками?
- : метрика Кендала
- : метрика Евкліда
- : метрика Хемінга
- : манхеттенська метрика
-
- S: Яка множина є базовою в моделі Холанда?

- : множина бінарних векторів B^n
- : множина векторів багатовимірного простору R^n ;
- : множина перестановок;
- : множина натуральних чисел;
-
- S: Який алгоритм називається складним?
- : який вирішує масову задачу за неpolіноміальний час від довжини входу
- : який вирішує масову задачу за polіноміальний час від довжини входу
- : який вирішує масову задачу методом перебору
- : який вирішує масову задачу кінцеве число кроків
-
- S: Який критерій зупинки не використовується в еволюційному алгоритмі?
- : досягнення точного розв'язку
- : обмеження часу розрахунку
- : обмеження числа поколінь
- : однаковість критерію для всіх розв'язків поточної популяції
-
- S: Якщо оптимізаційна задача має фрагментарною структурою, то пошук розв'язку можна звести до:
- : оптимізації на множині перестановок
- : оптимізації на графах
- : оптимізації в багатовимірному евклідовому просторі
- : оптимізації на множині бінарних послідовностей
-
-

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. І.В.Козін, Еволюційні моделі в дискретній оптимізації, Запоріжжя, Запорізький національний університет, 2019. – 204с.
2. Sean Luke, 2009, Essentials of Metaheuristics, Lulu, available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>.
3. Kozin, I.V., Batovskyi, S.E. Fragmentary Structures in a Two-Dimensional Strip Packing Problem. *Cybern Syst Anal* **55**, 943–948 (2019) doi:10.1007/s10559-019-00204-w
4. I. V. Kozin, N. K. Maksyshko, V. A. Perepelitsa Fragmentary Structures in Discrete Optimization Problems, *Cybernetics and Systems Analysis* November 2017, Volume 53, Issue 6, P 931–936. <https://doi.org/10.1007/s10559-017-9995-6>
5. Рутковська Д. Нейронні мережі, генетичні алгоритми і нечіткі системи. – К.: Наукова думка, 2019. – 452с.
6. Скобцов Ю.А. Основи еволюційних обчислень / Ю.А. Скобцов. – Донецьк.: ДонГУ, 2013. – 326с.
7. Суботін С.О. Ітеративні, еволюційні та мультиагентні методи синтезу нечітко логічних і нейромережових моделей / С.О.Суботін, А.О.Олійник, О.О.Олійник. – Запоріжжя: ЗНТУ, 2019. – 375с.
8. О. А. Щербина Метаевристичні алгоритми для задач комбінаторної оптимізації // Таврический вісник інформатики і математики 1(24) 2014 р. стр 3-5 tvim.info/files/56_72_Shcherbina.pdf

Додаткова:

1. Kozin I.V., Selyutin E.K., Polyuga S.I. Jumping frog method for optimal classifications / *International Academy Journal Web of Scholar*. 2(52). doi: 10.31435/rsglobal_wos/30042021/7519
2. Ігор Козін, Наталія Максишко, Ярослав Терешко Метод імітації відпалу для задачі рівноважного розміщення, *Фізико-математичне моделювання та інформаційні технології*, Львів 2021, вип.32, с.152-158. doi.org10.15407/fmmit2021.32.152
3. Holland J. H. *Adaptation in Natural and Artificial Systems* / J. H. Holland. – Boston, MA: MIT Press, 1992. – 288 p.
4. A.E. Eiben, E.H.L. Aarts, and K.M. van Hee. Global convergence of genetic algorithms: A Markov chain analysis. In H.-P. Schwefel and R. M_anner, editors, *Parallel problem Solving from Nature*, pp. 4-12. Springer, Berlinand Heidelberg, 1991.
5. Eremeev A.V., Reeves C.R. Evolutionary algorithms in discreteoptimisation. In: *Book of abstracts of Discrete Optimization andOperations Research Conference*

(DAOR-2002.P.40-45

http://iitam.omsk.net.uk/eremeev/PAPERS.MAT/ea_surv6.ps

6. Goldberg D.E. Genetic algorithms in search, optimization and machine learning. - Reading: Addison Wesley, 1989.
7. E.Goncharov and Yu.Kochetov, Behavior of a Probabilistic Tabu Search Algorithm for the Multi Stage Uncapacitated Facility Location Problem // Proceedings INFORMS-KORMS, Seoul 2000.
<http://math.nsc.uk/LBRT/k5/Kochetov/seoul.doc>

Інформаційні ресурси:

1. Аналітичні технології [Електронний ресурс]. – Режим доступу <http://www.neuroproject.ua/what.php>.
2. Kasahara Lab Waseda University Optimal Schedules for Prototype Standard Task Graph Set [Електронний ресурс]. – Режим доступу www.kasahara.elec.waseda.ac.jp/schedule/index.html

Навчально-методичне видання
(українською мовою)

Козін Ігор Вікторович
Максишко Наталія Костянтинівна

Метаеврістики в розв'язанні економічних задач

Методичні рекомендації до лабораторних занять
для здобувачів наукового ступеня PhD зі спеціальності «Економіка»

Рецензент
Відповідальний за випуск *Н.К. Максишко*
Коректор *В.В. Рянічева*