ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

Козін І.В. , Максишко Н.К.

СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА БАЗІ ШТУЧНОГО ІНТЕЛЕКТУ В ЕКОНОМІЦ

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

Затверджено
Вченою радою ЗНУ
Протокол № від

ЗАПОРІЖЖЯ
2024

УДК: 330.46 (075.8)
К591

Системи підтримки прийняття рішень на базі штучного інтелекту в економіці: методичні рекомендації до лабораторних занять для здобувачів другого ступеня вищої освіти освітньої програми «Економічна кібернетика» / Укладачі: І.В. Козін, Н.К. Максишко – Запоріжжя : ЗНУ, 2024. – 38с.

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 магістрів зі спеціальності «Економіка».

Рецензент
М.М. Іванов, доктор економічних наук, професор, завідувач кафедри управління персоналом і маркетингу ЗНУ

Відповідальний за випуск
Н.К. Максишко, доктор економічних наук, професор, завідувач кафедри економічної кібернетики ЗНУ

ЗМІСТ

ВСТУП

Дисципліна «*Системи підтримки прийняття рішень на базі штучного інтелекту в економіці*» відноситься до циклу дисциплін базової професійної та практичної підготовки здобувача ступеня магістра з економіки. *Предметом* курсу є методи пошуку оптимальних рішень за допомогою елементів штучного інтелекту, а саме алгоритмів, що засновані на метаеврістиках.

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 зрозуміли 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 «*Системи підтримки прийняття рішень на базі штучного інтелекту в економіці*»: набуття фундаментальних теоретичних знань і практичних навичок з питань постановки та розв'язування задач, щодо створення та застосування математичних моделей та інформаційних систем на рівні управління виробництвом. Підготувати кваліфікованих спеціалістів у сфері управління фінансами та керування сучасними економічними системами

Предметом вивчення є математичні моделі і методи розв'язання складних економічних задач на базі метаеврістик.

Основними завданнями вивчення дисципліни «*Системи підтримки прийняття рішень на базі штучного інтелекту в економіці*» є ознайомлення з основними поняттями, сучасними концепціями та математичними методами теорії метаевристик, практичне ознайомлення з низкою складних

економічних задач, для пошуку розв'язків яких доцільно використовувати метаеврістики.

У результаті вивчення дисципліни студент повинен **отримати такі результати навчання** (знання, уміння тощо):

- мати передові концептуальні та методологічні знання з предметної області та на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з відповідного напряму, отримання нових знань та/або здійснення інновацій;

- формулювати і перевіряти гіпотези; використовувати для обґрунтування висновків належні докази, зокрема, результати теоретичного аналізу, експериментальних досліджень, спостережень, наявні літературні дані з метою розв'язання значущих наукових та науково-прикладних проблем

- розробляти та досліджувати фундаментальні та прикладні моделі соціально-економічних процесів і систем, ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів у економіці та дотичних міждисциплінарних напрямах з метою досягнення економічного та соціального розвитку в умовах глобалізації;

Та набути такі **компетенції** як:

– здатність до розуміння основних концепцій, історичних витоків, сучасного стану та тенденції розвитку економіки; оволодіння термінологією з досліджуваного наукового напряму;

– здатність здійснювати планування та виконання оригінальних досліджень, досягати наукових результатів, які створюють нові знання як в предметній області, так і в міждисциплінарних напрямах, і можуть бути опубліковані у провідних вітчизняних та міжнародних наукових виданнях з галузі соціальні та поведінкові науки та суміжних галузей;

– здатність використовувати сучасні методології, методи та інструменти емпіричних і теоретичних досліджень у галузі, методи комп'ютерного моделювання, сучасні цифрові технології, бази даних та інші електронні ресурси, спеціалізоване програмне забезпечення у науковій та науково- педагогічній діяльності;

– здатність усно і письмово презентувати та обговорювати результати наукових досліджень та/або інноваційних розробок українською та іноземною мовами, демонструвати глибоке розуміння іншомовних наукових текстів за напрямом досліджень; володіти навичками академічного письма;

– здатність обґрунтовувати та готувати економічні рішення на основі розуміння закономірностей розвитку соціально-економічних систем і процесів, питань європейської та євроатлантичної інтеграції; із застосуванням математичних методів та моделей з врахуванням економічних ризиків та можливих соціально-економічних наслідків.

Лабораторні заняття з дисципліни «*Системи підтримки прийняття рішень на базі штучного інтелекту в економіці*» складаються з чотирьох лабораторних робіт, які логічно і змістовно відповідають робочій програмам дисципліни. Виконання лабораторних робіт дозволить студенту оволодіти необхідними знаннями та уміннями, які передбачені даною дисципліною, а також здобути відповідні практичні навички.

Набуті студентам знання та практичні навички з дисципліни «*Системи підтримки прийняття рішень на базі штучного інтелекту в економіці*» будуть необхідні їм при виконанні аналітичних досліджень під час написання кваліфікац. роботи.

Змістовий модуль 1. Дискретні задачі управління

Лабораторна робота № 1

Тема: ДВІЙКОВІ ПРЕДСТАВЛЕННЯ

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

Ключові слова: двійкові представлення, код.

Завдання:

1. Ознайомитись із програмою «Двійкові представлення»
2. Виконати тренувальні розрахунки
3. Виконати 3 типи розрахунків по 8 завдань у кожному та зберегти їх у базі даних
   а) двійкові представлення цілих чисел
   б) код Грея
   в) двійкові представлення 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 представлення цілих чисел

Нехай задано ціле число x.

Двійкове представлення цілого числа є рядок $zb_nb_{n-1}....b_1b_0$, де кожне з чисел $z, b_i \;\; i = 1,2,...,n$ 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 виявиться рівним 0. Число z вибирається рівним 1, якщо число x було від'ємним і 0 в іншому випадку.

Зворотне перетворення двійкового числа $zb_nb_{n-1}....b_1b_0$ на десяткове число проводиться за формулою

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

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

Код Ґрея. Алгоритм обчислення коду Грея

Коди Грея легко виходять з двійкових чисел шляхом побітової операції «Додавання за модулем 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 «Додавання за модулем 2»; біти нумеруються справа на ліво, починаючи з меншого.

Приклад: Перетворити двійкове число 11010 на 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 наведений вище 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 представлення 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);
- кількість знаків k у дробовій частині двійкового числа.

В двійковому представлення дробового числа є два рядки рядок $zb_nb_{n-1}....b_1b_0$ і $f_1f_2....f_k$. Кожне з чисел z, b_i, f_j $i=1,2,...,n$, $j=1,2,...,k$ дорівнює 0 або 1. Послідовність $zb_nb_{n-1}....b_1b_0$ є двійковим представленням 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 $zb_nb_{n-1}....b_1b_0$ і $f_1f_2....f_k$ відновлює десяткове дробове число x. Для цього спочатку по послідовності $zb_nb_{n-1}....b_1b_0$ відновлюється ціле число y. По послідовності $f_1f_2....f_k$ отримуємо дробову частину числа за такою формулою: $r = \frac{f_1}{2} + \frac{f_2}{4} + ... + \frac{f_k}{2^k}$. Обчислюване число x обчислюємо за формулою: $x = s*\left(abs(y) + r\right)$ 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 представлення 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 представлення 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 (рис.1.1).

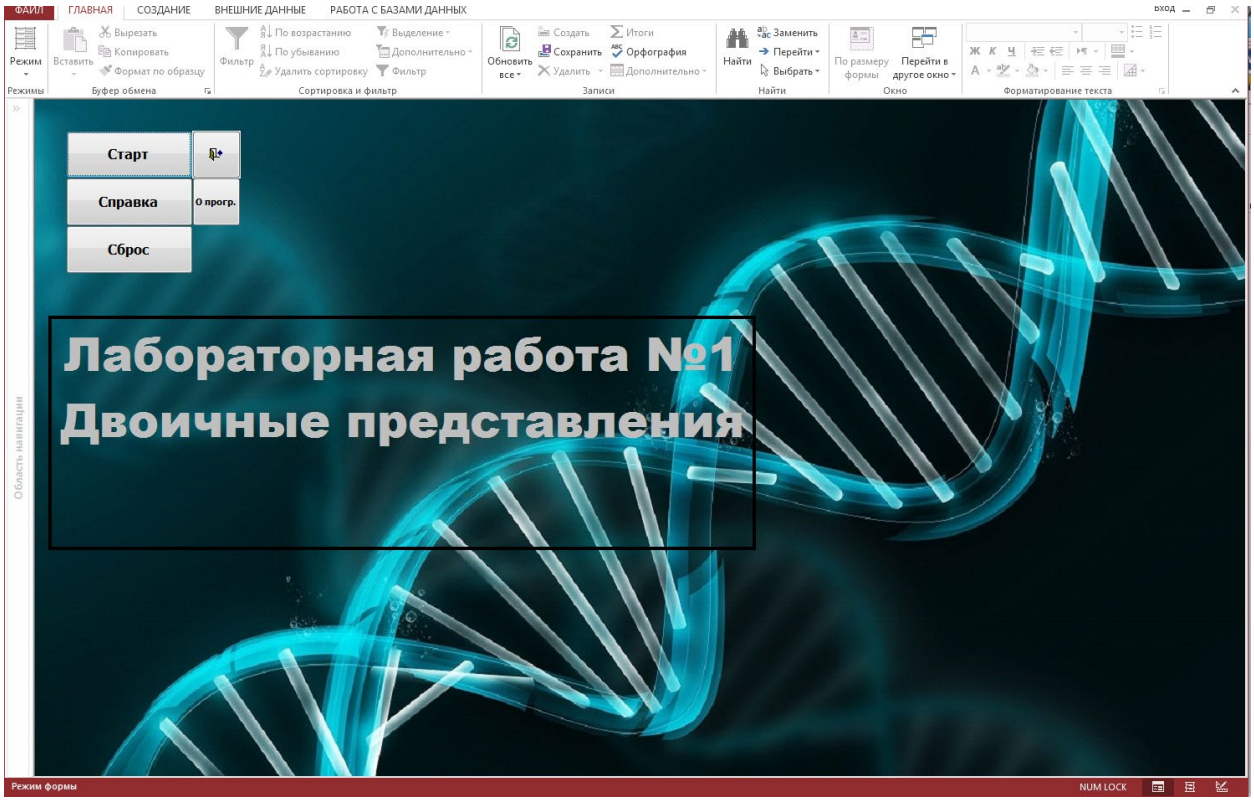


Рис. 1.1 – Початкове вікно програми «Двійкові представлення чисел»

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

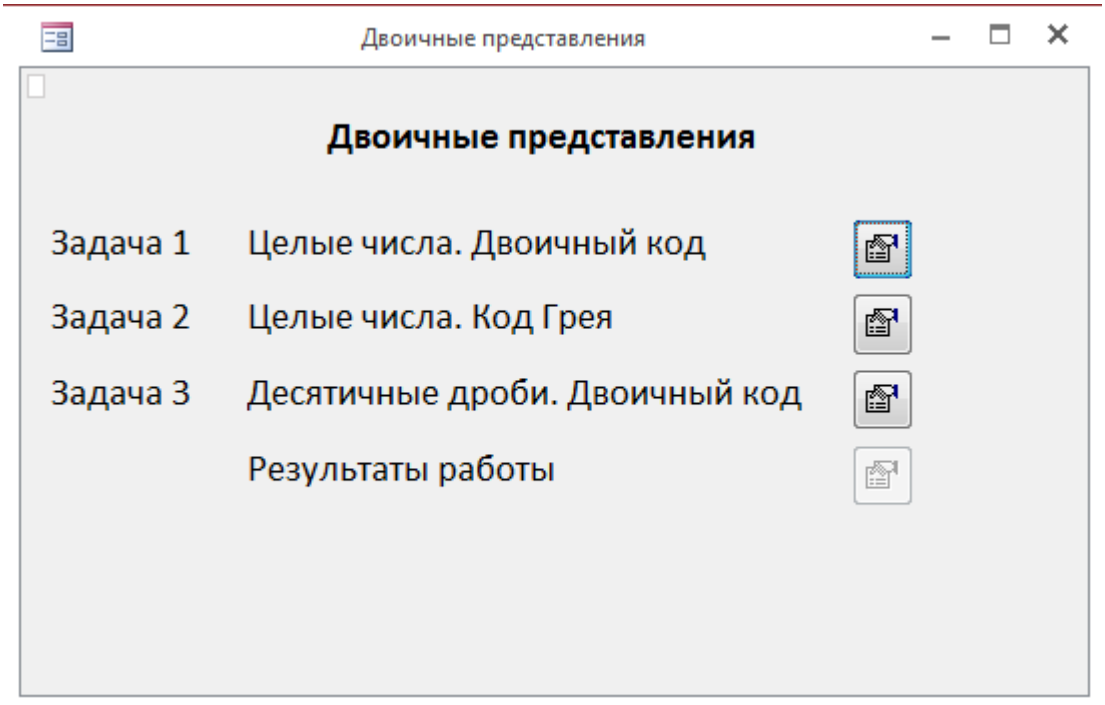


Рис.1.2 – Перше робоче вікно програми «Двійкові представлення чисел»

На початку виконання роботи 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 рис. 1.3.

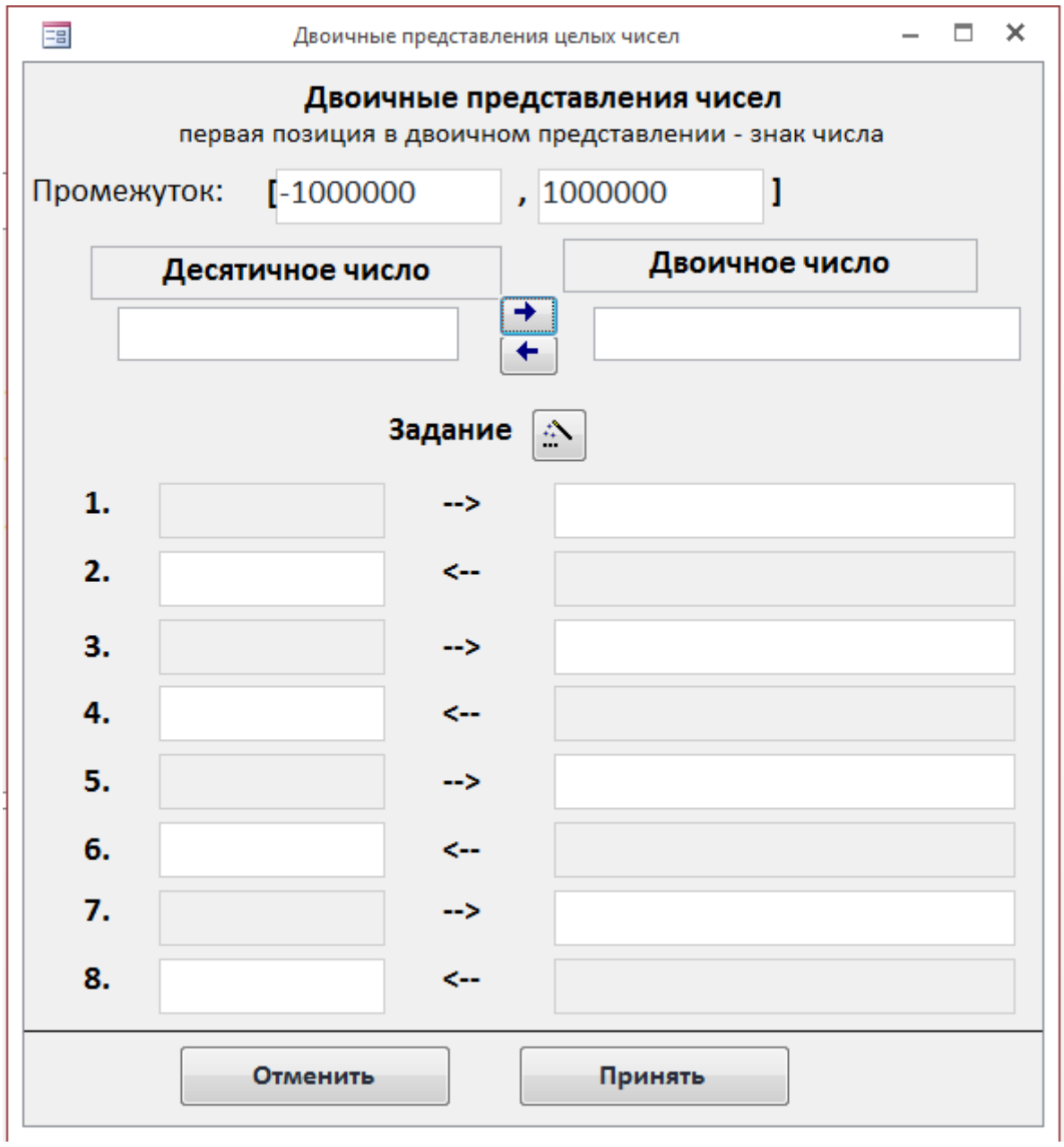


Рис. 1.3. – Вікно задачі №1

Генерація восьми завдань відбувається за натисканням кнопки. Завдання полягають в 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 (рис. 1.4).

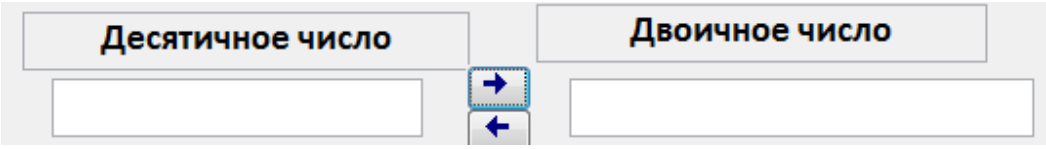


Рис. 1.4. – Вікно пробних обчислень у вікні задачі №1

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 представлення 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.

Змістовий модуль 2. Бінарний простір і генетичний алгоритм.

Лабораторна робота № 2

Тема: ПОБУДОВА ЕВОЛЮЦІЙНОЇ МОДЕЛІ.

***Мета роботи**:* познайомиться зі складовими еволюційної моделі. Познайомитися з генетичним алгоритмом та його узагальненнями Розглянути проблеми, що виникають при побудові еволюційних моделей. Побудувати еволюційну модель для задачі пошуку оптимуму функції.

Ключові слова: еволюційна модель, фітнес-функція, оператор кросоверу, оператор мутації, турнірний відбір.

☐ *Завдання:*

1. Познайомитися з комп'ютерною системою «Простий генетичний алгоритм».
2. Згенерувати функцію для пошуку та знайти її точки мінімуму.
3. Розглянути окремі етапи роботи генетичного алгоритму для згенерованої функції. Знайти наближений оптимальний розв'язок задачі.
4. Оформити звіт з лабораторної роботи, який включає висновки щодо використання генетичного алгоритму.

☐*Теоретичні відомості*

Модель Холланда та класичний генетичний алгоритм

У класичній моделі Холланда припустимі розв'язки оптимізаційної задачі є бінарними послідовностями фіксованої довжини *N*.

Нехай U – множина припустимих розв'язків задачі $F(u) \to \max\limits_{u \in U}$.

Кодом будемо називати ін'єктивне відображення W цієї множини в множну $\{0,1\}^N$.

Кодом елемента $u \in U$ називається його образ $W(u)$.

Образ $X=W(U)$ множини припустимих рішень називається множиною припустимих кодів. Саме ця множина є базовою для еволюційної моделі оптимізаційної задачі. Для простоти суперпозицію цільової функції та коду будемо також позначати $F : X \to R^1$. Крім того, будемо припускати також, що $\forall x \in X \quad F(x) > 0$.

Опишемо інші складові класичної моделі.

Початкова популяція Y_0 формується випадковим чином. Імовірність входження елемента з X початкову популяцію дорівнює $1/|X|$.

Правило селекції – пропорційне, на k-му етапі ймовірність вибору елемента для схрещування пропорційна значенню цільової функції на цьому елементі, тобто ймовірність $P(y)$ вибору елемента для схрещування визначається формулою

$$P(y) = \frac{F(y)}{\sum_{z \in Y_k} F(z)}$$

Опишемо тепер одноточкове правило кросовера в моделі Холланда. Нехай задані два припустимі рішення у вигляді слів $x = (x_1, x_2, ..., x_n)$ і . $y = (y_1, y_2, ..., y_n)$. Випадковим чином вибирається номер $i \in \{1, 2, ..., n\}$. З двох рішень-батьків будуються два нових рішення-нащадки $x' = (x_1, x_2, ..., x_i, y_{i+1}, ..., y_n)$ і $y' = (y_1, y_2, ..., y_i, x_{i+1}, ..., x_n)$, які заміняють рішення-батьків у популяції (рис.2.1).

Оператор мутації змінює у рішенні $x = (x_1, x_2, ..., x_n)$ випадково вибраний елемент j за правилом $x_j \to 1 - x_j$.

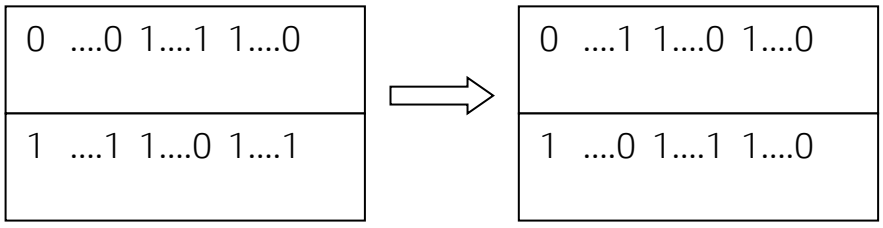


Рис.2.1 – Кросовер на бінарних послідовностях

Правило відбору видаляє із популяції рішення-батьків після виконання процедури кросоверу. Правило зупинки є стандартним – або досягнення певного числа поколінь, або досягнення граничного часу роботи алгоритму.

Наведену вище еволюційну модель Холланда будемо позначати G1. Генетичний алгоритм, що відповідає моделі G_1, отримав назву класичного генетичного алгоритму на бінарних послідовностях (КГА).

Як правило, модель G_1 використовується при розв'язанні задач багатовимірної безумовної оптимізації в евклідовому просторі.

Хід лабораторної роботи з використанням програми «Простий генетичний алгоритм».

Програма передається студентам як файл Work_2_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.

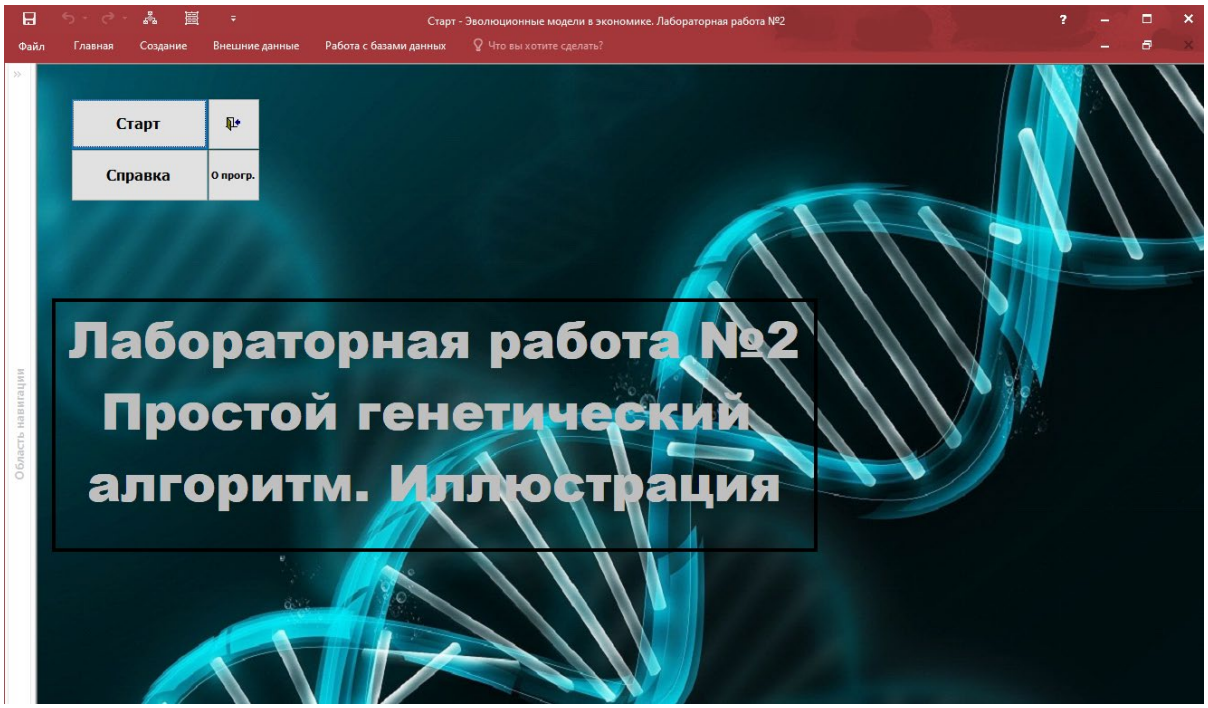


Рис.2.2 – Початкове вікно програми «Простий генетичний алгоритм»

Після натискання кнопки «Старт» відбувається перехід у перше робоче вікно програми (рис.2.3).

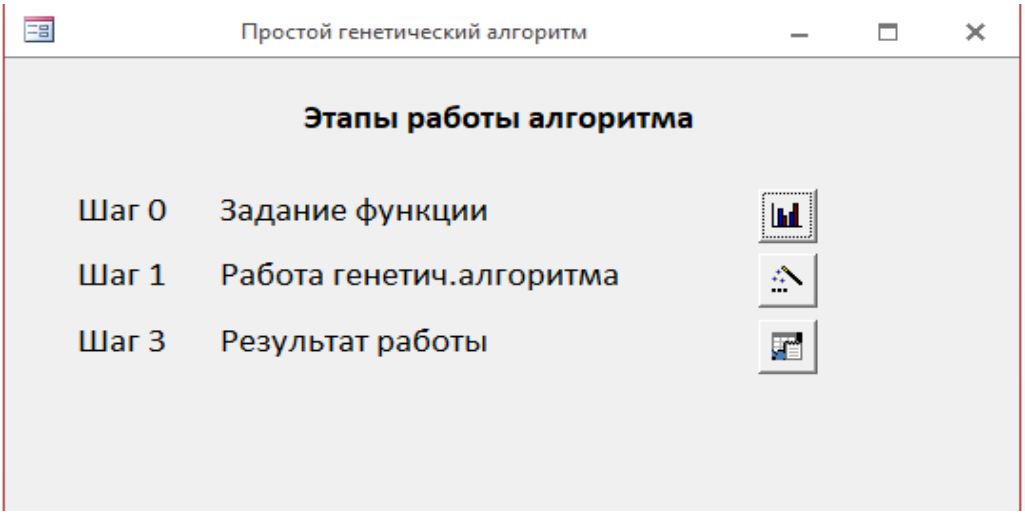


Рис.2.3 – Вікно задачі №1

Задаються параметри тестової функції. Генерація тестової функції відбувається при натисканні кнопки формування (рис.2.4).

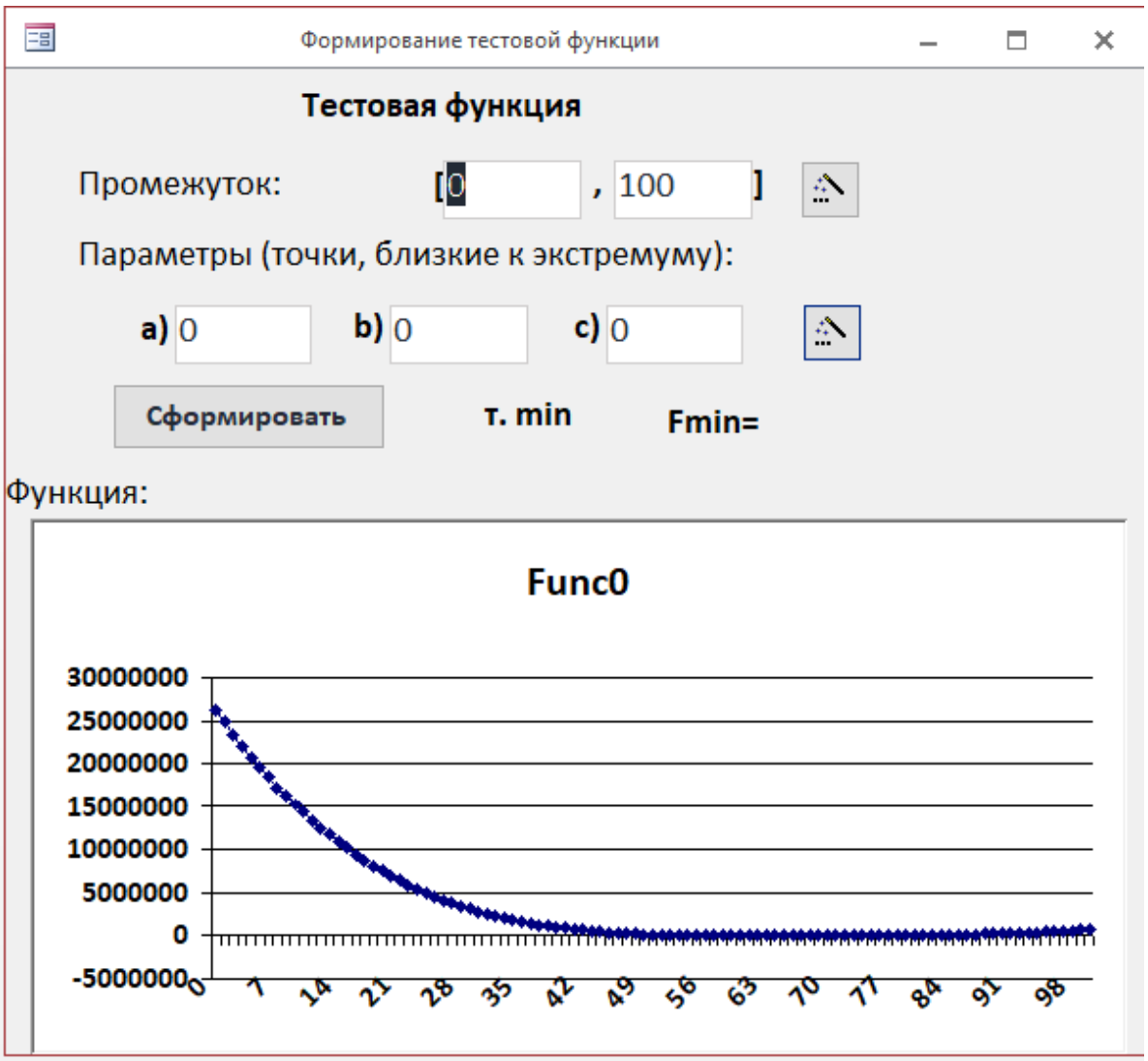


Рис.2.4 – Вікно вибору параметрів тестової функції

?

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?

Змістовий модуль 3. Метричні простори. Простори зі структурою опуклості.

Лабораторна робота № 3

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 метаевристики».

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

Завдання:

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

Теоретичні відомості та опис комп'ютерної програми

Комп'ютерна система «Фрагментарні структури та метаевристики» призначена для тестування та оцінки якості різних метаевристик 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

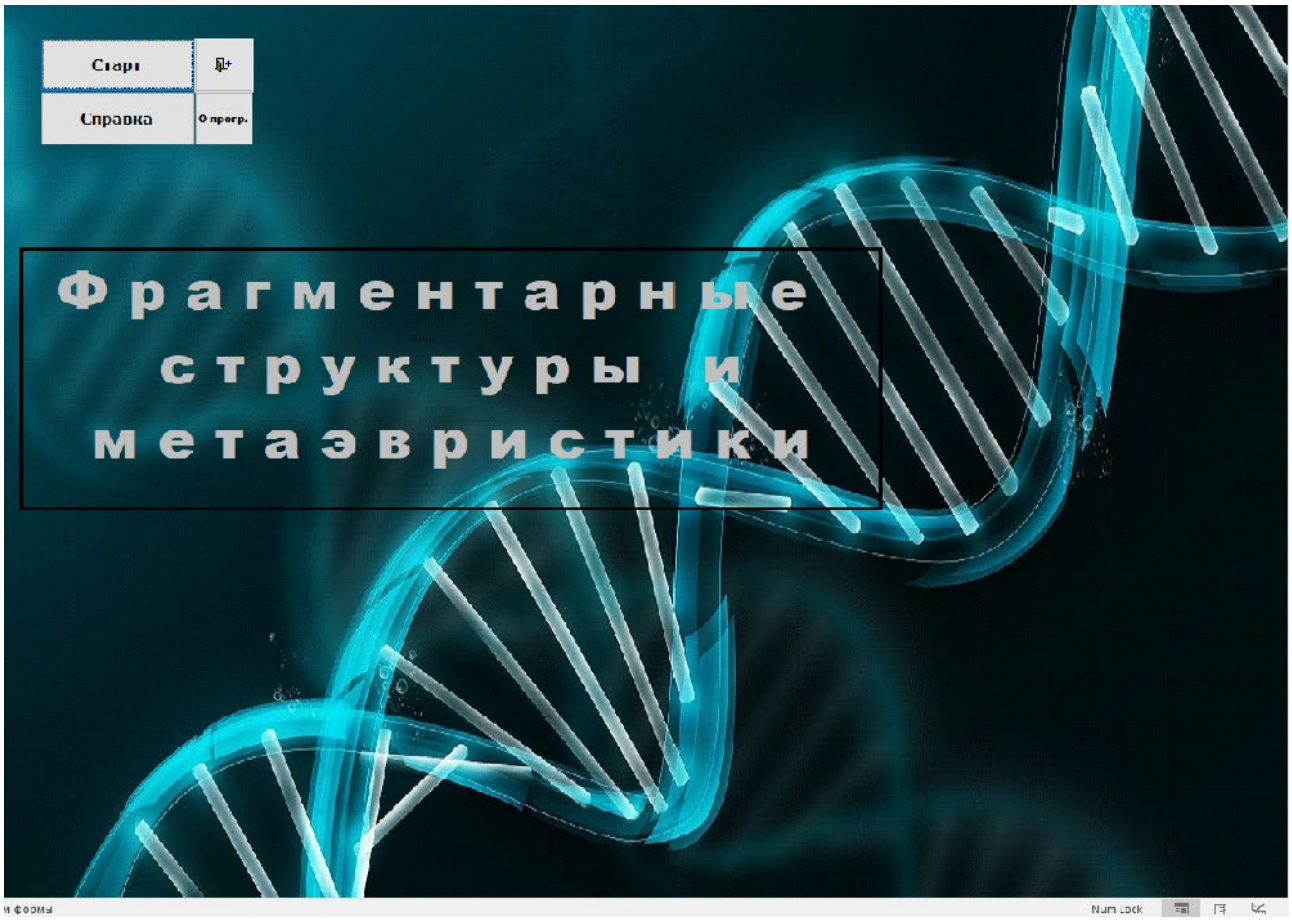


Рис. 3.1 Комп'ютерна система «Фрагментарні структури та метаеврістики»

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.

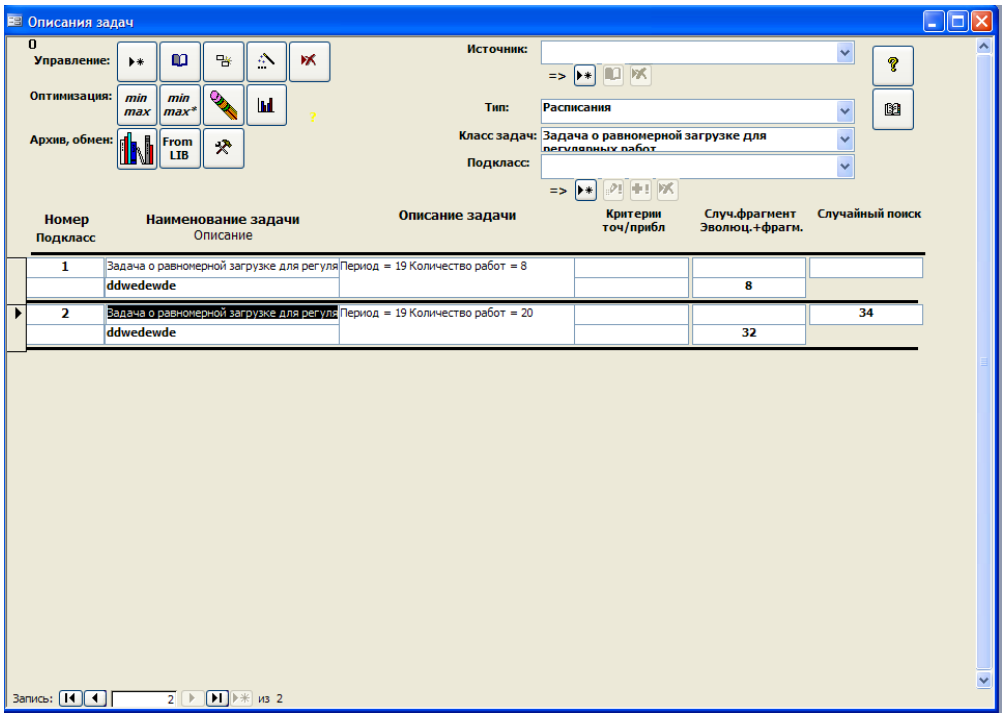


Рис 3.2 – Головне вікно системи «Фрагментарні структури та метаеврістики»

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 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).

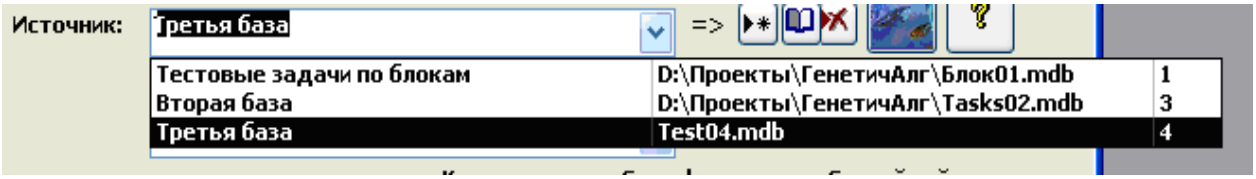


Рис. 3.3 – Вибір джерел даних

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)	Количество работ = 11	1556	2198	1636
1	Версия №1.			1615	
2	Задача директора (задача одного станка)	Количество работ = 91	153315	214719	193372
1	Версия №2.			188431	
3	Задача директора (задача одного станка)	Количество работ = 94	134087	205646	175898
1	Версия №3.			180697	
4	Задача директора (задача одного станка)	Количество работ = 54	57499	79919	68619
1	Версия №4.			67814	
5	Задача директора (задача одного станка)	Количество работ = 100	183017	267660	239875
1	Версия №5.			232042	
6	Задача дир[illegible]ча одного станка)	Количество работ = 42	31057	41610	[illegible]

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 припустимих 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).

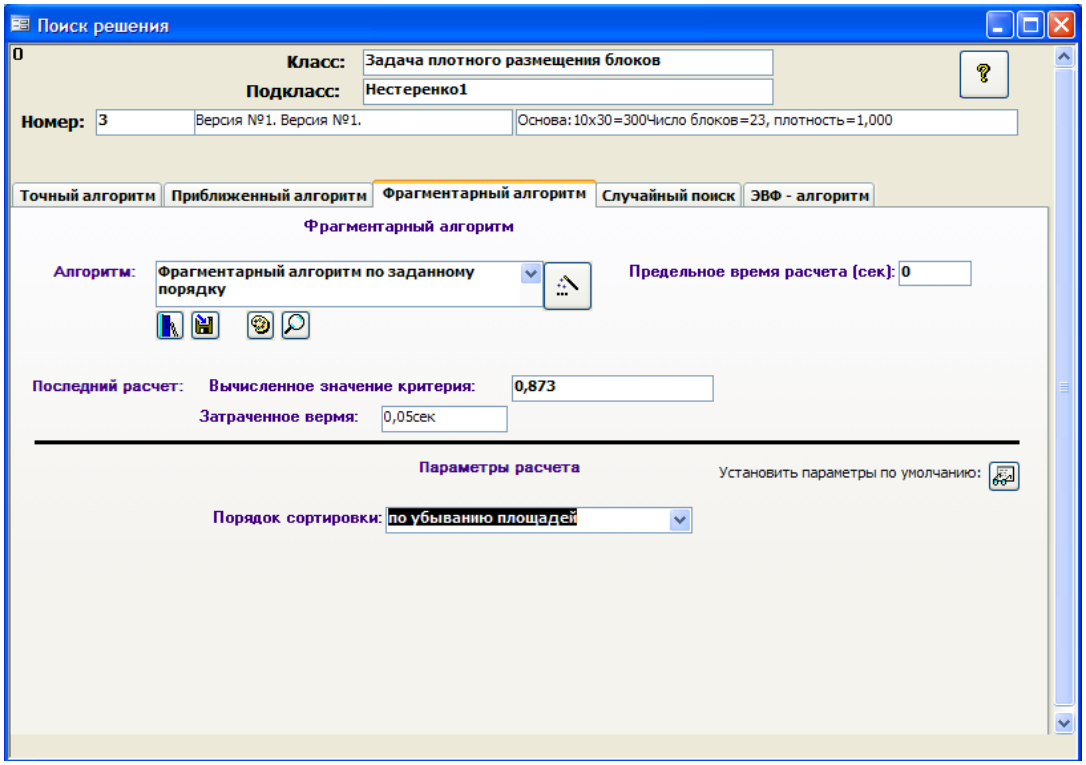


Рис. 3.5 – Вікно пошуку рішень

Область даних форми "Пошук рішення" розбита на п'ять зон, кожна з яких присвячена одному з типів алгоритмів для обраного класу задач. 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 виноситься 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 та формування звіту

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).

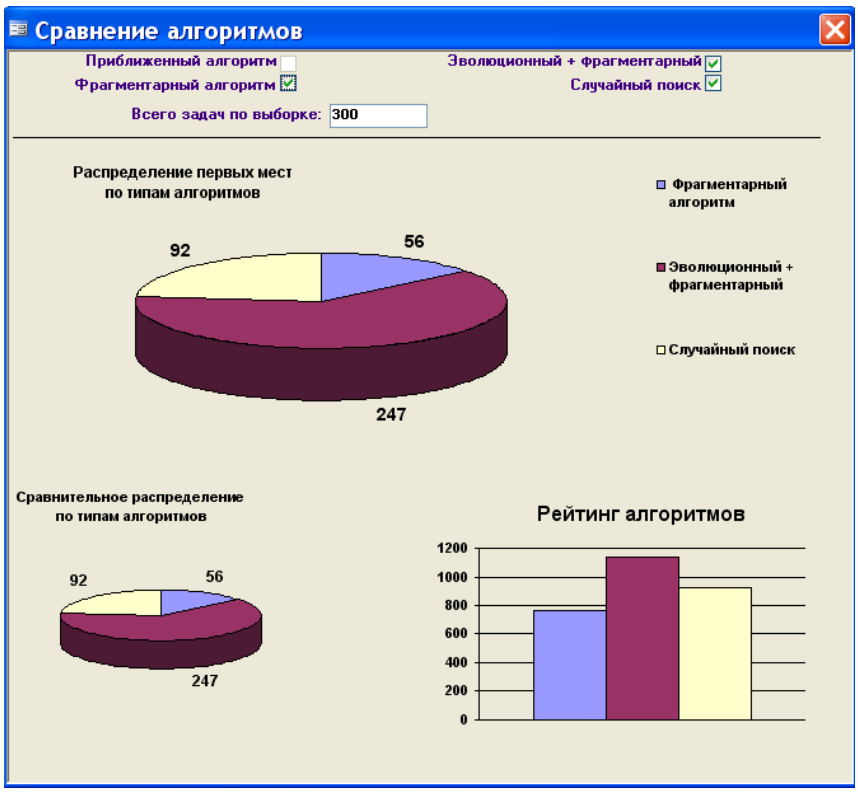


Рис. 3.6 – Вікно діаграм порівняння якості роботи різних алгоритмів

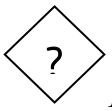
У заголовку вікна виділяються типи алгоритмів порівняння (типи алгоритмів, які описані цієї серії завдань, недоступні для позначки).

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-те два бали, за 4-те 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 фрагментарної структури.
2. Опишіть особливості еволюційної моделі на фрагментарній структурі.
3. Для чого призначена Комп'ютерна система «Фрагментарні структури та метаевристики»?
4. Як працює фрагментарний алгоритм?
5. Наведіть приклад роботи фрагментарного алгоритму для задачі комівояжера?

Змістовий модуль 4. Прикладні задачі економічного управління.

Лабораторна робота № 4

Тема: МЕТАЕВРІСТИКИ ДЛЯ ЗАДАЧІ ДОСТАВКИ ВАНТАЖІВ

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

Ключові слова: фрагментарна модель, еволюційно-фрагментарний алгоритм, задача доставки вантажів.

Теоретичні відомості

Задача доставки вантажів

Розглянемо завдання доставки вантажів з точки з номером 0 (депо) до n інших точок площини за допомогою декількох однакових машин, вантажопідйомність та час роботи яких обмежені. Маршрут кожної машини починається і закінчується депо, причому протягом маршруту машина може кілька разів повертатися в депо для завантаження. У кожній точці задана потреба у вантажі, який передбачається ділимим. Завдання полягає у побудові системи маршрутів, мінімальної вартості задоволення всіх потреб у вантажі.

В найпростішому випадку, коли немає обмежень на вантажопідйомність та час роботи машини маємо класичну задачу комівояжеру.

Вказано вартість оренди однієї машини та вартість одиниці колії машини. Нижче буде показано, що завдання доставки вантажу в наведеному формулюванні може розглядатися як оптимізаційна задача на фрагментарній структурі і, відповідно, може бути зведена до завдання безумовної оптимізації на безлічі перестановок.

Задача оптимізації на орієнтованій фрагментарній структурі

Орієнтованою фрагментарною структурою (X, E) на кінцевій множині будемо називати множину кінцевих послідовностей E елементів множини X,

таких, якщо послідовність $\{x_1, x_2, ..., x_m\}$ належить множині *E*, то і будь-яка її початкова підпослідовність $\{x_1, x_2, ..., x_k \mid k < m\}$ також належить множині *E*.

Послідовності елементів із множини *E* називаються припустимими фрагментами. Припустимі фрагменти потужності 1 називаються елементарними. Припустимий фрагмент називатимемо максимальним, якщо він не є підпослідовністю жодного іншого припустимого фрагмента. За визначенням порожня множина є припустимим фрагментом.

Якщо задана ефективна процедура перевірки належності послідовності множині *E*, то кожен максимальний фрагмент можна побудувати наступним жадібним алгоритмом:

а) спочатку вибирається деяке лінійне впорядкування множини X;

б) на початковому кроці вибирається порожня послідовність $X_0 = \varnothing$;

в) на кроці з номером *k* +1 вибирається перший за заданим порядком елемент $x_{k+1} \in X \setminus \{x_1, x_2, ..., x_k\}$ такий, що виконується умова $\{x_1, x_2, ..., x_{k+1}\} \in E$. Цю умову називатимемо умовою приєднання;

г) алгоритм закінчує роботу, якщо на черговому *k*-му кроці не вдалося знайти елемент $x \in X \setminus X_k$ з необхідною властивістю.

Результат роботи фрагментарного алгоритму (максимальний фрагмент) залежить від того, які будуть вибиратися елементи на кожному кроці, тобто від початкового впорядкування множини *X*. Таким чином, виникає природне відображення з множини перестановок S_n розмірності $n = \mid X \mid$ в множину максимальних фрагментів фрагментарної структури. Очевидно, це відображення є сюр'єктивним.

Нехай задана фрагментарна структура на скінченній множині *X*. Нехай визначена монотонна за включенням функція - критерій $f : E \to R^1$, яка кожному припустимому фрагменту ставить у відповідність деяке дійсне число, тобто $\forall E_1, E_2 \in E$ з умови $E_1 \subseteq E_2$ випливає, що $f\left(E_1\right) \le f\left(E_2\right)$ (або $f\left(E_1\right) \ge f\left(E_2\right)$).

Одним із способів завдання подібної функції є адитивний. А саме: кожному елементу $x \in X$ ставиться у відповідність невід'ємне число (вага) $w(x) \ge 0$. Функція на послідовностях фрагментарної структури визначається співвідношенням $f(\{x_1, x_2, .., x_k\}) = \sum_{i=1}^{k} w(x_i)$.

Задача оптимізації на фрагментарній структурі полягає у відшуканні припустимого фрагмента з максимальним значенням критерію. Очевидно,

оптимальним розв’язком цієї задачі при монотонно зростаючому критерії буде один із максимальних фрагментів. Принаймні максимальний фрагмент завжди є серед оптимальних розв’язків. Зазначимо, що умова максимальності фрагмента може бути обов'язковою у задачі оптимізації і без додаткових припущень про цільову функцію.

Таким чином будь-яка задача оптимізації на фрагментарній структурі (з умовою максимальності фрагмента-розв’язку) може бути зведена до комбінаторної задачі оптимізації на множині перестановок розмірності *n*. З кожною функцією $f : E \to R^1$, що задана на фрагментарній структурі, пов'язана накриваюча функція $F : S_n \to R^1$, яка відображає множину перестановок в R^1. Значення цієї функції на перестановці визначається наступним алгоритмом. Спочатку з перестановки з допомогою фрагментарного алгоритму будується максимальний фрагмент $e \in E$, а потім обчислюється значення функції $f(e)$ на цьому фрагменті.

Еволюційний алгоритм для функцій, що задані на множині перестановок

Розглянемо тепер задачу пошуку оптимального значення функції $F : S_n \to R^1$, заданої на множині перестановок *n* елементів. Для пошуку субоптимальних розв’язків задачі оптимізації цієї функції застосуємо еволюційний алгоритм із геометричним оператором кросоверу. Опишемо принцип роботи такого алгоритму. Як базова множина розв’язків вибирається множина всіх перестановок S_n з *n* елементів. Спочатку за допомогою оператору початкової популяції будується початкова множина розв’язків $Y_0 \subseteq S_n$. Кожен елемент цієї множини – випадково обрана перестановка. На кожному наступному етапі передбачається заданим деяка множина перестановок - поточна популяція. На першому кроці це множина $Y = Y_0$. Для кожного з елементів множини *Y* обчислюється значення критерію селекції, який в даному випадку є накриваючим відображенням вихідної задачі.

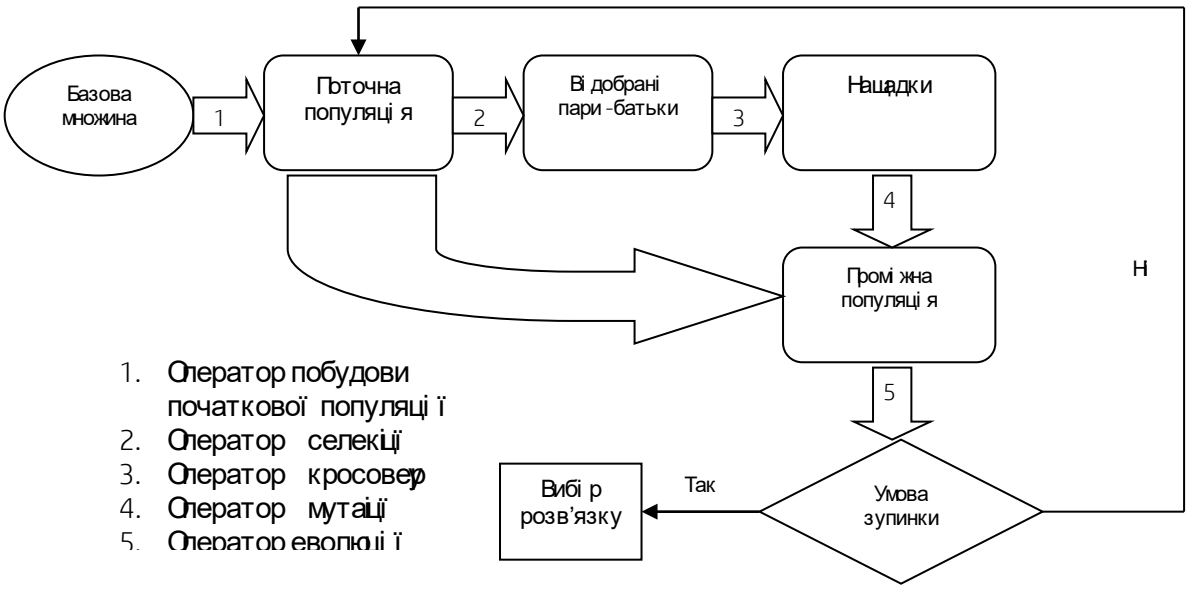


Рис.4.1 – Блок-схема еволюційного алгоритму

Далі за допомогою оператора відбору поточної популяції Y вибирається множина пар для кросоверу. До кожної пари з вибраної множини пар застосовується оператор кросоверу, який кожній парі перестановок «батьків» ставить у відповідність перестановку-«нащадок» а потім до результату кросоверу з ймовірністю $\alpha \in (0,1)$ застосовується оператор мутації.

Таким шляхом знаходиться множина елементів – нащадків $\tilde{Y}$. До проміжної популяції $Y \cup \tilde{Y}$, що є об'єднанням поточної популяції та множини нащадків, застосовується оператор еволюції, який виділяє цієї множині нову поточну популяцію. Процес еволюції повторюється до того часу, доки буде виконано умову зупинки еволюційного алгоритму. «Найкращий» за значенням функції F розв'язок в останній поточній популяції береться як наближений (субоптимальний) розв'язок задачі оптимізації. Блок схему еволюційного алгоритму наведено на рис.4.1.

Геометричний кросовер на множині перестановок

Опишемо тепер геометричний оператор кросоверу на перестановках. Множину перестановок можна розглядати як метричний простір з метрикою Кендала. У метриці Кендала відстань $\rho(u,v)$ між двома перестановками $u,v \in S_n$ визначається як мінімальна кількість транспозицій сусідніх елементів, які необхідно виконати, щоб перевести одну перестановку в іншу.

Кросовер $K : S_n \times S_n \to S_n$ є геометричним в метриці Кендала, якщо для будь-яких перестановок має місце рівність $\rho(u,v) = \rho(u,K(u,v)) + \rho(K(u,v),v)$. Це означає, що результат кросовера лежить на відрізку, що з'єднує перестановки u,v.

Пропонується наступний алгоритм для реалізації геометричног7о кросоверу. Нехай $s=(s_1,s_2,...,s_n)$ і $t=(t_1,t_2,...,t_n)$ – дві довільні перестановки. Перестановка на відрізку поміж них будується так: послідовності *s* і *t* проглядаються у порядку прямування елементів. На *k*-му етапі вибирається будь-який з перших елементів послідовностей і додається в нову перестановку-результат. Потім цей елемент вилучається з двох послідовностей *s* та *t.* Наприклад, результатом роботи алгоритму на перестановках (2,3,6,1,7,8,4,5) та (4,6,7,1,3,2,8,5) буде перестановка (2,3,4,6,1,2,8,5). Перестановка-результат завжди знаходиться на відрізку [*s*, *t*] в метриці Кендала.

Фрагментарна модель задачі про доставку вантажу

Повернемося тепер до задачі про доставку вантажу, умови якої було описано вище. Покажемо, що ця задача може розглядатися як задача оптимізації на фрагментарній структурі. Множина *X* складається з точок маршруту. Причому кожна точка крім точки 0 входить у цю множину 1 раз, а точка 0 входить у множину *X* у кількості *n*+1 екземпляра. Тепер опишемо припустимий фрагмент як послідовність точок множини *X*. Кожен припустимий фрагмент можна уявити як конкатенацію послідовностей $E_1,E_2,...,E_k$. Кожній з таких послідовностей відповідає маршрут, який починається в точці 0, проходить послідовно через всі точки послідовності. Причому довжина цього маршруту не перевищує довжини пробігу машини з відрахуванням відстані від останньої точки маршруту до точки 0. У послідовності можуть бути кілька точок 0, які відповідають поверненням в депо і перезавантаження машини. Сума потреб у точках між двома черговими поверненнями в депо не повинна перевищувати вантажопідйомності машини. У конкатенації послідовностей $E_1,E_2,...,E_k$ кожна точка $x\in X$, за винятком депо (точка 0), входить рівно по одному разу. Легко переконається, що такі фрагменти утворюють фрагментарну структуру. Припустимому розв'язанню задачі відповідає максимальний фрагмент, який містить усі точки множини *X*. Нехай вартість оренди однієї машини дорівнює *c*, а вартість однієї одиниці довжини маршруту – *p*. Тоді значення цільової функції на припустимому фрагменті визначається формулою $kc+pL$, де L - довжина маршруту, що відповідає даному припустимому фрагменту.

Таким чином, задача доставки вантажу є оптимізаційною задачею на фрагментарній структурі і, отже, для пошуку субоптимальних розв’язків цієї задачі може бути використаний еволюційний алгоритм для задач оптимізації

на фрагментарній структурі.

Завдання:

Задана матриця вартостей перевозок між 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

Приклад розв'язування завдання

Розглянемо варіант задачі комівояжера з 5-ма очками

Варіант № 1

0	106	128	181	154
95	0	31	125	161
118	124	0	20	9
69	43	23	0	66
127	92	87	91	0

1 крок. Задача має фрагментарну структуру. Припустимими фрагментами тут є будь-які послідовності номерів точок 1,2,3,4,5, які починаються з номеру 1 (депо) і всі номери точок не повторюються. Максимальним фрагмент – це перестановка з п'яти чисел, яка починається з числа 1. Значення цільової функції – довжина відповідного маршруту, що починається та закінчується у першій точці послідовності.

2 крок. Вибираємо послідовність s_1 точок, починаючи з 1 и додаючи інші номери точок без повторень, поки всі точки не будуть обрані. Наприклад: ***1-***3-5-4-2 . Значення критерію на цієї послідовності дорівнює

$$F(s_1)= 128+9+91+43+95=366.$$

3 крок. Вибираємо іншу послідовність точок (припустимий фрагмент) s_2. Наприклад, ***1-2***-4-5-3 . Значення критерію на цієї послідовності дорівнює

$$F(s_2)=106+125+66+87+118=502.$$

4 крок. Виконаємо операцію кросоверу для перестановок s_1 та s_2.

$$K(s_1,s_2)= 1\text{-}3\text{-}2\text{-}4\text{-}4$$

Виконаємо операцію Мутації (переставимо номери другої і четвертої позиції): $s_3 = M(K(s_1,s_2))= 1\text{-}4\text{-}2\text{-}3\text{-}4$. Значення критерію для цієї перестановки

$$F(s_3)=181+43+31+20+69=344$$

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 геометрічним?
3. Які особливості має метаевристика 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 припустимий 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 неполіноміальний час від довжини входу
- : який вирішує масову задачу за поліноміальний 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 фрагментарну структуру, 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. Optimization Methods and Applications : In Honor of Ivan V. Sergienko's 80th Birthday / eds. S. Butenko, P. M. Pardalos, V. Shylo. New York : Springer, 2017. 639 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi66/0048529.pdf.
2. Applied Metaheuristic Computing / ed. by P.-Y. Yin, R. Chang, Y. Gheraibia [et al]. Basel : MDPI, 2022. 684 p. URL: http://files.znu.edu.ua/files/Bibliobooks/Inshi71/0052038.pdf.
3. Genetic Algorithms / ed. by S. Ventura, J. M. Luna, J. M. Moyano. London : IntechOpen, 2022. 164 p. URL: http://files.znu.edu.ua/files/Bibliobooks/Inshi71/0052032.pdf.
4. Kozin V., Maksyshko N. K., Perepelitsa V. A. Fragmentary Structures in Discrete Optimization Problems. *Cybernetics and Systems Analysis*. 2017. Vol. 53, Issue 6. P. 931–936. URL: https://link.springer.com/article/10.1007/s10559-017-9995-6.
5. The Application of Ant Colony Optimization / ed. by A. Soofastaei. London : IntechOpen, 2022. 87 p. URL: http://files.znu.edu.ua/files/Bibliobooks/Inshi71/0052033.pdf.
6. Kozin I. V., Batovskyi S. E. Fragmentary Structures in a Two-Dimensional Strip Packing Problem. *Cybernetics and Systems Analysis*. 2019. Vol. 55. P. 943–948. URL: https://link.springer.com/article/10.1007/s10559-019-00204-w.
7. Kozin I. V., Maksyshko N. K., Selyutin E. K. The Usage of Evolutionary Algorithms for Searching Optimal Classifications. *Bulletin Zaporizhzhya national university. Economic sciences*. 2019. № 2 (42). P. 73-78.
8. Козін І. В. Еволюційні моделі в дискретній оптимізації: монографія. Запоріжжя: Запорізький національний університет, 2019. 204 с.

Додаткова:

1. Kozin I. V., Selyutin E. K., Polyuga S. I. Jumping frog method for optimal classifications. *International Academy Journal*. 2021. Vol. 2(52). URL: https://rsglobal.pl/index.php/wos/article/view/1891.
2. Козін І. В., Землянський О. О. Фрагментарна модель для задачі редагування кластеру. Міжнародний науковий симпозіум «Інтелектуальні

рішення-С» Обчислювальний інтелект (результати, проблеми, перспективи). Теорія прийняття рішень: праці X Міжнар. школи-семінару (Ужгород, 29 верес. 2021 р.). Ужгород, Ужгородський національний університет, 2021. С. 50-51.

3. Козін І., Максишко Н., Терешко Я. Метод імітації відпалу для задачі рівноважного розміщення. *Фізико-математичне моделювання та інформаційні технології*. Львів, 2021. Вип. 32. С. 152-158.

4. Козин И.В., Борю С.Ю., Кривцун Е.В. Математическая модель комбинированной задачи транспортной логистики. *Вісник Запорізького національного університету. Економічні науки*. 2018. № 1. С. 44-51.

5. Kozin I., Selyutin Y. The metaheuristic application in classification problems. Інформаційні технології: теорія і практика: матеріали III Всеукр. наук.-практ. інтернет-конф. здобувачів вищої освіти і молодих учених (Харків, 2020 р.). Харків : ХНУМГ імені О. М. Бекетова, 2020. С. 22-23. URL: https://knit.kname.edu.ua/images/new/web2020/theses_2020.pdf.

6. Advances and Novel Approaches in Discrete Optimization / ed. F. Werner. Basel : MDPI, 2020. 354 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi66/0048566.pdf.

7. Application of Optimization in Production, Logistics, Inventory, Supply Chain Management and Block Chain / eds. B. Sarkar, M. Sarkar. Basel : MDPI, 2020. 618 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi66/0048721.pdf.

8. Applied (Meta)-Heuristic in Intelligent Systems / ed. by P.-Y. Yin. Basel : MDPI, 2022. 184 p. URL: http://files.znu.edu.ua/files/Bibliobooks/Inshi71/0052043.pdf.

9. Cognitive Big Data Intelligence with a Metaheuristic Approach / ed. by S. Mishra [et al.]. London : Academic Press, 2022. 356 p. URL: http://files.znu.edu.ua/files/Bibliobooks/Inshi70/0051084/.

10. Cottle R. W., Thapa M. N. Linear and Nonlinear Optimization. New York : Springer, 2017. 614 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi66/0048527.pdf.

11. Gilli M., Maringer D., Schumann E. Numerical Methods and Optimization in Finance. Cambridge : Elsevier, 2019. 614 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi64/0047491.zip.

12. Multi-Objective Combinatorial Optimization Problems and Solution Methods / ed. by M. Toloo, S. Talatahari, I. Rahimi. London : Academic Press, 2022. 290 p. URL: http://files.znu.edu.ua/files/Bibliobooks/Inshi71/0052034/.

13. Nayak S. Fundamentals of Optimization Techniques with Algorithms. London : Academic Press, 2021. 305 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi70/0050984/.

14. Ng X. W. Concise Guide to Optimization Models and Methods : A Problem-Based Test Prep for Students. Cham : Springer, 2022. 122 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi68/0050242.pdf.

15. Numerical and Evolutionary Optimization 2020 / eds. M. Quiroz, O. Schutze, J. G. Ruiz, L. G. de la Fraga. Basel : MDPI, 2021. 364 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi66/0048722.pdf.

16. Romeo G. Elements of Numerical Mathematical Economics with Excel : Static and Dynamic Optimization. Cambridge : Elsevier, 2020. 816 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/ScienceDirect/0046123.zip.

17. Sustainable Transportation and Smart Logistics: Decision-Making Models and Solutions / ed. by J. Faulin. Amsterdam : Elsevier, 2019. 507 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi68/0050260/.

18. Uncertain Multi-Criteria Optimization Problems / ed. D. Pamucar. Basel : MDPI, 2021. 86 p. URL: http://ebooks.znu.edu.ua/files/Bibliobooks/Inshi66/0048553.pdf.

19. Yang X. Nature-Inspired Optimization Algorithms. 2nd ed. London : Academic Press, 2021. 292 p. URL: http://files.znu.edu.ua/files/Bibliobooks/Inshi71/0052047/.

20.

Інформаційні ресурси:

1. Аналитичні технології. URL: http://www.neuroproject.ua/what.php.

2. Kasahara Lab Waseda University Optimal Schedules for Prototype Standard Task Graph Set. URL: www.kasahara.elec.waseda.ac.jp/schedule/index.html.

3. Sean Luke Essentials of Metaheuristics. Lulu. URL: http://cs.gmu.edu/~sean/book/metaheuristics/.

Навчально-методичне видання
(українською мовою)

Козін Ігор Вікторович
Максишко Наталія Костянтинівна

Системи підтримки прийняття рішень на базі штучного інтелекту в економіці

Методичні рекомендації до лабораторних занять
для здобувачів другого ступеня вищої освіти зі спеціальності «Економіка»

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