**Опис комп'ютерної системи «Фрагментарні структури та метаеврістики»**

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

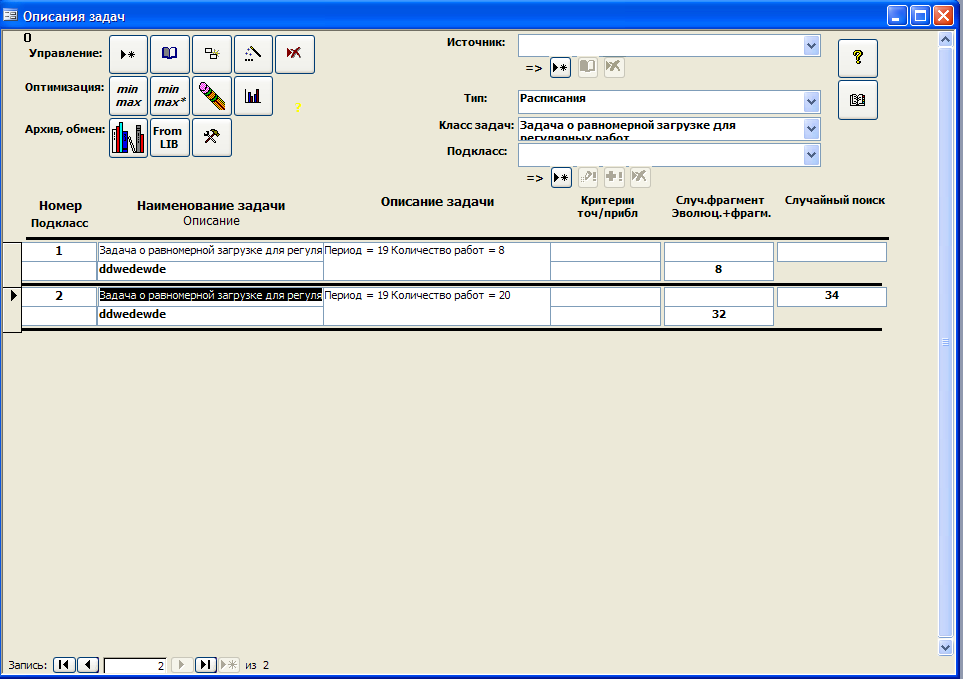


Рис 3.2 – Головне вікно системи «Фрагментарні структури та метаеврістики»

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).

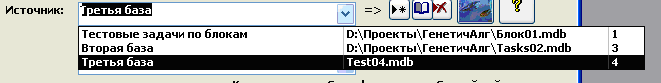


Рис. 3.3 – Вибір джерел даних

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.



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).

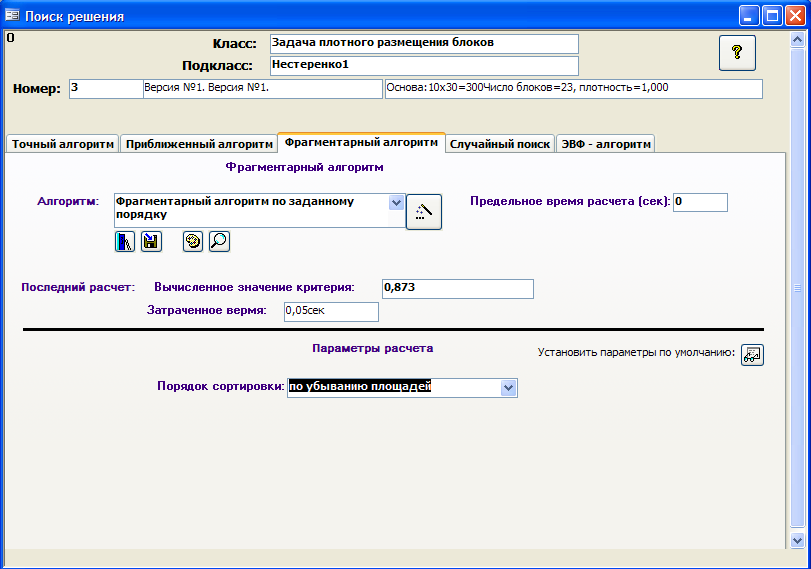


Рис. 3.5 – Вікно пошуку рішень

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 виноситься значення, отримане на останньому етапі відповідного алгоритму.

Значення критерію може мати вигляд , 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).

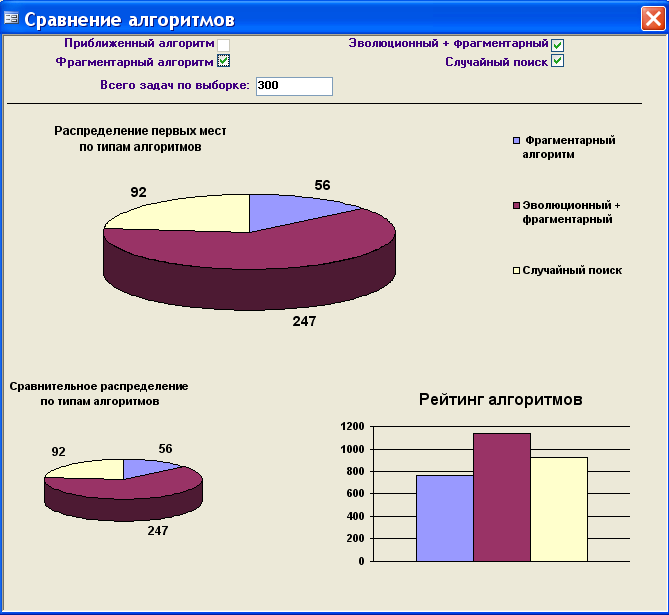


Рис. 3.6 – Вікно діаграм порівняння якості роботи різних алгоритмів

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-те два бали, за 4-те 1 бал та за п'яте місце порівняно – 0 балів.

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