**

***Міністерство освіти і науки України***

*Національний технічний університет України*

*«Київський політехнічний інститут імені Ігоря Сікорського»*

***Кафедра АСОІУ***

***Спеціальність 126 Інформаційні системи та технології***

**КУРСОВА РОБОТА**

**з дисципліни**

**“Додаткові розділи дослідження операцій”**

**на тему**

**“**Двокритеріальна задача планування діяльності**”**

|  |  |  |
| --- | --- | --- |
| *Виконали*  студент групи ІС-81  Салабай К.Я.  N зал. кн. ІС-8127  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис)  студент групи ІС-81  Чорний М.М.  N зал. кн. ІС-8129  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оцінка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оцінка) | *Прийняли*  Доц. каф. АСОІУ, к.т.н.  ЖДАНОВА О.Г.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис)  Доц. каф. АСОІУ, к.т.н.  ПОПЕНКО В.Д.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис)  Доц. каф. АСОІУ, к.т.н.  СПЕРКАЧ М.О.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) |

**Київ-2021**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Національний технічний університет України*  *«Київський політехнічний інститут імені Ігоря Сікорського»* | | | | | | | | | | | | |
| (назва вищого навчального закладу) | | | | | | | | | | | | |
| Кафедра | АСОІУ | | | | | | | | | | | |
| Дисципліна | | Дослідження операцій | | | | | | | | | | |
| Спеціальність | | | | ***126 Інформаційні системи та технології*** | | | | | | | | |
| Курс | | | | | 3 | | Група | | ІС-81 | | Семестр | 6 |
| **Завдання**  НА КУРСОВУ РоБОТУ СТУДЕНТУ | | | | | | | | | | | | |
| ***Салабаю Костянтину Ярославовичу***  ***Чорному Максиму Миколайовичу*** | | | | | | | | | | | | |
| 1. Тема роботи | | | Двокритеріальна задача планування діяльності | | | | | | | | | |
|  | | | | | | | | | | | | |
| 2. Термін здачі студентом закінченої роботи | | | | | | | | *30 травня 2021 року* | | | | |
| 3. Завдання курсової роботи | | | | | |  | | | | | | |
| *Вид робіт* | | | | | | | | | | *Виконавець* | | |
| *Побудувати математичну модель задачі.* | | | | | | | | | | Салабай К. Чорний М. | | |  |
| *Виконати аналіз можливих методів розв’язання* | | | | | | | | | | Салабай К. Чорний М. | | |
| *Розробити жадібний алгоритм розв’язання задачі.* | | | | | | | | | | Салабай К. Чорний М. | | |
| *Розробити генетичний алгоритм розв’язання задачі* | | | | | | | | | | Салабай К. | | |
| *Виконати програмну генетичного алгоритм.* | | | | | | | | | | Салабай К. | | |
| *Розробити бджолиний алгоритм розв’язання задачі* | | | | | | | | | | Чорний М. | | |
| *Виконати програмну бджолиного алгоритму.* | | | | | | | | | | Чорний М. | | |
| *Програмно реалізувати генератор індивідуальних задач* | | | | | | | | | | Салабай К. Чорний М. | | |
| *Експериментально дослідити розроблені алгоритми, виконати їх порівняльний аналіз* | | | | | | | | | | Салабай К. Чорний М. | | |
| *Визначити складність розроблених алгоритмів (теоретичну та експериментальну* | | | | | | | | | | Салабай К. Чорний М. | | |
| 4. Зміст розрахунково-пояснювальної записки | | | | | | | | | | | | |
| *Змістовні постановки задач. Математична постановка задачі.* | | | | | | | | | | | | |
| *Аналіз можливих методів розв’язання. Опис розроблених алгоритмів.* | | | | | | | | | | | | |
| *Опис програмного продукту. Результати експериментального дослідження алгоритмів.* | | | | | | | | | | | | |
| *Оцінка складності розроблених алгоритмів. Висновки* | | | | | | | | | | | | |
| 5. Перелік графічного матеріалу (с точним зазначенням обов’язкових креслень) | | | | | | | | | | | | |
|  | | | | | | | | | | | | |

|  |  |
| --- | --- |
|  | |
| 6. Дата видачі завдання | *10 лютого 2021 року* |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Назва етапів виконання курсової роботи** | | | **Строк виконання етапів роботи** | **Примітка** |
| 1 | *Опис постановки задачі* | | | *06.03* |  |
| 2 | *Пошук літературних джерел за темою курсової роботи. Порівняльний аналіз існуючих методів розв’язання задачі* | | | *15.03* |  |
| 3 | *Вивчення теоретичних положень жадібних алгоритмів, методу генетичного алгоритму, методу бджолиного алгоритму* | | | *31.03* |  |
| 4 | *Розробка жадібного алгоритму, генетичного алгоритму та бджолиного алгоритму поставленої задачі* | | | *10.04* |  |
| 5 | *Розв’язання задач “вручну”* | | | *15.04* |  |
| 6 | *Консультація в керівника – погодження інтерфейсу програми* | | | *15.04* |  |
| 7 | *Програмна реалізація алгоритмів та генератору індивідуальних задач. Відлагодження програм.* | | | *10.05* |  |
| 8 | *Експериментальне дослідження розроблених алгоритмів, їх порівняльний аналіз. Оцінка складності алгоритмів розв’язання задачі* | | | *15.05* |  |
| 9 | *Оформлення пояснювальної записки* | | | *20.05* |  |
| 10 | *Захист курсової роботи* | | | *31.05* |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  | | | | | |
| Студенти | |  | *Салабай К.Я.* | | |
|  | | (підпис) | (прізвище, ініціали) | | |
|  | |  | *Чорний М.М.* | | |

|  |  |  |
| --- | --- | --- |
|  | (підпис) | (прізвище, ініціали) |
| Керівник |  | *Жданова О.Г.* |
|  | (підпис) | (прізвище, ініціали) |
| «*20*» *лютого 2021 року* | | |

# 1 ПОСТАНОВКА ЗАДАЧІ

## Змістовна постановка задачі

**Деяке** підприємство здатне здійснювати видів діяльності. Для -го виду діяльності відомі:

* необхідні для здійснення кошти в обсязі , , ;
* прибуток , ;
* вплив на екологію, для цього використовується величина , відповідай вкрай негативному впливу, – діаметрально протилежному (позитивному), .

Усього на запуск усіх видів діяльності виділено одиниць вартості.

**Визначити** такий набір видів діяльності, за якого досягає:

* **максимуму** рентабельність (відношення сумарного прибутку на сумарні витрати);
* **максимуму** екологічність діяльності підприємства в цілому (сума величин ). циклу

## Постановка індивідуальних задач

### Постановка індивідуальної задачі 1

Підприємство SALABAYKOSTYA здатне будувати 5 типів будинків. Для побудови j –го типу будинка відомі :

* необхідні для будівництва кошти в обсязі , ; (див. табл. 1.)
* прибуток , ; (див. табл. 1.)
* вплив будівництва на екологію (див. табл. 1.)

Усього на будівництво виділено 200 одиниць вартості.

**Визначити** такий набір типів будинків, за якого досягає:

* **максимуму** рентабельність (відношення сумарного прибутку на сумарні витрати);
* **максимуму** екологічність діяльності підприємства (сума величин ).

***Таблиця 1 –*** *дані для побудови будинків.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип будинку** | **Необхідні кошти (од.варт.)** | **прибуток (од.варт.)** | вплив на екологію |
| Одноповерхові | 50 | 60 | +0.1 |
| Двоповерхові | 50 | 75 | -0.25 |
| Триповерхові | 90 | 135 | +0.2 |
| Мансардні | 70 | 90 | +0.6 |
| Блоковані | 30 | 75 | -0.5 |

#### Перший допустимий розв’язок

***Таблиця 2 -*** *Перший набір типів будинків.*

|  |
| --- |
| Одноповерхові |
| Двоповерхові |
| Мансардні |
| Блоковані |

Витрати на будівництво

200 од. варт 200 од.варт.

Рентабельність:

L1= = = 1.5

Екологічність:

L2= 0.1 – 0.25 + 0.6 – 0.5 = -0.05

#### Другий допустимий розв’язок

***Таблиця 3 -*** *Другий набір типів будинків.*

|  |
| --- |
| Одноповерхові |
| Двоповерхові |
| Триповерхові |

Витрати на будівництво

190 од. варт 200 од.варт.

Рентабельність:

L1= = 1.42

Екологічність:

L2= 0.1 – 0.25 + 0.2 = 0.05

Другий допустимий розв’язок є кращим, оскільки рентабельність знизилась незначно, а екологічність стала набагато кращою в порівнянні з першим допустимим розв’язком.

### Постановка індивідуальної задачі 2

Підприємство здатне виготовляти 5 типів пакетів. Для виготовлення j –го типу пакету відомі :

* необхідні для виготовлення пакету кошти в обсязі , ; (див. табл. 1.)
* прибуток , ; (див. табл. 1.)
* вплив виготовлення на екологію (див. табл. 1.)

## Усього на виготовлення 105 од. в.

**Визначити** такий набір видів пакетів, за якого досягає:

* **максимуму** рентабельність (відношення сумарного прибутку на сумарні витрати);
* **максимуму** екологічність діяльності підприємства (сума величин ).

***Таблиця 4 –*** *дані для виготовлення пакетів.*

|  |  |  |  |
| --- | --- | --- | --- |
| Тип пакету | Необхідні кошти (од.варт.) | прибуток (од.варт.) | вплив на екологію |
| Поліетиленовий низького тиску | 15 | 28 | -0.3 |
| Поліетиленовий високого тиску | 20 | 30 | -0.2 |
| Пеперматч | 30 | 40 | -0.1 |
| Крафт | 40 | 45 | +0.2 |
| Мелована бумага | 50 | 55 | +0.3 |

**Перший допустимий розв’язок:**

* поліетиленовий високого тиску;
* пеперматч;
* мелована бумага;

**Сумарні витрати** 100 105 од. в.

**Рентабельність:**



**Екологічність:**



**Другий допустимий розв’язок:**

* поліетиленовий низького тиску;
* крафт;
* мелована бумага;

**Сумарні витрати** 105 105 од. в.

**Рентабельність:**



**Екологічність:**



## Математична постановка задачі

**Змінні**

xj - чи входить вид діяльності в набір.

**Обмеження**

1.  (0 - якщо діяльність не входить в набір, 1 - якщо діяльність входить в набір)
2. Обмеження на кошти виділені на запуск усіх видів діяльності:



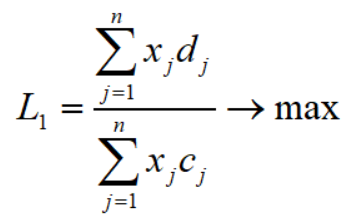
1. Кількість робіт повинна бути не менше 0 і не більше n:



**Цільова функція**

Необхідно визначити такий набір видів діяльності, за якого досягає:

* максимуму рентабельність (відношення сумарного прибутку на сумарні витрати);
* максимуму екологічність діяльності підприємства в цілому.





# ОПИС МЕТОДІВ РОЗВ’ЯЗАННЯ ЗАДАЧІ

Оскільки в досліджуваній задачі цільова функція залежить від декількох параметрів – екологічності та рентабельності, то задача відноситься до класу задач дискретної багатомірної оптимізації, а саме – двокритеріальної.

## Наявні методи розв'язання

Існує декілька методів розв’язання задач багатомірної оптимізації, а саме:

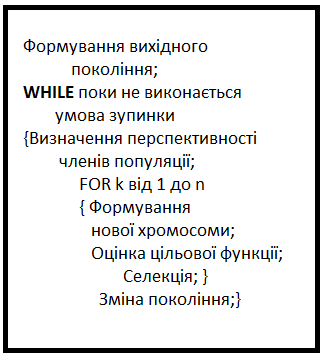
* Еволюційні методи;
* Інтерактивність;
* Метод дослідження простору параметрів.

### Еволюційні методи

Еволюційні методи відносяться до числа ефективних наближених (евристичних) методів для вирішення завдань оптимізації[1]. Вони засновані на використанні принципів оптимального пристосування організмів в живій природі до умов навколишнього середовища. До числа еволюційних відносяться генетичні методи, мурашиних колоній, поведінки натовпу, бджолиний на інші. Найбільш розвинені і затребувані в даний час генетичні алгоритми.

Початок застосування генетичних алгоритмів для вирішення задач оптимізації поклав Д.Холланд в 1975 р [3]. Фундаментальний внесок у розвиток GA вніс Д.Гольдберг [4]. Надалі генетичні алгоритми успішно застосовувалися для різних завдань синтезу конструкцій, складання розкладів, маршрутизації транспортних засобів, компонування обладнання, розкрою матеріалів і ін.

Для застосування еволюційних методів до вирішення конкретного завдання потрібно, по-перше, сформулювати безліч керованих параметрів X, по-друге, розробити модель програми у вигляді алгоритму обчислення цільової функції F (X), по-третє, розробити алгоритмічну реалізацію еволюційного методу.



*Рис.1 Загальна схема обчислень на основі еволюційних методів.*

Генетичні алгоритми засновані на пошуку кращих розв’язків за допомогою наслідування і посилення корисних властивостей об'єктів в процесі імітації їх еволюції.

Метод колонії мурах заснований на імітації поведінки мурах, які мінімізують довжину своїх маршрутів на шляху від мурашиної купи до джерела їжі.

### Інтерактивність

Часто рішення задачі багатокритеріальної оптимізації відбувається за участю експерта - людини, який вибирає і приймає рішення на основі інформації, представленої системою підтримки прийняття рішень. Можлива участь групи з кількох експертів. У разі участі людини в пошуку рішення алгоритми і методи називають інтерактивними[10].

Інтерактивні методи складаються з сукупності ітерацій, кожна з яких включає в себе етап аналізу, що виконується експертом, і етап розрахунку, що виконується багатокритеріальною оптимізаційною системою. За характером інформації, одержуваної багатокритеріальною оптимізаційною системою від експерта на етапі аналізу, можна виділити класи інтерактивних методів, в яких експерт:

* Безпосередньо призначає вагові коефіцієнти приватних критеріїв оптимальності;
* Накладає обмеження на значення приватних критеріїв оптимальності;
* Виконує оцінку пропонованих багатокритеріальною оптимізаційною системою альтернатив.

Основою для розвитку сучасних інтерактивних методів вирішення МКО-задач став метод Дайера-Джіоффріона, метод SIGMOP (SequentialInformationGeneratorforMultipleObjectiveProblems), метод Зайонца-Валленіуса [11, 12].

Одним із сучасних інтерактивних методів багатокритеріальної оптимізації, який відноситься до класу методів із застосуванням інформації від ОПР про бажаних рівнях критеріїв, є метод NIMBUS (NondifferentiableInteractiveMultiobjectiveBundle-basedoptimizationSystem). Цей метод розроблений в Гельсінському університеті (місто Гельсінкі, Фінляндія) під керівництвом професора К. Мієттінена. [13].

### Метод дослідження простору параметрів

Метод заснований на побудові допустимого і Парето-оптимального множин рішень. Дозволяє вирішувати завдання проектування, ідентифікації.

Оптимальність по Парето - такий стан системи, при якому значення кожного окремого критерію, що описує стан системи, не може бути покращено без погіршення становища інших елементів[8].

Принцип, за словами самого Парето говорить так: «Будь-яка зміна, яке не приносить збитків, а яка деяким людям приносить користь (за їх власною оцінкою), є поліпшенням». Таким чином визнається право на будь-які зміни, які не приносять нікому додаткової шкоди.

Метод Дослідження Простору Параметрів набув широкого застосування в задачах проектування технічних систем, які є завданнями з багатьма обмеженнями і мінімізується або максимізується критеріями. Дається опис моделей розрахунку, методів обчислення значень функцій і критеріїв, а також неформальна постановка задач багатокритеріальної оптимізації. Одне з головних призначень методу Досліджень Простору Параметрів (метод ДПП), який набув широкого застосування в різних сферах людської діяльності, полягає в тому, що з його допомогою будується множина допустимих розв’язків. Застосування методу ДПП для оптимізації об'єктів дозволяє коректно будувати багато геометричних форм оптимізованих конструкцій. В результаті визначається до допустимих безліч D. Дослідження оптимізованих конструкції істотно збільшує можливість отримання найкращої технології процесу виробництва, включаючи вибір способу обробки[9].

Отримання альбому представлень слід вважати однією з найважливіших особливостей розглянутого класу задач. Він допомагає фахівцям знайти раніше невідоме різноманіття геометричних форм оптимізованої конструкції, а значить, можна говорити про великі можливості, які відкриваються перед фахівцями, для пошуку принципово нових рішень.

### Порівняльна характеристика методів

Одним з недоліків генетичного алгоритму є відсутність гарантії виявлення глобального рішення для прийнятного часу.

Також генетичний алгоритм не гарантує, що знайдене рішення буде оптимальним. Тим не менш, він застосовується для пошуку "досить хорошого" рішення завдань.

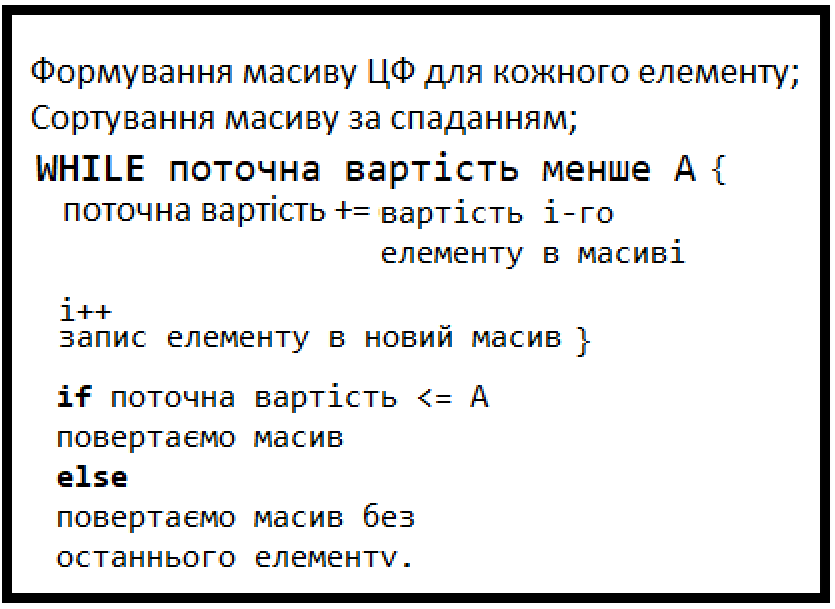
“Досить короткий час”. Однак даний алгоритм має переваги перед іншими алгоритмами при дуже великих розмірах завдань та відсутністю упорядкованості у вихідних даних, коли альтернативний спосіб є методом повного перебору варіантів. Головною перевагою генетичного алгоритму є те, що він може застосовуватись для вирішення складних неформалізованих завдань, для яких не розроблено спеціальних методів.

## Розробка жадібного алгоритму розв’язання задачі

Жадібні алгоритми - це ціле сімейство алгоритмів, тому немає якогось конкретного жадібного алгоритму, який підходив би до кожної окремої задачі оптимізації. Проте, всі жадібні алгоритми побудовані згідно одного принципу - обирати найкращий розв’язок на поточному кроці, не зважаючи на кроки, які були зроблені до або будуть зроблені після. Іншими словами, жадібний алгоритм робить локально оптимальний вибір у сподіванні, що він приведе до глобально оптимального розв'язку, проте так стається далеко не завжди.

В нашому випадку кожен вид діяльності буде розглядатись окремо. В набір буде входити та діяльність, цільова функція якої на поточному кроці є найкращою.

Для цього значення величини впливу діяльності на екологію нормується з до .



*Рис.2 Схема жадібного алгоритму.*

## Опис Генетичного алгоритму

Генетичний алгоритм - це в першу чергу еволюційний алгоритм, іншими словами, основна фішка алгоритму - схрещування (комбінування). Як нескладно здогадатися ідея алгоритму нахабним чином взята у природи. Так ось, шляхом перебору і найголовніше відбору виходить правильна «комбінація». Алгоритм ділиться на три етапи: схрещування, селекція, формування нового покоління. Якщо результат нас не влаштовує, ці кроки повторюються до тих пір, поки результат нас не почне задовольняти або станеться одна з нижче перерахованих умов:

* Значення цільової функції не буде змінюватись 30 циклів.

На кроці створення нової популяції формується початкова популяція, яка, цілком можливо, виявиться не кошерною, проте велика ймовірність, що алгоритм цю проблему виправить. Головне, щоб вони відповідали «формату» і були «пристосовані до розмноження». Щодо розмноження, ну тут все як у людей, для отримання нащадка потрібно два батька. Головне, щоб нащадок (дитина) міг успадкувати від батьків їх риси. При цьому розмножуються всі, а не тільки ті, що вижили (ця фраза особливо абсурдна, але так як у нас все в сферичному вакуумі, то можна все), в іншому випадку виділиться один альфа самець, гени якого перекриють всіх інших, а нам це принципово не підходить, адже ми можемо попасти в локальний оптимум а не глобальний. Щодо мутації, вони схожі з розмноженням, з мутантів вибирають певну кількість особин і змінюють їх відповідно до заздалегідь визначеними операціями. Відбір. Тут починається найсолодше, ми починаємо вибирати з популяції частку тих, хто «піде далі». При цьому частка «вижили» після нашого відбору ми визначаємо заздалегідь руками, вказуючи у вигляді параметра. Як не сумно, інші особини повинні загинути.

**Основні поняття**

**Ген –** структурна і функціональна одиниця спадковості, що контролює розвиток певної ознаки чи властивості[5].

**Хромосома** – сукупність генів, яка характеризує особину[5].

**Популяція** – сукупність усіх особин[5].

**Локус** – позиція у хромосомі[5].

**Алель** – можливі значення гена(сукупності генів, що йдуть поспіль) [5].

### Розробка алгоритму розв’язання задачі

#### Оператори генетичного алгоритму

Для розробки алгоритму слід визначити основні оператори генетичного алгоритму**.**

##### Оператор створення популяції

Популяція буде складати 7 особин. Кожна особина генерується випадковим чином. Особина, яка перевищує сумарну вагу - відкидається, а замість неї знову випадковим чином генерується нова особина.

##### Оператор вибору батьків

В якості оператору вибору батьків був обраний ,,*Турнірний відбір*”. Усі особини популяції випадковим чином діляться на дві групи і з кожної групи вибирається найкраща особина, яка і буде відігравати роль батька. Мінусом даного оператору вибору батьків є те, що якщо трапляється дуже гарна особина, то вона буде завжди вибиратися в ролі батька і це призведе до того, що ми будем попадати в локальний оптимум.

##### Оператор створення нащадків

У якості оператору створення нащадків був обраний універсальний оператор кросинговеру. Ідея полягає в визначенні двійкової маски(штучного батька), довжина якої дорівнює довжині хромосоми. Отримання нащадків виконується на основі бінарної операції “додавання по модулю 2” над генами батьків та маски.

##### Оператор реанімації недопустимих розв’язків

У разі недопустимого нащадку, відбувається зміна значення випадкового гену поки розв’язок стане допустимим.

##### Оператор мутації

Після створення нащадків, одному з них буде притаманна точкова мутація. Випадковим чином вибирається ген та інвертується.

Значення величини впливу діяльності на екологію нормується з до .

### Алгоритм розв’язання задачі

1. *Використовуючи оператор створення популяції, формуємо початкову популяцію.*
2. *Поки кількість ітерацій менше тридцяти:*
   1. *Використовуючи оператор вибору батьків, обираємо батьків.*
   2. *Використовуючи оператор створення нащадків, створюємо двох нащадків.*
   3. *Перевіряємо допустимість вибраних нащадків за початковими обмеженнями.*
      1. *Якщо нащадок містить сумарну вагу більше А, то використовуємо оператор реанімації.*
   4. *Використовуємо оператор мутації, в ітераціях в яких остача від ділення на 2 буде 0.*
   5. *Оновлюємо популяцію:*
      1. *Якщо нащадок кращий від найгіршої особини в поточній популяції та його гени не співпадають з жодною особиною із популяції, робимо заміну.*
      2. *Інакше відкидаємо отриманого нащадка.*

## Опис бджолиного алгоритму

Бджолиний алгоритм один з поліноміальних евристичних алгоритмів для розв‘язку оптимізаційних задач в області інформатики та дослідження операцій та заснований на імітації поведінки колонії медоносних бджіл при зборі нектару в природі [14]. Бджолиний алгоритм відноситься до ройового інтелекту та описує колективну поведінку децентралізованої системи, яка здатна до самоорганізації. Термін був уведений Херардо Бені й Ван Цизмом у 1989 році, у контексті системи клітинних роботів[14]. Системи ройового інтелекту складаються з множини агентів. Самі агенти зазвичай досить прості, але всі разом, взаємодіючи, вони створюють, так званий, ройовий інтелект. До такого ройового інтелекту і відноситься бджолиний алгоритм. Ідея бджолиного алгоритму полягає в тому, що усі бджоли на кожному кроці будуть вибирати як найкращі локальні ділянки для дослідження, так і ділянки в околиці найкращих, що дозволить:

* урізноманітнити популяцію пошуків на наступних ітераціях;
* збільшити ймовірність виявлення розв‘язку близького до оптимального.

### Розробка алгоритму розв’язання задачі

Значення величини впливу діяльності на екологію нормується з до . , де j = (1…n).

Параметри алгоритму:

= 7 - кількість бджіл розвідників.

= 3 - кількість найкращих квіткових ділянок = 3.

=10 - кількість бджіл-фуражирів =10.

- кількість фуражирів, відправлених на ділянку j. Визначається пропорційно в залежності від значення ЦФ за формулою:

);

r - розмір квіткової ділянки(розмір околу) = 1 (конвертуються 1 ген).

**Алгоритм**

1. генеруємо бджіл розвідників та розраховуємо для них значення ЦФ.
2. вибираємо найкращих розв’язків,
3. Розсилаємо бджіл-фуражирів в кількості на обрані ділянки та розраховуємо для них значення ЦФ.
4. Серед всіх фуражирів обираємо того хто знайшов найкращий розв’язок і якщо цей розв’язок кращий за розв’язок бджоли розвідника

то розвідник переміщається на місце найкращого фуражира.

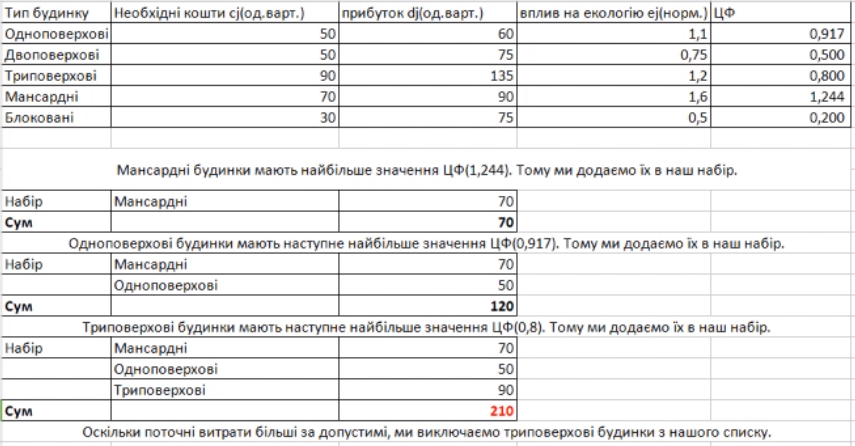
1. Додаємо - бджіл розвідників на пошук нових ділянок.
2. Переходимо до кроку 1.
3. Критерій зупинки(розв’язок не покращується 30 ітерацій).

Процес пошуку фуражирів для кожного з розвідників програмно буде розподілений на декілька паралельних процесів, за рахунок чого значно збільшиться його швидкість.

## Приклади застосування розроблених алгоритмів

### Приклад розв'язання жадібним алгоритмом

Демонстрація використання жадібного алгоритму буде виконуватись на прикладі індивідуальної задачі 1(див. розділ 1.2.1).



Відповідь: значення цільової функції = 2.161.

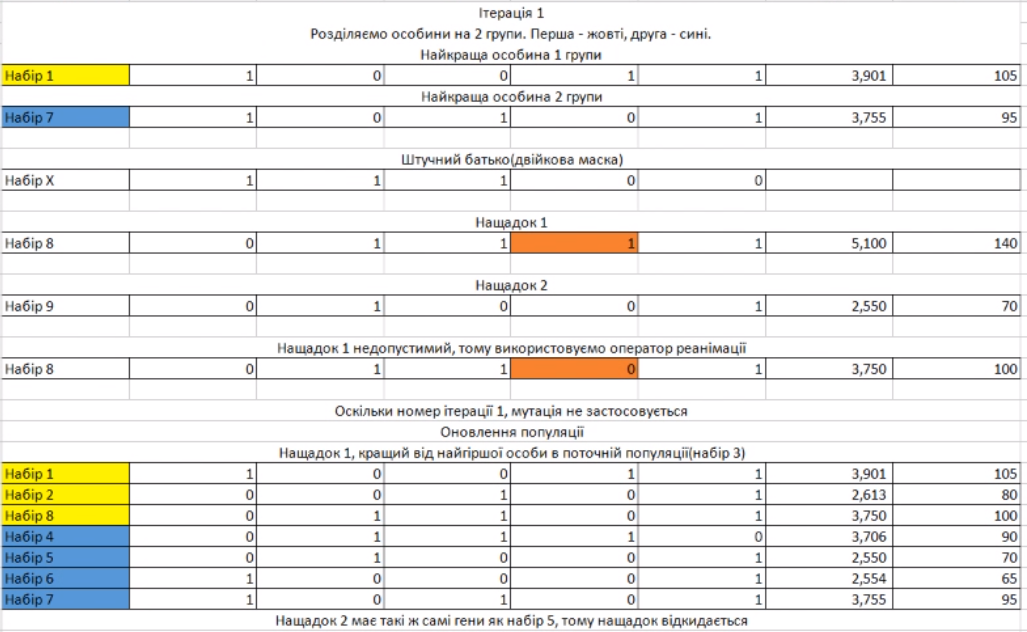
В набір входять такі будинки:

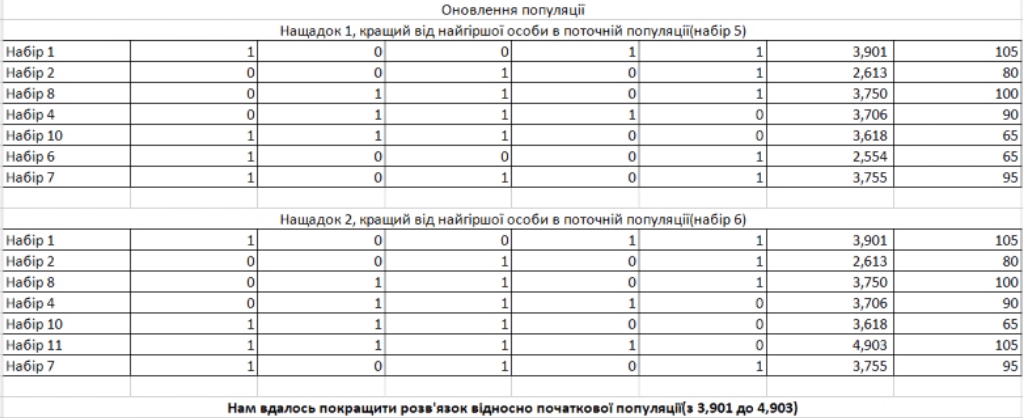
* мансардні;
* одноповерхові.

### Приклад розв'язання генетичним алгоритмом

Демонстрація використання генетичного алгоритму буде виконуватись на прикладі індивідуальної задачі 2(див. розділ 1.2.2)







Відповідь:

Поточний рекордний розв’язок(значення ЦФ 4,903)

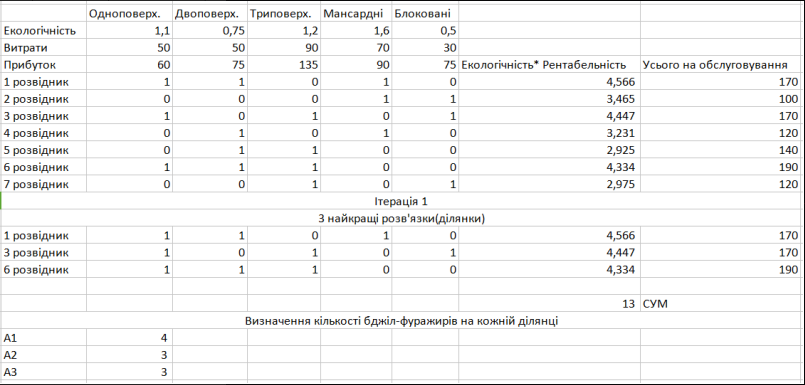


На поточній ітерації оптимальним набором пакетів є:

* поліетиленовий низького тиску;
* поліетиленовий високого тиску;
* пеперматч;
* крафт.

### Приклад розв'язання бджолиним алгоритмом

Демонстрація використання бджолиного алгоритму буде виконуватись на прикладі індивідуальної задачі 1(див. розділ 1.2.1).





На поточній ітерації вдалось покращити розв’язок з (4,566 до 5,925).

Поточний рекордний розв’язок(значення ЦФ = 5,925)



Відповідь: На поточній ітерації оптимальним набором є:

* одноповерхові;
* двоповерхові;
* мансардні;
* блоковані.

# Планування експериментів

## Класифікація індивідуальних задач

***Умовно-постійні параметри задач.***

n - кількість типів продукції,

- Масив значень прибутків від виготовлення j-го типу продукції.

- Масив значень впливу виготовлення продукції на екологію j-го типу;

- Масив значень необхідних для виготовлення коштів,

A – виділені кошти.

j = 1,n;

Класифікація буде відбуватись за такими значеннями параметрів:

а) розмірність n {10,20, …, 100};

б) параметри розподілу :

* {30;100};
* {0.1; 0.5}; – дисперсія.

в) параметри розподілу :

* [0.1 , 1.9];
* [0.6 , 1.3];

г) параметри розподілу А:

* [30; 50];
* {0.5 ; 0.7 };
* {0.1A; 0.25A} – дисперсія.

## Параметри алгоритмів

### Параметри генетичного алгоритму

##### Спосіб створення популяції

Популяція буде складати N особин. Будуть розглядатися такі значення N = 7, 20, 50;

##### Спосіб вибору батьків

В якості оператору вибору батьків будуть розглянуті:

* турнірний відбір;
* методрулетки (випадковий + випадковий).

### Параметри Бджолиного алгоритму

Параметри алгоритму:

= 7 - кількість бджіл розвідників.

= 3 - кількість найкращих квіткових ділянок = 3.

=10 - кількість бджіл-фуражирів =10.

- кількість фуражирів, відправлених на ділянку j. Визначається пропорційно в залежності від значення ЦФ за формулою:

, де j = (1,n);

r - розмір квіткової ділянки(розмір околу) = 1 (конвертуються 1 ген).

## План експериментів

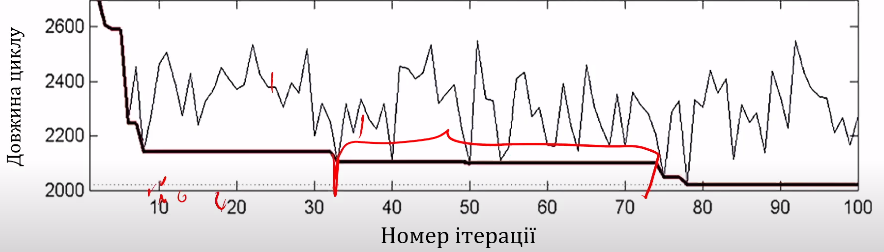
Далі наведені плани експериментів для оптимізації роботи алгоритмів.

### План експериментів для генетичного алгоритму

#### Експеримент №1

*Мета:* Дослідження динаміки зміни поточного рекорду

Результатом експерименту буде графік змін поточного рекорду. Наразі якщо значення цільової функції не буде змінюватись впродовж g циклів то алгоритм завершує роботу. Проаналізувавши графік, цей параметр можна буде змінити. Наприклад може вийти ситуація, коли поточний рекорд зміниться на ітерації g>30(див. рис3). В такому випадку точність алгоритму погіршуєтсья. Таким чином ми оптимізуємо роботу нашого алгоритму.



*Рис.3 Приклад графіку зміни поточного рекорду.*

#### Експеримент №2

*Мета*: Дослідження впливу значення параметру вибору батьків на результати роботи алгоритму.

В процесі експерименту будуть протестовані такі оператори вибору батьків як:

* турнірний відбір;
* метод рулетки (випадковий + випадковий).

– ймовірність вибору *і*-ї особини( – значення ЦФ *і*-ї особини);

N = 10 – кількість прогонів алгоритму для кожного оператору вибору батьків

Для задач з розмірністю n {5, 20, 60, 100} запустити генетичний алгоритм N разів для кожного оператору вибору батьків. Зберегти до файлу час роботи алгоритму та значення рекорду для кожного з операторів.

Час роботи алгоритму

t(rez)=, де - час роботи алгоритму *j*-го прогону.

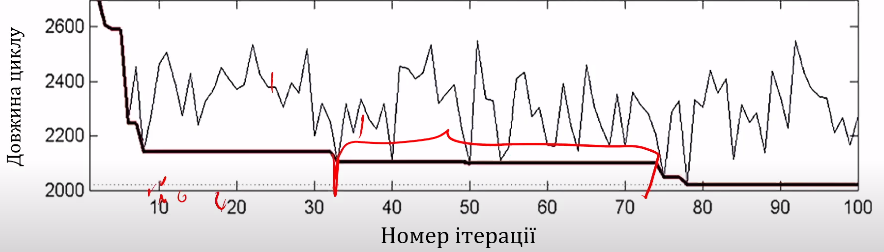
Результати будуть наведені в табличці. Проаналізувавши їх, ми виберемо оптимальний оператор вибору батьків.

### План експериментів для бджолиного алгоритму

#### Експеримент №1

*Мета:* Дослідження динаміки зміни поточного рекорду

Результатом експерименту буде графік змін поточного рекорду. Наразі якщо значення цільової функції не буде змінюватись впродовж g циклів то алгоритм завершує роботу. Проаналізувавши графік, цей параметр можна буде змінити. Наприклад може вийти ситуація, коли поточний рекорд зміниться на ітерації g>30(див. рис4). В такому випадку точність алгоритму погіршуєтсья. Таким чином ми оптимізуємо роботу нашого алгоритму.



*Рис.4 Приклад графіку зміни поточного рекорду.*

#### Експеримент №2

*Мета*: Дослідження впливу значень параметрів на результати роботи алгоритму.

- кількість бджіл розвідників.

- кількість найкращих квіткових ділянок.

- кількість бджіл-фуражирів =10.

N = 10 – кількість прогонів алгоритму для кожного набору параметрів.

1. Згенерувати індивідуальну задачу з 50 видами діяльностей.
2. **for** : = 5 to 10

**for** : = 2 to 5

**for** : = 2 to 5

**Calculate () –** зберігаємо значення рекорду та час роботи алгоритму;

Результати експерименту будуть представлені у вигляді таблиці:

***Таблиця 5 -*** *Результати експерименту №2.*

|  |  |  |  |
| --- | --- | --- | --- |
| Параметри |  |  |  |
| Найкращий розв’язок |  |  |  |
| Час роботи алгоритму |  |  |  |

Таким чином ми зможемо визначити оптимальні значення для параметрів .

### План загальних експериментів

#### Експеримент №1

*Мета*: порівняння розроблених алгоритмів за часом та точністю

Результатом експерименту буде таблиця зі значеннями цільової функції та часом роботи алгоритмів. Для проведення експерименту будуть взяті задачі з розмірністю n {5,10, 20, 50, 100}.

***Таблиця 6 -*** *Результати експерименту №1.*

|  |  |  |  |
| --- | --- | --- | --- |
| n | Алгоритм | Час роботи алгоритму | Значення ЦФ найкращого розв’язку |
|  |  |  |  |

# ОПИС ПРОГРАМНОГО ПРОДУКТУ

## Опис функціональної моделі

Програмний продукт (ПП) містить наступні складові:

1. **робота з індивідуальною задачею** (ІЗ):
   1. введення вручну даних ІЗ;
   2. генерація (випадковим чином) даних ІЗ;
   3. збереження в файл/читання з файлу/редагування даних ІЗ;
   4. розв’язання ІЗ усіма розробленими методами (з виведенням на екран відповідних розв’язків та значень ЦФ);
2. **експериментальне дослідження розроблених алгоритмів:** 
   1. задання діапазонів зміни параметрів експериментів:
      * розмірність задач;
      * кількість ІЗ, яку необхідно згенерувати для кожної розмірності;
      * діапазоні зміні коефіцієнтів (ЦФ та обмежень);
   2. генерація множини ІЗ
   3. розв’язання множини згенерованих ІЗ усіма розробленими методами;

## Рішення з інформаційного забезпечення

### Вхідні дані

n - кількість типів продукції,

- Масив значень прибутків від виготовлення j-го типу продукції.

- Масив значень впливу виготовлення продукції на екологію j-го типу;

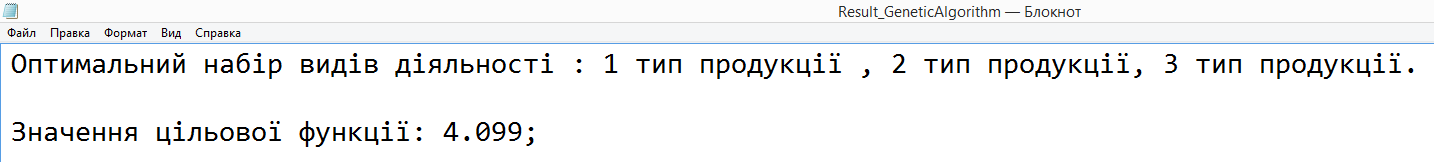
- Масив значень необхідних для виготовлення коштів,

A – виділені кошти.

j = 1,n;

### Вихідні дані

Результатом роботи програми є набір видів діяльності та значення цільової функції. Результати демонструються у консольному вікні застосунку та записуються до файлу **Result*\_****назва\_алгоритму***.txt**.



*Рис.5 Приклад текстового файлу з результатами роботи генетичного алгоритму.*

### Опис інформаційного забезпечення

Для зберігання інформації не буде реалізовано ніякої бази даних. Вся інформація буде зберігатись в оперативній пам’яті в класах List <T> з простору імен *System.Collections.Generic*. Цей клас представляє найпростіший список однотипних об'єктів.

***Таблиця 7 -*** *опис списків класу GeneticAlgorithm****.***

|  |  |
| --- | --- |
| ***Список*** | ***Семантика списку*** |
| **string[] production\_names** | масив назв видів діяльностей |
| **public List<float> Ecologies** | список впливу на екологію |
| **public List<int> Costs** | список коштів, необхідних для здійснення діяльностей |
| **public List<int> Profits** | список прибутків від здійснення діяльностей |
| **PublicList<List<int>>** **CurrentPopulation** | список хромосом у початковій популяції |
| **List<List<int>>BluePopulation** | список хромосом першої чистини популяії |
| **List<List<int>>YellowPopulation** | список хромосом другої чистини популяції |
| **public List<int> CurrentBlueParent** | список генів першого батька |
| **publicList<int> CurrentYellowParent** | список генів другого батька |
| **public List<int> CurrentMask** | список генів штучного батька |
| **public List<int> CurrentBlueChild** | список генів першого нащадка |
| **publicList<int> CurrentYellowChild** | список генів другого нащадка |

***Таблиця 8 -*** *опис списків класу BeesAlgorithm.*

|  |  |
| --- | --- |
| ***Список*** | ***Семантика списку*** |
| **public List<float> Ecologies** | список впливу на екологію |
| **public List<int> Costs** | список коштів, необхідних для здійснення діяльностей |
| **public List<int> Profits** | список прибутків від здійснення діяльностей |
| **public List<List<int>> CurrentPopulation** | список бджіл розвідників |
| **public List<List<int>> TheBestScouts** | список найкращих квіткових ділянок |
| **public List<List<int>> Foraging bees TheBestScouts** | Список бджіл фуражирів |

***Таблиця 9 -*** *опис списків класу GreedyAlgorithm****.***

|  |  |
| --- | --- |
| ***Список*** | ***Семантика списку*** |
| **public List<float> Ecologies** | список впливу на екологію |
| **public List<int> Costs** | список коштів, необхідних для здійснення діяльностей |
| **public List<int> Profits** | список прибутків від здійснення діяльностей |
| **public List<float> cf** | ***Список значень цільових функцій*** |

## Рішення з програмного забезпечення

### Опис обраних засобів розробки

Розробка програмного забезпечення здійснюється на платформі .NET CORE, мовою *C #* у вигляді звичайного консольного застосунку*. C # є* об'єктно-орієнтованою мовою програмування. Перша версія мови вийшла разом з релізом Microsoft Visual Studio .NET в лютому 2002 року. Поточною версією мови є версія C # 9.0, яка вийшла 10 листопада 2020 року разом з релізом .NET 5(*.NET Core*).

*.NET Core* - це модульна платформа для розробки програмного забезпечення з відкритим вихідним кодом. Сумісна з такими операційними системами як Windows, Linux і macOS. Була випущена компанією Microsoft.

Основні риси платформи *.NET:*

* *Підтримка декількох мов -* основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує декілька мов.
* *Кросплатформеність* - .NET 5 підтримується на більшості сучасних ОС Windows, MacOS, Linux.
* *Потужна бібліотека класів* - .NET представляє єдину для всіх підтримуваних мов бібліотеку класів.
* *Продуктивність* - Згідно ряду тестів веб-додатки на .NET 5 в ряді категорій сильно випереджають веб-додатки, побудовані за допомогою інших технологій.

**Технологія LINQ**

Взаємодія даних різних списків відбувається з використанням технології *LINQ*.

*LINQ* (Language-Integrated Query) представляє просту і зручну мову запитів до джерела даних. Як джерело даних може виступати об'єкт, який реалізує інтерфейс IEnumerable (наприклад, стандартні колекції, масиви), набір даних DataSet, документ XML. Але незалежно від типу джерела LINQ дозволяє застосувати до всіх один і той же підхід для вибірки даних[15].

Існує декілька різновидів *LINQ*. Ми будем використовувати *LINQ to Objects*, що застосовується для роботи з масивами і колекціями.



*Рис.6 Найпростіше визначення запиту LINQ*

### Архітектура програмного забезпечення

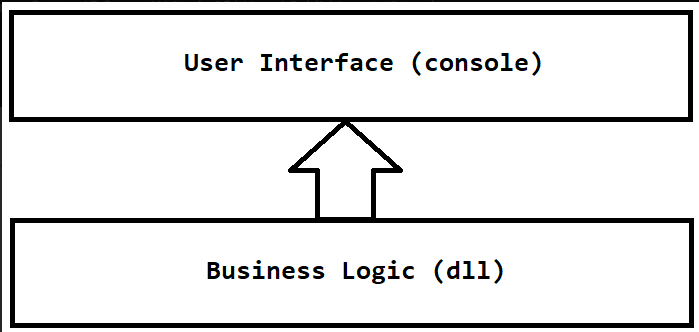
#### Схема архітектури ПП

Програмне забезпечення складається з 2 частин:

* Бібліотеки класів з реалізацією алгоритмів(.dll).
* Додаток з інтерфейсом(.exe).

Шар моделі (model) – класи **GeneticAlgorithm, BeesAlgorithm, GreedyAlgorithm**, які описують реалізації генетичного, бджолиного та жадібного алгоритму відповідно.

Шар уявлення (view) – клас **UI**, який відповідає за відображення інформації користувачеві, в графічному (GUI) або текстовому вигляді (консоль).



*Рис.7 Схема архітектури ПЗ.*



*Рис.8 Схема архітектури .net.*

#### Діаграма класів (об'єктів)

#### 

*Рис.9 Діаграма класів.*

#### Специфікація функцій

***Таблиця 10 -*** *специфікація функцій класу GeneticAlgorithm.*

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я функції | Призначення функції | Параметри | Семантика параметрів |
| *GetStartPopulation* | Функція формування початкової популяції | int | Кількість хромосом у популяції |
| *CreateChromosome* | Функція створення хромосоми. | int | Кількість видів діяльності |
| CreateMask | Функція створення штучного батька | - | - |
| GetChildren | Функція створення нащадків. | - | - |
| UpdatePopulation | Функція оновлення популяції | List<List<int>> | Список хромосом поточної популяції |
| Mutation | Функція мутації випадкового гену нащадка | - | - |
| Reverse | Функція інвертування гена | int | Ген хромосоми |
| CheckAdmissibility | Функція перевірки допустимості хромосоми | List<int> | Список генів хромосоми |
| CalculateChild | Функція створення нащадків | List<int>, List<int> | Список хромосом батька, список хромосом маски(штучного батька) |
| CalculateCF | Функція розрахунку значення ЦФ | List<int> | Список генів хромосоми |
| CalculateCost | Функція розрахунку значення витрат | List<int> | Список генів хромосоми |
| GetTheBestChromosome | Функція отримання найкращої хромосоми зі значенням ЦФ. | List<List<int>> | Список поточної популяції |
| GetParents | Функція створення батьків | - | - |

***Таблиця 11 -*** *специфікація функцій класу BeesAlgorithm.*

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я функції | Призначення функції | Параметри | Семантика параметрів |
| CalculateCf | Функція розрахунку значення ЦФ | List<int> | Список характеристик бджоли |
| GetScout | Функція генерації бджіл розвідників | - | - |
| CalculateCost | Функція розрахунку значення витрат | List<int> | Список характеристик бджоли |
| GetTheBestScouts | Функція пошуку найкращих розвідників | - | - |
| RunForagingBees | Функція пошуку найкращих ділянок серед фуражирів | - | - |
| UpdateScout | Функція оновлення списку бджіл розвідників | List<List<int>> | Список найкращих квіткових ділянок з попередньої ітерації |
| LocalUpdate | Функція локальної оптимізації | - | - |

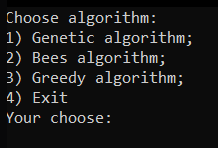
***Таблиця 12 -*** *специфікація функцій класу GreedyAlgorithm.*

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я функції | Призначення функції | Параметри | Семантика параметрів |
| Execute | Функція визначення набору діяльностей | - | - |
| CalculateCf | Функція розрахунку значення ЦФ | float, int, int | Екологічність, вартість, прибуток типу діяльності |

## Керівництво користувача

### Інструкція користувача

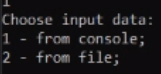
При запуску програми для користувача стане доступне меню. Користувач має змогу вибрати один з пунктів меню для виконання певного алгоритму.



*Рис.10 Головне меню користувача.*

Натиснувши пункт 1 меню - користувач матимете змогу застосувати генетичний алгоритм, пункт 2 – бджолиний. Пункт 3 – жадібний.

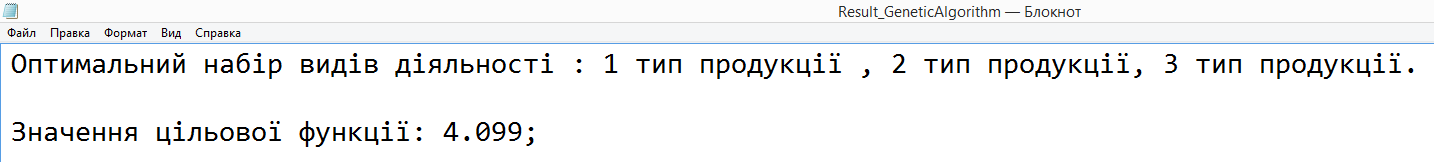
При виборі кожного пункту, вам буде запропоновано обрати спосіб введення вхідних даних(з консолі або з файлу) та ввестивхідні дані:



*Рис.11 Спосіб введення даних.*

* n - кількість типів продукції,
* - Масив значень прибутків від виготовлення j-го типу продукції.
* - Масив значень впливу виготовлення продукції на екологію j-го типу;
* - Масив значень необхідних для виготовлення коштів,
* A – виділені кошти.

Після цього програма зробить всі необхідні розрахунки. Результатом роботи програми є набір видів діяльності та значення цільової функції. Результати демонструються у консольному вікні застосунку та записуються до файлу **Result*\_****назва\_алгоритму***.txt**.



*Рис.12 Приклад текстового файлу з результатами роботи генетичного алгоритму.*

Після успішної роботи, програма повертає користувача у головне меню, де він може працювати з іншими алгоритмами або вийти із програми, обравши пункт 4 у головному меню.

У разі некоректно введених даних користувачем, програма повідомить про помилку та запропонує знову ввести дані.

### Методика випробувань

Для перевірки працездатності алгоритмів були написані Unit тести.

Модульне тестування (Unit testing) — це метод тестування програмного забезпечення, який полягає в окремому тестуванні кожного модуля коду програми. Модулем називають найменшу частину програми, яка може бути протестованою. У процедурному програмуванні модулем вважають окрему функцію або процедуру. В об'єктно-орієнтованому програмуванні — метод [18].

Для написання тестів був використаний спеціальний фреймворк xUnit.net. ***xUnit.net*** - фреймворк тестування для платформи .NET. Найбільш популярний фреймворк для роботи саме з .NET Core і ASP.NET Core.

Результати тестування порівнювались з результатами розв’язання індивідуальних задач в Excel(див. розділ 2.5). Також працездатність була перевірена експериментом ,,**Дослідження динаміки зміни поточного рекорду,,** (див. розділ 3.3) результатом якого є графік зі змінами поточного рекорду(див. *Рис.7*).

# **Експериментальне дослідження алгоритмів**

Для проведення досліджень був розроблений програмний продукт, який включає в себе модуль генерації індивідуальних задач згідно з їх класифікацією ( див. розділ 3.1 *Класифікація індивідуальних задач*).

## Експерименти для генетичного алгоритму

### Експеримент №1

*Мета:* Дослідження динаміки зміни поточного рекорду

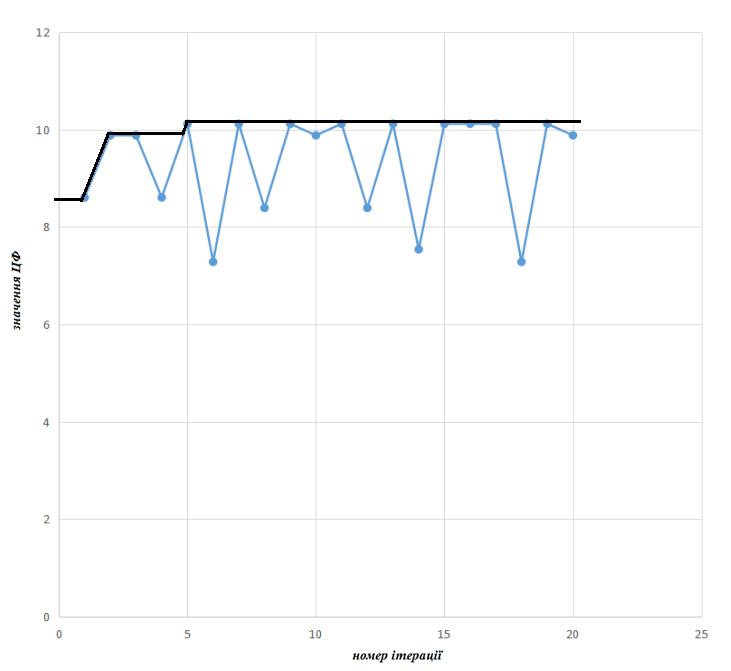
В даному експерименті буде досліджуватись параметр зупинки алгоритму g – кількість ітерацій за яких не змінюється рекордне значення цільової функції.

Будуть протестовані такі значення для параметру g {15, 30, 50, 100} на таких класах задач:

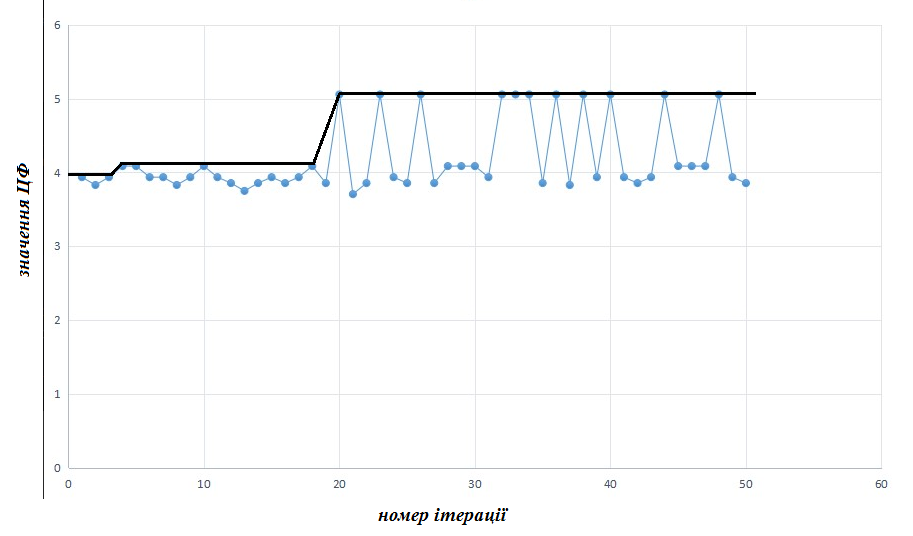
* 5/S,L/S,L/L,S/L,S;
* 30/L,S/S,L/L,S/S,L;
* 60/L,S/L,S/L,S/L,S.

***Таблиця 13 -*** *результати експерименту №1.*

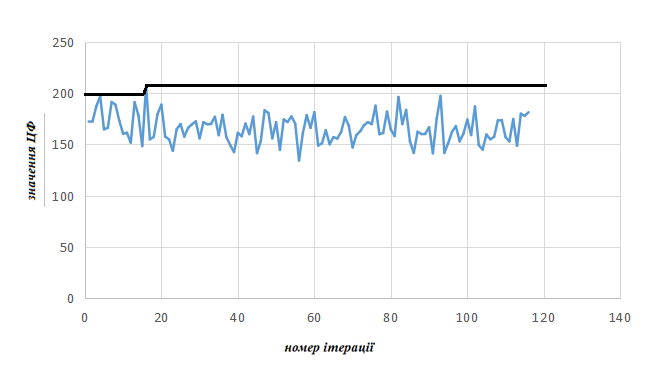
|  |  |  |  |
| --- | --- | --- | --- |
| n | g | Час роботи алгоритму(ms) | Значення ЦФ найкращого розв’язку |
| 5 | 15 | 34 | 10,117279 |
| 5 | 30 | 57 | 10,117279 |
| 5 | 50 | 69 | 10,117279 |
| 5 | 100 | 119 | 10,117279 |
| 20 | 15 | 92 | 54,72027 |
| 20 | 30 | 148 | 57,123844 |
| 20 | 50 | 380 | 61,526432 |
| 20 | 100 | 581 | 61,526432 |
| 60 | 15 | 95 | 192,9607 |
| 60 | 30 | 113 | 192,9607 |
| 60 | 50 | 268 | 195,11017 |
| 60 | 100 | 716 | 207,25603 |



*Рис.13 Графік зміни поточного рекорду генетичного алгоритму при g=15 n=5 .*



*Рис.14 Графік зміни поточного рекорду генетичного алгоритму при g=30, n=30.*



*Рис.15 Графік зміни поточного рекорду генетичного алгоритму при g=100, n=60.*

Таким чином, можна зробити висновки, що для задач з розмірністю менше 10 оптимальним значення параметру g є g = 15, тому що значення цільової функції перестає покращуватись. Для задач з розмірністю 10 < n < 60 оптимальним значення параметру g є g = 50. Для задач з розмірністю n > 60 оптимальним значення параметру g є g = 100.

### Експеримент №2

*Мета*: Дослідження впливу значення параметру вибору батьків на результати роботи алгоритму.

Були протестовані такі оператори вибору батьків як:

* турнірний відбір;
* метод рулетки (випадковий + випадковий).

Експерименти проводились для таких класів задач:

* 5/S,L/S,L/L,S/S,L;
* 20/L,S/S,L/L,S/L,S;
* 60/L,S/L,S/L,S/L,S;
* 100/S,L/L,S/L,S/L,S.

***Таблиця 14 -*** *результати експерименту №2*

|  |  |  |  |
| --- | --- | --- | --- |
| n | Оператор вибору батьків | Час роботи алгоритму | Значення рекорду |
| 5 | Турнірний відбір | 36 | 14,873852 |
| 5 | Метод рулетки | 28 | 14,873852 |
| 20 | Турнірний відбір | 215 | 57,92574 |
| 20 | Метод рулетки | 188 | 57,662865 |
| 60 | Турнірний відбір | 695 | 149,73677 |
| 60 | Метод рулетки | 750 | 146,44733 |
| 100 | Турнірний відбір | 781 | 261,83685 |
| 100 | Метод рулетки | 824 | 258,72955 |

Проаналізувавши дані таблиці №12, можна зробити висновки, що для задач з розмірністю n < 20, оптимальним вибором оператору створення батьків є метод рулетки, оскільки він дає такі ж результати як і оператор турнірного відбору, але за менший час. Щодо задач з розмірністю n > 20, оператор турнірного відбору знаходить кращий розв’язок за менший час.

## Експерименти для бджолиного алгоритму

### Експеримент №1

*Мета:* Дослідження динаміки зміни поточного рекорду

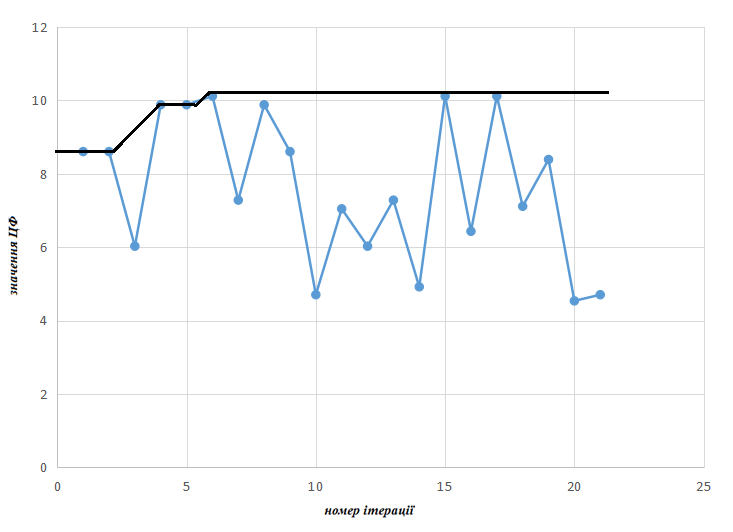
В даному експерименті буде досліджуватись параметр зупинки алгоритму g – кількість ітерацій за яких не змінюється рекордне значення цільової функції.

Будуть протестовані такі значення для параметру g {15, 30, 50, 100} на таких класах задач:

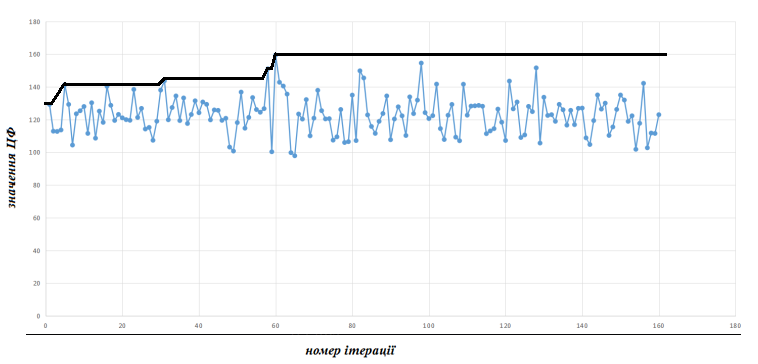
* 5/S,L/S,L/L,S/L,S;
* 20/L,S/S,L/L,S/S,L;
* 60/L,S/L,S/L,S/L,S.

***Таблиця 15 -*** *результати експерименту №1.*

|  |  |  |  |
| --- | --- | --- | --- |
| n | g | Час роботи алгоритму(ms) | Значення ЦФ найкращого розв’язку |
| 5 | 15 | 41 | 10,117279 |
| 5 | 30 | 99 | 10,117279 |
| 5 | 50 | 115 | 10,117279 |
| 5 | 100 | 141 | 10,117279 |
| 20 | 15 | 45 | 34,589245 |
| 20 | 30 | 166 | 38,456635 |
| 20 | 50 | 484 | 40,175697 |
| 20 | 100 | 498 | 40,912266 |
| 60 | 15 | 287 | 151,14546 |
| 60 | 30 | 307 | 154,26074 |
| 60 | 50 | 570 | 155,86809 |
| 60 | 100 | 1144 | 159,45049 |



*Рис.16 Графік зміни поточного рекорду генетичного алгоритму при g=15 n=5 .*



*Рис.17 Графік зміни поточного рекорду генетичного алгоритму при g=100, n=60.*

Таким чином, можна зробити висновки, що для задач з розмірністю менше 10 оптимальним значення параметру g є g = 15, тому що значення цільової функції перестає покращуватись. Для задач з розмірністю 10 < n < 60 оптимальним значення параметру g є g = 50. Для задач з розмірністю n > 60 оптимальним значення параметру g є g = 100.

### Експеримент №2

*Мета*: Дослідження впливу значень параметрів на результати роботи алгоритму.

N = 3 – кількість прогонів алгоритму для кожного набору параметрів.

Тести проводились на таких класах задач:

* 30/S,L/S,L/L,S/L,S;
* 30/L,S/S,L/S,L/S,L;
* 30/L,S/L,S/L,S/L,S;

***Таблиця 16 -*** *результати експерименту №2.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Клас задачі | Параметри |  |  |  |
| 1 | Середній найкращий розв’язок | 67,31128 | 63,44477 | 68,20417 |
| 1 | Cередній час роботи алгоритму | 127 | 135 | 234 |
| 2 | Середній найкращий розв’язок | 85,543915 | 79,377525 | 80,49629 |
| 2 | Cередній час роботи алгоритму | 123 | 49 | 77 |
| 3 | Середній найкращий розв’язок | 100,564735 | 89,060486 | 89,404945 |
| 3 | Cередній час роботи алгоритму | 54 | 74 | 74 |

Згідно з результатами в таблиці №16, можна зробити висновок, що оптимальним значенням параметрів є . Хоча значення цільової функції краще при , проте алгоритм працює майже в 2 рази повільніше.

## Загальні експерименти

### Експеримент №1

*Мета*: порівняння розроблених алгоритмів за часом та точністю

Тести проводились на таких класах задач:

* 5/S,L/S,L/L,S/L,S;
* 10/L,S/S,L/S,L/S,L;
* 20/S,L/S,L/L,S/L,S;
* 50/L,S/L,S/L,S/L,S;
* 100/L,S/L,S/L,S/S,L.

***Таблиця 17 -*** *результати експерименту №1.*

|  |  |  |  |
| --- | --- | --- | --- |
| n | Алгоритм | Час роботи алгоритму | Значення ЦФ найкращого розв’язку |
| 5 | Жадібний | 13 | 10,006453 |
| 5 | Генетичний | 28 | 12,5769415 |
| 5 | Бджолиний | 34 | 12,5769415 |
| 10 | Жадібний | 10 | 33,19274 |
| 10 | Генетичний | 88 | 43,017475 |
| 10 | Бджолиний | 123 | 43,017475 |
| 20 | Жадібний | 15 | 62,949898 |
| 20 | Генетичний | 234 | 67,5062 |
| 20 | Бджолиний | 124 | 64,23118 |
| 50 | Жадібний | 31 | 151,43344 |
| 50 | Генетичний | 182 | 199,39381 |
| 50 | Бджолиний | 363 | 163,93246 |
| 100 | Жадібний | 34 | 260,07993 |
| 100 | Генетичний | 460 | 282,64313 |
| 100 | Бджолиний | 937 | 280,78162 |

Згідно результатів експерименту, можна зробити висновки, що для задач з розмірністю n 10 краще застосовувати генетичний алгоритм, адже він дає такий самий результат як і бджолиний за менший час. Для задач з розмірністю n 20 генетичний алгоритм працює довше ніж бджолиний і дає трішки кращий результат. Для задач з розмірністю n 50 бджолиний алгоритм працює повільніше ніж генетичний та дає гірші результати. Щодо жадібного алгоритму, він дає найгірші розв’язки, проте сильно виграє по часу роботи. В середньому жадібний алгоритм працює в 6 разів швидше за генетичний та в 11 разів швидше за бджолиний. Таким чином, якщо нам потрібні непогані допустимі розв’язки за мінімальний час, можна скористатись жадібним алгоритмом.

В цілому ж найкраще себе проявив генетичний алгоритм, який в більшості випадків дає кращий розв’язок за середній час.

**Використані джерела**

1. <http://bigor.bmstu.ru/?cnt/?doc=045_ga/1010.mod/?cou=Default/Evolution.cou>
2. <http://cad.kpi.ua/attachments/093_2016d_Leshyk.pdf>
3. **Holland J. Adaptation in Natural and Artificial Systems. – Univ. of Michigan Press, 1975.**
4. **Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. - Addison-Welsey, 1989.**
5. Конспекти лекцій з предмету ,,Дослідження операцій,, доц. каф. АСОІУ, к.т.н. Жданової О.Г.
6. Соболь И.М. Выбор оптимальных параметров в задачах со многими критериями. — М.: Дрофа, 2006. — 175 с. — ISBN 5-7107-7989-X.
7. Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen та Roman Slowinski. Multiobjective Optimization: Interactive and Evolutionary Approaches (Lecture Notes in Computer Science) (англ.). — Springer. — ISBN 3-540-88907-8.
8. <https://cybernetics.wikia.org/ru/wiki/%D0%AD%D1%84%D1%84%D0%B5%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%BF%D0%BE_%D0%9F%D0%B0%D1%80%D0%B5%D1%82%D0%BE>
9. [http://naukarus.com/metod-issledovaniya-prostranstva-parametrov-v-zadachah-proektirovaniya-legkovyh-avtomobiley](http://naukarus.com/metod-issledovaniya-prostranstva-parametrov-v-zadachah-proektirovaniya-legkovyh-avtomobiley" \t "_blank" \o "http://naukarus.com/metod-issledovaniya-prostranstva-parametrov-v-zadachah-proektirovaniya-legkovyh-avtomobiley)
10. <http://technomag.edu.ru/doc/547747.html>
11. Лотов А.В., Поспелова И.И. Многокритериальные задачи принятия решений: учеб. пособие. М.: МАКС Пресс, 2008. 197 c.
12. Ларичев О.И. Теория и методы принятия решений. М.: Университетская книга, Логос, 2002. 392 c.
13. Miettinen K., Makela M.M. Interactive Bundle-based Method for Nondifferentiable Multiobjective Optimization: NIMBUS // Optimization: A Journal of Mathematical Programming and Operations Research. 1995. Vol. 34, no. 3. P. 231-246.
14. <https://phm.cuspu.edu.ua/nauka/konferentsii/fizyka-tekhnolohii-navchannia/99-2017/komp-iuterni-nauky-ta-informatsiini-tekhnolohii/1122-alhorytmy-royovoho-intelektu-ta-yikh-zastosuvannya.html>
15. <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/linq/>