

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

19 травня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційна технологія автоматичної перевірки завдань на
навчальній платформі Міх»
здобувача групи ІН.м-25 Резніка Миколи Миколайовича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

Микола РЕЗНІК

_____ (підпис)

Керівник
старший викладач кафедри
комп'ютерних наук, к.м.-ф.н.

Оксана ШОВКОПЛЯС

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття освітнього ступеня магістра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН.м-25 Резніка Миколи Миколайовича

1. Тема роботи: «Інформаційна технологія автоматичної перевірки завдань на навчальній платформі Міх»

затверджую наказом по СумДУ від «06» грудня 2023 року № 1412-VI

2. Термін задачі здобувачем кваліфікаційної роботи до 16 грудня 2023 року

3. Вихідні дані до кваліфікаційної роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми та постановка задачі. 2) Стандартний генетичний алгоритм.

3) Інформаційне та програмне забезпечення 4) Аналіз результатів роботи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти до проєкту (роботи) із зазначенням розділів проєкту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «б» листопада 2023 р.

Завдання прийняв до виконання _____

Керівник _____

(підпис)

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми та постановка задачі</i>		
2	<i>Стандартний генетичний алгоритм</i>		
3	<i>Інформаційне та програмне забезпечення</i>		
4	<i>Аналіз результатів роботи</i>		
5	<i>Оформлення кваліфікаційної магістерської роботи</i>		

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

АНОТАЦІЯ

Записка: 69 стор., 8 рис., 1 додаток, 20 використаних джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки присвячена розв’язанню важливої практичної задачі застосування інтерактивних компонентів електронного навчання аналізу доцільності, розроблення відповідних методів, моделей та інформаційної технології.

Об’єкт дослідження — процес автоматичної перевірки завдань в електронному навчанні.

Мета роботи — програмна реалізація тренажеру з інтерактивними компонентами на платформі Міх Сумського державного університету

Методи дослідження — генетичний алгоритм.

Результати — виконано програмну реалізацію тренажеру з інтерактивними компонентами, що дозволяє перевірити та закріплювати навички з використання стандартних кроків генетичних алгоритмів. Для досягнення цієї мети були виконані такі завдання: створено вхідний математичний опис тренажеру, сформульовано математичну модель етапів стандартного генетичного алгоритму, програмно реалізовано алгоритм роботи тренажеру, що перевіряє знання основних етапів застосування еволюційних механізмів, також була протестована працездатність розробленого додатку.

ТРЕНАЖЕР, JAVASCRIPT, ГЕНЕТИЧНИЙ АЛГОРИТМ, МУТАЦІЯ.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Системи електронного навчання	6
1.2 Системи управління навчанням.....	14
1.3 Інтерактивні елементи електронного навчання	18
1.4 Актуальність	23
1.3 Постановка задачі.....	24
2 СТАНДАРТНИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ.....	25
2.1 Опис алгоритму	25
2.2 Математична модель алгоритму	32
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	39
3.1 Формування вихідних даних.....	39
3.2 Короткий опис програмної реалізації.....	39
3.3 Результати тестування додатку.....	43
ВИСНОВКИ.....	47
СПИСОК ЛІТЕРАТУРИ	48
ДОДАТОК	50

ВСТУП

Актуальність. Сьогодні стрімкий розвиток електронного навчання спровокований глобальним локдауном виниклим через COVID-19 змусив організації та освітні установи активно вести свої бізнес процеси та навчальну діяльність у дистанційному форматі з використанням цифрових технологій. Ці фактори спричинили стрімкий ріст попиту на програмні рішення у сфері електронного навчання та наукові дослідження з приводу його ефективності в порівнянні класичним навчальним процесом. Таким чином розроблена технологія автоматичної перевірки завдань буде мати попит серед розробників електронних навчальних курсів.

Об'єкт дослідження – процес автоматичної перевірки завдань в електронному навчанні.

Предмет дослідження – методи і технології автоматичної перевірки завдань на навчальній платформі Міх.

Гіпотеза. Використання інтерактивних компонентів електронної освіти у віртуальних тренажерах може підвищити якість навчальних матеріалів.

Структура. Дана робота складається зі вступу, аналізу публікацій, постановки задачі дослідження, вибір методики та інструментів для рішення поставленої проблеми, опису програмної реалізації тренажеру, висновків, списку літератури та додатка.

Зв'язок роботи з науковою темою. Кваліфікаційна робота виконана на кафедрі комп'ютерних наук та пов'язана з виконанням науково-дослідної роботи № 0118U006971 «Методи, математичні моделі та інформаційні технології аналізу і синтезу інфокомунікаційних систем» (2018-2023).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Системи електронного навчання

Електронне навчання - це освітній підхід, який використовує цифрові технології та інтернет-ресурси для забезпечення процесу отримання знань, навичок та компетентностей. Воно охоплює різноманітні напрямки, включаючи комп'ютерне навчання, онлайн-освіту, а також використання електронних книг, платформ соціальних мереж, аудіовізуальних технологій і цифрових мереж мовлення. На відміну від традиційного навчального процесу, використання цифрових інструментів, що дає можливість здобувати знання дистанційно в онлайн-режимі, не обов'язково перебуваючи в аудиторії.

Головна ідея це використання систем електронного навчання для постачання освітнього контенту, оцінювання навчального прогресу та налагодження зв'язку між студентами та викладачами. Сьогодні електронне навчання є необхідним інструментом для вирішення дефіциту кваліфікованих фахівців у сучасному технологічному світі. З моменту виникнення COVID-19 та через терміновий перехід від очного до онлайн-навчання, навчальні заклади, включаючи вищі навчальні установи, широко використовують системи електронного навчання. [1]

Електронне навчання поділяють на синхронне та асинхронне. В асинхронному електронному навчанні викладання та навчання відбуваються в різний час. Викладач створює онлайн-урок, а учень має доступ до нього пізніше, не обов'язково в той самий час, що й інші учні. Матеріали асинхронного електронного навчання призначені головним чином для індивідуального самостійного навчання. Зазвичай учні взаємодіють з асинхронними курсами власним темпом та мають контроль над вибором тем і елементів завдяки опціям навігації. Асинхронні курси, як правило, пропонують менше можливостей для соціальної присутності, ніж синхронні заняття, хоча асинхронна співпраця

можлива через форуми обговорень. Асинхронні курси покладаються на самодисципліну та само мотивацію учня для їх проходження.

Синхронне відсилає на події, які відбуваються в той самий час, отже, в синхронному електронному навчанні викладання і навчання відбуваються одночасно. Група студентів відвідує заняття з живим інструктором у мережі. Синхронне електронне навчання, яке загалом відоме як віртуальні класи, зазвичай є інструкторським і призначене для участі в ньому кількох учнів одночасно за допомогою програмних інструментів, таких як Zoom або MS Teams. У синхронному електронному навчанні існують численні можливості для соціальної присутності, включаючи відео та аудіо інструктора і учасників, чат, маркування на дошці та робочі кімнати. Віртуальні класи проводяться за попереднім графіком, що може вимагати від деяких учнів приєднатися в різні години в залежності від їх часового поясу. Оскільки синхронне електронне навчання є запланованою подією, відсоток завершивших курс учнів часто вище порівняно з асинхронними курсами, які покладаються на рівні управління часом навчання учнів. Як у звичайному класі, інструктор контролює подію, встановлюючи темп та організовуючи навчальні елементи, такі як практика, обговорення та інше.

Оптимальна модель навчання може варіюватися залежно від цілей курсу та змісту, але зазвичай використовують комбінацію асинхронних та синхронних моделей навчання, щоб поєднати слабкі та сильні сторони обох моделей (див. табл. 1.1). Синхронне навчання ідеально підходить для створення соціальної взаємодії та швидкої обміну інформацією, допомагаючи будувати навички командної роботи та ведення дискусії[2].

Таблиця 1.1 Порівняння синхронного та асинхронного навчання

Характеристика	Асинхронне навчання	Синхронне навчання
Часовий режим	Студенти можуть вчитися у відповідності до свого графіка	Заняття та лекції проходять в реальному часі з участю всіх студентів
Взаємодія	Студенти працюють незалежно та обмінюються матеріалами через платформи	Заняття проводяться в реальному часі, дозволяючи студентам спілкуватися між собою та інструкторами
Темп навчання	Залежить від темпу студента	Графіки занять визначають ритм навчання
Спілкування	Необов'язкове інтерактивне спілкування	Активне спілкування, можливість групової роботи та дискусії
Програма	Студенти можуть працювати з матеріалами самостійно та повертатися до них	Фіксована програма, яку слід дотримуватися
Підтримка вчителя	Зазвичай обмежена в реальному часі, але є можливість спілкування через форуми та електронну пошту	Велика кількість реального часу для відповідей, питань та допомоги

Однак це вимагає планування об'єднаних часів для студентів та вчителів, часто з урахуванням різниці в часових поясах, і схильне до технічних труднощів та обмежень доступності, по технічним причинам. Асинхронне навчання, навпаки, є часово більш гнучкішим. Це не тільки дозволяє студентам більше часу для вивчення та спілкування з матеріалами, але і робить матеріал курсу доступним для більш широкого кола студентів. Навіть простий запис заняття у класі додає можливість, його перегляду у форматі асинхронного відтворення.

Через різні підходи до навчального процесу, під час асинхронного та синхронного електронного навчання, не всі медіа будуть еквівалентними. Наприклад:

Друковані матеріали, такі як книги, є недорогими, самостійними та портативними. Однак традиційні книги обмежені друкованим текстом та кількома графічними зображеннями.

Тренінги, які веде інструктор в аудиторії, надають високу соціальну присутність та можливості для практичної роботи, але вони залежить від темпу інструктора та повільно змінюють свій зміст.

Цифрові технології представляють один з найгнучкіших варіантів медіа, оскільки вони можуть підтримувати мультимедійні елементи, такі як друкований текст, графіка (нерухома, анімована, відео та 3D) та аудіо, а також інтерактивну взаємодію зі зворотнім зв'язком. Є можливості для унікальної участі у симуляціях або реалістичних імерсійних середовищах, які у деяких випадках можуть бути неможливі для відтворення в фізичному середовищі.

Симуляції пропонують стислий досвід, який може прискорити здобуття експертизи.

Вчителі повинні бути навчені використовувати різні медіа та знати принципи та інтерактивні програми під час процесу електронного навчання або мати сучасний путівник щодо того, як навчати та оцінювати студентів. Оскільки, як було описано в цьому огляді, від студентів в основному очікується поєднання синхронних та асинхронних методів.

Електронне навчання має багато сильних сторін (див табл. 1.2) та не менше потенційних проблем (див табл. 1.3). Усі ці особливості можуть як підвищити ефективність навчання, так і зробити курс надто перевантаженим, коли використовуються не вірно. Цифрові середовища легко можуть перевантажити здатність людини обробляти всю надану інформацію. Електронне навчання може стати перевантажним для системи обробки інформації учня, якщо воно не враховує, як саме людина сприймає інформацію.

Правильно організоване електронне навчання надає можливість створювати послідовні, зручні, цілеспрямовані, адаптивні, масштабовані та захоплюючі навчальні середовища. Це може значно поліпшити навчальний процес, забезпечуючи студентам зручний доступ до матеріалів, адаптованих до їхніх потреб та темпу навчання.

Такі середовища можуть також заохочувати активну участь, інтерактивність та зацікавленість студентів, що позитивно впливає на їхню успішність та розвиток[3].

Таблиця 1.2 Потенційні переваги електронного навчання

Потенційні переваги електронного навчання	Опис
Гнучкість навчальних процесів	Асинхронне електронне навчання індивідуалізує навчання, адаптуючи зміст та методи навчання під унікальні потреби учнів. Контроль учня дозволяє просуватися власним темпом та вибирати теми та методи з урахуванням їхніх потреб.
Залучення до навчання	Включає як поведінкове, так і психологічне залучення. Поведінкове залучення - активності, які виконує учень, такі як введення відповіді чи взаємодія з матеріалом. Психологічне залучення - когнітивна обробка контенту для засвоєння нових знань та навичок.
Мультимедіа	Використання комбінації тексту, аудіо, статичних та рухових візуальних засобів для ефективного передачі змісту.
Навчання через ігри	Гейміфікація - використання ігор для створення мотивуючого та ефективного навчального досвіду.
Прискорення отримання досвіду	Електронне навчання дозволяє учням занурюватися в реалістичне робоче середовище, прискорюючи отримання досвіду та вирішення завдань.

Таблиця 1.3 Потенційні проблеми електронного навчання

Потенційні проблеми електронного навчання	Опис
Занадто яскраве представлення	Яскраве та перевантажене представлення може відволікати учнів та обмежувати їхню здатність зосередитися на змісті. Важливо збалансувати використання анімацій, звуків та тексту.
Недостатньо яскраве представлення	Мінімалістичне оформлення може привести до втрати ефективності, особливо якщо не використовуються функції, які сприяють навчанню. Важливо забезпечити адекватне використання візуальних елементів та взаємодії.
Втрата ефективності	Брак оцінки результатів навчання може призвести до невпевненості у ефективності програми. Оцінка та вдосконалення ресурсів підготовки є важливими для забезпечення якісного навчання.
Навчання через відкриття	Використання навчальних середовищ, які дозволяють учням вільно будувати навчальний план, може призвести до втрати структури та контролю. Рекомендується використовувати структуровані форми електронного навчання для надання відповідного керівництва.

Класичну структуру електронного навчання можна поділити на три компоненти:

- платформа навчання
- об'єкт навчання
- інструмент навчання

Інструментами зазвичай є пристрої, які дозволяють користувачам отримувати доступ та взаємодіяти з цифровим освітнім контентом.

Це можуть бути різноманітні пристрої, такі як особисті комп'ютери (включаючи настільні, ноутбуки та нетбуки), планшети, а також смартфони. Мобільні додатки для навчання стають все більш популярними, дозволяючи навчання в будь-якому місці та часі, зокрема під час подорожей.

Об'єктом електронного навчання є конкретний курс, який охоплює освітні цілі та результати, встановлені авторами курсу чи навчальної програми. Об'єкт навчання слугує центральною точкою, навколо якої структуровані матеріали, завдання та оцінки електронного навчання. У електронному навчанні об'єкт навчання може значно варіюватися залежно від характеру курсу та задекларованих навчальних цілей.

Платформа навчання - це місце, де користувач проймає курси. Зазвичай це представлено у вигляді веб-сайту або системи управління навчанням (Learning Management System, LMS). Платформа навчання надає інфраструктуру для навчального процесу, включаючи доступ до матеріалів, інтерактивні можливості, систему оцінювання та взаємодію між учасниками курсу.

Вона є основним середовищем, яке об'єднує інструмент та об'єкт навчання, створюючи зручний простір для здійснення навчання [4].

1.2 Системи управління навчанням

Системи управління навчанням (Learning Management Systems, LMS) використовують для організації електронного навчання. LMS надають вчителям та учням онлайн-клас, для супроводження навчального процесу. У віртуальних класних середовищах LMS підтримують вчителів та учнів у процесі навчання. Стандартна LMS забезпечує інклюзивне навчальне середовище для академічного прогресу з впровадженням структур, що сприяють спільній-онлайн роботі груп, професійній підготовці, обговоренням та комунікації між різними користувачами.

Ще з 1950-х років, інженери комп'ютерних систем вважали застосування систем LMS з метою освіти за необхідне. З плином часу технології та інструменти, які підтримують структури онлайн-навчання, розвивалися з часів появи онлайн-навчання в середині та наприкінці 1990-х років. Композиції LMS включають різноманітні медіа та комунікаційні інструменти що сприяють вибору учня у засобах засвоєння матеріалів[5].

Системи управління курсами (Course Management Systems, CMS) Система управління курсами CMS надає користувачам набір інструментів, які структурують онлайн-взаємодії. Вона надає процес керування інформацією в одному основному місці. CMS надають онлайн- та змішані курси, де користувачі можуть отримати доступ до тек курсового матеріалу, а також інструментів та інших матеріалів, що містять важливу інформацію про курс.

CMS включають можливості відстеження прогресу, оцінювання та комунікативні платформи для групових обговорень, чатів та розміщення інформації. Хоча CMS та LMS мають деякі спільні риси, CMS спеціалізується на управлінні та створенні навчального контенту, тоді як LMS надає системну платформу для тренування, що проводить класи електронного навчання для відстеження завершення курсу та оцінювання результатів.

Головною метою CMS та LMS систем є забезпечення учнів і викладачів, зручними у використанні платформами, які сприятимуть ефективному навчанню та тренуванню учнів, незалежно від того, чи йдеться про навчальні заклади, корпоративні структури чи інші освітні середовища.

LMS система забезпечує структуру для асинхронних методів доставки, які включають електронну пошту, обговорення в групах, аудіо презентації обговорень та газети для взаємодії вчителів та учнів. Асинхронні методи доставки дозволяють учням спілкуватися одне з одним без роз'єднання відстанню та часом. Асинхронні середовища дозволяють учням працювати разом із іншими зобов'язаннями та відповідальностями. Додатково, в асинхронних структурах учні повинні навчитися користуватися LMS для вивчення курсових матеріалів, здійснення ефективного спілкування та керування технологіями курсу.[6]

LMS також надає структуру для синхронних методів доставки за допомогою відео та онлайн-конференцій. Учні можуть бачити своїх викладачів через відео, онлайн-обговорення, чат в реальному часі, а також презентації та файли. Відео конференційна комунікація через синхронні структури сприяє взаємодії між учнями та їх викладачами.

Як асинхронні, так і синхронні методи використовують LMS для забезпечення учнів позитивними впливами на полегшення їхнього навчання. Порівнюючи асинхронне та синхронне навчання, важливо відмітити, що в дистанційному навчальному середовищі синхронне навчання є кращим для розвитку критичних навичок рефлексії студента. Асинхронні системи навчання не надають учням миттєвого зворотного зв'язку. Асинхронні системи навчання також можуть залишити учнів відчуттям ізольованими та віддаленими від їхнього соціального середовища.[7] Однак розширене використання асинхронних інструментів навчання надає гнучкість, оскільки учні можуть рефлексувати та завершувати завдання.

Змішані навчальні середовища надають студентам можливість користуватися як синхронними, так і асинхронними методами. Студенти, які використовують обидва методи, досягають кращих оцінок порівняно з традиційним стилем навчання.

У 2019 був реалізований експеримент з 49 студентами, які використовували LMS Blackboard з обома асинхронними та синхронними методами для проходження курсу з фізики. Протягом курсу студентам слід було використовувати синхронний метод відеоконференцій для лекцій з викладачем та обговорень курсу протягом першого семестру, і асинхронний метод відеоконференцій з Blackboard протягом наступних сім тижнів. У кінці кожних семи тижнів студенти склали екзамен. Після завершення двох семи тижнів їхні оцінки порівняли. Результати показали, що 46,6% учасників віддали перевагу синхронному методу відеоконференцій, а 53,4% студентів віддали перевагу асинхронному методу відеоконференцій з Blackboard[8].

Результати дослідження свідчать про те, що використання асинхронного формату відеоконференцій з Blackboard позитивно впливало на сприйняття студентами матеріалу.

LMS системи використовуються як в освітніх установах, так і в великих компаніях. На сьогоднішній день існує безліч LMS систем, і кожна відрізняється пропонованими функціями і ціною. Організації стикаються з рішенням вибору LMS із пропрієтарною системою чи відкритою системою. Рішення в основному залежить від наявних ресурсів та рівня знань чи кваліфікації користувачів LMS в організаціях.

Пропрієтарна система використовує винятковий код, де організації купують підписку або ліцензію для доступу та використання функцій LMS. Деякі приклади пропрієтарних систем включають Blackboard, PowerSchool, SchoolWires, Edline, eSchoolView та SchoolPointe.

Відкриті системи використовують безкоштовну ліцензію, де користувачі мають можливість вільного доступу та використання системи. Приклади відкритих систем включають Moodle, WordPress і Drupal.

П'ятдесят дев'ять відсотків шкільних організацій обирають пропрієтарні системи, де вендори стягують плату за обслуговування згідно з кількістю користувачів LMS у хмарних пропрієтарних системах. Організації які обирають пропрієтарні системи, отримують переваги співпраці з компанією, яка займається створенням та розповсюдженням онлайн-рішень для підтримки навчання. Клієнти, які використовують пропрієтарні системи, також можуть отримати послуги навчання кінцевих користувачів.[9]

Недолік використання пропрієтарної системи полягає в тому, що клієнти обмежені і можуть отримати доступ лише до визначених функцій LMS. Організації із пропрієтарними системами починають інтегрувати хмарні рішення LMS, де торговець LMS зберігає дані клієнта в Інтернеті. Хмарні рішення LMS дозволяють користувачам підтримувати фізичну інфраструктуру для запуску LMS на місці.

Деякі освітні організації просувають використання відкритих систем для створення онлайн-середовищ, які використовують різноманітні медіа та комунікаційні засоби. Ці відкриті системи продовжують розвиватися, зазвичай вони працюють у веб-середовищах. Організації можуть використовувати відкриті системи для забезпечення своїх потреб. Відкриті системи дозволяють користувачам розширювати функціональність системи та використовувати плагіни від спільноти або власної розробки. Студенти коледжів та університетів є найбільшою групою користувачів відкритих систем.[10].

1.3 Інтерактивні елементи електронного навчання

Велику роль у дистанційних курсах відіграє інтерактивний елемент навчання. Інтерактивність - це процес діалогу між учнями та засобами електронного навчання через який учні стають залученими в процес електронного навчання. Саме завдяки взаємодії користувача з вмістом на основі ІКТ, в якій двосторонній обмін між учнем та вмістом керується візуалізаціями, активною участю учня та впливом на вміст, а також вивченням, яке підсилюється мотиваційними відгуківими сигналами або підказками. Навіть при використанні багатьох інструментів, що використовуються в обчисленнях, інтерактивне електронне навчання не тотожне інтерактивному навчанню, в якому "включення соціальних та міських обчислень у дизайн та доставку курсу" визначено як основна мета.

Зазвичай визначають наступні рівні інтерактивності(див табл. 2.1).

Таблиця 2.1 Форми інтерактивної взаємодії

Форма Взаємодії	Визначення	Вплив на Результати Навчання
Студент - Вчитель	Активна взаємодія між студентом і вчителем в онлайн-активностях.	Заохочення навчання, соціальна взаємодія, комунікація, підтримка.
Студент - Зміст	Активна участь студента взаємодії з вмістом курсу, включаючи семінари, лекції тощо.	Збагачення знань, побудова знань, підтримка академічних цілей.

Продовження Таблиці 2.1

Студент - Система	Взаємодія студента з технічними аспектами навчальної системи.	Вплив на задоволеність студента, користування технічними ресурсами.
Студент - Студент	Взаємодія між студентами, основана на підтримці, обговореннях та спільній роботі.	Збільшення взаєморозуміння, покращення результатів завдяки груповій роботі.

До інтерактивних елементів в електронному навчанні можна віднести:

1. Мультимедійні елементи:

1) Відео та аудіо матеріали: Навчання за допомогою відео- та аудіофайлів може зробити процес більш доступним та цікавим.

2) Анімації та інтерактивні симуляції: Засоби, які дозволяють студентам взаємодіяти з анімованими чи інтерактивними сценаріями для кращого розуміння концепцій.

b. Інтерактивні завдання:

1) Тести та опитування: Використання онлайн-тестів для оцінювання рівня знань та розуміння студентів.

2) Вправи та завдання: Створення вправ, які дозволяють студентам застосовувати свої знання на практиці.

c. Форуми та обговорення:

1) Онлайн-форуми: Створення місця для обговорення концепцій, обміну думками та взаємодії між учнями.

2) Чати та групи: Можливість спілкування в реальному часі для обговорення матеріалів та вирішення питань.

d. Відстеження прогресу:

1) Системи відстеження: Засоби, що дозволяють вчителям та учням відслідковувати прогрес у навчанні та виконання завдань.

е. Взаємодія з викладачем:

- 1) Віртуальні консультації: Організація онлайн-зустрічей або консультацій для вирішення питань студентів.
- 2) Електронна пошта та обмін повідомленнями: Засоби для комунікації між студентами та викладачами.

ф. Спільна робота:

- 1) Групові проекти: Завдання, які вимагають співпраці між студентами для досягнення спільних цілей.
- 2) Інтерактивні вправи в реальному часі: Можливість спільно вирішувати завдання або виконувати вправи у реальному часі.

Особливості інтерактивного навчання з методичної точки зору полягає в розробці такої структури змісту електронного курсу і форми завдань, які б сприяли активізації роботи учнів та підвищення їх рівня зацікавленості у навчальному процесі. Кінцевою метою має бути сприяння їх оптимальному розвитку й вихованню. [11]

На відміну від звичайного заняття, заняття з залученням інтерактивних технологій фокусується на стимулюванні ініціативи учнів. Що має призвести до зростання якості їх творчої та самостійної пізнавальної діяльності.

Використання вище описаних особливостей при проектуванні є одним з найважливіших принципів побудови інтерактивного навчально-виховного процесу. Впровадження методики інтерактивного навчання при викладанні дисциплін дозволяє змінити відношення до об'єкта навчання, перетворивши об'єкта на суб'єкт. Студенти будуть активно взаємодіють між собою, обмінюючись ідеями та думками. Централізація уваги на студенті підкреслює важливість його ролі у власному навчанні. Він стає не лише приймачем інформації, але і активним учасником навчального процесу. Звісно це неможливо без взаєморозуміння та повазі до думки кожного студента. [12]

Як приклад розберемо вирішені приклади(Worked-out examples) це загальноприйнята методика в підручниках, пов'язаних із добре структурованими областями, такими як математика, фізика і програмування. Зазвичай вони складаються з формулювання проблеми, одного чи кількох кроків розв'язання та остаточного розв'язку проблеми. Показуючи, як вирішити проблему крок за кроком, вирішені приклади надають учням модель експерта щодо того, як вирішити певну проблему, якою вони можуть скористатися для направлення своїх власних зусиль у вирішенні проблем. Оскільки вони виконують цю роль, їх зазвичай поєднують з однією чи кількома практичними завданнями, де учням подається формулювання проблеми, і їхньою відповідальністю є заповнення окремих кроків розв'язання та надання остаточного розв'язку.

Вирішені приклади не лише представляють собою високо оцінений джерело інструкцій для учнів, але й ефективний засіб, особливо для учнів із низьким попереднім досвідом у галузі.

Далі розглянуті (див табл. 2.2) інтерактивні елементи для вирішених прикладів де описано їхню роль у сприянні активному навчанню та покращенні розуміння матеріалу.

Таблиця 2.2 Інтерактивні елементи для підходу вирішених прикладів

Елемент	Опис
Пропуски	Вставлені в вирішені приклади пропуски або неповні кроки, які вимагають від учнів доповнення або розширення, сприяючи активному мисленню і самостійному розв'язанню.
Підказки	Додаткові вказівки або наведення, які надаються учням за їхнім запитанням або на вимогу. Можуть містити мінімальні пояснення або розгорнуті інструкції в залежності від потреби.
Подавання вказівок	Зазначення конкретних напрямків або вказівок, спрямованих на активний розгляд центральних аспектів матеріалів навчання.
Допомога за вимогою	Системи допомоги, які стають доступними на вимогу учня. Надають інструкційні пояснення або допомагають учням розуміти складні концепції при їхньому запитанні.

Інтерактивні елементи, такі як пропуски, підказки та засоби допомоги в середовищах навчання на прикладах, можуть сприяти навчанню. Однак ефективність не полягає у самому інтерактивні. Пропуски чи підказки, які відволікають увагу від центральних аспектів навчального матеріалу, мало в чому допомагають або навіть можуть завдати шкоди. Крім того, інтерактивні засоби допомоги можуть "спотворити" учнів так, що вони більше не оброблятимуть активно центральну інформацію.

Інтерактивні елементи ефективні лише тоді, коли вони викликають процеси навчання, які безпосередньо стосуються центральних навчальних цілей. Важливим є не сам "дія" в інтерактивні, а обробка, викликана цією "дією"

Підсумовуючи, інтерактивність сама по собі не є ефективною. Однак інтерактивні елементи можна використовувати для виклику обробки центральних аспектів навчального матеріалу. Для цього інтерактивні елементи повинні бути ретельно розроблені і, якщо можливо, перевірені в емпіричних дослідженнях.

1.4 Актуальність

Під час пандемії та війни у Україні відбувся вимушений перехід до дистанційного навчання "близько 800 тис. школярів змінили форму здобуття освіти з денної на дистанційну (з 17 669 учнів/ учениць у 2021 році до 772 909 у 2022)". Такий ризиковий перехід призвів до погіршення результатів навчання в освітньому секторі. Частина респондентів у дослідженні пов'язала ключові виклики з невмінням учнів/учениць самостійно вчитися — 39 % та брак зворотного зв'язку щодо навчального поступу від учителів — 7 %. [13].

Згідно з зарубіжними дослідженнями, учні дванадцятих класів, які перейшли з очної форми навчання на дистанційну, втратили приблизно 35% навчального рівня за звичайний рік[14]. Такі результати свідчать що погано спроектоване електронне навчання менш ефективне, ніж традиційні формати навчання.

Оскільки з розвитком технологій, які мають все більшу здатність до постачання різноманітних мультимедійних матеріалів, навчальні програми стають більш складними, підтримуючи більшу взаємодію, реалістичні графічні зображення, захопливі середовища, відео високої якості, захоплюючі ігри та інші форми анімації. Деякі переваги форматів електронного навчання полягають у тому, що середовища навчання пропонують більшу доступність, зручність, індивідуалізацію та масштабованість порівняно з традиційними очними форматами навчання. [15]

Ефективність електронного навчання залежить від адекватності використовуваних методів навчання, тому ця робота акцентується на принципах дизайну, підтверджених доказами, які були показані як ефективні.

Використання електронного навчання значущим чином впливає на процес будування знань у учнів і сприяє їх мотивації докладати більше зусиль для вивчення будь-яких джерел інформації, які їм можуть бути представлені. Таким чином, є потреба в розгляді різних поглядів для забезпечення учня якісним матеріалом.[16]

1.3 Постановка задачі

Метою роботи є програмна реалізація тренажеру, що дозволить перевірити навички з розуміння принципів роботи стандартного генетичного алгоритму.

Щоб досягти мети необхідно виконати наступні завдання:

1. Сформувати математичну модель еволюційних механізмів, що використовується в стандартному генетичному алгоритмі.
2. Програмно реалізувати алгоритм роботи тренажеру, що перевіряє знання з основних етапів застосування еволюційних механізмів стандартного генетичного алгоритму.
3. Перевірити працездатність розробленого тренажеру.

2 СТАНДАРТНИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ

2.1 Опис алгоритму

Генетичний алгоритм(ГА), розроблений Джоном Голландом та його співробітниками в 1960-1970 роках (Голланд, 1975; Де Жонг, 1975), представляє собою абстраговану модель біологічної еволюції на основі теорії природного відбору Чарльза Дарвіна. Ймовірно, Голланд став першим, хто використав мутацію, кросовер, рекомбінацію та відбір в дослідженні адаптивних систем. Саме використані їм оператори становлять основну частину стандартного генетичного алгоритму (див. рис. 2.1.) для розв'язання оптимізаційних задач. Це оператори:

- відбір
- кросовер
- мутація
- заміщення

Найважливішу роль відіграє оператор кросоверу в генетичному алгоритмі. Кросовер аналогічний природній рекомбінації генетичної інформації в живій природі, що відбувається під час схрещування організмів в процесі розмноження.

Оператор кросоверу в генетичному алгоритмі включає обмін генетичною інформацією між двома батьківськими особинами з метою створення нових потомків. Це сприяє комбінуванню корисних характеристик різних рішень-кандидатів у сподіванні отримання ще кращого рішення.

Існує багато переваг генетичних алгоритмів перед традиційними алгоритмами оптимізації. Два найбільш помітні - це можливість вирішення складних проблем та паралелізм. Генетичні алгоритми можуть працювати з різними видами оптимізації, незалежно від того, чи є об'єктивна (фітнес) функція стаціонарною чи нестаціонарною (змінюється з часом), лінійною чи нелінійною, неперервною чи розривною, чи з випадковим шумом.

Генетичний алгоритм базується на концепції популяційного підходу, де кожна особина в популяції є незалежним агентом, що працює над власним

рішенням. Це надає велику ступінь паралельності та розподіленості обчислень в рамках алгоритму. Різні параметри та навіть різні групи закодованих рядків можуть бути оброблені одночасно [17].

Генетичні алгоритми, як і будь-який інший метод оптимізації, мають свої недоліки та виклики. Визначення правильної функції пристосованості важливо для ефективності генетичного алгоритму. Неправильно сформульована функція пристосованості може призвести до некоректної оцінки рішень. Вибір розміру популяції важливий для збалансованості між різноманітністю та обчислювальною ефективністю. Параметри, такі як швидкість мутації і кросовера, повинні бути налаштовані відповідно до конкретної задачі. Неправильний вибір може вплинути на ефективність алгоритму. Методи відбору та формулювання критеріїв для формування нової популяції впливають на швидкість та якість збіжності. Генетичні алгоритми не завжди гарантують знаходження глобального оптимуму, а можуть застрягти в локальних мінімумах.

Незважаючи на ці недоліки, генетичні алгоритми широко використовуються завдяки їхній здатності оптимізації в різноманітних задачах, а також здатності працювати з великими та складними просторами пошуку.

Для опису процесів та об'єктів генетичних алгоритмів використовуються визначення запозичені з генетики такі як:

- Популяція - групи потенційних рішень (індивідів або хромосом), які еволюціонують разом під час кожного покоління. Кожна хромосома в популяції представляється генетичною структурою, яка кодує потенційне рішення задачі.
- Хромосоми (індивіди) - визначають генетичну структуру індивіда і кодує його характеристики або параметри. Аналогією з біологією, де хромосоми містять гени, які визначають властивості організму, в генетичних алгоритмах хромосома містить гени, що визначають характеристики індивіда.

- Ген - це атомарний елемент генотипу, він кодує конкретні властивості чи параметри індивіда. Наприклад, якщо ми вирішуємо задачу оптимізації, гени можуть кодувати значення параметрів, що піддаються оптимізації.
- Генотип (генетична структура) - складається з хромосом індивіда. Генотип включає всю генетичну інформацію, яка передається від батьків до нащадків через процеси схрещування і мутації. В залежності від конкретної задачі і кодування може використовуватися різноманітна репрезентація генотипу, така як бінарні рядки, числові послідовності, списки тощо. В контексті генетичних алгоритмів, оптимізації та еволюційних алгоритмів генотип часто служить основою для обчислення функції придатності.
- Фенотип - це набір значень, він визначає, наскільки добре даний індивід розв'язує конкретну задачу або відповідає поставленому завданню.
- Алель - вказує на конкретне значення або варіант гена, який знаходиться на певному локусі в генотипі організму чи індивіда.
- Локус - вказує на конкретне місце на хромосомі, де зберігається інформація про певний аспект індивіда. Кожен ген може бути розташований на своєму унікальному локусі.

Одним з найважливіших понять у генетичних алгоритмах вважається функція пристосованості (fitness function) або функцією оцінки. Ця функція є основним критерієм для визначення того, наскільки добре особина або рішення відповідає поставленій задачі.

Відповідно до еволюційного принципу виживання «найсильніших», функція пристосованості визначає, які рішення отримують більше шансів передати свої гени на наступне покоління.

Також функція пристосованості служить для кількісної оцінки того, наскільки ефективно розв'язок (особина) вирішує поставлену задачу.

У генетичних алгоритмах, як правило, максимізується функція пристосованості. На основі значень функції пристосованості здійснюється оцінка та відбір особин для створення нового покоління. Особини з більшими значеннями функції пристосованості мають більше шансів потрапити у нове покоління.

У задачах оптимізації функція пристосованості є цільовою функцією, яку генетичний алгоритм намагається максимізувати. У задачах мінімізації може бути використана змінена форма функції пристосованості.

На основі оцінки функції пристосованості генетичний алгоритм формує нове покоління, використовуючи операції схрещування, мутації та відбору. Функція пристосованості грає критичну роль у визначенні ефективності генетичних алгоритмів та їх здатності знаходження оптимальних рішень у просторі пошуку.

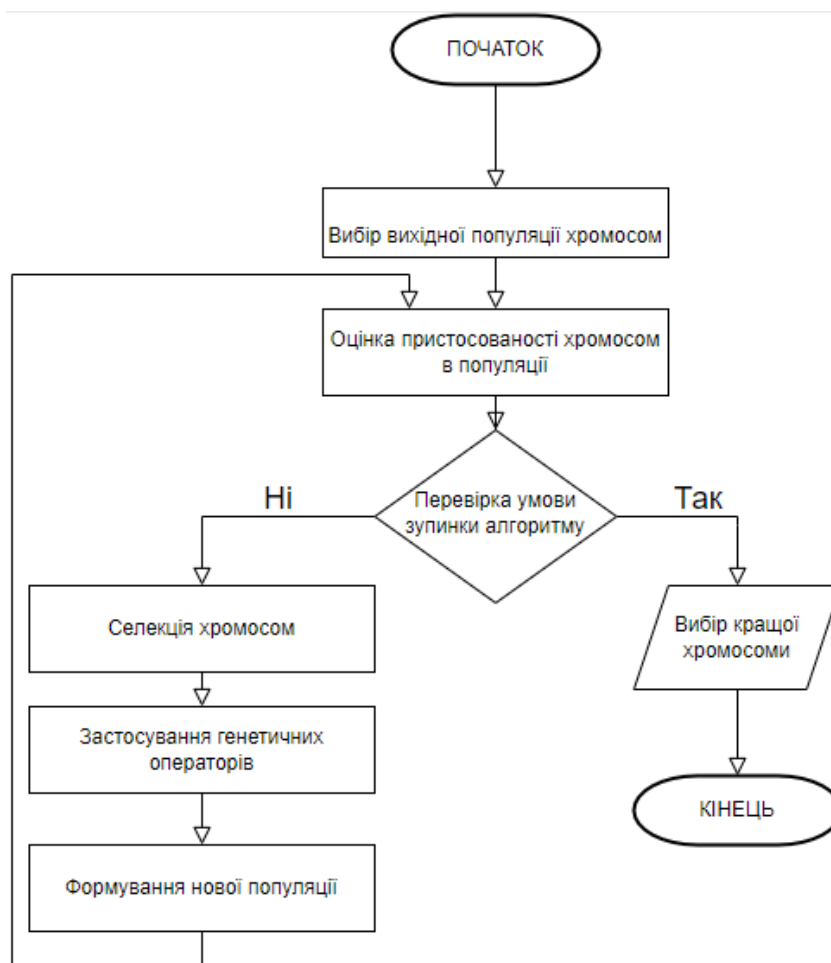


Рисунок 2.1 – Блок-схема генетичного алгоритму

Селекція (відбір): Цей етап визначає, які особини отримають можливість передати свої гени на наступне покоління, або які особини виживуть. Селекція сприяє щоб особини з кращими характеристиками мали більше впливу на наступне покоління.

Схрещування (кросовер): Цей етап відтворює ідею схрещування в природі, коли два батьки обмінюються своїми генетичними матеріалами для створення нащадка. Схрещування сприяє комбінації корисних характеристик батьків у нащадках, що сприяє покращенню поточного покоління.

Генетична інформація між батьками обмінюється в області, яка розташована від початку геному до вибраної точки схрещування. Якщо вибрана точка схрещування - це 3, то область до третього гена включно буде

обмінюватися між батьками на (рис. 2.2). Новостворені частини геному об'єднуються, утворюючи геном потомства. Таким чином, створюється нова особина, яка буде входити в наступне покоління.

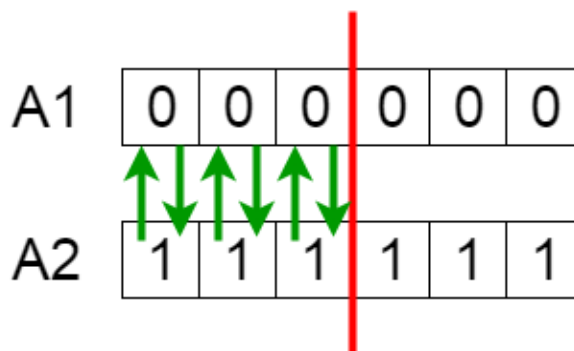


Рисунок 2.2 – Механізм кросинговеру

Мутація є операцією, яка застосовується до індивідуальних особин популяції з невеликою ймовірністю змінити їх генетичний матеріал. Це сприяє введенню нового генетичного різноманіття та дозволяє алгоритму досліджувати нові області простору пошуку.

Мутація гарантує, що в популяції буде введено нові генетичні елементи, навіть якщо обмін генами (кросовер) привів до створення подібних особин. Вона дозволяє алгоритму досліджувати нові області простору пошуку та уникати втрати різноманітності в популяції. (рис. 2.3)

Before Mutation

A5

1	1	1	0	0	0
---	---	---	---	---	---

After Mutation

A5

1	1	0	1	1	0
---	---	---	---	---	---

Рисунок 2.3 – Механізм мутації

Формування нового покоління: На цьому етапі створюється нове покоління, яке складається з нащадків, що виникли в результаті схрещування та відбору. Необхідний для забезпечення загальної якості популяції, збереження кращих характеристик та продовження процесу еволюції.[18]

Якщо функції пристосованості не досягло мети, ці кроки повторюються до тих пір поки результат функції пристосованості не дорівнюватиме шуканому або буде виконана одна з нижче перерахованих умов:

- Значення пройдених циклів досягне максимуму(визначається в умовах задачі)
- Вичерпаний час на мутацію

Етап створення нової популяції. На цьому етапі створюється Початкова популяція створюється шляхом випадкового вибору генетичних елементів для кожної особини. Це може включати в себе випадковий вибір значень для генів або використання алгоритмів для генерації. Особини створюються з урахуванням обмежень або "формату", який визначає, які генетичні елементи або комбінації генів є припустимими для створення ефективних рішень, а також особини в новій популяції обираються так, щоб брати участь в подальших етапах генетичного алгоритму, зокрема, у кроці розмноження. Це може включати в себе врахування пристосованості особин та їхню спроможність до створення ефективного нащадка.

Під час розмноження обираються два батьки з поточної популяції. Вибір може відбуватися за допомогою різних стратегій, таких як вибір осіб з вищою пристосованістю чи випадковий вибір. Гени від обох батьків перемішуються в нащадка, дозволяючи комбінувати різні риси та властивості. Зазвичай розмноження відбувається для всіх пар батьків, а не лише для тих, які "вижили". Це допомагає забезпечити більшу різноманітність в популяції та уникнути виокремлення одного "альфа самця" з перевагами.

Відбір. Тепер необхідно вибирати з популяції ту частку, яка “підє далі”. При цьому частку "виживших" після нашого відбору визначаємо заздалегідь вручну, вказуючи у вигляді параметра. Як не дивно, інші особини повинні загинути.[19]

2.2 Математична модель алгоритму

Стандартний ГА - це еволюційний алгоритм який застосовують для вирішення задач дискретної оптимізації для задач максимізації мінімізації значення функції $f(s)$ за умови, що $s \in \Omega = [0,1]^n$. Фітнес функція має вигляд $f: \Omega \rightarrow R$; $s \in \Omega$ - n -мірний двійковий вектор з дискретної множини Ω - хромосома з довжиною n . Дискретна множина $\Omega = [0,1]^n$ є множиною вершин гіперкуба з n вимірами кожне ребро якого дорівнює 1; R - це множина дійсних чисел.

Двійковий вектор-хромосома s кодує початковий змінний параметр V у двійковому вигляді, його фізичний зміст буде визначено деталями конкретної задачі. Кількість закодованих бітів(довжина хромосоми n), буде залежати від необхідної точності для визначення оптимального параметру V Число бітів при такім кодуванні, або довжина хромосоми n , залежить від необхідної точності визначення оптимальної величини параметра від необхідної точності визначення оптимальної величини параметра V яке зможе задовільнити умову

$$n \geq \log_2 \left(\frac{V_{max} - V_{min}}{\Delta} \right)$$

де V_{max} і V_{min} - мінімальне і максимальне значення параметра V ; Δ - задана погрішність визначення оптимального значення V .

Загальна схема роботи Стандартного ГА:

1. Визначаємо кількість пройдених циклів(епох) ГА $t=0$. Формуємо випадкову початкову популяцію (розміром M) $S(0) = \{s_1, s_2, \dots, s_m\}$

2. Для кожної із M істот розраховуємо пристосованість $f(s_i) = Fit(s_i), i = 1 \dots M$ і для усієї популяції $F_t = Fit(S_t)$.
3. З популяції $s_{c1} = Get(S_t)$ обрати істоту s_{c1} .
4. З деякою імовірністю P_c . обрати істоту з популяції $s_{c2} = Get(S_t)$ тепер виконуємо операцію кросингверу між двома істотами $s_c = Crossing(s_{c1}, s_{c2})$.
5. З деякою імовірністю P_m , виконуємо операцію мутації $s_m = Mutation(s_c)$.
6. З деякою імовірністю P_i , виконуємо операцію інверсії $s_{in} = Inversion(s_m)$
7. Отриману хромосому необхідно додати до нову популяції $Insert(S_{t+1}, s_{in})$.
8. Операції 3-7 повторюємо M разів.
9. Збільшуємо значення пройдених циклів $t = t + 1$.
10. За умови досягнення параметрів закінчення роботи: завершити цикл, у іншому випадку повторити кроки з 2. Кількість істот (M) в популяції впливає на ефективність кожної епохи (чим більше істот тим більше нових нащадків).

На практиці часто використовують нормативне подання функції придатності $f(s)$. Воно має вигляд $f: \Omega \rightarrow [0,1]$, його виводять в наслідок лінійного перетворення функції $f(s)$.

$$\bar{f}(s) = \frac{f(s) - f_{min}}{f_{max} - f_{min}}$$

де f_{max} і f_{min} - мінімальне і максимальне значення функції f .

У пунктах 3-4 можна використати такі схеми добору:

- a) пропорційний добір (fitness proportionate selection, FPS) $P(s_i) = \frac{f(s_i)}{\sum_{j=1}^M f(s_j)}$
- b) лінійне ранжирування (linear ranking)

$$P(s_i) = \frac{[\eta_{max} - (\eta_{max} - \eta_{min}) \cdot \frac{(i-1)}{(M-1)}]}{M} \text{ де } \eta_{min} = 2 - \eta_{max}, 1 \leq \eta_{max} \leq 2;$$

с) рівномірне ранжирування (uniform ranking)

$$P(s_i) = \begin{cases} \frac{1}{k}, 1 \leq i \leq k; \\ \prod_{0, k < i \leq M.} \end{cases}$$

Також можна використати елітний добір. Кращі індивіди, вимірювані за функцією пристосованості, автоматично переходять до наступного покоління без застосування кросовера чи мутації. Це забезпечує збереження високо-пристосованих хромосом. Однак використання елітного добору може призвести до занурення алгоритму в локальні мінімуми. Це тому, що кращі рішення мають тенденцію залишатися в популяції, а нові рішення не завжди отримують шанс вносити свій вклад через кросовер чи мутацію

Кросинговер діє в такий спосіб:

- вибір істот-предків. З популяції вибираються дві істоти-предки, які будуть використовуватися для створення нащадка.
- визначення точки розриву (точка схрещування) хромосом. Випадковим чином визначається точка розриву на хромосомі, де відбудеться кросовер.
- конкатенація частин хромосом. Частини хромосом істот-предків, які знаходяться до та після точки розриву, з'єднуються, утворюючи хромосому нащадка. Наприклад, якщо точка розриву знаходиться у середині хромосоми, перша половина хромосоми береться від одного предка, а друга половина - від іншого.
- випадковий вибір результуючої хромосоми. Випадковим чином вибирається одна з результуючих хромосом нащадка. Це може бути хромосома від одного з предків або обрана випадковим чином з двох конкатенованих хромосом.

На практиці кросинговер найчастіше проводять з ймовірністю $P_c=0,6$. Кросинговер може бути одноточковий та багатоточковий, коли точок розриву

декілька. Граничним випадком є рівномірний кросинговер, коли відповідно до визначеної ймовірності обмінюється кожна пара генів з двох хромосом.

Кросинговер балансує між використанням вже відомих областей з високим рівнем пристосованості так і дослідженням нових. Завдяки використанню кросинговеру та добору призводить до того, що хромосоми що мають більш середню пристосованість зустрічаються частіше на нових епохах.

Оператор мутації в генетичних алгоритмах використовується для введення випадкових змін в гени хромосоми, що сприяє розмаїтості та розширенню простору пошуку. Зазвичай ген замінюється із ймовірністю P_m . Це ймовірність того, що будь-який конкретний ген у хромосомі буде змінений в результаті мутації. Зазвичай визначається як величина від 0 до 1. Мутація сприяє додаванню випадковості в генетичний алгоритм, що робить його більш адаптивним та здатним до експлорації нових областей простору пошуку. Використання мутацій разом з кросовером дозволяє зберігати різноманітність в популяції та уникати застрягання в локальних мінімумах.

Зазвичай, $P_m \approx 0,001$

Використовується також оператор інверсії - різновид мутації, який діє так. Частина хромосоми (зазвичай одна з двох розділених частин) обертається на 180° . Це означає, що порядок генів в цій частині хромосоми змінюється на протилежний. Операція інверсії спрямована на впровадження розмаїтості в популяцію та дозволяє змінювати порядок генів в хромосомі. Операція інверсії допомагає в алгоритмах оптимізації рухатися по простору пошуку, якщо порядок генів важливий для досягнення оптимального розв'язку. Використання інверсії разом із звичайною мутацією та кросовером забезпечує додатковий рівень різноманітності в популяції та сприяє виявленню більш широкого спектру можливих рішень. Після завершення інверсії нові хромосоми формують нову популяцію S_{t+1} .

Важливий момент - визначають момент, коли алгоритм зупиняється. Це важливо для ефективного використання обчислювальних ресурсів та отримання

задовільного результату. Вони можуть використовуватися для обмеження максимальної кількості ітерацій або епох, щоб алгоритм зупиняється після визначеної максимальної кількості ітерацій або епох. Якщо пристосованість не змінюється значущим чином протягом певної кількості епох, можна вважати, що алгоритм зійшовся до оптимального розв'язку. Або алгоритм може зупинитися, коли популяція досягає певного рівня пристосованості, яке вважається задовільним.

Використання двійкового подання змінних є одним із способів кодування цілих чисел у генетичних алгоритмах. У цьому методі кожна змінна кодується у вигляді бінарної послідовності, або хромосоми, де кожен біт представляє один біт числа. Довжина хромосоми повинна бути достатньою для подання всіх можливих значень даної змінної.

Наприклад, якщо ви кодуєте ціле число від 0 до 15, вам потрібно 4 біти ($2^4 = 16$ можливих значень). Якщо числа можуть бути більшими, необхідно збільшити довжину хромосоми.

Один з можливих недоліків використання двійкового подання для кодування цілих чисел у генетичних алгоритмах. Цей недолік пов'язаний з тим, що сусідні числа можуть відрізнятися в значеннях декількох бітів, що може призводити до менш ефективного функціонування алгоритму та збільшувати час його збіжності. Щоб цього уникнути використовують код Грея. Його можна подати як двійковий з вагами розрядів $g_i = 2^i - 1$

Для кодування декількох змінних, використовують метод злиття (concatenation) двійкових подань кожної змінної в один вектор. Цей вектор потім розбивається на тетради (групи по 4 біта), і кожна тетрада конвертується відповідно до коду Грея.

При кодуванні декількох змінних необхідно провести кодування кожної у двійковий код, об'єднати отримані значення в один вектор і, розбиваючи його на тетради, перетворити за кодом Грея (див. табл. 2.1).

Метод кодування змінних, що приймають дійсні значення, називається кодуванням за зрізом (slice encoding) або методом розділення діапазону. Цей метод часто використовується для представлення змінних, що приймають значення з плаваючою точкою у генетичних алгоритмах.

Процедура кодування наступна:

- Поділ інтервалу: Розбити весь діапазон можливих значень змінної на підінтервали з необхідною точністю.
- Визначення номера інтервалу: Значення змінної кодується як номер підінтервалу, в середині якого знаходиться вихідне число. Це може бути просто ціле число, що вказує на номер підінтервалу.

Такий підхід дозволяє представити змінні з плаваючою точкою у формі цілих чисел та спростити їх обробку у генетичних алгоритмах.

Таблиця 2.1 – Відповідність кодів

Десятковий	Двійкові	Грея
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Для коду Грея декодувальна функція має вигляд:

$$V(s_r) = a + \frac{D(s_r)}{2^n - 1} (b - a)$$

де a і b - межі відрізка $[a, b]$, що містять значення змінної V , $D(s_r)$ - десяткове подання коду Грея. При кодуванні нечислових даних необхідно попередньо перетворити їх у числа.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Формування вихідних даних

Вихідні дані для тренажеру було сформовано за завданням з лабораторної роботи по генетичним алгоритмам дистанційного курсу “Інтелектуальні інформаційні системи”. Нижче наведено постановку задачі:

Для знаходження максимуму функції $f(x_1, x_2, x_3)$, де $x_i \in [\min i, \max i]$ було застосовано стандартний генетичний алгоритм.

При формуванні хромосоми область визначення кожної змінної поділяли на 2^n інтервали, значення змінної визначали як номер інтервалу (ціле число подане двійковим кодом). Після чого отримані значення об'єднували в один вектор, і розбиваючи його на тетради, перетворювали за кодом Грея.

Для двох хромосом C_1, C_2 , сформованих випадковим чином, провести кросинговер, мутацію та інверсію. Серед нових представників популяції обчислити найбільш пристосованого.

$$f = (3 - x_1 + x_3)x_2x_1 \in [0, 1.5], x_2 \in [-0.7, 0.8] x_3 \in [0.5, 2], \Delta x_i = 0.1$$

- Точка розриву при кросинговері знаходиться між 5 та 6 генами.
- Операція мутації виконується для гена 1 та 3.
- Операція інверсія виконується для частини хромосоми за 5 геном.

3.2 Короткий опис програмної реалізації

Задля забезпечення реалізації вимог до програмної реалізації було проаналізовано наступні програмні засоби (див табл. 3.1)

Таблиця 3.1 Порівняння інструментів програмної реалізації

Інструмент	Сфера використання	Переваги	Недоліки
HTML	Веб-розробка, структурування	Легко вивчається, підтримується усіма браузерами	Статична, не може створювати динамічні сторінки
CSS	Стилізація веб-сторінок	Простий, поліпшує швидкість завантаження сторінки	Проблеми з різнобраузерністю, плутанина з різними рівнями
JavaScript	Фронтенд та бекенд веб-розробка	Легкий, простий синтаксис, універсальний	Відсутність підтримки множинного успадкування, менша безпека
Python	Бекенд, наука про дані, веб-додатки	Легко вивчається, ефективна мова бекенду	Інтерпретована (повільна), не підходить для мобільних додатків
Java	Бекенд веб-розробка, ігри, мобільні додатки	Підтримка багатозадачності, масштабована, ефективно обробляє великі обсяги даних	Велике споживання пам'яті, повільніше порівняно з C/C+

Продовження Таблиці 3.1

TypeScript	Розширення JavaScript для великих проєктів	Засоби відлагодження, типобезпечність	Код потребує трансляції, додаткове вивчення
Objective C	Розробка для iOS	Гнучкість, динамічний типізація	Складно виявляти помилки, слабка підтримка ООП
C++	Загального призначення, підтримка ООП	Підтримка ООП, ефективна компіляція	Відсутність автоматичного збору сміття, складнощі з покажчиками
PHP	Бекенд веб-розробка	Легко вивчається, сумісна з основними ОС	Обробка помилок не відмінна, може бути повільною
C#	Бекенд розробка, ігри, мобільні додатки	Резервне копіювання пам'яті, багатий функціонал	Складно вправляти багатозадачні програми, менша гнучкість

Продовження Таблиці 3.1

Angular	Фронтенд веб-розробка	Мінімальний код, двостороннє зв'язування даних	Крута крива навчання, потребує підтримки JavaScript
Ruby on Rails	Бекенд веб-розробка	Графічні інструменти, дотримання веб-стандартів	Складно відлагоджувати, порівняно повільна
Golang	Ефективні веб-додатки, мережеве програмування	Вбудовані засоби тестування, функції безпеки пам'яті	Неефективне управління помилками, відсутність гнучкості
Scala	Рішення для аналізу даних	Підтримка паралельного програмування, підтримка IDE	Повільний вбудований компілятор, обмежена підтримка спільноти
Kotlin	Орієнтована на JVM, сумісна з Java	Зрозумілий синтаксис, безпека щодо значень null, розширені функції	М'яка крива навчання, більший розмір збірки, може відсутніти підтримка в деяких сценаріях

Програмна реалізація запропонованого алгоритму виконувалася мовою JavaScript. Для розробки окремих елементів проекту було застосовано HTML та CSS. HTML. Крім того виконувалася інтеграція даного віртуального тренажеру в систему дистанційного навчання СумДУ.

Основні функції, що використовувалися в роботі наведено в табл. 3.1. Повний код наведено в Додатку.

Таблиця 3.1 – Основні функції проекту

Функція	Опис
showHelpModal	Показує модальне вікно з підказкою для поточного питання.
startQuiz	Починає віртуальний тренажер, скидаючи індекс питання та рахунок. Показує перше питання.
showQuestion	Показує поточне питання разом з відповідями та таблицями, використовуючи buildTables.
showScore	Показує рахунок користувача та відгуки відповідей. Викликається при завершенні тренажеру.
resetState	Скидає стан кнопок відповідей (видаляє їх та додає нові), ховає кнопку "Далі".
selectAnswer	Обробляє вибір відповіді, встановлюючи класи для кольорового виділення правильних та неправильних відповідей, враховуючи коректність.
handleNextButton	Обробляє натискання на кнопку "Далі". Якщо є інші питання, викликає showQuestion, інакше - showScore.

3.3 Результати тестування додатку

При перевірці працездатності тренажера було проаналізовано його основні функції. Після активації тренажеру було відображено вікно з першим завданням варіантами відповіді та кнопкою допомоги. Перший етап перевіряє знання студента щодо особливостей виконання операції кросинговеру (Рис. 3.1а). Необхідно обрати вірний варіант відповіді на питання, якщо буде обрана правильна відповідь вона змінить колір на зелений(Рис. 3.1б) і на червоний якщо відповідь обрана не вірно (Рис. 3.1в). Перехід до другого етапу виконується при натисненні кнопки «Далі» яка з'являється після вибору одного з варіантів.

Help

Генетичний алгоритм

1. Обрати нові частини хромосом після операції односточкового кросингверу, якщо точка розриву при кросингвері знаходиться між 5 та 6 генами

Перша хромосома

1	0	0	1	0	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

Друга хромосома

1	1	1	1	0	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

0011000 0010110	0001110 0011100
-----------------	-----------------

0101011 0110110	0110111 0010111
-----------------	-----------------

а)

Help

Генетичний алгоритм

1. Обрати нові частини хромосом після операції односточкового кросингверу, якщо точка розриву при кросингвері знаходиться між 5 та 6 генами

Перша хромосома

1	0	0	1	0	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

Друга хромосома

1	1	1	1	0	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

0011000 0010110	0001110 0011100
-----------------	-----------------

0101011 0110110	0110111 0010111
-----------------	-----------------

Далі

б)

Help

Генетичний алгоритм

1. Обрати нові частини хромосом після операції односточкового кросингверу, якщо точка розриву при кросингвері знаходиться між 5 та 6 генами

Перша хромосома

1	0	0	1	0	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

Друга хромосома

1	1	1	1	0	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

0011000 0010110	0001110 0011100
-----------------	-----------------

0101011 0110110	0110111 0010111
-----------------	-----------------

Далі

в)

Рисунок 3.1 – Перше завдання «Кросингвер»: а) екран з тренажером, б) правильна відповідь користувача в) неправильна відповідь,

Другий етап та третій етап тестує вміння студента з операції мутація (Рис. 3.2 а) та інверсії (Рис. 3.2 б).

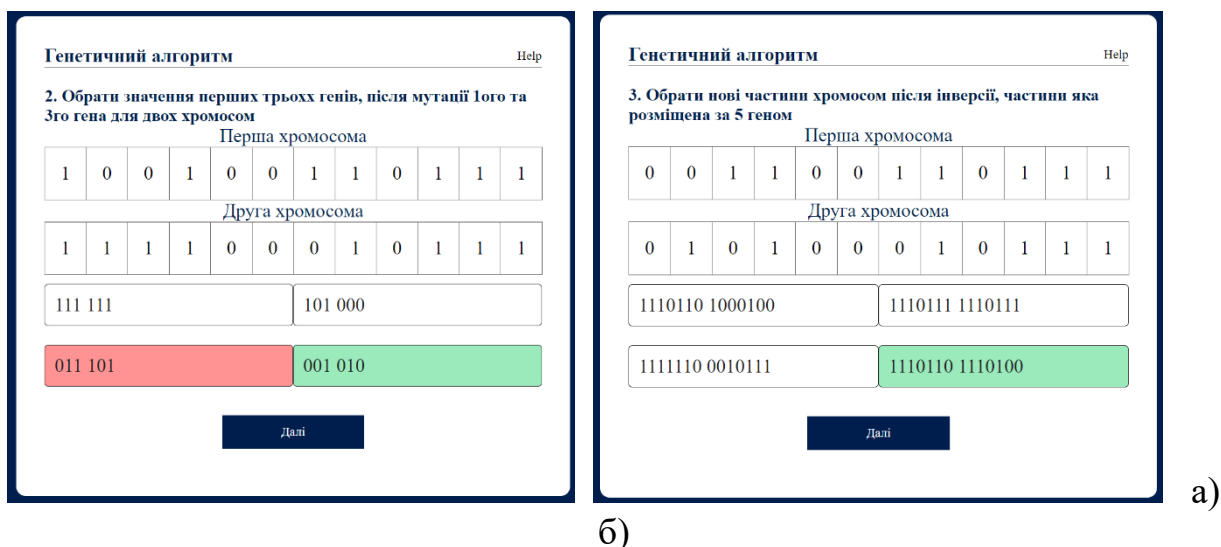


Рисунок 3.2 – Відображення етапів а) «Мутація» та б) «Інверсія»

Останніми перевіряються вміння декодувати хромосоми декодування сформованих хромосом (Рис. 3.3).

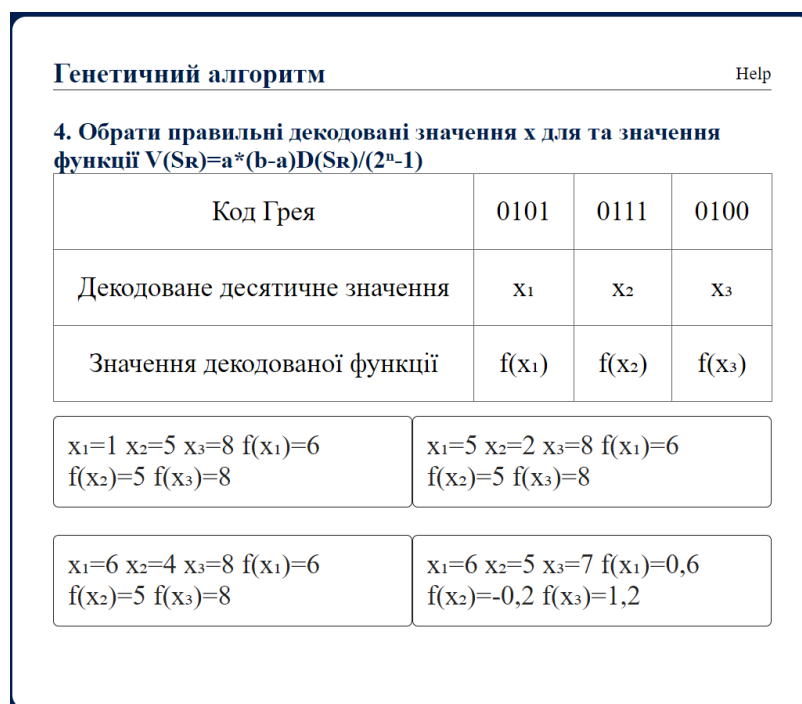


Рисунок 3.3 – Відображення етапу з декодування хромосоми

Для кожного питання тренажеру наведена унікальна підказка. Для її показу потрібно натиснути кнопку «help» (Рис. 3.4). У кінці тренажер показує кількість набраних балів за його проходження (Рис. 3.5).

Таким чином, тренажер перевіряє навички з усіх еволюційні механізми, що використовується в стандартному алгоритмі.

Кросингвер



Кросингвер діє в такий спосіб: з популяції вибираються дві істоти-предки; визначається випадковим чином точка розриву хромосом; хромосоми істот-нащадків визначаються як конкатенація частин хромосом першої та другої істоти-предка, потім випадковим чином вибирається одна з результуючих хромосом нащадків.

Got it!

Рисунок 3.4 – Відображення підказки

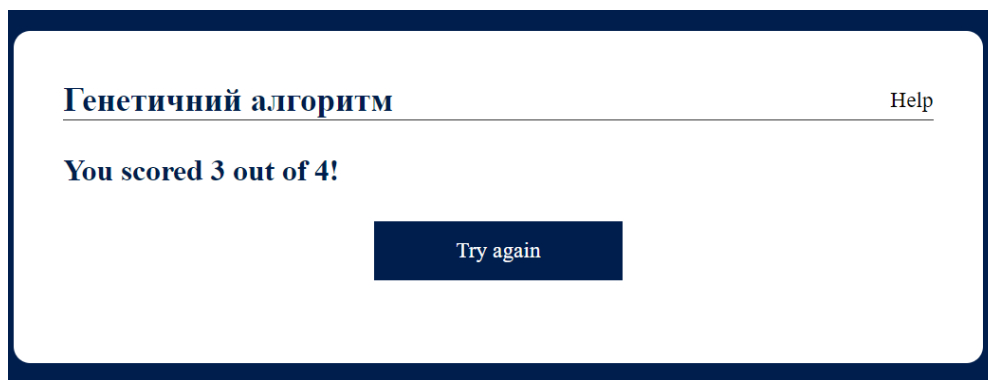


Рисунок 3.5 – Відображення результатів проходження тренажеру

ВИСНОВКИ

В кваліфікаційній магістерській роботі проведено програмну реалізацію тренажеру, що дозволяє перевірити знання операторів стандартного генетичного алгоритму. При цьому виконано такі основні завдання:

1. Сформовано вхідний математичний опис матеріалів тренажеру.
2. Сформовано математичну модель еволюційних механізмів, що використовується в стандартному алгоритмі.
3. Програмно реалізовано алгоритм роботи тренажеру, що перевіряє знання основних етапів застосування еволюційних механізмів стандартного генетичного алгоритму.
4. Перевірено працездатність розробленого тренажеру.

СПИСОК ЛІТЕРАТУРИ

1. The role of students' relational identity and autotelic experience for their innovative and continuous use of e-learning / M. Akbari та ін. Education and Information Technologies. 2022.
2. Amiti F. SYNCHRONOUS AND ASYNCHRONOUS E-LEARNING. European Journal of Open Education and E-learning Studies. 2020. Т. 5, № 2.
3. Clark R. C., Mayer R. E. E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning. – John Wiley & Sons, 2023.
4. Software Data Engineering for Network eLearning Environments / ред.: S. Caballé, J. Conesa. Cham : Springer International Publishing, 2018.
5. Shotarova S., Stoyanova-Petrova S. INTERACTIVE LEARNING IN ENGINEERING EDUCATION BASED ON LMS AND LTI. 15th International Conference on Education and New Learning Technologies, м. Palma, Spain, 3–4 лип. 2023 р. 2023.
6. Alshammari S. H., Bilal Ali M., Rosli M. S. LMS, CMS and LCMS: The confusion among them // Science International. – 2018. – Т. 30. – №. 3. – С. 455–459.
7. Alia A. A. H. The Analysis of a Learning Management System from a Design and Development Perspective. International Journal of Information and Education Technology. 2022. Т. 12, № 4. С. 280–289.
8. Alzahrani, A. A. (2019). The effect of distance learning delivery methods on student performance and perception. International Journal for Research in Education, 43(1), 12.
9. Turnbull D., Chugh R., Luck J. Learning Management Systems: An Overview. Encyclopedia of Education and Information Technologies. Cham, 2019. С. 1–7.
10. Firman F., Muhsin M. A., Goestina G. Online Based Learning Management System (LMS) on Student Academic Performance. AL-ISHLAH: Jurnal Pendidikan. 2021. Т. 13, № 1. С. 788–793.
11. Боднар Т. О. ПРИНЦИПИ ПОБУДОВИ ЕФЕКТИВНОГО ОНЛАЙН-НАВЧАННЯ. 2021. Т. 1, № 1. С. 52–57.

12. Гула Л. В. Впровадження інтерактивних технологій у вищих навчальних закладах. – 2019.
13. ДОСЛІДЖЕННЯ ЯКОСТІ ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ В УМОВАХ ВІЙНИ [Електронний ресурс] – Режим доступу до ресурсу: [yakist-osvity-v-umovah-viyny-web-3.pdf \(sqe.gov.ua\)](http://sqe.gov.ua/yakist-osvity-v-umovah-viyny-web-3.pdf).
14. Betthäuser B. A., Bach-Mortensen A. M., Engzell P. A systematic review and meta-analysis of the evidence on learning during the COVID-19 pandemic. *Nature Human Behaviour*. 2023.
15. Луцик Н. ВИКОРИСТАННЯ АКТИВНИХ ТА ІНТЕРАКТИВНИХ МЕТОДІВ НАВЧАННЯ У ДИСТАНЦІЙНІЙ ОСВІТІ ЗВО. MODALITĂȚI CONCEPTUALE DE DEZVOLTARE A ȘTIINȚEI MODERNE. 2020.
16. Cao J., Li L., Xie Y. Relationship between College Students' Engagement and Social Software Use. ICETC'20: 2020 12th International Conference on Education Technology and Computers, м. London United Kingdom. New York, NY, USA, 2020.
17. Mirjalili S. *Evolutionary Algorithms and Neural Networks*. Cham : Springer International Publishing, 2019.
18. Lambora A., Gupta K., Chopra K. Genetic Algorithm- A Literature Review. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), м. Faridabad, India, 14–16 лют. 2019 р.
19. Грицюк В. В. Порівняльний аналіз алгоритмів чисельного розв'язання задач нелінійного програмування. Генетичний алгоритм : master's thesis. 2020.
20. BEST PROGRAMMING LANGUAGES FOR WEB DEVELOPMENT [Електронний ресурс] – Режим доступу до ресурсу: [22 Best Web Development Languages to Learn in 2024 \(globalmediainsight.com\)](https://globalmediainsight.com/22-Best-Web-Development-Languages-to-Learn-in-2024/)

ДОДАТОК

```

<!DOCTYPE html>
<link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
<html>
<head>
  <title>Genetic Algorithm</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
<div class="app">
  <div class="modal__top">
    <div class="modal__title">Генетичний алгоритм</div>
    <button id="help-button" class="modal__close">Help</button>
  </div>
  <div class="quiz">
    <h1 id="question">?????</h1>

    <div id="tableOne">
      <table class="table">
        <p>Перша хромосома</p>
        <tbody id="tableOneBody">
        </tbody>
      </table>
    </div>

    <div id="tableTwo">
      <table class="table">
        <p>Друга хромосома</p>
        <tbody id="tableTwoBody">
        </tbody>
      </table>
    </div>

    <div id="grayTable", style="display: none;">
      <table class="table">
        <tbody id="grayTableBody">
        </tbody>
      </table>
    </div>

    <div id="answer-buttons">
      <div>
        <button class="btn">Answer 1</button>
        <button class="btn">Answer 2</button>
      </div>

```

```

        <div>
            <button class="btn">Answer 3</button>
            <button class="btn">Answer 4</button>
        </div>
    </div>
    <button id="next-btn">Наступне питання</button>
</div>
</div>
<script src="script.js"></script>
</body>
</html>

const questions = [
    {
        question: "Обрати нові частини хромосом після операції однострочкового кросинговеру, якщо точка розриву при кросинговері знаходиться між 5 та 6 генами",
        answers: [
            {text: "0011000 0010110", correct: false},
            {text: "0001110 0011100", correct: false},
            {text: "0101011 0110110", correct: false},
            {text: "0110111 0010111", correct: true},
        ]
    },
    {
        question: "Обрати значення перших трьох генів, після мутації 1ого та 3го гена для двох хромосом",
        answers: [
            {text: "111 111", correct: false},
            {text: "101 000", correct: false},
            {text: "011 101", correct: false},
            {text: "001 010", correct: true},
        ]
    },
    {
        question: "Обрати нові частини хромосом після інверсії, частини яка розміщена за 5 геном",
        answers: [
            {text: "1110110 1000100", correct: false},
            {text: "1110111 1110111", correct: false},
            {text: "1111110 0010111", correct: false},
            {text: "1110110 1110100", correct: true},
        ]
    },
    {
        question: "Обрати правильні декодовані значення x для та значення функції  $V(SR)=a*(b-a)D(SR)/(2^n-1)$ ",
    }
]

```

```

        answers: [
            {text: `x1=1 x2=5 x3=8\nf(x1)=6 f(x2)=5 f(x3)=8`, correct: false},
            {text: "x1=5 x2=2 x3=8\nf(x1)=6 f(x2)=5 f(x3)=8", correct: false},
            {text: "x1=6 x2=4 x3=8\nf(x1)=6 f(x2)=5 f(x3)=8", correct: false},
            {text: "x1=6 x2=5 x3=7\nf(x1)=0,6 f(x2)=-0,2 f(x3)=1,2", correct:
true},
        ]
    }
];

const clues = [
    {
        title: `Підказка№1`,
        help: `Для знаходження максимуму функції\n  $f(x_1, x_2, x_3) = (3 - x_1 + x_3)^{x_2}$ \n
+
        в області де  $x_1$  належить  $[0; 1.5]$ ,  $x_2$  належить  $[-0.7; 0.8]$ ,  $x_3$  належить
[0.5; 2] було застосовано\n`
    },
    {
        title: `Підказка№2`,
        help: "Обрати нові"
    },
    {
        title: `Підказка№3`,
        help: "Обрати нові частини "
    },
    {
        title: `Підказка№4`,
        help: "Обрати нові частини "
    }
];

const tablesData = [
    [1,0,0,1,0,0,0,1,0,1,1,1],
    [1,1,1,1,0,0,1,1,0,1,1,1],
    [1,0,0,1,0,0,1,1,0,1,1,1],
    [1,1,1,1,0,0,0,1,0,1,1,1],
    [0,0,1,1,0,0,1,1,0,1,1,1],
    [0,1,0,1,0,0,0,1,0,1,1,1],
]

const grayTablesData = [
    ["Код Грея", "0101", "0111", "0100"],
    ["Декодоване десятичне значення", "x1", "x2", "x3"],
    ["Значення декодованої функції", "f(x1)", "f(x2)", "f(x3)"],
]

```

```

const helpButton = document.getElementById("help-button");

const questionElement = document.getElementById("question");
const answerButtons = document.getElementById("answer-buttons");
const nextButton = document.getElementById("next-btn");
var tableOneBody = document.getElementById('tableOneBody');
var tableTwoBody = document.getElementById('tableTwoBody');
var grayTableBody = document.getElementById('grayTableBody');
var tableOne = document.getElementById('tableOne');
var tableTwo = document.getElementById('tableTwo');
var grayTable = document.getElementById('grayTable');

let currentQuestionIndex = 0;
let score = 0;

function showQuestion(){
  resetState();
  let currentQuestion = questions[currentQuestionIndex];
  let questionNo = currentQuestionIndex + 1;
  questionElement.innerHTML = questionNo + ". " + currentQuestion.question;

  turnTablesOn();
  buildTables(questionsData, currentQuestionIndex);

  currentQuestion.answers.forEach(answer => {
    const button = document.createElement("button");
    button.innerHTML = answer.text;
    button.classList.add("btn");
    answerButtons.appendChild(button);
    if(answer.correct) {
      button.dataset.correct = answer.correct;
    }
    button.addEventListener("click", selectAnswer)
  });
}

function showHelpModal(titleHtml, contentHtml, buttons) {
  const modal = document.createElement("div");

  modal.classList.add("modal");
  modal.innerHTML = `
    <div class="modal__inner">
      <div class="modal__top">
        <div class="modal__title">${titleHtml}</div>
        <button class="modal__close" type="button">
          <span class="material-icons">close</span>
        </button>

```

```

        </div>
        <div class="modal__content">${contentHtml}</div>
        <div class="modal__bottom"></div>
    </div>
`;

for (const button of buttons) {
    const element = document.createElement("button");

    element.setAttribute("type", "button");
    element.classList.add("modal__button");
    element.textContent = button.label;
    element.addEventListener("click", () => {
        if (button.triggerClose) {
            document.body.removeChild(modal);
        }

        button.onClick(modal);
    });

    modal.querySelector(".modal__bottom").appendChild(element);
}

modal.querySelector(".modal__close").addEventListener("click", () => {
    document.body.removeChild(modal);
});

document.body.appendChild(modal);
}

helpButton.addEventListener("click", ()=>{
    showHelpModal(clues[currentQuestionIndex].title,
clues[currentQuestionIndex].help, [
        {
            label: "Got it!",
            onClick: (modal) => {
                console.log("The button was clicked!");
            },
            triggerClose: true
        },
    ],
]);
})

function startQuiz(){
    currentQuestionIndex = 0;
    score = 0;
    nextButton.innerHTML = "Далі";
}

```

```
        showQuestion();
    }

function showQuestion(){
    resetState();
    let currentQuestion = questions[currentQuestionIndex];
    let questionNo = currentQuestionIndex + 1;
    questionElement.innerHTML = questionNo + ". " + currentQuestion.question;

    turnTablesOn()
    buildTables(currentQuestionIndex);

    currentQuestion.answers.forEach(answer => {
        const button = document.createElement("button");
        button.innerHTML = answer.text;
        button.classList.add("btn");
        answerButtons.appendChild(button);
        if(answer.correct) {
            button.dataset.correct = answer.correct;
        }
        button.addEventListener("click", selectAnswer)
    });
}

function turnTablesOff() {

    tableOne.style.display = "none";
    tableTwo.style.display = "none";
}

function turnTablesOn() {

    tableOne.style.display = "inline";
    tableTwo.style.display = "inline";
}

function turnGrayTableOn() {
    grayTable.style.display = "inline";
}

function turnGrayTableOff() {
    grayTable.style.display = "none";
}

function showScore(){
    turnGrayTableOff()
    turnTablesOff();
}
```

```

        resetState();
        questionElement.innerHTML = 'You scored ' + score + ' out of ' +
questions.length + '!';
        nextButton.innerHTML = "Try again";
        nextButton.style.display = "block";
    }

function resetState() {

    nextButton.style.display = "none";
    while(answerButtons.firstChild){
        answerButtons.removeChild(answerButtons.firstChild);
    }
}

function selectAnswer(e){
    const selectBtn = e.target;
    const isCorrect = selectBtn.dataset.correct === "true";
    if(isCorrect){
        selectBtn.classList.add("correct");
        score++;
    }else{
        selectBtn.classList.add("incorrect");
    }
    Array.from(answerButtons.children).forEach(button => {
        if(button.dataset.correct === "true"){
            button.classList.add("correct");
        }
        button.disabled = true;
    });
    nextButton.style.display = "block";
}

function handleNextButton(){
    currentQuestionIndex++;
    if(currentQuestionIndex < questions.length){
        showQuestion();
    }else{
        showScore();
    }
}

nextButton.addEventListener("click", ()=>{
    if(currentQuestionIndex < questions.length) {
        handleNextButton();
    }else{
        startQuiz();
    }
}

```



```

    }
  })

function buildTables(number){
  var row = ``;

  if (number < tablesData.length/2) {
    var tableId = number/2
    for(var i = 0; i < tablesData[2*number].length; i++){
      row += `<td>${tablesData[2*number][i]}</td>`
    }

    tableOneBody.innerHTML = `<tr>${row}</tr>`

    row = ``;
    for(var i = 0; i < tablesData[2*number+1].length; i++){
      row += `<td>${tablesData[2*number+1][i]}</td>`
    }

    tableTwoBody.innerHTML = `<tr>${row}</tr>`

  } else {
    turnTablesOff();
    turnGrayTableOn();
    grayTableBody.innerHTML = `<tr>${row}</tr>`;

    for(var i = 0; i < grayTablesData.length; i++){
      row=``;

      for(var j = 0; j < grayTablesData[i].length; j++){
        row += `<td>${grayTablesData[i][j]}</td>`
      }
      grayTableBody.innerHTML += `<tr>${row}</tr>`
    }

  }
}

startQuiz();

* {
  margin: 0;
  padding: 0;
  font-family: 'Poppins', 'sans-serif';
  box-sizing: border-box;

```

```
}
body{
    background:#001e4d;
}
.app{
    background: #fff;
    width: 90%;
    max-width: 600px;
    margin: 100px auto 0;
    border-radius: 10px;
    padding: 30px;
}
.quiz{
    padding: 20px 0;
}
.quiz h1{
    font-size: 18px;
    color: #001e4d;
    font-weight: 600;
}
.btn{
    display:inline-block;
    background: #fff;
    color: #222;
    font-size: 20px;
    font-weight: 500;
    width: 50%;
    border: 1px solid #222;
    padding: 10px;
    margin: 10px 0;
    text-align: left;
    border-radius: 4px;
    cursor: pointer;
    transition: all 0.3s;
}
.btn:hover:not([disabled]){
    background: #222;
    color: #fff;
}
.btn:disabled{
    cursor: no-drop;
}
#next-btn{
    background: #001e4d;
    color: #fff;
    font-weight: 500;
    width: 150px;
```

```
border: 0;
padding: 10px;
margin: 20px auto 0;
cursor: pointer;
display: none;
}
.correct{
  background: #9aeabc
}
.incorrect{
  background: #ff9393;
}
.table {
  border-collapse: collapse;
  width: 525px;
  font-size: 20px;
}
p{
  text-align: center;
  color: #001e4d;
  font-size: 20px;
}
td, th{
  border: 1px solid #666;
  padding: 16px;
  text-align: center;
}
.modal {
  --gap: 15px;

  position: fixed;
  top: 0;
  left: 0;
  width: 100vw;
  height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  box-sizing: border-box;
  padding: var(--gap);
  background: rgba(0, 0, 0, 0.5);
  font-family: sans-serif;
}

.modal__inner {
  background: #ffffff;
  width: 100%;
```

```
    max-width: 800px;
    overflow: hidden;
    border-radius: 4px;
}

.modal__top {
  display: flex;
  align-items: center;
  border-bottom: 1px solid #333;
}

.modal__title {
  font-size: 25px;
  color: #001e4d;
  font-weight: 600;
  padding-bottom: 30px;
  flex-grow: 1;
  padding: 0 var(--gap);
  font-size: 20px;
}

.modal__close {
  display: flex;
  align-items: center;
  cursor: pointer;
  padding: var(--gap);
  background: none;
  border: none;
  outline: none;
}

.modal__content {
  padding: 0 var(--gap);
  line-height: 1.5;
}

.modal__bottom {
  text-align: right;
  padding: 0 var(--gap) var(--gap) var(--gap);
}

.modal__button {
  display: inline-block;
  padding: 6px 12px;
  background: #001e4d;
  border: none;
  outline: none;
}
```

```

border-radius: 3px;
color: #ffffff;
cursor: pointer;
font-size: 18px;
}

.modal__button:not(:last-child) {
margin-right: var(--gap);
}

.modal__button:hover {
background:#222;
}
const questions = [
{
question: "Обрати нові частини хромосом після операції однострочкового кросинговеру, якщо точка розриву при кросинговері знаходиться між 5 та 6 генами",
answers: [
{text: "0011000 0010110", correct: false},
{text: "0001110 0011100", correct: false},
{text: "0101011 0110110", correct: false},
{text: "0110111 0010111", correct: true},
]
},
{
question: "Обрати значення перших трьохх генів, після мутації 1ого та 3го гена для двох хромосом",
answers: [
{text: "111 111", correct: false},
{text: "101 000", correct: false},
{text: "011 101", correct: false},
{text: "001 010", correct: true},
]
},
{
question: "Обрати нові частини хромосом після інверсії, частини яка розміщена за 5 геном",
answers: [
{text: "1110110 1000100", correct: false},
{text: "1110111 1110111", correct: false},
{text: "1111110 0010111", correct: false},
{text: "1110110 1110100", correct: true},
]
},
{
question: "Обрати правильні декодовані значення x для та значення функції  $V(SR)=a*(b-a)D(SR)/(2^n-1)$ ",

```

```

    answers: [
      {text: `x1=1 x2=5 x3=8\nf(x1)=6 f(x2)=5 f(x3)=8`, correct: false},
      {text: "x1=5 x2=2 x3=8\nf(x1)=6 f(x2)=5 f(x3)=8", correct: false},
      {text: "x1=6 x2=4 x3=8\nf(x1)=6 f(x2)=5 f(x3)=8", correct: false},
      {text: "x1=6 x2=5 x3=7\nf(x1)=0,6 f(x2)=-0,2 f(x3)=1,2", correct:
true},
    ]
  }
];

const clues = [
  {
    title: `Кросинговер`,
    help: `Кросинговер діє в такий спосіб:
з популяції вибираються дві істоти-предки;
визначається випадковим чином точка розриву хромосом;
хромосоми істот-нащадків визначаються як конкатенація частин хромосом
першої та другої істоти-предка,
потім випадковим чином вибирається одна з результуючих хромосом
нащадків.`
  },
  {
    title: `Мутація`,
    help: "Оператор мутації виконує випадкову зміну значення тобто для
кожного гена хромосоми його значення змінюється з 0 на 1 чи 1 на 0"
  },
  {
    title: `Інверсія`,
    help: "Інверсія це різновид мутації, який діє так, що хромосома
поділяється на дві частини і" +
" частина хромосоми після оберту на 180 градусів знов займає попереднє
положення"
  },
  {
    title: `Родуванні змінні`,
    help: `При кодуванні змінних, що подані числами з крапкою, що плаває
застосовують таку послідовність:
поділити весь інтервал припустимих значень змінної на ділянки з
необхідною точністю;
значення змінної визначають як номер інтервалу (ціле число), в середині
якого знаходиться число.
`
  }
];

const tablesData = [
  [1,0,0,1,0,0,0,1,0,1,1,1],

```

```

    [1,1,1,1,0,0,1,1,0,1,1,1],
    [1,0,0,1,0,0,1,1,0,1,1,1],
    [1,1,1,1,0,0,0,1,0,1,1,1],
    [0,0,1,1,0,0,1,1,0,1,1,1],
    [0,1,0,1,0,0,0,1,0,1,1,1],
  ]

const grayTablesData = [
  ["Код Грея", "0101", "0111", "0100"],
  ["Декодоване десятичне значення", "x1", "x2", "x3"],
  ["Значення декодованої функції", "f(x1)", "f(x2)", "f(x3)"],
]

const helpButton = document.getElementById("help-button");

const questionElement = document.getElementById("question");
const answerButtons = document.getElementById("answer-buttons");
const nextButton = document.getElementById("next-btn");
var tableOneBody = document.getElementById('tableOneBody');
var tableTwoBody = document.getElementById('tableTwoBody');
var grayTableBody = document.getElementById('grayTableBody');
var tableOne = document.getElementById('tableOne');
var tableTwo = document.getElementById('tableTwo');
var grayTable = document.getElementById('grayTable');

let currentQuestionIndex = 0;
let score = 0;

function showQuestion(){
  resetState();
  let currentQuestion = questions[currentQuestionIndex];
  let questionNo = currentQuestionIndex + 1;
  questionElement.innerHTML = questionNo + ". " + currentQuestion.question;

  turnTablesOn();
  buildTables(questionsData, currentQuestionIndex);

  currentQuestion.answers.forEach(answer => {
    const button = document.createElement("button");
    button.innerHTML = answer.text;
    button.classList.add("btn");
    answerButtons.appendChild(button);
    if(answer.correct) {
      button.dataset.correct = answer.correct;
    }
    button.addEventListener("click", selectAnswer)
  });
}

```

```

}

function showHelpModal(titleHtml, contentHtml, buttons) {
  const modal = document.createElement("div");

  modal.classList.add("modal");
  modal.innerHTML = `
    <div class="modal__inner">
      <div class="modal__top">
        <div class="modal__title">${titleHtml}</div>
        <button class="modal__close" type="button">
          <span class="material-icons">close</span>
        </button>
      </div>
      <div class="modal__content">${contentHtml}</div>
      <div class="modal__bottom"></div>
    </div>
  `;

  for (const button of buttons) {
    const element = document.createElement("button");

    element.setAttribute("type", "button");
    element.classList.add("modal__button");
    element.textContent = button.label;
    element.addEventListener("click", () => {
      if (button.triggerClose) {
        document.body.removeChild(modal);
      }

      button.onClick(modal);
    });

    modal.querySelector(".modal__bottom").appendChild(element);
  }

  modal.querySelector(".modal__close").addEventListener("click", () => {
    document.body.removeChild(modal);
  });

  document.body.appendChild(modal);
}

helpButton.addEventListener("click", ()=>{
  showHelpModal(clues[currentQuestionIndex].title,
clues[currentQuestionIndex].help, [
  {

```



```

        label: "Got it!",
        onClick: (modal) => {
            console.log("The button was clicked!");
        },
        triggerClose: true
    },
    ]);
})

function startQuiz(){
    currentQuestionIndex = 0;
    score = 0;
    nextButton.innerHTML = "Далі";
    showQuestion();
}

function showQuestion(){
    resetState();
    let currentQuestion = questions[currentQuestionIndex];
    let questionNo = currentQuestionIndex + 1;
    questionElement.innerHTML = questionNo + ". " + currentQuestion.question;

    turnTablesOn()
    buildTables(currentQuestionIndex);

    currentQuestion.answers.forEach(answer => {
        const button = document.createElement("button");
        button.innerHTML = answer.text;
        button.classList.add("btn");
        answerButtons.appendChild(button);
        if(answer.correct) {
            button.dataset.correct = answer.correct;
        }
        button.addEventListener("click", selectAnswer)
    });
}

function turnTablesOff() {

    tableOne.style.display = "none";
    tableTwo.style.display = "none";
}

function turnTablesOn() {

    tableOne.style.display = "inline";
    tableTwo.style.display = "inline";
}

```

```

}

function turnGrayTableOn() {
    grayTable.style.display = "inline";
}

function turnGrayTableOff() {
    grayTable.style.display = "none";
}

function showScore(){
    turnGrayTableOff()
    turnTablesOff();
    resetState();
    questionElement.innerHTML = 'You scored ' + score + ' out of ' +
questions.length + '!';
    nextButton.innerHTML = "Try again";
    nextButton.style.display = "block";
}

function resetState() {

    nextButton.style.display = "none";
    while(answerButtons.firstChild){
        answerButtons.removeChild(answerButtons.firstChild);
    }
}

function selectAnswer(e){
    const selectBtn = e.target;
    const isCorrect = selectBtn.dataset.correct === "true";
    if(isCorrect){
        selectBtn.classList.add("correct");
        score++;
    }else{
        selectBtn.classList.add("incorrect");
    }
    Array.from(answerButtons.children).forEach(button => {
        if(button.dataset.correct === "true"){
            button.classList.add("correct");
        }
        button.disabled = true;
    });
    nextButton.style.display = "block";
}

function handleNextButton(){

```

```

    currentQuestionIndex++;
    if(currentQuestionIndex < questions.length){
        showQuestion();
    }else{
        showScore();
    }
}

nextButton.addEventListener("click", ()=>{
    if(currentQuestionIndex < questions.length) {
        handleNextButton();
    }else{
        startQuiz();
    }
})

function buildTables(number){
    var row = ``;

    if (number < tablesData.length/2) {
        var tableId = number/2
        for(var i = 0; i < tablesData[2*number].length; i++){
            row += `<td>${tablesData[2*number][i]}</td>`
        }

        tableOneBody.innerHTML = `<tr>${row}</tr>`

        row = ``;
        for(var i = 0; i < tablesData[2*number+1].length; i++){
            row += `<td>${tablesData[2*number+1][i]}</td>`
        }

        tableTwoBody.innerHTML = `<tr>${row}</tr>`

    } else {
        turnTablesOff();
        turnGrayTableOn();
        grayTableBody.innerHTML = `<tr>${row}</tr>`;

        for(var i = 0; i < grayTablesData.length; i++){
            row=``;

            for(var j = 0; j < grayTablesData[i].length; j++){
                row += `<td>${grayTablesData[i][j]}</td>`
            }
            grayTableBody.innerHTML += `<tr>${row}</tr>`
        }
    }
}

```

```

    }
}

startQuiz();

<!DOCTYPE html>
<link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
<html>
<head>
  <title>Genetic Algorithm</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
<div class="app">
  <div class="modal_top">
    <div class="modal_title">Генетичний алгоритм</div>
    <button id="help-button" class="modal_close">Help</button>
  </div>
  <div class="quiz">
    <h1 id="question">?????</h1>

    <div id="tableOne">
      <table class="table">
        <p>Перша хромосома</p>
        <tbody id="tableOneBody">
        </tbody>
      </table>
    </div>

    <div id="tableTwo">
      <table class="table">
        <p>Друга хромосома</p>
        <tbody id="tableTwoBody">
        </tbody>
      </table>
    </div>

    <div id="grayTable", style="display: none;">
      <table class="table">
        <tbody id="grayTableBody">
        </tbody>
      </table>
    </div>

```

```
<div id="answer-buttons">
  <div>
    <button class="btn">Answer 1</button>
    <button class="btn">Answer 2</button>
  </div>
  <div>
    <button class="btn">Answer 3</button>
    <button class="btn">Answer 4</button>
  </div>
</div>
<button id="next-btn">Наступне питання</button>
</div>
</div>
<script src="script.js"></script>
</body>
</html>
```