

ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ
ДЛЯ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ УДАРСТІЙКИХ ПОВЕРХОНЬ

Сергієнко А.В.

канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, ORCID: <https://orcid.org/0000-0003-1328-2572>, sergienko_a_v@pstu.edu;

Волобуєв Є.С.

магістр, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, volobuiev_y_s@students.pstu.edu

У статті представлено дослідження застосування генетичних алгоритмів як інструменту автоматизованого проєктування ударостійких поверхонь. Актуальність роботи зумовлена потребою інженерних галузей у матеріалах і структурах, здатних витримувати інтенсивні механічні навантаження за умов обмежень щодо маси та вартості. У традиційних системах моделювання, що базуються на методі скінченних елементів або молекулярній динаміці, точність супроводжується високими обчислювальними витратами, що обмежує можливість швидкої оптимізації. У цій роботі показано, що еволюційні алгоритми можуть бути ефективним альтернативним або додатковим інструментом для дослідження великих просторових областей параметрів та знаходження оптимальних структур поверхонь. Для дослідження було розроблено комп'ютерний додаток на основі рушія Unity. У роботі використано два різні підходи: одноцільовий генетичний алгоритм (SGA), що орієнтований на максимізацію інтегрального коефіцієнта придатності, та багатоцільовий алгоритм недовідомованого генетичного сортування NSGA-II, здатний формувати фронт Парето для задач із суперечливими критеріями: мінімальною масою та мінімальною вартістю поверхні. Проведені симуляції включали моделювання зіткнення швидкісного об'єкта з поверхнями, виготовленими з різних матеріалів – сталі, алюмінію, вуглецевого волокна, пластику та дерева. Алгоритм SGA визначив найкращим матеріалом вуглецеве волокно, що забезпечило максимальний показник придатності завдяки поєднанню малої щільності та високої стійкості. Показано, що дерев'яні та пластикові поверхні, навіть за максимальної товщини, не здатні витримати удар, проте еволюційний алгоритм усе одно забезпечив відбір найкращих можливих рішень у межах обмежень. Алгоритм NSGA-II продемонстрував здатність формувати набір компромісних рішень та дозволив визначити альтернативні оптимальні варіанти залежно від пріоритетів – мінімальної вартості, мінімальної маси або збалансованої конфігурації. Порівняння обчислювальної ефективності показало, що швидкість NSGA-II значно падає при збільшенні кількості зразків, що є важливим фактором під час вибору алгоритму для практичного застосування. Результати роботи підтверджують, що генетичні алгоритми є потужним методом пошуку оптимальних конфігурацій ударостійких поверхонь у задачах фізичного моделювання. Отримані висновки можуть бути використані для подальшого вдосконалення систем автоматичного проєктування, інтеграції багатоцільової оптимізації та розширення до складніших реалістичних моделей удару.

Ключові слова: автоматичне проєктування, еволюційний алгоритм, фізичне моделювання, штучний інтелект.

Постановка проблеми

Проєктування та оптимізація ударостійких поверхонь становлять важливу сферу досліджень у багатьох інженерних дисциплінах, включаючи авіакосмічну, автомобільну, оборонну, цивільну інфраструктуру та біомедичні застосування. Здатність поверхні витримувати та розсіювати ударну енергію без катастрофічного руйнування є ключовою умовою забезпечення безпеки, довговічності та продуктивності. Такі традиційні підходи до моделювання ударостійких поверхонь, як метод скінченних елементів, молекулярна динаміка та континуальна механіка пошкоджень, широко застосовуються для прогнозування механізмів деформації, розподілу напружень та поведінки руйнування за умов динамічного навантаження [1]. Хоча ці фізично обґрунтовані моделі забезпечують цінні теоретичні та практичні висновки, їх застосування часто обмежується значною обчислювальною складністю.

В останні роки штучний інтелект (ШІ) постає перспективною парадигмою для вдосконалення моделювання ударостійких поверхонь. Зокрема, алгоритми

машинного навчання дають змогу виявляти нелінійні взаємозв'язки та приховані закономірності у великих наборах даних, що охоплюють властивості матеріалів, структурні конфігурації та результати експериментів. На відміну від традиційних методів, моделі, керовані ШІ, здатні швидко прогнозувати реакцію поверхні на удар у широкому діапазоні умов, водночас зменшуючи потребу в обчислювально затратних симуляціях і повторних фізичних випробуваннях [2]. Крім того, методи ШІ сприяють багатокритеріальній оптимізації, забезпечуючи систематичне дослідження простору проєктних рішень для досягнення бажаних компромісів між міцністю, вагою, вартістю та технологічністю [3].

Актуальність обраної теми зумовлена зростаючою потребою у матеріалах і структурах, здатних витримувати екстремальні механічні навантаження, залишаючись при цьому легкими, економічно ефективними та сталими. Сфера застосувань охоплює ключові галузі, такі як аерокосмічна промисловість, оборона, транспорт, цивільне будівництво та засоби індивідуального захисту, де відмова матеріалів під час удару

може мати серйозні наслідки для безпеки, економіки та суспільства.

Аналіз останніх досліджень та публікацій

При аналізі методів автоматизації моделювання ударостійких поверхонь за допомогою еволюційних алгоритмів можна виділити два можливі підходи:

Перший підхід – використання багатоцільових еволюційних алгоритмів. Ці алгоритми є особливо ефективними тоді, коли потрібно максимізувати ударостійкість, одночасно мінімізуючи такі параметри, як вага, вартість чи товщина. Серед них NSGA-II (Non-Dominated Sorting Genetic Algorithm II) є одним із найпоширеніших методів, відомим своєю стабільністю та здатністю формувати фронти Парето для аналізу компромісів [4]. Більш сучасний варіант, NSGA-III, краще підходить для задач багатоцільової оптимізації з більш ніж трьома-чотирма критеріями [5, 6]. Інший варіант, MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition), розкладає багатокритеріальну задачу на підзадачі, що робить його особливо придатним для великих і складних просторів проектування [7, 8].

Коли ж кілька критеріїв зводяться до однієї зваженої функції – наприклад, зваженої суми ударостійкості, ваги та вартості – ефективно застосовуються традиційні еволюційні алгоритми. Генетичні алгоритми (GA) є стандартним вибором для оптимізації шаруватих матеріалів, оскільки вони можуть працювати як з дискретними змінними (тип матеріалу), так і з неперервними (товщина). Диференціальна еволюція (DE), навпаки, є більш ефективною для неперервних змінних і добре підходить для оптимізації товщини шарів у широких діапазонах. Еволюційні стратегії, зокрема CMA-ES, також показують високу ефективність у неперервних просторах проектування, особливо коли кількість шарів є фіксованою, а основним параметром для оптимізації виступає розподіл товщини.

В цій роботі було використано два генетичних алгоритми: SGA та NSGA-II.

Одноцільовий генетичний алгоритм (SGA) – це метод оптимізації, натхненний процесом природного відбору та генетики, розроблений для пошуку оптимального або майже оптимального рішення задачі, визначеної однією цільовою функцією. SGA працює шляхом підтримки популяції кандидатів на рішення, кожне з яких представлено як хромосома, що кодує можливі значення змінних рішення. Шляхом ітеративного застосування операторів відбору, кросовера та мутації алгоритм еволюціонує популяцію до кращих рішень відносно цільової функції.

Процес починається з ініціалізації популяції, яка зазвичай генерується випадковим чином у межах можливого простору пошуку. Кожна особина в цій популяції оцінюється за допомогою цільової функції, яка кількісно визначає її придатність або якість як рішення проблеми. Потім механізм відбору ймовірно надає перевагу особинам з вищими значеннями придатності,

дозволяючи їм дати більше потомства наступному поколінню. Поширені методи відбору включають відбір за допомогою рулетки, турнірний відбір та відбір на основі рангів, кожен з яких по-різному балансує дослідження та експлуатацію.

Кросовер, або рекомбінація, застосовується до пар вибраних особин для отримання потомства, яке поєднує ознаки обох батьків. Цей оператор призначений для використання існуючих хороших рішень шляхом змішування їхнього генетичного матеріалу, створюючи таким чином нові рішення-кандидати, які можуть успадкувати корисні характеристики. Мутація, з іншого боку, вносить випадкові зміни в окремі хромосоми з низькою ймовірністю, забезпечуючи генетичну різноманітність і допомагаючи алгоритму уникнути локального оптимуму. Обидва оператори мають вирішальне значення для підтримки балансу між інтенсифікацією (пошук навколо перспективних областей) та диверсифікацією (дослідження нових областей простору пошуку).

Після генерації потомства вони формують нову популяцію для наступного покоління, і процес повторюється, доки не буде задоволено критерій завершення. Звичайні умови завершення включають досягнення заздалегідь визначеної кількості поколінь, досягнення задовільного об'єктивного значення або спостереження застою в покращенні популяції з часом. Кінцевий результат SGA зазвичай є найкращим рішенням, яке зустрічається під час еволюційного процесу.

Алгоритм генетичного сортування за недовомінованістю II (NSGA-II) є одним із найбільш поширених багатокритеріальних еволюційних алгоритмів. Він став одним із найвпливовіших підходів для розв'язання задач оптимізації, у яких необхідно враховувати кілька суперечливих критеріїв одночасно. Алгоритм особливо придатний для інженерних задач проектування, таких як оптимізація ударостійких поверхонь, де природно виникають компроміси між такими характеристиками, як поглинання енергії, вага та вартість.

Багатокритеріальні задачі оптимізації не мають єдиного оптимального розв'язку, натомість формують множину так званих Парето-оптимальних рішень. Розв'язок вважається Парето-оптимальним, якщо не існує іншого рішення, яке покращує один критерій без погіршення іншого. NSGA-II ефективно наближує цю фронту Парето, еволюціонуючи популяцію кандидатів рішень упродовж багатьох поколінь [9].

Ключові особливості NSGA-II:

- Сортування за недовомінованістю. Алгоритм розподіляє популяцію на різні фронти на основі відношення домінування. Перший фронт складається з недовомінованих рішень, другий – з тих, які домінуються лише рішеннями з першого фронту, і так далі. Це сортування гарантує, що рішення з кращим рангом мають вищий пріоритет під час відбору [10].

- Призначення відстані скупченості. Щоб зберегти різноманітність уздовж фронти Парето, NSGA-II вводить поняття відстані скупченості. Це міра

близькості рішення до його сусідів у просторі цілей. Рішення у менш скупчених регіонах мають перевагу, що сприяє рівномірному розподілу рішень уздовж фронти.

- Швидкий елітарний відбір. NSGA-II використовує елітарний підхід, поєднуючи батьківську та дочірню популяції перед відбором. Із цього об'єднаного набору індивіди сортується за рангом непомітності та відстанню скупченості, що гарантує збереження високоякісних рішень у процесі еволюції [11, 12].

Послідовність роботи NSGA-II відповідає типовому еволюційному алгоритму з певними вдосконаленнями для багатокритеріальної оптимізації:

- Ініціалізація. Генерується початкова популяція рішень, зазвичай випадковим чином. Кожен індивід кодує можливу конфігурацію змінних проєктування (наприклад, товщину шарів та вибір матеріалів).

- Оцінювання. Кожен кандидат оцінюється відповідно до визначених цілей. Для ударостійких поверхонь це можуть бути результати симуляцій для поглинання енергії, ваги та вартості.

- Відбір. Індивіди відбираються для відтворення на основі їхнього рангу непомітності та відстані скупченості, з перевагою для тих, які ближчі до фронти Парето й розташовані в менш насичених областях.

- Варіація. До вибраних рішень застосовуються генетичні оператори, такі як кросовер і мутація, для створення нащадків.

- Комбінація та сортування. Батьківська й дочірня популяції об'єднуються в єдиний пул. Застосовуються сортування за непомітністю та обчислення відстані скупченості для формування нового покоління [13].

- Ітерація. Процес повторюється доти, доки не буде досягнуто критерію зупинки, зазвичай максимальної кількості поколінь або збіжності фронту Парето.

Мета статті

Метою статті є аналіз та вдосконалення сучасних алгоритмів моделювання ударостійких поверхонь з використанням еволюційних алгоритмів. Досягнення поставленої мети передбачає вирішення комплексу взаємопов'язаних завдань. Зокрема, необхідно здійснити аналіз існуючих методів та алгоритмів моделювання ударостійких поверхонь, дослідити теоретичні основи еволюційних алгоритмів, розробити алгоритм моделювання ударостійких поверхонь з використанням еволюційних алгоритмів, провести експериментальну перевірку алгоритму на результатах фізичних симуляцій, а також оцінити ефективність запропонованого підходу та порівняти його з існуючими методами.

Матеріали та методи

Об'єкт дослідження – процес моделювання ударостійких поверхонь.

Предметом дослідження є алгоритми моделювання ударостійких поверхонь за допомогою еволюційних алгоритмів.

Виклад основного матеріалу

Для реалізації експерименту, за допомогою кросплатформового ігрового рушія Unity, було розроблено додаток, який включає наступні компоненти:

1. Головне меню (рис. 1) – дозволяє зайти в меню налаштувань та запустити обидва еволюційні алгоритми;

2. Графічний інтерфейс користувача (рис. 2) – забезпечує відображення інформації про кожний зразок симуляції, дозволяє керувати швидкістю візуалізації симуляції, дозволяє обрати яке з поколінь симуляцій візуалізувати, інформує про прогрес симуляції, дає користувачу поради щодо кнопок, якими можна контролювати візуалізацію симуляцій;



Рис. 1 – Головне меню



Рис. 2 – Графічний інтерфейс користувача

3. Візуалізатор симуляцій – дозволяє користувачам побачити абстрактну репрезентацію процесу зіткнення швидкісного об'єкту та ударостійкої поверхні для кожної окремої симуляції кожного покоління;

4. Менеджер еволюції – виконує розрахунки необхідні для симуляції кожного зіткнення та самого процесу еволюції, визначає характеристики матеріалів ударостійких поверхонь, зберігає інформацію про кожне покоління етап еволюції для подальшої візуалізації за допомогою візуалізатора симуляцій.

Робота симулятором виконується поетапно:

- 1) запустити додаток;
- 2) обрати бажаний алгоритм;
- 3) почекати поки симуляція завершиться;

4) переглянути результати симуляції у візуалізаторі або графіку;

5) за допомогою клавіш «N» та «P» обрати симуляцію для відображення або запустити автоматичний перегляд усіх симуляцій покоління по черзі за допомогою клавіші «Space»;

6. У графічному інтерфейсі користувача, у контрольній панелі справа знизу, обрати бажану швидкість візуалізації;

7. У контрольній панелі обрати бажане покоління для аналізу процесу еволюції. Це можна зробити, написавши номер покоління у полі вводу або натискаючи на стрілки для переходу на наступне або попереднє покоління;

8. Для завершення роботи потрібно закрити вікно програми.

Використовувались наступне технічне забезпечення:

- обчислювальні засоби – комп'ютер з операційною системою Windows 10 або Windows 11, з процесором 64-бітної архітектури та підтримкою інструкцій SSE2, графічний процесор, що підтримує графічний API DX11 та 8 ГБ оперативної пам'яті;

- засоби введення інформації – клавіатура та миша;

- засоби відображення інформації – монітор.

Для проведення експериментів було підготовлено 5 матеріалів для генерації ударостійких поверхонь (табл. 1). Усі зазначені характеристики вимірюються у абстрактних відносних одиницях, які можуть давати очікувані результати лише в контексті даного комп'ютерного додатку, розробленого з використанням рушія Unity

Таблиця 1
Характеристики використаних матеріалів

	Щільність	Стійкість	Ціна за одиницю ваги
Алюміній	2700	5	2
Вуглецеве волокно	1600	10	6
Пластик	950	2	1
Сталь	7850	8	3.5
Дерево	600	1	0.5

Додаткові характеристики симуляції: маса швидкісного об'єкту – 50, гравітація – 9,81, висота падіння – 5, кількість поколінь еволюції – 200, кількість зразків у кожному поколінні – 100, мінімальна товщина поверхні – 0.001, максимальна товщина поверхні – 0,1 м.

Коефіцієнт придатності зразка, що вистояв зіткнення, розраховується за формулою 1:

$$fitness = \frac{1}{1+weight*0.5+cost*0.5} \quad (1)$$

де fitness – коефіцієнт придатності;

weight – вага поверхні;

cost – вартість поверхні.

Коефіцієнт придатності зразка, що не вистояв зіткнення зі швидкісним об'єктом та зламався, розраховується за формулою 2:

$$fitness = \frac{0.05 * (\frac{resistance}{impactStrength})}{1+weight+cost} \quad (2)$$

де fitness – коефіцієнт придатності;

weight – вага поверхні;

cost – вартість поверхні;

resistance – стійкість поверхні до зіткнень;

impactStrength – сила зіткнення.

В результаті еволюції було отримано зразок (рис. 3) з найкращою придатністю з наступними характеристиками:

- матеріал: вуглецеве волокно;
- товщина поверхні: 0,0245 м;
- придатність: 0,0072.

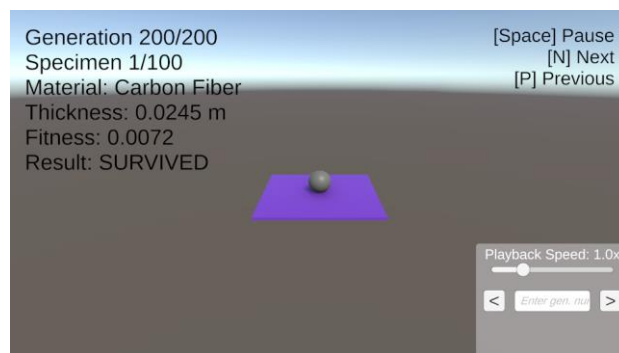


Рис. 3 – Вуглецеве волокно, найпридатніший зразок

Результат цієї симуляції показує, що незважаючи на високу вартість, вуглецеве волокно вигідно використовувати тому, що його висока стійкість зменшує необхідну товщину поверхні, що, у свою чергу, зменшує масу поверхні. Менша маса означає, що треба закупити менше матеріалу, тому витрачається менше грошей.

Другий по придатності матеріал це алюміній. Найпридатніший зразок (рис. 4), що використовує алюміній має товщину поверхні: 0,0491 м та придатність 0,0050.

Вартість алюмінію нижча за вартість вуглецевого волокна у 3 рази, але й стійкість нижча у 2 рази. Вуглецеве волокно також перевершує алюміній в щільності. Це приводить до того, що на кожен одиницю товщини поверхні приходить більше маси матеріалу. Показник придатності відображає цю негативну залежність у вигляді меншої цифри ніж у поверхні з вуглецевого волокна.

Третій по придатності матеріал поверхні це сталь. Найпридатніший зразок зі сталі (рис. 5) має товщину поверхні 0,0307 м та придатність: 0,0018.

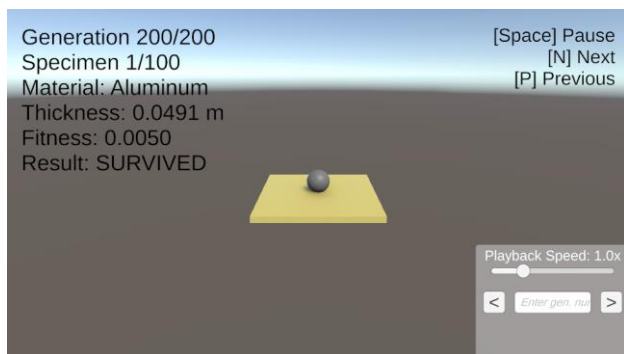


Рис. 4 – Алюміній, найпридатніший зразок

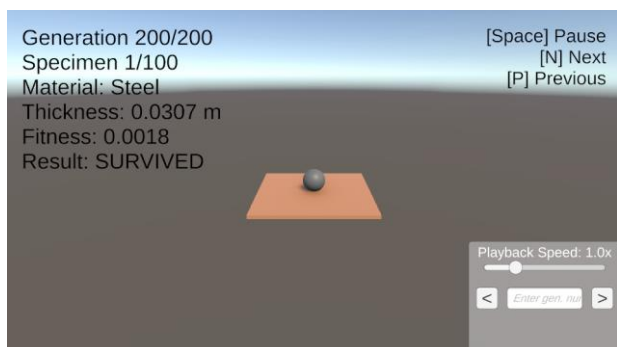


Рис. 5 – Сталь, найпридатніший зразок

Ціна сталі вище за ціну алюмінію, але це не головна причина різкого зменшення максимальної придатності. Щільність сталі більше за щільність алюмінію майже у 3 рази, що значно перевершує збільшення стійкості, яка є більшою всього в 1,6 рази. Більша стійкість дає змогу робити поверхні з меншою товщиною, але значно підвищена щільність скасовує цю перевагу, роблячи вагу та ціну поверхні вище. Функція придатності описала ці компроміси, обчисливши значно меншу цифру.

Четвертий по придатності матеріал поверхні це дерево. Важливо зазначити, що при наявних лімітах товщини (максимальна товщина – 0,1), усі поверхні з дерева та пластику провалюють симуляцію, ламаючись при зіткненні, але еволюційний алгоритм все ще намагається максимізувати значення придатності. Найпридатніший зразок з дерева (рис. 6) має товщину поверхні 0,1000 м та придатність: 0,000224.

Дерево має найменші показники ціни та щільності, але це також стосується показника стійкості. Розрахована мінімальна стійкість для успішного витримування зіткнення при максимальній товщині поверхні – 2,4535. Формула придатності для невдалих поверхонь нагороджує за те, наскільки близькою була поверхня до успішного витримування зіткнення, тому зразки з мінімальною товщиною (для мінімізації ціни) були породжені у ході еволюції.

Останній по придатності матеріал поверхні це пластик. Найпридатніший зразок з пластику (рис. 7)

має товщину поверхні 0,1000 м та придатність: 0,000213.

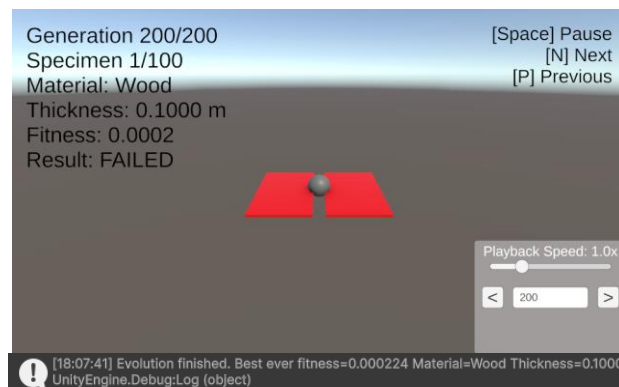


Рис. 6 – Дерево, найпридатніший зразок

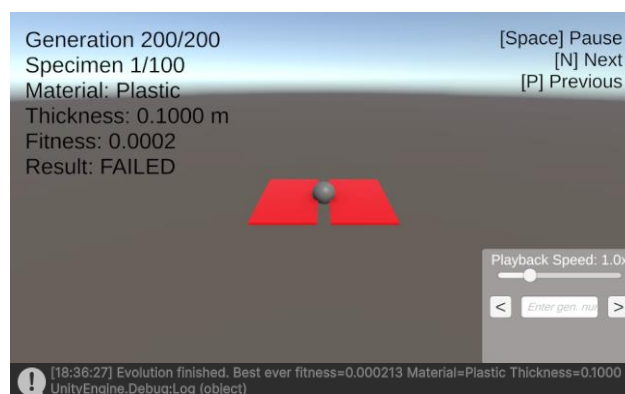


Рис. 7 – Пластик, найпридатніший зразок

Маючи більші за дерево ціну та щільність, пластик має трохи нижчу придатність. Нагороди за близькість до успішного витримування зіткнення було недостатньо для перевернення дерева за значенням придатності.

Підсумки проведених експериментів демонструють здатність генетичних алгоритмів знаходити оптимальні комбінації матеріалів та товщини ударостійкої поверхні з оглядом на декілька критеріїв оцінки одночасно. В даних експериментах було використано спрощену абстрактну фізичну модель, але універсальність генетичних алгоритмів дає змогу використовувати цей підхід у більш складних симуляціях. Слід зазначити, що одноцільовий алгоритм штовхає еволюцію к одному єдиному «найкращому» рішення, яке базується на вагах, призначених кожному критерію оцінки у функції придатності. У реальних випадках частіше не існує «універсального найкращого рішення», тож приходится шукати множину оптимальних рішень, фронт Парето, який можна аналізувати та знаходити потрібні компроміси. Наявний елітарний одноцільовий генетичний алгоритм не може цього робити. Для цієї задачі потрібно адаптувати концепцію вирішення проблеми моделювання ударостійких поверхонь за допомогою

еволюційних алгоритмів під використання багатоцільових алгоритмів, таких як NSGA-II, NSGA-III та MOEA/D.

При використанні алгоритму NSGA-II генерується множина результатів (фронт Парето), аналізуючи які можна обрати бажане компромісне рішення.

Результат роботи NSGA-II можна побачити на рисунку 8.

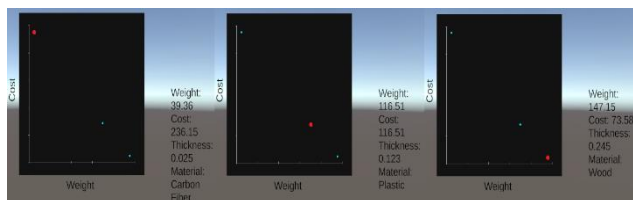


Рис. 8 – Результат роботи NSGA-II

Було знайдено три рішення: два для граничних значень ціни та ваги та одне для однакових значень. У цьому випадку, залежно від потреб, можна обирати рішення: чи потрібна дешева поверхня, чи потрібна легка поверхня, або щось середнє.

Важливо зазначити, що алгоритм NSGA-II значно повільніший за SGA при великій кількості зразків. Порівняння швидкості роботи обох алгоритмів можна побачити у таблиці 2.

Таблиця 2
Порівняння швидкості алгоритмів SGA та NSGA-II

	SGA	NSGA-II
50 зразків	1,46 сек	1,47 сек
100 зразків	1,46 сек	3,05 сек
150 зразків	1,46 сек	4,93 сек
200 зразків	1,46 сек	7,25 сек
300 зразків	1,46 сек	12,06 сек
500 зразків	1,47 сек	25,68 сек

Як можна побачити, кількість зразків майже не впливає на швидкість роботи алгоритму SGA, тоді як швидкість роботи NSGA-II прогресивно падає зі збільшенням кількості зразків. При числі зразків 500, NSGA-II потребує приблизно в 17 разів більше часу ніж SGA, це дуже важливе врахування при виборі між двома алгоритмами.

Підсумовуючи різницю між одноцільовим алгоритмом SGA та багатоцільовим алгоритмом NSGA-II, можна сказати, що SGA краще використовувати коли потрібно знайти лише одне компромісне рішення, описане функцією придатності. Це підходить для задач, де не існує потреби в порівнянні багатьох рішень та бажані характеристики добре відомі. Якщо ж задача потребує більшої гнучкості у виборі – алгоритм NSGA-II може забезпечити користувача множиною можливих рішень, серед яких користувач може обирати потрібне компромісне рішення залежно від поставленої задачі та обмежень по ресурсам.

Алгоритм NSGA-II не слід використовувати при великій кількості зразків, коли швидкість роботи алгоритму є важливим міркуванням.

Висновки

В результаті дослідження було встановлено, що еволюційні алгоритми здатні вирішувати широкий спектр задач, пов'язаних з фізичною симуляцією, одна з яких це автоматичне моделювання ударостійких поверхонь та оптимізація їх ціни та ваги шляхом пошуку оптимальних комбінацій матеріалів та товщини.

В ході роботи було розроблено комп'ютерну програму, яка симулює процес еволюції та абстрактну фізичну модель зіткнення поверхні з швидкісним об'єктом, графічно представляє результати еволюційного процесу.

Також було порівняно два типи еволюційних алгоритмів (одноцільовий та багатоцільовий) на прикладі алгоритмів SGA та NSGA-II. При порівнянні було виявлено важливе обмеження швидкості роботи алгоритму NSGA-II при великій кількості зразків.

Перелік використаних джерел

- [1] Bheel N., Mohammed B. S., Woen E. L. Modelling and optimizing the impact resistance of engineered cementitious composites with Multiwalled carbon nanotubes using response surface methodology. *Scientific Reports*. 2024. Vol. 14, no. 1. Article 24107. DOI: <https://doi.org/10.1038/s41598-024-75641-5>.
- [2] Ben-Zion Y., Einhorn Zarzecki R., Glazer J., Finkelstein N. D. Leveraging AI for Rapid Generation of Physics Simulations in Education: Building Your Own Virtual Lab. *The Physics Teacher*. 2025. Vol. 63, no. 6. Pp. 424-427. DOI: <https://doi.org/10.1119/5.0252343>.
- [3] Slowik A., Kwasnicka H. Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*. 2020. Vol. 32, no. 16. Pp. 12363-12379. DOI: <https://doi.org/10.1007/s00521-020-04832-8>.
- [4] Zhao Z., Liu B., Zhang C., Liu H. An improved adaptive NSGA-II with multi-population algorithm. *Applied Intelligence*. 2018. Vol. 49, no. 2. Pp. 569-580. DOI: <https://doi.org/10.1007/s10489-018-1263-6>.
- [5] Pérowski A., Ben-Hamida S. Multi-objective Optimization. *Evolutionary Algorithms*. Hoboken, NJ, USA, 2017. Pp. 165-182. DOI: <https://doi.org/10.1002/9781119136378.ch5>.
- [6] Yuan Y., Xu H., Wang B. An improved NSGA-III procedure for evolutionary many-objective optimization. *Genetic and Evolutionary Computation Conference: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, Vancouver, Canada, 12-16 July 2014*. New York, USA, 2014. Pp. 661-668. DOI: <https://doi.org/10.1145/2576768.2598342>.

- [7] Whitney W., Rana S., Dzubera J., Mathias K. E. Evaluating evolutionary algorithms. *Artificial Intelligence*. 1996. Vol. 84, no. 1-2. Pp. 357-358. DOI: [https://doi.org/10.1016/0004-3702\(96\)81371-3](https://doi.org/10.1016/0004-3702(96)81371-3).
- [8] Scardua L. A. Multiobjective Evolutionary Algorithm Based on Decomposition. *Applied Evolutionary Algorithms for Engineers Using Python*. 2021. Pp. 149-158. DOI: <https://doi.org/10.1201/9780429298028-12>.
- [9] Deb K., Pratap A., Agarwal S., Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 2002. Vol. 6, no. 2. Pp. 182-197. DOI: <https://doi.org/10.1109/4235.996017>.
- [10] Deb K., Jain H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*. 2014. Vol. 18, no. 4. Pp. 577-601. DOI: <https://doi.org/10.1109/tevc.2013.2281535>.
- [11] Keller A. A. Metaheuristic Optimization Algorithms. *Multi-Objective Optimization In Theory and Practice II: Metaheuristic Algorithms*. 2019. P. 54-83. DOI: <https://doi.org/10.2174/9781681087054119010004>.
- [12] Domínguez J., Montiel O., Sepúlveda R., Medina N. High Performance Architecture for NSGA-II. *Recent Advances on Hybrid Intelligent Systems*. Berlin, Heidelberg, 2013. Pp. 451-461. DOI: https://doi.org/10.1007/978-3-642-33021-6_35.
- [13] Scardua L. A. Non-Dominated Sorted Genetic Algorithm II. *Applied Evolutionary Algorithms for Engineers Using Python*. 2021. Pp. 137-148. DOI: <https://doi.org/10.1201/9780429298028-11>.

RESEARCH INTO THE USE OF GENETIC ALGORITHMS FOR AUTOMATED DESIGN OF IMPACT-RESISTANT SURFACES

Sergienko A.V.	PhD (Engineering), associate professor, SHEI «Priazovskyi state technical university», Dnipro, https://orcid.org/0000-0003-1328-2572 , e-mail: sergienko_a_v@pstu.edu ; ORCID:
Volobuev Y.S.	master's student, SHEI «Priazovskyi state technical university», Dnipro, e-mail: volobuev_y_s@students.pstu.edu

The article presents a study on the use of genetic algorithms as a tool for automated design of impact-resistant surfaces. The relevance of the work is driven by the need in engineering fields for materials and structures capable of withstanding intense mechanical loads under constraints on mass and cost. In traditional modelling systems based on the finite element method or molecular dynamics, accuracy is accompanied by high computational costs, which limits the possibilities for rapid optimization. This study demonstrates that evolutionary algorithms can serve as an effective alternative or complementary tool for exploring large parameter spaces and finding optimal surface structures. For the research, a computer application was developed using the Unity engine. Two different approaches were employed: a single-objective genetic algorithm (SGA), aimed at maximizing the integral fitness coefficient, and the multi-objective Non-Dominated Sorting Genetic Algorithm II (NSGA-II), capable of forming a Pareto front for problems with conflicting criteria: minimal mass and minimal surface cost. The simulations included modelling collisions of a high-velocity object with surfaces made from various materials – steel, aluminum, carbon fiber, plastic, and wood. The SGA algorithm identified carbon fiber as the best material, providing the highest fitness score due to its combination of low density and high resistance. It was shown that wooden and plastic surfaces, even at maximum thickness, cannot withstand the impact; however, the evolutionary algorithm still ensured the selection of the best possible solutions within the given constraints. The NSGA-II algorithm demonstrated the ability to generate a set of compromise solutions and made it possible to identify alternative optimal options depending on priorities – minimal cost, minimal mass, or a balanced configuration. A comparison of computational efficiency showed that the performance of NSGA-II declines significantly as the number of samples increases, which is an important factor when choosing an algorithm for practical applications. The results confirm that genetic algorithms are a powerful method for searching optimal configurations of impact-resistant surfaces in physical modelling tasks. The findings can be used for further improvement of automated design systems, integration of multi-objective optimization, and extension to more complex and realistic impact models.

Keywords: automatic design, evolutionary algorithm, physical modeling, artificial intelligence.

References

- [1] N. Bheel, B.S. Mohammed, and E.L., “Woen Modeling and optimizing the impact resistance of engineered cementitious composites with Multiwalled carbon nanotubes using response surface methodology,” *Scientific Reports* vol. 14, no. 1, article 24107, 2024. doi: 10.1038/s41598-024-75641-5.
- [2] Y. Ben-Zion, R. Einhorn Zarzecki, J. Glazer, and N.D. Finkelstein, “Leveraging AI for Rapid Generation of Physics Simulations in Education: Building

- Your Own Virtual Lab,” *The Physics Teacher*, vol. 63, no. 6, pp. 424-427, 2025. doi: 10.1119/5.0252343.
- [3] A. Slowik, and H. Kwasnicka, “Evolutionary algorithms and their applications to engineering problems,” *Neural Computing and Applications*, vol. 32, no. 16, pp. 12363-12379, 2020. doi: 10.1007/s00521-020-04832-8.
- [4] Z. Zhao, B. Liu, C. Zhang, and H. Liu, “An improved adaptive NSGA-II with multi-population algorithm,” *Applied Intelligence*, vol. 49, no. 2, pp. 569–580, 2018. doi: 10.1007/s10489-018-1263-6.
- [5] A. Pétrowski, and S. Ben-Hamida, “Multi-objective Optimization,” in *Evolutionary Algorithms*. Hoboken, USA, 2017, pp. 165-182. doi: 10.1002/9781119136378.ch5.
- [6] Y. Yuan, H. Xu, and B. Wang, “An improved NSGA-III procedure for evolutionary many-objective optimization,” in *Proc. of the 2014 Annual Conference on Genetic and Evolutionary Computation*, Vancouver, Canada, July 12-16, 2014, pp. 661-668. doi: 10.1145/2576768.2598342.
- [7] W. Whitney, S. Rana, J. Dzuber, and K.E. “Mathias, Evaluating evolutionary algorithms,” *Artificial Intelligence*, vol. 84, no. 1-2, p. 357-358, 1996. doi: 10.1016/0004-3702(96)81371-3.
- [8] L.A. Scardua, “Multiobjective Evolutionary Algorithm Based on Decomposition,” in *Applied Evolutionary Algorithms for Engineers Using Python*, 2021, pp. 149-158. doi: 10.1201/9780429298028-12.
- [9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002. doi: 10.1109/4235.996017.
- [10] K. Deb, and H. Jain, “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, 2014. doi: 10.1109/tevc.2013.2281535.
- [11] A.A. Keller, “Metaheuristic Optimization Algorithms,” in *Multi-Objective Optimization In Theory and Practice II: Metaheuristic Algorithms*, 2019, pp. 54-83. doi: 10.2174/9781681087054119010004.
- [12] J. Domínguez, O. Montiel, R. Sepúlveda, and N. Medina, “High Performance Architecture for NSGA-II,” in *Recent Advances on Hybrid Intelligent Systems*. Berlin, Heidelberg, 2013, pp. 451-461. doi: 10.1007/978-3-642-33021-6_35.
- [13] L.A. Scardua, “Non-Dominated Sorted Genetic Algorithm II,” in *Applied Evolutionary Algorithms for Engineers Using Python*, 2021, pp. 137-148. doi: 10.1201/9780429298028-11.

Стаття надійшла 30.10.2025

Стаття прийнята 09.12.2025

Стаття опублікована 29.12.2025

Цитуйте цю статтю як: Сергієнко А. В., Волобуєв Є. С. Дослідження використання генетичних алгоритмів для автоматизованого проектування ударостійких поверхонь. *Вісник Приазовського державного технічного університету. Серія: Технічні науки*. 2025. Вип. 52. С. 25-32. DOI: <https://doi.org/10.31498/2225-6733.52.2025.350946>.