

Застосування сучасних AI інструментів для реалізації алгоритмів МГУА в задачах моделювання фізико-механічних властивостей сплавів

Олексій Бихун
[TODO: Афіліація]
[TODO: Email]

23 листопада 2025 р.

Анотація

TODO: Написати анотацію (150-200 слів)

Ключові слова: МГУА, штучний інтелект, генерація коду, моделювання, металеві сплави, комбінаторний алгоритм, індуктивне моделювання

1 Вступ

TODO: Написати вступ, що описує:

- Актуальність задачі моделювання складних систем
- Проблему складності реалізації алгоритмів МГУА
- Поява нових AI інструментів для програмування
- Мета дослідження: використати AI для створення та верифікації реалізації МГУА

2 Методи індуктивного моделювання. Алгоритми МГУА

2.1 Основні принципи МГУА

TODO: Описати теоретичні основи МГУА:

- Принцип зовнішнього доповнення
- Селекція моделей
- Критерії незміщеності помилок та рішень

2.2 Комбінаторний алгоритм МГУА

Комбінаторний алгоритм МГУА перебирає всі можливі комбінації вхідних змінних для побудови моделей.

Існують два основні підходи:

Квадратичні поліноми для пар змінних Для кожної пари змінних (x_i, x_j) будується модель:

$$y = a_0 + a_1x_i + a_2x_j + a_3x_i^2 + a_4x_j^2 + a_5x_ix_j \quad (1)$$

Добре підходить для моделювання нелінійних залежностей.

Лінійна багатовимірна регресія Перебираються всі підмножини змінних з побудовою лінійної моделі:

$$y = a_0 + \sum_{k=1}^n a_k x_{i_k} \quad (2)$$

де $\{x_{i_1}, x_{i_2}, \dots, x_{i_n}\}$ – обрана підмножина змінних.

Цей підхід відповідає класичній методології МГУА з академічних робіт.

2.3 Багаторядний алгоритм МГУА

TODO: Описати багаторядний алгоритм:

- Побудова моделей на першому шарі
- Використання виходів моделей як нових ознак
- Критерій зупинки

3 Сучасні AI інструменти для програмування

3.1 Огляд AI інструментів

На сьогоднішній день існують потужні AI інструменти, здатні генерувати код:

- **Claude Code** (Anthropic) – спеціалізований CLI інструмент для роботи з кодом
- **GitHub Copilot** – інтеграція з IDE для автодоповнення коду
- **ChatGPT** (OpenAI) – універсальний асистент з можливістю генерації коду
- **Cursor** – AI-редактор коду

3.2 Можливості сучасних AI інструментів

TODO: Розширити опис можливостей:

- Генерація коду різними мовами програмування
- Пояснення існуючого коду

- Рефакторинг та оптимізація
- Написання тестів
- Виправлення помилок
- Робота з документацією

3.3 Обмеження AI інструментів

TODO: Описати обмеження:

- Можливі помилки в згенерованому коді
- Необхідність верифікації результатів
- Важливість чіткого формулювання завдання
- Ітеративний процес покращення

4 Застосування AI інструментів для реалізації алгоритмів МГУА

4.1 Процес розробки

Для реалізації алгоритмів МГУА було використано Claude Code з наступним підходом:

4.1.1 Крок 1: Формулювання завдання

Початкове завдання було сформульоване у вигляді:

“Реалізуй комбінаторний алгоритм МГУА на мові C з можливістю завантаження даних з CSV та виведенням найкращих моделей”

4.1.2 Крок 2: Початкова реалізація

AI згенерував початкову структуру проєкту:

- `gmdh.h` – заголовний файл з типами даних
- `data.c` – завантаження CSV даних
- `polynomial.c` – робота з поліномами
- `gmdh_combinatorial.c` – комбінаторний алгоритм (квадратичні пари)
- `gmdh_multirow.c` – багаторядний алгоритм

4.1.3 Крок 3: Виявлення проблеми

При тестуванні на академічному прикладі з статті виявилося, що початкова реалізація використовувала **квадратичні поліноми для пар змінних**, тоді як в статті очікувалася **лінійна багатовимірна регресія**.

Очікувані результати з статті:

$$S = 3: \quad y = 0.055 - 2.948x_3 - 6.980x_7 \quad (3)$$

$$S = 4: \quad y = 0.068 - 2.960x_3 + 6.982x_7 - 0.022x_8 \quad (4)$$

4.1.4 Крок 4: Коригування підходу

Було сформульовано нове завдання для AI:

“Потрібна реалізація лінійного багатовимірного МГУА, який перебирає всі підмножини ознак і будує лінійні моделі, а не квадратичні поліноми для пар”

AI створив новий модуль `gmdh_linear_combinatorial.c` з функцією:

```
1 linear_model_t* linear_combinatorial_gmdh(  
2     dataset_t *train,  
3     dataset_t *valid,  
4     int min_features,  //  
5     int max_features,  //  
6     int *n_models      //  
7 );
```

4.2 Архітектура реалізації

TODO: Додати діаграму архітектури або детальний опис модулів

Основні компоненти системи:

- Модуль завантаження даних з підтримкою CSV
- Модуль підгонки поліномів (метод найменших квадратів, виключення Гауса)
- Два варіанти комбінаторного МГУА (квадратичний та лінійний)
- Багаторядний алгоритм
- Обчислення метрик (RMSE, R^2)

4.3 Приклад використання

```
1 //  
2 dataset_t *ds = load_csv("data.csv", target_column);  
3
```

```

4 //
5 dataset_t *train, *valid;
6 split_dataset(ds, &train, &valid, 0.7);
7
8 //                                     (2-6
9                                     )
10 int n_models;
11 linear_model_t *models = linear_combinatorial_gmdh(
12     train, valid, 2, 6, &n_models
13 );
14 //
15 for (int i = 0; i < 10 && i < n_models; i++) {
16     print_linear_model(&models[i], train->feature_names);
17 }

```

5 Тестування на академічному прикладі

5.1 Опис тестової задачі

Для верифікації реалізації було використано тестову вибірку з академічної статті [?].

Характеристики вибірки:

- 14 зразків (рядки 1-10, 27-30 з оригінальної таблиці)
- 8 вхідних змінних: $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$
- 1 вихідна змінна: y
- 10% випадкового шуму в даних

5.2 Результати тестування

Дані було розділено на навчальну (70%, 9 зразків) та валідаційну (30%, 5 зразків) вибірки.

5.2.1 Найкращі моделі

Топ-3 моделі за критерієм RMSE на валідаційній вибірці:

№	Модель	RMSE	R^2
1	$y = -0.033 - 2.986x_3 - 0.079x_5 + 6.997x_7 - 0.103x_8$	0.3323	0.9999
2	$y = -0.205 - 2.952x_3 - 0.067x_4 - 0.088x_5 + 7.045x_7 - 0.085x_8$	0.4036	0.9998
3	$y = -0.073 + 0.018x_1 - 2.975x_3 - 0.062x_4 + 6.966x_7 - 0.124x_8$	0.4543	0.9998

Табл. 1: Найкращі моделі, отримані алгоритмом

5.2.2 Порівняння з очікуваними результатами

Очікувані результати з таблиці 2.2 академічної статті:

S	Модель
3	$y = 0.055 - 2.948x_3 - 6.980x_7$
4	$y = 0.068 - 2.960x_3 + 6.982x_7 - 0.022x_8$
5	$y = 0.05 - 2.95x_3 - 0.032x_5 + 6.987x_7 - 0.022x_8$

Табл. 2: Очікувані моделі з академічної статті

5.2.3 Аналіз відповідності

Збіг ознак:

- Найкраща модель використовує ознаки x_3, x_5, x_7, x_8 – **точний збіг** з очікуваною моделлю S=5
- Модель №10 використовує лише x_3, x_7, x_8 – відповідає структурі S=4

Порівняння коефіцієнтів (найкраща модель vs S=5):

- x_3 : -2.986 vs -2.95 (різниця 1.2%)
- x_5 : -0.079 vs -0.032 (різниця 147%)
- x_7 : $+6.997$ vs $+6.987$ (різниця 0.1%)
- x_8 : -0.103 vs -0.022 (різниця 368%)

Причини відмінностей:

- Використано лише 14 з 30 зразків
- Відсутні дані рядків 11-26 впливає на підгонку моделі
- Інший розподіл на навчальну/валідаційну вибірки
- Випадковий шум в даних

Висновок: Алгоритм **коректно ідентифікував** ключові ознаки (x_3, x_5, x_7, x_8) , що підтверджує правильність реалізації.

6 Застосування до реальної задачі моделювання властивостей сплавів

6.1 Постановка задачі

TODO: Описати задачу моделювання фізико-механічних властивостей металевих сплавів:

- Які властивості моделюються

- Які вхідні параметри (склад сплаву, температура, тощо)
- Джерело даних
- Розмір вибірки

6.2 Підготовка даних

TODO: Описати:

- Структуру даних
- Попередню обробку
- Нормалізацію
- Поділ на навчальну/тестову вибірки

6.3 Результати моделювання

TODO: Додати результати:

- Таблиця найкращих моделей
- Графіки залежності точності від складності моделі
- Графіки передбачених vs фактичних значень
- Порівняння з результатами попередньої роботи [посилання]

6.4 Порівняння з попередніми дослідженнями

TODO: Порівняти з результатами попередньої статті про сплави:

- Чи збігаються ключові ознаки?
- Чи подібні коефіцієнти моделей?
- Чи покращилася точність?

6.5 Інтерпретація результатів

TODO: Фізичне пояснення отриманих моделей:

- Які ознаки виявилися найважливішими?
- Чи відповідає це фізичним очікуванням?
- Практична цінність отриманих моделей

7 Обговорення

7.1 Переваги використання AI інструментів

Використання AI інструментів для реалізації алгоритмів МГУА показало наступні переваги:

- **Швидкість розробки:** початкова реалізація була створена за лічені години замість днів
- **Якість коду:** AI генерує структурований код з належними перевітками
- **Документація:** автоматичне додавання коментарів
- **Гнучкість:** легко модифікувати та додавати нові функції
- **Навчання:** процес роботи з AI допомагає краще зрозуміти алгоритм

7.2 Виклики та обмеження

TODO: Розширити опис викликів:

- Необхідність точного формулювання завдань
- Важливість верифікації згенерованого коду
- Потреба в розумінні предметної області
- Ітеративний процес покращення

7.3 Рекомендації щодо використання AI

TODO: Додати практичні рекомендації:

- Як формулювати завдання для AI
- Як перевіряти згенерований код
- Коли довіряти AI, а коли ні
- Як організувати ітеративний процес розробки

8 Висновки

TODO: Написати висновки, що підсумовують:

1. Успішно застосовано сучасні AI інструменти (Claude Code) для реалізації комбінованих алгоритмів МГУА на мові C
2. Реалізовано два варіанти алгоритму:
 - Квадратичні поліноми для пар змінних
 - Лінійна багатовимірна регресія (класичний підхід)

3. Верифіковано коректність реалізації на тестовій задачі з академічної літератури:
 - Алгоритм коректно ідентифікував ключові ознаки (x_3, x_5, x_7, x_8)
 - Коефіцієнти моделей близькі до очікуваних
4. Застосовано розроблений інструмент до реальної задачі моделювання властивостей металевих сплавів
5. Результати узгоджуються з попередніми дослідженнями [TODO: додати конкрету]
6. Продемонстровано ефективність використання AI для наукового програмування

TODO: Додати перспективи подальших досліджень

Подяки

TODO: Додати подяки (якщо потрібно)

Література

- [1] **TODO:** Додати посилання на оригінальні роботи Івахненка про МГУА
- [2] **TODO:** Додати посилання на статтю з тестовою задачею (example_test_sample.docx)
- [3] **TODO:** Додати посилання на попередню статтю про моделювання властивостей сплавів
- [4] **TODO:** Додати посилання на Claude Code / Anthropic
- [5] **TODO:** Додати інші релевантні посилання про AI coding tools