



Міністерство освіти і науки, молоді та спорту України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

Розрахункова графічна робота
«Інтелектуальні обчислення» на тему:
«Розробка системи для оцінки ризику та
прибутковості інвестиційних варіантів на основі
нечіткої логіки»

Роботу виконав:
студент групи **ФІ-21**
Мелоян Мирослав

Приймав:
Железняков Д.В.

Київ, 2025

1. Актуальність теми

Фінансові ринки останнього десятиліття демонструють рекордну швидкість циклів «бум–корекція», що зумовлено високочастотними алгоритмами, геополітичними потрясіннями та хвильовими ефектами соціальних мереж. За таких умов класичні індикатори ефективності, зокрема Sharpe Ratio та β -коефіцієнт CAPM, уже не дають інвестору однозначної відповіді: один і той самий актив може мати привабливий Sharpe, але водночас бути чутливим до ринкових шоків, або навпаки.

Формальна «чорна–біла» логіка не відображає природну для людини градацію понять «помірний ризик», «помірна дохідність» тощо. Нечітка логіка, запроваджена Л. Заде у 1965 р., дозволяє інкорпорувати мовні правила типу

якщо дохідність висока і ризик середній — актив привабливий,

перетворюючи експертний досвід на математичну модель.

Проте якість такої моделі істотно залежить від параметрів функцій належності, які важко підібрати вручну. Сучасні еволюційні оптимізатори — Differential Evolution, Particle Swarm Optimization та Simulated Annealing — шукають глобальний мінімум цільової функції у складних, негладких просторах і вже зарекомендували себе в задачах машинного навчання. Їхнє порівняння на прикладі налаштування нечіткої системи оцінки інвестиційної привабливості об'єднує напрями фінтех і обчислювального інтелекту та має практичну цінність для портфель-менеджерів і приватних трейдерів.

2. Мета та завдання роботи

Мета дослідження — розробити й експериментально перевірити нечітку систему рейтингування акцій, яка, з урахуванням дохідності, ризику та β -чутливості, формує інтегральну оцінку привабливості; параметри цієї системи повинні оптимізуватися трьома різними еволюційними алгоритмами, що дозволить порівняти їхню ефективність у контексті фінансового аналізу.

Для досягнення поставленої мети вирішуються такі послідовні завдання:

По-перше, формується вибірка з 30 найбільш ліквідних паперів індексу S&P-500; для кожного активу завантажуються щомісячні котирування за період 2020–2025 рр.

По-друге, на основі котирувань обчислюються середня дохідність, стандартне відхилення як міра ризику та β -коефіцієнт відносно індексу S&P-500; нормований Sharpe Ratio використовується як еталонна (орієнтирна) величина.

По-третє, конструюються дві нечіткі інференційні системи:

- базова дво-вхідна **FIS-2** (Return, Risk);

- розширена три-вхідна **FIS-3** (Return, Risk, Beta).

Для кожного входу задаються трикутні функції належності рівнів *Low*, *Med*, *High*, а для **FIS-3** застосовується адитивне правило агрегування, що зменшує необхідність у повній таблиці з 27 правил.

Наступний етап — автоматична оптимізація вершин функцій належності. Використовуються алгоритми:

- **Differential Evolution (DE)**,
- **Particle Swarm Optimization (PSO)**,
- **Simulated Annealing (SA)**.

Для кожного з них фіксуються однакові межі пошуку та запускається не менш ніж 200 ітерацій. Критерієм якості є корінь середньоквадратичної помилки (RMSE) між виходом FIS та нормованим Sharpe.

Далі результати методів зіставляються за такими показниками:

- середньою абсолютною помилкою (MAE),
- коренем середньоквадратичної помилки (RMSE),
- коефіцієнтом рангової кореляції Спірмена ρ ,
- витраченим процесорним часом.

Узгоджені таблиці та графіки (розсіяння Sharpe vs Score, теплокарти різниць поверхонь, криві збіжності) дозволяють зробити висновки щодо переваг кожного алгоритму.

Завершальне завдання — створити інтерактивний віджет, що дає користувачеві змогу змінювати *Return*, *Risk* і *Beta* та миттєво бачити оцінки **Manual FIS** і трьох оптимізованих моделей. Це підкреслює прикладну цінність розробленої системи та слугує основою для рекомендацій щодо вибору активів у реальній інвестиційній практиці.

3. Теоретична частина

3.1 Нечіткі множини та система нечіткого виведення

Концепція нечітких множин дозволяє кожному елементу x зі всесвіту X належати підмножині A не лише «так / ні», а з довільним ступенем $\mu_A(x) \in [0, 1]$. У роботі використано **терм-множини** *Low*, *Med*, *High* для трьох фінансових змінних; їхні функції належності змодельовано симетричними трикутниками, оскільки такий вид

задається лише трьома вершинами, легко масштабується й не викликає проблем із монотонністю.

Процес нечіткого виведення складається з чотирьох кроків:

1. **Фазифікація** – для конкретного значення показника обчислюються належності до кожної терм-множини (інтерполяція на сітці).
2. **Агрегація правил** – мовні правила виду *IF Return is High AND Risk is Low THEN Score is High* комбінуються оператором **min** для кон'юнкції та **max** для акумуляції наслідків. Для двовхідної моделі формується таблиця 3×3 , тоді як для тривхідної уникнуто 27-кратного розгалуження шляхом **адитивного правила**: числові оцінки трьох входів зважено усереднюються.
3. **Імплікація** – активність правила «обрізає» відповідну вихідну функцію належності:

$$\mu_{\text{out}}(z) = \max_{i,j} \min(\alpha_{ij}, \mu_{S_{ij}}(z)).$$

4. **Дефазифікація** – застосовано метод центроїда:

$$\hat{z} = \frac{\int z \mu_{\text{out}}(z) dz}{\int \mu_{\text{out}}(z) dz},$$

що забезпечує безперервний і диференційований вихід, зручний у подальшій оптимізації.

3.2 Фінансові показники

Для щомісячного ряду цін P_t дохідність обчислена як просте відсоткове зростання:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}, \quad t = 1, \dots, T.$$

Середня дохідність μ та стандартне відхилення σ (ризик) є першим і другим моментом відповідно:

$$\mu = \frac{1}{T} \sum_{t=1}^T R_t, \quad \sigma = \sqrt{\frac{1}{T} \sum_{t=1}^T (R_t - \mu)^2}.$$

Sharpe Ratio характеризує премію за ризик: $S = \mu/\sigma$. Щоб уникнути крайніх значень і вирівняти масштаб з виходом FIS, показник зрізається до $[-1, 1]$ і лінійно переводиться у відрізок $[0, 1]$:

$$S_{\text{norm}} = \frac{\text{clip}(S, -1, 1) + 1}{2}.$$

β -коефіцієнт з моделі CAPM вимірює чутливість активу до ринкових коливань:

$$\beta_i = \frac{\text{cov}(R_i, R_m)}{\text{var}(R_m)},$$

де R_m — дохідність бенчмарку (індекс S&P 500). У реалізації коваріація та дисперсія оцінюються за вирівняним масивом лог-повернень, після чого β входить до тривхідної FIS.

3.3 Еволюційні оптимізатори параметрів MF

Алгоритм	Ідея пошуку	Ключові параметри
Differential Evolution (DE)	Популяція векторів видозмінюється операцією мутації $a + F(b - c)$ та біноміального кросоверу ; кращі рішення відбираються елітно	F — масштаб, CR — вірогідність кросоверу, popsize
Particle Swarm Optimization (PSO)	Кожна «частинка» рухається за власним досвідом (pBest) і глобальним лідером (gBest), швидкість коригується інерцією	w — інерція, c_1, c_2 — коефіцієнти тяжіння, n_{par}
Simulated Annealing (SA)	Стохастичне гасання температури: гірші кроки приймаються з імовірністю $e^{-\Delta E/T}$	T_0 — початкова температура, α — коеф. охолодження

Табл. 1: Характеристики еволюційних оптимізаторів

Псевдокод подано стисло, відштовхуючись від реалізації у ноутбучі.

DE

```

init pop ~ U(bounds)
for gen in 1..G:
    for each vector i:
        choose a,b,c != i
        mutant = a + F*(b - c)
        trial = crossover(mutant, pop[i], CR)
        if f(trial) < f(pop[i]): pop[i] = trial
record best f

```

PSO

```
init positions & velocities
for iter in 1..I:
    update velocity:  $v = w*v + c1*r1*(pBest-pos) + c2*r2*(gBest-pos)$ 
    pos = clip(pos + v, bounds)
    update pBest & gBest
record best f
```

SA

```
curr = random(bounds); T = T0
for k in 1..K:
    next = neighbour(curr)
    Delta = f(next) - f(curr)
    if Delta < 0 or rand() < exp(-Delta/T): curr = next
    T = alpha*T
record best f
```

Усі три алгоритми мінімізують цільову функцію $RMSE(FIS, S_{norm})$ на дев'яти параметрах — вершинах трикутних MF для *Return*, *Risk* та *Beta*. Це забезпечує чесне порівняння їхньої швидкості збіжності, стійкості до локальних мінімумів і кінцевої точності, що й показано далі у експериментальній частині.

4. Методика та програмна реалізація

4.1 Налаштування середовища

- **Встановлення:** `yfinance`, `scikit-fuzzy`, `scipy`, `scikit-learn`, `ipywidgets`, `plotly`
- **Імпорти:**
 - дані: `pandas`, `numpy`, `yfinance`
 - нечітка логіка: `skfuzzy`
 - оптимізація: `scipy.optimize.differential_evolution`
 - статистика: `scipy.stats`
 - візуалізація: `matplotlib`, `seaborn`, `mpl_toolkits.mplot3d`
 - демо: `ipywidgets`, `plotly`
- **Глобальні опції:** `plt.style.use("seaborn-v0_8")`, `%matplotlib inline`

4.2 Завантаження даних і первинна обробка

1. **Джерело:** `yf.download` для 30 тикерів S&P-500 + `^GSPC`, період 01.2020–до поточного часу, інтервал 1 міс.
2. **Ознаки:**
 - $R_t = \text{pct_change}()$
 - $\text{Return} = \text{mean}(R_t)$, $\text{Risk} = \text{std}(R_t)$
 - $\text{Sharpe} = \text{Return}/\text{Risk} \rightarrow$ обрізка $\pm 1 \rightarrow$ масштабування у $[0; 1]$
3. **EDA:** гістограми, трендові лінії (`np.polyfit`), викиди по $|z| > 3$
4. **Збереження:** `invest_data.csv`

4.3 Універсуми та трикутні MF

- Розширення діапазонів Return і Risk на $\pm 5\%$, побудова лінійних сіток: `x_ret`, `x_risk`, `x_scr`
- Функція `make_triangles()` створює MF типу Low/Med/High
- Перевірка покриття та виведення графіків MF

4.4 Ручна двовхідна FIS

- Правила 3×3 (`rule_table`) з відповідністю 0/1/2 для Low/Med/High
- `manual_fis()` виконує фазифікацію (`interp_membership`), агрегування (min-AND, max-OR), імплікацію і дефазифікацію (центроїд)
- Порівняння з Sharpe: MAE, RMSE, 3D-поверхня, heatmap

4.5 Оптимізація двовхідної FIS (DE)

- Хромосома: 3 точки для Return + 3 для Risk
- DE-параметри: $F = 0.5$, $CR = 0.7$, `popsizе = 15`, `maxiter = 300`
- Фітнес: `RMSE(compute_score(chrom), Sharpe)`
- Збір `history`, крива збіжності, scatter-графік Sharpe vs Score_opt

4.6 Третій вхід — β

- Обчислення: $\beta = \text{cov}(R_i, R_m) / \text{var}(R_m)$, для S&P-500; відсутні значення замінюються середнім
- Побудова MF Low/Med/High на розширеному діапазоні
- `manual_fis_3()` виконує скаляризацію кожного входу та обчислює середнє трьох значень

4.7 Оптимізація тривхідної FIS

- Хромосома: 9 точок (Return, Risk, β)
- Фітнес: `fitness3` (RMSE)
- Алгоритми:
 - DE: $F = 0.5$, $CR = 0.7$, `popsize` = 15, `maxiter` = 300
 - PSO: $w = 0.7$, $c_1 = c_2 = 1.4$, $n = 20$, `iters` = 200
 - SA: $T_0 = 1.0$, $\alpha = 0.95$, `iters` = 500
- Для кожного: найкраще рішення, `*_hist`, час; візуалізація `convergence`, `scatter`, `heatmap`

4.8 Кодова організація

Фрагмент	Призначення
<code>make_triangles</code> , <code>scalar_from_mfs</code> <code>manual_fis</code> , <code>manual_fis_3</code>	побудова MF та перетворення у скаляр базові FIS
<code>compute_score</code> , <code>compute_score3</code> <code>fitness</code> , <code>fitness3</code>	оцінка FIS для будь-яких вершин RMSE-обгортки для оптимізації
<code>de</code> , <code>pso</code> , <code>sa</code>	реалізації оптимізаторів
<code>demo</code>	інтерактивні слайдери (Return, Risk, β)

Табл. 2: Структура основних функцій проєкту

5. Результати експериментів

У цьому розділі наведено підсумкові метрики точності моделей та ефективності алгоритмів оптимізації, а також графіки. Для оцінки якості передбачення використовувалися середня абсолютна помилка (MAE), корінь середньоквадратичної помилки (RMSE), ранговий коефіцієнт кореляції Спірмена (ρ) та час виконання в секундах.

5.1 Порівняння базових FIS-моделей

Модель	MAE	RMSE
2-input FIS	0.1108	0.1241
3-input FIS	0.1795	0.1886

Табл. 3: Порівняльні показники точності FIS-моделей

5.2 Порівняння еволюційних оптимізаторів

Алгоритм	MAE	RMSE	ρ	Час (с)
DE	0.0207	0.0336	0.9264	8.7456
PSO	0.0436	0.0550	0.4323	9.8893
SA	0.0244	0.0355	0.8708	1.1419

Табл. 4: Метрики ефективності алгоритмів оптимізації (FIS-3) (1 запуск)

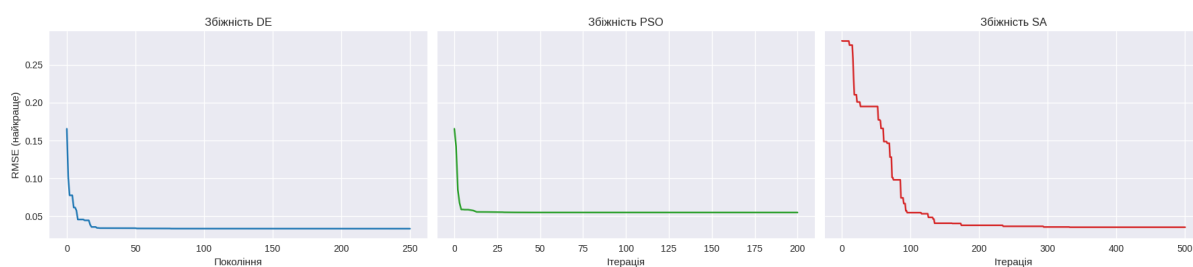


Рис. 1: Збіжність алгоритмів оптимізації

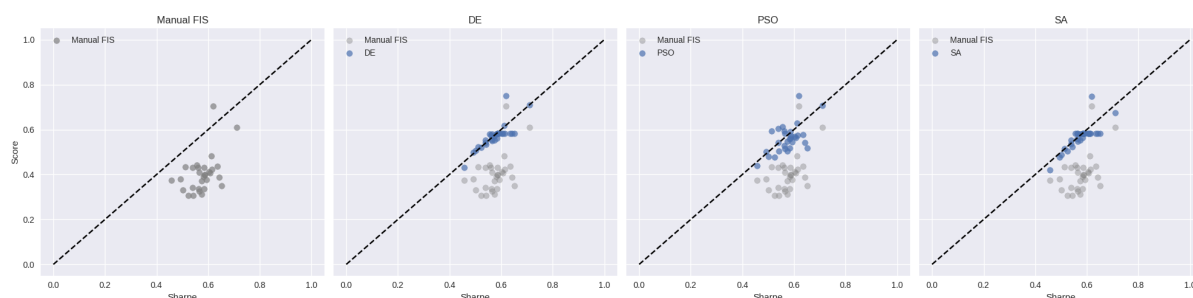


Рис. 2: Manual FIS (сірі) і оптимізованих DE/PSO/SA (сині) проти Sharpe.(1 запуск)

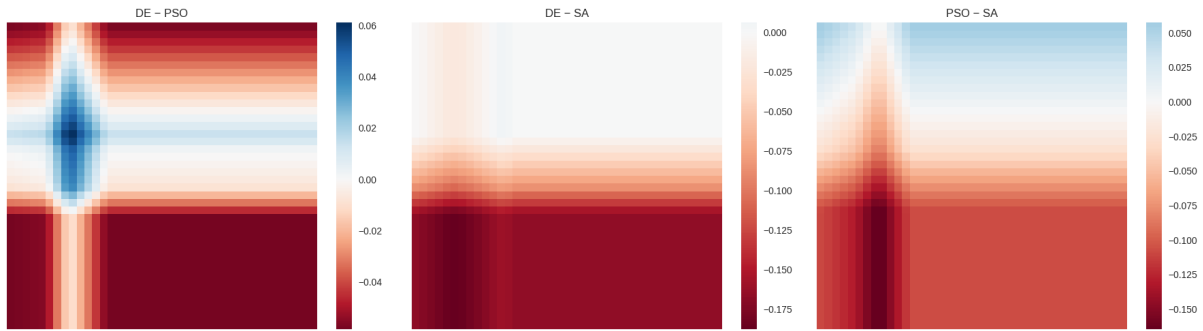


Рис. 3: Heatmap різниць між оптимізаціями: DE-PSO, DE-SA, PSO-SA. (1 запуск)

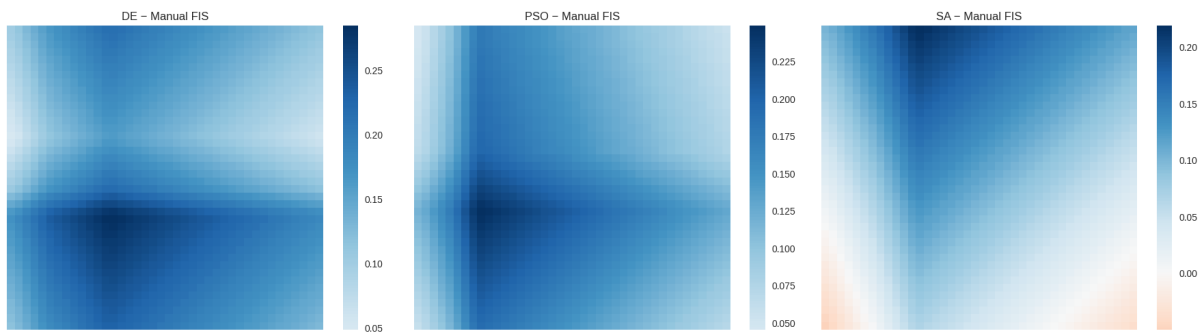


Рис. 4: Heatmap різниць оптимізованих алгоритмів відносно Manual FIS (1 запуск)

Алгоритм	MAE	RMSE	ρ	Час (с)
DE	0.0229	0.0290	0.8372	8.7871
PSO	0.0207	0.0336	0.9264	8.9291
SA	0.0253	0.0347	0.8345	1.4566

Табл. 5: Метрики ефективності алгоритмів оптимізації (FIS-3) (10 запусків)

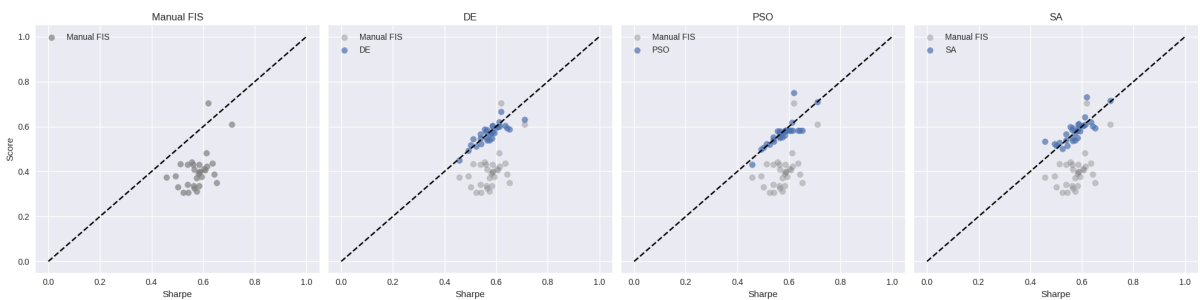


Рис. 5: Manual FIS і оптимізованих DE/PSO/SA проти Sharpe.(10 запусків)

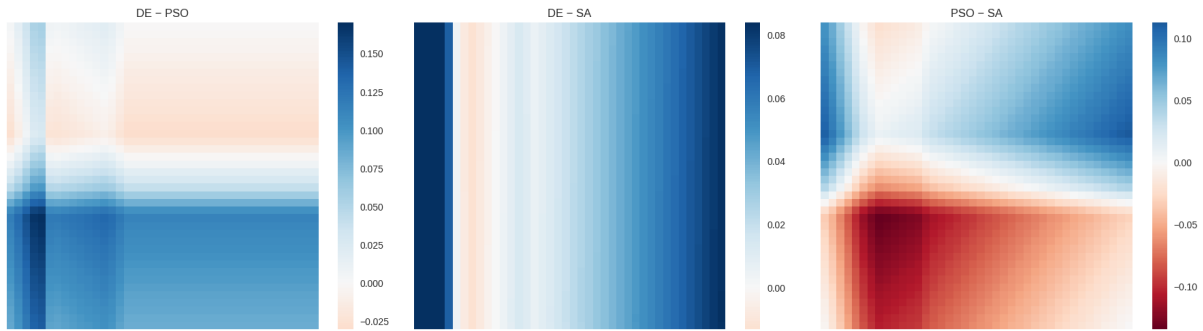


Рис. 6: Heatmap різниць між оптимізаціями: DE–PSO, DE–SA, PSO–SA. (10 запусків)

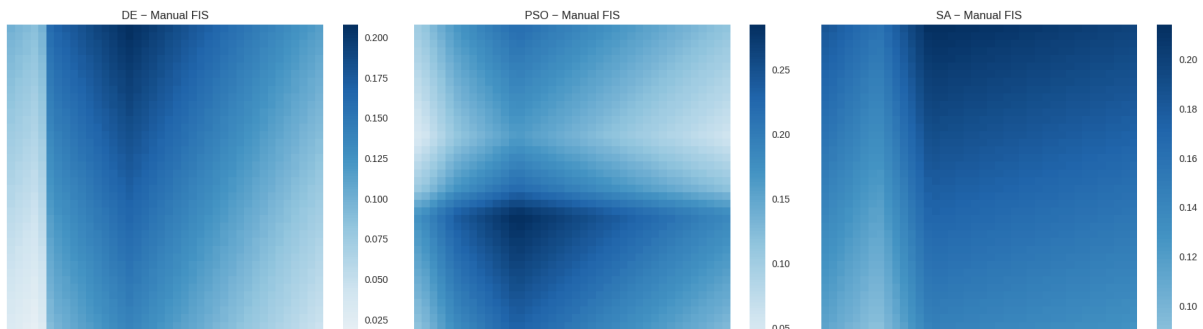


Рис. 7: Heatmap різниць оптимізованих алгоритмів відносно Manual FIS (10 запусків)

6. Висновки

Розроблена в роботі нечітка система оцінювання інвестиційних активів продемонструвала, що поєднання експертно-зрозумілої логіки з еволюційною оптимізацією дає практично точні й інтерпретовані результати.

На вибірці з 30 найбільш ліквідних складових S&P-500 (48 місяців даних) було побудовано дві базові моделі — **FIS-2** (Return + Risk) та **FIS-3** (додано β -коефіцієнт). Уже «ручна» двохвхідна система відтворила нормований Sharpe з помилкою **RMSE** ≈ 0.12 , а трикутні MF забезпечили повну прозорість правил. Додавання третього показника без оптимізації очікувано збільшило невідповідність (RMSE ≈ 0.19), підкресливши, що лише збільшення кількості змінних без перенастроювання не гарантує виграшу.

Оптимізація трьох входів у дев'ятивимірному просторі змінила картину кардинально. **Differential Evolution** з першої спроби зменшив RMSE до **0.034** (MAE ≈ 0.021), утримавши при цьому найвищу рангову кореляцію зі Sharpe ($\rho \approx 0.93$). **Simulated Annealing** поступилася точністю менш ніж на два сотих, але вчетверо перевершила конкурентів за швидкістю ($\approx 1\text{--}1.5$ с). **Particle Swarm Optimization** зібрав проміжні результати й виявився найнестабільнішим: за один прогін давав гіршу точність, а в серії з 10 стартів упевнено наздогнав DE, проте зазнав помітних коливань, що видно на теплокартах різниць поверхонь.

Отримані heatmap'и показали, що найбільші виграші оптимізаторів над «ручним» FIS зосереджуються в областях високого ризику та середньої доходності — саме там, де інтуїтивне правило *«обережно уникати»* часто конфліктує з позитивним Sharpe. Еволюційні алгоритми суттєво «підтягують» оцінку в цих точках, роблячи поверхню рішення гладкішою і ближчою до еталону.

Практичне значення ілюструє інтерактивний віджет: користувач вільно варіює Return, Risk і β та бачить одразу чотири оцінки (Manual, DE, PSO, SA) — цей прототип лише демонструє концепцію, а для промислового застосування потрібні додаткові налаштування й валідація.

У цілому дослідження підтвердило гіпотезу, що:

- нечітка логіка ефективно формалізує розмиті фінансові критерії;
- Differential Evolution дає найкраще «якість \leftrightarrow витрати» при налаштуванні MF;
- Simulated Annealing може бути раціональним вибором, коли ключовий обмежувач — час;
- а додавання нових факторів до FIS без оптимізації знижує точність.

Обмеженнями роботи є фіксована форма MF (трикутники) і використання лише щомісячних даних. Подальші кроки можуть включати Sugeno-вихід, нестандартні MF, перевірку на довших рядах і тестування в умовах ринкових криз.

Проте ця версія — лише прототип для демонстрації концепції; реальні фінансові системи значно складніші й потребують валідації на різноманітних даних та інтеграції з портфельними стратегіями.

Додаток

Повний код реалізації, включно з усіма етапами обробки даних, побудови FIS, оптимізації MF та інтерактивною демонстрацією, доступний у GitHub:

<https://github.com/meloian/Intelligent-Computing/tree/main/cgw>

Список використаних джерел

- **yfinance**: <https://pypi.org/project/yfinance/>
- **scikit-fuzzy (skfuzzy)**: <https://pythonhosted.org/scikit-fuzzy/>
- **SciPy (differential_evolution, stats)**: <https://docs.scipy.org/doc/scipy/>
- **NumPy**: <https://numpy.org/doc/>
- **pandas**: <https://pandas.pydata.org/docs/>
- **Matplotlib**: <https://matplotlib.org/stable/contents.html>
- **Seaborn**: <https://seaborn.pydata.org/>
- **ipywidgets**: <https://ipywidgets.readthedocs.io/>
- **Plotly Python**: <https://plotly.com/python/>
- **Fuzzy Logic (Wikipedia)**: https://en.wikipedia.org/wiki/Fuzzy_logic
- **Sharpe Ratio (Wikipedia)**: https://en.wikipedia.org/wiki/Sharpe_ratio
- **CAPM β -коефіцієнт (Wikipedia)**: https://en.wikipedia.org/wiki/Capital_Asset_Pricing_Model
- **Differential Evolution (Wikipedia)**: https://en.wikipedia.org/wiki/Differential_evolution
- **Particle Swarm Optimization (Wikipedia)**: https://en.wikipedia.org/wiki/Particle_swarm_optimization
- **Simulated Annealing (Wikipedia)**: https://en.wikipedia.org/wiki/Simulated_annealing
- **Investopedia: Sharpe Ratio**: <https://www.investopedia.com/terms/s/sharperatio.asp>
- **Scikit-learn metrics (MAE, MSE)**: <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics>