

**AGH**

**AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE**

## Ćw.3: Metody skalaryzacji.

Optymalizacja Wielokryterialna

Kacper Błachowiak  
Jakub Ciura

Automatyka i Robotyka  
II stopień 2025/2026

Kraków, 17 listopad 2025 r.

# 1 Wstęp

Wielokryterialne problemy optymalizacyjne stanowią istotną klasę zagadnień, w których dąży się do jednoczesnej poprawy kilku, często sprzecznych ze sobą, funkcji celu. W takim przypadku nie istnieje zazwyczaj pojedyncze rozwiązanie absolutnie najlepsze — zamiast tego dąży się do wyznaczenia zbioru rozwiązań Pareto-optymalnych, reprezentujących kompromisy pomiędzy poszczególnymi kryteriami. Aby umożliwić ich wyznaczenie, stosuje się różne metody skalarizacji, które przekształcają problem wielokryterialny w sekwencję problemów jednokryterialnych.

Celem niniejszego ćwiczenia było praktyczne zbadanie i porównanie trzech klas metod skalarizacji:

1. S1 — metoda sumy ważonej,
2. S2 — metoda ograniczeń typu  $\varepsilon$  (epsilon-constraints),
3. S3 — metoda minimalizacji odległości od punktu odniesienia.

W eksperymencie wykorzystano dwa przykładowe problemy: dwukryterialny ( $F : R^2 \rightarrow R^2$ ) oraz trzykryterialny ( $G : R^3 \rightarrow R^3$ ). Dla obu funkcji przeprowadzono losowe próbkowanie przestrzeni decyzyjnej, estymując punkty idealne i nadir, a następnie zastosowano każdą z trzech metod skalarizacji w celu wygenerowania aproksymacji frontu Pareto. Otrzymane wyniki zwizualizowano na wykresach oraz zapisano do plików wynikowych.

Celem sprawozdania jest przedstawienie przebiegu eksperymentów, interpretacja rezultatów oraz porównanie efektywności zastosowanych metod skalarizacji w kontekście jakości uzyskanej aproksymacji zbioru Pareto.

# 2 Opis implementacji

W ramach ćwiczenia przygotowano kompletny skrypt w języku Python realizujący generowanie aproksymacji frontów Pareto dla dwóch problemów wielokryterialnych. Kod implementuje trzy metody skalarizacji (S1, S2 oraz S3), a następnie wizualizuje oraz zapisuje otrzymane wyniki. Poniżej przedstawiono szczegółowy opis najważniejszych komponentów programu.

## 2.1 Funkcje pomocnicze

W kodzie zdefiniowano funkcje odpowiedzialne za wyznaczanie zbioru punktów niezdominowanych:

- `is_dominated(v, V)` — sprawdza, czy wektor  $v$  jest zdominowany przez którykolwiek punkt ze zbioru  $V$ .
- `nondominated_set(V)` — zwraca zbiór punktów Pareto-niezdominowanych spośród punktów przekazanych w macierzy  $V$ .

Funkcje te umożliwiają późniejszą ekstrakcję przybliżonego zbioru Pareto z punktów uzyskanych metodami skalarizacji.

## 2.2 Definicje funkcji celu

W pliku zawarto dwie funkcje wielokryterialne wykorzystywane do testów:

- `F_obj(x)` — funkcja dwukryterialna  $F : R^2 \rightarrow R^2$ ,
- `G_obj(x)` — funkcja trzykryterialna  $G : R^3 \rightarrow R^3$ .

Każda z nich zwraca wektor wartości funkcji celu. Zastosowano kombinacje wyrażeń kwadratowych, wielomianowych oraz składników nieliniowych (sinus), aby zachować niemonotoniczność i uzyskać nieliniowe fronty Pareto.

## 2.3 Próbkowanie przestrzeni decyzyjnej

Do estymacji punktu idealnego i nadir wykorzystano funkcję:

- `sample_objectives(obj_fun, bounds, n_samples)`.

Losowo generuje ona próbki z przestrzeni decyzyjnej na podstawie przedziałów ograniczeń (każda zmienna z przedziału  $[-1, 1]$ ). Dla każdej próbki obliczana jest wartość funkcji celu. Pozwala to na:

- ocenę globalnego rozkładu wartości funkcji,
- wygenerowanie punktów odniesienia (ideal, nadir),
- wizualizację całej chmury próbkowanych punktów w przestrzeni celów.

## 2.4 Metody skalaryzacji

Zaimplementowano trzy klasyczne metody przekształcania problemu wielokryterialnego do jednokryterialnego.

### 2.4.1 Metoda S1 — suma ważona

Zrealizowana w funkcji:

- `scalarize_S1_weighted_sum(obj_fun, bounds, lambdas)`.

Metoda ta rozwiązuje problem:

$$\min_x \sum_i \lambda_i f_i(x),$$

gdzie  $\lambda_i$  to wybrane wagi spełniające  $\sum \lambda_i = 1$ . W praktyce zastosowano regularną siatkę wag, co pozwala uzyskać równomierne próbkowanie frontu Pareto w przypadku problemów wypukłych.

#### 2.4.2 Metoda S2 — ograniczenia typu $\varepsilon$

Zrealizowana w funkcji:

- `scalarize_S2_epsilon_constraints(obj_fun, bounds, main_index, epsilons).`

Podejście to minimalizuje wybraną funkcję celu, a pozostałe ogranicza wartościami  $\varepsilon$ :

$$\min_x f_k(x) \quad \text{przy ograniczeniach} \quad f_i(x) \leq \varepsilon_i.$$

Metoda działa również dla trzech kryteriów, gdzie wartości  $\varepsilon$  są dobierane na siatce dwuwymiarowej.

#### 2.4.3 Metoda S3 — minimalizacja odległości od punktu idealnego

Zrealizowana w funkcji:

- `scalarize_S3_distance(obj_fun, bounds, ref_point, p).`

Rozwiązuje problem:

$$\min_x \|F(x) - z^*\|_p,$$

gdzie  $z^*$  jest oszacowanym punktem idealnym, a norma  $p$  może być równa 2 lub  $\infty$ .

### 2.5 Procedury eksperymentalne

Przeprowadzono dwa osobne eksperymenty:

- `run_bi_objective_experiment()` dla funkcji dwukryterialnej,
- `run_tri_objective_experiment()` dla funkcji trzykryterialnej.

Każda procedura:

1. generuje próbki losowe oraz wyznacza punkt idealny i nadir,
2. stosuje wszystkie trzy metody skalaryzacji,
3. gromadzi uzyskane punkty w przestrzeni celów,
4. identyfikuje zbiór Pareto-niezdominowany,
5. tworzy wizualizację wyników,
6. zapisuje dane w formacie `.csv` oraz wykresy w formacie `.png`.

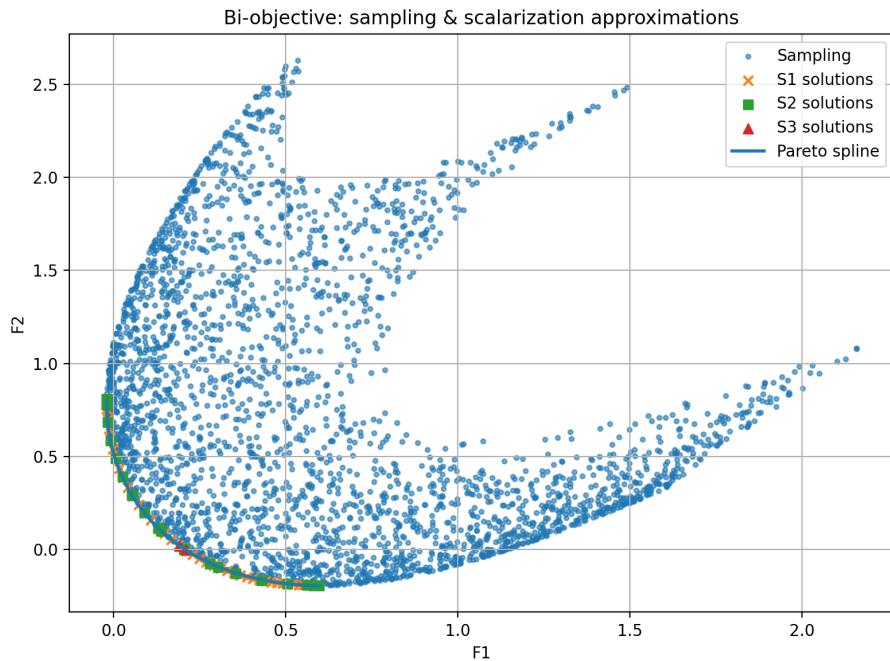
Główna funkcja programu:

- `main()` — uruchamia oba eksperymenty i zapisuje wszystkie wyniki do katalogu `output`.

### 3 Wyniki eksperymentów

W ramach przeprowadzonych eksperymentów otrzymano aproksymacje frontów Pareto zarówno dla problemu dwukryterialnego, jak i trzykryterialnego. Poniżej przedstawiono analizę uzyskanych wyników wraz z odniesieniem do dołączonych wykresów.

#### 3.1 Problem dwukryterialny



Na rysunku przedstawiającym wyniki dla funkcji  $F : R^2 \rightarrow R^2$  widoczna jest chmura punktów pochodzących z losowego próbkowania przestrzeni decyzyjnej. Punkty te pozwalają oszacować topografię przestrzeni celów, w tym przybliżone położenie punktów idealnego oraz nadir.

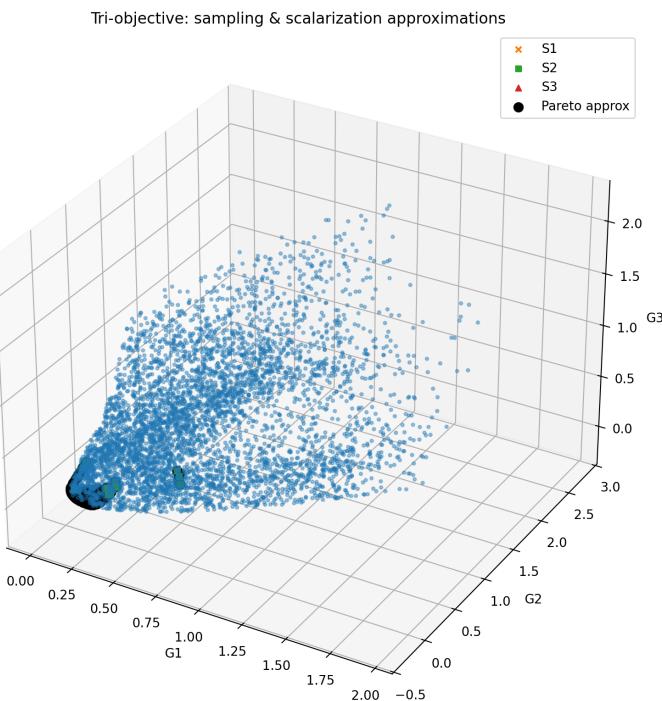
Zastosowanie metod skalaryzacji (S1, S2, S3) pozwoliło uzyskać zestaw rozwiązań rozłożonych wzduż frontu Pareto:

- metoda S1 (suma ważona) generuje rozwiązania dobrze pokrywające obszary wypukłe frontu,
- metoda S2 (ograniczenia  $\varepsilon$ ) umożliwia dotarcie również do fragmentów niewypukłych,

- metoda S3 (minimalizacja odległości do punktu idealnego) dostarcza kilku punktów skupionych w pobliżu krańcowych regionów kompromisu.

Połączenie wyników wszystkich metod oraz usunięcie punktów zdominowanych pozwoliło zrekonstruować gładką aproksymację frontu Pareto, przedstawioną na wykresie w formie krzywej wygładzonej spline’em. Otrzymana aproksymacja potwierdza, że front ma charakter nieliniowy i zawiera zarówno części wypukłe, jak i lokalnie wklęsłe.

### 3.2 Problem trzykryterialny



W przypadku funkcji  $G : R^3 \rightarrow R^3$  wyniki zaprezentowano na trójwymiarowym wykresie punktowym. Losowa chmura próbek ponownie umożliwia wyznaczenie struktury przestrzeni celów i przybliżenie punktów odniesienia (ideal, nadir).

Uzyskane rozwiązania metod skalaryzacji prezentują różny stopień pokrycia frontu:

- metoda S1 generuje dużą liczbę punktów, jednak koncentruje się głównie na częściach wypukłych,
- metoda S2, dzięki dwóm zmiennym ograniczającym, zapewnia bardziej równomierne pokrycie frontu,

- metoda S3 dostarcza pojedynczy punkt w pobliżu optimum względem punktu idealnego.

Wyznaczenie zbioru Pareto-niezdominowanego pozwoliło uzyskać dyskretną aproksymację frontu Pareto w trzech wymiarach. Punkty te stanowią reprezentację kompromisów pomiędzy trzema funkcjami celu, a ich rozmieszczenie wskazuje na złożoną geometrię problemu, obejmującą fragmenty zarówno monotoniczne, jak i silnie nieliniiowe.

Uzyskane wykresy umożliwiają porównanie jakości i różnic pomiędzy metodami skalaryzacji oraz ich zdolności do eksploracji przestrzeni rozwiązań. Dane wynikowe zostały dodatkowo zapisane w postaci plików CSV, co umożliwia dalszą analizę numeryczną.

## 4 Podsumowanie i wnioski

W przeprowadzonych eksperymentach zaprezentowano trzy klasyczne metody skalaryzacji stosowane w optymalizacji wielokryterialnej: metodę sumy ważonej (S1), metodę ograniczeń  $\varepsilon$  (S2) oraz metodę minimalizacji odległości od punktu idealnego (S3). Analiza wyników dla problemu dwukryterialnego oraz trzykryterialnego pozwoliła na sformułowanie następujących wniosków:

- **Metoda S1 (suma ważona):** zapewnia równomierne pokrycie frontu Pareto w obszarach wypukłych. W przypadku problemów nieliniiowych lub posiadających fragmenty wklęsłe metoda może nie generować punktów w pełni reprezentujących całą geometrię frontu.
- **Metoda S2 (ograniczenia  $\varepsilon$ ):** umożliwia eksplorację również fragmentów niewypukłych frontu Pareto, zapewniając bardziej kompleksową aproksymację kompromisów między kryteriami. Wadą jest większa liczba obliczeń związana z tworzeniem siatki wartości  $\varepsilon$ .
- **Metoda S3 (minimalizacja odległości do punktu idealnego):** skutecznie identyfikuje punkty w pobliżu ekstremalnych rozwiązań, reprezentujących kompromis w kierunku punktu idealnego. Zaletą jest prostota, jednak metoda nie zapewnia równomiernego pokrycia całego frontu.

Porównując wszystkie metody, można stwierdzić, że najlepszą praktyką jest ich kombinacja: metoda S1 generuje szeroki przegląd frontu, metoda S2 uzupełnia obszary niewypukłe, a metoda S3 pozwala na dokładną lokalizację punktów bliskich optimum idealnemu.

Wyniki eksperymentów pokazują również, że przy wzroście liczby kryteriów (z dwóch do trzech) złożoność frontu Pareto rośnie, co wymaga zastosowania metod zapewniających zarówno równomierne pokrycie przestrzeni, jak i możliwość eksploracji fragmentów wklęszych. Calkowity zbiór punktów Pareto-niezdominowanych stanowi wartościową aproksymację kompromisów i może być wykorzystywany do podejmowania decyzji w kontekście rzeczywistych problemów wielokryterialnych.