AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Zarządzania

# Analiza i optymalizacja funkcji przy użyciu metod analizy przedziałowej

Autorzy: *Marcin Mika, Maciej Tajs, Wojciech Liberacki, Jakub Sornat, Filip Kopańko*

Kierunek studiów: *Informatyka i Ekonometria*

Prowadzący: *dr hab.Iwona Skalna, prof. uczelni*

Kraków, 2024

## 1. Cel projektu

Celem projektu jest zastosowanie różnych metod analizy przedziałowej do globalnej optymalizacji funkcji testowych takich jak f. Rastrigina, Rosenbrocka, liniowa, kwadratowa, wielomianowa trzeciego stopnia i wielomianowa czwartego stopnia. Porównaliśmy uzyskane wyniki i szybkość działania metod. Logika arytmetyki przedziałowej została zaczerpnięta z biblioteki mpmath.iv: Link

## 2. Opis Metod

Przeanalizowano trzy metody przytoczone w badaniu:

**Tradycyjny Algorytm Globalnej Minimalizacji Analizy Przedziałowej z Testem Monotoniczności (TIAM)** polega na dekompozycji przestrzeni poszukiwań na mniejsze podprzedziały i analizie właściwości funkcji w tych przedziałach. Dla każdego podprzedziału oblicza się przedział możliwych wartości funkcji, co daje przedział możliwych wyników dla danej funkcji w tym przedziale. Następnie sprawdza się, czy funkcja jest monotoniczna w danym przedziale, co pozwala na precyzyjniejsze oszacowanie minimalnej wartości funkcji w tym przedziale. Jeśli funkcja jest monotoniczna, można szybko odrzucić przedziały, w których minimalna możliwa wartość funkcji jest większa od najlepszej dotychczas znalezionej wartości minimalnej. Proces ten jest powtarzany, a przedziały są coraz bardziej dzielone, aż do osiągnięcia zadowalającej dokładności wyników. TIAM jest szczególnie efektywny w przypadkach, gdy funkcja jest monotoniczna na dużych obszarach przestrzeni poszukiwań, co pozwala na szybkie odrzucanie dużych przedziałów i tym samym skrócenie czasu obliczeń. Metoda ta, poprzez systematyczne zawężanie przestrzeni poszukiwań, umożliwia precyzyjne znalezienie globalnego minimum funkcji.

**Algorytm Globalnej Minimalizacji Analizy Przedziałowej z Wykorzystaniem Informacji o Gradiencie (IAG)** to zaawansowana metoda optymalizacji, która łączy tradycyjną analizę przedziałową z informacjami o gradientach funkcji. Proces rozpoczyna się od podziału przestrzeni poszukiwań na mniejsze podprzedziały, co umożliwia bardziej szczegółową analizę. Dla każdego przedziału oblicza się przedział możliwych wartości funkcji oraz przedział możliwych wartości gradientu funkcji. Wykorzystując informacje o gradientach, można lepiej oszacować minimum w przedziale, co pozwala na efektywniejsze zawężanie przestrzeni poszukiwań. Gradienty wskazują kierunek największego spadku funkcji, co jest pomocne w szybkim odrzucaniu przedziałów, w których minimalna możliwa wartość funkcji jest większa od najlepszej dotychczas znalezionej wartości minimalnej. Proces jest powtarzany, a przedziały są coraz bardziej dzielone, aż do osiągnięcia wymaganej precyzji wyników. IAG jest szczególnie skuteczny w przypadkach, gdy gradienty funkcji mogą być dokładnie oszacowane i wykorzystane do szybszego zawężania przestrzeni poszukiwań. Metoda ta pozwala na znacznie szybsze i dokładniejsze znajdowanie globalnych minimów funkcji w porównaniu do metod, które nie wykorzystują informacji o gradientach.

**Technika Obliczeń Przedziałowych Zorientowana na Arytmetykę Przedziałową (IAOICT)** to metoda optymalizacji skupiająca się na wykorzystaniu zaawansowanych operacji arytmetyki przedziałowej do znajdowania globalnych minimów funkcji. Proces rozpoczyna się od podziału przestrzeni poszukiwań na mniejsze przedziały, co umożliwia bardziej szczegółową analizę. Następnie, dla każdego przedziału, wykorzystuje się zaawansowane operacje arytmetyki przedziałowej do obliczania przedziałów możliwych wartości funkcji. Techniki te obejmują m.in. przedziałowe obliczenia pochodnych oraz zastosowanie metod redukcji szerokości przedziałów, co pozwala na bardziej precyzyjne oszacowanie wartości funkcji. Przedziały, w których minimalna możliwa wartość funkcji jest większa od najlepszej dotychczas znalezionej wartości minimalnej, są odrzucane, co pozwala na efektywne zawężanie przestrzeni poszukiwań. Proces ten jest powtarzany z coraz dokładniejszym podziałem przedziałów i bardziej zaawansowanymi obliczeniami arytmetyki przedziałowej, aż do osiągnięcia wymaganej dokładności wyników. IAOICT wykorzystuje zaawansowane techniki arytmetyki przedziałowej, co pozwala na precyzyjniejsze i efektywniejsze znajdowanie minimów funkcji w porównaniu do prostszych metod przedziałowych. Dzięki zastosowaniu zaawansowanych technik obliczeń przedziałowych, metoda ta jest szczególnie efektywna w bardziej skomplikowanych przypadkach, gdzie tradycyjne metody mogą być mniej dokładne lub wolniejsze.

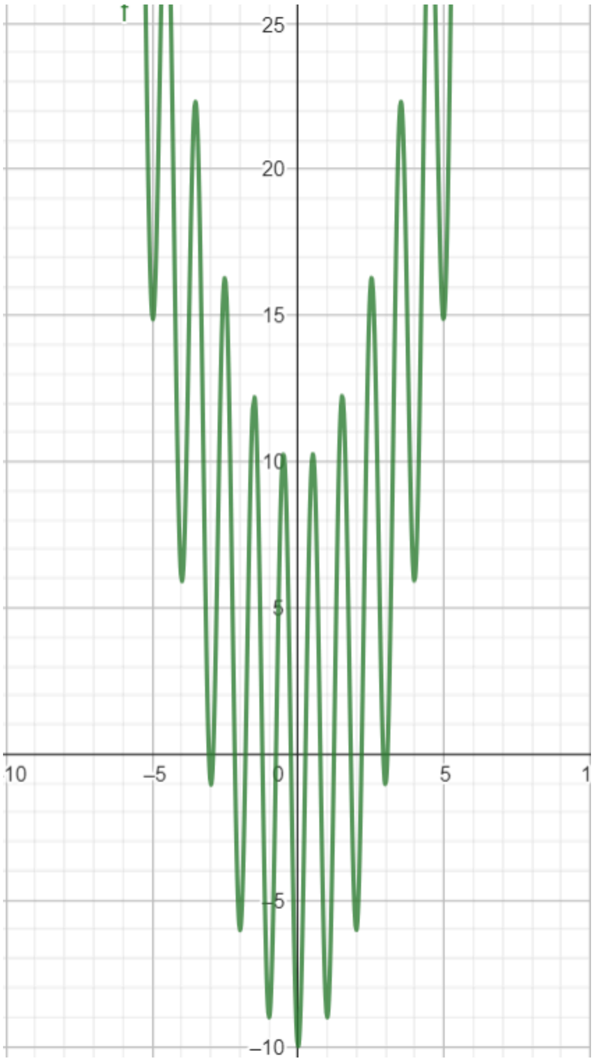
# 3. Opis funkcji

- Funkcja Rastrigina

Funkcja o wzorze:

$$f(\mathbf{x}) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i)\right]$$

Tak prezentuje się wykres funkcji:



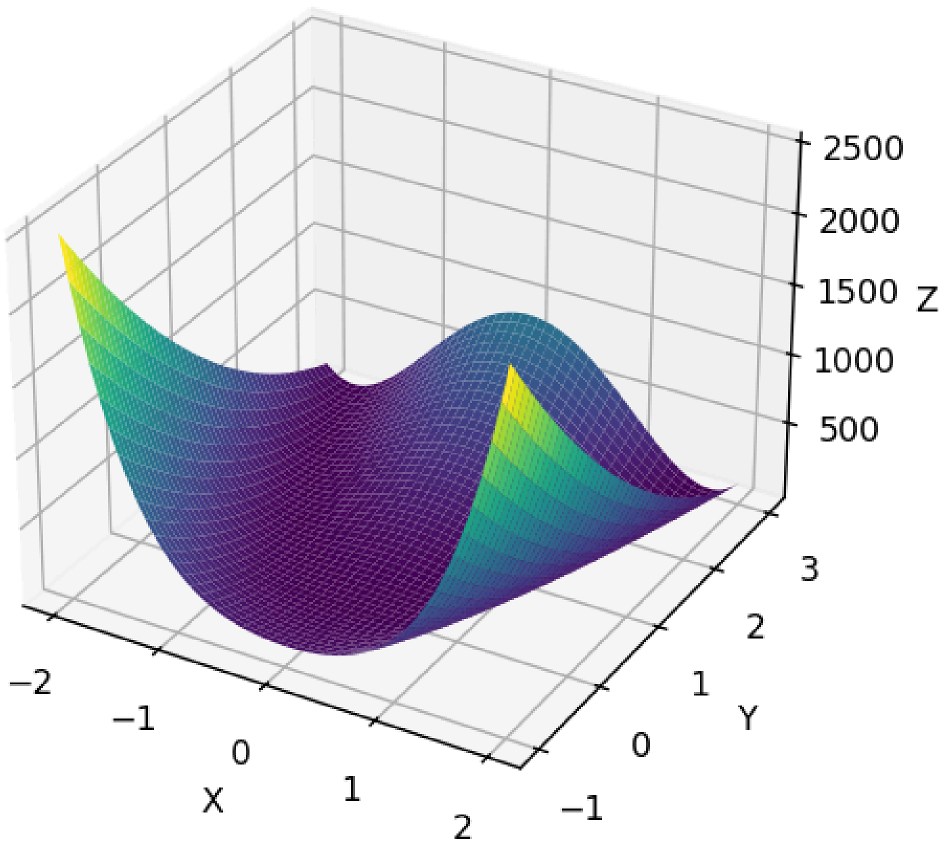

Funkcja osiąga minimum globalne w punkcie o odciętej równej 0.

- Funkcja Rosenbrocka

  Funkcja o wzorze:

$$f(x) = \sum_{i=1}^{N-1} \left[ (1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2 \right] \quad \forall x \in \mathbb{R}^N .$$

  Tak prezentuje się wykres funkcji:

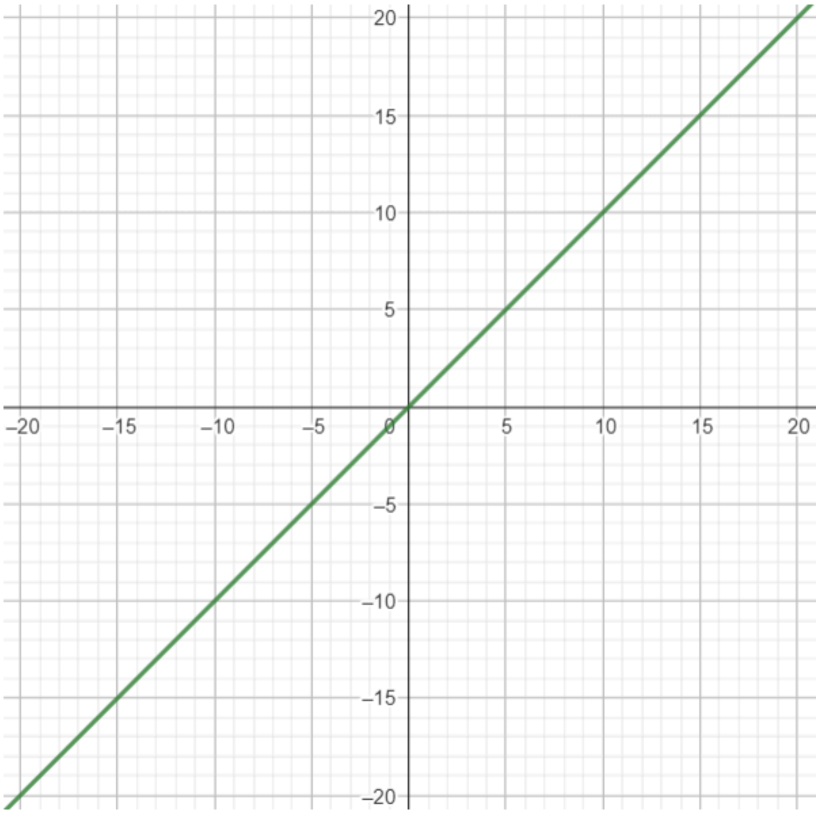


- Funkcja liniowa

  Funkcja o wzorze:

$$f(x) = x$$

  Tak prezentuje się wykres funkcji:

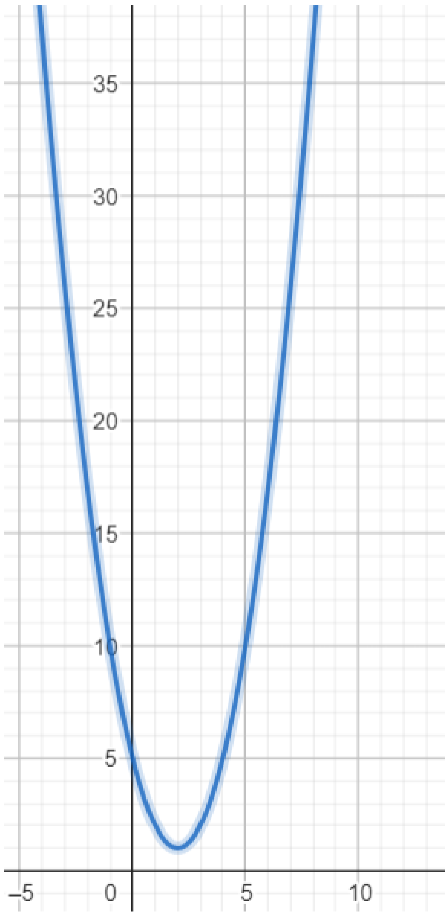


- Funkcja kwadratowa

  Funkcja o wzorze:

$$f(x) = x^2 - 4x + 5$$

  Tak prezentuje się wykres funkcji:



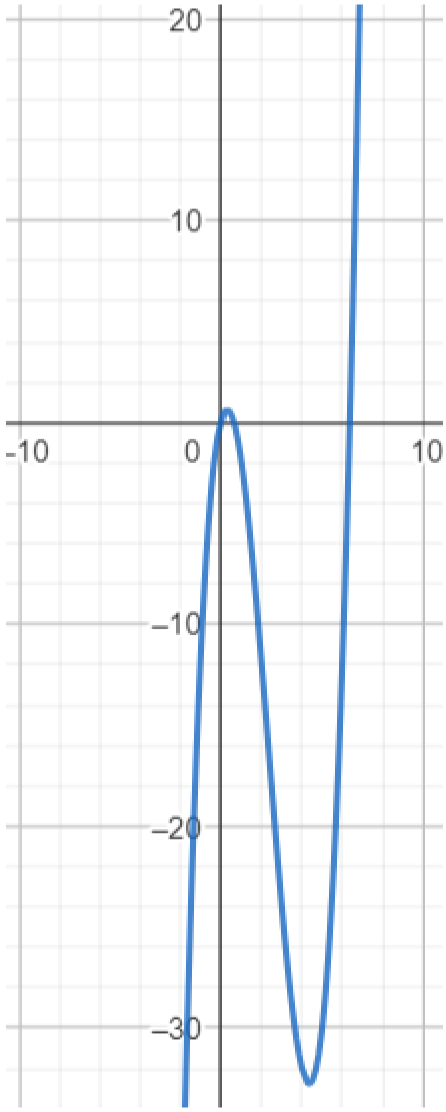

  Funkcja osiąga minimum globalne w punkcie o odciętej równej 2.

- Funkcja wielomianowa 3 stopnia

  Funkcja o wzorze:

$$f(x) = x^3 - 7x^2 + 4x$$

  Tak prezentuje się wykres funkcji:

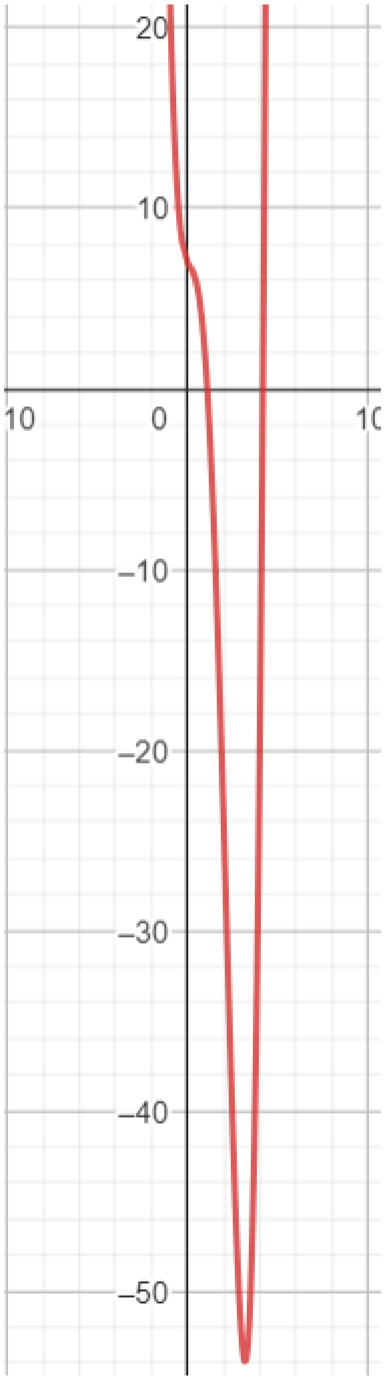


- Funkcja wielomianowa 4 stopnia

  Funkcja o wzorze:

$$f(x) = 2x^4 - 9x^3 + 3x^2 - 2x + 7$$

Tak prezentuje się wykres funkcji:



# 4. Usprawnienie działania

Cały proces początkowo trwał za długo, więc aby przyspieszyć jego działanie wprowadziliśmy kilka usprawnień:

- **Równoległe przetwarzanie**: Użyliśmy *ThreadPoolExecutor* do równoległego przetwarzania przedziałów.
- **Zmniejszenie liczby iteracji**: Ustawiliśmy maksymalną liczbę iteracji na *100*.
- **Zwiększenie tolerancji**: Ustawiliśmy tolerancję na $1 * 10^{-6}$. Tolerancja opisuje o ile nietrafione wyniki są akceptowane.

# 5. Wyniki

Podczas przeprowadzania testów użyte zostały trzy metody, do znalezienia minimum lokalnego dla sześciu funkcji na pięciu różnych przedziałach, co w efekcie dało 90 wyników. Każdy z wyników mieścił się w zadanej tolerancji, więc uznane zostały za akceptowalne.
Tabele wynikowe dla poszczególnych metod prezentują się następująco:

**TIAM:**

| function | x_value | y_value | nfe | interval | time_elapsed |
|---|---|---|---|---|---|
| Rastrigin | 2.000000e+00 | 4.000000e+00 | 4 | (2.0, 7.0) | 0.004069 |
| Rosenbrock | 4.500000e+00 | 0.000000e+00 | 2 | (2.0, 7.0) | 0.001909 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 4 | (2.0, 7.0) | 0.001993 |
| Linear | 2.000000e+00 | 2.000000e+00 | 8 | (2.0, 7.0) | 0.003000 |
| Third degree polynomial | 4.360921e+00 | -3.274535e+01 | 6 | (2.0, 7.0) | 0.002983 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 18 | (2.0, 7.0) | 0.003991 |
| Rastrigin | 0.000000e+00 | 0.000000e+00 | 2 | (-1.0, 1.0) | 0.002005 |
| Rosenbrock | 0.000000e+00 | 0.000000e+00 | 2 | (-1.0, 1.0) | 0.001513 |
| Quadratic | 1.000000e+00 | 2.000000e+00 | 4 | (-1.0, 1.0) | 0.001468 |
| Linear | -1.000000e+00 | -1.000000e+00 | 4 | (-1.0, 1.0) | 0.001173 |
| Third degree polynomial | -1.000000e+00 | -1.200000e+01 | 4 | (-1.0, 1.0) | 0.001945 |
| Fourth degree polynomial | 1.000000e+00 | 1.000000e+00 | 4 | (-1.0, 1.0) | 0.001868 |
| Rastrigin | -4.000000e+00 | 1.600000e+01 | 4 | (-13.0, -4.0) | 0.001732 |
| Rosenbrock | -8.500000e+00 | 0.000000e+00 | 2 | (-13.0, -4.0) | 0.000966 |
| Quadratic | -4.000000e+00 | 3.700000e+01 | 4 | (-13.0, -4.0) | 0.001997 |
| Linear | -1.300000e+01 | -1.300000e+01 | 8 | (-13.0, -4.0) | 0.002129 |
| Third degree polynomial | -1.300000e+01 | -3.432000e+03 | 4 | (-13.0, -4.0) | 0.002094 |
| Fourth degree polynomial | -4.000000e+00 | 1.151000e+03 | 4 | (-13.0, -4.0) | 0.000875 |
| Rastrigin | -1.864986e-08 | 6.927792e-14 | 10 | (-5.0, 9.0) | 0.001977 |
| Rosenbrock | 2.000000e+00 | 0.000000e+00 | 2 | (-5.0, 9.0) | 0.001027 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 2 | (-5.0, 9.0) | 0.002283 |
| Linear | -5.000000e+00 | -5.000000e+00 | 10 | (-5.0, 9.0) | 0.001678 |
| Third degree polynomial | 4.360921e+00 | -3.274535e+01 | 18 | (-5.0, 9.0) | 0.003560 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 14 | (-5.0, 9.0) | 0.002725 |
| Rastrigin | 0.000000e+00 | 0.000000e+00 | 2 | (-10000.0, 10000.0) | 0.000999 |
| Rosenbrock | 0.000000e+00 | 0.000000e+00 | 2 | (-10000.0, 10000.0) | 0.002016 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 8 | (-10000.0, 10000.0) | 0.001948 |
| Linear | -1.000000e+04 | -1.000000e+04 | 20 | (-10000.0, 10000.0) | 0.003017 |
| Third degree polynomial | -1.000000e+04 | -1.000700e+12 | 14 | (-10000.0, 10000.0) | 0.001995 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 20 | (-10000.0, 10000.0) | 0.004228 |

## IAG:

| function | x_value | y_value | nfe | interval | time_elapsed |
|---|---|---|---|---|---|
| Rastrigin | 2.000000e+00 | 4.000000e+00 | 2 | (2.0, 7.0) | 0.001994 |
| Rosenbrock | 4.500000e+00 | 0.000000e+00 | 1 | (2.0, 7.0) | 0.000610 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 2 | (2.0, 7.0) | 0.001995 |
| Linear | 2.000000e+00 | 2.000000e+00 | 3 | (2.0, 7.0) | 0.001862 |
| Third degree polynomial | 4.360921e+00 | -3.274535e+01 | 3 | (2.0, 7.0) | 0.001994 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 9 | (2.0, 7.0) | 0.002995 |
| Rastrigin | 0.000000e+00 | 0.000000e+00 | 1 | (-1.0, 1.0) | 0.001143 |
| Rosenbrock | 0.000000e+00 | 0.000000e+00 | 1 | (-1.0, 1.0) | 0.001853 |
| Quadratic | 1.000000e+00 | 2.000000e+00 | 2 | (-1.0, 1.0) | 0.000994 |
| Linear | -1.000000e+00 | -1.000000e+00 | 2 | (-1.0, 1.0) | 0.001008 |
| Third degree polynomial | -1.000000e+00 | -1.200000e+01 | 2 | (-1.0, 1.0) | 0.001041 |
| Fourth degree polynomial | 1.000000e+00 | 1.000000e+00 | 2 | (-1.0, 1.0) | 0.001266 |
| Rastrigin | -4.000000e+00 | 1.600000e+01 | 2 | (-13.0, -4.0) | 0.001881 |
| Rosenbrock | -8.500000e+00 | 0.000000e+00 | 1 | (-13.0, -4.0) | 0.001001 |
| Quadratic | -4.000000e+00 | 3.700000e+01 | 2 | (-13.0, -4.0) | 0.001991 |
| Linear | -1.300000e+01 | -1.300000e+01 | 4 | (-13.0, -4.0) | 0.001000 |
| Third degree polynomial | -1.300000e+01 | -3.432000e+03 | 2 | (-13.0, -4.0) | 0.002069 |
| Fourth degree polynomial | -4.000000e+00 | 1.151000e+03 | 2 | (-13.0, -4.0) | 0.001411 |
| Rastrigin | -1.164859e-08 | 2.664535e-14 | 6 | (-5.0, 9.0) | 0.001725 |
| Rosenbrock | 2.000000e+00 | 0.000000e+00 | 1 | (-5.0, 9.0) | 0.001007 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 1 | (-5.0, 9.0) | 0.001667 |
| Linear | -5.000000e+00 | -5.000000e+00 | 5 | (-5.0, 9.0) | 0.001034 |
| Third degree polynomial | 4.360921e+00 | -3.274535e+01 | 9 | (-5.0, 9.0) | 0.002487 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 7 | (-5.0, 9.0) | 0.001876 |
| Rastrigin | 0.000000e+00 | 0.000000e+00 | 1 | (-10000.0, 10000.0) | 0.000758 |
| Rosenbrock | 0.000000e+00 | 0.000000e+00 | 1 | (-10000.0, 10000.0) | 0.002006 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 4 | (-10000.0, 10000.0) | 0.001698 |
| Linear | -1.000000e+04 | -1.000000e+04 | 3 | (-10000.0, 10000.0) | 0.001487 |
| Third degree polynomial | -1.000000e+04 | -1.000700e+12 | 7 | (-10000.0, 10000.0) | 0.001770 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 10 | (-10000.0, 10000.0) | 0.001984 |

**IAOICT:**

| function | x_value | y_value | nfe | interval | time_elapsed |
|---|---|---|---|---|---|
| Rastrigin | 2.000000e+00 | 4.000000e+00 | 570 | (2.0, 7.0) | 0.149596 |
| Rosenbrock | 4.500000e+00 | 0.000000e+00 | 200 | (2.0, 7.0) | 0.047872 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 400 | (2.0, 7.0) | 0.070883 |
| Linear | 2.000000e+00 | 2.000000e+00 | 408 | (2.0, 7.0) | 0.069741 |
| Third degree polynomial | 4.360921e+00 | -3.274535e+01 | 438 | (2.0, 7.0) | 0.073802 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 454 | (2.0, 7.0) | 0.075797 |
| Rastrigin | 0.000000e+00 | 0.000000e+00 | 446 | (-1.0, 1.0) | 0.075527 |
| Rosenbrock | 0.000000e+00 | 0.000000e+00 | 200 | (-1.0, 1.0) | 0.047060 |
| Quadratic | 1.000000e+00 | 2.000000e+00 | 400 | (-1.0, 1.0) | 0.069628 |
| Linear | -1.000000e+00 | -1.000000e+00 | 400 | (-1.0, 1.0) | 0.066821 |
| Third degree polynomial | -1.000000e+00 | -1.200000e+01 | 400 | (-1.0, 1.0) | 0.067878 |
| Fourth degree polynomial | 1.000000e+00 | 1.000000e+00 | 400 | (-1.0, 1.0) | 0.069754 |
| Rastrigin | -4.000000e+00 | 1.600000e+01 | 690 | (-13.0, -4.0) | 0.102276 |
| Rosenbrock | -8.500000e+00 | 0.000000e+00 | 200 | (-13.0, -4.0) | 0.045870 |
| Quadratic | -4.000000e+00 | 3.700000e+01 | 400 | (-13.0, -4.0) | 0.068817 |
| Linear | -1.300000e+01 | -1.300000e+01 | 420 | (-13.0, -4.0) | 0.067817 |
| Third degree polynomial | -1.300000e+01 | -3.432000e+03 | 400 | (-13.0, -4.0) | 0.067820 |
| Fourth degree polynomial | -4.000000e+00 | 1.151000e+03 | 400 | (-13.0, -4.0) | 0.068796 |
| Rastrigin | -4.972014e-09 | 3.552714e-15 | 788 | (-5.0, 9.0) | 0.113320 |
| Rosenbrock | 2.000000e+00 | 0.000000e+00 | 200 | (-5.0, 9.0) | 0.045876 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 398 | (-5.0, 9.0) | 0.066823 |
| Linear | -5.000000e+00 | -5.000000e+00 | 422 | (-5.0, 9.0) | 0.067820 |
| Third degree polynomial | -5.000000e+00 | -3.200000e+02 | 436 | (-5.0, 9.0) | 0.070775 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 456 | (-5.0, 9.0) | 0.072841 |
| Rastrigin | 0.000000e+00 | 0.000000e+00 | 398 | (-10000.0, 10000.0) | 0.083775 |
| Rosenbrock | 0.000000e+00 | 0.000000e+00 | 200 | (-10000.0, 10000.0) | 0.048869 |
| Quadratic | 2.000000e+00 | 1.000000e+00 | 422 | (-10000.0, 10000.0) | 0.074834 |
| Linear | -1.000000e+04 | -1.000000e+04 | 1478 | (-10000.0, 10000.0) | 0.158573 |
| Third degree polynomial | -1.000000e+04 | -1.000700e+12 | 410 | (-10000.0, 10000.0) | 0.068821 |
| Fourth degree polynomial | 3.162864e+00 | -5.392956e+01 | 638 | (-10000.0, 10000.0) | 0.090757 |

Następnie metody zostały poddane porównaniu ze względu na ilość iteracji potrzebnych do optymalizacji oraz tego, ile czasu to zajęło.

| | iterations | time_elapsed |
|---|---|---|
| **method** | | |
| **TIAM** | 210 | 0.065164 |
| **IAG** | 98 | 0.047608 |
| **IAOICT** | 13472 | 2.268841 |

Najlepsze wyniki zapewniła metoda IAG - szczególnie pod względem ilości iteracji potrzebnych do otrzymania zadowalającego wyniku. Nieznacznie gorsza była metoda TIAM, natomiast zwraca uwagę zdecydowanie wyższa ilość iteracji metody IAOICT, oraz wyraźnie dłuższy czas potrzebny, bo otrzymać zadowalające minimum.
Czasy działania w metodach TIAM oraz IAG wydają się być stałe, podczas gdy w IAOICT można dostrzec znacznie większe wahania - może to wskazywać na zależność efektywności od długości przedziału. Aby to zbadać wyizolowane zostały kolumny nfe, elapsed_time oraz długość przedziału:

| | nfe | time_elapsed | interval_length |
|---|---|---|---|
| **function** | | | |
| **Rastrigin** | 570 | 0.149596 | 5 |
| **Rosenbrock** | 200 | 0.047872 | 5 |
| **Quadratic** | 400 | 0.070883 | 5 |
| **Linear** | 408 | 0.069741 | 5 |
| **Third degree polynomial** | 438 | 0.073802 | 5 |
| **Fourth degree polynomial** | 454 | 0.075797 | 5 |
| **Rastrigin** | 446 | 0.075527 | 2 |
| **Rosenbrock** | 200 | 0.047060 | 2 |
| **Quadratic** | 400 | 0.069628 | 2 |
| **Linear** | 400 | 0.066821 | 2 |

Następnie zmierzone zostały korelacje pomiędzy tymi zmiennymi.

| | nfe | time_elapsed | interval_length |
|---|---|---|---|
| **nfe** | 1.000000 | 0.873678 | 0.305207 |
| **time_elapsed** | 0.873678 | 1.000000 | 0.234593 |
| **interval_length** | 0.305207 | 0.234593 | 1.000000 |

Dane te faktycznie pokazują słabą dodatnią korelację pomiędzy wielkością przedziału, a potrzebną ilością iteracji oraz czasu, by znaleźć minimum. Biorąc pod uwagę wszystkie powyższe czynniki oraz dane najlepszą metodą wydaje się być IAG, nieznacznie gorzej radzi sobie TIAM, zaś znacznie odstaje IAOICT, która wymaga wielokrotnie więcej iteracji, oraz kilkukrotnie dłuższego czasu działania, by znaleźć minimum, którego błąd jest zadowalający.