

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Przeszukiwanie i optymalizacja
Semestr 23Z
Sprawozdanie z projektu

Temat nr 6
Algorytm roju cząstek z modyfikacjami

Natalia Iwańska, Mikołaj Wewiór

Warszawa, 2024

Spis treści

1. Wstęp	2
1.1. Opis algorytmu roju cząstek	2
1.2. Implementacja algorytmu	3
1.2.1. Inicjalizacja	3
1.2.2. Ograniczenia	3
1.3. Modyfikacje	3
1.3.1. Modyfikacja 1	3
1.3.2. Modyfikacja 2	4
1.3.3. Modyfikacja 3	4
2. Opis testów i funkcji	5
3. Przebiegi	8
3.1. Funkcja Wielomianowa	9
3.2. Booth	10
3.3. Rosenbrock	11
3.4. Ackley	12
3.5. Michalewicz	13
3.6. Holder Table	14
4. Analiza wyników	16
4.1. Krzywe zbieżności	16
4.2. Krzywe ECDF	18
4.3. Wykresy pudełkowe	20
5. Wyższe wymiary	22
5.1. 5 wymiarów	22
Funkcja wielomianowa	22
Funkcja Rosenbrock	22
Funkcja Ackley'a	22
Funkcja Michalewicz'a	23
5.2. 10 wymiarów	23
Funkcja wielomianowa	23
Funkcja Rosenbrock	23
Funkcja Ackley'a	23
Funkcja Michalewicz'a	24
5.3. Analiza wyników	24
5.3.1. Krzywe zbieżności	24
5.3.2. Krzywe ECDF	26
5.3.3. Wykresy pudełkowe	27
6. Podsumowanie	29
6.1. Porównania tabelaryczne	29
6.1.1. Najlepszy wynik	29
6.1.2. Najlepsza średnia	29
6.1.3. Krzywe zbieżności	29
6.1.4. Wykresy pudełkowe	29
6.1.5. Sumacyjne wyniki	30
6.2. Podsumowanie	31
7. Bibliografia	32

1. Wstęp

Celem naszego projektu jest implementacja algorytmu roju cząstek z modyfikacjami dotyczącymi współczynnika bezwładności, a następnie porównanie wyników działania algorytmu z jego podstawową wersją.

Pracę rozpoczęliśmy nad zaimplementowaniem standardowej wersji algorytmu. Gdy udało się zrealizować tę część, musieliśmy się zastanowić nad możliwymi modyfikacjami współczynnika bezwładności. Po odpowiednim przejrzeniu literatury doszliśmy do dwóch głównych sposobów zmiany tego parametru.

Pierwszym z nich okazała się zmiana wartości współczynnika wraz z kolejnymi iteracjami. Mieliśmy tutaj do rozważenia dwie kwestie. Pierwszą z nich był wybór funkcji, która opisywałaby zmianę parametru w czasie - w naszym przypadku padło na najprostszą realizację, czyli funkcję liniową. Drugim kryterium było to czy funkcja jest rosnąca czy malejąca w czasie; sprawdziliśmy dwie opcje.

Innym sposobem na zmienność współczynnika bezwładności w czasie była modyfikacja, która przypisywała odpowiednie wartości w zależności od jakości rozwiązania. Tak jak w pierwszym przypadku tutaj też mogliśmy wybrać to, czy parametr ma się zwiększać, czy zmniejszać w zależności od wyniku. Zdecydowaliśmy się na jedną z opcji mianowicie na malejący współczynnik, dla lepszych rozwiązań.

1.1. Opis algorytmu roju cząstek

Algorytm Roju Cząstek (PSO, ang. *particle swarm optimization*) jest metaheurystycznym algorytmem optymalizacyjnym, inspirowanym zachowaniem stad w przyrodzie, który na początku służył do symulowania zachowań społecznych występujących wśród ławic ryb, czy kłuczy ptaków.

Dla danej funkcji celu $f(\mathbf{x})$, gdzie \mathbf{x} to wektor zmiennych decyzyjnych, PSO korzysta z populacji cząstek poruszających się w przestrzeni poszukiwań, w celu znalezienia optymalnego rozwiązania problemu. Każda cząstka reprezentuje pozycję oraz potencjalne rozwiązanie (wartość w tym punkcie) problemu.

Pozycja cząstki w przestrzeni poszukiwań jest określana przez wektor $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})$, gdzie n to liczba zmiennych decyzyjnych.

Ruch cząstki jest aktualizowany na podstawie dwóch wpływów: jej własnego najlepszego rozwiązania (lokalne) oraz najlepszego rozwiązania w całej populacji (globalne).

Aktualizacja pozycji cząstki jest opisana równaniem:

$$\mathbf{v}_i^{t+1} = w \cdot \mathbf{v}_i^t + c_1 \cdot r_1 \cdot (\mathbf{p}_i^t - \mathbf{x}_i^t) + c_2 \cdot r_2 \cdot (\mathbf{g}^t - \mathbf{x}_i^t) \quad (1.1)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (1.2)$$

gdzie:

- \mathbf{v}_i^t to prędkość cząstki w chwili t ,
- \mathbf{p}_i^t to najlepsze dotąd osiągnięte przez cząstkę rozwiązanie,
- \mathbf{g}^t to najlepsze dotąd osiągnięte w populacji globalne rozwiązanie,
- w to współczynnik bezwładności,
- c_1 i c_2 to współczynniki akceleracji,
- r_1 i r_2 to losowe liczby z przedziału $(0, 1)$.

1.2. Implementacja algorytmu

Algorytm został zaimplementowany głównie w funkcji `PSO.py` w kodach źródłowych dołączonych do dokumentacji projektowej.

1.2.1. Inicjalizacja

Inicjalizacja algorytmu obejmuje wygenerowanie 25 cząstek, a następnie przypisanie każdej cząsteczce losowej początkowej pozycji w przestrzeni poszukiwań, gdzie każda zmienna decyzyjna jest ustawiana na losową wartość z zakresu dziedziny badanej funkcji. Dodatkowo, inicjalizowane są również losowe początkowe prędkości cząsteczki, a także ustawiany jest atrybut reprezentujący najlepszy osiągnięty wynik, który początkowo jest ustawiony na nieskończoność. Te inicjalne parametry stanowią punkt wyjścia dla procesu optymalizacji rojem cząsteczek. Pozostałe parametry algorytmu:

- współczynnik bezwładności : 0.72984
- liczba cząsteczek: 25
- liczba testów: 500
- maksymalna liczba iteracji: 60
- współczynniki akceleracji (c_1 i c_2): 1.496

1.2.2. Ograniczenia

W celu skutecznej obsługi naruszonych ograniczeń dziedziny badanych funkcji w algorytmie optymalizacyjnym, zdecydowaliśmy się zaimplementować technikę rzutowania. Ta metoda umożliwiła utrzymanie rozwiązania w obszarze dopuszczalnym, poprzez rzutowanie współrzędnych znajdujących się poza dziedziną funkcji na współrzędne graniczne. Przyczyniło się to do poprawy działania algorytmu.

1.3. Modyfikacje

Ponieważ projekt dotyczy badania wpływu zmiany parametru bezwładności cząsteczek na wyniki działania algorytmu, wszystkie pozostałe parametry pozostawiliśmy jednakowe. Przyjęte one zostały na podstawie rekomendowanych parametrów, opracowanych w badaniu przez M. Clerc'a oraz J. Kennedy'ego. W podstawowej wersji algorytmu współczynnik bezwładności jest stały. Zaimplementowaliśmy trzy różne modyfikacje algorytmu, które sprawiają, że współczynnik ten jest zmienny w czasie.

1.3.1. Modyfikacja 1

W pierwszej modyfikacji wartość współczynnika rośnie wraz z każdą iteracją algorytmu.

$$\mathbf{w} = w_{\text{init}} \cdot \frac{t}{t_{\text{max}}} + 0.2 \quad (1.3)$$

gdzie:

- w_{init} to początkowa wartość współczynnika bezwładności równa 0.72984
- t to numer aktualnej iteracji
- t_{max} to maksymalna liczba iteracji w naszym przypadku równa 60

Współczynnik startuje ze współczynnikiem bezwładności równym 0.2 i z każdą iteracją rośnie, aż do wartości równej $0.72984 + 0.2$ czyli 0.92984

1.3.2. Modyfikacja 2

W drugiej modyfikacji wartość współczynnika bezwładności zależy od jakości rozwiązania. Im jakość rozwiązania jest bliższa najlepszemu globalnemu rozwiązaniu tym inercja ma mniejszą wartość.

$$\mathbf{w} = (w_{\text{init}} + 0.1) \cdot \frac{dif}{dif + 1} \quad (1.4)$$

gdzie:

- dif to różnica między najlepszym globalnym wynikiem, a najlepszym dotychczasowym wynikiem cząstki.
- w_{init} to początkowa wartość współczynnika bezwładności równa 0.72984

1.3.3. Modyfikacja 3

W drugiej modyfikacji wartość współczynnika maleje wraz z każdą iteracją algorytmu.

$$\mathbf{w} = w_{\text{init}} \cdot \frac{t_{\text{max}} - t}{t_{\text{max}}} + 0.2 \quad (1.5)$$

gdzie:

- w_{init} to początkowa wartość współczynnika bezwładności równa 0.72984

Współczynnik startuje ze współczynnikiem bezwładności równym 0.92984 i z każdą iteracją maleje, aż do wartości równej 0.2.

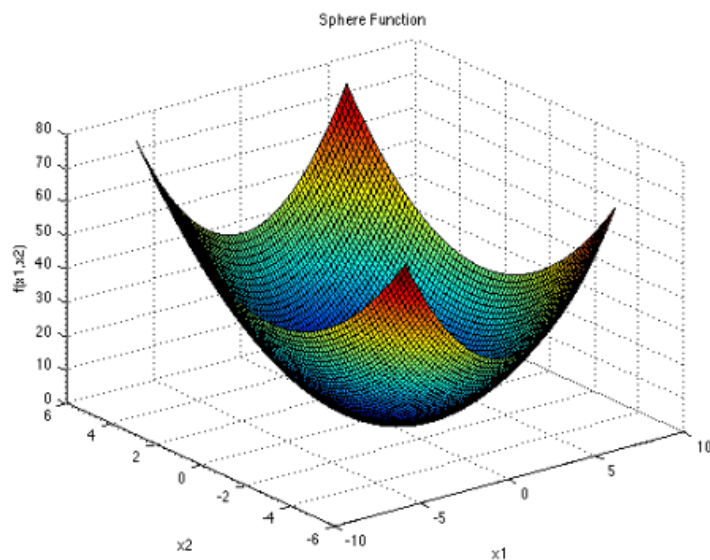
W dalszej części dokumentu rozróżnienie poszczególnych trybów wygląda następująco:

- tryb 1 - brak modyfikacji - stała inercja równa 0.72984
- tryb 2 - modyfikacja 1. inercja rosnąca liniowo w czasie
- tryb 3 - modyfikacja 2. inercja malejąca wraz z poprawą jakości rozwiązania.
- tryb 4 - modyfikacja 3. inercja malejąca liniowo w czasie

2. Opis testów i funkcji

Testy dla algorytmu wykonaliśmy dla sześciu różnych funkcji dwuwymiarowych, tj. wielomianowej (sferycznej), Booth'a, Rosenbrocka, Ackley'a, Michalewicz'a oraz funkcji holder-table. Każda z modyfikacji algorytmu szukała najlepszego rozwiązania dla każdej z powyższych funkcji. Po zakończonych testach dla każdej z funkcji analizowane były wyniki oraz porównywane ich efekty. Poniżej znajdują się wykresy tych funkcji:

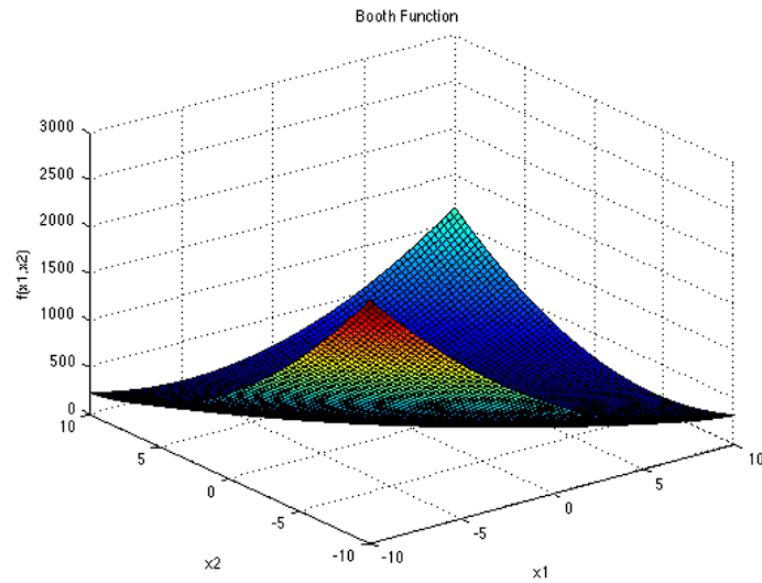
SPHERE FUNCTION



$$f(\mathbf{x}) = \sum_{i=1}^d x_i^2$$

Rys. 2.1: funkcja wielomianowa

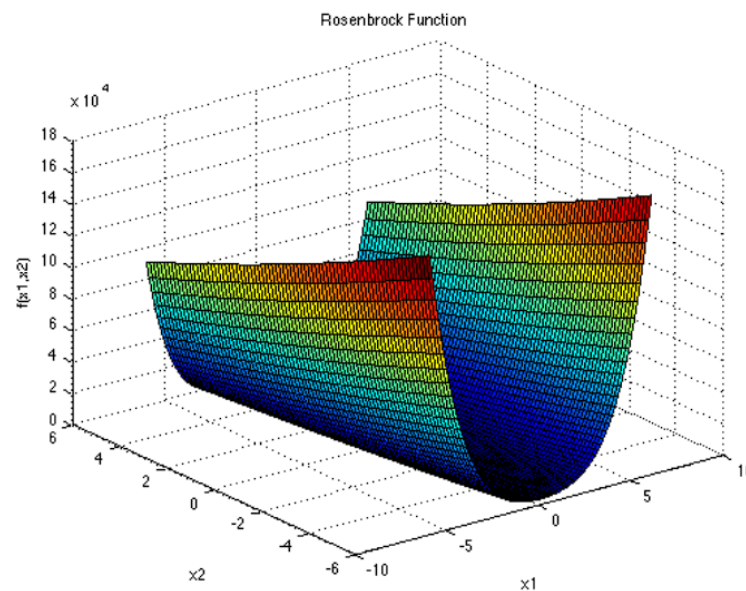
BOOTH FUNCTION



$$f(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

Rys. 2.2: funkcja Bootha

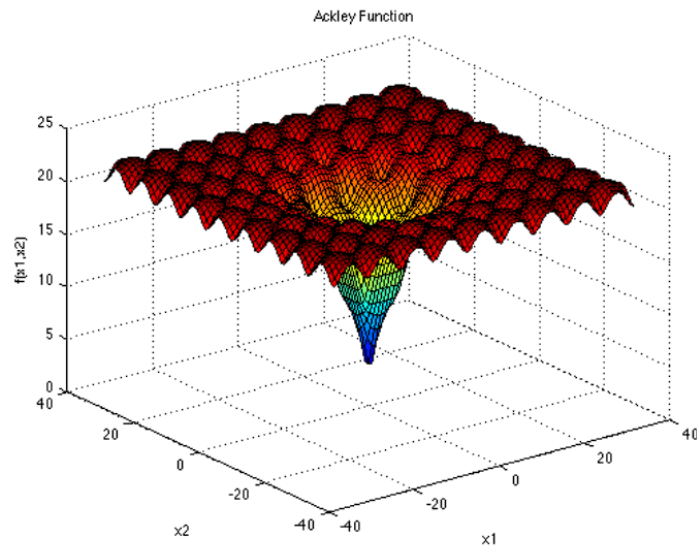
ROSENBROCK FUNCTION



$$f(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

Rys. 2.3: funkcja Rosenbrocka

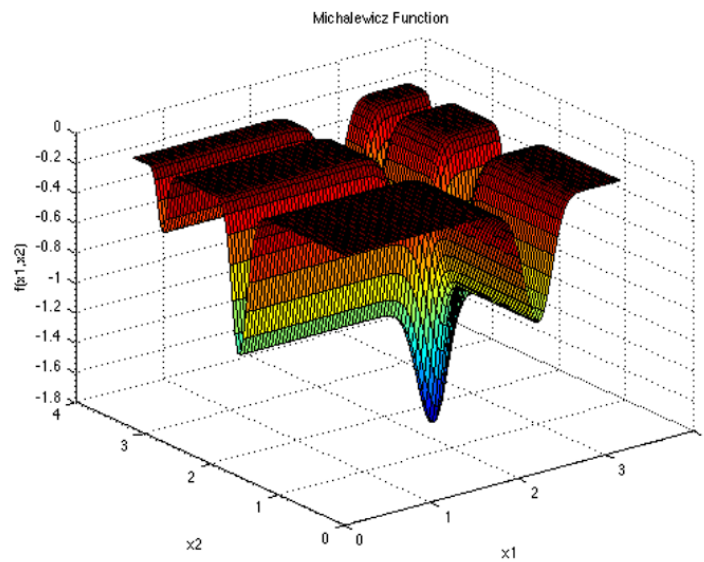
ACKLEY FUNCTION



$$f(\mathbf{x}) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1)$$

Rys. 2.4: funkcja Ackley'a

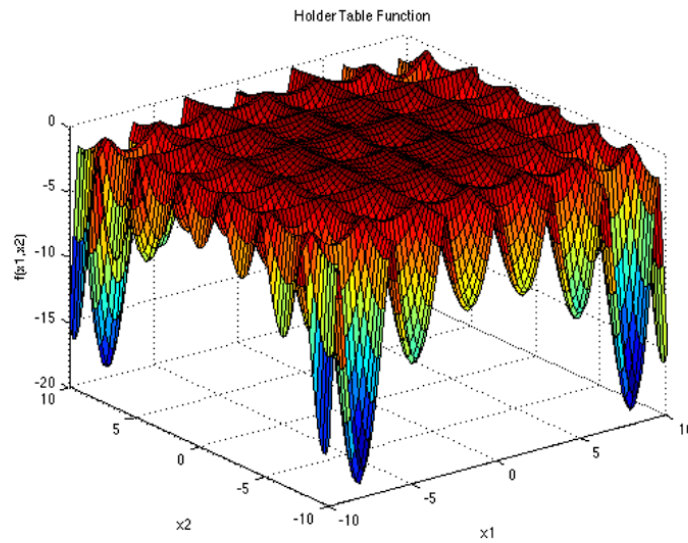
MICHALEWICZ FUNCTION



$$f(\mathbf{x}) = - \sum_{i=1}^d \sin(x_i) \sin^{2m} \left(\frac{\pi x_i^2}{\pi} \right)$$

Rys. 2.5: funkcja Michalewicza

HOLDER TABLE FUNCTION



$$f(\mathbf{x}) = - \left| \sin(x_1) \cos(x_2) \exp \left(\left| 1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right| \right) \right|$$

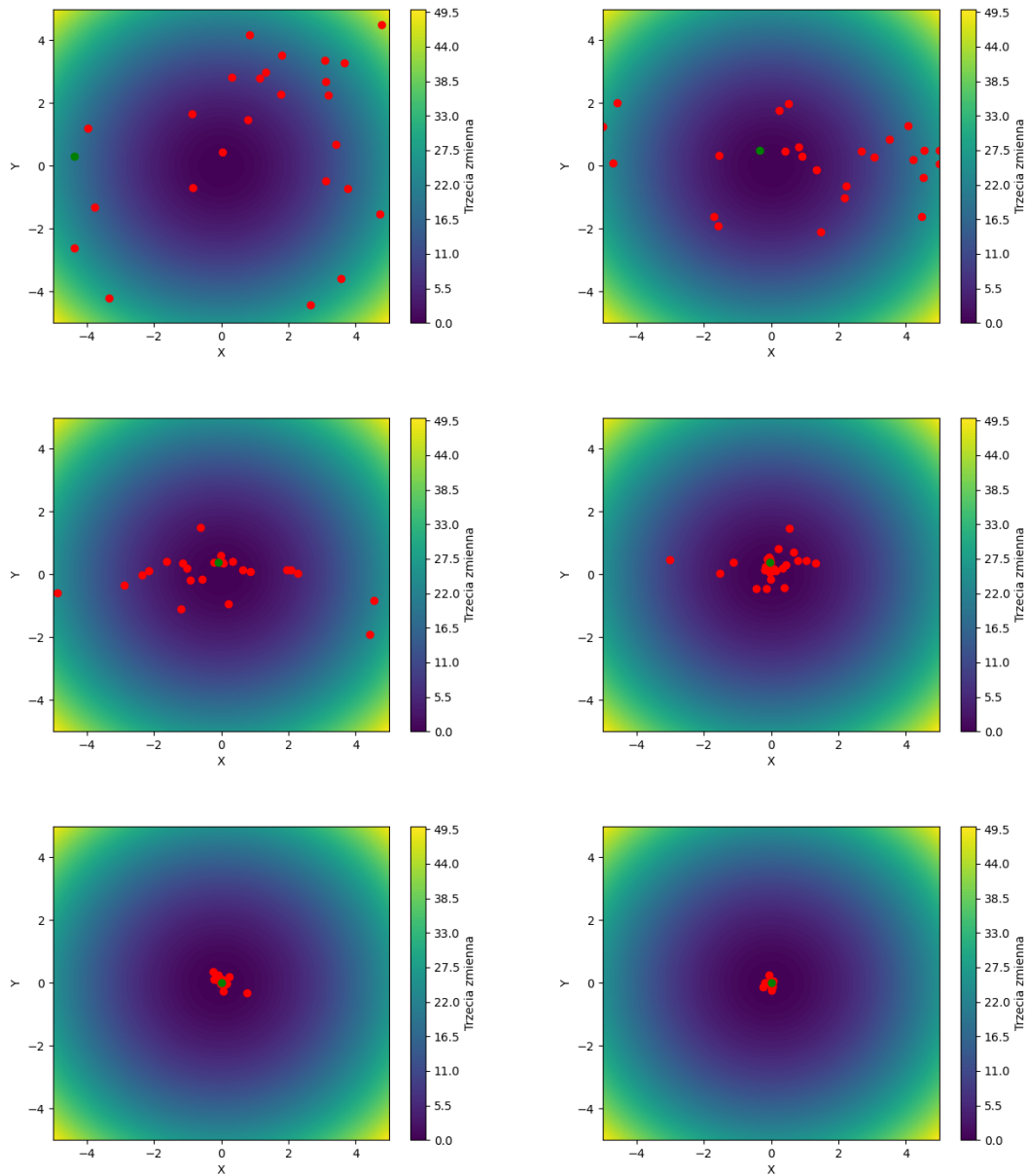
Rys. 2.6: funkcja Holder-Table

3. Przebiegi

Poniżej zostały przedstawione fragmenty wizualizacji przebiegów algorytmu w formie zrzutów wykresu funkcji z zaznaczonymi cząsteczkami w danej iteracji (czerwone punkty) oraz najlepszego globalnego rozwiązania z poprzedniej iteracji (zielony punkt).

W tabelach znajdują się otrzymane wyniki naszych testów. Testy polegały na uruchomieniu algorytmu w każdym trybie i dla każdej funkcji po 500 razy. Z powodu pojedynczych znaczących rozbieżności między wynikami naszych testów a oczekiwanymi wartościami, postanowiliśmy eliminować 5% najgorszych rezultatów w celu uzyskania bardziej reprezentatywnych danych. Poniższe tabele zawierają kolumnę przedstawiającą najgorszy uzyskany wynik, jednak pozostała analiza (między innymi średni wynik) całkowicie pomija wyeliminowane wartości.

3.1. Funkcja Wielomianowa

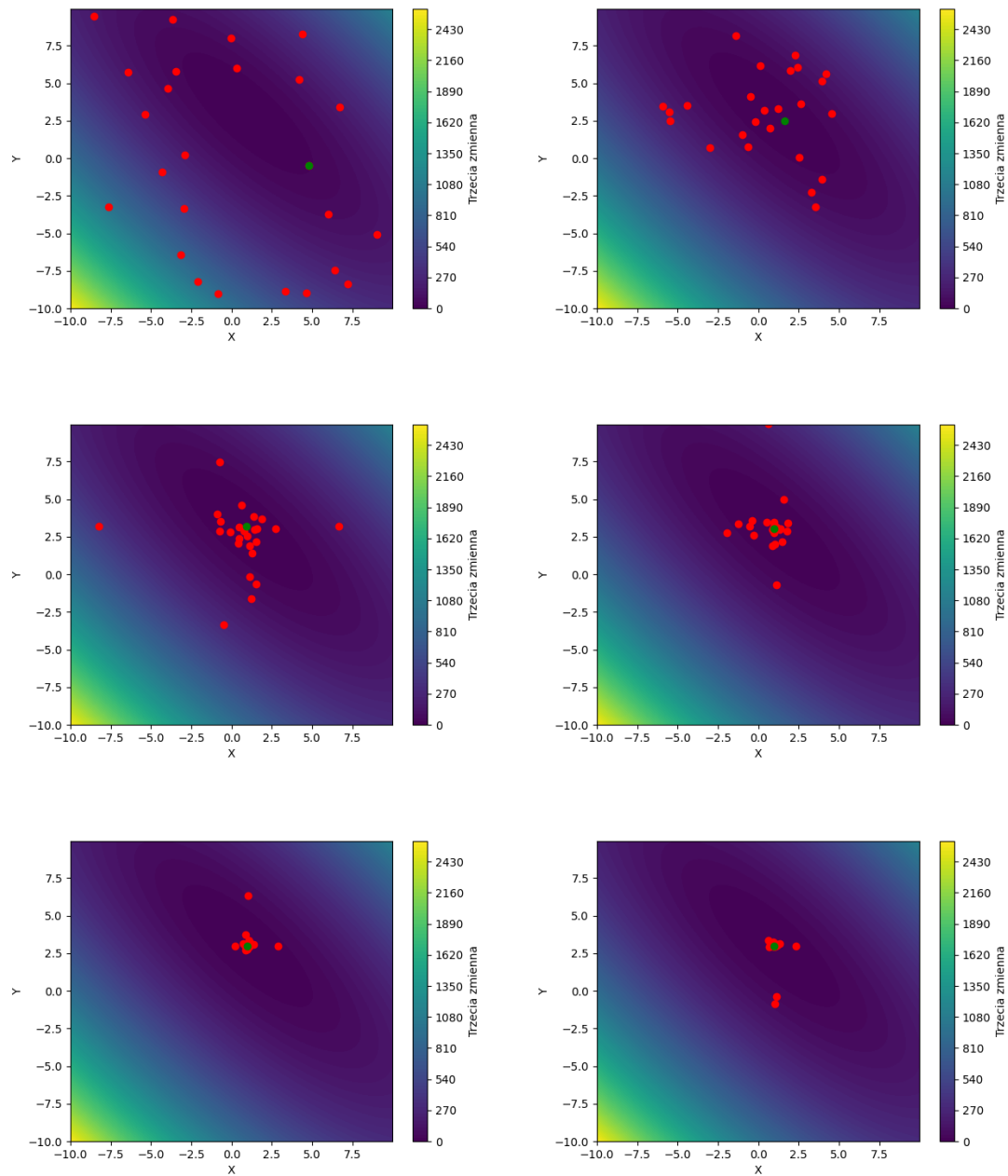


Dziedzina przeszukiwań = $[-5, 5]$. Optymalna wartość = 0, dla punktu (0, 0).

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$3,36 \cdot 10^{-13}$	$3,18 \cdot 10^{-7}$	$1,7 \cdot 10^{-2}$	$5,17 \cdot 10^{-7}$	1
$1,19 \cdot 10^{-18}$	$2,35 \cdot 10^{-14}$	3,71	$3,40 \cdot 10^{-14}$	2
$8,02 \cdot 10^{-42}$	$4,59 \cdot 10^{-15}$	1,13	$2,09 \cdot 10^{-13}$	3
$9,92 \cdot 10^{-10}$	$1,83 \cdot 10^{-12}$	$8,26 \cdot 10^{-5}$	$3,03 \cdot 10^{-12}$	4

Tab. 3.1: Porównanie wyników dla funkcji wielomianowej w różnych trybach.

3.2. Booth

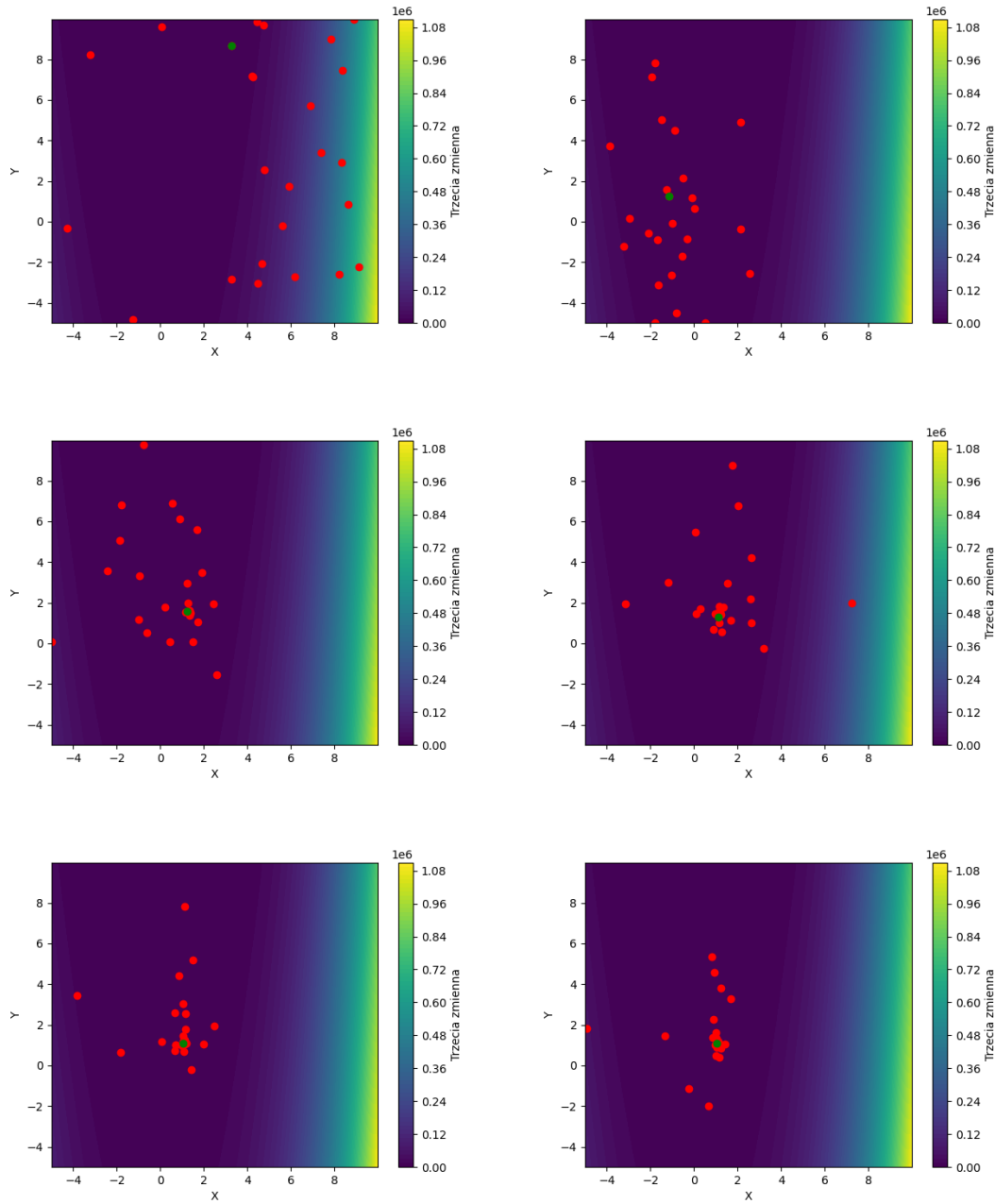


Dziedzina przeszukiwań = $[-10, 10]$. Optymalna wartość = 0, dla punktu $(1, 3)$.

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$2,01 \cdot 10^{-9}$	$3,29 \cdot 10^{-6}$	$1,00 \cdot 10^{-4}$	$5,51 \cdot 10^{-6}$	1
$5,81 \cdot 10^{-16}$	$8,97 \cdot 10^{-13}$	$5,25 \cdot 10^{-11}$	$1,43 \cdot 10^{-12}$	2
$3,32 \cdot 10^{-27}$	$1,02 \cdot 10^{-6}$	$1,91 \cdot 10^{-3}$	$6,14 \cdot 10^{-6}$	3
$3,00 \cdot 10^{-14}$	$6,84 \cdot 10^{-11}$	$3,54 \cdot 10^{-9}$	$9,27 \cdot 10^{-11}$	4

Tab. 3.2: Porównanie wyników dla funkcji Booth w różnych trybach.

3.3. Rosenbrock

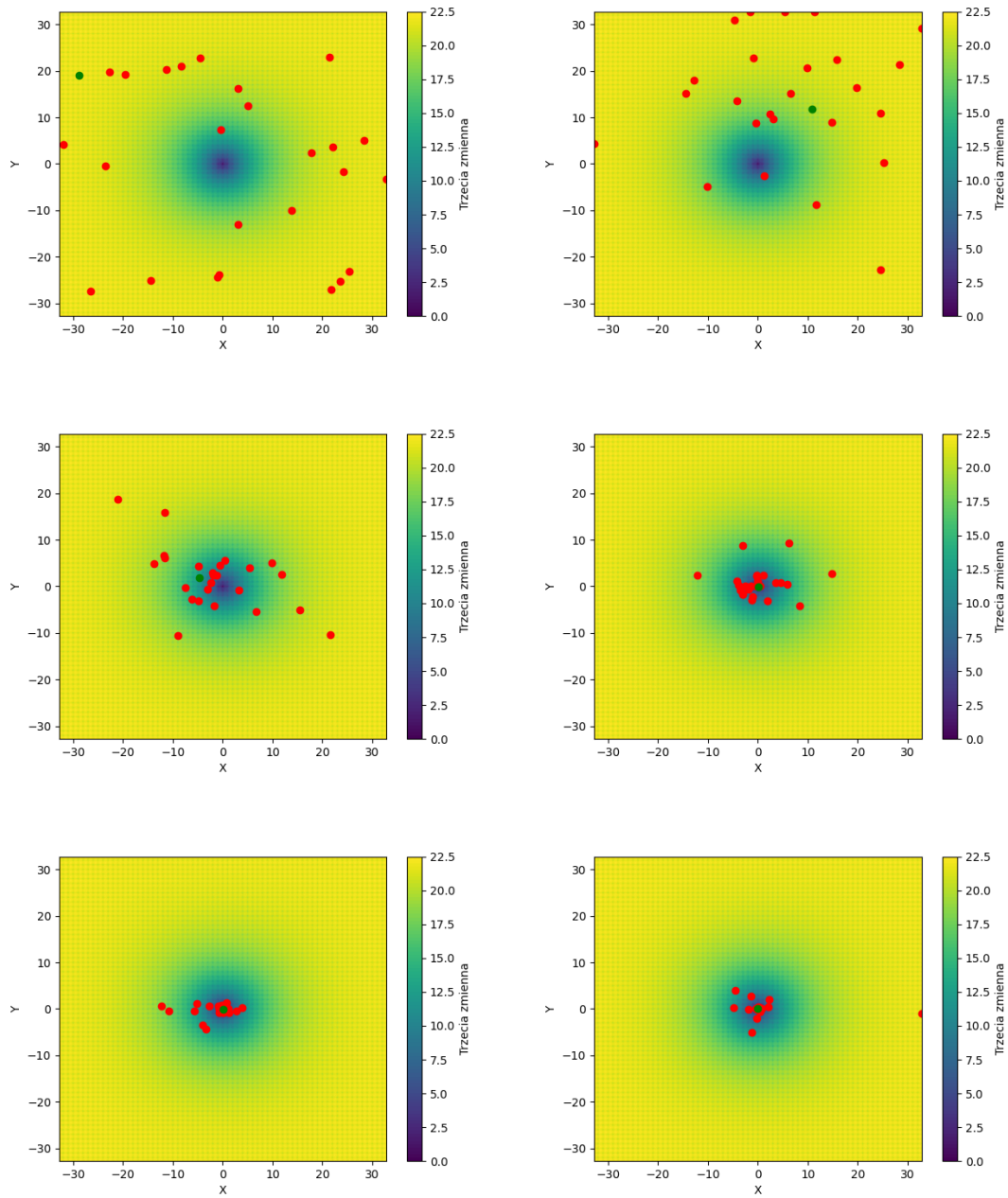


Dziedzina przeszukiwań = $[-5, 10]$. Optymalna wartość = 0, dla punktu (1, 1).

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$1,27 \cdot 10^{-7}$	$1,36 \cdot 10^{-2}$	$1,71 \cdot 10^1$	$3,59 \cdot 10^{-2}$	1
$6,81 \cdot 10^{-9}$	$2,35 \cdot 10^{-2}$	$1,74 \cdot 10^1$	$4,49 \cdot 10^{-2}$	2
$1,94 \cdot 10^{-8}$	$2,00 \cdot 10^{-2}$	5,89	$5,14 \cdot 10^{-2}$	3
$6,18 \cdot 10^{-9}$	$2,48 \cdot 10^{-2}$	$1,48 \cdot 10^1$	$1,23 \cdot 10^{-1}$	4

Tab. 3.3: Porównanie wyników dla funkcji Rosenbrock w różnych trybach.

3.4. Ackley

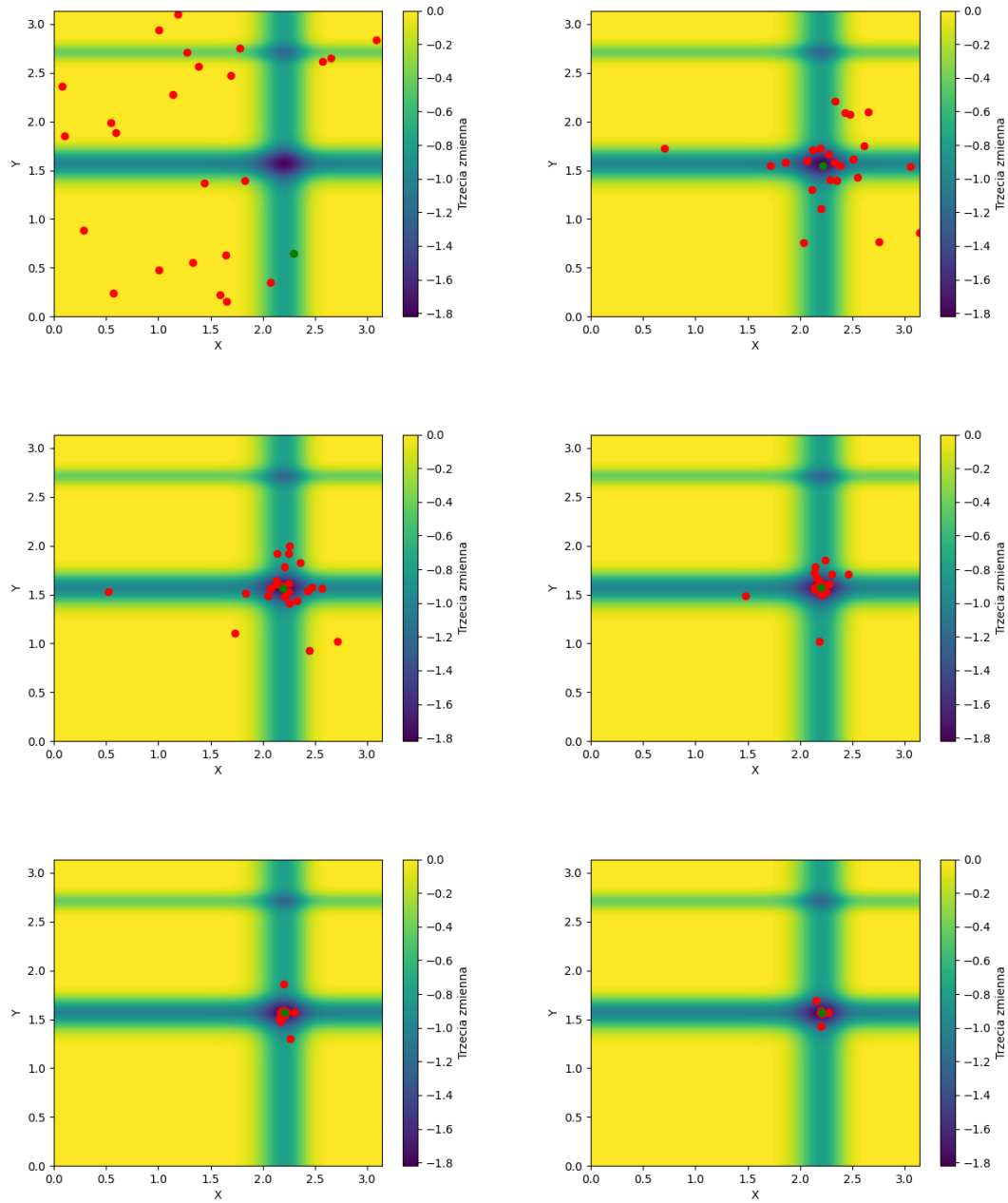


Dziedzina przeszukiwań = $[-32.768, 32.768]$. Optymalna wartość = 0, dla punktu (0, 0).

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$1,02 \cdot 10^{-4}$	$8,10 \cdot 10^{-3}$	$1,59 \cdot 10^1$	$7,54 \cdot 10^{-3}$	1
$2,78 \cdot 10^{-8}$	$1,95 \cdot 10^{-6}$	$1,36 \cdot 10^1$	$1,82 \cdot 10^{-6}$	2
$4,94 \cdot 10^{-13}$	$4,20 \cdot 10^{-6}$	$1,08 \cdot 10^{-2}$	$3,35 \cdot 10^{-5}$	3
$8,06 \cdot 10^{-7}$	$2,08 \cdot 10^{-5}$	$1,19 \cdot 10^1$	$1,88 \cdot 10^{-5}$	4

Tab. 3.4: Porównanie wyników dla funkcji Ackley w różnych trybach.

3.5. Michalewicz

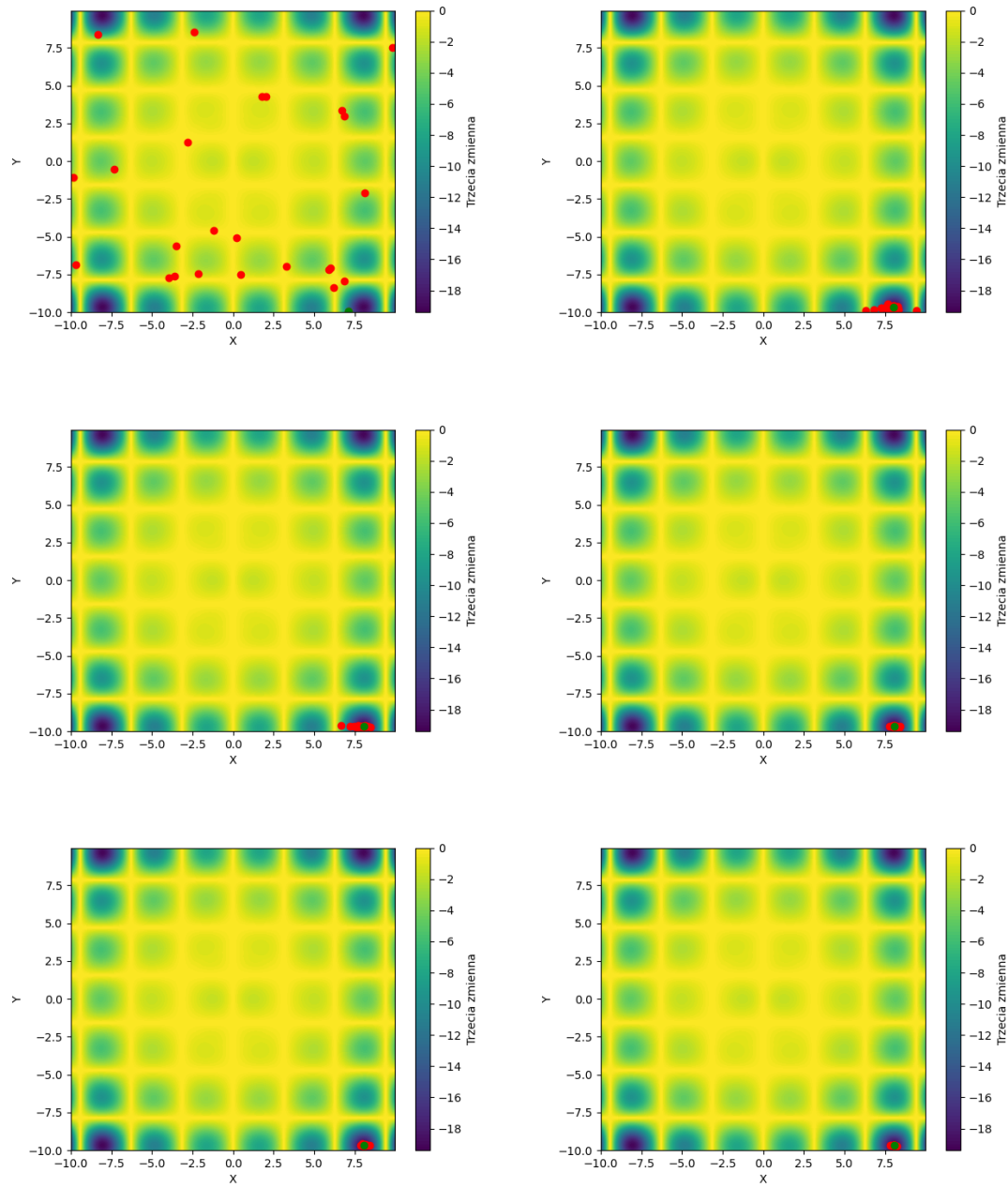


Dziedzina przeszukiwań = $[-0, \pi]$. Optymalna wartość = -1.8013, dla punktu (2.20, 1.57).

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
-1.801303	-1.801294	-1.0	$5,81 \cdot 10^{-7}$	1
-1.801303	-1.801303	-1,0	$6,64 \cdot 10^{-13}$	2
-1.801303	-1.801303	-1.0	$1,52 \cdot 10^{-13}$	3
-1.801303	-1.801303	-1.0	$5,16 \cdot 10^{-12}$	4

Tab. 3.5: Porównanie wyników dla funkcji Michalewicz w różnych trybach.

3.6. Holder Table



Dziedzina przeszukiwań = $[-10, 10]$. Optymalna wartość = -19.2085, dla punktów: (8.05502, 9.66459), (-8.05502, 9.66459), (-8.05502, -9.66459).

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
-19.2085	-18.8182	-15.14	$5,38 \cdot 10^{-1}$	1
-19.2085	-19.2085	-11.0695	$1,09 \cdot 10^{-1}$	2
-19.2085	-18.9600	-16.27	$4,81 \cdot 10^{-1}$	3
-19.2085	-18.8305	-15.14	$5,49 \cdot 10^{-1}$	4

Tab. 3.6: Porównanie wyników dla funkcji Holder Table w różnych trybach.

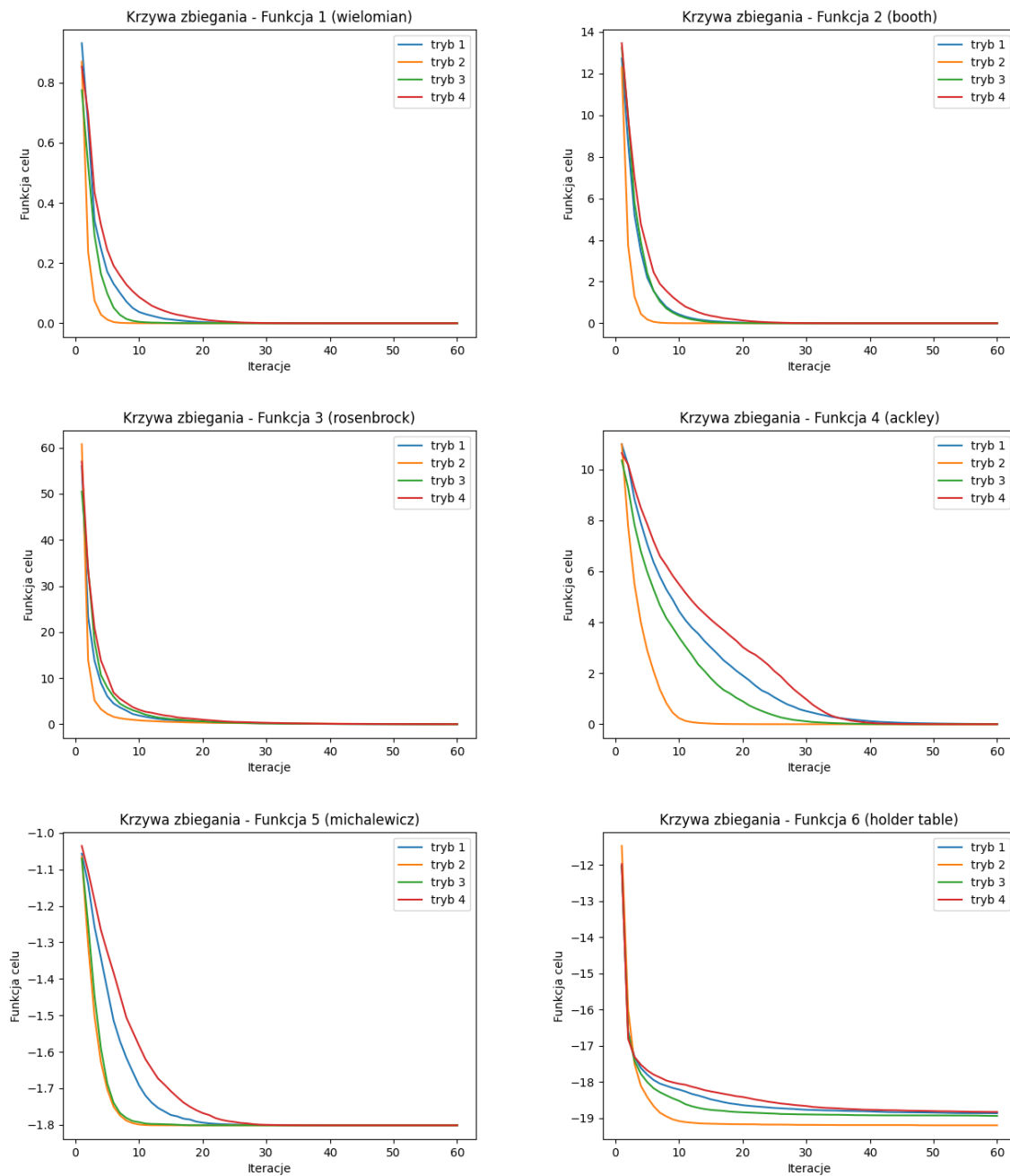
Analiza powyższych wyników pozwala stwierdzić, że najbliższą oczekiwanej wartości uzyskał zazwyczaj algorytm działający w trybie 3. Jednak najbliższy oczekiwanej wartości średni

wynik miał algorytm działający w trybie 2. Oznacza to, że tryb 3 pozwala na większą eksploatację przestrzeni przeszukiwań, jednak to trybowi 2 udało się częściej znaleźć optimum.

4. Analiza wyników

4.1. Krzywe zbieżności

Poniżej graficzne przedstawienie zmian wartości funkcji celu w kolejnych iteracjach, ilustrujące proces dążenia do minimum optymalnej wartości przez algorytm.

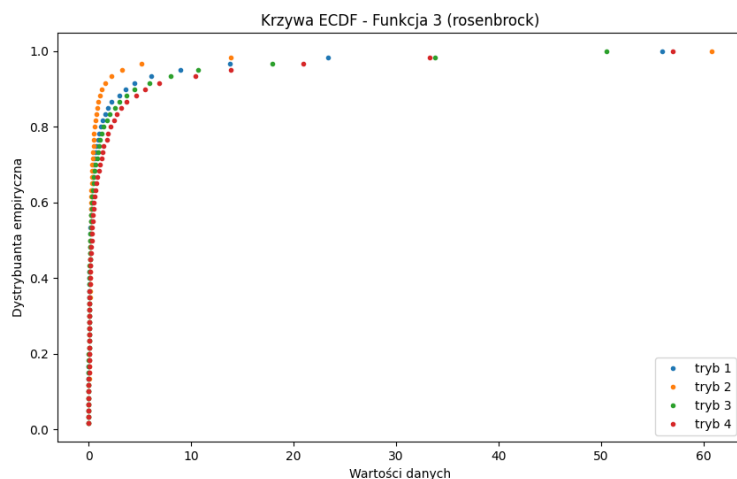
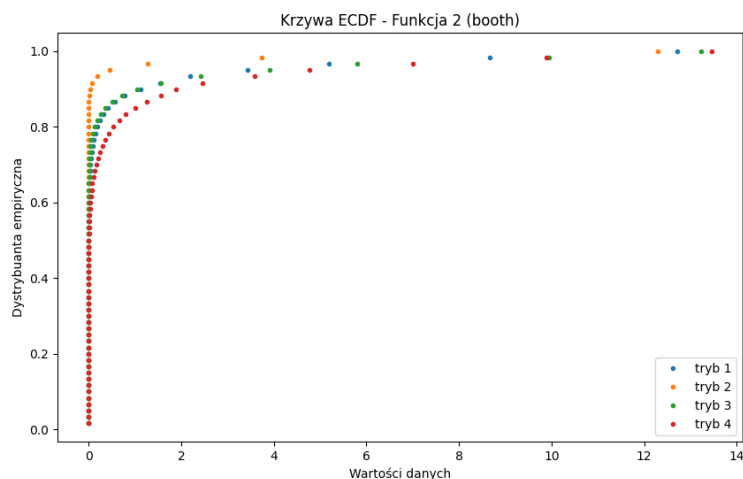
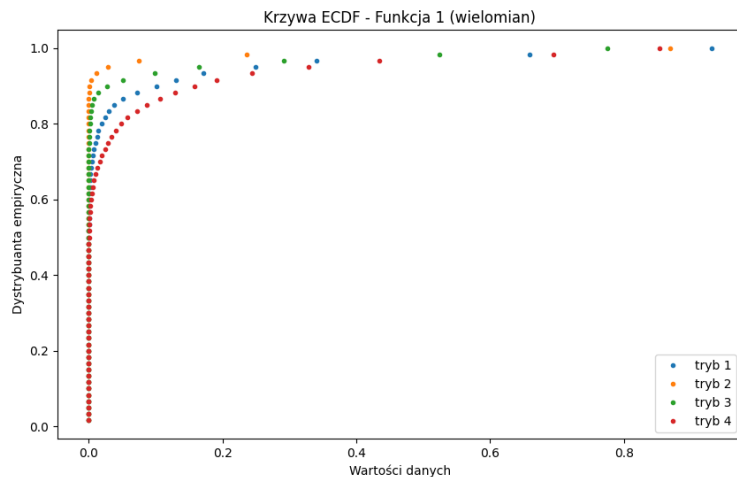


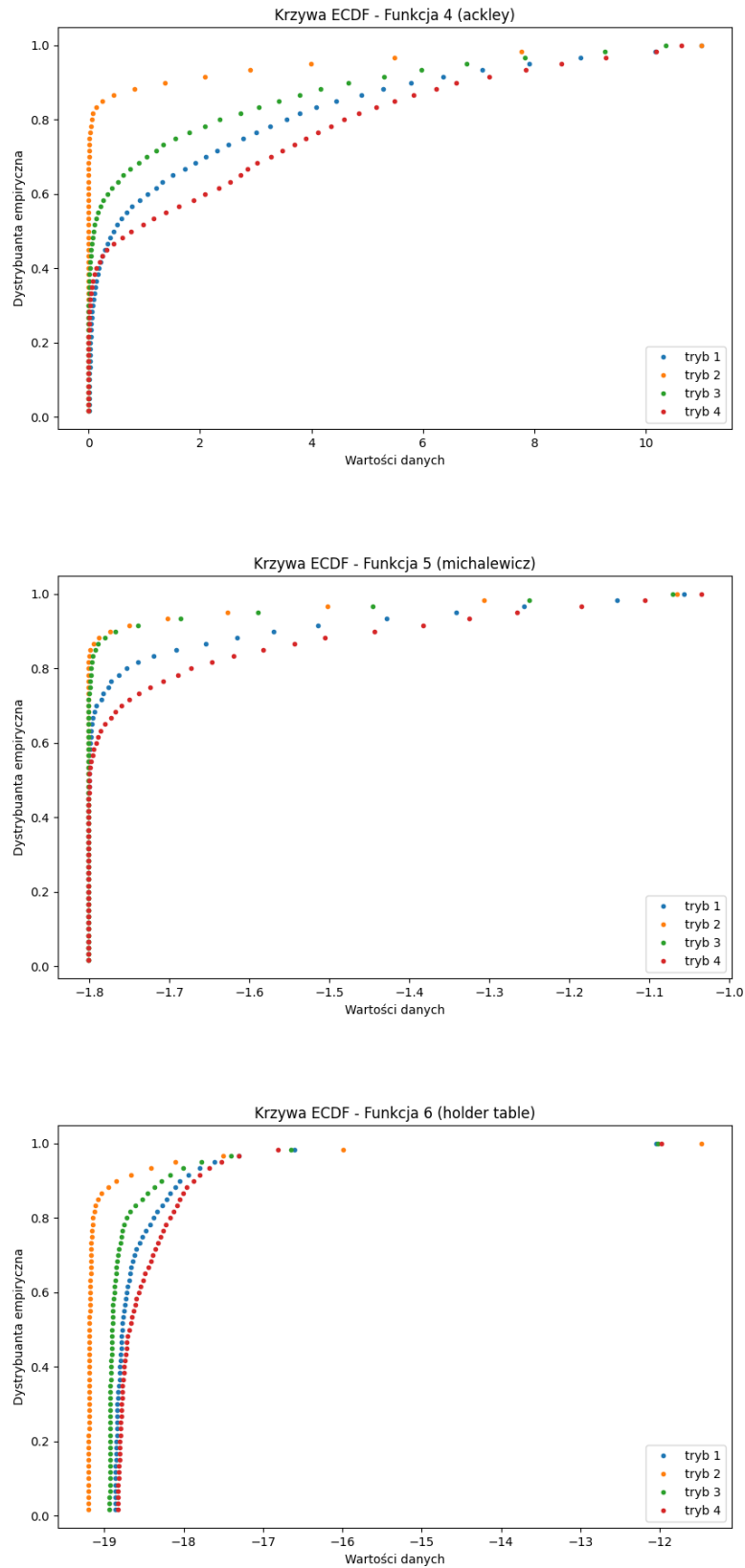
Rys. 4.1: Krzywe zbieżności

Powyższe wykresy pozwalają nam z łatwością zauważyć, że najszybciej funkcja celu docierała do optimum w algorytmie z trybem 2. W przypadku funkcji nr 6 (Holder Table) widzimy ciekawe zjawisko, ponieważ tylko algorytm z trybem nr 2 skutecznie odnajdywał optimum. Pozostałe tryby miały problem by zawsze odnaleźć minimum funkcji. Prawdopodobnie spowodowane jest to sposobem obsługi ograniczeń jakim jest rzutowanie. Ponieważ Holder Table to jedyna funkcja, która posiada swoje optimum w pobliżu granicy swojej dziedziny, rzutowanie może zaburzać proces eksploatacji przestrzeni wokół minimum.

4.2. Krzywe ECDF

Krzywa empirycznej dystrybuanty skumulowanej (ECDF) to graficzna reprezentacja empirycznego rozkładu danych, przedstawiająca procentowy udział wartości w próbie, które są mniejsze lub równe danej wartości.

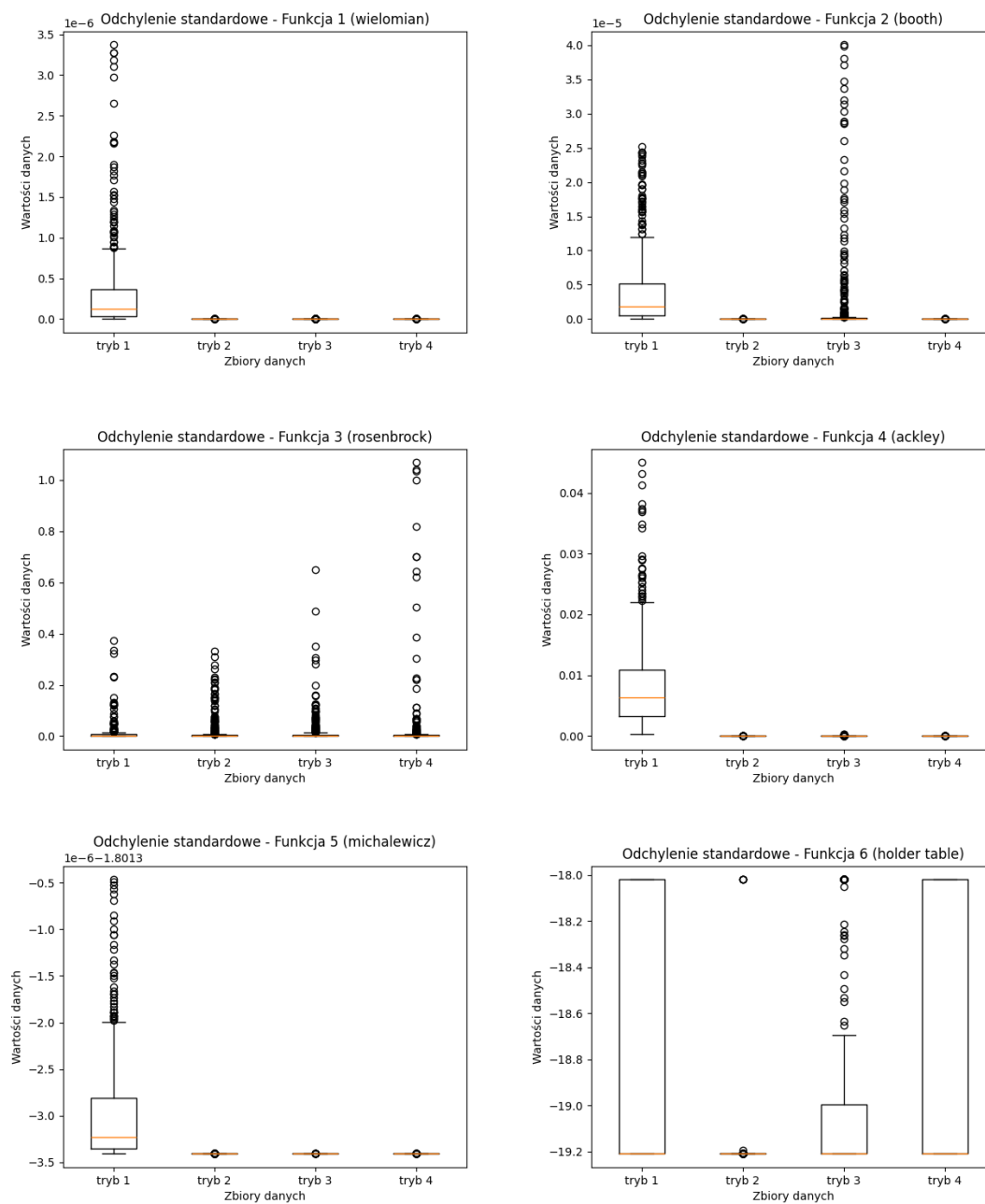




Rys. 4.3: Krzywe ECDF

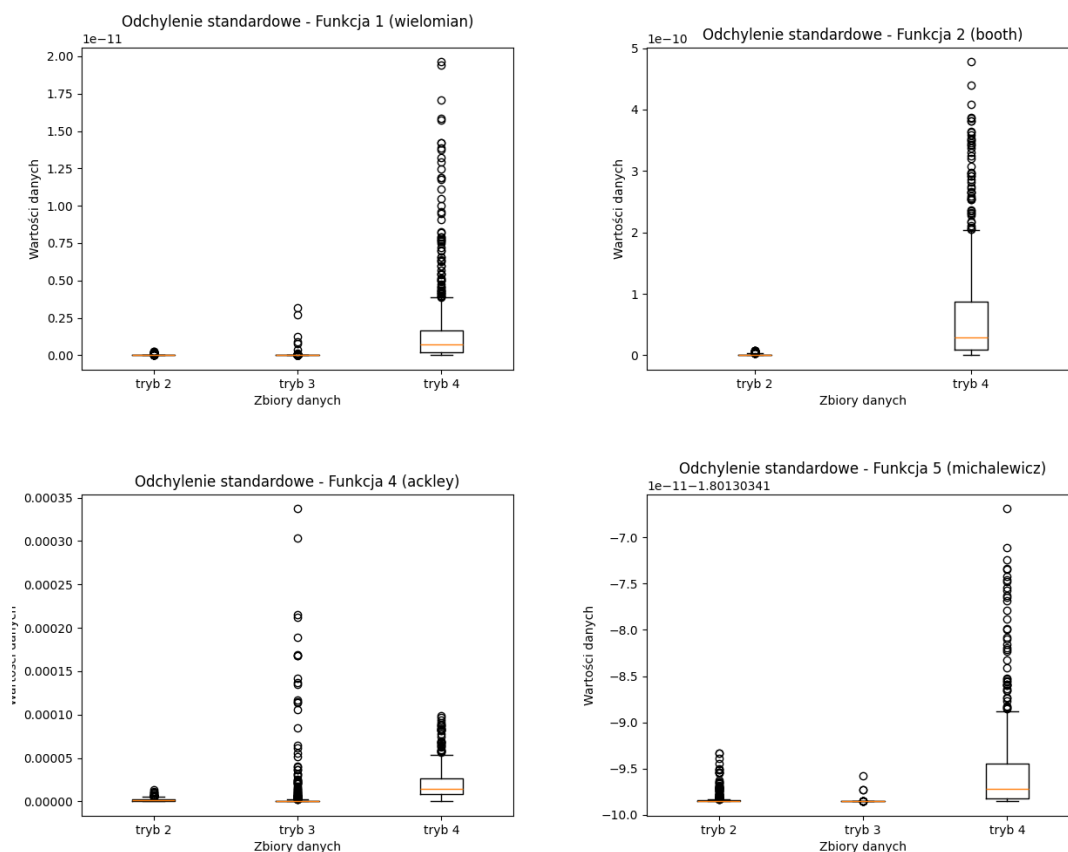
4.3. Wykresy pudełkowe

Wykresy pudełkowe graficznie przedstawiają rozkład danych, prezentując kwartyle, medianę i zakres interkwartylowy, co umożliwia analizę skupienia oraz rozproszenia wartości w zbiorze.



Rys. 4.4: Wykresy pudełkowe

Wykresy pudełkowe dla funkcji, dla których z powyższego wykresu nie da się wybrać najlepszego trybu.



Rys. 4.5: Wykresy pudełkowe

Na powyższych wykresach pudełkowych ponownie widzimy, że najlepiej radzi sobie algorytm w trybie 2, ponieważ zwykle posiada najmniej wyników odbiegających znacząco od optimum.

5. Wyższe wymiary

Dla lepszej oceny jakości modyfikacji algorytmu PSO przeprowadziliśmy również analizę dla funkcji o wyższych wymiarach. Przetestowane zostało 5 oraz 10 wymiarowe funkcje: wielomianowa, Rosenbrocka, Ackley’a i Michalewicz’a.

5.1. 5 wymiarów

Przeprowadzono 300 testów z maksymalną liczbą iteracji równą 100.

Funkcja wielomianowa

Dziedzina przeszukiwań $x_i \in [-5, 5]$, dla $i = 1, \dots, 5$.
Optimum = 0, dla punktu $x = (0, \dots, 0)$.

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$5,46 \cdot 10^{-9}$	$4,57 \cdot 10^{-1}$	$1,62 \cdot 10^1$	1,5651	1
$2,91 \cdot 10^{-17}$	2,4692	$2,45 \cdot 10^1$	4,3228	2
$1,57 \cdot 10^{-8}$	$7,91 \cdot 10^{-1}$	$1,78 \cdot 10^1$	1,9681	3
$1,99 \cdot 10^{-16}$	$6,23 \cdot 10^{-1}$	$2,01 \cdot 10^1$	1,7849	4

Tab. 5.1: Porównanie wyników dla funkcji wielomianowej w 5 wymiarach w różnych trybach.

Funkcja Rosenbrock

Dziedzina przeszukiwań $x_i \in [-5, 10]$, dla $i = 1, \dots, 5$.
Optimum = 0, dla punktu $x = (1, \dots, 1)$.

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$1,57 \cdot 10^{-4}$	$8,40 \cdot 10^2$	$3,87 \cdot 10^4$	$1,79 \cdot 10^3$	1
$9,78 \cdot 10^{-4}$	$3,46 \cdot 10^3$	$6,10 \cdot 10^4$	$6,18 \cdot 10^3$	2
$9,45 \cdot 10^{-2}$	$5,28 \cdot 10^2$	$4,46 \cdot 10^4$	$1,25 \cdot 10^3$	3
$6,62 \cdot 10^{-2}$	$8,96 \cdot 10^2$	$5,63 \cdot 10^4$	$2,09 \cdot 10^3$	4

Tab. 5.2: Porównanie wyników dla funkcji Rosenbrock w 5 wymiarach w różnych trybach.

Funkcja Ackley’a

Dziedzina przeszukiwań $x_i \in [-32.768, 32.768]$, dla $i = 1, \dots, 5$.
Optimum = 0, dla punktu $x = (0, \dots, 0)$.

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$1,51 \cdot 10^{-3}$	1,6962	$1,98 \cdot 10^1$	4,8387	1
$7,31 \cdot 10^{-8}$	3,4348	$2,00 \cdot 10^1$	5,9962	2
$9,48 \cdot 10^{-5}$	2,2039	$1,97 \cdot 10^1$	5,2112	3
$2,34 \cdot 10^{-7}$	1,5420	$2,00 \cdot 10^1$	4,5194	4

Tab. 5.3: Porównanie wyników dla funkcji Ackley w 5 wymiarach w różnych trybach.

Funkcja Michalewicz'a

Dziedzina przeszukiwań $x_i \in [0, \pi]$, dla $i = 1, \dots, 5$.
Optimum = -4,687658

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
-4,687 651	-4,2958	-2,8558	$3,69 \cdot 10^{-1}$	1
-4,687 658	-4,2951	-2,9479	$3,52 \cdot 10^{-1}$	2
-4,687 658	-4,2465	-2,5714	$3,74 \cdot 10^{-1}$	3
-4,687 658	-4,1932	-2,6571	$3,99 \cdot 10^{-1}$	4

Tab. 5.4: Porównanie wyników dla funkcji Michalewicz w 5 wymiarach w różnych trybach.

5.2. 10 wymiarów

Przeprowadzono 300 testów z maksymalną liczbą iteracji równą 200.

Funkcja wielomianowa

Dziedzina przeszukiwań $x_i \in [-5, 5]$, dla $i = 1, \dots, 5$.
Optimum = 0, dla punktu $x = (0, \dots, 0)$.

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$3,94 \cdot 10^{-10}$	$1,84 \cdot 10^1$	$6,55 \cdot 10^1$	$1,66 \cdot 10^1$	1
$7,97 \cdot 10^{-5}$	$2,63 \cdot 10^1$	$5,81 \cdot 10^1$	$1,73 \cdot 10^1$	2
$1,60 \cdot 10^{-3}$	$1,57 \cdot 10^1$	$5,75 \cdot 10^1$	$1,55 \cdot 10^1$	3
$3,25 \cdot 10^{-13}$	$1,75 \cdot 10^1$	$5,62 \cdot 10^1$	$1,55 \cdot 10^1$	4

Tab. 5.5: Porównanie wyników dla funkcji wielomianowej w 5 wymiarach w różnych trybach.

Funkcja Rosenbrock

Dziedzina przeszukiwań $x_i \in [-5, 10]$, dla $i = 1, \dots, 5$.
Optimum = 0, dla punktu $x = (1, \dots, 1)$.

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$3,56 \cdot 10^{-1}$	$6,17 \cdot 10^4$	$4,27 \cdot 10^5$	$6,99 \cdot 10^4$	1
4,5778	$8,78 \cdot 10^4$	$5,14 \cdot 10^5$	$7,80 \cdot 10^4$	2
3,3961	$4,63 \cdot 10^4$	$3,52 \cdot 10^5$	$5,52 \cdot 10^4$	3
$4,08 \cdot 10^{-1}$	$5,80 \cdot 10^4$	$5,46 \cdot 10^5$	$5,86 \cdot 10^4$	4

Tab. 5.6: Porównanie wyników dla funkcji Rosenbrock w 5 wymiarach w różnych trybach.

Funkcja Ackley'a

Dziedzina przeszukiwań $x_i \in [-32.768, 32.768]$, dla $i = 1, \dots, 5$.
Optimum = 0, dla punktu $x = (0, \dots, 0)$.

Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
$1,79 \cdot 10^{-4}$	5,9385	$2,06 \cdot 10^1$	8,8042	1
1,6462	$1,17 \cdot 10^1$	$2,08 \cdot 10^1$	6,1319	2
$1,09 \cdot 10^{-1}$	9,4832	$2,07 \cdot 10^1$	7,9143	3
$1,86 \cdot 10^{-6}$	7,0258	$2,05 \cdot 10^1$	8,7486	4

Tab. 5.7: Porównanie wyników dla funkcji Ackley w 5 wymiarach w różnych trybach.

Funkcja Michalewicz'a

Dziedzina przeszukiwań $x_i \in [0, \pi]$, dla $i = 1, \dots, 5$.
Optimum = -9,66015.

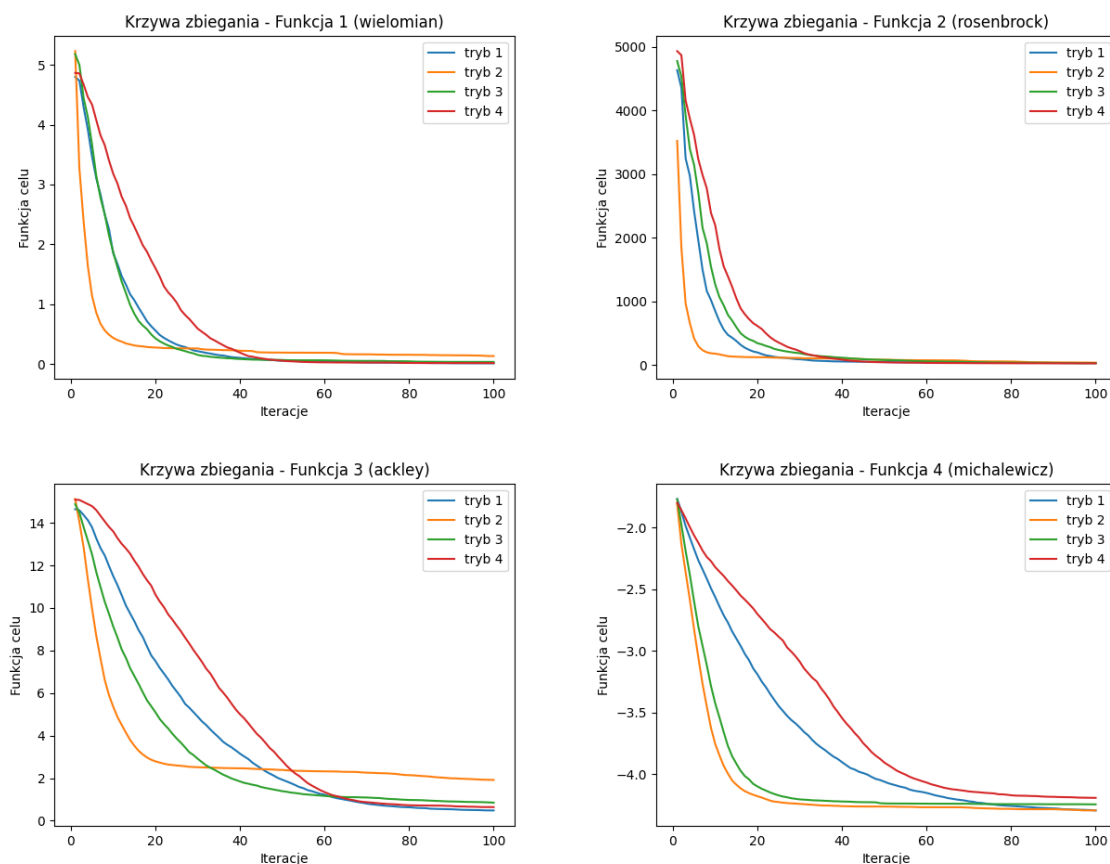
Najlepszy	Średni	Najgorszy	Odchylenie standardowe	Tryb
-9,386 64	-7,5271	-5,2036	$7,69 \cdot 10^{-1}$	1
-9,369 41	-7,4807	-5,2040	$7,82 \cdot 10^{-1}$	2
-9,420 70	-7,3826	-5,2202	$7,68 \cdot 10^{-1}$	3
-9,269 42	-7,0662	-4,8317	$7,98 \cdot 10^{-1}$	4

Tab. 5.8: Porównanie wyników dla funkcji Michalewicz w 5 wymiarach w różnych trybach.

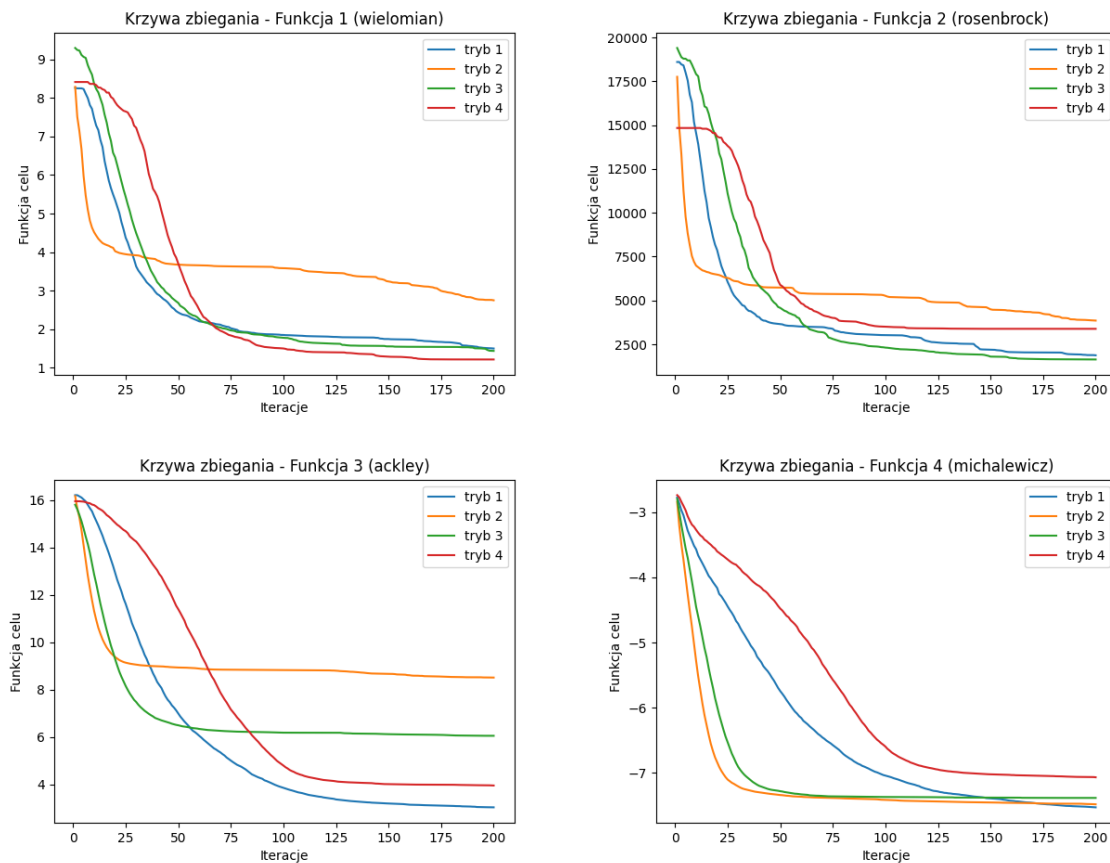
Widzimy, że dla większej ilości wymiarów algorytm radzi sobie znacznie gorzej niż dla dwuwymiarowych funkcji. Nawet najlepsze wyniki dla różnych trybów odbiegają potrafią znacząco od wartości minimalnej funkcji. Żadna z modyfikacji nie wysuwa się na prowadzenie biorąc pod uwagę dokładność wyniku optymalizacji oraz średnie wyników.

5.3. Analiza wyników

5.3.1. Krzywe zbieżności



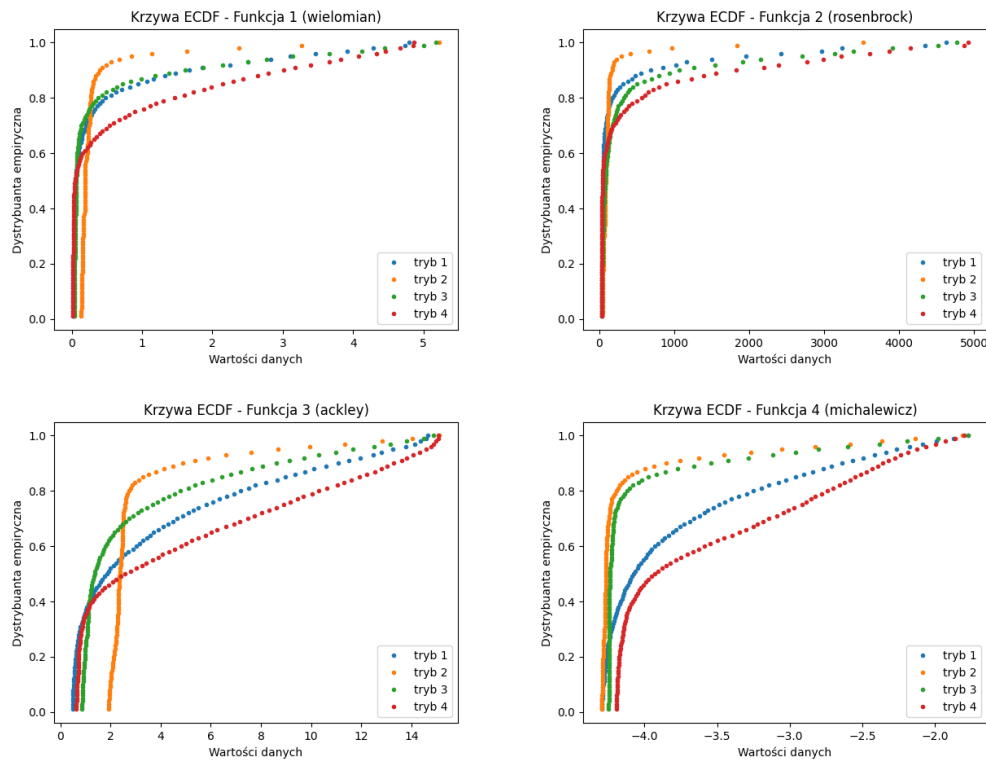
Rys. 5.1: Krzywe zbieżności - 5 wymiarowe



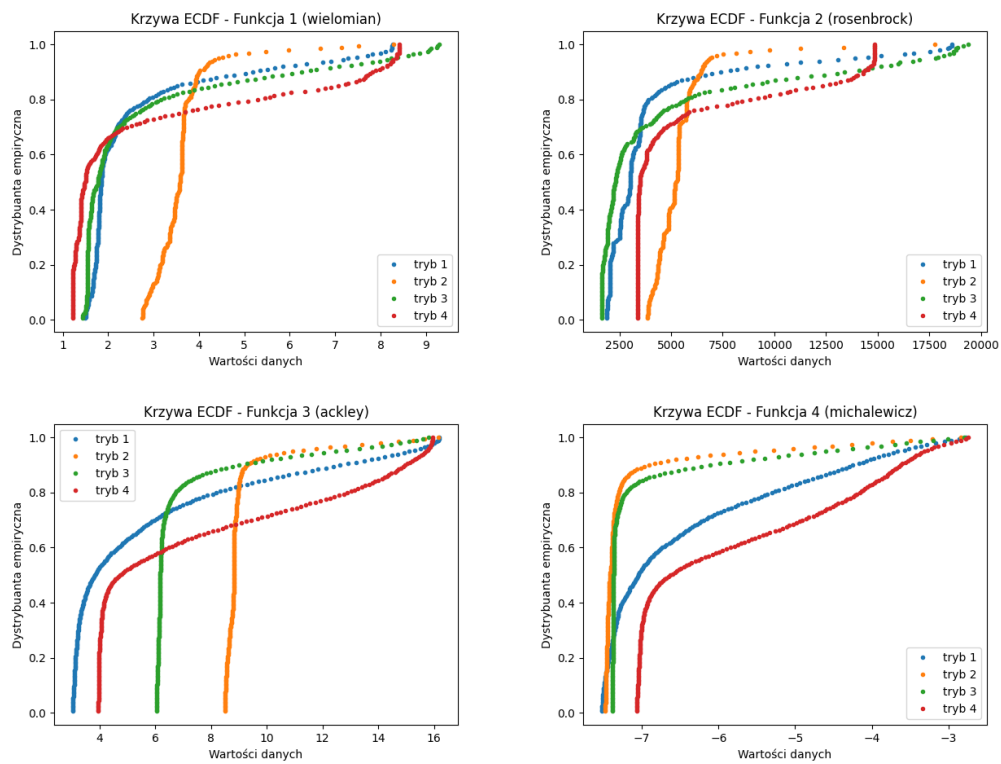
Rys. 5.2: Krzywe zbieżności - 10 wymiarowe

Powyższe wyniki pokazują, że dla wyższych wymiarów dobór modyfikacji nie jest już tak oczywisty jak dla dwuwymiarowych funkcji. W większości przypadków tryb nr 2 okazuje się być początkowo najszybszym algorytmem, ale zwykle nie zbliżającym się w dalszych iteracjach do optimum, zostając daleko w tyle za pozostałymi modyfikacjami. Z tego powodu z powyższych krzywych zbieżności ciężko jest wskazać najlepszą modyfikację.

5.3.2. Krzywe ECDF

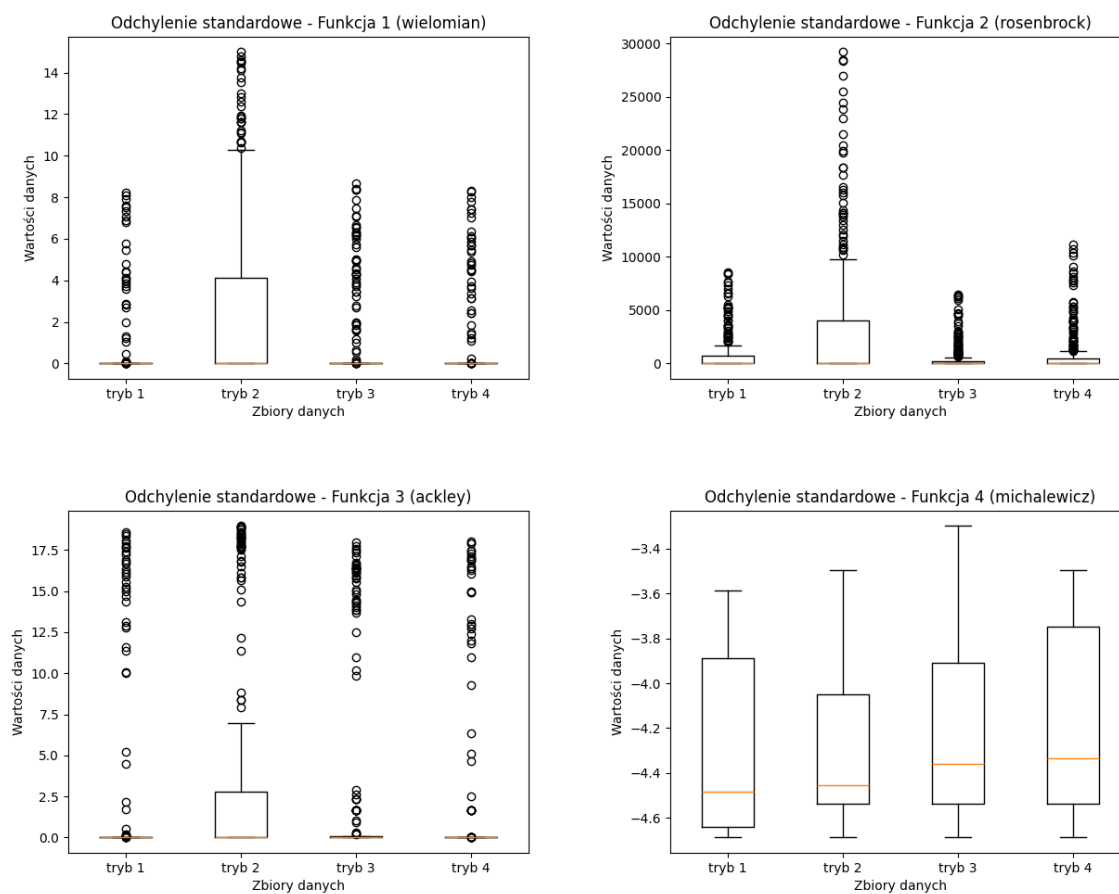


Rys. 5.3: Krzywe ECDF - 5 wymiarowe

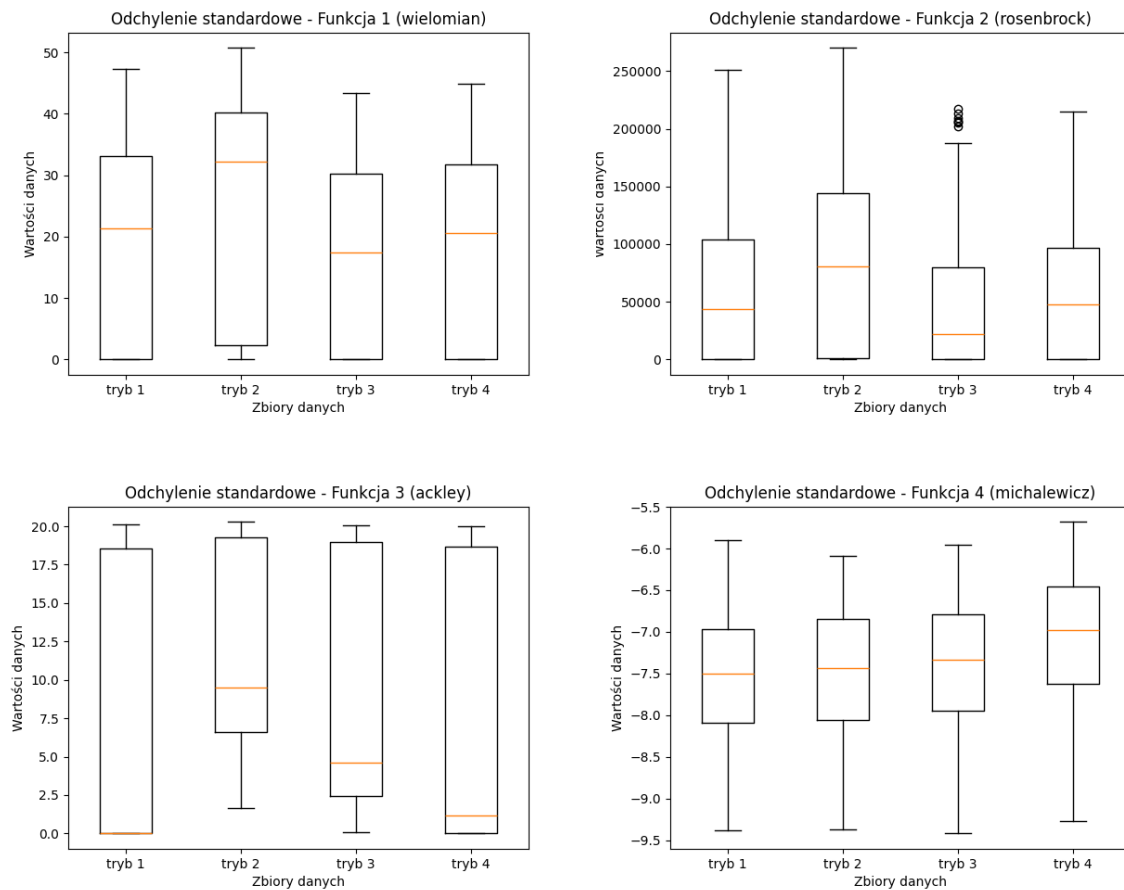


Rys. 5.4: Krzywe ECDF - 10 wymiarowe

5.3.3. Wykresy pudełkowe



Rys. 5.5: Wykresy pudełkowe - 5 wymiarowe



Rys. 5.6: Wykresy pudełkowe - 10 wymiarowe

Powyżej widzimy, jak zwiększenie wymiarów spowodowało trudności ze znalezieniem optimum pomimo znacznego zwiększenia czasu iteracji. Wykresy pudełkowe również nie wskazują, by któraś z modyfikacji dawała lepsze rezultaty niż pozostałe.

6. Podsumowanie

6.1. Porównania tabelaryczne

Poniżej oceniany jest każdy tryb w różnych kategoriach w skali od 1 - najlepszy, do 4 - najgorszy

6.1.1. Najlepszy wynik

	2 wymiary						5 wymiarów				10 wymiarów				średnia ranga
	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F1	F2	F3	F4	
tryb 1	3	4	4	4	2.5	2.5	3	1	4	4	2	1	2	2	2.786
tryb 2	2	2	2	2	2.5	2.5	1	2	1	2	3	4	4	3	2.357
tryb 3	1	1	3	1	2.5	2.5	4	4	3	2	4	3	3	1	2.5
tryb 4	4	3	1	3	2.5	2.5	2	3	2	2	1	2	1	4	2.357

6.1.2. Najlepsza średnia

	2 wymiary						5 wymiarów				10 wymiarów				średnia ranga
	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F1	F2	F3	F4	
tryb 1	4	4	1	4	4	4	1	3	2	1	3	3	1	1	2,571
tryb 2	2	1	3	1	2	1	4	2	4	2	4	4	4	2	2,571
tryb 3	1	3	2	2	2	2	3	1	3	3	1	1	3	3	2,143
tryb 4	3	2	4	3	2	3	2	4	1	4	2	2	2	4	2,714

6.1.3. Krzywe zbieżności

	2 wymiary						5 wymiarów				10 wymiarów				średnia ranga
	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F1	F2	F3	F4	
tryb 1	3	2.5	2	3	3	3	2.5	2	2.5	2.5	2.5	2.5	2.5	2	2.536
tryb 2	1	1	1	1	1.5	1	2.5	1	2.5	1	2.5	2.5	2.5	2	1.643
tryb 3	2	2.5	3	2	1.5	2	2.5	3	2.5	2.5	2.5	2.5	2.5	2	2.357
tryb 4	4	4	4	4	4	4	2.5	4	2.5	4	2.5	2.5	2.5	4	3.464

6.1.4. Wykresy pudełkowe

	2 wymiary						5 wymiarów				10 wymiarów				średnia ranga
	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F1	F2	F3	F4	
tryb 1	4	4	2	4	4	3.5	2	3	1.5	1	3	2.5	1	1.5	2,643
tryb 2	1	1	1	1	2	1	4	4	4	2	4	4	4	1.5	2,464
tryb 3	2	3	3	2	1	2	2	1	3	4	1	1	3	3	2,214
tryb 4	3	2	4	3	3	3.5	2	2	1.5	3	2	2.5	2	4	2.679

6.1.5. Sumacyjne wyniki

modyfikacja	suma z średnich rang	ostateczna ranga
tryb 1	2.63	3
tryb 2	2.26	1
tryb 3	2.30	2
tryb 4	2.80	4

6.2. Podsumowanie

Wyniki przeprowadzonych przez nas testów wskazują dwóch faworytów. Są to modyfikacja druga - współczynnik zwiększający swoją wartość z czasem trwania programu oraz modyfikacja trzecia, czyli współczynnik zależny od jakości rozwiązania. Druga modyfikacja była zdecydowanie najszybsza w większości testów, zazwyczaj najszybciej znajdowała rozwiązanie w sąsiedztwie optimum, co szczególnie widać na krzywych zbieżności. Modyfikacja trzecia za to najczęściej zwracała wynik w najmniejszym otoczeniu założonego optimum - rezultaty tej modyfikacji były najdokładniejsze. Działo się to jednak kosztem liczby iteracji potrzebnych do znalezienia tych wartości oraz potencjalnej większej liczby przeprowadzonych podejść.

Analiza funkcji dwuwymiarowych niezaprzeczalnie wskazywała na wyższość modyfikacji numer 2 i 3 jednak dla wyższych wymiarów okazało się, że sposób modyfikacji a nawet jego brak nie wpływa jednoznacznie na jakość wyników optymalizacji. co prawda dalej algorytm z modyfikacją nr 2 najszybciej zmniejsza swoją funkcję celu jednak do pewnego momentu oraz kosztem ostatecznego wyniku, który często znacząco odbiega od oczekiwanego optimum oraz wyniku innych, wolniejszych modyfikacji.

Dlatego w kwestii ostatecznego zwycięzcy warto rozważyć na czym nam zależy bardziej, ilu wymiarową funkcję zamierzamy optymalizować oraz czy mamy ograniczony czas wykonywania programu. Jeżeli chodzi o szybkość działania oraz przeciętnie otrzymany wynik to modyfikacja druga okaże się najszybsza. W przypadku, gdy możemy pozwolić sobie na więcej testów, które potencjalnie mogą zająć więcej czasu, a zależy nam najbardziej na dokładności rozwiązania, to modyfikacja trzecia okaże się odpowiednią opcją.

7. Bibliografia

1. https://en.wikipedia.org/wiki/Particle_swarm_optimization
2. <https://www.sfu.ca/~ssurjano/>
3. <https://pure.port.ac.uk/ws/files/83267/geccoPP1.pdf>
4. M. Clerc and J. Kennedy. The particle swarm - explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002