

Bartosz Jurczewski 234067 234067@edu.p.lodz.pl
Karol Podlewski 234106 234106@edu.p.lodz.pl

Zadanie 2: Optymalizacja wielokryterialna funkcji

1. Cel

Celem zadania była optymalizacja wielokryterialna funkcji wielu zmiennych na zadanym przedziale. Należało wykorzystać algorytmy do rozwiązywania problemów, dla których znany jest front Pareto.

2. Opis implementacji

Zadanie zostało zrealizowane w języku Python, w wersji 3.7.5, z wykorzystaniem bibliotek: Matplotlib, NumPy, Platypus-Opt (algorytmy genetyczne) oraz Pymoo (ewaluacja).

3. Wprowadzenie

3.1. Algorytmy genetyczne

W przygotowanym rozwiązaniu porównane ze sobą zostały trzy algorytmy przeznaczone do optymalizacji wielokryterialnej funkcji: NSGA-II, SPEA-II oraz PESA-II.

— **NSGA-II** - jest to obecnie używana, poprawiona wersja algorytmu NSGA (ang. *Non-dominated Sorting Genetic Algorithm*). Kroki algorytmu:

1. Losowe utworzenie populacji $P(0)$. $t = 0$.
2. Przez mutację i krzyżowanie na zbiorze $P(t)$ zostaje utworzony zbiór $Q(t)$.
3. Zbiory $P(t)$ oraz $Q(t)$ zostają połączone w jeden zbiór $R(t)$. Dla każdego rozwiązania ze zbioru $R(t)$ zostaje wyliczona siła w zależności od poziomu w którym jest ona niezdominowana. Następnie wartości te są sortowane.
4. Zbiór $R(t)$ zostaje zmniejszony do wielkości populacji przez usunięcie najsłabszych wartości - algorytm stara się usunąć takie rozwiązania, wokół których znajduje się wiele innych najsłabszych rozwiązań.
5. Jeżeli osiągnięto wymaganą liczbę epok t , algorytm zostaje przerwany, zbiór $P(t + 1)$ jest jego wynikiem.
6. $t = t + 1$. Jeżeli nie, należy wrócić do punktu 2.

[1][4]

— **SPEA-II** - jest to najpopularniejsze rozszerzenie algorytmu SPEA (ang. *Strength Pareto Evolutionary Algorithm*). Kroki algorytmu:

1. Losowe utworzenie populacji $P(0)$ oraz utworzenie pustego zbioru rozwiązań niezdominowanych $A(0)$. $t = 0$.
2. Wyliczenie dopasowania dla każdego ze zbiorów ($P(t)$ oraz $A(t)$).
3. Skopiowanie wszystkich niezdominowanych rozwiązań z obu zbiorów do nowego zbioru rozwiązań niezdominowanych $A(t + 1)$. Jeżeli zbiór przekracza maksymalną wielkość ustaloną przed rozpoczęciem działania algorytmu, zostaje zmniejszony z wykorzystaniem algorytmu klastrowania. Jeżeli zbiór jest za mały, zostaje wypełniony rozwiązaniami zdominowanymi o najlepszym dopasowaniu.

4. Jeżeli osiągnięto wymaganą liczbę epok t , algorytm zostaje przerwany, zbiór $A(t+1)$ jest jego wynikiem.
5. Jeżeli nie, zbiór $A(t+1)$ zostaje poddany selekcji turniejowej w celu utworzenia zbioru o wielkości populacji, który, będąc poddanym mutacji oraz krzyżowaniu, tworzy zbiór $P(t+1)$.
6. $t = t + 1$. Należy przejść do kroku 2.

Algorytm SPEA-II względem pierwowzoru poprawił obliczanie dopasowania, metody zmniejszania oraz wypełniania zbioru $A(t+1)$ oraz zmienił zbiór, który jest poddany mutacji [1][3][5].

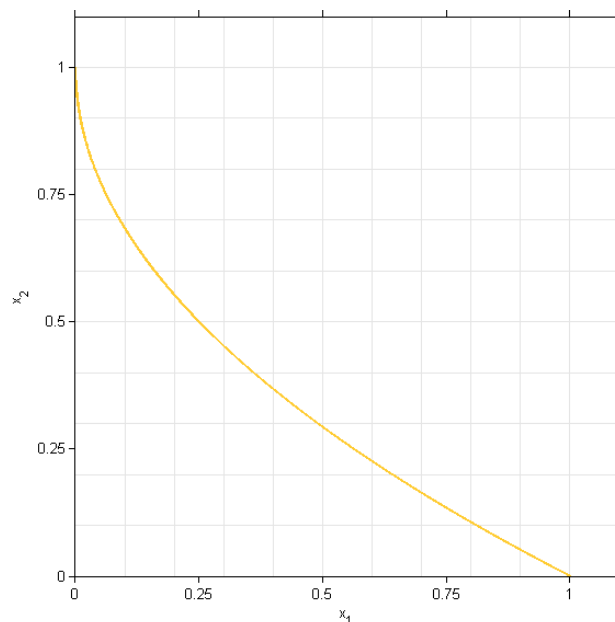
— **PESA-II** - jest to druga wersja algorytmu PESA (ang. *Pareto Envelope-based Selection Algorithm*). Kroki algorytmu:

1. Losowe utworzenie populacji $P(0)$ oraz utworzenie pustego zbioru rozwiązań niezdominowanych $A(0)$. $t = 0$.
2. Skopiowanie wszystkich niezdominowanych rozwiązań ze zbioru $P(t)$ do zbioru $A(t)$.
3. Jeżeli osiągnięto wymaganą liczbę epok t , algorytm zostaje przerwany. Zbiór $A(t)$ jest jego wynikiem.
4. Jeżeli nie, przez mutację i krzyżowanie na zbiorze $A(t)$, zostaje utworzony zbiór $P(t+1)$.
5. $t = t + 1$. Należy przejść do kroku 2.

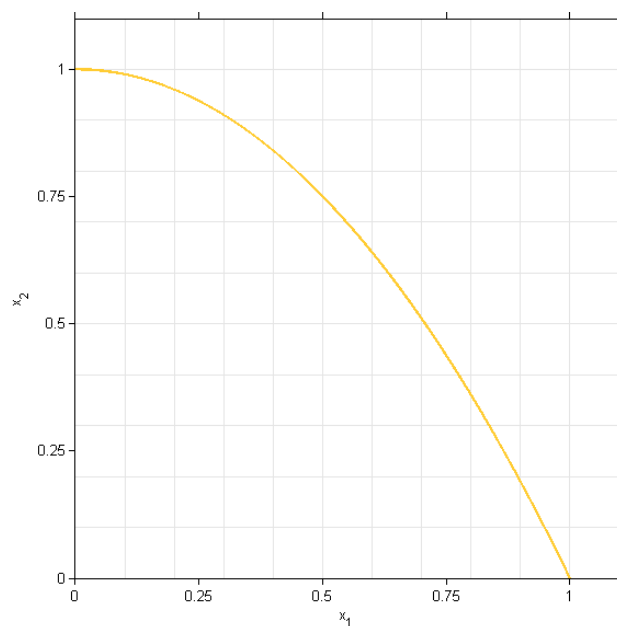
[2]

3.2. Problemy testowe

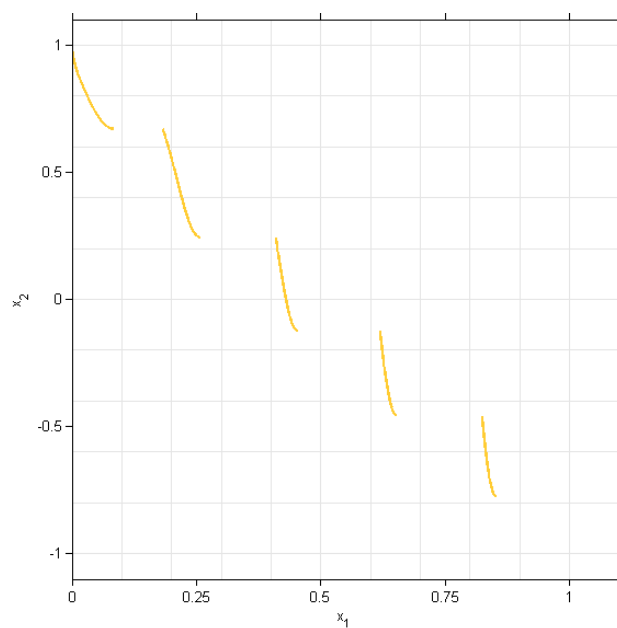
Przygotowane rozwiązanie obsługuje trzy problemy testowe: **ZDT1**, **ZDT2** i **ZDT3**. Dla każdego z nich rozwiązania są tym lepsze, im bliżej frontu Pareto leżą.



Rysunek 1. Problem ZDT1 <http://people.ee.ethz.ch/~sop/download/supplementary/testproblems/zdt1/index.php>



Rysunek 2. Problem ZDT2 <http://people.ee.ethz.ch/~sop/download/supplementary/testproblems/zdt2/index.php>



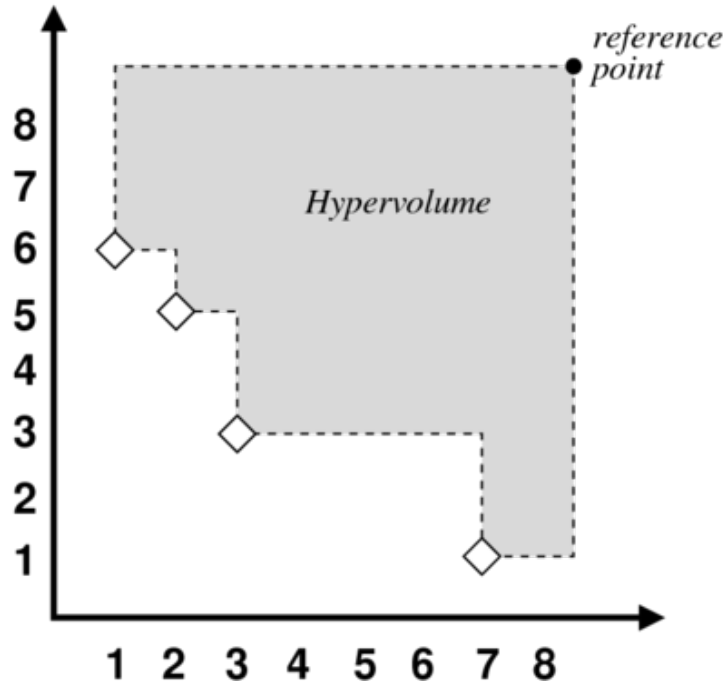
Rysunek 3. Problem ZDT3 <http://people.ee.ethz.ch/~sop/download/supplementary/testproblems/zdt3/index.php>

3.3. Miary oceny algorytmów

Przygotowane rozwiązanie ocenia działanie algorytmu przy użyciu dwóch miar:

1. **Generational Distance (GD)** - miara oblicza średnią odległość euklidesową każdego rozwiązania do najbliższego punktu na froncie Pareto. Im mniejszy wynik, tym algorytm znalazł lepsze rozwiązanie.

2. **Hypervolume (HV)** - miara oblicza pole powierzchni między niezdominowanymi rozwiązaniami, a wybranym punktem referencyjnym (w przypadku prezentowanego rozwiązania jest to punkt $x=1, y=5$). Im wyższy wynik, tym algorytm znalazł lepsze rozwiązanie.



Rysunek 4. Prezentacja miary Hypervolume dla 2 wymiarowej przestrzeni
<http://lopez-ibanez.eu/hypervolume>

4. Wyniki

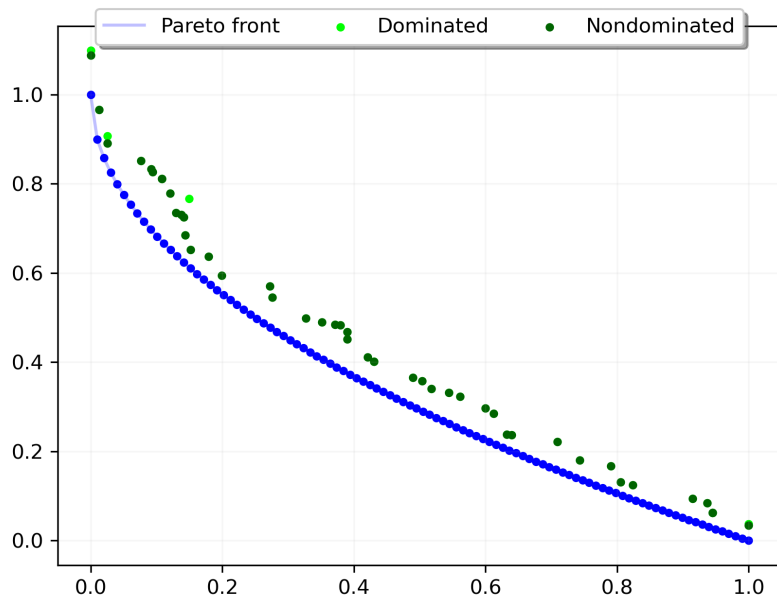
Dla każdego algorytmu została porównana wielkość populacji oraz prawdopodobieństw operatorów. W przypadku algorytmów SPEA-II oraz PESA-II został porównany także dodatkowy atrybut, charakterystyczny dla danego algorytmu, z uwzględnieniem najlepszych uzyskanych wyników dla wcześniejszych testów.

4.1. Algorytm NSGA-II

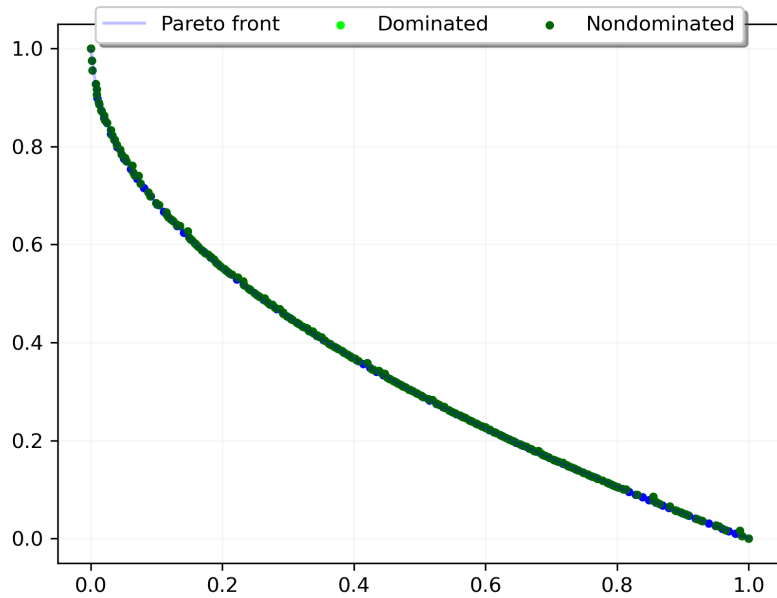
4.1.1. Problem testowy ZDT1

Tabela 1. Wyniki algorytmu NSGA-II dla problemu ZDT1

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	1,353	2,789
1000	50	0,5	1,344	2,939
1000	50	0,75	1,442	3,008
1000	100	0,25	2,033	2,337
1000	100	0,5	1,786	2,523
1000	100	0,75	1,66	2,669
1000	200	0,25	2,168	2,183
1000	200	0,5	2,191	2,098
1000	200	0,75	2,273	2,074
10000	50	0,25	0,05	4,589
10000	50	0,5	0,158	4,438
10000	50	0,75	0,228	4,359
10000	100	0,25	0,159	4,44
10000	100	0,5	0,235	4,338
10000	100	0,75	0,251	4,292
10000	200	0,25	0,511	3,96
10000	200	0,5	0,471	4,025
10000	200	0,75	0,601	3,921
100000	50	0,25	0,005	4,656
100000	50	0,5	0,028	4,622
100000	50	0,75	0,074	4,557
100000	100	0,25	0,005	4,661
100000	100	0,5	0,033	4,616
100000	100	0,75	0,087	4,538
100000	200	0,25	0,005	4,663
100000	200	0,5	0,042	4,604
100000	200	0,75	0,112	4,508



Rysunek 5. Algorytm NSGA-II, problem ZDT1,
generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25

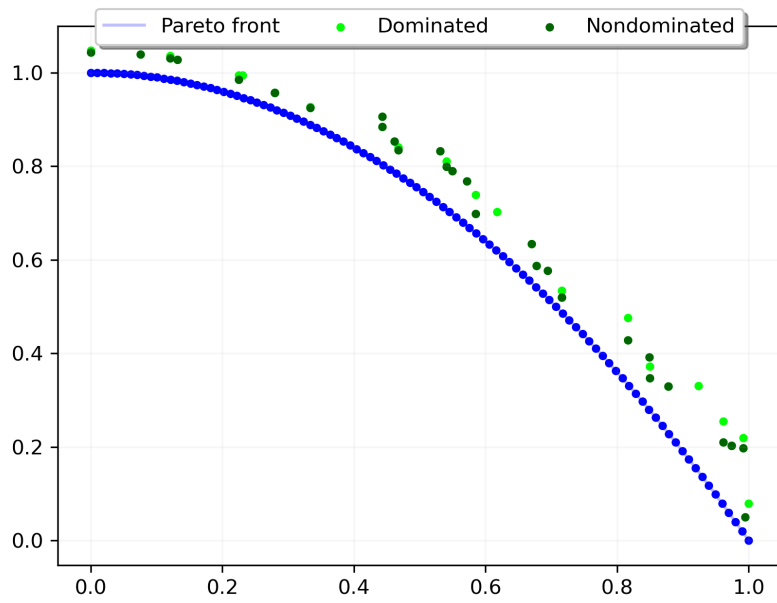


Rysunek 6. Algorytm NSGA-II, problem ZDT1,
generacje - 100 000, populacja - 200, prawdopodobieństwo - 0,25

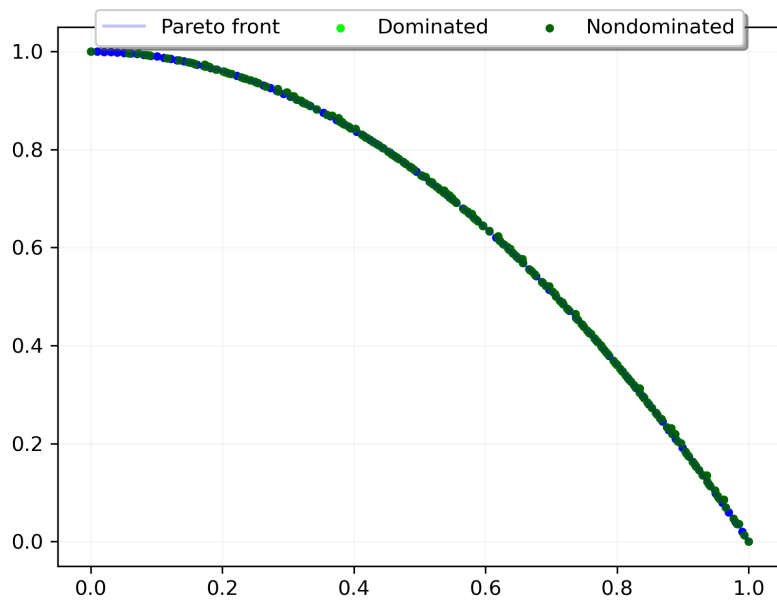
4.1.2. Problem testowy ZDT2

Tabela 2. Wyniki algorytmu NSGA-II dla problemu ZDT2

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	2,339	1,689
1000	50	0,5	2,01	2,065
1000	50	0,75	1,662	2,424
1000	100	0,25	2,813	1,194
1000	100	0,5	2,761	1,295
1000	100	0,75	2,07	1,923
1000	200	0,25	2,949	1,086
1000	200	0,5	3,001	1,022
1000	200	0,75	2,66	1,258
10000	50	0,25	0,047	4,246
10000	50	0,5	0,243	3,982
10000	50	0,75	0,314	3,885
10000	100	0,25	0,119	4,15
10000	100	0,5	0,356	3,844
10000	100	0,75	0,402	3,811
10000	200	0,25	0,321	3,679
10000	200	0,5	0,742	3,513
10000	200	0,75	0,576	3,555
100000	50	0,25	0,004	4,323
100000	50	0,5	0,032	4,282
100000	50	0,75	0,128	4,146
100000	100	0,25	0,004	4,328
100000	100	0,5	0,049	4,256
100000	100	0,75	0,144	4,132
100000	200	0,25	0,004	4,329
100000	200	0,5	0,063	4,233
100000	200	0,75	0,175	4,074



Rysunek 7. Algorytm NSGA-II, problem ZDT2,
generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25

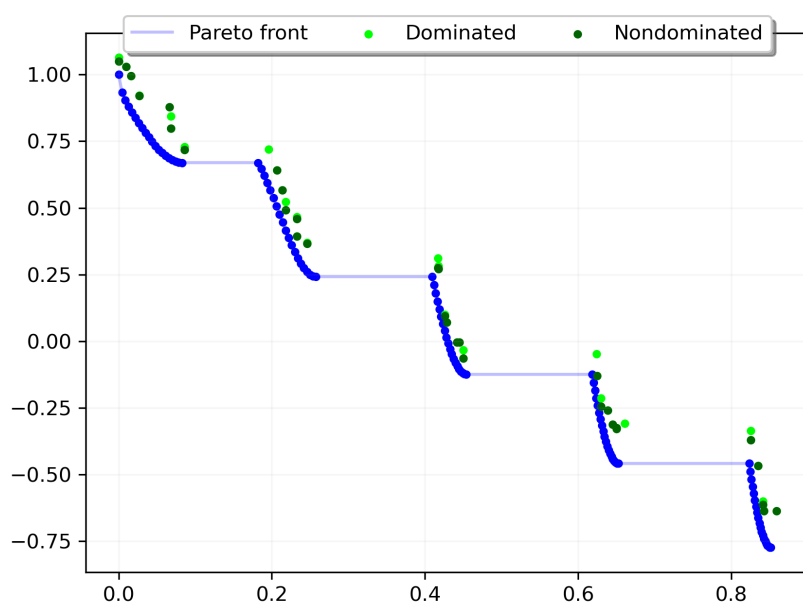


Rysunek 8. Algorytm NSGA-II, problem ZDT2,
generacje - 100 000, populacja - 200, prawdopodobieństwo - 0,25

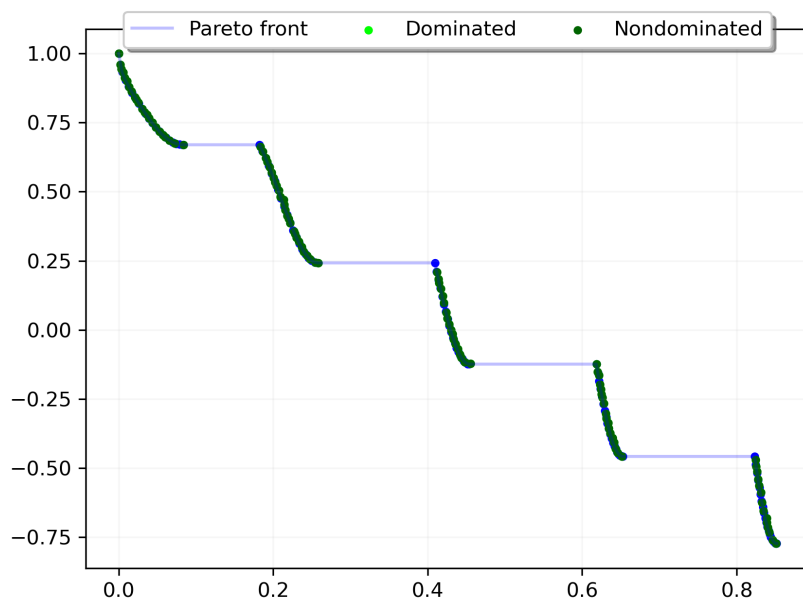
4.1.3. Problem testowy ZDT3

Tabela 3. Wyniki algorytmu NSGA-II dla problemu ZDT3

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	1,529	2,929
1000	50	0,5	1,507	2,947
1000	50	0,75	1,422	3,013
1000	100	0,25	1,695	2,508
1000	100	0,5	1,725	2,884
1000	100	0,75	1,724	2,716
1000	200	0,25	2,002	2,509
1000	200	0,5	1,875	2,5
1000	200	0,75	2,095	2,564
10000	50	0,25	0,022	4,937
10000	50	0,5	0,087	4,773
10000	50	0,75	0,144	4,682
10000	100	0,25	0,085	4,784
10000	100	0,5	0,148	4,683
10000	100	0,75	0,234	4,546
10000	200	0,25	0,518	4,198
10000	200	0,5	0,374	4,311
10000	200	0,75	0,492	4,178
100000	50	0,25	0,006	5,038
100000	50	0,5	0,013	4,997
100000	50	0,75	0,039	4,905
100000	100	0,25	0,006	5,04
100000	100	0,5	0,016	4,983
100000	100	0,75	0,059	4,881
100000	200	0,25	0,006	5,039
100000	200	0,5	0,021	4,966
100000	200	0,75	0,044	4,883



Rysunek 9. Algorytm NSGA-II, problem ZDT3, generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25



Rysunek 10. Algorytm NSGA-II, problem ZDT3, generacje - 100 000, populacja - 100, prawdopodobieństwo - 0,25

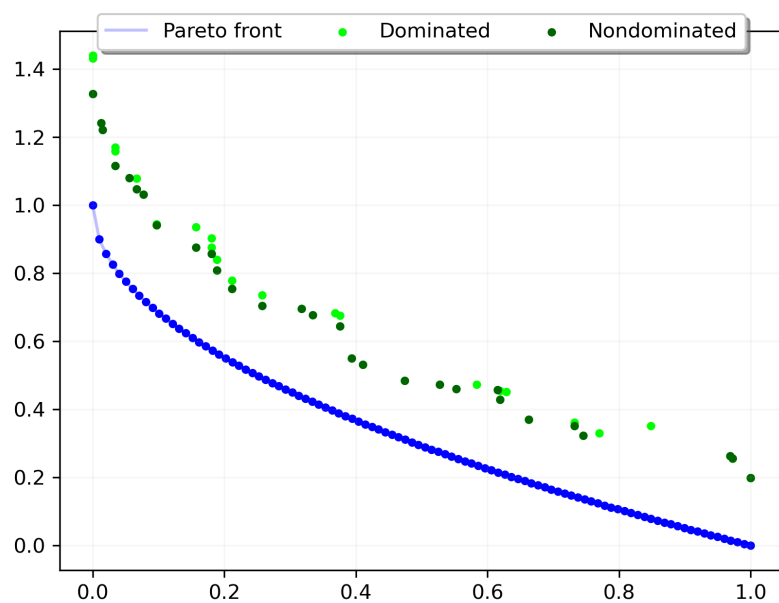
4.2. Algorytm SPEA-II

Ze względu na bardzo długi czas działania algorytmu dla 100 000 generacji oraz 200 epok, wyniki przy tej konfiguracji nie zostały uwzględnione w tabelach.

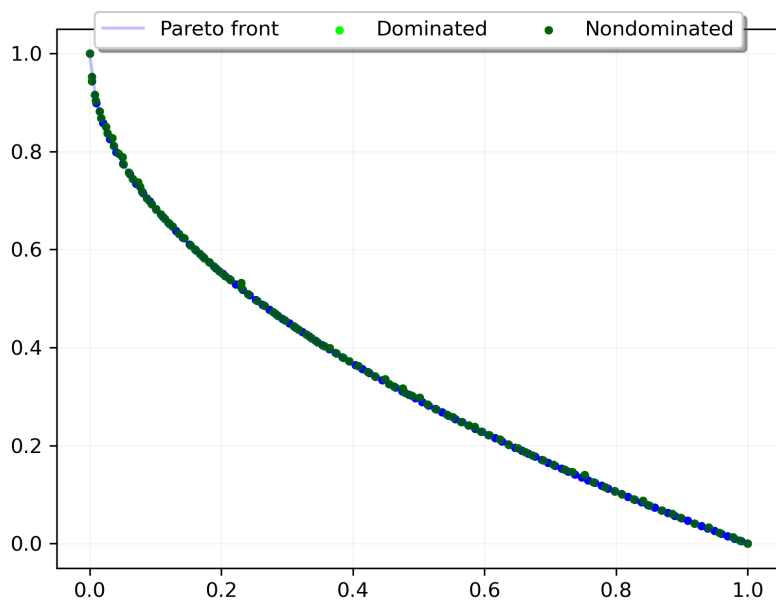
4.2.1. Problem testowy ZDT1

Tabela 4. Wyniki algorytmu SPEA-II dla problemu ZDT1

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	2,337	2,143
1000	50	0,5	1,811	2,404
1000	50	0,75	1,796	2,51
1000	100	0,25	2,543	1,906
1000	100	0,5	2,436	1,994
1000	100	0,75	2,349	2,017
1000	200	0,25	2,54	1,826
1000	200	0,5	2,599	1,716
1000	200	0,75	2,229	2,005
10000	50	0,25	0,164	4,418
10000	50	0,5	0,244	4,328
10000	50	0,75	0,276	4,285
10000	100	0,25	0,485	4,022
10000	100	0,5	0,474	4,002
10000	100	0,75	0,477	3,958
10000	200	0,25	1,178	3,174
10000	200	0,5	0,922	3,43
10000	200	0,75	0,987	3,325
100000	50	0,25	0,005	4,656
100000	50	0,5	0,028	4,62
100000	50	0,75	0,084	4,539
100000	100	0,25	0,005	4,661
100000	100	0,5	0,049	4,595
100000	100	0,75	0,106	4,515



Rysunek 11. Algorytm SPEA-II, problem ZDT1, generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25

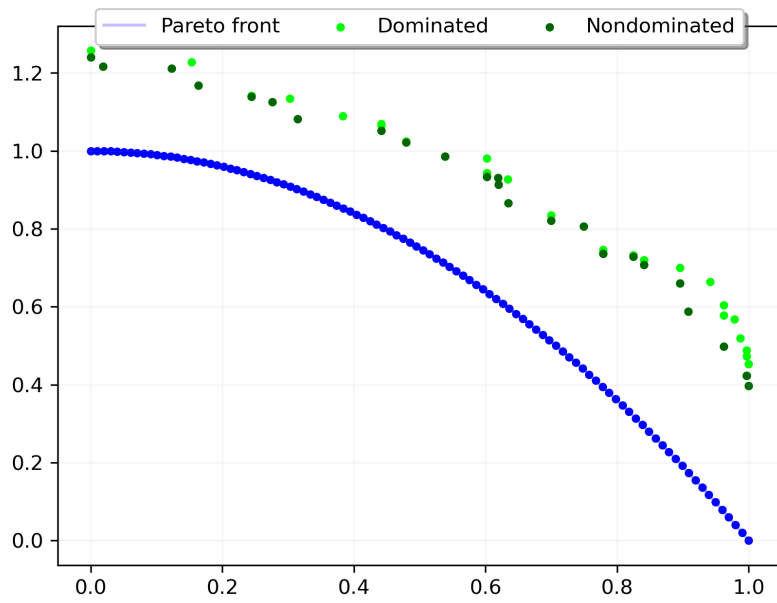


Rysunek 12. Algorytm SPEA-II, problem ZDT1, generacje - 100 000, populacja - 100, prawdopodobieństwo - 0,25

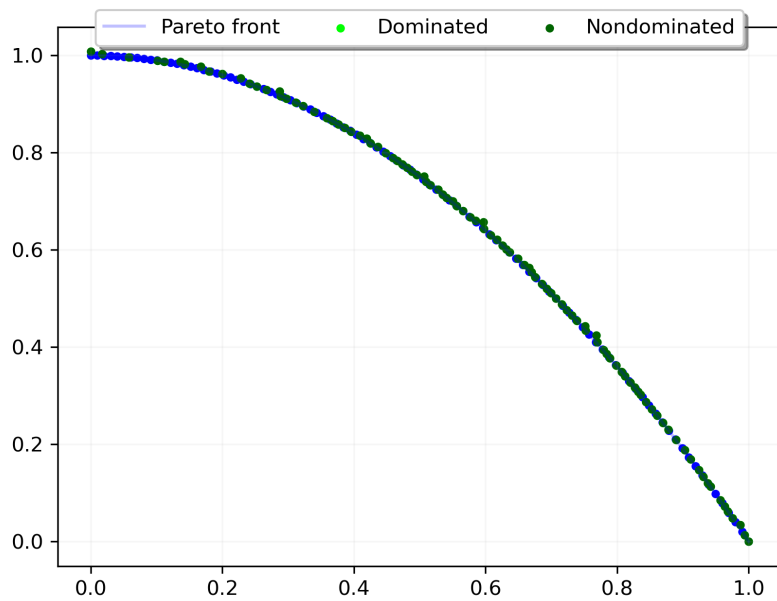
4.2.2. Problem testowy ZDT2

Tabela 5. Wyniki algorytmu SPEA-II dla problemu ZDT2

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	3,186	0,939
1000	50	0,5	2,609	1,442
1000	50	0,75	2,808	1,259
1000	100	0,25	3,38	0,663
1000	100	0,5	3,43	0,623
1000	100	0,75	3,259	0,915
1000	200	0,25	3,539	0,752
1000	200	0,5	3,386	0,693
1000	200	0,75	3,435	0,806
10000	50	0,25	0,205	4,033
10000	50	0,5	0,407	3,77
10000	50	0,75	0,529	3,644
10000	100	0,25	0,823	3,304
10000	100	0,5	0,863	3,254
10000	100	0,75	0,82	3,279
10000	200	0,25	1,886	2,175
10000	200	0,5	1,635	2,432
10000	200	0,75	1,64	2,439
100000	50	0,25	0,004	4,324
100000	50	0,5	0,051	4,249
100000	50	0,75	0,141	4,12
100000	100	0,25	0,004	4,327
100000	100	0,5	0,07	4,231
100000	100	0,75	0,185	4,062



Rysunek 13. Algorytm SPEA-II, problem ZDT2, generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25

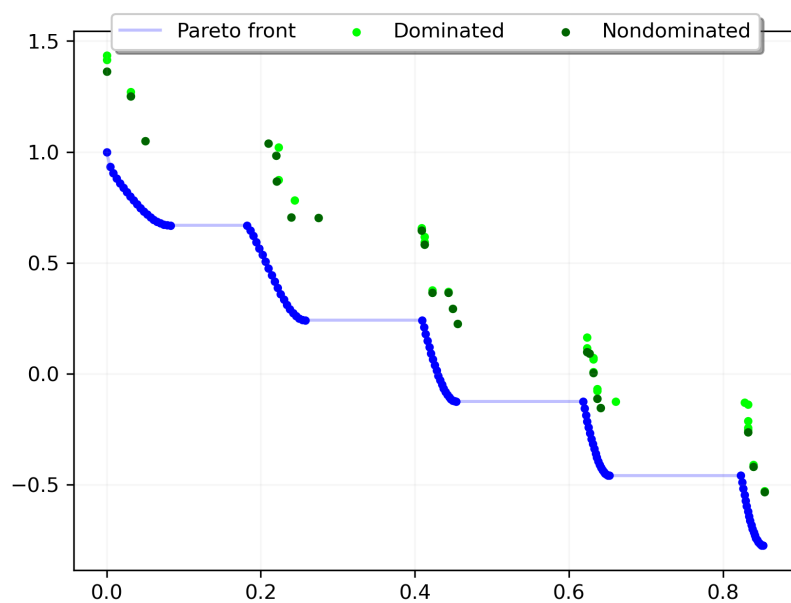


Rysunek 14. Algorytm SPEA-II, problem ZDT2, generacje - 100 000, populacja - 100, prawdopodobieństwo - 0,25

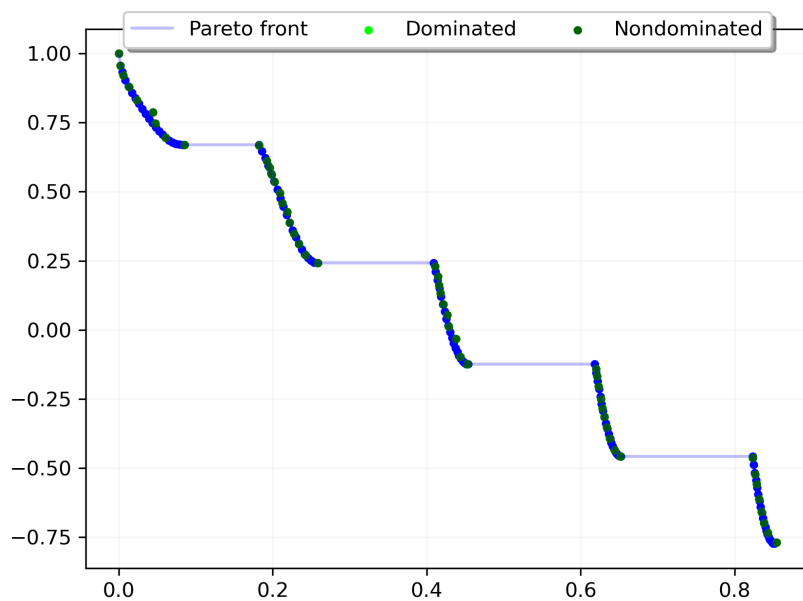
4.2.3. Problem testowy ZDT3

Tabela 6. Wyniki algorytmu SPEA-II dla problemu ZDT3

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	1,891	2,458
1000	50	0,5	1,714	2,727
1000	50	0,75	1,738	2,665
1000	100	0,25	2,362	2,169
1000	100	0,5	2,13	2,246
1000	100	0,75	2,27	2,288
1000	200	0,25	2,476	2,017
1000	200	0,5	2,318	2,045
1000	200	0,75	2,246	2,312
10000	50	0,25	0,143	4,676
10000	50	0,5	0,175	4,613
10000	50	0,75	0,249	4,477
10000	100	0,25	0,359	4,313
10000	100	0,5	0,506	4,18
10000	100	0,75	0,458	4,175
10000	200	0,25	1,073	3,411
10000	200	0,5	0,986	3,508
10000	200	0,75	0,911	3,607
100000	50	0,25	0,006	5,037
100000	50	0,5	0,017	4,975
100000	50	0,75	0,043	4,889
100000	100	0,25	0,006	5,037
100000	100	0,5	0,018	4,958
100000	100	0,75	0,068	4,847



Rysunek 15. Algorytm SPEA-II, problem ZDT3, generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25



Rysunek 16. Algorytm SPEA-II, problem ZDT3, generacje - 100 000, populacja - 50, prawdopodobieństwo - 0,25

4.3. Zmiana parametru k

Parametr k odpowiada za poszukiwanie k-tego najbliższego punktu na froncie Pareto. Sprawdzono czy zmiana tego parametru (domyślnie jego wartość wynosi 1) wpłynie zauważalnie na rezultaty algorytmu dla ustawień: generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25.

Tabela 7. Wyniki algorytmu SPEA-II (generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25) dla problemu ZDT1

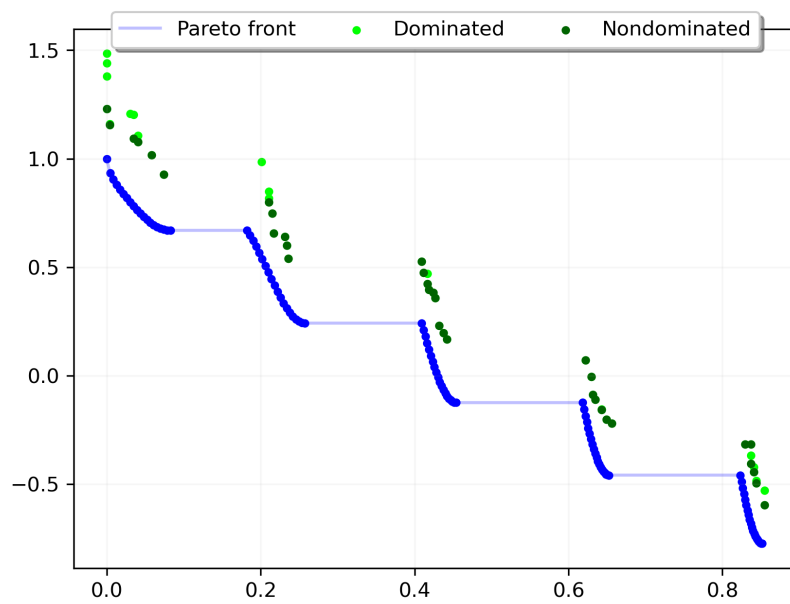
k	GD	HV
1	0,164	4,418
2	0,192	4,383
3	0,187	4,395

Tabela 8. Wyniki algorytmu SPEA-II (generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25) dla problemu ZDT2

k	GD	HV
1	0,205	4,033
2	0,266	3,951
3	0,281	3,952

Tabela 9. Wyniki algorytmu SPEA-II (generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25) dla problemu ZDT3

k	GD	HV
1	0,143	4,676
2	0,128	4,705
3	0,089	4,777



Rysunek 17. Algorytm SPEA-II, problem ZDT3, generacje - 100 000, populacja - 50, prawdopodobieństwo - 0,25, k - 3

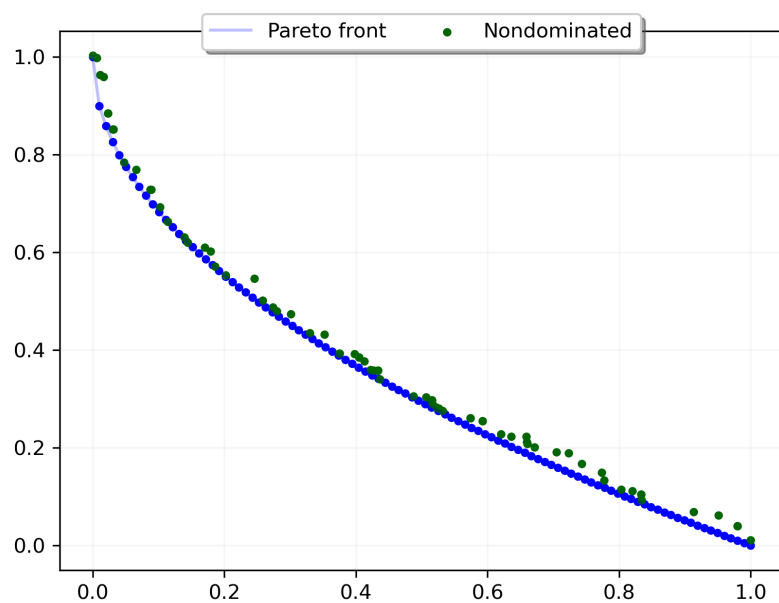
4.4. Algorytm PESA-II

Ze względu na specyfikację algorytmu PESA-II, na wykresach nie znajdują się zdominowane wyniki.

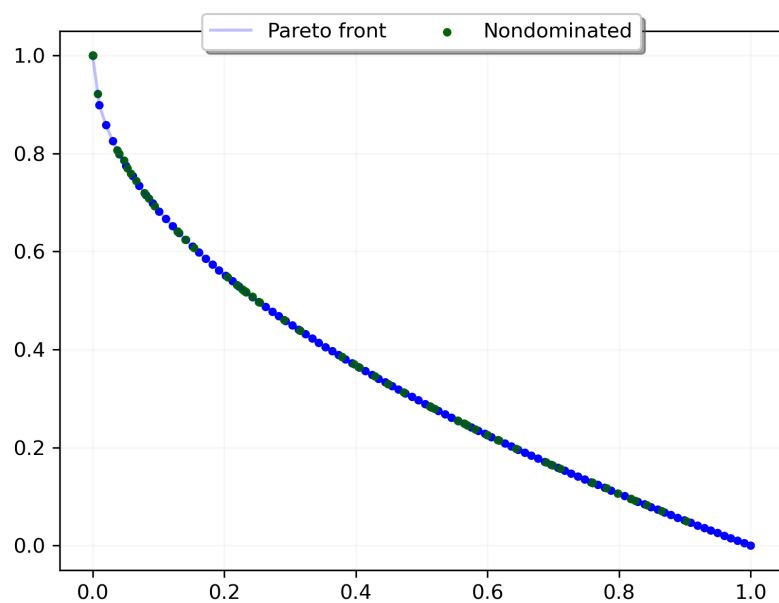
4.4.1. Problem testowy ZDT1

Tabela 10. Wyniki algorytmu PESA-II dla problemu ZDT1

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	0,849	3,604
1000	50	0,5	0,86	3,526
1000	50	0,75	0,667	3,74
1000	100	0,25	1,708	2,648
1000	100	0,5	1,508	2,934
1000	100	0,75	1,186	3,281
1000	200	0,25	2,03	2,52
1000	200	0,5	1,476	2,868
1000	200	0,75	1,742	2,603
10000	50	0,25	0,015	4,639
10000	50	0,5	0,089	4,53
10000	50	0,75	0,143	4,45
10000	100	0,25	0,019	4,635
10000	100	0,5	0,09	4,537
10000	100	0,75	0,182	4,397
10000	200	0,25	0,041	4,605
10000	200	0,5	0,105	4,503
10000	200	0,75	0,173	4,403
100000	50	0,25	0,003	4,651
100000	50	0,5	0,021	4,636
100000	50	0,75	0,063	4,564
100000	100	0,25	0,004	4,653
100000	100	0,5	0,021	4,635
100000	100	0,75	0,078	4,549
100000	200	0,25	0,004	4,651
100000	200	0,5	0,02	4,635
100000	200	0,75	0,065	4,566



Rysunek 18. Algorytm PESA-II, problem ZDT1, generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25

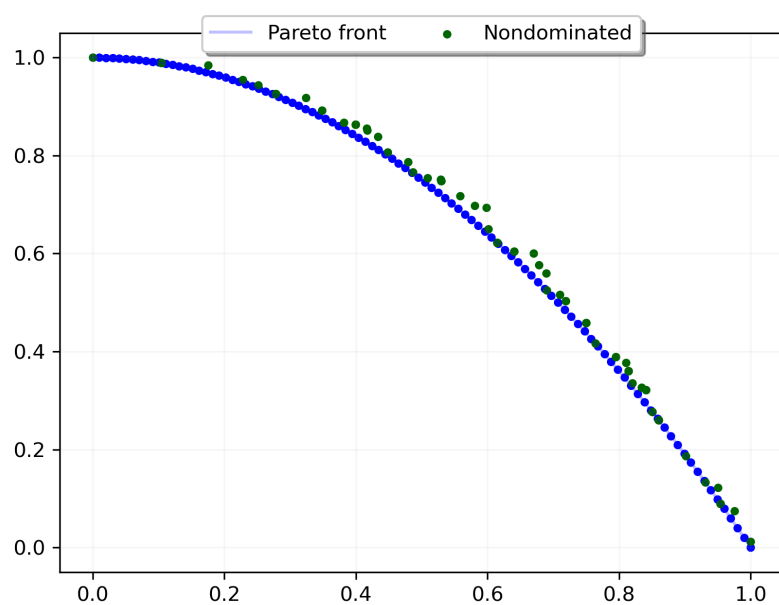


Rysunek 19. Algorytm PESA-II, problem ZDT1, generacje - 100 000, populacja - 50, prawdopodobieństwo - 0,25

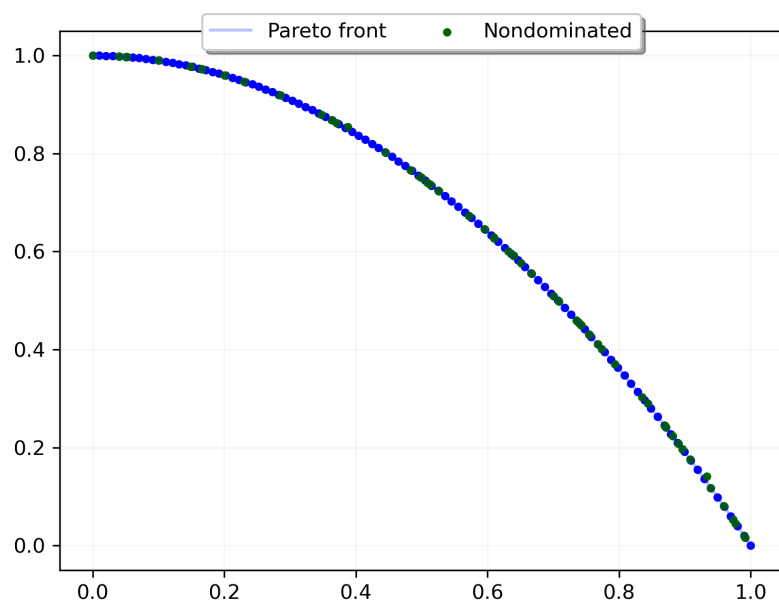
4.4.2. Problem testowy ZDT2

Tabela 11. Wyniki algorytmu PESA-II dla problemu ZDT2

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	1,127	2,996
1000	50	0,5	0,496	3,504
1000	50	0,75	0,884	3,177
1000	100	0,25	1,616	2,454
1000	100	0,5	1,303	2,715
1000	100	0,75	1,066	2,934
1000	200	0,25	2,701	1,408
1000	200	0,5	1,912	2,206
1000	200	0,75	1,757	2,304
10000	50	0,25	0,011	4,31
10000	50	0,5	0,11	4,174
10000	50	0,75	0,22	4,008
10000	100	0,25	0,012	4,305
10000	100	0,5	0,088	4,189
10000	100	0,75	0,198	4,032
10000	200	0,25	0,038	4,27
10000	200	0,5	0,105	4,165
10000	200	0,75	0,229	3,988
100000	50	0,25	0,004	4,32
100000	50	0,5	0,022	4,295
100000	50	0,75	0,092	4,185
100000	100	0,25	0,004	4,32
100000	100	0,5	0,025	4,289
100000	100	0,75	0,1	4,183
100000	200	0,25	0,005	4,321
100000	200	0,5	0,018	4,299
100000	200	0,75	0,117	4,159



Rysunek 20. Algorytm PESA-II, problem ZDT2, generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25

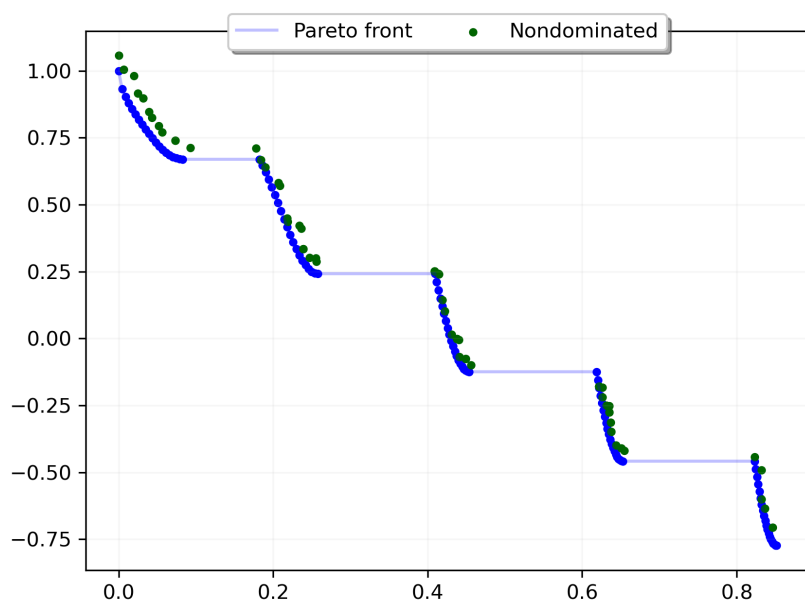


Rysunek 21. Algorytm PESA-II, problem ZDT2, generacje - 100 000, populacja - 50, prawdopodobieństwo - 0,25

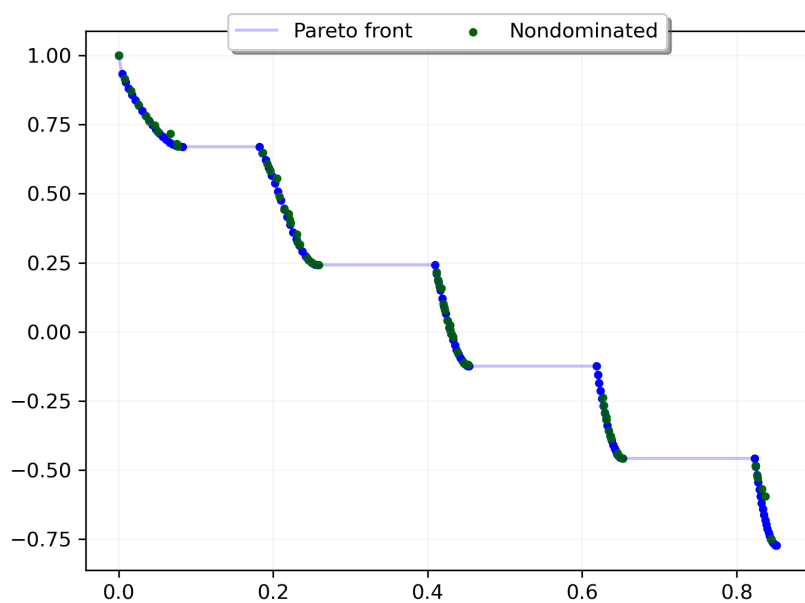
4.4.3. Problem testowy ZDT3

Tabela 12. Wyniki algorytmu PESA-II dla problemu ZDT3

Generacje	Populacja	Prawdo.	GD	HV
1000	50	0,25	1,264	3,313
1000	50	0,5	0,752	3,766
1000	50	0,75	0,67	3,799
1000	100	0,25	1,083	3,334
1000	100	0,5	0,991	3,449
1000	100	0,75	1,194	3,348
1000	200	0,25	1,534	3,158
1000	200	0,5	1,317	3,06
1000	200	0,75	1,448	3,023
10000	50	0,25	0,013	4,991
10000	50	0,5	0,039	4,915
10000	50	0,75	0,126	4,703
10000	100	0,25	0,02	4,977
10000	100	0,5	0,082	4,815
10000	100	0,75	0,12	4,746
10000	200	0,25	0,023	4,922
10000	200	0,5	0,068	4,877
10000	200	0,75	0,104	4,749
100000	50	0,25	0,006	5,034
100000	50	0,5	0,012	5,006
100000	50	0,75	0,037	4,902
100000	100	0,25	0,005	5,031
100000	100	0,5	0,01	5,007
100000	100	0,75	0,033	4,929
100000	200	0,25	0,005	5,032
100000	200	0,5	0,011	4,996
100000	200	0,75	0,044	4,92



Rysunek 22. Algorytm PESA-II, problem ZDT3,
generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25



Rysunek 23. Algorytm PESA-II, problem ZDT3,
generacje - 100 000, populacja - 200, prawdopodobieństwo - 0,25

4.5. Zmiana parametru divisions

Sprawdzono czy zmiana parametru divisions (domyślnie jego wartość wynosi 8) wpłynie zauważalnie na rezultaty algorytmu dla ustawień: generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25.

Tabela 13. Wyniki algorytmu PESA-II (generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25) dla problemu ZDT1

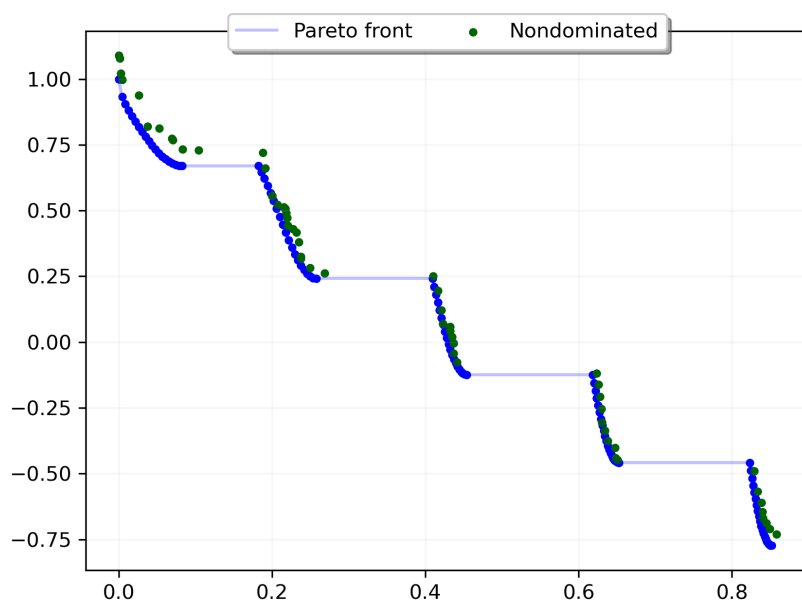
divisions	GD	HV
4	0,016	4,636
8	0,015	4,639
12	0,02	4,628

Tabela 14. Wyniki algorytmu PESA-II (generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25) dla problemu ZDT2

divisions	GD	HV
4	0,013	4,304
8	0,011	4,31
12	0,018	4,294

Tabela 15. Wyniki algorytmu PESA-II (generacje - 10 000, populacja - 50, prawdopodobieństwo - 0,25) dla problemu ZDT3

divisions	GD	HV
4	0,014	4,998
8	0,013	4,991
12	0,017	4,987



Rysunek 24. Algorytm PESA-II, problem ZDT3, generacje - 100 000, populacja - 50, prawdopodobieństwo - 0,25, divisions - 4

5. Wnioski

- Większa liczba generacji pozytywnie wpływa na uzyskanie wyniki optymalizacji, jednocześnie zauważalnie wydłużając czas działania algorytmów. Dla 100 000 generacji rezultaty pokrywały się z frontem Pareto.
- Dla niskich wartości generacji, prawdopodobieństwo 0,25 było najgorszym wyborem. Optymalny wybór nie był ściśle powiązany z żadnym algorytmem bądź problemem.
- Dla każdego algorytmu oraz dla wysokich wartości generacji (10 000 oraz 100 000) najlepszym prawdopodobieństwem było 0,25.
- Wszystkie algorytmy osiągały najlepsze rezultaty dla populacji 50 - dla 100 000 generacji wielkość populacji nie miała znaczenia. Duża populacja przy miarach GD oraz Hypervolume nie wpływa pozytywnie na wyniki.
- Średnio, najlepsze wyniki uzyskał algorytm PESA-II, a najgorsze SPEA-II. Dla 100 000 generacji różnice te są pomijalne.
- Zmiana parametrów konkretnych argumentów nie wpływa znacząco na rezultaty.

Bibliografia

- [1] Jason Brownlee. *Clever Algorithms: Nature-Inspired Programming Recipes*. <http://www.cleveralgorithms.com>.
- [2] David Corne, Joshua Damian Knowles,, Martin J. Oates. *NSGA2 based of multi-criteria decision analysis for multi-objective optimization of fuzzy logic controller for non linear system. TheParetoEnvelop-basedSelectionAlgorithmforMulti-ObjectiveOptimization*.
- [3] Rahman Ghararia, Navid Poursalehib, Mohammadreza Abbasib,, Mahdi Aghaie. *Implementation of Strength Pareto Evolutionary Algorithm II in the Multiobjective Burnable Poison Placement Optimization of KWU Pressurized Water Reactor*. <https://www.sciencedirect.com/science/article/pii/S1738573316300493#bib14>.
- [4] Kheireddine Lamamra, Khaled Belarbi, Ahmed Belhani,, Souaad Boukhtini. *NSGA2 based of multi-criteria decision analysis for multi-objective optimization of fuzzy logic controller for non linear system*. https://www.researchgate.net/publication/286493028_NSGA2_based_of_multi-criteria_decision_analysis_for_multi-objective_optimization_of_fuzzy_logic_controller_for_non_linear_system.
- [5] Fuqing Zhao. *An Improved SPEA2 Algorithm with Adaptive Selection of Evolutionary Operators Scheme for Multiobjective Optimization Problems*. <https://www.hindawi.com/journals/mpe/2016/8010346/>.