Raport – Metaheurystyka genetyczna (AG)

Laboratorium - Metaheurystyki

Ernest Przybył

1. Wstęp

Metaheurystyka genetyczna

Potocznie zwana **algorytmem genetycznym**. Jest to metaheurystyką inspirowaną naturą. W uproszczeniu. Inicjalizujemy losową populację często o określonej wielkości, charakteryzującą się daną formą **genotypu**. Następnie generujemy kolejne pokolenie z rodziców wybranych określony sposób. Chodzi o to, żeby w następnym pokoleniu w „dzieciach" przetrwała przynajmniej część dobrze ocenianych cech rodziców, jednak w pewien sposób zmodyfikowana. Chcemy przeglądać w każdym pokoleniu nowe możliwości dlatego oprócz krzyżowania rodziców stosuje się operator mutacji który dodatkowo wprowadza losowe zmiany w „dziecku".

TTP - Traveling Thief Problem

Popularny problem służący jako benchmark dla wielu algorytmów. Tytułowy złodziej musi odwiedzić N miast w których może ukraść określone przedmioty. Musimy znaleźć dla niego trasę oraz listę przedmiotów do ukradzenia w taki sposób, aby zysk z przedmiotów był jak największy po odjęciu kosztów podróży. Pojemność plecaka jest ograniczona, a większa ilość przedmiotów znajdujących się w nim sprawia, że złodziej porusza się wolniej. Ostatecznie problem TTP możemy rozbić na dwa pod problemy:

- TSP – czyli znajdywanie jak najlepszej trasy dla złodzieja
- Problem Plecakowy – znajdywanie jak najlepszego sposobu na upakowanie przedmiotów w plecaku.

Problem: Jak połączyć problemy TSP i Plecakowy w AG?

Na razie nie dysponujemy takimi narzędziami, a stworzenie genotypu z zawartości plecaka i kolejności miast byłoby bardzo niekorzystne w AG. Dlaczego?

W AG liczymy na to, że **niewielka zmiana genotypu** przyniesie również niewielkie zmiany w ocenie osobnika. Tymczasem, gdyby zmienić kolejność któregoś z miast wycena plecaka mogłaby się zmienić diametralnie, a zawartość plecaka byłaby nieadekwatna do trasy.

Potencjalne rozwiązania:

- Uruchomić AG do wyznaczenia trasy, a następnie dla każdego utworzonego osobnika AG dla plecaka:
  Niestety takie podejście zabiłoby wydajność naszego programu i nie byłoby z tego powodu lepsze od innych rozwiązań.
- Uruchomić AG dla plecaka i wyznaczyć trasę zachłannie:
  Takie podejście miałoby sens, ale przy dużej pojemności plecaka w stosunku do wielkości przedmiotów algorytm zachłanny jest w stanie poradzić sobie całkiem dobrze i w czasie krótszym od AG. Dlatego rezygnuję z tego rozwiązania.

- Uruchomić AG do wyznaczenia trasy, a plecak wyznaczać z pomocą algorytmu zachłannego: Algorytm zachłanny w większości wypadków powinien poradzić sobie w miarę dobrze z plecakiem. Natomiast AG zapewni nam trasę bliską optymalnej.

Reprezentacja/Genotyp

Każdy osobnik jest reprezentowany jako permutacja zbioru od 0 do N-1, gdzie N to liczba miast. Kolejność miast w permutacji jest równoważna z kolejnością odwiedzania przez „złodzieja". Przedmioty w plecaku są wyznaczane zachłannie w sposób **deterministyczny**.

Selektor zachłanny przedmiotów

Dla określonej z góry trasy obliczamy dla każdego przedmiotu jego rzeczywistą wartość. Robimy to za pomocą takiego wzoru:

$$W_r = W_b - (m \cdot d)$$

Gdzie W_r to realna wartość przedmiotu. W_b wartość przedmiotu bazowa. m - masa przedmiotu, d dystans jaki musi przebyć przedmiot do końca trasy.

Ta metoda ma pewną wadę, ponieważ nie uwzględnia tego, że złodziej ma określoną minimalną prędkość.

Selekcja rodziców

Zaimplementowano dwa sposoby selekcji rodziców. Ruletka oraz Turniej. Ruletka ma jednak pewną wadę w stosunku do turnieju. Dla każdego problemu musimy zastosować odpowiednią funkcję, która będzie nam „oddalała" rozwiązania, jeżeli różnice pomiędzy wszystkimi osobnikami będą stosunkowo niewielkie. Zdecydowałem się, że najłatwiej będzie podnosić oceny osobników do n-tej potęgi co uwypukli różnicę między osobnikami na tablicy ruletowej.

W obu metodach określamy pojęcie tj. **ciśnienie selekcyjne**. Dostosowujące parametry tych metod możemy określić jak mocno są faworyzowane obecnie najlepsze osobnik.

Zbyt duże ciśnienie selekcyjne sprawi, że nasza populacja może utknąć w optimum lokalnym, ponieważ do rozrodu będą wybierane tylko osobniki zawierające się w nim przez to **zróżnicowanie populacji** mocno spadnie i nie będziemy eksplorować nowych terenów.

Z kolei zbyt małe ciśnienie sprawi, że osobnikom nie będzie się opłacało posiadać dobre cechy i wszelkie dobre tropy do jakiegoś optimum zostaną zbyt szybko zapomniane. Poszukiwanie będzie przypominało coraz bardziej losowe przeszukiwanie przestrzeni rozwiązań.

Krzyżowanie rodziców

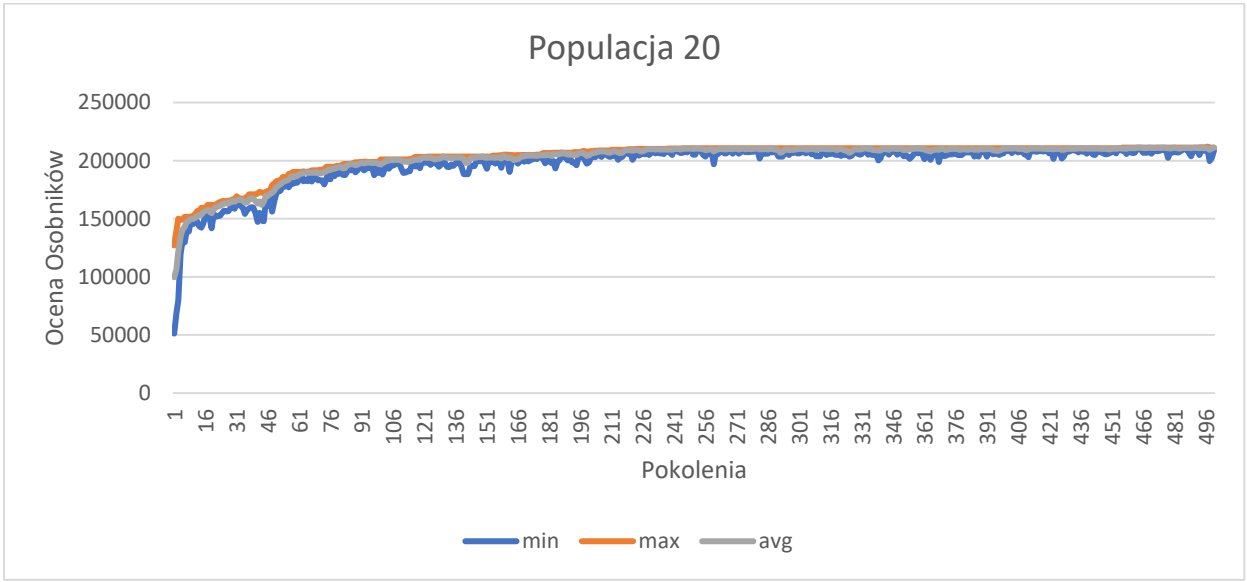
Mutacja

2. Eksperymenty

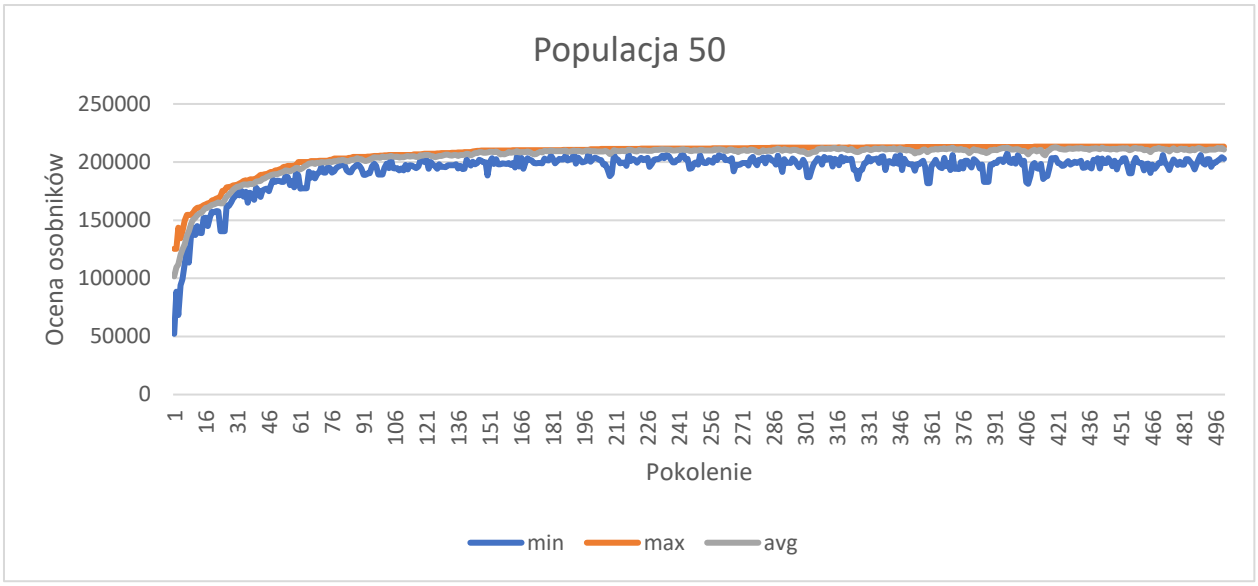
Uwaga. Wszystkie eksperymenty przeprowadzono na zestawie medium_4.ttp

Wielkość populacji

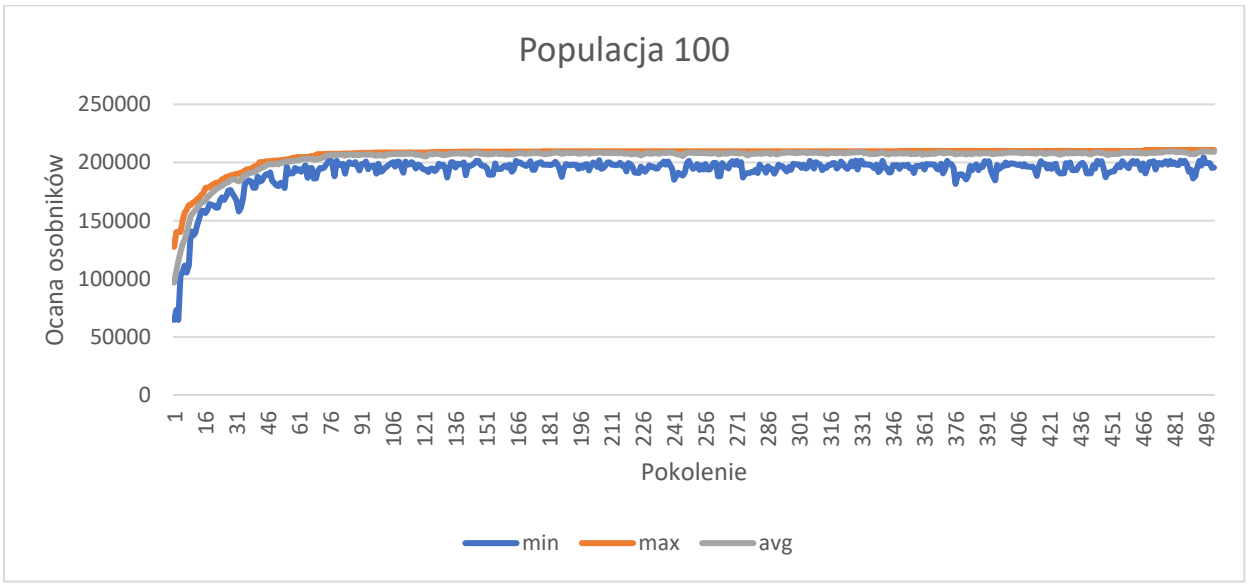
- Cel: Określenie wpływu wielkości populacji na efektywność AG.
- Stałe
  - Pokolenia: 500
  - Mutacja: Swap; intensywność = 0.1; prawdopodobieństwo = 0.2
  - Krzyżowanie: Ordered; prawdopodobieństwo – 0.7
  - Selekcja: Turniej(3/52);
- Zmienne:
  - Wielkość populacji: [20, 50, 100, 200]



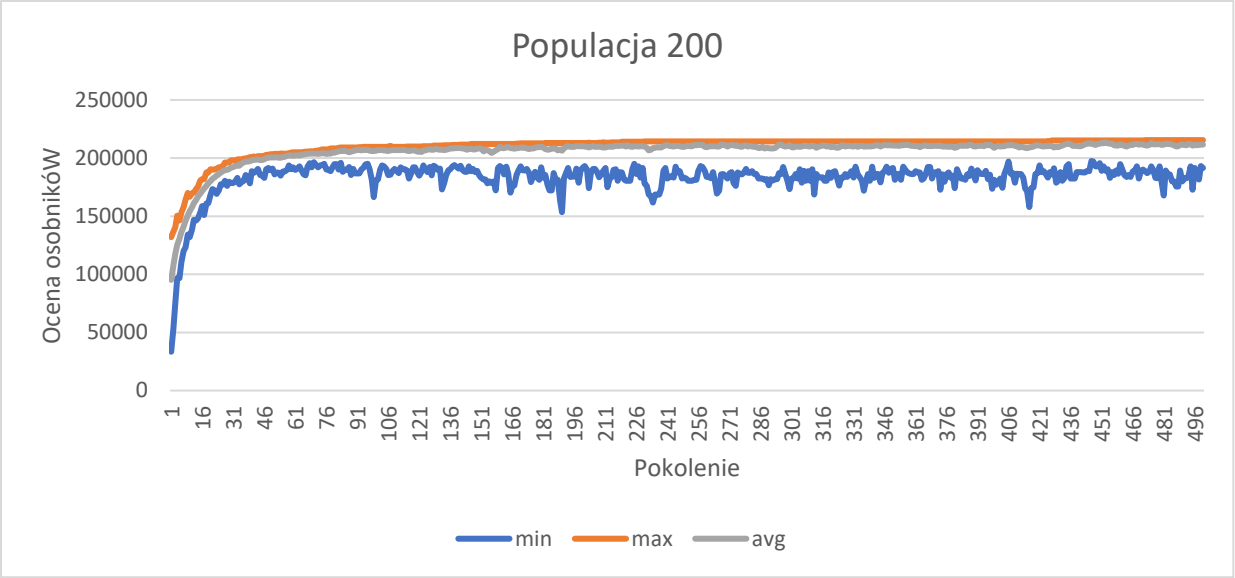
Wykr. 1


Populacja 50
Ocena osobników
250000
200000
150000
100000
50000
0
1
16
31
46
61
76
91
106
121
136
151
166
181
196
211
226
241
256
271
286
301
316
331
346
361
376
391
406
421
436
451
466
481
496
Pokolenie
min
max
avg

Wykr. 2


Populacja 100
Ocana osobników
250000
200000
150000
100000
50000
0
1
16
31
46
61
76
91
106
121
136
151
166
181
196
211
226
241
256
271
286
301
316
331
346
361
376
391
406
421
436
451
466
481
496
Pokolenie
min
max
avg

Wykr. 3



Wykr. 4

Obserwacje i wnioski:

Na wykresie 1. W okolicach 45 pokolenia obserwujemy spadek jakości populacji. Z byt mała wielkość populacji może spowodować, że zbyt szybko stracimy dobry trop i pójdziemy szukać dalej. Można temu zapobiec zwiększając ciśnienie selekcyjne, ale to może spowodować, że nasza populacja utknie w tym „dobrym tropie”. Zbyt mała wielkość populacji nie pozwala nam zachować śladu pewnych dobrych cech rozwiązania które próbujemy osiągnąć.

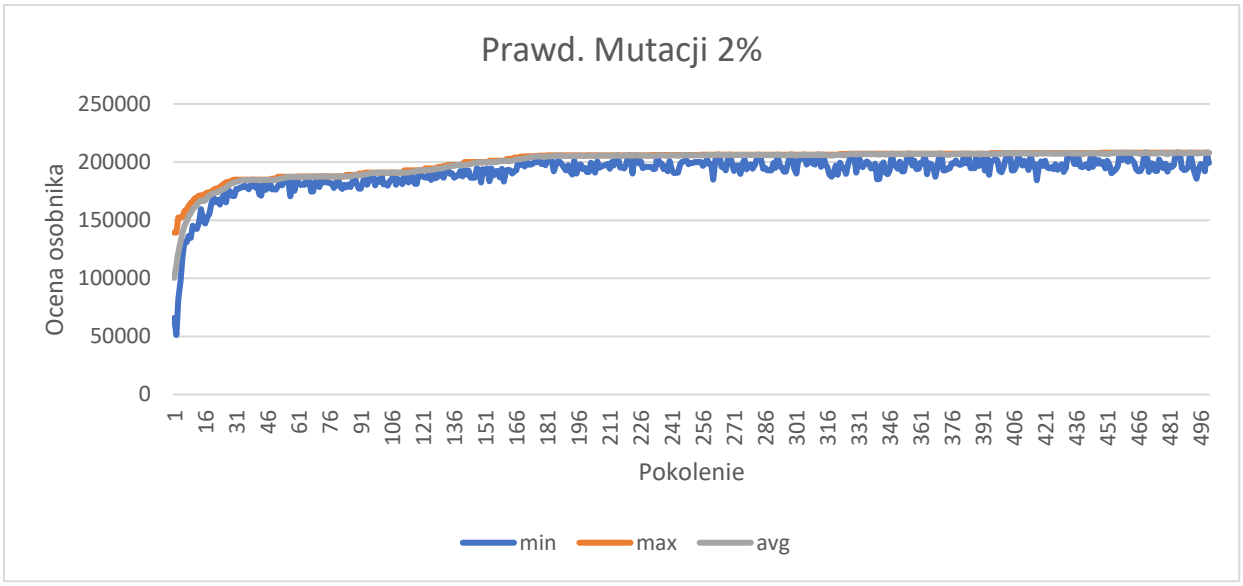
Dodatkowo mała populacja powoduje, zmniejszenie różnorodności populacji co widać na wykresie 1. Wartości min i max są znacznie bliżej niż na wykresach 3 i 4. Mała różnorodność może źle wpłynąć na poszukiwania optymalnego rozwiązania.

Różnica po między efektywnością populacji o wielkości 100 i 200 (wykr. 3 i 4). Nie jest proporcjonalnie 2 razy większa. Pomimo iż musimy przeszukać 2x więcej osobników otrzymujemy podobne wyniki w kolejnych pokoleniach.

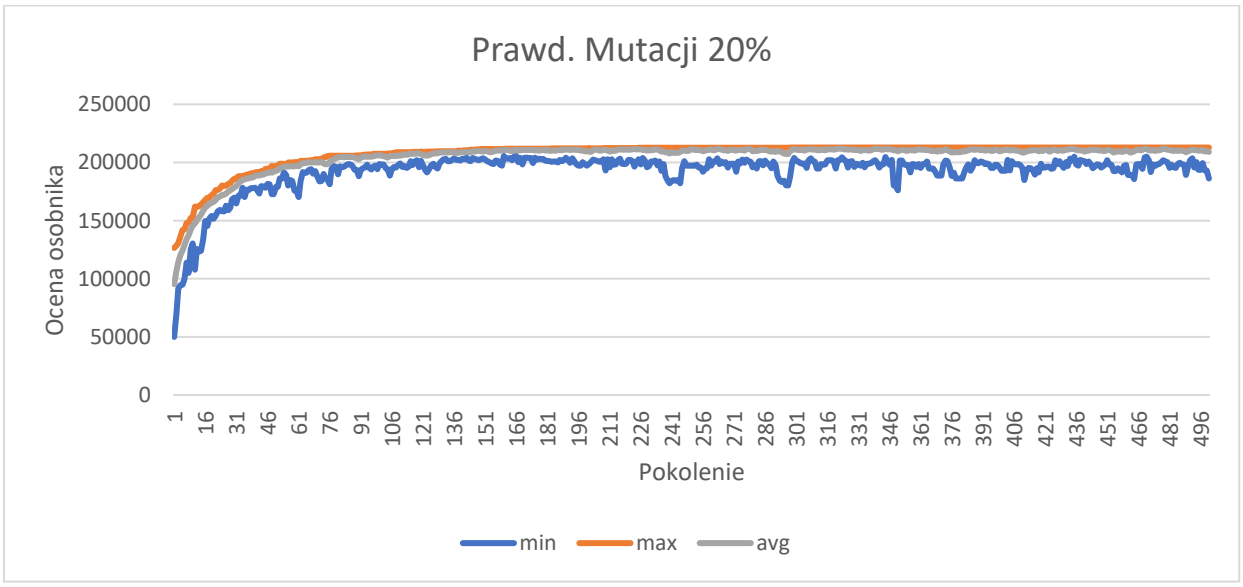
Mam wrażenie, że zbyt mała ilość miast dla problemu medium(52) nie pozwala dokonywać pewnych obserwacji.

Prawdopodobieństwo mutacji

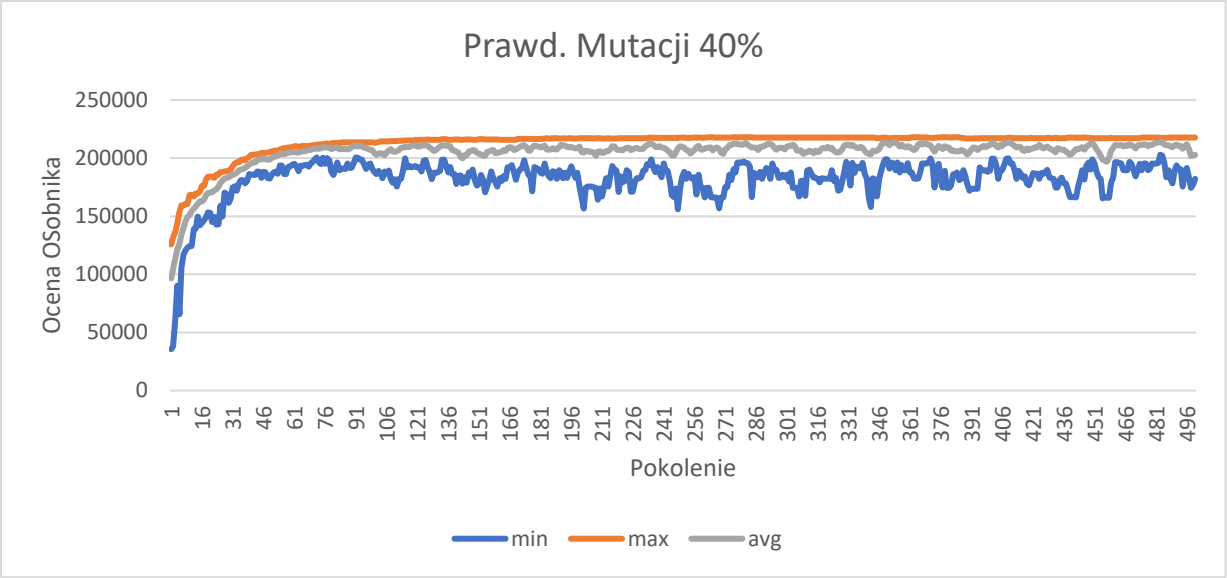
- Cel: Określenie wpływu prawdopodobnieństwa mutacji na efektywność AG.
- Stałe
    - Pokolenia: 500
    - Krzyżowanie: Ordered; prawdopodobieństwo – 0.7
    - Wielkość populacji: 100
    - Selekcja: Turniej(3/52);
- Zmienne:
    - Mutacja: Swap; intensywność = 0.15; prawdopodobieństwo = [0.02, 0.2, 0.4]


Prawd. Mutacji 2%
250000
200000
150000
100000
50000
0
Ocena osobnika
1
16
31
46
61
76
91
106
121
136
151
166
181
196
211
226
241
256
271
286
301
316
331
346
361
376
391
406
421
436
451
466
481
496
Pokolenie
min
max
avg

Wykr. 5


Prawd. Mutacji 20%
250000
200000
150000
100000
50000
0
Ocena osobnika
1
16
31
46
61
76
91
106
121
136
151
166
181
196
211
226
241
256
271
286
301
316
331
346
361
376
391
406
421
436
451
466
481
496
Pokolenie
min
max
avg

Wykr. 6



Wykr. 7

Obserwacje i wnioski:

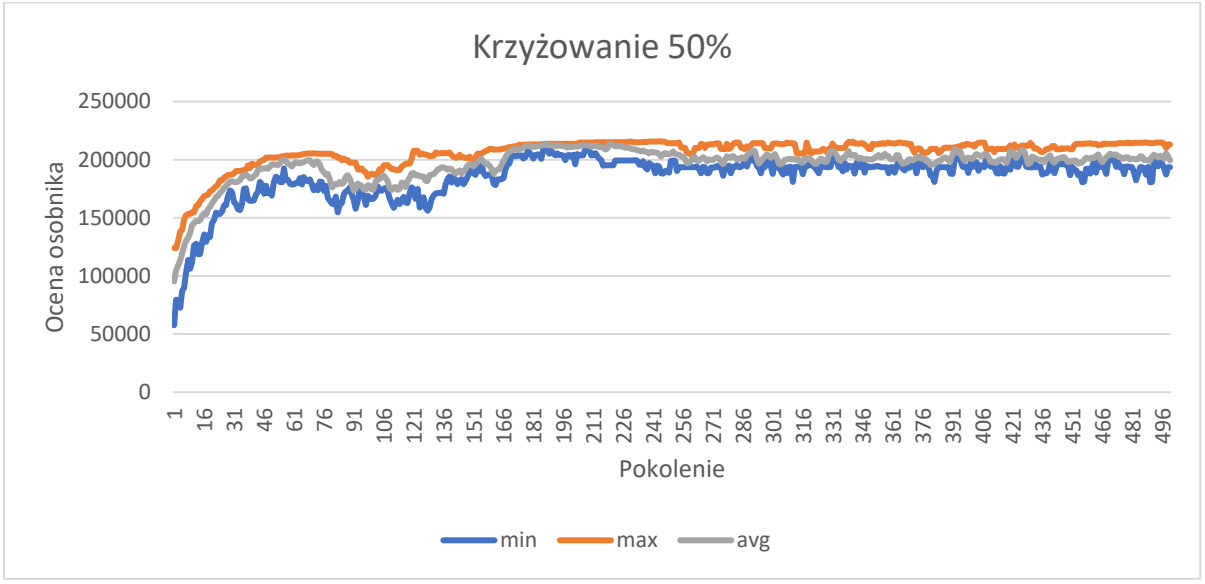
Na wykresie 5 i 6 widać, że większa ilość mutacji przyspieszyła znajdywanie dobrych rozwiązań. Na wykresie 5 nadal jest prowadzona eksploatacja rozwiązań ponieważ wciąż dokonujemy krzyżowania osobników które ma w sobie element losowości.

Wyższe wartości prawd. Mutacji sprawiają, że wartości min i max oddalają się od siebie. Oznacza to, że rośnie różnorodność naszej populacji i możemy lepiej przeszukiwać przestrzeń rozwiązań.

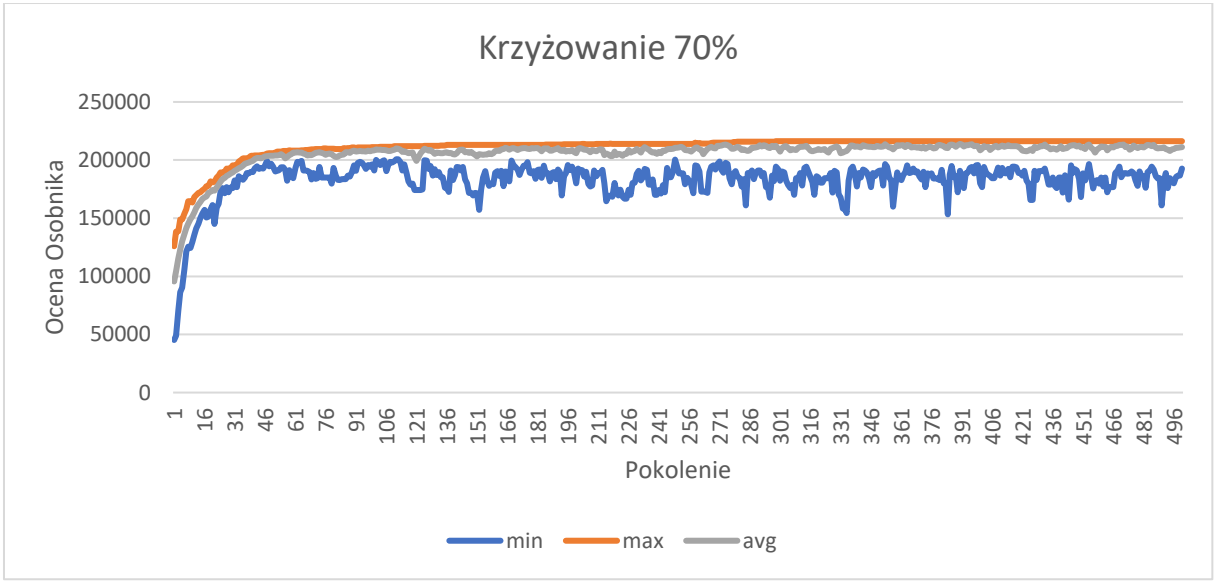
Na wykresie 7 widać, że zbyt duża ilość mutacji wpływa także na zachwiania jakości całej populacji (avg) choć mimo wszystko udało się oszczędzić najlepszego osobnika (max).

Prawd. krzyżowania

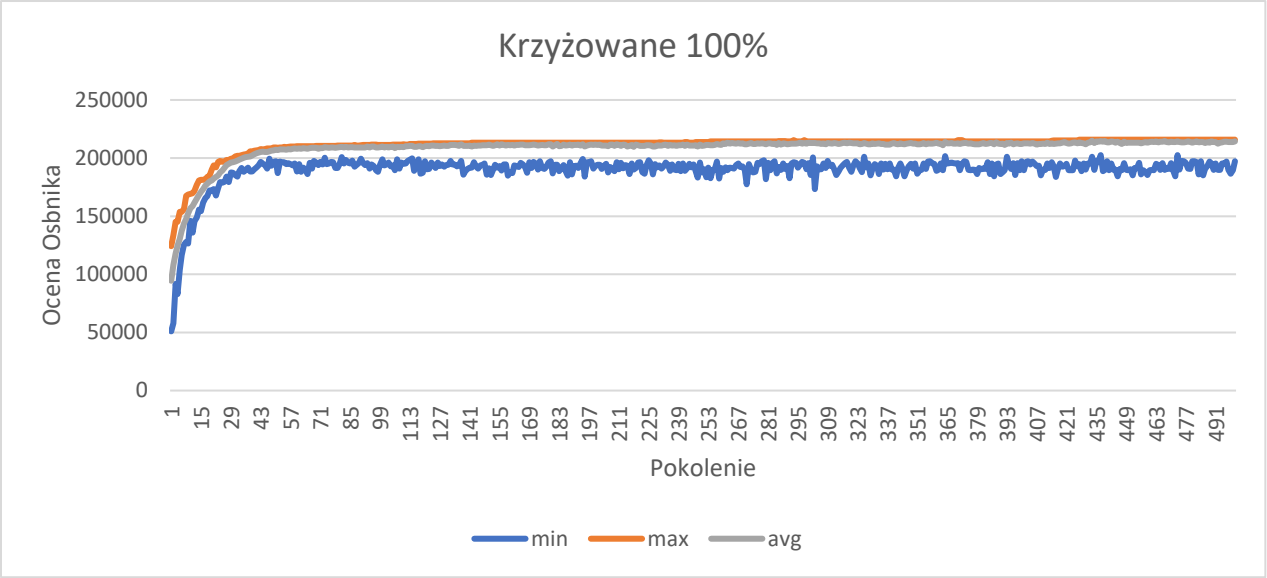
- Cel: Określenie wpływu częstości krzyżowań na efektywność AG.
- Stałe
    - Pokolenia: 500
    - Mutacja: Swap; intensywność = 0.15 -> 0.05; prawdopodobieństwo = 0.2 -> 1.0
    - Wielkość populacji: 100
    - Selekcja: Turniej(3/52);
- Zmienne:
    - Krzyżowanie: Ordered; prawdopodobieństwo –[0,5; 0.7; 1]



Wykr. 8



Wykr. 9



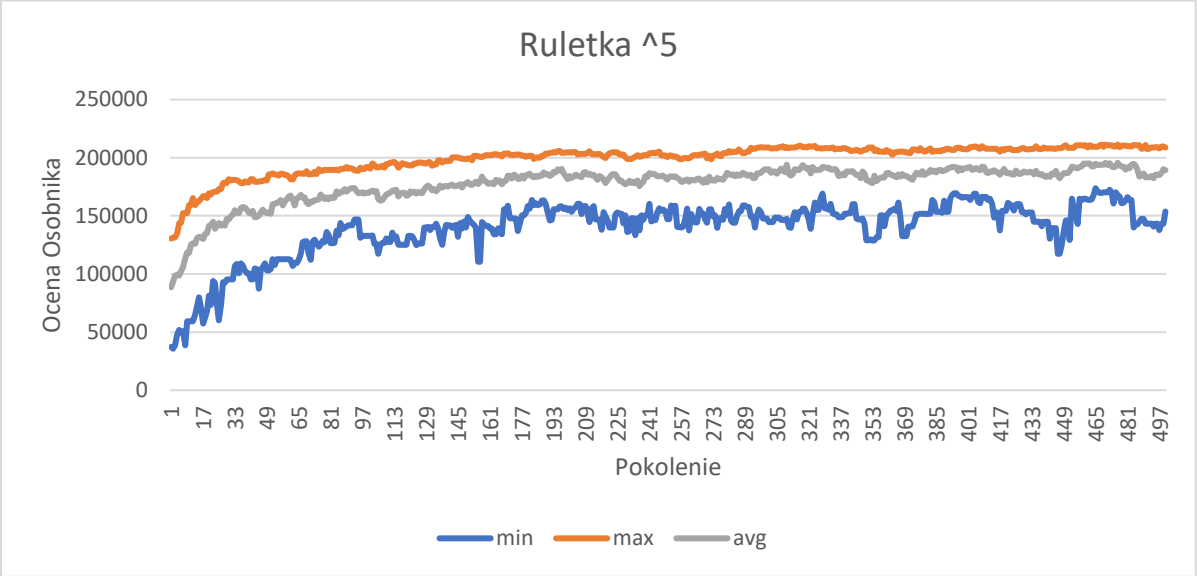
Wykr. 10

Obserwacje i wnioski:

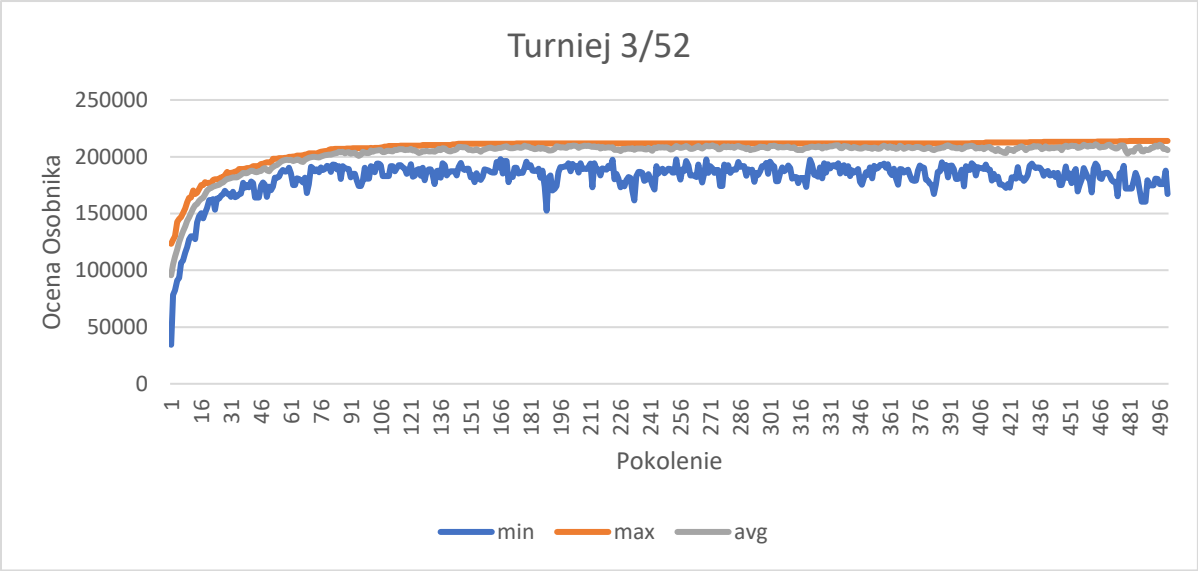
Spodziewał bym się, że stabilniejszy wykres będzie, gdy ilość krzyżowań jest mniejsza. W końcu wtedy większa ilość niezmienionych osobników przenika następnego pokolenia. Jednak okazało się, że jest inaczej i wykres 10 staje się bardzo szybko stabilny.

Metoda Selekcji

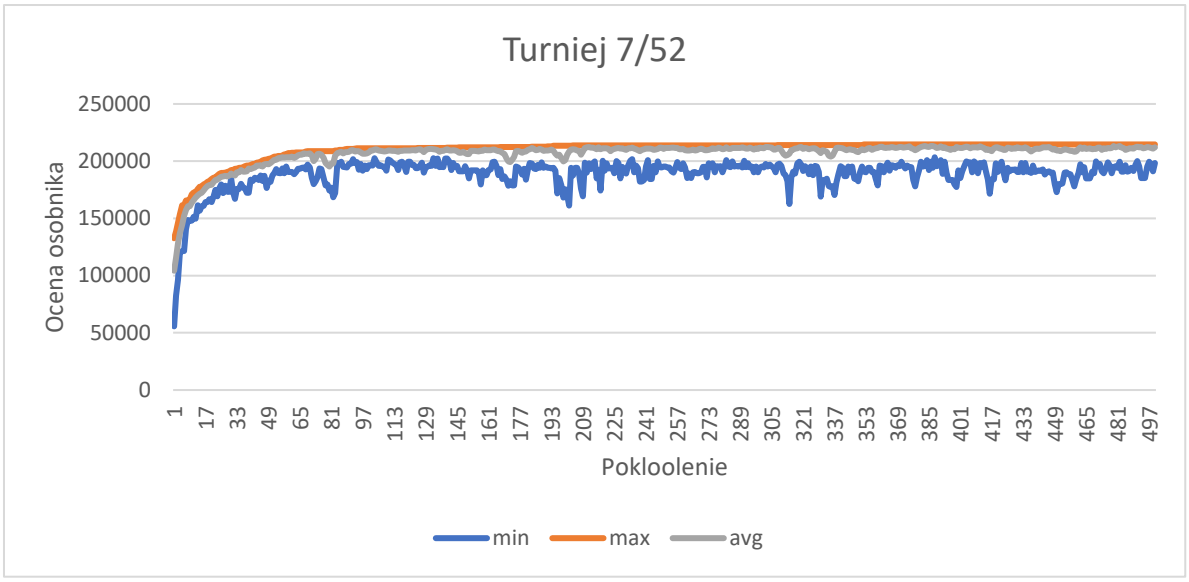
- Cel: Określenie wpływu metody selekcji na wyniki AG
- Stałe
    - Pokolenia: 500
    - Mutacja: Swap; intensywność = 0.15; prawdopodobieństwo = 0.2
    - Wielkość populacji: 100
    - Krzyżowanie: Ordered; prawdopodobieństwo – 0.7
    -
- Zmienne:
    - Selekcja: [Turniej(3/52); Turniej(7/52); Turniej(1/52);Turniej(52/52) ; Ruletka(^5)]



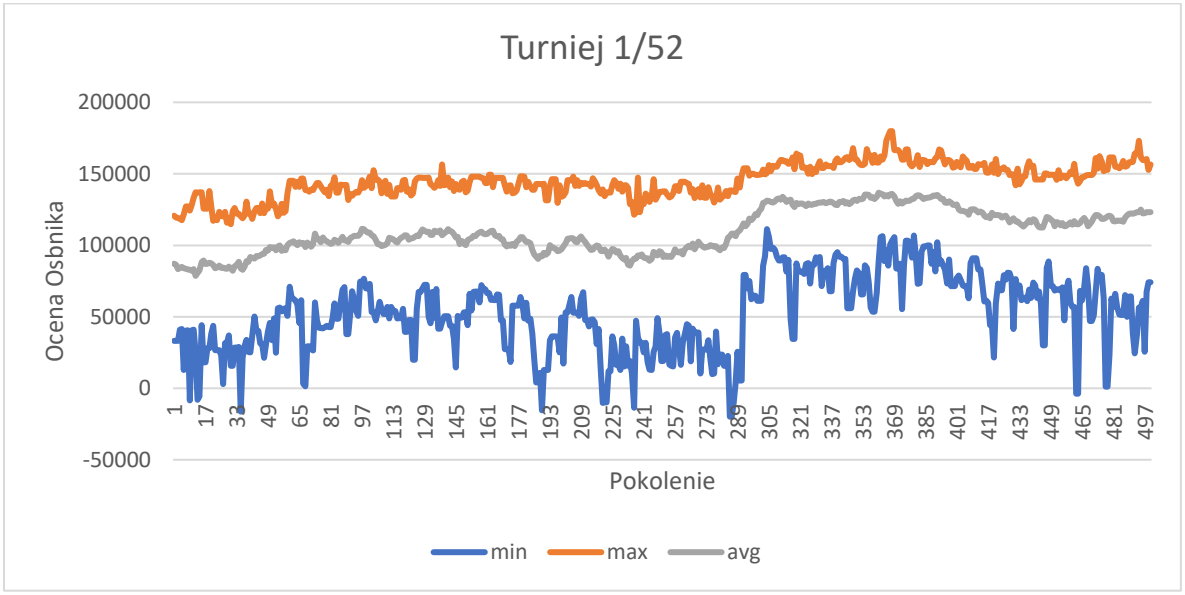
Wykr. 11



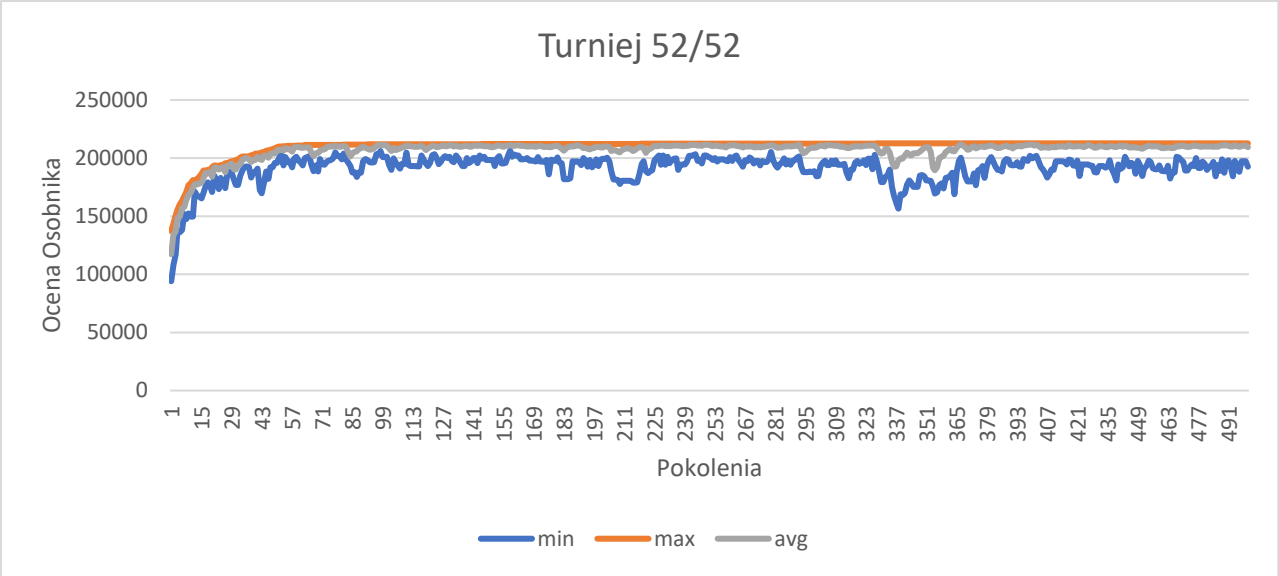
Wykr. 12



Wykr. 13



Wykr. 14



Wykr. 15

Obserwacje i wnioski:

Na wykresie 11 wartości min max odbiegają od siebie znacznie bardziej niż na wykresie 12. Może to oznaczać, że przy zastosowanych przez nas parametrach uzyskaliśmy mniejsze ciśnienie selekcyjne niż przy Turnieju. Szczególnie, że wartość avg i max na wykresie 11 są także bardzo odległe w porównaniu do 12. Oznacza to tyle, że populacja nie jest bardziej zróżnicowana.

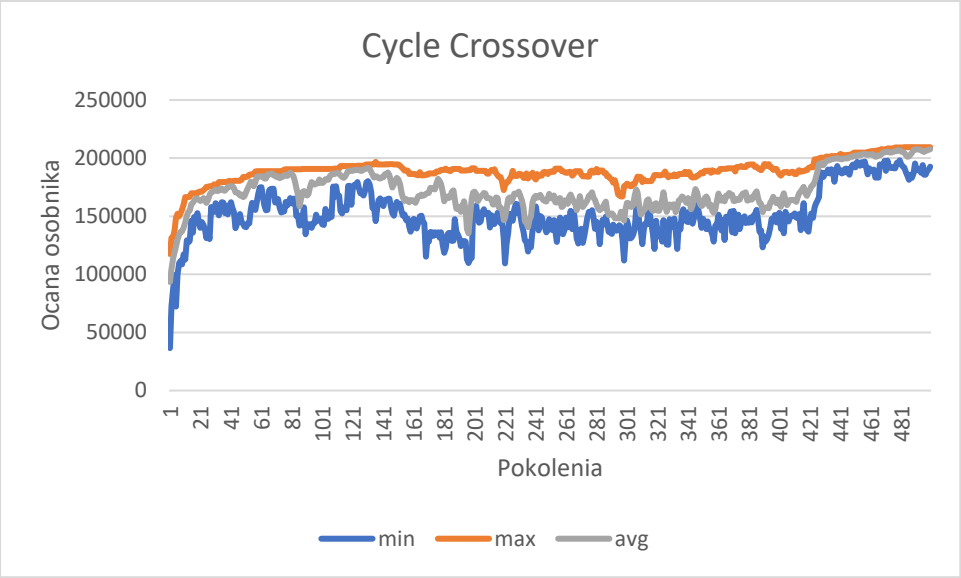
Na wykresie 14 widać, że turniej dla 1 osobnika nie różni się wiele od metody losowej.

Nie obserwowałem dużych różnic pomiędzy turniejami wielkości 2 3 7 20 52. Może jest coś złego w implementacji?

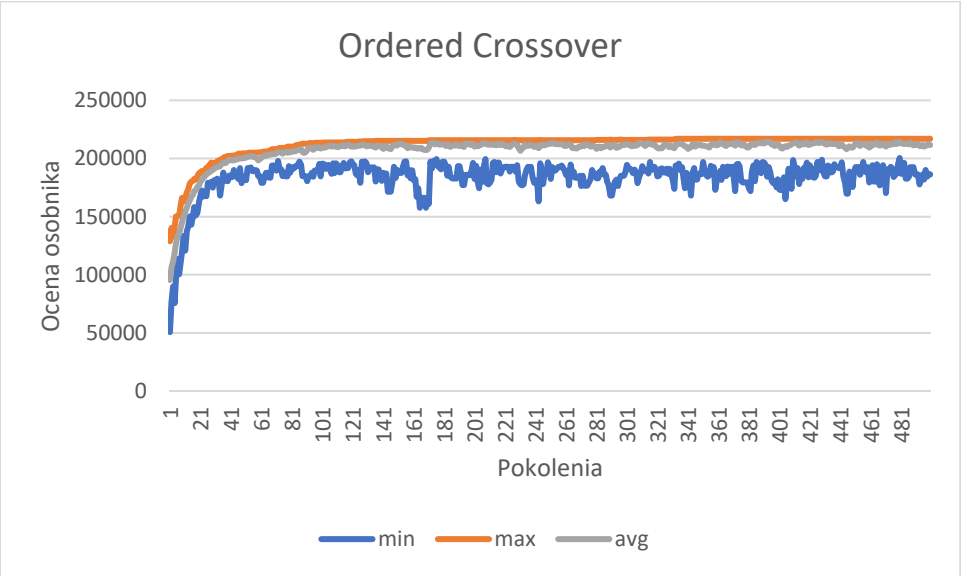
Teoretycznie przy turnieju 52 powinien zawsze być wybierany najlepszy osobnik. A populacja powinna się ujednolicić.

Porównanie metod krzyżowania

- Cel: Określenie wpływu metody krzyżowania na populację AG
- Stałe
    - Pokolenia: 500
    - Mutacja: Swap; intensywność = 0.15; prawdopodobieństwo = 0.2
    - Wielkość populacji: 100
    - Selekcja: Turniej(3/52);
- Zmienne:
    - Krzyżowanie: [Ordered, Cycle]; prawdopodobieństwo = 0.7



Wykr. 16



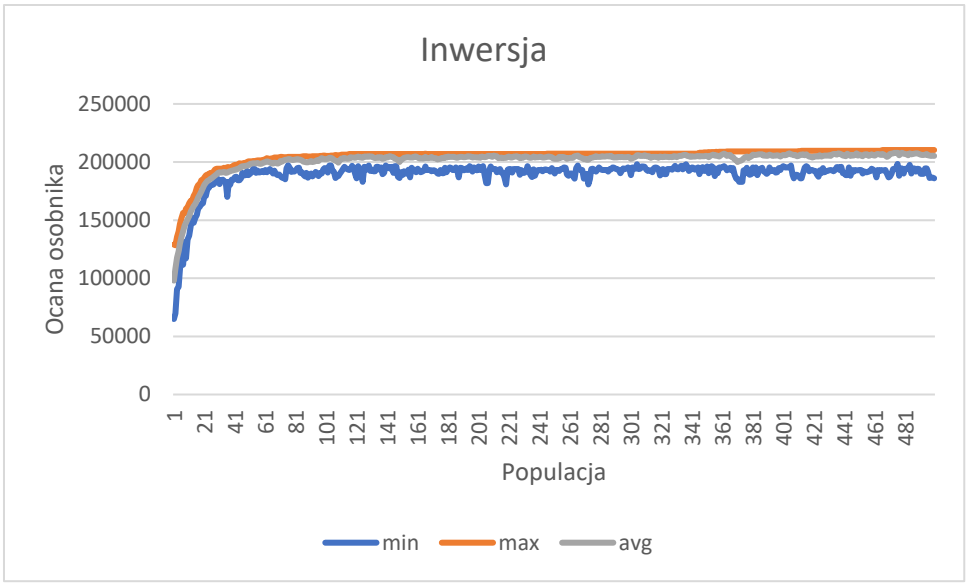
Wykr. 17

Obserwacje i wnioski:

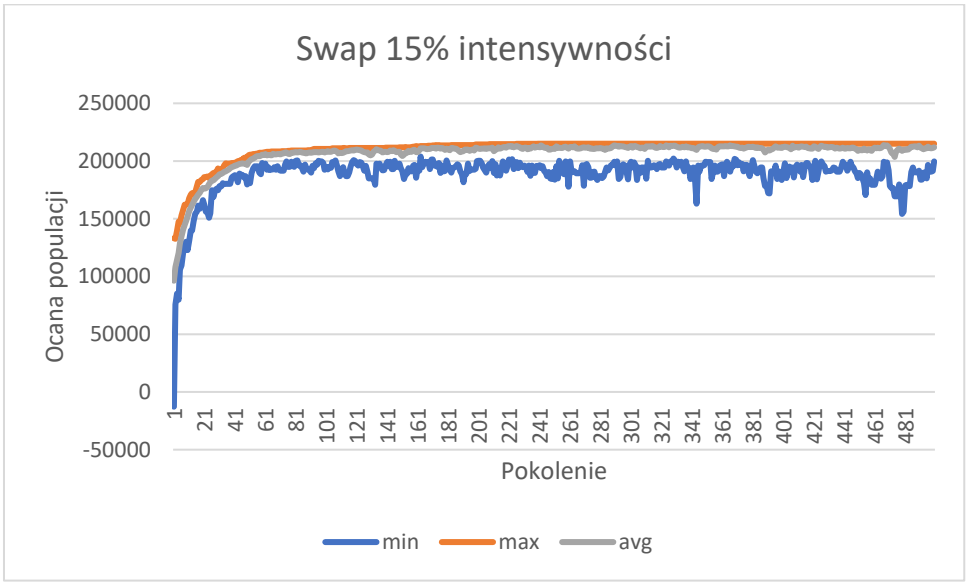
…

Porównanie metod mutacji

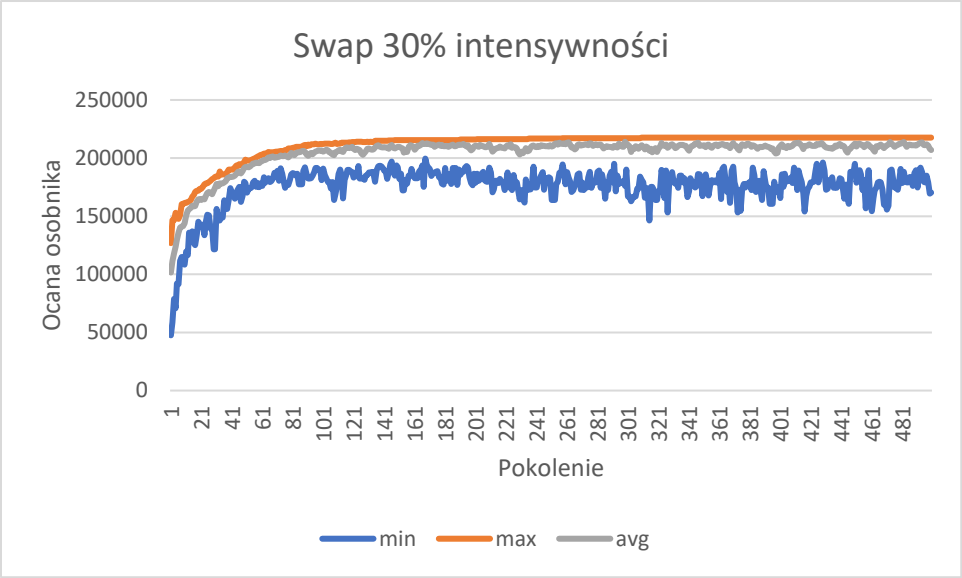
- Cel: Określenie wpływu metod mutacji na efektywność AG.
- Stałe
    - Pokolenia: 500
    - Wielkość populacji: 100
    - Selekcja: Turniej(3/52);
    - Krzyżowanie: [Ordered]; prawdopodobieństwo = 0.7
- Zmienne:
    - Mutacja:[Swap(0.15); Swap(30); Inverse]; prawdopodobieństwo = 0.2



Wykr. 18



Wykr. 19



Wykr. 20

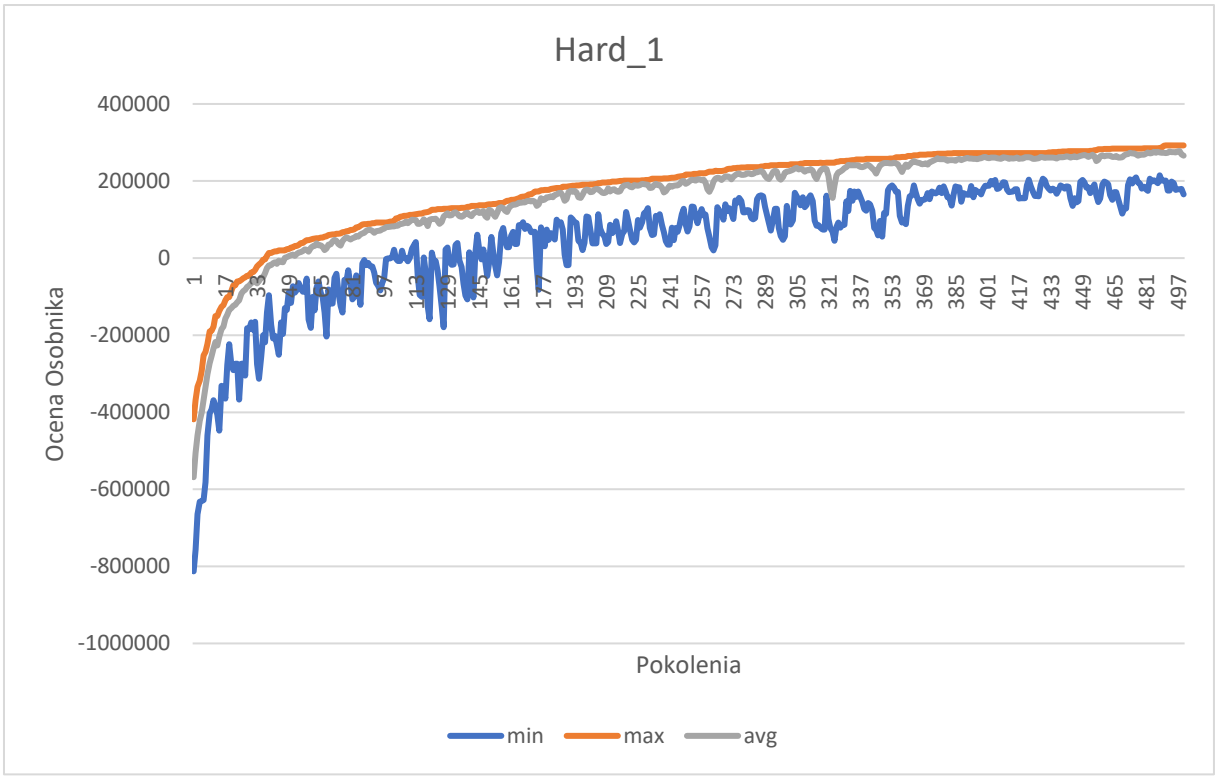
3. Porównanie AG, zachłannego i losowego

Instancja	Opt. wynik	Alg. Losowy[10k]				Alg. Zachłanny[N]				Alg. Ewolucyjny[10x]			
		best	worst	avg	std	best	worst	avg	std	best	worst	avg	std
Easy_3		-26266	-85347	-45105	8449	-36928	-51398	-43653	3510	14447	14447	14447	0
Easy_4		-23945	-74980	-39634	7123	-32480	-47833	-39137	4446	17602	17602	17602	0
Medium_3		173703	117485	155768	7599	158848	144252	151807	4457	214970	209481	212539	1727
Medium_4		225746	172537	210168	6352	211545	196919	204954	4189	268914	266543	268178	693
Hard_0		-502271	-1447402	-905929	143616	-193917	-2309747	-875340	718810	45601	-95843	5311	40651
Hard_1		-386036	-1097802	-685597	96394	-135734	-1666911	-486691	475441	429214	253052	330986	65243

Obserwacje:

Algorytm losowy radzi sobie całkiem nieźle w porównaniu do alg. Zachłannego w przypadkach easy. Mimo iż przestrzeń poszukiwań to aż 56! (ok. 7.1e+74), a więc całkiem spora różnica w stosunku do 1e+4. Spodziewałem się, że algorytm zachłanny już na tym etapie okaże się lepszy od losowego jednak ten zyskuję przewagę dopiero przy problemie hard gdzie przestrzeń rozwiązań to 439!.

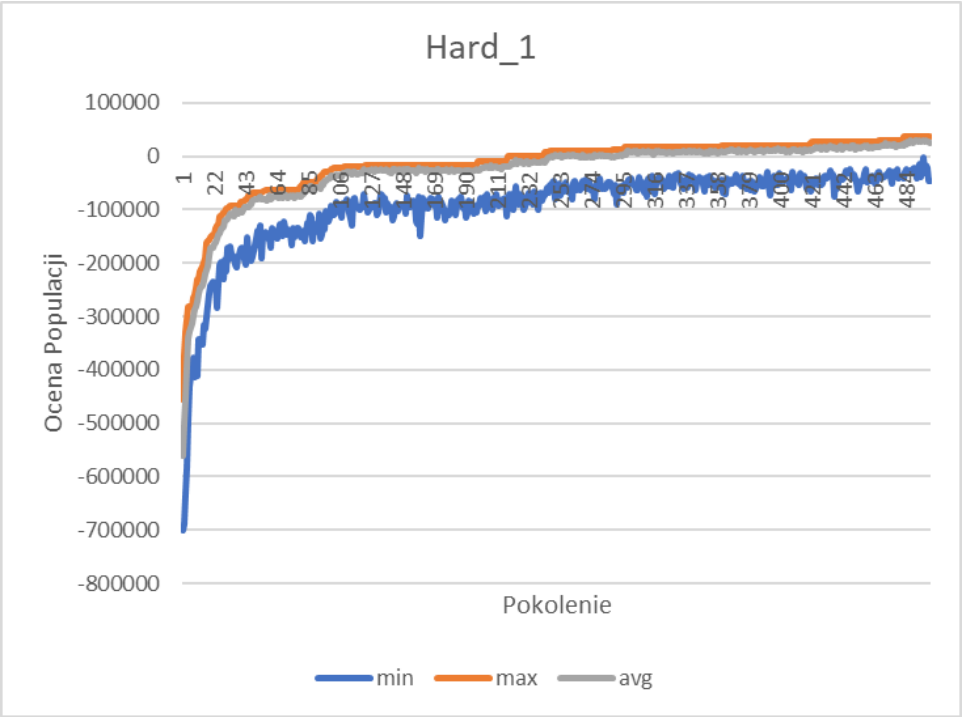
Żaden z pozostałych algorytmów nawet nie zbliża się do wyników AG. Co więcej wyniki AG charakteryzują się większą powtarzalnością dobrych wyników (niskie odchylnie standardowe).



Wykr. 21 – hard_1 przykładowy wykres dla danych z tabelki;

Ustawienia dla danych z tabelki

- Pokolenia: 500
- Wielkość populacji: 500
- Selekcja: Turniej(8) dla hard; Turniej(3) dla med. I easy;
- Krzyżowanie: [Ordered]; prawdopodobieństwo = 0.7
- Mutacja: swap; prawd=0.2; intensywność=0.3



Wykr. 23 – hard_1 przykładowy wykres dla danych z tabelki;

Ustawienia dla danych z tabelki

- Pokolenia: 500
- Wielkość populacji: 100
- Selekcja: Turniej(8) dla hard; Turniej(3) dla med. I easy;
- Krzyżowanie: [Ordered]; prawdopodobieństwo = 0.7
- Mutacja: swap; prawd=0.2; intensywność=0.15