

C1 - Algorytm poszukiwania harmonii

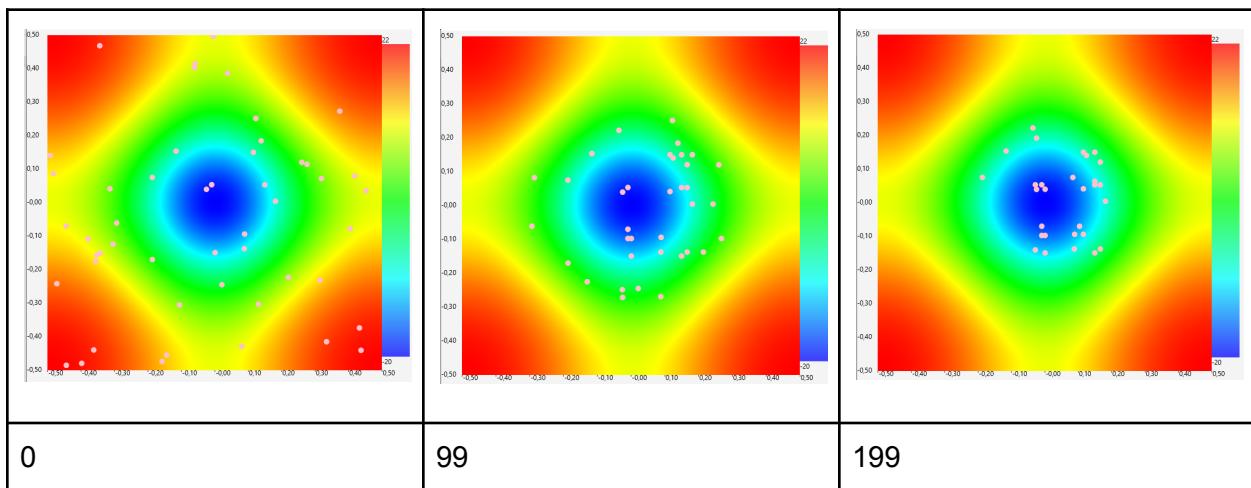
Autorzy: Michał Wójcik, Katarzyna Grygorowicz

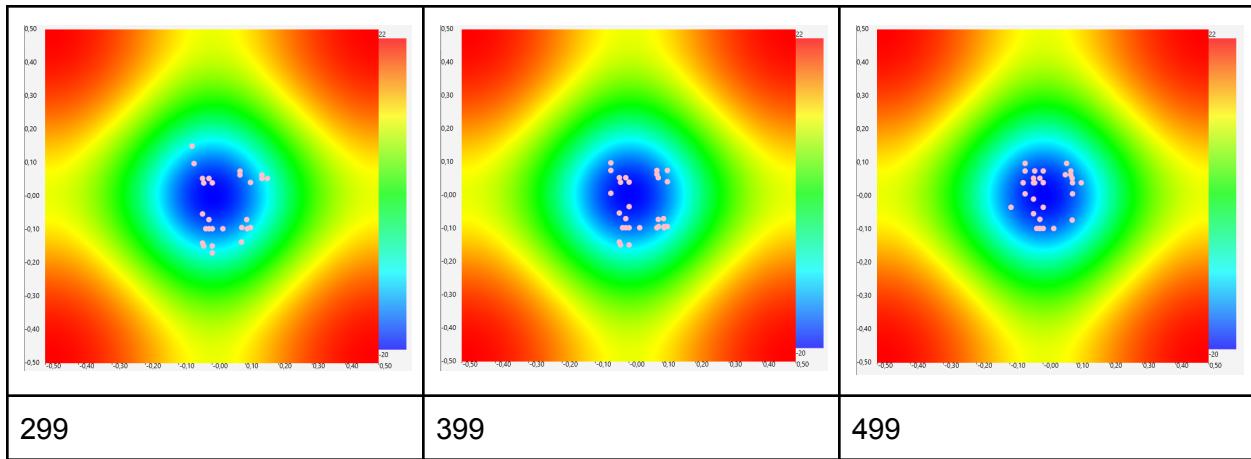
Funkcja Ackleya

To funkcja mogąca posiadać liczne minima lokalne, które mogą powodować "zablokowanie" algorytmu. Globalne optymalne minimum optimum wynosi $f(x,y) = 20.0$, dla $x = 0$ oraz $y = 0$

Dane domyślne

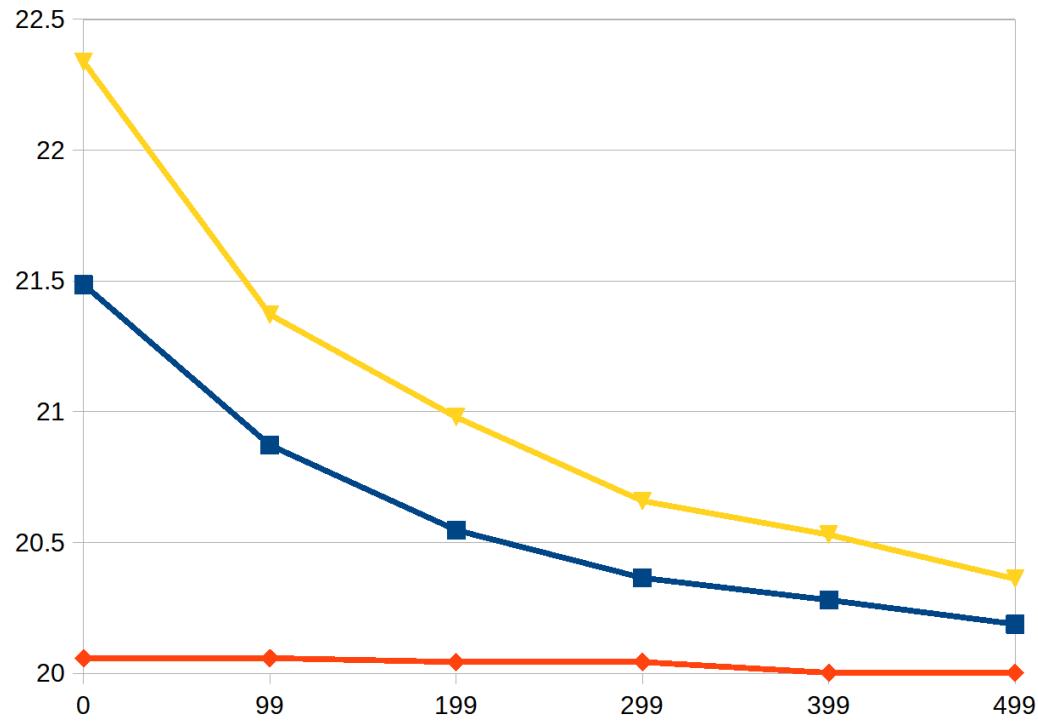
Zakres od	-0.5
Zakres do	0.5
Liczba improwizacji	500
Rozmiar pamięci	50
Wsp. odwołań do pamięci	0.95
Wsp. dostosowania	0.7
Promień dostosowania	1.0





iteracje	min	max	avg
0	20.0579799314041	22.337895916134	21.4864747848754
99	20.0579799314041	21.3715090702406	20.8721946224268
199	20.0430939420431	20.9790341745131	20.5473078005462
299	20.0430939420431	20.6587021130322	20.364769663599
399	20.0017552933429	20.5298669623608	20.2799172094332
499	20.0017552933429	20.3608117747748	20.1876631052095

Wykres harmonii

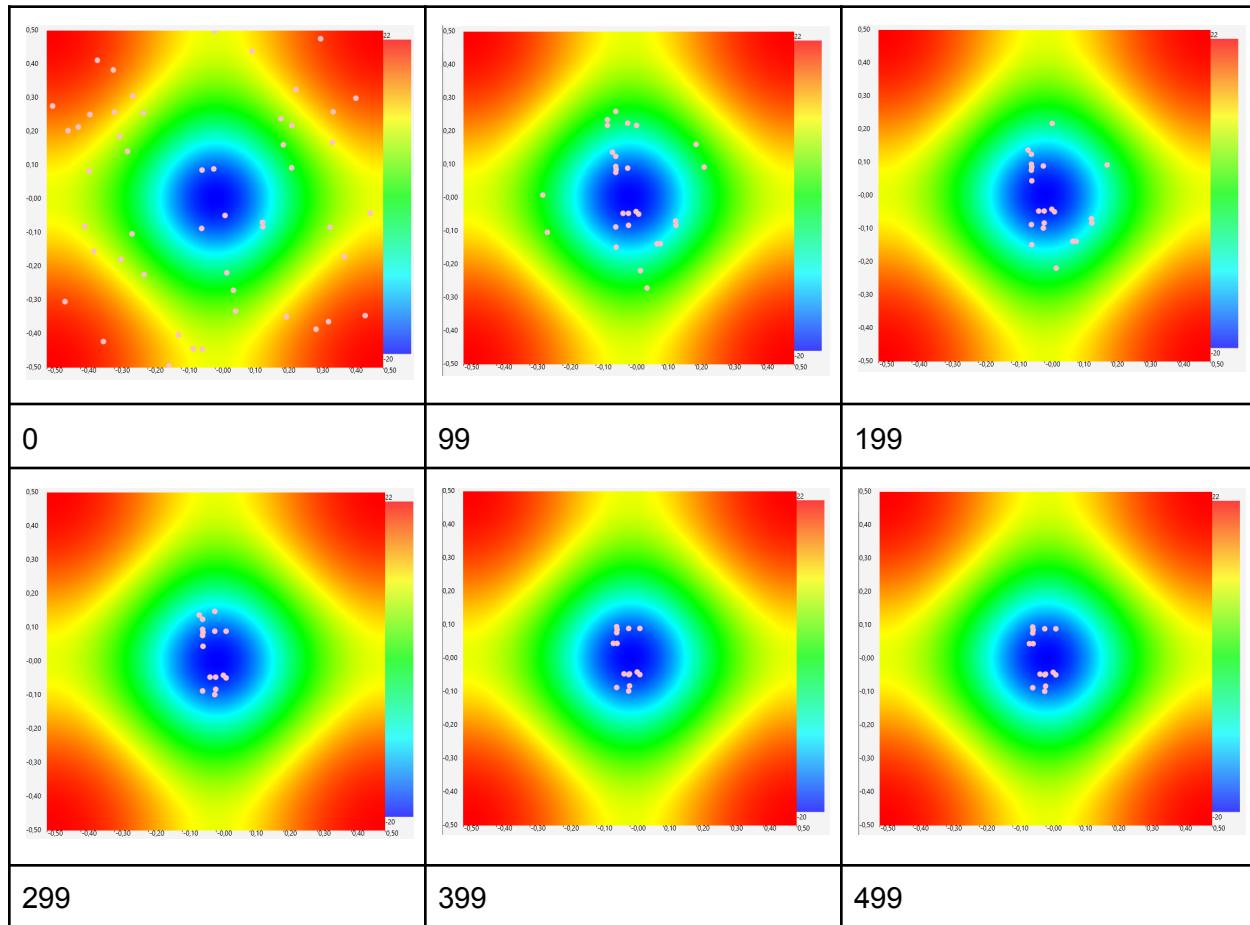


Obserwacje

- Funkcja wraz ze zwiększeniem liczby iteracji zmiany przybliżające nas do minimum globalnego (optimum) są coraz mniejsze.
- Po 300 iteracjach nie uzyskujemy dużo lepszych rezultatów.

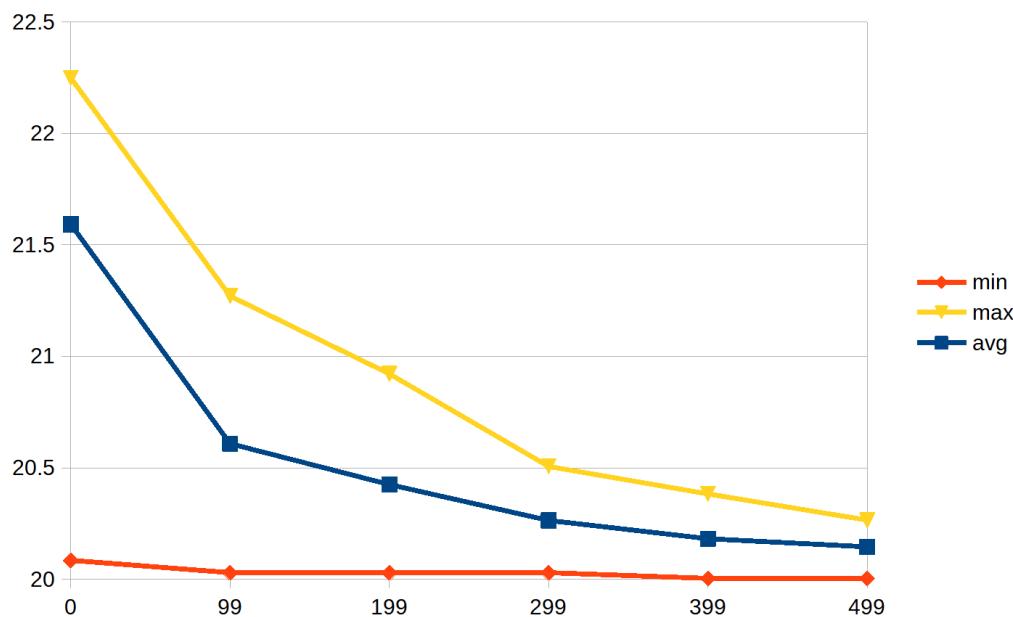
Większy promień dostosowania (3.0)

Zakres od	-0.5
Zakres do	0.5
Liczba improwizacji	500
Rozmiar pamięci	50
Wsp. odwołań do pamięci	0.95
Wsp. dostosowania	0.7
Promień dostosowania	3.0



iteracje	min	max	avg
0	20.0849980551047	22.2481240222088	21.5921541056292
99	20.0296318083064	21.2702673193402	20.6082282279171
199	20.0296318083064	20.9210801194567	20.4250493514449
299	20.0296318083064	20.5061771573855	20.2637849837187
399	20.0041227528074	20.3829892594294	20.1819506637658
499	20.0041227528074	20.2652911377262	20.1453679821029

Wykres harmonii



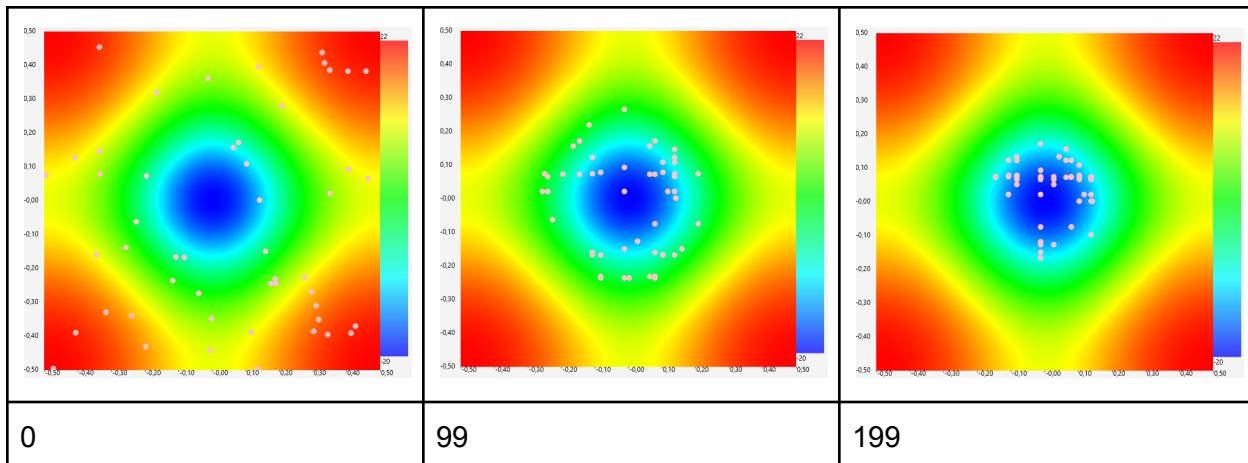
Obserwacje

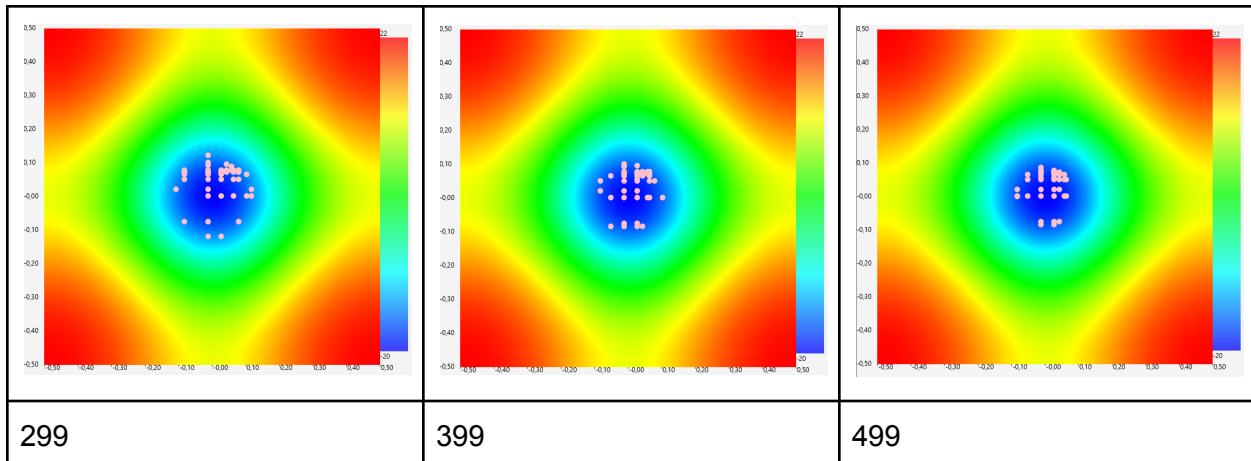
- Zwiększenie wartości parametru promienia spowodowało, że szybciej zbliżamy się do minimum globalnego w początkowych iteracjach. Szczególnie widać to w przedziale 0-99, gdzie widać gwałtowny spadek funkcji.
- Dla iteracji 299-499 otrzymujemy podobne wyniki jak dla wariantu z parametrami domyślnymi (po 300-tnej iteracji wykres spłaszcza się). Po około 300 iteracjach moglibyśmy zakończyć działanie algorytmu.

- Zwiększenie promienia spowodowało, że nowo losowane wartości przyjmowały częściej wartości skrajne badanego przedziału, co poprawiło zbieżność metody dla funkcji Ackleya. Jest to prawdopodobnie spowodowane tym, że dla badanego przedziału funkcja zbliżając się do minimum globalnego jest bardzo stroma, dzięki czemu wartości mogą szybciej “wpaść” do minimum globalnego dzięki mutacjom.

Mniejszy współczynnik dostosowania (0.1)

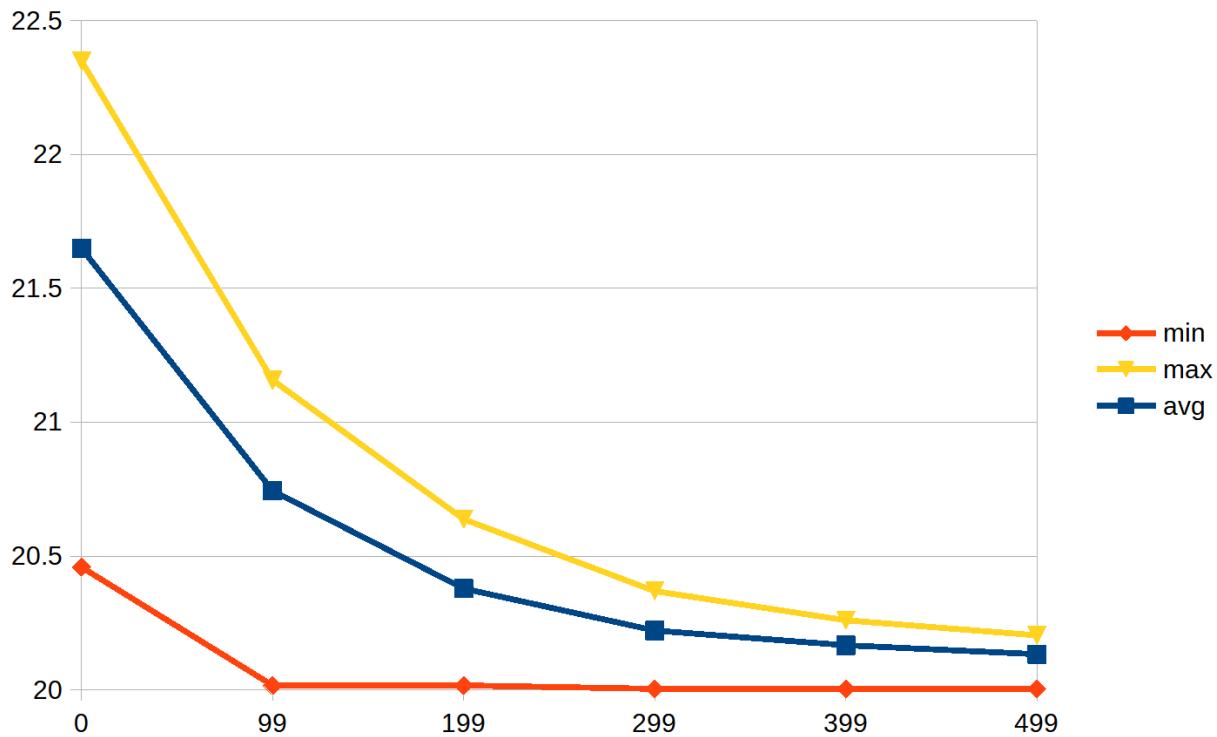
Zakres od	-0.5
Zakres do	0.5
Liczba improwizacji	500
Rozmiar pamięci	50
Wsp. odwołań do pamięci	0.95
Wsp. dostosowania	0.1
Promień dostosowania	1.0





iteracje	min	max	avg
0	20.4581985504872	22.3475875283238	21.6485478284531
99	20.0167987753721	21.1568947522859	20.7434381794185
199	20.0167987753721	20.6374424892845	20.3784154969552
299	20.0040093316254	20.368831398629	20.2216474561913
399	20.0040093316254	20.2600947186152	20.166748712254
499	20.0040093316254	20.2037146091046	20.1332971268903

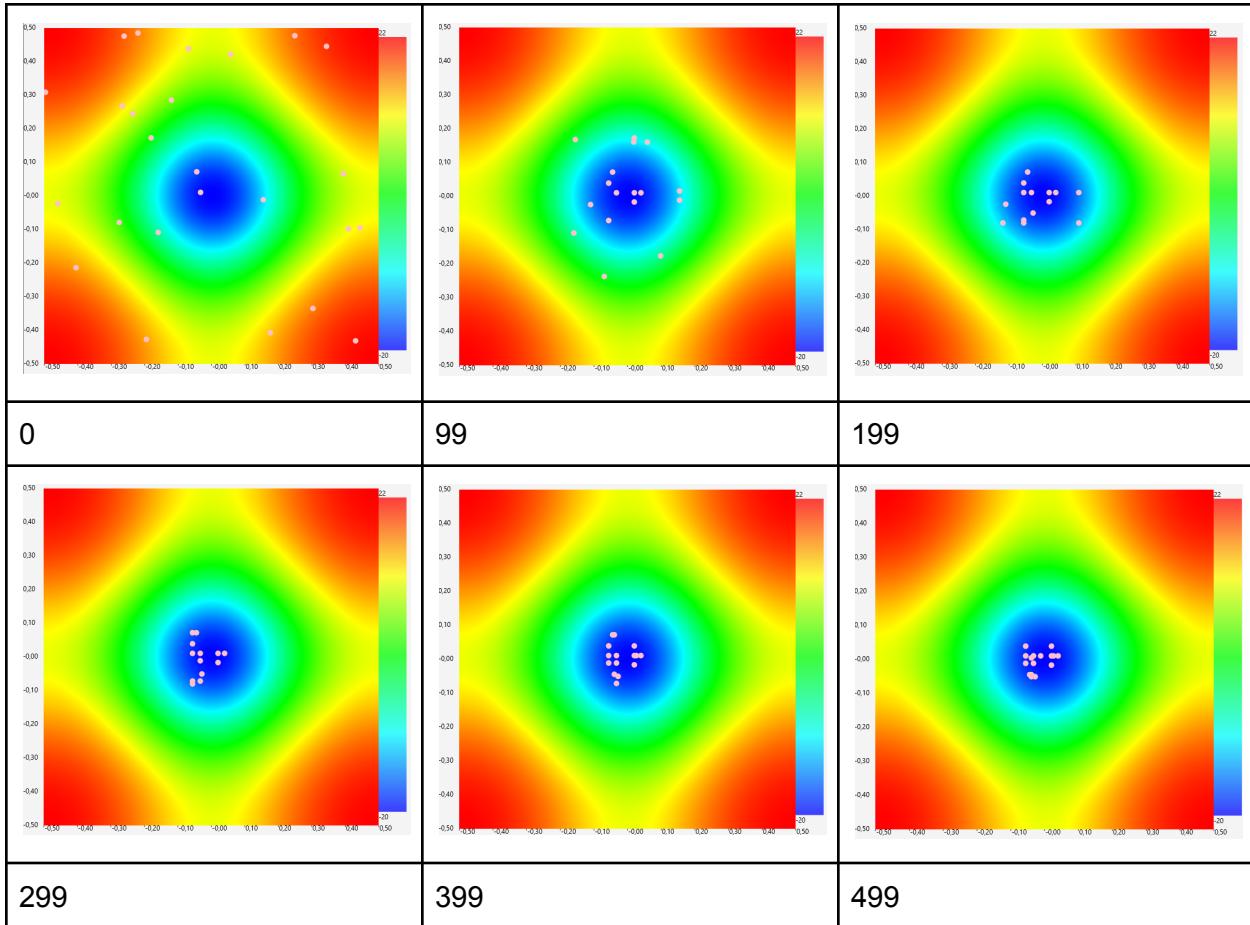
Wykres harmonii



Obserwacje

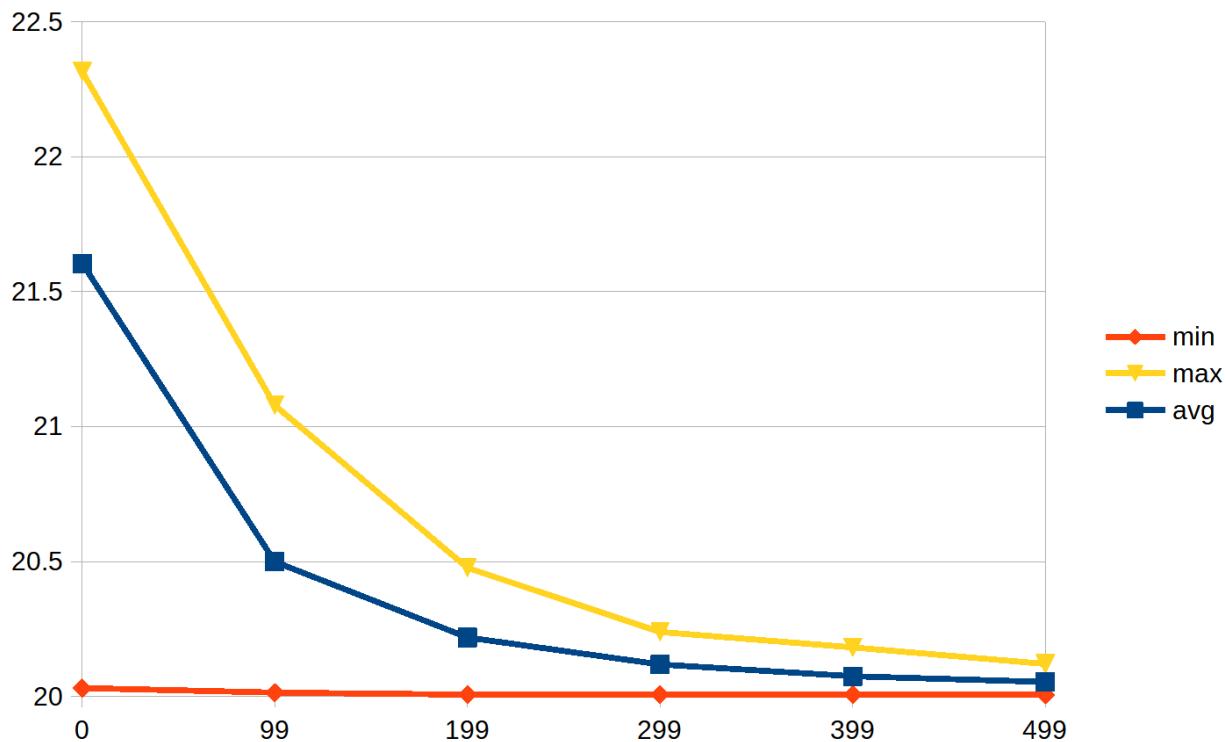
- Dla mniejszego współczynnika dostosowania uzyskaliśmy szybszą zbieżność niż w wariancie z parametrami domyślnymi, ale gorszą zbieżność niż w wariancie z większym promieniem.
- Współczynnik dostosowania określa prawdopodobieństwo mutacji pojedynczego tonu w harmonii. Ponieważ dla tego wariantu uzyskaliśmy lepsze wyniki niż dla wariantu domyślnego, podejrzewamy, że zmniejszenie prawdopodobieństwa mutacji tonu, a zatem ograniczenie wpływu losowości może poprawić wyniki działania algorytmu.

Mniejsza pamięć (25)



iteracje	min	max	avg
0	20.0310597062126	22.3148302334432	21.6046544338684
99	20.0143885347007	21.0784359859243	20.4990838782401
199	20.0066580437627	20.4765345975737	20.2186314536176
299	20.0066580437627	20.2389501955106	20.1185854439804
399	20.0066580437627	20.1817533668622	20.0747430125416
499	20.0065205034886	20.1204527141553	20.0538973572949

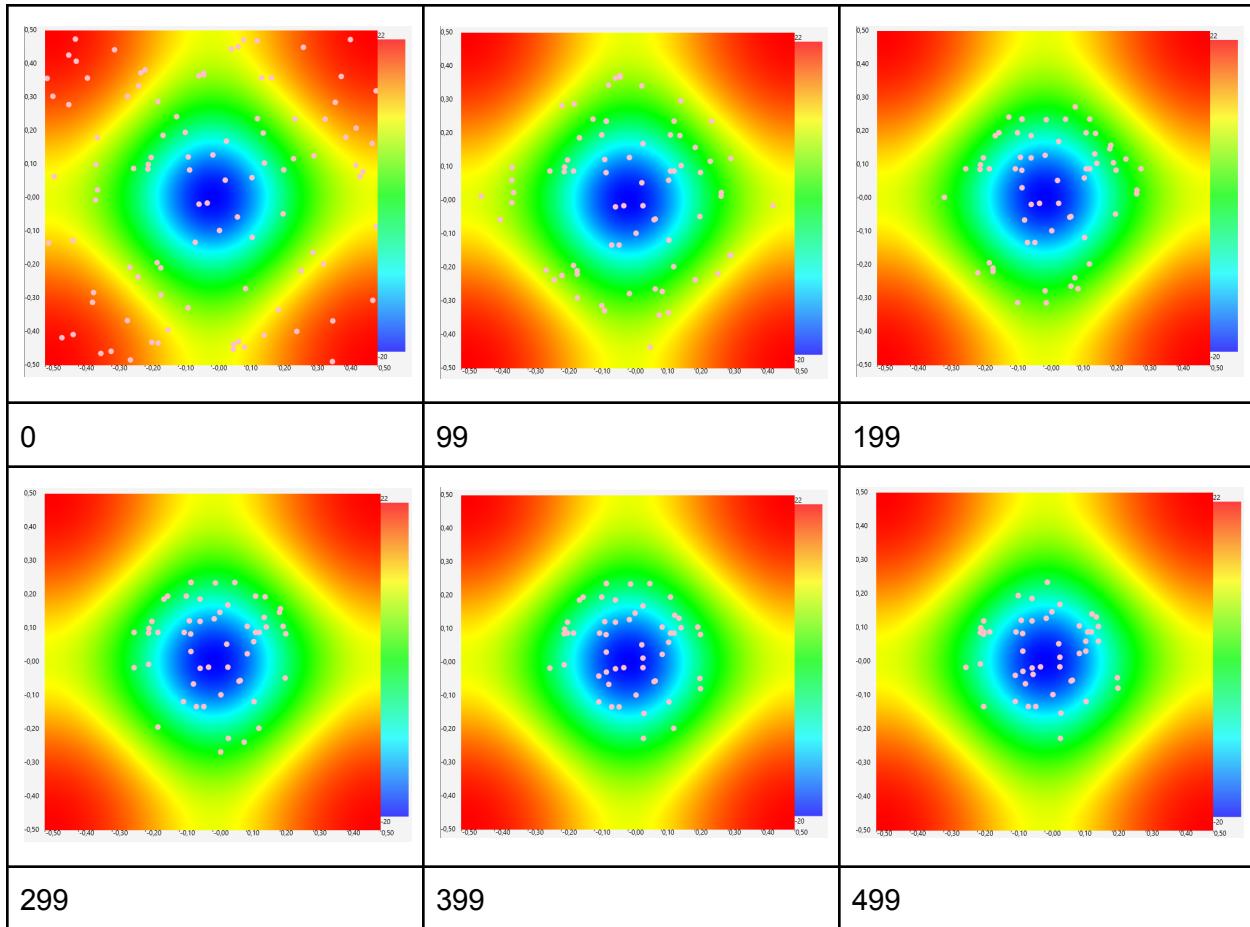
Wykres harmonii



Obserwacje

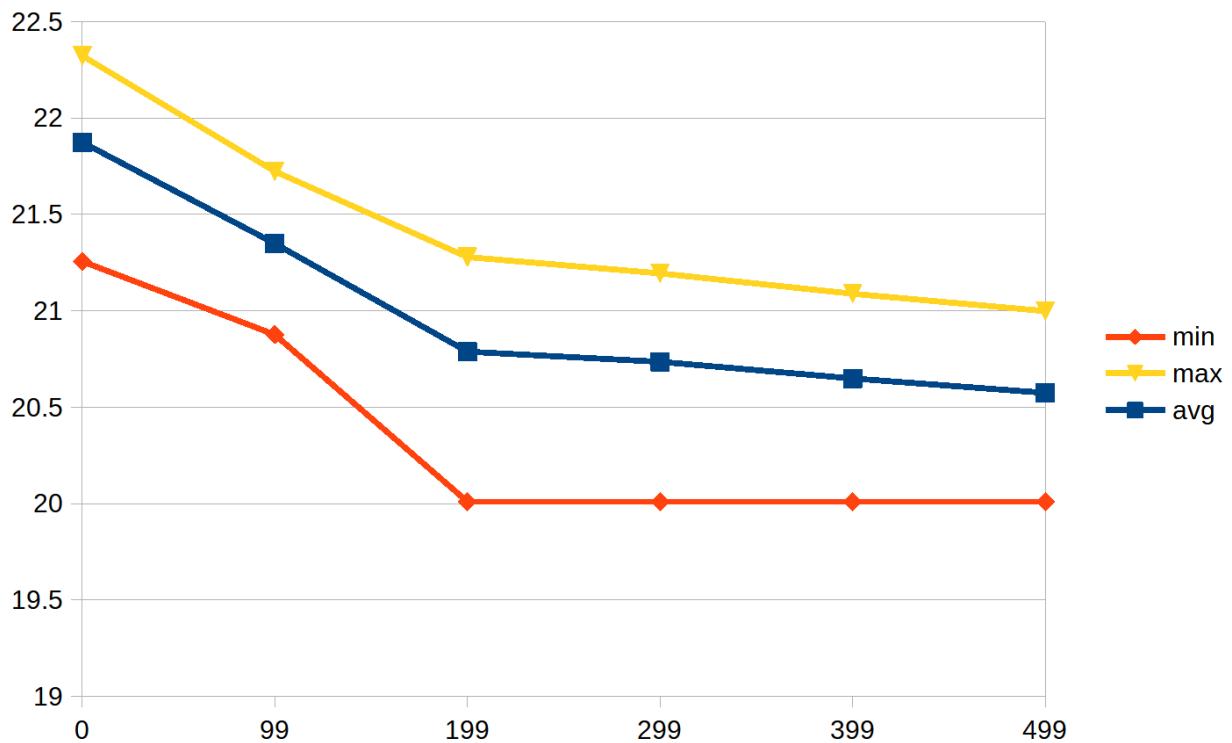
- Zmniejszenie rozmiaru pamięci dało lepsze rezultaty - funkcja jest szybciej zbieżna.
- Rozmiar pamięci definiuje ilość harmonii, które algorytm przechowuje i pamięta. Ponieważ zmniejszyliśmy ilość pamięci, ograniczyliśmy liczbę harmonii w pamięci o połowę, co spowodowało, że w pamięci szybciej pojawiały się harmonie o lepszych brzmieniach (ponieważ najgorzej brzmiąca harmonia w pamięci jest zastępowana nową, jeśli brzmi lepiej od tej najgorszej), co dało zauważalnie szybszą zbieżność do optymalnego rozwiązania.

Większa pamięć (100)



iteracje	min	max	avg
0	20.0310597062126	22.3148302334432	21.6046544338684
99	20.0143885347007	21.0784359859243	20.4990838782401
199	20.0066580437627	20.4765345975737	20.2186314536176
299	20.0066580437627	20.2389501955106	20.1185854439804
399	20.0066580437627	20.1817533668622	20.0747430125416
499	20.0065205034886	20.1204527141553	20.0538973572949

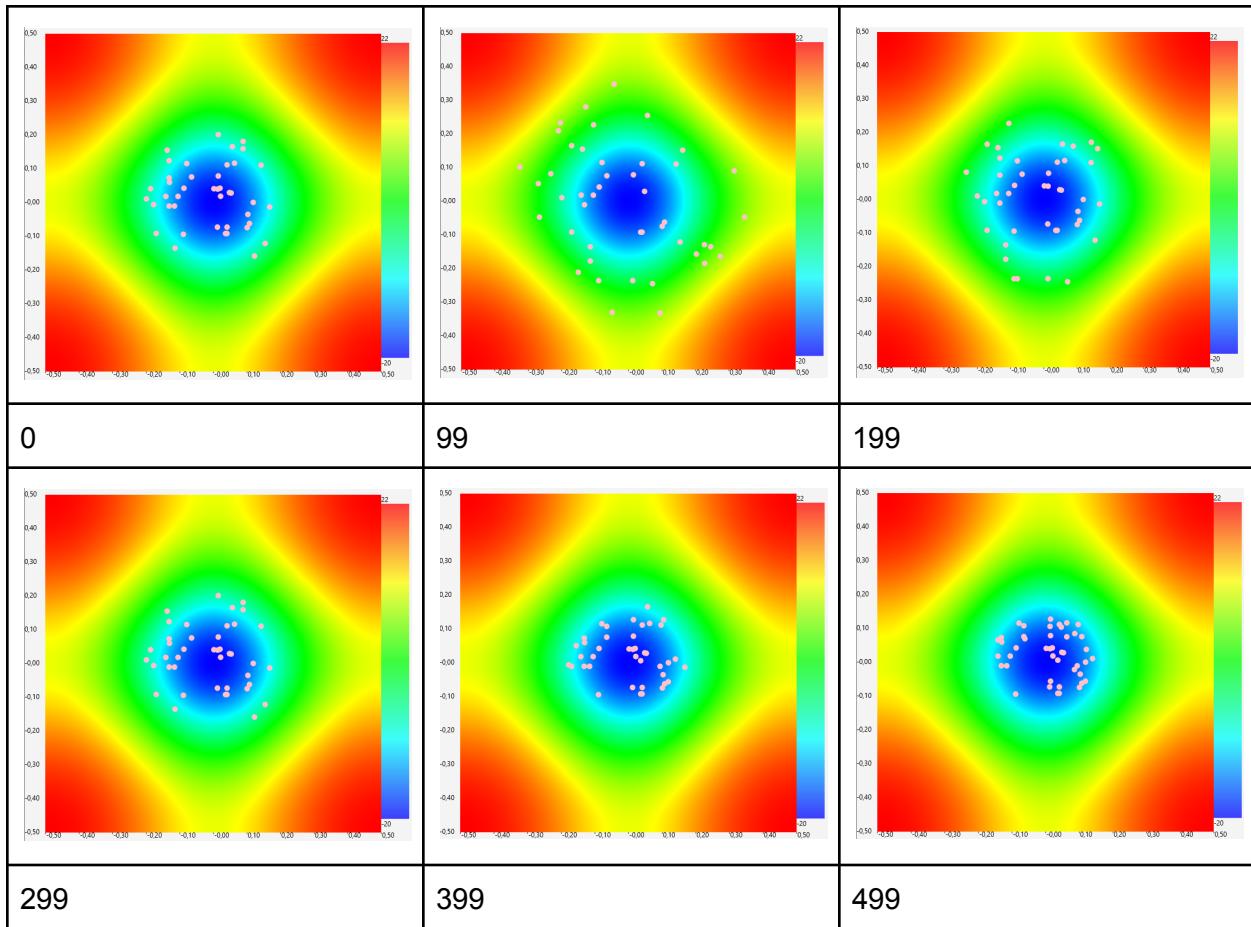
Wykres harmonii



Obserwacje

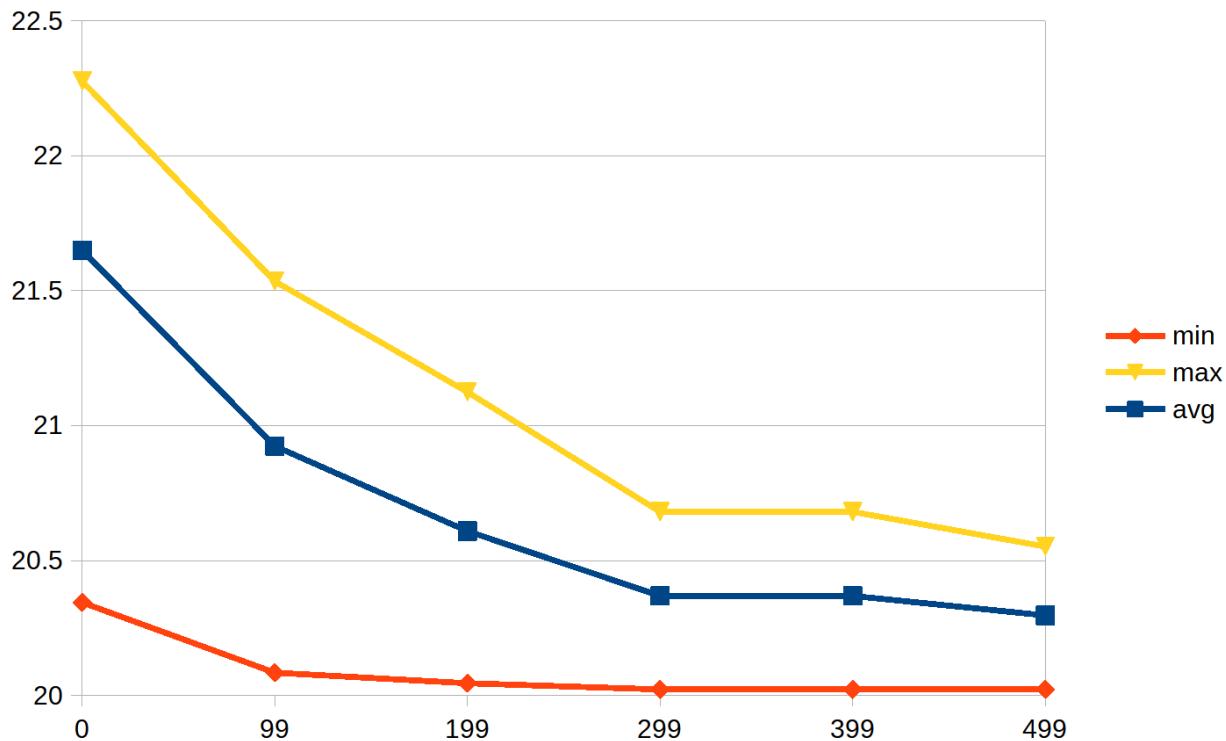
- Zwiększenie pamięci dało nam odwrotne rezultaty niż jej zmniejszenie
- Zwiększając ilość pamięci zwiększyliśmy liczbę harmonii dwukrotnie. W pamięci było w związku z tym więcej miejsca na harmonie gorzej brzmiące, co spowodowało wolniejszą zbieżność do optymalnego rozwiązania.

Wsp.odw do pamięci (0.4)



iteracje	min	max	avg
0	20.3444390807203	22.2752327153483	21.6487454624024
99	20.0843988292094	21.5343343969211	20.9237177817232
199	20.0451437696665	21.1239583438513	20.6088140449985
299	20.0223881950194	20.6804696342889	20.3687394847926
399	20.0223881950194	20.6804696342889	20.3687394847926
499	20.0223881950194	20.5515007692081	20.2964856197876

Wykres harmonii



Obserwacje

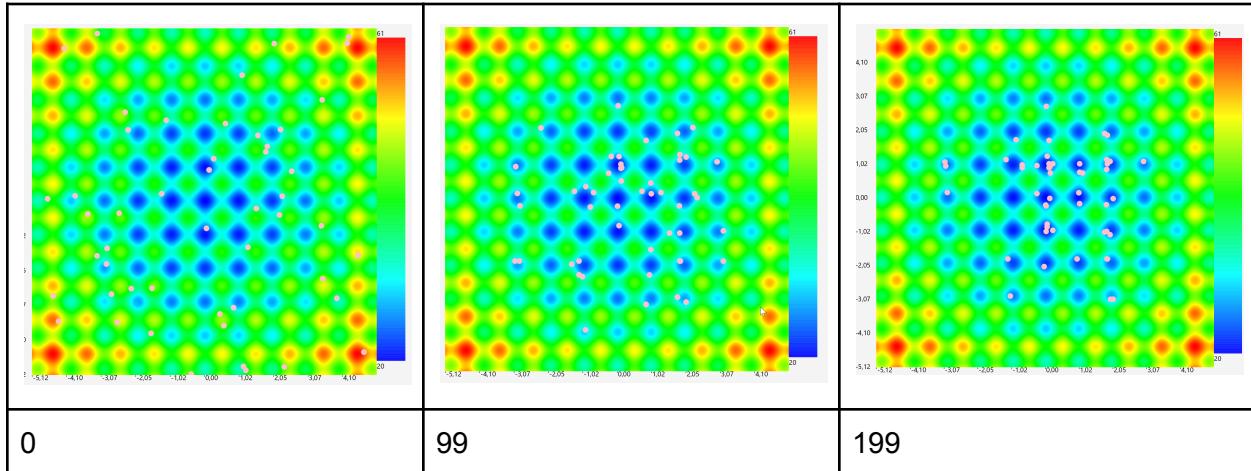
- Zmniejszając czynnik odwołania do pamięci, zwiększyliśmy szansę na tworzenie nowych rozwiązań. Ponieważ współczynnik ten wynosi 0.4, prawdopodobieństwo stworzenie losowego rozwiązania (harmonii) wynosi 60%. Jest to znacznie więcej niż domyślna wartość 5%.
- Znaczne zwiększenie wpływu losowości na działanie algorytmu obniżyło jego efektywność - algorytm jest wolniej zbieżny. Wariant ten potwierdza wnioski i obserwacje opisane w wariantie, gdzie zastosowaliśmy mniejszy współczynnik dostosowania.
- W szczególności wolniejszą zbieżność algorytmu widać w iteracjach 299-399, gdzie wartość funkcji w ogóle nie zbliżała się do minimum globalnego - podejrzewamy, że funkcja mogła "utknąć" w jednym z wielu minimum lokalnym algorytmu.

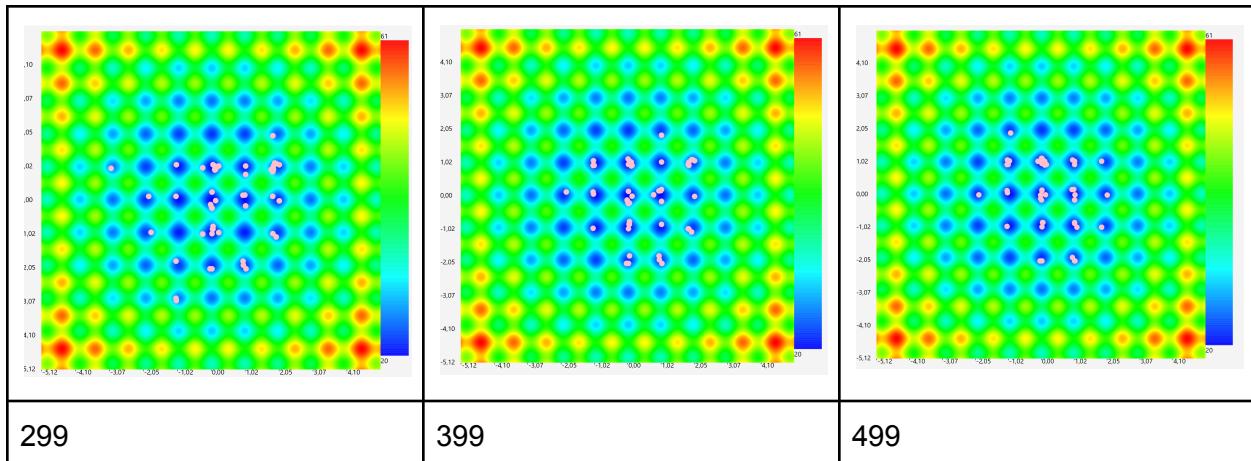
Funkcja Rastragina

Funkcja Rastragin charakteryzuje się obecnością wielu minimów lokalnych przy jednoczesnym istnieniu tylko 1 ekstremum globalnego. Wartości tych minimów lokalnych zwiększały się wraz z oddaleniem od minimum globalnego. Minimum globalne poszukiwane jest w przedziale $x \in [-5.12; 5.12]$ i wynosi $f(x,y) = -20.0$, dla $x = 0$ oraz $y = 0$

Parametry domyślne

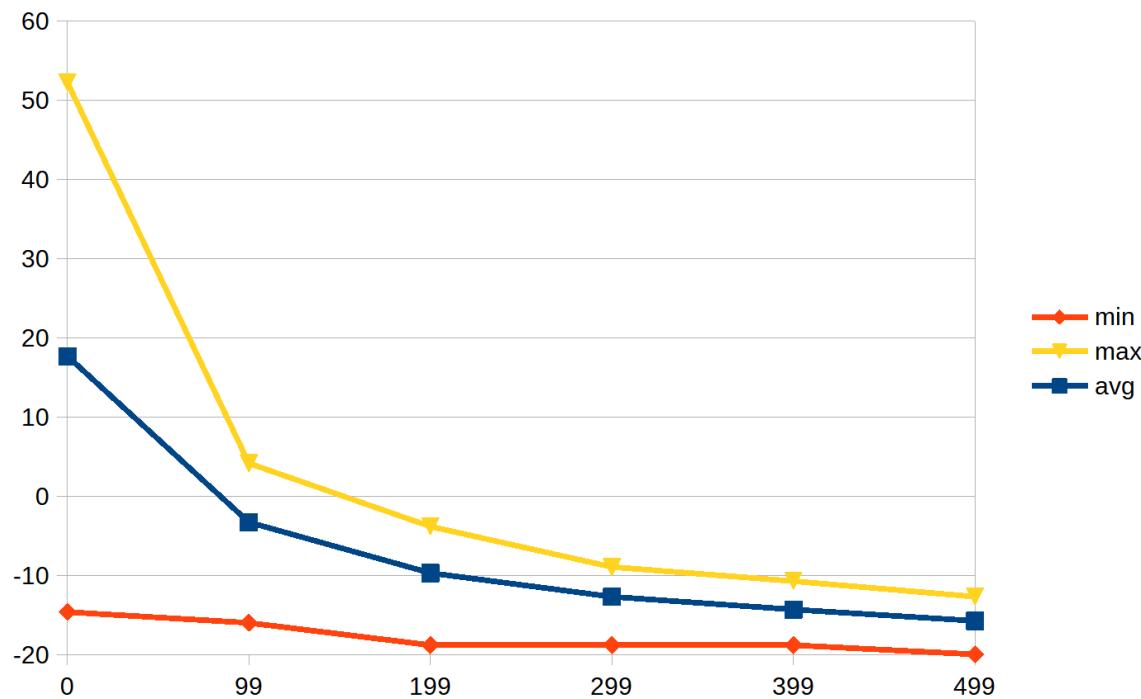
Zakres od	-5.12
Zakres do	5.12
Liczba improwizacji	500
Rozmiar pamięci	50
Wsp. odwołań do pamięci	0.95
Wsp. dostosowania	0.7
Promień dostosowania	1.0





iteracje	min	max	avg
0	-14.592372975698	52.2297998312257	17.6972873629227
99	-15.9579704016236	4.1707308104129	-3.26893324380763
199	-18.7746627550888	-3.79741399846434	-9.66836575285604
299	-18.7746627550888	-8.90636108984922	-12.6596057247769
399	-18.7746627550888	-10.6959985604133	-14.2740276706661
499	-19.9350204002973	-12.6573014068208	-15.7083754236603

Wykres harmonii



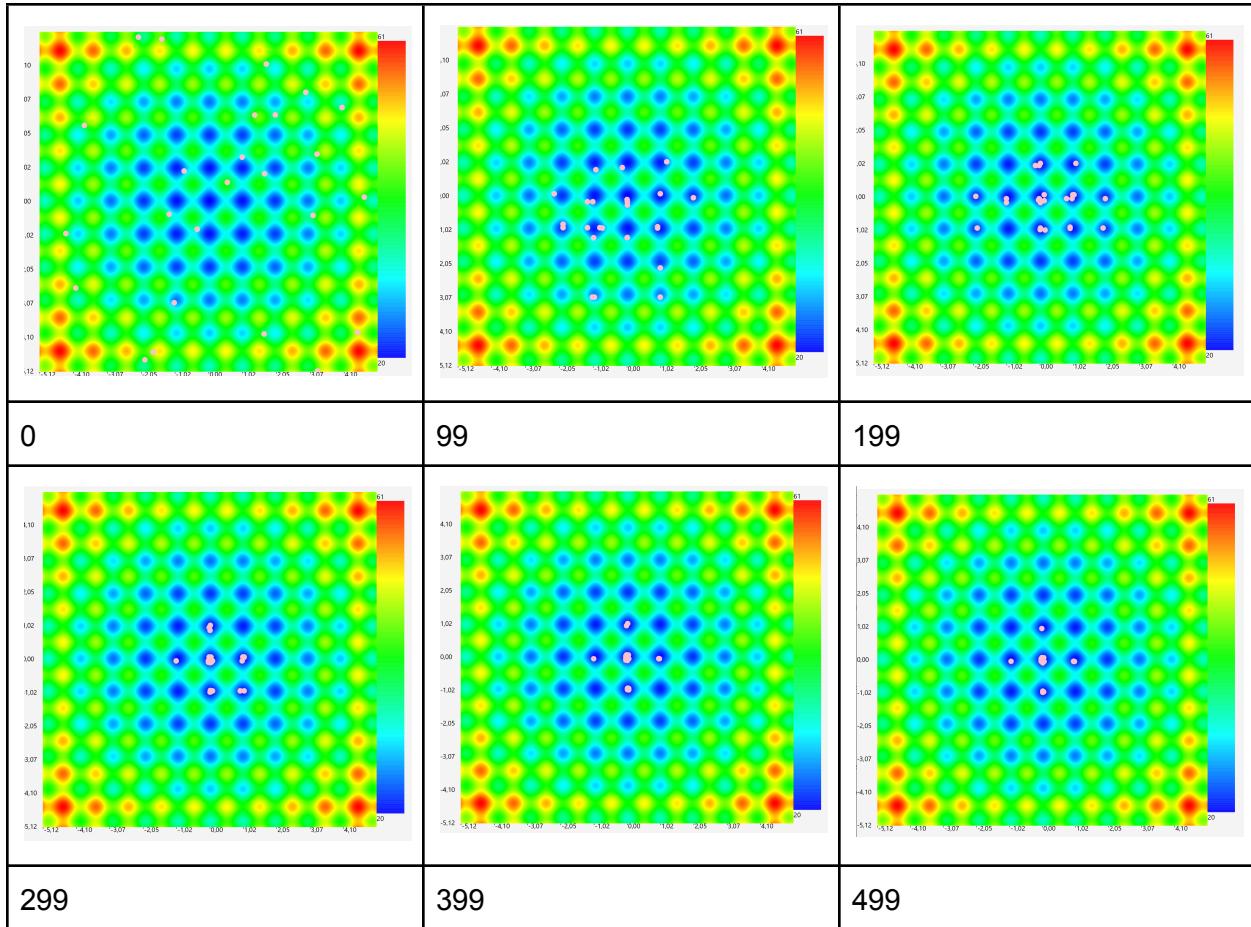
Obserwacje

- Podobnie jak w przypadku funkcji Ackleya, wraz ze zwiększaniem liczby iteracji zmiany przybliżające nas do minimum globalnego (optimum) są coraz mniejsze (tj. rozmiar kroku jest coraz mniejszy)
- Mniej więcej po około 300 iteracjach nie uzyskujemy dużo lepszych rezultatów.
- W przeciwieństwie do funkcji Ackleya, wartości w większym stopniu odbiegają od wartości minimum globalnego. Dla funkcji Ackleya różnice między wartością średnią a minimum globalnym były rzędu wartości dziesiętnej, tutaj są to wielkości dziesiętne lub jedności. Jest to najprawdopodobniej spowodowane większym zakresem, w którym szukamy globalnego minimum oraz obecności wielu minimów lokalnych przy jednoczesnym istnieniu tylko jednego ekstremum globalnego

Mniejsza pamięć (25)

Zakres od	-5.12
Zakres do	5.12
Liczba improwizacji	500

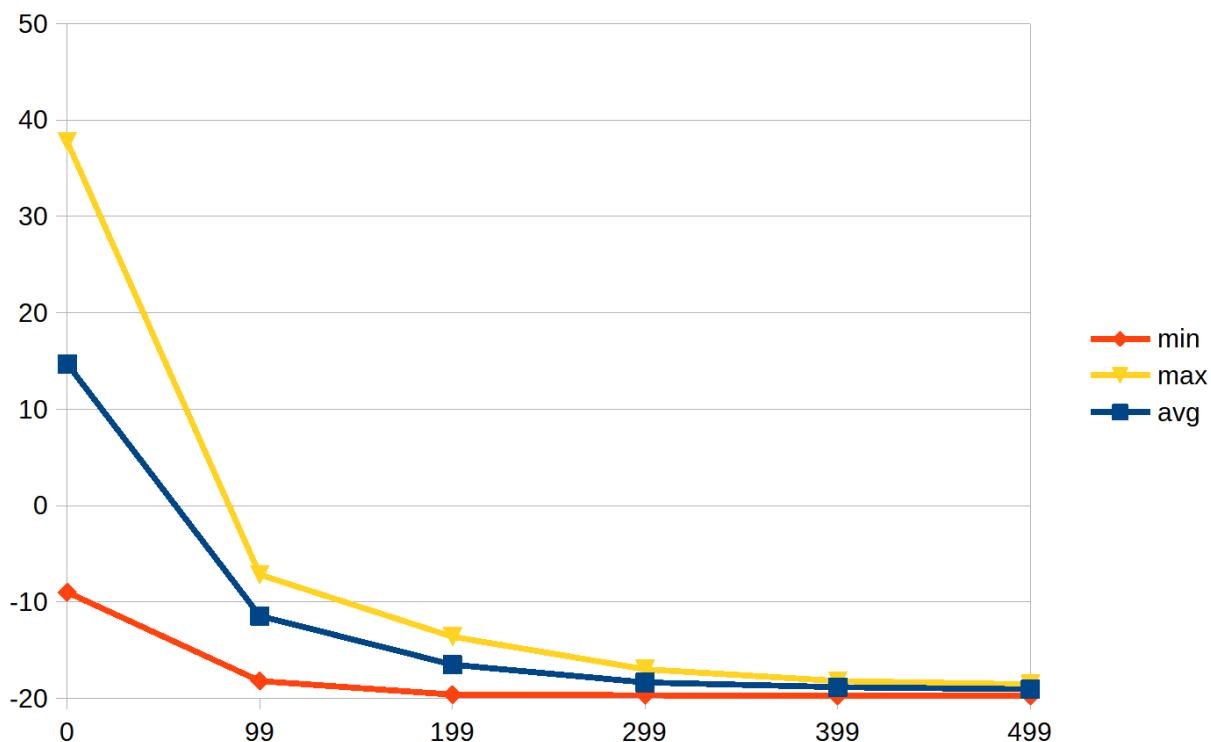
Rozmiar pamięci	25
Wsp. odwołań do pamięci	0.95
Wsp. dostosowania	0.7
Promień dostosowania	1.0



iteracje	min	max	avg
0	-8.98254688715032	37.7450656517378	14.7051464024129
99	-18.1943107200651	-7.152957149109	-11.4420056605038
199	-19.6014690801273	-13.591448009151	-16.4942092288739
299	-19.6714868952426	-16.9717091757948	-18.3341755260783

399	-19.7349283542126	-18.1943107200651	-18.8267885996401
499	-19.7349283542126	-18.54225298457	-19.0519459525293

Wykres harmonii

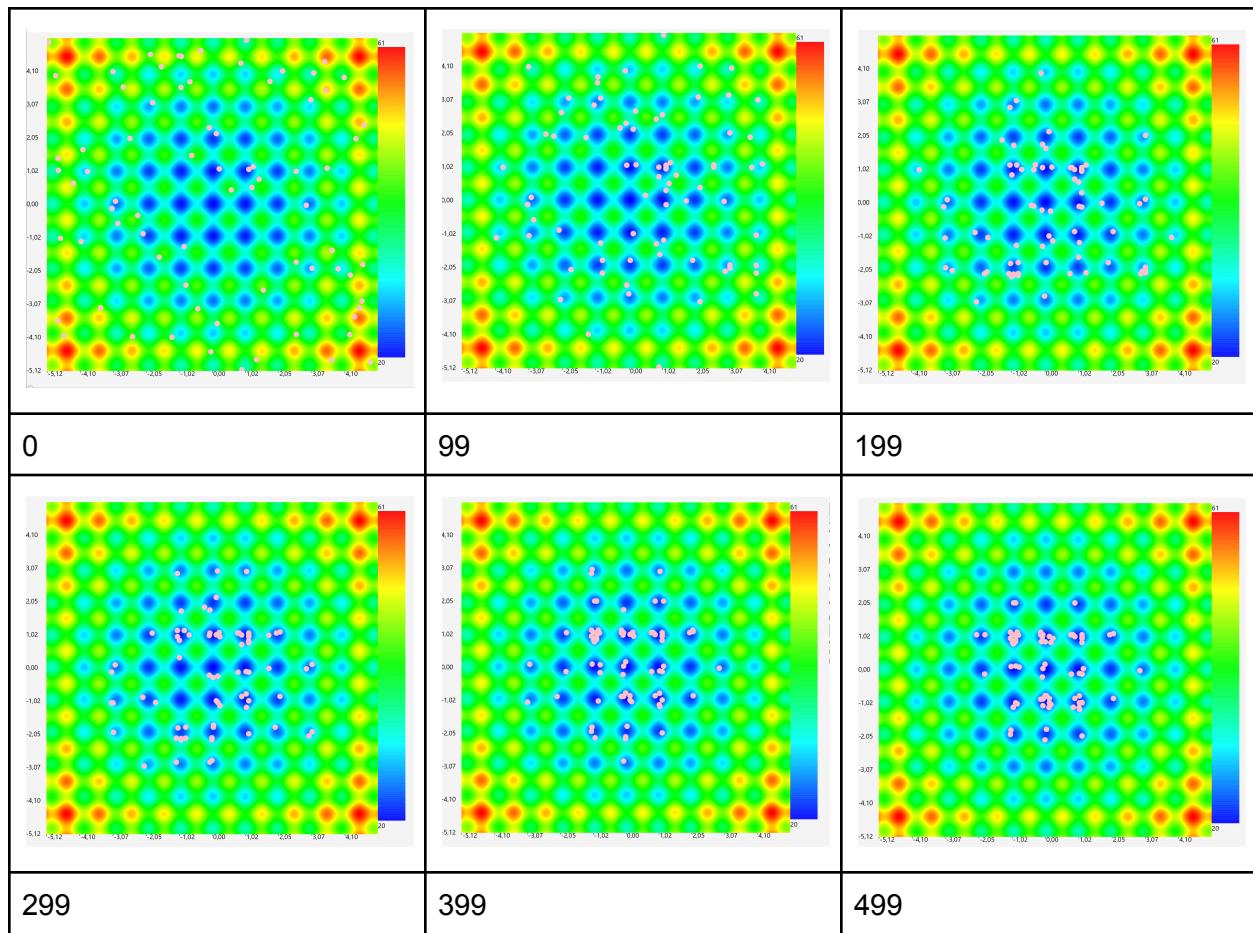


Obserwacje

- Zmniejszenie pamięci znacząco poprawiło rezultaty działania programu - większość próbek dla liczby iteracji większej niż 300 przyjmuje wartość około -20, czyli szukanego optimum. Po 300 iteracjach moglibyśmy zakończyć działanie algorytmu i otrzymać satysfakcyjujący rezultat.
- Wartości średnie, minimalne i maksymalne w mniejszym stopniu odbiegają od siebie, porównując wykres z poprzednimi wariantami.
- Spadek dla iteracji 0-99 jest o wiele bardziej stromy niż dla wariantu z domyślnymi parametrami.

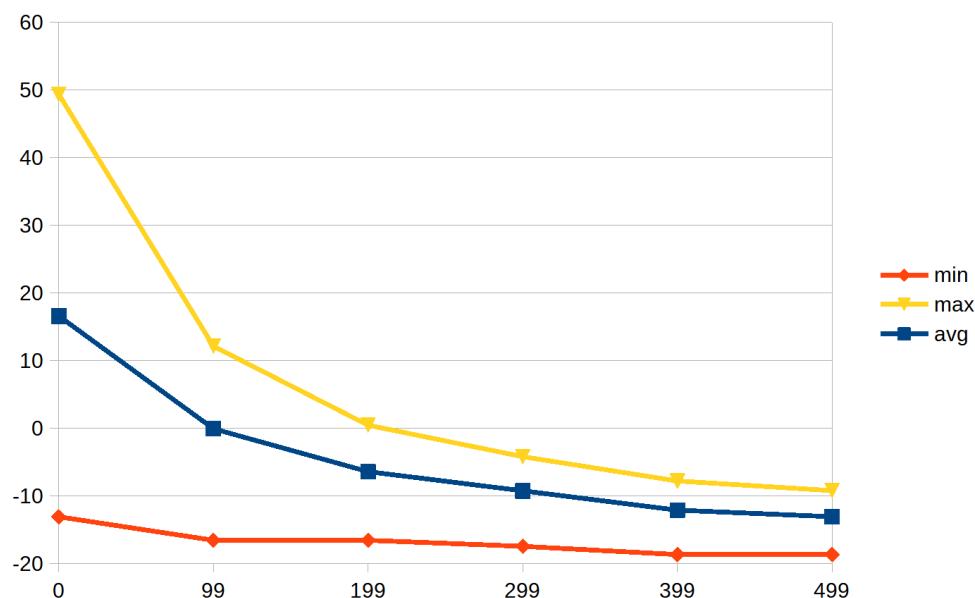
Większa pamięć (75)

Zakres od	-5.12
Zakres do	5.12
Liczba improwizacji	500
Rozmiar pamięci	75
Wsp. odwołań do pamięci	0.95
Wsp. dostosowania	0.7
Promień dostosowania	1.0



iteracje	min	max	avg
0	-13.0758070657247	49.3365426384582	16.5847122037197
99	-16.5518904825443	12.1192692739906	-0.0647600657465731
199	-16.5518904825443	0.451907078955914	-6.41085435949694
299	-17.4482194393933	-4.19506539293147	-9.24358690755917
399	-18.6602468703108	-7.79114064697768	-12.0917647755053
499	-18.6602468703108	-9.22015321094263	-13.070450661838

Wykres harmonii

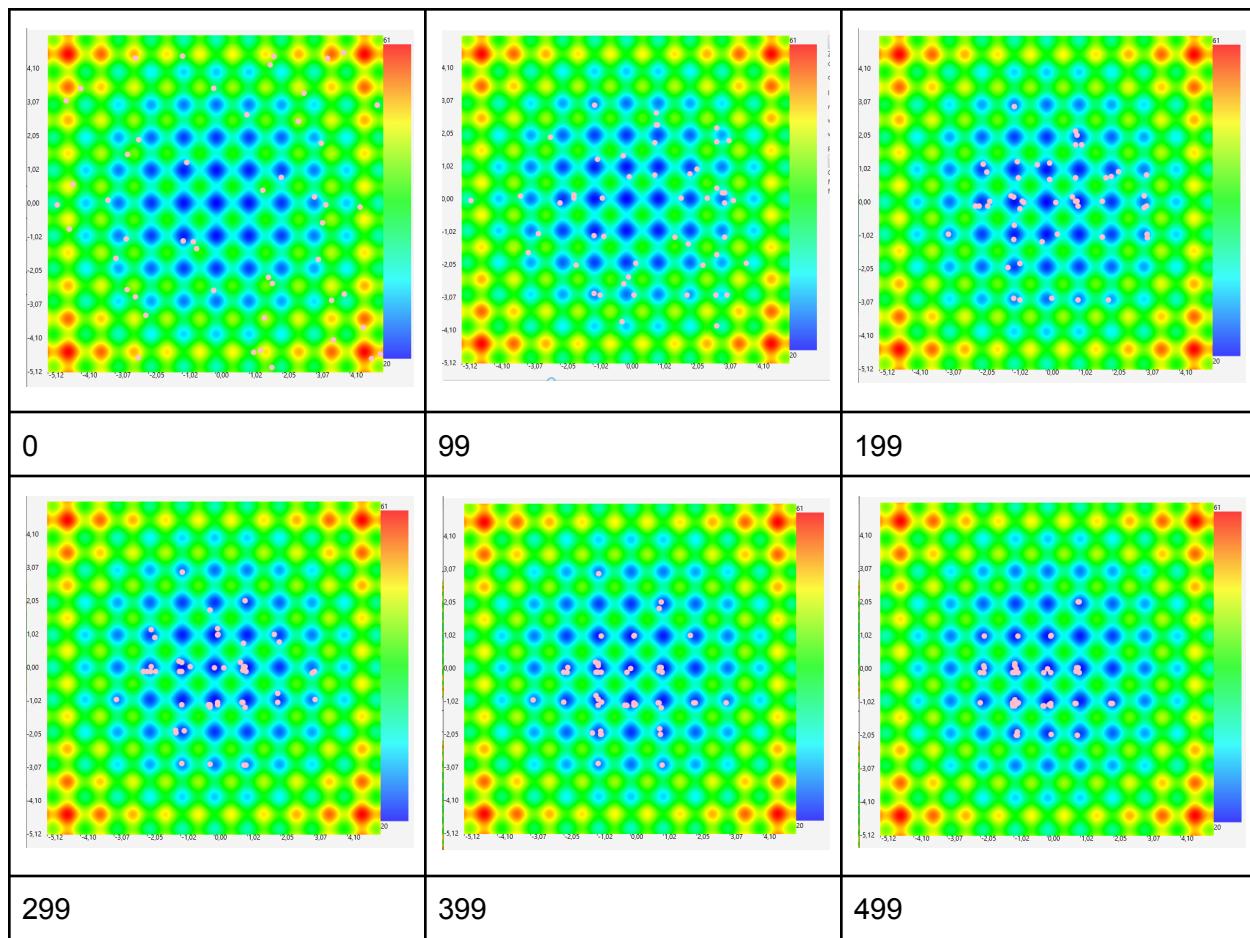


Obserwacje

- Przy lekkim zwiększeniu pamięci o 25 (z 50 na 75) otrzymaliśmy gorsze rezultaty. Taką samą zależność zaobserwowaliśmy dla funkcji Ackleya, gdzie zwiększenie pamięci również skutkowało gorszą zbieżnością do minimum globalnego.

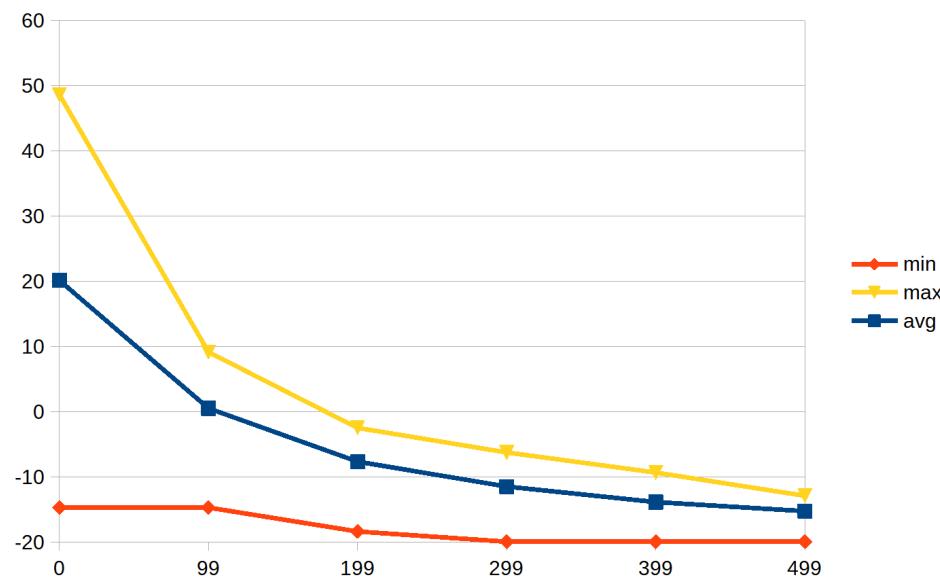
Większy promień dostosowania (2.0)

Zakres od	-5.12
Zakres do	5.12
Liczba improwizacji	500
Rozmiar pamięci	50
Wsp. odwołań do pamięci	0.95
Wsp. dostosowania	0.7
Promień dostosowania	2.0



iteracje	min	max	avg
0	-14.6823660176238	48.6266333522166	20.1301114271753
99	-14.6823660176238	9.16401907136637	0.530582046211876
199	-18.3460292856682	-2.47199034452849	-7.64382114764805
299	-19.9196180254676	-6.23114562260985	-11.4589549502349
399	-19.9196180254676	-9.31334371202617	-13.8426536394861
499	-19.9196180254676	-12.8752531971286	-15.2286380758629

Wykres harmonii

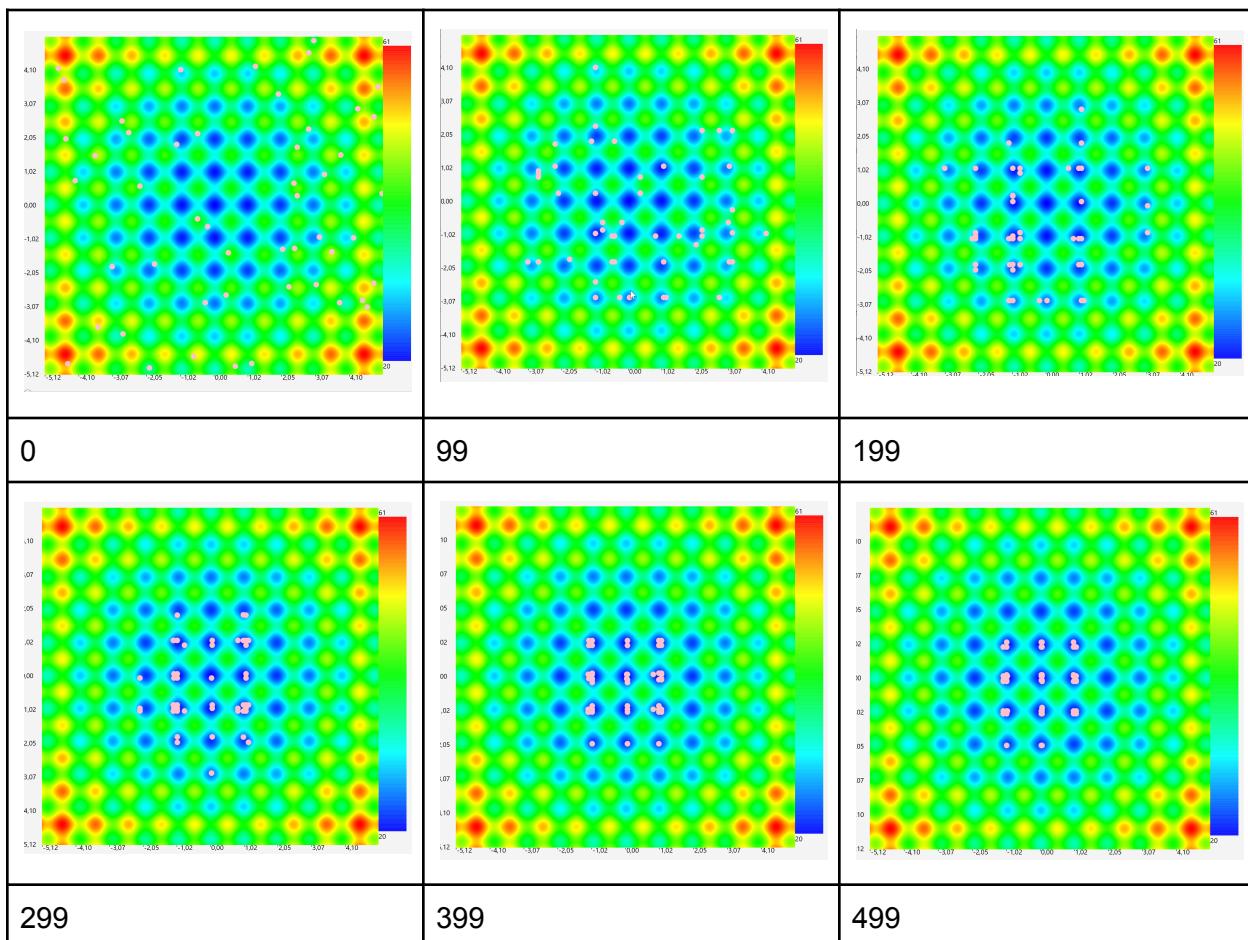


Obserwacje

- W przeciwnieństwie do funkcji Ackleya, zwiększenie wartości promienia dostosowania spowodowało, że uzyskaliśmy gorsze rezultaty - funkcja wolniej zbiera do minimum globalnego.
- Jest to prawdopodobnie spowodowane kształtem funkcji, tzn. w funkcji obecnych jest wiele minimów lokalnych - droga do minimum nie jest łagodna jak funkcja Ackleya, funkcja natrafia na wiele minimów lokalnych o dużych wartościach.

Mniejszy współczynnik dostosowania (0.1)

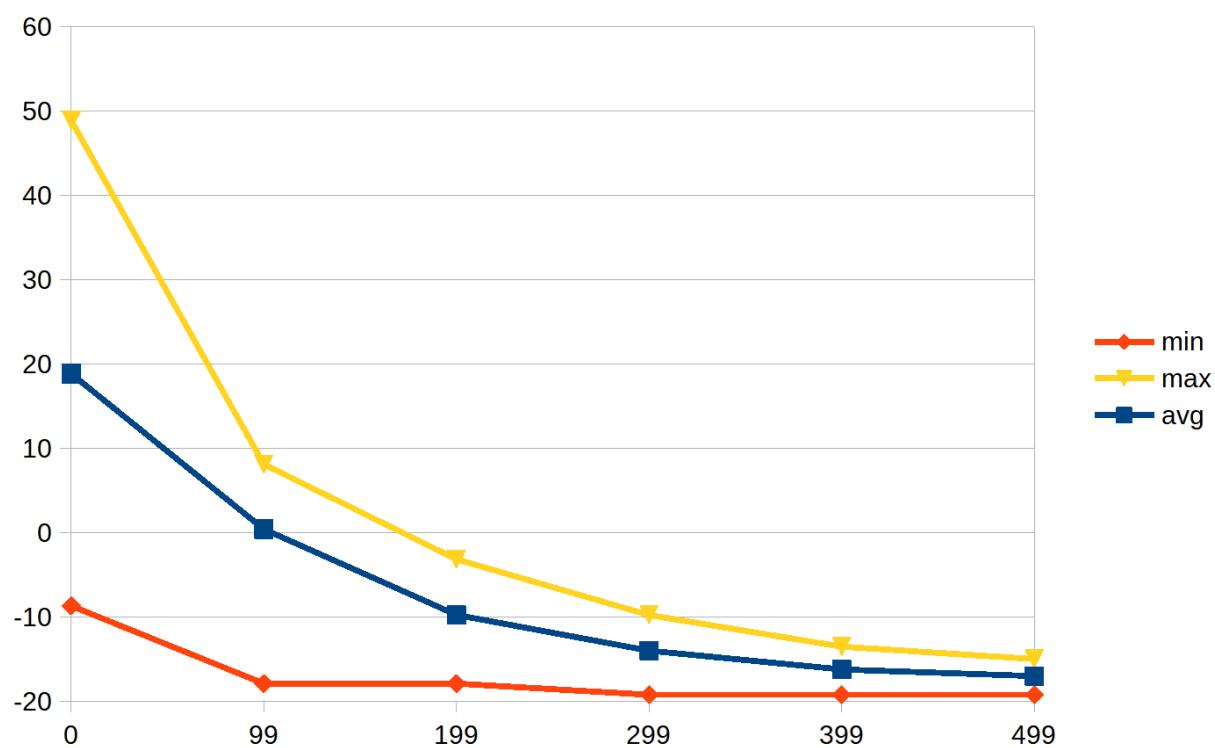
Zakres od	-5.12
Zakres do	5.12
Liczba improwizacji	500
Rozmiar pamięci	50
Wsp. odwołań do pamięci	0.95
Wsp. dostosowania	0.1
Promień dostosowania	1.0



iteracje	min	max	avg
0	-8.66701576980892	48.8649674765123	18.8449283715698

99	-17.8829790082648	8.11418213792209	0.430262929987389
199	-17.8829790082648	-3.13797689406718	-9.74063517876523
299	-19.2033648586768	-9.75380302933149	-13.9823834850947
399	-19.2033648586768	-13.5056296911475	-16.2091836798674
499	-19.2033648586768	-14.9728822276654	-16.9892916960058

Wykres harmonii



Obserwacje

- Zmniejszenie współczynnika dostosowania daje nam gorszą zbieżność do minimum globalnego dla pierwszych 200 iteracji, ale daje lepszą zbieżność dla iteracji 200-499.
- Wyniki te są zgodne z teorią badanej funkcji - dla wartości dalszych od minimum globalnego wyniki minimów lokalnych zwiększały się, co w połączeniu z mniejszym prawdopodobieństwem mutacji powoduje, że algorytm "dłużej" błądzi po minimach lokalnych. Wraz ze zmniejszeniem odległości od minimum globalnego wartości te stają się mniejsze, co przyspiesza dążenie do minimum globalnego, dlatego dla iteracji 200-499 obserwujemy bardziej stromy wykres, niż dla wariantu badanej funkcji z domyślnymi parametrami.

Wnioski

- Dla kolejnych iteracji, efekt poprawy wyniku (tj. wielkości kroku do poszukiwanego minimum globalnego) jest coraz mniejszy.
- Dla badanych funkcji zwiększanie rozmiaru pamięci powodowało pogorszenie rezultatów działania algorytmów. Mniejsza liczba harmonii dla badanych funkcji powoduje przechowywanie mniejszej liczby harmonii, dzięki czemu w pamięci szybciej pojawiają się harmonie lepsze, co ma pozytywny wpływ na działanie funkcji.
- Zmiana parametrów mających wpływ na losowość uzyskiwanych wyników w kolejnych iteracjach zmienia działanie algorytmu w zależności od wybranej funkcji. Dla funkcji Ackleya zmniejszenie prawdopodobieństwa losowości, tzn. zmniejszenie współczynnika dostosowania spowodowało, że uzyskaliśmy szybszą zbieżność. Przy zwiększeniu czynnika losowości, tj. współczynnika odwołań do pamięci uzyskaliśmy gorsze rezultaty - zbieżność była wolniejsza.
- Z kolei zwiększając promień dostosowania dla funkcji Ackleya, uzyskaliśmy szybszą zbieżność - jest to prawdopodobnie spowodowane tym, że dla badanego przedziału funkcja zbliżając się do minimum globalnego jest bardzo stroma, dzięki czemu wartości mogą szybciej "wpaść" do minimum globalnego dzięki mutacjom.
- Dla funkcji Rastrigina zwiększenie promienia dostosowania dało nam gorsze rezultaty ze względu na kształt funkcji - algorytm dłużej błądził po minimach lokalnych o dużych wartościach, wartość nie może "wpaść" do minimum globalnego jak w przypadku funkcji Ackleya, gdzie funkcja jest bardzo stroma wokół minimum globalnego.
- Bardziej skomplikowanym przypadkiem dla funkcji Rastrigina jest zmniejszenie współczynnika dostosowania, gdzie początkowo funkcja zbiega wolniej, jednak dla kolejnych iteracji (200-499) zbiega szybciej. Jest to spowodowane tym, że dla wartości dalszych od minimum globalnego wyniki minimów lokalnych zwiększają się. W związku z tym na początku mniejsza ilość mutacji przyspiesza zbieżność, jednak w późniejszych iteracjach, gdzie wartości minimów lokalnych nie są już tak duże, mutacje mogą "przeszkadzać" w osiągnięciu minimum globalnego.
- Aby w pełni przetestować wpływ parametrów na działanie funkcji, powinniśmy zmieniać kilka parametrów jednocześnie i porównywać wyniki z parametrami domyślnymi programu.
- Porównując funkcję Ackleya i Rastrigina, występują o wiele większe różnice między wynikiem w danej iteracji a poszukiwaną wartością minimum globalnego. Jest to spowodowane różnymi kształtami obu funkcji - dla funkcji Ackleya już od początku punkty przyjmują wartości zbliżone do poszukiwanego minimum globalnego. Dla funkcji Rastrigina taka sytuacja nie zachodzi ze względu na duże wartości minimów lokalnych, dodatkowo zwiększających się wraz z oddaleniem od minimum globalnego.