Grzegorz Topór 337294

WSI LAB1 - ALGORYTMY EWOLUCYJNE I GENETYCZNE

1. Opis zaimplementowanego algorytmu

Solver do znalezienia minimum globalnego podanej funkcji używa strategii ewolucyjnej (1+1) oraz posiada dwa warunki stopu: maksymalna liczba wywołań funkcji zależna od wymiarowości wektora zmiennych (n x 100) oraz maksymalna liczba wywołań funkcji bez znalezienia nowej, lepszej wartości funkcji. Solver zwraca dictionary gdzie: dict[t]= [f(x_t), best-so-far].
jak działa algorytm w dużym uproszczeniu:

1. stwórz nowego potomka:
   x' ← x + moc adaptacji * N(0, I) (rozklad gaussa)
2. Oblicz wartość funkcji:
   q ← q(x')

3. Jeśli q' < najlepsza_wartość:
   najlepsza_wartość ← q

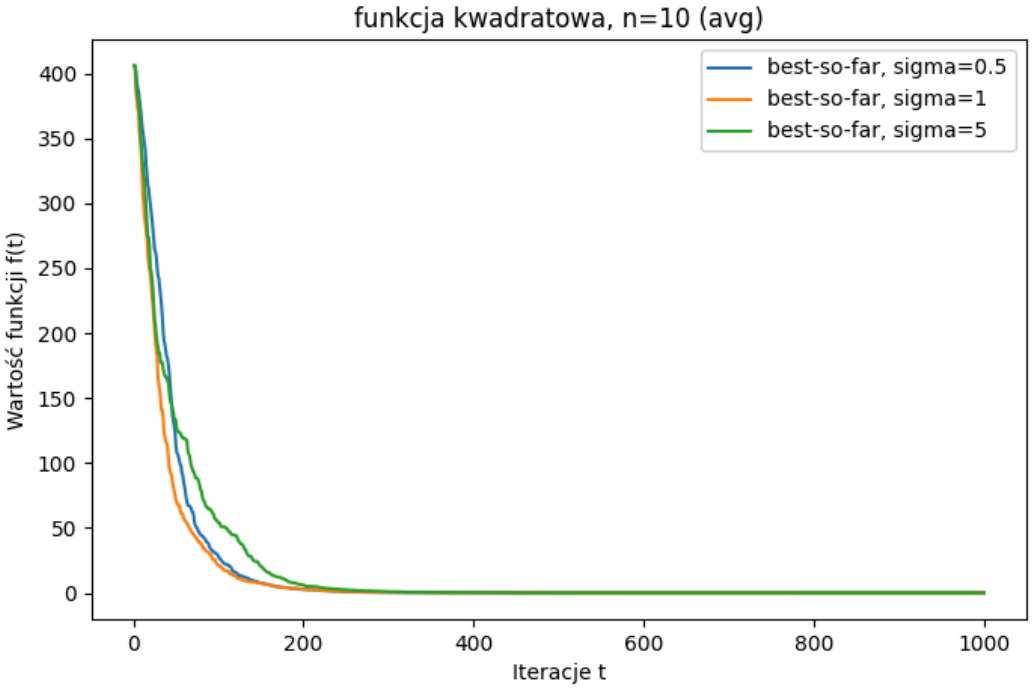
4. co okres adaptacji mocy::
   jeśli (liczba popraw / K) ≥ 1/5:
      σ ← σ * 1.21
   w przeciwnym razie:
      σ ← σ * 0.82
5. liczba_iteracji ← liczba_iteracji + 1

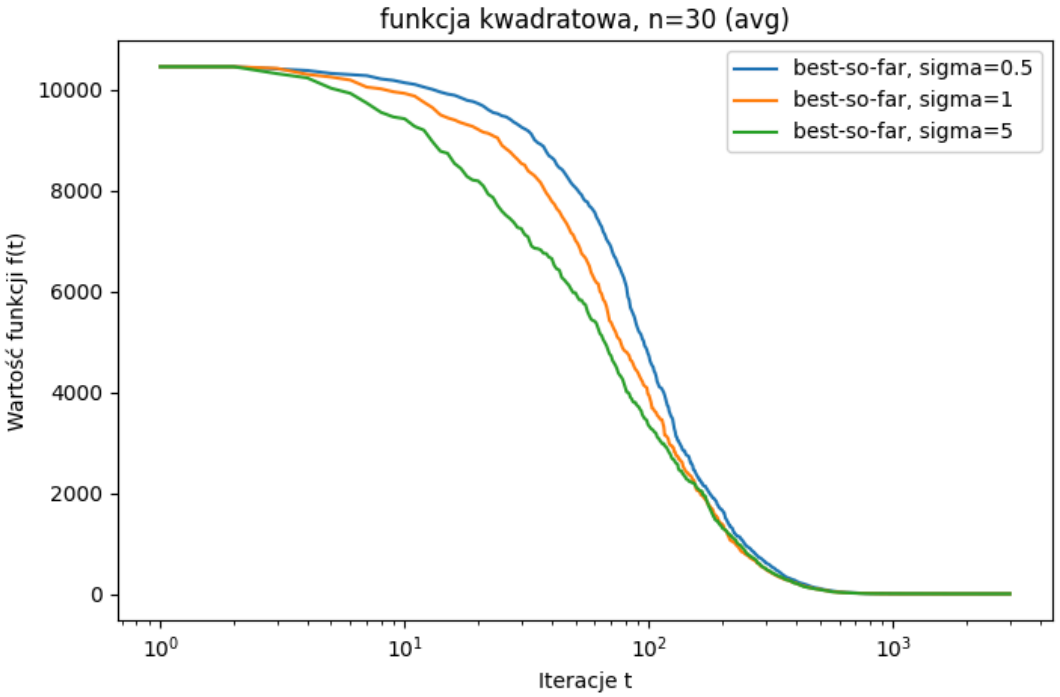
2. Planowane eksperymenty numeryczne

Eksperymenty skupiają się na sprawdzaniu **zależności mocy siły adaptacji w stosunku do możliwości eksploracji.** Program tworzy 12 wykresów, z czego 6 pokazuje średnie wartości best so far w zależności od iteracji z określonej liczby uruchomień (R=10). Pozostałe 6 wykresów pokazuje zestawienie pojedynczych uruchomień funkcji oraz zachowanie best so far value i f(x_t). Eksperymenty zostały przeprowadzone na trzech funkcjach (kwadratowa, ackley'a i rosenbrocka) dla wymiarowości n = 10, 30. Moc mutacji = 0.5, 1, 5.(oraz 2 dla funkcji ackley'a). Liczba wywołań funkcji została ograniczona do 200. Moc jest adopotowana co 10 iteracji. Za każdym uruchomieniem funkcji wektor początkowy jest taki sam, tak aby wybrany punkt nie miał wpływu na przebieg eksperymentów.

3. Opis uzyskanych wyników

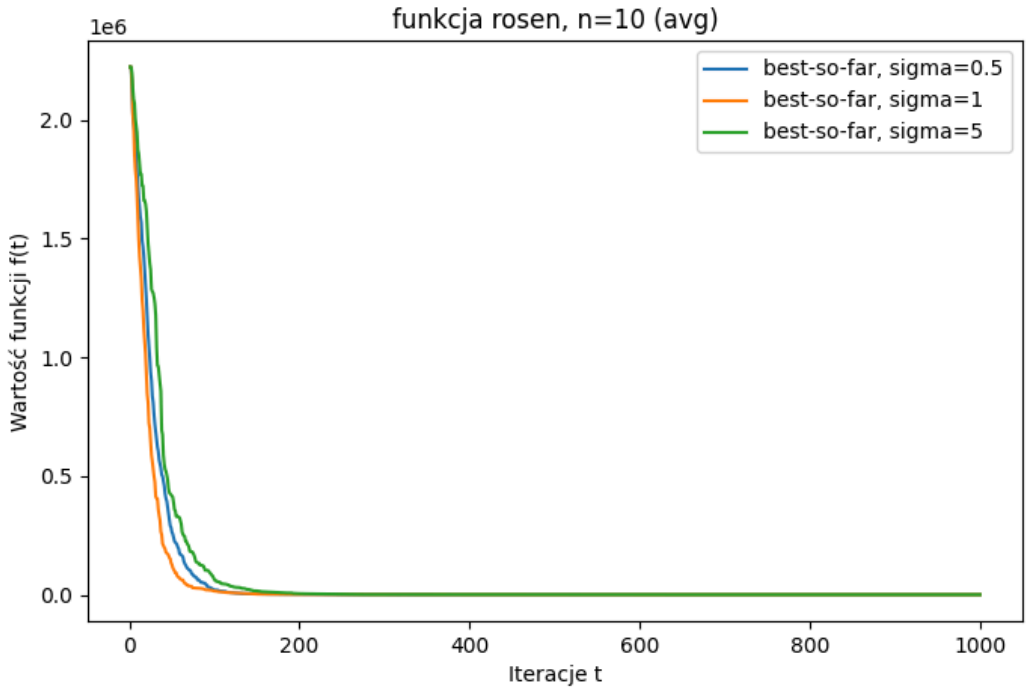
Funkcja kwadratowa

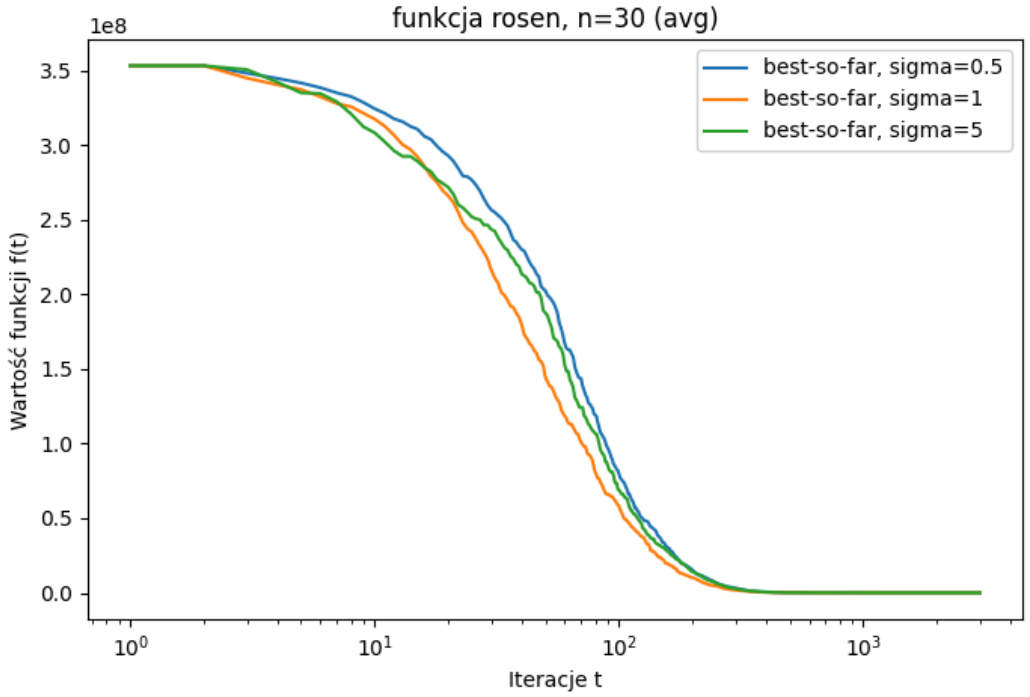




funkcja kwadratowa jest prosta dla algorytmu (1+1), znajduje minimum globalne zarówno dla n=10 i n=30, wielkość sigmy nie jest znacząca i nie wpływa na wynik badania

Funkcja rosenbrock’a:

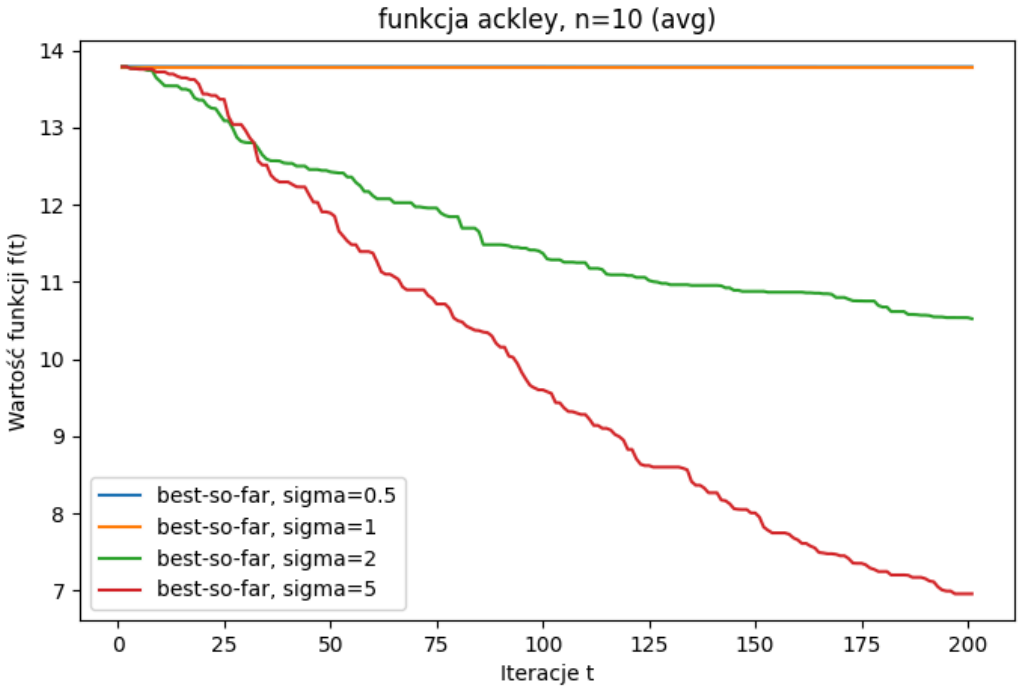




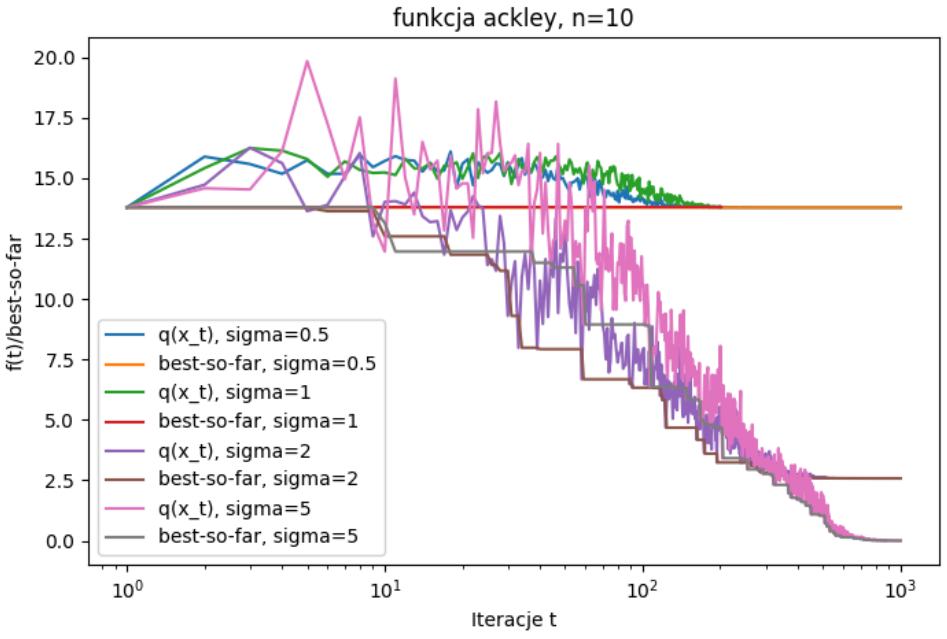
funkcja rosenbrocka jest dość prosta dla algorytmu (1+1),nawet dla małych sigmy=0.5 nie utyka w minimum lokalnym i znajduje minima globalne (nawet dla n=30)

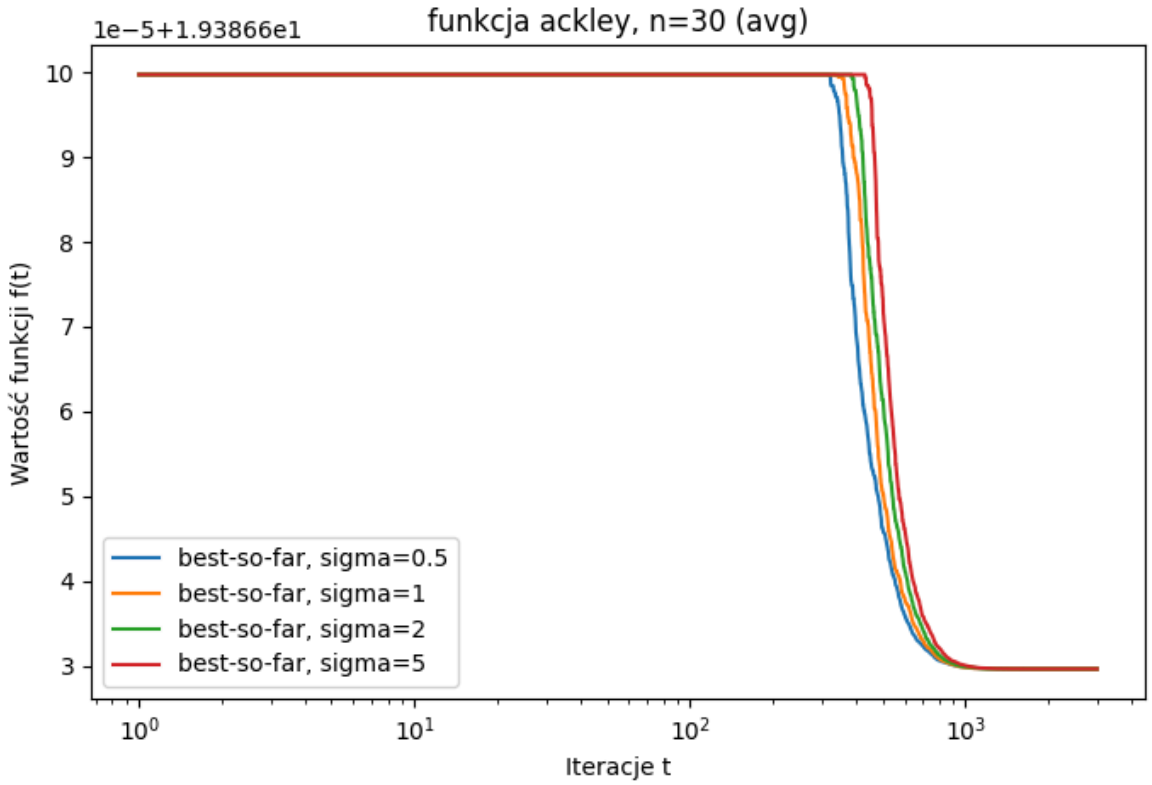
Funkcja ackleya:

funkcja ackleya jest najtrudniejsza dla solvera i najciekawiej widać problem strategii (1+1) i jego wady wynikające z braku populacji i słabą eksploracją.



Dla wymiarowości n=10 widać że moc 0.5 i 1 jest zbyt mała i funkcja utyka w minimach lokalnych, natomiast 2 i 5 wychodzą z nich i znajdują lepsze minima. Gdy sigma jest zbyt mała, wdrapuje się lecz nie daje rady przeskoczyć do kolejnego minima, widać to na wykresie zestawiającym best so far i current value:





Dla wymiarowości n=30, kiedy usuniemy ograniczenie iteracji bez znalezienia nowej, najlepszej wartości algorytm po dłuższym utknięciu, wydostaje się przypadkowo znajdując lepszą “dolinę” w funkcji ackleya. Aby znaleźć szybciej minimum należałoby użyć algorytmu z populacją, która gwarantuje szybszą eksploracje.

4. Wnioski z przeprowadzonych badań

Przeprowadzone eksperymenty numeryczne uwydatniają sposób działania strategii ewolucyjnej (1+1) oraz znaczenia dobrania odpowiednich parametrów, takich jak siła adaptacji.

Podczas badania na funkcji Ackleya objawiły się cechy charakterystyczne tej strategii,to jest dłuższe utknięcie i nagłe, przypadkowe znalezienie lepszego obszaru w funkcji. Z tego wynika, że do trudniejszych funkcji trzeba odpowiednio dobrać wielkość adaptacji, tak aby nie utknęła na zawsze, a także jak istotna jest liczba iteracji, pomimo pozornej stagnacji w jednym minimum lokalknym.