

Zadanie 1. Strategie Ewolucyjne

Julia Jodczyk

10 listopada 2021

1 Treść zadania

Zaimplementuj strategię ewolucyjną typu $(\mu/\mu, \lambda)$, w której rozważysz dwa mechanizmy adaptacji zasięgu mutacji σ :

- samoadaptację (SA)
- metodę logarytmiczno-gaussowską (LMR).

Zbadaj zbieżność obu metod w zależności od rozmiaru populacji potomnej oraz początkowej wartości zasięgu mutacji. Do badań wykorzystaj

- funkcję sferyczną
- funkcję w postaci

$$q(x) = [(||x||^2 - D)^2]^{1/8} + D^{-1}(1/2||x||^2 + \sum_{i=1}^D x_i) + 1/2$$

gdzie

$$x \in (-100, 100)^D$$

, gdzie D to wymiarowość funkcji.

2 Wprowadzenie

Strategie ewolucyjne są typem algorytmów ewolucyjnych. Działają na wektorach liczb zmiennoprzecinkowych. Stosowane są głównie do optymalizacji funkcji rzeczywistych wielowymiarowych. Schemat działania składa się z reprodukcji, krzyżowania i mutacji, chociaż nie wszystkie z tych etapów muszą być uwzględnione. W poniższym algorytmie nie korzystam z krzyżowania, mutacja zachodzi w punkcie 2.a, natomiast reprodukcja w 2.c.

Opis algorytmu

Zastosowany przeze mnie algorytm strategii $(\mu/\mu, \lambda)$ wygląda następująco:

1. Wybierz punkt początkowy i załaduj go do x^t , początkowy współczynnik mutacji załaduj do σ^t .
2. Dopóki niezaspokojony jest warunek stopu wykonaj:
 - (a) Na podstawie x^t wygeneruj λ punktów pochodnych.
 - (b) Dokonaj ewaluacji wygenerowanych punktów.
 - (c) Wybierz μ najlepszych punktów.
 - (d) Na ich podstawie oblicz następny punkt $x^t = \sum_{i=1}^{\mu_i} x_i^t \cdot w_i$, gdzie $\forall_i w_i = 1/\mu$.
 - (e) Oblicz następny współczynnik mutacji.

Ewaluacja punktów w 2.b następuje na podstawie wartości funkcji oceny (w tym wypadku funkcji celu). Im mniejsza jest ta wartość, tym wyższa ocena.

Generowanie punktów pochodnych różni się w zależności od użytego mechanizmu adaptacji zasięgu mutacji (które opisane są w dalszej części).

- dla SA

$$\sigma_i^t = \sigma^t \cdot \exp(\tau \mathcal{N}_i(0, 1))$$

$$x_i^t = x^t + \sigma_i^t \cdot \mathcal{N}_i(0, I_D), i \in 1 \dots \lambda$$

gdzie σ^t wyliczana jest w punkcie 2.e na podstawie wzoru

$$\sigma_t = \sum_{i=1}^{\mu_i} \sigma_i^t \cdot w_i$$

- dla LMR

$$x_i^t = x^t + \sigma^t \cdot \mathcal{N}_i(0, I_D)$$

gdzie

$$\sigma^t = \sigma^{t-1} \cdot e^{\tau \mathcal{N}_i(0,1)}$$

W mechanizmie samoadaptacji ewolucji podlega parametr mutacji. Usprawnia to działanie całego algorytmu. Aby strategia ewolucyjna znalazła ekstremum globalne funkcji chcielibyśmy, aby początkowe mutacje były duże, pozwalając na szeroką eksplorację przestrzeni. W miarę przybliżania się do ekstremum, parametr mutacji powinien maleć i punkty bliskie sobie powinny być rozpatrywane przez algorytm (eksploatacja). Na takie zmiany parametru mutacji pozwala właśnie samoadaptacja, ponieważ przy okazji promowania najlepszych wartości x , promowane są ich współczynniki mutacji.

Metoda logarytmiczno-gaussowska nie zakłada promowania dobrych rozwiązań. Kolejne wartości σ są dobierane bez względu na najlepsze osobniki. Z tego powodu powinna działać gorzej od samoadaptacji i być bardziej wrażliwa na dobór parametrów strojących.

Uwagi implementacyjne

Domyślną wartością μ jest 0.5λ , $\tau = b \cdot 1/\sqrt{D}$, gdzie domyślnie $b = 1$.

Warunek stopu składa się z dwóch części:

1. budżet funkcji $> 1000 \cdot D$
2. różnica między kolejnymi wartościami $f(x^t)$, (gdzie f jest funkcją celu) jest mniejsza od ustalonego epsilon (domyślnie 0.0000001) przez pewną liczbę iteracji (domyślnie 20).

Dodatkowo, algorytm posiada zmienną **best**, w której przechowywany jest punkt, dla którego wartość funkcji celu była dotychczas najmniejsza. Pod koniec działania algorytmu, to jego wartość jest zwracana, a nie ostatnia znaleziona wartość. Jest to zabezpieczenie przed sytuacją, w której, przy źle dobranych parametrach, algorytm zacznie oddalać się od najmniejszej wartości.

3 Opis eksperymentów numerycznych

Skuteczność strategii ewolucyjnej zależy przede wszystkim od wielkości tworzonych populacji i współczynnika mutacji. Dla obu mechanizmów adaptacji zasięgu mutacji sprawdzę zależność wartości funkcji celu od ilości iteracji przy różnych wartościach λ i σ .

Eksperymenty przeprowadzę dla losowego punktu początkowego $x = (41.29, 16.8, 12.29, 47.18, 15.75, 11.3, 95.79, 87.4, 16.05, 7.87)$ oraz dla punktu $x = (100, 100, 100, 100, 100, 100, 100, 100, 100, 100)$.

Jeden eksperyment to pięciokrotne wywołanie algorytmu dla tych samych parametrów (punktu początkowego, funkcji celu, λ (kolejno 10, 50, 100), σ (kolejno 1, 10, 0.1)). Aby zapewnić powtarzalność eksperymentów używam ziarna `numpy.random.seed()` z wartościami 997, 998, 999, 1000, 1001 dla kolejnych iteracji w eksperymencie.

4 Wyniki eksperymentów

Wyniki eksperymentów dla funkcji sferycznej:

1. losowy punkt

λ / σ	1	10	0.1
10	9.402714869010427e-08	2.483360158641352e-08	1.8960720803986048e-08
50	4.0915925925514456e-10	2.192364633153346e-09	7.376841244460082e-10
100	0.00040262501309946356	8.575807585912866e-10	14593.340746970429

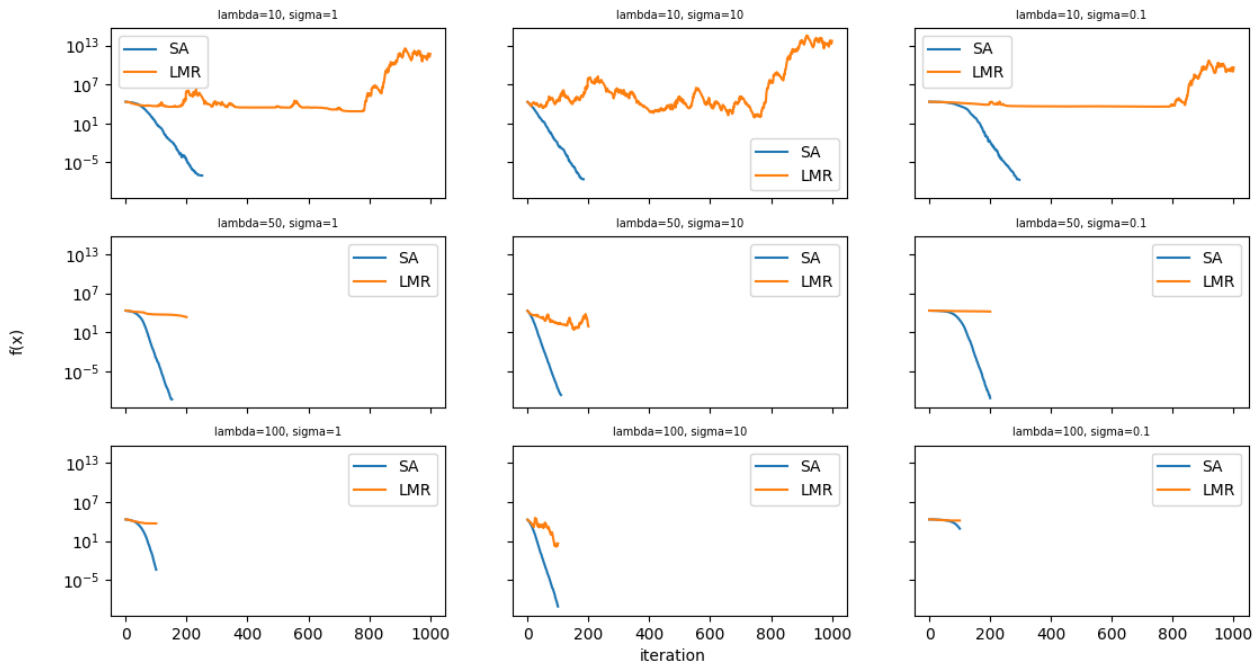
Tabela 1: Samoadaptacja, wartości znalezionej minimum funkcji sferycznej w zależności od λ i σ dla punktu losowego.

λ / σ	1	10	0.1
10	550.7546783134923	1.6697071977557427	3821.6726431663456
50	2188.7080942104717	0.5296413593937308	15391.053342584011
100	5285.259674580069	0.46380901372506206	14593.340746970429

Tabela 2: Metoda Logarytmiczno-Gaussowska, wartości znalezione minimum funkcji sferycznej w zależności od λ i σ dla punktu losowego.

Wartości znalezione przez algorytm z metodą samoadaptacji są bardzo bliskie minimum dla prawie wszystkich testowanych wartości parametrów λ i σ . Jedyną kombinacją dającą bardzo duży wynik to $\lambda = 100$, $\sigma = 0.1$. Mały parametr mutacji nie pozwala na dobrą eksploatację przestrzeni, a kolejno generowane punkty są blisko siebie. Przez to budżet wywołań funkcji celu wyczerpuje się zanim algorytm zdąży zejść do minimum (a przez dużą wartość parametru λ budżet wywołań funkcji starcza na małą ilość iteracji algorytmu).

Algorytm z metodą logarytmiczno-gaussowską działa znacznie gorzej. Najbliżej minimum dochodzimy przy kombinacjach $\lambda = 50$, $\sigma = 10$ i $\lambda = 100$, $\sigma = 10$. Jak wcześniej wspomniałam, ta metoda jest wrażliwa na parametry strojące i te przeze mnie testowane nie pozwoliły na znalezienie minimum globalnego.



Rysunek 1: Wykresy zależności wartości funkcji sferycznej od ilości iteracji dla różnych wartości λ i σ dla punktu losowego.

Na wykresach widać, że przy samoadaptacji wartość funkcji celu maleje z każdą iteracją. Przy metodzie LMR zachowują się bardzo różnie w zależności od dobranych parametrów. W przypadkach, w których udało się znaleźć wartość najbliższą minimum, wykresy pokazują stopniowy spadek (choć nie jest on tak gładki i szybki jak w przypadku samoadaptacji).

2. punkt z granicy przedziału

λ / σ	1	10	0.1
10	3.472729752725441e-08	3.0163828357027685e-08	1.5632259264792307e-08
50	6.441256696172573e-10	5.571676860699168e-10	1.7847746669664947e-05
100	2.5257388632667346	4.279234128325984e-08	20198.757809441617

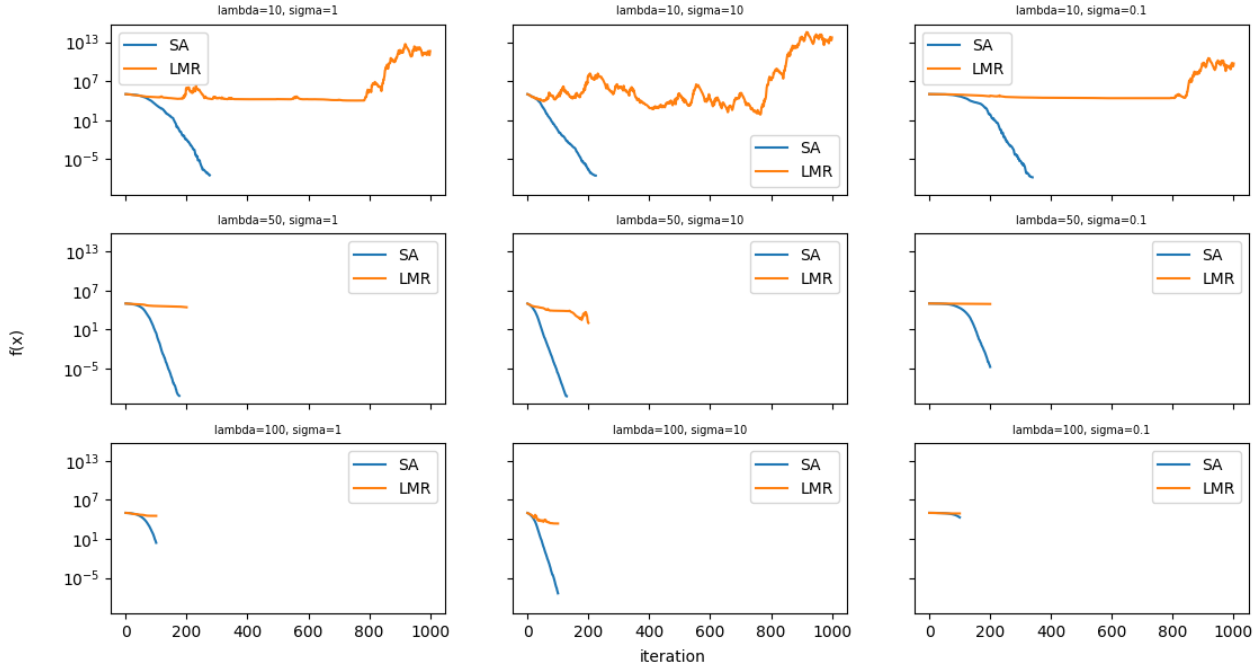
Tabela 3: Samoadaptacja, wartości znalezione minimum funkcji sferycznej w zależności od λ i σ dla punktu z granicy przedziału.

λ / σ	1	10	0.1
10	9553.393925130029	0.3278389850212903	22935.917440771816
50	26533.550753098447	2.532227379134338	84673.91559469451
100	34837.21943393121	2287.3675832857894	81192.09425528407

Tabela 4: Metoda Logarytmiczno-Gaussowska, wartości znalezione minimum funkcji sferycznej w zależności od λ i σ dla punktu z granicy przedziału.

Wyniki dla punktu granicznego przy samoadaptacji nie różnią się znacząco od tych dla punktu losowego. Wartości znalezione minimum są większe, ponieważ algorytm zaczynał pracę w punkcie bardziej oddalonym od minimum.

Przy LMR jest, jednak dla eksperymentu $\lambda = 100$, $\sigma = 10$, poprzednio najlepszego, wynik jest znacząco gorszy i tym razem to kombinacja $\lambda = 10$, $\sigma = 10$ działa najlepiej. Można wnioskować, że skuteczność algorytmu zależy nie tylko od wartości λ i σ , ale również od punktu początkowego.



Rysunek 2: Wykresy zależności wartości funkcji sferycznej od ilości iteracji dla różnych wartości λ i σ dla punktu z granicy przedziału.

Wyniki dla funkcji $q(x)$

1. losowy punkt

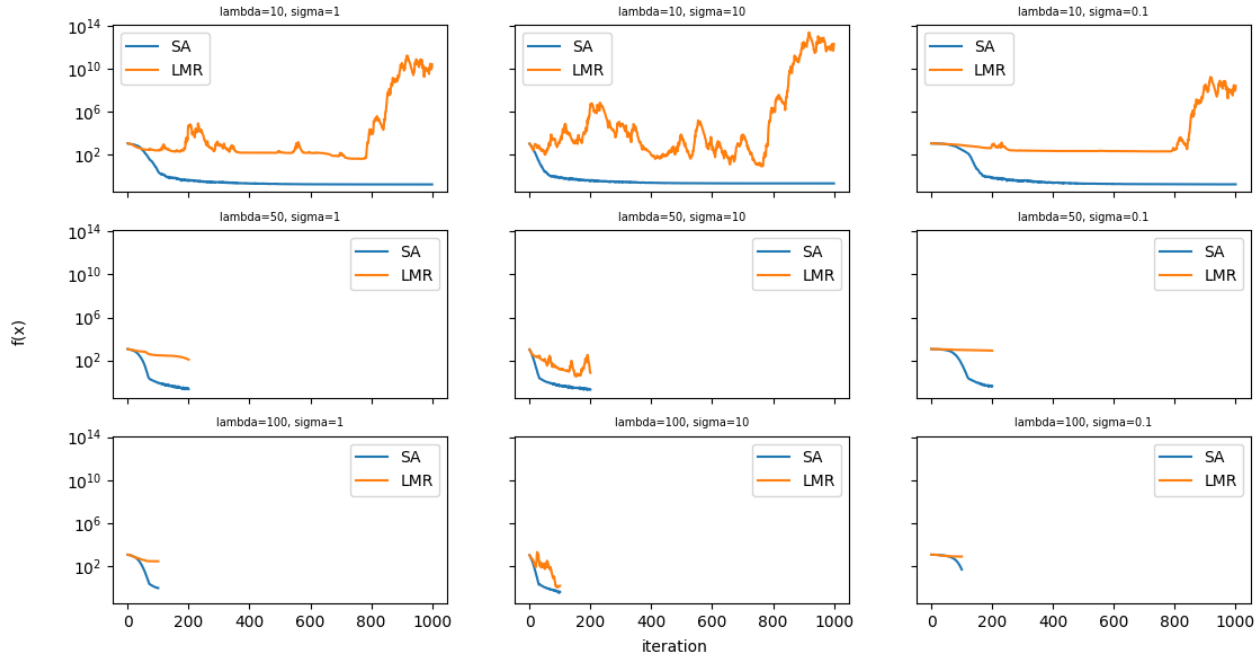
λ / σ	1	10	0.1
10	0.17216190371686774	0.2148244195622011	0.17578961771005605
50	0.13906422665814464	0.11621948009403363	0.25191865273468317
100	0.8544474745838672	0.21568317435890288	47.57854117648933

Tabela 5: Samoadaptacja, wartości znalezione minimum funkcji $q(x)$ w zależności od λ i σ dla punktu losowego.

λ / σ	1	10	0.1
10	31.82122807676061	1.135069413035843	200.26535163244472
50	119.03887959731313	0.876519227063253	809.9426193597553
100	279.987395268316	0.7910297884167924	767.3976996237659

Tabela 6: Metoda Logarytmiczno-Gaussowska, wartości znalezione minimum funkcji $q(x)$ w zależności od λ i σ dla punktu losowego.

Nie znane jest minimum funkcji $q(x)$, ale na podstawie otrzymanych wyników można je oszacować na liczbę z przedziału $(0.15, 0.25)$. Ponownie algorytm z samoadaptacją poradził sobie niezależnie od dobranych parametrów, a z LMR znalazł wartości o wiele większe. Najlepsze wyniki uzyskano dla takich samych kombinacji λ i σ jak w przypadku funkcji sferycznej.



Rysunek 3: Wykresy zależności wartości funkcji $q(x)$ od ilości iteracji dla różnych wartości λ i σ dla punktu z granicy przedziału

Wykresy wyglądają podobnie do tych uzyskanych przy eksperymentach na funkcji sferycznej. Widać, jednak, że na znalezienie minimum potrzeba więcej iteracji. Jest tak ponieważ funkcja $q(x)$ ma wiele minimów lokalnych w okolicach minimum globalnego. Dlatego po zbliżeniu się do ekstremum, algorytm potrzebuje więcej czasu na eksploatację obszaru.

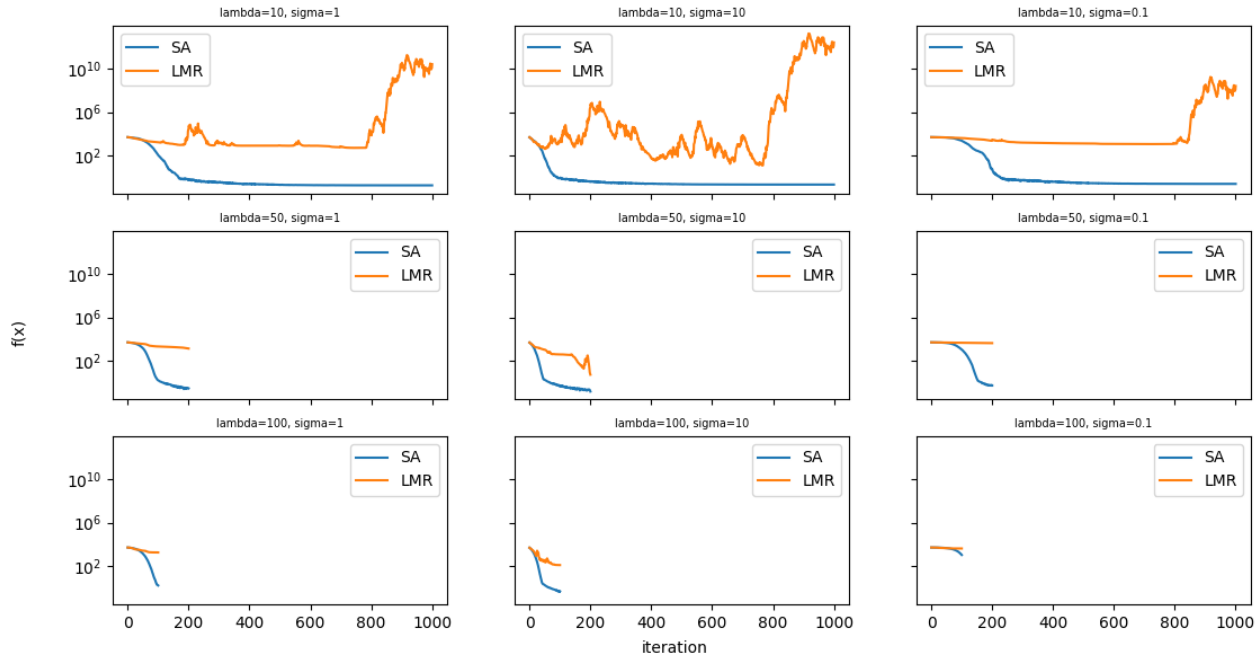
λ / σ	1	10	0.1
10	0.19135134360523257	0.22509193086792081	0.26065034745975585
50	0.185655855696994	0.12435473564301729	0.4431843398740384
100	1.6691639463693897	0.36430319265447475	1060.3241024732608

Tabela 7: Samoadaptacja, wartości znalezione minimum funkcji sferycznej w zależności od λ i σ dla punktu z granicy przedziału.

λ / σ	1	10	0.1
10	494.8043564016157	1.1909023099661624	1181.4894287962366
50	1377.6751950659207	1.0049052595763643	4342.959253194914
100	1796.7092150688713	123.90453185039885	4165.964155085903

Tabela 8: Metoda Logarytmiczno-Gaussowska, wartości znalezione minimum funkcji $q(x)$ w zależności od λ i σ dla punktu z granicy przedziału.

Wyniki eksperymentów z punktem granicznym są analogiczne jak w funkcji sferycznej.



Rysunek 4: Wykresy zależności wartości funkcji sferycznej od ilości iteracji dla różnych wartości λ i σ dla punktu losowego.

5 Wnioski

Strategie ewolucyjne pozwalają na znalezienie ekstremum globalnego funkcji. Używanie samoadaptacji jako mechanizmu mutacji działa o wiele lepiej od metody logarytmiczno-gaussowskiej. Przy samoadaptacji algorytm znajdował minimum funkcji szybciej i było ono mniejsze. Ponadto wartości funkcji celu malały dla kolejnych znajduwanych punktów, w przeciwieństwie do LMR, gdzie wykres zależności wartości funkcji celu od ilości iteracji był niemonotoniczny. Dodatkowo samoadaptacja jest odporna na dobór parametrów strojących, LMR bardzo wrażliwy. Dlatego SA jest bardziej bezpieczny i uniwersalny.

Samoadaptacja działa lepiej dla większych początkowych wartości parametru mutacji σ , ponieważ sprzyja on dużej eksploatacji w pierwszych iteracjach algorytmu. Mniejsze σ powodują przedwczesną eksploatację i wolne przybliżanie do minimum globalnego (bądź znalezienie jedynie minimum lokalnego).

Przy metodzie logarytmiczno-gaussowskiej zdaje się nie istnieć metoda pozwalająca znaleźć optymalne wartości parametrów strojących. Jedynym sposobem na ich znalezienie jest przeprowadzanie eksperymentów numerycznych.

Wyniki eksperymentów sugerują, że dobór parametrów jest niezależny od funkcji celu, ponieważ najlepsze wyniki uzyskano dla tych samych λ i σ .