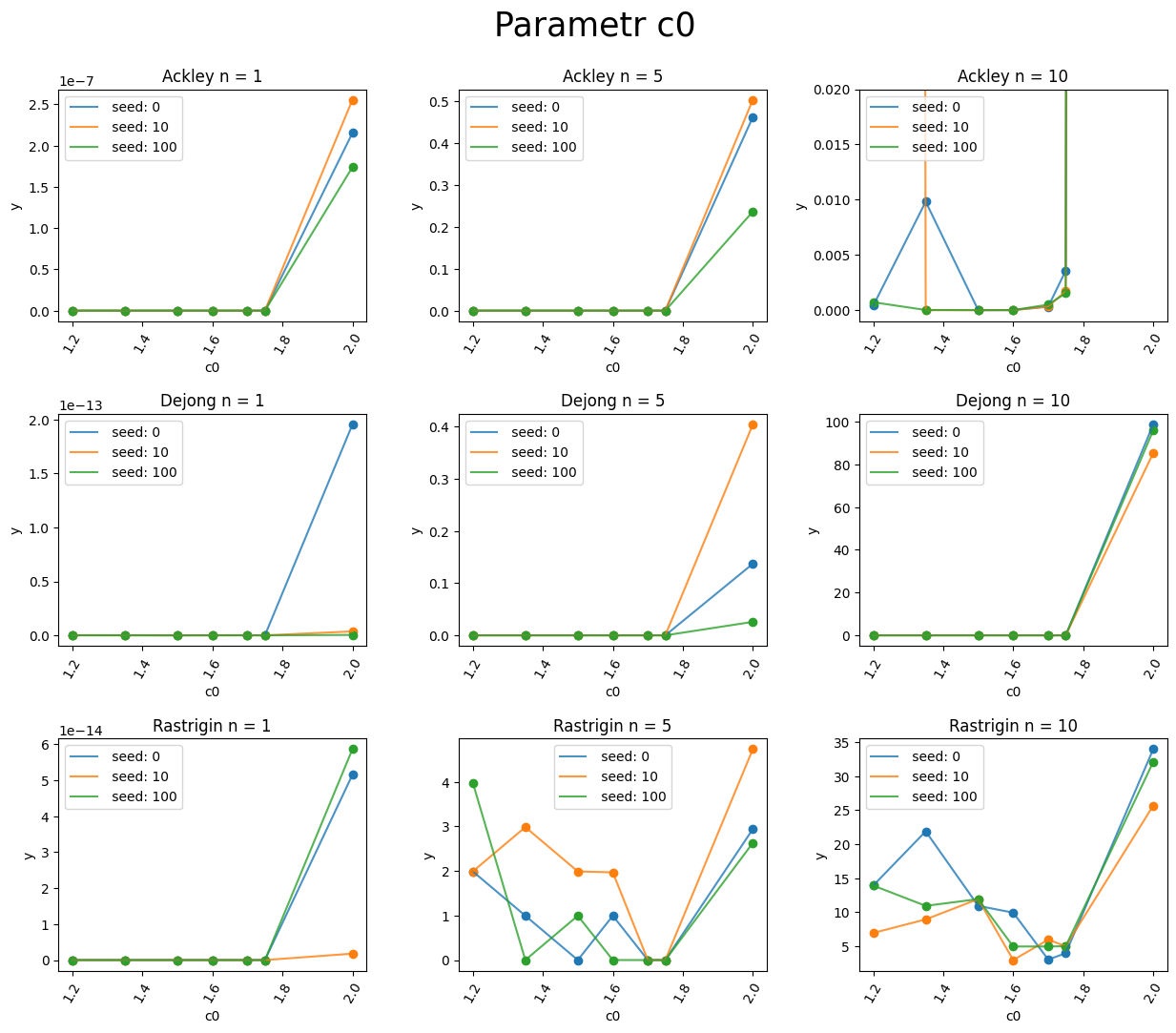
Sprawozdanie

Ćw. 9 – Algorytm ewolucyjny (AE)  
Filip Horst 311257

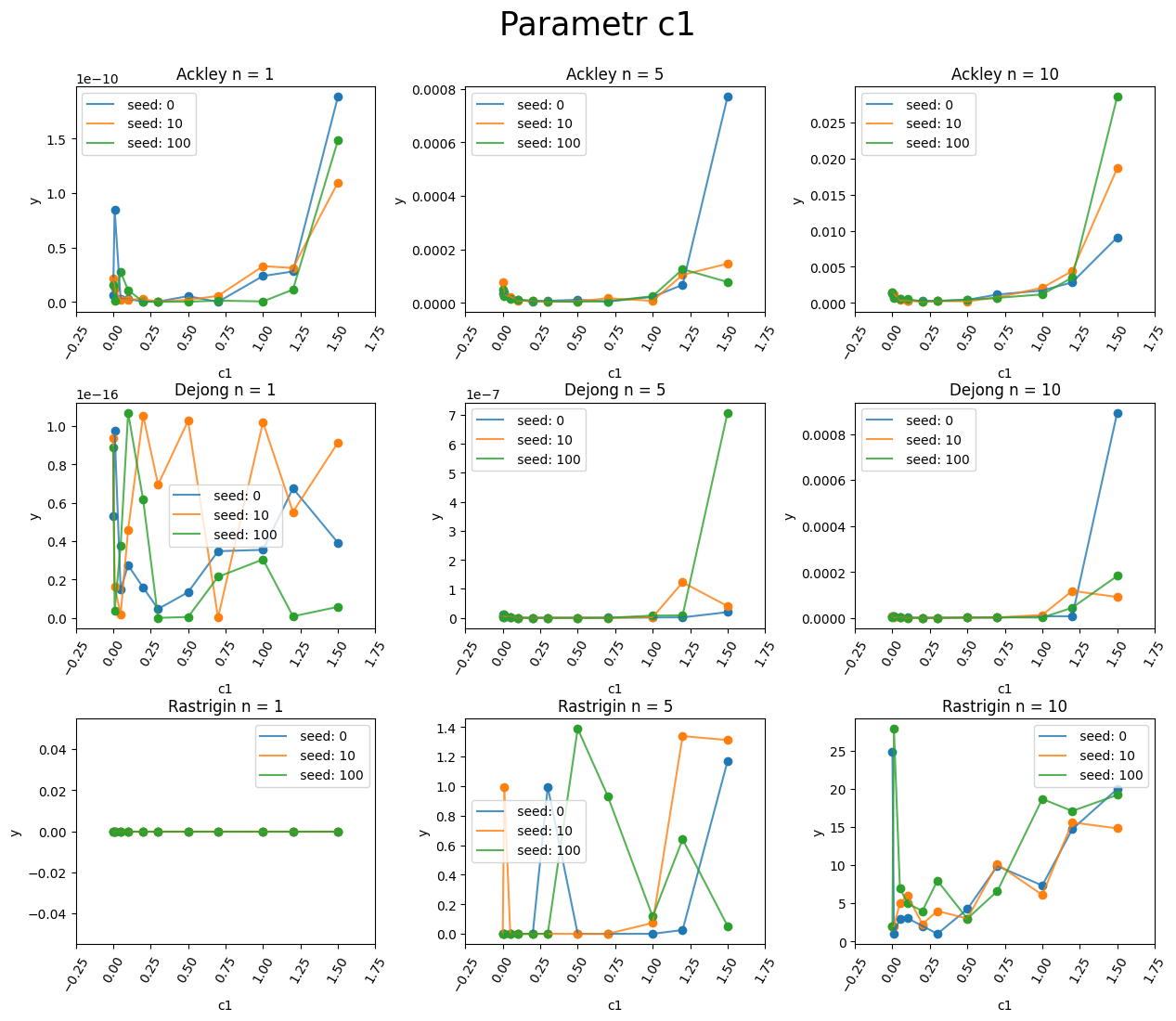
# Badanie wpływu parametrów na wyniki

## C0



Najważniejszym wnioskem płynącym z eksperymentu jest to, że wartość c0 posiada pewne optimum, czyli może być zarówno zbyt wysokie, jak i zbyt niskie. Szczególnie dobrze widać to w przypadku funkcji Rastrigina o 10 wymiarach. Progi, których przekroczenie powoduje duży błąd są inne w zależności od zadania, ale dobrym przedziałem startowym wydaje się być 1.4 – 1.8.

## C1

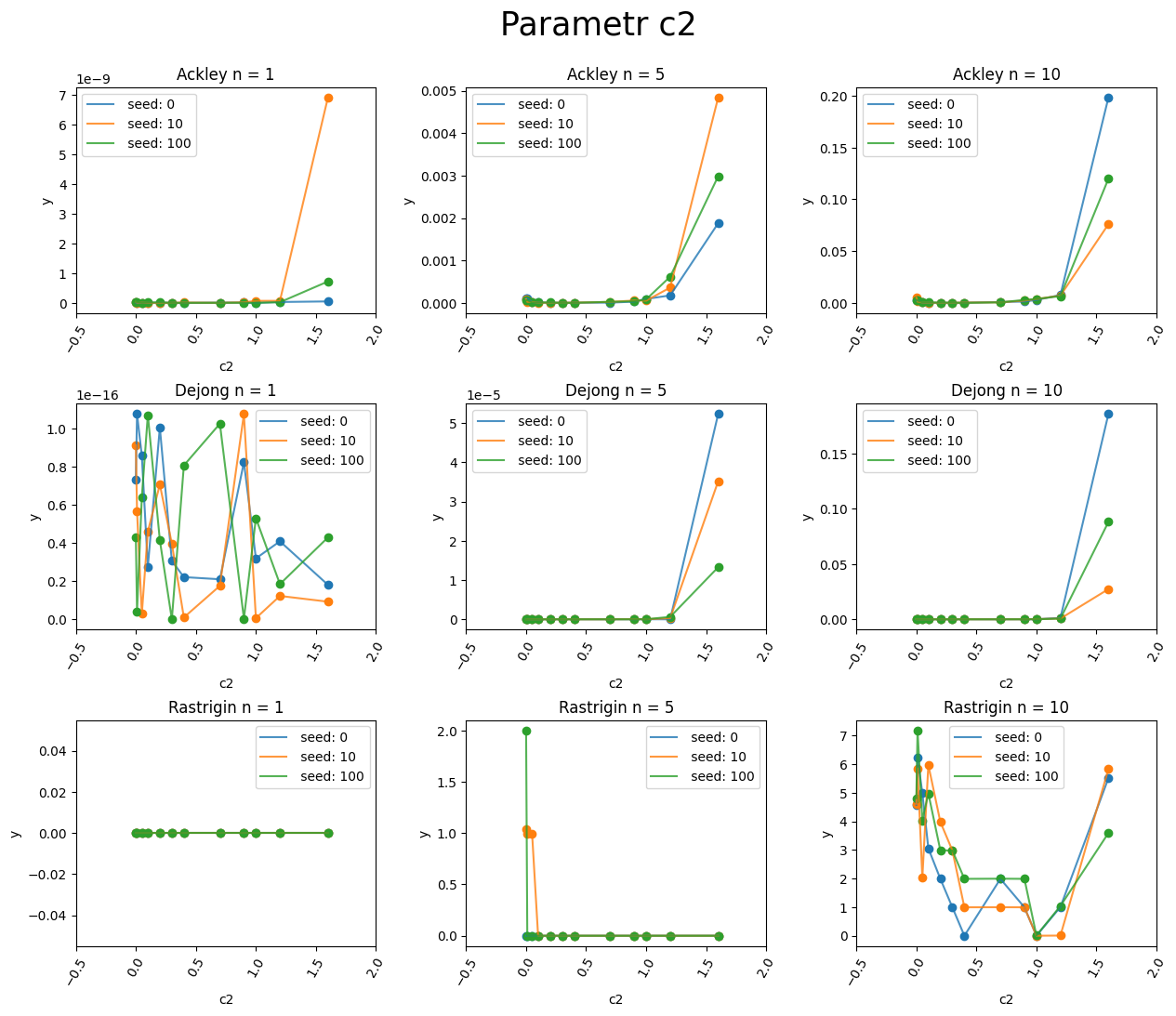


Dla parametru C1 zależność jest podobna do C0, co widać jednak tylko na 10 wymiarowej funkcji Rastrigina. Różnicą jest jednak przede wszystkim wartość – tym razem przedział startowy lepiej ustalić na 0.1 – 0.4.

Warto zaznaczyć, że losowe skoki dla funkcji jednowymiarowych są skutkiem bardzo małych wartości wyników (rząd e-10, e-16), gdzie zwykłe elementy losowości powodują skoki wartości – ostatecznie każdy z nich jest praktycznie idealny.

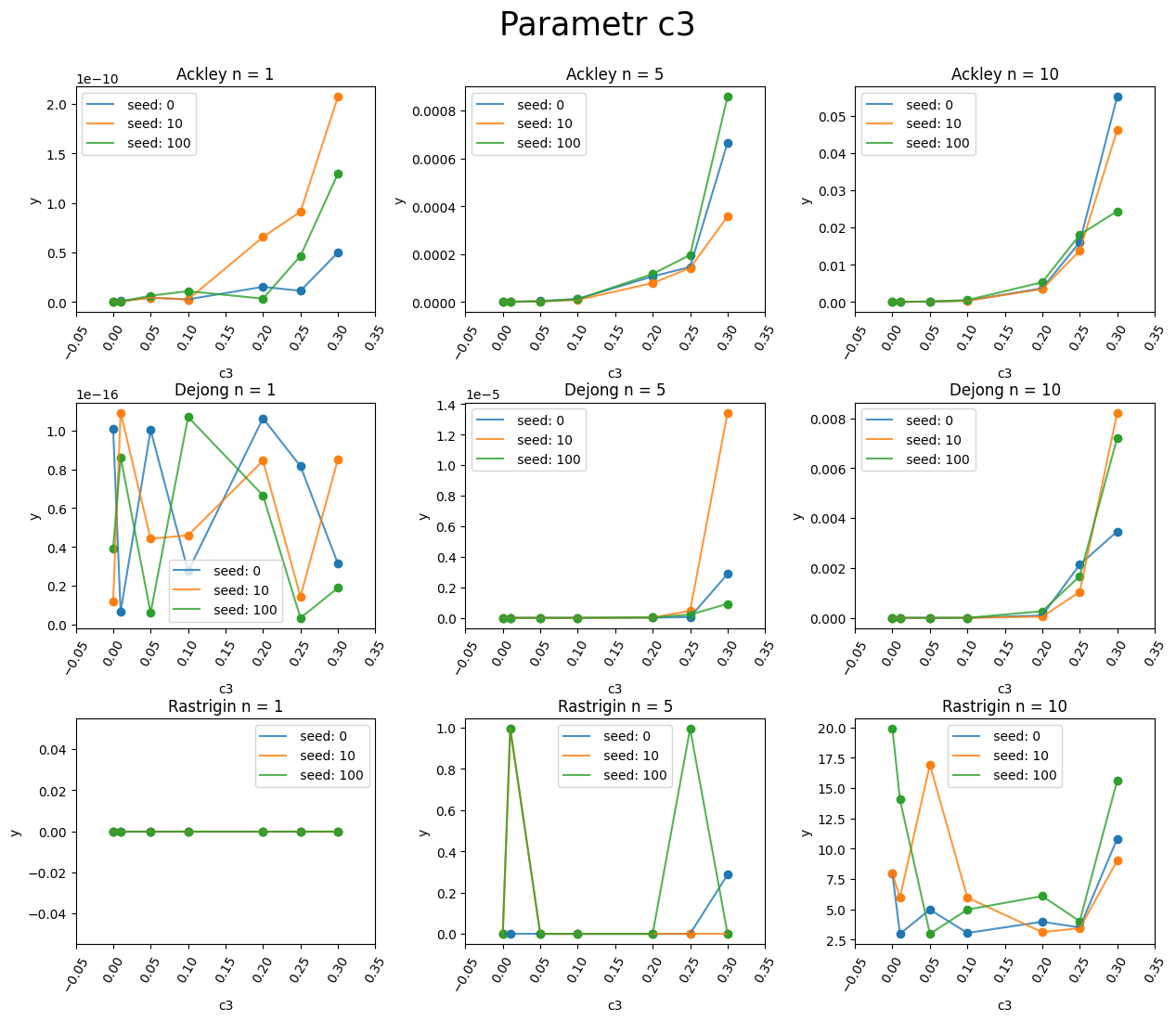
Wyniki dla f. Ratrigina są bardzo nietypowe, ponieważ zauważalne są losowe oscylacje również dla wersji 5- wymiarowej, natomiast dla funkcji 10 – wymiarowej najbardziej zauważalny jest charakter paraboli (oczywiście mówiąc bardzo ogólnie) zależności między c1, a wynikiem y.

## C2



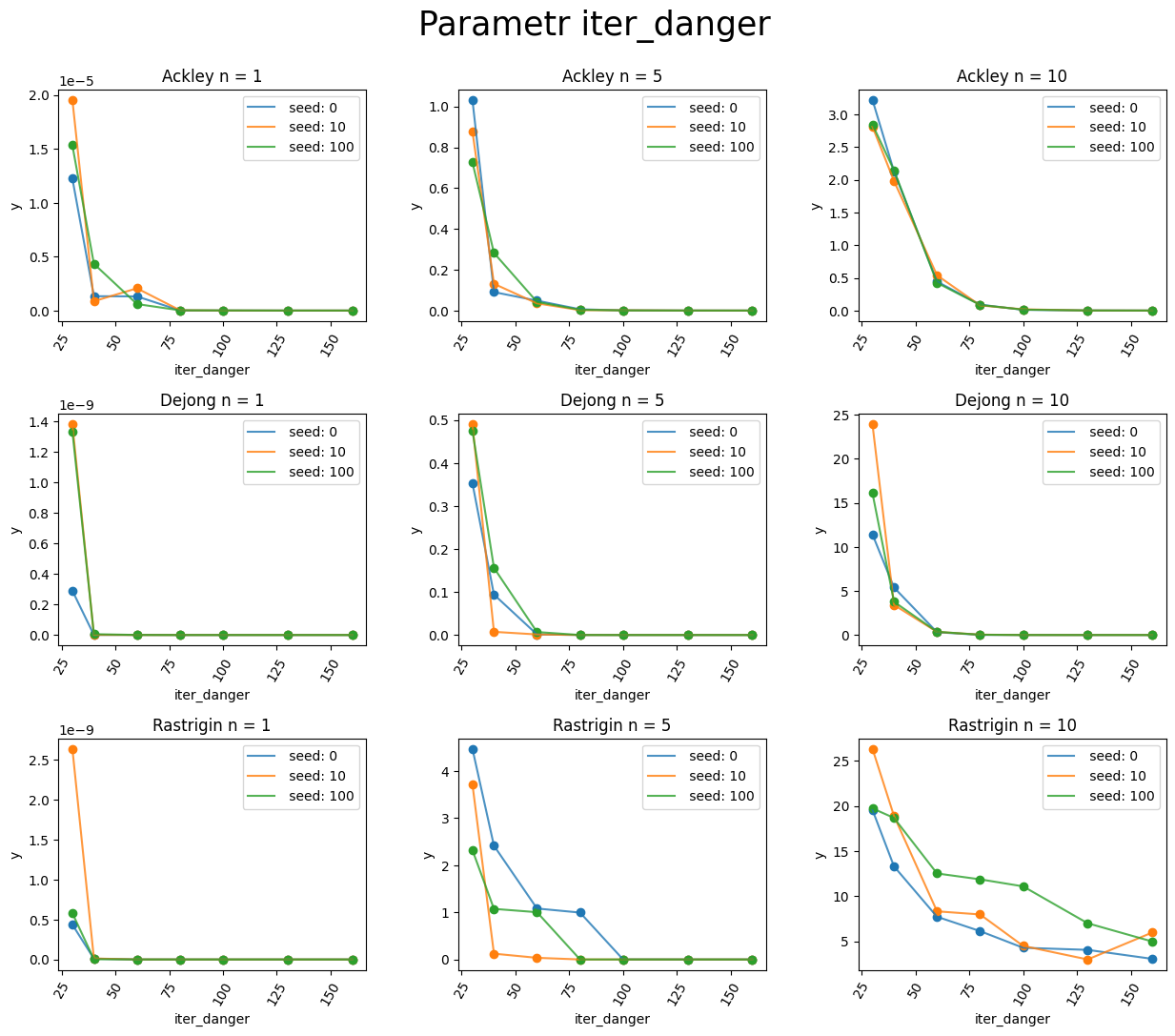
Analogicznie do swoich poprzedników wyniki pokazują, że istnieje pewien przedział najlepszych ustawień dla problemu. Biorąc pod uwagę wyniki z różnych zadań dobrym przedziałem startowym wydaje się być 0.5-1.

## C3



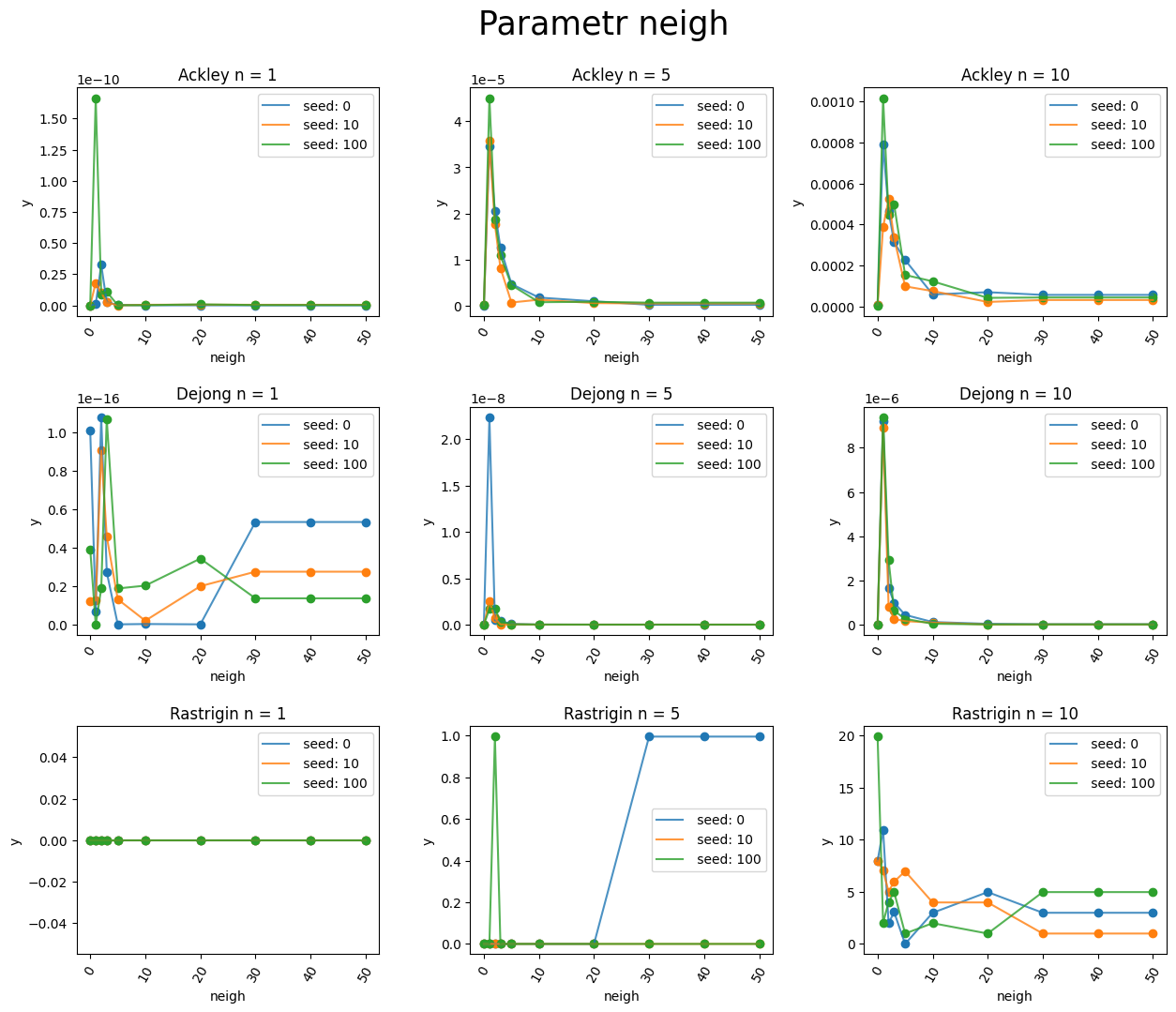
Wrażenia analogiczne do poprzednich parametrów „c”. Przedział wartości to 0.05 – 0.2.

## Iter\_danger



Parametr iter\_danger zdaje się nie mieć wartości „zbyt dużej”. W pewnym momencie ustawienie powoduje coś w stylu nasycenia i dalsze zwiększanie nie ma sensu. Na tym etapie nie ma zauważalnych wad dużych wartości parametru, więc można podejrzewać że po prostu nie jest opłacalne w tym problemie, ponieważ potencjalna wada jaką jest zwiększona podatność na minima lokalne nie jest wystarczająco groźna. Mówiąc inaczej, brak zjawiska pogorszenia się wyników może być spowodowany zbyt łatwym zadaniem dla algorytmu (inne parametry radzą sobie z problemem minimów).

## Sasiedztwo

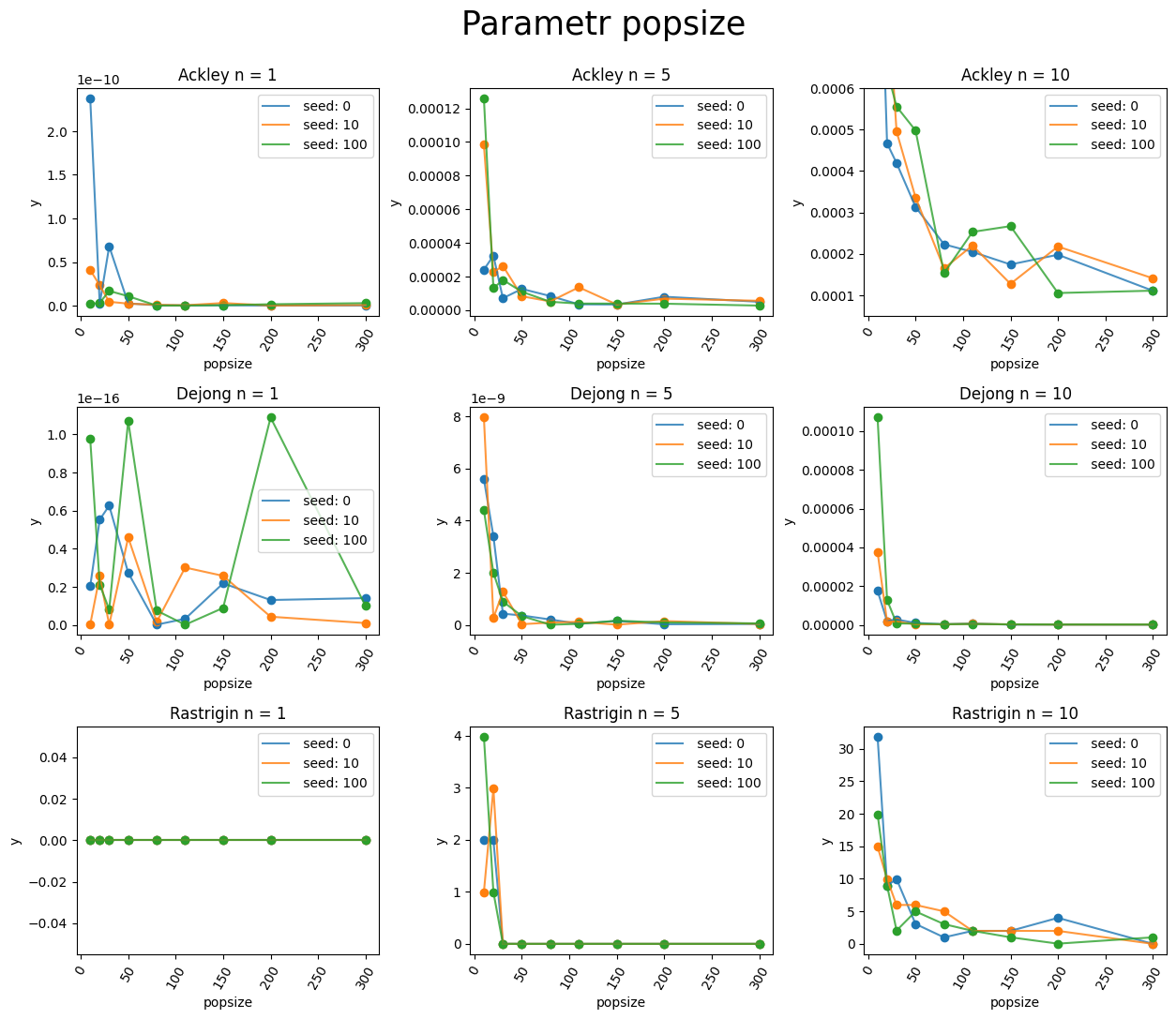


Sąsiedztwo podobnie do poprzedniego iter\_danger wydaje się w pewnym momencie nasycać. Zauważalne jest jednak, że po przekroczeniu tego progu wyniki całkowicie się stabilizują, co wskazuje na to, że tworzone grupy sąsiedztwa zaczynają być tak podobne między sobą (nachodzą na siebie), że populacja traci różnorodność.

Sam mechanizm sąsiedztwa jest jednak czasem opłacalny. Dla łatwiejszych zadań można zauważyć, że w ustawieniu 0 wartość jest bardzo niska, a często minimalna, jednak dla skomplikowanej funkcji 10 wymiarowej Rastrigina, algorytm lepiej sobie radzi z małym sąsiedztwem.

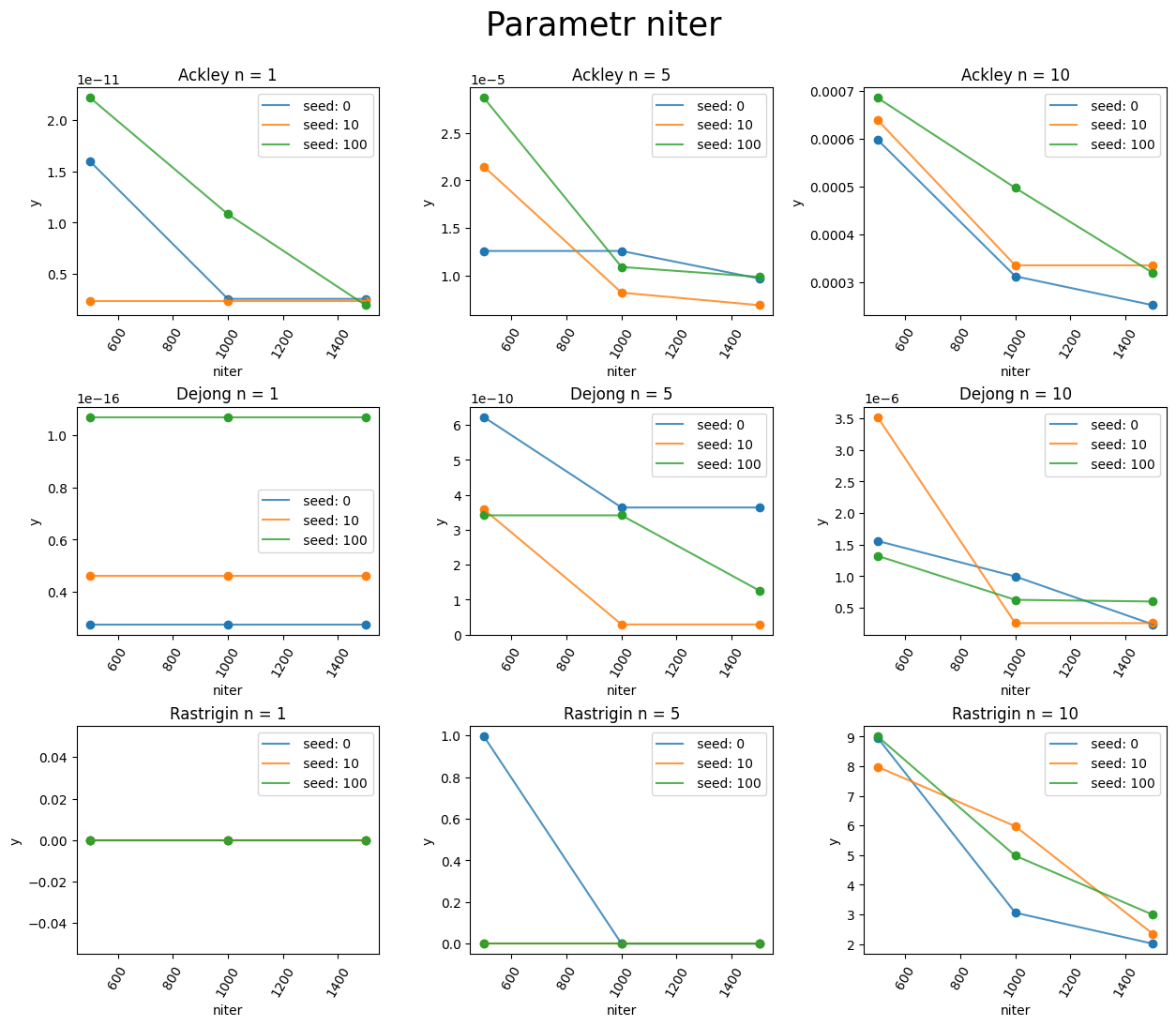
Ogólnie można powiedzieć, że sąsiedztwo ma sens tylko dla trudniejszych problemów, a nawet wtedy jest stosunkowo niewielkie – grupy kilku osobników.

## pop\_size



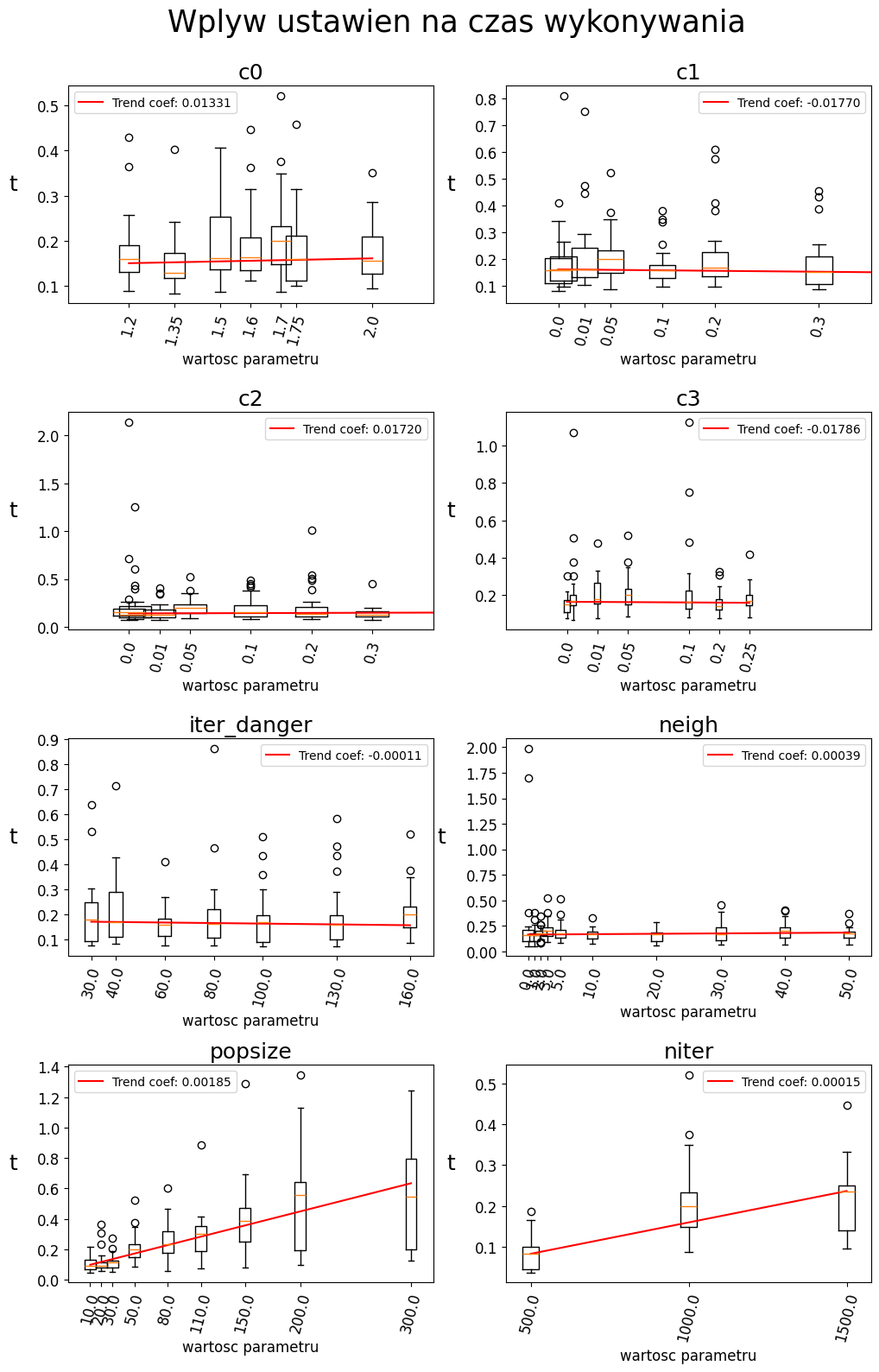
Liczebność populacji na ogół wpływa pozytywnie na działanie algorytmu. Nie ma zauważalnego punktu zbyt dużej populacji tak jak w przypadku algorytmu ewolucyjnego nawet dla najprostszych funkcji. Jedyną zaobserwowaną wadą większych populacji jest czas trwania algorytmu.

## niter



Liczba iteracji jest oczywista – im więcej tym lepiej, choć czasami poprawy są na tyle małe (lub dosłownie zerowe), że nie ma sensu kontynuowanie obliczeń. Wadą jest też wydłużenie czasu pracy.

# Wpływ parametrów na szybkość wykonania



Realnie, tylko parametry popsize oraz niter mają wpływ na czas wykonania, z czego pierwszy z nich wydaje się być bardziej znaczący.

W pozostałych przypadkach, mimo teoretycznie większego nachylenia prostej trendu realne zmiany w czasie nawet między skrajnymi ustawieniami nie mają zupełnie żadnego znaczenia z perspektywy użytkownika.

Ciekawym przypadkiem jest iter\_danger. Zwiększanie liczby iteracji przed restartem jest bardzo lekko proporcjonalne do spadku czasu trwania.