

**Politechnika Wrocławskaw
Wydział Informatyki i Telekomunikacji**

Kierunek: **ISA**
Specjalność: **IPS**

**PRACA DYPLOMOWA
MAGISTERSKA**

**Badanie wpływu metody próbkowania na wartości miar
przestrzeni rozwiązań dla wybranego problemu
optymalizacji dyskretnej**

Kacper Leśniański

Opiekun pracy
dr inż. Andrzej Gnatowski

Slowa kluczowe: optymalizacja, próbkowanie, przestrzeń rozwiązań, krajobraz adaptacyjny

WROCŁAW (2023)

*Serdecznie dziękuję mojemu
promotorowi, dr inż. Andrzejowi
Gnatowskiemu za nieocenioną
pomoc w realizacji tej pracy.*

Spis treści

1 Wprowadzenie	3
1.1 Cel pracy	4
1.2 Zakres Pracy	4
2 Przegląd literatury	5
2.1 Problemy optymalizacyjne	5
2.1.1 Problem komiwojażera	5
2.1.2 Problem kwadratowego przydziału	6
2.2 Cele badania przestrzeni rozwiązań	7
2.3 Próbkowanie przestrzeni rozwiązań	8
3 O przestrzeni rozwiązań	11
3.1 Sieć optimów lokalnych	11
3.2 Wierzchołki	11
3.3 Krawędzie	12
3.3.1 Basin-transition	12
3.3.2 Escape Edges	12
3.3.3 Perturbation edges	12
3.4 Struktury lejowe	13
3.5 Metryki przestrzeni rozwiązań	13
3.6 Operacja 2-exchange	15
4 Badania eksperymentalne	17
4.1 Zaimplementowane algorytmy	17
4.1.1 Próbkowanie dwufazowe	17
4.1.2 Snowball	17
4.1.3 Przegląd zupełny	21
4.2 Instancje testowe	21
4.3 Opis badań	23
4.3.1 Porównanie wartości metryk dla małych instancji	25
4.3.2 Badanie stabilności dla średnich instancji	42
4.3.3 Badanie stabilności dla dużych instancji	45
4.3.4 Korelacja wzajemna miar	48
4.3.5 Podsumowanie wyników	50
5 Opis implementacji	61
5.1 Program próbujący	61
5.1.1 Uruchamianie i parametry	61
5.1.2 Pliki wejściowe i wyjściowe	62

5.2	Program obliczający wartości miar	64
5.2.1	Uruchamianie i parametry	64
5.2.2	Pliki wejściowe i wyjściowe	64
5.3	Generatory instancji testowych	64
5.3.1	Uruchamianie i parametry	64
5.3.2	Pliki wyjściowe	65
5.4	Skrypty pomocnicze	65
5.5	Wykorzystane biblioteki	66
6	Podsumowanie	67
A	Wykresy	69
	Literatura	152
	Spis rysunków	155
	Spis tablic	160

Rozdział 1

Wprowadzenie

Analiza przestrzeni rozwiązań, znanej również jako krajobraz adaptacyjny, jest jednym z rzadziej rozważanych aspektów badań nad problemami optymalizacyjnymi. Przestrzeń ta pozwala na reprezentację problemu jako „krajobraz”, w którym każde rozwiązanie jest punktem na płaszczyźnie, a wysokość punktu wartością funkcji celu danego rozwiązania. Niektóre z cech tej przestrzeni, takie jak liczba optimów lokalnych i ich rozkład, obecność rozległych obszarów płaskich, licznych wzniesień lub wgłębień, mogą dostarczać informacji o właściwościach danego problemu, lub jego konkretnej instancji [15]. Dokładna reprezentacja przestrzeni rozwiązań, obejmująca wszystkie rozwiązania, wymagałaby przechowywania olbrzymiej ilości danych. Jednym z bardziej kompaktowych formatów jej reprezentacji jest sieć optimów lokalnych (ang. Local Optima Network, LON) — graf, którego wierzchołki reprezentują optima lokalne przestrzeni, a krawędzie relacje pomiędzy nimi. Wartości miar opisujące graf LON, takie jak wagи krawędzi, czy długości ścieżek między wierzchołkami, można wykorzystać m.in do predykcji jakości wyników uzyskiwanych przez techniki heurystyczne, dostrajania ich parametrów, lub do projektowania całkiem nowych algorytmów. Z powodu dużego rozmiaru przestrzeni rozwiązań zbudowanie grafu stanowiącego jej dokładną reprezentację jest trudne, czasochłonne i zwykle niemożliwe do wykonania w akceptowalnym czasie. W badaniach zamiast przeglądu zupełnego wykorzystuje się zatem próbkowanie, zakładając przy tym, że wartości miar uzyskane z obszaru nim objętego są w pewnym stopniu reprezentatywne dla całej przestrzeni. Próbkowanie to można wykonać na wiele różnych sposobów, a wybór algorytmu, oraz długość procesu może mieć istotny wpływ na wartości miar obliczonych na podstawie spróbowanej przestrzeni.

Większość najnowszych publikacji z dziedziny zajmuje się badaniami nad możliwościami praktycznego zastosowania analizy przestrzeni rozwiązań w różnych zadaniach. Niniejsza praca skupia się zaś głównie na samym aspekcie próbkowania. Uzyskane z eksperymentów dane pozwolą na sprawdzenie, w jaki sposób wartości miar zmieniają się wraz z postępem procesu próbkowania. Pozwoli to ocenić, które z nich zmieniają się w sposób na tyle przewidywalny, by możliwe było szacowanie ich wartości dla pełnej przestrzeni na podstawie danych z jej spróbowanego fragmentu. Wyniki ukażą również różnice pomiędzy sieciami LON budowanymi przez różne techniki próbkowania i ujawnią mocne oraz słabe strony poszczególnych algorytmów.

1.1 Cel pracy

W pracy zostaną przeprowadzone badania nad przestrzenią rozwiązań jednego z najbardziej znanych problemów optymalizacji dyskretnej — problemu komiwojażera.

Do poszczególnych celów pracy należą:

- Sprawdzenie dokładności, z jaką algorytmy próbkowania odwzorowują rzeczywistą przestrzeń rozwiązań,
- Zbadanie wpływu metody próbkowania, oraz przeznaczonego na nie nakładu obliczeniowego na wartości miar uzyskanej spróbkowanej przestrzeni,
- Zbadanie korelacji wzajemnej miar przestrzeni rozwiązań oraz wpływu metody próbkowania na wartości współczynników korelacji.

Cel zostanie zrealizowany poprzez wykonanie eksperymentów na kilkudziesięciu różnorodnych — pod względem liczby miast i ich ułożenia — instancjach problemu.

1.2 Zakres Pracy

Praca składa się z pięciu rozdziałów. W rozdziale pierwszym znajduje się krótkie wprowadzenie do tematyki pracy. Rozdział drugi zawiera przegląd najnowszych publikacji z dziedziny, opisane są w nim zastosowania analizy przestrzeni rozwiązań, oraz algorytmy próbkowania znalezione w literaturze. Rozdział trzeci poświęcony jest objaśnieniu niektórych terminów związanych z tematyką analizy przestrzeni rozwiązań. Znajduje się w nim również opis miar analizowanych w reszcie pracy. Rozdział czwarty zawiera opis eksperymentu. Przedstawione są w nim testowe instancje problemu i wykorzystane algorytmy próbkowania. W dalszej części rozdziału opisany jest przebieg eksperymentu oraz uzyskane wyniki. Dla małych instancji problemu, dla których możliwe jest wykonanie przeglądu zupełnego zostanie wykonane porównanie wartości miar uzyskanych z próbkowania do tych uzyskanych z przeglądu zupełnego. Pozwoli to na ocenę dokładności próbkowania. Dla kilkudziesięciu większych instancji wykonane zostanie długotrwałe próbkowanie przestrzeni rozwiązań, przy użyciu dwóch różnych algorytmów. Uzyskane dane posłużą do obliczenia wartości miar i przedstawieniu ich w relacji do wielkości spróbkowanej przestrzeni. Umożliwi to ocenę wpływu algorytmu na wartości poszczególnych miar oraz ich stabilność. Dane te posłużą również do obliczenia współczynników wzajemnej korelacji Pearsona poszczególnych miar. Współczynniki zostaną obliczone osobno dla dwóch algorytmów, co pozwoli na ocenę wpływu metody próbkowania na ich wartość. Rozdział piąty jest technicznym opisem implementacji i zawiera informacje o programach i skryptach napisanych na potrzeby pracy. Przedstawiono w nim krótkie instrukcje uruchomienia najważniejszych programów. Opisane zostały również wykorzystywane przez nie formaty plików. W rozdziale znajduje się także spis najważniejszych zewnętrznych bibliotek i narzędzi wykorzystanych w tworzeniu projektu. W rozdziale szóstym znajduje się krótkie podsumowanie całości pracy oraz wnioski wyciągnięte na podstawie uzyskanych wyników.

Rozdział 2

Przegląd literatury

W tym rozdziale przedstawiony jest przegląd najnowszych publikacji z dziedzin optymalizacji dyskretnej i badania przestrzeni rozwiązań. Pierwsza sekcja poświęcona jest wybranym problemom optymalizacji dyskretnej. Druga sekcja opisuje praktyczne zastosowania analizy przestrzeni rozwiązań znalezione w literaturze. Trzecia sekcja skupia się na przedstawionych w cytowanych pracach algorytmach próbkoowania.

2.1 Problemy optymalizacyjne

Sekcja ma na celu przybliżyć problemy optymalizacyjne stanowiące obiekt badań tej pracy lub prac cytowanych w przeglądzie literatury. Są to: problem kwadratowego przypisania oraz problem komiwojażera. Zostaną przedstawione opisy problemów oraz kilka prac dotyczących ich rozwiązywania.

2.1.1 Problem komiwojażera

Problem komiwojażera (ang. Traveling Salesman Problem, TSP) jest znanym problemem optymalizacyjnym sformułowanym w następujący sposób: mając do dyspozycji listę miast i odległości między nimi, należy odnaleźć najkrótszą ścieżkę przechodzącą przez wszystkie miasta zaczynającą i kończącą się w ustalonym punkcie. Problem ten jest problemem NP-trudnym i z tego powodu do rozwiązywania większych jego instancji konieczne jest stosowanie algorytmów heurystycznych.

Istnieje wiele różnych metod rozwiązywania problemu komiwojażera. Do metod dokładnych zalicza się metody programowania dynamicznego, oparte na memoizacji oraz algorytmy typu *Branch and Bound*. Ze względu na nieefektywność metod dokładnych dla większych instancji problemu, zwykle wykorzystuje się metody przybliżone. Oprócz „uniwersalnych” heurystyk, takich jak symulowane wyżarzanie czy przeszukiwanie Tabu, stosuje się również metody stworzone specjalnie dla tego problemu. Do takich metod należą m.in. 2-opt, polegający na „rozplataniu” skrzyżowanych ścieżek na trasie oraz Furthest Insertion, budujący trasę poprzez dodawanie do niej miast położonych najdalej od tych już odwiedzonych [10]. Do jednej z najlepszych heurystyk należy algorytm Lin-Kerninghan. Opiera się ona na operacji $\lambda - opt$ - operacji przypominającej 2-opt, ale dokonującą zmiany λ krawędzi. Algorytm z każdą iteracją sprawdza możliwe zamiany od 2 do λ krawędzi, poszukując takiej, która najbardziej skróci ścieżkę. Proces może być powtarzony kilkukrotnie z innymi warunkami początkowymi [8]. Rozwiążanie początkowe może być losowe, lub wybrane inną metodą.

Metody rozwiązywania problemu komiwojażera są ciągle rozwijane i nadal powstają nowe algorytmy. Obecnie często wykorzystuje się metody uczenia maszynowego. Przykładem jest praca *H-TSP: Hierarchically Solving the Large-Scale Travelling Salesman Problem* [17]. W pracy wykorzystano hierarchiczny algorytm oparty o uczenie ze wzmacnianiem. Konstrukcję trasy oparto o metodę dziel i rządź - algorytm najpierw znajduje najkrótsze ścieżki dla mniejszych podzbiorów zbioru miast, a następnie łączy je w dłuższe trasy. Badania autorów wykazały, że otrzymane tą metodą wyniki są porównywalne z wynikami najlepszych znanych heurystyk (takich jak LKH3), a zarazem metoda ta cechuje się znacznie krótszym czasem obliczeń, w niektórych przypadkach krótszym o dwa rzędy wielkości. Innym przykładem wykorzystania uczenia maszynowego jest praca *NeuroLKH: Combining Deep Learning Model with Lin-Kernighan-Helsgaun Heuristic for Solving the Traveling Salesman Problem* [23]. W pracy wykorzystano sieć neuronową do przyspieszenia procesu tworzenia zbioru krawędzi-kandydatów przez algorytm Lin-Kernighan-Helsgaun. Opracowana modyfikacja, nazwana NeuroLKH zdołała porównana z oryginalnym algorytmem poprzez porównanie wyników otrzymanych po takim samym czasie działania. W każdym przetestowanych przypadków NeuroLKH znalazła znacznie lepsze rozwiązanie.

Istnieją również metody inspirowane naturą. W pracy *Ant colony optimization for traveling salesman problem based on parameters optimization* opracowano algorytm mrówkowy z automatycznym doborem wartości części parametrów. Dobór wartości dwóch najważniejszych parametrów, kontrolujących wpływ feromonów i innych czynników na wybór ścieżki, oparto o metodę wyszukiwania organizmów symbiotycznych (ang. symbiotic organisms search). Algorytm porównano z innymi algorytmami mrówkowymi, w porównaniu do nich osiągnięto lepsze lub podobne wyniki w mniejszej liczbie iteracji. W pracy *Discrete sparrow search algorithm for symmetric traveling salesman problem* [25] przedstawiono algorytm inspirowany zachowaniem wróblów. Oryginalny algorytm Sparrow Search zmodyfikowano poprzez dodanie etapów mutacji Gaussowskiej, oraz lokalnej optymalizacji 2opt. Algorytm porównano z innymi metodami (D-GWO, DSFLA, DSMO), wykorzystując w tym celu instancje ze zbioru TSPLIB. Nowy algorytm uzyskał lepsze wyniki i krótszy czas konwergencji.

2.1.2 Problem kwadratowego przydziału

Problem kwadratowego przydziału (ang. Quadratic Assignment Problem, QAP) jest problemem optymalizacyjnym sformułowanym w następujący sposób: należy przydzielić N placówek do N lokalizacji tak, aby całkowity koszt transportu towarów pomiędzy nimi był jak najmniejszy. Dane są dwie macierze. Pierwsza macierz opisuje dystans pomiędzy poszczególnymi lokalizacjami, natomiast druga macierz zawiera liczby reprezentujące ilość towaru do przetransportowania między danymi placówkami. Koszt przetransportowania towaru jest definiowany jako odległość pomnożona przez ilość towaru. Całkowity koszt to suma wszystkich kosztów transportu między placówkami [11].

Problem kwadratowego przydziału jest problemem NP-trudnym i do rozwiązywania większych instancji konieczne jest zastosowanie algorytmów heurystycznych. Wśród nich można wymienić przeszukiwanie tabu, algorytmy mrówkowe i genetyczne [1]. W badaniu przestrzeni rozwiązań stosuje się zaś proste algorytmy przeszukiwania lokalnego typu *hillclimb*. Oprócz wymienionych metod istnieją również podejścia hybrydowe. Przykładem jest algorytm przedstawiony w pracy *A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem* [24], który łączy w sobie algorytm genetyczny z przeszukiwaniem Tabu. Populacja losowo wygenerowanych rozwiązań poddawana jest operacjom krzyżowania i mutacji, a następnie najlepsze rozwiązanie

poddawane jest dalszej optymalizacji algorytmem przeszukiwania Tabu. Algorytm porównano z innymi metodami (w tym DBA o WAITS) i osiągnięto od nich lepsze wyniki. W pracy *Memetic search for the quadratic assignment problem* [2] wykorzystano algorytm memetyczny korzystający z metody optymalizacji lokalnej BLS. Wykonane na 133 instancjach ze zbioru QAPLIB badania wykazały, że jest w stanie znaleźć najlepsze rozwiązania dla 130 z nich w stosunkowo krótkim czasie. W pracy *Artificial bee colony optimization for the quadratic assignment problem* [6] przedstawiono metodę inspirowaną zachowaniem pszczelego roju, wykorzystującą algorytm *Robust Taboo Search* w fazie „zbierania nektaru”. Zaletą algorytmu jest możliwość jego zrównoleglenia, zapewniająca wysoką skalowalność. Metoda dobrze spisała się przy rozwiązywaniu instancji testowych QAPLIB, znajdując optymalne rozwiązania w większości przypadków.

2.2 Cele badania przestrzeni rozwiązań

Analiza przestrzeni rozwiązań pozwala na lepsze poznanie niektórych cech problemu, a także zbadanie, w jakim stopniu cechy te zależne są od rodzaju badanej instancji. Przedmiotem badań z tej dziedziny najczęściej są znane problemy NP-trudne, takie jak problem komiwojażera czy problem kwadratowego przydziału. Jednym z proponowanych zastosowań analizy przestrzeni rozwiązań jest wykorzystanie jej do wyboru najlepszego algorytmu heurystycznego dla danej instancji problemu. W pracy *Local Optima Networks in Solving Algorithm Selection Problem for TSP* [3] sieci optimów lokalnych wygenerowane na podstawie próbkowania przestrzeni rozwiązań wykorzystano do nauczenia modeli regresji, których zadaniem było przewidzenie, który algorytm heurystyczny da lepsze wyniki dla danej instancji problemu. Badanie wykazało, że analiza przestrzeni rozwiązań może zostać z powodzeniem wykorzystana w tym celu. Problemem pozostaje natomiast długi czas trwania próbkowania przestrzeni, zwykle dłuższy niż czas działania samego algorytmu optymalizacji. W pracy *Mapping the global structure of TSP Fitness landscapes* [14] zbadano przestrzeń rozwiązań dla różnych instancji problemu komiwojażera. Zauważono, że instancje wygenerowane losowo zwykle mają mniejszą neutralność i mniej globalnych optimów od instancji rzeczywistych. Zaobserwowano również, że sposób rozłożenia miast (w klastrach, równomierny) wpływa na wzajemną korelację wartości różnych miar.

W ostatnich latach w podobny sposób przeprowadzono analizy przestrzeni konfiguracji wieloparametrowych algorytmów optymalizacyjnych. W pracy *Understanding Parameter Spaces using Local Optima Networks: A Case Study on Particle Swarm Optimization* [4] wykorzystano sieci lokalnych optimów, oraz pochodne struktury CMLON do analizy i wizualizacji przestrzeni parametrów algorytmu roju częstek. Analiza wykazała istnienie dużej ilości lokalnych optimów, niską neutralność, oraz istnienie wielu ścieków (ang sinks) nie znajdujących się w optimum globalnym. Sugeruje to, że naiwne metody dobierania parametrów mogą łatwo doprowadzić do suboptimalnej konfiguracji i w efekcie nie otrzymania najlepszych wyników. Podobne badanie wykonano dla przestrzeni parametrów procesu AutoML w pracy *Understanding AutoML Search Spaces with Local Optima Networks* [18]. AutoML jest procesem automatyzacji konfiguracji procesów (ang. pipelines) uczenia maszynowego obejmującym m. in. wybór zastosowanego przetwarzania wstępnego, właściwego algorytmu uczenia, oraz jego hiperparametrów. W pracy zbadano przestrzeń konfiguracji AutoML dla zadania klasyfikacji. Jako funkcję celu przyjęto uśrednioną wartość metryki F-score dla danej konfiguracji procesu. Wartości F-score uzyskano poprzez przetestowanie procesu na kilku zbiorach danych. Dla każdej z badanych przestrzeni konfiguracji utworzono trzy sieci LON. Każda z sieci była oparta o inny model

krawędzi. Przeszukano wszystkie możliwe rozwiązania, ale ze względu na złożoność obliczeniową sieć LON budowano poprzez przeglądanie ograniczonego sąsiedztwa każdego z wierzchołków. Badanie wykonano dla kilku wielkości sąsiedztwa — 20, 30, 50 i 100 sąsiadów. Zauważono wpływ modelu krawędzi na powstałą sieć — w sieciach z krawędziami typu basin-transition nie zauważono obecności ścieków, natomiast istniało wiele źródeł. Sieci oparte o escape edges miały z kolei dużo ścieków i mało źródeł. W przypadku perturbation edges w sieci nie było ścieków ani źródeł niezależnie od rozmiaru sąsiedztwa. Zauważono, że w badanych przypadkach optima globalne skupiały się w pewnym rejonie przestrzeni rozwiązań, w niewielkiej od siebie odległości. Nie były rozłożone równomierne. Sugeruje to, że w problemie konfiguracji procesu uczenia maszynowego, najlepsze ze wszystkich możliwych konfiguracji niewiele się od siebie różnią.

2.3 Próbkowanie przestrzeni rozwiązań

Przegląd zupełny przestrzeni rozwiązań problemów trudnych obliczeniowo jest w praktyce niemożliwy, poza bardzo małymi instancjami problemu. Z tego powodu analizę tą przeprowadza się na części przestrzeni zbadanej w procesie próbkowania.

W tym miejscu pojawiają się pytanie: w jaki sposób próbować przestrzeń, aby cechy jej zbadanego fragmentu jak najlepiej odzwierciedlały cechy całej przestrzeni?

Próbkowanie przestrzeni rozwiązań wykonuje się zwykle poprzez zastosowanie pewnej odmiany iteracyjnego przeszukiwania lokalnego (ang. ILS — Iterated Local Search) [15], a wyniki zapisuje się w postaci sieci optimów lokalnych. Najczęściej stosowane algorytmy próbkowania oparte o ILS można podzielić na trzy kategorie: Próbkowanie dwufazowe, *Markov-chain* oraz *Snowball*.

Próbkowanie dwufazowe składa się z dwóch faz — najpierw próbowane są wierzchołki poprzez generowanie losowego rozwiązania, a następnie wykorzystaniu algorytmu optymalizacji lokalnej do znalezienia najbliższego lokalnego optimum. Następuje po tym wylosowanie nowego punktu i powtórzenie procedury. Druga faza to faza próbkowania krawędzi, polegająca na poddaniu znalezionych wcześniej rozwiązań operacji perturbacji, a następnie wykonaniu optymalizacji lokalnej. Jeśli otrzymane w ten sposób rozwiązanie jest innym lokalnym optimum znajdującym się w zbiorze wierzchołków, to dodawana jest odpowiednia krawędź. Metoda ta po raz pierwszy została zaprezentowana w pracy *Data-Driven Local Optima Network Characterization of QAPLIB Instances* [9] do analizy problemu kwadratowego przydziału. W pracy *Local Optima Networks in Solving Algorithm Selection Problem for TSP* [3] algorytm oparty o tę samą metodę został wykorzystany do badania przestrzeni problemu komiwojażera. Zastosowana tam odmiana wykorzystuje algorytm 2-opt do optymalizacji lokalnej i procedurę *2-exchange* jako operację perturbacji.

Metoda *Markov-chain* została zaprezentowana w pracy *Perturbation Strength and the Global Structure of QAP Fitness Landscapes* [13]. Zaczyna się od wybrania losowego rozwiązania i jego optymalizacji. Następnie otrzymane optimum lokalne zostaje poddane operacji perturbacji i otrzymywane w ten sposób rozwiązanie staje się nowym punktem startowym. Procedura jest powtarzana przez określoną liczbę iteracji, z każdą iteracją zapisywane są informacje o nowych krawędziach i wierzchołkach. Metoda została wykorzystana w pracy *Mapping the global structure of TSP fitness landscapes* [14] do badania przestrzeni problemu komiwojażera. Do optymalizacji lokalnej wykorzystany został algorytm Lin-Kerninghan, a jako operacja perturbacji procedura *Double-bridge*.

Metoda *Snowball* składa się z dwóch etapów wykonywanych naprzemiennie. Pierwszy polega na kilkukrotnym poddaniu rozwiązania początkowego perturbacji w celu uzy-

skania sąsiednich rozwiązań, a następnie ich optymalizacji. Procedura jest rekurencyjnie powtarzana dla znalezionych w ten sposób lokalnych optimów aż do osiągnięcia pewnej z góry ustalonej głębokości. Jedno z lokalnych optimów sąsiadujących z rozwiązaniem początkowym zostaje wybrane jako punkt startowy kolejnej iteracji. Dodatkowo w pamięci przechowywana jest lista rozwiązań początkowych — jeśli rozwiązanie już się w nim znajduje, to nie może być wybrane ponownie. Jeśli nie istnieje rozwiązanie sąsiednie spełniające ten warunek, za kolejne rozwiązanie początkowe przyjmowane jest lokalne optimum otrzymane z optymalizacji rozwiązania losowego. Algorytm *Snowball* wywodzi się z technik wykorzystywanych w badaniach z dziedziny socjologii, a w kontekście badania przestrzeni rozwiązań został zaprezentowany po raz pierwszy w pracy *Sampling Local Optima Networks of Large Combinatorial Search Spaces: The QAP Case* [22].

W pracy *Clarifying the Difference in Local Optima Network Sampling Algorithms* [19] zostało przeprowadzone statystyczne porównanie algorytmów próbkowania *Snowball* oraz *Markov-chain*. Eksperyment polegał na spróbkowaniu 30 instancji problemu kwadratowego przydziału, a następnie użyciu zebranych danych do stworzenia modeli regresji przewidujących jakość rozwiązań, które zostaną uzyskane przez algorytm optymalizacji uruchomiony na instancji. Wybranymi algorytmami heurystycznymi były *Robust Taboo Search* Taillarda oraz *Improved ILS* Stützla. Z zebranych danych utworzono grafy LON i zbadaano takie właściwości, jak średnia wartość funkcji celu, średni stopień wychodzący wierzchołków w grafie, promień grafu, liczba lokalnych optimów i liczba krawędzi. Stworzono modele regresji typu liniowego oraz lasu losowego. Obliczono wartości korelacji pomiędzy właściwościami przestrzeni rozwiązań a jakością rozwiązań zwracanych przez algorytmy optymalizacji. Badania wykazały, że dane pozyskane z próbkowania *Markov-chain* były w większym stopniu skorelowane z jakością rozwiązań algorytmów optymalizacji, a modele regresji utworzone na ich podstawie dokonywały lepszej predykcji niż te utworzone na podstawie danych ze *Snowball*. Z kolei *Snowball* okazał się bardziej przewidywalny i łatwiejszy w doborze parametrów.

Podobne badanie przeprowadzono w pracy *Inferring Future Landscapes: Sampling the Local Optima Level* [20], tym razem na większej liczbie instancji — 124 instancji ze zbioru QAPLIB i dodatkowych 60 instancji o rozmiarze N=11. Dla tych dodatkowych instancji, oprócz próbkowania wykonano również przegląd zupełny. Zauważono, że algorytm *Snowball* produkuje gęstszą sieć od algorytmu *Markov-chain*. Dostrzeżono wadę algorytmu *Snowball* — duży wpływ parametrów próbkowania na wartości metryk opartych o gęstość i wzory połączeń krawędzi, oraz metryk opisujących struktury lejowe. Stworzono modele regresji przewidujące jakość rozwiązań generowanych przez heurystyki *Taboo Search* i *ILS*. Model przewidujący odpowiedź *ILS* oparty o dane z algorytmu *Snowball* był nieco lepszy od modelu opartego o dane z algorytmu *ILS*. Między predykcjami modeli przewidujących odpowiedź algorytmu *TS* nie było znaczcej różnicy. Wśród wszystkich czterech testowanych modeli cechą przestrzeni będącą najlepszym predyktorem okazała się średnia wartość funkcji dopasowania (ang. mean fitness). Porównanie spróbkowanych sieci LON małych instancji z sieciami uzyskanymi poprzez przegląd zupełny wykazało, że zarówno *Markov-chain* jak i *Snowball* dobrze aproksymują liczbę wierzchołków i krawędzi w sieci. *Snowball* odnalazł prawie wszystkie optima lokalne obecne w pełnej sieci. *Markov-chain* znalazł ich mniej, ale nadal dawał zadowalający wynik.

Z innych prac z dziedziny warto wymienić *First-improvement vs. Best-improvement Local Optima Networks of NK Landscapes* [16], w którym porównano algorytmy wstępujące (hillclimb) typu *best-improvement* i *first-improvement* w próbkowaniu przestrzeni NK. Wykorzystanie algorytmu typu *first-improvement* prowadziło do powstania gęstszej, bardziej kompletnej sieci optimów lokalnych. Pętle obecne w sieci miały mniejszą wagę

w przypadku algorytmu *first-improvement* co sugeruje, że łatwiej wychodzi z optimum lokalnego. Dodatkową zaletą tego algorytmu jest krótszy czas wykonywania, spowodowany brakiem konieczności przeglądania za każdym razem całego sąsiedztwa rozwiązania początkowego.

Z kolei w pracy *How Perturbation Strength Shapes the Global Structure of TSP Fitness Landscapes* [12] zbadano wpływ siły perturbacji (stopnia, w jakim operacja perturbacji zmienia rozwiązanie początkowe) na właściwości spróbkowanej przestrzeni. Wybranym algorytmem próbującym był algorytm typu *Markov-chain* pochodzący z artykułu [14]. Operacją perturbacji tego algorytmu jest procedura double-bridge. Badania przeprowadzono na 180 instancjach o znanych rozwiązaniach optymalnych, uzyskanych przy pomocy generatora DIMACS TSP . Celem było znalezienie takiego parametru k - siły perturbacji - aby uzyskać jak największy wskaźnik sukcesu. Za wskaźnik sukcesu przyjęto stosunek liczby przebiegów algorytmu, które znalazły globalne optimum, do wszystkich przebiegów. Nie znaleziono uniwersalnej najlepszej wartości parametru k. Zauważono jednak, że niższa siła perturbacji sprawdzała się dla instancji, w których miasta rozmieszczone są w klastrach. Dla instancji z miastami rozmieszczonymi równomiernie zwiększenie wartości k powodowało zmniejszenie ilości suboptimalnych lejów w przestrzeni. Zaobserwowano również, że instancje z miastami ułożonymi w klastrach były generalnie prostsze do rozwiązania, niż instancje z rozkładem równomiernym, a także, że rozkład optimów lokalnych w przestrzeni rozwiązań ma większy wpływ na trudność znalezienia optymalnego rozwiązania dla danej instancji, niż ich ilość.

Rozdział 3

O przestrzeni rozwiązań

W rozdziale przedstawiono definicje przestrzeni rozwiązań oraz sieci optimów lokalnych. Opisane zostały również miary analizowane w pozostałą części pracy, oraz operacja 2-exchange stanowiąca operator permutacji w wykorzystanych algorytmach próbkoowania.

Przestrzeń rozwiązań (Krajobraz adaptacyjny, ang. Fitness Landscape) jest pojęciem wywodzącym się z biologii ewolucyjnej. W kontekście biologicznym jest to model opisujący relację między genotypem i fenotypem organizmów, a ich przystosowaniem (ang. fitness), które jest miarą opisującą sukces reprodukcyjny [7].

W kontekście optymalizacji Fitness landscape to trójka (S, V, f) , gdzie:

- S jest zbiorem wszystkich możliwych rozwiązań - przestrzenią przeszukiwania,
- V jest funkcją przypisującą każdemu rozwiązaniu $s \in S$ zbiór sąsiadów $V(s)$,
- f jest funkcją $f : S \rightarrow \mathbb{R}$ przypisującą danemu rozwiązaniu wartość przystosowania - zwykle jest to wartość funkcji celu dla danego rozwiązania.

3.1 Sieć optimów lokalnych

Sieć optimów lokalnych (ang. Local Optima Network, LON) jest formą reprezentacji przestrzeni rozwiązań zaprezentowaną po raz pierwszy w artykule *Complex-network analysis of combinatorial spaces: The NK landscape case* [21], a następnie rozwiniętym w pracy *Local Optima Networks: A New Model of Combinatorial Fitness Landscapes* [15].

Jest to graf $G = (N, E)$ przedstawiający występujące w przestrzeni rozwiązań optima lokalne (zbiór wierzchołków N) i relacje między nimi (zbiór krawędzi E).

3.2 Wierzchołki

Wierzchołki w sieci optimów lokalnych reprezentują optima lokalne w przestrzeni rozwiązań. Do optimów zaliczamy minima i maksima; w problemach optymalizacyjnych zazwyczaj poszukujemy tych pierwszych. Minimum lokalne to takie rozwiązanie s , w którego sąsiedztwie $V(s)$ nie znajduje się żadne rozwiązanie x , dla którego $f(x) < f(s)$. Każde rozwiązanie w przestrzeni rozwiązań można przyporządkować do pewnego lokalnego minimum. Aby znaleźć lokalne minimum $n \in N$, do którego „prowadzi” dane rozwiązanie $s \in S$, wykonuje się lokalną optymalizację z tym rozwiązaniem przyjętym jako punkt startowy. W dalszej części pracy takie przyporządkowanie będzie oznaczane jako $h(s) \rightarrow n \in N$. Wierzchołkom w sieci LON można przypisać wagę równą wartości funkcji celu w danym optimum lokalnym.

3.3 Krawędzie

Krawędzie w sieci lokalnych optimów mogą być zdefiniowane na jeden z kilku sposobów. W literaturze [15] [18] zostały opisane trzy różne modele: *basin-transition*, *escape edges* i *perturbation edges*.

3.3.1 Basin-transition

Basen przyciągania optimum lokalnego n jest zdefiniowany jako zbiór:

$$b_i = \{s \in S \mid h(s) = n\}$$

Rozmiarem basenu jest liczność tego zbioru oznaczana jako $|b_i|$. Dla każdej pary rozwiązań w przestrzeni można obliczyć prawdopodobieństwo przejścia z jednego rozwiązania do drugiego $p(s \rightarrow s')$. Dla rozwiązań reprezentowanych permutacją o długości M , prawdopodobieństwo takie wynosi:

$$\begin{aligned} p(s \rightarrow s') &= \frac{1}{M(M-1)/2}, \quad \text{jeżeli } s' \in V(s), \\ p(s \rightarrow s') &= 0, \quad \text{jeżeli } s' \notin V(s), \end{aligned}$$

Mając informacje o prawdopodobieństwach przejścia między poszczególnymi rozwiązaniami można obliczyć prawdopodobieństwo przejścia od rozwiązania s do dowolnego rozwiązania należącego do basenu b_j :

$$p(s \rightarrow b_j) = \sum_{s' \in b_j} p(s \rightarrow s')$$

Całkowite prawdopodobieństwo przejścia z basenu jednego optimum lokalnego do drugiego wynosi więc:

$$p(b_i \rightarrow b_j) = \frac{1}{|b_i|} \cdot \sum_{s \in b_i} p(s \rightarrow b_j)$$

To całkowite prawdopodobieństwo stanowi wagę krawędzi w grafie.

Krawędzie typu *basin-transition* tworzą gęstszą sieć od krawędzi typu *escape edges*.

3.3.2 Escape Edges

Escape Edges zdefiniowane są przy pomocy funkcji dystansu d zwracającej najmniejszą odległość między dwoma rozwiązaniami, oraz liczby całkowitej D . Krawędź e_{ij} między lokalnymi optimami n_i i n_j istnieje, jeśli istnieje rozwiązanie s takie, że:

$$d(s, n_i) \leq D \wedge h(s) = n_j \tag{3.1}$$

Wagą takiej krawędzi jest liczba rozwiązań spełniających powyższy warunek.

3.3.3 Perturbation edges

W tym modelu wagę krawędzi pomiędzy lokalnymi optimami n_i i n_j uzyskuje się poprzez kilkukrotne wykonanie operacji perturbacji na n_i , a następnie optymalizacji lokalnej otrzymanego rozwiązania. Liczba przypadków, w których po optymalizacji otrzymujemy rozwiązanie n_j podzielona przez liczbę prób stanowi wagę krawędzi.

$$w_{ij} = \frac{|\{opt(pert(n_i)) = n_j\}|}{n_{trials}}$$

3.4 Struktury lejowe

W sieci optimów lokalnych możemy wyróżnić sekwencje złożone z optimów lokalnych, których wartość przystosowania jest niemalejąca. Sekwencje te zwane są sekwencjami monotonicznymi [14]. Sekwencje monotoniczne zmierzające do tego samego ścieku (ang. sink, wierzchołek bez krawędzi wychodzących) tworzy strukturę zwaną lejem. Wspomniany ściek stanowi spód leja (ang. funnel bottom), a liczba wierzchołków zawartych w tej strukturze określa jej rozmiar. Ponadto w przestrzeni rozwiązań można wyróżnić lej pierwszorzędny (ang. primary funnel), kończący się w globalnym optimum i leje drugorzędne, kończące się w optimach lokalnych. Jeden wierzchołek w grafie może przynależeć jednocześnie do wielu lejów.

Leje w sieci optimów lokalnych można zidentyfikować poprzez usunięcie z grafu krawędzi prowadzących od lepszych rozwiązań do gorszych, zidentyfikowanie ścieków, odwrócenie krawędzi i wykonanie przeszukiwania w głąb lub wszerz w celu odnalezienia wierzchołków należących do leja.

3.5 Metryki przestrzeni rozwiązań

W tej sekcji opisane zostaną metryki przestrzeni rozwiązań wykorzystane w tej pracy. Definicje niektórych miar zostały oparte o dokumentację biblioteki *igraph* [5].

- **num_sinks** — liczba ścieków. Ściek (ang. sink) jest wierzchołkiem grafu nie posiadającym krawędzi wychodzących. Pętle nie są uwzględniane.
- **num_sources** — liczba źródeł. Źródło (ang. source) jest wierzchołkiem grafu nie posiadającym krawędzi wchodzących. Pętle nie są uwzględniane.
- **num_sub sinks** — Liczba wierzchołków, które nie posiadają krawędzi wychodzących do rozwiązań o niższej wartości funkcji celu.
- **edge_to_node** — stosunek liczby krawędzi do liczby wierzchołków w grafie.
- **avg_fitness** — średnia wartość funkcji celu lokalnych optimów w sieci.
- **distLO** — średnia odległość rozwiązań od rozwiązania z najniższą wartością funkcji celu. Odległość jest zdefiniowana jako odwrotność wagi krawędzi łączającej rozwiązanie z rozaniem najlepszym. Rozwiązania nie połączone krawędzią z najlepszym rozwiązaniem nie sąbrane pod uwagę.
- **conrel** — Stosunek liczby rozwiązań połączonych krawędzią z najlepszym rozwiązaniem do liczby pozostałych rozwiązań.
- **avg_out_degree, max_out_degree** — średni i maksymalny stopień wychodzący rozwiązań w grafie. Stopień wychodzący wierzchołka to liczba wychodzących z niego krawędzi. Pętle nie sąbrane pod uwagę.
- **avg_in_degree, max_in_degree** — średni i maksymalny stopień wchodzący rozwiązań w grafie. Stopień wchodzący wierzchołka to liczba wchodzących do niego krawędzi. Pętle nie sąbrane pod uwagę.

- **assortativity** — współczynnik różnorodności grafu. Różnorodność (ang. assortativity) grafu skierowanego jest zdefiniowana następującym wzorem:

$$\text{assortativity} = \frac{1}{\sigma_o \sigma_i} \sum_{(j,k) \in E} \deg(j) \cdot \deg(k) \cdot (e_{jk} - q_j^o q_k^i) \quad (3.2)$$

Gdzie:

- e_{ij} — część krawędzi łączących wierzchołki i i j w stosunku do liczby wszystkich krawędzi (ułamek z zakresu 0 do 1),
 - $q_i^o = \sum_{j \in V} e_{ij}$
 - $q_i^i = \sum_{j \in V} e_{ji}$
 - σ_o — odchylenie standardowe q^o
 - σ_i — odchylenie standardowe q^i
- **clustering_coeff** — współczynnik klasteryzacji grafu (ang. clustering coefficient, transitivity) opisuje prawdopodobieństwo istnienia połączenia pomiędzy sąsiednimi wierzchołkami. Współczynnik klasteryzacji opisany jest wzorem:

$$cc = \frac{N_{\text{triangles}}}{N_{\text{triples}}}$$

Gdzie $N_{\text{triangles}}$ to liczba trójkątów w grafie, a N_{triples} to liczba połączonych trójkę.

Trójkąt to trójka wierzchołków (x, y, z) taka, że $(x, y), (y, z), (x, z) \in E$.

Połączona trójka to trójka wierzchołków (x, y, z) taka, że $(x, y), (y, z) \in E$.

- **density** — gęstość grafu. Jest to stosunek liczby krawędzi w grafie do maksymalnej liczby krawędzi, jaka mogłaby istnieć w tym grafie. Dana jest wzorem:

$$\text{density} = \frac{|E|}{|V|(|V| - 1)}$$

- **largest_clique_size** — rozmiar największej kliki. Klika w grafie jest podzbiorem zbioru wierzchołków, w którym wszystkie wierzchołki są sąsiednie — istnieje krawędź pomiędzy każdą parą wierzchołków należących do zbioru.
- **reciprocity** — Wzajemność. Wzajemność jest miarą zdefiniowaną tylko dla grafów skierowanych. Jest to stosunek wierzchołków wzajemnie połączonych do wierzchołków, które są połączone krawędzią tylko w jednym kierunku. Dana jest wzorem:

$$\text{reciprocity} = \frac{|(i, j) \in E \mid (j, i) \in E|}{|(i, j) \in E \mid (j, i) \notin E|} \quad (3.3)$$

- **mean_funnel_size** — średnia wielkość leja. Wielkość leja to liczba należących do niego optimów lokalnych.
- **max_funnel_size** — wielkość największego leja
- **go_path_ratio** — stosunek liczby wierzchołków z istniejącą ścieżką do najlepszego rozwiązania do liczby wszystkich wierzchołków.

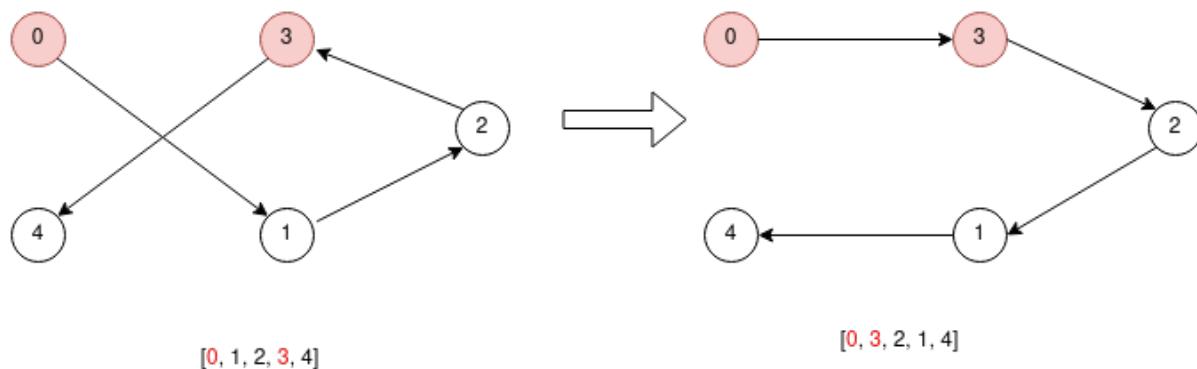
- **avg_go_path_len** — średnia długość ścieżki do najlepszego rozwiązania. Wierzchołki bez ścieżki do najlepszego rozwiązania nie sąbrane pod uwagę. Długość ścieżki definiowana jest jako liczba krawędzi wchodzących w skład ścieżki pomiędzy wierzchołkami.
- **max_go_path_len** — długość najdłuższej ścieżki do najlepszego rozwiązania.
- **num_cc** — Liczba spójnych podgrafów grafu
- **largest_cc** — Wielkość (liczba wierzchołków) największego spójnego podgrafa.
- **largest_cc_radius** — Promień największego spójnego podgrafa. Promień grafu to najmniejsza acentryczność wierzchołka wśród wszystkich wierzchołków grafu. Acentryczność (ang. eccentricity) wierzchołka to największa z odległości wierzchołka do innych wierzchołków grafu.

3.6 Operacja 2-exchange

Operacja 2-exchange jest operacją modyfikującą rozwiązanie problemu komiwojażera. Polega na wybraniu dwóch wierzchołków (miast) i zamiany krawędzi prowadzących do następnych wierzchołków. Procedura jest wykorzystywana w algorytmie heurystycznym 2opt, w którym w ten sposób „rozplatane” są skrzyżowane krawędzie.

Algorytmy przeszukiwania przestrzeni rozwiązań zaprezentowane w tej pracy wykorzystują operację 2-exchange jako operację mutacji. Mutacja permutacji polega na D-krotnym wykonaniu operacji 2-exchange na losowych wierzchołkach, gdzie D to maksymalna odległość z definicji 3.1.

Operacja 2-exchange oraz operator mutacji zostały przedstawione na listingu 1 oraz rysunku 3.1.



Rysunek 3.1 Przykład procedury 2-exchange

Algorithm 1: Operacja 2exchange — pseudokod

```

function 2exchange( $a, b, perm$ ):
     $a \leftarrow a + 1;$ 
    while  $a < b$  do
        zamien( $perm[a], perm[b]$ );
         $a \leftarrow a + 1;$ 
         $b \leftarrow b + 1;$ 
    end
    return  $perm$ ;
end

function 2exchangeMutacja( $perm, D$ ):
    for  $i \leftarrow 1$  to  $D$  do
         $a \leftarrow losowaZZakresu(0, length(perm) - 3);$ 
         $b \leftarrow losowaZZakresu(a + 2, length(perm) - 1);$ 
         $perm \leftarrow 2exchange(a, b, perm);$ 
    end
    return  $perm$ ;
end

function 2exchangeWszystkiePermutacje( $perm$ ):
     $perms = \{\};$ 
    for  $a \leftarrow 0$  to  $n - 2$  do
        for  $b \leftarrow a + 1$  to  $n - 1$  do
             $perm \leftarrow 2exchange(a, b);$ 
             $perms \leftarrow perms \cup \{perm\};$ 
        end
    end
    return  $perms$ ;
end
```

Rozdział 4

Badania eksperymentalne

Rozdział opisuje przeprowadzone w pracy badania. Sekcja 4.1 przedstawia wykorzystane algorytmy próbkowania, a sekcja 4.2 zawiera opis analizowanych instancji problemu komiwojażera. W kolejnych sekcjach opisane są przeprowadzone eksperymenty, oraz uzyskane z nich wyniki.

4.1 Zaimplementowane algorytmy

Na potrzeby badania wykonano implementacje dwóch algorytmów próbkowania - próbkowania dwufazowego oraz próbkowania *snowball*. Dodatkowo, aby możliwe było porównanie wyników próbkowania z rzeczywistą przestrzenią rozwiązań, zaimplementowano algorytm przeglądu zupełnego.

4.1.1 Próbkowanie dwufazowe

Próbkowanie dwufazowe swoją nazwę zawdzięcza procesowi próbkowania składającemu się z dwóch oddzielnego faz - próbkowania wierzchołków oraz próbkowania krawędzi — wykonywanych jedna po drugiej. Istotną zaletą tego podejścia jest jego stosunkowo prosta implementacja oraz możliwość łatwego zrównoleglenia.

Zaimplementowany algorytm pochodzi z pracy [3]. Został on przygotowany specjalnie do próbkowania przestrzeni rozwiązań problemu komiwojażera. Próbkowanie wierzchołków odbywa się poprzez generowanie losowych rozwiązań, a następnie ich optymalizacji algorytmem 2-opt. Próbkowanie krawędzi polega na wielokrotnym poddaniu każdego ze znalezionych wcześniej lokalnych optimów n_i operacji perturbacji typu 2-exchange, a następnie poddaniu powstałego rozwiązania optymalizacji algorytmem 2-opt typu *first-improvement* uzyskując w ten sposób lokalne optimum n_j . Następnie dodawana jest krawędź między n_i a n_j , lub — jeśli już taka istnieje — jej waga jest zwiększana o 1.

Algorytm przyjmuje trzy parametry: pożądaną liczbę wierzchołków do wygenerowania (n_{max}), maksymalną liczbę prób generowania wierzchołka (n_{att}) oraz maksymalną liczbę prób generowania krawędzi (e_{att}). Implementacja zastosowana w tej pracy dodatkowo powtarza cały proces kilkukrotnie, za każdym razem zapisując zebrane próbki do pliku.

Algorytm w postaci pseudokodu został przedstawiony na listingu 2.

4.1.2 Snowball

Próbkowanie typu Snowball wywodzi się z techniki używanej w badaniach z dziedziny socjologii, w której ludzie należący do próby z populacji rekrutują kolejnych uczestników

badania spośród swoich znajomych. W kontekście badania przestrzeni rozwiązań technika ta została zaprezentowana w pracy [22], gdzie została wykorzystana do próbkowania przestrzeni problemu kwadratowego przypisania (QAP).

Próbkowanie składa się z etapów procedury *snowball* próbującej „wgłęb” i losowego spaceru(ang. *random walk*). Próbkowanie *snowball* polega na wybraniu rozwiązania startowego i przeszukaniu jego najbliższego sąsiedztwa. Następnie operacja ta jest powtarzana dla każdego rozwiązania w tym sąsiedztwie. Proces powtarza się aż do osiągnięcia z góry ustalonej głębokości przeszukiwania. Następnie rozpoczyna się procedura losowego spaceru — wybierane jest kolejne rozwiązanie startowe ze zbioru sąsiadów poprzedniego rozwiązywania startowego (lub rozwiązanie losowe, jeśli to sąsiedztwo jest puste) i proces *snowball* rozpoczyna się od nowa. Procedura jest powtarzana aż osiągnięty zostanie z góry ustalony limit długości spaceru.

Zaimplementowany algorytm jest próbą adaptacji tej techniki do zadania przeszukiwania przestrzeni problemu komiwojażera. Do najważniejszych modyfikacji należy zastąpienie funkcji optymalizacji lokalnej typu *hillclimb* optymalizacją 2opt, implementacja odpowiedniej funkcji celu oraz operacji mutacji typu 2-exchange.

Algorytm w postaci pseudokodu został przedstawiony na listingu 3.

Algorithm 2: Próbkowanie dwufazowe — pseudokod**Data:**

n_{max} — żądana liczba wierzchołków
 n_{att} — liczba prób generowania wierzchołków
 e_{att} — liczba prób generowania krawędzi
 n_{runs} — liczba powtórzeń
 D — stała D krawędzi

```

 $N \leftarrow \{\};$ 
 $E \leftarrow \{\};$ 
for  $i \leftarrow 1$  to  $n_{runs}$  do
|   probkujWierzcholki(N, nmax, natt);
|   probkujKrawedzie(N, E, eatt);
|   zapiszDoPliku(N, E);
end

function probkujWierzcholki(N, nmax, natt):
|   for  $i \leftarrow 1$  to  $n_{max}$  do
|   |   for  $i \leftarrow 1$  to  $n_{att}$  do
|   |   |    $s \leftarrow losoweRozwiazanie();$ 
|   |   |    $s \leftarrow 2opt(s);$ 
|   |   |    $N \leftarrow N \cup \{s\};$ 
|   |   end
|   |   end
|   end
end

function probkujKrawedzie(N, E, eatt):
|   foreach  $n \in N$  do
|   |   for  $i \leftarrow 1$  to  $e_{att}$  do
|   |   |    $s \leftarrow 2exchangeMutacja(n, D);$ 
|   |   |    $s \leftarrow 2optFirstImprovement(s);$ 
|   |   |   if  $s \in N$  then
|   |   |   |    $E \leftarrow E \cup \{(n, s)\};$ 
|   |   |   |    $w_{ns} \leftarrow w_{ns} + 1;$ 
|   |   |   end
|   |   end
|   |   end
|   end
end

```

Algorithm 3: Próbkowanie snowball — pseudokod**Data:**

w_{len} — długość losowego spaceru
 m — liczba prób przeszukania sąsiedztwa
 $depth$ — głębokość przeszukiwania
 D — stała D krawędzi
 s_{tresh} — interwał zapisu

```

 $s_1 \leftarrow losoweRozwiazanie();$ 
 $n_1 \leftarrow 2opt(s_1);$ 
 $N \leftarrow \{n_1\};$ 
 $E \leftarrow \{\}$ ;
for  $j \leftarrow 1$  to  $n_{runs}$  do
  for  $i \leftarrow 1$  to  $w_{len}$  do
    |  $snowball(n_i, m, depth);$ 
    |  $n_{i+1} \leftarrow losowySpacer(n_i);$ 
  end
end
 $zapiszDoPliku(N, E);$ 

function  $snowball(n, m, depth):$ 
  if  $d > 0$  then
    for  $i \leftarrow 1$  to  $m$  do
      |  $s \leftarrow 2opt(2exchangeMutacja(n, D));$ 
      |  $N \leftarrow N \cup \{s\};$ 
      | if  $|N| \ mod \ s_{tresh} = 0$  then
        |   |  $zapiszDoPliku(N, E);$ 
      | end
      | if  $(n, s) \in E$  then
        |   |  $w_{ns} \leftarrow w_{ns} + 1;$ 
      | else
        |   |  $E \leftarrow E \cup \{(n, s)\};$ 
        |   |  $w_{ns} \leftarrow 1;$ 
        |   |  $snowball(s, m, d - 1);$ 
      | end
    end
  end
end

function  $losowySpacer(n_i):$ 
   $neighbours \leftarrow \{s : (n_i, s) \in E \wedge s \notin \{n_0 \dots n_i\}\};$ 
  if  $neighbours \neq \emptyset$  then
    |  $n_{i+1} \leftarrow losowyElementZeZbioru(neighbours);$ 
  else
    |  $s \leftarrow losoweRozwiazanie();$ 
    |  $n_{i+1} \leftarrow 2opt(s);$ 
    |  $N \leftarrow N \cup \{n_{i+1}\};$ 
    | if  $|N| \ mod \ s_{tresh} = 0$  then
      |   |  $zapiszDoPliku(N, E);$ 
    | end
  end
  return  $n_{i+1}$ 
end

```

4.1.3 Przegląd zupełny

Ze względu na złożoność problemu komiwojażera oraz rozmiar przestrzeni rozwiązań, przegląd zupełny można zastosować tylko do bardzo małych instancji problemu. Przegląd polega na wygenerowaniu wszystkich możliwych rozwiązań danej instancji, wykonaniu na nich optymalizacji 2-opt w celu znalezienia optimów lokalnych a następnie znalezieniu krawędzi oraz obliczeniu ich wag. Dla każdego z rozwiązań generowane są wszystkie permutacje, które mogą powstać poprzez D-krotne wykonanie na rozwiązaniu operacji 2-exchange. Jeśli wśród tych permutacji znajduje się jedno ze znalezionych wcześniej lokalnych optimów, oznacza to, że spełniony jest warunek 3.1 i dodawana jest nowa krawędź lub zwiększoza zostaje waga istniejącej. Algorytm został przedstawiony na listingu 4.

4.2 Instancje testowe

Do badań wykorzystano instancje testowe wygenerowane losowo oraz wybrane instancje ze zbioru TSPLIB. Zaimplementowano trzy generatory tworzące różne typy instancji testowych: z miastami rozłożonymi równomiernie, z miastami rozłożonymi w klikach oraz z miastami ułożonymi na siatce. Wygenerowano instancje testowe każdego z trzech typów instancji losowych o rozmiarach 7, 8, 9, 10, 11 oraz 20, 50, 80 i 100. Uzyskano w ten sposób 27 instancji problemu. Ze zbioru TSPLIB wybrano instancje o podobnych rozmiarach: **burma14**, **ulysses22**, **att48**, **berlin52**, **pr76**, **eil76**, **rat99**, **bier127**. W sumie badanie przeprowadzono na 35 instancjach problemu.

Miasta rozmieszczone równomiernie

Generator losowo rozmieszcza miasta na wirtualnej planszy o ustalonym rozmiarze. Współrzędne miast generowane są losowo z rozkładu równomiernego. W dalszej części dokumentu instancje wygenerowane tym generatorem będą nazywane **uniform_<liczba miast>**.

Przykład wygenerowanej instancji został przedstawiony na rysunku 4.1.

Miasta rozmieszczone w klikach

Miasta umieszczane są blisko siebie w kilku grupach oddzielonych większymi odległościami. W dalszej części dokumentu instancje wygenerowane tym generatorem będą nazywane **cliques_<liczba miast>**. Przykład wygenerowanej instancji został przedstawiony na rysunku 4.2.

Miasta rozmieszczone na siatce

Miasta umieszczane są na siatce, w stałej odległości od swoich sąsiadów. W dalszej części dokumentu instancje wygenerowane tym generatorem będą nazywane **grid_<liczba miast>**. Przykład wygenerowanej instancji został przedstawiony na rysunku 4.3.

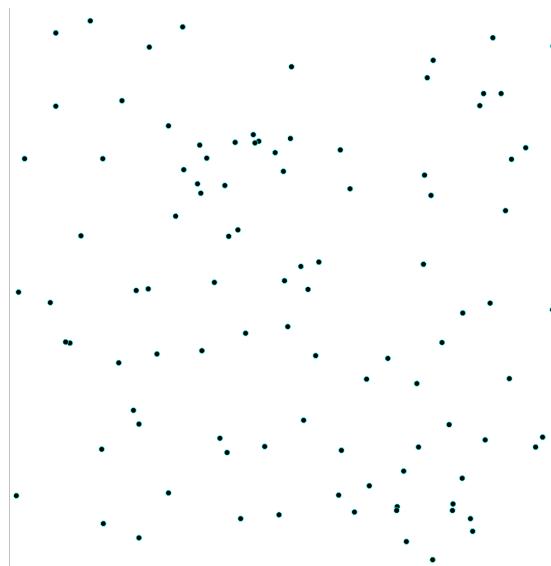
Algorithm 4: Przegląd zupełny

```

 $S \leftarrow \{\};$ 
 $P \leftarrow wygenerujWszystkiePermutacje();$ 
foreach  $p \in P$  do
     $lo \leftarrow 2optFirstImprovement(p);$ 
     $S \leftarrow S \cup \{(p, lo)\};$ 
end
foreach  $(p, lo) \in S$  do
    foreach  $n \in N$  do
        if  $wZasiegu2Exchange(p, n, D)$  then
            if  $(n, lo) \in E$  then
                 $w_{n,lo} \leftarrow w_{n,lo} + 1;$ 
            else
                 $E \leftarrow E \cup \{(n, lo)\};$ 
                 $w_{n,lo} = 1;$ 
            end
        end
    end
end
end

function  $wZasiegu2Exchange(p, n, D):$ 
     $permutacje \leftarrow \{p\};$ 
    for  $i \in 1..D$  do
         $nowe\_perm \leftarrow \{\};$ 
        foreach  $perm \in permutacje$  do
             $pochodne\_perm \leftarrow 2exchangeWszystkiePermutacje(permutacje);$ 
            foreach  $poch \in pochodne\_perm$  do
                if  $poch = n$  then
                    return  $true;$ 
                end
                 $nowe\_perm \leftarrow nowe\_perm \cup \{poch\};$ 
            end
        end
         $permutacje \leftarrow nowe\_perm;$ 
    end
    return  $false;$ 
end

```



Rysunek 4.1 Wizualizacja przykładowej wygenerowanej instancji z miastami rozmieszczonymi równomiernie dla 100 miast

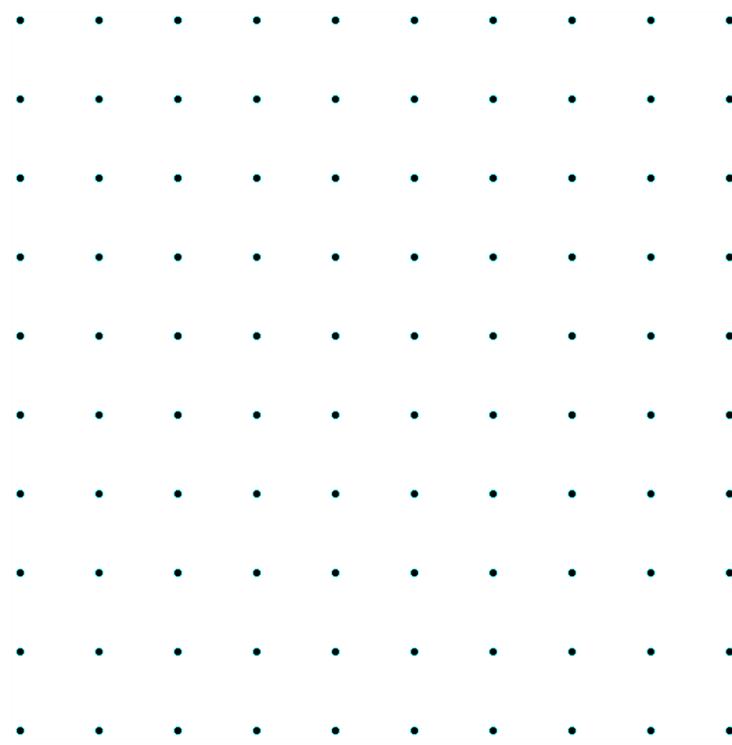


Rysunek 4.2 Wizualizacja przykładowej wygenerowanej instancji z miastami rozmieszczonymi w klikach dla 100 miast

4.3 Opis badań

Badania podzielono na trzy etapy, odpowiadające poszczególnym celom pracy:

1. Porównanie wartości metryk z przeglądem zupełnym dla małych instancji, celem sprawdzenia dokładności odwzorowania przestrzeni rozwiązań przez algorytmy próbkoowania (sekcja 4.3.1),



Rysunek 4.3 Wizualizacja przykładowej wygenerowanej instancji z miastami rozmieszczonymi na siatce dla 100 miast

2. Badanie stabilności średnich i dużych instancji - długotrwałe próbkowanie przeprowadzone w celu zbadania wpływu metody i długości próbkowania na wartości miar (sekcje 4.3.2 i 4.3.3),
3. Badanie korelacji wzajemnej poszczególnych miar, oraz wpływu metody próbkowania na ich wartości (sekcja 4.3.4).

W tym rozdziale przedstawione zostaną tabele i wykresy wygenerowane z wyników próbkowania. Nazwy metryk są identyczne z nazwami objaśnionymi w rozdziale 3.5. Ponadto występują pojęcia:

- node_count — liczba wierzchołków,
- edge_count — liczba krawędzi,
- opt_count — liczba wywołań funkcji 2opt,
- oracle_count — liczba wykonanych obliczeń długości ścieżki (metodą obliczenia przyrostu długości stosowanej w 2opt).
- twophase — algorytm dwufazowy

Wszystkie badania wykonano wykorzystując model krawędzi typu *Escape Edges* z parametrem D o wartości 2.

4.3.1 Porównanie wartości metryk dla małych instancji

Aby sprawdzić, z jaką dokładnością algorytmy próbkowania odwzorowują rzeczywistą przestrzeń rozwiązań, porównano wartości miar uzyskane z próbkowania z rzeczywistymi wartościami uzyskanymi z przeglądu zupełnego. Wykorzystano w tym celu instancje wygenerowane losowo (uniform, grid, cliques) o rozmiarach 7 do 11, dla których wykonanie takiego przeglądu jest możliwe. Aby ocenić, ile czasu jest potrzebne na wykonanie próbkowania zadowalającą dokładnością wyników, uruchomiono algorytmy kilkukrotnie, za każdym razem zwiększając czas ich działania.

Dla algorytmu dwufazowego wykonano próbkowanie dla różnych wartości parametru n_{max} (liczby żądanego wierzchołków): 10, 40, 100 oraz 1000. Parametry n_{att} , e_{att} ustawiono na równe n_{max} , a parametr n_{runs} na 1. Dla algorytmu *snowball* wykonano próbkowanie dla różnych wartości parametru w_{len} (długości losowego spaceru): 1, 10, 100 oraz 1000. Pozostałe parametry ustawiono następująco: $m = 100$, $depth = 3$.

Uzyskane wartości porównano z wartościami uzyskanymi z przeglądu zupełnego. Wyniki przedstawiono w tabelach od 4.1 do 4.30. W tabeli przedstawiono wartości metryk uzyskane w przeglądzie zupełnym oraz wartości uzyskane z próbkowania. Dodatkowo w nawiasie podana została wartość błędu obliczona z wzoru 4.1.

$$E = \frac{|R - S|}{R} \cdot 100\% \quad (4.1)$$

Gdzie:

- E — wartość błędu
- R — wartość rzeczywista (z przegl. zupełnego)
- S — wartość otrzymana z próbkowania.

Wartości, dla których błąd nie wyniósł 0 zostały dodatkowo oznaczone w tabeli grubą czcionką. Przebadano wszystkie miary opisane w sekcji 3.5 za wyjątkiem *assortativity_deg* — wartość ta dla małych instancji zwykle przyjmowała nieprawidłową wartość (NaN), więc zdecydowano o jej wykluszeniu.

Wyniki

Tabela 4.1 Instancja uniform_7 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	720.00	501.00	1608.00	10698.00	101798.00
oracle_count	42362.00	30912.00	97125.00	650391.00	6179229.00
node_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
edge_count	16.00	15.00 (6.25%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	4.00	3.75 (6.25%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
avg_fitness	2720.50	2720.50 (0.00%)	2720.50 (0.00%)	2720.50 (0.00%)	2720.50 (0.00%)
distLO	0.02	0.05 (203.37%)	0.02 (2.24%)	0.00 (79.17%)	0.00 (97.45%)
conrel	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_loop_weight	57.00	83.75 (46.93%)	257.00 (350.88%)	1863.00 (3168.42%)	17944.75 (31382.02%)
go_path_ratio	0.75	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_out_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_out_degree	3.00	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_in_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_in_degree	3.00	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
density	1.33	1.25 (6.25%)	1.33 (0.00%)	1.33 (0.00%)	1.33 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.08 (8.33%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
reciprocity	1.00	0.91 (9.09%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.2 Instancja uniform_7 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	720.00	106.00	1661.00	10014.00	1000010.00
oracle_count	42362.00	7525.00	126699.00	788384.00	80004896.00
node_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
edge_count	16.00	11.00 (31.25%)	15.00 (6.25%)	15.00 (6.25%)	16.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	4.00	2.75 (31.25%)	3.75 (6.25%)	3.75 (6.25%)	4.00 (0.00%)
avg_fitness	2720.50	2720.50 (0.00%)	2720.50 (0.00%)	2720.50 (0.00%)	2720.50 (0.00%)
distLO	0.02	0.42 (2237.80%)	0.05 (163.64%)	0.02 (26.72%)	0.00 (88.20%)
conrel	3.00	1.00 (66.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_loop_weight	57.00	3.75 (93.42%)	18.25 (67.98%)	43.25 (24.12%)	382.50 (571.05%)
go_path_ratio	0.75	0.25 (66.67%)	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	3.00	2.50 (16.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_out_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_out_degree	3.00	1.75 (41.67%)	2.75 (8.33%)	2.75 (8.33%)	3.00 (0.00%)
max_in_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_in_degree	3.00	1.75 (41.67%)	2.75 (8.33%)	2.75 (8.33%)	3.00 (0.00%)
density	1.33	0.92 (31.25%)	1.25 (6.25%)	1.25 (6.25%)	1.33 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.12 (12.50%)	1.08 (8.33%)	1.08 (8.33%)	1.00 (0.00%)
largest_clique_size	4.00	3.00 (25.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
reciprocity	1.00	0.57 (42.86%)	0.91 (9.09%)	0.91 (9.09%)	1.00 (0.00%)

Tabela 4.3 Instancja cliques_7 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	720.00	301.00	1210.00	10300.00	101200.00
oracle_count	46570.00	19467.00	79989.00	681198.00	6671805.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	2401.00	2401.00 (0.00%)	2401.00 (0.00%)	2401.00 (0.00%)	2401.00 (0.00%)
distLO	0.02	0.03 (33.15%)	0.00 (78.81%)	0.00 (96.68%)	0.00 (99.70%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	100.00	121.50 (21.50%)	489.00 (389.00%)	4266.00 (4166.00%)	41638.50 (41538.50%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.4 Instancja cliques_7 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	720.00	102.00	1623.00	10003.00	1000006.00
oracle_count	46570.00	8190.00	143165.00	889106.00	89524393.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	2401.00	2401.00 (0.00%)	2401.00 (0.00%)	2401.00 (0.00%)	2401.00 (0.00%)
distLO	0.02	0.13 (477.43%)	0.03 (46.55%)	0.02 (27.19%)	0.00 (92.16%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	100.00	5.50 (94.50%)	32.00 (68.00%)	66.50 (33.50%)	715.50 (615.50%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.5 Instancja grid _ 7 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	720.00	301.00	1210.00	10300.00	101200.00
oracle_count	40111.00	18564.00	72912.00	624981.00	6124188.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	741.00	741.00 (0.00%)	741.00 (0.00%)	741.00 (0.00%)	741.00 (0.00%)
distLO	0.01	0.03 (124.07%)	0.01 (49.98%)	0.00 (93.94%)	0.00 (99.43%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	99.50	127.50 (28.14%)	507.50 (410.05%)	4289.00 (4210.55%)	42077.50 (42188.94%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.6 Instancja grid _ 7 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	720.00	102.00	1622.00	10002.00	1000002.00
oracle_count	40111.00	7516.00	131943.00	834126.00	84311158.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	3.00 (25.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	1.50 (25.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	741.00	741.00 (0.00%)	741.00 (0.00%)	741.00 (0.00%)	741.00 (0.00%)
distLO	0.01	0.17 (1371.33%)	0.03 (149.11%)	0.01 (22.30%)	0.01 (48.57%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	99.50	8.00 (91.96%)	34.00 (65.83%)	66.50 (33.17%)	727.00 (630.65%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	0.50 (50.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	0.50 (50.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	1.50 (25.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	0.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.7 Instancja uniform_8 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	5040.00	401.00	1210.00	10300.00	101200.00
oracle_count	570090.00	34384.00	106120.00	905660.00	8916992.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	3224.00	3224.00 (0.00%)	3224.00 (0.00%)	3224.00 (0.00%)	3224.00 (0.00%)
distLO	0.03	0.02 (15.71%)	0.01 (65.97%)	0.00 (96.21%)	0.00 (99.64%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	206.00	171.50 (16.75%)	532.50 (158.50%)	4520.00 (2094.17%)	44341.50 (21425.00%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.8 Instancja uniform_8 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	5040.00	102.00	1623.00	10003.00	1000002.00
oracle_count	570090.00	13685.00	231804.00	1456655.00	146845408.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	3.00 (25.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	1.50 (25.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	3224.00	3224.00 (0.00%)	3224.00 (0.00%)	3224.00 (0.00%)	3224.00 (0.00%)
distLO	0.03	0.13 (389.16%)	0.19 (591.20%)	0.11 (316.44%)	0.01 (60.73%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	206.00	7.00 (96.60%)	35.00 (83.01%)	66.00 (67.96%)	670.50 (225.49%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	0.50 (50.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	0.50 (50.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	1.50 (25.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	0.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.9 Instancja cliques_8 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	5040.00	501.00	1508.00	10598.00	101498.00
oracle_count	419679.00	42252.00	127932.00	895076.00	8632344.00
node_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
edge_count	16.00	15.00 (6.25%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	4.00	3.75 (6.25%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
avg_fitness	1718.00	1718.00 (0.00%)	1718.00 (0.00%)	1718.00 (0.00%)	1718.00 (0.00%)
distLO	0.01	0.06 (388.30%)	0.01 (14.48%)	0.01 (56.23%)	0.00 (97.32%)
conrel	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_loop_weight	141.75	82.25 (41.98%)	247.00 (74.25%)	1991.75 (1305.11%)	19076.00 (13357.50%)
go_path_ratio	0.75	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_out_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_out_degree	3.00	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_in_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_in_degree	3.00	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
density	1.33	1.25 (6.25%)	1.33 (0.00%)	1.33 (0.00%)	1.33 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.08 (8.33%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
reciprocity	1.00	0.91 (9.09%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.10 Instancja cliques_8 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	5040.00	111.00	1648.00	10010.00	100007.00
oracle_count	419679.00	12363.00	206732.00	1313485.00	133437863.00
node_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
edge_count	16.00	14.00 (12.50%)	15.00 (6.25%)	16.00 (0.00%)	16.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	4.00	3.50 (12.50%)	3.75 (6.25%)	4.00 (0.00%)	4.00 (0.00%)
avg_fitness	1718.00	1718.00 (0.00%)	1718.00 (0.00%)	1718.00 (0.00%)	1718.00 (0.00%)
distLO	0.01	0.48 (3679.85%)	0.09 (585.96%)	0.05 (288.51%)	0.00 (77.78%)
conrel	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_loop_weight	141.75	3.25 (97.71%)	23.00 (83.77%)	42.25 (70.19%)	453.25 (219.75%)
go_path_ratio	0.75	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_out_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_out_degree	3.00	2.50 (16.67%)	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)
max_in_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_in_degree	3.00	2.50 (16.67%)	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)
density	1.33	1.17 (12.50%)	1.25 (6.25%)	1.33 (0.00%)	1.33 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.17 (16.67%)	1.08 (8.33%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
reciprocity	1.00	0.80 (20.00%)	0.91 (9.09%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.11 Instancja grid_8 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	5040.00	401.00	1210.00	10300.00	101200.00
oracle_count	448412.00	34216.00	102984.00	868392.00	8524068.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	800.00	800.00 (0.00%)	800.00 (0.00%)	800.00 (0.00%)	800.00 (0.00%)
distLO	0.01	0.03 (370.57%)	0.01 (65.35%)	0.00 (82.66%)	0.00 (98.44%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	214.00	180.50 (15.65%)	541.00 (152.80%)	4540.00 (2021.50%)	44727.50 (20800.70%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.12 Instancja grid_8 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	5040.00	102.00	1623.00	10002.00	1000002.00
oracle_count	448412.00	12381.00	211038.00	1344823.00	135267881.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	800.00	800.00 (0.00%)	800.00 (0.00%)	800.00 (0.00%)	800.00 (0.00%)
distLO	0.01	0.19 (2786.15%)	0.03 (401.94%)	0.02 (223.27%)	0.01 (9.76%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	214.00	7.00 (96.73%)	36.50 (82.94%)	80.50 (62.38%)	744.00 (247.66%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.13 Instancja uniform_9 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	40320.00	401.00	1210.00	10300.00	101200.00
oracle_count	6071702.00	44460.00	136116.00	1165968.00	11466720.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	2482.00	2482.00 (0.00%)	2482.00 (0.00%)	2482.00 (0.00%)	2482.00 (0.00%)
distLO	0.00	0.04 (730.53%)	0.02 (252.09%)	0.00 (73.72%)	0.00 (98.09%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	390.00	180.50 (53.72%)	528.00 (35.38%)	4527.50 (1060.90%)	44516.00 (11314.36%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.14 Instancja uniform_9 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	40320.00	103.00	1622.00	10002.00	100003.00
oracle_count	6071702.00	19296.00	335931.00	2093776.00	211786272.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	3.00 (25.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	1.50 (25.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	2482.00	2482.00 (0.00%)	2482.00 (0.00%)	2482.00 (0.00%)	2482.00 (0.00%)
distLO	0.00	0.17 (3488.73%)	0.08 (1529.09%)	0.05 (948.82%)	0.00 (38.83%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	390.00	8.00 (97.95%)	40.00 (89.74%)	79.50 (79.62%)	806.00 (106.67%)
go_path_ratio	0.50	0.00 (100.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	0.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	0.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	0.50 (50.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	0.50 (50.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	1.50 (25.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	0.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.15 Instancja cliques_9 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	40320.00	401.00	1310.00	10300.00	101200.00
oracle_count	6306952.00	47880.00	156960.00	1235592.00	12167676.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	1458.00	1458.00 (0.00%)	1458.00 (0.00%)	1458.00 (0.00%)	1458.00 (0.00%)
distLO	0.00	0.03 (900.91%)	0.01 (194.47%)	0.00 (61.40%)	0.00 (95.67%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	388.00	181.00 (53.35%)	589.00 (51.80%)	4638.50 (1095.49%)	45549.50 (11639.56%)
go_path_ratio	0.50	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.16 Instancja cliques_9 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	40320.00	104.00	1622.00	10002.00	1000002.00
oracle_count	6306952.00	21499.00	359182.00	2284332.00	229659303.00
node_count	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_count	4.00	3.00 (25.00%)	3.00 (25.00%)	4.00 (0.00%)	4.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	2.00	1.50 (25.00%)	1.50 (25.00%)	2.00 (0.00%)	2.00 (0.00%)
avg_fitness	1458.00	1458.00 (0.00%)	1458.00 (0.00%)	1458.00 (0.00%)	1458.00 (0.00%)
distLO	0.00	0.25 (7669.83%)	0.02 (630.78%)	0.18 (5523.88%)	0.00 (56.05%)
conrel	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_loop_weight	388.00	7.00 (98.20%)	31.50 (91.88%)	66.00 (82.99%)	701.50 (80.80%)
go_path_ratio	0.50	0.00 (100.00%)	0.50 (0.00%)	0.50 (0.00%)	0.50 (0.00%)
avg_go_path_len	1.00	0.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	0.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_out_degree	1.00	0.50 (50.00%)	0.50 (50.00%)	1.00 (0.00%)	1.00 (0.00%)
max_in_degree	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_in_degree	1.00	0.50 (50.00%)	0.50 (50.00%)	1.00 (0.00%)	1.00 (0.00%)
density	2.00	1.50 (25.00%)	1.50 (25.00%)	2.00 (0.00%)	2.00 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
reciprocity	1.00	0.00 (100.00%)	0.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.17 Instancja grid_9 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	40320.00	2501.00	5401.00	17386.00	108686.00
oracle_count	4548760.00	240012.00	530136.00	1697868.00	10488780.00
node_count	16.00	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
edge_count	236.00	131.00 (44.49%)	191.00 (19.07%)	228.00 (3.39%)	236.00 (0.00%)
num_sub sinks	16.00	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
edge_to_node	14.75	8.19 (44.49%)	11.94 (19.07%)	14.25 (3.39%)	14.75 (0.00%)
avg_fitness	941.00	941.00 (0.00%)	941.00 (0.00%)	941.00 (0.00%)	941.00 (0.00%)
distLO	0.01	0.34 (2442.55%)	0.28 (1974.80%)	0.05 (281.88%)	0.01 (34.25%)
conrel	15.00	4.33 (71.11%)	15.00 (0.00%)	15.00 (0.00%)	15.00 (0.00%)
avg_loop_weight	119.62	90.25 (24.56%)	152.38 (27.38%)	496.12 (314.73%)	3191.19 (2567.66%)
go_path_ratio	0.94	0.69 (26.67%)	0.94 (0.00%)	0.94 (0.00%)	0.94 (0.00%)
avg_go_path_len	1.00	1.09 (9.09%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	16.00	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
num_sources	16.00	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	15.00	12.00 (20.00%)	14.00 (6.67%)	15.00 (0.00%)	15.00 (0.00%)
avg_out_degree	13.75	7.44 (45.91%)	10.94 (20.45%)	13.25 (3.64%)	13.75 (0.00%)
max_in_degree	15.00	11.00 (26.67%)	15.00 (0.00%)	15.00 (0.00%)	15.00 (0.00%)
avg_in_degree	13.75	7.44 (45.91%)	10.94 (20.45%)	13.25 (3.64%)	13.75 (0.00%)
density	0.98	0.55 (44.49%)	0.80 (19.07%)	0.95 (3.39%)	0.98 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	16.00	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
largest_cc_radius	1.00	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.08	1.34 (23.59%)	1.27 (17.31%)	1.12 (3.08%)	1.08 (0.00%)
largest_clique_size	14.00	8.00 (42.86%)	10.00 (28.57%)	12.00 (14.29%)	12.00 (14.29%)
reciprocity	0.93	0.72 (22.06%)	0.79 (14.96%)	0.97 (4.79%)	1.00 (7.84%)

Tabela 4.18 Instancja grid_9 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	40320.00	122.00	1838.00	10141.00	1000067.00
oracle_count	4548760.00	11663.00	240147.00	1576822.00	168817024.00
node_count	16.00	10.00 (37.50%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
edge_count	236.00	38.00 (83.90%)	140.00 (40.68%)	167.00 (29.24%)	225.00 (4.66%)
num_sub sinks	16.00	10.00 (37.50%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
edge_to_node	14.75	3.80 (74.24%)	8.75 (40.68%)	10.44 (29.24%)	14.06 (4.66%)
avg_fitness	941.00	941.00 (0.00%)	941.00 (0.00%)	941.00 (0.00%)	941.00 (0.00%)
distLO	0.01	0.49 (3563.83%)	0.18 (1292.29%)	0.36 (2623.22%)	0.18 (1262.31%)
conrel	15.00	4.00 (73.33%)	15.00 (0.00%)	7.00 (53.33%)	7.00 (53.33%)
avg_loop_weight	119.62	3.71 (96.90%)	11.25 (90.60%)	24.19 (79.78%)	222.62 (86.10%)
go_path_ratio	0.94	0.90 (4.00%)	0.94 (0.00%)	0.94 (0.00%)	0.94 (0.00%)
avg_go_path_len	1.00	1.11 (11.11%)	1.07 (6.67%)	1.13 (13.33%)	1.07 (6.67%)
max_go_path_len	1.00	2.00 (100.00%)	2.00 (100.00%)	2.00 (100.00%)	2.00 (100.00%)
num_sinks	16.00	10.00 (37.50%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
num_sources	16.00	10.00 (37.50%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
mean_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_funnel_size	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_out_degree	15.00	4.00 (73.33%)	11.00 (26.67%)	14.00 (6.67%)	15.00 (0.00%)
avg_out_degree	13.75	3.10 (77.45%)	7.75 (43.64%)	9.44 (31.36%)	13.06 (5.00%)
max_in_degree	15.00	8.00 (46.67%)	15.00 (0.00%)	15.00 (0.00%)	15.00 (0.00%)
avg_in_degree	13.75	3.10 (77.45%)	7.75 (43.64%)	9.44 (31.36%)	13.06 (5.00%)
density	0.98	0.42 (57.06%)	0.58 (40.68%)	0.70 (29.24%)	0.94 (4.66%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	16.00	10.00 (37.50%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
largest_cc_radius	1.00	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.08	1.79 (65.33%)	1.43 (32.31%)	1.38 (26.92%)	1.13 (4.23%)
largest_clique_size	14.00	5.00 (64.29%)	8.00 (42.86%)	10.00 (28.57%)	13.00 (7.14%)
reciprocity	0.93	0.39 (58.25%)	0.50 (46.08%)	0.66 (28.58%)	0.91 (1.96%)

Tabela 4.19 Instancja uniform_10 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	362880.00	601.00	2006.00	11496.00	102496.00
oracle_count	70346836.00	86985.00	293130.00	1674675.00	14796000.00
node_count	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
edge_count	35.00	20.00 (42.86%)	30.00 (14.29%)	32.00 (8.57%)	32.00 (8.57%)
num_sub sinks	2.00	3.00 (50.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	5.83	3.33 (42.86%)	5.00 (14.29%)	5.33 (8.57%)	5.33 (8.57%)
avg_fitness	2111.67	2111.67 (0.00%)	2111.67 (0.00%)	2111.67 (0.00%)	2111.67 (0.00%)
distLO	0.01	0.08 (1073.45%)	0.13 (1844.98%)	0.03 (301.38%)	0.00 (38.98%)
conrel	5.00	1.00 (80.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_loop_weight	352.50	98.60 (72.03%)	262.33 (25.58%)	1550.00 (339.72%)	14307.83 (3958.96%)
go_path_ratio	0.83	0.67 (20.00%)	0.83 (0.00%)	0.83 (0.00%)	0.83 (0.00%)
avg_go_path_len	1.00	1.25 (25.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	3.00 (50.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	5.00	2.67 (46.67%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
max_funnel_size	5.00	4.00 (20.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
max_out_degree	5.00	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_out_degree	4.83	2.50 (48.28%)	4.00 (17.24%)	4.33 (10.34%)	4.33 (10.34%)
max_in_degree	5.00	3.00 (40.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_in_degree	4.83	2.50 (48.28%)	4.00 (17.24%)	4.33 (10.34%)	4.33 (10.34%)
density	1.17	0.67 (42.86%)	1.00 (14.29%)	1.07 (8.57%)	1.07 (8.57%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.03	1.48 (43.23%)	1.20 (16.13%)	1.13 (9.68%)	1.13 (9.68%)
largest_clique_size	6.00	4.00 (33.33%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
reciprocity	0.97	0.67 (30.95%)	0.75 (22.32%)	0.85 (12.36%)	0.85 (12.36%)

Tabela 4.20 Instancja uniform_10 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	362880.00	112.00	1693.00	10037.00	1000026.00
oracle_count	70346836.00	22080.00	440811.00	2812073.00	287817627.00
node_count	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
edge_count	35.00	21.00 (40.00%)	29.00 (17.14%)	32.00 (8.57%)	33.00 (5.71%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	5.83	3.50 (40.00%)	4.83 (17.14%)	5.33 (8.57%)	5.50 (5.71%)
avg_fitness	2111.67	2111.67 (0.00%)	2111.67 (0.00%)	2111.67 (0.00%)	2111.67 (0.00%)
distLO	0.01	0.72 (10677.02%)	0.18 (2558.91%)	0.07 (977.16%)	0.01 (110.83%)
conrel	5.00	2.00 (60.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_loop_weight	352.50	5.60 (98.41%)	22.00 (93.76%)	40.17 (88.61%)	398.83 (13.14%)
go_path_ratio	0.83	0.83 (0.00%)	0.83 (0.00%)	0.83 (0.00%)	0.83 (0.00%)
avg_go_path_len	1.00	1.20 (20.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	5.00	4.50 (10.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
max_funnel_size	5.00	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
max_out_degree	5.00	4.00 (20.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_out_degree	4.83	2.67 (44.83%)	3.83 (20.69%)	4.33 (10.34%)	4.50 (6.90%)
max_in_degree	5.00	4.00 (20.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_in_degree	4.83	2.67 (44.83%)	3.83 (20.69%)	4.33 (10.34%)	4.50 (6.90%)
density	1.17	0.70 (40.00%)	0.97 (17.14%)	1.07 (8.57%)	1.10 (5.71%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.03	1.44 (39.35%)	1.27 (22.58%)	1.13 (9.68%)	1.10 (6.45%)
largest_clique_size	6.00	4.00 (33.33%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
reciprocity	0.97	0.50 (48.21%)	0.70 (27.95%)	0.85 (12.36%)	0.89 (7.94%)

Tabela 4.21 Instancja cliques_10 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	362880.00	601.00	1808.00	10898.00	101698.00
oracle_count	66863792.00	88020.00	263925.00	1626795.00	15107760.00
node_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
edge_count	16.00	13.00 (18.75%)	15.00 (6.25%)	16.00 (0.00%)	16.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	4.00	3.25 (18.75%)	3.75 (6.25%)	4.00 (0.00%)	4.00 (0.00%)
avg_fitness	3356.50	3356.50 (0.00%)	3356.50 (0.00%)	3356.50 (0.00%)	3356.50 (0.00%)
distLO	0.01	0.03 (196.42%)	0.07 (530.44%)	0.01 (3.83%)	0.00 (82.48%)
conrel	3.00	1.00 (66.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_loop_weight	422.00	119.50 (71.68%)	364.75 (13.57%)	2219.75 (426.01%)	20851.00 (4841.00%)
go_path_ratio	0.75	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)
avg_go_path_len	1.00	1.33 (33.33%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	3.00	2.50 (16.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_out_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_out_degree	3.00	2.25 (25.00%)	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)
max_in_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_in_degree	3.00	2.25 (25.00%)	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)
density	1.33	1.08 (18.75%)	1.25 (6.25%)	1.33 (0.00%)	1.33 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.25 (25.00%)	1.08 (8.33%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	4.00	3.00 (25.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
reciprocity	1.00	0.89 (11.11%)	0.91 (9.09%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.22 Instancja cliques_10 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	362880.00	105.00	1647.00	10009.00	100007.00
oracle_count	66863792.00	24344.00	469045.00	2970174.00	303045402.00
node_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
edge_count	16.00	13.00 (18.75%)	16.00 (0.00%)	16.00 (0.00%)	16.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	4.00	3.25 (18.75%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
avg_fitness	3356.50	3356.50 (0.00%)	3356.50 (0.00%)	3356.50 (0.00%)	3356.50 (0.00%)
distLO	0.01	0.67 (5735.76%)	0.08 (582.35%)	0.03 (196.55%)	0.00 (66.86%)
conrel	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_loop_weight	422.00	4.33 (98.97%)	23.50 (94.43%)	41.50 (90.17%)	439.25 (4.09%)
go_path_ratio	0.75	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	3.00	2.50 (16.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_out_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_out_degree	3.00	2.50 (16.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_in_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_in_degree	3.00	2.50 (16.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
density	1.33	1.08 (18.75%)	1.33 (0.00%)	1.33 (0.00%)	1.33 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.17 (16.67%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
reciprocity	1.00	0.80 (20.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.23 Instancja grid_10 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	362880.00	2301.00	5301.00	19184.00	111184.00
oracle_count	60644109.00	297585.00	679185.00	2476485.00	14224680.00
node_count	18.00	18.00 (0.00%)	18.00 (0.00%)	18.00 (0.00%)	18.00 (0.00%)
edge_count	276.00	143.00 (48.19%)	206.00 (25.36%)	252.00 (8.70%)	275.00 (0.36%)
num_sub sinks	14.00	17.00 (21.43%)	14.00 (0.00%)	14.00 (0.00%)	14.00 (0.00%)
edge_to_node	15.33	7.94 (48.19%)	11.44 (25.36%)	14.00 (8.70%)	15.28 (0.36%)
avg_fitness	1100.44	1100.44 (0.00%)	1100.44 (0.00%)	1100.44 (0.00%)	1100.44 (0.00%)
distLO	0.13	0.23 (75.72%)	0.18 (34.40%)	0.13 (1.30%)	0.02 (81.95%)
conrel	17.00	3.50 (79.41%)	8.00 (52.94%)	5.00 (70.59%)	17.00 (0.00%)
avg_loop_weight	222.94	97.38 (56.32%)	163.72 (26.56%)	587.56 (163.54%)	3454.78 (1449.61%)
go_path_ratio	0.94	0.67 (29.41%)	0.94 (0.00%)	0.94 (0.00%)	0.94 (0.00%)
avg_go_path_len	1.00	1.25 (25.00%)	1.06 (5.88%)	1.12 (11.76%)	1.00 (0.00%)
max_go_path_len	1.00	2.00 (100.00%)	2.00 (100.00%)	2.00 (100.00%)	1.00 (0.00%)
num_sinks	14.00	17.00 (21.43%)	14.00 (0.00%)	14.00 (0.00%)	14.00 (0.00%)
num_sources	2.00	5.00 (150.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	4.79	1.76 (63.13%)	4.14 (13.43%)	4.79 (0.00%)	4.86 (1.49%)
max_funnel_size	5.00	2.00 (60.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
max_out_degree	17.00	15.00 (11.76%)	14.00 (17.65%)	16.00 (5.88%)	17.00 (0.00%)
avg_out_degree	14.33	7.22 (49.61%)	10.44 (27.13%)	13.00 (9.30%)	14.28 (0.39%)
max_in_degree	17.00	12.00 (29.41%)	16.00 (5.88%)	17.00 (0.00%)	17.00 (0.00%)
avg_in_degree	14.33	7.22 (49.61%)	10.44 (27.13%)	13.00 (9.30%)	14.28 (0.39%)
density	0.90	0.47 (48.19%)	0.67 (25.36%)	0.82 (8.70%)	0.90 (0.36%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	18.00	18.00 (0.00%)	18.00 (0.00%)	18.00 (0.00%)	18.00 (0.00%)
largest_cc_radius	1.00	2.00 (100.00%)	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.16	1.43 (23.99%)	1.40 (20.90%)	1.24 (6.78%)	1.16 (0.28%)
largest_clique_size	15.00	8.00 (46.67%)	9.00 (40.00%)	12.00 (20.00%)	14.00 (6.67%)
reciprocity	0.85	0.62 (27.83%)	0.76 (11.42%)	0.81 (4.78%)	0.87 (2.21%)

Tabela 4.24 Instancja grid_10 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	362880.00	122.00	1837.00	10136.00	1000151.00
oracle_count	60644109.00	16098.00	342834.00	2287711.00	248177074.00
node_count	18.00	10.00 (44.44%)	17.00 (5.56%)	18.00 (0.00%)	18.00 (0.00%)
edge_count	276.00	33.00 (88.04%)	136.00 (50.72%)	184.00 (33.33%)	255.00 (7.61%)
num_sub sinks	14.00	9.00 (35.71%)	14.00 (0.00%)	14.00 (0.00%)	14.00 (0.00%)
edge_to_node	15.33	3.30 (78.48%)	8.00 (47.83%)	10.22 (33.33%)	14.17 (7.61%)
avg_fitness	1100.44	1086.20 (1.29%)	1094.24 (0.56%)	1100.44 (0.00%)	1100.44 (0.00%)
distLO	0.13	0.24 (79.99%)	0.58 (343.20%)	0.27 (108.33%)	0.06 (53.51%)
conrel	17.00	2.33 (86.27%)	0.89 (94.77%)	3.50 (79.41%)	17.00 (0.00%)
avg_loop_weight	222.94	3.44 (98.46%)	13.65 (93.88%)	23.94 (89.26%)	247.89 (11.19%)
go_path_ratio	0.94	0.90 (4.71%)	0.94 (0.35%)	0.94 (0.00%)	0.94 (0.00%)
avg_go_path_len	1.00	1.33 (33.33%)	1.81 (81.25%)	1.35 (35.29%)	1.00 (0.00%)
max_go_path_len	1.00	2.00 (100.00%)	3.00 (200.00%)	2.00 (100.00%)	1.00 (0.00%)
num_sinks	14.00	9.00 (35.71%)	14.00 (0.00%)	14.00 (0.00%)	14.00 (0.00%)
num_sources	2.00	6.00 (200.00%)	6.00 (200.00%)	5.00 (150.00%)	2.00 (0.00%)
mean_funnel_size	4.79	1.44 (69.82%)	2.43 (49.25%)	3.43 (28.36%)	4.50 (5.97%)
max_funnel_size	5.00	2.00 (60.00%)	4.00 (20.00%)	5.00 (0.00%)	5.00 (0.00%)
max_out_degree	17.00	4.00 (76.47%)	11.00 (35.29%)	13.00 (23.53%)	17.00 (0.00%)
avg_out_degree	14.33	2.40 (83.26%)	7.00 (51.16%)	9.22 (35.66%)	13.17 (8.14%)
max_in_degree	17.00	6.00 (64.71%)	16.00 (5.88%)	17.00 (0.00%)	17.00 (0.00%)
avg_in_degree	14.33	2.40 (83.26%)	7.00 (51.16%)	9.22 (35.66%)	13.17 (8.14%)
density	0.90	0.37 (59.35%)	0.50 (44.57%)	0.60 (33.33%)	0.83 (7.61%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	18.00	10.00 (44.44%)	17.00 (5.56%)	18.00 (0.00%)	18.00 (0.00%)
largest_cc_radius	1.00	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.16	1.52 (31.63%)	1.54 (32.72%)	1.48 (28.32%)	1.23 (5.93%)
largest_clique_size	15.00	4.00 (73.33%)	8.00 (46.67%)	9.00 (40.00%)	15.00 (0.00%)
reciprocity	0.85	0.00 (100.00%)	0.45 (46.78%)	0.55 (35.01%)	0.76 (10.93%)

Tabela 4.25 Instancja uniform_11 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	3628800.00	401.00	1408.00	10898.00	101698.00
oracle_count	979464676.00	72105.00	253110.00	1986930.00	18375555.00
node_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
edge_count	16.00	10.00 (37.50%)	14.00 (12.50%)	14.00 (12.50%)	14.00 (12.50%)
num_sub sinks	2.00	3.00 (50.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	4.00	2.50 (37.50%)	3.50 (12.50%)	3.50 (12.50%)	3.50 (12.50%)
avg_fitness	2632.50	2632.50 (0.00%)	2632.50 (0.00%)	2632.50 (0.00%)	2632.50 (0.00%)
distLO	0.00	0.10 (3840.85%)	0.10 (3748.92%)	0.01 (400.38%)	0.00 (9.32%)
conrel	3.00	1.00 (66.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_loop_weight	738.25	120.33 (83.70%)	306.50 (58.48%)	2400.00 (225.09%)	23065.00 (3024.28%)
go_path_ratio	0.75	0.50 (33.33%)	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	3.00 (50.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	3.00	1.67 (44.44%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_funnel_size	3.00	2.00 (33.33%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_out_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_out_degree	3.00	1.75 (41.67%)	2.50 (16.67%)	2.50 (16.67%)	2.50 (16.67%)
max_in_degree	3.00	2.00 (33.33%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_in_degree	3.00	1.75 (41.67%)	2.50 (16.67%)	2.50 (16.67%)	2.50 (16.67%)
density	1.33	0.83 (37.50%)	1.17 (12.50%)	1.17 (12.50%)	1.17 (12.50%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.22 (22.22%)	1.17 (16.67%)	1.17 (16.67%)	1.17 (16.67%)
largest_clique_size	4.00	3.00 (25.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
reciprocity	1.00	0.57 (42.86%)	0.80 (20.00%)	0.80 (20.00%)	0.80 (20.00%)

Tabela 4.26 Instancja uniform_11 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	3628800.00	105.00	1652.00	10004.00	1000013.00
oracle_count	979464676.00	33153.00	637759.00	4067144.00	416068389.00
node_count	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
edge_count	16.00	12.00 (25.00%)	15.00 (6.25%)	16.00 (0.00%)	16.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	4.00	3.00 (25.00%)	3.75 (6.25%)	4.00 (0.00%)	4.00 (0.00%)
avg_fitness	2632.50	2632.50 (0.00%)	2632.50 (0.00%)	2632.50 (0.00%)	2632.50 (0.00%)
distLO	0.00	0.47 (17820.66%)	0.08 (2867.18%)	0.03 (1116.68%)	0.00 (77.18%)
conrel	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_loop_weight	738.25	5.50 (99.25%)	24.00 (96.75%)	44.75 (93.94%)	463.00 (37.28%)
go_path_ratio	0.75	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)	0.75 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
mean_funnel_size	3.00	2.50 (16.67%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_funnel_size	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
max_out_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_out_degree	3.00	2.00 (33.33%)	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)
max_in_degree	3.00	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)	3.00 (0.00%)
avg_in_degree	3.00	2.00 (33.33%)	2.75 (8.33%)	3.00 (0.00%)	3.00 (0.00%)
density	1.33	1.00 (25.00%)	1.25 (6.25%)	1.33 (0.00%)	1.33 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.42 (41.67%)	1.08 (8.33%)	1.00 (0.00%)	1.00 (0.00%)
largest_clique_size	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
reciprocity	1.00	0.50 (50.00%)	0.91 (9.09%)	1.00 (0.00%)	1.00 (0.00%)

Tabela 4.27 Instancja cliques_11 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	3628800.00	1001.00	1906.00	11996.00	102096.00
oracle_count	941261556.00	182710.00	340065.00	2181245.00	18669915.00
node_count	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
edge_count	36.00	32.00 (11.11%)	35.00 (2.78%)	35.00 (2.78%)	36.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	6.00	5.33 (11.11%)	5.83 (2.78%)	5.83 (2.78%)	6.00 (0.00%)
avg_fitness	991.67	991.67 (0.00%)	991.67 (0.00%)	991.67 (0.00%)	991.67 (0.00%)
distLO	0.01	0.26 (2667.10%)	0.15 (1466.64%)	0.02 (130.30%)	0.00 (85.78%)
conrel	5.00	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_loop_weight	729.67	141.83 (80.56%)	254.33 (65.14%)	1638.50 (124.55%)	13907.67 (1806.03%)
go_path_ratio	0.83	0.83 (0.00%)	0.83 (0.00%)	0.83 (0.00%)	0.83 (0.00%)
avg_go_path_len	1.00	1.20 (20.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	2.00 (100.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
mean_funnel_size	5.00	4.50 (10.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
max_funnel_size	5.00	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
max_out_degree	5.00	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_out_degree	5.00	4.33 (13.33%)	4.83 (3.33%)	4.83 (3.33%)	5.00 (0.00%)
max_in_degree	5.00	4.33 (13.33%)	4.83 (3.33%)	4.83 (3.33%)	5.00 (0.00%)
avg_in_degree	5.00	1.07 (11.11%)	1.17 (2.78%)	1.17 (2.78%)	1.20 (0.00%)
density	1.20	1.07 (11.11%)	1.17 (2.78%)	1.17 (2.78%)	1.20 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.13 (13.33%)	1.03 (3.33%)	1.03 (3.33%)	1.00 (0.00%)
largest_clique_size	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
reciprocity	1.00	0.85 (15.38%)	0.97 (3.45%)	0.97 (3.45%)	1.00 (0.00%)

Tabela 4.28 Instancja cliques_11 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	3628800.00	111.00	1675.00	10010.00	1000015.00
oracle_count	941261556.00	30696.00	608502.00	3969080.00	410313529.00
node_count	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
edge_count	36.00	22.00 (38.89%)	29.00 (19.44%)	31.00 (13.89%)	36.00 (0.00%)
num_sub sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
edge_to_node	6.00	3.67 (38.89%)	4.83 (19.44%)	5.17 (13.89%)	6.00 (0.00%)
avg_fitness	991.67	991.67 (0.00%)	991.67 (0.00%)	991.67 (0.00%)	991.67 (0.00%)
distLO	0.01	0.83 (8602.17%)	0.25 (2517.90%)	0.22 (2146.49%)	0.02 (108.35%)
conrel	5.00	2.00 (60.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_loop_weight	729.67	4.33 (99.41%)	19.33 (97.35%)	43.67 (94.02%)	429.83 (41.09%)
go_path_ratio	0.83	0.33 (60.00%)	0.83 (0.00%)	0.83 (0.00%)	0.83 (0.00%)
avg_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
max_go_path_len	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
num_sinks	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sources	4.00	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)	4.00 (0.00%)
mean_funnel_size	5.00	3.00 (40.00%)	4.00 (20.00%)	4.00 (20.00%)	5.00 (0.00%)
max_funnel_size	5.00	3.00 (40.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
max_out_degree	5.00	4.00 (20.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_out_degree	5.00	2.67 (46.67%)	3.83 (23.33%)	4.17 (16.67%)	5.00 (0.00%)
max_in_degree	5.00	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)	5.00 (0.00%)
avg_in_degree	5.00	2.67 (46.67%)	3.83 (23.33%)	4.17 (16.67%)	5.00 (0.00%)
density	1.20	0.73 (38.89%)	0.97 (19.44%)	1.03 (13.89%)	1.20 (0.00%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	6.00	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
largest_cc_radius	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
avg_path_len	1.00	1.24 (23.81%)	1.27 (26.67%)	1.17 (16.67%)	1.00 (0.00%)
largest_clique_size	6.00	4.00 (33.33%)	6.00 (0.00%)	6.00 (0.00%)	6.00 (0.00%)
reciprocity	1.00	0.50 (50.00%)	0.70 (30.43%)	0.80 (20.00%)	1.00 (0.00%)

Tabela 4.29 Instancja grid_11 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.

metryka	p. zupełny	snowball_1	snowball_10	snowball_100	snowball_1000
opt_count	3628800.00	1401.00	16501.00	40156.00	142756.00
oracle_count	852898736.00	221870.00	2660240.00	6491430.00	22904695.00
node_count	46.00	34.00 (26.09%)	46.00 (0.00%)	46.00 (0.00%)	46.00 (0.00%)
edge_count	1467.00	141.00 (90.39%)	846.00 (42.33%)	1082.00 (26.24%)	1230.00 (16.16%)
num_sub sinks	18.00	31.00 (72.22%)	19.00 (5.56%)	18.00 (0.00%)	18.00 (0.00%)
edge_to_node	31.89	4.15 (87.00%)	18.39 (42.33%)	23.52 (26.24%)	26.74 (16.16%)
avg_fitness	1194.57	1182.06 (1.05%)	1194.57 (0.00%)	1194.57 (0.00%)	1194.57 (0.00%)
distLO	0.26	0.55 (112.61%)	0.26 (0.76%)	0.17 (35.04%)	0.16 (38.38%)
conrel	8.20	0.42 (94.92%)	2.07 (74.80%)	2.29 (72.13%)	2.54 (69.04%)
avg_loop_weight	223.83	69.67 (68.87%)	183.93 (17.82%)	427.70 (91.08%)	1708.43 (663.29%)
go_path_ratio	0.98	0.32 (66.93%)	0.96 (2.22%)	0.98 (0.00%)	0.98 (0.00%)
avg_go_path_len	1.09	1.45 (33.58%)	1.36 (25.23%)	1.29 (18.37%)	1.27 (16.33%)
max_go_path_len	2.00	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)	2.00 (0.00%)
num_sinks	18.00	31.00 (72.22%)	19.00 (5.56%)	18.00 (0.00%)	18.00 (0.00%)
num_sources	5.00	24.00 (380.00%)	10.00 (100.00%)	8.00 (60.00%)	7.00 (40.00%)
mean_funnel_size	23.89	1.65 (93.11%)	15.63 (34.57%)	19.44 (18.60%)	20.50 (14.19%)
max_funnel_size	29.00	4.00 (86.21%)	26.00 (10.34%)	28.00 (3.45%)	29.00 (0.00%)
max_out_degree	43.00	15.00 (65.12%)	26.00 (39.53%)	34.00 (20.93%)	37.00 (13.95%)
avg_out_degree	30.89	3.79 (87.72%)	17.41 (43.63%)	22.52 (27.09%)	25.74 (16.68%)
max_in_degree	45.00	10.00 (77.78%)	37.00 (17.78%)	41.00 (8.89%)	44.00 (2.22%)
avg_in_degree	30.89	3.79 (87.72%)	17.41 (43.63%)	22.52 (27.09%)	25.74 (16.68%)
density	0.71	0.13 (82.27%)	0.41 (42.33%)	0.52 (26.24%)	0.59 (16.16%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	46.00	34.00 (26.09%)	46.00 (0.00%)	46.00 (0.00%)	46.00 (0.00%)
largest_cc_radius	1.00	2.00 (100.00%)	2.00 (100.00%)	2.00 (100.00%)	2.00 (100.00%)
avg_path_len	1.31	1.87 (42.07%)	1.65 (25.57%)	1.51 (14.82%)	1.43 (8.75%)
largest_clique_size	29.00	7.00 (75.86%)	12.00 (58.62%)	14.00 (51.72%)	16.00 (44.83%)
reciprocity	0.70	0.37 (46.70%)	0.57 (18.45%)	0.66 (5.98%)	0.73 (4.53%)

Tabela 4.30 Instancja grid_11 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.

metryka	p. zupełny	dwufazowy_10	dwufazowy_40	dwufazowy_100	dwufazowy_1000
opt_count	3628800.00	114.00	2177.00	10579.00	1001183.00
oracle_count	852898736.00	19106.00	394010.00	2713661.00	330519796.00
node_count	46.00	10.00 (78.26%)	36.00 (21.74%)	46.00 (0.00%)	46.00 (0.00%)
edge_count	1467.00	29.00 (98.02%)	430.00 (70.69%)	789.00 (46.22%)	1266.00 (13.70%)
num_sub sinks	18.00	9.00 (50.00%)	17.00 (5.56%)	18.00 (0.00%)	18.00 (0.00%)
edge_to_node	31.89	2.90 (90.91%)	11.94 (62.55%)	17.15 (46.22%)	27.52 (13.70%)
avg_fitness	1194.57	1149.20 (3.80%)	1184.28 (0.86%)	1194.57 (0.00%)	1194.57 (0.00%)
distLO	0.26	0.73 (181.69%)	0.36 (39.83%)	0.67 (156.08%)	0.21 (20.55%)
conrel	8.20	1.50 (81.71%)	3.00 (63.41%)	0.70 (91.42%)	14.33 (74.80%)
avg_loop_weight	223.83	3.50 (98.44%)	9.66 (95.69%)	16.96 (92.42%)	167.65 (25.10%)
go_path_ratio	0.98	0.60 (38.67%)	0.97 (0.62%)	0.98 (0.00%)	0.98 (0.00%)
avg_go_path_len	1.09	1.17 (7.14%)	1.31 (20.70%)	1.96 (79.59%)	1.11 (2.04%)
max_go_path_len	2.00	2.00 (0.00%)	2.00 (0.00%)	3.00 (50.00%)	2.00 (0.00%)
num_sinks	18.00	9.00 (50.00%)	17.00 (5.56%)	18.00 (0.00%)	18.00 (0.00%)
num_sources	5.00	8.00 (60.00%)	19.00 (280.00%)	17.00 (240.00%)	8.00 (60.00%)
mean_funnel_size	23.89	1.22 (94.88%)	8.71 (63.56%)	15.33 (35.81%)	22.28 (6.74%)
max_funnel_size	29.00	2.00 (93.10%)	18.00 (37.93%)	29.00 (0.00%)	29.00 (0.00%)
max_out_degree	43.00	4.00 (90.70%)	19.00 (55.81%)	23.00 (46.51%)	38.00 (11.63%)
avg_out_degree	30.89	1.90 (93.85%)	10.97 (64.48%)	16.15 (47.71%)	26.52 (14.14%)
max_in_degree	45.00	5.00 (88.89%)	31.00 (31.11%)	43.00 (4.44%)	45.00 (0.00%)
avg_in_degree	30.89	1.90 (93.85%)	10.97 (64.48%)	16.15 (47.71%)	26.52 (14.14%)
density	0.71	0.32 (54.53%)	0.34 (51.85%)	0.38 (46.22%)	0.61 (13.70%)
num_cc	1.00	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)	1.00 (0.00%)
largest_cc	46.00	10.00 (78.26%)	36.00 (21.74%)	46.00 (0.00%)	46.00 (0.00%)
largest_cc_radius	1.00	2.00 (100.00%)	2.00 (100.00%)	2.00 (100.00%)	1.00 (0.00%)
avg_path_len	1.31	1.47 (11.82%)	1.97 (50.27%)	1.79 (36.62%)	1.41 (7.43%)
largest_clique_size	29.00	4.00 (86.21%)	11.00 (62.07%)	14.00 (51.72%)	22.00 (24.14%)
reciprocity	0.70	0.00 (100.00%)	0.33 (52.13%)	0.40 (42.55%)	0.60 (13.82%)

Najbardziej rzuającymi się w oczy metrykami są avg_loop_weight i distLO. We wszystkich przypadkach występuje duża rozbieżność między wartościami tych miar uzyskanymi z próbkowania, a tymi z przeglądu zupełnego. Nie jest to zaskakujące. Algorytm przeglądu zupełnego jednokrotnie sprawdza każde rozwiązanie w przestrzeni przeszukiwania pod kątem możliwości utworzenia krawędzi między optimami lokalnymi, lub zwiększenia wagi istniejącej. Po sprawdzeniu wszystkich rozwiązań algorytm kończy pracę. Algorytmy próbkowania nie kończą pracy aż do momentu osiągnięcia warunku zatrzymania. Ze względu na swój losowy charakter, po operacji perturbacji często trafiają na te same rozwiązania (szczególnie w przypadku małych instancji) i powodują zwiększenie wagi krawędzi, w tym pętli, do wartości znacznie większych od rzeczywistych. Dlatego metryki, których wartość zależy od bezwzględnych wartości wag, a nie od stosunku wag różnych krawędzi nie dają użytecznej informacji na temat małych instancji. Z tego też powodu ich wartość nie będzie brana pod uwagę w ocenie działania algorytmu.

W przypadku najmniejszych instancji o rozmiarze 7, błąd uzyskany po porównaniu wartości metryk wynosił 0 lub był bliski 0 dla wszystkich metryk oprócz avg_loop_weight i distLO. Wyjątkiem była instancja uniform_7, gdzie wartości błędów były wyższe dla obu algorytmów. W tej instancji algorytm snowball poradził sobie znacznie lepiej — brak błędu dla większości metryk wystąpił już dla parametru w_{len} równego 10, natomiast algorytm dwufazowy podobny wynik uzyskał dopiero dla parametru n_{max} równego 1000 — wymagało to ponad 600 razy więcej wywołań funkcji optymalizującej. Dla instancji tej wielkości przegląd zupełny jest szybki, dlatego nie ma sensu stosować dla nich próbkowania.

Obu algorytmom udało się dobrze odwzorować cechy przestrzeni rozwiązań instancji uniform_8 i grid_8 (z wyjątkiem distLO i avg_loop_weight), w czasie próbkowania kilkukrotnie krótszym od czasu trwania przeglądu zupełnego. Wyjątkiem okazała się instancja cliques_8, której dokładne spróbkowanie udało się dopiero dla n_{max} równego 100 — próbkowanie trwało prawie dwa razy tyle czasu, co przegląd zupełny. Algorytm snowball poradził sobie znakomicie już dla wartości w_{len} równej 10 — próbkowanie było ponad dwukrotnie krótsze od przeglądu zupełnego.

Dla instancji uniform_9 i cliques_9 oba algorytmy dobrze odwzorowały przestrzeń w czasie znacznie krótszym od czasu przeglądu zupełnego. Jednak w przypadku instancji grid_9 nawet najdłuższe przebiegi testowanych algorytmów nie pozwoliły na uzyskanie zerowego błędu dla wszystkich miar z wyłączeniem distLO i avg_loop_weight. Algorytm snowball ujawnia tu swoją wadę — mimo znalezienia wszystkich krawędzi i wierzchołków wartości miar largest_clique_size i reciprocity nie zgodne z wartościami uzyskany mi z przeglądu zupełnego. Jest to spowodowane tym, że zarówno przegląd zupełny jak i algorytm dwufazowy używają algorytmu 2opt typu fist-improvement w procedurze perturbacji. Algorytm snowball używa zaś optymalizacji typu best-improvement, co powoduje nieznaczne, ale widoczne różnice topologii zbudowanej sieci LON. Pomimo korzystania z 2opt typu fist-improvement, algorytm dwufazowy nie poradził sobie z dokładnym odwzorowaniem przestrzeni instancji grid_9 i ostatecznie lepszy w tym przypadku okazał się algorytm snowball.

W przypadku instancji o rozmiarze 10 coraz bardziej widoczna staje się różnica między poszczególnymi typami instancji — instancje z miastami ulożonymi w klikach nie sprawiają problemów — oba algorytmy dla większości miar otrzymują wartość prawdziwą w czasie krótszym od przeglądu zupełnego. Instancja uniform_10 nie jest już idealnie odwzorowywana — algorytmy nie docierają do niektórych krawędzi i pojawiają się niewielkie błędy. Instancja grid_10 sprawia problemy obu algorytmom. Warto jednak zauważyć, że wartości błędów dla przebiegu snowball_1000 są mniejsze od błędów z przebiegu dwufa-

zowy _1000 pomimo tego, że ten drugi trwa 10 razy dłużej.

Instancja uniform_11 okazała się rzadkim przypadkiem, w którym algorytm dwufazowy poradził sobie lepiej od algorytmu snowball w podobnym czasie. Snowball okazuje się lepszy dla instancji cliques_11 — oba algorytmy w najdłuższym przebiegu znajdują prawdziwą wartość większości miar, ale snowball robi to szybciej. Instancja grid_11 zawiera dużą ilość wierzchołków i krawędzi — nawet najdłuższe przebiegi obu algorytmów nie były w stanie znaleźć wszystkich, co poskutkowało znacznymi błędami.

4.3.2 Badanie stabilności dla średnich instancji

Aby zbadać wpływ długości i metody próbkowania na wartości miar przestrzeni rozwiązań, wykonano długotrwałe próbkowanie przestrzeni, podczas którego zapisywano jej stan w postaci sieci LON w określonych interwałach. Na podstawie zebranych danych obliczono wartości miar w poszczególnych etapach i przedstawiono je w relacji do wielkości spróbowanej przestrzeni, wyrażonej jako liczba obecnych w niej wierzchołków. Pozwoliło to na zidentyfikowanie miar stabilnych i niestabilnych oraz na obserwację różnic w działaniu poszczególnych algorytmów próbkowania.

Instancje **burma14**, **ulysses22** ze zbioru tsplib oraz wygenerowane instancje **cliques_20**, **uniform_20** oraz **grid_20** są zbyt małe, żeby próbkować je z takimi samymi parametrami, jak większe instancje. Z tego powodu badania na nich przeprowadzone umieszczone w osobnej sekcji. Dla instancji tego typu wykonano bardziej szczegółowe próbkowanie z mniejszymi interwałami zapisu. Badania większych instancji przedstawione w sekcji 4.3.3.

Próbkowanie algorytmem snowball wykonano z następującymi parametrami:

- w_{len} — długość losowego spaceru — 10000
- m — liczba prób przeszukania sąsiedztwa — 100
- $depth$ — głębokość przeszukiwania — 3

Zapis wyników wykonywano przy każdym znalezionym wierzchołku. Wyjątkiem była instancja grid_20 — tu zapis wykonywano co 100 wierzchołków z powodu dużej liczby optimów lokalnych.

Próbkowanie algorytmem dwufazowym przeprowadzono z następującymi parametrami:

- n_{max} — żądana liczba wierzchołków — 1
- n_{att} — liczba prób generowania wierzchołków — 1000
- e_{att} — liczba prób generowania krawędzi — 10
- n_{runs} — liczba powtórzeń — 1000

Zapis wyników wykonywano po zakończeniu każdego powtórzenia.

Utworzono wykresy przedstawiające zależność wartości miar od liczby spróbowanych wierzchołków. Przedstawiono je na rysunkach od A.1 do A.27.

Wyniki

Wykresy nie są wymagane do zrozumienia wyników. Zostały one zawarte w załączniku. Dla wygody czytelnika w opisie wyników umieszczone odnośniki do poszczególnych wykresów.

Zależność liczby krawędzi od liczby wierzchołków jest nieliniowa dla wszystkich zbadanych instancji i obu algorytmów (rys. A.1, rys. A.2). W przypadku metryki assortativity_deg można zaobserwować pewną właściwość. Na początku próbkowania wartość metryki zmienia się gwałtownie, zwiększaając i zmniejszając się naprzemiennie (rys. A.3). Po pewnym czasie jednak się stabilizuje, widoczne jest to zwłaszcza dla instancji z większą ilością optimów lokalnych. Nadal obecne są wahania wartości, ale są one znacznie mniejsze. Wyjątkiem jest instancja grid_20 próbkowana algorymem snowball. W tym przypadku wartość zmienia się znacznie przez cały okres próbkowania. W wielu przypadkach ostatnia uzyskana wartość różni się znacznie pomiędzy algorytmami.

W większości przypadków średnia wartość funkcji celu przestrzeni zwiększa się wraz z ilością wierzchołków (rys. A.4). W niektórych przypadkach występuje początkowo duża wartość i gwałtowny spadek, zwykle w przypadku próbkowania dwufazowego. W innych przypadkach wartość wzrasta przez cały czas. Wykresy wartości uzyskane z obu algorytmów różnią się, ale wartości uzyskane na końcu próbkowania są do siebie podobne. Niska wartość funkcji celu na początku i powolny wzrost oznacza, że w badanych instancjach szybko znajdowane były najlepsze rozwiązania w przestrzeni.

Wartość współczynnika conrel jest nieprzewidywalna (rys. A.5). Dla niektórych instancji utrzymuje się na podobnym poziomie przez większość czasu, ale gwałtownie zwiększa się pod koniec próbkowania. W innych wartość miary znacznie zmienia się w trakcie procesu, formując ostre piki. Gęstość grafu w każdym przypadku osiągała wysoką wartość, by następnie gwałtownie spaść i ustabilizować się w pewnym zakresie (rys. A.6). W kilku przypadkach występował wzrost gęstości grafu pod koniec próbkowania. Miara jest najbardziej stabilna dla instancji z miastami rozłożonymi równomiernie. Miara distLO jest niestabilna, występują częste nagle wzrosty i spadki jej wartości (rys. A.7). W kilku przypadkach wystąpił duży spadek pod koniec próbkowania, w innych — gwałtowny wzrost. Współczynnik wzajemności (reciprocity) również jest niestabilny (rys. A.8). W niektórych przypadkach występuje gwałtowny wzrost i spadek wartości na początku próbkowania. Ciekawa jest instancja grid_20 — po spróbkowaniu odpowiednio dużej liczby wierzchołków wzrost tej miary jest zbliżony do wzrostu liniowego.

Rozmiar największej kliki zmienia się „schodkowo” w momencie, w którym albo ze spróbkowanych wierzchołków powstanie większa klika lub nowy wierzchołek zostanie odnaleziony w już istniejącej (rys. A.9). Wyniki sugerują, że możliwe jest oszacowanie wielkości największej kliki przestrzeni na podstawie wartości z krótkiego próbkowania, zwłaszcza na podstawie danych z próbkowania dwufazowego, gdzie wartość zmienia się mniej gwałtownie. Ze względu na stopniowy i nieliniowy charakter wzrostu zmiennej, dokładność takiego szacowania nie będzie duża. Liczba źródeł, ścieków i tzw. subsinks w grafie w wielu przypadkach początkowo rośnie, by nagle zmaleć na końcu próbkowania. W przypadkach ze zbioru TSPLIB po pewnym czasie liczba ścieków i subsinks przestała się zmieniać. Wszystkie trzy miary są niestabilne, mają charakter nieprzewidywalny (rys. A.11, rys. A.10, rys. A.12).

Charakterystyka wzrostu wartości średniego stopnia wchodzącego wierzchołka jest bardziej zbliżona do liniowej w przypadku próbkowania dwufazowego. W instancjach cliques_20, uniform_20 i burma14 występuje przyspieszenie wzrostu pod koniec próbkowania (rys. A.13). W tych przypadkach z dużym prawdopodobieństwem algorytmy znalazły wszystkie optima lokalne. Może być to powodem tego zjawiska. W przypadku instancji

grid_20 o liczbie optimów lokalnych znacznie wyższej od pozostałych instancji tego rozmiaru, wynik z próbkowania snowball ma charakter eksponencjalny. Podobnie wyglądają miary największego stopnia wchodzącego, średniego stopnia wychodzącego i maksymalnego stopnia wychodzącego (rys. A.14, rys. A.15, rys A.16).

Średnia waga pętli w grafie jest niestabilna, w niektórych przypadkach wzrost jest bliski liniowego, w innych eksponencjalnego, a w pozostałych jest chaotyczny (rys. A.17). Średnia długość ścieżki w grafie zwiększa się gwałtownie na początku, po czym stabilizuje (rys. A.18). Pojawiają się przypadki odstające (w grid_20 przy próbkowaniu snowball wartość dość szybko maleje), ale dla większości przypadków wahania wartości po ustabilizowaniu są niewielkie. Podobnie jest w przypadku średniej długości ścieżek do globalnego optimum. Długość najdłuższej ścieżki do globalnego optimum zmniejsza i zwiększa się gwałtownie. Jest nieprzewidywalna, podobnie jest z wartością maksymalną (rys. A.19, rys. A.20).

Współczynnik go_path_ratio zmienia się zupełnie inaczej w zależności od algorytmu próbkowania (rys. A.21). Próbkowanie dwufazowe rośnie gwałtownie i bardzo wcześnie się stabilizuje. Próbkowanie snowball prowadzi natomiast do powolnego wzrostu z okazjonalnymi spadkami wartości. Oba algorytmy osiągają podobną końcową wartość miary, ale próbkowanie dwufazowe osiąga ją przy mniejszej liczbie spróbkowanych wierzchołków.

Rozmiar największego leja w przestrzeni rośnie liniowo dla danych uzyskanych z próbkowania dwufazowego. Wartość uzyskana z próbkowania snowball nie jest tak stabilna (rys. A.22). Wzrost średniego rozmiaru lejów dla próbkowania dwufazowego również jest bliski liniowego, widać jednak na nim pewne odchylenia (rys. A.23). W ramach wzrostu ilości optimów lokalnych w przestrzeni, wykres uzyskany ze snowball coraz bardziej zbliża się do wzrostu eksponencjalnego.

Liczba spójnych podgrafów szybko stabilizuje się na stałej wartości dla większości przypadków (rys. A.24). Rozmiar największego spójnego podgrafa rośnie liniowo dla wszystkich instancji i algorytmów (rys. A.25). Promień największego spójnego podgrafa zmienia się wielokrotnie o wartość 1 dla większości przypadków (rys. A.26).

Im więcej wierzchołków znajduje się w zbiorze, tym więcej prób tworzenia krawędzi w algorytmie dwufazowym się powiedzie. Dlatego można zaobserwować delikatne wypłaszczenie krzywej na wykresie liczby nieudanych prób. Wyjątkiem jest tu instancja grid_20 — jest to spowodowane tym że algorytm nie znalazł wszystkich optimów lokalnych przed końcem próbkowania (rys. A.27).

4.3.3 Badanie stabilności dla dużych instancji

W tej sekcji opisano badania przeprowadzone dla instancji o rozmiarze 48 i większym. Podobnie jak w eksperymentach opisanych w sekcji 4.3.2, wykonano długotrwałe próbkiwanie przestrzeni rozwiązań i przedstawiono uzyskane wartości miar na wykresach. Ponieważ instancje badane w tej sekcji cechują się znacznie większym rozmiarem przestrzeni rozwiązań, próbkiwanie przeprowadzono dla innego zestawu parametrów, wykonując zapisy w większych interwałach.

Próbkowanie algorytmem *snowball* wykonano z następującymi parametrami:

- w_{len} — długość losowego spaceru — 10000
- m — liczba prób przeszukania sąsiedztwa — 100
- $depth$ — głębokość przeszukiwania — 3

Próbkowanie prowadzono do momentu zakończenia losowego spaceru lub osiągnięcia liczby 100000 wierzchołków. Dla instancji **cliques_50** z powodu małej liczby optimów lokalnych zapis wykonywano co 100 znalezionych wierzchołków. Dla pozostałych instancji stan próbkowania zapisywany był co 1000 znalezionych wierzchołków.

Próbkowanie algorytmem dwufazowym przeprowadzono z następującymi parametrami:

- n_{max} — żądana liczba wierzchołków — 1000
- n_{att} — liczba prób generowania wierzchołków — 1000
- e_{att} — liczba prób generowania krawędzi — 1000
- n_{runs} — liczba powtórzeń — 100

Próbkowanie przeprowadzono przez ustaloną z góry liczbę powtórzeń, zapisując stan przestrzeni po zakończeniu każdego powtórzenia. Po zakończeniu działania algorytmów, dla każdego zapisanego stanu przestrzeni obliczono wartości badanych miar. W ten sposób uzyskano ich wartości w różnych etapach próbkowania przestrzeni. Utworzono wykresy przedstawiające zależność wartości miar od liczby wierzchołków w grafie LON. Przedstawiono je na rysunkach od A.28 do A.80.

Ze względu na długi czas obliczeń i ograniczony czas realizacji projektu, dla instancji **cliques_50**, **cliques_80**, **uniform_50**, **att48**, **berlin52** i **pr76** i próbkowania dwufazowego nie wykonano obliczeń wartości następujących miar:

- num_sub sinks
- num_cc
- largest_cc
- largest_cc_radius
- avg_path_len
- largest_clique_size
- reciprocity

Wyniki

Wykresy nie są wymagane do zrozumienia wyników. Zostały one zawarte w załączniku. Dla wygody czytelnika w opisie wyników umieszczone odnośniki do poszczególnych wykresów.

Wzrost liczby krawędzi w stosunku do liczby wierzchołków jest zbliżony do liniowego dla próbkowania *snowball* (rys. A.28). Na wykresie wartości stosunku krawędzi do wierzchołków (rys. A.29), w niektórych przypadkach widać co prawda gwałtowne zmiany, ale ich bezwzględna wartość jest niewielka. W przypadku próbkowania dwufazowego (rys. A.54, rys. A.55), ta zależność nie jest liniowa dla części instancji, najlepiej widać to w przypadku instancji *cliques_50*, *cliques_80*, *att48*, *berlin52*, oraz *pr76*. Dla instancji typu *grid* oraz pozostałych instancji ze zbioru *tsplib* zależność ma charakterystykę liniową lub zbliżoną do liniowej.

Współczynnik różnorodności grafu nie jest stabilny. Wykres jego wartości wygląda inaczej w niemal każdym przypadku próbkowania *snowball* (rys. A.30). Dla próbkowania dwufazowego (rys. A.56) można zaobserwować eksponencjalny spadek wartości taj miary dla części instancji. W innych przypadkach wzrasta ona szybko i oscyluje wokół pewnej wartości. Odstającym przypadkiem są instancje *grid_80* i *grid_100*. Wykres pokazuje dla nich stałą wartość 0. W rzeczywistości otrzymano tutaj wartość nieprawidłową (NaN). Najprawdopodobniej mianownik wyrażenia 3.2 miał wartość 0.

Dla większości instancji o rozmiarze 50 średnia wartość funkcji celu jest niska na początku i wzrasta podczas dalszego próbkowania. Zależność tą możemy zaobserwować zarówno w przypadku próbkowania *snowball* jak i dwufazowego (rys. A.31, rys. A.57). W przypadku instancji większych oraz typu *grid* po początkowym spadku wartości, maleje ona dalej lub stabilizuje się na pewnym poziomie. Nie jest to zaskakujące. W mniejszych instancjach problemu występuje mniej lokalnych optimów, więc zwiększa się szansa, że algorytm natrafi na nie podczas krótszego próbkowania. Instancje większe zawierają znacznie więcej optimów lokalnych, więc nawet po długim próbkowaniu algorytm wciąż natrafia na nowe o niższej wartości funkcji celu. Instancje typu *grid* charakteryzują się dużą liczbą optimów lokalnych, więc problem ten występuje również dla mniejszych instancji tego typu.

Wartość współczynnika *conrel* jest niestabilna. Różni się ona w zależności od algorytmu próbkowania (rys. A.32, rys. A.58). Dla większych instancji typu *grid* i próbkowania dwufazowego ma ona stałą wartość 0. Wartość gęstości grafu (*density*) zmienia się w przewidywalny sposób dla próbkowania *snowball* (rys. A.33) i maleje eksponencjalnie. W przypadku próbkowania dwufazowego (rys. A.59) taki spadek występuje tylko w instancjach typu *grid*, instancji *cliques_100* oraz większych instancjach ze zbioru *tsplib*. Dla pozostałych instancji występują kolejne wzrosty i spadki wartości tej miary. Z otrzymanych danych można wywnioskować, że wartość tej metryki jest w dużym stopniu zależna od wykorzystanego algorytmu próbkowania. Wartość współczynnika *distLO* jest niestabilna. Zmienia się w nieprzewidywalny sposób zarówno dla algorytmu *snowball* jak i dwufazowego (rys. A.34, rys. A.60). Ciekawym przypadkiem są instancje *grid* dla próbkowania *snowball* — wartość *distLO* jest dla nich stała i wynosi 1. Oznacza to, że każda krawędź połączona z najlepszym rozwiązaniem miała wagę 1 przez cały okres próbkowania.

Wzajemność grafu (*reciprocity*) w większości przypadkach gwałtownie zmienia się na początku, a potem stabilizuje w okolicach pewnej wartości. W przypadku instancji *grid_80* i *grid_100* otrzymano wartości nieprawidłowe (NaN) dla całego okresu próbkowania dwufazowego (rys. A.61). Jest to spowodowane tym, że wszystkie spróbkowane w tych instancjach krawędzie są pętlami. Podobne nieprawidłowe wartości występują również na początku próbkowania dużych instancji rzeczywistych. Jest to spowodowane działaniem

algorytmu próbkowania. Na początku wiele optimów lokalnych nie znajduje się w zbiorze wierzchołków. Z tego powodu nie udaje się utworzyć dużej ilości krawędzi prowadzących do nieodkrytych jeszcze wierzchołków. Im większa przestrzeń przeszukiwania, tym dłużej będzie występował ten problem. Zjawisko to nie występuje dla algorytmu *snowball*, gdyż tam wierzchołki są próbkowane równocześnie z krawędziami i każda krawędź, która może powstać jest tworzona od razu (rys. A.35).

Rozmiar największej klasy w grafie wzrasta „schodkowo” dla obu algorytmów (rys. A.36, (rys. A.62)). Dla największych instancji jego wartość nie zmienia się przez cały okres próbkowania. Z danych można wywnioskować, że im mniejsza ilość lokalnych optimów w przestrzeni przeszukiwania, tym częściej zmienia się wartość tej miary. Stała wartość dla dużych instancji nie oznacza, że można ją przyjąć za wartość miary dla całej przestrzeni — najprawdopodobniej zmieni się ona po jeszcze dłuższym próbkowaniu.

Liczba źródeł w grafie rośnie liniowo dla większości instancji przy próbkowaniu *snowball* (rys. A.37). W przypadku próbkowania dwufazowego zależność liniowa występuje jedynie w przypadku największych instancji i instancji typu grid (rys. A.63). Ciekawym przypadkiem jest instancja *cliques_50*, w której jako jedynej wartość tej miary zaczyna spadać pod koniec próbkowania. Liczba ścieków w grafie również rośnie liniowo dla większości instancji w przypadku próbkowania *snowball* (rys. A.38). W przypadku próbkowania dwufazowego jest zupełnie inaczej. O ile liniowy wzrost wciąż występuje dla niektórych instancji, to w innych przypadkach wartość ta maleje lub utrzymuje się na stałym poziomie. Instancja *cliques_50* również się tu wyróżnia — w przestrzeni rozwiązań występują jedynie dwa ścieki przez cały okres próbkowania (rys. A.64). Wykresy ilości subsinks wyglądają niemal identycznie do wykresów ilości ścieków — oznacza to, że ich liczba jest zawsze zbliżona (rys. A.39, rys. A.65).

Średni stopień wchodzący wierzchołków jest miarą nieprzewidywalną dla próbkowania *snowball* (rys. A.40). W przypadku próbkowania dwufazowego wartość miary jest bardziej przewidywalna — przez cały czas wzrasta dla większości przypadków (rys. A.66). Wyjątkiem są instancje *grid_80* i *grid_100*, w których przez cały okres próbkowania otrzymywana jest wartość 0. Maksymalny stopień wchodzący dla części instancji wzrasta „schodkowo”, a dla innych łagodniej (rys. A.41, rys. A.67). W przypadku algorytmu dwufazowego dla niektórych instancji wzrost jest bliski liniowego. Instancje *grid_80* i *grid_100* mają maksymalny stopień wchodzący równy 0. Oznacza to, że wszystkie krawędzie w tych grafach to pętle. Wyjaśnia to również wartość *distLO* równą 1 oraz nieprawidłową wartość *reciprocity*. Podobnie prezentują się wykresy wartości średniej i maksymalnej stopnia wychodzącego (rys. A.42, rys. A.43, rys. A.68, rys. A.69).

Średnia waga pętli w grafie jest nieprzewidywalna dla próbkowania *snowball* (rys. A.44, rys. A.70). Dla próbkowania dwufazowego wielkość ta wzrasta niemal liniowo dla każdej instancji oprócz *cliques_50*. Średnia długość ścieżki w grafie jest nieprzewidywalna dla większości instancji próbkowanych algorymem *snowball* (rys. A.45). Wyjątkiem są instancje typu grid, dla których wartość tej metryki stale wzrasta. Dla próbkowania dwufazowego również nie uzyskano stabilnych wartości. Pętle nie sąbrane pod uwagę w obliczaniu wartości tej metryki. Z tego powodu wynosi ona 0 dla największych instancji typu grid (rys. A.71). Średnia długość ścieżki do najlepszego rozwiązania jest nieprzewidywalna niezależnie od algorytmu próbującego (rys. A.46, rys. A.72). Nagłe zmiany w wartości są spowodowane najprawdopodobniej znalezieniem nowego najlepszego rozwiązania w przestrzeni. Podobnie jest w przypadku wartości maksymalnej (rys. A.47, rys. A.73). Współczynnik *go_path_ratio* jest nieprzewidywalny. O ile w niektórych przypadkach przy próbkowaniu dwufazowym stabilizuje się na pewnej wartości po początkowym wzroście, to w innych gwałtownie zmniejsza się i zwiększa (rys. A.48, rys. A.74).

Rozmiar największego leja dla niektórych instancji wzrasta liniowo. Dla innych wzrasta „schodkowo”, a w przypadku instancji grid_80 i grid_100 dla próbkowania dwufazowego — ma stałą wartość 1. Oznacza to, że nie da się dokładnie przewidzieć wartości tej miary. Warto jednak zauważyć, że „schodkowe” zwiększenie się wartości tej miary występuje w instancjach o największej liczbie lokalnych optimów. Widoczne są wyraźne różnice pomiędzy wykresami uzyskanymi z próbkowania *snowball* i dwufazowego (rys. A.49, rys. A.75). Nieco inaczej przedstawiają się wykresy średniej wielkości lejów. O ile dla próbkowania *snowball* wartość zmienia się w sposób całkowicie nieprzewidywalny, to wszystkie wykresy uzyskane z algorytmu dwufazowego (oprócz grid_80 i grid_100) wykazują tendencję wzrostową (choć w niektórych instancjach pojawiają się małe spadki wartości.) (rys. A.50, rys. A.76).

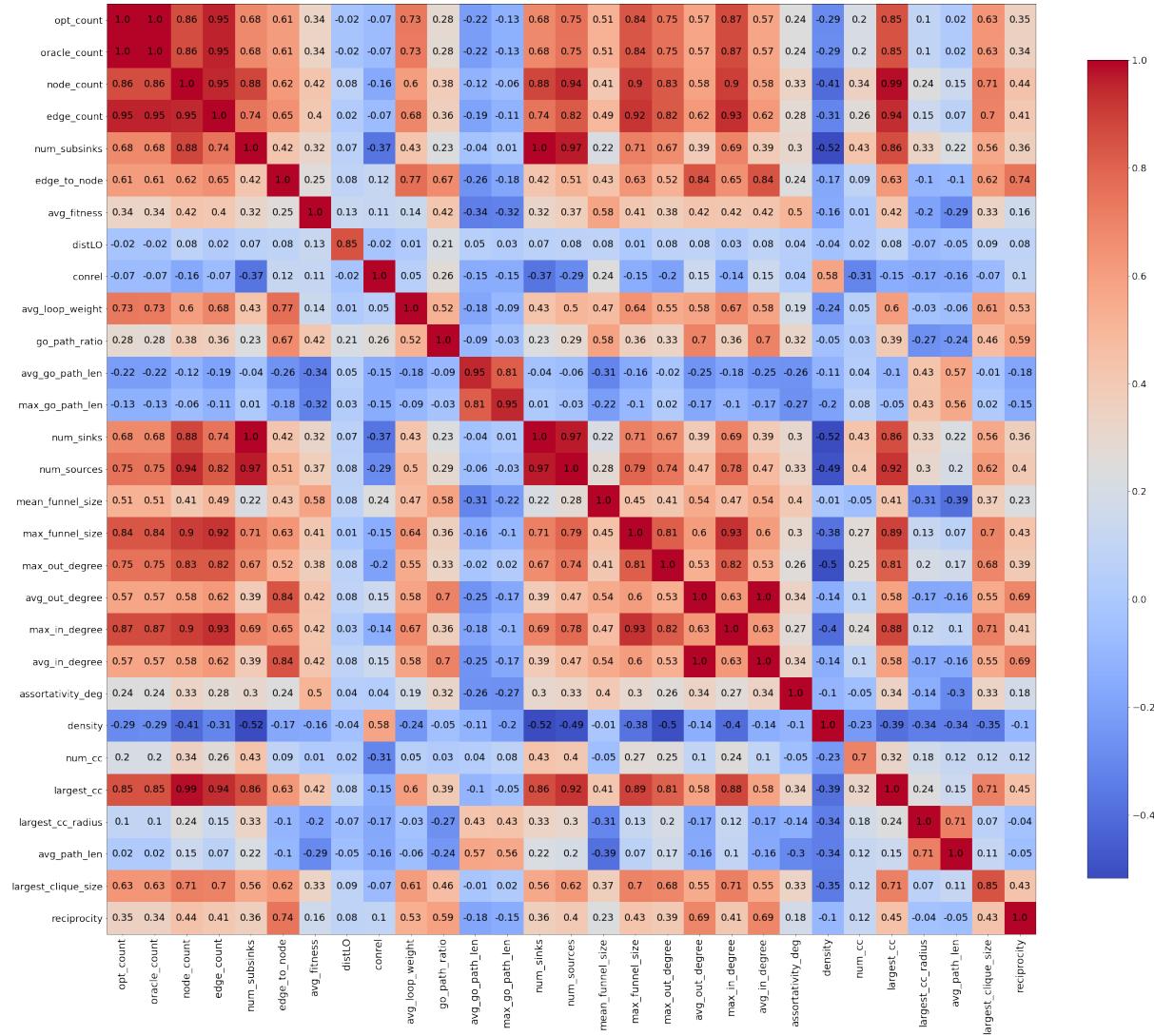
Liczba spójnych podgrafów sieci LON jest stała dla wielu instancji w przypadku próbkowania *snowball* (rys. A.51). Dla innych zmienia się ona w nieprzewidywalny sposób. W przypadku próbkowania dwufazowego i instancji grid liczba podgrafów jest równa liczbie wierzchołków, gdyż każdy wierzchołek w tych grafach jest wierzchołkiem izolowanym. W przypadku instancji ze zbioru tsplib, liczba ta nie jest równa liczbie wierzchołków, ale jest bardzo bliska — oznacza to, że znaczna większość wierzchołków w tych grafach również jest izolowana (rys. A.77). W przypadku próbkowania *snowball* rozmiar największego spójnego podgrafa wzrasta liniowo dla większości instancji. Dla instancji uniform_100, oraz eil76, rat99 i bier127 potrafi utrzymywać się na stałym poziomie przez dłuższy czas i nagle rosnąć (rys. A.52). W przypadku próbkowania dwufazowego nie wzrasta on liniowo. Dla niektórych instancji wartość utrzymuje się na niskim poziomie przez długi czas (rys. A.78). Dla instancji rat99 i bier127 zmienia się częściej niż w przypadku próbkowania *snowball*. Promień największego spójnego podgrafa sieci zmienia się w sposób nieprzewidywalny dla obu algorytmów (rys. A.53, rys. A.79). Zależność liczby nieudanych prób tworzenia krawędzi przez algorytm dwufazowy wygląda bardzo podobnie dla wszystkich instancji (rys. A.80).

Na wykresach A.81 i A.82 możemy zobaczyć odpowiednio liczbę wierzchołków i liczbę krawędzi w zależności od liczby wywołań funkcji optymalizującej. Liczba wywołań funkcji optymalizującej jest zarazem liczbą wywołań funkcji celu, gdyż algorytm 2opt wykonuje pełną kalkulację długości ścieżki tylko raz. Na wykresach kolorem niebieskim oznaczono algorytm *snowball*, a pomarańczowym — algorytm dwufazowy. Oś X została przedstawiona w skali logarytmicznej. Na wykresie wyraźnie widać, że spróbkowanie tej samej liczby wierzchołków zawiera o wiele więcej czasu algorytmowi dwufazowemu. Należy jednak pamiętać, że w niektórych przypadkach (np. cliques_50) szybszy algorytm *snowball* nie odnalazł wszystkich optimów lokalnych w przestrzeni. Dla wszystkich instancji cliques, uniform_50 i uniform_80, att48, berlin52 oraz pr76 algorytm dwufazowy znalazł również o wiele większą ilość krawędzi. Co ważne, większość z nich została znaleziona w bardzo późnej fazie próbkowania.

4.3.4 Korelacja wzajemna miar

Aby sprawdzić, w jakim stopniu skorelowane są ze sobą poszczególne miary przestrzeni rozwiązań, obliczono współczynnik korelacji Pearsona dla każdej pary miar. Współczynniki uzyskane ze wszystkich instancji badanych w sekcjach 4.3.2 i 4.3.3 uśredniono i przedstawiono na wykresach. W przypadku, gdy wartość miary w danej instancji była stała, jako współczynnik korelacji przyjmowano wartość 0. Wykonano osobne wykresy dla algorytmu dwufazowego i *snowball*, aby ocenić wpływ metody na korelację miar. Wyniki dla algorytmu *snowball* przedstawiono w tabeli 4.4. Wyniki dla algorytmu dwufazowego

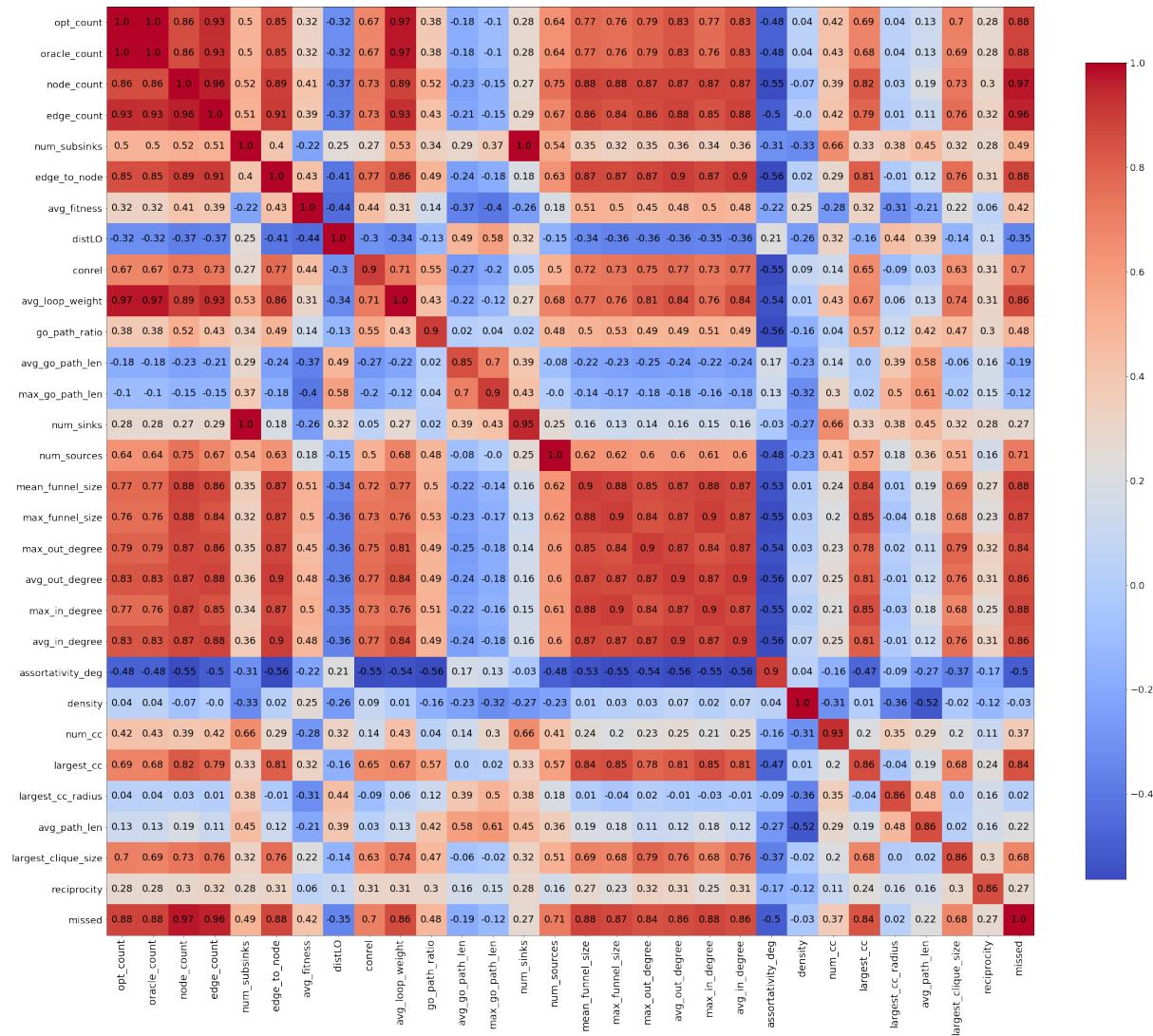
przedstawiono w tabeli 4.5.



Rysunek 4.4 Wzajemna korelacja miar — próbkowanie snowball

Występuje wysoka dodatnia korelacja występuje pomiędzy liczbą wywołań funkcji celu (opt_count), liczbą obliczeń długości ścieżki metodą przyrostową (oracle_count), liczbą wierzchołków (node_count) i krawędzi (edge_count). Wspomniane wartości silnie korelują dodatnio z liczbą źródeł i ścieków, a także z metrykami lejowymi oraz średnimi i maksymalnymi wartościami stopni wierzchołków. Korelacja ta występuje dla obu algorytmów, ale jest nieco silniejsza dla algorytmu dwufazowego. Występuje bardzo silna korelacja między liczbą wywołań funkcji celu i średnią wagą pętli w algorytmie dwufazowym. Nie jest to zaskakujące, gdyż — szczególnie dla instancji z dużą liczbą optimów lokalnych — algorytm ten ma tendencję do tworzenia pętli o bardzo dużych wagach.

Dla algorytmu *snowball* występuje korelacja o wartościach bliskich 1 pomiędzy liczbą źródeł, ścieków i lejów w przestrzeni. Dla algorytmu dwufazowego korelacja między liczbą źródeł i ścieków wynosi jedynie 0.25. Metryki lejowe oraz dotyczące stopni wierzchołków są ze sobą silniej skorelowane w przypadku próbkowania dwufazowego. Dla tego rodzaju próbkowania widoczna jest również dość silna ujemna korelacja między różnorodnością (assortativity_deg) a tymi metrykami. Różnorodność koreluje również ujemnie z metrykami conrel, go_path_ratio i avg_loop_weight. W przypadku próbkowania *snow-*



Rysunek 4.5 Wzajemna korelacja miar — próbkowanie dwufazowe

ball występuje silna ujemna korelacja między gęstością grafu (density) a liczbą ścieków, liczbą źródeł i maksymalnym stopniem wychodzącym. Metryka ta, dla obu rodzajów próbkowania, koreluje ujemnie z każdą inną metryką oprócz conrel.

Obliczono wartość średnią współczynników korelacji z innymi metrykami dla każdej metryki, aby sprawdzić, które z nich są w najmniejszym stopniu zależne od innych. Wyniki przedstawiono w tabeli 4.31. Najbliższymi 0 średnimi wartościami korelacji cechują się metryki: distLO, avg_go_path_len, max_go_path_len, largest_cc_radius. Dla algorytmu snowball bliska zera jest również wartość dla współczynnika conrel, a dla algorytmu dwufazowego — wartość dla miary density.

4.3.5 Podsumowanie wyników

W przypadku próbkowania małych instancji algorytm *snowball* zwykle buduje sieć LON bliższą rzeczywistej. Algorytm dwufazowy również radzi sobie z tym zadaniem, zwykle wymaga jednak znacznie dłuższych obliczeń. Instancje typu grid — z miastami ułożonymi na siatce — sprawiają największe trudności i już dla jedenastu miast algorytmy nie znalazły wszystkich wierzchołków i krawędzi. Dla większości instancji o rozmiarze 8 i większym

Tabela 4.31 Uśredniona wartość korelacji z innymi metrykami dla każdej metryki

Metryka	alg. snowball	alg. dwufazowy
node_count	0.52	0.53
edge_count	0.51	0.53
num_sub sinks	0.44	0.37
edge_to_node	0.42	0.51
avg_fitness	0.25	0.18
distLO	0.07	-0.07
conrel	-0.01	0.42
avg_loop_weight	0.40	0.51
go_path_ratio	0.32	0.32
avg_go_path_len	-0.03	0.01
max_go_path_len	0.01	0.06
num_sinks	0.44	0.27
num_sources	0.47	0.42
mean_funnel_size	0.30	0.50
max_funnel_size	0.49	0.49
max_out_degree	0.46	0.49
avg_out_degree	0.41	0.50
max_in_degree	0.49	0.49
avg_in_degree	0.41	0.50
assortativity_deg	0.21	-0.30
density	-0.19	-0.05
num_cc	0.15	0.27
largest_cc	0.52	0.48
largest_cc_radius	0.11	0.12
avg_path_len	0.07	0.21
largest_clique_size	0.42	0.44
reciprocity	0.32	0.24
missed	-	0.53

oba algorytmy poradziły sobie ze spróbkowaniem przestrzeni w czasie krótszym od przeglądu zupełnego. Metryki oparte o bezwzględne wartości wag krawędzi (`avg_loop_weight` i `distLO`) nie dają informacji o wartościach w pełnej przestrzeni rozwiązań.

Nie występuje liniowa zależność między liczbą wierzchołków w przestrzeni, a liczbą krawędzi (rys. 4.6, rys. 4.7). Większość miar przestrzeni rozwiązań zmienia się w sposób nieprzewidywalny (przykładem jest długość najdłuższej ścieżki do optimum globalnego rys. 4.8). Pojawiają się jednak i takie, których wartość stabilizuje się lub wzrasta w przewidywalny sposób. Przykładem jest liczba źródeł i ścieków w przestrzeni uzyskanej w próbkowaniu *snowball* (rys. 4.9). Charakter wzrostu tych zmiennych umożliwia wykorzystanie ekstrapolacji liniowej do przewidzenia ich liczby dla większej ilości spróbkowanych wierzchołków. Istnieje jednak ryzyko, że przewidywana wartość będzie nieprawdziwa — dla instancji o rozmiarze 20, oraz `cliques_50` zaobserwowano wypłaszczenie krzywej, a nawet szybki spadek wartości tej miary (rys. 4.10). Podobny spadek zaobserwowano w niektórych przypadkach próbkowania dwufazowego. Maksymalny stopień wierzchołków również można oszacować z pewną dokładnością, ale tylko dla instancji o rozmiarze mniejszym od 50. Im więcej optimów lokalnych w przestrzeni, tym rzadziej zmienia się ta wartość i tym szersze „schodki” widoczne są na wykresie (rys. 4.11).

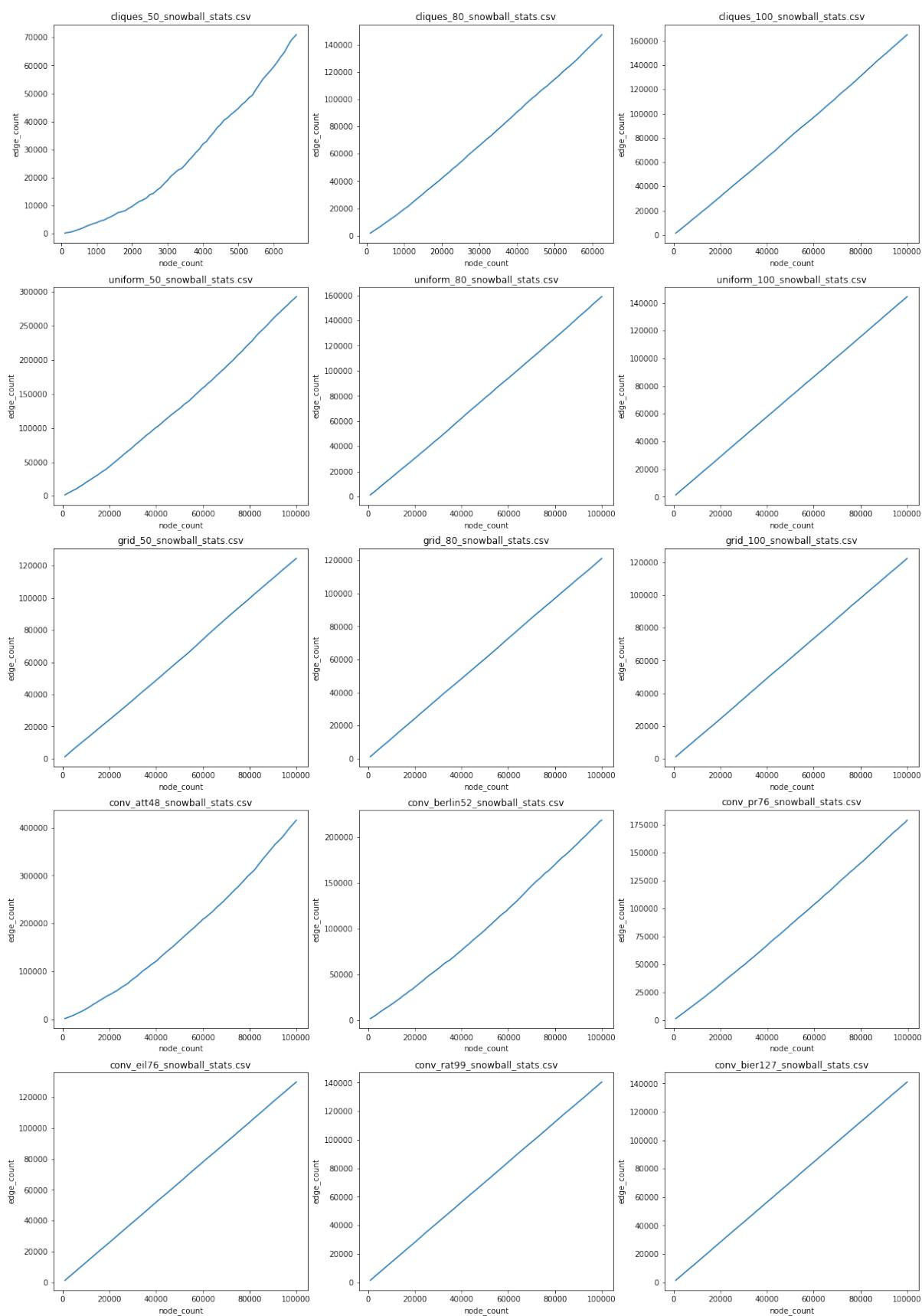
Rodzaj zastosowanego algorytmu ma duży wpływ na to, jak wartości miary zmieniają się w miarę próbkowania. Algorytm dwufazowy zaczyna tworzyć krawędzie dopiero po pewnym czasie, gdy w zbiorze wierzchołków znajduje się odpowiednia ich ilość. Z tego powodu na wykresie liczby krawędzi dla algorytmu dwufazowego widzimy krzywą eksponentyczną. Algorytm dwufazowy nie radzi sobie z największymi (pod względem liczby optimów lokalnych)instancjami - zdecydowaną większość krawędzi (lub w przypadku większych instancji grid — wszystkie krawędzie) stanowią w nich pętle, które nie mogą być wykorzystane do obliczenia wartości większości metryk. Dla tych instancji zbudowanie gęstszej sieci LON wymagałoby znacznego zwiększenia i tak już dużego wkładu obliczeniowego. Warto jednak zwrócić uwagę na to, że w niektórych instancjach (np. `cliques_50`) algorytm *snowball* znalazł o wiele mniej optimów lokalnych od algorytmu dwufazowego, pomimo długiego czasu próbkowania. Oznacza to, że algorytm dwufazowy może odnaleźć optima lokalne, których znalezienie sprawia trudności algorytmowi *snowball*. Najprawdopodobniej jest to spowodowane bardziej losowym charakterem algorytmu dwufazowego. Snowball kolejny etap przeszukiwania zaczyna od jednego z sąsiednich rozwiązań, algorytm dwufazowy za każdym razem zaczyna od rozwiązania całkowicie losowego.

Zaobserwowano różnice między różnymi rodzajami instancji. Instancja `cliques_50` z miastami ułożonymi w klikach ma o wiele mniej optimów lokalnych od instancji innego typu o podobnym rozmiarze. Instancje typu grid z miastami ułożonymi na siatce mają ich z kolei znacznie więcej. Instancje tego typu są również najtrudniejsze do próbkowania algorytmem dwufazowym. Instancje typu uniform często wykazują inne właściwości od innych instancji tego samego typu. Nie jest to zaskakujące — są to instancje losowe bez ustalonej struktury i wiele zależy od ułożenia miast w konkretnej instancji.

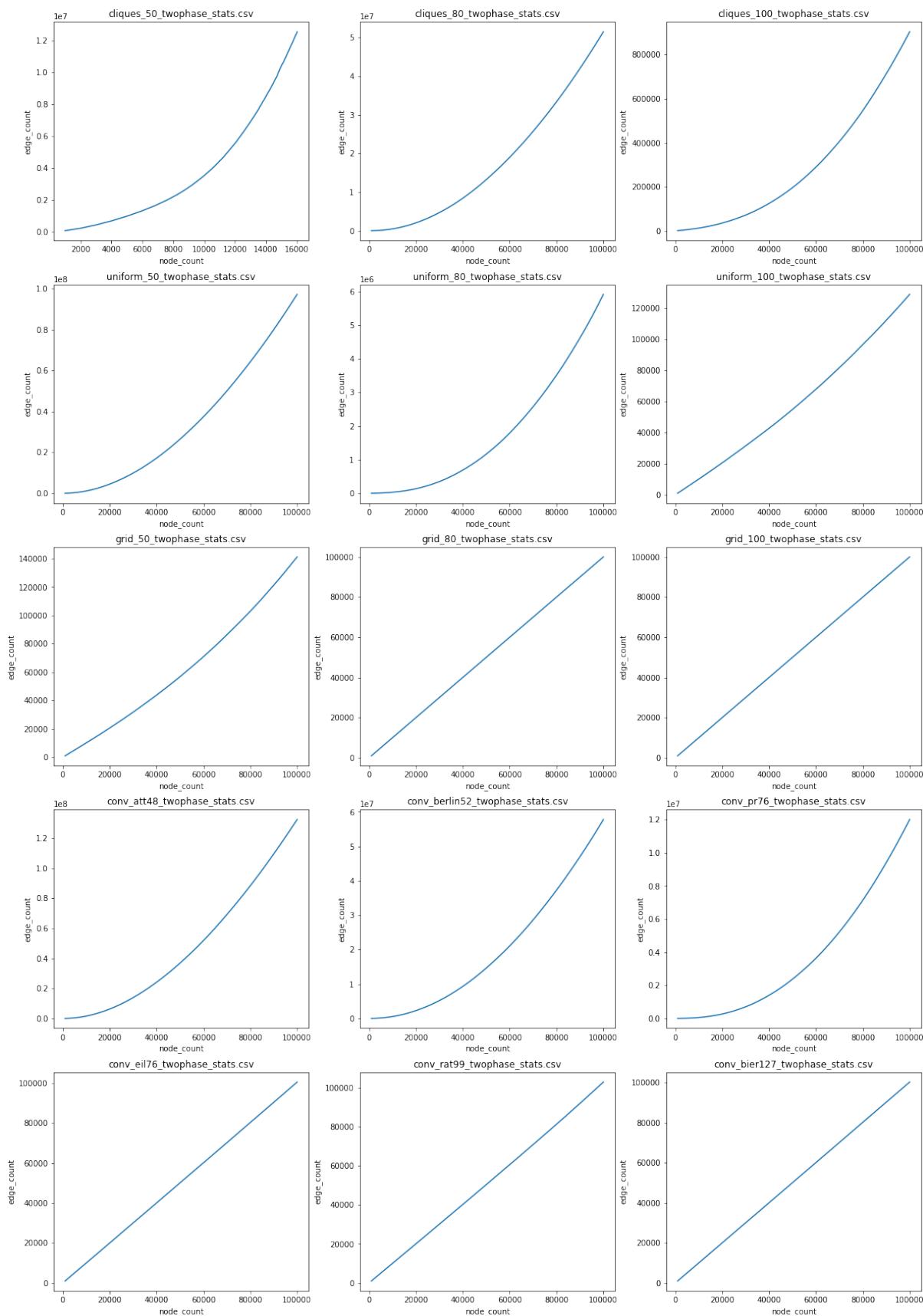
Instancje `cliques_50`, `cliques_80`, `uniform_50`, `att48`, `berlin52` i `pr76` cechują się wyjątkowo dużą liczbą krawędzi w stosunku do liczby wierzchołków (stosunek ten był największy dla instancji `att48` i wynosił ponad 1300 do 1). Krawędzie te nie zostały odnalezione przez algorytm *snowball*, a jedynie w późnym etapie próbkowania dwufazowego (4.12). Istnienie milionów krawędzi znacznie wydłużyło długość obliczeń, oraz rozmiar plików z reprezentacją grafu i była powodem rezygnacji z obliczenia wartości części miar w tym badaniu.

Wiele miar jest ze sobą silnie skorelowanych, większość dodatnio. Istnieje jednak kilka miar, których wartość jest praktycznie niezależna od innych. Wybór algorytmu próbkowa-

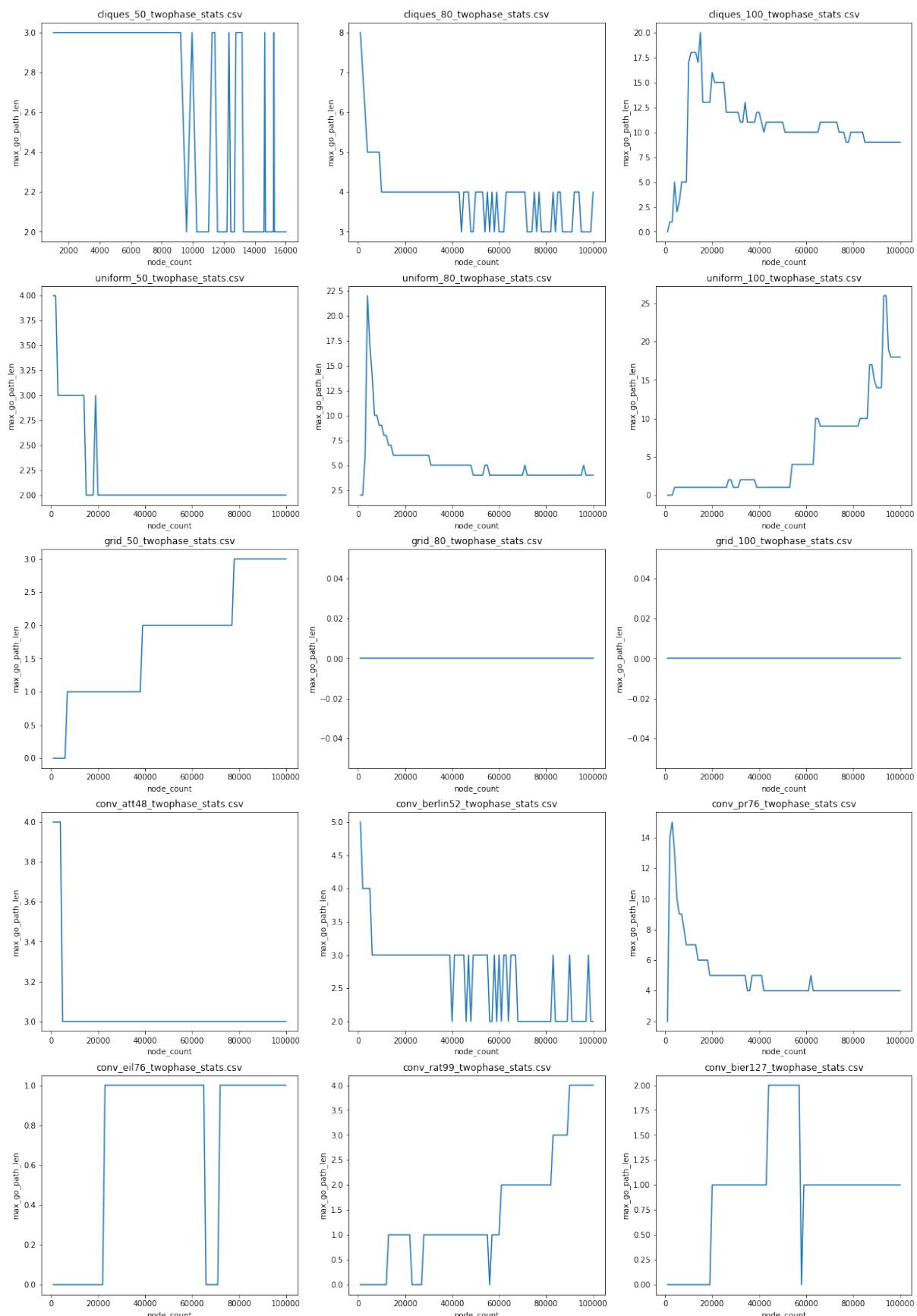
nia ma znaczący wpływ na wzajemną korelację pomiędzy wartościami niektórych miar.



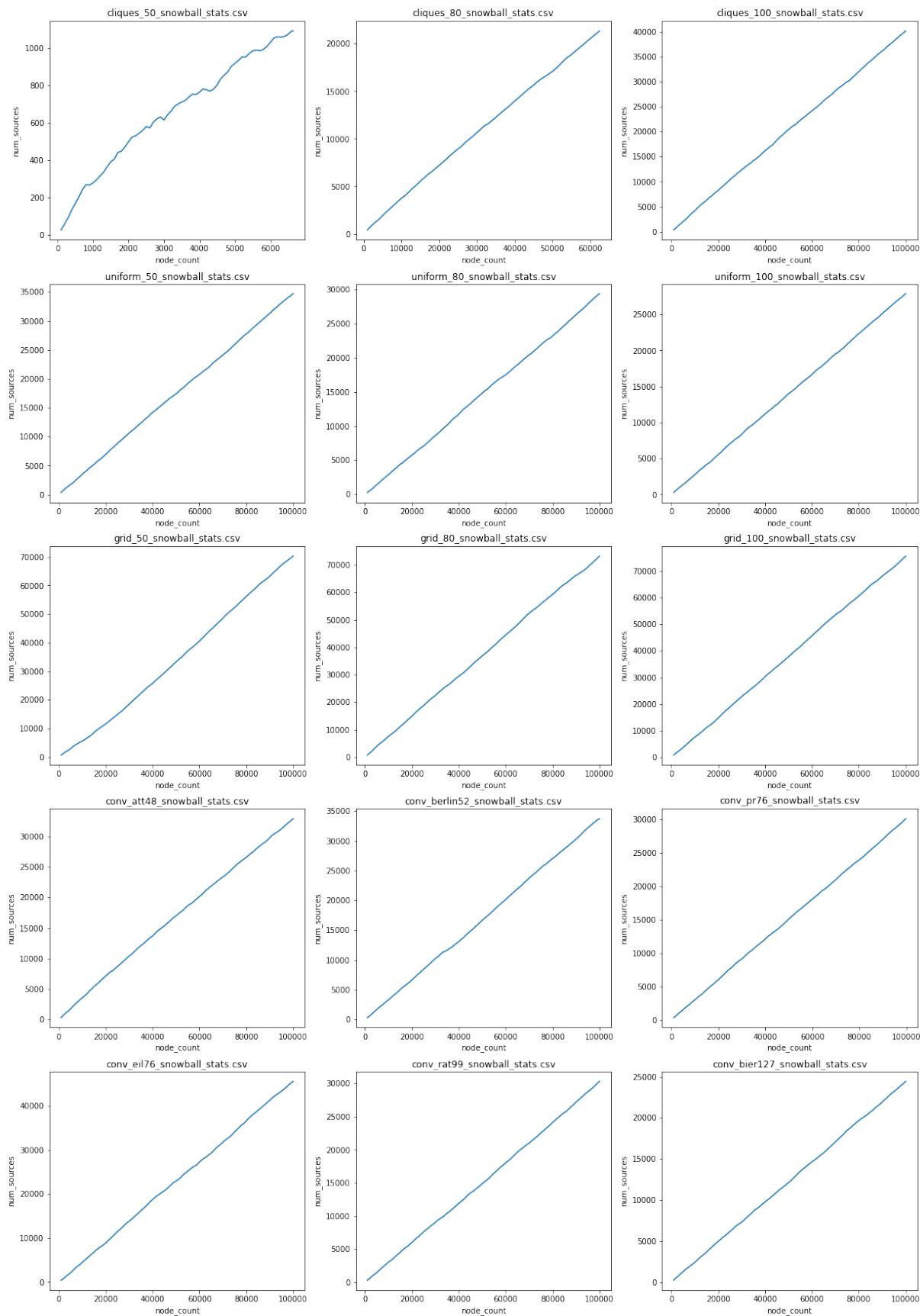
Rysunek 4.6 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) — próbkowanie snowball



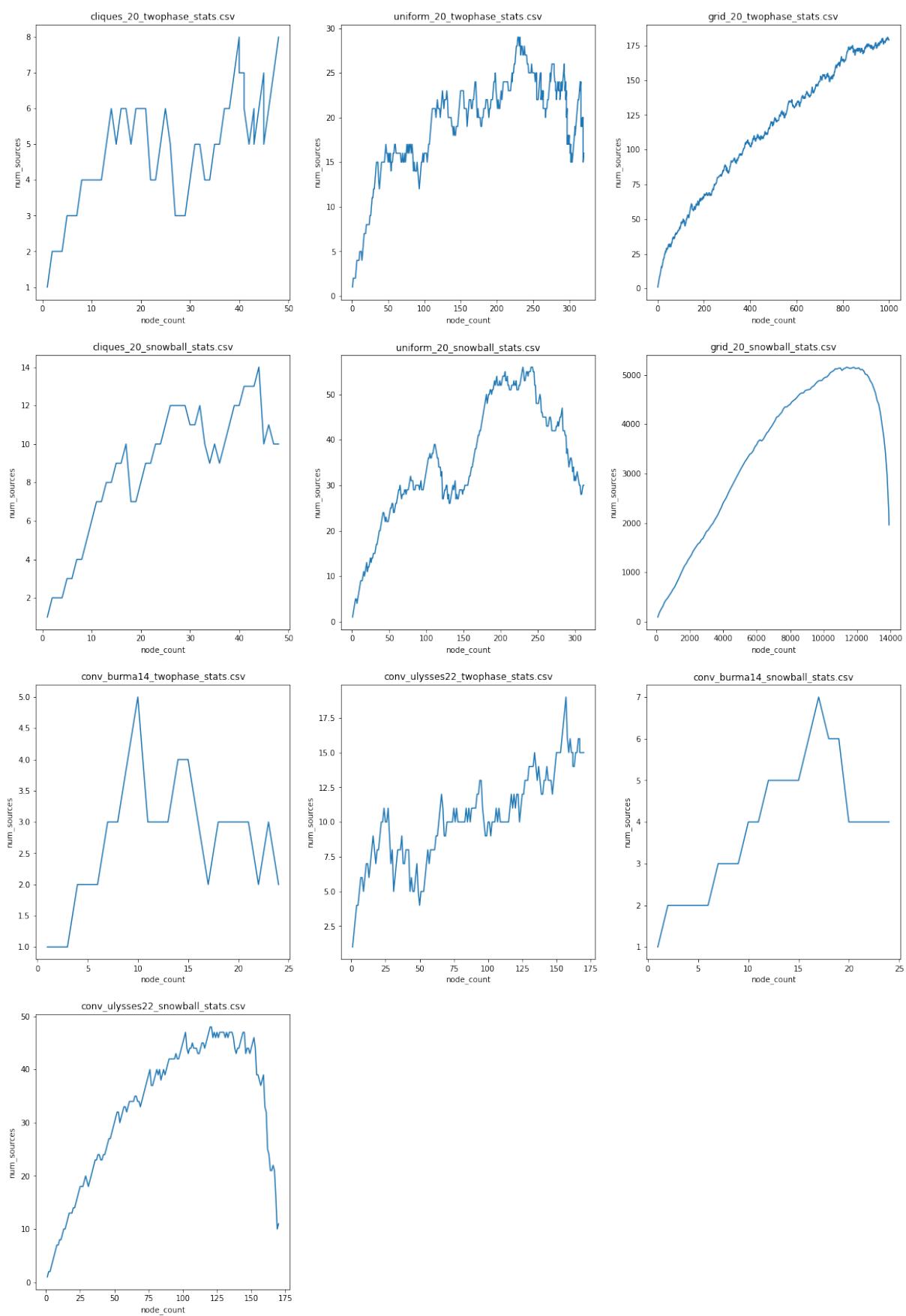
Rysunek 4.7 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) — próbkowanie dwufazowe



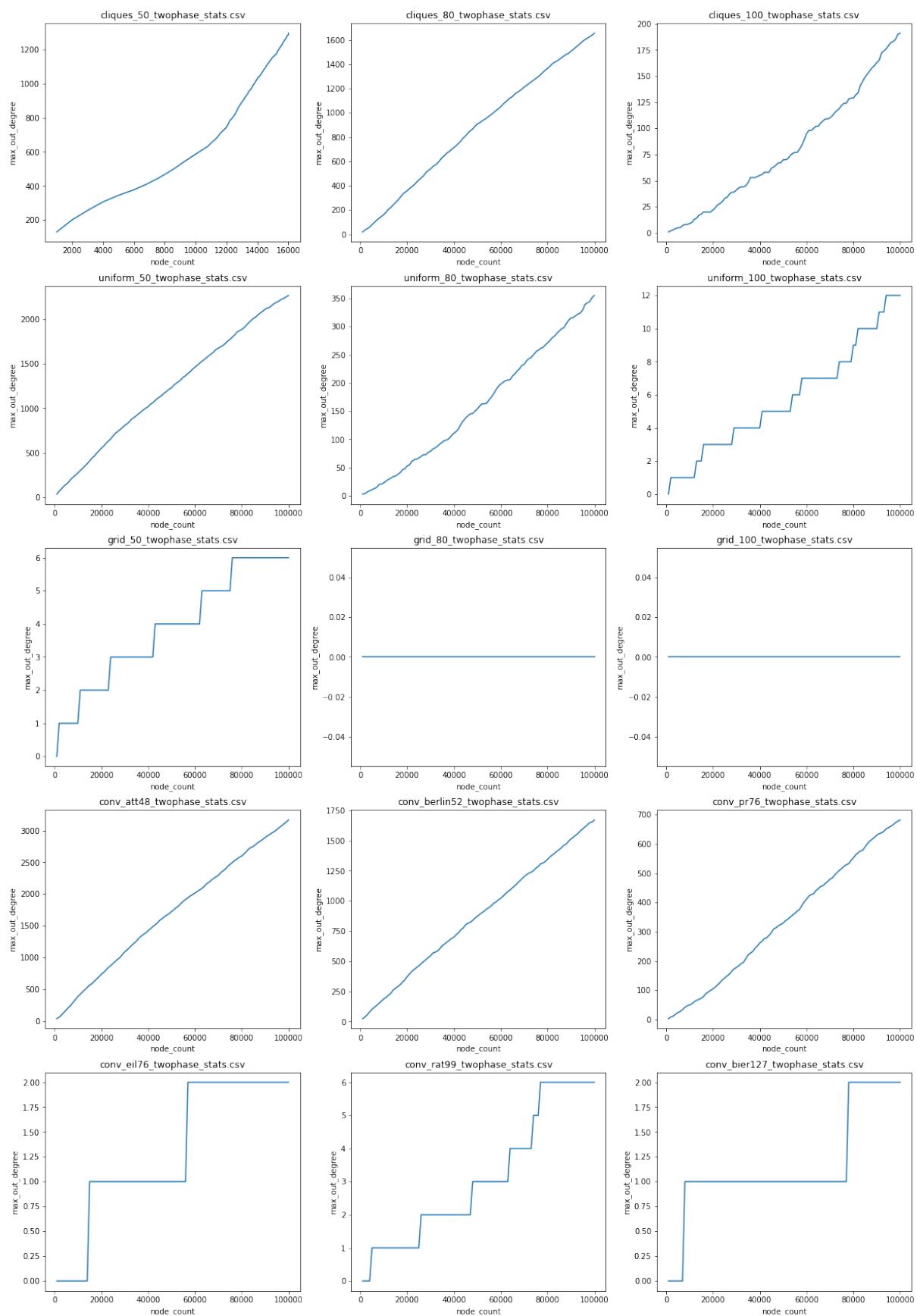
Rysunek 4.8 Wykres miary `max_go_path_len` dla próbkowania dwufazowego — przykład miary nieprzewidywalnej



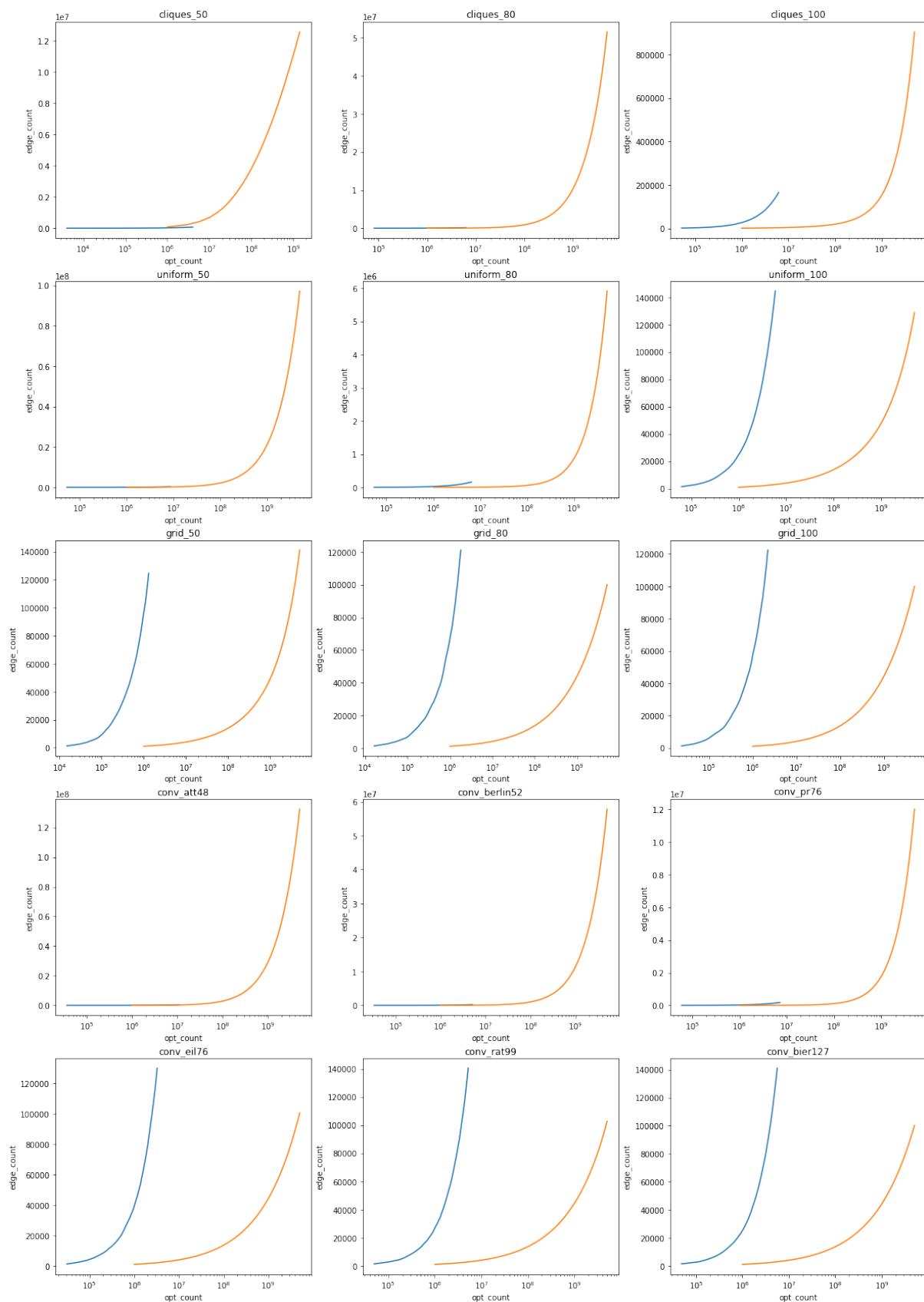
Rysunek 4.9 Wykres miary num_sources dla próbkowania snowball — przykład miary o wartościach zmieniających się w przewidywalny sposób



Rysunek 4.10 Wykres miary num_sources dla instancji średniej wielkości



Rysunek 4.11 Wykres miary max_out_degree dla próbkowania dwufazowego



Rysunek 4.12 Wykres zależności liczby krawędzi od liczby wywołań funkcji celu. Kolor niebieski - próbkowanie snowball, kolor pomarańczowy — próbkowanie dwufazowe. Oś X w skali logarytmicznej

Rozdział 5

Opis implementacji

W tym rozdziale opisane zostało oprogramowanie utworzone na potrzeby projektu. Zawiera on opis poszczególnych programów i skryptów, instrukcje uruchomienia, format obsługiwanych plików oraz spis wykorzystanych zewnętrznych bibliotek. Kod źródłowy oprogramowania jest publicznie dostępny na platformie Github pod adresem:
https://github.com/gero0/FLA_research.

Opracowano następujące narzędzia:

- Program wykonujący próbkowanie przestrzeni rozwiązań „tsp_samplers”,
- Program obliczający wartości miar na podstawie spróbkowanej przestrzeni „stat_calculator”,
- Generatory instancji testowych,
- Skrypty pomocnicze.

5.1 Program próbujący

Ponieważ próbkowanie przestrzeni rozwiązań wymaga dużego nakładu obliczeniowego, implementację algorytmów wykonano w języku Rust. Rust jest językiem kompilowanym do kodu maszynowego, wspierającym programowanie wielowątkowe.

5.1.1 Uruchamianie i parametry

Program ma postać pliku wykonywalnego uruchamianego z linii poleceń. Program wymaga podania nazwy pliku wejściowego, nazwę algorytmu oraz wartości parametrów tego algorytmu. Opcjonalnie jako parametr możliwe jest podanie nazwy folderu, do którego mają zostać zapisane wyniki próbkowania, jeśli nie zostanie ona podana program zapisze wyniki do folderu o nazwie „<algorytm>_latest”.

Format komendy wygląda więc następująco:

```
1  tsp_samplers <plik_wejsciowy> [plik_wyjsciowy] \
2  <algorytm> <parametry_algorytmu>
```

Akceptowane wartości parametru „algorytm” to:

- tp — próbkowanie dwufazowe
- snowball — próbkowanie snowball
- exhaustive — przegląd zupełny

Algorytm przeglądu zupełnego nie przyjmuje żadnych parametrów wejściowych.

Algorytm dwufazowy przyjmuje parametry w następującej kolejności:

1 <ITERS> <N_MAX> <N_ATT> <E_ATT> <MUT_D>

Gdzie:

- ITERS (n_{runs}) — liczba iteracji,
- N_MAX (n_{max}) — żądana liczba wierzchołków,
- N_ATT (n_{att}) — liczba prób generowania wierzchołków,
- E_ATT (e_{att}) — liczba prób generowania krawędzi,
- MUT_D (D) — parametr D

Algorytm został opisany szczegółowo w sekcji 4.1.1 i listingu 2.

Algorytm *snowball* przyjmuje parametry w następującej kolejności:

1 <WALK_LEN> <N_EDGES> <DEPTH> <MUT_D> <SAVE_THRESHOLD>

Gdzie:

- WALK_LEN (w_{len}) — długość losowego spaceru
- N_EDGES (m) — liczba prób przeszukania sąsiedztwa
- DEPTH> ($depth$) — głębokość przeszukiwania
- MUT_D (D) — parametr D
- SAVE_THRESHOLD ($s_{_}thresh$) — interwał zapisu

Algorytm został opisany szczegółowo w sekcji 4.1.2 i listingu 3.

5.1.2 Pliki wejściowe i wyjściowe

Plik wejściowy musi być plikiem txt w następującym formacie:

- Linia 1 — nazwa instancji
- Linia 2 — liczba miast w instancji
- N kolejnych linii — Pełna macierz odległości pomiędzy miastami podanych w postaci liczb całkowitych. Kolejne wartości w rzędzie macierzy rozdzielone spacją, wiersze rozdzielone znakiem nowej linii

Przykładowy plik wejściowy przedstawiającyinstancję burma14 ze zbioru TSPLIB został przedstawiony na listingu 5.1.

Listing 5.1 Instancja burma14 w formacie akceptowanym przez program

```
1 burma14
2 14
3 0 153 510 706 966 581 455 70 160 372 157 567 342 398
4 153 0 422 664 997 598 507 197 311 479 310 581 417 376
5 510 422 0 289 744 390 437 491 645 880 618 374 455 211
6 706 664 289 0 491 265 410 664 804 1070 768 259 499 310
7 966 997 744 491 0 400 514 902 990 1261 947 418 635 636
```

```

8      581 598 390 265 400 0 168 522 634 910 593 19 284 239
9      455 507 437 410 514 168 0 389 482 757 439 163 124 232
10     70 197 491 664 902 522 389 0 154 406 133 508 273 355
11     160 311 645 804 990 634 482 154 0 276 43 623 358 498
12     372 479 880 1070 1261 910 757 406 276 0 318 898 633 761
13     157 310 618 768 947 593 439 133 43 318 0 582 315 464
14     567 581 374 259 418 19 163 508 623 898 582 0 275 221
15     342 417 455 499 635 284 124 273 358 633 315 275 0 247
16     398 376 211 310 636 239 232 355 498 761 464 221 247 0

```

Wyniki programu zapisuje okresowo do docelowego folderu w plikach „samples_<i>.json”. Pliki zawierają następujące informacje:

- nodes — lista wierzchołków grafu. Każdy wierzchołek opisywany jest 3-elementową listą: [unikalny identyfikator, rozwiązanie, wartość funkcji celu]. Identyfikator oraz wartość funkcji celu są liczbami całkowitymi, rozwiązanie ma zaś postać listy identyfikatorów miast w kolejności, w jakiej przebiega przez nie trasa,
- edges — lista krawędzi skierowanych. Każda krawędź jest opisane 3-elementową listą — [id. wierzchołka początkowego, id. wierzchołka docelowego, waga],
- opt_count — liczba wywołań funkcji optymalizującej (2opt),
- oracle_count — liczba wykonanych obliczeń długości ścieżki,
- time_ms — czas działania programu podany w ms. Liczony jest jedynie czas spędzony na próbkowaniu (bez liczenia czasu spędzonego na zapisywanie wyników),
- comment — komentarz, zawiera parametry, z którymi uruchomiono algorytm,
- missed — tylko dla próbkowania dwufazowego, licznik prób dodania krawędzi, które zakończyły się niepowodzeniem z powodu braku wierzchołka należącego do krawędzi w zbiorze spróbkowanych wierzchołków.

Przykładowy plik wyjściowy programu został przedstawiony na listingu 5.2.

Listing 5.2 Przykład pliku wyjściowego po krótkim próbkowaniu małej instancji

```

1  {
2  "opt_count":101802,
3  "oracle_count":5114163,
4  "time_ms":199,
5  "comment":"n_max:100 n_att:100 e_att:100 iters:10 file:cliques_6",
6  "missed":0,
7  "nodes": [
8  [1,[0, 1, 4, 5, 3, 2],1622],
9  [0,[0, 2, 3, 5, 4, 1],1622]],
10 "edges": [
11 [0,0,335],
12 [1,0,93],
13 [0,1,665],
14 [1,1,907]]
15 }

```

5.2 Program obliczający wartości miar

Program obliczający wartości miar został napisany w języku Python. Większość miar grafu obliczone zostało przy pomocy biblioteki igraph. Aby zwiększyć wydajność, do obliczeń metryk **avg_loop_weight** i **num_sub sinks** wykorzystano moduł natywny napisany w języku Rust.

5.2.1 Uruchamianie i parametry

Program jest uruchamiany przy użyciu interpretera języka Python, jako argument przyjmuje ścieżkę do folderu zawierającego pliki z próbami. Opcjonalnymi parametrami są: nazwa pliku wyjściowego limit plików wejściowych (-l), oraz nazwy miar do policzenia (-s, -stats). Jeśli nazwa pliku wyjściowego nie zostanie podana, wyniki zapisywane są do pliku „results.csv”. Jeśli limit nie zostanie podany, obliczenia są wykonywane dla wszystkich plików w folderze. Jeśli nie podane zostaną żadne nazwy miar do policzenia, domyślnym działaniem jest obliczenie ich wszystkich.

Format komendy wygląda następująco:

```
1 python stat_calculator/main.py <folder_wejsciowy> \
2 [plik_wyjsciowy] [-l <N>] [—stats ...]
```

5.2.2 Pliki wejściowe i wyjściowe

Program na wejście przyjmuje ścieżkę do folderu. W wybranym folderze wyszukiwane są pliki JSON wygenerowane przez program próbujący. Pliki są sortowane na podstawie numeracji (natsort) i dla każdego wykonywany jest proces wczytania danych, utworzenia grafu LON i obliczenia wartości miar. Wszystkie pliki w folderze są traktowane jako pliki z próbami, dlatego nie można umieszczać w nim innych plików.

Dane wyjściowe zapisywane są w postaci pliku csv zgodnego ze strukturami DataFrame biblioteki pandas. Dodatkowo generowany jest plik „results_corr.csv” zawierający tablicę z wartościami współczynników korelacji między wartościami miar. Pliki jako separatora używają znaku średnika ;. Przykładowy plik wyjściowy został przedstawiony na listingu 5.3.

Listing 5.3 Przykład pliku wyjściowego z obliczonymi miarami num_cc i largest_cc

```
1 ;time_ms;opt_count;oracle_count;node_count;edge_count;num_cc;largest_cc
2 0;0;1;140;1;0;1;1
3 1;0;15;1204;2;1;2;1
4 2;1520;1010200;85157744;2;4;1;2
```

5.3 Generatory instancji testowych

Generatory instancji testowych napisano w języku Python. Są to trzy proste skrypty, każdy generujący inny rodzaj instancji problemu. Programy generują pliki tekstowe opisujące pozycje miast, oraz plik graficzny zawierający wizualizację instancji.

5.3.1 Uruchamianie i parametry

Skrypt uniform.py generuje instancję o miastach ułożonych równomiernie. Uruchomienie skryptu wygląda następująco:

```
1     python uniform.py <N> <area_size> [-o OUTPUT]
```

Parametr N określa liczbę miast w instancji. Parametr area_size określa rozmiar boku kwadratowej planszy, na której zostaną umieszczone miasta. Opcjonalny parametr output pozwala na ustawienie nazwy folderu wyjściowego (domyślnie uniform_output).

Skrypt grid.py generujeinstancję o miastach ułożonych na siatce. Uruchomienie skryptu wygląda następująco:

```
1     python grid.py <N> <gap> [-o OUTPUT]
```

Parametr N określa liczbę miast w instancji. Parametr gap określa odległość pomiędzy sąsiadującymi miastami w siatce. Opcjonalny parametr output pozwala na ustawienie nazwy folderu wyjściowego (domyślnie grid_output).

Skrypt cliques.py generujeinstancję o miastach ułożonych w klikach. Uruchomienie skryptu wygląda następująco:

```
1     python cliques.py <N> <N_cliques> [-minld MINLD] [-maxld MAXLD] \
2     [-mincd MINCD] [-maxcd MAXCD] [-o OUTPUT]
```

Parametr N określa liczbę miast w instancji. Parametr N_cliques określa liczbę klik. W odróżnieniu od pozostałych generatorów generator cliques.py może przyjąć dużą liczbę parametrów opcjonalnych:

- minld — minimalna odległość między miastami w tym samym klice (domyślnie 5)
- maxld — maksymalna odległość między miastami w tym samym klice (domyślnie 30)
- mincd — minimalna odległość między środkami klik (domyślnie 100)
- maxcd — maksymalna odległość między środkami klik (domyślnie 300)
- output — nazwa folderu wyjściowego (domyślnie cliques_output)

Możliwe jest że programowi nie uda się wygenerować instancji o zadanych parametrach. W takim przypadku w konsoli wyświetlany jest stosowny komunikat.

5.3.2 Pliki wyjściowe

Generatory zapisują trzy pliki do foldery wyjściowej:

- points.txt — zawiera listę miast w formacie: identyfikator, koordynat x, koordynat y
- matrix.txt — macierz odległości w formacie obsługiwany przez program próbujący
- vis.png — wizualizacja instancji

5.4 Skrypty pomocnicze

Skrypty pomocnicze wykonano w języku Python. Należą do nich:

- plotting.py — Skrypt rysujący wykresy poszczególnych miar w zależności od liczby wierzchołków. Skrypt przyjmuje na wejście pliki z wartościami miar wygenerowane przez program stat_calculator

- convert_tsplib.py — Skrypt konwertujący pliki w formacie tsplib na uproszczony format przyjmowany przez program próbujący. Korzysta z biblioteki tsplib95.
- visualize.py — Prosty skrypt tworzący wizualizację grafu LON na podstawie pliku wejściowego json z próbami.
- run_exhaustive.py — Skrypt uruchamiający po kolei przegląd zupełny dla podanej listy instancji
- run_tp.py — Skrypt uruchamiający po kolei próbkowanie dwufazowe dla podanej listy instancji
- run_snowball.py — Skrypt uruchamiający po kolei próbkowanie *snowball* dla podanej listy instancji
- run_stat_calc.py — Skrypt uruchamiający program obliczający wartości miar dla podanej listy folderów w próbkami
- run_plotting.py — Skrypt uruchamiający po kolei skrypt plotting.py dla podanej listy plików z obliczonymi wartościami miar

5.5 Wykorzystane biblioteki

Najważniejsze zewnętrzne biblioteki wykorzystane w projekcie:

Rust:

- rand_chacha — szybki i deterministyczny generator liczb losowych,
- clap — parser argumentów konsoli,
- rustc-hash — szybka hashmapa,
- permutations_iter — generator permutacji metodą Steinhause-Johnsona-Trottera,
- PyO3 — narzędzie do tworzenia natywnych modułów Pythona w języku Rust,
- tsptools — generator losowego rozwiązania problemu TSP, implementacja 2opt oraz funkcja obliczająca długość ścieżki.

Python:

- igraph — obliczanie wartości miar grafu LON, generowanie wizualizacji,
- matplotlib — rysowanie wykresów,
- numpy, pandas — przetwarzanie danych,
- Pillow — generowanie obrazów,
- tsplib95 — odczytywanie plików w formacie tsplib.

Rozdział 6

Podsumowanie

W pracy wykonano badanie stabilności wartości miar dla problemu komiwojażera. Wykonano implementację generatorów instancji testowych, algorytmów próbkowania przestrzeni rozwiązań, oraz programu obliczającego wartości miar. Zbadano 24 różne miary dla 35 instancji problemu pochodzących z generatorów oraz ze zbioru tsplib. Wykorzystano algorytm dwufazowy oraz *snowball*.

Dla małych instancji o rozmiarze do 11 miast, wykonano przegląd zupełny i porównano wartości miar z estymacjami uzyskanymi z próbkowania. Wykazano, że dla większości małych instancji możliwe jest uzyskanie dokładnych estymacji większości miar w czasie znacznie krótszym od czasu potrzebnego na wykonanie algorytmu przeglądu zupełnego. Wyniki sugerują również, że dla instancji tego rozmiaru bardziej efektywne obliczeniowo jest wykorzystanie algorytmu *snowball*.

Większe instancje poddano badaniu stabilności, polegającemu na okresowym zapisywaniu estymowanych wartości miar podczas próbkowania. Uzyskano w ten sposób wartości szacowanych wielkości w odniesieniu do liczby optimów lokalnych w spróbkowanej przestrzeni. Stwierdzono, że wartość większości miar jest niemożliwa do przewidzenia z dużą dokładnością. Zauważono duży wpływ długości próbkowania oraz wybranego algorytmu na estymowane wartości miar przestrzeni rozwiązań. Zidentyfikowano słabe punkty obu algorytmów próbkowania. Algorytm dwufazowy nie radzi sobie z odnajdywaniem krawędzi w instancjach z dużą liczbą optimów lokalnych przy niedostatecznie długim próbkowaniu. Algorytm *snowball* może mieć trudności ze znalezieniem wszystkich optimów lokalnych w przestrzeni.

Zauważono różnice między różnymi typami instancji. Instancje z miastami ułożonymi na siatce cechują się dużym stosunkiem liczby optimów lokalnych do liczby miast. Instancje z miastami rozmiieszczonymi w klikach mają zwykle mniej optimów lokalnych od podobnych rozmiarem instancji innego typu.

Przeprowadzenie badań wymagało wykonania ponad 20 miliardów wywołań procedury 2opt oraz funkcji obliczającej długość ścieżki. Próbkowanie oraz obliczanie wartości miar wymagało kilkudniowych obliczeń na wydajnej, 64-procesorowej maszynie. Obliczanie wartości miar dla największych sieci LON wymagało ponad 50GB pamięci RAM.

Z próbkowania uzyskano setki plików JSON reprezentujących otrzymane sieci LON, których przechowywanie wymagało wykorzystania ponad 300GB przestrzeni dyskowej.

Ze względu na ograniczony czas i moc obliczeniową, eksperymenty zostały przeprowadzone dla niewielkiej liczby instancji. Rozwój badań w tym kierunku powinien obejmować przeprowadzenie analizy dla dużej liczby instancji tego samego typu i rozmiaru, np. kilkanaście instancji o rozmiarze 100 i rozkładzie równomiernym. Wtedy będzie można stwierdzić z większą pewnością, na które cechy przestrzeni wpływa dany sposób rozmieszczenia miar.

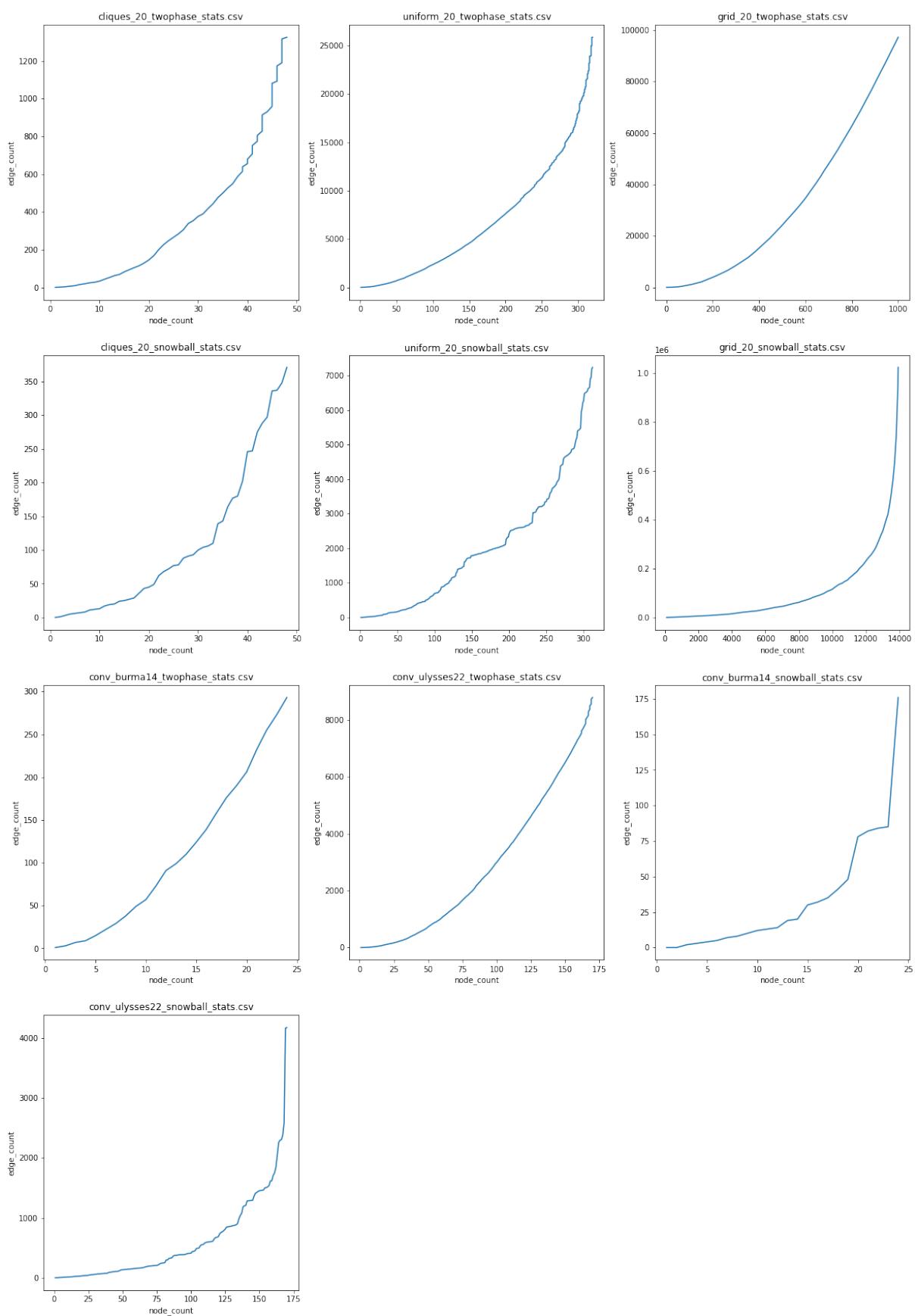
czenia miast. Należy również przeprowadzić dłuższe próbkowanie, szczególnie dla instancji z dużą ilością lokalnych optimów.

Aby dłuższe próbkowanie było możliwe, należy zadbać o wydajną implementację algorytmów. Oprócz wymyślania nowych algorytmów, można również dokonywać prób zrównoleglenia istniejących. Na potrzeby tej pracy zaimplementowano algorytm dwufazowy w sposób umożliwiający wykorzystanie wielu procesorów. Kolejnym krokiem może być uruchomienie algorytmu w systemie rozproszonym i/lub na kartach GPU.

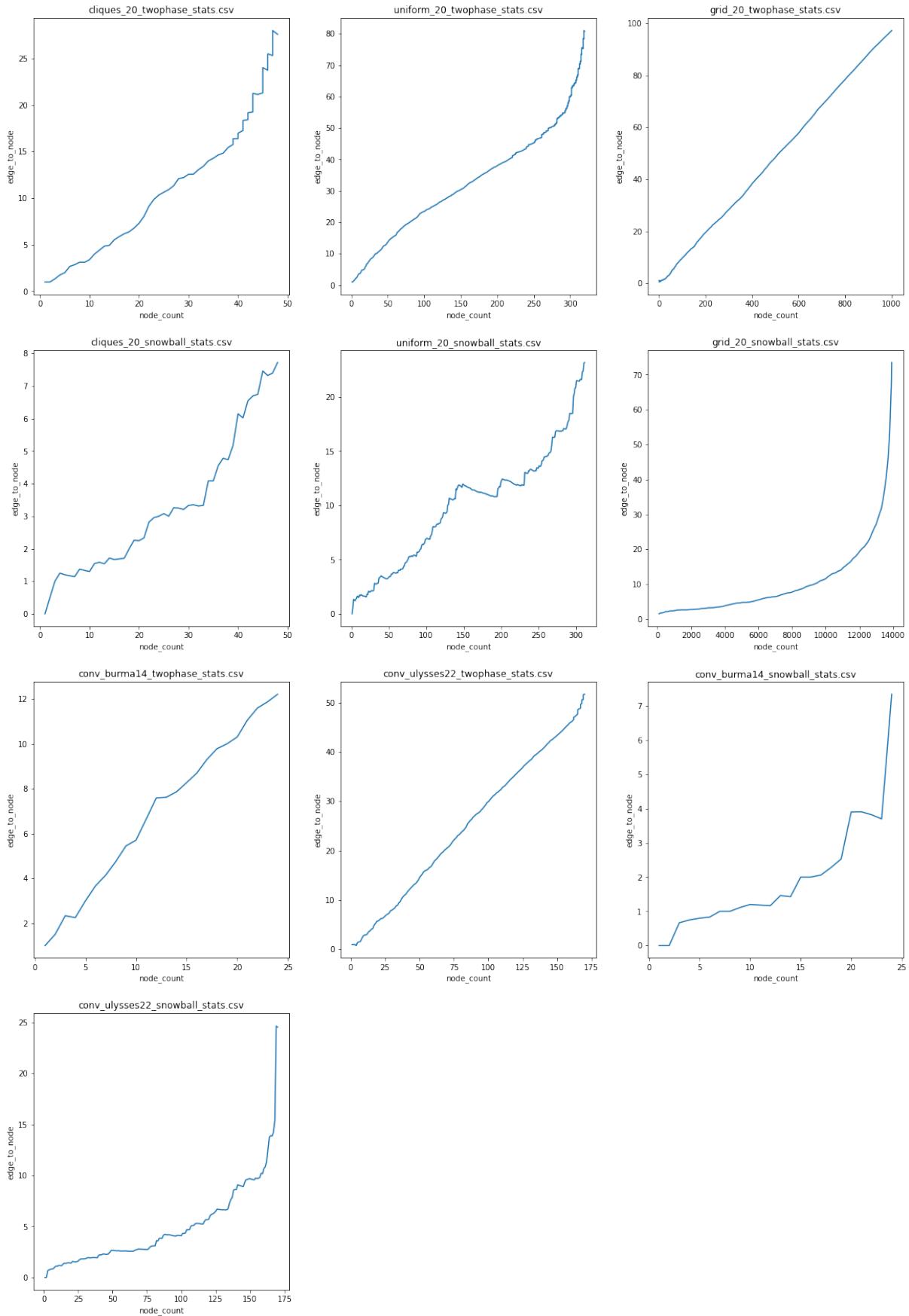
Dodatek A

Wykresy

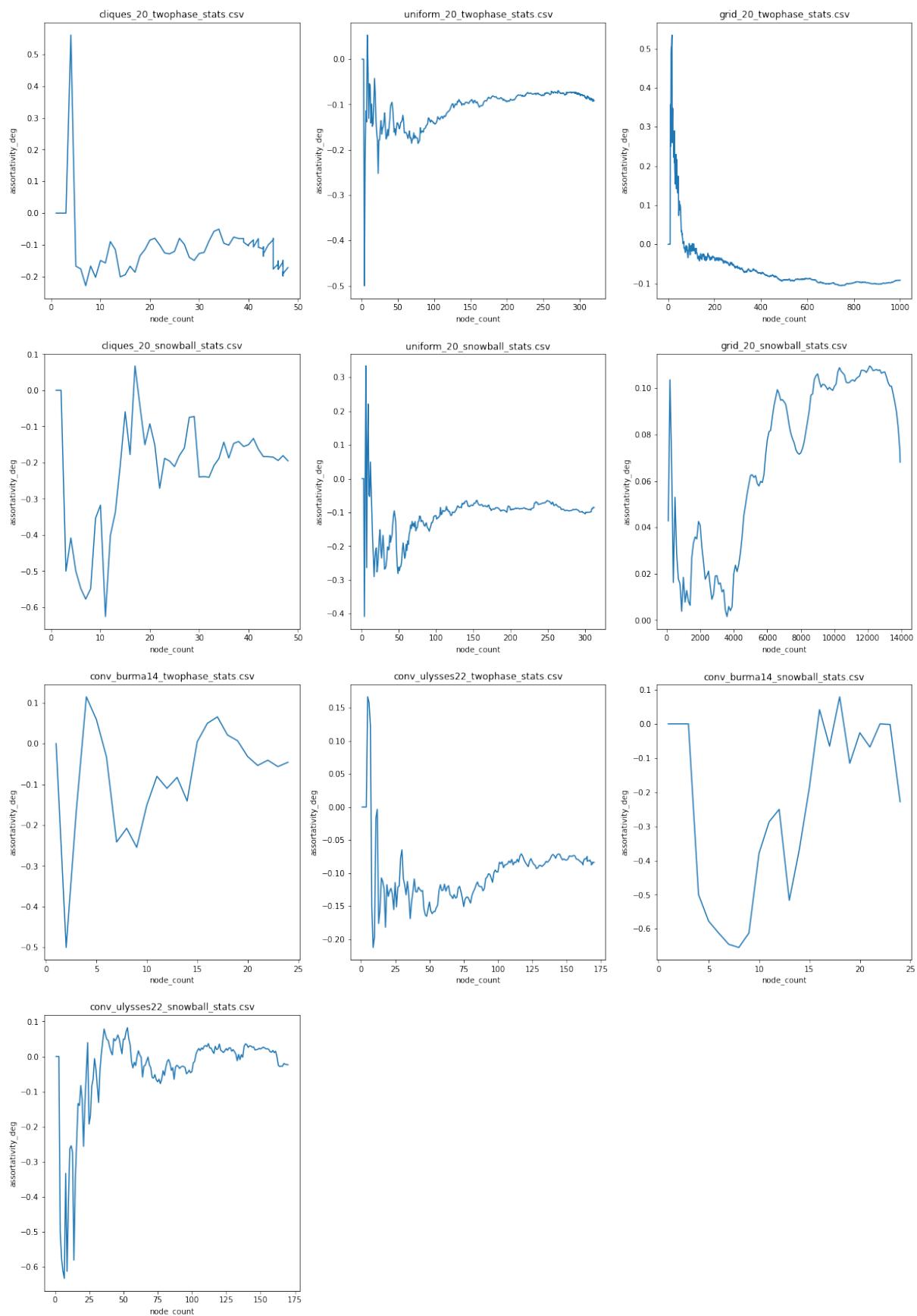
Dodatek zawiera wykresy wartości wszystkich miar poddanych badaniu stabilności w sekcjach 4.3.2 i 4.3.3.



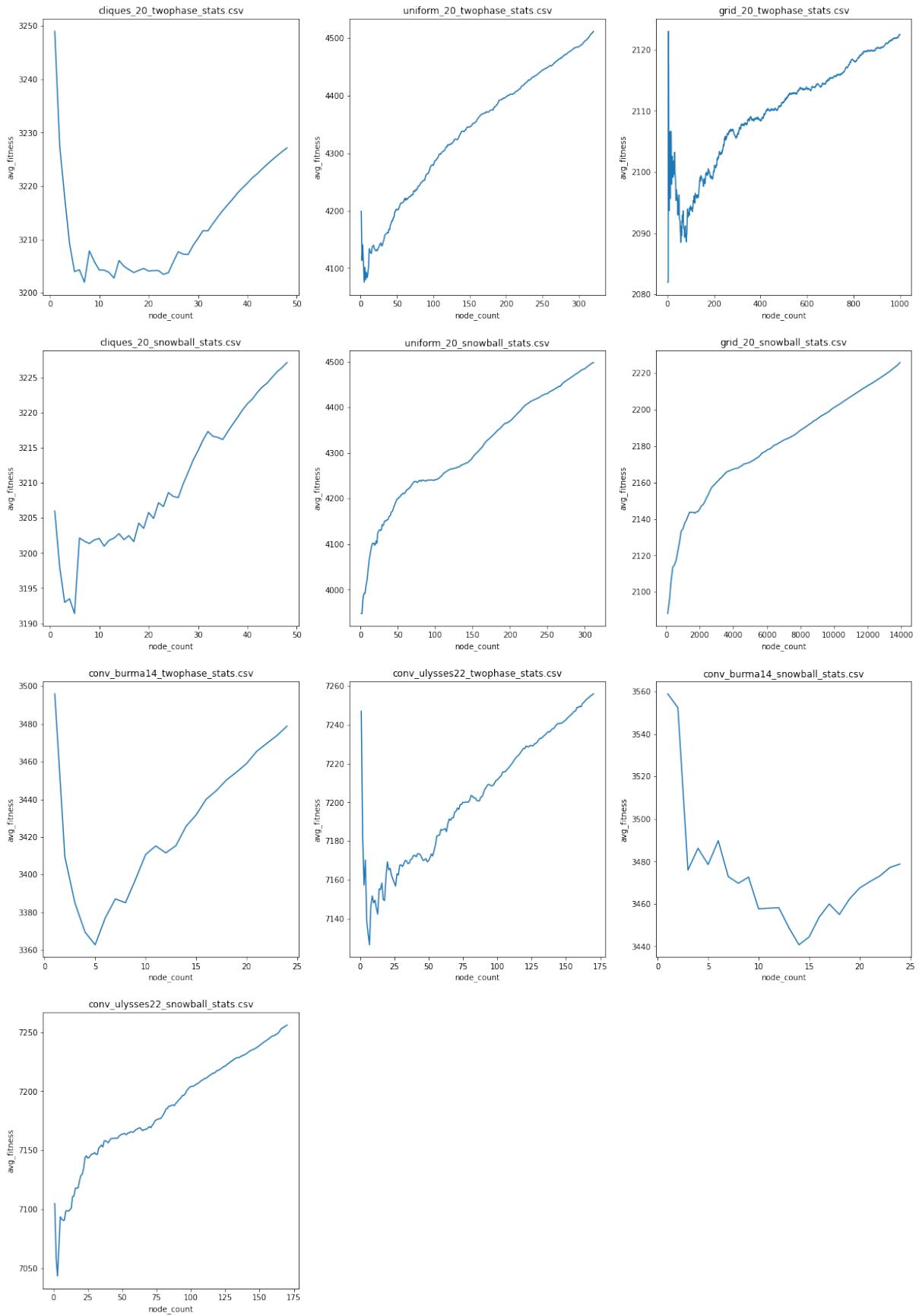
Rysunek A.1 Liczba krawędzi (edge_count) w zależności od liczby wierzchołków



Rysunek A.2 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) w zależności od liczby wierzchołków



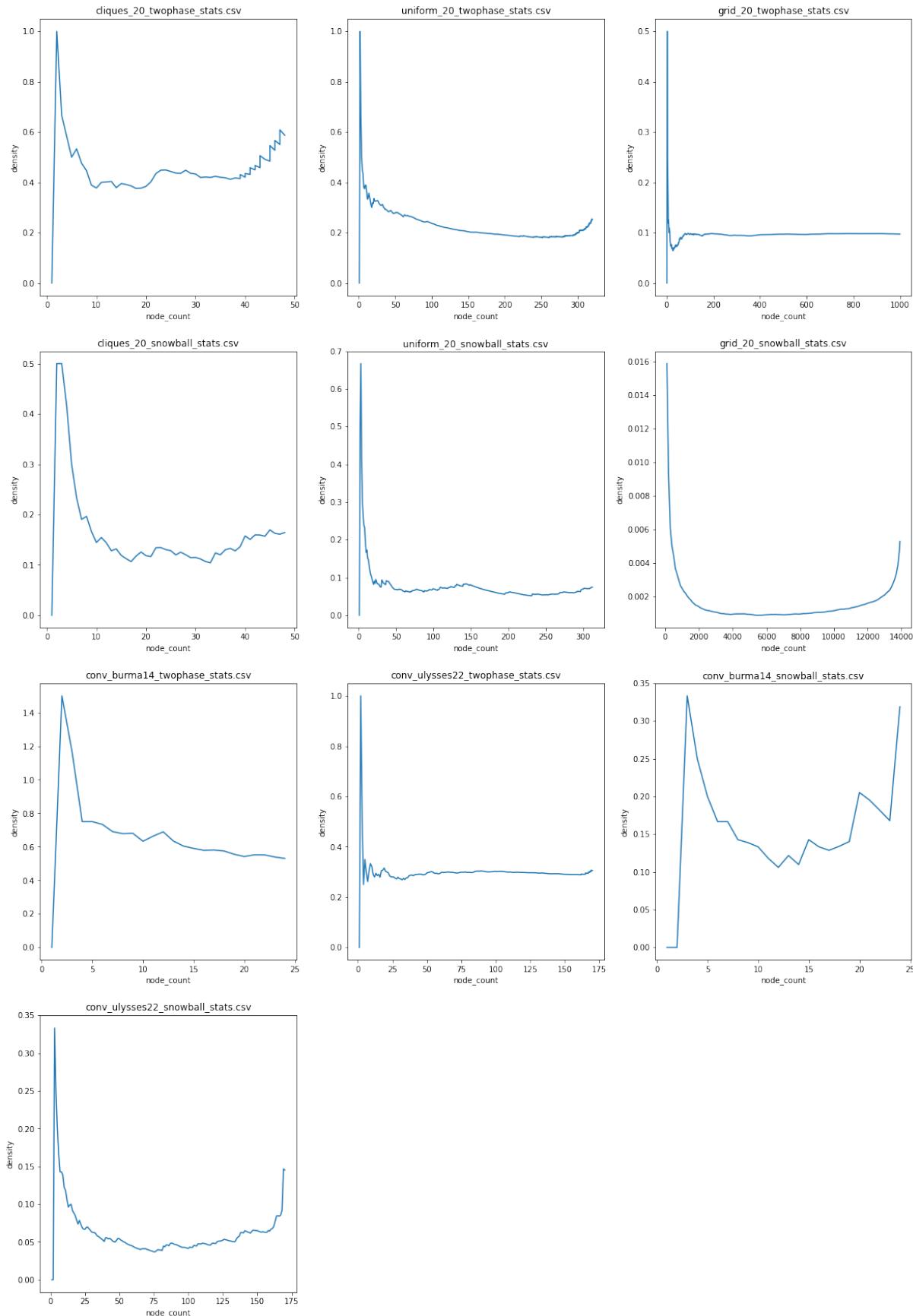
Rysunek A.3 Współczynnik różnorodności grafu (`assortativity_deg`) w zależności od liczby wierzchołków



Rysunek A.4 Średnia wartość funkcji celu w znalezionych optimach lokalnych (avg_fitness) w zależności od liczby wierzchołków



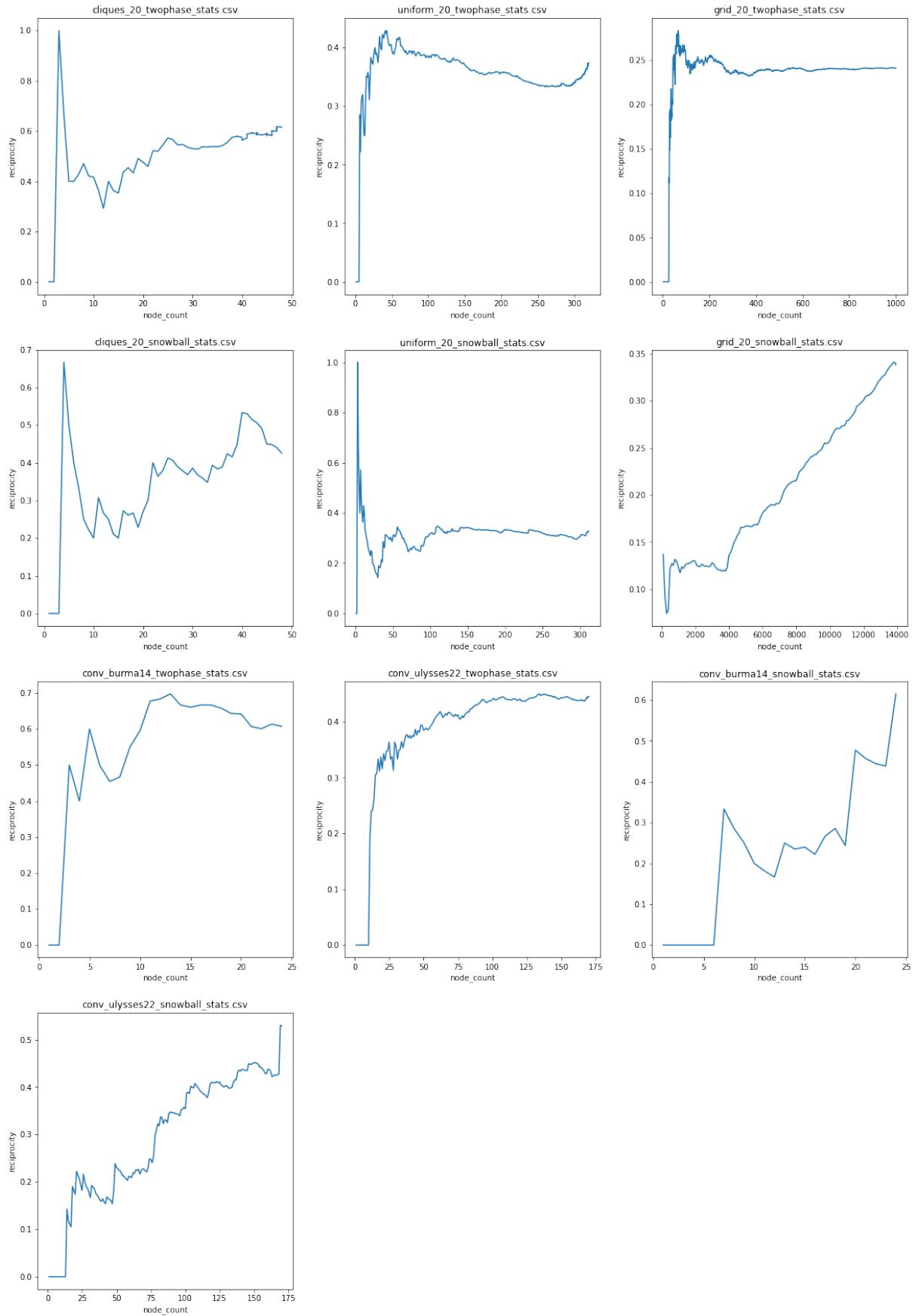
Rysunek A.5 Współczynnik conrel w zależności od liczby wierzchołków



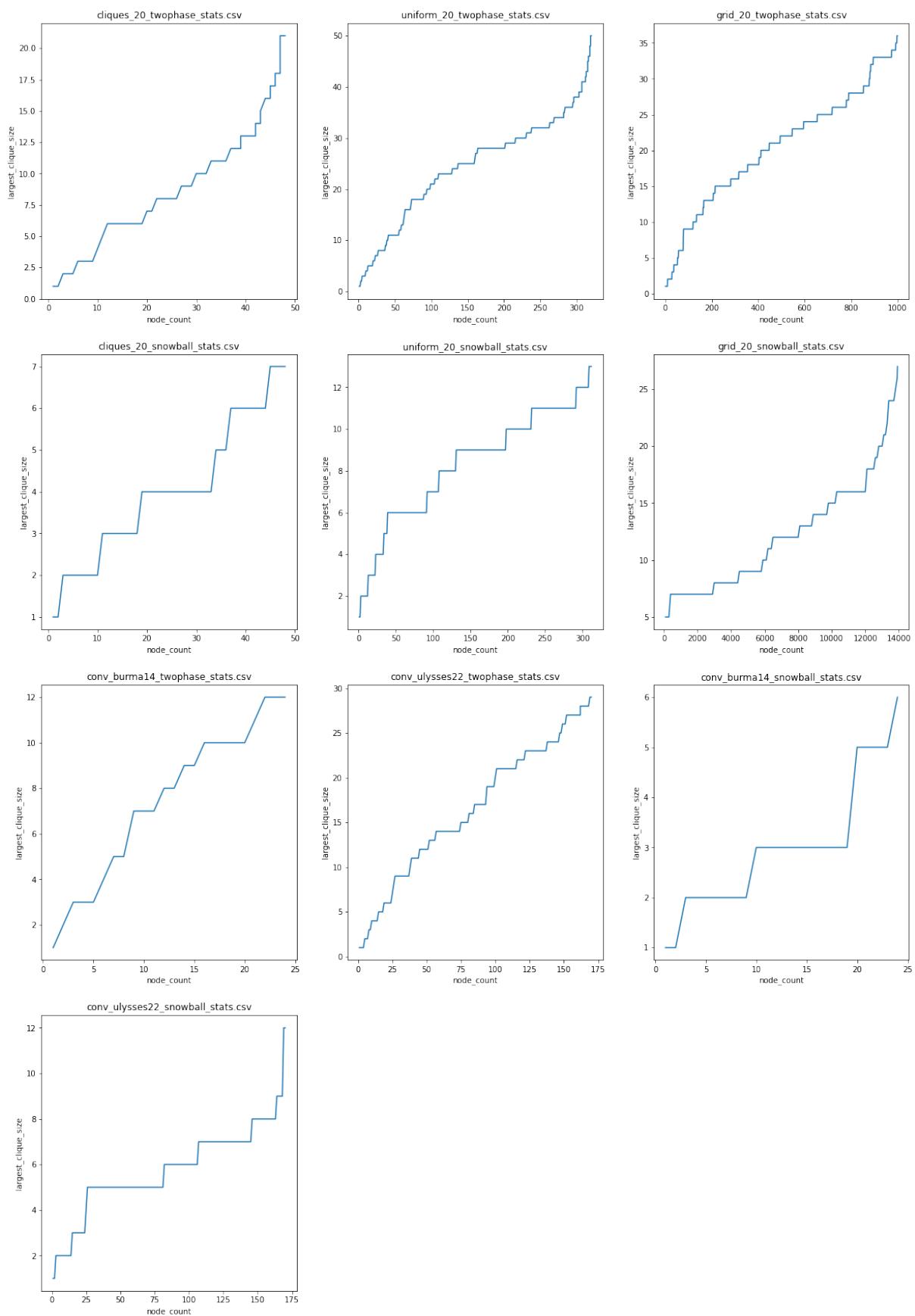
Rysunek A.6 Gęstość grafu (density) w zależności od liczby wierzchołków



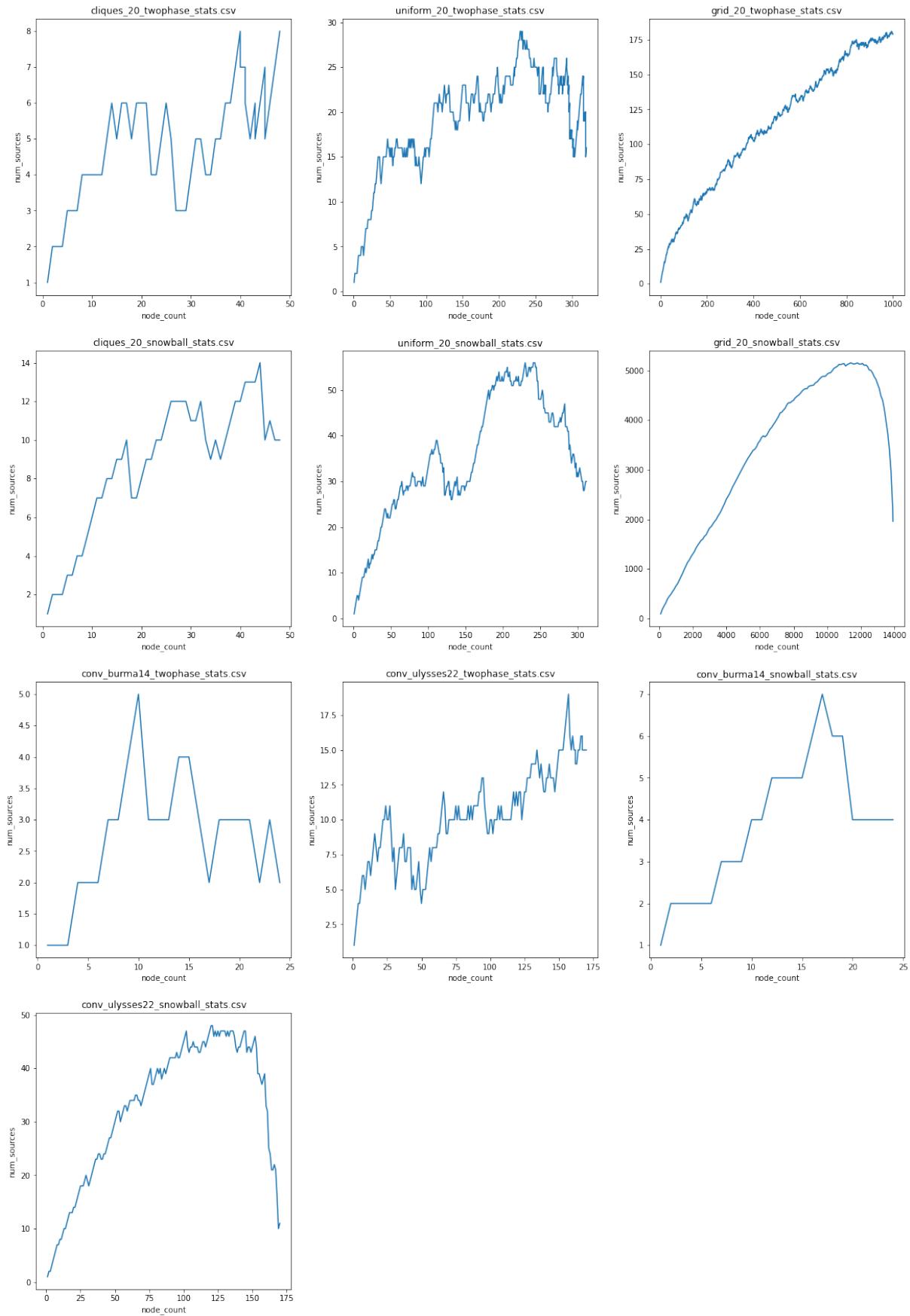
Rysunek A.7 Współczynnik distLO w zależności od liczby wierzchołków



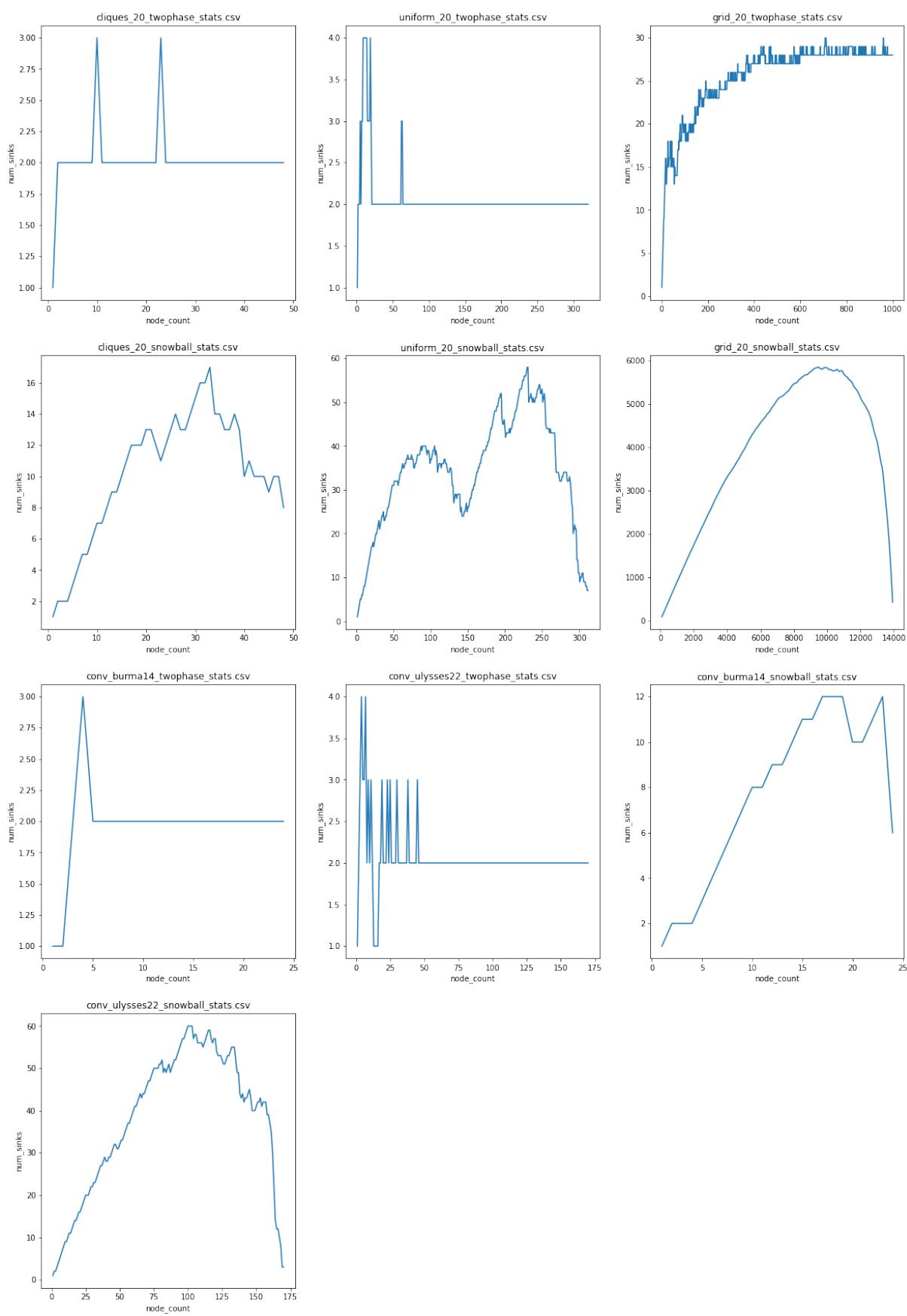
Rysunek A.8 Współczynnik wzajemności grafu (reciprocity) w zależności od liczby wierzchołków



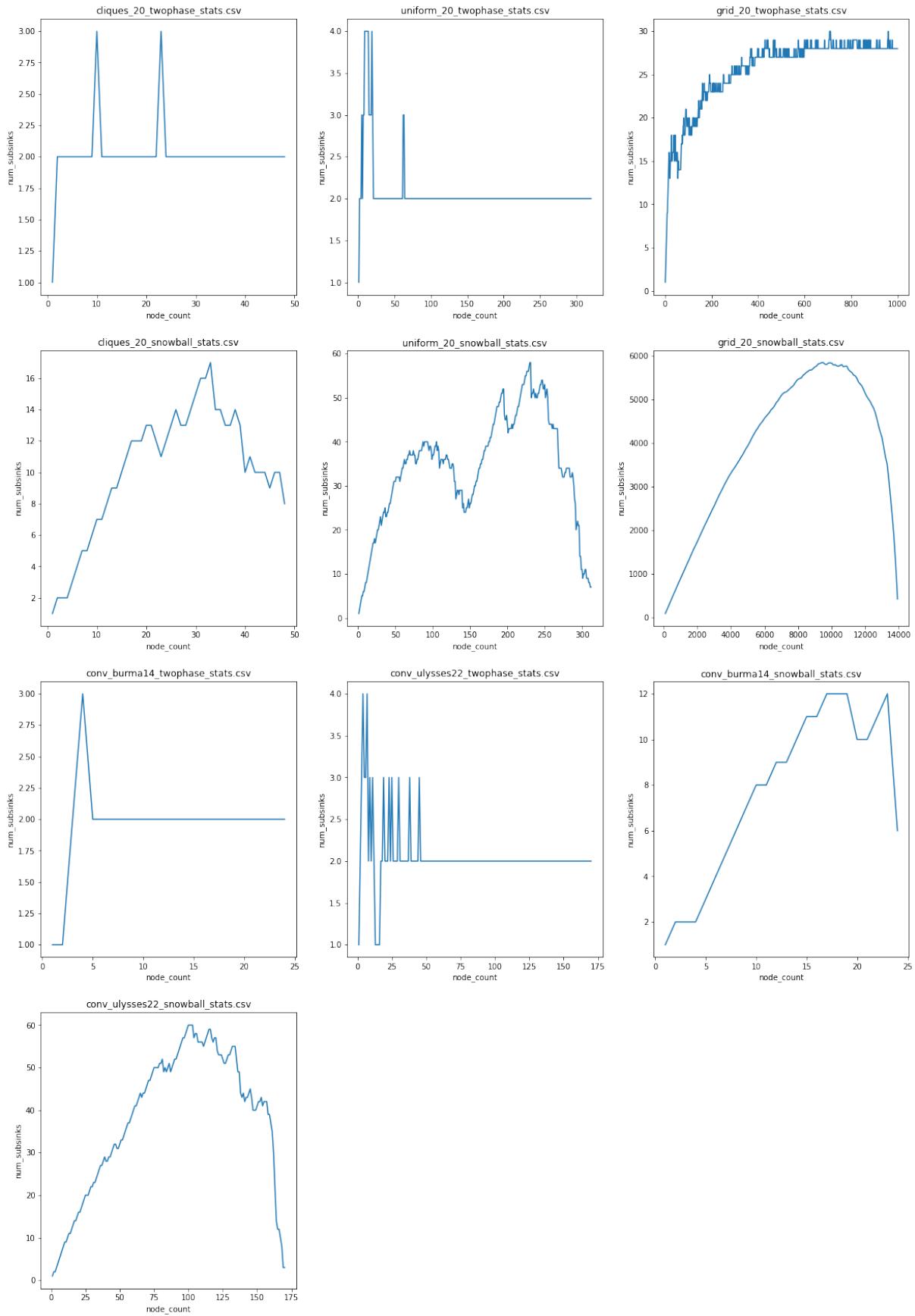
Rysunek A.9 Rozmiar największej kliki w grafie (largest_clique_size) w zależności od liczby wierzchołków



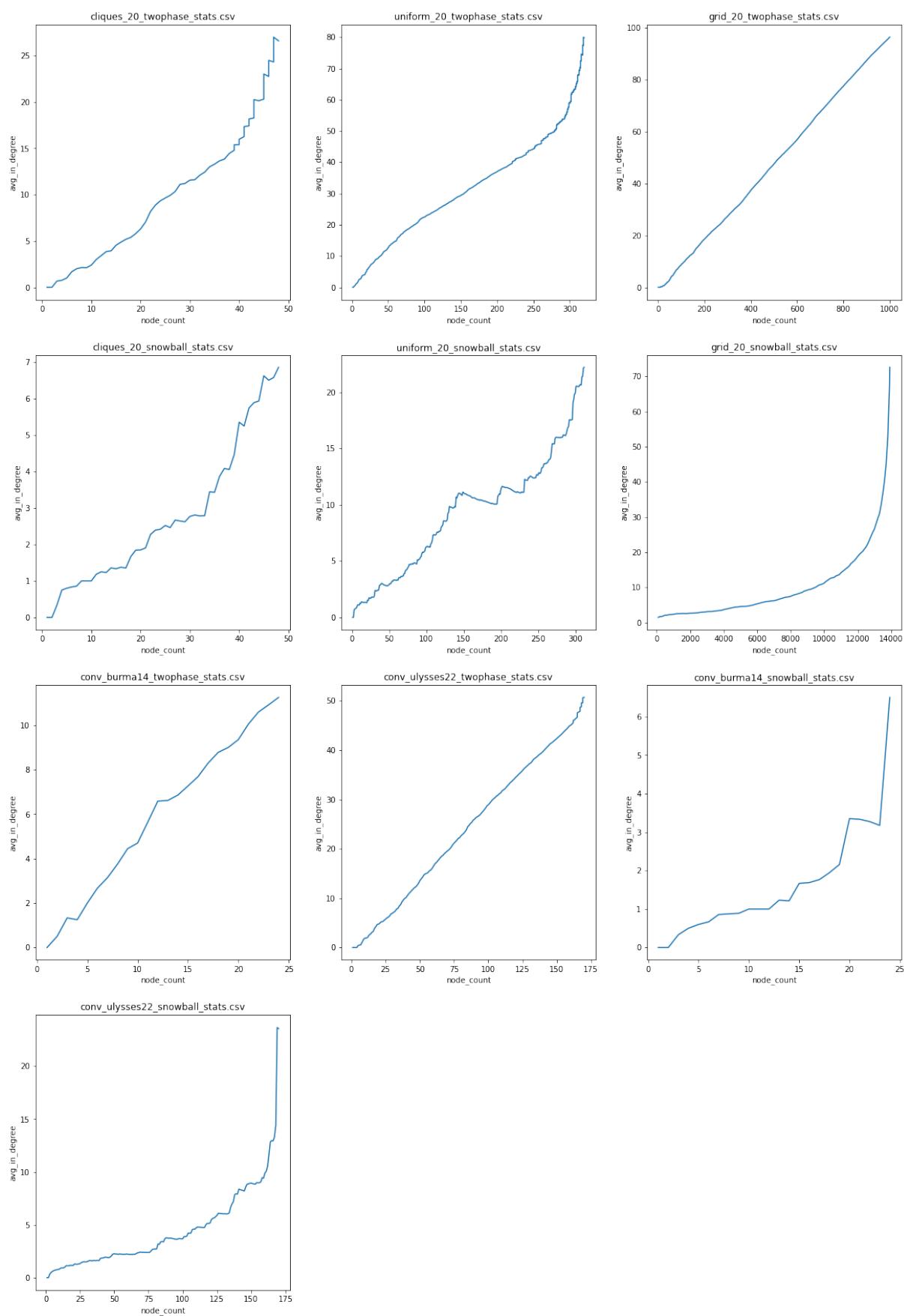
Rysunek A.10 Liczba źródeł w grafie (`num_sources`) w zależności od liczby wierzchołków



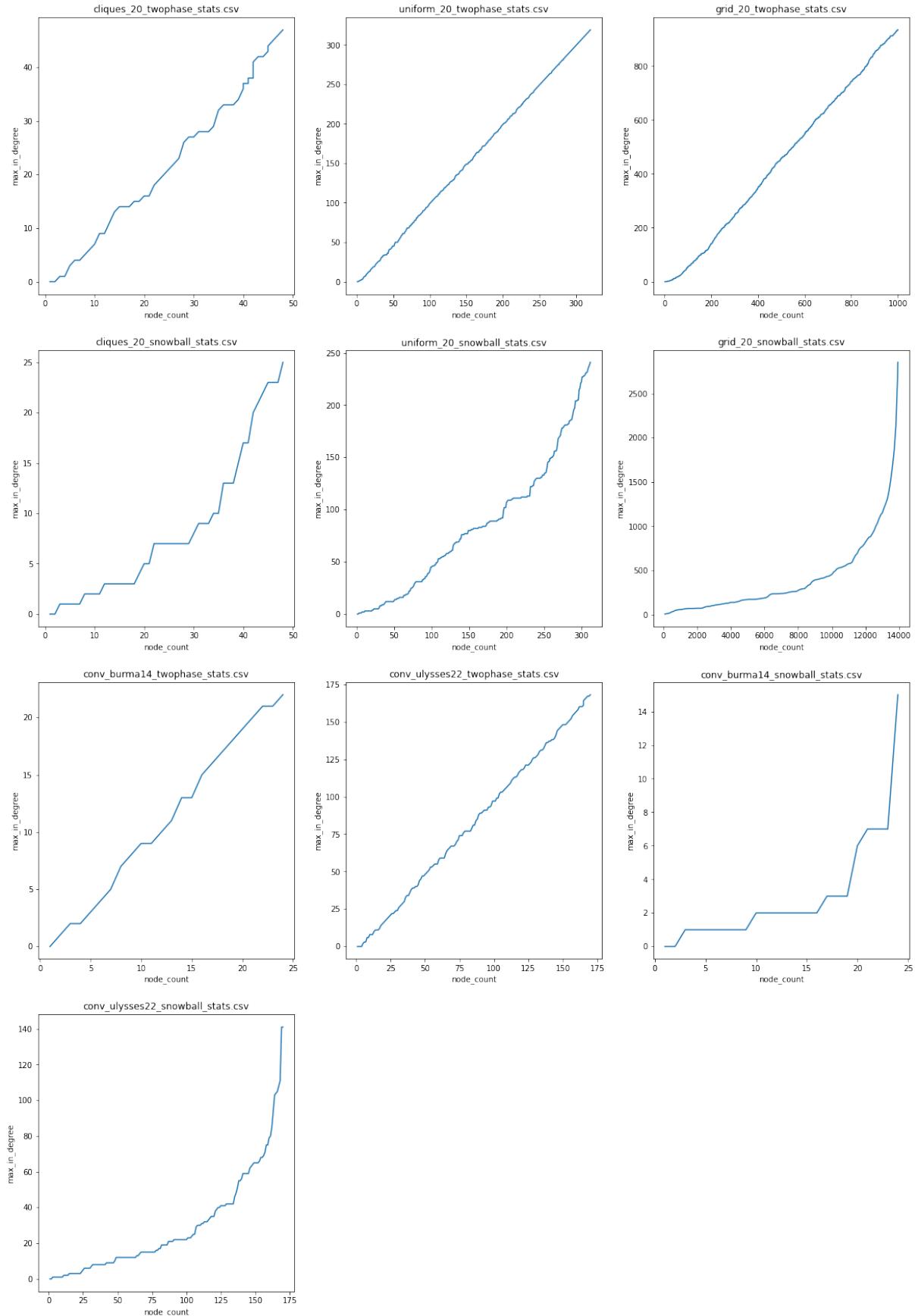
Rysunek A.11 Liczba ścieków w grafie (`num_sinks`) w zależności od liczby wierzchołków



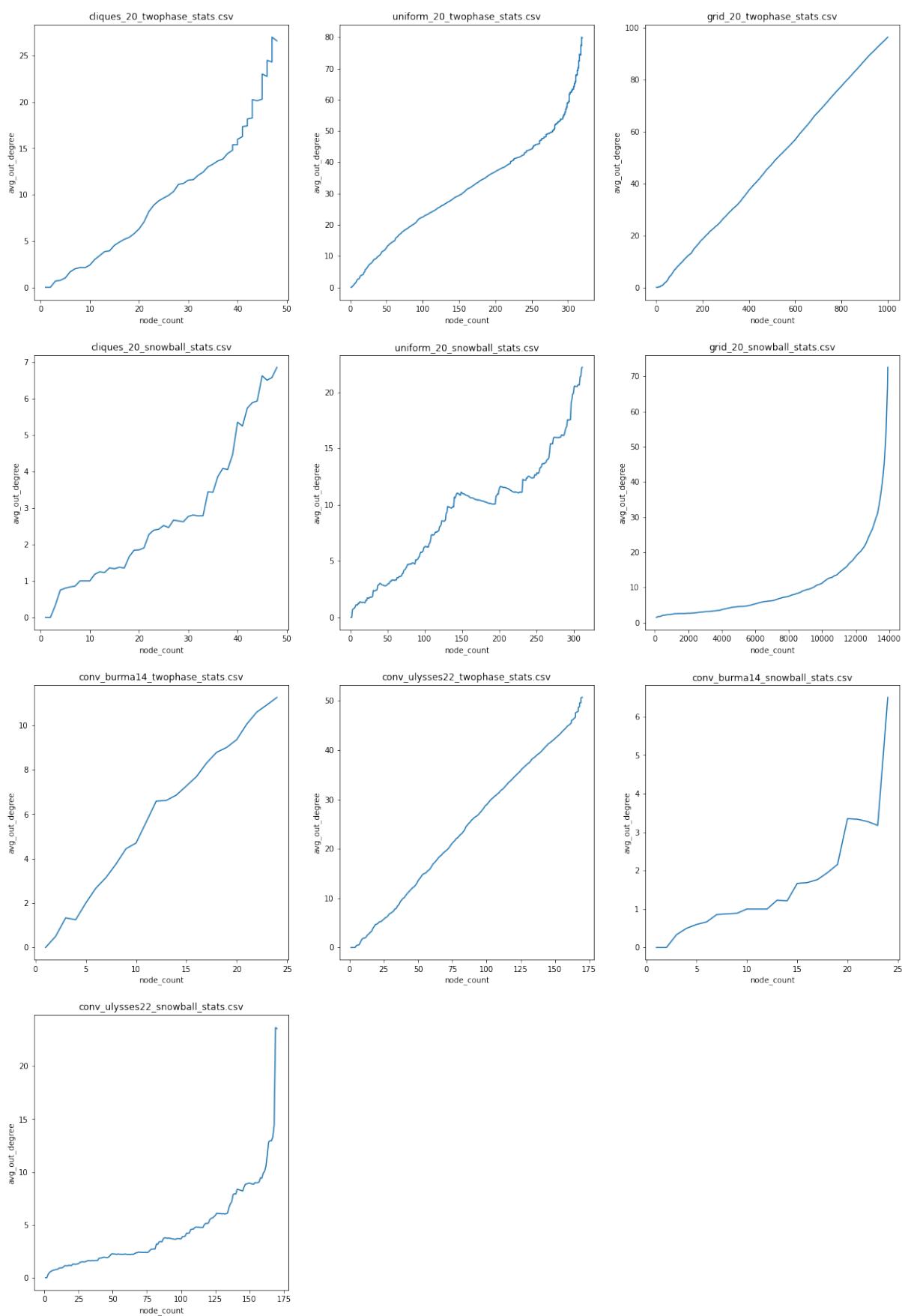
Rysunek A.12 Liczba subsinks w grafie (num_sub sinks) w zależności od liczby wierzchołków



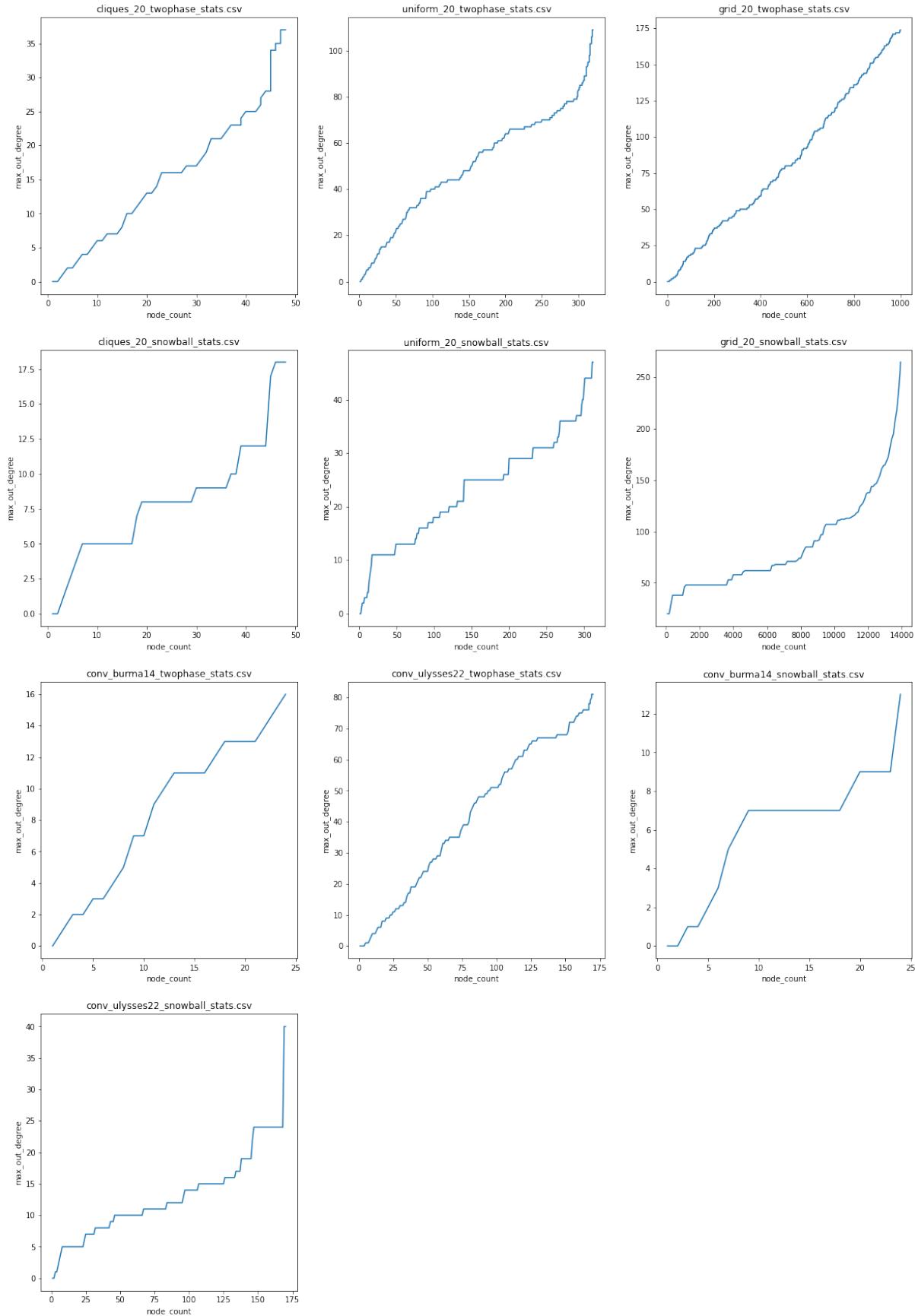
Rysunek A.13 Średni stopień wchodzący wierzchołków w grafie (`avg_in_degree`) w zależności od liczby wierzchołków



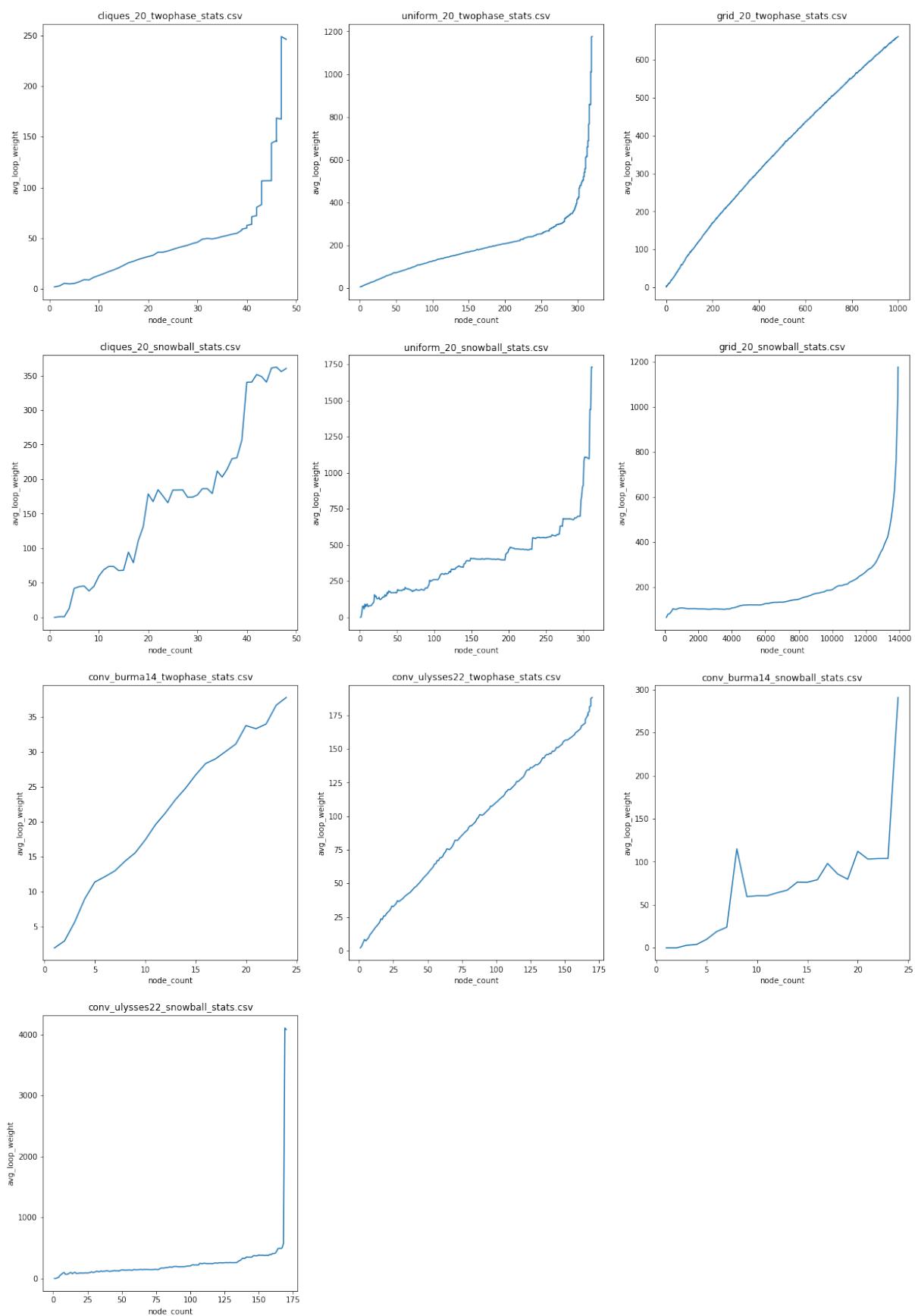
Rysunek A.14 Maksymalny stopień wchodzący wśród wierzchołków w grafie (max_in_degree) w zależności od liczby wierzchołków



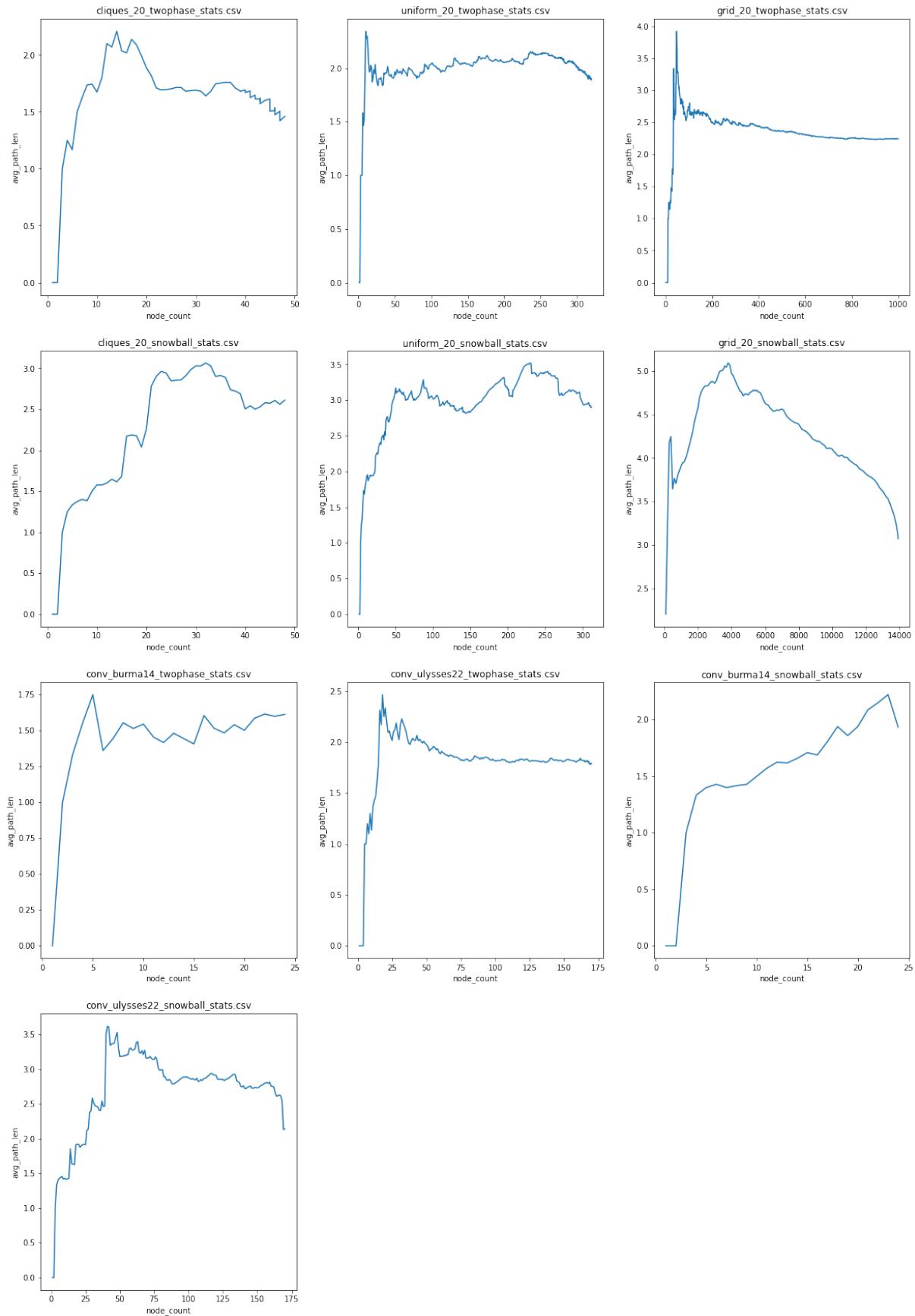
Rysunek A.15 Średni stopień wychodzący wśród wierzchołków w grafie (avg_out_degree) w zależności od liczby wierzchołków



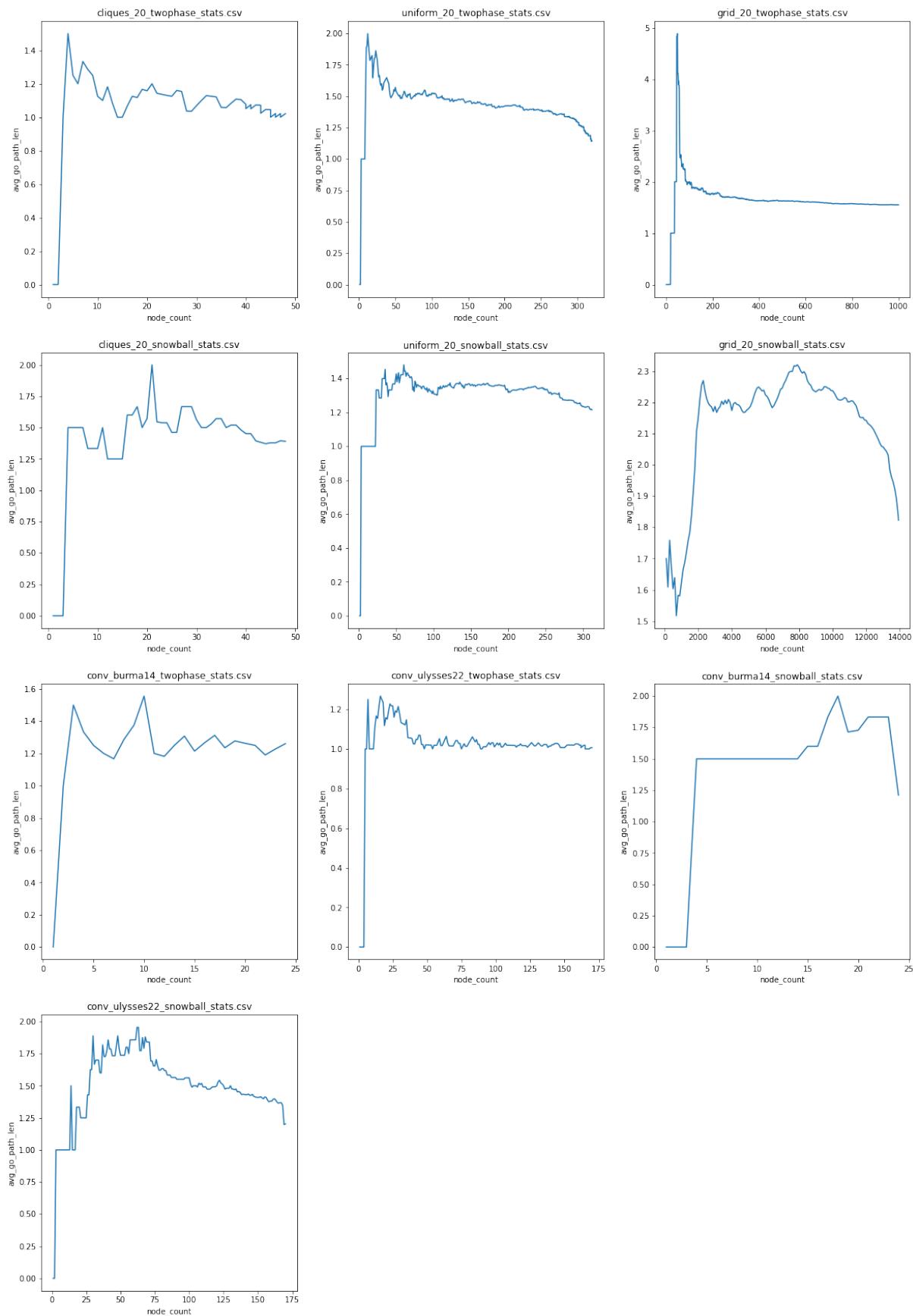
Rysunek A.16 Maksymalny stopień wychodzący wierzchołków w grafie (max_out_degree) w zależności od liczby wierzchołków



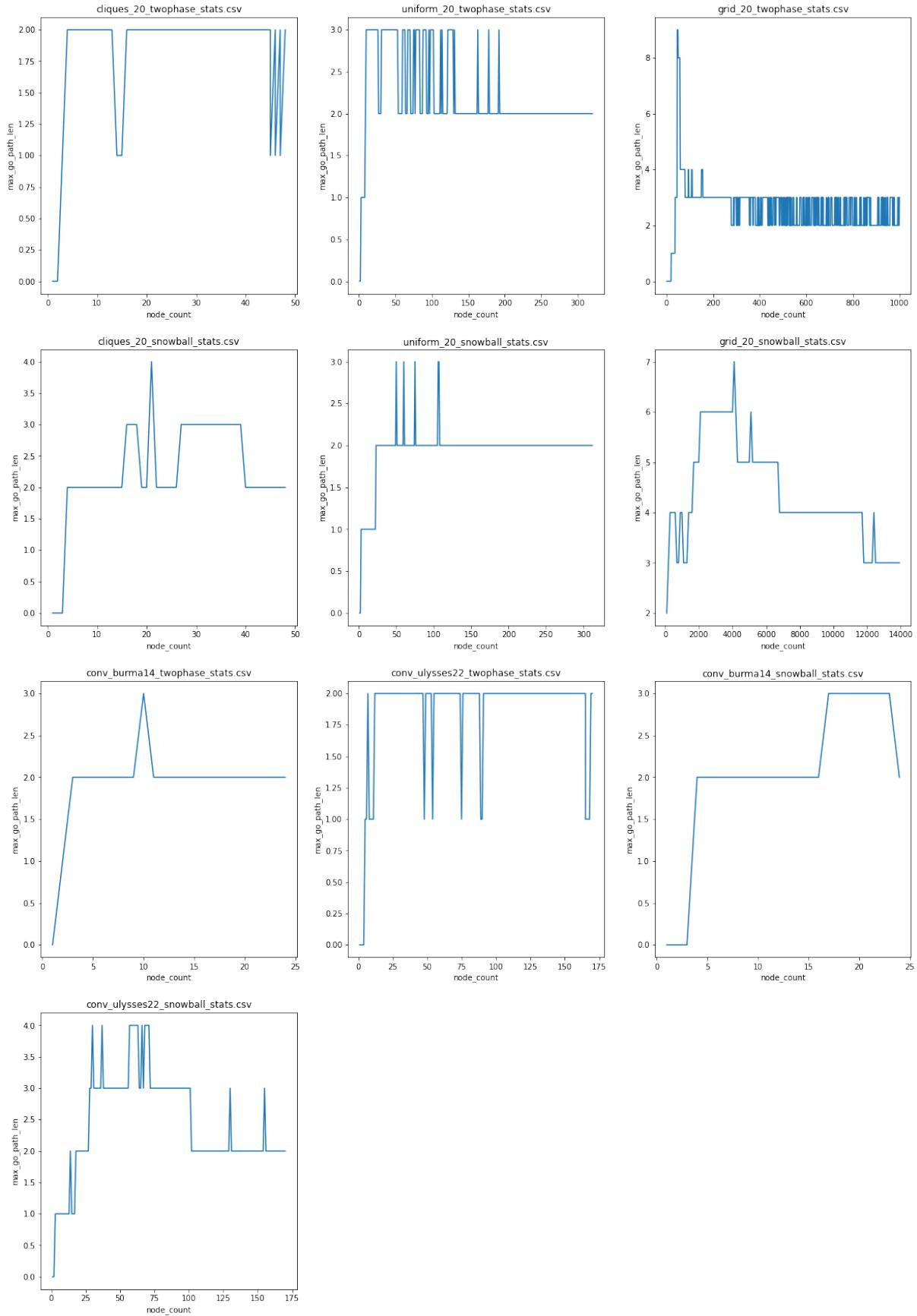
Rysunek A.17 Średnia waga pętli w grafie (`avg_loop_weight`) w zależności od liczby wierzchołków



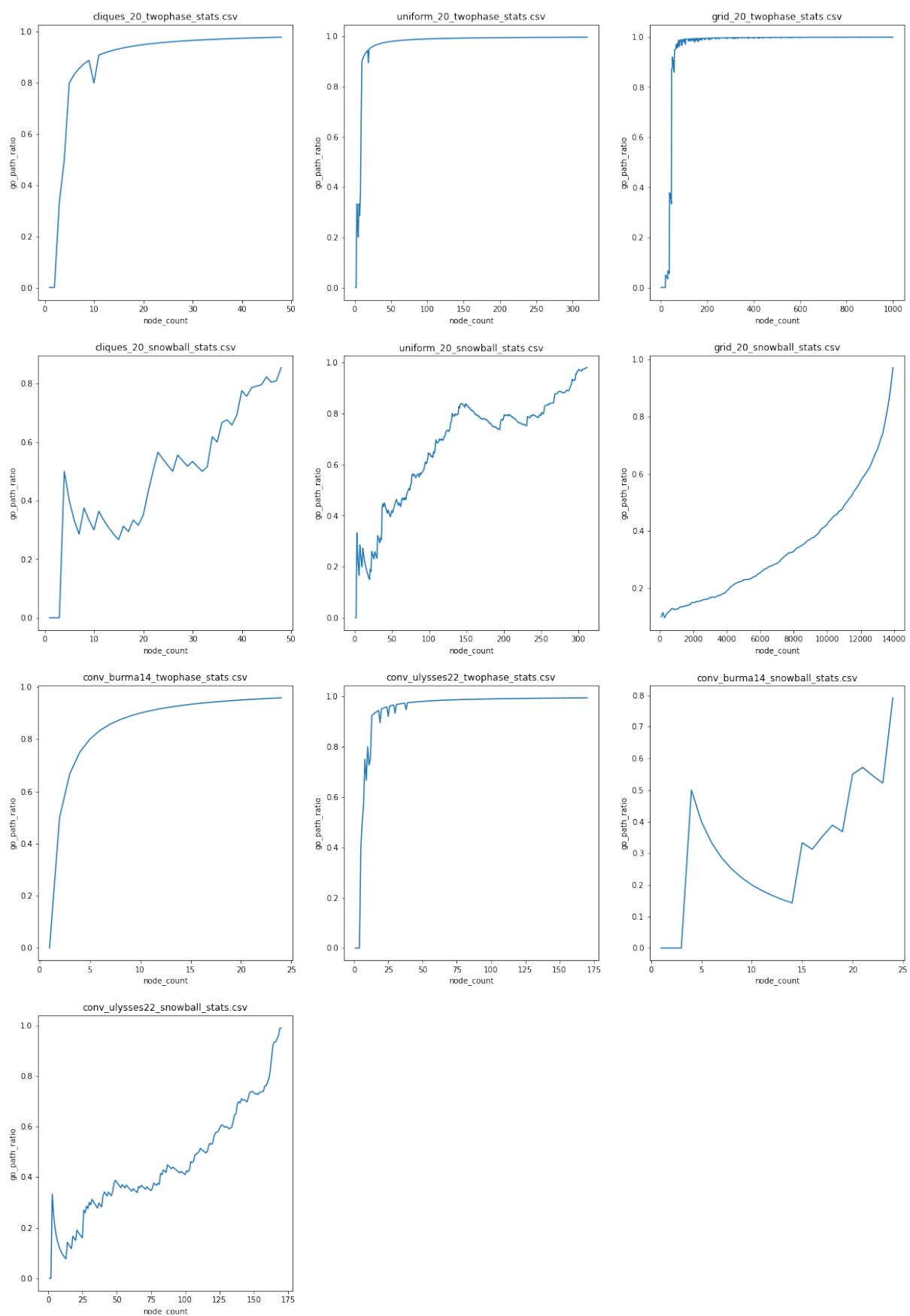
Rysunek A.18 Średnia długość ścieżki w grafie (avg_path_len) w zależności od liczby wierzchołków



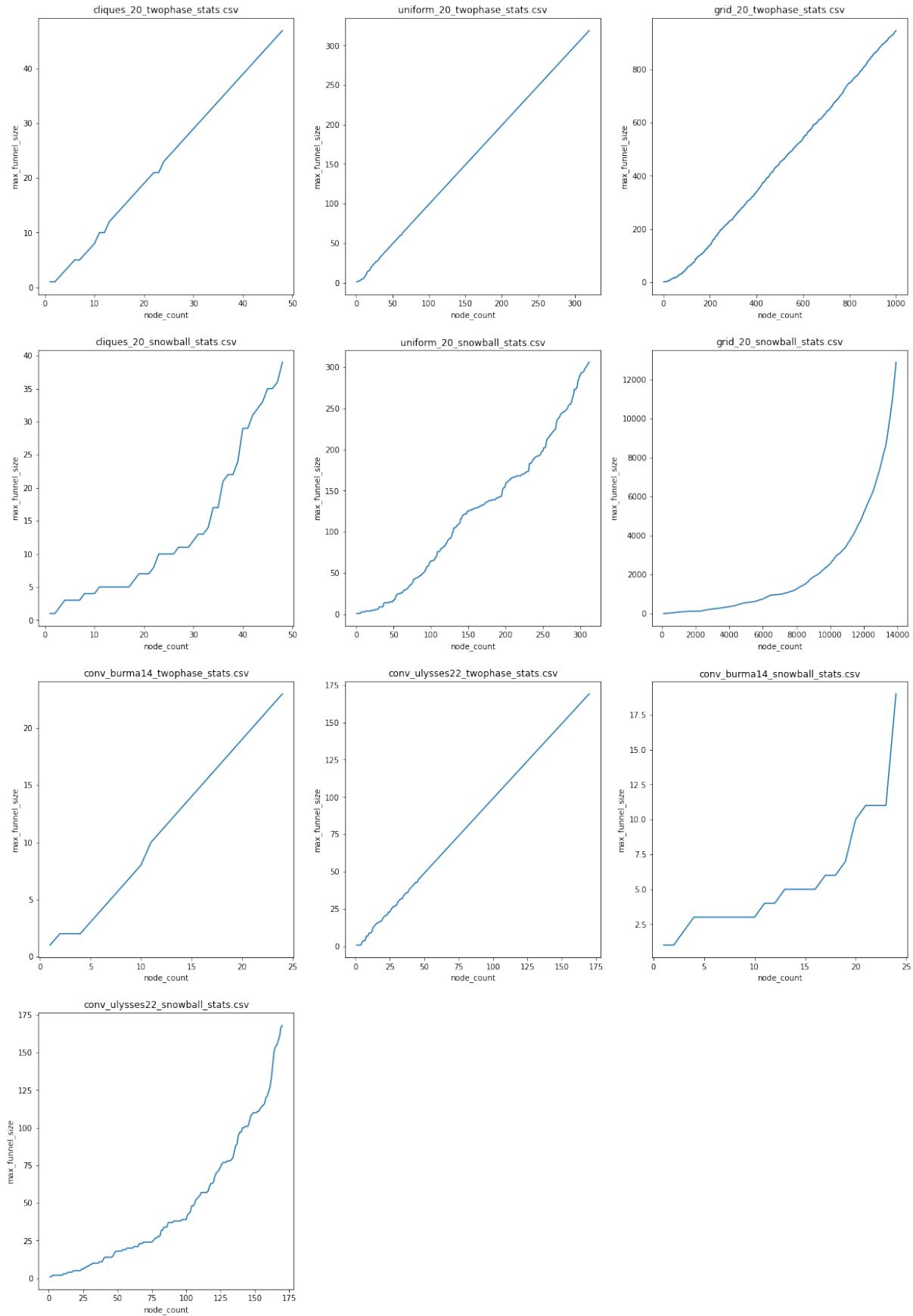
Rysunek A.19 Średnia długość istniejących ścieżek do globalnego optimum (avg_go_path_len) w zależności od liczby wierzchołków



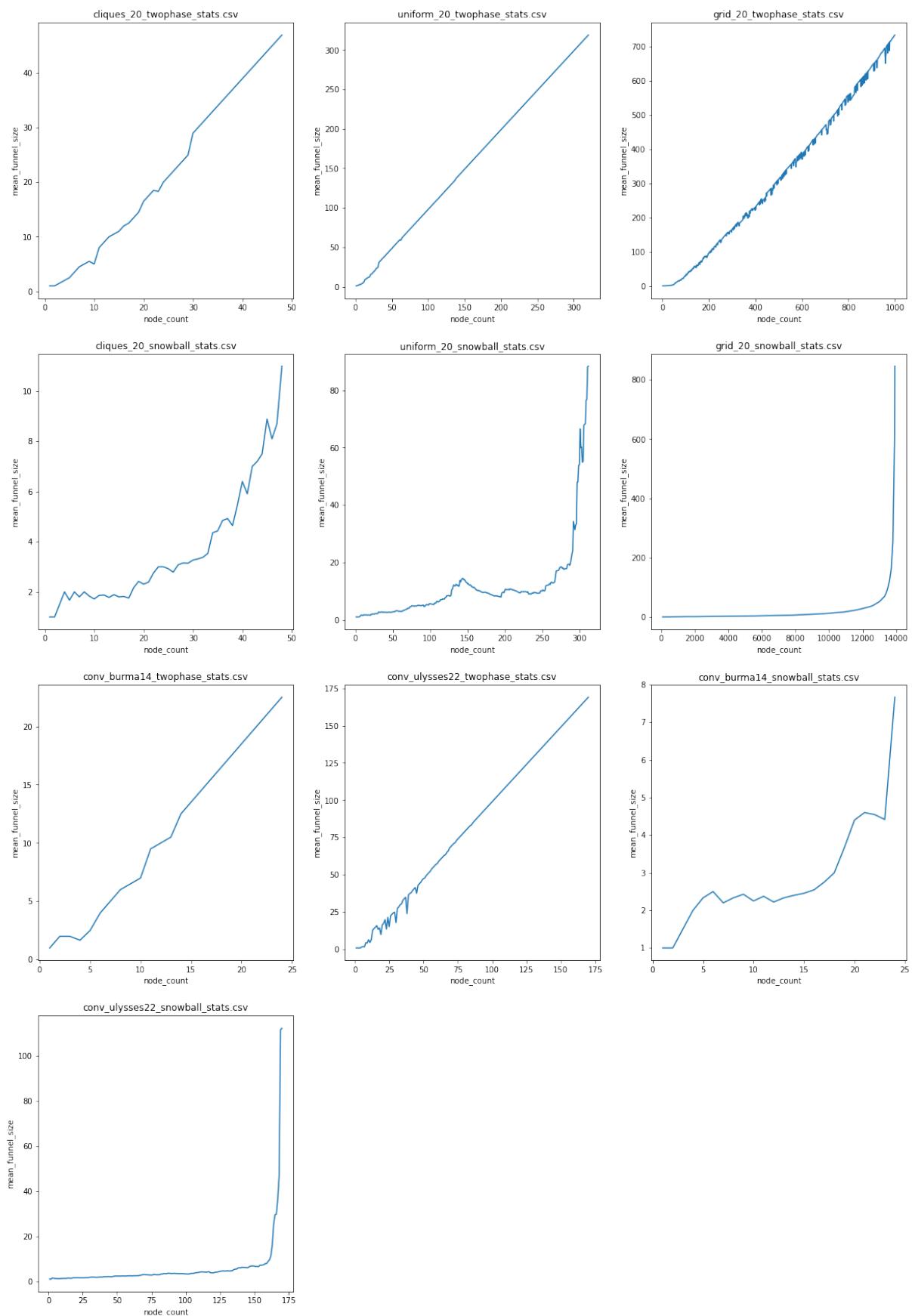
Rysunek A.20 Długość najdłuższej istniejącej ścieżki do globalnego optimum (`max_go_path_len`) w zależności od liczby wierzchołków



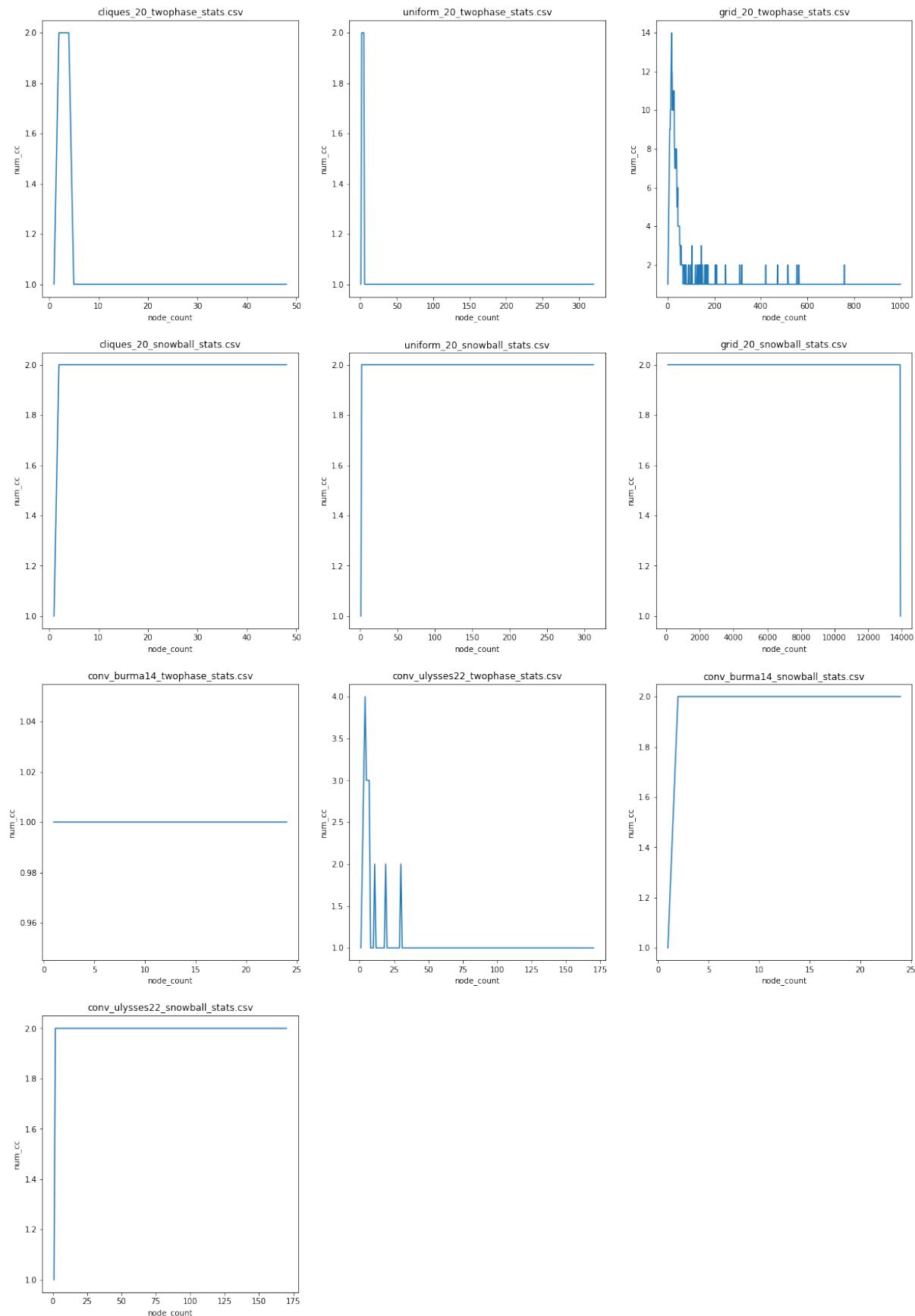
Rysunek A.21 Stosunek liczby wierzchołków, z których istnieje ścieżka do globalnego optimum do liczby wszystkich wierzchołków (go_path_ratio) w zależności od liczby wierzchołków



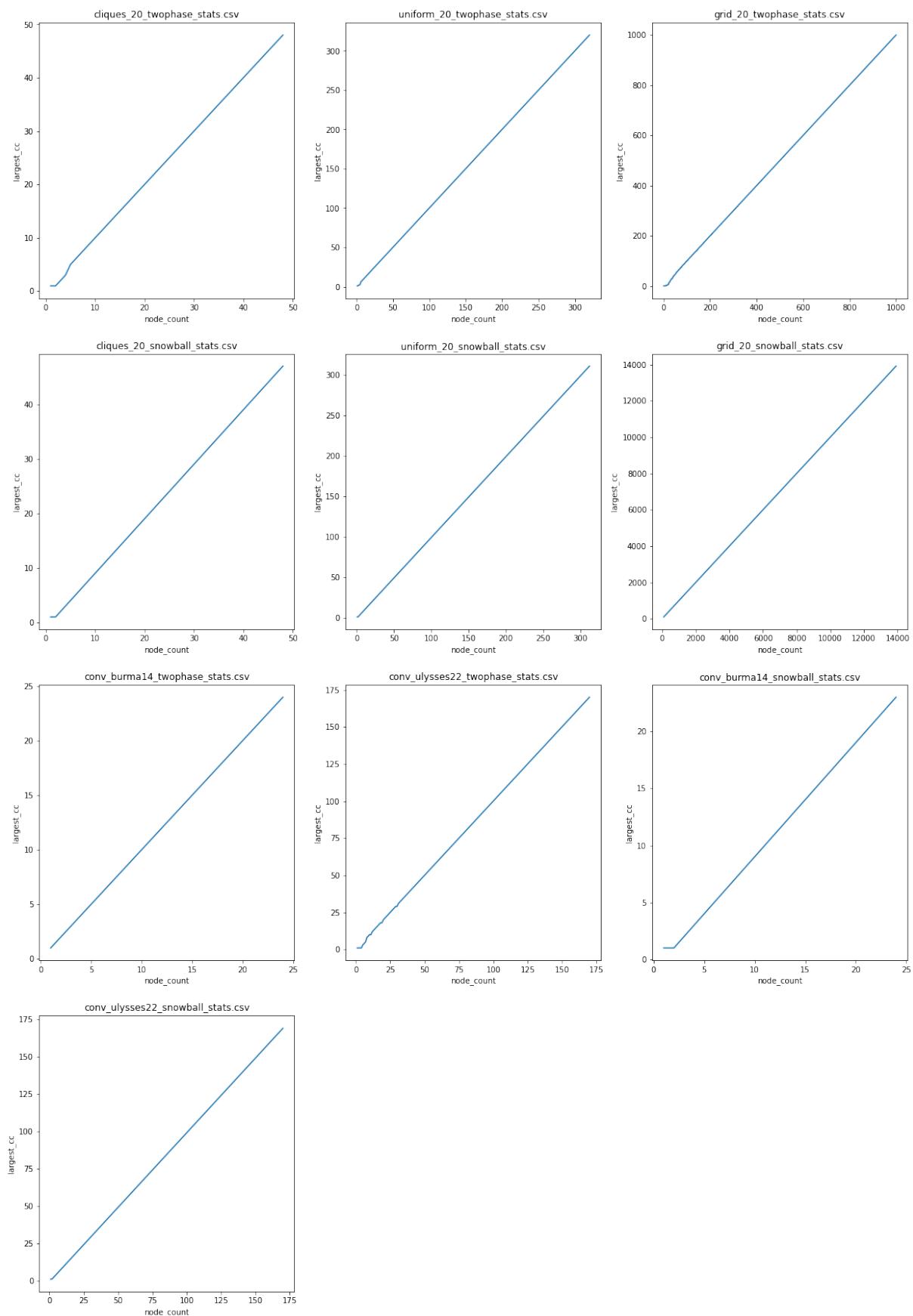
Rysunek A.22 Rozmiar największego leja w przestrzeni (max_funnel_size) w zależności od liczby wierzchołków



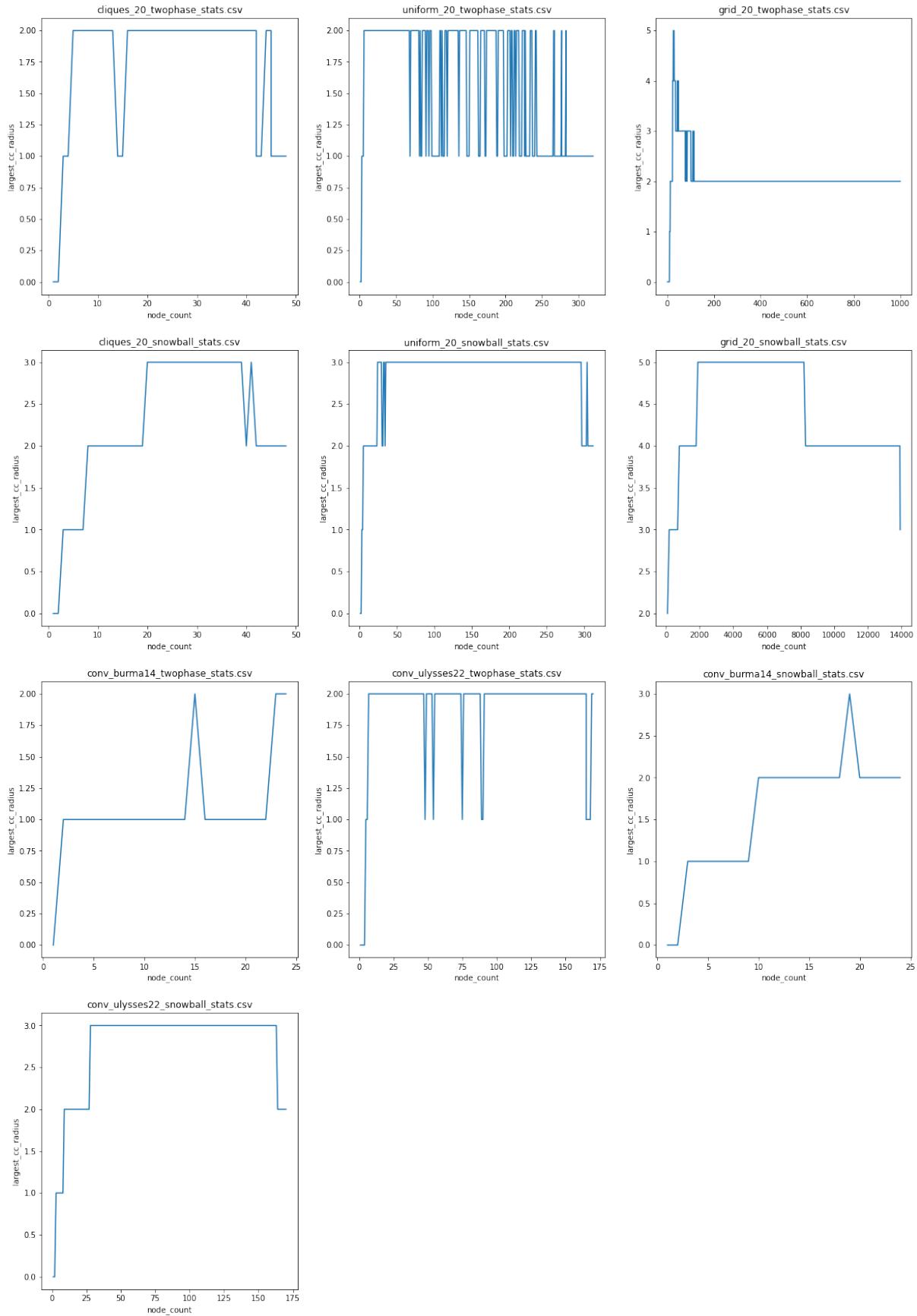
Rysunek A.23 Średni rozmiar lejów w przestrzeni (mean_funnel_size) w zależności od liczby wierzchołków



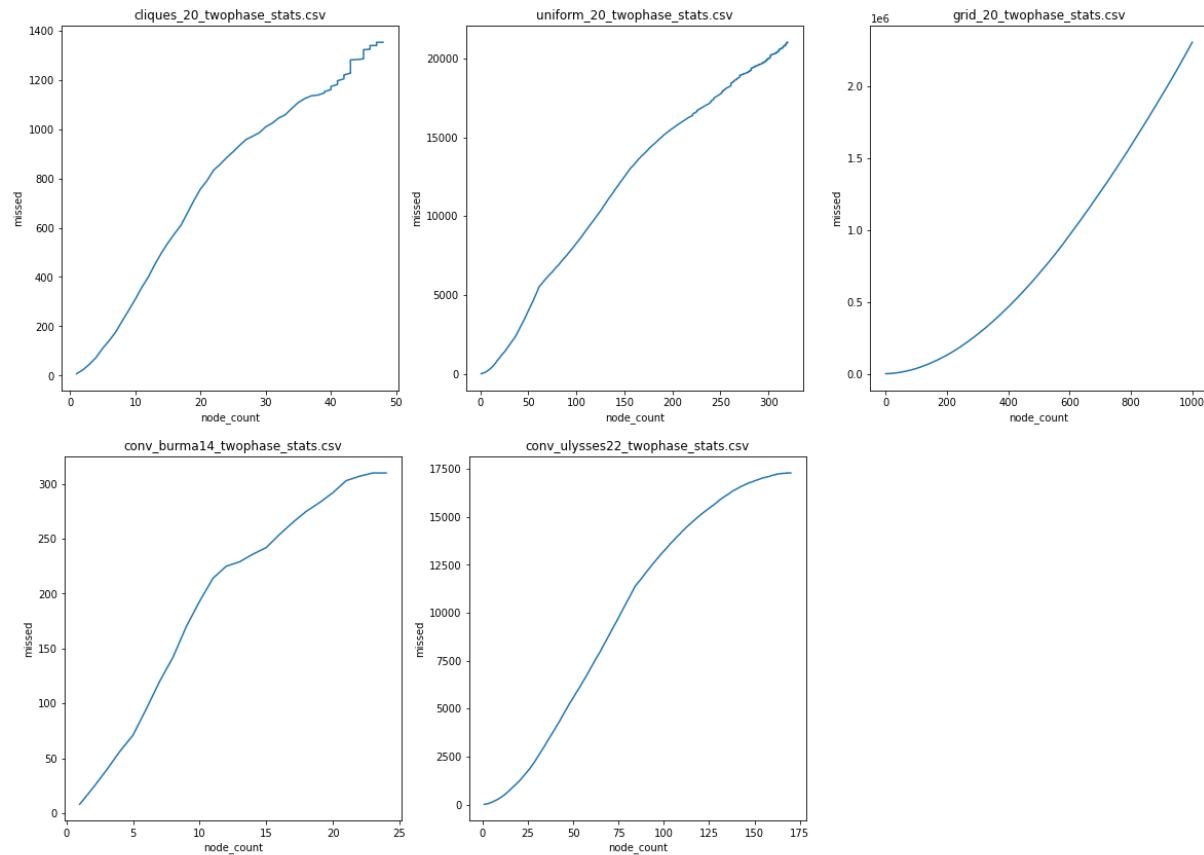
Rysunek A.24 Liczba spójnych podgrafów (num_cc) w zależności od liczby wierzchołków



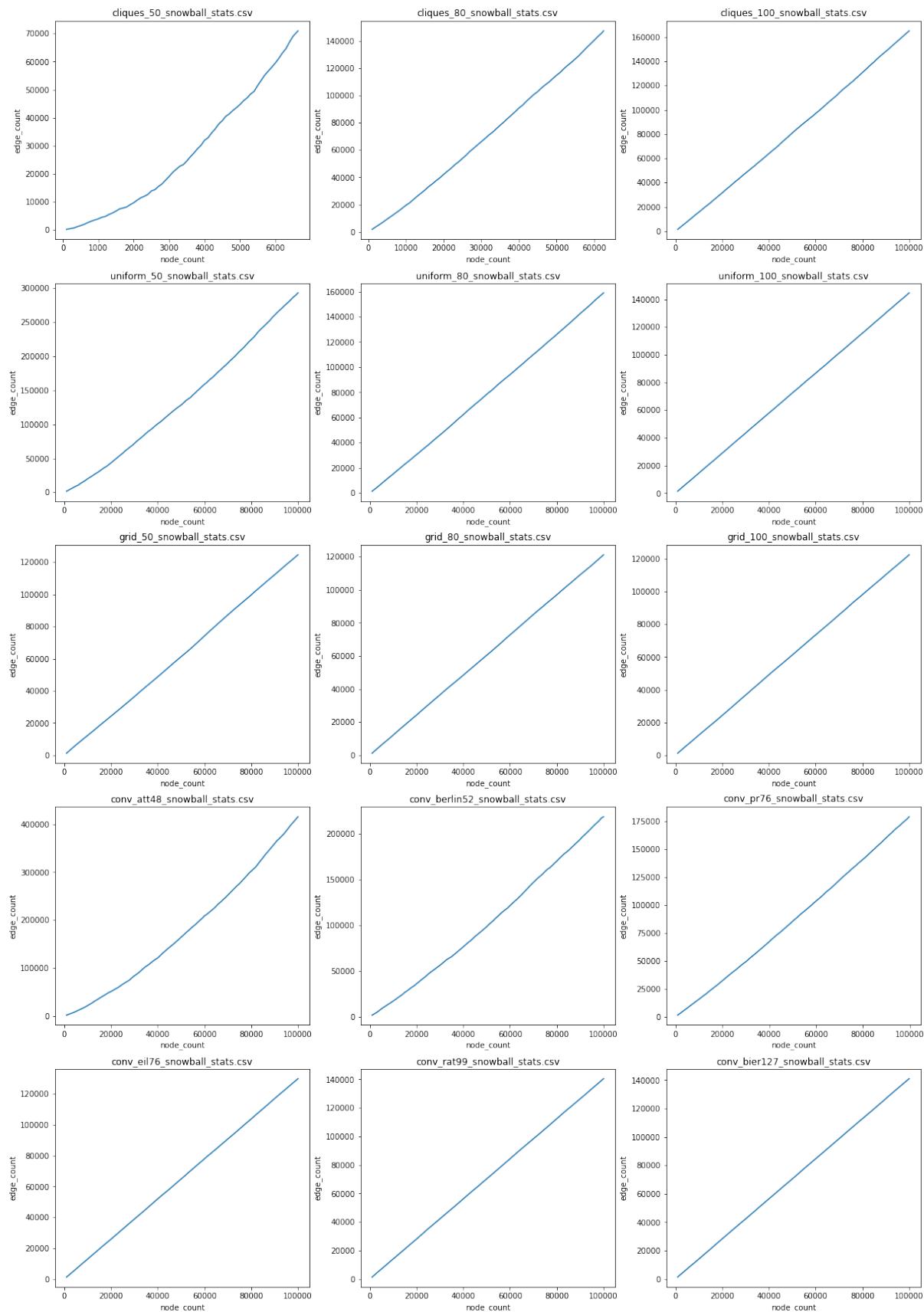
Rysunek A.25 Rozmiar największego spójnego podgrafa (largest_cc) w zależności od liczby wierzchołków



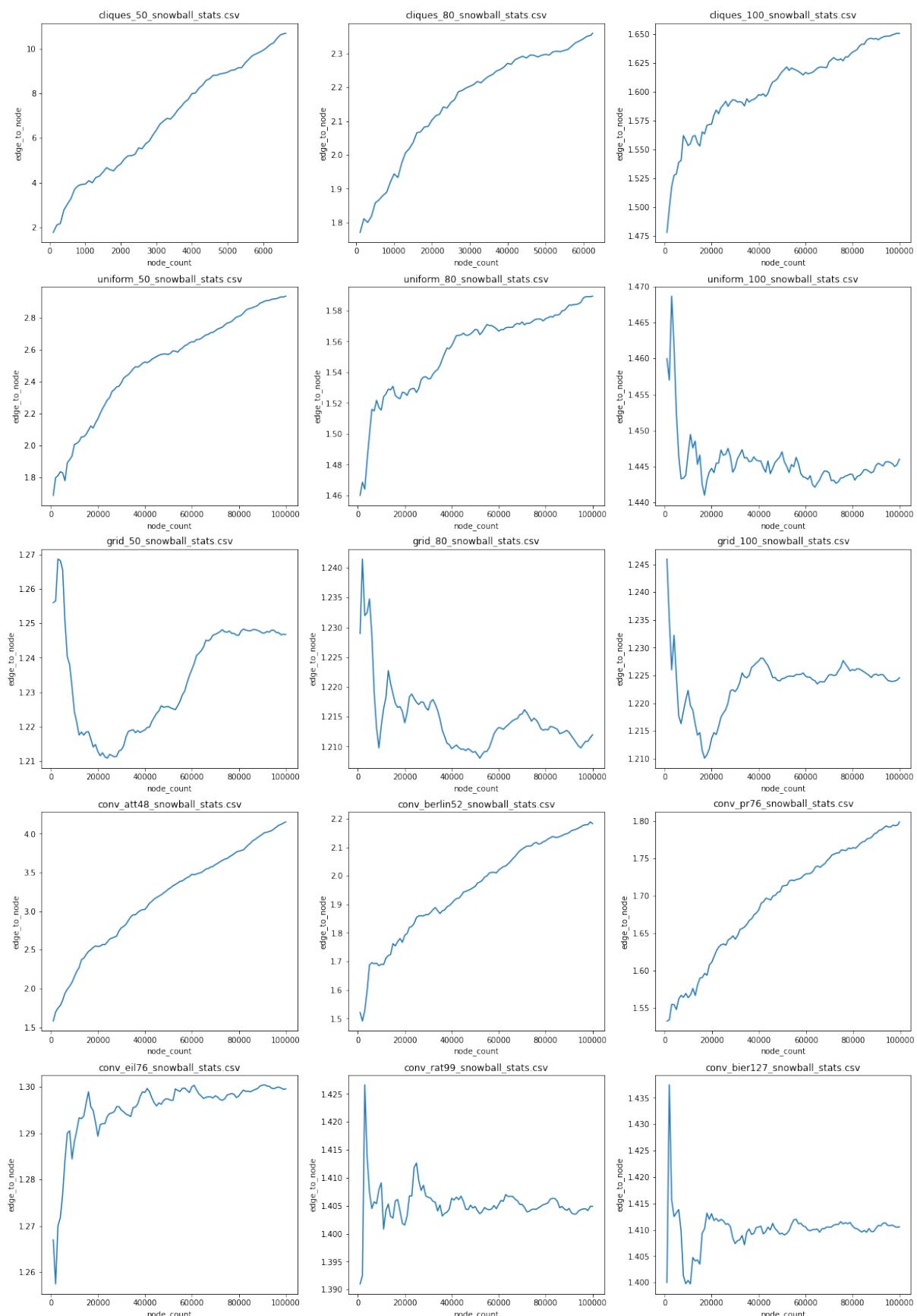
Rysunek A.26 Promień największego spójnego podgrafu (largest_cc_radius) w zależności od liczby wierzchołków



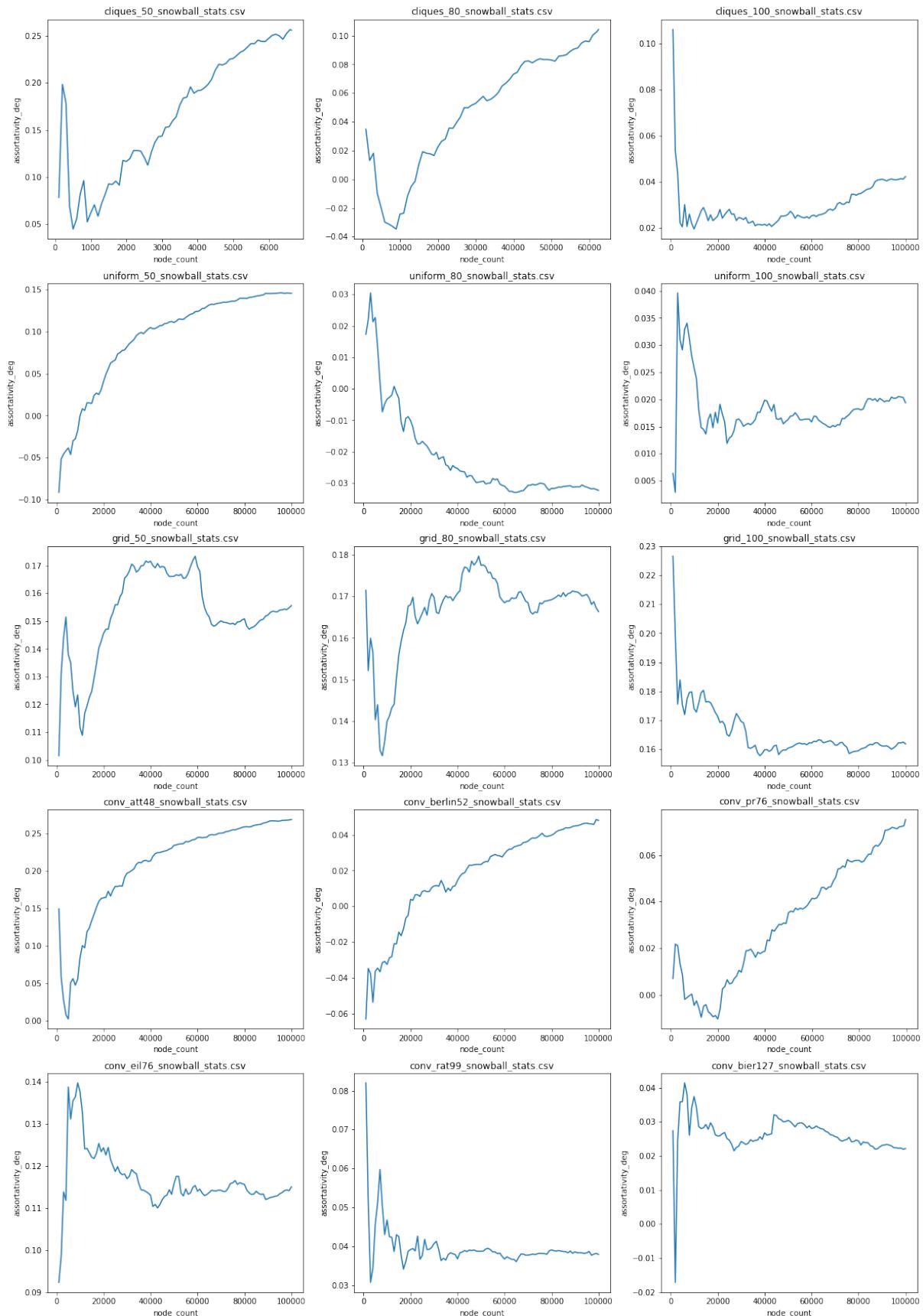
Rysunek A.27 Liczba nieudanych prób tworzenia krawędzi w zależności od liczby wierzchołków — próbkowanie dwufazowe



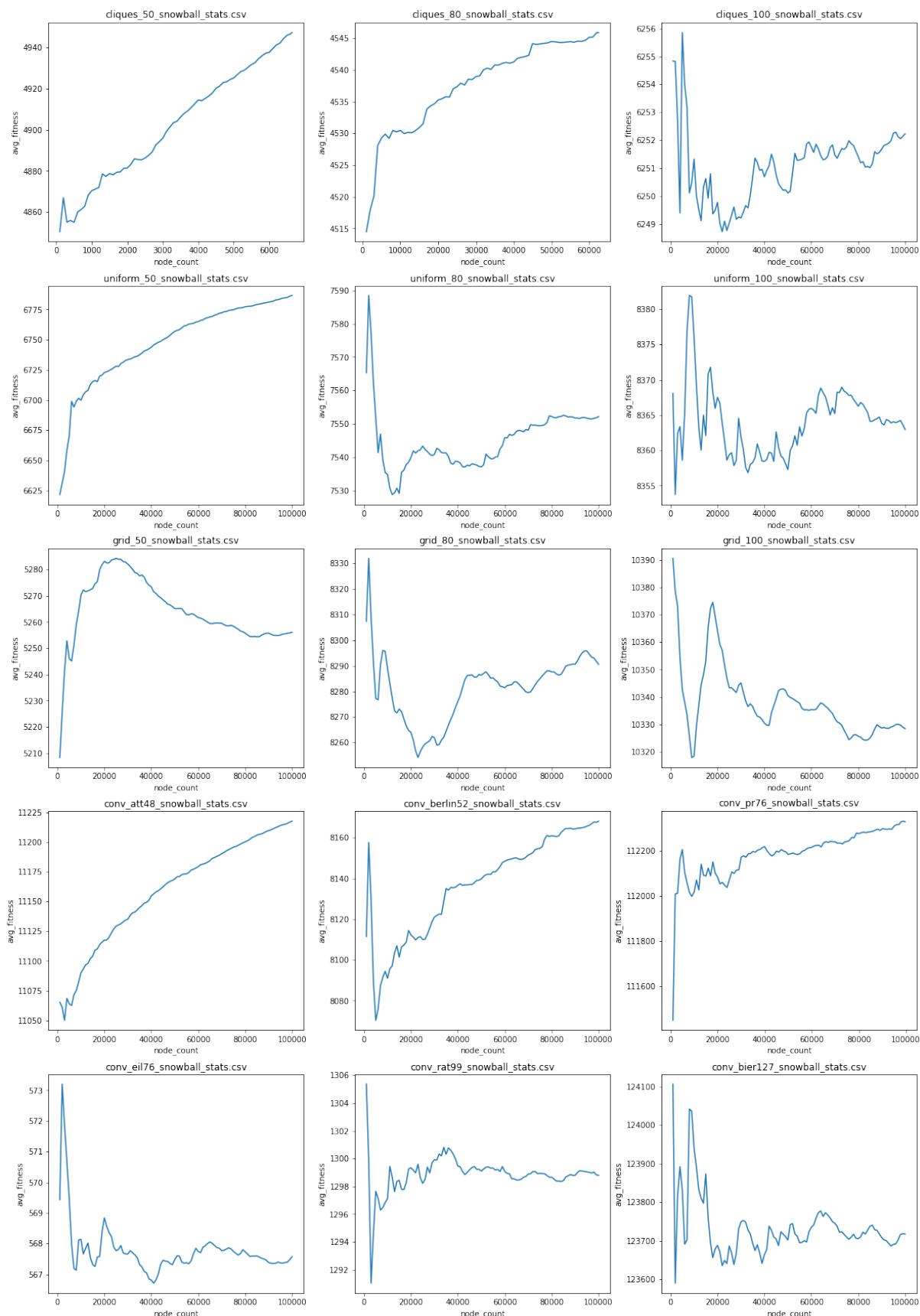
Rysunek A.28 Liczba krawędzi (edge_count) w zależności od liczby wierzchołków — próbkowanie snowball



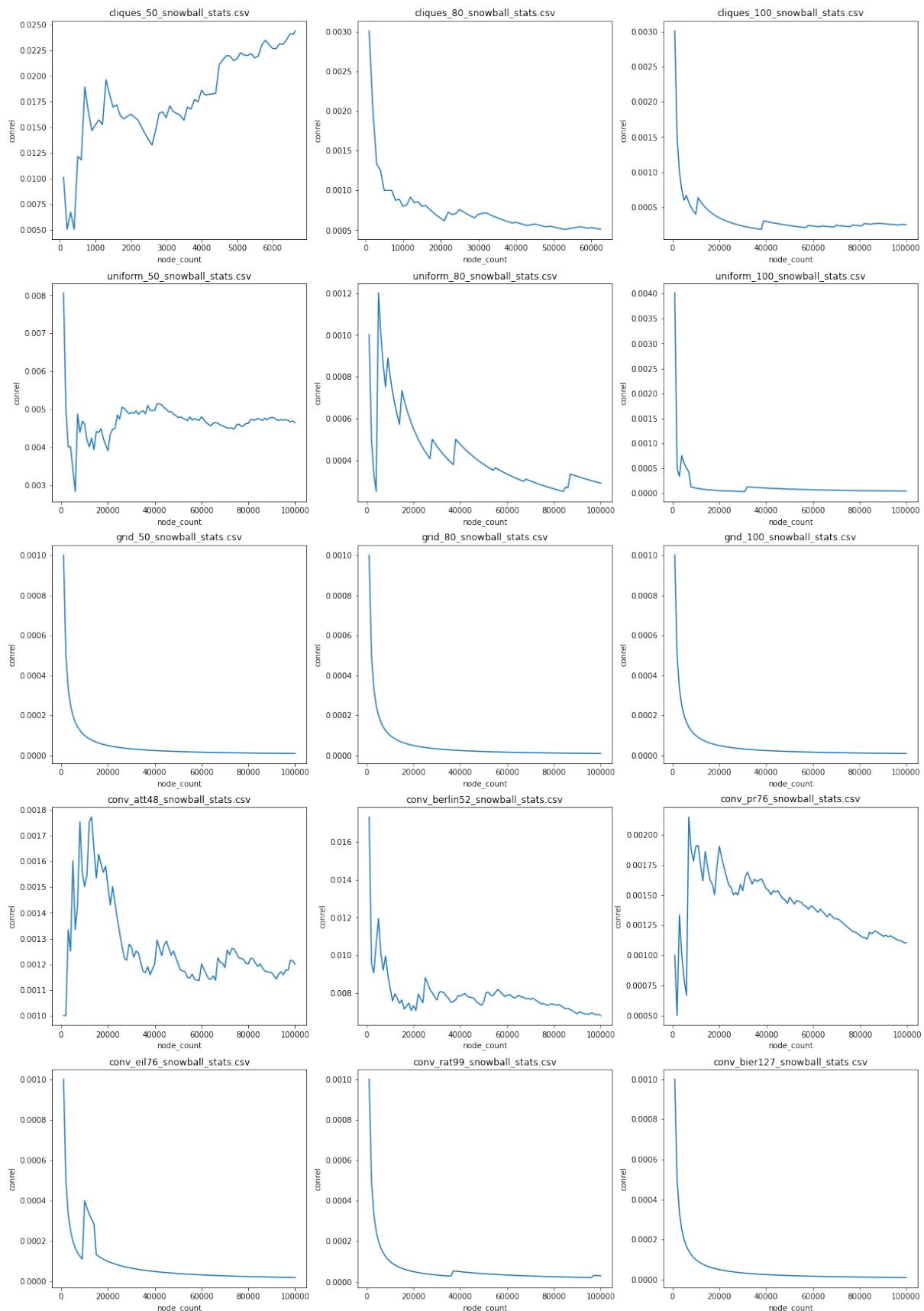
Rysunek A.29 Stosunek liczby krawędzi do liczby wierzchołków (`edge_to_node`) w zależności od liczby wierzchołków — próbkowanie snowball



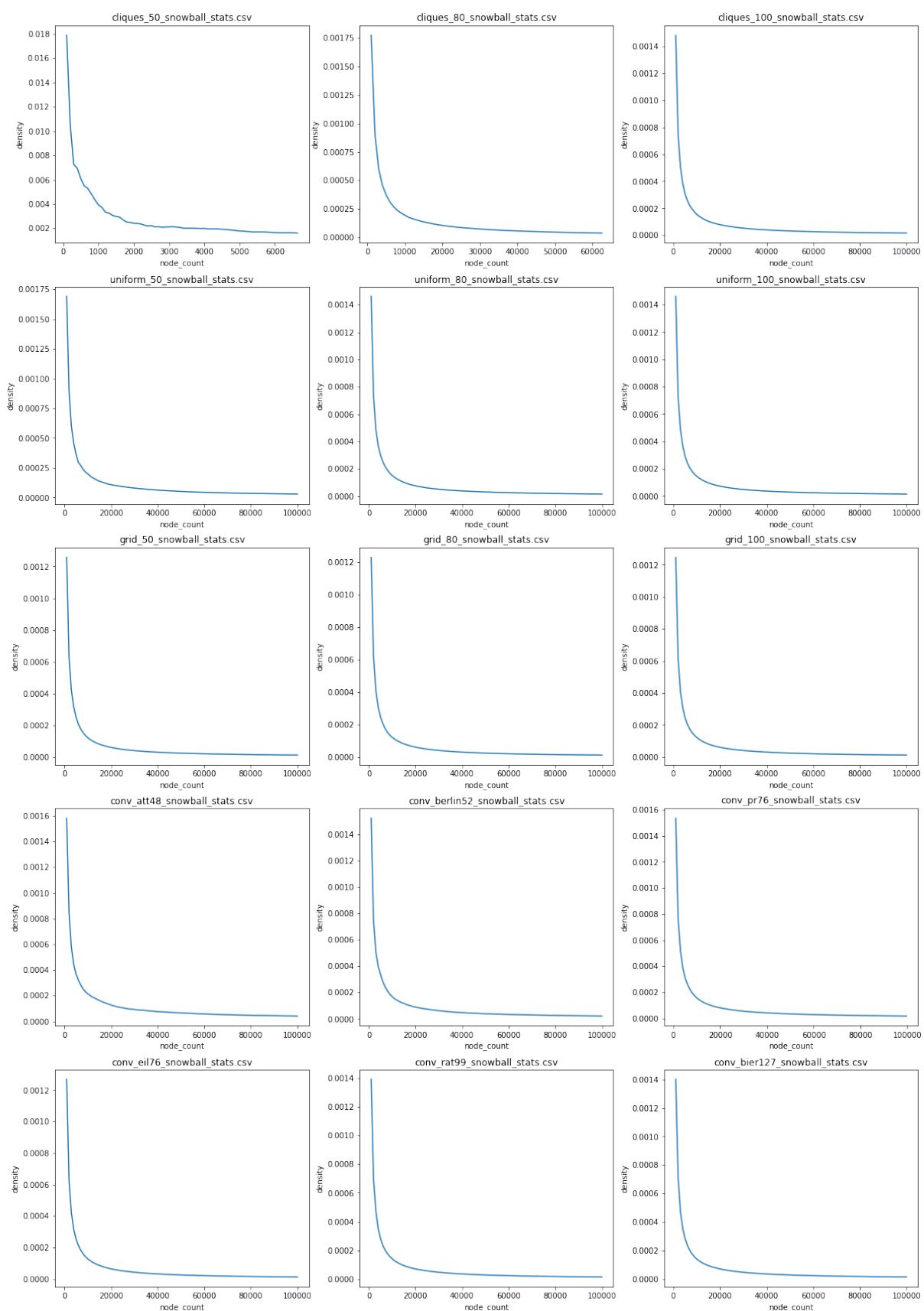
Rysunek A.30 Współczynnik różnorodności grafu (assortativity_deg) w zależności od liczby wierzchołków — próbkowanie snowball



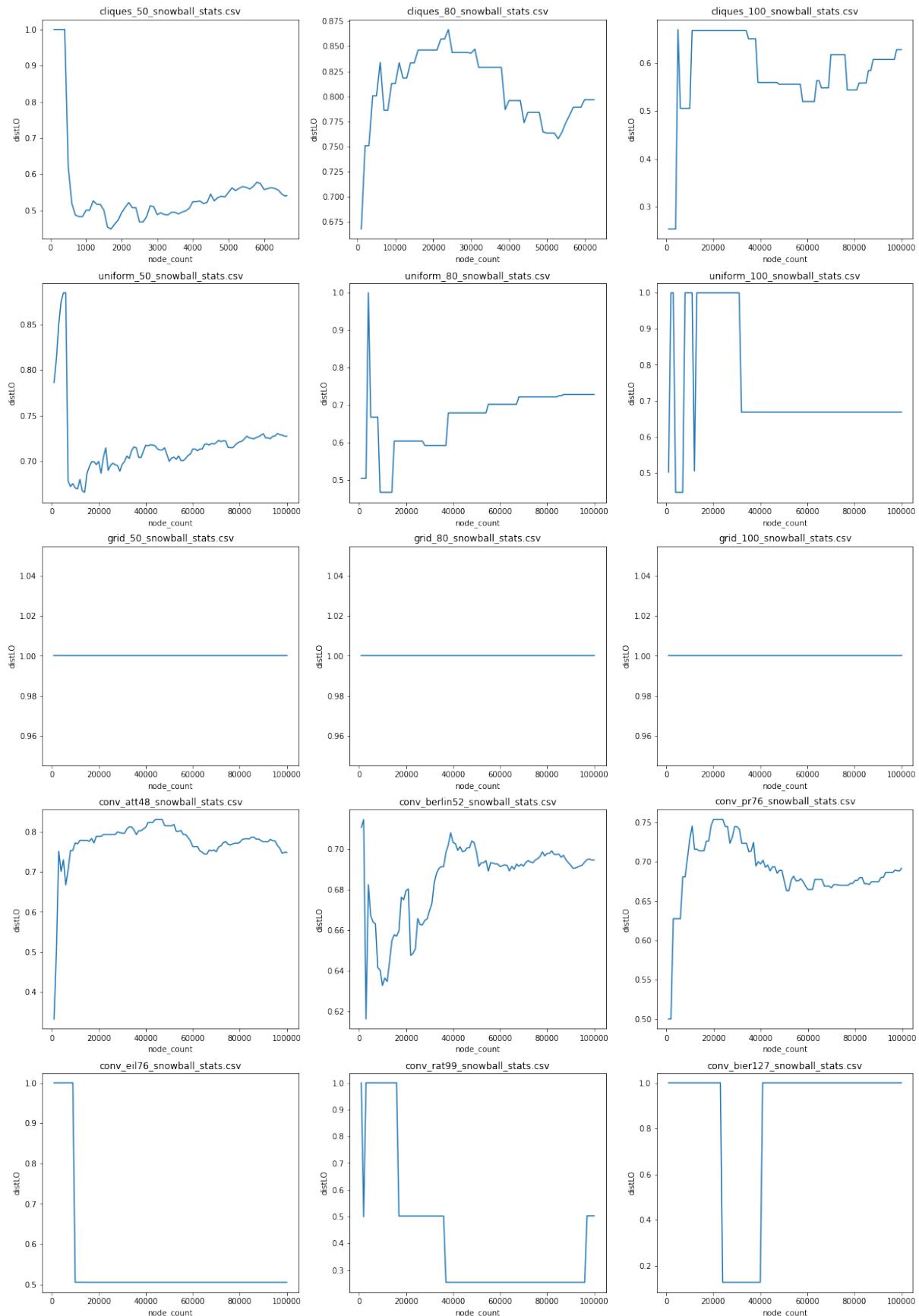
Rysunek A.31 Średnia wartość funkcji celu w znalezionych optimach lokalnych (avg_fitness) w zależności od liczby wierzchołków — próbkowanie snowball



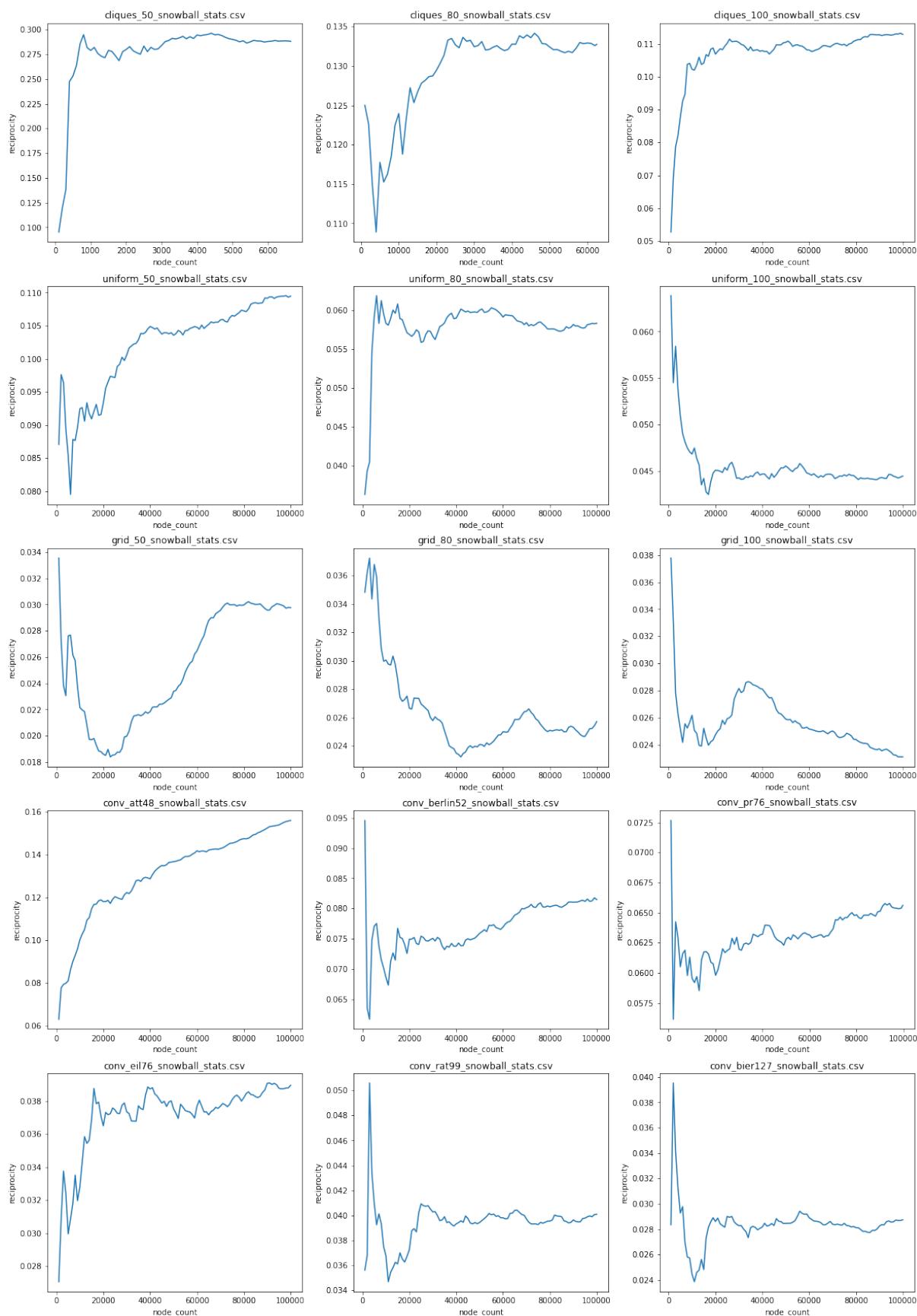
Rysunek A.32 Współczynnik conrel w zależności od liczby wierzchołków — próbkowanie snowball



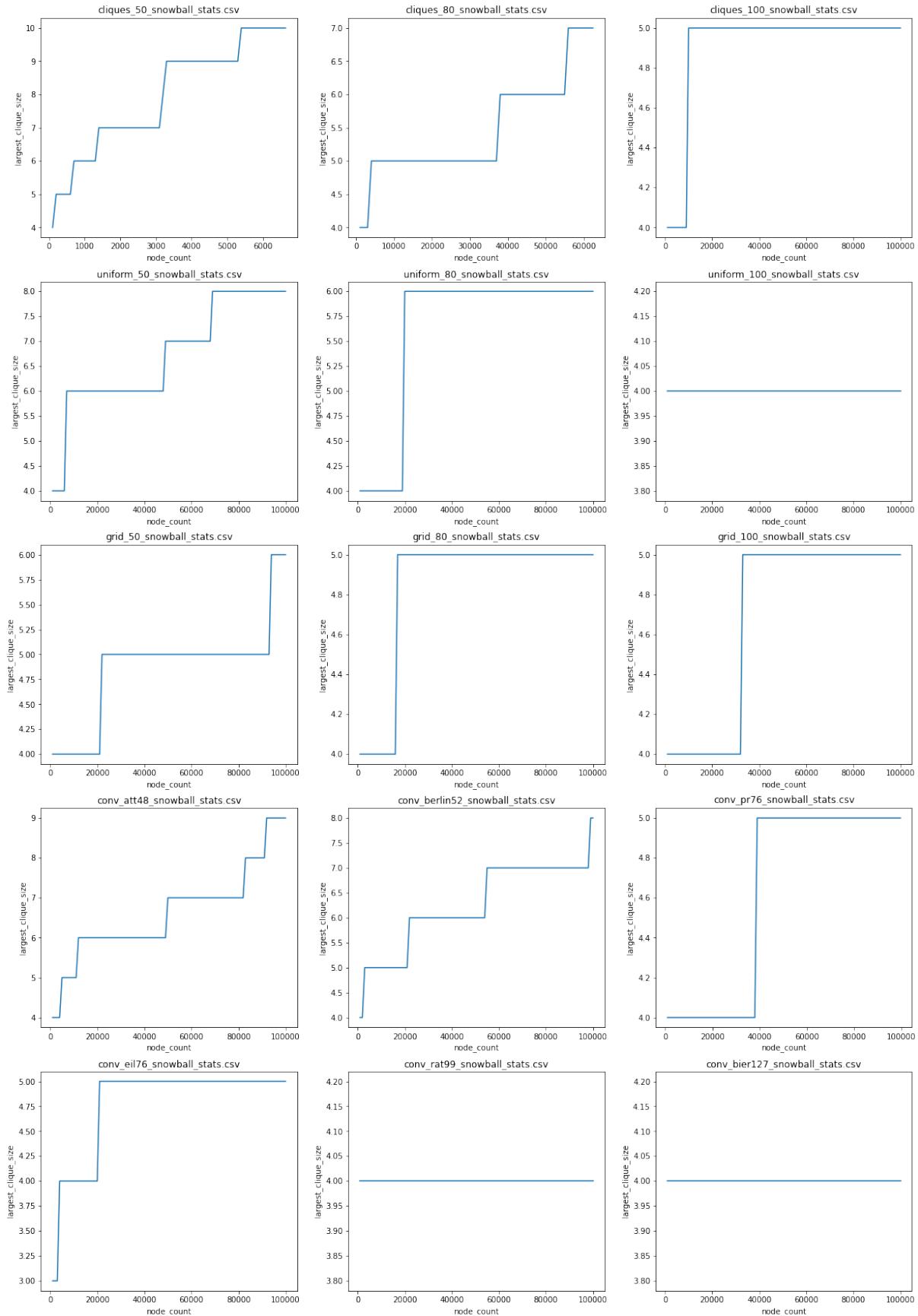
Rysunek A.33 Gęstość grafu (density) w zależności od liczby wierzchołków — próbkowanie snowball



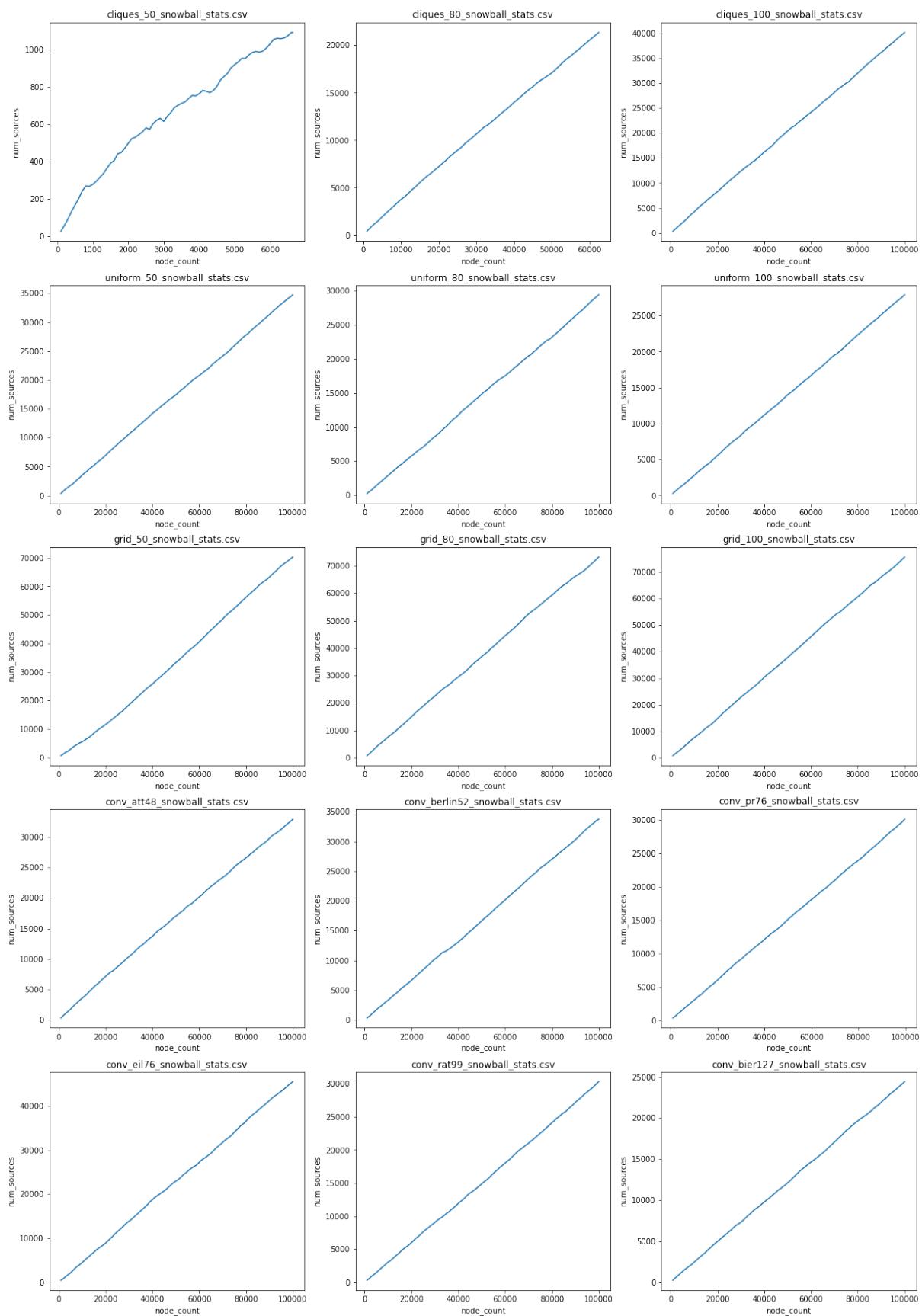
Rysunek A.34 Współczynnik distLO w zależności od liczby wierzchołków — próbkowanie snowball



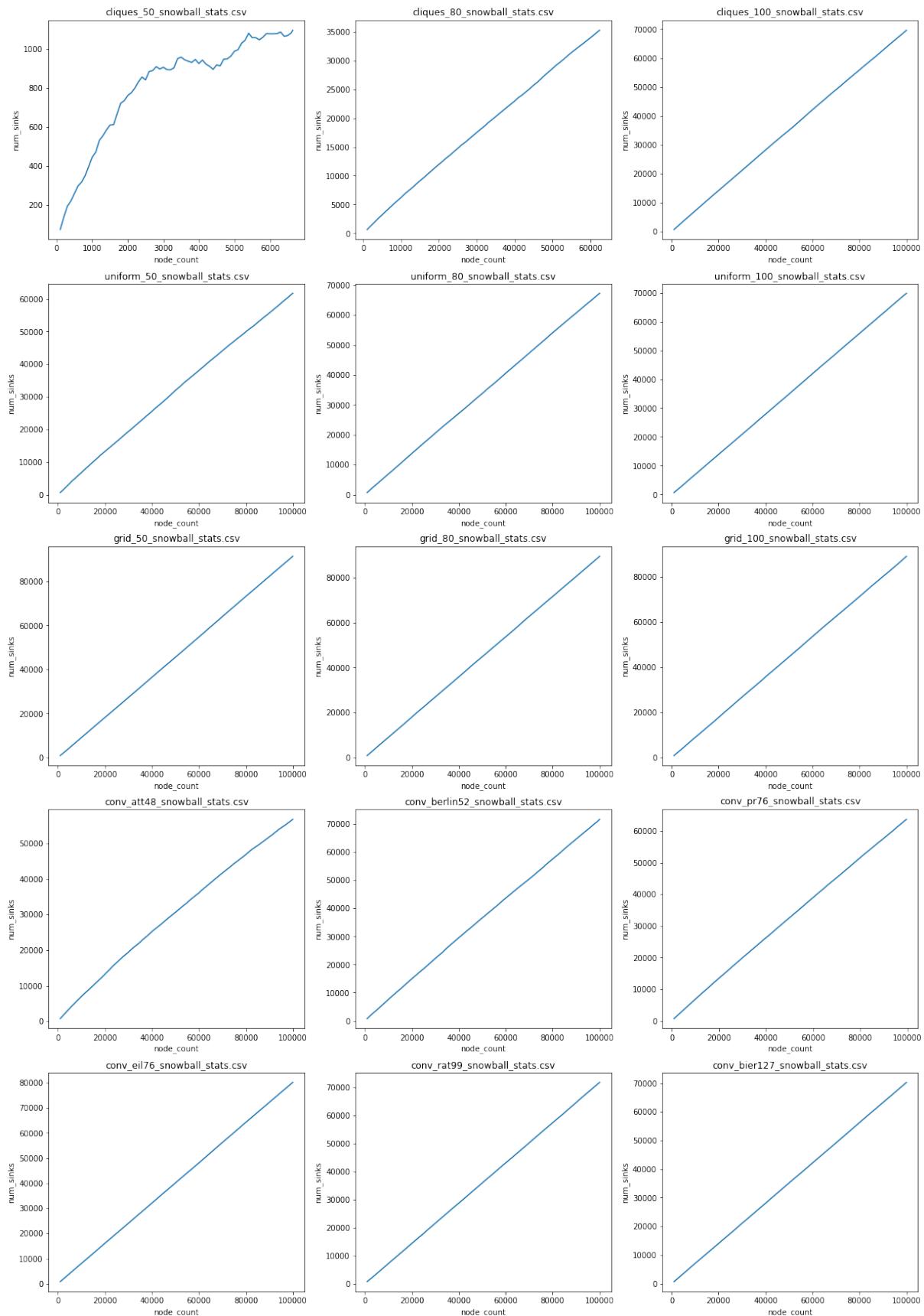
Rysunek A.35 Współczynnik wzajemności grafu (reciprocity) w zależności od liczby wierzchołków — próbkowanie snowball



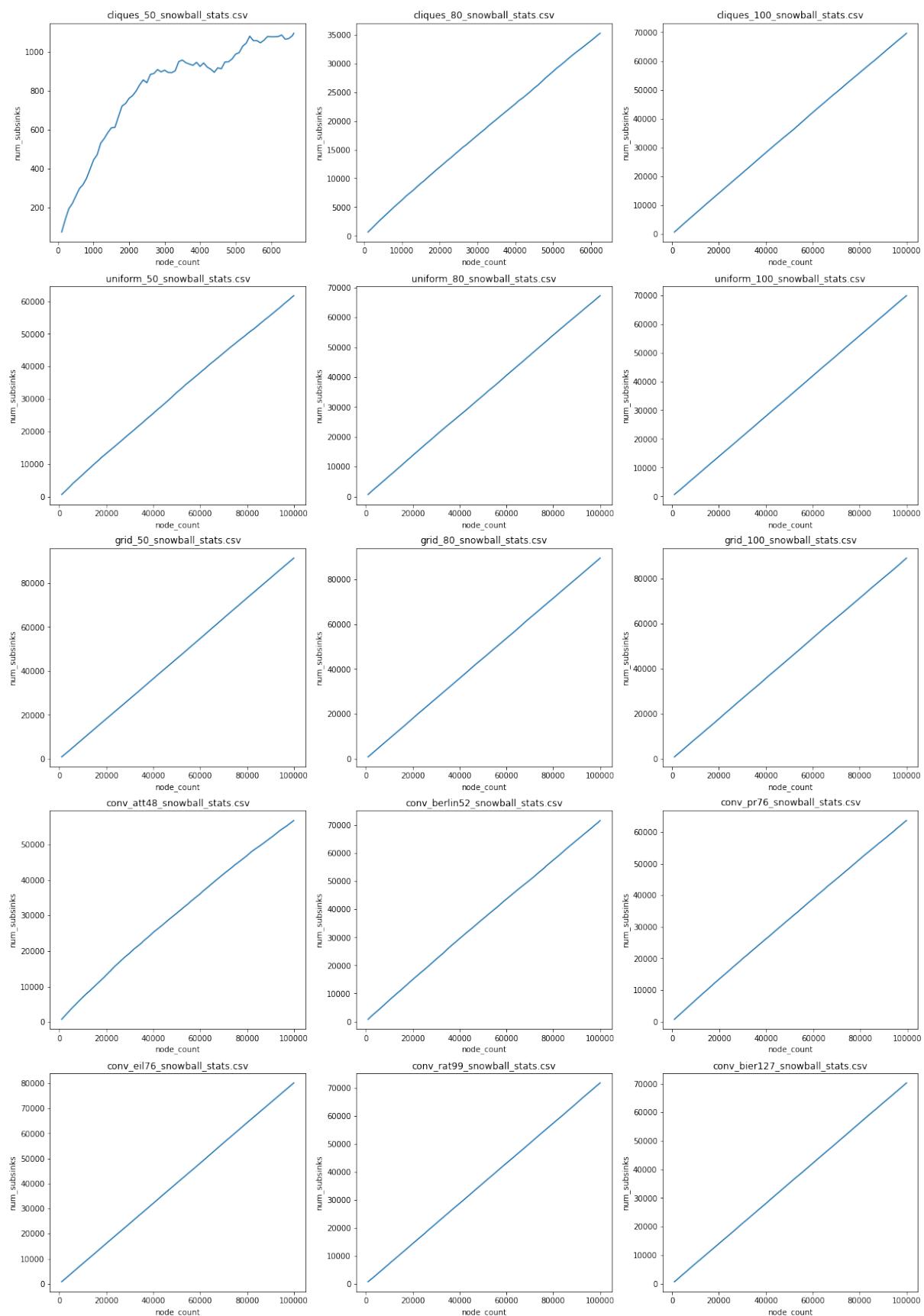
Rysunek A.36 Rozmiar największej kliki w grafie (largest_clique_size) w zależności od liczby wierzchołków — próbkowanie snowball



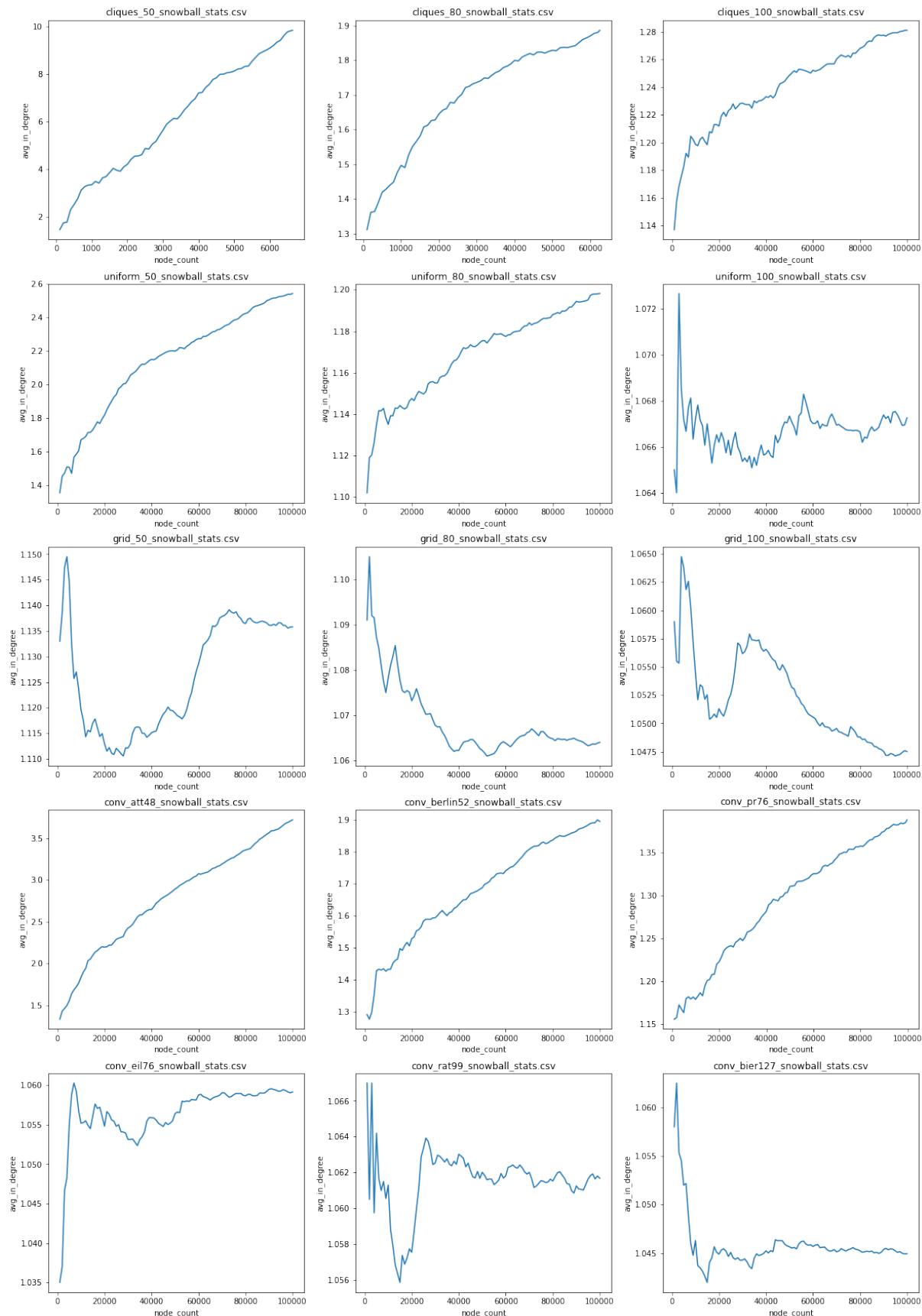
Rysunek A.37 Liczba źródeł w grafie (num_sources) w zależności od liczby wierzchołków — próbkowanie snowball



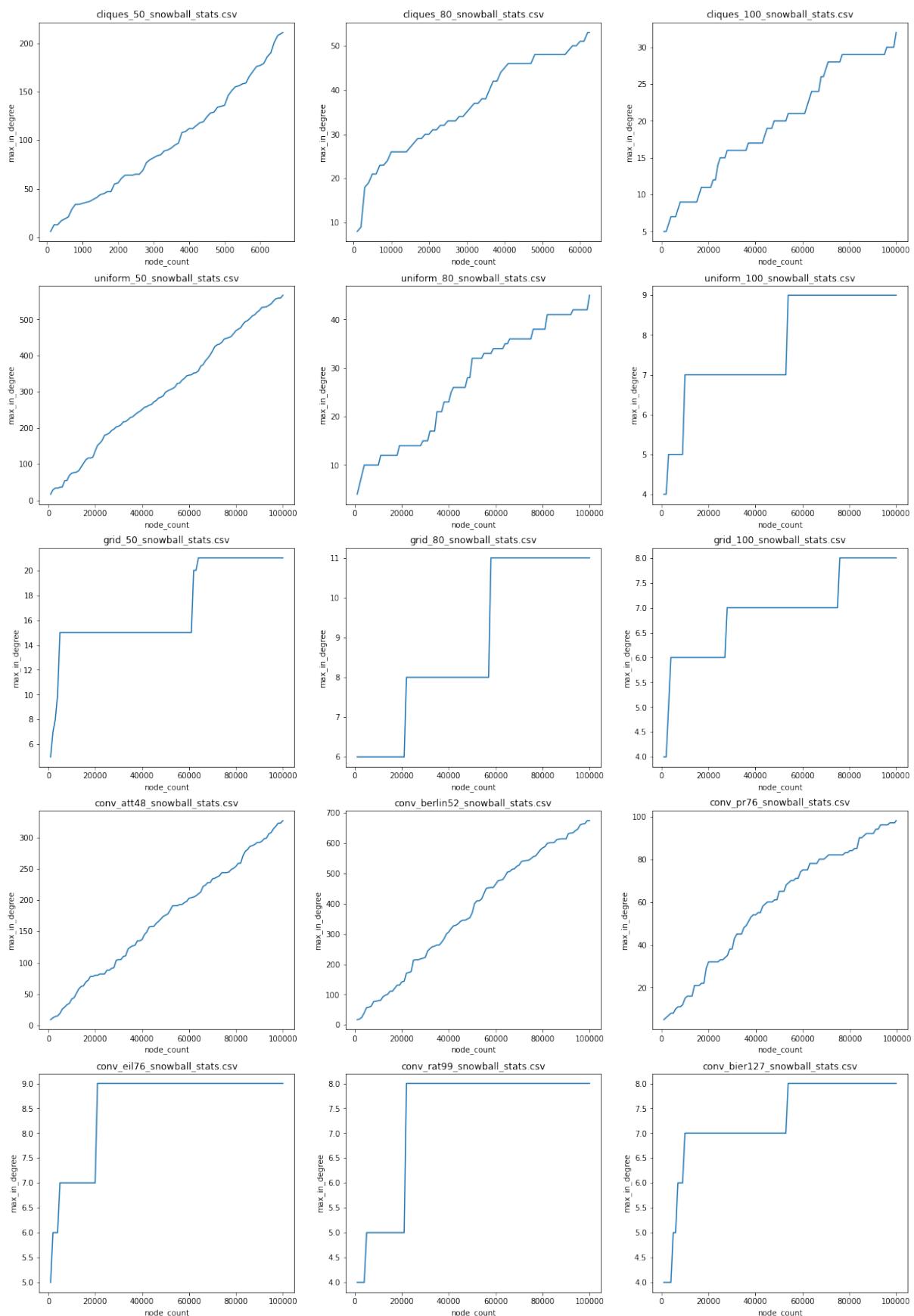
Rysunek A.38 Liczba ścieków w grafie (num_sinks) w zależności od liczby wierzchołków — próbkowanie snowball



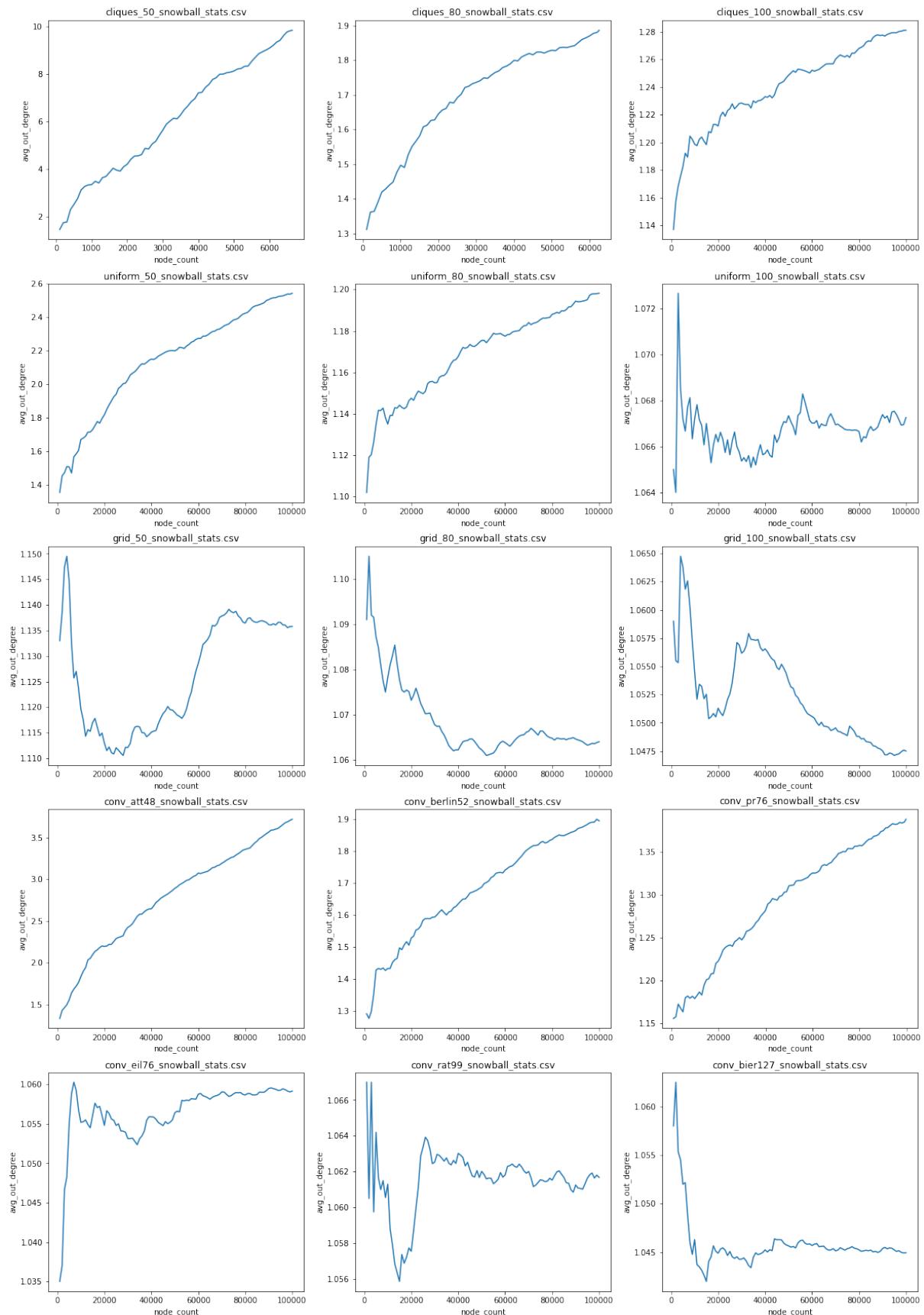
Rysunek A.39 Liczba subsinks w grafie (num_subsink) w zależności od liczby wierzchołków — próbkowanie snowball



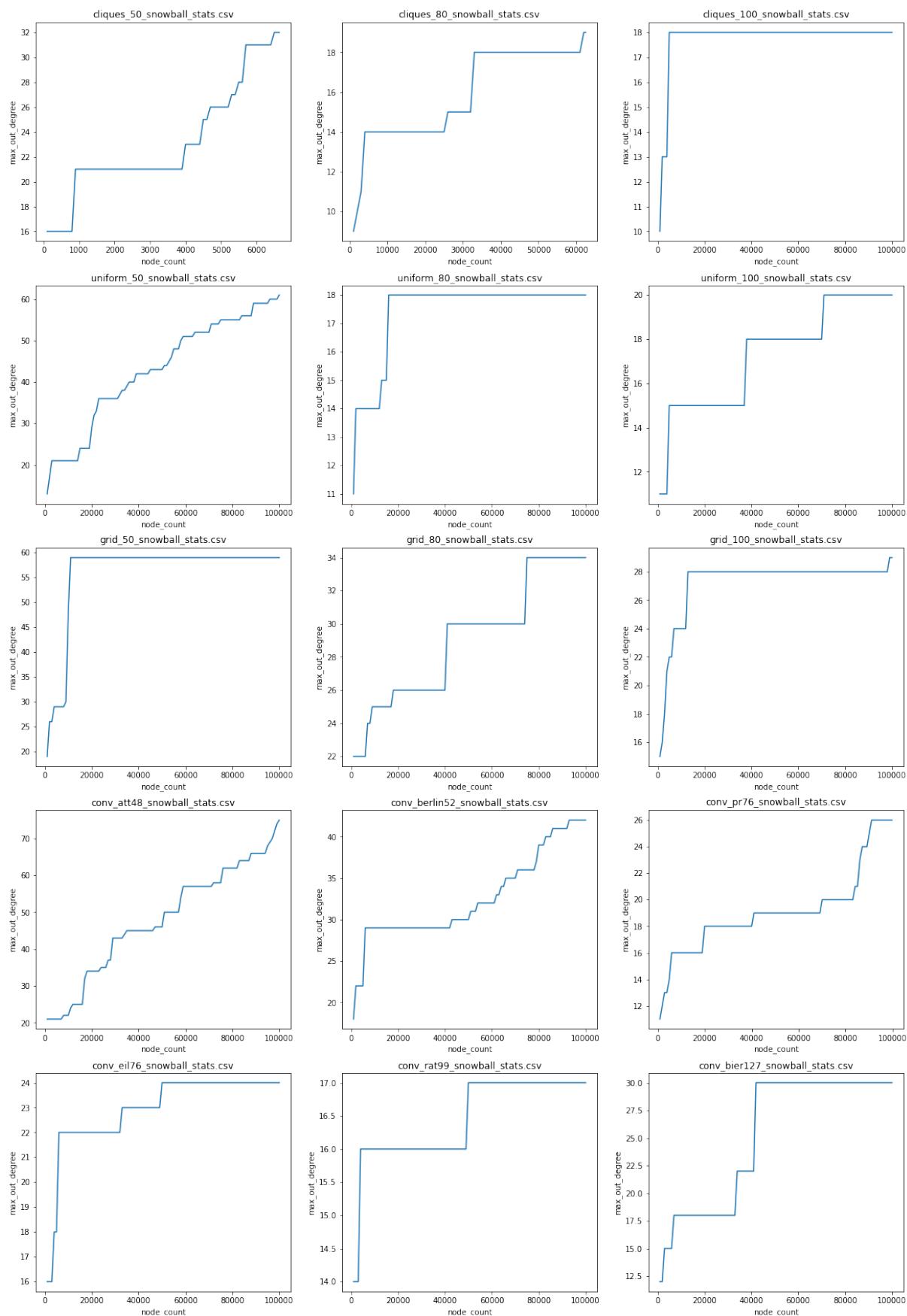
Rysunek A.40 Średni stopień wchodzący wierzchołków w grafie (avg_in_degree) w zależności od liczby wierzchołków — próbkowanie snowball



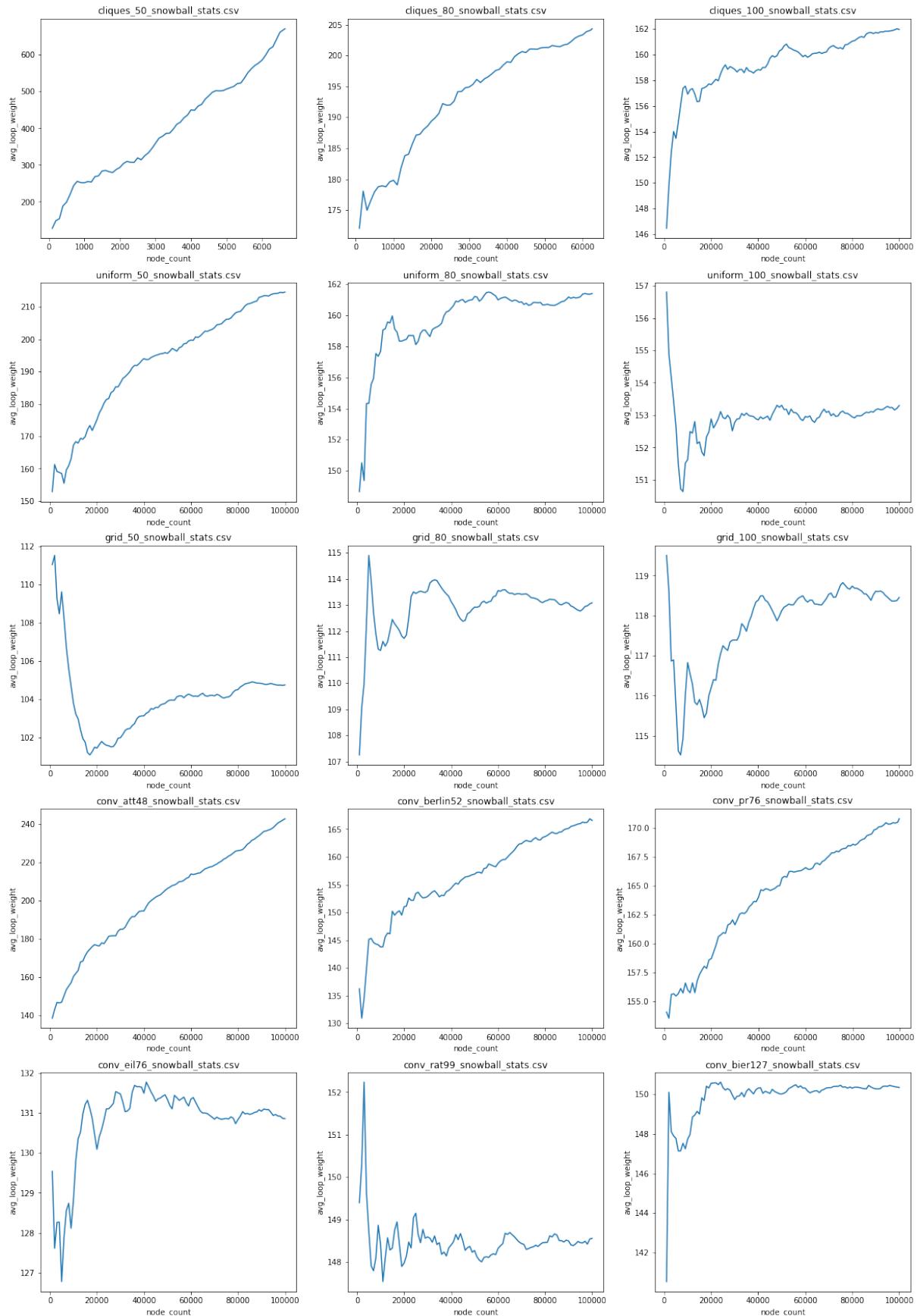
Rysunek A.41 Maksymalny stopień wchodzący wśród wierzchołków w grafie (max_in_degree) w zależności od liczby wierzchołków — próbkowanie snowball



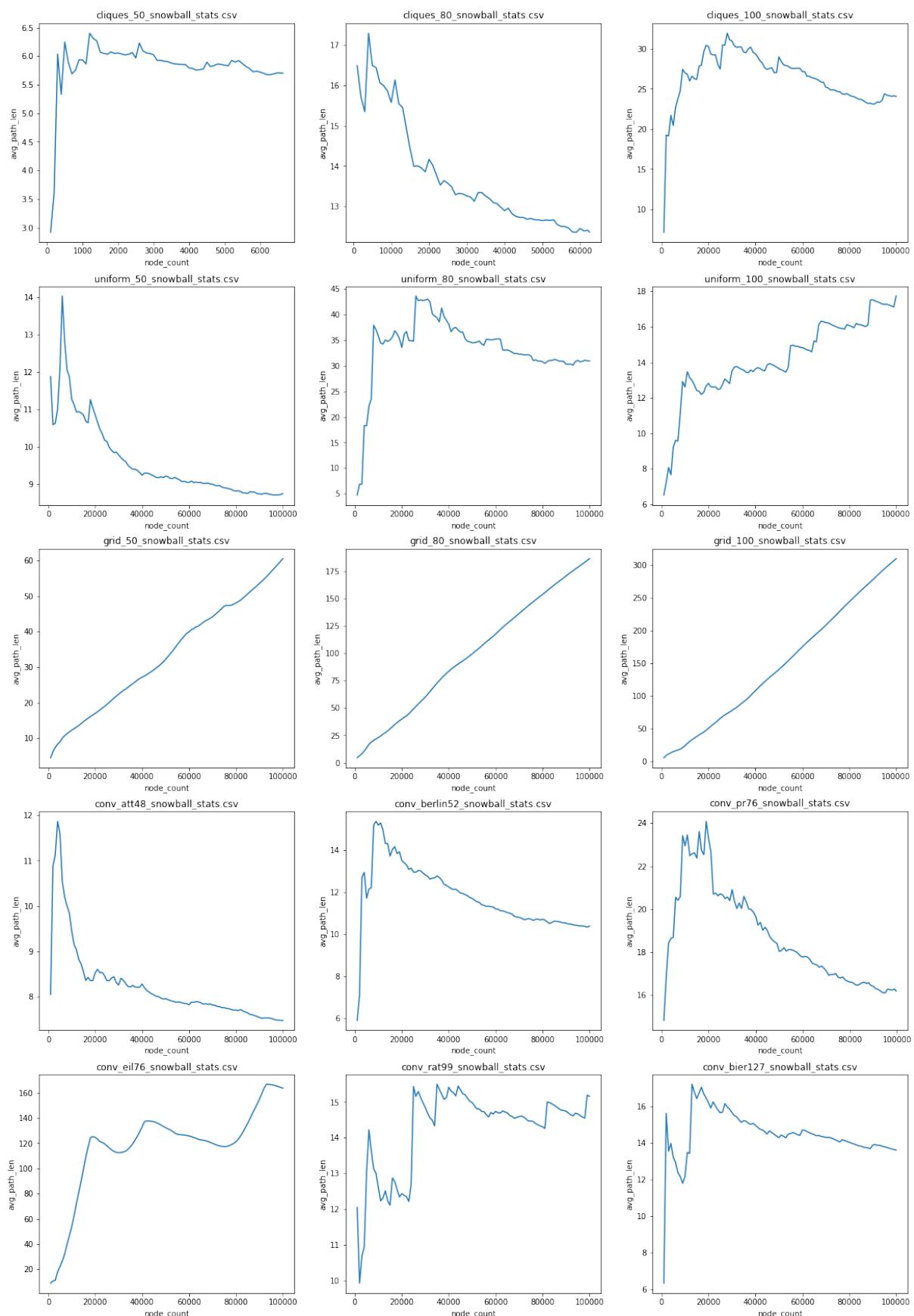
Rysunek A.42 Średni stopień wychodzący wśród wierzchołków w grafie (avg_out_degree) w zależności od liczby wierzchołków — próbkowanie snowball



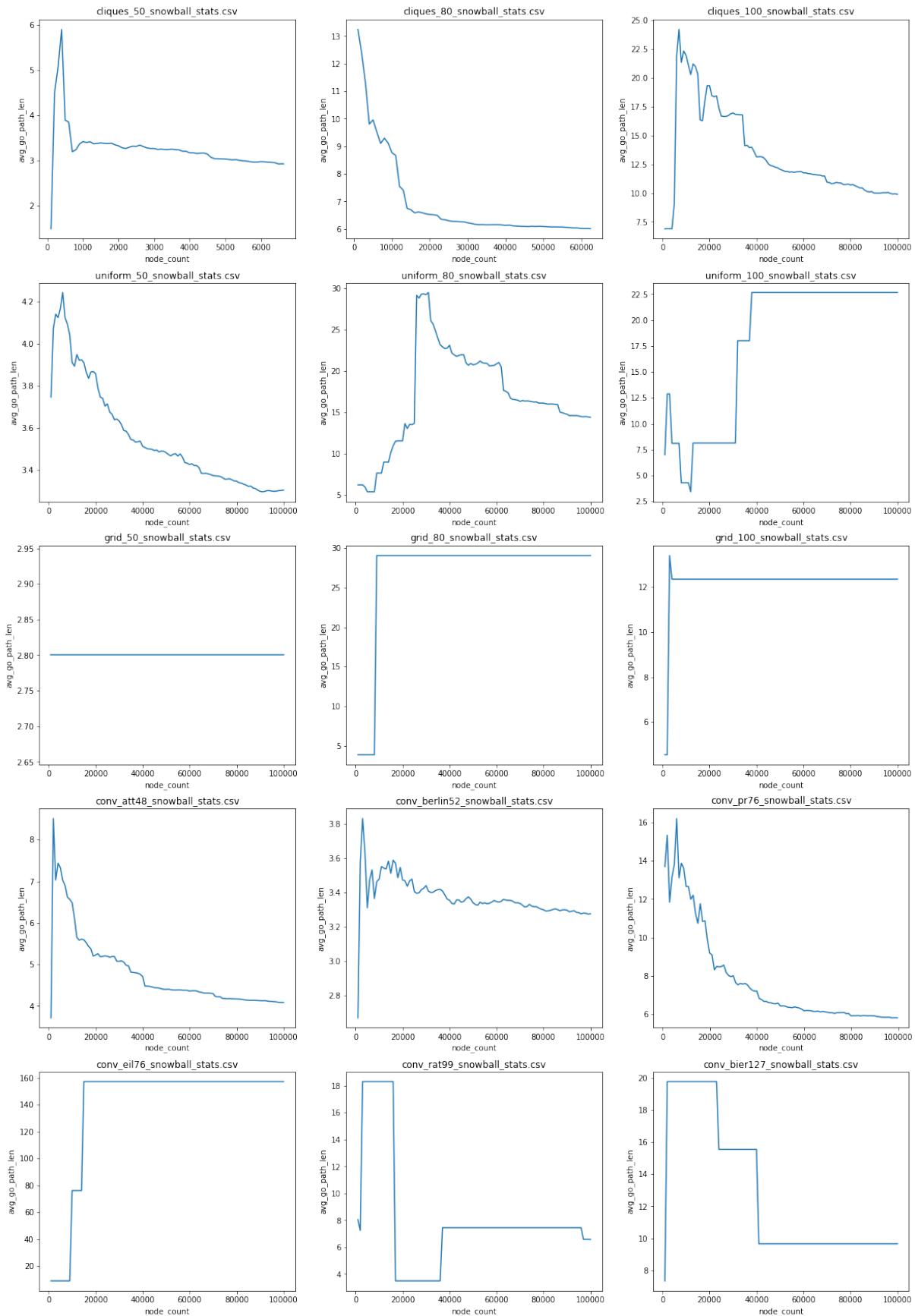
Rysunek A.43 Maksymalny stopień wychodzący wierzchołków w grafie (max_out_degree) w zależności od liczby wierzchołków — próbkowanie snowball



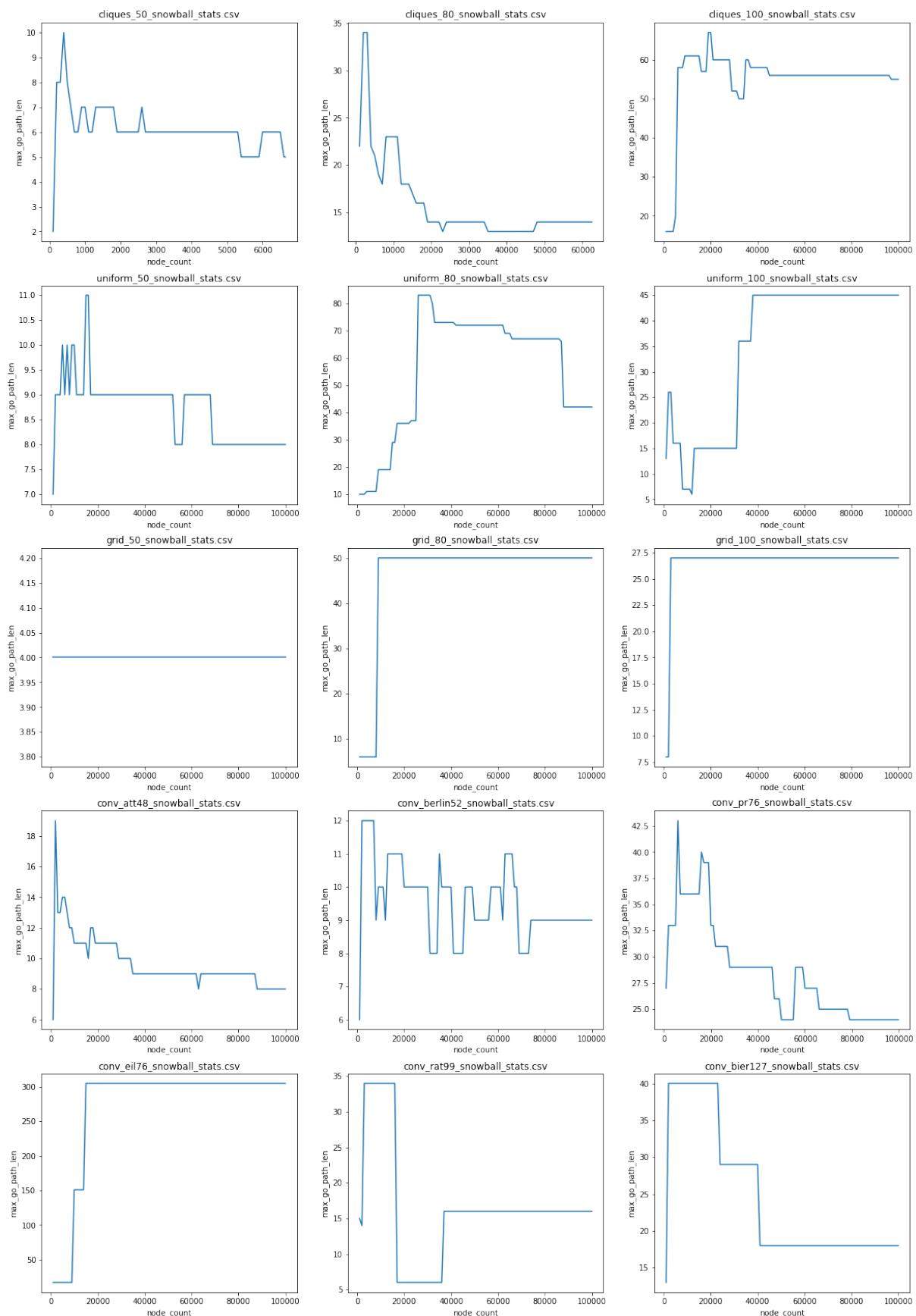
Rysunek A.44 Średnia waga pętli w grafie (avg_loop_weight) w zależności od liczby wierzchołków — próbkowanie snowball



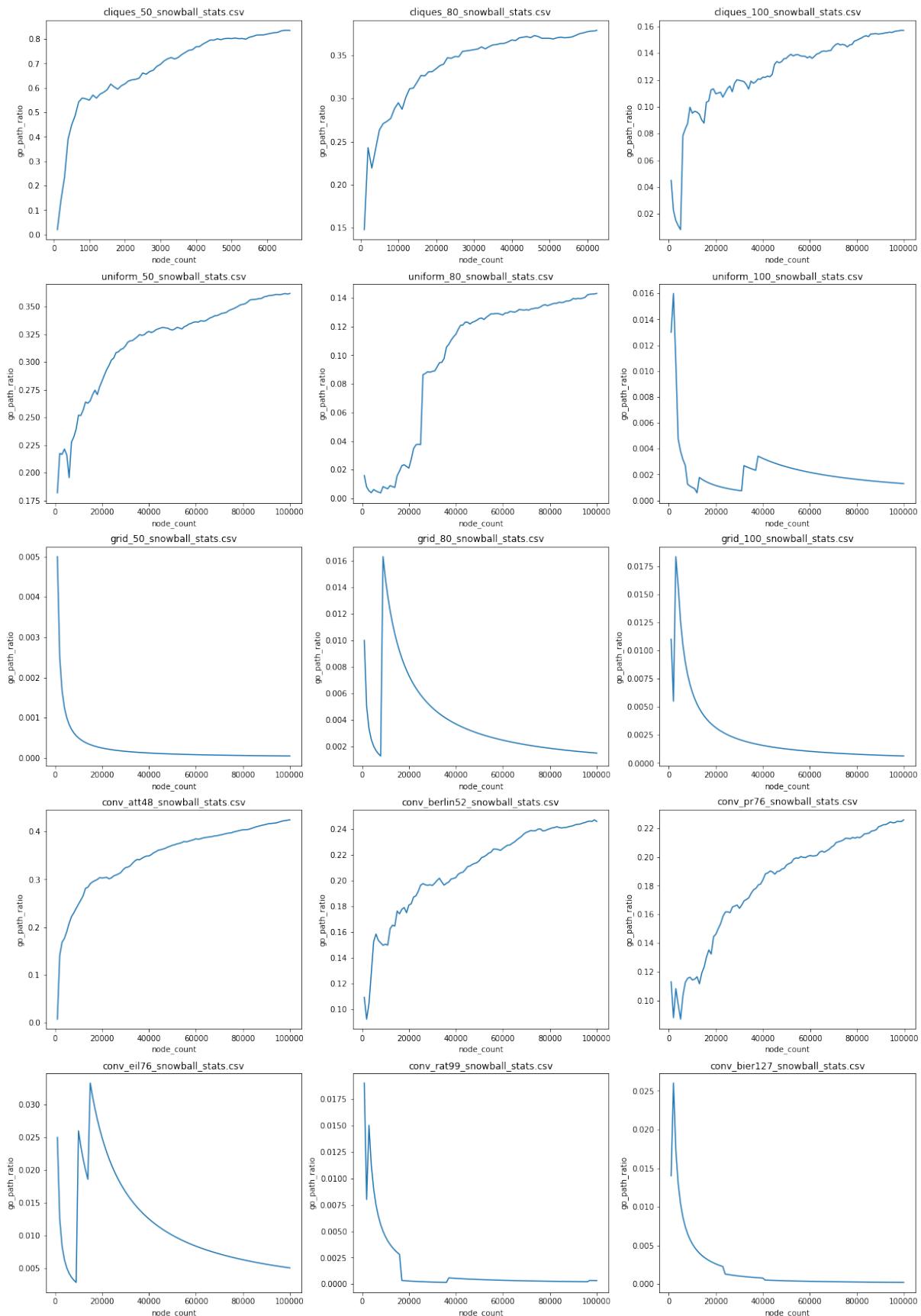
Rysunek A.45 Średnia długość ścieżki w grafie (avg_path_len) w zależności od liczby wierzchołków — próbkowanie snowball



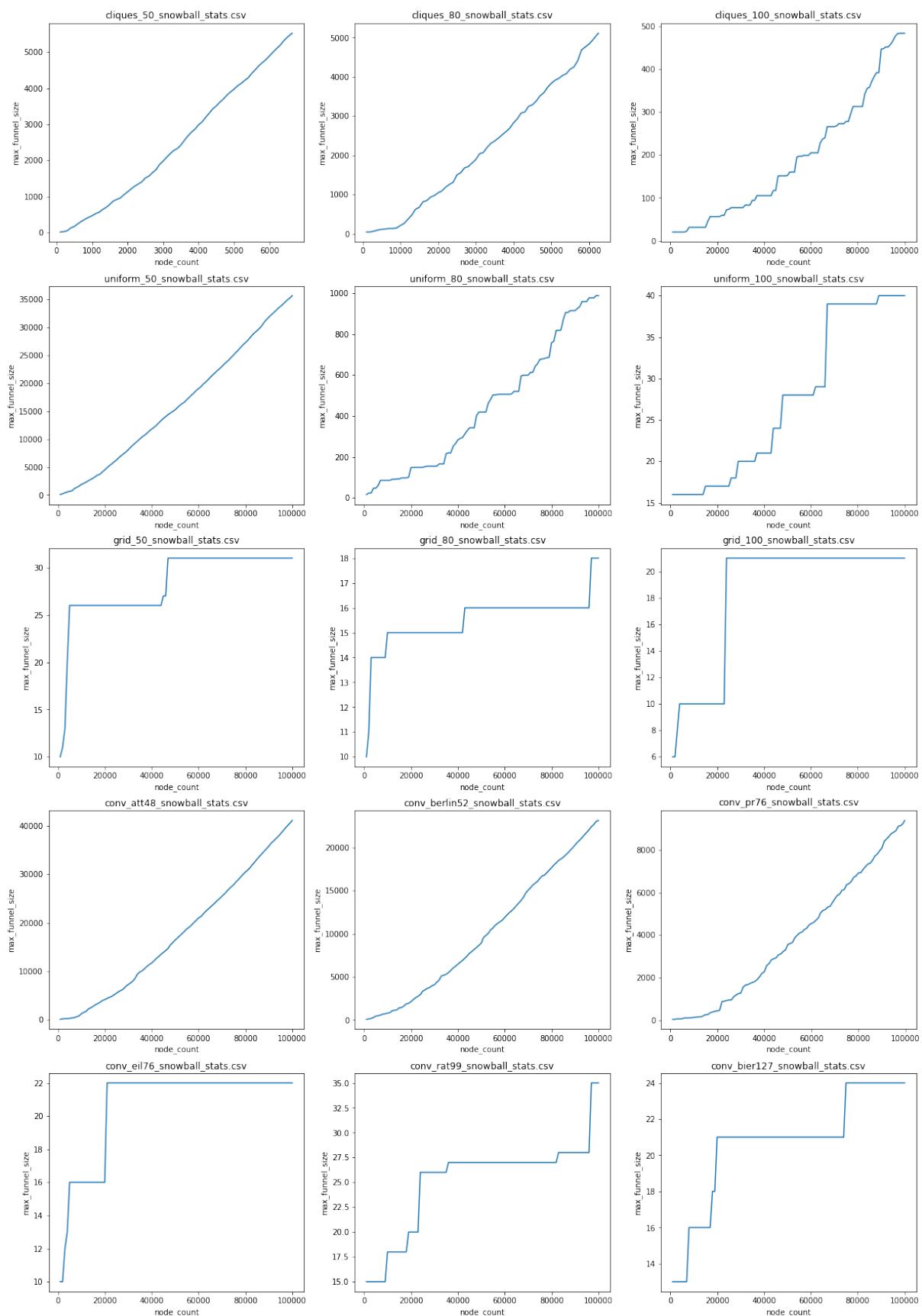
Rysunek A.46 Średnia długość istniejących ścieżek do globalnego optimum (avg_go_path_len) w zależności od liczby wierzchołków — próbkowanie snowball



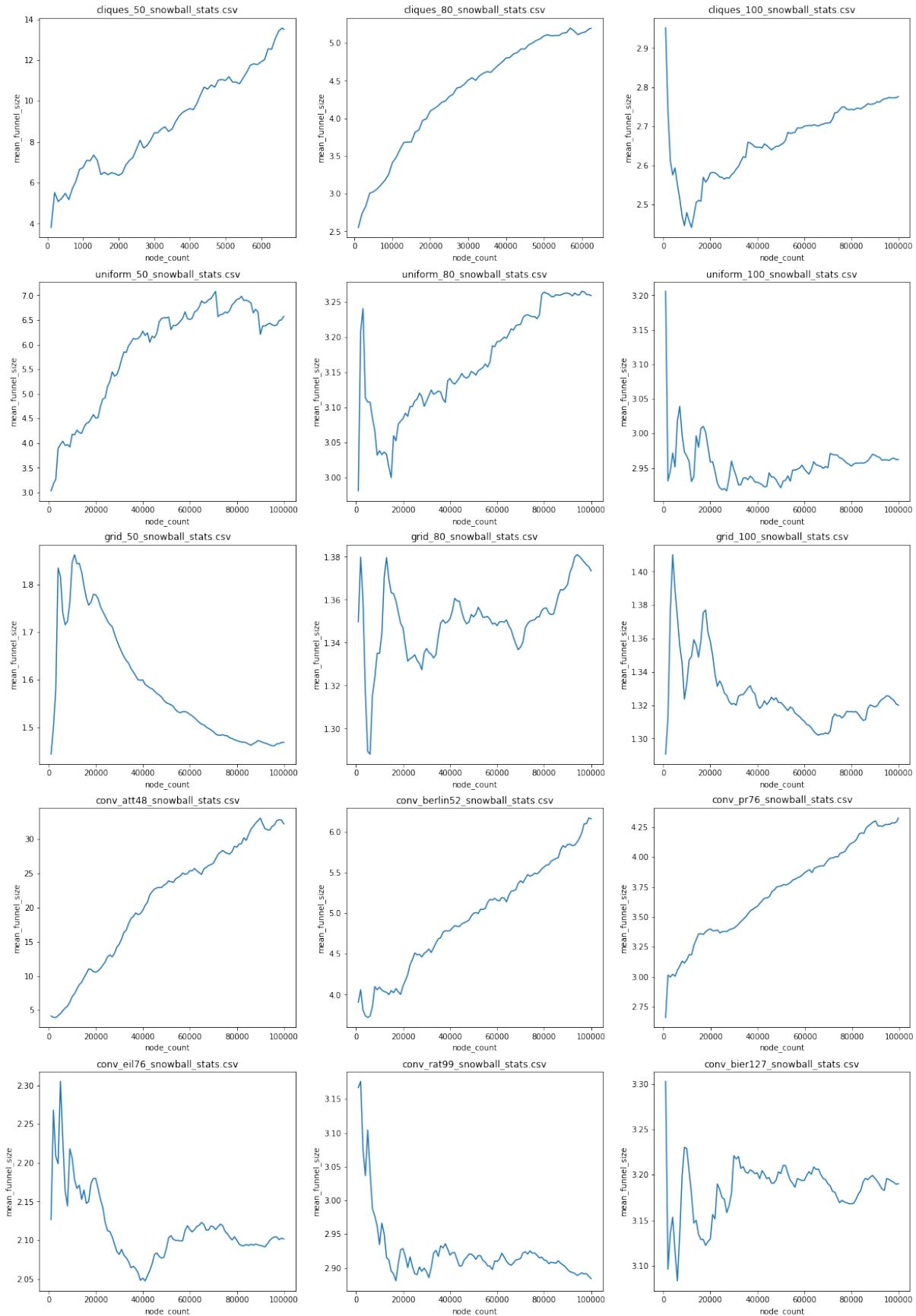
Rysunek A.47 Długość najdłuższej istniejącej ścieżki do globalnego optimum (max_go_path_len) w zależności od liczby wierzchołków — próbkowanie snowball



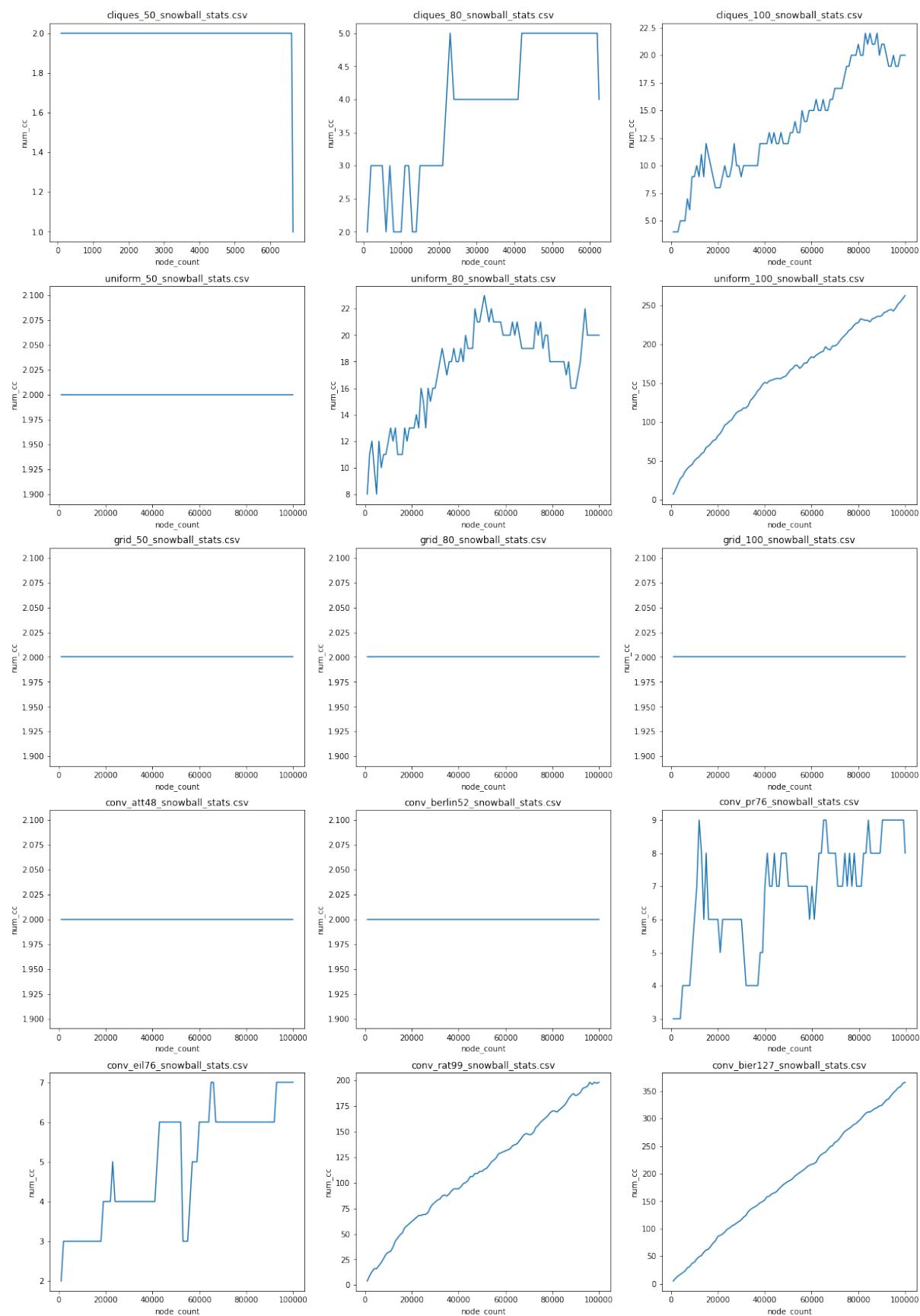
Rysunek A.48 Stosunek liczby wierzchołków, z których istnieje ścieżka do globalnego optimum do liczby wszystkich wierzchołków (go_path_ratio) w zależności od liczby wierzchołków — próbkowanie snowball



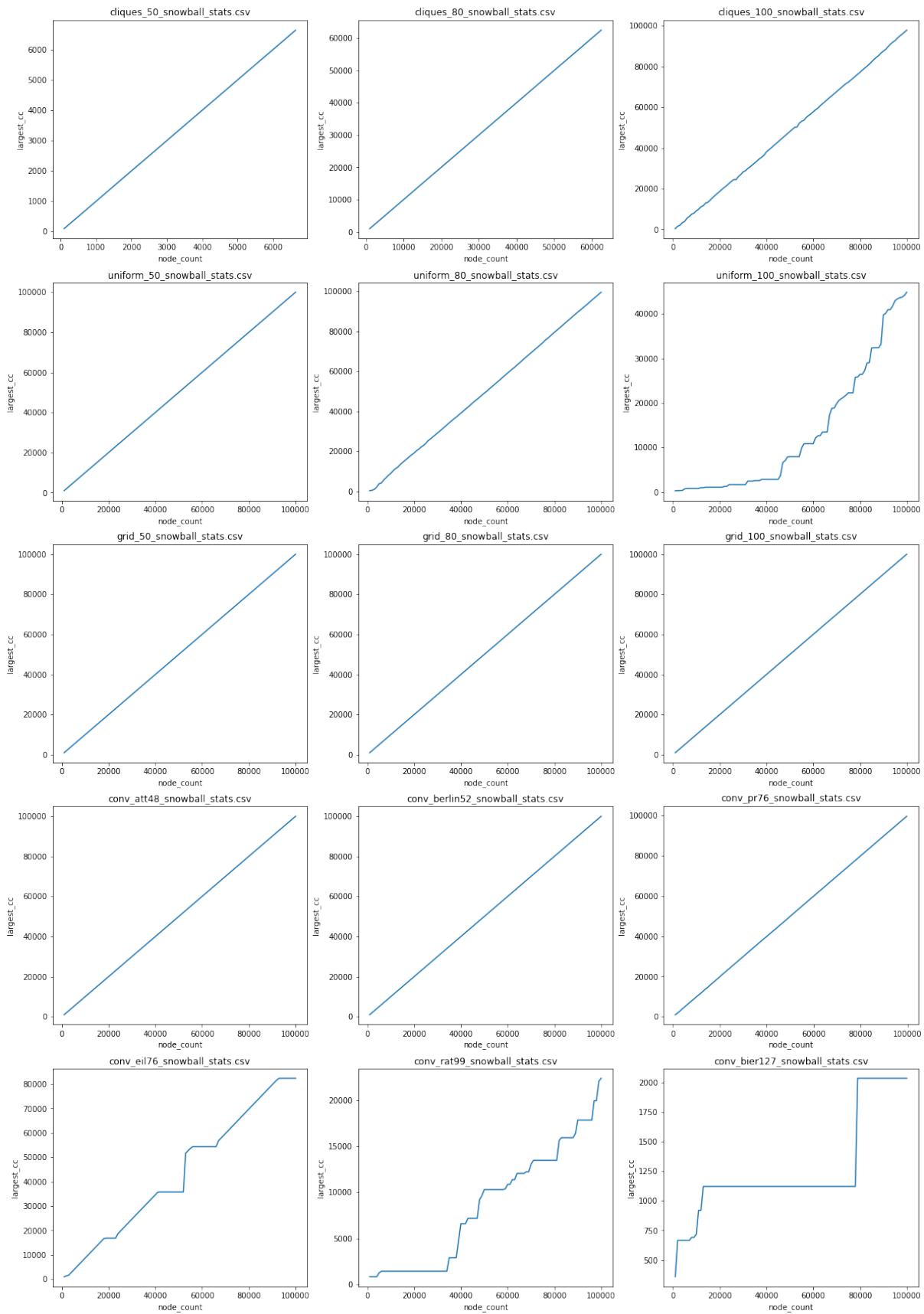
Rysunek A.49 Rozmiar największego leja w przestrzeni (max_funnel_size) w zależności od liczby wierzchołków — próbkowanie snowball



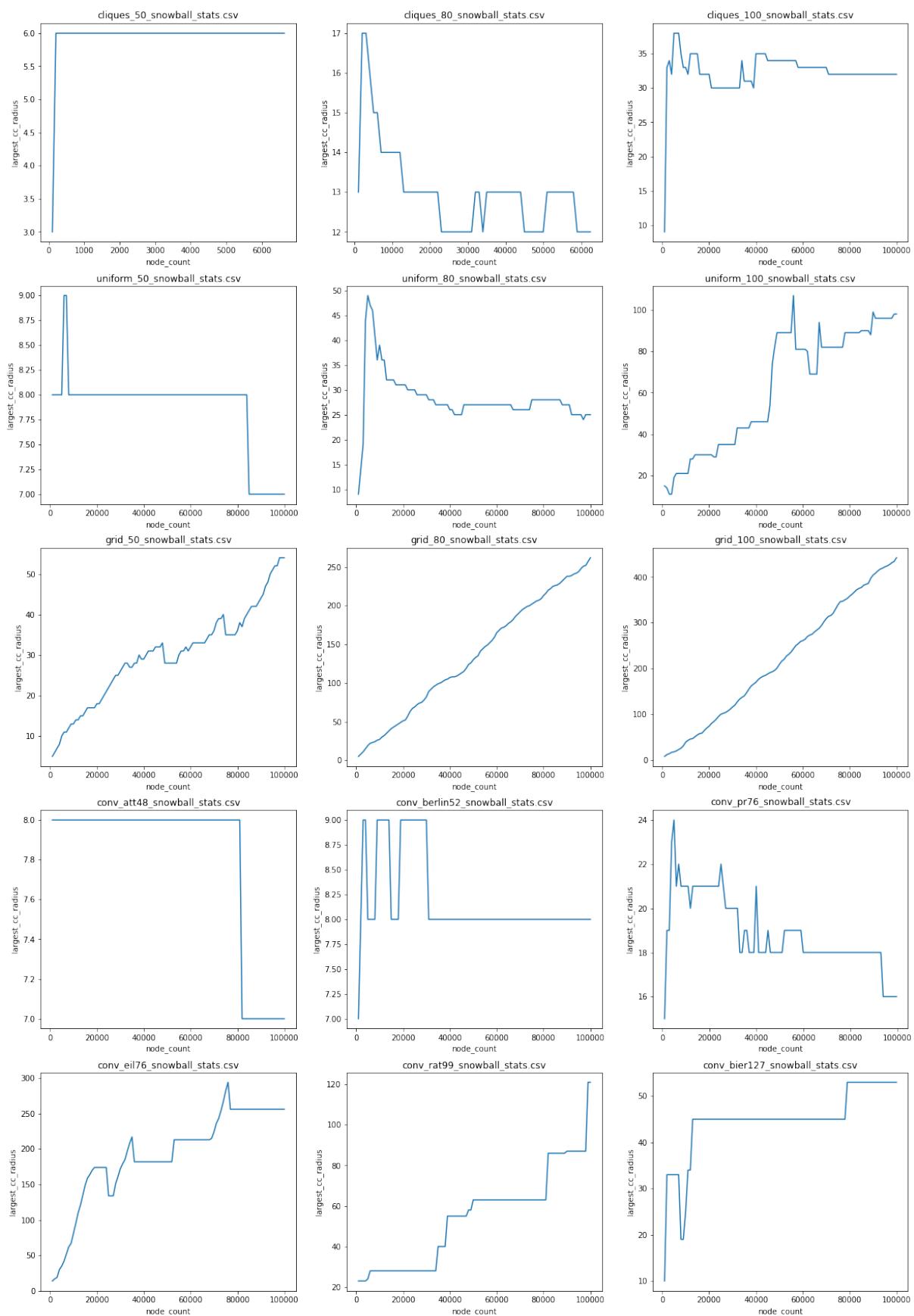
Rysunek A.50 Średni rozmiar lejów w przestrzeni (mean_funnel_size) w zależności od liczby wierzchołków — próbkowanie snowball



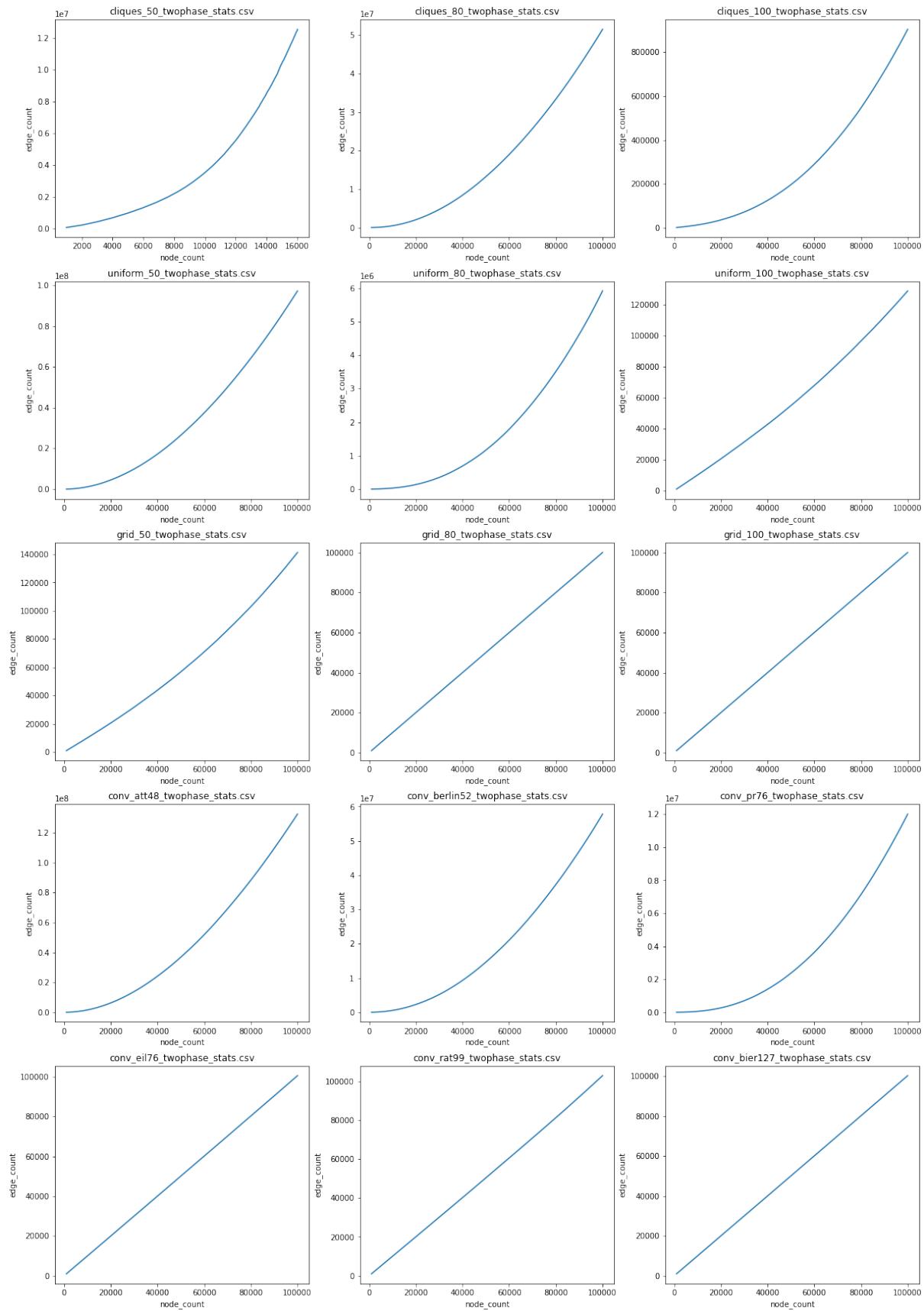
Rysunek A.51 Liczba spójnych podgrafów (num_cc) w zależności od liczby wierzchołków — próbkowanie snowball



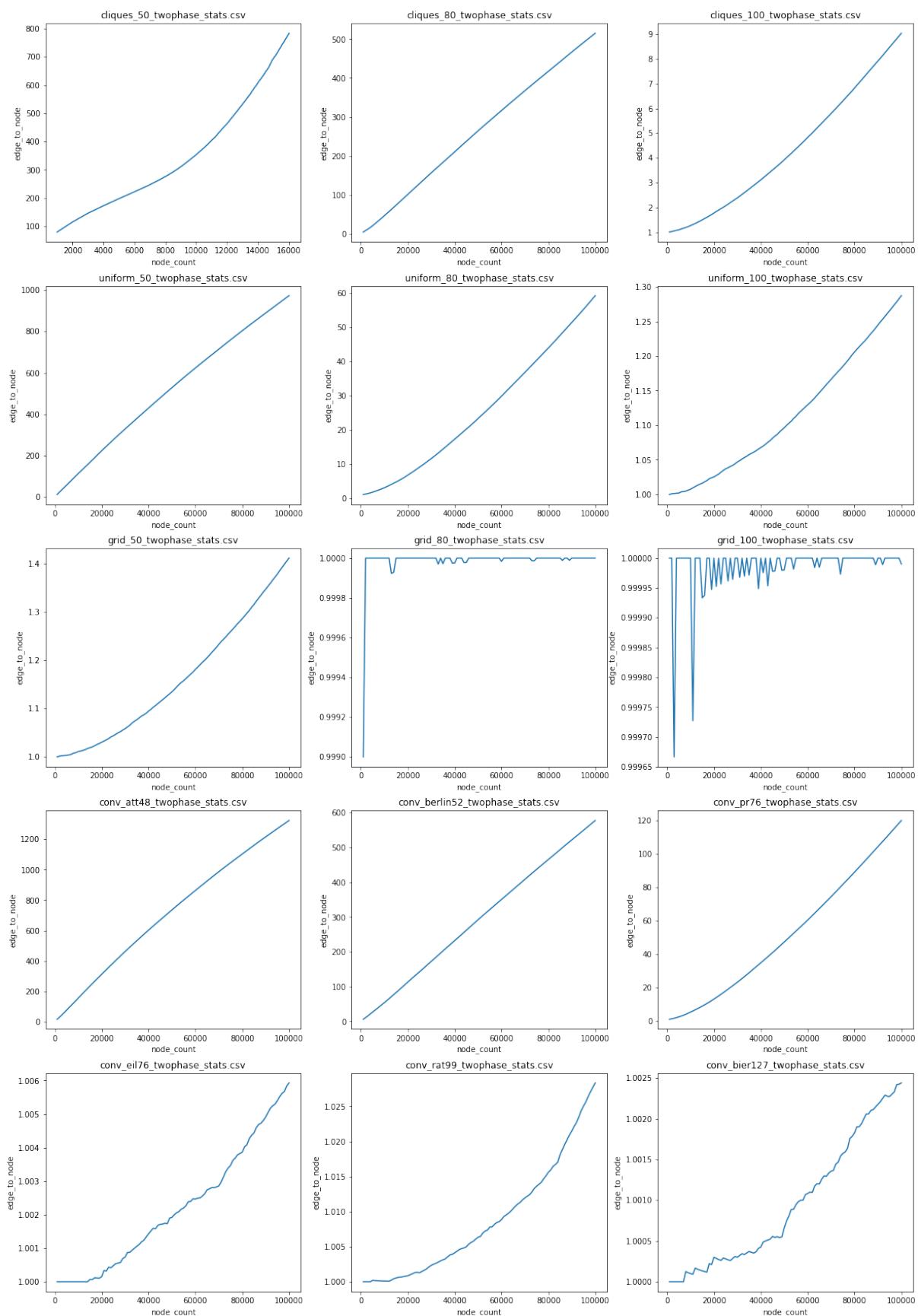
Rysunek A.52 Rozmiar największego spójnego podgrafa (largest_cc) w zależności od liczby wierzchołków — próbkowanie snowball



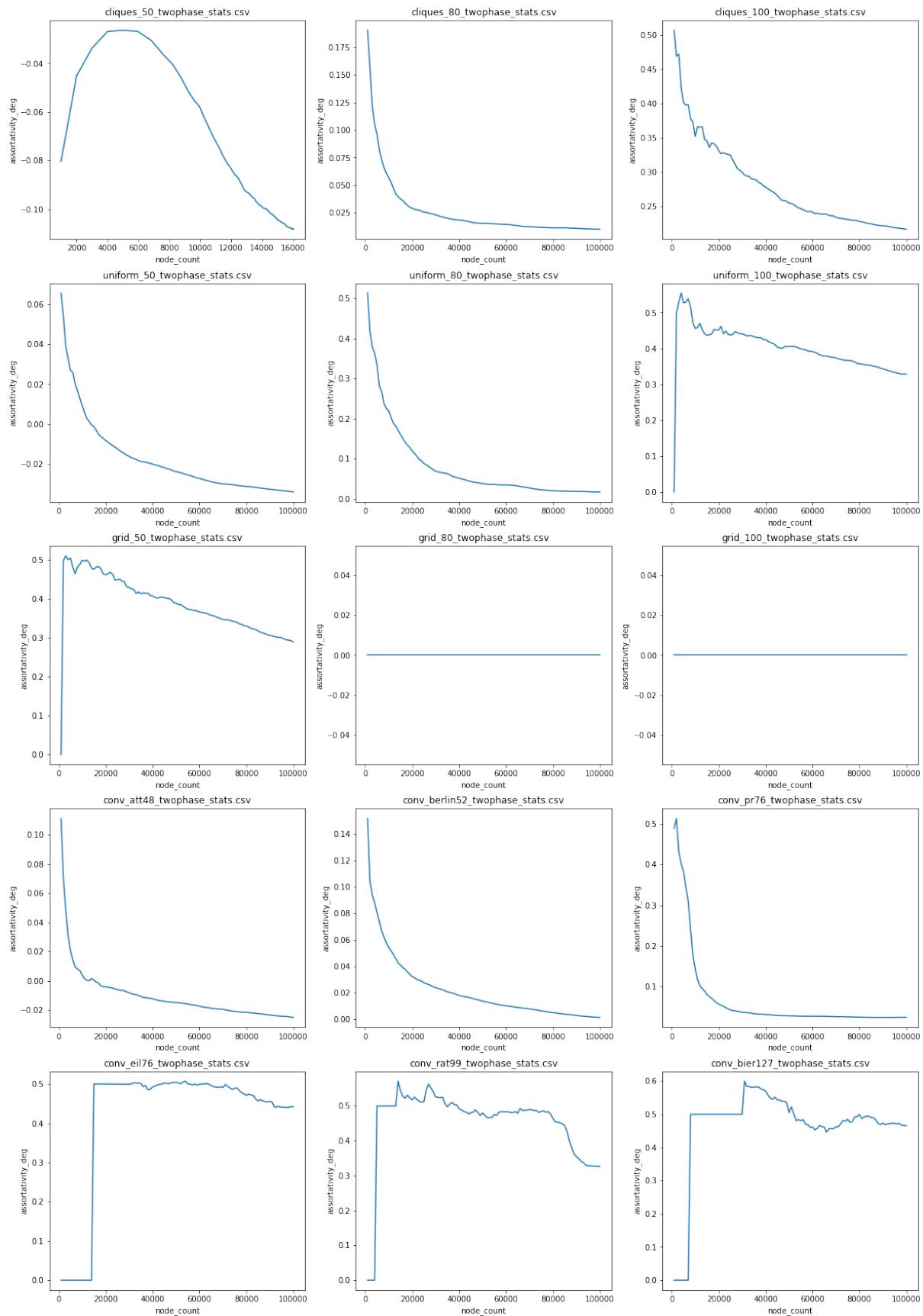
Rysunek A.53 Promień największego spójnego podgrafu (largest_cc_radius) w zależności od liczby wierzchołków — próbkowanie snowball



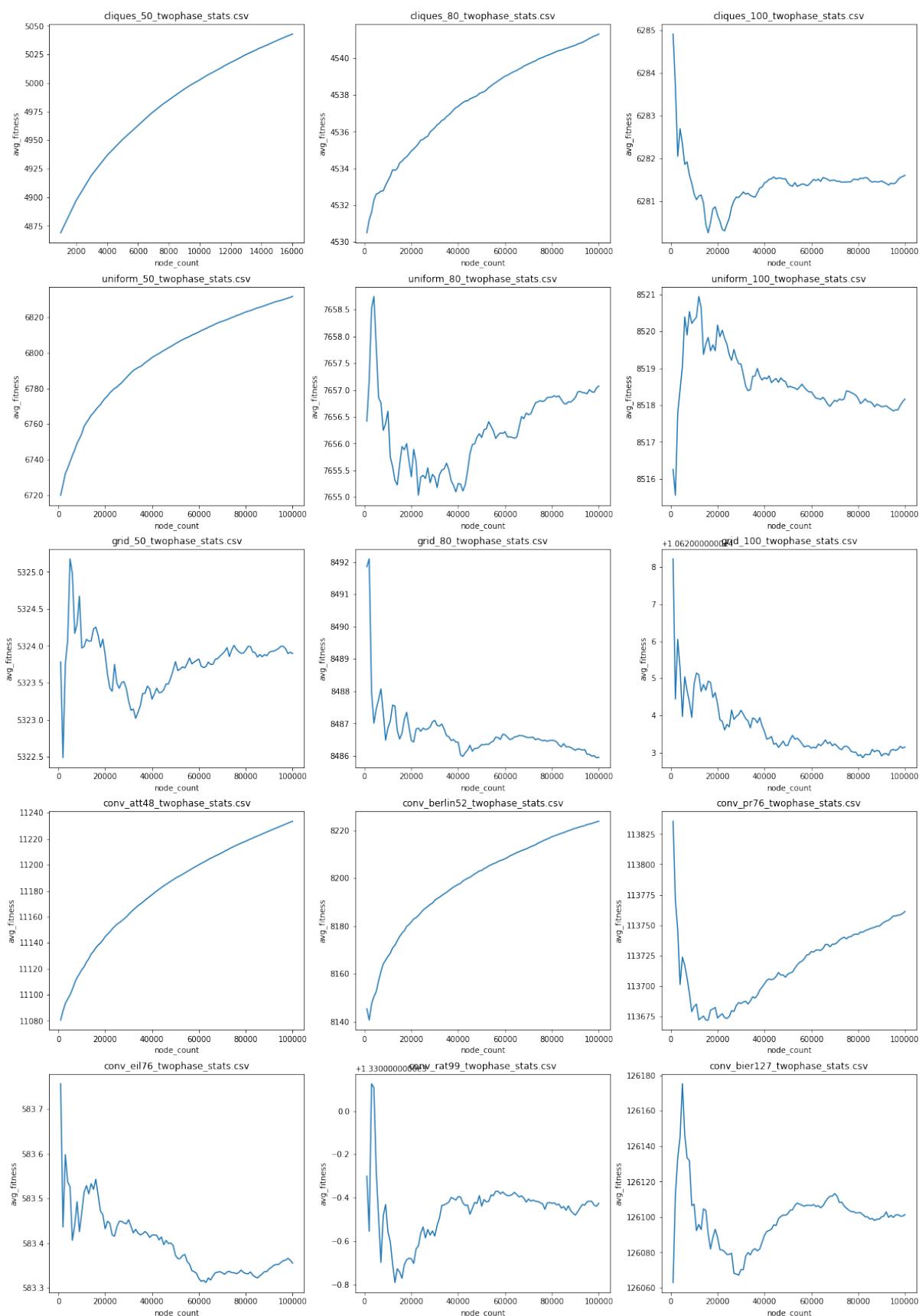
Rysunek A.54 Liczba krawędzi (edge_count) w zależności od liczby wierzchołków — próbkowanie dwufazowe



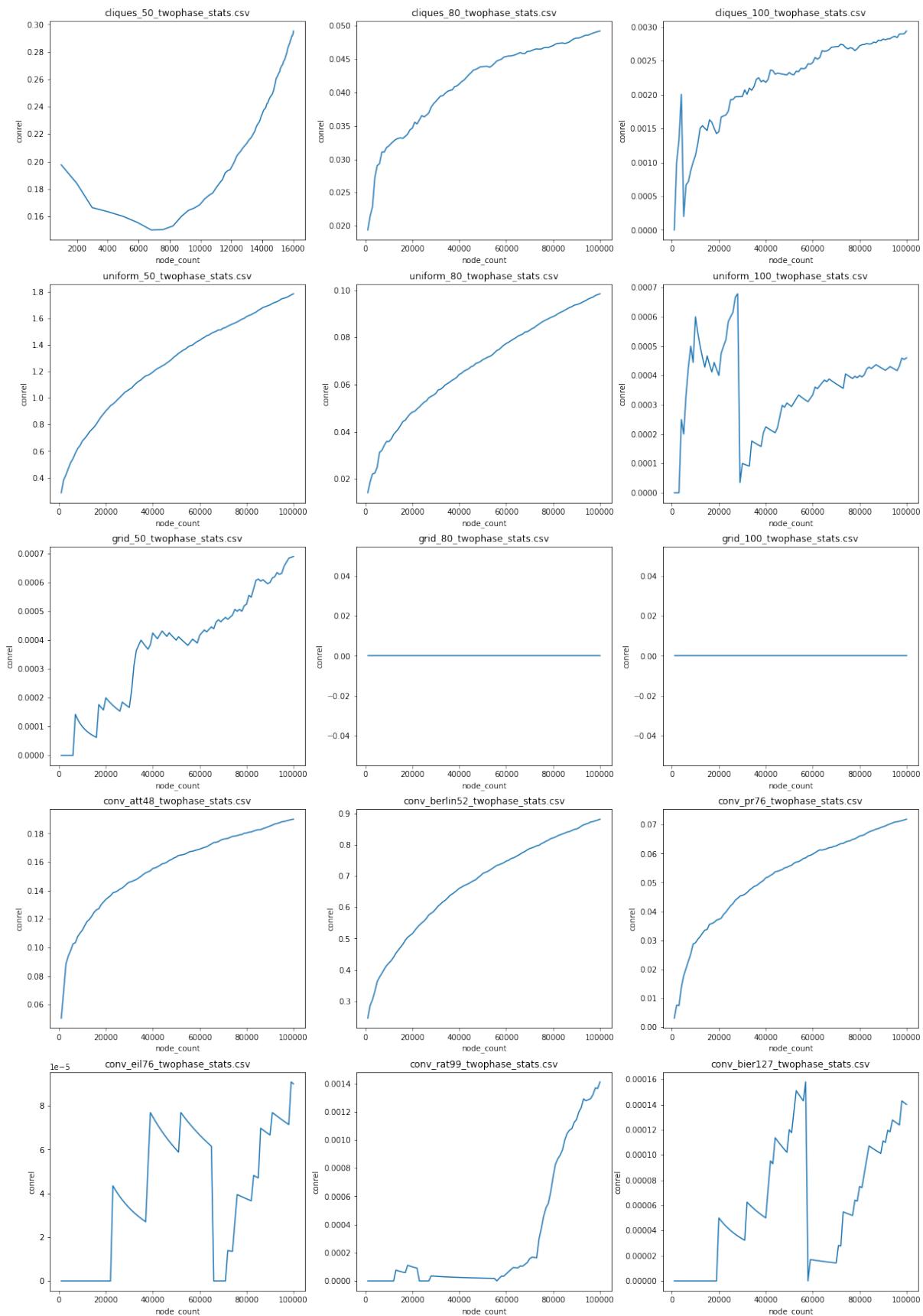
Rysunek A.55 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) w zależności od liczby wierzchołków — próbkowanie dwufazowe



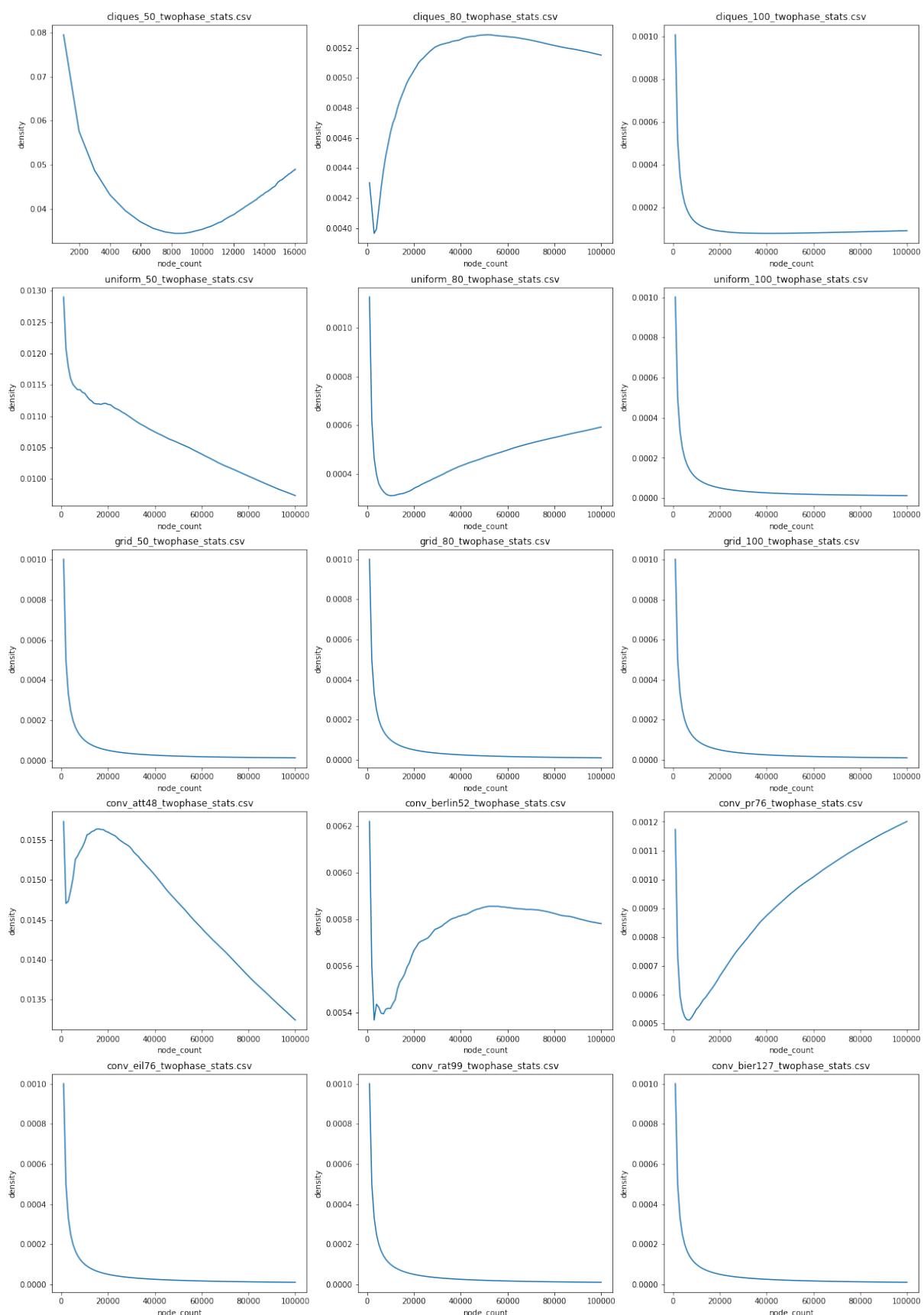
Rysunek A.56 Współczynnik różnorodności grafu (assortativity_deg) w zależności od liczby wierzchołków — próbkowanie dwufazowe



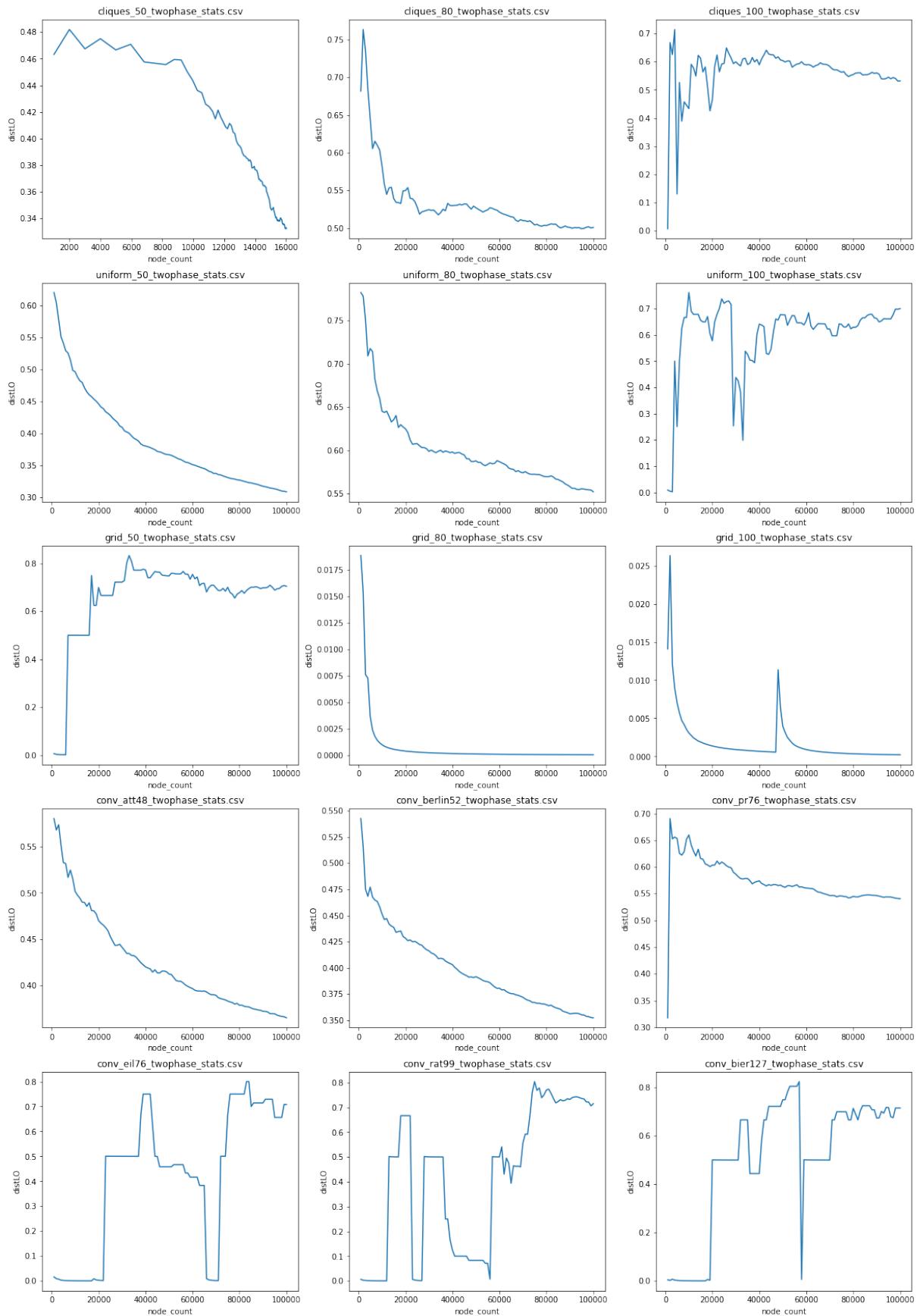
Rysunek A.57 Średnia wartość funkcji celu w znalezionych optimach lokalnych (avg_fitness) w zależności od liczby wierzchołków — próbkowanie dwufazowe



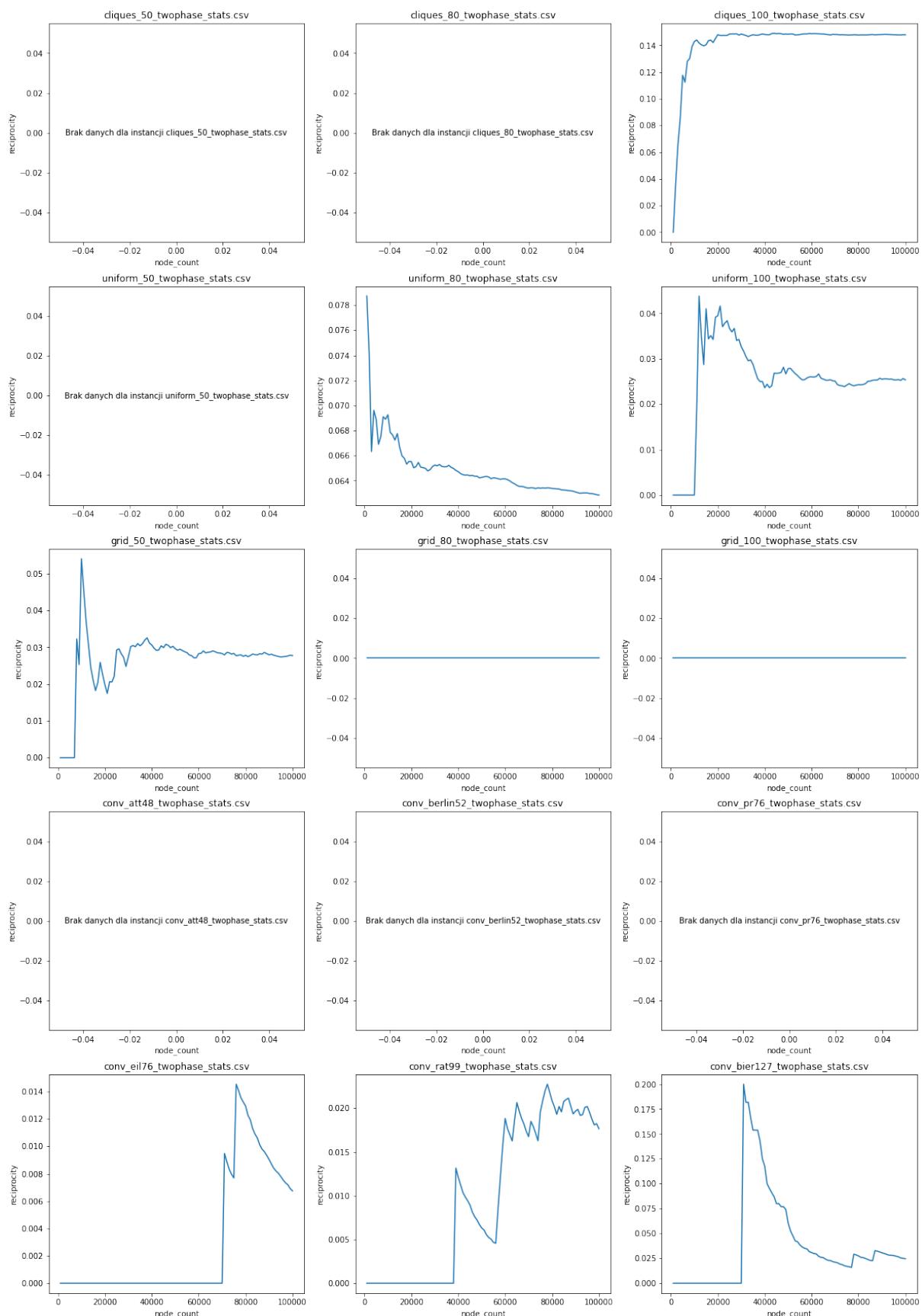
Rysunek A.58 Współczynnik conrel w zależności od liczby wierzchołków — próbkowanie dwufazowe



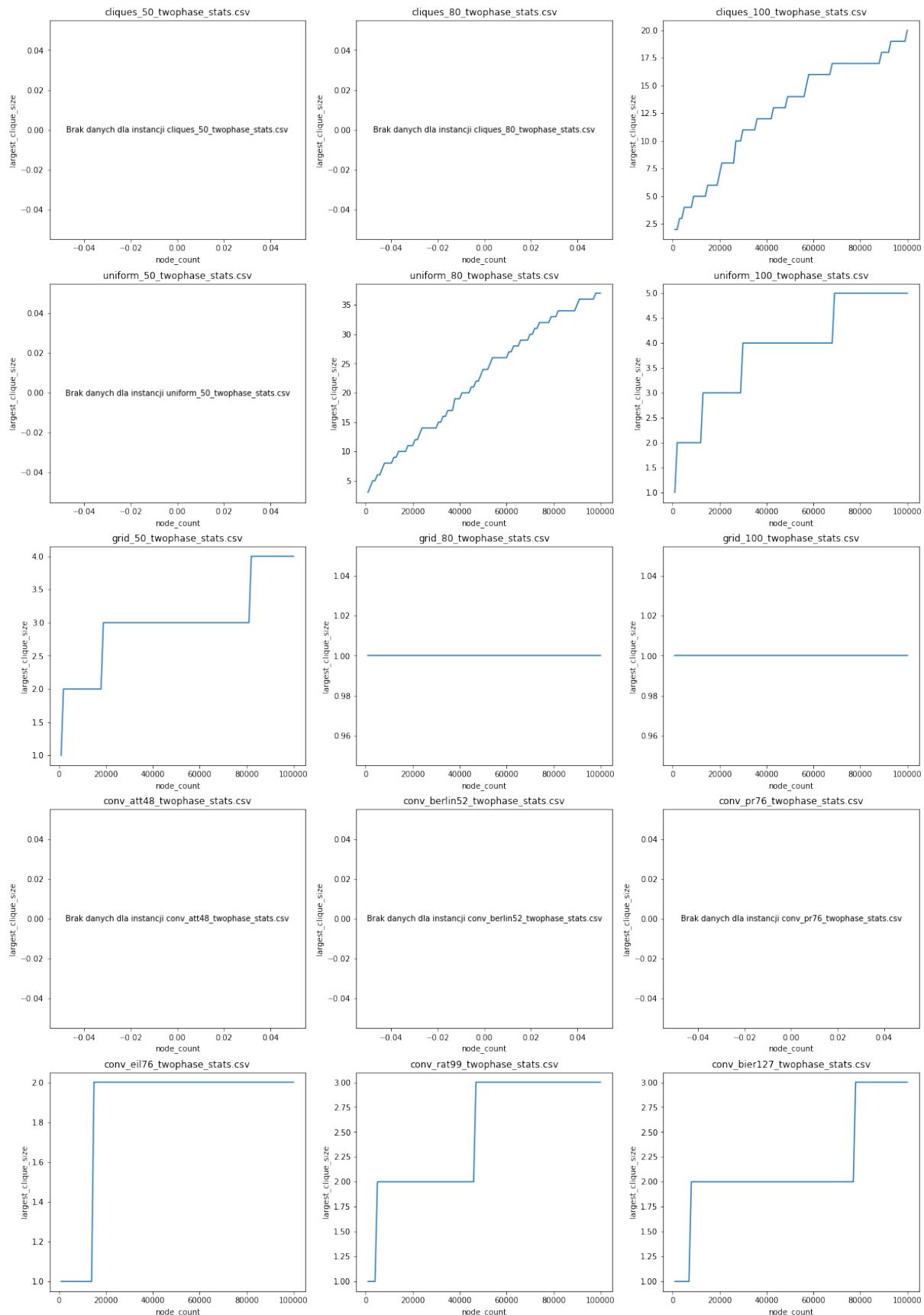
Rysunek A.59 Gęstość grafu (density) w zależności od liczby wierzchołków — próbkowanie dwufazowe



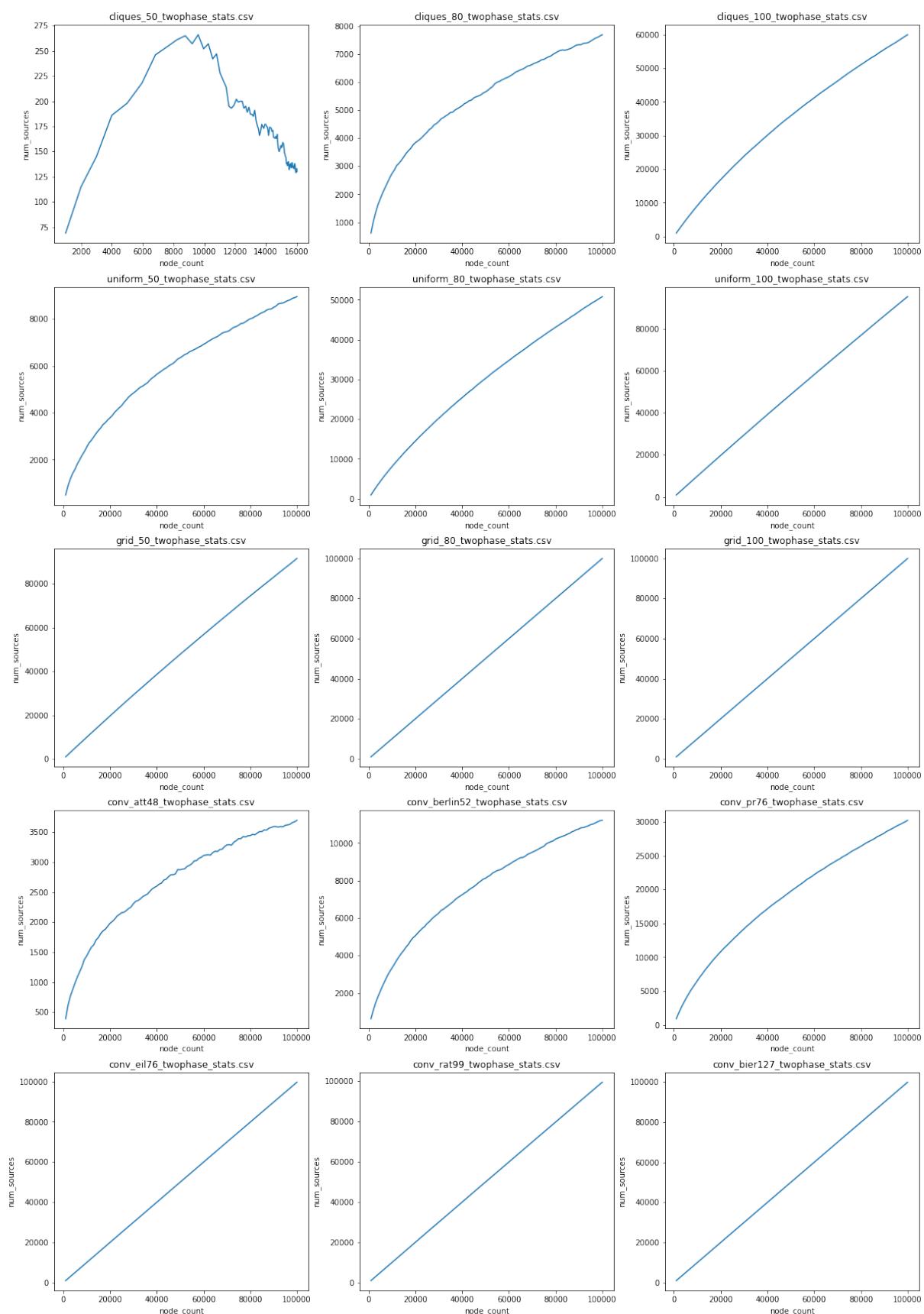
Rysunek A.60 Współczynnik distLO w zależności od liczby wierzchołków — próbkowanie dwufazowe



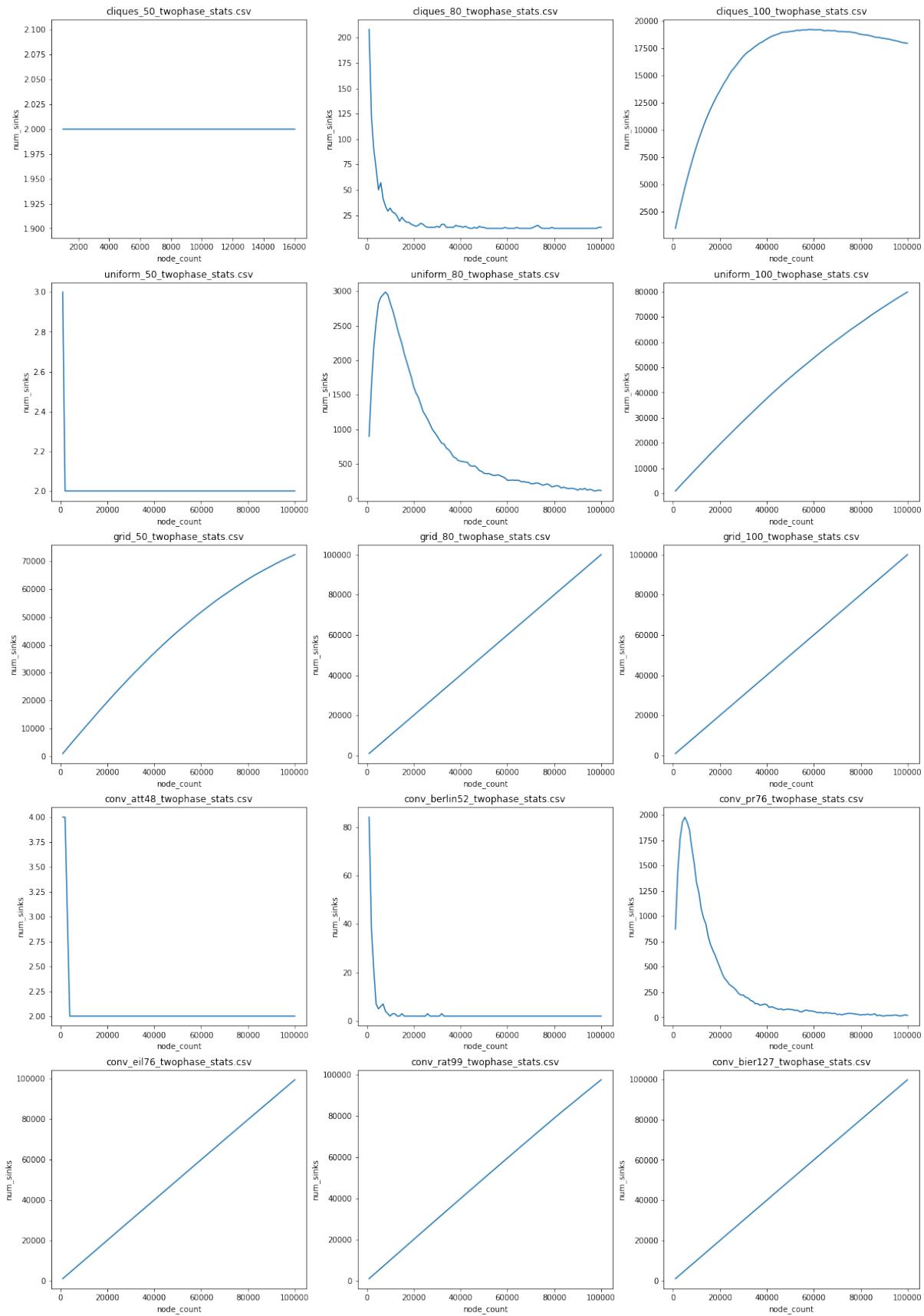
Rysunek A.61 Współczynnik wzajemności grafu (reciprocity) w zależności od liczby wierzchołków — próbkowanie dwufazowe



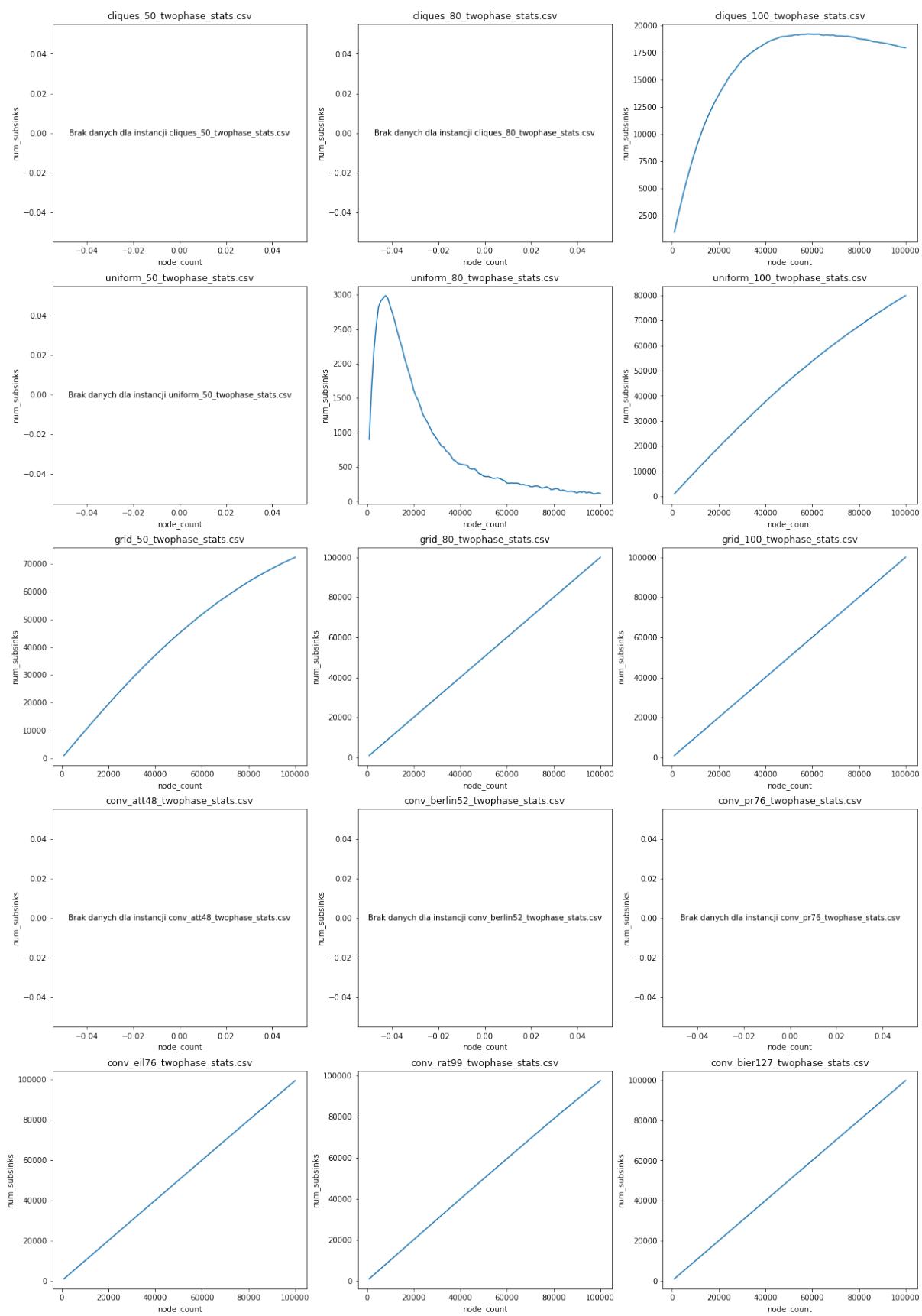
Rysunek A.62 Rozmiar największej kliki w grafie (largest_clique_size) w zależności od liczby wierzchołków — próbkowanie dwufazowe



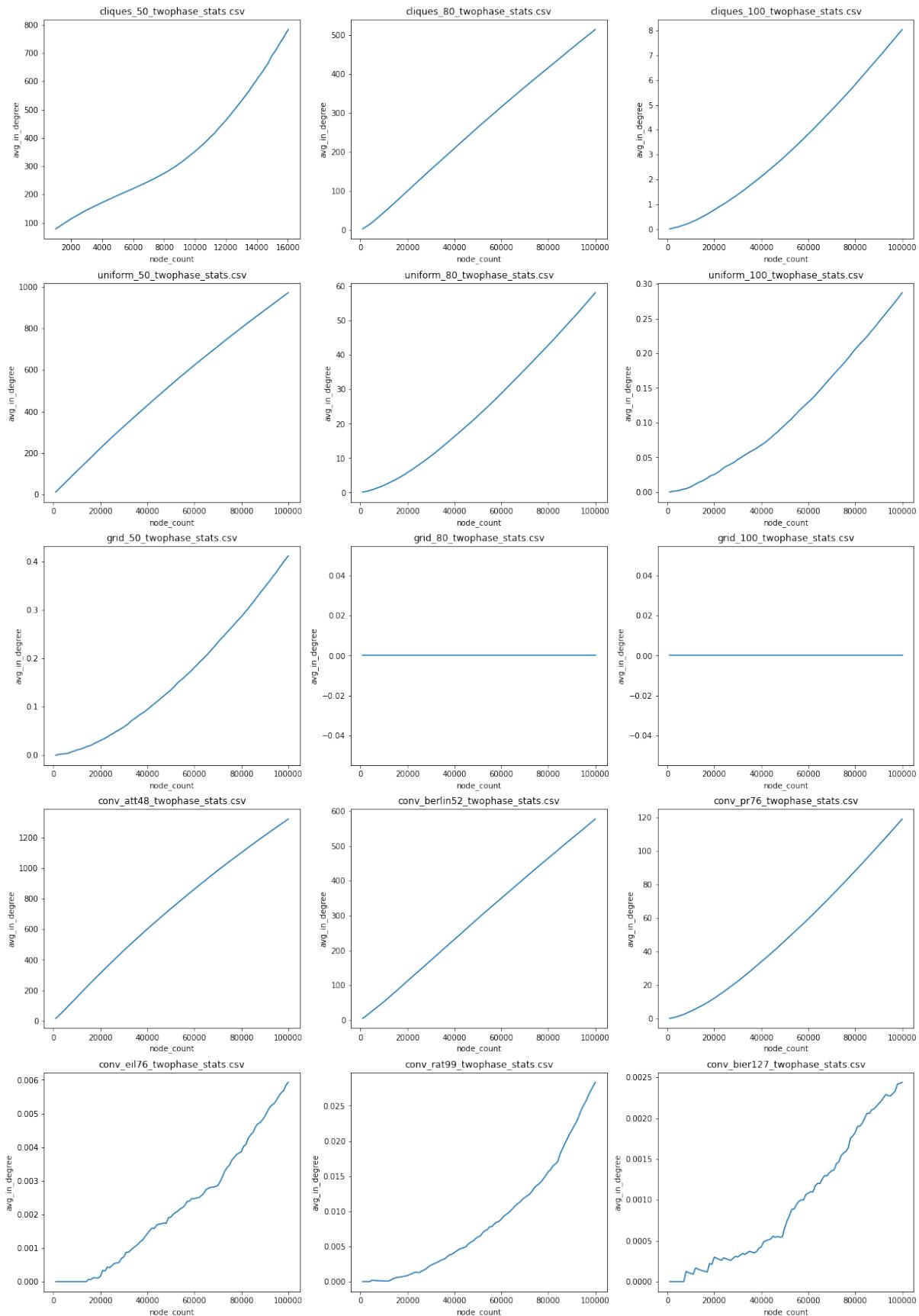
Rysunek A.63 Liczba źródeł w grafie (num_sources) w zależności od liczby wierzchołków — próbkowanie dwufazowe



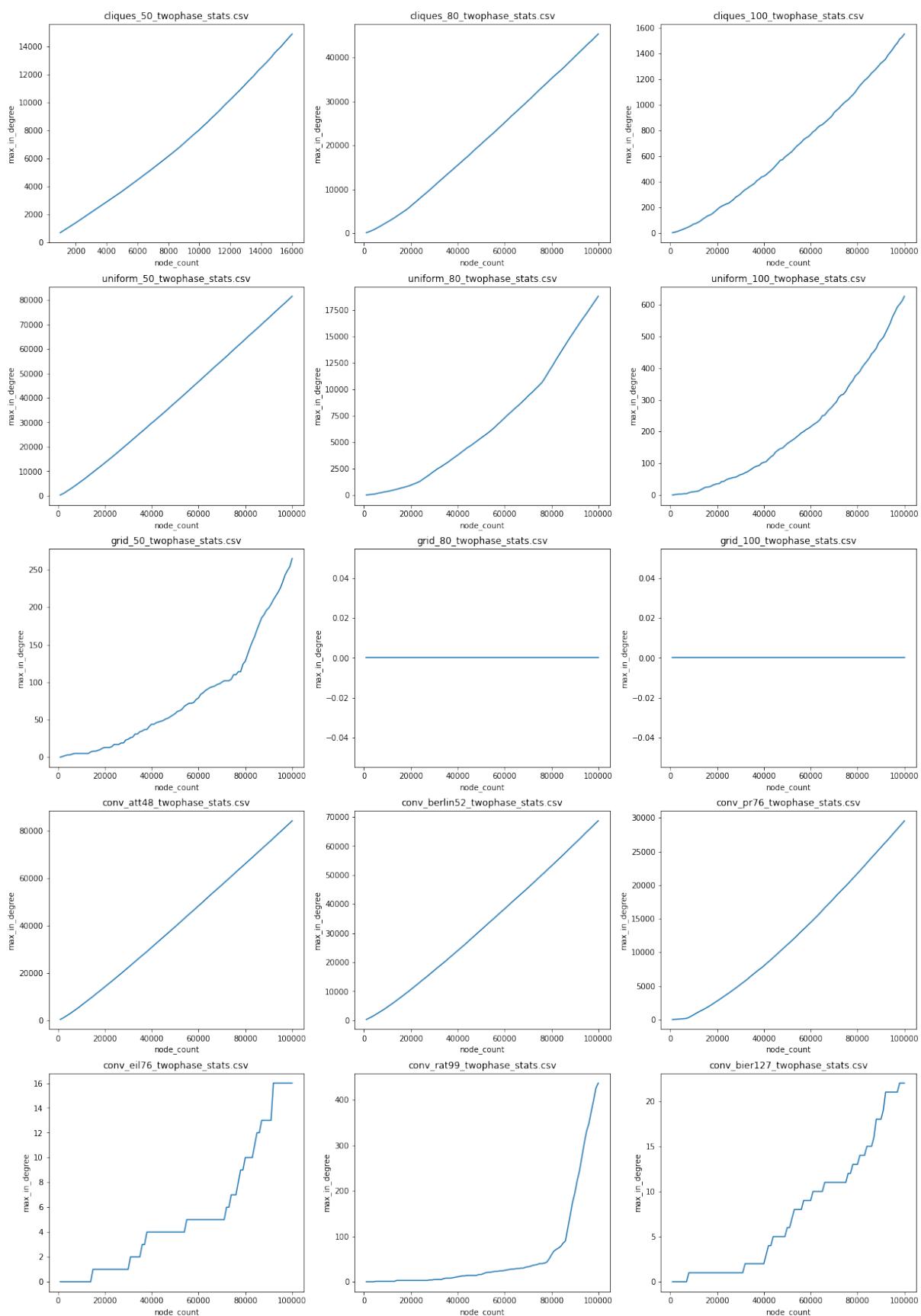
Rysunek A.64 Liczba ścieków w grafie (num_sinks) w zależności od liczby wierzchołków — próbkowanie dwufazowe



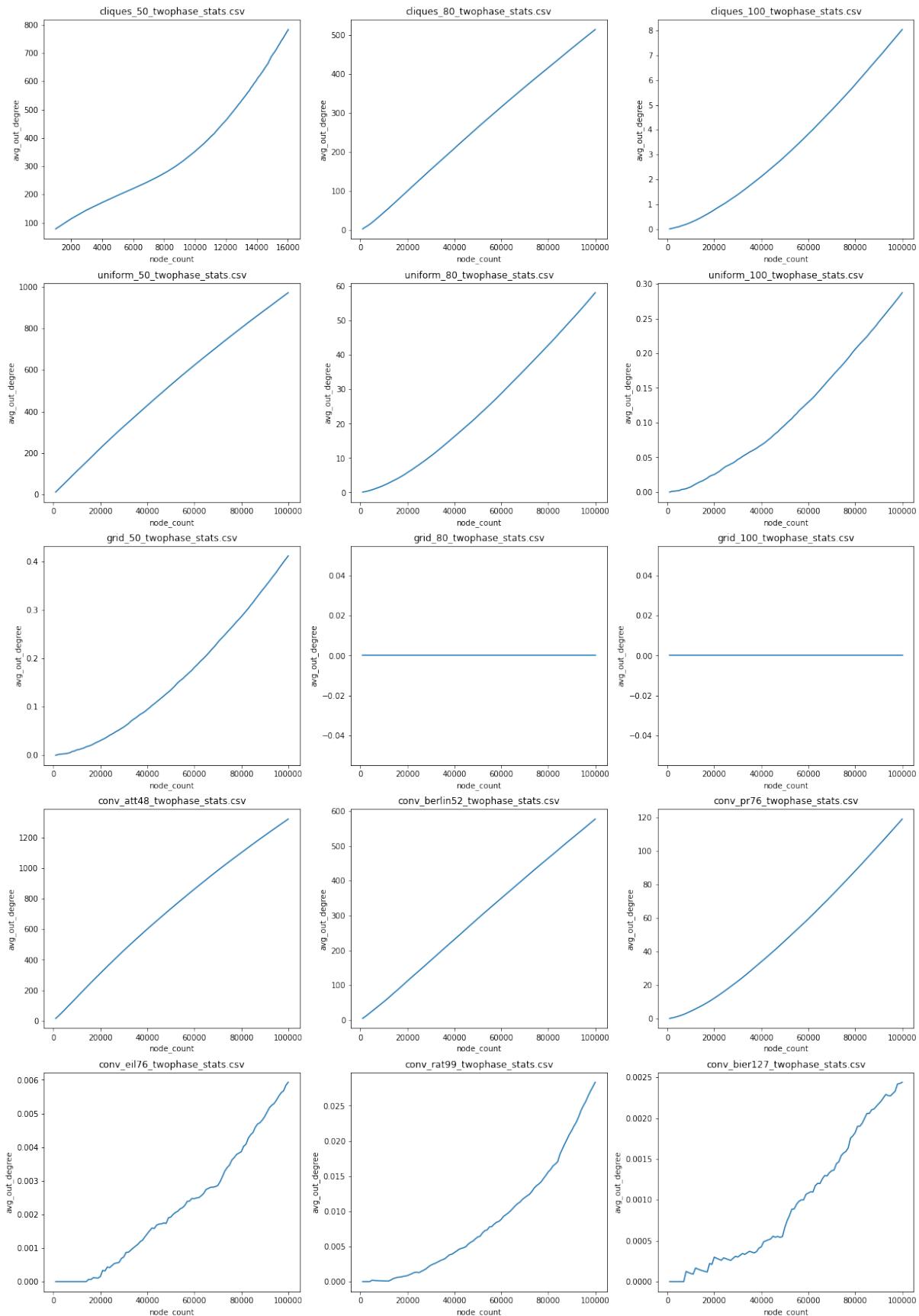
Rysunek A.65 Liczba subsinks w grafie (num_subsink) w zależności od liczby wierzchołków — próbkowanie dwufazowe



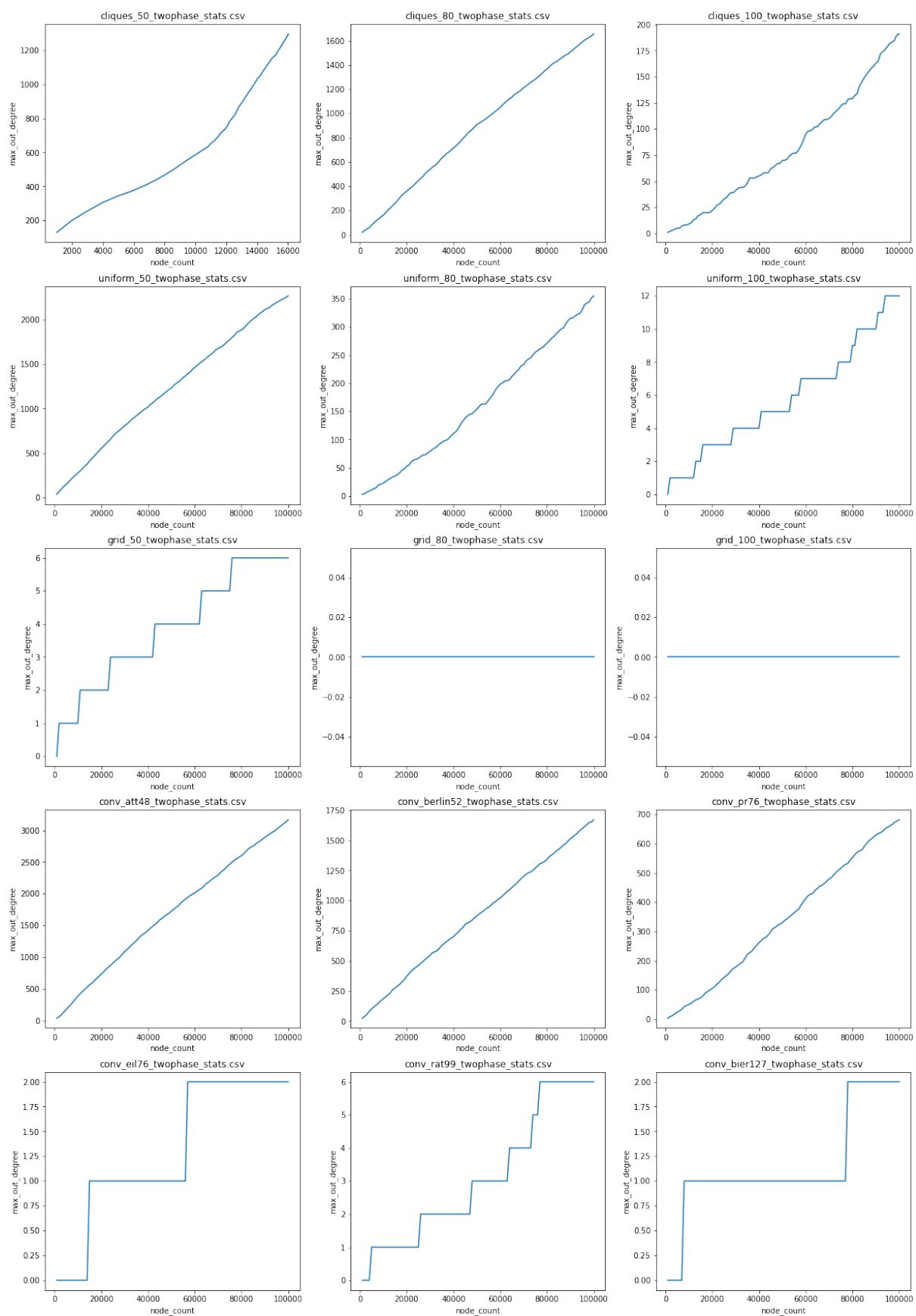
Rysunek A.66 Średni stopień wchodzący wierzchołków w grafie (avg_in_degree) w zależności od liczby wierzchołków — próbkowanie dwufazowe



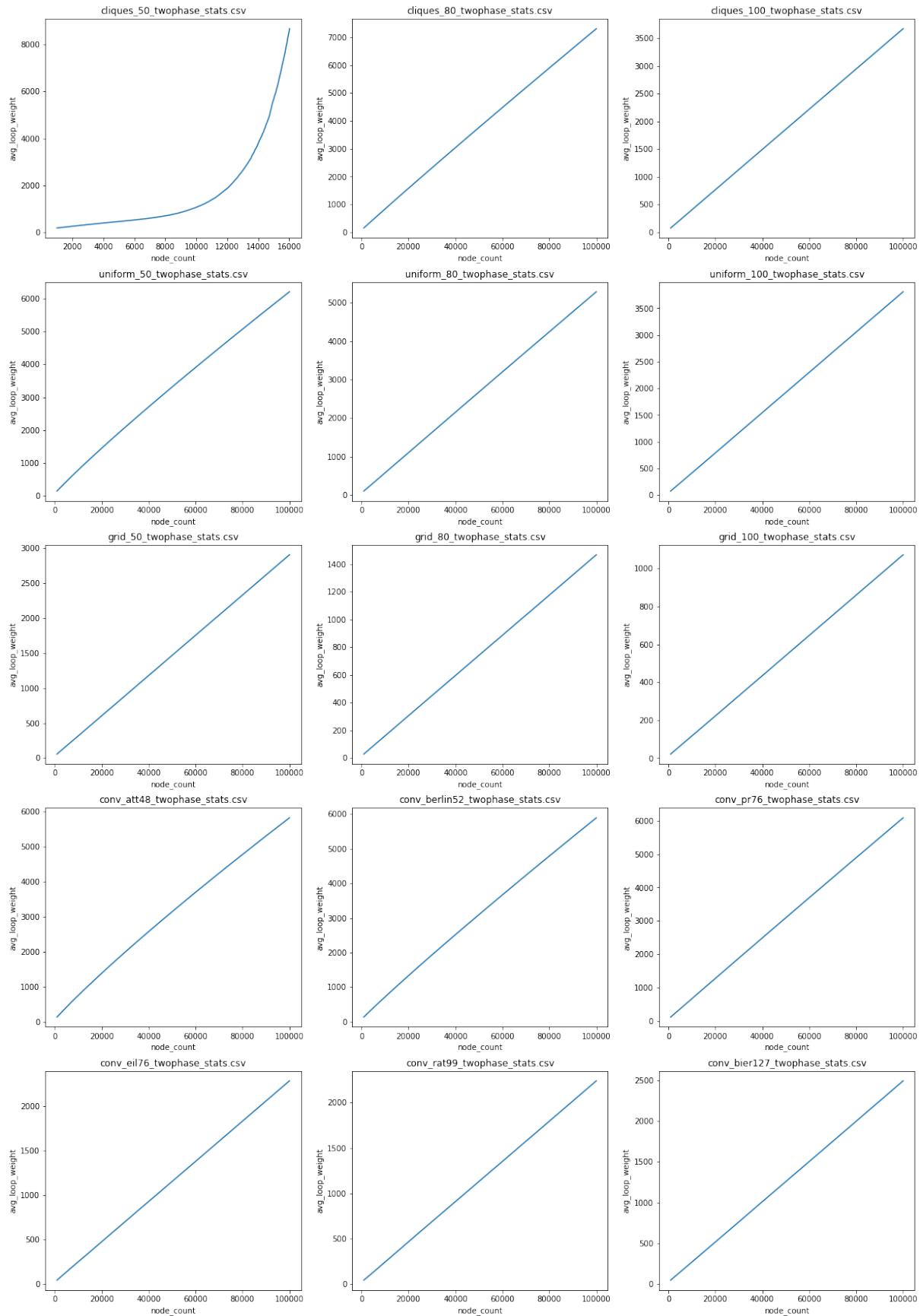
Rysunek A.67 Maksymalny stopień wchodzący wśród wierzchołków w grafie (max_in_degree) w zależności od liczby wierzchołków — próbkowanie dwufazowe



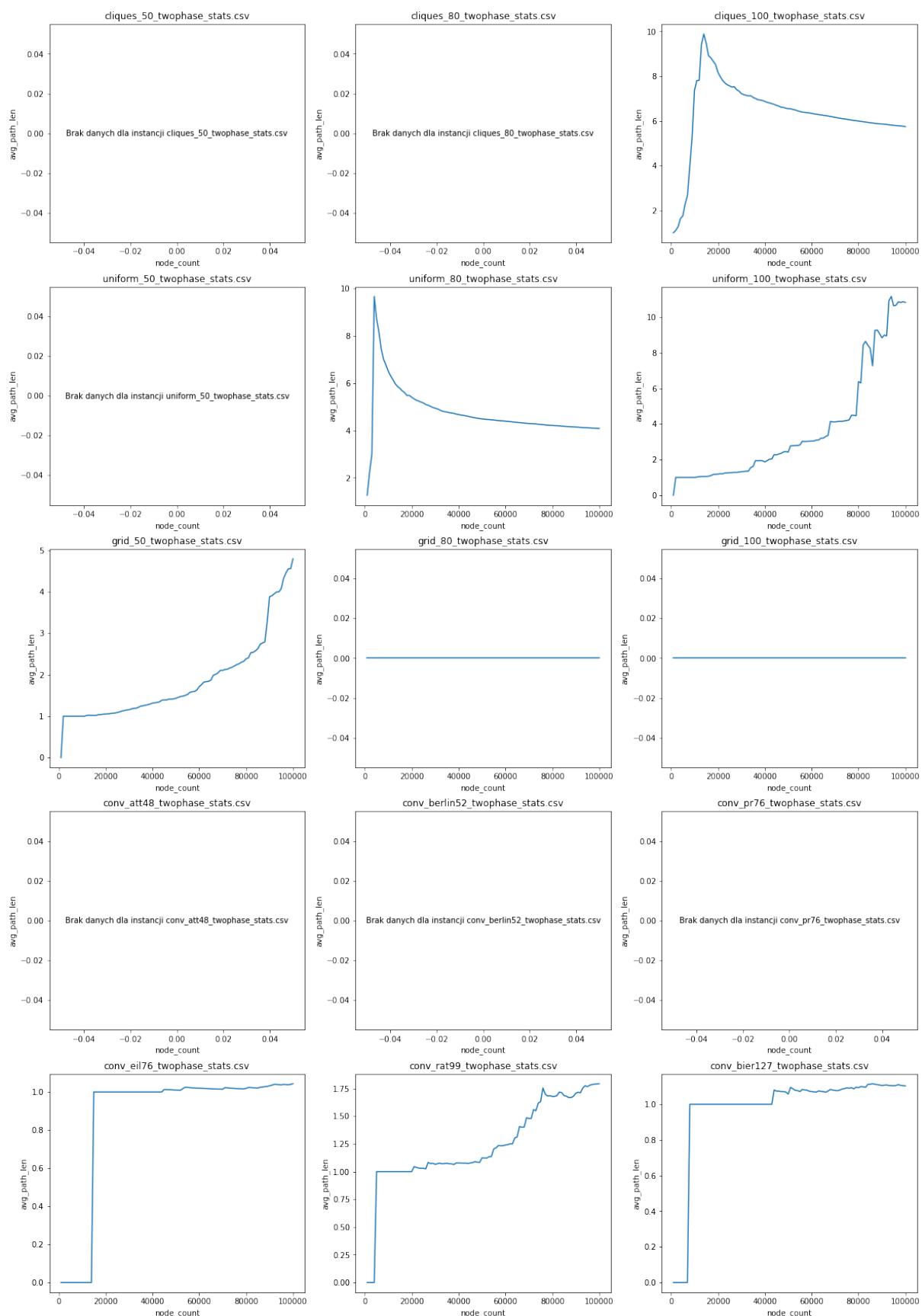
Rysunek A.68 Średni stopień wychodzący wśród wierzchołków w grafie (avg_out_degree) w zależności od liczby wierzchołków — próbkowanie dwufazowe



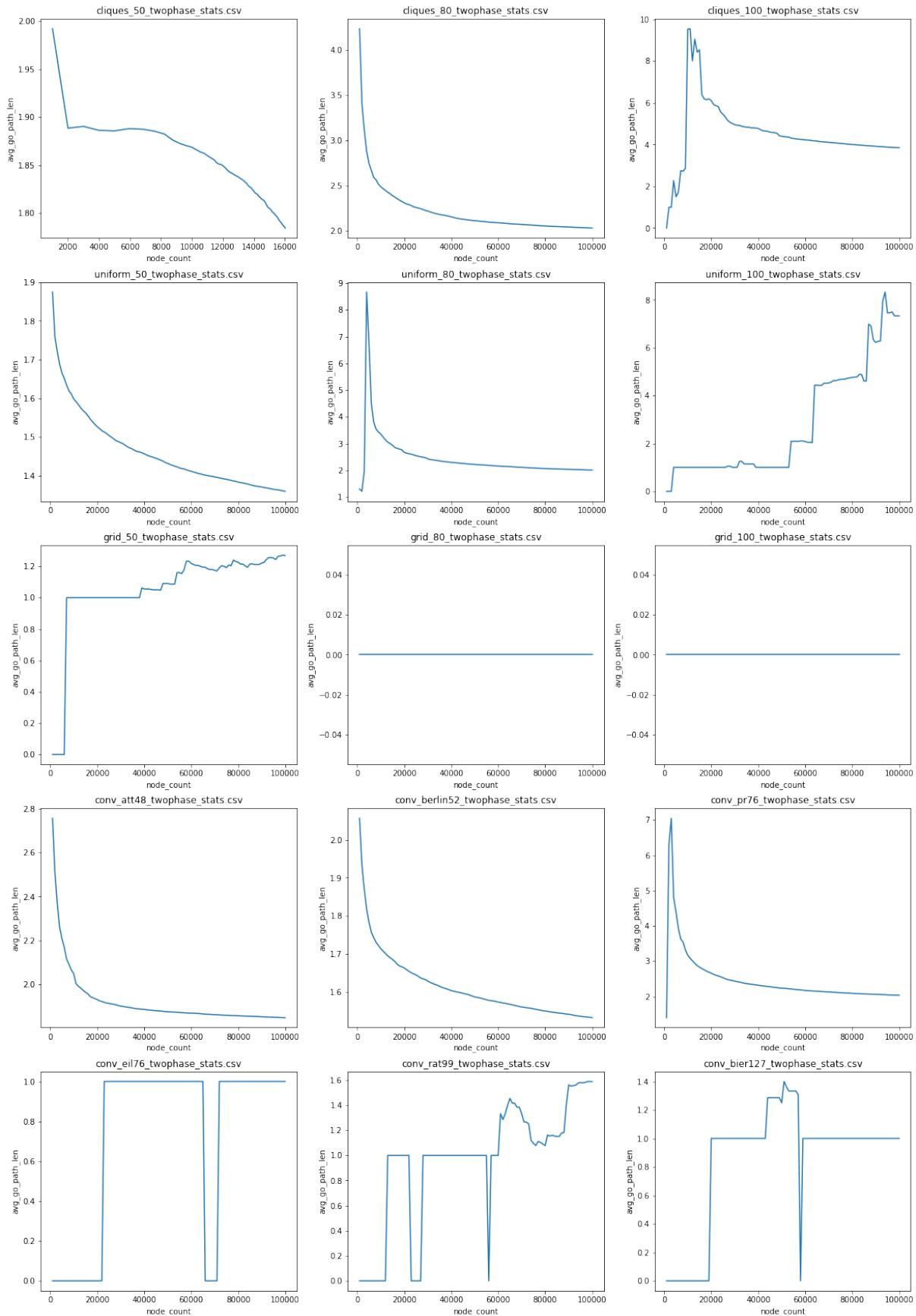
Rysunek A.69 Maksymalny stopień wychodzący wierzchołków w grafie (`max_out_degree`) w zależności od liczby wierzchołków — próbkowanie dwufazowe



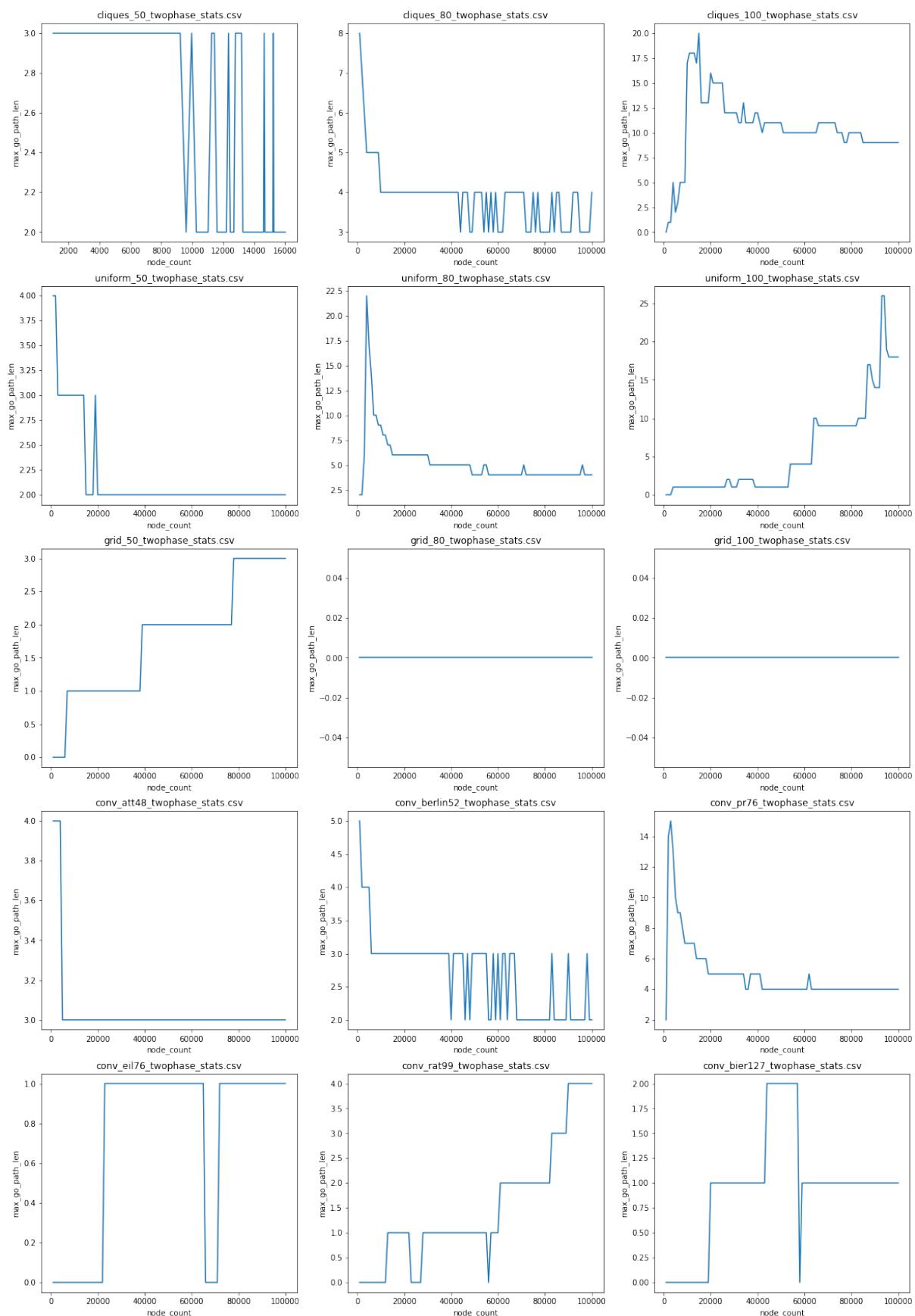
Rysunek A.70 Średnia waga pętli w grafie (avg_loop_weight) w zależności od liczby wierzchołków — próbkowanie dwufazowe



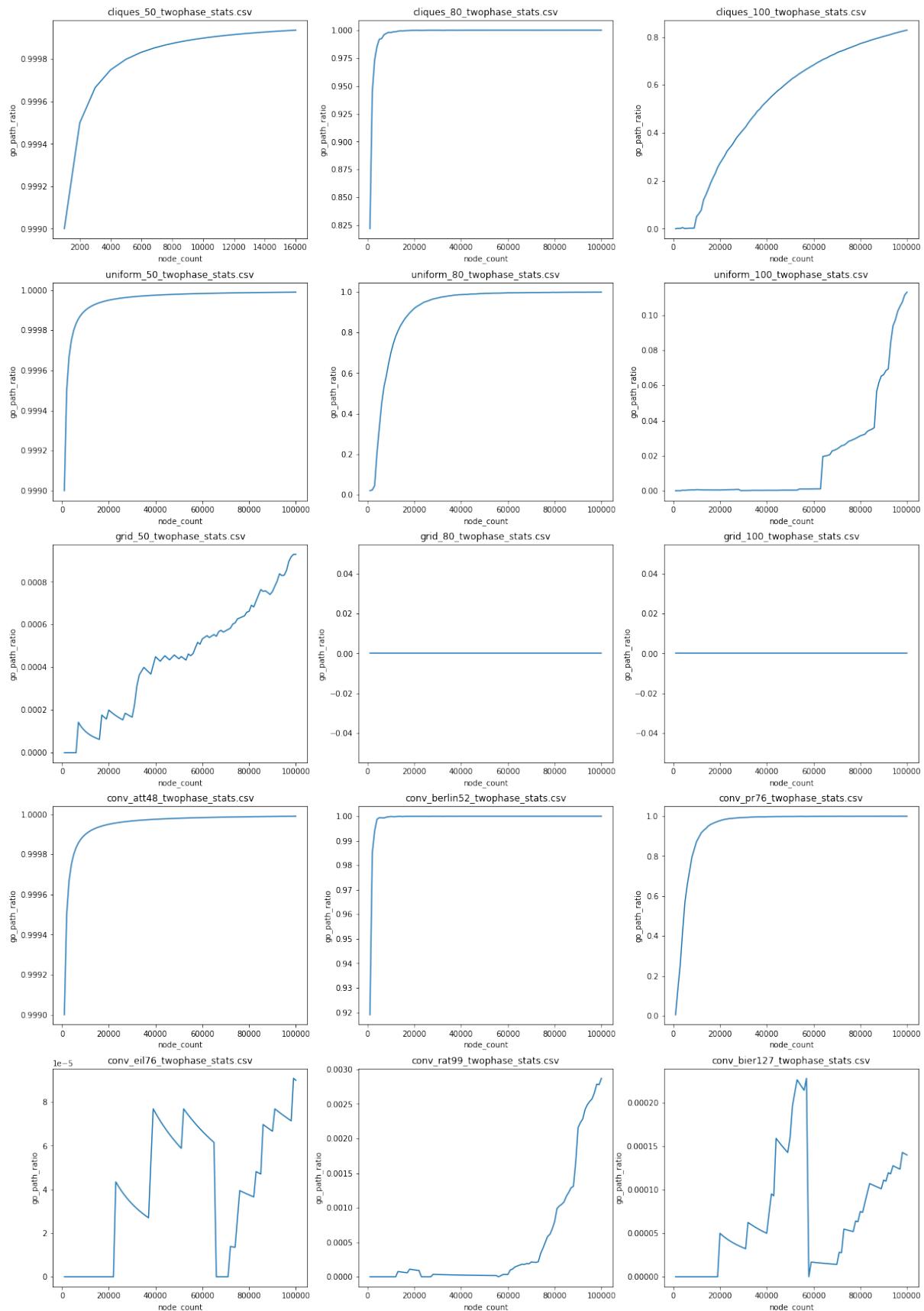
Rysunek A.71 Średnia długość ścieżki w grafie (avg_path_len) w zależności od liczby wierzchołków — próbkowanie dwufazowe



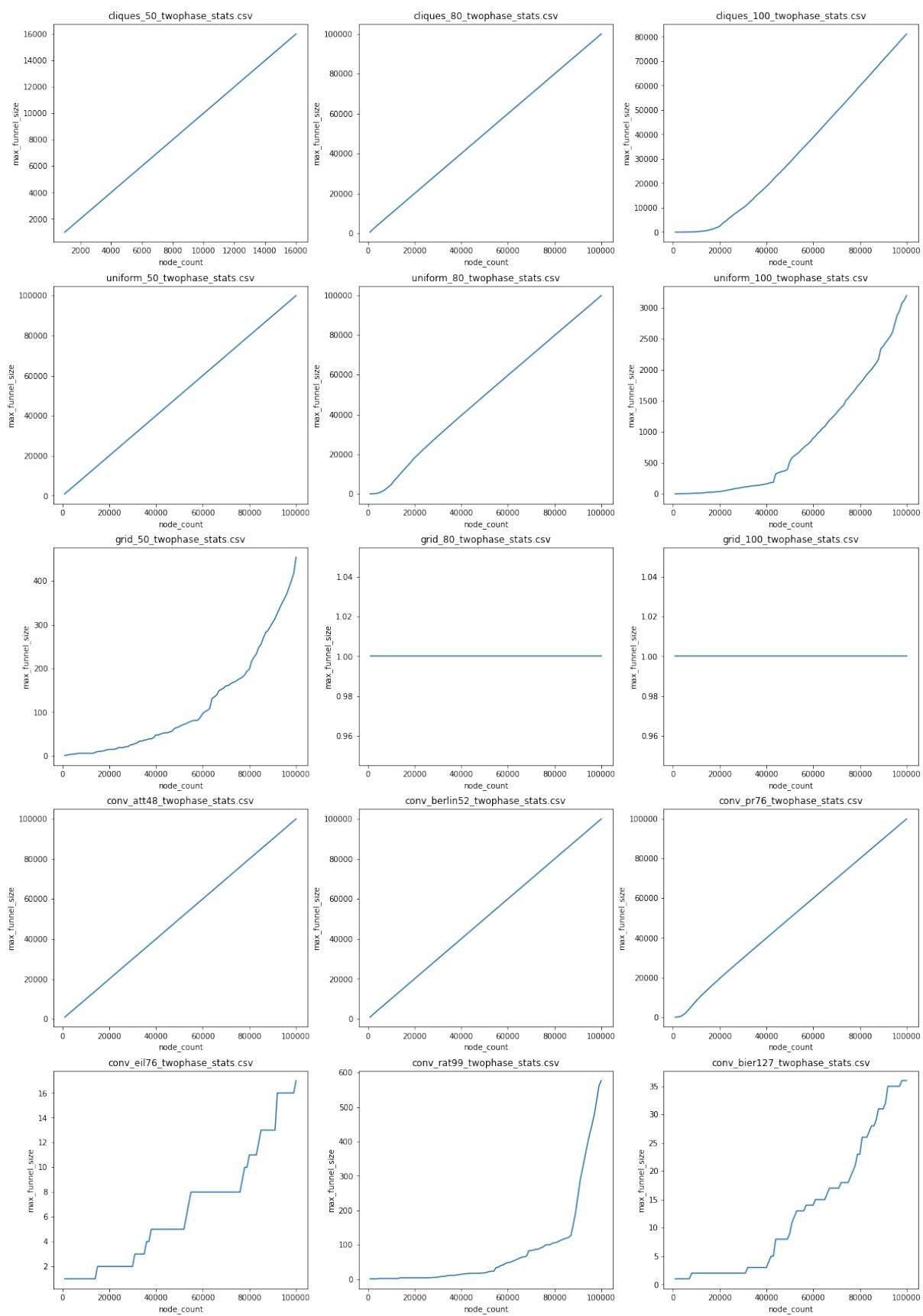
Rysunek A.72 Średnia długość istniejących ścieżek do globalnego optimum (`avg_go_path_len`) w zależności od liczby wierzchołków — próbkowanie dwufazowe



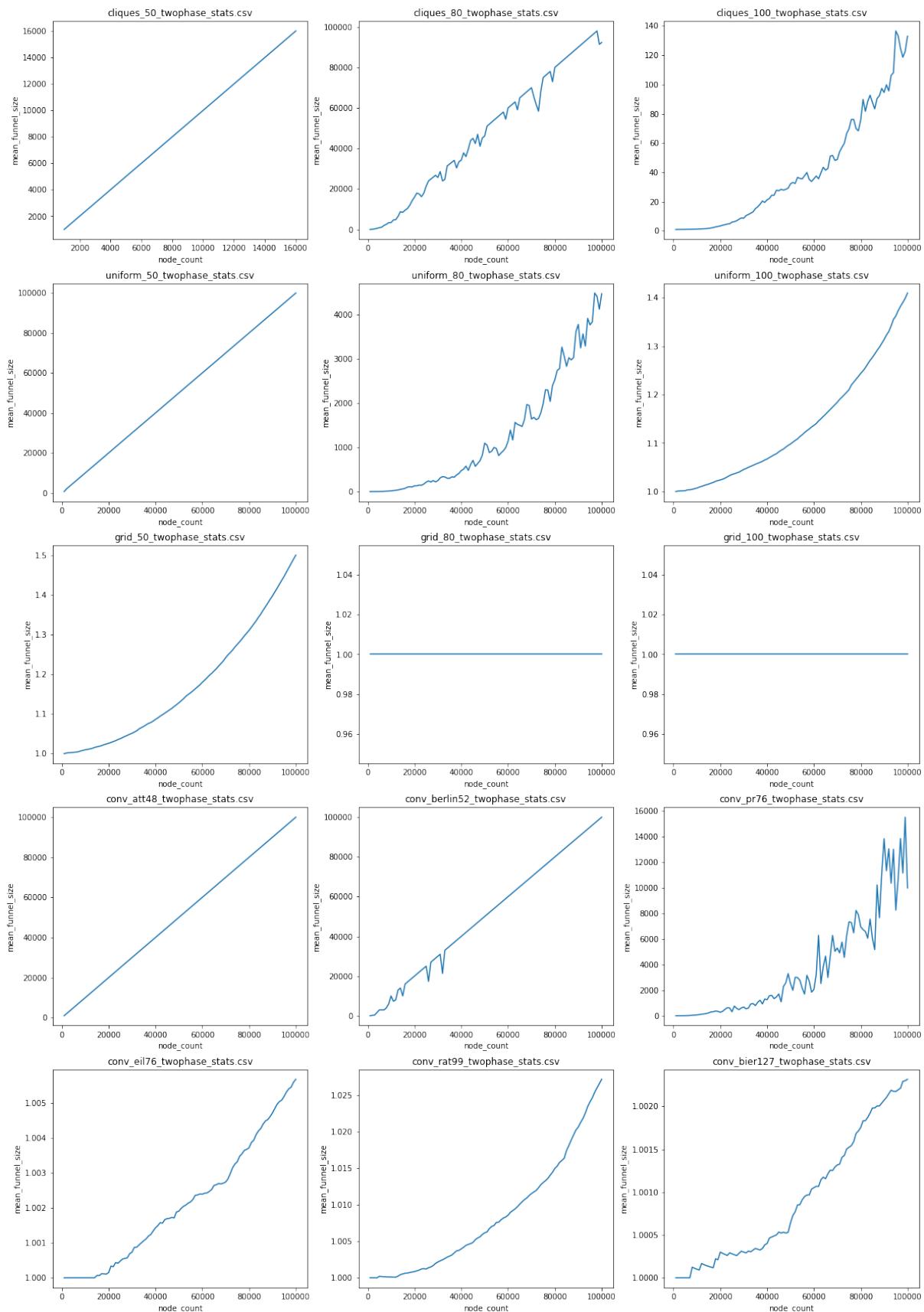
Rysunek A.73 Długość najdłuższej istniejącej ścieżki do globalnego optimum (max_go_path_len) w zależności od liczby wierzchołków — próbkowanie dwufazowe



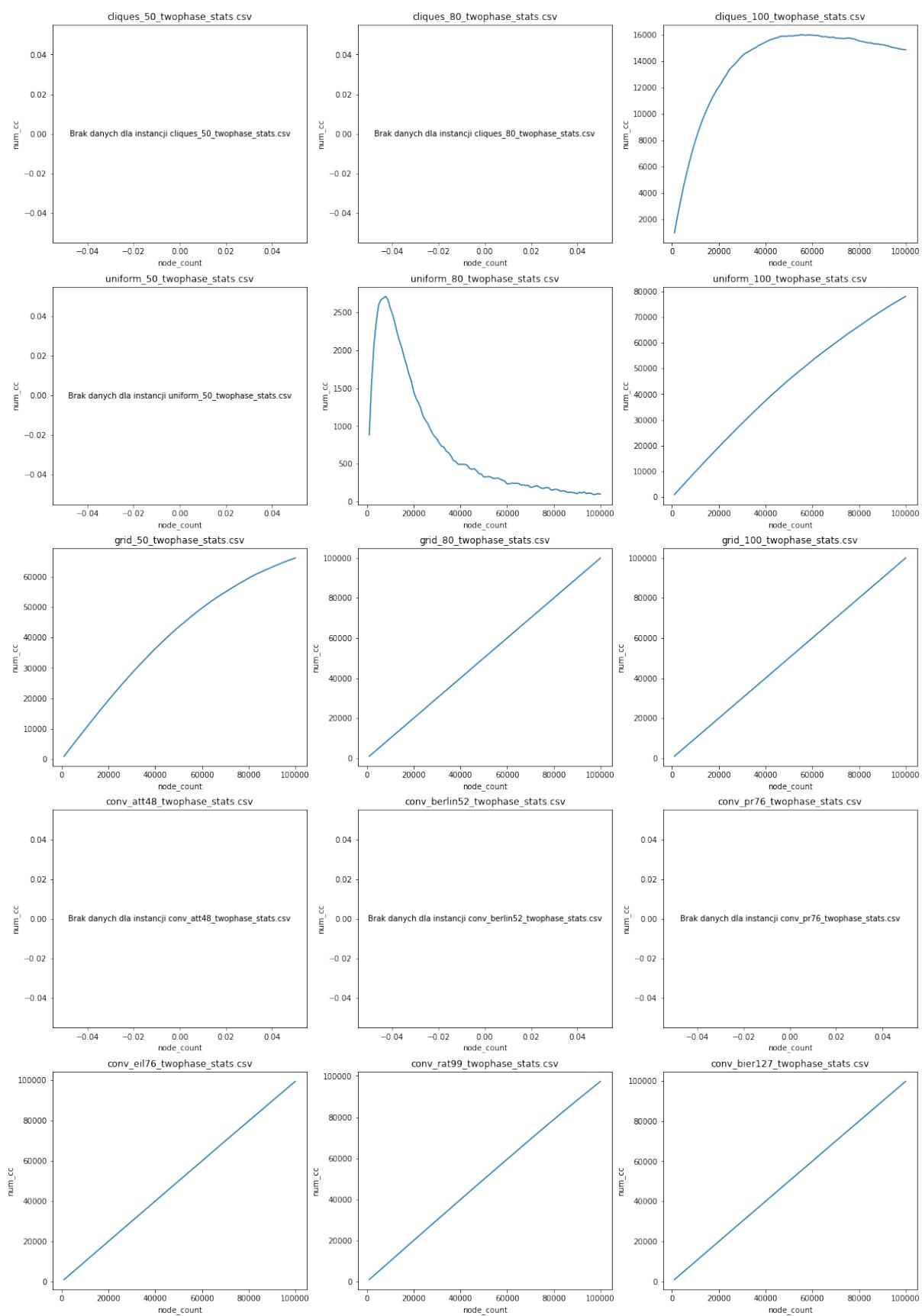
Rysunek A.74 Stosunek liczby wierzchołków, z których istnieje ścieżka do globalnego optimum do liczby wszystkich wierzchołków (go_path_ratio) w zależności od liczby wierzchołków — próbkowanie dwufazowe



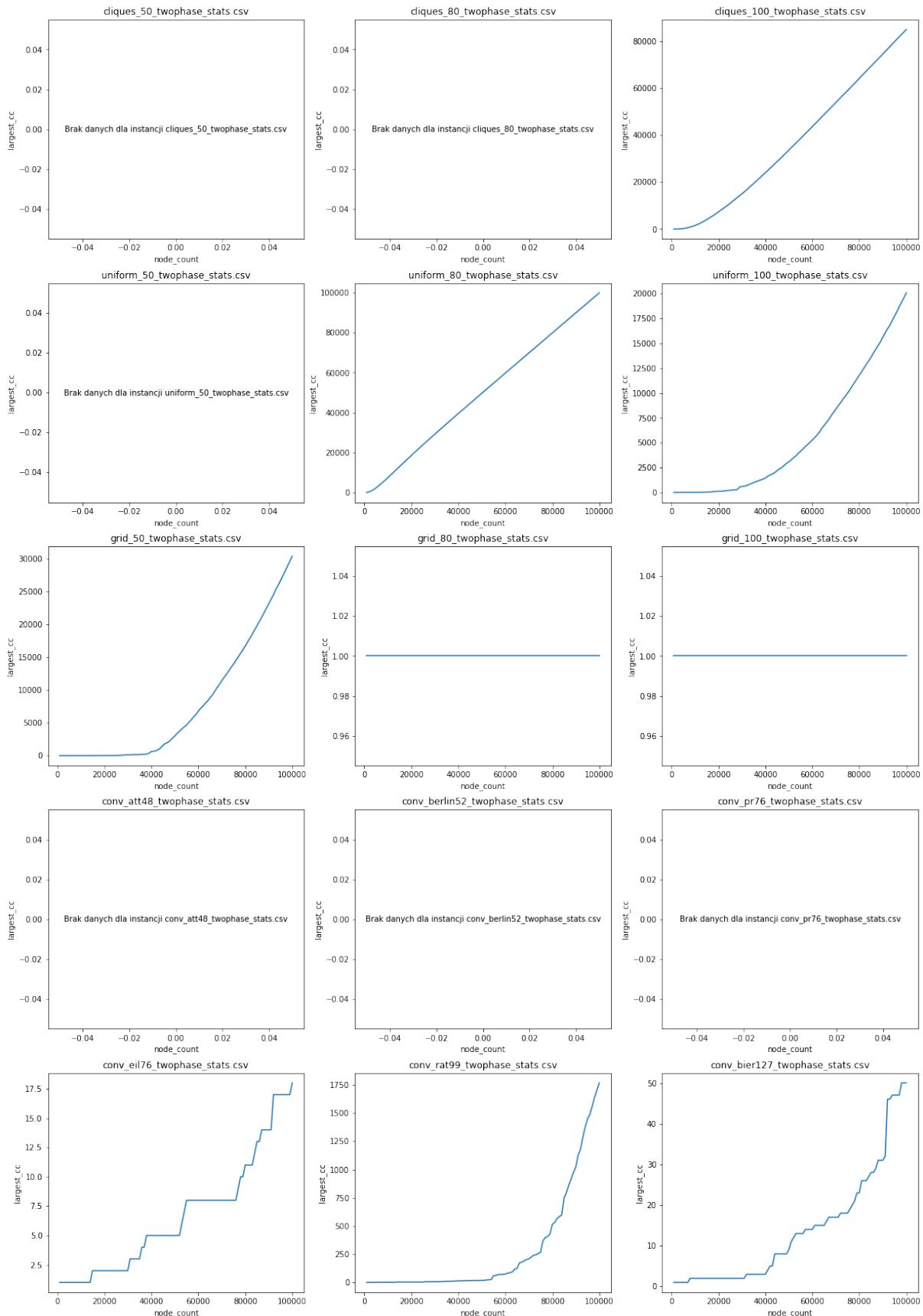
Rysunek A.75 Rozmiar największego leja w przestrzeni (max_funnel_size) w zależności od liczby wierzchołków — próbkowanie dwufazowe



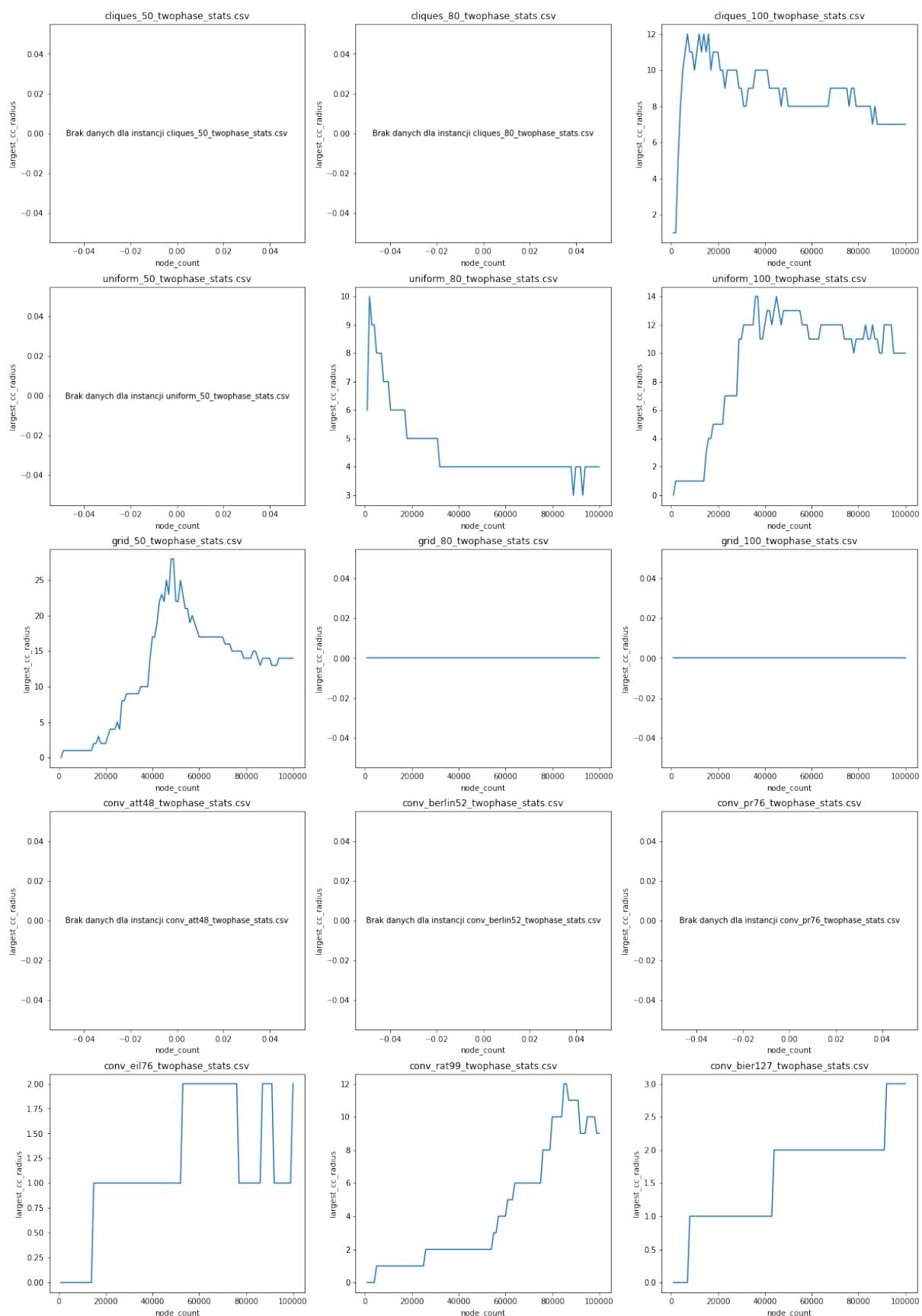
Rysunek A.76 Średni rozmiar lejów w przestrzeni (mean_funnel_size) w zależności od liczby wierzchołków — próbkowanie dwufazowe



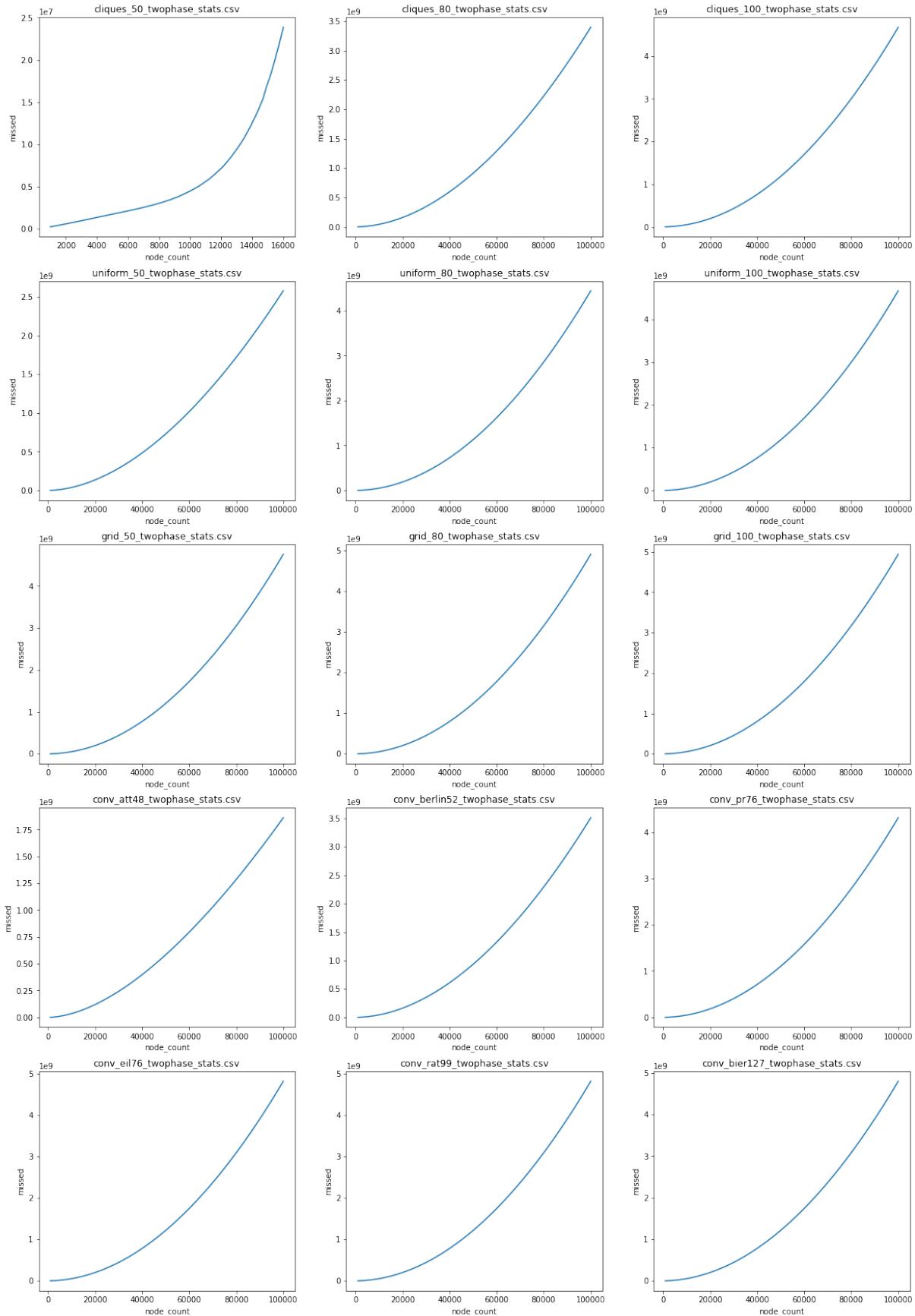
Rysunek A.77 Liczba spójnych podgrafów (num_cc) w zależności od liczby wierzchołków — próbkowanie dwufazowe



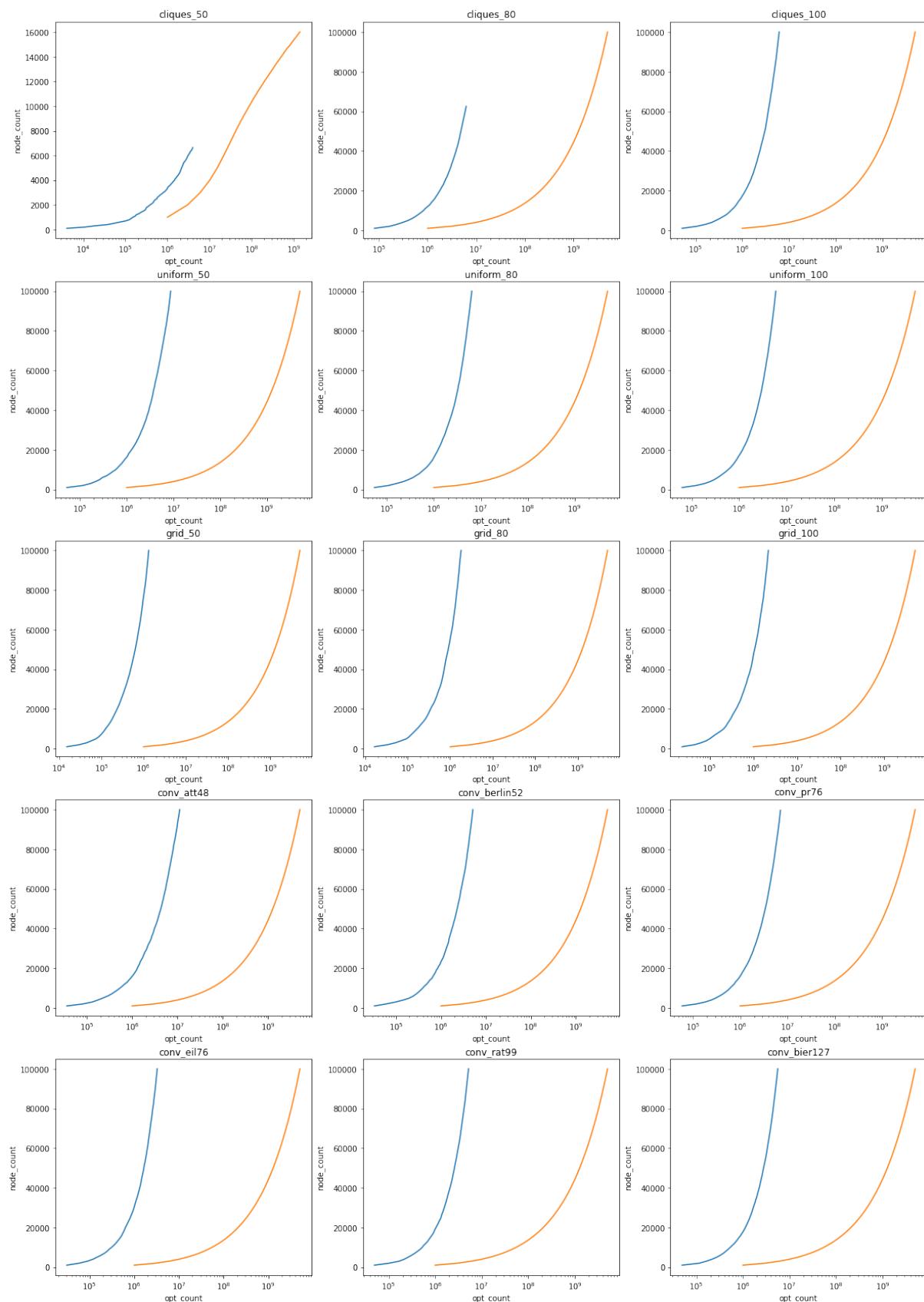
Rysunek A.78 Rozmiar największego spójnego podgrafa (largest_cc) w zależności od liczby wierzchołków — próbkowanie dwufazowe



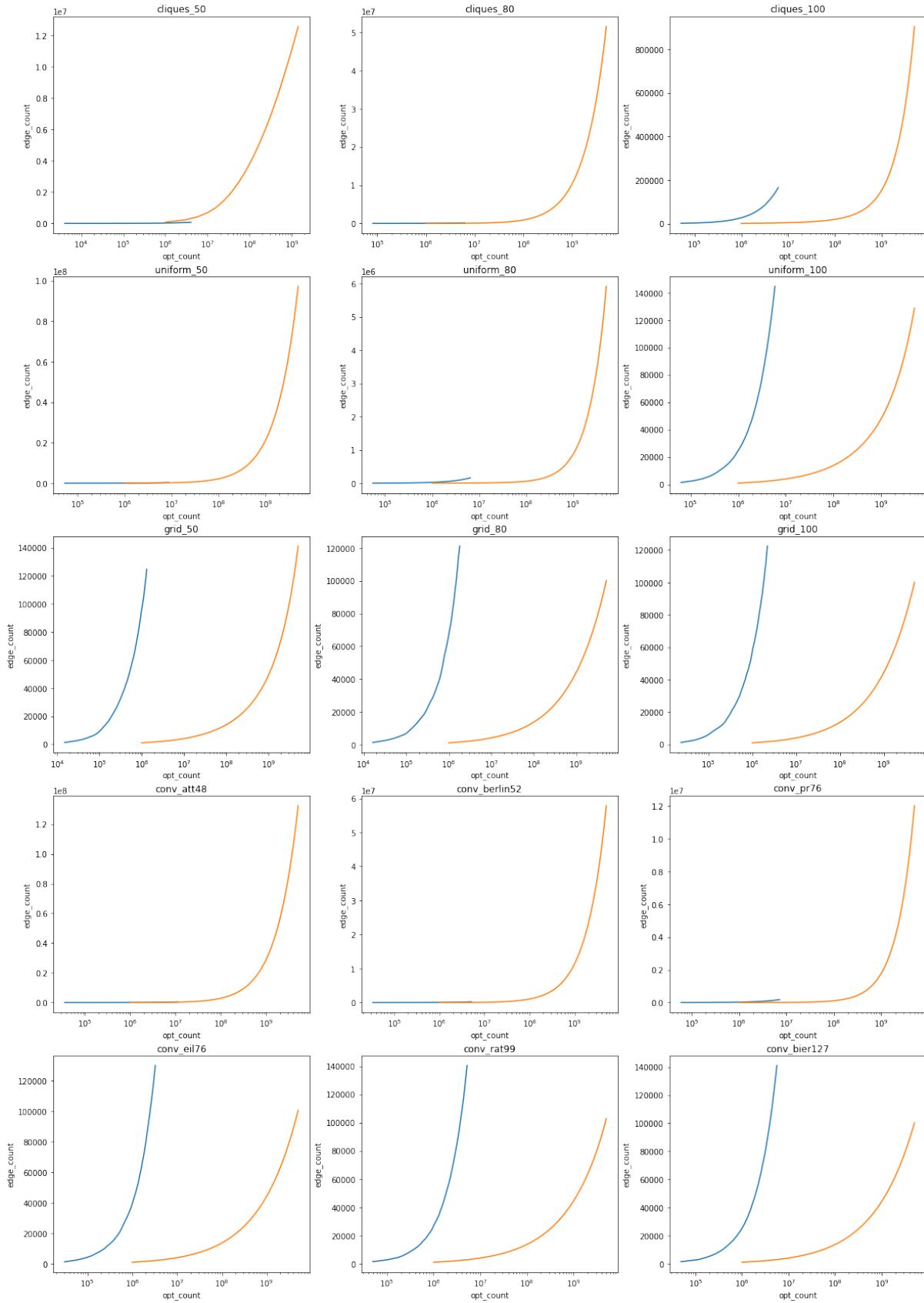
Rysunek A.79 Promień największego spójnego podgrafu (largest_cc_radius) w zależności od liczby wierzchołków — próbkowanie dwufazowe



Rysunek A.80 Liczba nieudanych prób tworzenia krawędzi w zależności od liczby wierzchołków — próbkowanie dwufazowe



Rysunek A.81 Wykres zależności liczby wierzchołków od liczby wywołań funkcji celu. Kolor niebieski - próbkowanie snowball, kolor pomarańczowy — próbkowanie dwufazowe. Oś X w skali logarytmicznej



Rysunek A.82 Wykres zależności liczby krawędzi od liczby wywołań funkcji celu. Kolor niebieski — próbkowanie snowball, kolor pomarańczowy — próbkowanie dwufazowe. Oś X w skali logarytmicznej

Literatura

- [1] T. ACHARY, S. M. PILLAY, S. M. PILLAI, M. MQADI, E. GENDERS, AND A. E. EZUGWU, *A performance study of meta-heuristic approaches for quadratic assignment problem*, Concurr. Comput. Pract. Exp., 33 (2021).
- [2] U. BENLIC AND J. HAO, *Memetic search for the quadratic assignment problem*, Expert Syst. Appl., 42 (2015), pp. 584–595.
- [3] W. BOZEJKO, A. GNATOWSKI, T. NIZYNSKI, M. AFFENZELLER, AND A. BEHAM, *Local optima networks in solving algorithm selection problem for TSP*, in Contemporary Complex Systems and Their Dependability - Proceedings of the 13th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, July 2-6, 2018, Brunów, Poland, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, eds., vol. 761 of Advances in Intelligent Systems and Computing, Springer, 2018, pp. 83–93.
- [4] C. W. CLEGHORN AND G. OCHOA, *Understanding parameter spaces using local optima networks: a case study on particle swarm optimization*, in GECCO '21: Genetic and Evolutionary Computation Conference, Companion Volume, Lille, France, July 10-14, 2021, K. Krawiec, ed., ACM, 2021, pp. 1657–1664.
- [5] G. CSÁRDI, T. NEPU SZ, V. TRAAG, S. HORVÁT, F. ZANINI, AND D. NOOM, *Igraph C Reference Manual*. <https://igraph.org/c/html/latest/>, dostęp: 27.05.2023.
- [6] T. DÖKEROGLU, E. SEVINÇ, AND A. COSAR, *Artificial bee colony optimization for the quadratic assignment problem*, Appl. Soft Comput., 76 (2019), pp. 595–606.
- [7] I. FRAGATA, A. BLANCKAERT, M. A. DIAS LOURO, D. A. LIBERLES, AND C. BANK, *Evolution in the light of fitness landscape theory*, Trends in Ecology & Evolution, 34 (2019), pp. 69–82.
- [8] K. HELSGAUN, *An effective implementation of the Lin-Kernighan traveling salesman heuristic*, Eur. J. Oper. Res., 126 (2000), pp. 106–130.
- [9] D. IC LANZAN, F. DAOLIO, AND M. TOMASSINI, *Data-driven local optima network characterization of qaplib instances*, in Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO '14, New York, NY, USA, 2014, Association for Computing Machinery, p. 453–460.
- [10] M. JÜNGER, G. REINELT, AND G. RINALDI, *Chapter 4 The traveling salesman problem*, in Network Models, vol. 7 of Handbooks in Operations Research and Management Science, Elsevier, 1995, pp. 225–330.

- [11] E. L. LAWLER, *The quadratic assignment problem*, Management Science, 9 (1963), pp. 586–599.
- [12] P. McMENEMY, N. VEERAPEN, AND G. OCHOA, *How perturbation strength shapes the global structure of TSP fitness landscapes*, in Evolutionary Computation in Combinatorial Optimization - 18th European Conference, EvoCOP 2018, Parma, Italy, April 4-6, 2018, Proceedings, A. Liefooghe and M. López-Ibáñez, eds., vol. 10782 of Lecture Notes in Computer Science, Springer, 2018, pp. 34–49.
- [13] G. OCHOA AND S. HERRMANN, *Perturbation Strength and the Global Structure of QAP Fitness Landscapes: 15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part II*, 01 2018, pp. 245–256.
- [14] G. OCHOA AND N. VEERAPEN, *Mapping the global structure of TSP fitness landscapes*, J. Heuristics, 24 (2018), pp. 265–294.
- [15] G. OCHOA, S. VÉREL, F. DAOLIO, AND M. TOMASSINI, *Local optima networks: A new model of combinatorial fitness landscapes*, CoRR, abs/1402.2959 (2014).
- [16] G. OCHOA, S. VÉREL, AND M. TOMASSINI, *First-improvement vs. best-improvement local optima networks of NK landscapes*, CoRR, abs/1207.4455 (2012).
- [17] X. PAN, Y. JIN, Y. DING, M. FENG, L. ZHAO, L. SONG, AND J. BIAN, *H-TSP: Hierarchically Solving the Large-Scale Travelling Salesman problem*, CoRR, abs/2304.09395 (2023).
- [18] M. C. TEIXEIRA AND G. L. PAPPA, *Understanding AutoML search spaces with local optima networks*, in GECCO '22: Genetic and Evolutionary Computation Conference, Boston, Massachusetts, USA, July 9 - 13, 2022, J. E. Fieldsend and M. Wagner, eds., ACM, 2022, pp. 449–457.
- [19] S. L. THOMSON, G. OCHOA, AND S. VÉREL, *Clarifying the difference in local optima network sampling algorithms*, in Evolutionary Computation in Combinatorial Optimization - 19th European Conference, EvoCOP 2019, Held as Part of EvoStar 2019, Leipzig, Germany, April 24-26, 2019, Proceedings, A. Liefooghe and L. Paquete, eds., vol. 11452 of Lecture Notes in Computer Science, Springer, 2019, pp. 163–178.
- [20] S. L. THOMSON, G. OCHOA, S. VÉREL, AND N. VEERAPEN, *Inferring future landscapes: Sampling the local optima level*, Evol. Comput., 28 (2020), pp. 621–641.
- [21] M. TOMASSINI, S. VÉREL, AND G. OCHOA, *Complex-network analysis of combinatorial spaces: The NK landscape case*, Phys. Rev. E, 78 (2008), p. 066114.
- [22] S. VÉREL, F. DAOLIO, G. OCHOA, AND M. TOMASSINI, *Sampling local optima networks of large combinatorial search spaces: The QAP case*, in Parallel Problem Solving from Nature - PPSN XV - 15th International Conference, Coimbra, Portugal, September 8-12, 2018, Proceedings, Part II, A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and L. D. Whitley, eds., vol. 11102 of Lecture Notes in Computer Science, Springer, 2018, pp. 257–268.
- [23] L. XIN, W. SONG, Z. CAO, AND J. ZHANG, *NeuroLKH: Combining Deep Learning Model with Lin-Kernighan-Helsgaun Heuristic for Solving the Traveling Salesman*

- Problem*, in Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds., 2021, pp. 7472–7483.
- [24] H. ZHANG, F. LIU, Y. ZHOU, AND Z. ZHANG, *A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem*, Inf. Sci., 539 (2020), pp. 347–374.
- [25] Z. ZHANG AND Y. HAN, *Discrete sparrow search algorithm for symmetric traveling salesman problem*, Appl. Soft Comput., 118 (2022), p. 108469.

Spis rysunków

3.1 Przykład procedury 2-exchange	15
4.1 Wizualizacja przykładowej wygenerowanej instancji z miastami rozmieszczonymi równomiernie dla 100 miast	23
4.2 Wizualizacja przykładowej wygenerowanej instancji z miastami rozmieszczonymi w klikach dla 100 miast	23
4.3 Wizualizacja przykładowej wygenerowanej instancji z miastami rozmieszczonymi na siatce dla 100 miast	24
4.4 Wzajemna korelacja miar — próbkowanie snowball	49
4.5 Wzajemna korelacja miar — próbkowanie dwufazowe	50
4.6 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) — próbkowanie snowball	54
4.7 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) — próbkowanie dwufazowe	55
4.8 Wykres miary max_go_path_len dla próbkowania dwufazowego — przykład miary nieprzewidywalnej	56
4.9 Wykres miary num_sources dla próbkowania snowball — przykład miary o wartościach zmieniających się w przewidywalny sposób	57
4.10 Wykres miary num_sources dla instancji średniej wielkości	58
4.11 Wykres miary max_out_degree dla próbkowania dwufazowego	59
4.12 Wykres zależności liczby krawędzi od liczby wywołań funkcji celu. Kolor niebieski - próbkowanie snowball, kolor pomarańczowy — próbkowanie dwufazowe. Oś X w skali logarytmicznej	60
A.1 Liczba krawędzi (edge_count) w zależności od liczby wierzchołków	70
A.2 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) w zależności od liczby wierzchołków	71
A.3 Współczynnik różnorodności grafu (assortativity_deg) w zależności od liczby wierzchołków	72
A.4 Średnia wartość funkcji celu w znalezionych optimach lokalnych (avg_fitness) w zależności od liczby wierzchołków	73
A.5 Współczynnik conrel w zależności od liczby wierzchołków	74
A.6 Gęstość grafu (density) w zależności od liczby wierzchołków	75
A.7 Współczynnik distLO w zależności od liczby wierzchołków	76
A.8 Współczynnik wzajemności grafu (reciprocity) w zależności od liczby wierzchołków	77
A.9 Rozmiar największej kliki w grafie (largest_clique_size) w zależności od liczby wierzchołków	78
A.10 Liczba źródeł w grafie (num_sources) w zależności od liczby wierzchołków	79

A.11 Liczba ścieków w grafie (num_sinks) w zależności od liczby wierzchołków	80
A.12 Liczba subsinks w grafie (num_sub sinks) w zależności od liczby wierzchołków	81
A.13 Średni stopień wchodzący wierzchołków w grafie (avg_in_degree) w zależności od liczby wierzchołków	82
A.14 Maksymalny stopień wchodzący wśród wierzchołków w grafie (max_in_degree) w zależności od liczby wierzchołków	83
A.15 Średni stopień wychodzący wśród wierzchołków w grafie (avg_out_degree) w zależności od liczby wierzchołków	84
A.16 Maksymalny stopień wychodzący wierzchołków w grafie (max_out_degree) w zależności od liczby wierzchołków	85
A.17 Średnia waga pętli w grafie (avg_loop_weight) w zależności od liczby wierzchołków	86
A.18 Średnia długość ścieżki w grafie (avg_path_len) w zależności od liczby wierzchołków	87
A.19 Średnia długość istniejących ścieżek do globalnego optimum (avg_go_path_len) w zależności od liczby wierzchołków	88
A.20 Długość najdłuższej istniejącej ścieżki do globalnego optimum (max_go_path_len) w zależności od liczby wierzchołków	89
A.21 Stosunek liczby wierzchołków, z których istnieje ścieżka do globalnego optimum do liczby wszystkich wierzchołków (go_path_ratio) w zależności od liczby wierzchołków	90
A.22 Rozmiar największego leja w przestrzeni (max_funnel_size) w zależności od liczby wierzchołków	91
A.23 Średni rozmiar lejów w przestrzeni (mean_funnel_size) w zależności od liczby wierzchołków	92
A.24 Liczba spójnych podgrafów (num_cc) w zależności od liczby wierzchołków	93
A.25 Rozmiar największego spójnego podgrafa (largest_cc) w zależności od liczby wierzchołków	94
A.26 Promień największego spójnego podgrafa (largest_cc_radius) w zależności od liczby wierzchołków	95
A.27 Liczba nieudanych prób tworzenia krawędzi w zależności od liczby wierzchołków — próbkowanie dwufazowe	96
A.28 Liczba krawędzi (edge_count) w zależności od liczby wierzchołków — próbkowanie snowball	97
A.29 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) w zależności od liczby wierzchołków — próbkowanie snowball	98
A.30 Współczynnik różnorodności grafu (assortativity_deg) w zależności od liczby wierzchołków — próbkowanie snowball	99
A.31 Średnia wartość funkcji celu w znalezionych optimach lokalnych (avg_fitness) w zależności od liczby wierzchołków — próbkowanie snowball	100
A.32 Współczynnik conrel w zależności od liczby wierzchołków — próbkowanie snowball	101
A.33 Gęstość grafu (density) w zależności od liczby wierzchołków — próbkowanie snowball	102
A.34 Współczynnik distLO w zależności od liczby wierzchołków — próbkowanie snowball	103
A.35 Współczynnik wzajemności grafu (reciprocity) w zależności od liczby wierzchołków — próbkowanie snowball	104

A.36 Rozmiar największej kliki w grafie (largest_clique_size) w zależności od liczby wierzchołków — próbkowanie snowball	105
A.37 Liczba źródeł w grafie (num_sources) w zależności od liczby wierzchołków — próbkowanie snowball	106
A.38 Liczba ścieków w grafie (num_sinks) w zależności od liczby wierzchołków — próbkowanie snowball	107
A.39 Liczba subsinks w grafie (num_sub sinks) w zależności od liczby wierzchołków — próbkowanie snowball	108
A.40 Średni stopień wchodzący wierzchołków w grafie (avg_in_degree) w zależności od liczby wierzchołków — próbkowanie snowball	109
A.41 Maksymalny stopień wchodzący wśród wierzchołków w grafie (max_in_degree) w zależności od liczby wierzchołków — próbkowanie snowball	110
A.42 Średni stopień wychodzący wśród wierzchołków w grafie (avg_out_degree) w zależności od liczby wierzchołków — próbkowanie snowball	111
A.43 Maksymalny stopień wychodzący wierzchołków w grafie (max_out_degree) w zależności od liczby wierzchołków — próbkowanie snowball	112
A.44 Średnia waga pętli w grafie (avg_loop_weight) w zależności od liczby wierzchołków — próbkowanie snowball	113
A.45 Średnia długość ścieżki w grafie (avg_path_len) w zależności od liczby wierzchołków — próbkowanie snowball	114
A.46 Średnia długość istniejących ścieżek do globalnego optimum (avg_go_path_len) w zależności od liczby wierzchołków — próbkowanie snowball	115
A.47 Długość najdłuższej istniejącej ścieżki do globalnego optimum (max_go_path_len) w zależności od liczby wierzchołków — próbkowanie snowball	116
A.48 Stosunek liczby wierzchołków, z których istnieje ścieżka do globalnego optimum do liczby wszystkich wierzchołków (go_path_ratio) w zależności od liczby wierzchołków — próbkowanie snowball	117
A.49 Rozmiar największego leja w przestrzeni (max_funnel_size) w zależności od liczby wierzchołków — próbkowanie snowball	118
A.50 Średni rozmiar lejów w przestrzeni (mean_funnel_size) w zależności od liczby wierzchołków — próbkowanie snowball	119
A.51 Liczba spójnych podgrafów (num_cc) w zależności od liczby wierzchołków — próbkowanie snowball	120
A.52 Rozmiar największego spójnego podgrafa (largest_cc) w zależności od liczby wierzchołków — próbkowanie snowball	121
A.53 Promień największego spójnego podgrafa (largest_cc_radius) w zależności od liczby wierzchołków — próbkowanie snowball	122
A.54 Liczba krawędzi (edge_count) w zależności od liczby wierzchołków — próbkowanie dwufazowe	123
A.55 Stosunek liczby krawędzi do liczby wierzchołków (edge_to_node) w zależności od liczby wierzchołków — próbkowanie dwufazowe	124
A.56 Współczynnik różnorodności grafu (assortativity_deg) w zależności od liczby wierzchołków — próbkowanie dwufazowe	125
A.57 Średnia wartość funkcji celu w znalezionych optimach lokalnych (avg_fitness) w zależności od liczby wierzchołków — próbkowanie dwufazowe	126
A.58 Współczynnik conrel w zależności od liczby wierzchołków — próbkowanie dwufazowe	127
A.59 Gęstość grafu (density) w zależności od liczby wierzchołków — próbkowanie dwufazowe	128

A.60 Współczynnik distLO w zależności od liczby wierzchołków — próbkowanie dwufazowe	129
A.61 Współczynnik wzajemności grafu (reciprocity) w zależności od liczby wierzchołków — próbkowanie dwufazowe	130
A.62 Rozmiar największej kliki w grafie (largest_clique_size) w zależności od liczby wierzchołków — próbkowanie dwufazowe	131
A.63 Liczba źródeł w grafie (num_sources) w zależności od liczby wierzchołków — próbkowanie dwufazowe	132
A.64 Liczba ścieków w grafie (num_sinks) w zależności od liczby wierzchołków — próbkowanie dwufazowe	133
A.65 Liczba subsinks w grafie (num_sub sinks) w zależności od liczby wierzchołków — próbkowanie dwufazowe	134
A.66 Średni stopień wchodzący wierzchołków w grafie (avg_in_degree) w zależności od liczby wierzchołków — próbkowanie dwufazowe	135
A.67 Maksymalny stopień wchodzący wśród wierzchołków w grafie (max_in_degree) w zależności od liczby wierzchołków — próbkowanie dwufazowe	136
A.68 Średni stopień wychodzący wśród wierzchołków w grafie (avg_out_degree) w zależności od liczby wierzchołków — próbkowanie dwufazowe	137
A.69 Maksymalny stopień wychodzący wierzchołków w grafie (max_out_degree) w zależności od liczby wierzchołków — próbkowanie dwufazowe	138
A.70 Średnia waga pętli w grafie (avg_loop_weight) w zależności od liczby wierzchołków — próbkowanie dwufazowe	139
A.71 Średnia długość ścieżki w grafie (avg_path_len) w zależności od liczby wierzchołków — próbkowanie dwufazowe	140
A.72 Średnia długość istniejących ścieżek do globalnego optimum (avg_go_path_len) w zależności od liczby wierzchołków — próbkowanie dwufazowe	141
A.73 Długość najdłuższej istniejącej ścieżki do globalnego optimum (max_go_path_len) w zależności od liczby wierzchołków — próbkowanie dwufazowe	142
A.74 Stosunek liczby wierzchołków, z których istnieje ścieżka do globalnego optimum do liczby wszystkich wierzchołków (go_path_ratio) w zależności od liczby wierzchołków — próbkowanie dwufazowe	143
A.75 Rozmiar największego leja w przestrzeni (max_funnel_size) w zależności od liczby wierzchołków — próbkowanie dwufazowe	144
A.76 Średni rozmiar lejów w przestrzeni (mean_funnel_size) w zależności od liczby wierzchołków — próbkowanie dwufazowe	145
A.77 Liczba spójnych podgrafów (num_cc) w zależności od liczby wierzchołków — próbkowanie dwufazowe	146
A.78 Rozmiar największego spójnego podgrafa (largest_cc) w zależności od liczby wierzchołków — próbkowanie dwufazowe	147
A.79 Promień największego spójnego podgrafa (largest_cc_radius) w zależności od liczby wierzchołków — próbkowanie dwufazowe	148
A.80 Liczba nieudanych prób tworzenia krawędzi w zależności od liczby wierzchołków — próbkowanie dwufazowe	149
A.81 Wykres zależności liczby wierzchołków od liczby wywołań funkcji celu. Kolor niebieski - próbkowanie snowball, kolor pomarańczowy — próbkowanie dwufazowe. Oś X w skali logarytmicznej	150
A.82 Wykres zależności liczby krawędzi od liczby wywołań funkcji celu. Kolor niebieski - próbkowanie snowball, kolor pomarańczowy — próbkowanie dwufazowe. Oś X w skali logarytmicznej	151

Spis tabel

4.1 Instancja uniform_7 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	26
4.2 Instancja uniform_7 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	26
4.3 Instancja cliques_7 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	27
4.4 Instancja cliques_7 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	27
4.5 Instancja grid_7 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	28
4.6 Instancja grid_7 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	28
4.7 Instancja uniform_8 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	29
4.8 Instancja uniform_8 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	29
4.9 Instancja cliques_8 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	30
4.10 Instancja cliques_8 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	30
4.11 Instancja grid_8 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	31
4.12 Instancja grid_8 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	31
4.13 Instancja uniform_9 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	32
4.14 Instancja uniform_9 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	32
4.15 Instancja cliques_9 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	33
4.16 Instancja cliques_9 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	33
4.17 Instancja grid_9 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	34
4.18 Instancja grid_9 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	34
4.19 Instancja uniform_10 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	35

4.20 Instancja uniform_10 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	35
4.21 Instancja cliques_10 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	36
4.22 Instancja cliques_10 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	36
4.23 Instancja grid_10 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	37
4.24 Instancja grid_10 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	37
4.25 Instancja uniform_11 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	38
4.26 Instancja uniform_11 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	38
4.27 Instancja cliques_11 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	39
4.28 Instancja cliques_11 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	39
4.29 Instancja grid_11 - porównanie wartości miar uzyskanych z próbkowania snowball z wartościami z przeglądu zupełnego.	40
4.30 Instancja grid_11 - porównanie wartości miar uzyskanych z próbkowania dwufazowego z wartościami z przeglądu zupełnego.	40
4.31 Uśredniona wartość korelacji z innymi metrykami dla każdej metryki	51