Analiza algorytmów. Lista 4

Piotr Berezowski, 236749

15 maja 2020

1 Implementacja zadań

Implementacja zadania została wykonana w języku *Julia* w wersji 1.4.1. Do uruchomienia skryptu z zadaniem wymagane jest doinstalowanie pakietu *Plots* odpowiedzialnego za rysowanie wykresów.

Załączone pliki:

- *zad11.jl* - zawiera implementację symulatora dla podpunktu b zadania, funkcje obliczające $P(n, q)$ oraz funkcje rysujące wykresy.

Uruchomienie skryptu poleceniem *julia zad11.jl* powinno stworzyć pliki zawierające wszystkie wykresy przedstawione w sprawozadniu.

2 Zadanie 11

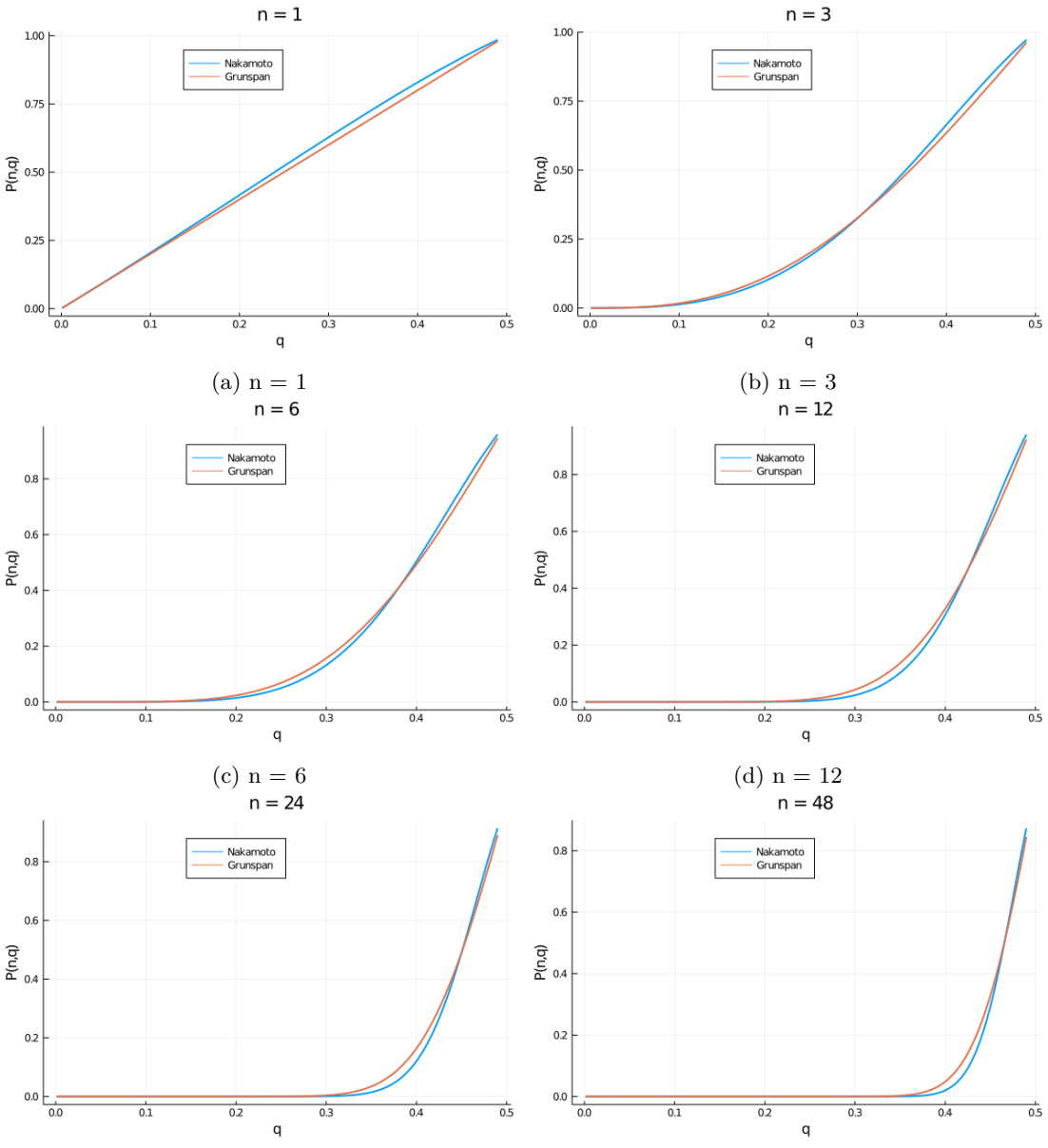
2.1 Opis zadania

Przeczytaj notatki do wykładu. Niech $0 < q < 1/2$ oznacza prawdopodobieństwo wydobycia kolejnego bloku przez adwersarza odpowiadające części mocy obliczeniowej będącej w jego posiadaniu. Niech n oznacza liczbę potwierdzeń (nadbudowanych bloków) potrzebnych by uznać transakcję za potwierdzoną. Niech $P(n, q)$ oznacza prawdopodobieństwo, że adwersarz o mocy q będzie dysponował łańcuchem bloków równym lub dłuższym niż ten budowany przez uczciwych użytkowników w momencie, gdy nadbudowali oni blok zawierający rozważaną transakcję n blokami lub kiedykolwiek później.

- Porównaj formuły na $P(n, q)$ uzyskane przez Nakamato i Grunspana. W szczególności:
  - ustal $n = 1, 3, 6, 12, 24, 48$ i przedstaw wykresy $P(n, q)$ w zależności od wartości q,
  - ustal dopuszczalne prawd. sukcesu adwersarza $P(n, q) = 0.1\%, 1\%, 10\%$ i narysuj wykresy przedstawiające jak należy dobrać wartość n w zależności od wartości q.
- Zaimplementuj symulator ataku „double spending", który umożliwi eksperymentalne przybliżenie prawdopodobieństwa zdarzenia $P(n, q)$ w zależności od wartościni q. Wskazówka: zaprojektuj eksperyment i powtórz go wielokrotnie (Metoda Monte Carlo). W raporcie starannie i dokładnie opisz ideę działania i kod symulatora.
- Porównaj wyniki symulatora do wyników analitycznych (wykresy). Jeśli pojawią się rozbieżności postaraj się je wyjaśnić.

2.2 Rozwiązanie

2.2.1 Porównanie formuł uzyskanych przez Nakamoto i Grunspana dla poszczególnych wartości n



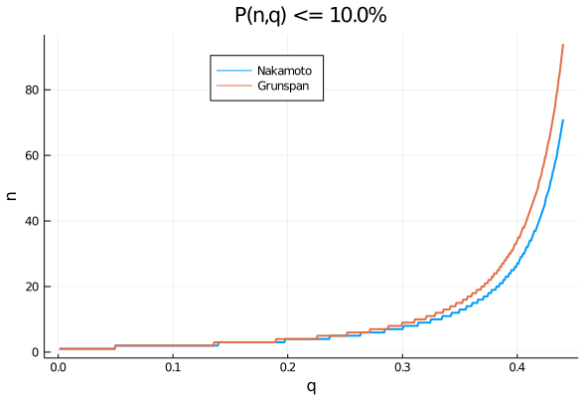
(a) n = 1 (b) n = 3

(c) n = 6 (d) n = 12

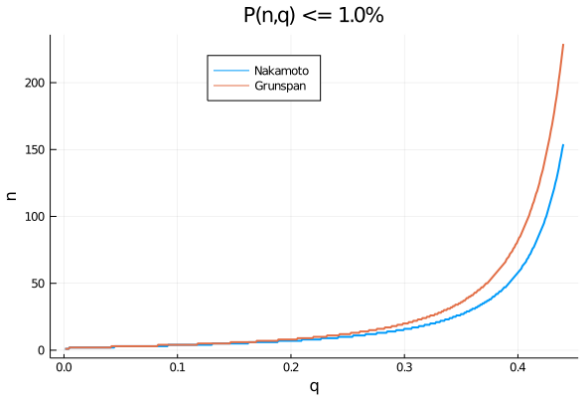
(e) n = 24 (f) n = 48

Rysunek 1: $P(n, q)$ w zależności od q dla różnych wartości n.

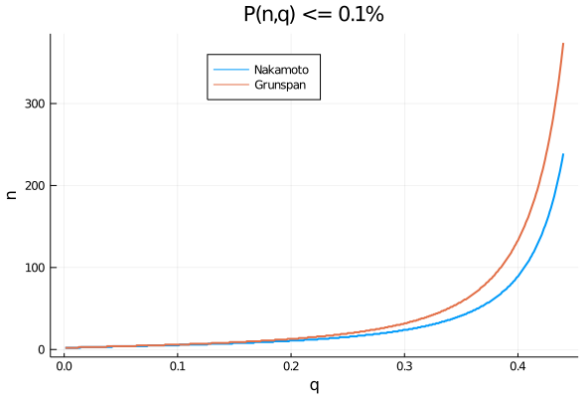
2.2.2 Porównanie wartości n spełniających dopuszczalną wartość $P(n, q)$ obliczaną przez obie formuły.



(a) $P(n, q) \leq 10\%$



(b) $P(n, q) \leq 1\%$



(c) $P(n, q) \leq 0.1\%$

Rysunek 2: Wartość n w zależności od q dla różnych dopuszczalnych wartości $P(n, q)$.

2.3 Symulator ataku „double spending”

1: doubleSpendingSimulation(n, q)

Input : n, q
Output: $P(n, q)$ - prawdopodobieństwa zrównania się gałęzi adwersarza z najdłuższą

```
1  advBlocks ← 0;
2  usrBlocks ← 0;
3  while usrBlocks < n do
4  |  r ← rand([0, 1));
5  |  if r < q then
6  |  |  advBlocks ← advBlocks + 1;
7  |  else
8  |  |  usrBlocks ← usrBlocks + 1;
9  if advBlocks ≥ usrBlocks then
10 |  return 1
11 else
12 |  return (q/(1−q))^(usrBlocks−advBlocks)
```

Powyższy pseudokod pokazuje w jaki sposób został zaimplementowany symulator ataku „double spending”.

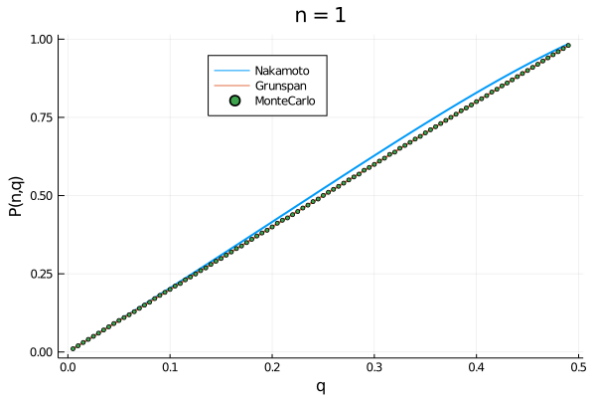
Sytuacja początkowa prezentuje się w ten sposób, że adwersarz zaczyna budować własną gałąź łańcucha zaczynając od ostatniego bloku najdłuższego istniejącego łańcucha. W takiej sytuacji początkowa długość nowej „uczciwej” gałęzi jak i gałęzi adwersarza jest równa 0. Następnie, dopóki gałąź uczciwych użytkowników nie osiągnie długości n, obie gałęzie budowane są w następujący sposób:

- losujemy liczbę z zakresu $[0, 1)$
- jeśli wylosowana liczba jest mniejsza niż q wydłużamy gałąź adwersarza
- w przeciwnym wypadku wydłużamy gałąź uczciwych użytkowników

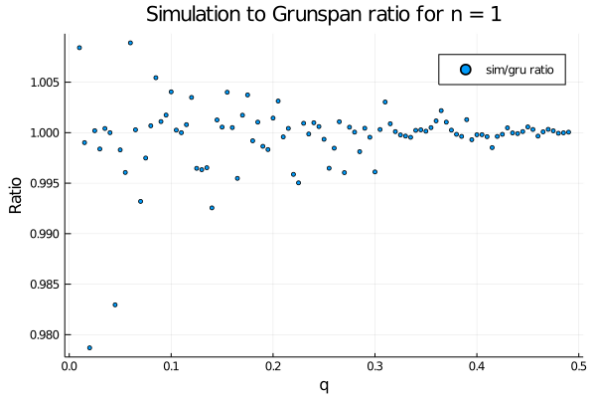
Kiedy długość gałęzi uczciwych użytkowników osiągnie długość równą n znajdujemy się w jednej z dwóch sytuacji:

- gałąź adwersarza jest dłuższa niż n - w tym przypadku zwracamy 1, ponieważ zdarzenie zostało spełnione
- gałąź adwersarza jest krótsza niż n i ma długość k - w tym przypadku zwracamy prawdopodobieństwo zrównania się gałęzi adwersarza z gałęzią uczciwych użytkowników równe $(\frac{q}{p})^{n-k}$

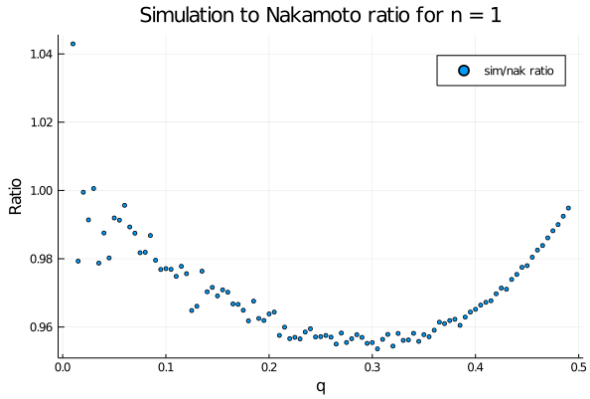
2.4 Porównanie wyników symulatora i wyników analitycznych



(a) $P(n, q)$ otrzymane z symulacji w porównaniu do formuł Nakamoto i Grunspana.

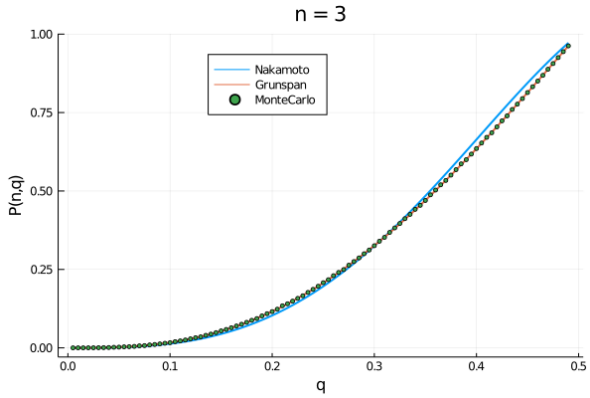


(b) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Grunspana.

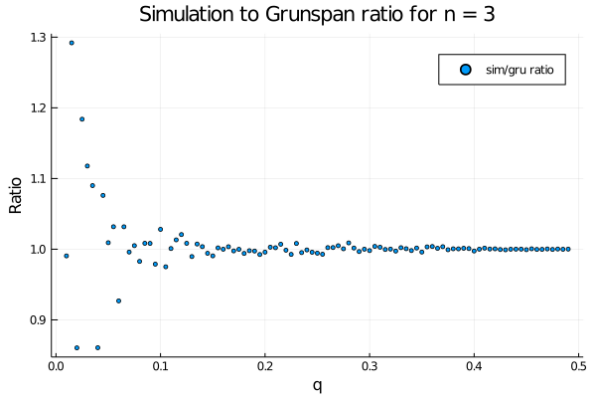


(c) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Nakamato.

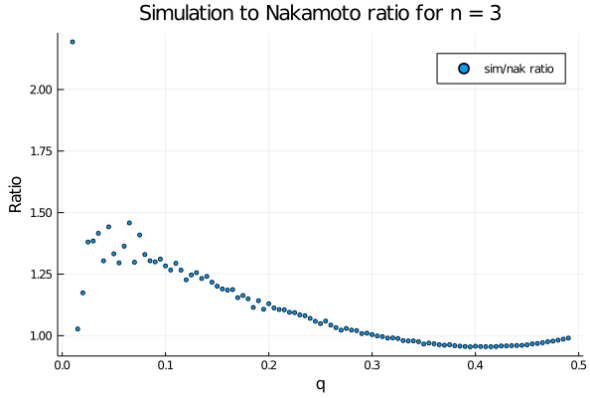
Rysunek 3: Wyniki dla wartości $n = 1$



(a) $P(n, q)$ otrzymane z symulacji w porównaniu do formuł Nakamoto i Grunspana.

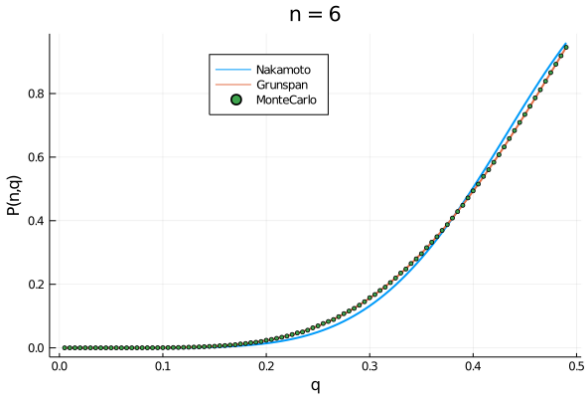


(b) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Grunspana.

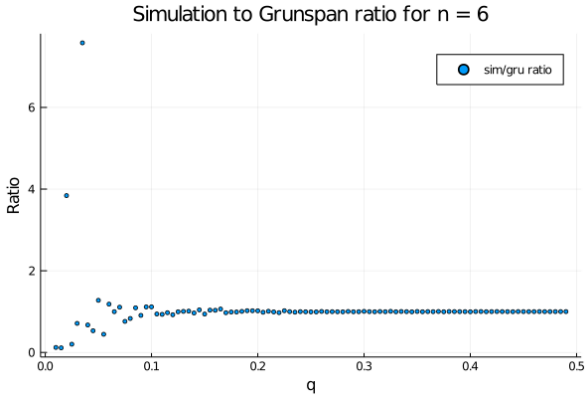


(c) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Nakamato.

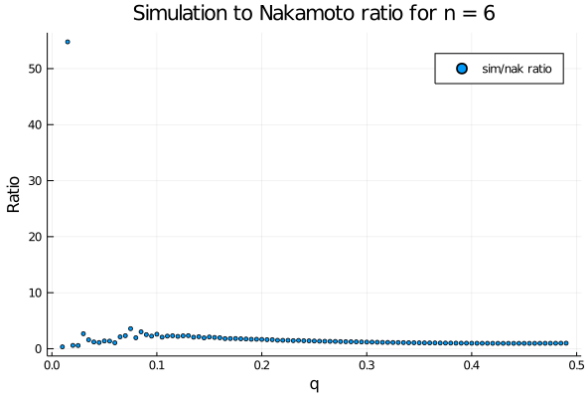
Rysunek 4: Wyniki dla wartości $n = 3$



(a) $P(n, q)$ otrzymane z symulacji w porównaniu do formuł Nakamoto i Grunspana.

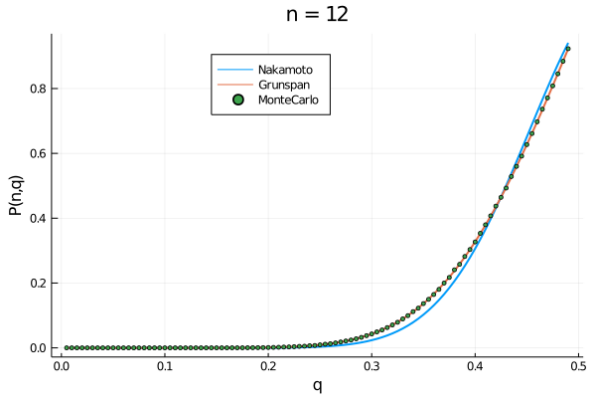


(b) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Grunspana.

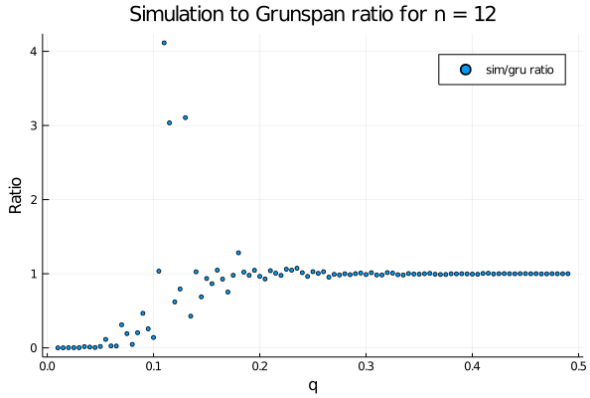


(c) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Nakamato.

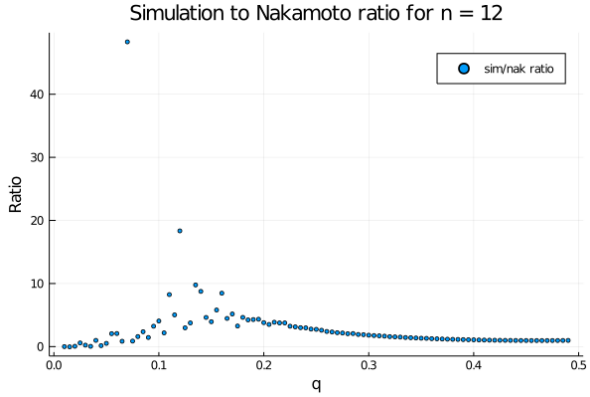
Rysunek 5: Wyniki dla wartości $n = 6$



(a) $P(n, q)$ otrzymane z symulacji w porównaniu do formuł Nakamoto i Grunspana.

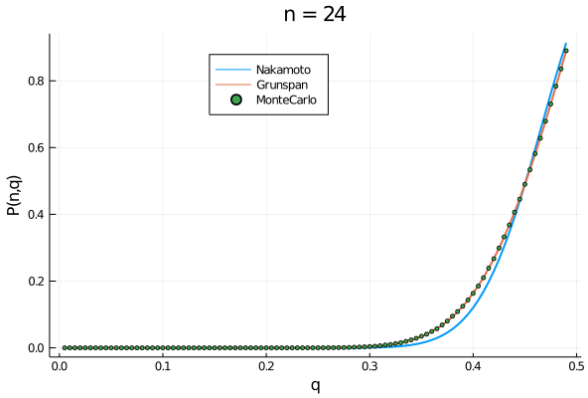


(b) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Grunspana.

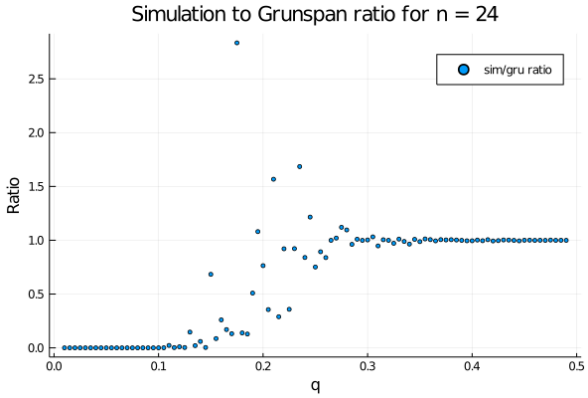


(c) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Nakamato.

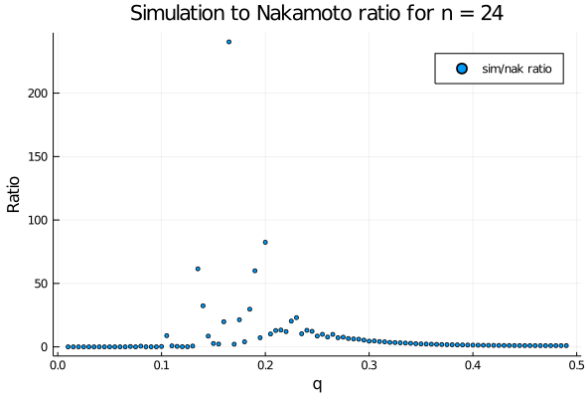
Rysunek 6: Wyniki dla wartości $n = 12$



(a) $P(n, q)$ otrzymane z symulacji w porównaniu do formuł Nakamoto i Grunspana.

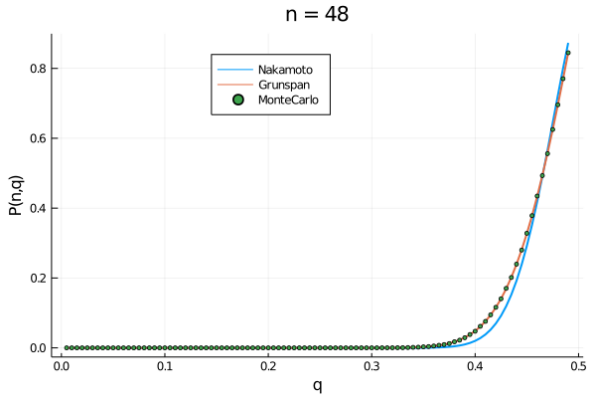


(b) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Grunspana.

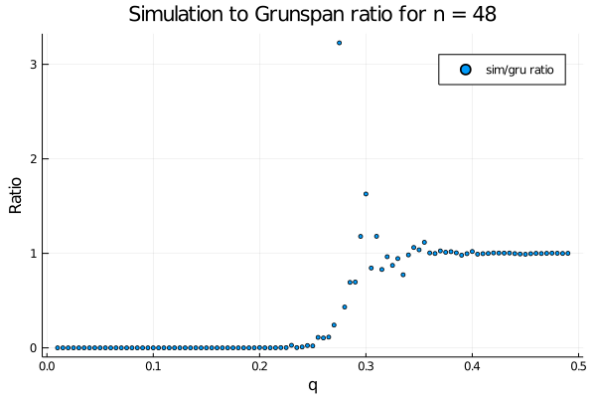


(c) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Nakamato.

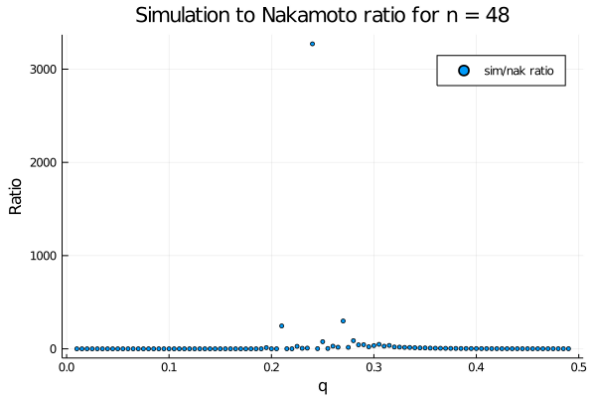
Rysunek 7: Wyniki dla wartości $n = 24$



(a) $P(n, q)$ otrzymane z symulacji w porównaniu do formuł Nakamoto i Grunspana.



(b) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Grunspana.



(c) Stosunek $P(n, q)$ otrzymanego w symulacji do formuły Nakamato.

Rysunek 8: Wyniki dla wartości $n = 48$