Analiza Algorytmów, Lista 4 Raport

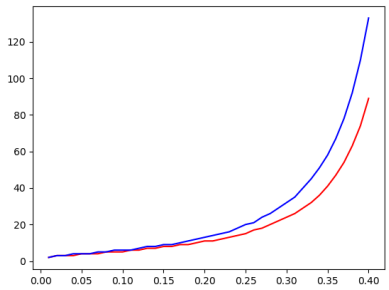
Tomasz Krent

May 15, 2020

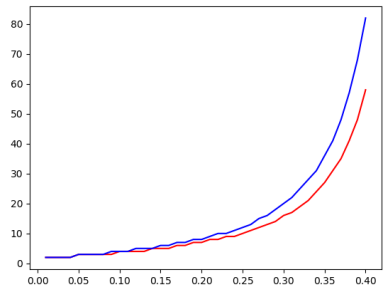
1 Zadanie 11

(a) Funkcje obliczajace prawdopodobieństwo udanego ataku adwersarza uzyskane dzieki formułom Nakamoto oraz Grunspana znajduja sie w pliku *formuly.py*.

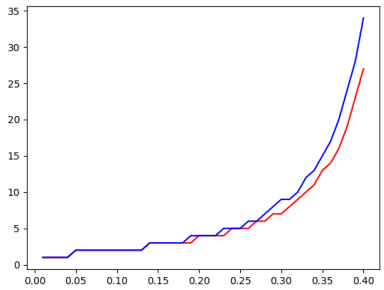
Wyniki działania programu *zad*11_*a*2.*py* sprawdzajacego jak należy dobrać wartość n w zależności od wartości q ustalajace dopuszczalne prawdopodobieństwo sukcesu dla różnych wartości α obliczonych za pomoca formuł uzyskanych przez Nakamoto (wykres zaznaczony czerwona linia) oraz Grunspana (wykres zaznaczony niebieska linia):



(a) $\alpha = 0.1\%$

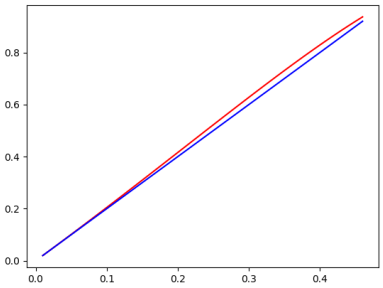


(b) $\alpha = 10\%$

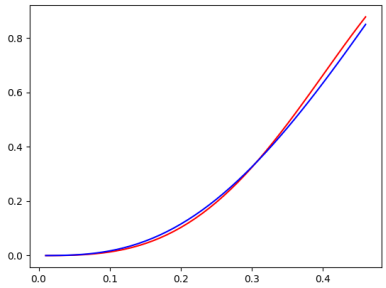


(c) $\alpha = 10\%$

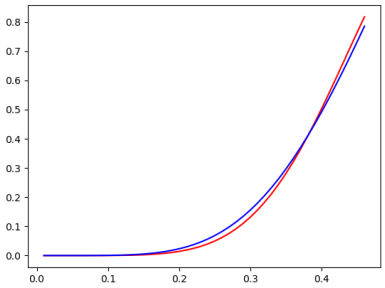
Wyniki działania programu *zad*11_*a*1.*py* przedstawiajace wykresy $P(n, q)$ w zależności od wartości q dla różnych $n = 1, 3, 6, 12, 24, 48$ obliczonych za pomoca formuł uzyskanych przez Nakamoto (wykres zaznaczony czerwona linia) oraz Grunspana (wykres zaznaczony niebieska linia):



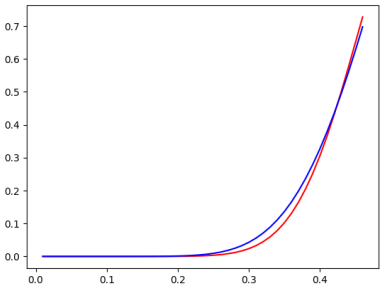
(a) $n = 1$



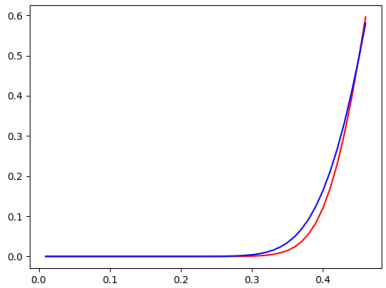
(b) $n = 3$



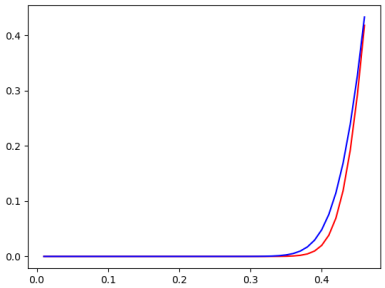
(c) $n = 6$



(d) $n = 12$



(e) $n = 24$



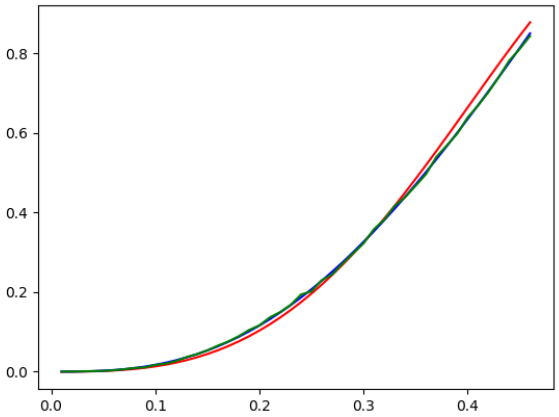
(f) $n = 48$

Z wykresów można wywnioskować, że wraz ze zwiekszaniem sie parametru n prawdopodobieństwo udanego ataku przez adwersarza sie zmniejsza. Widać również, że w momencie, w którym moc obliczeniowa adwersarza jest ,,zbliża sie" do mocy obliczeniowej uczciwego użytkownika to prawdopodobieństwo odniesienia sukcesu drastycznie sie zwieksza.

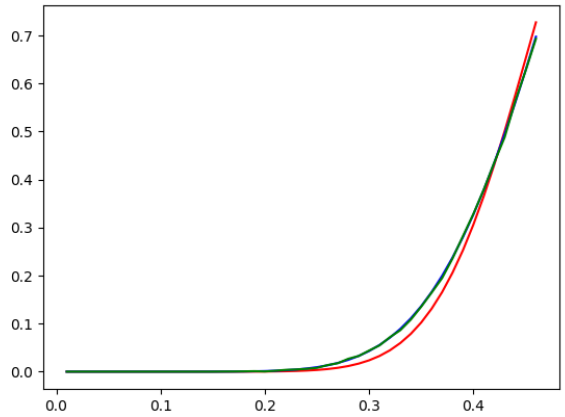
(b) W programie *sym.py* zaimplementowana został funkcja symulujaca ataku ,,double spending", który umożliwia eksperymentalnie sprawdzenie prawdopodobieństwa $P(n, q)$. Funkcja w swoim działaniu symuluje każdy krok w taki sposób, że losowane sa liczby z zakresu od 0 do 1, dzieki którym sprawdzane jest zdarzenie wydobycia kolejnego bloku przez adwersarza (jeśli wylosowana liczba dla adwersarza jest mniejsza od q) oraz przez uczciwych użytkowników (jeśli wylosowana liczba dal uczciwego użytkownika jest mniejsza niż $1 - q$). Po wykonaniu n kroków sprawdzane jest zdarzenie takie, że adwersarz ma tyle samo lub wiecej wydobytych bloków niż uczciwy użytkownik. Jeśli tak to atak sie powiódł, w przeciwnym przypadku wykonywane sa kolejne kroki i po każdym kroku sprawdzane jest czy adwersarz ma tyle samo wydobytych bloków co uczciwy użytkownik, jeśli tak to atak sie powiódł. Jeśli adwersarz nie ,,dogoni" uczciwego użytkownika do pewnego określonego momentu (np. przypadku działania programu po upływie 200 kroków), stwierdza sie, że atak sie nie powiódł.

Dla pojedynczych wartości n i q program *zad*11_*a*1.*py* wielokrotnie powtarza funkcje symulujaca atak w celu uzyskania przybliżonego prawdopodobieństwa powodzenia ataku przez adwersarza.

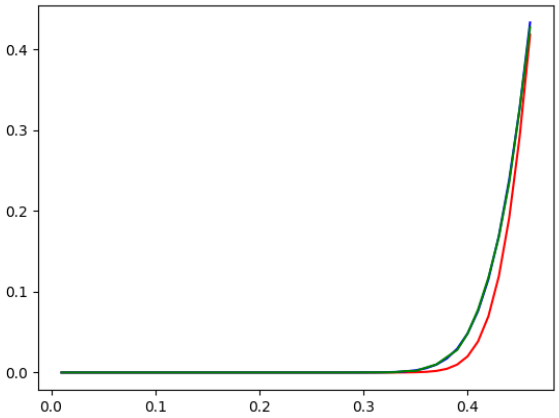
Wyniki działania programu $zad11_a1.py$ przedstawiajace wykresy $P(n, q)$ w zależności od wartości q dla różnych $n = 3, 12, 48$ obliczonych za pomoca formuł uzyskanych przez Nakamoto (wykres zaznaczony czerwona linia), Grunspana (wykres zaznaczony niebieska linia) oraz przez eksperyment wyznaczajacy prawdopodobieństwo za pomoca symulatora metoda ,,Monte carlo"(wykres zaznaczony czerwona linia) :



(a) $n = 3$



(b) $n = 12$



(c) $n = 48$

Wyniki uzyskana przy pomocy symulatora niemal pokrywaja sie z formuła prawdopodobieństwo Grunspana, czyli formuła w dokładny sposób przybliża prawdopodobieństwo. Formuła Nakamoto natomiast w swojej formie przyjmuje uproszczenie, że faktyczny czas wydobywania bloków równy wartości oczekiwanej czasu wydobycia tych bloków przez co nie jest tak precyzyjna jak formuła Grunspana.

.