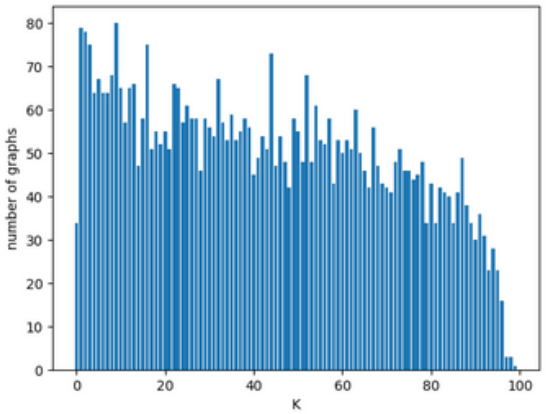
Mateusz Budzyński
(praca samodzielna)

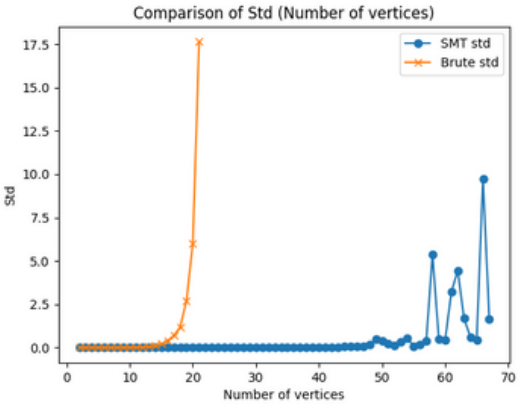
A) Grafy w u mnie generują się w konstruktorze to znaczy, przy tworzeniu grafu podajemy mu liczbę wierzchołków a potem losowo wybieramy mu liczbę krawędzi z legalnego zakresu. Gdy znamy już liczbę wierzchołków i krawędzi generowanie grafu to po prostu iterowanie tyle razy ile wynosi liczba krawędzi i losowanie jakiejś nie wybranej.

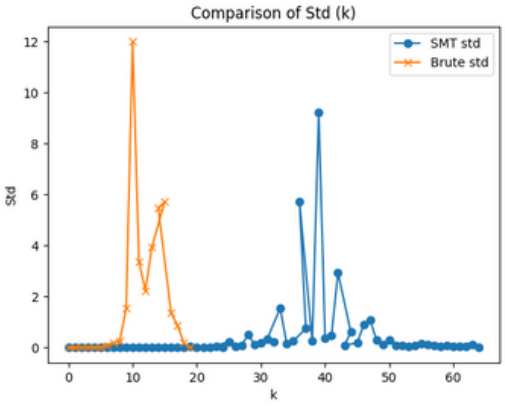
F) Funkcja vertex_cover_smt rozwiązuje problem Vertex Cover za pomocą solvera SMT (Satisfiability Modulo Theories). Tworzy kopię grafu, generuje zbiór zmiennych boolowskich dla każdego wierzchołka i dodaje ograniczenia, które zapewniają, że każda krawędź ma przynajmniej jeden z końców w pokryciu. Następnie dodaje warunek na liczność pokrycia, ograniczając liczbę wybranych wierzchołków do k. Solver SMT sprawdza, czy istnieje satysfakcjonujące rozwiązanie, a jeśli tak, zwraca wierzchołki wchodzące w skład pokrycia. W przeciwnym razie zwraca None. Wykorzystuje do tego bibliotekę z3.

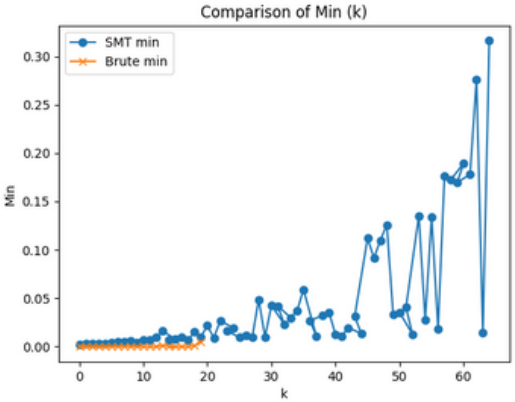
I) Smt potrafi przetworzyć grafy nawet o 106 wierzchołkach, Gdzie brute średnio zacina się już przy 23 chodź jak wynika z jego działania przy dużej dozy szczęścia przetworzyłby nawet grafy takiej wielkości ile zbiorów byłby w stanie w dwie minuty wygenerować (od razu znalazłby rozwiązanie)

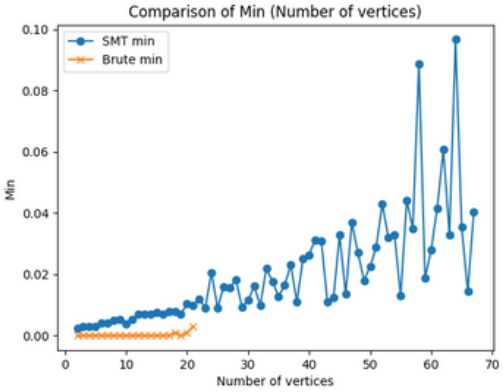


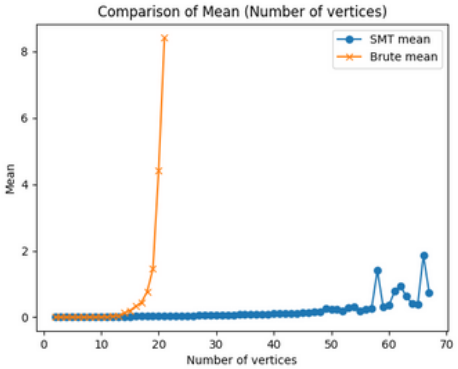
H

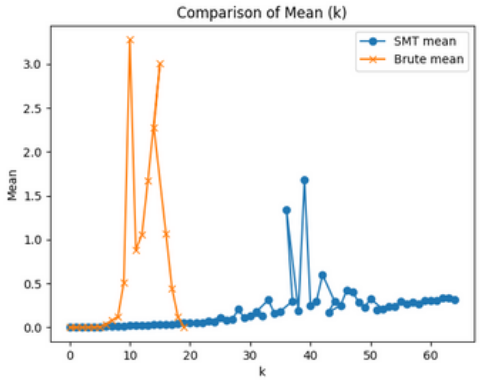


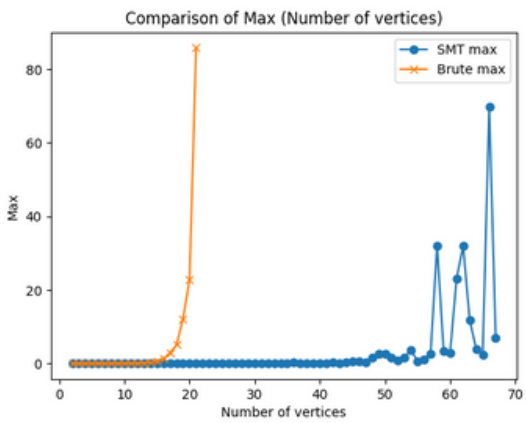


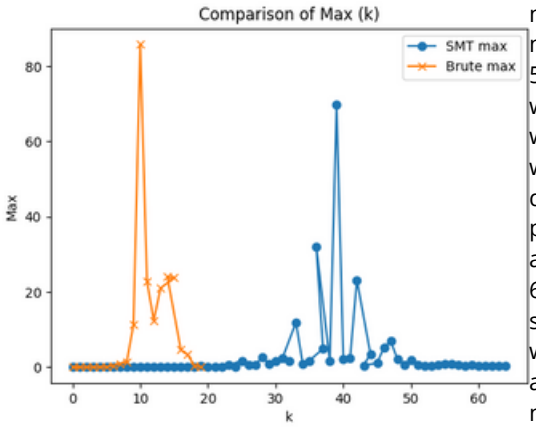












Opis eksperymentu

porównującego czasy działania. Wykresy zamieszczone po niżej opisują 4 statystyki, dwukrotnie tzn. raz zależne od liczby wierzchołków a raz zależne od liczby k. Te statystyki to minimalna wartość, maksymalna wartość, średnia i odchylenie standardowe. Grafy do eksperymentu zostały wygenerowane w następujący sposób: po 50 grafów dla danej liczby wierzchołków w zakresie od 2 do 101 czyli w sumie 5000 grafów. Ich histogram po K możemy zobaczyć po lewej stronie. Samo testowanie przebiegło w następujący sposób. Programy brute i smt przetwarzały posortowane grafy (po liczbie wierzchołków) do momentu przejrzenia wszystkich grafów lub upływu 10 min. Czasy przetworzenia konkretnego grafu były przechowywane w dwóch słownikach. Jeden słownik w kluczach miał liczbę k a w wartościach czasy wykonania dla tej liczby k, drugi w kluczach miał liczbę wierzchołków i analogiczne wartości. Na podstawie tych danych zostały wykonane wszystkie poniższe wykresy

wnioski z eksperymentu

Z porównań możemy wyciągnąć kilka bardzo ważnych wniosków mianowicie

1) smt działa znacznie szybciej od bruta

2) brut ma znacznie większe odchyleni standardowe (co jest naturalną konsekwencją jego działania tzn sprawdzamy wszystkie możliwości których jest bardzo dużo i rozwiązanie możemy znaleźć na samym początku lub końcu lub przejrzeć wszystkie i nie znaleźć wcale.

3) minimalne czasy są bliskie 0 (case gdy szybko znajdujemy rozwiązanie)

4) wykresy dla k charakteryzują się większym “szumem”, nieregularnością co wynika z faktu ,że dla danego k liczba wierzchołków grafu może być do niego bliska lub zdecydowanie od niego większa.

5) wraz z wzrostem k/n czasy wykonania rosną z tym ,że nie jest to wzrost regularny i nie koniecznie większe n będzie równało się dłuższemu wykonaniu co wynika np z pkt 2) co wynika z charakteru algorytmów i losowych próbek

6) wzrosty działania obu algorytmów są stosunkowo szybkie co w wynika z wykładniczego czasu działania obu algorytmów. (brute dużo szybciej bo nie jest w żaden sposób optymalizowany jak smt)