Mateusz Chechliński

Sławomir Płodczyk

Temat: Zastosowanie algorytmu APPROX-VERTEX-COVER do wyznaczania maksymalnej kliki w grafie

Dokumentacja końcowa

1 Opis problemu:

Dany jest graf nieskierowany G=(V,E). Rozwiązaniem problemu jest znalezienie takiego podzbioru V' zbioru wierzchołków V o maksymalnej liczności, dla którego dowolne 2 wierzchołki z podzbioru V' są połączone w grafie G krawędzią. Korzystamy z algorytmu aproksymacyjnego, zatem nie musi on znaleźć podzbioru V' o maksymalnej liczności, lecz tylko jego dolne przybliżenie.

2 Instrukcja obsługi

Po uruchomieniu programu dostępne jest okno programu głównego. Dostępne są 2 przyciski:

- Wczytaj plik – powoduje otwarcie okna, w którym jest możliwy wybór odpowiedniego pliku z zapisanym grafem (o rozszerzeniu .grph). Po wybraniu pliku dane grafu są wczytywane do programu. W przypadku błędnych danych wyświetlany jest odpowiedni komunikat.

- Rozwiąż – dla wczytanego grafu program rozwiązuje problem i zapisuje do pliku w tym samym miejscu co wczytywany plik, z tą samą nazwą i rozszerzeniem .clq. W przypadku braku wczytanych danych lub innych błędów wyświetlany jest odpowiedni komunikat.

3 Opis testów

3.1 Testy sprawdzające poprawność

Dla powyższego problemu zostało stworzone 14 przypadków testowych. Otrzymano następujące wyniki:

Przykład 1 – graf pełny o 8 wierzchołkach.
Program znalazł klikę o rozmiarze 8 równą rzeczywistej maksymalnej klice.

Przykład 2 – graf pełny o 14 wierzchołkach.
Program znalazł klikę o rozmiarze 14 równą rzeczywistej maksymalnej klice.

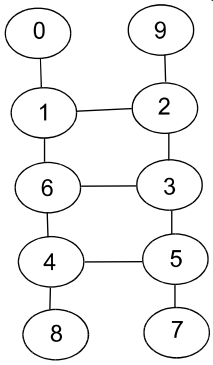
Przykład 3– graf pełny o 12 wierzchołkach z usuniętymi wszystkimi krawędziami łączącymi ze sobą wierzchołki o indeksach 0,1,2,3 (usunięte krawędzie tworzą klikę 4-wierzchołkową).
Program znalazł klikę o rozmiarze 8 (tworzą ją wierzchołki o indeksach: 4,5,..,11).
Maksymalna klika ma rozmiar 9 (tworzą ją np. wierzchołki o indeksach: 0,4,5,..,11).

Przykład 4– graf pełny o 18 wierzchołkach z usuniętymi wszystkimi krawędziami łączącymi ze sobą wierzchołki o indeksach 0,2,4,6,8,10 (usunięte krawędzie tworzą klikę 5-wierzchołkową).
Program znalazł klikę o rozmiarze 12 (tworzą ją wierzchołki o indeksach: 1,3,5,7,9,11,12,13,14,15,16,17). Maksymalna klika ma rozmiar 13 (tworzą ją np. wierzchołki o indeksach: 0,1,3,5,7,9,11,12,13,14,15,16,17).

Przykład 5 – graf pełny o 12 wierzchołkach z usuniętymi krawędziami: (0,1),(1,2),(2,3),(3,4),(4,5),(5,6),(6,7),(7,0)(usunięte krawędzie tworzą cykl z 8 wierzchołkami).
Program znalazł klikę o rozmiarze 4 (tworzą ją wierzchołki o indeksach:8,9,10,11).
Maksymalna klika ma rozmiar 8 (tworzą ją np. wierzchołki 0,2,4,6,8,9,10,11).

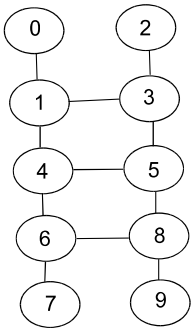
Przykład 6 – graf pełny o 9 wierzchołkach z usuniętymi krawędziami: (0,1),(1,8),(8,2),(2,3),(3,7),(7,5),(5,4),(4,6),(6,0)(usunięte krawędzie tworzą cykl z 9 wierzchołkami).
Program znalazł klikę o rozmiarze 3 (tworzą ją wierzchołki o indeksach:6,7,8). Maksymalna klika ma rozmiar 4 (tworzą ją np. wierzchołki 1,2,4,7).

Przykład 7 -graf pełny o 10 wierzchołkach z usuniętymi tymi krawędziami jak na rysunku:



Program znalazł klikę rozmiaru 4 (tworzą ją wierzchołki 6,7,8,9). Maksymalna klika ma rozmiar 5 (tworzą ją np. wierzchołki 0,2,5,6,8).

Przykład 8 - graf pełny o 10 wierzchołkach z usuniętymi tymi krawędziami jak na rysunku:



Program nie znalazł kliki (klika o rozmiarze 0). Maksymalna klika ma rozmiar 5 (tworzą ją np. wierzchołki 0,3,4,7,8).

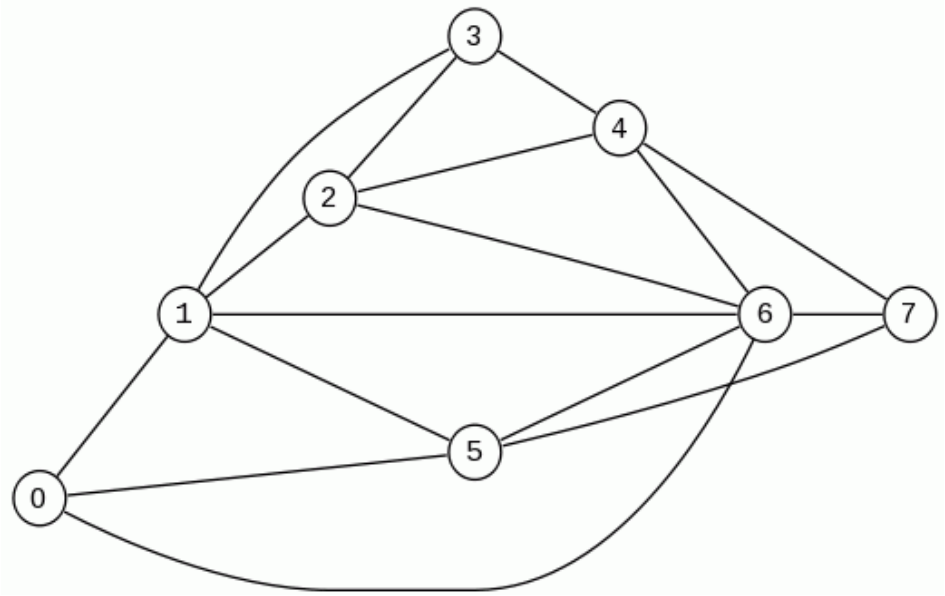
Przykład 9 – graf pełny o 10 wierzchołkach z usuniętymi krawędziami (0,1),(2,3),(4,5),(6,7),(8,9).

Program nie znalazł kliki (klika o rozmiarze 0). Maksymalna klika ma rozmiar 5 (tworzą ją np. wierzchołki o indeksach 0,2,4,6,8).

Przykład 10– graf pełny o 16 wierzchołkach z usuniętymi krawędziami (0,8),(1,9),(2,10),(3,11),(4,12),(5,13),(6,14),(7,15).

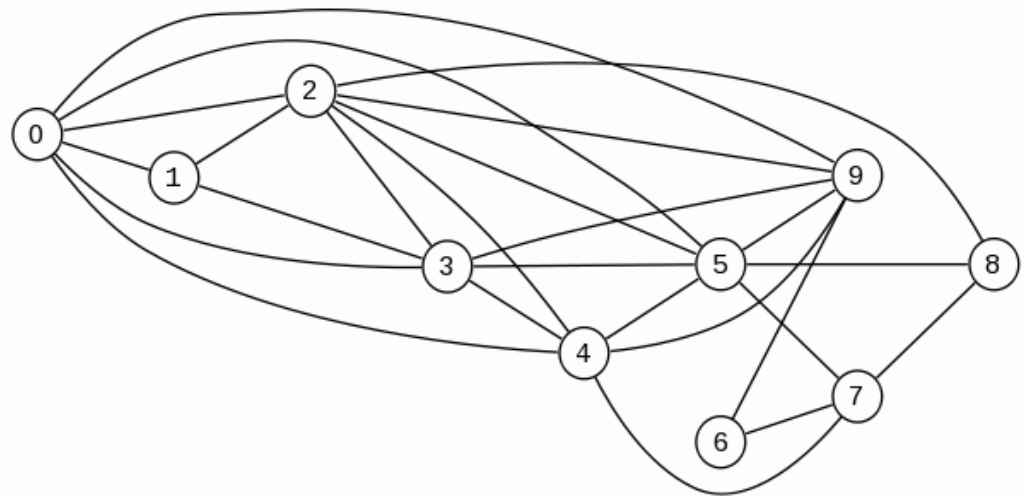
Program nie znalazł kliki (klika o rozmiarze 0). Maksymalna klika ma rozmiar 8 (tworzą ją np. wierzchołki o indeksach 0,1,2,3,4,5,6,7).

Przykład 11 – graf jak na rysunku:



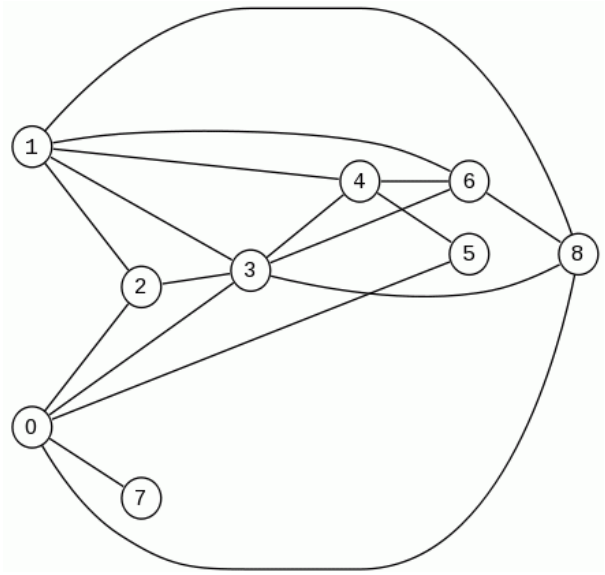
Program znalazł klikę rozmiaru 2(tworzą ją wierzchołki 6,7).Maksymalna klika ma rozmiar 4 (tworzą ją wierzchołki 0,1,5,6).

Przykład 12 – graf jak na rysunku:



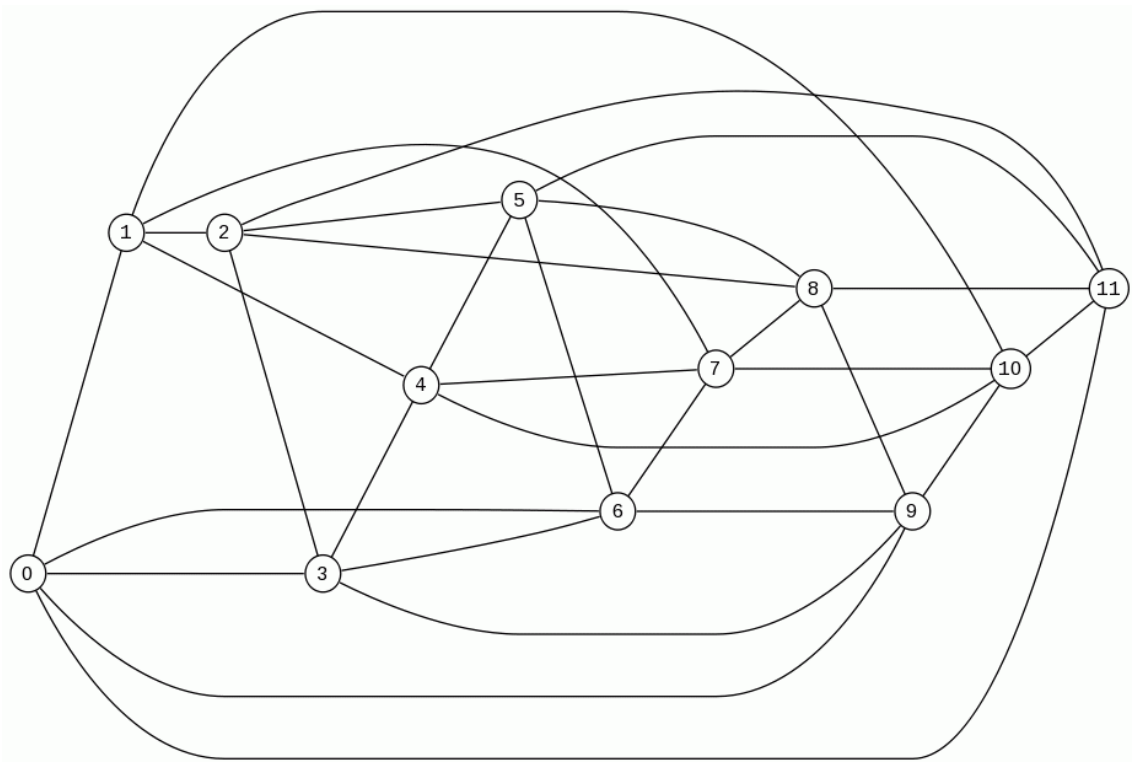
Program znalazł klikę rozmiaru 2(tworzą ją wierzchołki 5,9).Maksymalna klika ma rozmiar 6 (tworzą ją wierzchołki 0,2,3,4,5,9).

Przykład 13 – graf jak na rysunku:



Program znalazł klikę rozmiaru 1(tworzy ją wierzchołek 8).Maksymalna klika ma rozmiar 4 (tworzą ją wierzchołki 1,3,4,6).

Przykład 14 – graf jak na rysunku:



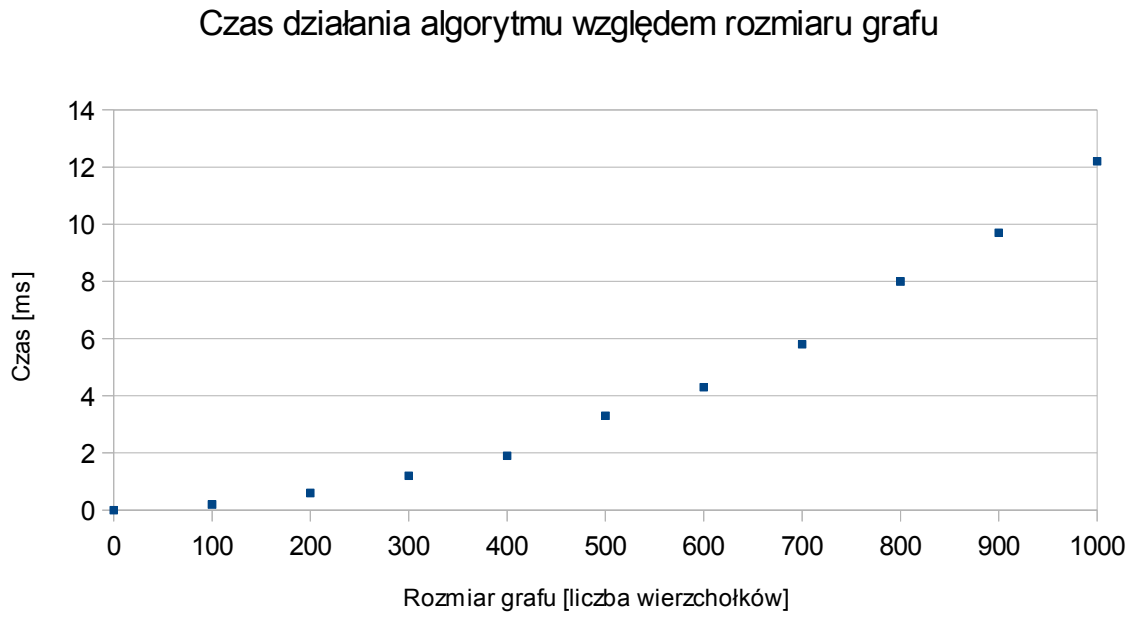
Program nie znalazł kliki .Maksymalna klika ma rozmiar 4 (tworzą ją wierzchołki 0,3,6,9).

Wizualizację przykładów 11-14 uzyskano za pomocą progamu ze strony http://www.algorytm.org/narzedzia/edytor-grafow.html

3.2 Testy wydajnościowe

Następnie zostały przeprowadzone testy wydajnościowe. W tym celu algorytm został uruchomiony dla grafów o dużej(100,200,...,1000) liczbie wierzchołków. Dla każdego z rozmiarów wygenerowano losowo po 10 grafów o różnej średniej gęstości krawędzi w grafie. Uzyskane czasy działania algorytmu uśredniono względem rozmiaru grafu. Uzyskano następujące wyniki:

Rozmiar grafu	Czas[ms]
100	0,2
200	0,6
300	1,2
400	1,9
500	3,3
600	4,3
700	5,8
800	8
900	9,7
1000	12,2



3.3 Wnioski

Analizując przytoczone przykłady możemy stwierdzić, że algorytm tylko w szczególnych przypadkach znajduje klikę równą maksymalnej lub jej bliską. W pewnych przypadkach (przykłady 5 i 6 oraz przykłady 7 i 8) różny sposób numeracji wierzchołków może spowodować otrzymanie wyników różnej jakości. Możemy zatem wyciągnąć stąd wniosek, że wybór odpowiedniej krawędzi w kroku algorytmu może spowodować otrzymanie wyników lepszej jakości. Może też to jednak spowodować komplikację algorytmu i nie zawsze objawi się to lepszymi wynikami (jak w przykładach 9 i 10). W przypadkach grafów bez szczególnej budowy (grafy 11-14) algorytm uzyskuje dość słabe wyniki.

Przeprowadzenie testów wydajnościowych pozwala stwierdzić, że obliczenia dla grafów o dużych rozmiarach nie wymagają długiego czasu. Związane jest to z tym, że złożoność tego algorytmu jest wielomianowa, w przeciwieństwie do algorytmu dokładnego o złożoności wykładniczej. Dla algorytmu dokładnego rozwiązywanie problemu o rozmiarze 100 mogłoby nie ukończyć się w rozsądnym czasie. Dodatkowo patrząc na wykres możemy utwierdzić się w przekonaniu (zgodnym z wcześniejszymi uzasadnieniami), że złożoność algorytmu aproksymacyjnego jest rzędu kwadratowego.