WSI - ćwiczenie 2

Agnieszka Wójtowicz

25 marca 2022

Spis treści

1 Polecenie oraz elementy języka

1.1 Polecenie Prowadzącego

Tematem drugich ćwiczeń są algorytmy genetyczne i ewolucyjne. Państwa zadaniem będzie zaimplementować klasyczny algorytm ewolucyjny bez krzyżowania, z selekcją turniejową i sukcesją generacyjną. W raporcie należałoby wskazać jak zmiana liczby osobników w populacji wpływa na jakość uzyskanych rozwiązań przy ograniczonym budżecie. Warto również opisać zachowanie algorytmu dla różnych rodzajów danych wejściowych oraz wpływ zmiany parametrów. Przykładowe zbiory danych i/lub ich generatory należy samemu skonstruować na potrzebę zadania.

Z powodu cięć budżetowych w Wolsce zmniejszono finansowanie policji tego wspaniałego kraju. W mieście X znajduje się n=25 placów (wierzchołki w grafie), z których można dotrzeć do sąsiadujących placów (istnieje krawędź w grafie). Policjant stojący na placu jest w stanie pokryć wszystkie krawędzie biegnące od tego placu do wszystkich jego sąsiadów. Gdzie powinniśmy umieścić policjantów by uzyskać jak największe pokrycie wszystkich ulic. (vertex cover problem)

1.2 Elementy języka

Wykorzystano język programowania Python w wersji 3.8. Skorzystano z pomocniczych bibliotek: *numpy 1.20.1*, *matplotlib 3.4.3*, *pandas 1.2.4*, *time*, *networkx 2.5*, *random*, pickle oraz (pomocniczo, w celu wyznaczenia rozwiązań metodą brutalną) *itertools 8.7.0*.

2 Cel zadania

Celem zadania jest implementacja algorytmu genetycznego rozwiązującego problem minimalnego pokrycia wierzchołkowego. Zbadane zostaną: liczba iteracji potrzebna do znalezienia minimum oraz jakość rozwiązania, a także wpływ parametrów takich jak wielkość populacji i prawdopodobieństwo mutacji na powyższe wyniki.

3 Przyjęte założenia

3.1 Problem

Rozwiązywane przez algorytm genetyczny problemy to grafy. Założenia przyjęte do generacji grafów były następujące:

- grafy były nieskierowane,
- grafy były spójne,
- koszty krawędzi grafów nie były rozważane, gdyż nie jest to istotne dla problemu,
- rozważano grafy pełne, pełne dwudzielne oraz losowe.

  Grafy były generowane przy pomocy biblioteki *networkx*.

3.2 Genom

Genom, członek populacji, reprezentowany jest przez tablice złożoną z zer i jedynek. Jest ona długości równej liczbie wierzchołków. Wartość 1 znajdująca się na indeksie tablicy odpowiadającemu danemu wierzchołkowi oznacza, że jest on włączany do problemu. Wartość 0 - nie jest włączany.

3.3 Populacja

Populacja początkowa generowana jest jeden raz, i składa się z zupełnie losowo wygenerowanych genomów. Przyjęto, że rozmiar populacji powinien być stały dla problemu o danym wymiarze N i wynoszący $8 * N$.

3.4 Selekcja

Stosowana jest selekcja turniejowa o stałym rozmiarze $k = 2$. Wykonywana jest tyle razy, by zachować rozmiar populacji.

3.5 Mutacja

Rozróżniane są dwa prawdopodobieństwa. Jedno decyduje, czy dany genom będzie mutowany. Drugie określa jak duży fragment genomu będzie mutowany (siła mutacji). W tym wypadku mutacja polega na zamianie wartości 0 na 1 w genomie i na odwrót. Zakłada się, że każdy indeks genomu ma taką samą szanse na mutacje. Założono również stałą siłę mutacji wynoszącą $max(1; 0,1 * N)$, gdzie N jest wymiarem problemu.

Zbadano jakość rozwiązania i szybkość znalezienia tego rozwiązania dla różnych wartości prawdopodobieństwa, że genom zostanie poddany mutacji.

3.6 Sukcesja

Nowa populacja, po przejściu procedury selekcji i mutacji, całkowicie zastępowała nową.

3.7 Funkcja oceny

Funkcja oceny dla danego problemu ma postać:

$$f(genom) = \begin{cases} 1 + \text{len(genom)} + \text{l. niepokrytych krawędzi}, & \textit{if} \text{ nie pokryto wszystkich krawędzi} \\ \text{l. wykorzystanych wierzchołków}, & \textit{if} \text{ pokryto krawędzie} \end{cases} \tag{1}$$

3.8 Rozwiązanie

Rozwiązaniem jest tablica informująca o wybraniu danych wierzchołków. Rozwiązane uznane jest jako poprawne jeśli pozwala na pokrycie wszystkich krawędzi. Jego oceną jest uzyskana dla niego wartości funkcji celu. Rozwiązania niepokrywające wszystkich krawędzi uznane są za niepoprawne.

4 Wyniki

4.1 Rozwiązanie brutalne

Pomocniczo, zaimplementowano algorytm rozwiązujący problem minimalnego wierzchołkowego pokrycia metodą brutalną. Wyniki uzyskane tą metodą nie były w żaden sposób wykorzystywane podczas dobierania parametrów algorytmu genetycznego, służyły jedynie weryfikacji poprawności uzyskanego ostatecznego rozwiązania.

4.2 Algorytm genetyczny

4.2.1 Prawdopodobieństwo mutacji

Operując cały czas na tych samych grafach, wywołano algorytm 10-krotnie, dla takich samych parametrów: rozmiaru ilości iteracji równej 500, liczności populacji 200, dla problemu grafu losowego o rozmiarze 25. Zmienna była prawodpodobieństwo mutacji. Tabela 1 zawiera podsumowanie wywołań dla różnych liczności. Kolumny dotyczące "liczby iteracji" odnoszą się do liczby iteracji, po których znalezione zostało rozwiązanie uznane potem za prawidłowe.

prawdopodonieństwo mutacji	ułamek prawidłowych	średni wynik	std wyniku	średnia l. iteracji	std l. iteracji
0,01	1	21,2	0,6	44,9	71,28
0,05	1	20,9	0,7	115,7	158,54
0,1	1	20,4	0,49	183,1	157,71
0,5	1	20,3	0,46	102,5	127,47

Tabela 1: Podsumowanie wywołań algorytmu dla różnych prawdopodobieństw mutacji

4.2.2 Wielkość populacji

Operując cały czas na tych samych grafach, wywołano algorytm 10-krotnie, dla takich samych parametrów: rozmiaru ilości iteracji równej 500, prawdopodobieństwa mutacji równemu $0,01$, dla problemu grafu losowego o rozmiarze 25. Zmienna była jedynie liczność populacji. Tabela 2 zawiera podsumowanie wywołań dla różnych liczności. Kolumny dotyczące "liczby iteracji" odnoszą się do liczby iteracji, po których znalezione zostało rozwiązanie uznane potem za prawidłowe.

liczność populacji	ułamek prawidłowych	średni wynik	std wyniku	średnia l. iteracji	std l. iterajcji
25	1	21,4	0,66	160,9	141,56
50	1	21,5	0,5	71,7	54,18
125	1	20,9	0,7	105	115,11
250	1	21,2	0,6	61,5	71,17

Tabela 2: Podsumowanie wywołań algorytmu dla różnych wielkości populacji

4.2.3 Różne rodzaje grafów

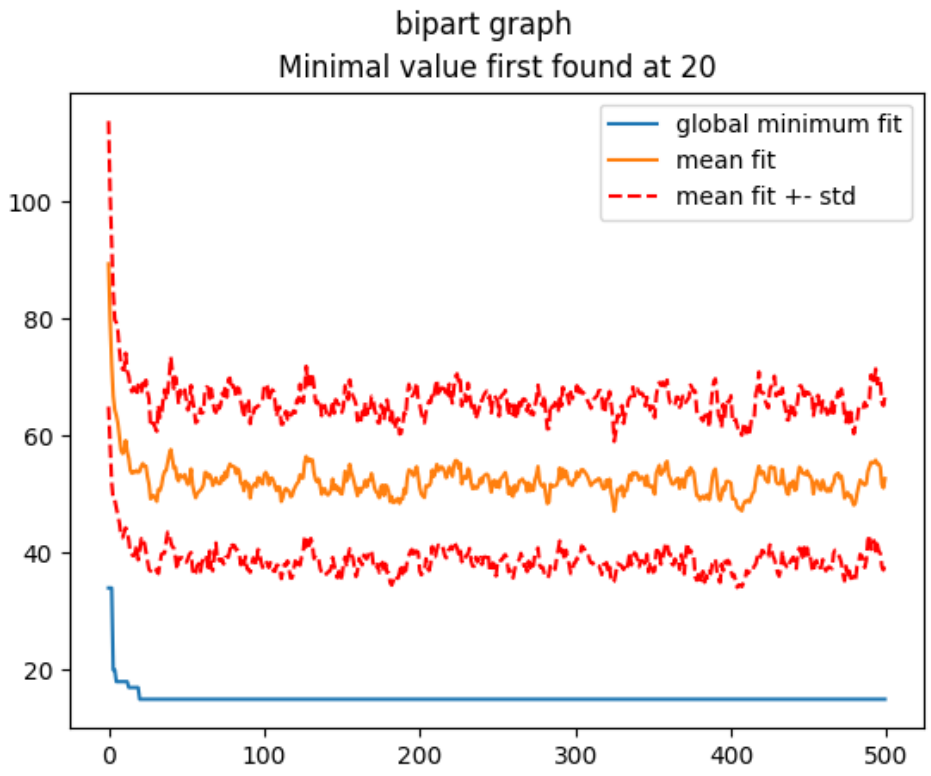
Zgodnie z poleceniem, wszystkie grafy miały 25 wierzchołków. Oba grafy pełne nie mają w sobie elementu losowości - tzn dla zadanej liczby wierzchołków można je zbudować tylko na jeden sposób. Graf losowy stworzony był poprzez usunięcie z grafu pełnego wybranej losowo połowy połączeń, dbając jednak o jego spójność.

Operując cały czas na tych samych grafach, wywołano algorytm 10-krotnie, dla takich samych parametrów: rozmiaru populacji równej 200, ilości iteracji równej 500, prawdopodobieństwa mutacji równemu $0,01$, dla problemu o rozmiarze 25. Eksperyment wykonano dla 3 różnych grafów. Tabela 3 zawiera podsumowanie wywołań w zależności od rodzaju grafu. Kolumny dotyczące "liczby iteracji" odnoszą się do liczby iteracji, po których znalezione zostało rozwiązanie uznane potem za prawidłowe.

ułamek prawidłowych	średni wynik	std wyniku	średnia l. iteracji	std l. iteracji	typ grafu	prawdziwy wynik	liczba rozwiązań
0,9	24,6	1,2	69,7	119,37	pełny	24	25
1	20,5	0,5	96,6	84,4	losowy	20	6
1	14	1,18	120	120,29	dwudzielny	12	1

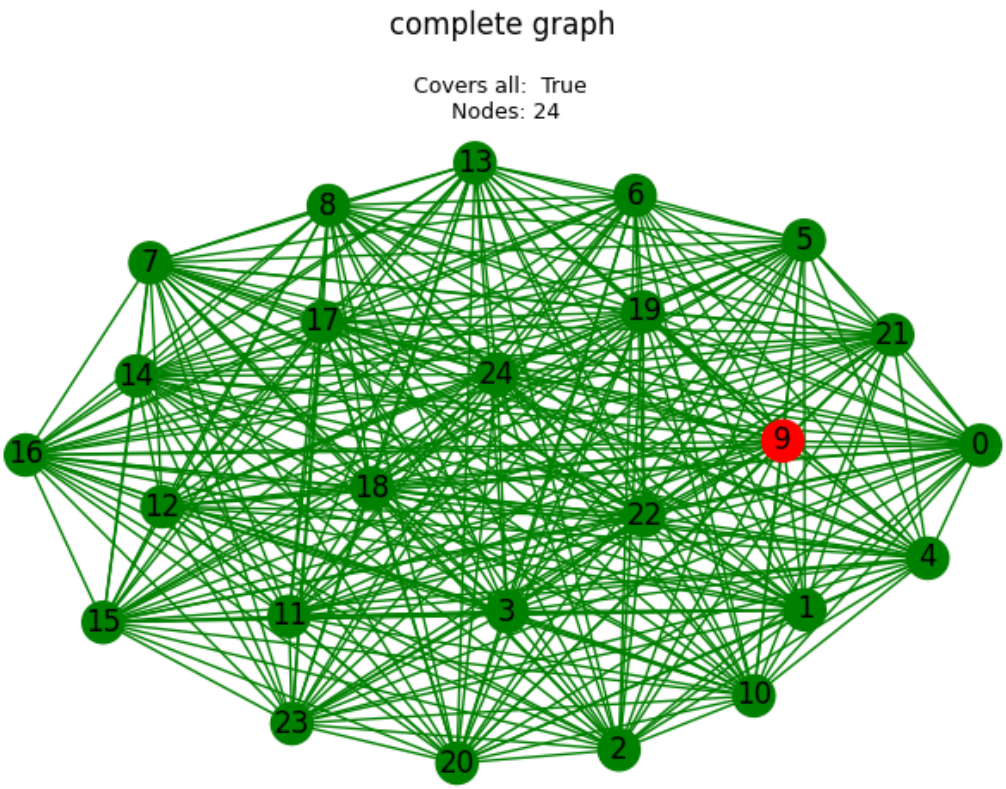
Tabela 3: Podsumowanie wywołań algorytmu dla różnych rodzajów grafów

Rysunek 1 zawiera historie przebiegu minimalnej wartości funkcji celu, średniej wartości funkcji celu w populacji oraz odchyleń standardowych jej wartości w zależności od iteracji, dla czterech rodzajów grafów. Jest to przebieg dla jednego z 10 wywołań funkcji. Nie zdecydowano się zamieścić przebiegów funkcji dla wszystkich wywołań oraz dla różnych rodzajów grafów, gdyż po analizie stwierdzono, że mają one podobny przebieg.

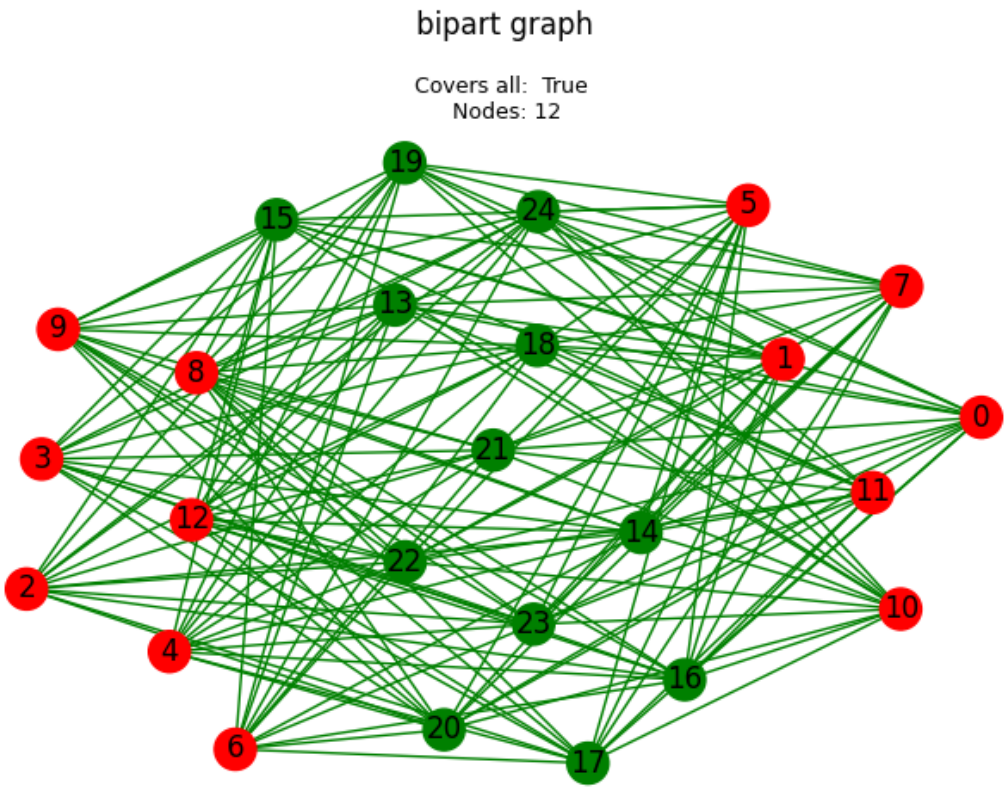


Rysunek 1: Historia przebiegu średniej funkcji celu.

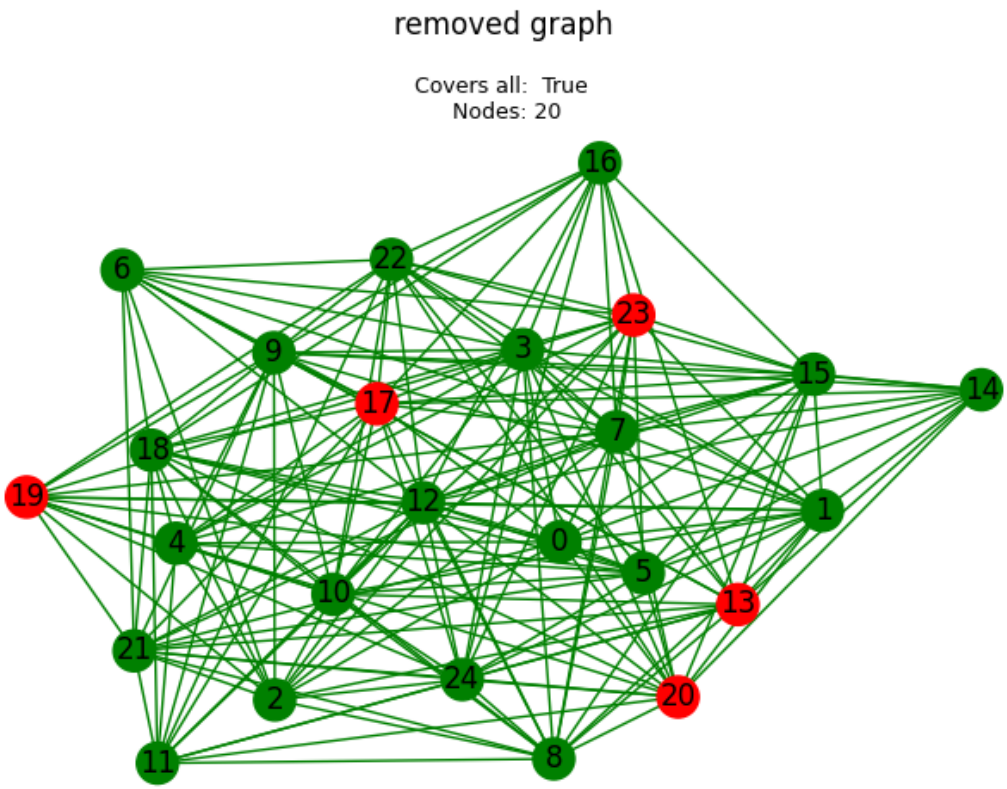
Rysunki 2,3,4 przedstawiają wybrane rozwiązania problemu minimalnego pokrycia uzyskane dzięki algorytmowi genetycznemu.



Rysunek 2: Rozwiązanie problemu minimalnego pokrycia (algorytm genetyczny) - graf pełny.

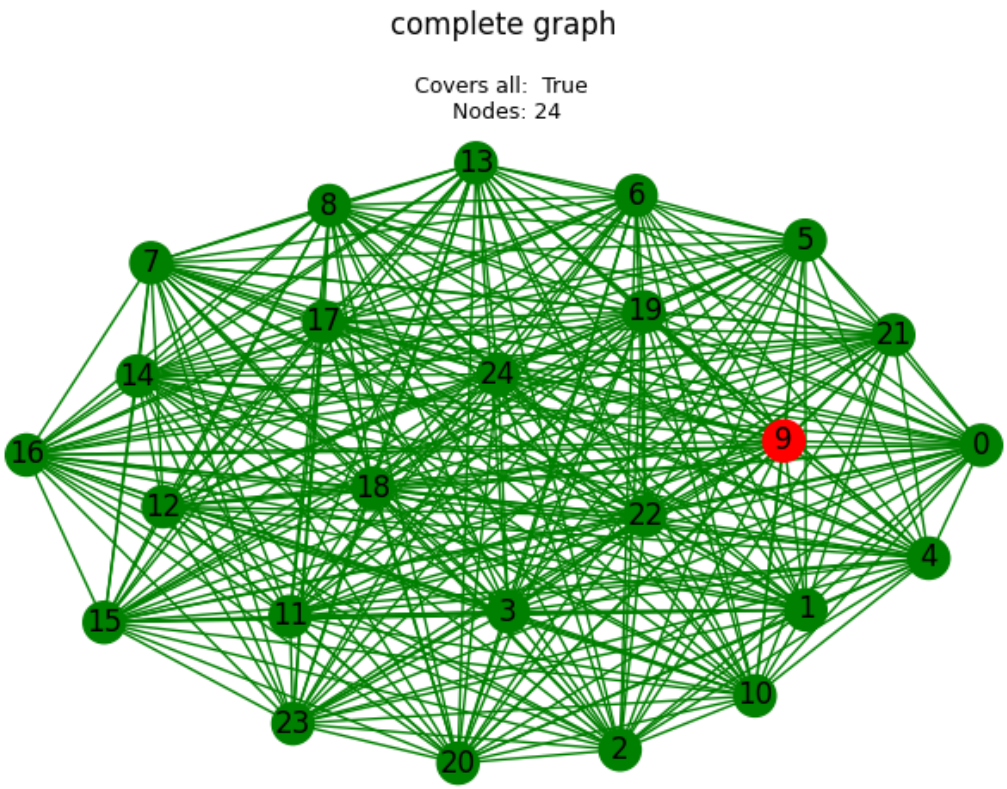


Rysunek 3: Rozwiązanie problemu minimalnego pokrycia (algorytm genetyczny) - graf pełny dwudzielny.

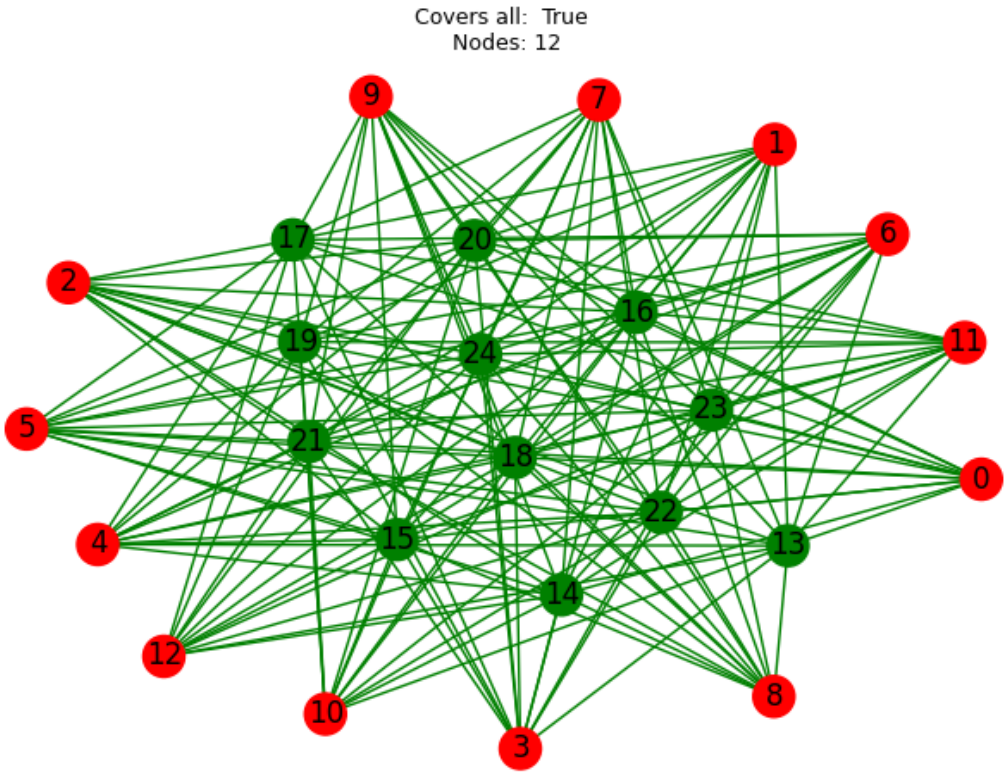


Rysunek 4: Rozwiązanie problemu minimalnego pokrycia (algorytm genetyczny) - graf losowy wersja 1.

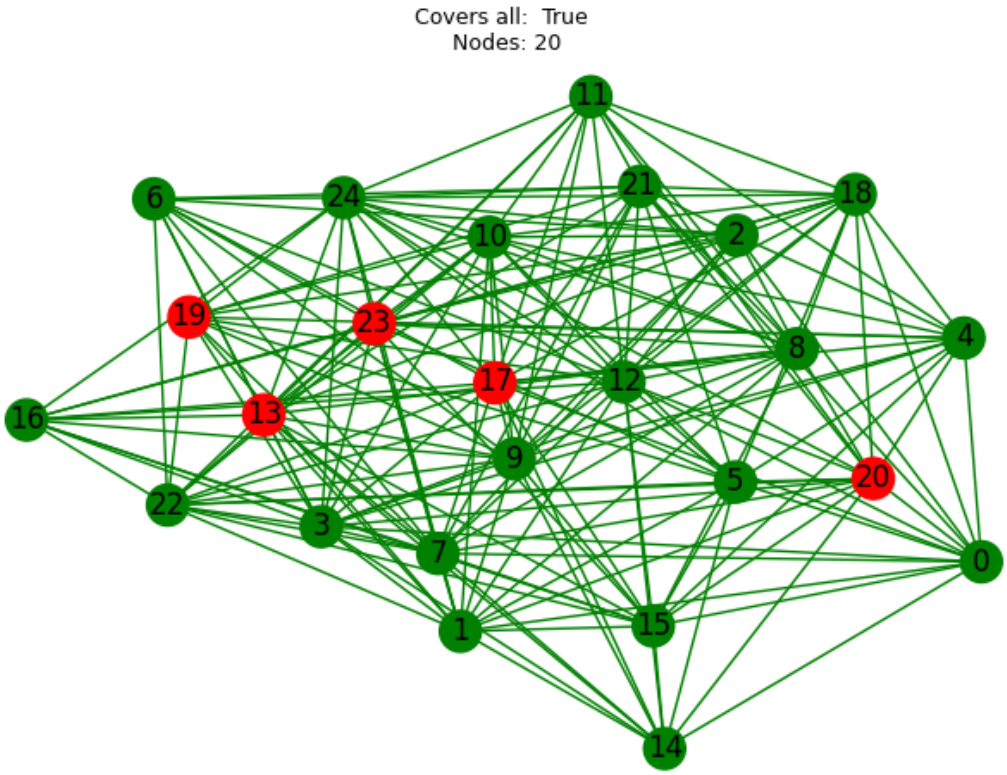
Rysunki 5,6,7 przedstawiają rozwiązania grafów losowych uzyskane metodą brutalną. W przypadku wielu rozwiązań zdecydowano się zamieścić tylko jedno z nich. Informacja o ilości rozwiązań zawarta jest w Tabeli 2.



Rysunek 5: Rozwiązanie problemu minimalnego pokrycia (metoda brutalna) - graf pełny.



Rysunek 6: Rozwiązanie problemu minimalnego pokrycia (metoda brutalna) - graf pełny dwudzielny.



Rysunek 7: Rozwiązanie problemu minimalnego pokrycia pokrycia (metoda brutalna) - graf losowy wersja 1.

5 Wnioski

5.1 Prawdopodobieństwo mutacji

Wraz ze wzrostem prawdopodobieństwa mutacji, średnia ilość iteracji potrzebnych na znalezienie minimum generalnie rośnie. Postawienie takiego wniosku jest jednak o tyle ryzykowne, że odchylenie standardowe uzyskanych wyników jest tak duże, że wskazuje duża losowość uzyskanych wyników. Prawdopodobnie należałoby przeprowadzić zdecydowanie więcej testów niż 10 by uzyskać prawdziwie reprezentatywny wynik. Taka tendencja mogłaby wynikać z tego, że przy prawdopodobieństwie mutacji tak dużym jak $0,5$, w kolejnych populacjach modyfikujemy zbyt dużo osobników. Następnie te osobniki uczestniczą w selekcji turniejowej, a kolejna nowo powstała populacja składa się z wielu nowych członków. Może to zaburzyć pożądaną w przypadku algorytmu genetycznego tendencje do skupiania się populacji wokół minimalnego rozwiązania. Warto jednak zwrócić uwagę na fakt, że wybrana siła mutacji jest stała w każdym z wywołań. Prawdopodobnie modyfikacja tej właśnie wartości miałaby zdecydowanie większy wpływ na wyniki działania algorytmu (a przede wszystkim przebieg średniej

wartości funkcji celu w populacji).

5.2 Rozmiar populacji

Zgodnie z przypuszczeniami, tendencja jest taka, że im większa jest populacja, tym szybciej znajdowane jest minimalne ostateczne rozwiązanie. Odchylenie standardowe iteracji oraz odchylenie standardowe ostatecznej oceny rozwiązania również maleje. Wynika to z tego, że przy dużej populacji, jest więcej szans na uzyskanie mutacji pozwalającej na uzyskanie minimum funkcji oceny. Oczywiście, wraz ze wzrostem rozmiaru populacji, koszt obliczeniowy wykonania algorytmu rośnie. Podobnie jak w punkcie poprzednim, warto jednak wspomnieć o ogromnym odchyleniu standardowym uzyskanych wyników.

5.3 Wpływ rodzaju grafu na działanie algorytmu

5.3.1 Rodzaj grafu, a ilość rozwiązań

Warto zauważyć, że w przypadku grafów pełnych o 25 wierzchołkach, minimalne pokrycie wierzchołkowe zawsze zawiera 24 wierzchołki i występuje 25 różnych równie dobrych rozwiązań. W przypadku grafu dwudzielnego z możliwie równym podziałem na niezależne podzbiory, dla parzystej liczby wierzchołków występowałyby dwa symetryczne rozwiązania, jednak dla liczby 25, jako nieparzystej - jedno jest lepsze i wynosi 12 wierzchołków.

Co do grafów o losowo usuniętych krawędziach nie możemy domniemywać jaki będzie rozmiar minimalnego pokrycia wierzchołkowego.

5.3.2 Działanie algorytmu

W przypadku wszystkich rodzajów grafów, praktycznie zawsze odnajdywane było rozwiązanie pokrywające wszystkie krawędzie. Końcowa jakoś rozwiązania nie odbiegała najczęściej od minimum globalnego w znacznym stopniu.

Najszybciej algorytm odnajdował rozwiązanie minimalne dla grafu pełnego, następnie dla grafu losowego i dwumianowego. Można zauważyć, że im więcej graf miał rozwiązań (w sensie rozwiązania problemu pokrycia wierzchołkowego), tym szybciej znajdowane było rozwiązanie będące ostatecznym minimum.

5.4 Historia przebiegu funkcji celu

Przebiegi wartości średniej funkcji celu dla wszystkich trzech rodzajów grafów są bardzo zbliżone. Wraz z przebiegiem kolejnych iteracji zaobserwować można: najpierw gwałtowny spadek wartości średniej oraz odchylenia standardowego, kończący się zwykle w okolicy znalezienia rozwiązania ostatecznego lub zbliżonego do ostatecznego, a następnie wypłaszczenie średniej wartości funkcji celu w populacji. Optymalnej liczby iteracji, pozwalającej na znalezienie zadowalająco dobrego rozwiązania, należałoby szukać w na początku wspomnianego wypłaszczenia, gdyż kolejne iteracje nie poprawiają/nie poprawiają w dużym stopniu funkcji celu.

6 Podsumowanie

Udało się zaimplementować algorytm genetyczny skutecznie rozwiązujący problem minimalnego pokrycia wierzchołkowego. Uzyskane wyniki najczęściej spójne były z rozwiązaniami uzyskanymi metodą brutalną. Zbadano wpływ wielkości populacji na szybkość znalezienia rozwiązania (im większa populacja tym szybciej) oraz wpływ prawdopodobieństwa mutacji osobnika (dla dużych wartości negatywnie wpływa na szybkość znalezienia rozwiązania). Porównano działanie algorytmu dla grafów pełnych, pełnych dwudzielnych oraz losowych - najszybciej rozwiązanie znajdowane było dla gafu pełnego, wolniej dla dwudzielnego i losowego.