

# Grafy i Sieci. Sprawozdanie 2.

SK11 Kolorowanie grafu za pomocą przeszukiwania z tabu.

Michał Aniserowicz, Jakub Turek

## 1 Temat projektu

SK11 Kolorowanie grafu za pomocą przeszukiwania z tabu.

## 2 Opis algorytmu

Zadaniem programu jest pokolorowanie wierzchołków zadanego grafu z użyciem jak najmniejszej liczby kolorów. Kolorowanie odbywa się z wykorzystaniem heurystycznego algorytmu przeszukiwania z tabu. Węzłem przestrzeni przeszukiwań jest pokolorowany (legalnie bądź nie) graf.

### 2.1 Funkcja celu

Algorytm dąży do minimalizacji funkcji celu<sup>1</sup>:

$$f(G) = - \sum_{i=1}^k C_i^2 + \sum_{i=1}^k 2C_i E_i \quad (1)$$

gdzie:

- $G$  - graf, dla którego liczona jest funkcja celu,
- $k$  - liczba kolorów użytych do pokolorowania grafu  $G$ ,
- $C_i$  - liczba wierzchołków grafu  $G$  pokolorowanych na  $i$ -ty kolor,
- $E_i$  - liczba krawędzi grafu  $G$ , których oba końce pokolorowane są na  $i$ -ty kolor.

Definicję funkcji należy rozumieć następująco:

1. z jednej strony, faworyzowane są pokolorowania z użyciem jak najmniejszej liczby kolorów,
2. z drugiej strony, dyskryminowane są pokolorowania nielegalne.

---

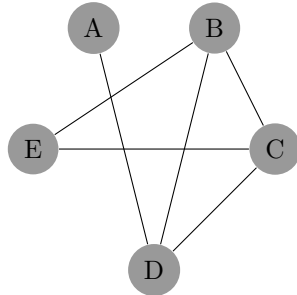
<sup>1</sup>Definicja funkcji celu zaczerpnięta z: D. S. Johnson, C. R. Aragon, L. A. McGeoch, C. Schevon, Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning, Operations Research, Vol. 39, No. 3, May-June 1991, pp. 378-406.

## 2.2 Lista tabu

Lista tabu zawiera ograniczoną liczbę ostatnich akcji podjętych przez algorytm. Pojedynczą akcją jest pokolorowanie pojedynczego wierzchołka na określony kolor.

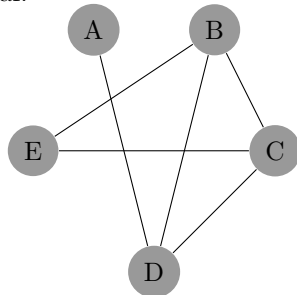
## 2.3 Przykład

Graf:



### 2.3.1 Krok 0

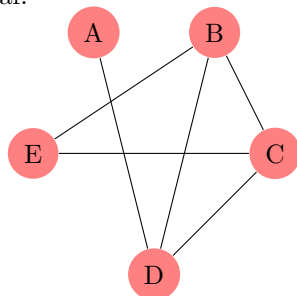
Graf:



Lista tabu jest pusta.

### 2.3.2 Krok 1

Graf:



Lista tabu:



### 3 Struktury danych

Podstawową strukturą danych użytą w programie jest graf, na który składa się zbiór wierzchołków. Pojedynczy wierzchołek zawiera:

- identyfikator,
- przypisany kolor,
- zbiór wskazań na sąsiadujące wierzchołki.

### 4 Projekty testów

### 5 Założenia programu

#### 5.1 Opcje

**Wielkość pamięci** Rozmiar tablicy tabu jest wymaganym parametrem aplikacji. Rozmiar tablicy tabu jest specyfikowany opcją `-m`.

Przykład: `<nazwa_programu> -m 5` uruchamia algorytm z pięcioelementową tablicą tabu.

**Plik(i) wejściowe** W opcjach programu można wyspecyfikować jeden lub więcej plików wejściowych. Nazwę pliku wejściowego specyfikuje się bez dodatkowych opcji, zaraz po nazwie programu.

Przykład: `<nazwa_programu> graf1.txt -i graf2.txt` wykona algorytm dla grafów opisanych w plikach *graf1.txt* oraz *graf2.txt*.

**Plik wyjściowy** W opcjach programu można wyspecyfikować nazwę pliku, do którego zostanie zapisane wyjście programu. Nazwę pliku wyjściowego specyfikuje się opcją `-o`. Nazwa pliku wyjściowego jest parametrem opcjonalnym. Domyślnie wyjście przekierowywane jest na standardowy strumień (konsolę).

Przykład: `<nazwa_programu> -o wyjscie1.txt` zapisze wyjście algorytmu do pliku *wyjscie1.txt*.

**Tryb „rozmowny”** W opcjach programu można włączyć tryb „rozmowny” (*verbose*), który wyprowadza dodatkowe informacje diagnostyczne na wyjście w trakcie działania algorytmu. W trybie domyślnym na wyjście wyprowadzany jest tylko wynik działania algorytmu. Tryb „rozmowny” specyfikuje się opcją `-v`.

Przykład: `<nazwa_programu> -v` uruchamia aplikację w trybie „rozmownym”.

**Maksymalna liczba iteracji** Maksymalna liczba iteracji określa liczbę przejść algorytmu, po której aplikacja wyłączy się (opisane w sekcji 5.2). Maksymalną liczbę iteracji specyfikuje się opcją `-i`.

Przykład: `<nazwa_programu> -i 500` uruchamia algorytm dla maksymalnie 500 iteracji.

**Maksymalna liczba iteracji bez zmiany rezultatu** Maksymalna liczba iteracji bez zmiany rezultatu określa liczbę przejść algorytmu, po której aplikacja wyłączy się, jeżeli wartość funkcji celu dla najlepszego pokolorowania nie zmieni się (opisane w sekcji 5.2). Maksymalną liczbę iteracji bez zmiany wyniku specyfikuje się opcją `-s`.

Przykład: `<nazwa_programu> -s 25` uruchamia algorytm dla maksymalnie 25 iteracji bez zmiany wyniku.

## 5.2 Kryteria stopu

**Maksymalna liczba iteracji** Wykonywanie programu zakończy się, gdy algorytm przekroczy maksymalną liczbę iteracji. Maksymalna liczba iteracji jest podana jako parametr aplikacji.

**Maksymalna liczba iteracji bez zmiany rezultatu** Wykonywanie programu zakończy się, gdy algorytm przekroczy maksymalną liczbę iteracji, w których nie zmieniła się wartość funkcji celu dla najlepszego pokolorowania. Maksymalna liczba iteracji bez zmiany wyniku jest podawana w parametrach aplikacji.