

# Opis zadania i zaproponowanego rozwiązania - sprawozdanie

## 1 Problem 1

### 1.1 Opis problemu, wybór metaheurystyki

Problem znalezienia jak największego zespołu naukowców, którego członkowie nie mają ze sobą żadnych wspólnych publikacji (tj. w zespole nie istnieje żadna para członków, która pracowała ze sobą wcześniej) sprowadza się do problemu znalezienia maksymalnego zbioru niezależnego w grafie, którego wierzchołkami są naukowcy, a krawędzie reprezentują fakt posiadania wspólnej publikacji przez połączonych krawędzią naukowców (wierzchołków). Do rozwiązania problemu zastosowałam metaheurystykę opartą na przeszukiwaniu lokalnym - algorytm wyszukiwania lokalnego (ILS).

### 1.2 Opis algorytmu

Wybrałam algorytm przeszukiwania lokalnego oparty na metodzie zachłannej.

*Krok I* Losujemy  $N^1$  (gdzie  $N$  to parametr funkcji) rozwiązań początkowych w następujący sposób: Inicjalizujemy pomocniczą listę złożoną z wszystkich wierzchołków grafu. Losujemy pierwszy wierzchołek naszego rozwiązania z tej listy, a następnie aktualizujemy listę: usuwamy z niej wylosowany wierzchołek i wszystkich jego sąsiadów. Dalej, losujemy kolejny wierzchołek z tej listy, aktualizujemy listę. Kontynuujemy losowanie kolejnych wierzchołków (a następnie aktualizowanie listy) dopóki lista wierzchołków jest niepusta. Z  $N$  wylosowanych rozwiązań wybieramy najlepsze (tj. o największej liczbie wierzchołków) - tym sposobem znaleźliśmy nasze rozwiązanie początkowe, które w następnym kroku będziemy próbować optymalizować.

*Krok II* Dla zadanego parametru  $m$  (warunek stopu - liczba o jaką różni się rozwiązanie po wykonaniu  $N$  iteracji) i  $N$  (liczba iteracji do wykonania, po której dokonujemy sprawdzenia rozwiązania) wykonujemy kolejno  $N$  iteracji, a w czasie każdej iteracji dokonujemy losowej mutacji rozwiązania: polega ona na wylosowaniu indeksu wierzchołka z naszego rozwiązania, która ma następnie zostać usunięty. Po usunięciu tego wierzchołka (nazwijmy go  $w$ ) dodajemy do naszej pomocniczej listy wierzchołków, sąsiadów  $w$ , o ile ci nie sąsiadują z którymś z wierzchołków z rozwiązania (sprawdzamy to uprzednio). Jeśli dokonana w ten sposób mutacja zwraca nam gorszy wynik (mniejszą liczbę wierzchołków), niż poprzednie rozwiązanie, to cofamy zmiany (funkcja `reverse_mutation`). Po każdych  $N$  wykonanych iteracjach pętla *while* sprawdza, czy liczba o jaką różni się rozwiązanie od rozwiązania sprzed cyklu  $N$  iteracji jest mniejsza od parametru  $m$  - z chwilą, gdy ten warunek jest spełniony kończy się wykonywanie programu.

---

<sup>1</sup>w mojej implementacji ten parametr ma nazwę *l\_losowan*

### 1.3 Wpływ wyboru parametrów heurystyki na jakość rozwiązania

Wykonano szereg prób dla kilku różnych zestawów wartości parametrów. Wyniki przedstawia poniższa tabela:

l-losowań	N (liczba iteracji)	m (war_stopu)	wynik (liczebność znalezionego zbioru)
20	10	2	2207
20	10	2	2187
20	10	2	2187
20	10	2	2203
20	10	2	2190
20	10	2	2179
20	10	2	2201
80	10	2	2200
80	10	2	2210
80	10	2	2207
80	10	2	2202
80	10	2	2218
80	10	2	2206
80	10	2	2205
80	15	2	2203
80	15	2	2202
80	15	2	2196
80	15	2	2218
80	15	2	2210
80	15	2	2203
80	15	2	2202
80	5	1	2199
80	5	1	2207
80	5	1	2217
80	5	1	2208
80	5	1	2211
80	5	1	2211
80	5	1	2226
80	5	1	2209

80	5	1	2209
80	10	1	2212
80	10	1	2209
80	10	1	2216
80	10	1	2203
80	10	1	2212
80	10	1	2202
80	10	1	2212
150	10	1	2222
150	10	1	2212
150	10	1	2214
150	10	1	2210
150	10	1	2203
150	10	1	2220
80	30	1	2186
80	30	1	2195
80	30	1	2184
80	30	1	2197
80	30	1	2197
80	30	1	2220

Wydaje się, że najrozsądniejszym wyborem parametrów heurystyki będzie:  $l\_losowan \sim 80$ ,  $N \sim 5-10$ ,  $m = 1$ . Co prawda, wydaje się, że dla kilkukrotnie większych wartości parametru  $l\_losowan$  możemy się spodziewać jeszcze lepszych średnich wyników (na co wskazują wyniki uzyskane dla  $l\_losowan = 150$ ), ale nieduża poprawa średniej jakości rozwiązania (chyba) nie rekompensuje dłuższego czasu wykonywania programu.

## 2 Problem 2

### 2.1 Opis problemu

Zadany problem sprowadza się do problemu znalezienia minimalnego pokrycia wierzchołkowego w grafie, którego wierzchołkami są ulice, a krawędziami połączone są wierzchołki odpowiadające ulicom, które się przecinają. Nietrudno zauważyć, że jest to problem dualny do poprzedniego (z tw. Halla wynika, że suma wszystkich wierzchołków grafu jest równa sumie maksymalnego zbioru niezależnego i minimalnego pokrycia wierzchołkowego, oraz, że **zbiór wierzchołków będący minimalnym pokryciem wierzchołkowym jest dopełnieniem maksymalnego zbioru niezależnego**). Można zatem użyć algorytmu z poprzedniego zadania do znalezienia maksymalnego zbioru niezależnego w grafie, a następnie wziąć jego dopełnienie.

### 2.2 Wpływ wyboru parametrów heurystyki na jakość rozwiązania

Wykonano szereg prób dla kilku różnych zestawów wartości parametrów. Wyniki przedstawiają się następująco:

l-losowań	N (liczba iteracji)	m (war_stopu)	wynik (liczebność znalezionej zbioru)
20	10	2	3060
20	10	2	3031
20	10	2	3044
20	10	2	3045
20	10	2	3042
20	10	2	3042
80	10	2	3028
80	10	2	3028
80	10	2	3038
80	10	2	3039
80	10	2	3037
80	10	2	3026
80	10	2	3026
80	10	2	3021
80	15	1	3033
80	15	1	3027
80	15	1	3042
80	15	1	3043
80	15	1	3025
80	15	1	3041
80	15	1	3036
80	10	1	3038
80	10	1	3015
80	10	1	3039
80	10	1	3028
80	10	1	3033
80	10	1	3031
80	10	1	3023
80	10	1	3032
150	10	1	3035
150	10	1	3029
150	10	1	3028
150	10	1	3034
150	10	1	3025
150	10	1	3043
300	10	1	3031
300	10	1	3032
300	10	1	3029
300	10	1	3018
300	10	1	3032
300	10	1	3021

150	5	1	3025
150	5	1	3025
150	5	1	3018
150	5	1	3031
150	5	1	3026
150	5	1	3011

Wydaje się, że wpływ wyboru wartości parametrów heurystyki na jakość rozwiązań wygląda dość podobnie jak w poprzednim zadaniu.