

Adrianna Mazur

Paweł Sienkiewicz

Projekt 3 - Spy Union

Opis zadania

Celem projektu jest znalezienie rozwiązania problemu Spy Union poprzez wskazanie możliwie największej liczby pracowników, których zwolnienie nie zagrozi funkcjonowaniu żadnej z dwóch organizacji: WSA oraz Trade Union of Spies. Rozwiązanie wygenerowane zostało z wykorzystaniem jednej z metod programowania liniowego - metody Sympleks.

Dane wejściowe

Dane wejściowe zadane są za pomocą pliku tekstowego o nazwie "numer" (gdzie "numer" należy do $\{0, 1, \dots, 10\}$) oraz rozszerzeniu .in (np. 10.in).

Przykładowe dane wejściowe:

```
5
1 0 1 2
2 0 1 2
2 1 2 0
2 1 0 1
1 3 0 0
```

Liczba w pierwszym wierszu oznacza liczbę szpiegów w obydwu organizacjach (wszyscy szpiegowie należą bowiem zarówno do WSA, jak i do Trade Union of Spies). Każdy z pracowników organizacji ma swój własny, niepowtarzalny numer ID, gdzie $ID \in \{0, 1, \dots, N - 1\}$

Kolejne N wierszy (gdzie N to liczba z pierwszego wiersza - w powyższym przykładzie równa 5) zawiera dane dla każdego ze szpiegów, przy założeniu, iż w wierszu drugim znajdują się dane dla szpiega o numerze ID = 0, w wierszu trzecim - dla pracownika o numerze ID = 1 etc.

Dane przedstawione są w następujący sposób:

Bw Bu Rw Ru

gdzie:

Bw - ID przełożonego w hierarchii WSA

Bu - ID przełożonego w hierarchii Trade Union of Spies

Rw - ilość szpiegów wymagana do funkcjonowania departamentu, którego dany szpieg jest szefem (ilość wierzchołków w poddrzewie o korzeniu danego szpiega) w organizacji WSA

Ru - ilość szpiegów wymagana do funkcjonowania departamentu, którego dany szpieg jest szefem (ilość wierzchołków w poddrzewie o korzeniu danego szpiega) w organizacji Trade Union of Spies

Opis algorytmu

W celu rozwiązania problemu Spy Union wykorzystany został algorytm składający się z następujących kroków:

1. Wczytywanie danych z pliku o rozszerzeniu .in (np. 10.in)
2. Zdefiniowanie funkcji "pracownicy" określającej ilość szpiegów, których dany szpieg jest przełożonym. Jako parametry podajemy:
 - boss - ID szpiega, dla którego funkcja zwrócić ma listę jego podwładnych
 - organizacja - zmienna zerojedynkowa, dla której wartość 0 oznacza WSA, natomiast wartość 1 - Trade Union of Spies

W tym miejscu warto zaznaczyć, iż funkcja ta może zostać zaimplementowana zarówno w sposób iteracyjny, jak i rekurencyjny. Druga metoda pozwala na uzyskanie wyników w krótszym czasie, co widoczne jest zwłaszcza dla testów końcowych (m.in. pliki 9.in oraz 10.in)

3. Zainicjowanie metody Sympleks w postaci MixedIntegerLinearProgram. Funkcja celu zdefiniowana została jako suma N zmiennych zero-jedynkowych. (każda z nich odpowiada jednemu ze szpiegów. Wartość 1 oznacza, iż dana osoba pozostanie pracownikiem organizacji, natomiast wartość równa 0 - szpieg zostanie zwolniony).

4. Dodawanie ograniczeń (constraints) do programu liniowego dla każdego z wierzchołków (każdy ze szpiegów generuje dwa takie ograniczenia, zapisane w wejściowej tabeli w kolumnach Rw i Ru)
5. Start właściwego programu liniowego
6. Dopasowanie otrzymanych danych do pożądanej formy prezentacji wyniku
7. Eksport danych końcowych do pliku tekstowego o nazwie out.out

Dane wyjściowe

Dane wyjściowe znajdują się w plikach o nazwie "numer".out (gdzie "numer" należy do $\{0, 1, \dots, 10\}$). Struktura każdego z nich jest następująca:

2

4 2

Liczba w pierwszym wierszu oznacza liczbę zwolnionych szpiegów. W toku przeprowadzania testów zaobserwowano, że wartość ta najczęściej może być przybliżona liczbą $N/2$.

Liczby w drugim rzędzie prezentują ID usuniętych z organizacji pracowników.