

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

Powyższa struktura danych wejściowych sugeruje, iż najlepszym sposobem podejścia do rozwiązania problemu Spy Union jest osobne rozpatrywanie hierarchii obu organizacji w kontekście struktury drzewa, którego wierzchołkami są pracujący w nich szpiedzy. Korzeniem danego drzewa jest pracownik nieposiadający szefa, co oznacza, że w danych wejściowych, w przypisanym do niego wierszu w kolumnie pierwszej bądź drugiej pojawi się jego własny numer ID (w naszym przykładzie pracownik o ID = 2 jest szefem organizacji WSA, natomiast pracownik o ID = 0 jest szefem Trade Union of Spies). Liśćmi natomiast są ci szpiedzy, którzy nie mają podwładnych - ich ID nie pojawia się w pierwszej bądź drugiej kolumnie danych wejściowych (w naszym przykładzie liśćmi drzewa hierarchii WSA są szpiedzy o ID = 0 oraz ID = 4, natomiast liśćmi drzewa hierarchii Trade Union of Spies są pracownicy o ID = 2 oraz ID = 4).

W przypadku, gdy liczba Rx przypisana do danego pracownika wynosi 0, natomiast liczba Rx przypisana do jego podwładnego jest różna od 0, zakładamy, że zwolnienie pracownika z Rx = 0 oznacza usunięcie jego departamentu, natomiast departament jego podwładnego pozostaje częścią organizacji.

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 liczbę pracowników, 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 nieco krótszym czasie, jednak w miarę zwiększania się liczby wierszy w danych wejściowych, różnica w czasie wykonywania obu wersji programu staje się coraz mniejsza.

3. Zainicjowanie metody Sympleks w postaci MixedIntegerLinearProgram. Funkcja celu (która w tym zadaniu jest minimalizowana) jest sumą N zmiennych zero-jedynkowych, zdefiniowanych następująco:

$$x_i = \begin{cases} 1 & \text{gdy pracownik o ID} = i \text{ pozostaje w organizacji} \\ 0 & \text{gdy pracownik o ID} = i \text{ zostaje zwolniony} \end{cases}$$

gdzie $i \in \{0, 1, \dots, N - 1\}$

4. Dodawanie ograniczeń (constraints) do programu liniowego dla każdego z wierzchołków drzew obu organizacji (każdy ze szpiegów generuje zatem dwa takie ograniczenia, zapisane w wejściowej tabeli w kolumnach R_w i R_u)
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:

Liczba w pierwszym wierszu oznacza liczbę zwolnionych szpiegów.

Liczby w drugim rzędzie prezentują ID usuniętych z organizacji pracowników.

Wyniki

Poniższa tabela (*Tab. 1*) prezentuje wyniki otrzymane za pomocą opisanego algorytmu dla każdego z 11 programów testowych. Z powodu długości plików numer.out zdecydowaliśmy się na zaprezentowanie jedynie pierwszego wiersza każdego z nich - a zatem maksymalnej możliwej liczby zwolnionych szpiegów dla danej ilości pracowników.

Tab. 1 Liczba zwolnionych szpiegów dla każdego z przeprowadzonych testów

Numer testu	Liczba szpiegów (N)	Liczba zwolnionych szpiegów	Procent zwolnionych szpiegów
0	5	2	40,000%
1	20	9	45,000%
2	100	53	53,000%
3	200	115	57,500%
4	500	302	60,400%
5	1000	585	58,500%
6	2000	1174	58,700%
7	5000	2754	55,080%
8	10000	5761	57,610%
9	20000	11492	57,460%
10	50000	26428	52,856%

Na podstawie wartości zaprezentowanych w *Tab. 1* możemy stwierdzić, że zaproponowany algorytm umożliwia redukcję liczby szpiegów o wartość z zakresu 40% - 60,4%, natomiast w przypadku organizacji o przynajmniej 100 członkach, okazuje się, że zwolnionych zostało przynajmniej 52,856%.