

POP projekt
Dokumentacja wstępna

Kacper Maj

Łukasz Topolski

1. Opis problemu

Mamy macierz prostokątną o wymiarach $4 \times N$ ($N \in \mathbb{N}$). Każdy z jej elementów jest liczbą całkowitą. Oprócz niej dostępnych jest m kart. Za pomocą nich można zakryć komórki macierzy, przy czym dwa zakryte elementy nie mogą być obok siebie (zarówno w pionie, jak i w poziomie). Zadanie polega na zakryciu komórek tak, żeby suma pozostałych elementów była jak największa. Nie ma przymusu wykorzystywania wszystkich kart.

Największą możliwą sumę można osiągnąć przy usunięciu wszystkich ujemnych liczb, dlatego też problem ten sprowadza się do zakrycia m najmniejszych liczb ujemnych tak, aby warunki zadania zostały spełnione. Ponadto z treści można wywnioskować, że liczba użytych kart jest z przedziału $[0, 2N]$ (w jednej kolumnie mogą być zakryte co najwyżej 2 komórki).

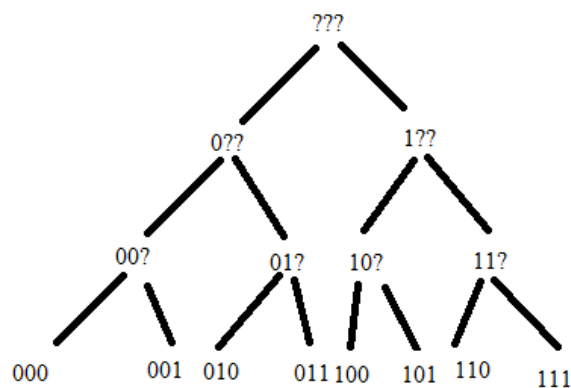
2. Propozycja rozwiązania

Do wyznaczenia najoptymalniejszego rozwiązania zostanie wykorzystany algorytm A^* wraz z heurystyką. Przestrzeń przeszukiwań to wektory składające się z zer i jedynek, gdzie 1 oznacza, że i -ta liczba ujemna została zakryta, a 0 – nie została zakryta. Pierwsza wartość jest wybierana losowo, a kolejne to następne liczby z tego samego wiersza. Gdy i -ta wartość jest ostatnią z wiersza, to wartością $i+1$ -tą jest kolejna liczba z następnego wiersza. Gdy jednak i -ta liczba jest ostatnią z ostatniego wiersza, to $i+1$ -ta wartość to następna liczba z pierwszego wiersza. Dla przykładowej macierzy z rysunku 1. i wartości -6 jako pierwszy element wektor zer i jedynek wygląda następująco: -4 odpowiada drugiej pozycji wektora, -1 odpowiada czwartej pozycji, a -5 – ósmej.

-5	-6	8	-4
-10	10	-1	15
0	-2	1	7
2	30	-7	-8

Rys. 1.: przykładowa macierz

Przykładowe drzewo przeszukiwań zostało zaprezentowane na rys. 2. (? – nie wiemy czy liczba i -ta została wybrana)



Rys. 2.: drzewo przeszukiwań dla 3 liczb ujemnych

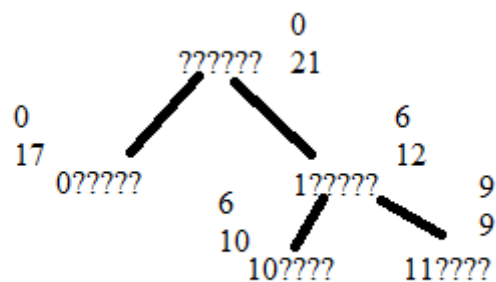
Funkcją celu jest moduł sumy (albo suma modułów – wykonujemy działania na liczbach o tym samym znaku) wybranych liczb. Funkcja heurystyczna to suma modułów co najwyżej $m_{max} - m_u$ najmniejszych (ujemnych) liczb, które potencjalnie mogą być zakryte (m_u – liczba zakrytych komórek, m_{max} – maksymalna możliwa liczba zakrytych komórek, $m_{max} = m$, gdy $m \leq 2N$ lub $m_{max} = 2N$, gdy $m > 2N$). W tej funkcji nie są brane pod uwagę elementy, które na pewno nie mogą być wybrane oraz te, które zostały już zakryte.

Poniżej znajduje się przykład przedstawiający kolejne kroki algorytmu:

Przykładowa plansza jest na rysunku 3., liczba kart = 4. Maksymalna możliwa liczba użytych kart wynosi 4. Pierwszym elementem niech będzie -6 (trzecia kolumna, drugi wiersz). Wartość funkcji celu wynosi 6 oraz funkcji heurystycznej – 12 ($7+2+3$) jeśli zakryjemy tę komórkę. Gdy nie zakryjemy, to wart. funkcji celu wynosi 0, a funkcji heurystycznej – 17 ($7+5+2+3$). Suma funkcji celu i funkcji heurystycznej jest większa przy wybraniu elementu -6, więc go zakrywamy. Kolejnym rozpatrywanym elementem będzie -3 (z pierwszej kolumny i ostatniego wiersza). Przy zakryciu go funkcja celu ma wartość 9, a funkcja heurystyczna – 9 ($7 + 2$). Natomiast gdy nie zostanie zakryty, to wart. funkcji celu wynosi 6, a funkcji heurystycznej – 10 ($7 + 2 + 1$). W tym przypadku element -3 zostaje zakryty. Analogicznie są wykonywane kolejne kroki aż dojdziemy do ostatniej liczby ujemnej lub gdy zakryjemy 4 liczby. Fragment drzewa przeszukiwań dla omówionego przypadku znajduje się na rysunku 4. (przy węźle drzewa pierwsza liczba oznacza wartość funkcji celu, a druga – wartość funkcji heurystycznej)

-1	8	-5	0
-7	3	-6	12
2	5	1	0
-3	7	6	-2

Rys. 3.: przykładowa tablica



Rys. 4.: Fragment drzewa przeszukiwań

3. Planowe eksperymenty numeryczne

W ramach testów planowane jest kilkukrotne uruchomienie algorytmu dla jednej macierzy i jej transpozycji z różnymi elementami startowymi i porównanie wyników (tzn. sum wszystkich elementów niewybranych tablicy). Również do porównania zostanie uwzględniona suma wszystkich wartości niezakrytych przy wybraniu co najwyżej m najmniejszych ujemnych liczb.

4. Wybrana technologia

Wybrany językiem programowania jest Python wraz z biblioteką NumPy.