Ćwiczenia 14

Problem zero–jedynkowego programowania liniowego

Dane: macierz A o wymiarach $m \times n$, wektor b długości m
Wynik: wektor zero–jedynkowy x, t.że $A \cdot x \leq b$

Problem zero–jedynkowego programowania liniowego jest NP–zupełny.

- problem należy do NP — wystarczy wybrać maszynę która zgaduje x, a następnie sprawdza czy $A \cdot x \leq b$,
- problem jest NP–zupełny — redukcja problemu 3–CNF–SAT,
  Dla danej formuły składającej się z k klauzul $c_1 \wedge c_2 \wedge \ldots \wedge c_k$ i n zmiennych $x_1, \ldots, x_n$. Tworzona jest macierz A o rozmiarze $2n \times (k+n)$. Macierz A jest zdefiniowana w następujący sposób (zapis $A[i,j]$ oznacza element w j–tym wierszu i i–tej kolumnie):
  - $A[2i-1, j] = -1$, $1 \leq i \leq n$, $1 \leq j \leq k$, gdy zmienna x_i występuje (bez negacji) w klauzuli c_j,
  - $A[2i, j] = -1$, $1 \leq i \leq n$, $1 \leq j \leq k)$, gdy gdy zmienna x_i występuje w postaci zanegowanej w klauzuli c_j,
  - $A[2i, j] = 0$, $A[2i-1, j] = 0$, $1 \leq i \leq n$, $1 \leq j \leq k)$, gdy gdy zmienna x_i nie występuje w klauzuli c_j,
  - $A[2i, k+i] = 1$, $A[2i-1, k+i] = 1$, dla $1 \leq i \leq n$,
  - $B[i] = -1$ dla $1 \leq i \leq k$,
  - $B[i] = 1$ dla $k < i \leq n+k$,

  Przykład: Dla formuły:

  $$(x_1 \vee x_2 \vee \neg\, x_3) \wedge (x_1 \vee \neg\, x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg\, x_1 \vee x_2 \vee \neg\, x_3)$$

  dane dla problemu zero–jedynkowego programowania liniowego mają następującą postać:

A: x_1	$\neg x_1$	x_2	$\neg x_2$	x_3	$\neg x_3$		X		B
-1	0	-1	0	0	-1		$X[1]$		-1
-1	0	0	-1	-1	0		$X[2]$		-1
-1	0	-1	0	-1	0	$\cdot$	$X[3]$	$=$	-1
0	-1	-1	0	0	-1		$X[4]$		-1
1	1	0	0	0	0		$X[5]$		1
0	0	1	1	0	0		$X[6]$		1
0	0	0	0	1	1				1

Problem najdłuższego cyklu prostego

Dane: graf G, k — liczba całkowita
Wynik: TAK, jeśli istnieje cykl prosty C_p, t.że $|C_p| \geq k$

- problem należy do NP — wystarczy wybrać maszynę Turinga losującą $k' \geq k$ i ciąg $C_{p'}$ ($|C_{p'}| = k'$), a następnie sprawdzającą, czy $C_{p'}$ jest prawidłowym cyklem prostym w G,
- problem jest NP–zupełny — redukcja problemu cyklu Hamiltona, Aby sprawdzić czy w grafie G istnieje cykl Hamiltona wystarczy sprawdzić czy graf zawiera cykl prosty o długości n, gdzie n — liczba wierzchołków.

Problem podziału zbioru

Dane: Zbiór liczb S: $s_1, s_2, \ldots, s_n$
Wynik: Zbiór $A \subseteq S$, t.że: $\sum_{x \in A} x = \sum_{x \in (S-A)} x$

- problem należy do NP — wystarczy wybrać maszynę Turinga losującą zbiór A, a następnie sprawdzającą sumy elementów A i $S - A$,
- problem jest NP–zupełny — redukcja problemu Subset–Sum Aby sprawdzić czy dla danego zbioru A istnieje podzbiór o sumie k wystarczy utworzyć zbiór $A' = A \cup \{|\sum A - 2x|\}$, i następnie zwrócić odpowiedź wygenerowaną przez algorytm dla problemu podziału zbioru.

Problem izomorfizmu podgrafu

Dane: grafy G_1, G_2
Wynik: czy G_1 jest podgrafem G_2

- problem należy do NP — wystarczy wylosować ułożenie G_1 w G_2 i sprawdzić czy jest poprawne,
- problem jest NP–zupełny — redukcja problemu maksymalnej kliki, Aby sprawdzić czy w grafie G występuje klika o rozmiarze k, wystarczy sprawdzić czy K_k (graf pełny o k wierzchołkach) jest podgrafem G.

Problem zbioru niezależnego

Dane: graf $G = (V, E)$, k
Wynik: TAK, jeśli istnieje zbiór $S \subseteq V$, $|S| \geq k$, t.że $\forall_{a,b \in S}(a, b) \notin E$

- problem należy do NP — np. maszyna losująca podzbiór wierzchołków o rozmiarze k, a następnie sprawdzająca czy jest on zbiorem niezależnym,

- problem jest NP–zupełny — redukcja problemu 3CNF–SAT
  Dla danej formuły $c_1 \wedge c_2 \wedge \ldots \wedge c_n$ (gdzie c_i jest postaci $x_j \vee x_k \vee x_l$) należy skonstruować graf o $3n$ wierzchołkach (każdy wierzchołek odpowiada wystąpieniu w zmiennej w klauzuli) i następującym zbiorze krawędzi:

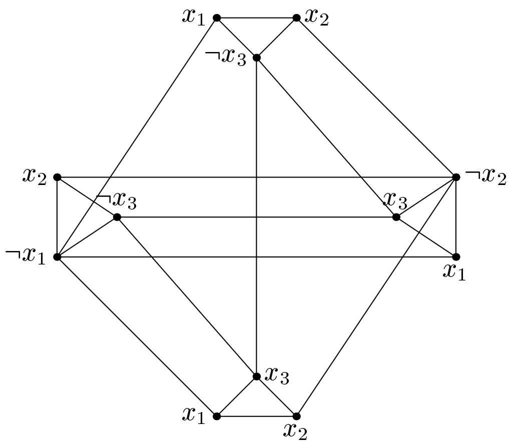
  - $v_{i,j}$ — oznacza wierzchołek przypisany do klauzuli c_i, i j-tej zmiennej ($j \in \{1, 2, 3\}$),
  - dla $j \neq k$, $(v_{i,j}, v_{i,k}) \in E$,
  - dla $i \neq j$, $(v_{i,k}, v_{j,l}) \in E$, jeśli $v_{i,j}$, $v_{j,l}$ oznaczają tą samą zmienną, lecz jeden z wierzchołków oznacza wystąpienie zmiennej bez negacji, drugi z negacją.

  Teraz pozostaje jedynie uruchomienie algorytmu dla sprawdzającego czy w tak skonstruowanym grafie istnieje zbiór niezależny o rozmiarze n. Jeśli taki zbiór istnieje, to formuła jest spełnialna, wpp nie.

  Przykład: Dla formuły:

  $$(x_1 \vee x_2 \vee \neg\, x_3) \wedge (x_1 \vee \neg\, x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg\, x_1 \vee x_2 \vee \neg\, x_3)$$

  zostanie skonstruowany następujący graf:



Problem pokrycia wierzchołkowego

Dane: graf $G = (V, E)$, k
Wynik: TAK, jeśli istniej $C \subseteq V$, $|C| \leq k$, t.że, $\forall_{e=(a,b)\in E}\{a, b\} \cap C \neq \emptyset$

Problem jest dualny do problemu maksymalnego zbioru niezależnego. Wystarczy zauważyć własność: dla grafu $G = (V, E)$, zbiór $C \subseteq V$ jest pokryciem wierzchołkowym G wtw, gdy zbiór $V - C$ jest zbiorem niezależnym G. Stąd jeśli dla grafu G minimalne pokrycie wierzchołkowe ma rozmiar k, to maksymalny zbiór niezależny ma rozmiar $|V| - k$.