

Vertex Cover

Fie $X = \{x_1, x_2, \dots, x_n\}$ o mulțime de variabile de tip bool. Numim formulă booleană peste mulțimea X o formulă CNF (conjunctive normal form) o expresie de forma $C_1 \wedge C_2 \wedge \dots \wedge C_m$ unde fiecare predicat (clause) C_i este o disjuncție a unui număr de variabile (e alcătuit din mai multe variabile cu simbolul \vee - logical or - între ele).

Exemplu de astfel de expresie:

$$(x_1 \vee x_3 \vee x_4) \wedge (x_2 \vee x_3 \vee x_7) \wedge (x_1 \vee x_5 \vee x_6) \wedge (x_2 \vee x_5 \vee x_7).$$

Evident că orice expresie de acest tip va fi evaluată cu "true" dacă toate elementele lui X iau valoarea true. Ne interesează în schimb, să aflăm un număr minim de elemente din X care trebuie să aibă valoarea *true* astfel încât toată expresia să fie *true*.

Fie următorul algoritm pentru problema în forma 3CNF

Greedy-3CNF(C, X)

1: $C = \{C_1, \dots, C_m\}$ mulțimea de predicate, $X = \{x_1, \dots, x_n\}$ - mulțime de variabile

2: cât timp $C \neq \emptyset$ execută

3: Alegem aleator $C_j \in C$.

4: Fie x_i una dintre variabilele din C_j .

5: $x_i \leftarrow \text{true}$.

6: Eliminăm din C toate predicatele ce îl conțin pe x_i .

7: return X

- a) Analizați factorul de aproximare (worst case) al algoritmului (0,5 p)
- b) Modificați algoritmul de mai sus, astfel încât acesta să fie un algoritm 3-aproximativ pentru problema inițială (și justificați) (0,5p)
- c) Reformulați problema de mai sus sub forma unei probleme de programare liniară (0,5p)
- d) Dați o soluție 3-aproximativă pentru problema de programare liniară (0,5p)

Rezolvare:

- a) Fie tripletele (x_i, x_i, x_i) , cu i luând valori între 1 și n .

Oricum am lua perechile, algoritmul se va termina în n pași, deci algoritmul este n -aproximativ în worst case.

- b) 1: $C = \{C_1, \dots, C_m\}$ mulțimea de predicate, $X = \{x_1, \dots, x_n\}$ - mulțime de variabile
 2: cât timp $C \neq \emptyset$ execută
 3: Alegem aleator $C_j \in C$.
 4: Fie x_a, x_b, x_c una dintre variabilele din C_j .
 5: $x_a \leftarrow \text{true}, x_b \leftarrow \text{true}, x_c \leftarrow \text{true}$.
 6: Eliminăm din C toate predicatele ce conțin pe x_a, x_b, x_c
 7: return X

Cum la fiecare pas se selectează cel mult 3 elemente, față de algoritmul precedent unde se selecta aleatoriu un singur element. Prin urmare, algoritmul este 3-aproximativ.

- c) Fie $X = \{x_1, \dots, x_n\}$ o mulțime de numere reale și $C = \{C_1, \dots, C_m\}$ o mulțime de predicate cu condițiile:
 - $0 \leq x_i \leq 1, \forall i \in \{1, \dots, n\}$
 - suma elementelor din fiecare clauza este mai mare sau egală cu 1

Minimizați suma

$$S = \sum_{1 \leq i \leq n} x_i .$$

d) Fie $f: \mathbb{R} \rightarrow \{0,1\}$, $f(x) = \begin{cases} 0, & \text{dacă } x < \frac{1}{3} \\ 1, & \text{altfel} \end{cases}$

Din a doua condiție a enunțului știm că cel puțin un element din cele 3 este mai mare sau egal decât $\frac{1}{3}$. Prin urmare,

$$ALG = \sum_{1 \leq i \leq n} f(x_i) \leq \sum_{1 \leq i \leq n} 3x_i \leq 3 \sum_{1 \leq i \leq n} x_i \leq 3 OPT$$

Deci algoritmul este 3-aproximativ.