Vertex Cover

Bianca-Mihaela Stan, grupa 232

March 2021

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

Exemplu de astfel de expresie:
$(x_1 \lor x_3 \lor x_4) \land (x_2 \lor x_3 \lor x_7) \land (x_1 \lor x_5 \lor x_6) \land (x_2 \lor x_5 \lor x_7)$

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 in forma 3CNF:

```
Greedy-3CNF(C, X)
    C = {C1, . . . , Cm} mulțimea de predicate, X = {x1, . . . , xn} mulțime de variabile
    Cât timp C != multimea vida execută:
        Alegem aleator Cj din C.
        Fie xi una dintre variabilele din Cj .
        xi = true
        Eliminăm din C toate predicatele ce il conțin pe xi .
    return X
```

a) Analizați factorul de aproximare (worst case) al algoritmului (0,5 p)

Un exemplu de worst case:

$$(x_2 \lor x_1) \land (x_3 \lor x_1) \land ... \land (x_n \lor x_1)$$

Rezultatul optim este E=x_1 (multimea variabilelor care trebuie sa fie true) si are cardinalul 1. Algoritmul poate sa nu aleaga niciodata variabila x_1, caz in care rezultatul va fi E'=$x_2, x_3, .., x_n$ cu cardinalul n-1.

Deci, intr-un worst case, factorul de approximare este n-1, unde n este cardinalul lui X.

b) Modificați algoritmul de mai sus, astfel încât acesta să fie un algoritm 3-aproximativ pentru problema inițială (si justificati) (0,5p)

```
Greedy-3CNF(C, X)
    C = {C1, . . . , Cm} mulțimea de predicate, X = {x1, . . . , xn} mulțime de variabile
    Cât timp C != multimea vida execută:
        Alegem aleator Cj din C.
        Cat timp Cj != multimea vida:
            Fie xi una dintre variabilele din Cj .
            xi = true
            Elimin xi din Cj.
            Eliminăm din C toate predicatele ce il conțin pe xi .
    return X
```

Demosnstratie ca algoritmul de mai sus este 3-aproximativ.
Fie:

- OPT = multimea variabilelor care trebuie sa fie true in solutia optima a problemei.
- E = multimea variabilelor care trebuie sa fie true din solutia algoritmului propus de mine

Presupunem ca algoritmul nostru este 3-aproximativ. Adica $|E| \leq 3 * |OPT|$. Vom demonstra prin inductie.

Pasul 1: verificare
C=$(x_1 \vee x_2 \vee x_3)$
In acest caz: $|OPT| = 1$
Algoritmul meu obtine $|E| = 3$. Deci se verifica $|E| \leq 3 * |OPT|$.

Pasul 2: inductie
Presupunem ca pana acum, pentru fiecare variabila aleasa care se afla si in OPT, s-au adaugat maxim 3 variabile in E.

Se alege o clauza din C. Daca aceasta clauza nu a fost eliminata pana acum, inseamna ca contine o variabila din OPT care nu se afla in E. Pentru aceasta variabila, algoritmul nostru va adauga maxim 3 variabile si va elimina clauzele rezolvate de variabilele adaugate.

Deci, prin inductie, $|E| \leq 3 * |OPT|$. Evident $|OPT| \leq |E|$. Deci:

$$|OPT| \leq |E| \leq 3 * |OPT|$$

si algoritmul este 3-aproximativ.

c) Reformulati problema de mai sus sub forma unei probleme de programare liniară (0,5p)

$X = x_1, x_2, .., x_n$ cu proprietatea ca $x_i \in \mathbb{R}$.
Trebuie sa minimizam

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

Pentru fiecare clauza, cu $k \leq 3$ variabile, cel putin una trebuie sa fie true, adica:

$$\sum_{i=1}^{k} x_i \geq 1 \qquad (1)$$

si pentru orice i:

$$0 \leq x_i \leq 1 \qquad (2)$$

d) Dați o soluție 3-aproximativă pentru problema de programare liniara (0,5p)

Cum proprietatea de true sau false nu poate fi modelata cu numere reale, vom transforma x_i in $f(x_i)$, unde f este:

f(x) = 1, daca $x \geq 1/3$
0, daca $x < 1/3$

Acum vom justifica ca este 3-aproximativa.

Definesc:

- OPT : solutia optima pentru problema noastra
- $OPT_{relaxed}$: solutia problemei de programare liniara in cazul in care acceptam si valori reale pentru x_i.

- S : multimea de variabile care trebuie sa fie true ce rezulta din rezolvarea problemei de programare liniara propusa de mine.
- ALG : valoarea rezultata in urma rularii problemei de programare liniara propusa de mine

Algoitmul nostru are valoarea finala:

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

$f(x_i) = 1$ doar pentru $v_i \in S$. Deci:

$$\sum_{1 \leq i \leq n} f(x_i) = \sum_{v_i \in S} x_i = \sum_{v_i \in S} 1$$

Daca $v_i \in S$ inseamna ca $x_i \geq 1/3 => 3 * x_i \geq 1$. Deci:

$$ALG = \sum_{v_i \in S} 1 \leq \sum_{v_i \in S} 3x_i \leq 3 \sum_{v_i \in S} x_i$$

Evident, suma x-urilor din S este mai mica sau egala decat suma tuturor x-urilor.

$$\sum_{v_i \in S} x_i \leq \sum_{1 \leq i \leq n} x_i$$

Deci:

$$ALG \leq 3 \sum_{v_i \in S} x_i \leq 3 \sum_{1 \leq i \leq n} x_i$$

Dar noi stim ca $\sum_{1 \leq i \leq n} x_i = OPT_{relaxed}$, adica solutia optima a problemei daca acceptam solutii in multimea numerelor reale.
De asemenea, stim solutia optima pentru problema noastra este varianta 1/0 a problemei de programare liniara. Cum insa aceasta este NP-hard nu o putem folosi. Insa restrangerea conditiilor de la 1/0 la intervalul [0,1] nu poate decat imbunatati solutia. Asadar, $OPT_{relaxed} \leq OPT$. Avem deci ca:

$$ALG \leq 3 \sum_{1 \leq i \leq n} x_i \leq 3OPT_{relaxed} \leq 3OPT$$

Evident, algoritmul nostru nu da o solutie mai buna decat opt pentru ca altfel OPT nu ar mai fi optim. Deci:

$$OPT \leq ALG \leq 3OPT$$