Petrescu Cosmin
Grupa 243

Tema 1 – Algoritmi avansati

Problema 4 – Vertex Cover

Fie X={x1 , x2 , ..., xn } o mulțime de variabile de tip bool. Numim formulă booleană peste mulțimea X o formulă CNF (conjunctive normal form) o expresie de forma C1∧C2∧...∧Cm unde fiecare predicat (clause) Ci este o disjuncție a unui număr de variabile (e alcătuit din mai multe variabile cu simbolul ∨ - logical or - între ele). Exemplu de astfel de expresie:

(x1 ∨ x3 ∨ x4) ∧ (x2 ∨ x3 ∨ x7) ∧ (x1 ∨ x5 ∨ x6)∧ (x2 ∨ x5 ∨ x7)

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)

1: C = {C1 , . . . , Cm} mulțimea de predicate, X = {x1 , . . . , xn} - mulțime de variabile

2: cât timp C ≠ ∅ execută

3: Alegem aleator Cj ∈ C.

4: Fie xi una dintre variabilele din Cj .

5: xi ← true.

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

7: return X

a) Analizați factorul de aproximare (worst case) al algoritmului.

Sa presupunem cazul in care variabila x_i apartine doar predicatului C_i, pentru orice i ≤ m. Desigur, pentru a completa restul expresiei, vom mai avea nevoie de cel putin doua variabile, care pot aparea in orice predicat. Sa presupunem ca la pasul 4, algoritmul alege intotdeauna pentru predicatul C_i, variabila x_i. Deci, rezulta faptul ca pentru aceasta situatie, agoritmul va asigna true primelor m (= numarul de predicate) variabile. Deoarece aceasta este un exemplu worst-case, putem spune ca factorul de aproximare este chiar m.

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

Petrescu Cosmin
Grupa 243

Fie algoritmul urmator:

Greedy-3CNF(C, X)

1: C = {C1 , . . . , Cm} mulțimea de predicate, X = {x1 , . . . , xn} - mulțime de variabile

2: cât timp C ≠ ∅ execută

3: Alegem aleator Cj ∈ C.

4: Fie x_a, x_b, x_c variabilele din Cj .

5: $x_a = x_b = x_c = true$

6: Eliminăm din C toate predicatele care contin cel putin una din cele 3 variabile .

7: return X

Lema:

Fie OPT numarul minim de variabile care trebuie sa aiba valoarea true astfel incat expresia sa devina true. Fie C' ⊂ C o multime de predicate element-disjuncte. Atunci, rezulta ca OPT ≥ |C'|.

Demonstratie: Fie S – un set de variabile de cardinal minim. Fiecare variabila din S va evalua ca true cel mult un predicat din C'. Deci |S|≥|C'|.

In primul rand, trebuie sa aratam ca solutia algoritmului este corecta (adica evalueaza expresia finala la true).

Justificare: Algoritmul elimina la fiecare pas cate un predicat pana cand nu mai exista niciunul si seteaza variabilele ce ii apartin ca true. Deci, toate predicatele se vor evalua ca true.

Trebuie sa aratam ca |S| ≤ 3OPT:

Fie C* multimea de predicate selectate la pasul 3 al algoritmului. C* este o multime de predicate element-disjuncte, deoarece odata ce un predicat de forma $(x_a \vee x_b \vee x_c)$ este adaugat la C*, toate celelalte predicate care contin cel putin unul din cele 3 variabile sunt sterse si nu vor fi niciodata selectate pentru a intra in C*.

Conform lemei de mai sus, OPT ≥ |C*| (deoarece predicatele din C* sunt element-disjuncte).
Dar, |C*| = $\frac{1}{3}$ |S| (deoarece 3 variabile apartin unui singur predicat). Deci, OPT ≥ $\frac{1}{3}$ |S| ⇒ 3OPT ≥ S.