

Readme

Andrei Gabriel 323CBa

Tema 2 AA Reducere k-Clique \leq_p SAT

Pentru rezolvarea problemelor am stocat datele citite din fisier intr-o matrice,
 $\text{graph}[v1][v2] = \text{true} \Leftrightarrow v1$ este adiacent cu $v2$, nefiind eficienta d.p.d.v al memoriei(o matrice de n^2).

Backtracking apartine $O(n^k)$

Pentru reducerea k-Clique la SAT cu graful $G = (V, M)$, V -multimea nodurilor, M -multimea muchiilor,
am considerat $n \cdot k$ variabile de forma :

x_{iv} , unde i apartine $[1, k]$ si v apartine V ; n = numarul de noduri

$x_{iv} = \text{true}$ daca v apartine lui k-Clique cu indicele i in k-Clique

Reducerea consta in urmatoarele codificari:

- pe pozitia i in k-Clique exista (cel putin) un nod

astfel rezulta clauza: $(x_{i1} \vee x_{i2} \vee \dots \vee x_{in})$

- un nod care este in k-Clique nu poate sa apara pe alta pozitie in clique si o pozitie este ocupata doar de un nod

astfel pentru pozitiile i si j ($i \neq j$; i, j pozitii din k-Clique) rezulta urmatoarea clauza:

$(\neg x_{iv} \vee \neg x_{jv})$

- daca v este in k-Clique doar nodurile adiacente pot fi in k-Clique (sunt k noduri adiacente care apartin k-Clique)

- astfel daca v apartine k-Clique nodurile care nu sunt adiacente cu v nu pot apartine lui k-Clique rezulta clauza:

$\neg x_{iv} \vee (\neg x_{jw1} \vee \neg x_{jw2} \vee \dots \vee \neg x_{jwm})$; unde $j \in [1, k]$, $j \neq i$, w apartine multimii nodurilor care nu sunt adiacente cu v .

sau

- pe pozitia i in k -Clique se afla $v \Rightarrow$ pe pozitia j se afla neaparat un nod adiacent cu v
($\sim x_{i1} \vee x_{i2} \vee \dots \vee x_{i,j} \vee \dots \vee x_{i,k}$) ; unde $j \in [1, k]$, $j \neq i$, u apartine multimii nodurilor care sunt adiacente cu v

Aceasta reducere are timp de calcul polinomial, apartine lui $O(n^2 * k^2)$

Algoritmul backtracking rezolva problemele intr-un timp foarte scurt (mai putin de 1s) pentru inputurile date, datorita valorii lui k care este relativ mica si favorizeaza timpul.

In cazul reducerii, se rezolva in aproximativ 14 secunde. Timpul reducerii propriu-zise este mai mic decat timpul backtracking-ului, dar timpul

rezolvarii SAT va creste foarte mult datorita numarului mare de clauze $2 * n * (k-1) * k + k$ si nk variabile.

In concluzie, pentru un input de date cu valori mai mari (in special pentru un numar k foarte mare), reducerea va avea rezultate mai bune ca backtrackingul.