Vertex cover

Όταν ένα πρόβλημα είναι NP-πλήρες, δεν περιμένουμε να υπάρχει αλγόριθμος πολυωνυμικού χρόνου που να απαντάει σωστά σε όλες τις εισόδους. Χρησιμοποιώντας σαν παράδειγμα το πρόβλημα vertex cover θα δούμε τρεις διαφορετικούς τρόπους αντιμετώπισης ενός προβλήματος που είναι NP-πλήρες.

Vertex cover. *Δίνεται μη κατευθυνόμενος γράφος $G = (V, E)$, με $|V| = n$ και $|E| = m$. Ζητείται σύνολο κόμβων $C \subseteq V$ ελάχιστου μεγέθους που να καλύπτει όλες τις ακμές, δηλαδή,*

$$(\forall (u, v) \in E)(u \in C \text{ ή } v \in C).$$

Ορίσαμε το πρόβλημα ως πρόβλημα βελτιστοποίησης και όχι ως πρόβλημα απόφασης. Είναι ξεκάθαρο πως να χρησιμοποιηθούν οι παρακάτω αλγόριθμοι για το αντίστοιχο πρόβλημα απόφασης.

Ακριβής αλγόριθμος.

Σε αυτή την ενότητα δίνουμε έναν αλγόριθμο που απαντάει σωστά σε κάθε είσοδο. Ο χρόνος εκτέλεσης θα είναι μη πολυωνυμικός σε κάποιες εισόδους. Θα δούμε όμως ότι είναι σημαντικά καλύτερος από τον τετριμμένο αλγόριθμο που ελέγχει κάθε υποσύνολο και χρειάζεται χρόνο $O(2^n)$.

Ο αλγόριθμος βασίζεται στην παρακάτω παρατήρηση, όπου το C είναι ένα βέλτιστο vertex cover και, για $u \in V$, $N(u) = \{v : (u, v) \in E\}$ είναι η *γειτονιά* του u.

Παρατήρηση.

- Αν $u \notin C$, τότε $N(u) \subseteq C$.
- Αν $u \in C$, τότε υπάρχει $v \in N(u)$ τέτοιο που $v \notin C$.

Το πρώτο ισχύει γιατί κάθε ακμή στην γειτονιά του u πρέπει να καλυφθεί από τον αντίστοιχο γείτονα του u (αφού $u \notin C$) και το δεύτερο γιατί αλλιώς και το $C' = C \setminus \{u\}$ θα ήταν vertex cover (το οποίο αντιτίθεται στην υπόθεση ότι το C είναι βέλτιστο).

Ο παρακάτω αλγόριθμος εκμεταλλεύεται αυτή την παρατήρηση. Για κόμβο u, γράφουμε $N'(u) = N(u) \cup \{u\}$ για την κλειστή γειτονιά του και $d(u) = |N(u)|$ για τον βαθμό του.

VC1(G):

1) Έστω $u \in V$ ελάχιστου βαθμού.
2) Return $\min_{v \in N'(u)}\{d(v) + VC1(G \setminus N'(v))\}$.

Η ορθότητα του αλγορίθμου προκύπτει από το ότι για κάθε κόμβο u, ξέρουμε (από την παρατήρηση) ότι τουλάχιστον ένας κόμβος της κλειστής γειτονιάς του δεν ανήκει στο βέλτιστο vertex cover. Άρα, αν δοκιμάσουμε κάθε μία από τις $|N'(u)|$ επιλογές, θα πρέπει να βρούμε τη βέλτιστη λύση. Ο λόγος που επιλέγουμε κόμβο ελάχιστου βαθμού θα αποκαλυφθεί στην ανάλυση της πολυπλοκότητας του αλγορίθμου.

Για την χρονική πολυπλοκότητα, αν $T(n)$ είναι άνω φράγμα στον αριθμό των υποπροβλημάτων που χρειάζεται να λύσουμε για γράφο με n κόμβους, τότε

$$T(n) \leq 1 + \sum_{v \in N'(u)} T(n - d(v) - 1) \leq 1 + (d(u) + 1)T(n - d(u) - 1). \tag{1}$$

Στην δεύτερη ανισότητα χρησιμοποιήσαμε ότι ο u είναι ελάχιστου βαθμού και άρα, για όλα τα $v \in N'(u)$,

$$d(u) \le d(v) \Longrightarrow n - d(v) - 1 \le n - d(u) - 1 \Longrightarrow T(n - d(v) - 1) \le T(n - d(u) - 1).$$

Για την τελευταία συνεπαγωγή χρησιμοποιήσαμε ότι η $T(n)$ είναι αύξουσα. Τώρα, θέτοντας $k = d(u) + 1$ στην (1), έχουμε για την αναδρομή $T'(n) = 1 + kT'(n - k)$ με $T'(1) = 0$,

$$T'(n) \le 1 + k + k^2 + \cdots + k^{n/k} = O(k^{n/k}).$$

Δεν είναι δύσκολο να ελέγξει κανείς ότι η $f(k) = k^{1/k}$ μεγιστοποιείται στου ακεραίους για $k = 3$. Προκύπτει ότι η χρονική πολυπλοκότητα του $VC1$ είναι $O(3^{n/3})$. (Για να γίνει η παραπάνω ανάλυση πιο αυστηρή, απομένει να δειχθεί (π.χ. με επαγωγή) ότι η $T(n)$ είναι αύξουσα και ότι $T(n) = (3^{n/3})$.

Παραμετρικός αλγόριθμος.

Σε αυτή την ενότητα δίνουμε ακόμη έναν αλγόριθμο $VC2$ που απαντάει σωστά σε κάθε είσοδο, οπότε πάλι ο χρόνος εκτέλεσης θα είναι μη πολυωνυμικός σε κάποιες εισόδους. Το πλεονέκτημα του $VC2$ είναι ότι αν η βέλτιστη λύση είναι μικρή, τότε τερματίζει γρήγορα. Π.χ. αν το βέλτιστο vertex cover είναι $O(\log_2 n)$, τότε ο αλγόριθμος τερματίζει σε πολυωνυμικό χρόνο.

Εδώ εκμεταλλευόμαστε την απλή παρατήρηση «για κάθε $(u, v) \in E$, $u \in C$ ή $v \in C$» και εκφράζουμε τον αλγόριθμο και πάλι αναδρομικά.

VC2(G):

1) Έστω $(u, v) \in E$.
2) Return $1 + \min\{VC2(G \setminus \{u\}), VC2(G \setminus \{v\})\}$.

Το «βάθος» της αναδρομής είναι το μέγεθος της λύσης. Οι αναδρομικές κλήσεις λοιπόν του αλγορίθμου αναπαρίστανται από δυαδικό δένδρο ύψους OPT. Προκύπτει ότι γίνονται το πολύ 2^{OPT} αναδρομικές κλήσεις και η χρονική πολυπλοκότητα του αλγορίθμου είναι $O(m2^{OPT})$.

Προσεγγιστικός αλγόριθμος.

Τέλος, δίνουμε έναν απλό πολυωνυμικό αλγόριθμο. Ο αλγόριθμος αυτός δεν βρίσκει την βέλτιστη λύση, αλλά το ενδιαφέρον είναι ότι μπορούμε να δείξουμε πως το σύνολο που βρίσκει δεν είναι ποτέ διπλάσιο του βέλτιστου!

VC3(G): Επαναλαμβάνουμε μέχρι $E = \emptyset$.

1) $C = \emptyset$.
2) Έστω $(u, v) \in E$.
3) $C \leftarrow C \cup \{(u, v)\}$.
4) $E \leftarrow E \setminus \{(x, y) : \{x, y\} \cap \{u, v\} \ne \emptyset\}$.

Αν ο αλγόριθμος επιλέξει k ακμές M, τότε βρίσκει vertex cover μεγέθους $2k$. Η σημαντική παρατήρηση είναι ότι κάθε vertex cover πρέπει να περιέχει τουλάχιστον έναν κόμβο από κάθε ακμή του M. Ο λόγος είναι ότι οι ακμές στο M είναι ξένες μεταξύ τους. Προκύπτει ότι κάθε vertex cover έχει μέγεθος k.