Θέματα Εφαρμογών Βάσεων Δεδομένων: Ιδιωτικότητα Δεδομένων

3. Δυναμικός Προγραμματισμός

Ζαγορίσιος Παναγώτης
Παπαοικονόμου Χριστίνα

Δυναμικός Προγραμματισμός

- Μέθοδος επίλυσης σύνθετων προβλημάτων.
- Όπως και η μέθοδος “διαίρει και βασίλευε”,ο δυναμικός προγραμματισμός επιλύει προβλήματα συνδιάζοντας λύσεις υποπροβλημάτων.
- ***Σημαντική διαφορά των μεθόδων:*** Η “διαίρει και βασίλευε” χωρίζει ένα πρόβλημα σε πολλά *ανεξάρτητα* υπο-προβλήματα. Ο δυναμικός προγραμματισμός είναι εφαρμόσιμος όταν τα διάφορα υποπροβλήματα **δεν** είναι ανεξάρτητα(= έχουν κοινά υποπροβλήματα).

Δυναμικός Προγραμματισμός

- Ενώ ένας αλγόριθμος “διαίρει και βασίλευε” θα έλυνε κοινά υποπροβλήματα ξανά και ξανά. Αντίθετα ένας αλγόριθμος δυναμικού προγραμματισμού λύνει κάθε υποπρόβλημα **ακριβώς μια φορά** και καταχωρεί τη λύση του σε έναν πίνακα απ΄ όπου μπορεί να τη διαβάσει κάθε φορά που το πρόβλημα ξανασυναντιέται.

- Εφαρμόζεται κατά κανόνα σε προβλήματα βελτιστοποίησης.
  - Πολλές διαφορετικές λύσεις
  - Μας ενδιαφέρει η λύση με την βέλτιση τιμή (ελάχιστη/μέγιστη)
    - Η λύση που εξασφαλίζει την βέλτιστη τιμή καλείται βέλτιστη λύση

Βασική Ιδέα

- “σπάσιμο” του αρχικού προβλήματος σε απλούστερα υποπροβλήματα.
- επίλυση κάθε υποπροβλήματος ακριβώς μια φορά και αξιοποίηση των υπολύσεων τους στην εύρεση λύσεων μεγαλύτερων υποπροβλημάτων.
- αποτέλεσμα, ο υπολογισμός της συνολικής βέλτιστης λύσης.(με μικρό υπολογιστικό κόστος)

Βήματα

- Χαρακτηρίζουμε την δομή μιας βέλτιστης λύσης
- Ορίζουμε αναδρομικά την τιμή της βέλτιστης λύσης
- Υπολογίζουμε την τιμή μιας βέλτιστης λύσης εργαζόμενοι “bottom-up” (από κάτω προς τα επάνω)
- Κατασκευάζουμε μια βέλτιστη λύση από τα δεδομένα που έχουμε υπολογίσει.
  - Το βήμα 4 παραλείπεται αν το ζητούμενο είναι η εύρεση της τιμής μόνο μιας βέλτιστης λύσης

Principle of Optimality(1/2)

«Οποιαδήποτε υπακολουθία της βέλτιστης ακολουθίας αποφάσεων είναι επίσης βέλτιστη για το υποπρόβλημα που αντιστοιχεί στη συγκεκριμένη υπακολουθία»

- Ο Δ.Π αποτελεί γενίκευση της greedy μεθόδου που χρησιμοποείται για προβλήματα των οποίων η λύση μπορεί να θεωρηθεί σαν μια ακολουθία αποφάσεων.

- Σε πολλά προβλήματα που οι αποφάσεις εξαρτώνται από “τοπικές” πληροφορίες οδηγούν σε βέλτιστες λύσεις (greedy τεχνικές)

Principle of Optimality(2/2)

- Σε άλλα, όμως, δεν ισχύει. Γι αυτό θα πρέπει να δημιουργηθούν όλες οι σειρές των αποφάσεων και να επιλεχθεί η καλύτερη.
- Εδώ είναι που χρειάζεται δυναμικός προγραμματισμός... αποκλείει σειρές αποφάσεων χαρακτηρίζοντάς αυτές ως μη βέλτιστες.

Βασικά Στοιχεία Δυναμικού Προγραμματισμού(1/3)

- Βελτιστη υποδομή: αν μια βέλτιστη λύση εμπεριέχει βέλτιστες λύσεις στα υποπροβλήματα.
  - Greedy αλγόριθμοι: η βέλτιστη υποδομή χρησιμοποιείται top-down και όχι bottom-up
  - Ανεξάρτητα υποπροβλήματα: η λύση σε κάποιο υποπρόβλημα δεν επηρεάζει την λύση σε κάποιο άλλο υποπρόβλημα του ίδιου προβλήματος.
    → δεν έχουν κοινούς πόρους

Βασικά Στοιχεία Δυναμικού Προγραμματισμού(2/3)

- Επικάλυψη προβλημάτων: όταν ο αναδρομικός αλγόριθμος επανέρχεται στο ίδιο πρόβλημα ξανά και ξανά
    - dynamic programming: χρήση πίνακα για αποθήκευση της λύσης
    - 2 υποπροβλήματα επικαλύπτονται εάν είναι το ίδιο υποπρόβλημα που εμφανίζεται ως υποπρόβλημα διαφορετικών προβλημάτων.

Βασικά Στοιχεία Δυναμικού Προγραμματισμού(3/3)

- Ανασύνθεση μια βέλτιστης λύσης
    - Αποθήκευση των επιλογών που κάνουμε σε κάθε υποπρόβλημα ώστε να μην είμαστε αναγκασμένοι να ανασυνθέσουμε αυτές τις πληροφορίες ξανά.

Αλγόριθμος Floyd-Warshall

- Δεδομένου ενός κατευθυνόμενου γράφου G=(V,E) με βάρη, στόχος είναι η εύρεση των ελάχιστων (ελαφρύτατων) διαδρομών μεταξύ όλων των ζευγαριών των κόμβων.
- Ο αλγόριθμος εξετάζει τους ενδιάμεσους κόμβους
    - Function: shortest_path(i,j,k) υπολογίζει και επιστρέφει το συντομότερο μονοπάτι από τον κόμβο i στον j με ενδιάμεσους κόμβους {1,...,k}

Δομή μιας ελαφρύτατης διαδρομής(1/2)

Έστω p διαδρομή ελάχιστου βάρους από τον i → j κόμβο.

- Αν ο k ΔΕΝ είναι ενδιάμεσος κόμβος της διαδρομής p τότε ενδιάμεσοι={1,...,k-1}
  - Μια ελαφρύτατη διαδρομή p από i → j με ενδιάμεσους κόμβους={1,..k-1} αποτελεί επίσης ελαφρύτατη διαδρομή από i → j με ενδιάμεσους κόμους={1,...,k}.

Δομή μιας ελαφρύτατης διαδρομής(2/2)

- Αν k είναι ενδιάμεσος κόμβος της διαδρομής p, τότε αναλύουμε την p στις υποδιαδρομές

  p1: i → k και p2: k → j

  - p1 ελαφρύτατη με ενδιάμεσους={1,...,k-1}
  - p2 ελαφρύτατη με ενδιάμεσους={1,...,k-1}

  – Επομένως,
    - i → k shortest path
    - k → j shortest path

    ⇒ i → j shortest path με ενδιάμεσο κόμβο τον k

Αναδρομική λύση

If w_{ij}: weight of edge between $i \rightarrow j$

$d^{(k)}_{ij}$: το βάρος μιας ελαφρύτατης διαδρομής $i \rightarrow j$ με ενδιάμεσους κόμβους ={1,...,k}

$$d^{(k)}_{ij} = w_{ij} \rightarrow k=0$$

$$d^{(k)}_{ij} = \min \{d^{(k-1)}_{ij}, \; d^{(k-1)}_{ik} + d^{(k-1)}_{kj}\} \rightarrow k \geq 1$$

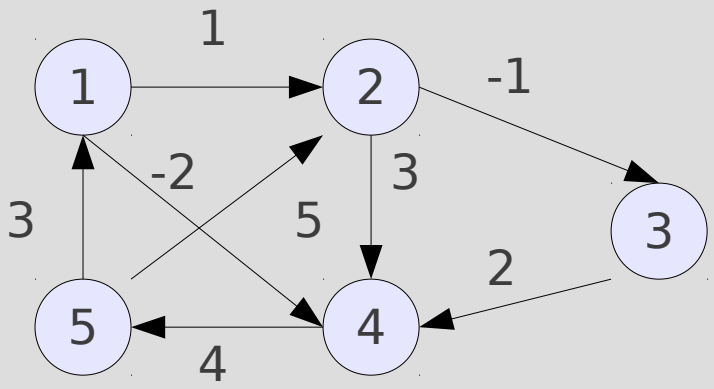
Bottop-up υπολογισμός

n = rows(W)
D(0)= W

for k=1:n
    for i=1:n
        for j=1:n
            $d^{(k)}_{ij} = \min \{d^{(k-1)}_{ij}, d^{(k-1)}_{ik} + d^{(k-1)}_{kj}\}$

return D(n)

Παράδειγμα



D(0)

0	1	∞	-2	∞
∞	0	-1	3	∞
∞	∞	0	2	∞
∞	∞	∞	0	4
3	5	∞	4	0

D(1)

0	1	∞	-2	∞
∞	0	-1	3	∞
∞	∞	0	2	∞
∞	∞	∞	0	4
3	4	∞	1	0

D(2)

0	1	0	-2	∞
∞	0	-1	3	∞
∞	∞	0	2	∞
∞	∞	∞	0	4
3	4	3	1	0

D(3)

0	1	0	-2	∞
∞	0	-1	1	∞
∞	∞	0	2	∞
∞	∞	∞	0	4
3	4	3	1	0

Παράδειγμα

D(4)

0	1	0	-2	2
∞	0	-1	1	5
∞	∞	0	2	6
∞	∞	∞	0	4
3	4	3	1	0

D(5)

0	1	0	-2	2
8	0	-1	1	5
9	10	0	2	6
7	8	7	0	4
3	4	3	1	0