Γραμμική και συνδυαστική βελτιστοποίηση

Τελική εργασία

Θέμα: Shortest path problem

Όνομα: Δημήτρης

Επώνυμο: Μπόκαρης

Αριθμός μητρώου: 1053588

Βιβλιογραφία: <https://en.wikipedia.org/wiki/Shortest_path_problem>

http://wiki.gis.com/wiki/index.php/Shortest\_path\_problem

<https://favtutor.com/blogs/bellman-ford-python>

https://stackabuse.com/courses/graphs-in-python-theory-and-implementation/lessons/dijkstras-algorithm/

Περιγραφή

Στη θεωρία γράφων, το πρόβλημα της συντομότερης διαδρομής είναι το πρόβλημα της εύρεσης μιας διαδρομής μεταξύ δύο κορυφών (ή κόμβων) σε ένα γράφημα έτσι ώστε το άθροισμα των βαρών των ακμών που την αποτελούν να ελαχιστοποιείται. Ένα παράδειγμα είναι η εύρεση του πιο γρήγορου τρόπου μετάβασης από μια τοποθεσία σε μία άλλη σε έναν οδικό χάρτη. Σε αυτήν την περίπτωση, οι κορυφές αντιπροσωπεύουν τοποθεσίες και οι ακμές αντιπροσωπεύουν τμήματα του δρόμου οι οποίες σταθμίζονται με το χρόνο που απαιτείται για τη διάσχισή τους.

Το πρόβλημα ονομάζεται μερικές φορές πρόβλημα συντομότερης διαδρομής ενός ζεύγους, για να το διακρίνουμε από τις ακόλουθες γενικεύσεις:

⦁ Το πρόβλημα συντομότερης διαδρομής μιας πηγής, στο οποίο πρέπει να βρούμε τα συντομότερα μονοπάτια από μια κορυφή πηγής v σε όλες τις άλλες κορυφές του γραφήματος.

⦁ Το πρόβλημα της συντομότερης διαδρομής ενός προορισμού, στο οποίο πρέπει να βρούμε τα συντομότερα μονοπάτια από όλες τις κορυφές του γραφήματος προς μια κορυφή προορισμού v. Αυτό μπορεί να μειωθεί στο πρόβλημα της συντομότερης διαδρομής μιας πηγής αντιστρέφοντας τις ακμές του γραφήματος.

⦁ Το πρόβλημα της συντομότερης διαδρομής όλων των ζευγών, στο οποίο πρέπει να βρούμε τα συντομότερα μονοπάτια μεταξύ κάθε ζεύγους κορυφών v, v' στο γράφημα.

Αυτές οι γενικεύσεις έχουν σημαντικά πιο αποτελεσματικούς αλγόριθμους από την απλοϊκή προσέγγιση της εκτέλεσης ενός αλγορίθμου συντομότερης διαδρομής ενός ζεύγους σε όλα τα σχετικά ζεύγη κορυφών.

Αλγόριθμοι

Οι πιο σημαντικοί αλγόριθμοι για την επίλυση αυτού του προβλήματος είναι:

⦁ Ο αλγόριθμος του Dijkstra λύνει τα προβλήματα συντομότερης διαδρομής ενός ζεύγους, μιας πηγής και ενός προορισμού.

⦁ Ο αλγόριθμος Bellman-Ford λύνει το πρόβλημα μιας πηγής εάν τα βάρη των ακμών μπορεί να είναι αρνητικά.

⦁ Ο αλγόριθμος αναζήτησης A\* επιλύει τη συντομότερη διαδρομή ενός ζεύγους χρησιμοποιώντας ευρετικές μεθόδους για να προσπαθήσει να επιταχύνει την αναζήτηση.

⦁ Ο αλγόριθμος Floyd-Warshall επιλύει όλα τα ζεύγη των συντομότερων μονοπατιών.

⦁ J C Πράσινος αλγόριθμος Ένας ευρετικός αλγόριθμος αναζήτησης που λύνει το πρόβλημα σε μικρότερο χρόνο n από άλλες υλοποιήσεις.

⦁ Ο αλγόριθμος του Johnson λύνει όλα τα ζεύγη των συντομότερων μονοπατιών και μπορεί να είναι ταχύτερος από τον Floyd-Warshall σε αραιά γραφήματα.

⦁ Η θεωρία των διαταραχών βρίσκει (στη χειρότερη περίπτωση) την τοπικά συντομότερη διαδρομή.

⦁ Ο αλγόριθμος Viterbi λύνει το πρόβλημα της συντομότερης στοχαστικής διαδρομής με ένα επιπλέον πιθανοτικό βάρος σε κάθε κόμβο.

Εφαρμογές

Οι αλγόριθμοι συντομότερης διαδρομής εφαρμόζονται για την αυτόματη εύρεση οδηγιών μεταξύ φυσικών τοποθεσιών, όπως οδηγίες πλοήγησης σε ιστότοπους χαρτογράφησης ιστού όπως το Mapquest ή οι Χάρτες Google. Για αυτήν την εφαρμογή είναι διαθέσιμοι γρήγοροι εξειδικευμένοι αλγόριθμοι.

Εάν κάποιος αντιπροσωπεύει μια μη ντετερμινιστική αφηρημένη μηχανή ως γράφημα όπου οι κορυφές περιγράφουν καταστάσεις και οι ακμές περιγράφουν πιθανές μεταβάσεις, μπορούν να χρησιμοποιηθούν αλγόριθμοι συντομότερης διαδρομής για την εύρεση μιας βέλτιστης ακολουθίας επιλογών για την επίτευξη μιας συγκεκριμένης κατάστασης στόχου ή για τον καθορισμό χαμηλότερων ορίων στο χρόνο που απαιτείται για φτάσει σε μια δεδομένη κατάσταση. Για παράδειγμα, εάν οι κορυφές αντιπροσωπεύουν τις καταστάσεις ενός παζλ όπως ο Κύβος του Ρούμπικ και κάθε κατευθυνόμενη άκρη αντιστοιχεί σε μία κίνηση ή στροφή, μπορούν να χρησιμοποιηθούν αλγόριθμοι συντομότερης διαδρομής για να βρεθεί μια λύση που χρησιμοποιεί τον ελάχιστο δυνατό αριθμό κινήσεων.

Σε μια νοοτροπία δικτύωσης ή τηλεπικοινωνιών, αυτό το πρόβλημα της συντομότερης διαδρομής ονομάζεται μερικές φορές πρόβλημα διαδρομής ελάχιστης καθυστέρησης και συνήθως συνδέεται με ένα πρόβλημα ευρύτερης διαδρομής. Για παράδειγμα, ο αλγόριθμος μπορεί να αναζητήσει τη συντομότερη (λεπτή καθυστέρηση) ευρύτερη διαδρομή ή τη μεγαλύτερη συντομότερη διαδρομή (λεπτή καθυστέρηση).

Μια πιο ανάλαφρη εφαρμογή είναι τα παιχνίδια "έξι βαθμών διαχωρισμού" που προσπαθούν να βρουν τη συντομότερη διαδρομή σε γραφήματα όπως οι αστέρες του κινηματογράφου που εμφανίζονται στην ίδια ταινία. Άλλες εφαρμογές περιλαμβάνουν «έρευνα λειτουργίας, διάταξη εγκαταστάσεων και εγκαταστάσεων, ρομποτική, μεταφορές και σχεδιασμός VLSI».

Παράδειγμα 1o

2 8

3

6 16 1

7 4 10

4

5 3

Θεωρούμε ότι τα βάρη των ακμών αναπαριστούν την απόσταση μεταξύ δύο κόμβων.

*Πρόβλημα:* Ποια είναι η πιο σύντομη διαδρομή από τον κόμβο 1 στον κόμβο 7?

*Απάντηση :* 1->3->6->7.

Λύση με γραμμικό προγραμματισμό.

Έστω δυαδική μεταβλητή Xij που δηλώνει την μεταφορά από τον κόμβο i στον κόμβο j.

Οι πιθανές διαδρομές από και προς γειτονικούς κόμβους για κάθε κόμβο , παρατίθονται παρακάτω :

Κόμβος (1): X12 + X13 + X14 = 1

Κόμβος (2): -X12 + X23+X24 + X25= 0

Κόμβος (3): -X13 - X23 + X36 + X35 = 0

Κόμβος (4): -X14 - X24 + X45 + X47 = 0

Κόμβος (5): -X35 - X45 - X25 + X57 = 0

Κόμβος (6): -X36 + X67 = 0

Κόμβος (7): - X67 - X47 - X57 = -1

Η μεταβλητή Xij  παίρνει την τιμή 1 όταν διανύουμε τον συγκεκριμένο σύνδεσμο, 0 όταν δεν ακολουθείται ο συγκεκριμένος σύνδεσμος . Το θετικό πρόσημο που έπεται των μεταβλητών δηλώνει μεταφορά από τον συγκεκριμένο κόμβο σε κάποιον άλλον, ενώ το αρνητικό πρόσημο από κάποιον άλλο κόμβο προς τον συγκεκριμένο.

Αναζητούμε το ελάχιστο της αντικειμενικής συνάρτησης που διαμορφώνεται ώς εξής :

Min Z = 6 X12 +2 X13 + 16 X14 + 7 X23 +5 X24 + 4 X25 + 8 X36 + 3 X35 + 4 X45 + 3 X47 +10 X57 + X67

Χρησιμοποιώντας τον επιλυτή pymprog , η βέλτιστη δυνατή διαδρομή διαμορφώνεται ως εξής : 1->3 ,3->6 ,6->7 με την συνολική απόσταση να ανέρχεται στα Ζ=11.

Παράδειγμα 2o

9 6

4 2 10 5

11 15

20 1 5

7 12

1 3

Θεωρούμε ότι τα βάρη των ακμών αναπαριστούν την απόσταση μεταξύ δύο κόμβων.

*Πρόβλημα:* Βρείτε την συντομότερη διαδρομή από τον κόμβο-πηγή 0 προς οποιοδήποτε κόμβο του γραφήματος.

*Απάντηση:*

⦁ 0-1 -> 4

⦁ 0-2 -> 11

⦁ 0-3 -> 17

⦁ 0-4 -> 9

⦁ 0-5 -> 22

⦁ 0-6 -> 7

⦁ 0-7-> 8

⦁ 0-8 -> 11

Λύση με αλγόριθμο dijkstra .

Πολυπλοκότητα: O(|E|+|V|log|V|)

Σε αυτόν τον αλγόριθμο, περνάμε κάθε ακμή μία φορά, πράγμα που έχει ως αποτέλεσμα χρονική πολυπλοκότητα O(|E|), όπου |E| αντιπροσωπεύει τον αριθμό των ακμών.

Επισκεπτόμαστε επίσης κάθε κόμβο μία φορά, κάτι που έχει ως αποτέλεσμα χρονική πολυπλοκότητα O(|V|), όπου |V| αντιπροσωπεύει τον αριθμό των κορυφών. Κάθε κορυφή θα τεθεί σε μια ουρά προτεραιότητας, όπου η εύρεση της επόμενης πλησιέστερης κορυφής θα γίνει σε σταθερό χρόνο O(1). Ωστόσο, χρησιμοποιούμε το χρόνο O(Vlog|V|) για να ταξινομήσουμε τις κορυφές σε αυτήν την ουρά προτεραιότητας.

Αυτό έχει ως αποτέλεσμα η χρονική πολυπλοκότητα αυτού του αλγορίθμου να είναι O(|E|+|V|log|V|).

Παράδειγμα 3o

Σε αντίθεση με τον αλγόριθμο του Dijkstra, ο αλγόριθμος bellman ford μπορεί επίσης να βρει τη μικρότερη απόσταση σε κάθε κορυφή στο σταθμισμένο γράφημα, ακόμη και με τις αρνητικές ακμές. Η μόνη διαφορά μεταξύ του αλγόριθμου Dijkstra και του αλγόριθμου bellman ford είναι ότι ο αλγόριθμος του Dijkstra απλώς επισκέπτεται τη γειτονική κορυφή σε κάθε επανάληψη, αλλά ο αλγόριθμος bellman ford επισκέπτεται κάθε κορυφή μέσω κάθε ακμής σε κάθε επανάληψη.

Η χρονική πολυπλοκότητα του αλγόριθμου bellman ford για την καλύτερη περίπτωση είναι O(E) ενώ η χρονική πολυπλοκότητα της μέσης και της χειρότερης περίπτωσης είναι O(NE) όπου N είναι ο αριθμός των κορυφών και E είναι οι συνολικές ακμές που πρέπει να χαλαρώσουν. Επίσης, η πολυπλοκότητα χώρου του αλγόριθμου bellman ford είναι O(N) επειδή το μέγεθος του πίνακα είναι N.