Αλγόριθμοι και Πολυπλοκότητα

4η σειρά γραπτών ασκήσεων

Σχέδιο Λύσεων

CoReLab

ΣΗΜΜΥ Ε.Μ.Π.

12 Φεβρουαρίου 2020

Άσκηση 1α: Επιβεβαίωση Συντομότερων μονοπατιών

Ο αλγόριθμος είναι μια απλή παραλλαγή του BFS

- Ξεκινάμε το BFS από την κορυφή v_1 με $p(v_1) = NULL$
- Έστω ότι είμαστε στην κορυφή v_i και έστω v_j γειτονάς του.
- Αν $\delta_i + w_{ij} < \delta_j$ οι αποστάσεις είναι λάθος. Αν $\delta_i + w_{ij} = \delta_j$, σημειώνουμε την κορυφή και ανανεώνουμε τον πίνακα προγόνων $p[j] = i$. Αλλιώς, προχωράμε.
- Αν ολοκληρωθεί η διαδικασία ελέγχουμε τον πίνακα προγόνων εάν έχει σε κάποια θέση εκτός την πρώτη $NULL$. Αν έχει, τότε απορρίπτουμε τις δοθείσες αποστάσεις. Αλλιώς, επιστρέφουμε τον πίνακα προγόνων που αποτελεί το ΔΣΜ.
- Η πολυπλοκότητα είναι $O(n + m)$.

Άσκηση 1β και 1γ: Αναπροσαρμογή Συντομότερων Μονοπατιών

- Μειώνεται το μήκος της ακμής $e = (x, y)$ σε $w'(x, y) < w(x, y)$. Άρα για κάθε ζεύγος κορυφών v_i, v_j υπάρχει περίπτωση να μειωθεί η απόσταση μεταξύ τους μόνο εάν το μήκος του μονοπατιού που περνά από την e είναι μικρότερο από το δοθέν $d(v_i, v_j)$.
- Το μήκος αυτό είναι $d(v_i, x) + w'(x, y) + d(y, v_j)$ και υπολογίζεται προφανώς σε ένα υπολογιστικό βήμα. Έτσι εφόσον έχουμε $O(n^2)$ ζεύγη, η πολυπλοκότητα του αλγορίθμου είναι $O(n^2)$.

Άσκηση 1β και 1γ: Αναπροσαρμογή Συντομότερων Μονοπατιών

- Αν αυξηθεί το μήκος της ακμής $e = (x, y)$ σε $w'(x, y) > w(x, y)$, δεν αρκεί στη γενική περίπτωση να συγκρίνουμε τη δοθείσα απόσταση $d(v_i, v_j)$ με το μήκος **ενός μόνο** μονοπατιού.
  Αν ισχύει πως $d(v_i, x) + w'(x, y) + d(y, v_j) = d(v_i, v_j)$ δεν χρειάζεται να αλλάξουμε τίποτα.
  Αν όμως $d(v_i, x) + w'(x, y) + d(y, v_j) > d(v_i, v_j)$ τότε θα πρέπει αναγκαστικά να συγκρίνουμε το $d(v_i, v_j)$ με το μήκος όλων των υπολοίπων μονοπατιών μεταξύ v_i και v_j.

Κριτήριο

Φτιάχνουμε ένα γράφημα $G(V, E)$ όπου:

- Για κάθε μεταβλητή x_i βάζουμε μία κορυφή v_i
- Για κάθε περιορισμό $x_i - x_j \leq b_{i,j}$ βάζουμε μια ακμή ανάμεσα στα v_i και v_j βάρους $b_{i,j}$

Το σύστημα ανισοτήτων S είναι ικανοποιήσιμο εάν και μόνο εάν το γράφημα G δεν έχει αρνητικό κύκλο

Ορθότητα Κριτηρίου

Αν το γράφημα δεν έχει αρνητικό κύκλο, τότε εφαρμόζοντας τον αλγόριθμο Bellman-Ford από μια εξωτερική κορυφή v που τη συνδέουμε με ακμές βάρους 0 προς όλες τις υπόλοιπες, βρίσκουμε συντομότερες αποστάσεις απ'το v. Αναθέτουμε σε κάθε μεταβλητή x_i τιμή ίση με την απόσταση απ'τη v όπως προκύπτει απ'το δέντρο ελάχιστων διαδρομών. Οι ανισότητες που αντιστοιχούν σε ακμές αυτού του δέντρου ικανοποιούνται με ισότητα. Οι υπόλοιπες ικανοποιούνται λόγω τριγωνικής ανισότητας στις αποστάσεις ελάχιστων διαδρομών. Άρα το S είναι ικανοποιήσιμο με αυτές τις τιμές.

Ορθότητα Κριτηρίου
Αν το γράφημα έχει αρνητικό κύκλο έστω τον $v'_1, v'_2, ..., v'_k, v'_1$ με μήκος $L < 0$, προσθέτοντας τις αντίστοιχες ανισότητες κατά μέλη, έχουμε:

- Στο πρώτο μέλος 0
- Στο δεύτερο μέλος κάτι αρνητικό καθώς ο κύκλος είναι αρνητικού μήκους
- Άρα: $0 \leq L < 0$ (Άτοπο)

Άρα για οποιαδήποτε ανάθεση τιμών στις μεταβλητές, το σύστημα δεν μπορεί να είναι ικανοποιήσιμο

Αποδοτικός Αλγόριθμος

Αλγόριθμος για εύρεση ύπαρξης κύκλου αρνητικού μήκους: Bellman Ford $n-1$ βήματα: Σε κάθε βήμα εξετάζονται όλες οι ακμές και ελέγχουμε εαν μπορούν να μας ελαττώσουν την απόσταση προς κάποια κορυφή. Αν μετά από τα $n-1$ βήματα, εξακολουθεί να υπάρχει ακμή που να ελαττώνει την απόσταση προς κάποια κορυφή, σημαίνει ότι υπάρχει κύκλος αρνητικού μήκους. Άρα τότε το σύστημα δεν είναι ικανοποιήσιμο. Διαφορετικά είναι. Η πολυπλοκότητα είναι $O(n*m)$ ($n-1$ βήματα, σε κάθε βήμα m ακμές)

Άσκηση 2β

Στην περίπτωση που το S είναι ικανοποιήσιμο ο Bellman-Ford κρατάει και τους προγόνους που οδηγούν σε κάθε κορυφή, οπότε έτσι μπορούμε να φτιάξουμε ένα δέντρο ελαχίστων διαδρομών από την x_1. Άρα, θέτοντας $x_1 = 0$ και τις υπόλοιπες μεταβλητές να ισούνται με την απόστασή τους από τη v_1, έχουμε ένα σύστημα που είναι ικανοποιήσιμο, όπως εξηγήσαμε πριν. Η πολυπλοκότητα στην περίπτωση αυτή είναι $O(n * m)$.

Άσκηση 2β

Διαφορετικά το ελάχιστο υποσύστημα που δεν είναι ικανοποιήσιμο υπολογίζεται ως εξής:
$L^{(I)}$: πίνακας συντομότερων μονοπατιών με το πολύ I ακμές.
$P^{(I)}$: πίνακας προγόνων
$L^{(I+1)}(i,j) = min\{L^{(I)}(i,k), L^{(1)}(k,j), L^{(I)}(i,j)\}$
Επομένως
$P^{(I+1)}(i,j) = -1$ αν $min\{L^{(I)}(i,k) + L^{(1)}(k,j)\} > L^{(I)}(i,j)$
διαφορετικά
$P^{(I+1)}(i,j) = argmin_k\{L^{(I)}(i,k) + L^{(1)}(k,j)\}$
Αν κατά τον υπολογισμό του $L^{(I+1)}$, για κάποια (i,j) έχουμε $L^{(I+1)}(i,j) + b_{j,i} < 0$ τότε έχουμε κύκλο αρνητικού μήκους με το πολύ $l+2$ ακμές που μπορούμε να εξάγουμε από τον πίνακα προγόνων. Πολυπλοκότητα: $O(n^4)$ για τον υπολογισμό του $L^{(n)}$. Για κάθε $L^{(I)}$ είναι $O(n^3)$. Μπορεί να βελτιωθεί σε $O(n^3 * logn)$ με χρήση δυαδικής αναζήτησης.

Άσκηση 2γ

$L^{(c)}$: πίνακας συντομότερων μονοπατιών με το πολύ βάρος c $P^{(c)}$: πίνακας προγόνων
$L^{(c+1)}(i,j) = min\{L^{(c-k')}(i,k) + L^{(k'+1)}(k,j), b_{i,j}$ αν $w_{i,j} \leq c+1\}$ Επομένως
$P^{(c+1)}(i,j) = (-1,-1)$ αν
$b_{i,j} < min\{L^{(c-k')}(i,k) + L^{(k'+1)}(k,j)\}$ και $w_{(i,j)} \leq c+1$
διαφορετικά
$P^{(c+1)}(i,j) = argmin_{(k',k)}\{L^{(c-k')}(i,k) + L^{(k'+1)}(k,j)\}$
Αν κατα τον υπολογισμό του $L^{(c+1)}$, για κάποια (i,j) έχουμε $L^{(c+1)}(i,j) + b_{j,i} < 0$ τότε έχουμε κύκλο αρνητικού μήκους.

Άσκηση 2γ

Συγκρίνουμε το βάρος του με το βάρος του ελάχιστου που έχουμε βρει ως τότε και ανανεώνουμε το ελάχιστο. Στο τέλος, από τον πίνακα προγόνων βρίσκουμε σε ποιόν κύκλο αντιστοιχεί ο ελάχιστος (το $(-1, -1)$ σημαίνει ότι αντιστοιχεί σε απευθείας ακμή).
Πολυπλοκότητα: $O(n^3 * C^2)$ για τον υπολογισμό του $L^{(C)}$ όπου C είναι το άθροισμα των βαρών. Άρα ο αλγόριθμός μας είναι ψευδοπολυωνυμικός.

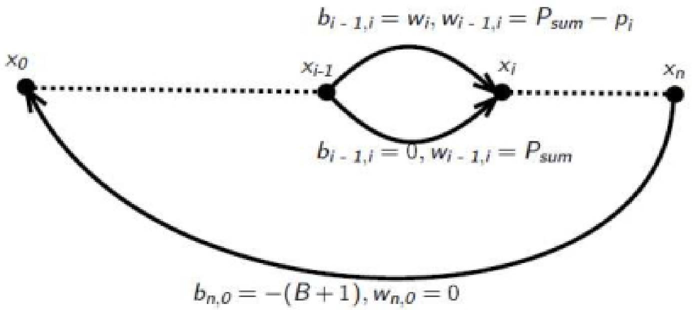
Άσκηση 2δ

Πρόβλημα Απόφασης
Δίνεται ένα σύστημα S με n μεταβλητές $x_1, ..., x_n$ και m ανισότητες της μορφής $x_i - x_j \leq b_{i,j}$, κάθε μία από τις οποίες έχει βάρος $w_{i,j}$.

Υπάρχει μη ικανοποιήσιμο υποσύστημα S' βάρους μικρότερου ή ίσου με C ;

Αναγωγή από Knapsack

Γενικό στιγμιότυπο από Knapsack: n στοιχεία, το i-οστό με βάρος w_i και κέρδος p_i. Υπάρχει υποσύνολό τους με συνολικό βάρος μικρότερο ή ίσο του B και συνολοκό κέρδος μεγαλύτερο ή ίσο του P ; Θα το μετασχηματίσουμε σε ένα ειδικό στιγμιότυπο του προβλήματός μας. Έστω P_{sum} το άθροισμα των κερδών. Για το i-οστό στοιχείο δημιουργώ μία κορυφή x_i. Συνδέω τη x_{i-1} με τη x_i με μία ακμή μήκους w_i και βάρους $P_{sum} - p_i$ και με μία ακμή μήκους 0 και βάρους P_{sum}. (Θεωρούμε και μία αρχική κορυφή x_0). Τέλος, βάζω μία ακμή από το x_n στο x_0 με μήκος $-(B+1)$ και βάρος 0.



Σχήμα: Reduction

Άσκηση 2δ

NP-hard
Κάθε αρνητικός κύκλος σε αυτό το πρόβλημα ισοδυναμεί με μία επιλογή στοιχείων του Knapsack με συνολικό βάρος μικρότερο ή ίσο του B. Τα στοιχεία που επιλέγονται είναι αυτά στα οποία επιλέγεται η πρώτη ακμή. Άρα, αν μπορούσαμε να λύσουμε το πρόβλημα της ύπαρξης αρνητικού κύκλου με βάρος μικρότερο ή ίσο από $n * P_{sum} - P$ που είναι ισοδύναμο με την εύρεση υποσυνόλου συνολικού βάρους μικρότερου ή ίσου του P) τότε μπορούμε να λύσουμε και το γενικό στιγμιότυπο του Knapsack. Όμως το τελευταίο είναι NP-complete. Άρα το πρόβλημά μας είναι τουλάχιστον τόσο δύσκολο (NP-hard).

Άσκηση 2δ

NP-complete
Προφανώς το πρόβλημα ανήκει στο NP καθώς εύκολα μπορούμε να επαληθεύσουμε αν ένα δοθέν υποσύνολο είναι μη ικανοποιήσιμο (αποτελεί αρνητικό κύκλο) και έχει βάρος μικρότερο ή ίσο του C.

Συνεπώς αφού ανήκει στο NP και είναι NP-hard, είναι NP-complete.

Άσκηση 3α

- Εφαρμόζουμε τον αλγόριθμο Dijkstra στο γράφημα G προκειμένου να υπολογίσουμε τις αποστάσεις $d(c_i)$ όλων των πόλεων $c_i \in C$ που διαθέτουν σταθμό φόρτισης από τον αρχικό κόμβο s.
- Δημιουργούμε ένα νέο γράφημα G' που είναι ουσιαστικά το G και στο οποίο έχουμε αντιστρέψει τη φορά όλων των ακμών. Ως αρχικό κόμβο θεωρούμε τώρα τον t.
- Εφαρμόζουμε ξανά τον Dijkstra στο G' προκειμένου να υπολογίσουμε τις αποστάσεις $d'(c_i)$ όλων των πόλεων που διαθέτουν σταθμό φόρτισης από τον αρχικό κόμβο t.
- Αναζητούμε το $min_{i \in C}(d(c_i) + d'(c_i))$
- Η πολυπλοκότητα είναι $O(mlogn)$ ή $O(m + nlogn)$.

Άσκηση 3β

- Για κάθε $c_i \in C$ υπολογίζουμε τα συντομότερα μονοπάτια από το s στο c_i και από το c_i στο t καθώς και από το c_i στο c_j για κάθε $c_i, c_j \in C$. Τέλος βρίσκουμε το συντομότερο μονοπάτι από το s στο t.
- Δημιουργούμε νέο γράφο με κόμβους όλες τις πόλεις που διαθέτουν σταθμό φόρτισης (s, t, c_i). Προσθέτω ως ακμές τα συντομότερα μονοπάτια που υπολόγισα αρκεί να μην ξεπερνούν το α. Εφαρμόζω Dijkstra στο νέο γράφο και παίρνω το συντομότερο μονοπάτι από το s στο t
- Η πολυπλοκότητα είναι $O(|C|mlogn)$ ή $O(|C|(m + nlogn))$.

Άσκηση 4α και 4β

- Για το (α): Δημιουργούμε το υπολειμματικό δίκτυο σε γραμμικό χρόνο και με ένα BFS (πάλι σε γραμμικό χρόνο) ελέγχουμε εάν υπάρχει επαυξητικό μονοπάτι. Αν δεν υπάρχει, τότε και μόνο τότε η δοθείσα ροή είναι μέγιστη.
- Για το (β): Θέλουμε να βρούμε μια εφικτή ροή στο νέο γράφημα και μετά να χρησιμοποιήσουμε τον αλγόριθμο Ford-Fulkerson ξεκινώντας από τη ροή αυτή για να φτάσουμε στη μέγιστη.
- Αν $c'_e \geq f_e$, τότε δεν χρειάζεται καμία αλλαγή. Αλλιώς θέτουμε $f'_e = c'_e$ και εκτελούμε τον ακόλουθο αλγόριθμο.
- Έστω $e = (x, y)$. Η εφικτή ροή υπολογίζεται με δύο BFS: Το πρώτο διασχίζοντας το υπογράφημα από το x στο s και το άλλο από το y στο t. Σκοπός είναι ξεκινώντας με τη μείωση κατά $f_e - c'_e$ από το x (προχωρώντας προς το s) και μετά όμοια από το y (προχωρώντας προς το t), να δημιουργηθεί μία εφικτή ροή.

Άσκηση 4β

- Κατά το BFS, έστω σε μία κορυφή v_i φτάνει νέα ροή f'_i και θέλουμε να μειώσουμε τη ροή στις προσπίπτουσες ακμές κατά k'. Μειώνεις τη ροή προς ένα γείτονα μέχρι είτε η ροή της ακμής να μηδενιστεί, είτε να έχουμε φτάσει στις k' μονάδες ροής.
- Αφού δημιουργήσουμε την εφικτή ροή f', εφαρμόζουμε τον αλγόριθμο Ford-Fulkerson. Εφόσον θα χρειαστεί προφανώς το πολύ k επαναλήψεις, η πολυπλοκότητα είναι $O(k * (V + E))$ όπου k σταθερά.

Άσκηση 4γ

- Αντιστρέφουμε όλες τις ακμές και αναζητούμε τη μέγιστη ροή. Η ιδέα είναι πως θέλουμε να δούμε πόση ροή μπορούμε να στείλουμε πίσω χωρίς να χαλάσουμε τους περιορισμούς.
- Βάζουμε νέες χωρητικότητες στο γράφημα για να λάβουμε υπόψη τους περιορισμούς: $c'_e = f_e - \ell_e$. Υπολογίζουμε τη μέγιστη ροή στο νέο γράφημα που συμβολίζουμε με h.
- Άρα η τελική ροή που είναι $g = f - h$ είναι προφανώς ελάχιστη και ισχύει επίσης
  $g_e = f_e - h_e \geq f_e - c'_e \geq f_e - f_e + \ell_e = \ell_e$ και
  $g_e = f_e - h_e \leq f_e \leq c_e$.

Άσκηση 5 3-Διαμέριση

Γενικό στιγμιότυπο του Partition: $A = w_1, ..., w_n$. Να βρεθεί διαμέριση A_1, A_2 του A ώστε $w(A_1) = w(A_2)$.
Το μετασχηματίζω σε ειδικό στιγμιότυπο του 3-Partition.
$A' = \{w_1, ..., w_n, W = \frac{w_1 + ... + w_n}{2}\}$.
Υποχρεωτικά, το W θα μπει σε ένα σύνολο και άρα τα άλλα δύο σύνολα είναι η διαμέριση του αρχικού προβλήματος του Partition. Άρα NP-hard. Προφανώς μπορούμε να επαληθεύσουμε εύκολα αν μία διαμέριση είναι ισοβαρής για ένα στιγμιότυπο του 3-Partition. Άρα ανήκει στο NP. Οπότε είναι NP-complete.

Θεώρημα:
Το LONGEST PATH είναι NP-COMPLETE

Απόδειξη:

- Πιστοποιητικό: Το μονοπάτι. Άρα LONGEST PATH $\in$ NP
- HAMILTON PATH $\leq$ LONGEST PATH

Άσκηση 5 Μακρύ Μονοπάτι-Αναγωγή

- Προσθέτουμε σε κάθε κορυφή γείτονες βαθμού 1.
- Αν έχουμε μονοπάτι Hamilton στο G, τότε θα υπάρχει μονοπάτι μήκους $|V| + 2$ στο G'
- Θέλουμε αυτό να είναι τουλάχιστον το $\frac{1}{4}$ του συνολικού αριθμού των κορυφών του G'
- Θα προσθέσουμε αρκετές καινούριες κορυφές ώστε να έχουμε $|V'| = 4|V| + 8$

Άσκηση 5 Πυκνό Γράφημα (Dense Subgraph)

Θεώρημα:
Το DENSE SUBGRAPH (DS) είναι NP-COMPLETE

Απόδειξη:

- Πιστοποιητικό: Το S. Άρα DS $\in$ NP
- DS γενίκευση του Clique ($B = \frac{k(k-1)}{2}$)

Απόδειξη

- Ανήκει στο NP: γιατί αν μας δοθεί η ανάθεση, επαληθεύουμε σε γραμμικό χρόνο ότι είναι έγκυρη.
- Είναι NP-hard: Υπάρχει πολυωνυμική αναγωγή από το 3-SAT.
- Έστω $\phi = \bigwedge_{j=1}^{m}(\ell_{j1} \vee \ell_{j2} \vee \ell_{j3})$ μία 3-CNF πρόταση για την οποία θέλουμε να ελέγξουμε αν υπάρχει ανάθεση αληθοτιμών που την ικανοποιεί (δηλαδή κάθε όρος της περιλαμβάνει τουλάχιστον ένα αληθές literal).

- Κατασκευάζουμε σε πολυωνυμικό χρόνο τη λογική πρόταση $\phi' = \bigwedge_{j=1}^{m}(\ell_{j1} \vee \ell_{j2} \vee \ell_{j3} \vee z)$ όπου z καινούρια μεταβλητή που δεν εμφανίζεται στην ϕ.
- Αν υπάρχει ανάθεση που να ικανοποιεί τον περιορισμό της ϕ' τότε η ϕ είναι ικανοποιήσιμη και το αντίστροφο: Για το αντίστροφο αν η ϕ είναι ικανοποιήσιμη, έχει τουλάχιστον ένα literal αληθές σε κάθε όρο άρα η ίδια ανάθεση, αν θέσουμε επιπλέον $z = false$ ικανοποιεί τον περιορισμό μας για την ϕ'. Για το ευθύ, αν έχουμε μία ανάθεση για το ϕ' που ικανοποιεί τον περιορισμό τότε προφανώς αν $z = false$ τότε η ίδια ανάθεση ικανοποιεί και την ϕ, διαφορετικά την ικανοποιεί η αντίθετή της. (δηλ. όποια μεταβλητή έχει true κάντη false και το αντίθετο).

Απόδειξη

- Ανήκει στο NP: Ελέγχουμε τα υποσύνολα που πρέπει να ικανοποιούν το ζητούμενο σε πολυωνυμικό χρόνο.
- Είναι NP-hard: Υπάρχει πολυωνυμική αναγωγή από το Max Independent Set (MIS).
- Για κάθε κόμβο του γραφήματος G, δημιουργούμε ένα σύνολο στο S με όλες τις προσπίπτουσες ακμές του κόμβου αυτού στο γράφημα. Η σταθερά k του Set Packing είναι ίση με την αντίστοιχη του MIS. Το U είναι το σύνολο των ακμών E.

- Έστω ότι έχουμε ανεξάρτητο σύνολο κορυφών με πληθάριθμο τουλάχιστον k. Κανένα ζεύγος κορυφών αυτού του συνόλου δεν θα έχει κοινή ακμή. Άρα και κανένα ζεύγος των αντίστοιχων υποσυνόλων του Set Packing δεν θα έχει κοινό στοιχείο. Δηλαδή, όλα τα υποσύνολα που θα δημιουργήσουμε θα είναι ξένα μεταξύ τους και τουλάχιστον k στο πλήθος.
- Αντίστροφα, έστω ότι έχουμε τουλάχιστον k ξένα υποσύνολα του U. Εφόσον το καθένα αντιστοιχεί σε κορυφή του αρχικού γραφήματος, θα έχουμε προφανώς στο G ίδιου πλήθους σύνολο κορυφών χωρίς κοινές ακμές, δηλαδή ένα ανεξάρτητο σύνολο με τον κατάλληλο πληθάριθμο.