Αλγόριθμοι και Πολυπλοκότητα

Ροή Δικτύου

Δημήτρης Μιχαήλ

Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο

Μοντελοποίηση Δικτύων Μεταφοράς

Τα γραφήματα χρησιμοποιούνται συχνά για την μοντελοποίηση των δικτύων μεταφοράς (transportation networks), δηλαδή δικτύων στα οποία οι ακμές μεταφέρουν κάποιο είδος κίνησης και οι κόμβοι ως ”μεταγωγείς” για τη μεταφορά της κίνησης στις διάφορες ακμές.

Παραδείγματα.

1. οδικό δίκτυο εθνικών οδών
   - ακμές είναι εθνικοί οδοί
   - κόμβοι είναι έξοδοι εθνικών οδών
2. δίκτυο υπολογιστών
   - ακμές είναι συνδέσεις που μεταφέρουν πακέτα
   - κόμβοι είναι μεταγωγείς (routers)
3. δίκτυο μεταφοράς υγρών
   - ακμές είναι σωληνώσεις
   - κόμβοι είναι συνδέσεις μεταξύ των σωληνώσεων

Χαρακτηριστικά Μοντέλων Δικτύων

Χωρητικότητα. Οι ακμές έχουν χωρητικότητα (capacity) η οποία δείχνει πόση ποσότητα μπορεί να μεταφερθεί.

Κόμβοι Προέλευσης. Κόμβοι προέλευσης (source) στο γράφημα οι οποίοι παράγουν κίνηση.

Κόμβοι Απόληξης. Κόμβοι απόληξης ή προορισμού (target) στο γράφημα οι οποίοι μπορούν να ”απορροφήσουν” την κίνηση που καταφθάνει.

Κίνηση. Η κίνηση που διαδίδεται μέσω των ακμών. Αναφερόμαστε στην κίνηση αυτή ως **ροή (flow)**.

Δίκτυο Ροής
Flow Network

Το **δίκτυο ροής** (flow network) είναι ένα κατευθυνόμενο γράφημα $G = (V, E)$ με τα ακόλουθα χαρακτηριστικά:

- Με κάθε ακμή $e \in E$ είναι συσχετισμένη μια χωρητικότητα $c_e \geq 0$.
- Υπάρχει ένας μόνο κόμβος προέλευσης $s \in V$.
- Υπάρχει ένας μόνο κόμβος απόληξης $t \in V$.

Οι υπόλοιποι κόμβοι, εκτός των s και t, θα ονομάζονται *εσωτερικοί* (internal) κόμβοι.

Το **δίκτυο ροής** (flow network) είναι ένα κατευθυνόμενο γράφημα $G = (V, E)$ με τα ακόλουθα χαρακτηριστικά:

- Με κάθε ακμή $e \in E$ είναι συσχετισμένη μια χωρητικότητα $c_e \geq 0$.
- Υπάρχει ένας μόνο κόμβος προέλευσης $s \in V$.
- Υπάρχει ένας μόνο κόμβος απόληξης $t \in V$.

Διάφορες Παραδοχές.

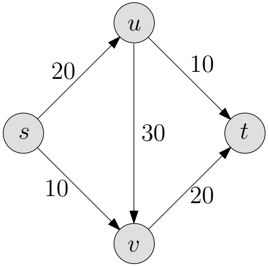
1. δεν υπάρχει ακμή που να εισέρχεται στην προέλευση $s \in V$ ή να εξέρχεται από την απόληξη $t \in V$,
2. υπάρχει τουλάχιστον μια ακμή προσκείμενη σε κάθε κόμβο
3. όλες οι χωρητικότητες είναι ακέραιοι αριθμοί

Δίκτυο Ροής

Flow Network

Το **δίκτυο ροής** (flow network) είναι ένα κατευθυνόμενο γράφημα $G = (V, E)$ με τα ακόλουθα χαρακτηριστικά:

- Με κάθε ακμή $e \in E$ είναι συσχετισμένη μια χωρητικότητα $c_e \geq 0$.
- Υπάρχει ένας μόνο κόμβος προέλευσης $s \in V$.
- Υπάρχει ένας μόνο κόμβος απόληξης $t \in V$.



Παράδειγμα δικτύου ροής με

- προέλευση s,
- απόληξη t και
- χωρητικότητες δίπλα στις ακμές.

Ροή

Λέμε ότι η **ροή** $s - t$ ($s - t$ flow) είναι μια συνάρτηση f που αντιστοιχίζει κάθε ακμή e σε έναν μη αρνητικό πραγματικό αριθμό $f : E \mapsto \mathcal{R}^{+}$.

Διαισθητικά η τιμή $f(e)$ αντιπροσωπεύει την ποσότητα ροής που μεταφέρεται από την ακμή e.

Η ροή πρέπει να ικανοποιεί τις παρακάτω δύο ιδιότητες.

1. (*Συνθήκες χωρητικότητας*) Για κάθε ακμή $e \in E$, έχουμε $0 \leq f(e) \leq c_e$.
2. (*Συνθήκες διατήρησης*) Για κάθε κόμβο v εκτός των s and t, έχουμε

$$\sum_{e \text{ προς το } v} f(e) = \sum_{e \text{ από το } v} f(e)$$

Τιμή Ροής

Η τιμή μιας ροής f, που συμβολίζουμε ως $v(f)$, ορίζεται ως η ποσότητα της ροής που παράγεται στην προέλευση s.

$$v(f) = \sum_{e \text{ από το } s} f(e)$$

Μερικοί Ορισμοί

Για να κάνουμε πιο συμπαγή τον τρόπο γραφής μας, ορίζουμε ότι

$$f^{out}(v) = \sum_{e \text{ από το } v} f(e),$$

και

$$f^{in}(v) = \sum_{e \text{ προς το } v} f(e).$$

Το ίδιο και για σύνολα κόμβων $S \subseteq V$ ορίζουμε πως

$$f^{out}(S) = \sum_{e \text{ από το } S} f(e),$$

και

$$f^{in}(S) = \sum_{e \text{ προς το } S} f(e).$$

Πρόβλημα Μέγιστης Ροής
Max Flow Problem

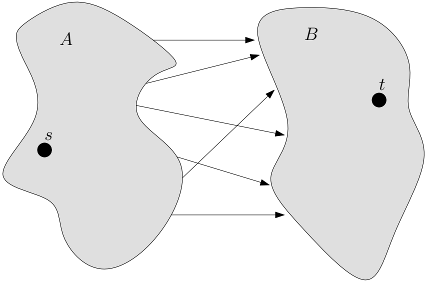
Πρόβλημα. Με δεδομένο ένα δίκτυο ροής, βρείτε μια ροή με τη μέγιστη δυνατή τιμή.

Πρόβλημα Μέγιστης Ροής

Max Flow Problem

Πρόβλημα. Με δεδομένο ένα δίκτυο ροής, βρείτε μια ροή με τη μέγιστη δυνατή τιμή.

Από τι εξαρτάται η μέγιστη ροή; Υποθέστε πως διαιρούμε τους κόμβους του γραφήματος σε δύο σύνολα, A και B, έτσι ώστε $s \in A$ και $t \in B$. Διαισθητικά, οποιαδήποτε ροή πρέπει να περνάει από το A στο B και άρα να χρησιμοποιεί κάποια από τις χωρητικότητες ακμής από το A στο B.



Πρόβλημα Μέγιστης Ροής

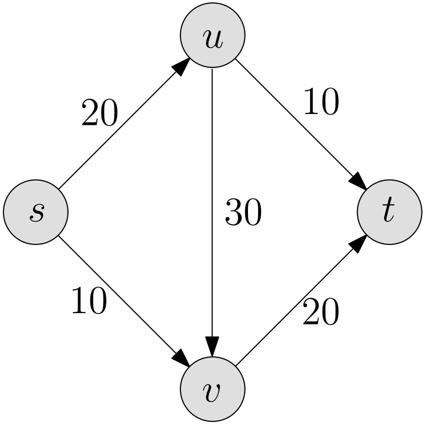
Κανόνας. Για να υπολογίσουμε μια μέγιστη ροή θα προσπαθήσουμε να βρούμε διαδρομές από το s στο t μέσω των οποίων να στείλουμε ροή. Θα φροντίσουμε να μην παραβιάσουμε τις χωρητικότητες των ακμών.

Άπληστος Αλγόριθμος. Μήπως λειτουργεί μια απλή άπληστη τακτική;

Πρόβλημα Μέγιστης Ροής

Κανόνας. Για να υπολογίσουμε μια μέγιστη ροή θα προσπαθήσουμε να βρούμε διαδρομές από το s στο t μέσω των οποίων να στείλουμε ροή. Θα φροντίσουμε να μην παραβιάσουμε τις χωρητικότητες των ακμών.

Άπληστος Αλγόριθμος. Μήπως λειτουργεί μια απλή άπληστη τακτική;

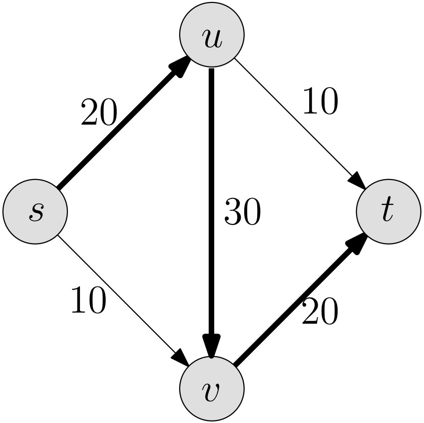


Πρόβλημα Μέγιστης Ροής

Κανόνας. Για να υπολογίσουμε μια μέγιστη ροή θα προσπαθήσουμε να βρούμε διαδρομές από το s στο t μέσω των οποίων να στείλουμε ροή. Θα φροντίσουμε να μην παραβιάσουμε τις χωρητικότητες των ακμών.

Άπληστος Αλγόριθμος. Μήπως λειτουργεί μια απλή άπληστη τακτική;

- Στο δίκτυο που βλέπουμε μπορούμε να στείλουμε 20 μονάδες ροής μέσω της διαδρομής $s \to u \to v \to t$.

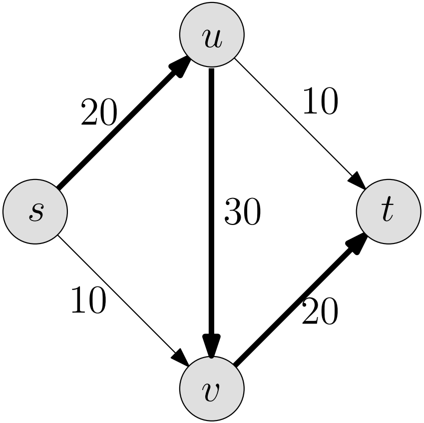


Πρόβλημα Μέγιστης Ροής

Κανόνας. Για να υπολογίσουμε μια μέγιστη ροή θα προσπαθήσουμε να βρούμε διαδρομές από το s στο t μέσω των οποίων να στείλουμε ροή. Θα φροντίσουμε να μην παραβιάσουμε τις χωρητικότητες των ακμών.

Άπληστος Αλγόριθμος. Μήπως λειτουργεί μια απλή άπληστη τακτική;

- Στο δίκτυο που βλέπουμε μπορούμε να στείλουμε 20 μονάδες ροής μέσω της διαδρομής $s \to u \to v \to t$.
- Η μέγιστη ροή όμως είναι 30.

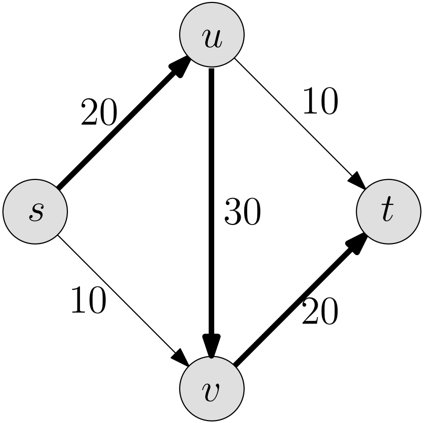


Πρόβλημα Μέγιστης Ροής

Κανόνας. Για να υπολογίσουμε μια μέγιστη ροή θα προσπαθήσουμε να βρούμε διαδρομές από το s στο t μέσω των οποίων να στείλουμε ροή. Θα φροντίσουμε να μην παραβιάσουμε τις χωρητικότητες των ακμών.

Άπληστος Αλγόριθμος. Μήπως λειτουργεί μια απλή άπληστη τακτική;

- Στο δίκτυο που βλέπουμε μπορούμε να στείλουμε 20 μονάδες ροής μέσω της διαδρομής $s \to u \to v \to t$.
- Η μέγιστη ροή όμως είναι 30.
- Δεν μπορούμε να στείλουμε τίποτα άλλο αν δεν πάρουμε πίσω κάποια ροή από την ακμή (u, v).

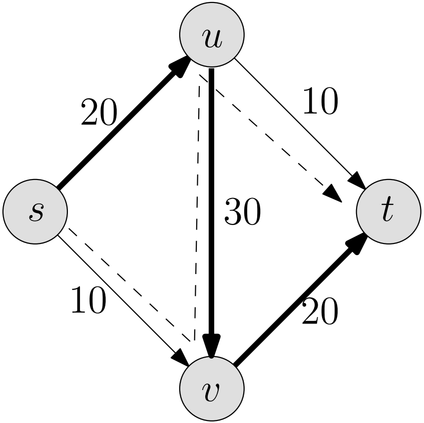


Πρόβλημα Μέγιστης Ροής

Κανόνας. Για να υπολογίσουμε μια μέγιστη ροή θα προσπαθήσουμε να βρούμε διαδρομές από το s στο t μέσω των οποίων να στείλουμε ροή. Θα φροντίσουμε να μην παραβιάσουμε τις χωρητικότητες των ακμών.

Άπληστος Αλγόριθμος. Μήπως λειτουργεί μια απλή άπληστη τακτική;

- Στο δίκτυο που βλέπουμε μπορούμε να στείλουμε 20 μονάδες ροής μέσω της διαδρομής $s \to u \to v \to t$.
- Η μέγιστη ροή όμως είναι 30.
- Δεν μπορούμε να στείλουμε τίποτα άλλο αν δεν πάρουμε πίσω κάποια ροή από την ακμή (u, v).
- Ουσιαστικά πρέπει να στείλουμε 10 μονάδες ροής μέσω της ακμής (s, v), να αναιρέσουμε 10 μονάδες ροής από την ακμή (u, v) και να στείλουμε 10 μονάδες μέσω της ακμής (u, t).



Πρόβλημα Μέγιστης Ροής

Τεχνική. Θα χρησιμοποιήσουμε τον εξής γενικότερο τρόπο για την προώθηση ροής.

- θα προωθούμε προς τα εμπρός (ευθύδρομα) σε ακμές που έχουν περίσσευμα χωρητικότητας
- θα προωθούμε προς τα πίσω (ανάδρομα) σε ακμές που μεταφέρουν ήδη ροή, έτσι ώστε να την εκτρέψουμε σε διαφορετική κατεύθυνση.

Για να κάνουμε αυτή την τεχνική πιο σαφής θα ορίσουμε το **υπολειπόμενο γράφημα** (residual graph).

Υπολειπόμενο Γράφημα

Residual Graph

Δεδομένου ενός γραφήματος G και μιας ροής f στο G, ορίζουμε το υπολειπόμενο γράφημα G_f του G ως προς την ροή f με τον εξής τρόπο.

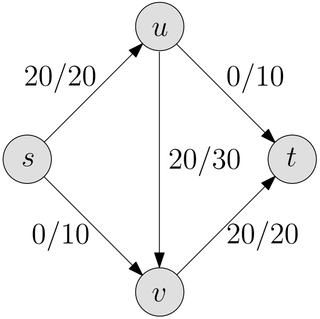
- Το G_f έχει το ίδιο σύνολο κόμβων με το G.
- Για κάθε ακμή $e = (u, v)$ του G όπου $f(e) < c_e$ υπάρχουν $c_e - f(e)$ μονάδες χωρητικότητας ελεύθερες. Συμπεριλαμβάνουμε την ακμή $e = (u, v)$ στο G_f με χωρητικότητα $c_e - f(e)$. Αυτές οι ακμές λέγονται ευθύγραμμες (forward edges).
- Για κάθε ακμή $e = (u, v)$ του G όπου $f(e) > 0$, μπορούμε να ”αναιρέσουμε” $f(e)$ μονάδες. Συμπεριλαμβάνουμε την ακμή $e' = (v, u)$ στο G_f με χωρητικότητα $f(e)$. Αυτές οι ακμές λέγονται ανάδρομες (backward edges).

Κάθε ακμή $e \in G$ μπορεί να δημιουργήσει το πολύ δύο ακμές στο G_f. Άρα το G_f έχει το πολύ διπλάσιες ακμές από το G.

Υπολειπόμενο Γράφημα

Residual Graph

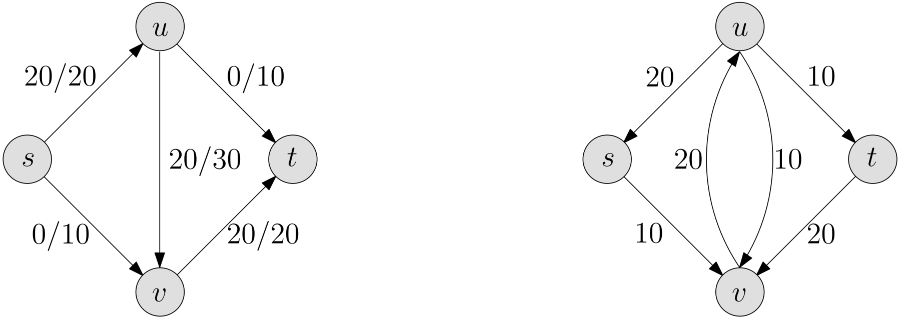
Όταν αναφερόμαστε στη χωρητικότητα ακμής στο υπολειπόμενο γράφημα, χρησιμοποιούμε τον όρο **υπολειπόμενη χωρητικότητα** (residual capacity).



Το υπολειπόμενο γράφημα μας δίνει ένα συστηματικό τρόπο να αλλάζουμε την ροή.

Υπολειπόμενο Γράφημα
Residual Graph

Όταν αναφερόμαστε στη χωρητικότητα ακμής στο υπολειπόμενο γράφημα, χρησιμοποιούμε τον όρο **υπολειπόμενη χωρητικότητα** (residual capacity).



Το υπολειπόμενο γράφημα μας δίνει ένα συστηματικό τρόπο να αλλάζουμε την ροή.

Διαδρομές Επαύξησης

Augmenting Paths

Οποιαδήποτε διαδρομή στο υπολειπόμενο γράφημα G_f από τον κόμβο προέλευσης s στον κόμβο απόληξης t ονομάζεται **διαδρομή επαύξησης**.

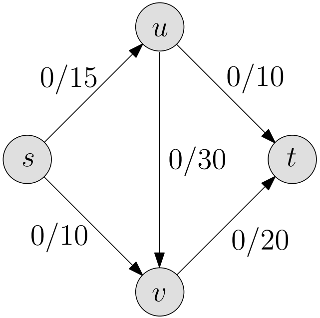
Ορισμός bottleneck. Έστω P μια απλή διαδρομή $s - t$ στο γράφημα G_f. Ορίζουμε ότι η $b(P, f)$ είναι η ελάχιστη υπολειπόμενη χωρητικότητα οποιασδήποτε ακμής της P, σε σχέση με τη ροή f.

Επαύξηση (augmentation). Έστω μια διαδρομή επαύξησης P στο G_f. Ορίζουμε την παρακάτω λειτουργία, η οποία παράγει μια νέα ροή f' στο G.

Για κάθε ακμή $e = (u, v) \in P$ της διαδρομής $P \in G_f$

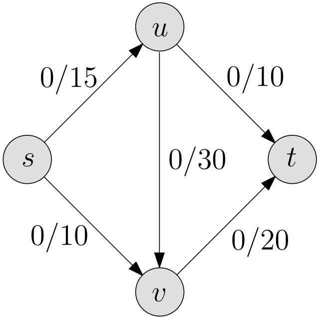
- Αν $e = (u, v)$ είναι ευθύδρομη ακμή αύξησε το $f(e)$ στο G κατά $b(P, f)$.
- Αν $e = (u, v)$ είναι ανάδρομη ακμή μείωσε το $f(v, u)$ στο G κατά $b(P, f)$.

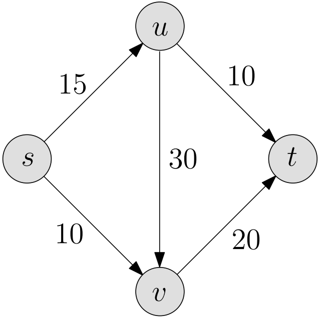
Παράδειγμα



Έστω το δίκτυο *G* στα αριστερά και *f* η μηδενική ροή.

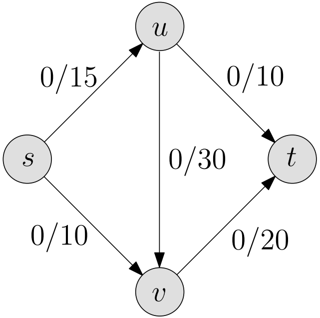
Παράδειγμα

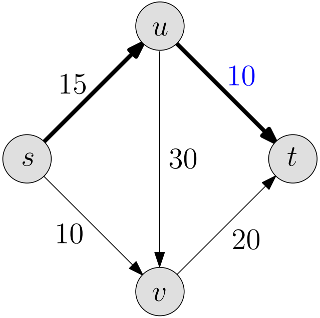




Το υπολειπόμενο γράφημα G_f στα δεξιά.

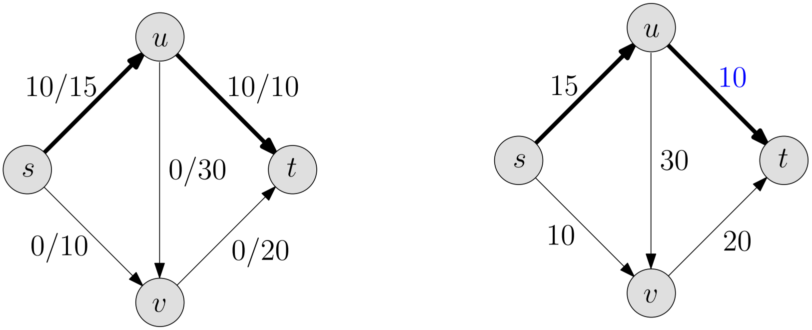
Παράδειγμα





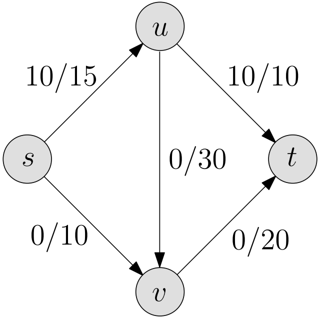
Έστω $P = \langle s, u, t \rangle$ μια διαδρομή επαύξησης με $b(P, f) = 10$

Παράδειγμα



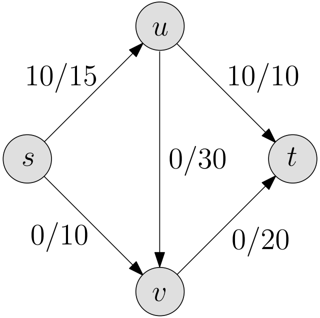
Στέλνουμε στο γράφημα G ροή 10 από το s στο t μέσω της διαδρομής P.

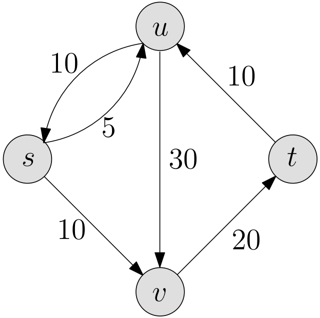
Παράδειγμα



Έχουμε μια καινούρια ροή f'

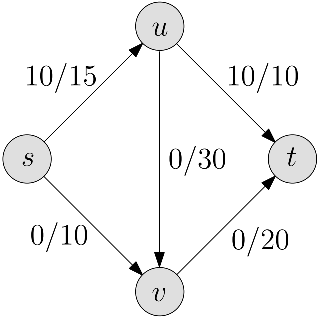
Παράδειγμα

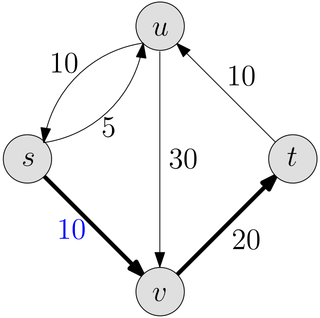




Ας δούμε τώρα το καινούριο υπολειπόμενο γράφημα G'_f.

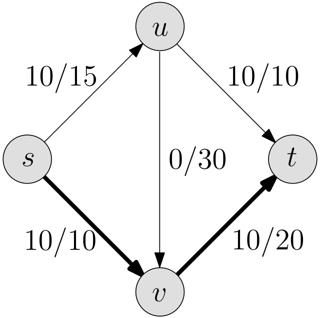
Παράδειγμα

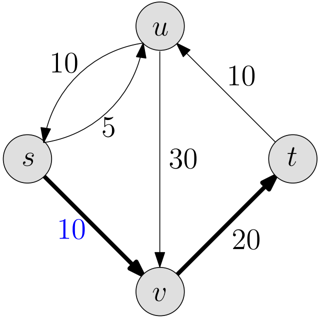




Έστω $P' = \langle s, v, t \rangle$ μια διαδρομή επαύξησης πάλι με $b(P', f') = 10$

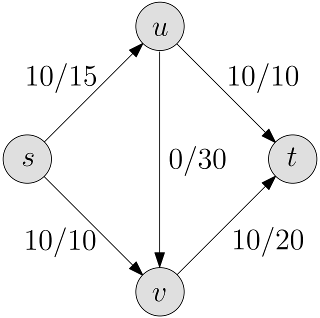
Παράδειγμα





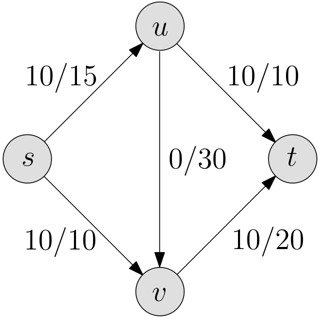
Στέλνουμε στο γράφημα G ροή 10 από το s στο t μέσω της διαδρομής P'.

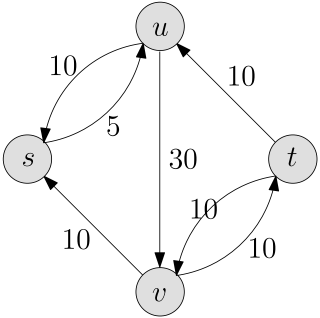
Παράδειγμα



Έχουμε μια καινούρια ροή f'

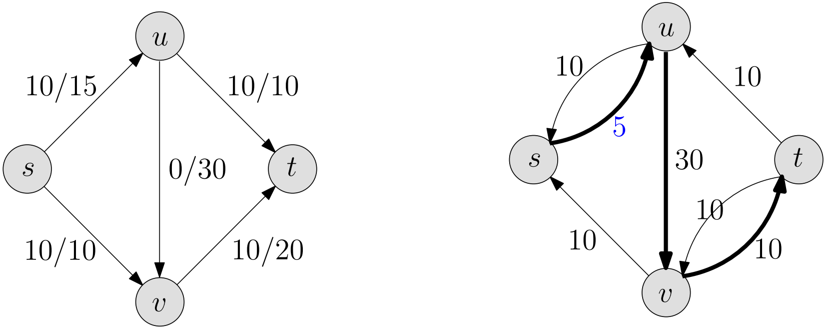
Παράδειγμα





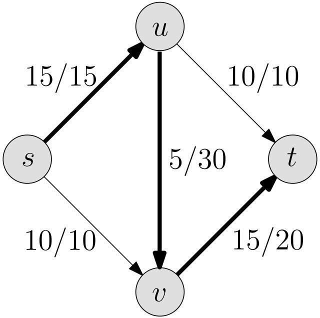
Ας δούμε τώρα το καινούριο υπολειπόμενο γράφημα G_f''.

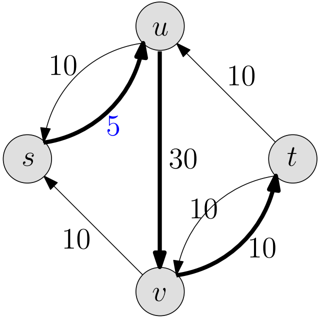
Παράδειγμα



Έστω $P'' = \langle s, u, v, t \rangle$ η μόνη διαδρομή επαύξησης με $b(P'', f') = 5$

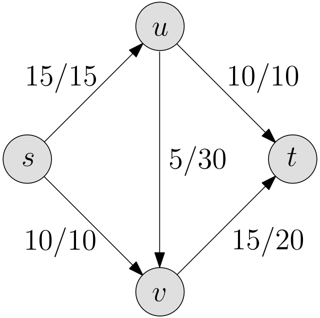
Παράδειγμα





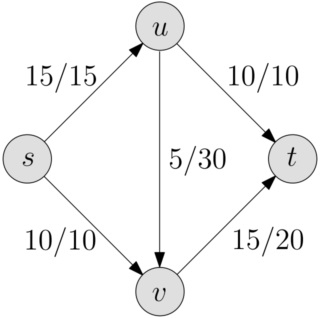
Στέλνουμε στο γράφημα G ροή 5 από το s στο t μέσω της διαδρομής P'.

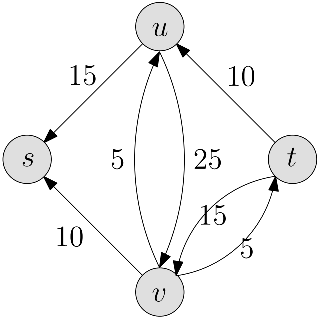
Παράδειγμα



Έχουμε μια καινούρια ροή f''

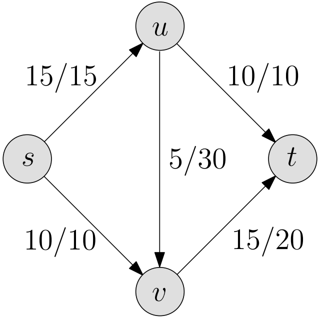
Παράδειγμα

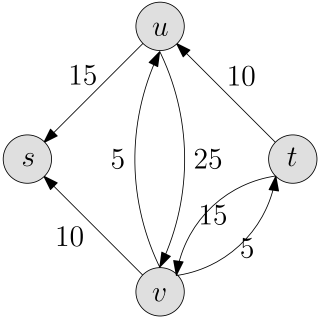




Ας δούμε τώρα το καινούριο υπολειπόμενο γράφημα G_f'''.

Παράδειγμα





Δεν υπάρχει $s - t$ διαδρομή στο G_f'''.

Επαύξηση

Λήμμα

Έστω G ένα δίκτυο ροής, f μια ροή και P μια απλή διαδρομή στο υπολειπόμενο γράφημα G_f. Έστω επίσης f' η συνάρτηση στο γράφημα G που προκύπτει από την επαύξηση κατά $b(f, P)$ μονάδες. Η f' είναι ροή.

Επαύξηση

Λήμμα

Έστω G ένα δίκτυο ροής, f μια ροή και P μια απλή διαδρομή στο υπολειπόμενο γράφημα G_f. Έστω επίσης f' η συνάρτηση στο γράφημα G που προκύπτει από την επαύξηση κατά $b(f, P)$ μονάδες. Η f' είναι ροή.

Απόδειξη

Πρέπει να επαληθεύσουμε τις συνθήκες χωρητικότητας και διατήρησης.

$\square$

Επαύξηση

Λήμμα

Έστω G ένα δίκτυο ροής, f μια ροή και P μια απλή διαδρομή στο υπολειπόμενο γράφημα G_f. Έστω επίσης f' η συνάρτηση στο γράφημα G που προκύπτει από την επαύξηση κατά $b(f, P)$ μονάδες. Η f' είναι ροή.

Απόδειξη

Η f' διαφέρει από την f μόνο κατά μήκος της P, άρα αρκεί να ελέγξουμε μόνο αυτές τις ακμές. Αν $e = (u, v)$ είναι ευθύδρομη ακμή η υπολειπόμενη χωρητικότητα της είναι $c_e - f(e)$ και έχουμε

$$0 \leq f(e) \leq f'(e) = f(e) + b(P, f) \leq f(e) + (c_e - f(e)) = c_e.$$

Αν $(u, v) \in G_f$ είναι ανάδρομη ακμή που προκύπτει από την $e = (v, u) \in G$, τότε η υπολειπόμενη χωρητικότητα της είναι $f(e)$ και άρα

$$c_e \geq f(e) \geq f'(e) = f(e) - b(P, f) \geq f(e) - f(e) = 0.$$

□

Επαύξηση

Λήμμα

Έστω G ένα δίκτυο ροής, f μια ροή και P μια απλή διαδρομή στο υπολειπόμενο γράφημα G_f. Έστω επίσης f' η συνάρτηση στο γράφημα G που προκύπτει από την επαύξηση κατά $b(f, P)$ μονάδες. Η f' είναι ροή.

Απόδειξη

Η f' διαφέρει από την f μόνο κατά μήκος της P, άρα αρκεί να ελέγξουμε μόνο αυτές τις ακμές. Αν $e = (u, v)$ είναι ευθύδρομη ακμή η υπολειπόμενη χωρητικότητα της είναι $c_e - f(e)$ και έχουμε

$$0 \leq f(e) \leq f'(e) = f(e) + b(P, f) \leq f(e) + (c_e - f(e)) = c_e.$$

Αν $(u, v) \in G_f$ είναι ανάδρομη ακμή που προκύπτει από την $e = (v, u) \in G$, τότε η υπολειπόμενη χωρητικότητα της είναι $f(e)$ και άρα

$$c_e \geq f(e) \geq f'(e) = f(e) - b(P, f) \geq f(e) - f(e) = 0.$$

Πρέπει να ελέγξουμε για διατήρηση τους εσωτερικούς κόμβους της P. Έστω v ένας τέτοιος κόμβος. Μπορούμε να επαληθεύσουμε πως η μεταβολή στην ποσότητα ροής που εισέρχεται στον κόμβο v είναι ίδια με την μεταβολή στην ποσότητα ροής που εξέρχεται από τον κόμβο v. Επειδή η f ικανοποιούσε τη συνθήκη διατήρησης στον v το ίδιο θα ισχύει και για την f'. $\square$

Αλγόριθμος Ford-Fulkerson

Έυρεση Μέγιστης Ροής

Input: Δίκτυο $G(V, E)$ με χωρητικότητες στις ακμές $\{c_e : e \in E\}$.
Output: Μέγιστη ροή
Αρχικά $f(e) = 0$ για όλα τα $e \in E$
while *υπάρχει* $s - t$ *διαδρομή* P *στο γράφημα* G_f **do**
  $f \leftarrow f +$ επαύξηση κατά $b(P, f)$
  ενημέρωσε το G_f με βάση την καινούρια ροή
end

Τερματισμός. Για να αποδείξουμε πως ο αλγόριθμος τερματίζει, θα δείξουμε πως κάνει συνέχεια βήματα προόδου.

Ακέραιες Τιμές Ροής. Είναι εύκολο μέσω επαγωγής να δείξουμε πως οι τιμές ροής $\{f(e)\}$ και οι υπολειπόμενες χωρητικότητες στο G_f είναι πάντα ακέραιοι αριθμοί.

Λήμμα

Έστω ότι f είναι μια ροή στο G, και έστω ότι P είναι μια απλή διαδρομή $s-t$ στο G_f. Τότε $v(f') = v(f) + b(P,f)$ και αφού $b(P,f) > 0$ έχουμε $v(f') > v(f)$.

Λήμμα

Έστω ότι f είναι μια ροή στο G, και έστω ότι P είναι μια απλή διαδρομή $s - t$ στο G_f. Τότε $v(f') = v(f) + b(P, f)$ και αφού $b(P, f) > 0$ έχουμε $v(f') > v(f)$.

Απόδειξη

Η πρώτη ακμή e στην διαδρομή P εξέρχεται από το s στο υπολειπόμενο γράφημα G_f, και αφού η διαδρομή είναι απλή, δεν έχουμε άλλη επίσκεψη στο s.

$\square$

Λήμμα

Έστω ότι f είναι μια ροή στο G, και έστω ότι P είναι μια απλή διαδρομή $s - t$ στο G_f. Τότε $v(f') = v(f) + b(P, f)$ και αφού $b(P, f) > 0$ έχουμε $v(f') > v(f)$.

Απόδειξη

Η πρώτη ακμή e στην διαδρομή P εξέρχεται από το s στο υπολειπόμενο γράφημα G_f, και αφού η διαδρομή είναι απλή, δεν έχουμε άλλη επίσκεψη στο s.

Το γράφημα G δεν έχει ακμές που εισέρχονται στον κόμβο s, και άρα μια εξερχόμενη ακμή από το s στο γράφημα G_f είναι υποχρεωτικά ευθύδρομη ακμή.

$\square$

Λήμμα

Έστω ότι f είναι μια ροή στο G, και έστω ότι P είναι μια απλή διαδρομή $s-t$ στο G_f. Τότε $v(f') = v(f) + b(P,f)$ και αφού $b(P,f) > 0$ έχουμε $v(f') > v(f)$.

Απόδειξη

Η πρώτη ακμή e στην διαδρομή P εξέρχεται από το s στο υπολειπόμενο γράφημα G_f, και αφού η διαδρομή είναι απλή, δεν έχουμε άλλη επίσκεψη στο s.

Το γράφημα G δεν έχει ακμές που εισέρχονται στον κόμβο s, και άρα μια εξερχόμενη ακμή από το s στο γράφημα G_f είναι υποχρεωτικά ευθύδρομη ακμή.

Η ακμή e είναι η μόνη ακμή του s που αλλάζει η ροή της και η ροή της αυξάνεται κατά $b(P,f)$. Άρα $f' = f + b(P,f)$. $\square$

Άνω Όριο. Ο αλγόριθμος υπολογίζει διαρκώς ακέραιες ροές και η ροή που υπολογίζει αυξάνει από βήμα σε βήμα. Για να δείξουμε πως τερματίζει αρκεί να βρούμε και ένα άνω όριο της μέγιστης ροής.

Εξερχόμενες ακμές από την προέλευση. Αν όλες οι ακμές που εξέρχονται από τον s μπορούσαν να κορεστούν πλήρως με ροή, η τιμή της ροής θα ήταν

$$\sum_{e \text{ από το } s} c_e.$$

Έστω ότι C συμβολίζει αυτό το άθροισμα. Έχουμε πως

$$v(f) \leq C$$

για κάθε $s-t$ ροή f.

Έχουμε πλέον όλα τα εργαλεία για να αποδείξουμε πως ο αλγόριθμος τερματίζει.

Θεώρημα

Υποθέστε ότι όλες οι χωρητικότητες σε ένα δίκτυο ροής G είναι ακέραιες. Ο αλγόριθμος Ford-Fulkerson τερματίζει μετά από το πολύ C επαυξήσεις.

Απόδειξη

Καμία ροή στο G δεν μπορεί να είναι μεγαλύτερη από C.

Κάθε ροή είναι ακέραιος και κάθε επαύξηση αυξάνει την ροή τουλάχιστον κατά μια μονάδα.

Η ροή στην αρχή είναι μηδέν και άρα θα εκτελεστούν το πολύ C επαυξήσεις. $\square$

Χρόνος Εκτέλεσης Ford-Fulkerson

Επειδή έχουμε υποθέσει πως όλοι οι κόμβοι έχουν τουλάχιστον μια προσκείμενη ακμή, ισχύει $m \geq n/2$ και άρα για απλούστευση θα χρησιμοποιήσουμε το γεγονός πως $\mathcal{O}(m+n) = \mathcal{O}(m)$.

Θεώρημα

Αν όλες οι χωρητικότητες είναι ακέραιοι, ο αλγόριθμος Ford-Fulkerson μπορεί να υλοποιηθεί ώστε να εκτελείται σε χρόνο $\mathcal{O}(mC)$.

Απόδειξη

Η επανάληψη εκτελείται το πολύ C φορές.

Σε κάθε επανάληψη κάνουμε τα εξής.

- φτιάχνουμε το G_f σε χρόνο $\mathcal{O}(m)$
- ψάχνουμε διαδρομή $s - t$, έστω P, στο γράφημα G_f σε χρόνο $\mathcal{O}(m+n) = \mathcal{O}(m)$ με DFS ή BFS.
- Υπολογίζουμε το f' σε χρόνο $\mathcal{O}(n)$ αφού η P έχει το πολύ $n-1$ ακμές.

$\square$

Μέγιστη Ροή

Έχουμε δείξει πως ο αλγόριθμος τερματίζει αλλά δεν έχουμε αποδείξει ακόμη πως υπολογίζει την μέγιστη δυνατή ροή.

Όρια του Δικτύου. Το βασικό ερώτημα είναι πως η δομή του δικτύου θέτει άνω όρια ως προς τη μέγιστη τιμή μιας ροής $s-t$.

Εξερχόμενη Χωρητικότητα του s. Είδαμε ήδη ένα άνω όριο: η τιμή $v(f)$ για οποιαδήποτε ροή $s-t$ είναι το πολύ ίση με $C = \sum_{e \text{ από το } s} c_e$. Θέλουμε όμως ένα πιο καλό όριο, αυτό το όριο είναι πολλές φορές πολύ αδύναμο.

Αποκοπή

Cut

Αποκοπή $s-t$. Μια αποκοπή $s-t$ είναι μια διαμέριση (A, B) του συνόλου των κορυφών V έτσι ώστε $s \in A$ και $t \in B$.

Χωρητικότητα. Η χωρητικότητα μιας αποκοπής (A, B), την οποία συμβολίζουμε ως $c(A, B)$, είναι απλώς το άθροισμα των χωρητικοτήτων για όλες τις ακμές που εξέρχονται από το A:

$$c(A, B) = \sum_{e \text{ από το } A} c_e.$$

Οι αποκοπές παρέχουν πολύ φυσικά όρια ως προς τις τιμές των ροών.

Αποκοπή

Cut

Λήμμα

Έστω ότι η f είναι κάποια $s - t$ ροή και (A, B) είναι κάποια αποκοπή $s - t$. Τότε $v(f) = f^{out}(A) - f^{in}(A)$.

Αποκοπή
Cut

Λήμμα

Έστω ότι η f είναι κάποια $s-t$ ροή και (A, B) είναι κάποια αποκοπή $s-t$. Τότε $v(f) = f^{out}(A) - f^{in}(A)$.

Απόδειξη

Από τον ορισμό $v(f) = f^{out}(s)$. Με βάση τις παραδοχές μας $f^{in}(s) = 0$ και άρα $v(f) = f^{out}(s) - f^{in}(s)$. Επίσης όλες οι κορυφές $v \in A$ εκτός του s είναι εσωτερικές και άρα $f^{out}(v) - f^{in}(v) = 0$. Έτσι

$$v(f) = \sum_{v \in A} (f^{out}(v) - f^{in}(v)).$$

$\square$

Λήμμα

Έστω ότι η f είναι κάποια $s-t$ ροή και (A,B) είναι κάποια αποκοπή $s-t$. Τότε $v(f) = f^{out}(A) - f^{in}(A)$.

Απόδειξη

Από τον ορισμό $v(f) = f^{out}(s)$. Με βάση τις παραδοχές μας $f^{in}(s) = 0$ και άρα $v(f) = f^{out}(s) - f^{in}(s)$. Επίσης όλες οι κορυφές $v \in A$ εκτός του s είναι εσωτερικές και άρα $f^{out}(v) - f^{in}(v) = 0$. Έτσι

$$v(f) = \sum_{v \in A} (f^{out}(v) - f^{in}(v)).$$

Θα αναδιατυπώσουμε το άθροισμα. Έστω μια ακμή $e = (u,v)$. Εαν $u \in A$ και $v \in A$ τότε η ακμή εμφανίζεται 2 φορές στο άθροισμα με αντίθετα πρόσημα, και άρα αλληλοαναιρούνται. Αν $u \notin A$ και $v \notin A$ δεν εμφανίζεται καθόλου. Αλλιώς εμφανίζεται μια φορά. Με βάση αυτά μπορούμε να γράψουμε:

$$v(f) = \sum_{v \in A} (f^{out}(v) - f^{in}(v)) = \sum_{e \text{ από το } A} f(e) - \sum_{e \text{ προς το } A} f(e) = f^{out}(A) - f^{in}(A).$$

$\square$

Αποκοπή

Cut

Λήμμα

Έστω ότι η f είναι κάποια $s-t$ ροή και (A, B) είναι κάποια αποκοπή $s-t$. Τότε $v(f) \leq c(A, B)$.

Αποκοπή

Cut

Λήμμα

Έστω ότι η f είναι κάποια $s-t$ ροή και (A, B) είναι κάποια αποκοπή $s-t$. Τότε $v(f) \leq c(A, B)$.

Απόδειξη

$$\begin{aligned} v(f) &= f^{out}(A) - f^{in}(A) \\ &\leq f^{out}(A) \\ &= \sum_{e \text{ από το } A} f(e) \\ &\leq \sum_{e \text{ από το } A} c_e \\ &= c(A, B) \end{aligned}$$

□

Λήμμα

Έστω ότι η f είναι κάποια $s-t$ ροή και (A, B) είναι κάποια αποκοπή $s-t$. Τότε $v(f) \leq c(A, B)$.

Απόδειξη

$$\begin{aligned} v(f) &= f^{out}(A) - f^{in}(A) \\ &\leq f^{out}(A) \\ &= \sum_{e \text{ από το } A} f(e) \\ &\leq \sum_{e \text{ από το } A} c_e \\ &= c(A, B) \end{aligned}$$

□

Με άλλα λόγια στην τιμή κάθε ροής τίθεται ένα άνω όριο με βάση τη χωρητικότητα κάθε αποκοπής.

Αλγόριθμος Ford-Fulkerson και Μέγιστη Ροή

Έστω ότι το $\bar{f}$ συμβολίζει τη ροή που επιστρέφεται από τον αλγόριθμο Ford-Fulkerson.

Στόχος. Θέλουμε να δείξουμε πως ο αλγόριθμος Ford-Fulkerson βρίσκει την μέγιστη ροή.

Μέθοδος. Θα βρούμε μια $s-t$ αποκοπή (A^*, B^*) για την οποία $v(\bar{f}) = c(A^*, B^*)$.

Η ύπαρξη αυτής της αποκοπής αμέσως αποδεικνύει πως η $\bar{f}$ είναι μέγιστη λόγω του προηγούμενου λήμματος.

Max-Flow Min-Cut Theorem

Θεώρημα

Αν η f είναι μια ροή $s - t$ τέτοια ώστε να μην υπάρχει διαδρομή $s - t$ στο υπολειπόμενο γράφημα G_f, τότε υπάρχει μια $s - t$ αποκοπή (A^*, B^*) στο G για την οποία $v(f) = c(A^*, B^*)$. Κατά συνέπεια, η f έχει τη μέγιστη τιμή από όλες τις ροές του G και η (A^*, B^*) έχει την ελάχιστη χωρητικότητα από όλες τις αποκοπές $s - t$ του G.

Max-Flow Min-Cut Theorem

Απόδειξη

Έστω A^* το σύνολο των κόμβων $v \in G$ που υπάρχει διαδρομή $s - v$ στο G_f και έστω $B^* = V \setminus A^*$. Το (A^*, B^*) είναι μια $s - t$ αποκοπή αφού δεν υπάρχει διαδρομή $s - t$ στο G_f και άρα $s \in A^*$ ενώ $t \in B^*$.

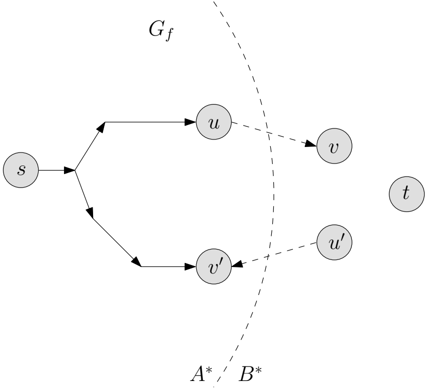
$\square$

Max-Flow Min-Cut Theorem

Απόδειξη

Έστω A^* το σύνολο των κόμβων $v \in G$ που υπάρχει διαδρομή $s - v$ στο G_f και έστω $B^* = V \setminus A^*$. Το (A^*, B^*) είναι μια $s - t$ αποκοπή αφού δεν υπάρχει διαδρομή $s - t$ στο G_f και άρα $s \in A^*$ ενώ $t \in B^*$.

- Έστω $e = (u, v) \in G$ έτσι ώστε $u \in A^*$ και $v \in B^*$. Τότε η ακμή είναι κορεσμένη, $f(e) = c_e$, αλλιώς η e θα ήταν ευθύδρομη στο G_f.
- Έστω $e' = (u', v') \in G$ έτσι ώστε $u' \in B^*$ και $v' \in A^*$. Τότε η ακμή είναι αχρησιμοποίητη, $f(e) = 0$, αλλιώς η ανάδρομη της $e'' = (v', u')$ θα υπήρχε στο G_f.

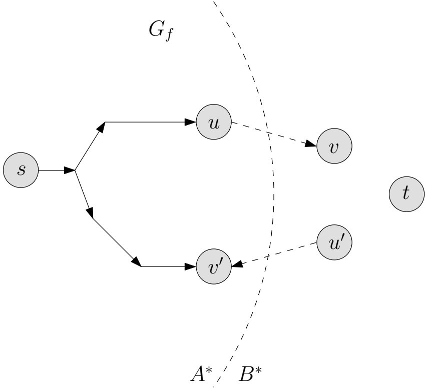


□

Απόδειξη

Έστω A^* το σύνολο των κόμβων $v \in G$ που υπάρχει διαδρομή $s - v$ στο G_f και έστω $B^* = V \setminus A^*$. Το (A^*, B^*) είναι μια $s - t$ αποκοπή αφού δεν υπάρχει διαδρομή $s - t$ στο G_f και άρα $s \in A^*$ ενώ $t \in B^*$.

- Έστω $e = (u, v) \in G$ έτσι ώστε $u \in A^*$ και $v \in B^*$. Τότε η ακμή είναι κορεσμένη, $f(e) = c_e$, αλλιώς η e θα ήταν ευθύδρομη στο G_f.
- Έστω $e' = (u', v') \in G$ έτσι ώστε $u' \in B^*$ και $v' \in A^*$. Τότε η ακμή είναι αχρησιμοποίητη, $f(e) = 0$, αλλιώς η ανάδρομη της $e'' = (v', u')$ θα υπήρχε στο G_f.



Άρα

$$
\begin{aligned}
v(f) &= f^{out}(A^*) - f^{in}(A^*) \\
&= \sum_{e \text{ από το } A^*} f(e) - \sum_{e \text{ προς το } A^*} f(e) \\
&= \sum_{e \text{ από το } A^*} c_e - 0 \\
&= c(A^*, B^*).
\end{aligned}
$$

□

Max-Flow Min-Cut Theorem

Μπορούμε να διατυπώσουμε πιο γενικά το τελευταίο αποτέλεσμα.

Θεώρημα

Σε όλα τα δίκτυα ροής η μέγιστη τιμή μιας ροής $s - t$ είναι ίση με την ελάχιστη χωρητικότητα μιας αποκοπής $s - t$.

Χρόνος Εκτέλεσης

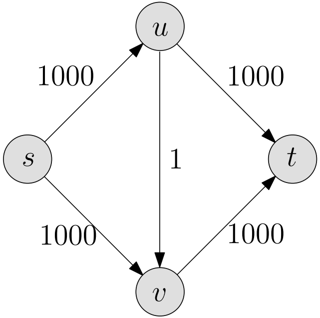
Καταλήγουμε πως ο αλγόριθμος Ford-Fulkerson τρέχει σε χρόνο $\mathcal{O}(mC)$ όπου C είναι η μέγιστη ροή του δικτύου.

Ψευδο-πολυωνυμικός Χρόνος. Έχουμε αναφερθεί και πάλι σε τέτοιου είδους χρόνους και τους αποκαλέσαμε ψευδοπολυωνυμικούς. Ο λόγος είναι πως τέτοιοι αλγόριθμοι είναι πολυωνυμικοί ως προς το μέγεθος των αριθμών εισόδου αλλά όχι ως προς το πλήθος των ψηφίων που απαιτούνται για την αναπαράσταση τους.

Χρόνος Εκτέλεσης

Καταλήγουμε πως ο αλγόριθμος Ford-Fulkerson τρέχει σε χρόνο $\mathcal{O}(mC)$ όπου C είναι η μέγιστη ροή του δικτύου.

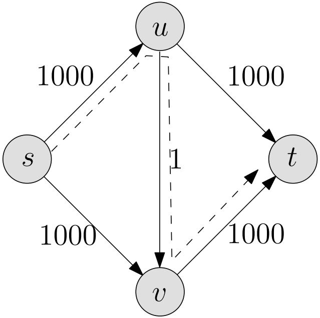
Ψευδο-πολυωνυμικός Χρόνος. Έχουμε αναφερθεί και πάλι σε τέτοιου είδους χρόνους και τους αποκαλέσαμε ψευδοπολυωνυμικούς. Ο λόγος είναι πως τέτοιοι αλγόριθμοι είναι πολυωνυμικοί ως προς το μέγεθος των αριθμών εισόδου αλλά όχι ως προς το πλήθος των ψηφίων που απαιτούνται για την αναπαράσταση τους.



Χρόνος Εκτέλεσης

Καταλήγουμε πως ο αλγόριθμος Ford-Fulkerson τρέχει σε χρόνο $\mathcal{O}(mC)$ όπου C είναι η μέγιστη ροή του δικτύου.

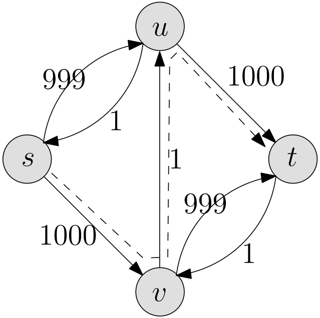
Ψευδο-πολυωνυμικός Χρόνος. Έχουμε αναφερθεί και πάλι σε τέτοιου είδους χρόνους και τους αποκαλέσαμε ψευδοπολυωνυμικούς. Ο λόγος είναι πως τέτοιοι αλγόριθμοι είναι πολυωνυμικοί ως προς το μέγεθος των αριθμών εισόδου αλλά όχι ως προς το πλήθος των ψηφίων που απαιτούνται για την αναπαράσταση τους.



Χρόνος Εκτέλεσης

Καταλήγουμε πως ο αλγόριθμος Ford-Fulkerson τρέχει σε χρόνο $\mathcal{O}(mC)$ όπου C είναι η μέγιστη ροή του δικτύου.

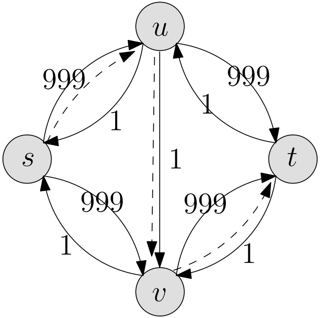
Ψευδο-πολυωνυμικός Χρόνος. Έχουμε αναφερθεί και πάλι σε τέτοιου είδους χρόνους και τους αποκαλέσαμε ψευδοπολυωνυμικούς. Ο λόγος είναι πως τέτοιοι αλγόριθμοι είναι πολυωνυμικοί ως προς το μέγεθος των αριθμών εισόδου αλλά όχι ως προς το πλήθος των ψηφίων που απαιτούνται για την αναπαράσταση τους.



Χρόνος Εκτέλεσης

Καταλήγουμε πως ο αλγόριθμος Ford-Fulkerson τρέχει σε χρόνο $\mathcal{O}(mC)$ όπου C είναι η μέγιστη ροή του δικτύου.

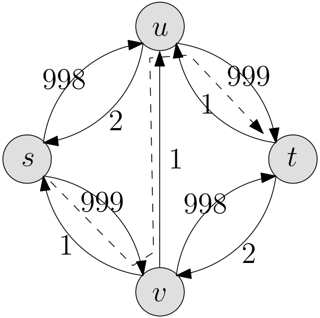
Ψευδο-πολυωνυμικός Χρόνος. Έχουμε αναφερθεί και πάλι σε τέτοιου είδους χρόνους και τους αποκαλέσαμε ψευδοπολυωνυμικούς. Ο λόγος είναι πως τέτοιοι αλγόριθμοι είναι πολυωνυμικοί ως προς το μέγεθος των αριθμών εισόδου αλλά όχι ως προς το πλήθος των ψηφίων που απαιτούνται για την αναπαράσταση τους.



Χρόνος Εκτέλεσης

Καταλήγουμε πως ο αλγόριθμος Ford-Fulkerson τρέχει σε χρόνο $\mathcal{O}(mC)$ όπου C είναι η μέγιστη ροή του δικτύου.

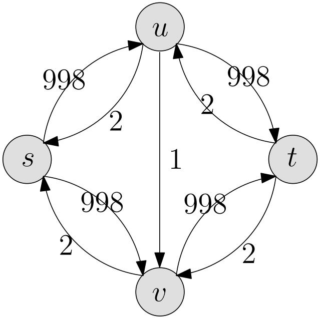
Ψευδο-πολυωνυμικός Χρόνος. Έχουμε αναφερθεί και πάλι σε τέτοιου είδους χρόνους και τους αποκαλέσαμε ψευδοπολυωνυμικούς. Ο λόγος είναι πως τέτοιοι αλγόριθμοι είναι πολυωνυμικοί ως προς το μέγεθος των αριθμών εισόδου αλλά όχι ως προς το πλήθος των ψηφίων που απαιτούνται για την αναπαράσταση τους.



Χρόνος Εκτέλεσης

Καταλήγουμε πως ο αλγόριθμος Ford-Fulkerson τρέχει σε χρόνο $\mathcal{O}(mC)$ όπου C είναι η μέγιστη ροή του δικτύου.

Ψευδο-πολυωνυμικός Χρόνος. Έχουμε αναφερθεί και πάλι σε τέτοιου είδους χρόνους και τους αποκαλέσαμε ψευδοπολυωνυμικούς. Ο λόγος είναι πως τέτοιοι αλγόριθμοι είναι πολυωνυμικοί ως προς το μέγεθος των αριθμών εισόδου αλλά όχι ως προς το πλήθος των ψηφίων που απαιτούνται για την αναπαράσταση τους.



Μια παραλλαγή των Edmonds και Karp

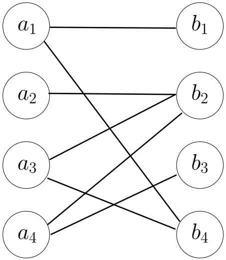
BFS. Εαν υλοποιήσουμε την αναζήτηση για μονοπάτια προσαύξησης στο G_f με BFS τότε ο αλγόριθμος Ford-Fulkerson τρέχει στην χειρότερη περίπτωση σε χρόνο $\mathcal{O}(m^2 n)$ αντι για $\mathcal{O}(mC)$.

Μικρότερος Αριθμός Ακμών. Με άλλα λόγια διαλέγουμε πάντα το μονοπάτι προσαύξησης $s-t$ που έχει τον μικρότερο αριθμό ακμών.

Στο παράδειγμα της προηγούμενη διαφάνειας ο αλγόριθμος τερματίζει με 2 προσαυξήσεις αντί για 1000.

Διμερές Ταίριασμα

Ένα γράφημα $G(V, E)$ είναι *διμερές* (bipartite) αν το σύνολο των κόμβων του V μπορεί να διαμεριστεί σε σύνολα A και B με τέτοιο τρόπο ώστε η κάθε ακμή να έχει το ένα άκρο στο A και το άλλο στο B.



Διμερές Ταίριασμα

Ένα *ταίριασμα* (matching) σε ένα γράφημα $G(V, E)$ είναι ένα σύνολο ακμών $M \subseteq E$ με την ιδιότητα ότι κάθε κόμβος εμφανίζεται το πολύ σε μια ακμή του M. Το M είναι ένα *τέλειο ταίριασμα* αν κάθε κόμβος εμφανίζεται σε μια μόνο ακμή του M.

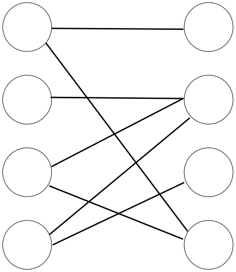
Με ταιριάσματα μοντελοποιούμε προβλήματα ανάθεσης, για παράδειγμα ανάθεση υπολογιστικών εργασιών σε υπολογιστές όπου μια ακμή υποδηλώνει την ικανότητα του εκάστοτε υπολογιστή να διεκπεραιώσει την λειτουργία.

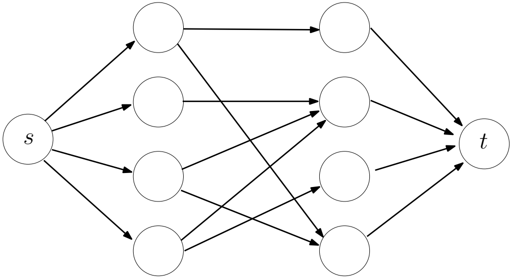
Διμερές Ταίριασμα

Το *πρόβλημα Διμερούς Ταιριάσματος* (Bipartite matching problem) είναι το ακόλουθο: Δεδομένου ενός διμερούς γραφήματος G, βρείτε ένα ταίριασμα μέγιστου μεγέθους.

Αναγωγή σε Max-Flow

Ξεκινώντας με το γράφημα G ενός προβλήματος Διμερούς Ταιριάσματος, κατασκευάζουμε ένα δίκτυο ροής G'.





Προσθέτουμε κόμβους s και t και όλες οι ακμές έχουν χωρητικότητα 1.

Η τιμή της μέγιστης ροής στο καινούριο δίκτυο $s - t$ είναι ίση με το μέγεθος τους μέγιστου ταιριάσματος στο G.

Αναγωγή σε Max-Flow

Λήμμα

Έστω στο G ένα μέγιστο ταίριασμα που αποτελείται από k ακμές $(a_1, b_1), \ldots, (a_k, b_k)$. Τότε υπάρχει ροή $s - t$ στο G' με τιμή k.

Απόδειξη

Έστω η ροή που στέλνει μια μονάδα κατά μήκος κάθε διαδρομής $\langle s, a_i, b_i, t \rangle$. $\square$

Αναγωγή σε Max-Flow

Λήμμα

Έστω στο G' μια μέγιστη ροή f' με τιμή k. Υπάρχει ταίριασμα στο G με k ακμές.

Απόδειξη

Από το θεώρημα ακέραιων τιμών για τις μέγιστες ροές γνωρίζουμε ότι υπάρχει μια ακέραια ροή f με τιμή k, και αφού $c_e = 1$ για κάθε $e \in G'$ έχουμε πως $f(e) \in \{0, 1\}$.

Έστω M' το σύνολο των ακμών της μορφής (a, b) για τις οποίες η τιμή της ροής είναι 1.

Έχουμε πως $|M'| = k$ και επίσης κάθε κόμβος είτε του A είτε του B έχει το πολύ μια μονάδα ροής που περνάει από αυτόν λόγω αρχής διατήρησης. $\square$

Χρόνος Εκτέλεσης

Έστω πως $n = |A| = |B|$ και έστω m το πλήθος των ακμών στο διμερές γράφημα G.

$s - t$ αποκοπή. Έχουμε $C = \sum_{e \text{ από το } s} c_e = |A| = n$.

Χρόνος Ford-Fulkerson. Το χρονικό όριο $\mathcal{O}(mC)$ είναι ουσιαστικά $\mathcal{O}(mn)$ σε αυτή την περίπτωση.