

Προηγμένα Θέματα Αλγορίθμων (ΣΗΜΜΥ ΕΜΠ) Αλγόριθμοι Δικτύων και Πολυπλοκότητα (ΕΜΠ - ΑΛΜΑ)

Διδάσκοντες: Δημήτρης Φωτάκης - Άρης Παγουρτζής

Προσεγγιστικοί Αλγόριθμοι

Επιμέλεια διαφανειών: Άρης Παγουρτζής

Άνοιξη 2022

- ▶ Αφορούν κυρίως σε προβλήματα βελτιστοποίησης: σε κάθε στιγμιότυπο αντιστοιχούν εφικτές (αποδεκτές / έγκυρες) λύσεις (feasible solutions) που η κάθε μια έχει μια τιμή μέσω μιας αντικειμενικής συνάρτησης (objective function) (συνήθως εκφράζει: κόστος, μήκος, βάρος, κέρδος, κ.λπ.). Ζητείται βέλτιστη λύση, δηλαδή εφικτή λύση με βέλτιστη τιμή.

- ▶ Αφορούν κυρίως σε **προβλήματα βελτιστοποίησης**: σε κάθε στιγμιότυπο αντιστοιχούν **εφικτές (αποδεκτές / έγκυρες) λύσεις** (feasible solutions) που η κάθε μια έχει μια **τιμή** μέσω μιας **αντικειμενικής συνάρτησης (objective function)** (συνήθως εκφράζει: κόστος, μήκος, βάρος, κέρδος, κ.λπ.). Ζητείται **βέλτιστη λύση**, δηλαδή εφικτή λύση με βέλτιστη τιμή.
- ▶ Προβλήματα ελαχιστοποίησης (minimization): **Shortest Paths, Dominating Set, Vertex Cover, TSP.**

- ▶ Αφορούν κυρίως σε **προβλήματα βελτιστοποίησης**: σε κάθε στιγμιότυπο αντιστοιχούν **εφικτές (αποδεκτές / έγκυρες) λύσεις** (feasible solutions) που η κάθε μια έχει μια **τιμή** μέσω μιας **αντικειμενικής συνάρτησης (objective function)** (συνήθως εκφράζει: κόστος, μήκος, βάρος, κέρδος, κ.λπ.). Ζητείται **βέλτιστη λύση**, δηλαδή εφικτή λύση με βέλτιστη τιμή.
- ▶ Προβλήματα ελαχιστοποίησης (minimization): **Shortest Paths, Dominating Set, Vertex Cover, TSP.**
- ▶ Προβλήματα μεγιστοποίησης (maximization): **Maximum Matching, Independent Set, Clique.**

- ▶ Αφορούν κυρίως σε **προβλήματα βελτιστοποίησης**: σε κάθε στιγμιότυπο αντιστοιχούν **εφικτές (αποδεκτές / έγκυρες) λύσεις** (feasible solutions) που η κάθε μια έχει μια **τιμή** μέσω μιας **αντικειμενικής συνάρτησης (objective function)** (συνήθως εκφράζει: κόστος, μήκος, βάρος, κέρδος, κ.λπ.). Ζητείται **βέλτιστη λύση**, δηλαδή εφικτή λύση με βέλτιστη τιμή.
- ▶ Προβλήματα ελαχιστοποίησης (minimization): **Shortest Paths, Dominating Set, Vertex Cover, TSP.**
- ▶ Προβλήματα μεγιστοποίησης (maximization): **Maximum Matching, Independent Set, Clique.**
- ▶ Κλάση **NPO (NP-optimization)**: κάθε εφικτή λύση έχει πολυωνυμικό μήκος ως προς είσοδο, και επαληθεύεται σε πολυωνυμικό χρόνο.
Το αντίστοιχο πρόβλημα απόφασης ανήκει στην κλάση **NP**.

- ▶ Π : πρόβλημα βελτιστοποίησης
- ▶ I : στιγμιότυπο (είσοδος) του προβλήματος
- ▶ $SOL_A(\Pi, I)$: η τιμή της λύσης που επιστρέφει ο αλγόριθμος A για το στιγμιότυπο I του προβλήματος Π .
- ▶ $OPT(\Pi, I)$: η τιμή της βέλτιστης λύσης για το στιγμιότυπο I του προβλήματος Π .

Σημείωση: Συχνά Π , A και I παραλείπονται.

Προσεγγισιμότητα: προβλήματα ελαχιστοποίησης

- ▶ **Λόγος προσέγγισης αλγορίθμου:** Το ελάχιστο ρ που ικανοποιεί

$$\frac{SOLA(I)}{OPT(I)} \leq \rho$$

για κάθε στιγμιότυπο I ενός προβλήματος ελαχιστοποίησης λέγεται **λόγος** (ή **παράγοντας**) **προσέγγισης** (**approximation ratio**) του αλγορίθμου A , και ο A λέγεται **ρ -προσεγγιστικός** αλγόριθμος για το πρόβλημα.

- ▶ **Λόγος προσέγγισης προβλήματος:** Αν για πρόβλημα Π υπάρχει ρ -προσεγγιστικός αλγόριθμος, λέμε ότι το Π προσεγγίζεται με λόγο (ή παράγοντα) ρ . Μας ενδιαφέρει το ελάχιστο ρ μεταξύ όλων των δυνατών προσεγγιστικών αλγορίθμων για το Π .

Σημείωση: συνήθως, ο όρος **προσεγγιστικός αλγόριθμος** αναφέρεται σε αλγόριθμο **πολυωνυμικού χρόνου** ως προς το μέγεθος εισόδου $|I|$.

- ▶ Προβλήματα μεγιστοποίησης: ο αλγόριθμος A λέγεται ρ -προσεγγιστικός για το Π αν για κάθε I :

$$\frac{SOL_A(I)}{OPT(I)} \geq \rho$$

(δηλαδή $\rho < 1$ για προβλήματα μεγιστοποίησης)

- ▶ **Λόγος προσέγγισης αλγορίθμου** για πρόβλημα μεγιστοποίησης: το μέγιστο ρ που ικανοποιεί την παραπάνω σχέση (για κάθε I).
- ▶ **Λόγος προσέγγισης προβλήματος**: το μέγιστο ρ μεταξύ όλων των αλγορίθμων που το επιλύουν.

Προσεγγισιμότητα: εναλλακτικοί ορισμοί

- ▶ Εναλλακτικός ορισμός του λόγου προσέγγισης, **κοινός για προβλήματα ελαχιστοποίησης και μεγιστοποίησης**:

Ενας αλγόριθμος A για ένα πρόβλημα βελτιστοποίησης Π λέγεται **ρ -προσεγγιστικός**, αν για κάθε έγκυρο στιγμιότυπο I :

$$\max\left\{\frac{SOL_A(I)}{OPT(I)}, \frac{OPT(I)}{SOL_A(I)}\right\} \leq \rho$$

Στο πλαίσιο αυτό $\rho > 1$ πάντοτε. Ακολουθείται σε κάποια βιβλιογραφία, εμείς θα χρησιμοποιήσουμε τους προηγούμενους ορισμούς.

- ▶ Παλαιότερος ορισμός εξετάζει το **σχετικό σφάλμα**. Ένας αλγόριθμος A έχει **σχετικό σφάλμα προσέγγισης** ϵ αν $\forall I$:

$$\frac{|SOL_A(I) - OPT(I)|}{OPT(I)} \leq \epsilon$$

- ▶ Ο λόγος προσέγγισης στην γενική περίπτωση είναι **συνάρτηση του μεγέθους (μήκους) της εισόδου**:

$$\forall I : \quad \frac{SOL_A(I)}{OPT(I)} \leq \rho(|I|) \quad (\geq \text{για max})$$

- ▶ Ασυμπτωτικός λόγος προσέγγισης: η ανισότητα ισχύει $\forall I, |I| \geq n_0$.
- ▶ Πιθανοτικοί προσεγγιστικοί αλγόριθμοι: ο λόγος προσέγγισης επιτυγχάνεται με **μεγάλη πιθανότητα** (τυπικοί ορισμοί αργότερα).

Κλάσεις προσεγγισιμότητας (i)

- ▶ **APX**: κλάση προβλημάτων της **NPO** που επιλύονται με σταθερό λόγο προσέγγισης σε πολυωνυμικό χρόνο.

Παράδειγμα: **Vertex Cover**

Κλάσεις προσεγγισιμότητας (i)

- ▶ **APX**: κλάση προβλημάτων της **NPO** που επιλύονται με σταθερό λόγο προσέγγισης σε πολυωνυμικό χρόνο.

Παράδειγμα: **Vertex Cover**

- ▶ **PTAS**: κλάση προβλημάτων της **NPO** που επιλύονται με **οποιοδήποτε** σταθερό λόγο προσέγγισης $1 + \varepsilon$ (μεγιστ/σης: $1 - \varepsilon$) σε πολυωνυμικό χρόνο ως προς $|I|$. Υποκλάση της **APX**.

Παράδειγμα: **Bin Packing**

Κλάσεις προσεγγισιμότητας (i)

- ▶ **APX**: κλάση προβλημάτων της **NPO** που επιλύονται με σταθερό λόγο προσέγγισης σε πολυωνυμικό χρόνο.

Παράδειγμα: **Vertex Cover**

- ▶ **PTAS**: κλάση προβλημάτων της **NPO** που επιλύονται με **οποιοδήποτε** σταθερό λόγο προσέγγισης $1 + \varepsilon$ (μεγιστ/σης: $1 - \varepsilon$) σε πολυωνυμικό χρόνο ως προς $|I|$. Υποκλάση της **APX**.

Παράδειγμα: **Bin Packing**

- ▶ **FPTAS**: κλάση προβλημάτων της **NPO** που επιλύονται με **οποιοδήποτε** σταθερό λόγο προσέγγισης $1 + \varepsilon$ (μεγιστ/σης: $1 - \varepsilon$) σε πολυωνυμικό χρόνο ως προς $|I|$ και $\frac{1}{\varepsilon}$. Υποκλάση της **PTAS**.

Παράδειγμα: **Knapsack**

Κλάσεις προσεγγισιμότητας (ii)

- ▶ **log-APX**: κλάση προβλημάτων της **NPO** που επιλύονται με λογαριθμικό (ως προς $|I|$) λόγο προσέγγισης σε πολυωνυμικό χρόνο. Υπερκλάση της **APX**.

Παράδειγμα: **Set Cover**

Κλάσεις προσεγγισιμότητας (ii)

- ▶ **log-APX**: κλάση προβλημάτων της **NPO** που επιλύονται με λογαριθμικό (ως προς $|I|$) λόγο προσέγγισης σε πολυωνυμικό χρόνο. Υπερκλάση της **APX**.

Παράδειγμα: **Set Cover**

- ▶ **poly-APX**: κλάση προβλημάτων της **NPO** που επιλύονται με πολυωνυμικό (ως προς $|I|$) λόγο προσέγγισης σε πολυωνυμικό χρόνο. Υπερκλάση της **log-APX**.

Παράδειγμα: **Max Independent Set**

Κλάσεις προσεγγισιμότητας (ii)

- ▶ **log-APX**: κλάση προβλημάτων της **NPO** που επιλύονται με λογαριθμικό (ως προς $|I|$) λόγο προσέγγισης σε πολυωνυμικό χρόνο. Υπερκλάση της **APX**.

Παράδειγμα: **Set Cover**

- ▶ **poly-APX**: κλάση προβλημάτων της **NPO** που επιλύονται με πολυωνυμικό (ως προς $|I|$) λόγο προσέγγισης σε πολυωνυμικό χρόνο. Υπερκλάση της **log-APX**.

Παράδειγμα: **Max Independent Set**

- ▶ Κλάση **NPO** (**NP**-optimization): κάθε εφικτή λύση έχει πολυωνυμικό μήκος ως προς είσοδο, και επαληθεύεται σε πολυωνυμικό χρόνο. Το αντίστοιχο πρόβλημα απόφασης ανήκει στην κλάση **NP**.

Παράδειγμα: **TSP**

$$\mathbf{FP} \subseteq \mathbf{FPTAS} \subseteq \mathbf{PTAS} \subseteq \mathbf{APX} \subseteq$$

$$\mathbf{log-APX} \subseteq \mathbf{poly-APX} \subseteq \mathbf{NPO}$$

Το πρόβλημα **Vertex Cover (VC)**

Vertex Cover

Δίνεται: γράφος $G(V, E)$

Ζητείται: **Κάλυμμα ή κάλυψη κορυφών (vertex cover)** ελάχιστης πληθικότητας, δηλαδή ελάχιστο σύνολο κορυφών V' έτσι ώστε κάθε ακμή να έχει τουλάχιστον μία από τις κορυφές της στο V' .

Το πρόβλημα **Vertex Cover (VC)**

Vertex Cover

Δίνεται: γράφος $G(V, E)$

Ζητείται: **Κάλυμμα ή κάλυψη κορυφών (vertex cover)** ελάχιστης πληθικότητας, δηλαδή ελάχιστο σύνολο κορυφών V' έτσι ώστε κάθε ακμή να έχει τουλάχιστον μία από τις κορυφές της στο V' .

Weighted Vertex Cover (WVC): οι κορυφές έχουν και βάρος και το ζητούμενο είναι το σύνολο V' να είναι ελαχίστου βάρους.

Σημείωση: Συχνά ο όρος **Vertex Cover** χρησιμοποιείται για την weighted εκδοχή του προβλήματος. Τότε για το πρώτο πρόβλημα χρησιμοποιούνται οι όροι “Unweighted” ή “Cardinality”.

Vertex Cover

Δίνεται: γράφος $G(V, E)$

Ζητείται: **Κάλυμμα ή κάλυψη κορυφών (vertex cover)** ελάχιστης πληθικότητας, δηλαδή ελάχιστο σύνολο κορυφών V' έτσι ώστε κάθε ακμή να έχει τουλάχιστον μία από τις κορυφές της στο V' .

Vertex Cover

Δίνεται: γράφος $G(V, E)$

Ζητείται: **Κάλυμμα ή κάλυψη κορυφών (vertex cover)** ελάχιστης πληθικότητας, δηλαδή ελάχιστο σύνολο κορυφών V' έτσι ώστε κάθε ακμή να έχει τουλάχιστον μία από τις κορυφές της στο V' .

- **NP-hard**: αναγωγή από **Satisfiability**.

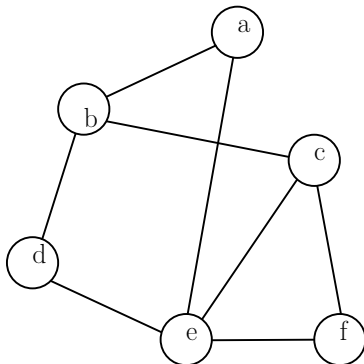
Vertex Cover

Δίνεται: γράφος $G(V, E)$

Ζητείται: **Κάλυμμα ή κάλυψη κορυφών (vertex cover)** ελάχιστης πληθικότητας, δηλαδή ελάχιστο σύνολο κορυφών V' έτσι ώστε κάθε ακμή να έχει τουλάχιστον μία από τις κορυφές της στο V' .

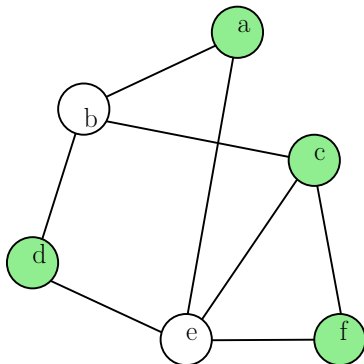
- ▶ **NP-hard**: αναγωγή από **Satisfiability**.
- ▶ Σχέση με **Independent Set**: ισοδύναμα ως προς επιλυσιμότητα (επακριβώς), σημαντική διαφορά ως προς προσεγγισιμότητα!

Παράδειγμα (**Vertex Cover**)



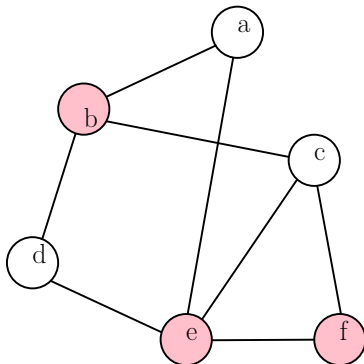
Σχήμα: Ένα στιγμιότυπο του προβλήματος **Vertex Cover**

Παράδειγμα (**Vertex Cover**)



Σχήμα: Μια μη βέλτιστη λύση

Παράδειγμα (**Vertex Cover**)



Σχήμα: Μια βέλτιστη λύση

Αλγόριθμος VC-Greedy

$C \leftarrow \emptyset$

Επανάλαβε για όσο $\exists e = \{u, v\} \in E : \{u, v\} \cap C = \emptyset$

- Βρές κορυφή v που καλύπτει μέγιστο πλήθος
ακάλυπτων ακμών
- $C \leftarrow C \cup \{v\}$

Ερώτηση: τι λόγο προσέγγισης επιτυγχάνει ο VC-Greedy ;

Αλγόριθμος VC-Greedy

$C \leftarrow \emptyset$

Επανάλαβε για όσο $\exists e = \{u, v\} \in E : \{u, v\} \cap C = \emptyset$

- Βρές κορυφή v που καλύπτει μέγιστο πλήθος
ακάλυπτων ακμών
- $C \leftarrow C \cup \{v\}$

Ερώτηση: τι λόγο προσέγγισης επιτυγχάνει ο VC-Greedy ;

Θα το απαντήσουμε σε λίγο ...

Αλγόριθμος VC-Match

- Βρες maximal matching $M_{\max}$ στον γράφο
- Επίστρεψε το σύνολο V' των κορυφών που είναι άκρα ακμών του $M_{\max}$

Αλγόριθμος VC-Match

- Βρες maximal matching $M_{\max}$ στον γράφο
- Επίστρεψε το σύνολο V' των κορυφών που είναι άκρα ακμών του $M_{\max}$

Θεώρημα

Ο αλγόριθμος VC-Match είναι 2-προσεγγιστικός.

Αλγόριθμος VC-Match

- Βρες maximal matching $M_{\max}$ στον γράφο
- Επίστρεψε το σύνολο V' των κορυφών που είναι άκρα ακμών του $M_{\max}$

Θεώρημα

Ο αλγόριθμος VC-Match είναι 2-προσεγγιστικός.

Απόδειξη.

1. Το V' είναι εφικτή λύση (γιατί;)

2-προσεγγιστικός αλγόριθμος για το **Vertex Cover**

Αλγόριθμος VC-Match

- Βρες maximal matching $M_{\max}$ στον γράφο
- Επίστρεψε το σύνολο V' των κορυφών που είναι άκρα ακμών του $M_{\max}$

Θεώρημα

Ο αλγόριθμος VC-Match είναι 2-προσεγγιστικός.

Απόδειξη.

1. Το V' είναι εφικτή λύση (γιατί;)
2. $|M_{\max}| \leq OPT$

2-προσεγγιστικός αλγόριθμος για το **Vertex Cover**

Αλγόριθμος VC-Match

- Βρες maximal matching $M_{\max}$ στον γράφο
- Επίστρεψε το σύνολο V' των κορυφών που είναι άκρα ακμών του $M_{\max}$

Θεώρημα

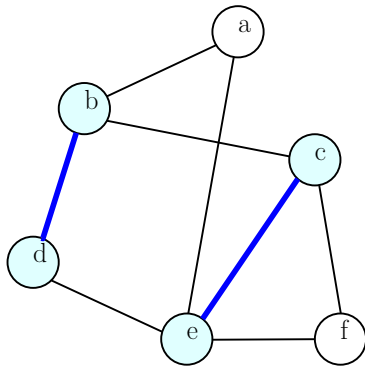
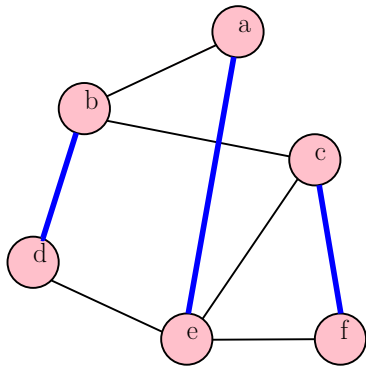
Ο αλγόριθμος VC-Match είναι 2-προσεγγιστικός.

Απόδειξη.

1. Το V' είναι εφικτή λύση (γιατί;)
2. $|M_{\max}| \leq OPT$
3. $SOL = |V'| = 2 \cdot |M_{\max}| \leq 2 \cdot OPT$



Παράδειγμα εκτέλεσης VC-Greedy



Σχήμα: Δύο εκτελέσεις του VC-Greedy με λύσεις κόστους 6 ($= 2OPT$) και 4 ($= \frac{4}{3}OPT$)

- ▶ Το άνω φράγμα 2 είναι **ανελαστικό (tight)**: αποδεικνύεται ότι ο αλγόριθμος VC-Match δεν μπορεί να πετύχει καλύτερο λόγο προσέγγισης. Δηλαδή, το 2 είναι και **κάτω φράγμα** για τον λόγο προσέγγισης του αλγορίθμου.

Ανελαστικά φράγματα (tight bounds)

- ▶ Το άνω φράγμα 2 είναι **ανελαστικό (tight)**: αποδεικνύεται ότι ο αλγόριθμος VC-Match δεν μπορεί να πετύχει καλύτερο λόγο προσέγγισης. Δηλαδή, το 2 είναι και **κάτω φράγμα** για τον λόγο προσέγγισης του αλγορίθμου.
- ▶ Συνήθης μέθοδος απόδειξης: εύρεση **ανελαστικού παραδείγματος (tight example)**, δηλαδή άπειρης οικογένειας στιγμιοτύπων για τα οποία ο αλγόριθμος δεν μπορεί να πετύχει λόγο προσέγγισης $2 - \varepsilon$, για οποιοδήποτε σταθερά $\varepsilon > 0$.

Ανελαστικά φράγματα (tight bounds)

- ▶ Το άνω φράγμα 2 είναι **ανελαστικό (tight)**: αποδεικνύεται ότι ο αλγόριθμος VC-Match δεν μπορεί να πετύχει καλύτερο λόγο προσέγγισης. Δηλαδή, το 2 είναι και **κάτω φράγμα** για τον λόγο προσέγγισης του αλγορίθμου.
- ▶ Συνήθης μέθοδος απόδειξης: εύρεση **ανελαστικού παραδείγματος (tight example)**, δηλαδή άπειρης οικογένειας στιγμιοτύπων για τα οποία ο αλγόριθμος δεν μπορεί να πετύχει λόγο προσέγγισης $2 - \varepsilon$, για οποιοδήποτε σταθερά $\varepsilon > 0$.
- ▶ Ένα tight example για αλγόριθμο VC-Match: **πλήρεις διμερείς γράφοι $K_{n,n}$** .

Κάτω φράγματα προσεγγισιμότητας προβλημάτων

- ▶ Δυσκολότερη απόδειξη: πρέπει να ισχύει για **κάθε αλγόριθμο**.

Κάτω φράγματα προσεγγισιμότητας προβλημάτων

- ▶ Δυσκολότερη απόδειξη: πρέπει να ισχύει για **κάθε αλγόριθμο**.
- ▶ Συνήθως **υπό συνθήκη**, π.χ. **$P \neq NP$** .

Κάτω φράγματα προσεγγισιμότητας προβλημάτων

- ▶ Δυσκολότερη απόδειξη: πρέπει να ισχύει για **κάθε αλγόριθμο**.
- ▶ Συνήθως **υπό συνθήκη**, π.χ. $\mathbf{P} \neq \mathbf{NP}$.
- ▶ Συνήθης μέθοδος: **αναγωγές εισαγωγής χάσματος (gap introducing reductions)** από γνωστά **NP**-πλήρη προβλήματα στο υπό εξέταση πρόβλημα βελτιστοποίησης.
Παράδειγμα: **TSP** (πώς;)

Κάτω φράγματα προσεγγισιμότητας προβλημάτων

- ▶ Δυσκολότερη απόδειξη: πρέπει να ισχύει για **κάθε αλγόριθμο**.
- ▶ Συνήθως **υπό συνθήκη**, π.χ. $\mathbf{P} \neq \mathbf{NP}$.
- ▶ Συνήθης μέθοδος: **αναγωγές εισαγωγής χάσματος (gap introducing reductions)** από γνωστά **NP**-πλήρη προβλήματα στο υπό εξέταση πρόβλημα βελτιστοποίησης.
Παράδειγμα: **TSP** (πώς;)
- ▶ Επίσης: **αναγωγές διατήρησης προσεγγισιμότητας (gap preserving reductions)** μεταξύ προβλημάτων βελτιστοποίησης.
Παράδειγμα: **Steiner Tree** (προσεχώς)

- ▶ Ένα διαφορετικό είδος ανελαστικότητας (tightness): αφορά στο λόγο της αντικειμενικής τιμής της βέλτιστης λύσης OPT προς το κάτω φράγμα $|M_{\max}|$ που χρησιμοποιεί ο VC-match.
- ▶ Για το **Vertex Cover**, εξετάζοντας πλήρεις γράφους (K_n) προκύπτει ότι το μέγεθος του maximal matching ως κάτω φράγμα για το OPT δεν μπορεί να οδηγήσει σε καλύτερο λόγο προσέγγισης από 2.

- ▶ Ένα διαφορετικό είδος ανελαστικότητας (tightness): αφορά στο λόγο της αντικειμενικής τιμής της βέλτιστης λύσης OPT προς το κάτω φράγμα $|M_{\max}|$ που χρησιμοποιεί ο VC-match.
- ▶ Για το **Vertex Cover**, εξετάζοντας πλήρεις γράφους (K_n) προκύπτει ότι το μέγεθος του maximal matching ως κάτω φράγμα για το OPT δεν μπορεί να οδηγήσει σε καλύτερο λόγο προσέγγισης από 2.
- ▶ Για παράδειγμα, δεν μπορεί να υπάρχει κανένας αλγόριθμος που να βρίσκει, σε κάθε γράφο, vertex cover μεγέθους $\leq \frac{3}{2} |M_{\max}|$.
- ▶ Φαινομενικά παράδοξο: ο VC-match επιτυγχάνει (σχεδόν) βέλτιστη λύση για κάθε γράφο K_n .

- ▶ Ένα διαφορετικό είδος ανελαστικότητας (tightness): αφορά στο λόγο της αντικειμενικής τιμής της βέλτιστης λύσης OPT προς το κάτω φράγμα $|M_{\max}|$ που χρησιμοποιεί ο VC-match.
- ▶ Για το **Vertex Cover**, εξετάζοντας πλήρεις γράφους (K_n) προκύπτει ότι το μέγεθος του maximal matching ως κάτω φράγμα για το OPT δεν μπορεί να οδηγήσει σε καλύτερο λόγο προσέγγισης από 2.
- ▶ Για παράδειγμα, δεν μπορεί να υπάρχει κανένας αλγόριθμος που να βρίσκει, σε κάθε γράφο, vertex cover μεγέθους $\leq \frac{3}{2} |M_{\max}|$.
- ▶ Φαινομενικά παράδοξο: ο VC-match επιτυγχάνει (σχεδόν) βέλτιστη λύση για κάθε γράφο K_n .
Εξήγηση: Η ανελαστικότητα αυτή μας δίνει μια αξιολόγηση του κάτω φράγματος και όχι κάποιου αλγορίθμου.

Το πρόβλημα **Weighted Vertex Cover (VC)**

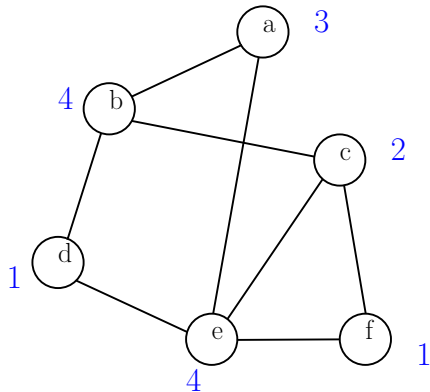
Weighted Vertex Cover

Δίνεται: γράφος $G(V, E)$, συνάρτηση βάρους $w : V \rightarrow \mathbb{Q}^+$

Ζητείται: **κάλυμμα κορυφών ελάχιστου βάρους**, δηλαδή σύνολο κορυφών V^* που να είναι κάλυμμα κορυφών για το G και το συνολικό βάρος των κορυφών του V^* να είναι το ελάχιστο δυνατό:

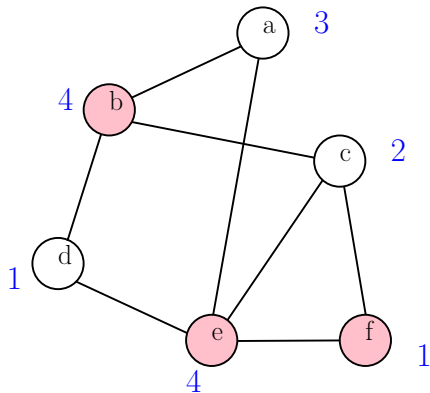
$$V^* = \arg \min_{V' \subseteq V, \forall \{x,y\} \in E: x \in V' \vee y \in V'} \sum_{v \in V'} w(v)$$

Παράδειγμα (**Weighted Vertex Cover**)



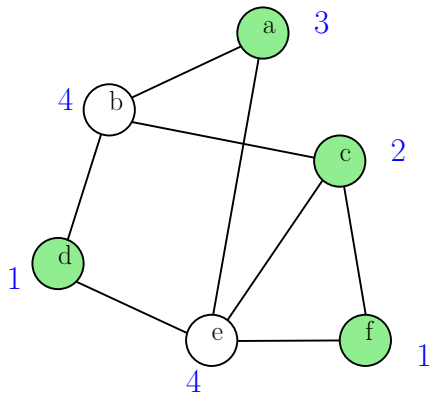
Σχήμα: Στιγμιότυπο του προβλήματος **Weighted Vertex Cover**

Παράδειγμα (**Weighted Vertex Cover**)



Σχήμα: Μια μη βέλτιστη λύση (κόστος 9)

Παράδειγμα (**Weighted Vertex Cover**)



Σχήμα: Μια βέλτιστη λύση (κόστος 7)

Προσεγγίζοντας το **Weighted Vertex Cover (WVC)**

Ορισμός

Degree-weighted συνάρτηση $w : V \rightarrow \mathbf{Q}$ (σε γράφο $G(V, E)$):
 $\exists c > 0, \forall v \in V, w(v) = c \cdot \text{deg}(v)$, όπου $\text{deg}(v)$ είναι ο βαθμός της κορυφής v .

Λήμμα

Αν η συνάρτηση βαρών w σε γράφο $G(V, E)$ είναι *degree-weighted* τότε $w(V) \leq 2 \cdot OPT_{WVC}$

Απόδειξη.

- ▶ Αν U είναι vertex cover τότε $deg(U) \geq |E|$
(συμβολισμός: $deg(U) = \sum_{u \in U} deg(u)$)

Λήμμα

Αν η συνάρτηση βαρών w σε γράφο $G(V, E)$ είναι *degree-weighted* τότε $w(V) \leq 2 \cdot OPT_{WVC}$

Απόδειξη.

- ▶ Αν U είναι vertex cover τότε $deg(U) \geq |E|$
(συμβολισμός: $deg(U) = \sum_{u \in U} deg(u)$)
- ▶ Όμως $deg(V) = 2|E| \leq 2deg(U)$

Λήμμα

Αν η συνάρτηση βαρών w σε γράφο $G(V, E)$ είναι *degree-weighted* τότε $w(V) \leq 2 \cdot OPT_{WVC}$

Απόδειξη.

- ▶ Αν U είναι vertex cover τότε $deg(U) \geq |E|$
(συμβολισμός: $deg(U) = \sum_{u \in U} deg(u)$)
- ▶ Όμως $deg(V) = 2|E| \leq 2deg(U)$
- ▶ Επομένως, αν w είναι degree-weighted: $w(V) \leq 2w(U)$

Λήμμα

Αν η συνάρτηση βαρών w σε γράφο $G(V, E)$ είναι *degree-weighted* τότε $w(V) \leq 2 \cdot OPT_{WVC}$

Απόδειξη.

- ▶ Αν U είναι vertex cover τότε $deg(U) \geq |E|$
(συμβολισμός: $deg(U) = \sum_{u \in U} deg(u)$)
- ▶ Όμως $deg(V) = 2|E| \leq 2deg(U)$
- ▶ Επομένως, αν w είναι degree-weighted: $w(V) \leq 2w(U)$
- ▶ Αυτό ισχύει και για vertex cover U_{OPT} ελαχίστου βάρους. Άρα:
$$w(V) \leq 2w(U_{OPT}) = 2OPT_{WVC}$$



Λήμμα

Αν η συνάρτηση βαρών w σε γράφο $G(V, E)$ είναι *degree-weighted* τότε $w(V) \leq 2 \cdot OPT_{WVC}$

Απόδειξη.

- ▶ Αν U είναι vertex cover τότε $deg(U) \geq |E|$
(συμβολισμός: $deg(U) = \sum_{u \in U} deg(u)$)
- ▶ Όμως $deg(V) = 2|E| \leq 2deg(U)$
- ▶ Επομένως, αν w είναι degree-weighted: $w(V) \leq 2w(U)$
- ▶ Αυτό ισχύει και για vertex cover U_{OPT} ελαχίστου βάρους. Άρα:
$$w(V) \leq 2w(U_{OPT}) = 2OPT_{WVC}$$



Ερώτηση: τι συμπεραίνουμε από το παραπάνω λήμμα;

Προσεγγίζοντας το **Weighted Vertex Cover (WVC)**

Ιδέα: διάσπαση της δοθείσης συνάρτησης βάρους σε degree-weighted συναρτήσεις.

Αλγόριθμος Degree-Weighted-WVC (layering)

Επανάλαβε για όσο υπάρχουν κορυφές στο γράφο

- αφαίρεσε κορυφές μηδενικού βαθμού
- βρές μέγιστο c τ.ώ. $\forall v \in V, c \cdot \text{deg}(v) \leq w(v)$
(όπου w η τρέχουσα συνάρτηση βάρους)
- $\forall v \in V$ θέσε $w(v) \leftarrow w(v) - c \cdot \text{deg}(v)$
- πρόσθεσε κορυφές βάρους 0 στην κάλυψη
και αφαίρεσέ τις από το γράφο.

Θεώρημα

Ο παραπάνω αλγόριθμος είναι 2-προσεγγιστικός για το πρόβλημα **WVC**.

Set Cover

Δίνεται: σύνολο U με n στοιχεία και συλλογή υποσυνόλων του U ,
 $\mathcal{S} = \{S_1, \dots, S_k\}$, $S_i \subseteq U$

Ζητείται: ελάχιστης πληθικότητας συλλογή $\mathcal{S}' \subseteq \mathcal{S}$ τ.ώ. κάθε στοιχείο του U να ανήκει σε τουλάχιστον ένα σύνολο της $\mathcal{S}'$: $\bigcup_{S \in \mathcal{S}'} S = U$.

Weighted Set Cover: τα υποσύνολα έχουν βάρος (κόστος) και το ζητούμενο είναι η $\mathcal{S}'$ να είναι ελαχίστου βάρους.

Σημείωση: Συχνά ο όρος **Set Cover** χρησιμοποιείται για την weighted εκδοχή του προβλήματος. Τότε για το πρώτο πρόβλημα χρησιμοποιούνται οι όροι “Unweighted” ή “Cardinality”.

Ο άπληστος αλγόριθμος για το **Set Cover**

Αλγόριθμος SC-Greedy

$C \leftarrow \emptyset$

Επανάλαβε για όσο $C \neq U$:

- Βρες $S_i \in S$ που μεγιστοποιεί $|S_i \setminus C|$
- $C \leftarrow C \cup S_i$

Ο άπληστος αλγόριθμος για το **Set Cover**

Αλγόριθμος SC-Greedy

$C \leftarrow \emptyset$

Επανάλαβε για όσο $C \neq U$:

- Βρες $S_i \in S$ που μεγιστοποιεί $|S_i \setminus C|$
- $C \leftarrow C \cup S_i$

Θεώρημα

Ο αλγόριθμος SC-Greedy είναι **log n-προσεγγιστικός** για το πρόβλημα **Set Cover**.

Ο άπληστος αλγόριθμος για το **Set Cover**

Αλγόριθμος SC-Greedy

$C \leftarrow \emptyset$

Επανάλαβε για όσο $C \neq U$:

- Βρες $S_i \in S$ που μεγιστοποιεί $|S_i \setminus C|$
- $C \leftarrow C \cup S_i$

Θεώρημα

Ο αλγόριθμος SC-Greedy είναι **log n-προσεγγιστικός** για το πρόβλημα **Set Cover**.

Ιδέα απόδειξης: μετά από $k = OPT$ επαναλήψεις καλύπτονται τουλάχιστον τα μισά (ακάλυπτα) στοιχεία του U .

Ο άπληστος αλγόριθμος για το **Set Cover**

Αλγόριθμος SC-Greedy

$C \leftarrow \emptyset$

Επανάλαβε για όσο $C \neq U$:

- Βρες $S_i \in \mathcal{S}$ που μεγιστοποιεί $|S_i \setminus C|$
- $C \leftarrow C \cup S_i$

Θεώρημα

Ο αλγόριθμος SC-Greedy είναι **log n-προσεγγιστικός** για το πρόβλημα **Set Cover**.

Ιδέα απόδειξης: μετά από $k = OPT$ επαναλήψεις καλύπτονται τουλάχιστον τα μισά (ακάλυπτα) στοιχεία του U .

Αν όχι, $\exists S^* \in C_{opt}$, που δεν επιλέχθηκε στις πρώτες k επαναλήψεις, που καλύπτει τουλάχιστον $\frac{n}{2k}$ ακάλυπτα στοιχεία. **Άτοπο!**

Tight example για τον αλγόριθμο SC-Greedy

$$U = \bigcup_{i=1}^t \{a_{2^{i-1}}, \dots, a_{2^i-1}\} \cup$$

$$\bigcup_{i=1}^t \{a'_{2^{i-1}}, \dots, a'_{2^i-1}\}$$

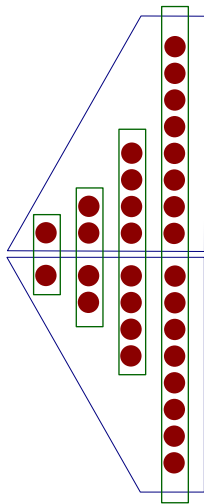
$$S_i = \{a_{2^{i-1}}, \dots, a_{2^i-1}\} \cup \{a'_{2^{i-1}}, \dots, a'_{2^i-1}\},$$
$$i = 1, \dots, t$$

$$S_{k+1} = \bigcup_{i=1}^t \{a_{2^{i-1}}, \dots, a_{2^i-1}\}$$

$$S_{k+2} = \bigcup_{i=1}^t \{a'_{2^{i-1}}, \dots, a'_{2^i-1}\}$$

$$SOL = t \approx \log n, \quad OPT = 2$$

Λόγος προσέγγισης: $\Theta(\log n)$



Αλγόριθμος Weighted SC-Greedy

$C \leftarrow \emptyset$

Επανάλαβε για όσο $C \neq U$:

- Βρές S_i με ελάχιστο $\alpha_i = \text{cost}(S_i) / |S_i - C|$
- $\forall e \in S_i - C$ θέσε $\text{price}(e) \leftarrow \alpha_i$
- $C \leftarrow C \cup S_i$

Σημείωση: $\text{price}(e_k)$ είναι η τιμή που “πληρώσαμε” για να καλυφθεί το στοιχείο e_k .

Συνολικό κόστος κάλυψης: $\text{SOL} = \sum_{k=1}^n \text{price}(e_k)$.

Ανάλυση του Weighted SC-Greedy

Θεώρημα

Ο αλγόριθμος *Weighted SC-Greedy* είναι H_n -προσεγγιστικός, όπου $H_n = 1 + \frac{1}{2} + \dots \frac{1}{n} < \ln n + 1$.

Απόδειξη.

Υποθέτουμε ότι τα στοιχεία του U αριθμούνται με τη σειρά που καλύπτονται από τον αλγόριθμο. Παρατηρούμε ότι:

- ▶ Σε οποιαδήποτε επανάληψη του αλγορίθμου μπορούμε να καλύψουμε όλα τα ακάλυπτα στοιχεία με κόστος το πολύ OPT (γιατί;).

Ανάλυση του Weighted SC-Greedy

Θεώρημα

Ο αλγόριθμος *Weighted SC-Greedy* είναι H_n -προσεγγιστικός, όπου $H_n = 1 + \frac{1}{2} + \dots \frac{1}{n} < \ln n + 1$.

Απόδειξη.

Υποθέτουμε ότι τα στοιχεία του U αριθμούνται με τη σειρά που καλύπτονται από τον αλγόριθμο. Παρατηρούμε ότι:

- ▶ Σε οποιαδήποτε επανάληψη του αλγορίθμου μπορούμε να καλύψουμε όλα τα ακάλυπτα στοιχεία με κόστος το πολύ OPT (γιατί;).
- ▶ Επομένως υπάρχει πάντα σύνολο με κόστος / νέο στοιχείο το πολύ $OPT / |U - C|$ (C : η τρέχουσα κάλυψη).

Ανάλυση του Weighted SC-Greedy

Θεώρημα

Ο αλγόριθμος *Weighted SC-Greedy* είναι H_n -προσεγγιστικός, όπου $H_n = 1 + \frac{1}{2} + \dots \frac{1}{n} < \ln n + 1$.

Απόδειξη.

Υποθέτουμε ότι τα στοιχεία του U αριθμούνται με τη σειρά που καλύπτονται από τον αλγόριθμο. Παρατηρούμε ότι:

- ▶ Σε οποιαδήποτε επανάληψη του αλγορίθμου μπορούμε να καλύψουμε όλα τα ακάλυπτα στοιχεία με κόστος το πολύ OPT (γιατί;).
- ▶ Επομένως υπάρχει πάντα σύνολο με κόστος / νέο στοιχείο το πολύ $OPT / |U - C|$ (C : η τρέχουσα κάλυψη).
- ▶ Πριν καλυφθεί το στοιχείο e_k για πρώτη φορά ισχύει $|U - C| \geq n - k + 1$. Άρα $price(e_k) \leq \frac{OPT}{n-k+1}$.

Συνολικό κόστος: $SOL \leq \sum_{k=1}^n \frac{OPT}{n-k+1} = H_n \cdot OPT$

□

Tight example

$$U = \{a_1, \dots, a_n\}$$

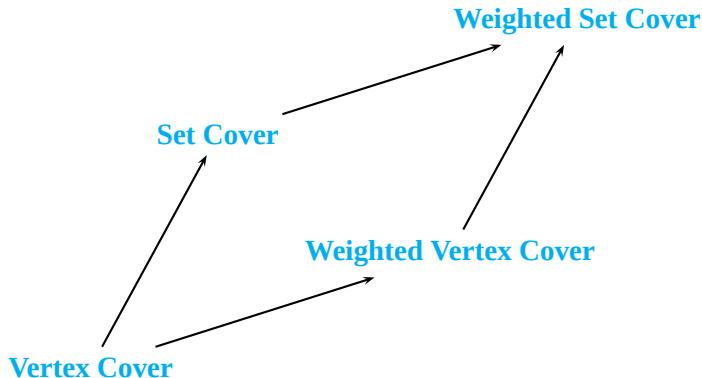
$$\forall i \in \{1, \dots, n\}, S_i = \{a_i\}, \text{cost}(S_i) = \frac{1}{i}$$

$$S_{n+1} = U, \text{cost}(S_{n+1}) = 1 + \varepsilon, \text{για } \varepsilon > 0$$

$$SOL = H_n, \quad OPT = 1 + \varepsilon$$

Επομένως $\rho(n) > H_n - \varepsilon'$ για οποιοδήποτε ε' .

Σημείωση: Αποδεικνύεται ότι ο αλγόριθμος έχει το ίδιο κάτω φράγμα λόγου προσέγγισης και για το **(Cardinality) Set Cover** και για το **Vertex Cover**!



Σημείωση: κάτω φράγματα στο λόγο προσέγγισης ‘διαδίδονται’ προς τα πάνω, άνω φράγματα προς τα κάτω

Μερικά γνωστά φράγματα

- ▶ **Weighted Set Cover**: H_n -προσεγγιστικός αλγόριθμος (ειδικότερα: $H_{|S_{\max}|}$).
- ▶ **Weighted Set Cover**: f -προσεγγιστικός αλγόριθμος.
- ▶ **(Weighted) Set Cover**: μη προσεγγίσιμο με λόγο $(1 - o(1)) \ln n$ εκτός εάν $\mathbf{P} = \mathbf{NP}$ [Dinur-Steurer 2013].
- ▶ **(Weighted) Vertex Cover**: μη προσεγγίσιμο με λόγο 1.3606 εκτός εάν $\mathbf{P} = \mathbf{NP}$ [Dinur-Safra 2005].

Μερικά γνωστά φράγματα

- ▶ **Weighted Set Cover**: H_n -προσεγγιστικός αλγόριθμος (ειδικότερα: $H_{|S_{\max}|}$).
- ▶ **Weighted Set Cover**: f -προσεγγιστικός αλγόριθμος.
- ▶ **(Weighted) Set Cover**: μη προσεγγίσιμο με λόγο $(1 - o(1)) \ln n$ εκτός εάν $\mathbf{P} = \mathbf{NP}$ [Dinur-Steurer 2013].
- ▶ **(Weighted) Vertex Cover**: μη προσεγγίσιμο με λόγο 1.3606 εκτός εάν $\mathbf{P} = \mathbf{NP}$ [Dinur-Safra 2005].
Μη προσεγγίσιμο με λόγο $2 - \varepsilon$ αν ισχύει η εικασία Unique Games Conjecture [Khot-Regev 2008].

Μερικά γνωστά φράγματα

- ▶ **Weighted Set Cover**: H_n -προσεγγιστικός αλγόριθμος (ειδικότερα: $H_{|S_{\max}|}$).
- ▶ **Weighted Set Cover**: f -προσεγγιστικός αλγόριθμος.
- ▶ **(Weighted) Set Cover**: μη προσεγγίσιμο με λόγο $(1 - o(1)) \ln n$ εκτός εάν $\mathbf{P} = \mathbf{NP}$ [Dinur-Steurer 2013].
- ▶ **(Weighted) Vertex Cover**: μη προσεγγίσιμο με λόγο 1.3606 εκτός εάν $\mathbf{P} = \mathbf{NP}$ [Dinur-Safra 2005].
Μη προσεγγίσιμο με λόγο $2 - \varepsilon$ αν ισχύει η εικασία **Unique Games Conjecture** [Khot-Regev 2008].
Καλύτερο γνωστό άνω φράγμα: $2 - \Theta(\frac{1}{\sqrt{\log n}})$ [Karakostas 2004].

Πρόβλημα μεγιστοποίησης: Maximum Coverage

Δίνεται: σύνολο U με n στοιχεία, συλλογή $\mathcal{S} = \{S_1, \dots, S_l\}$ υποσυνόλων του U , και ακέραιος k .

Ζητείται: Μία συλλογή $\mathcal{S}' \subseteq \mathcal{S}$ αποτελούμενη από k σύνολα τ.ώ. το πλήθος των στοιχείων που καλύπτει η $\mathcal{S}'$ να είναι μέγιστο.

Πρόβλημα μεγιστοποίησης: **Maximum Coverage**

Δίνεται: **σύνολο** U με n στοιχεία, συλλογή $\mathcal{S} = \{S_1, \dots, S_l\}$ υποσυνόλων του U , και **ακέραιος** k .

Ζητείται: Μία **συλλογή** $\mathcal{S}' \subseteq \mathcal{S}$ αποτελούμενη από k σύνολα τ.ώ. το πλήθος των στοιχείων που καλύπτει η $\mathcal{S}'$ να είναι μέγιστο.

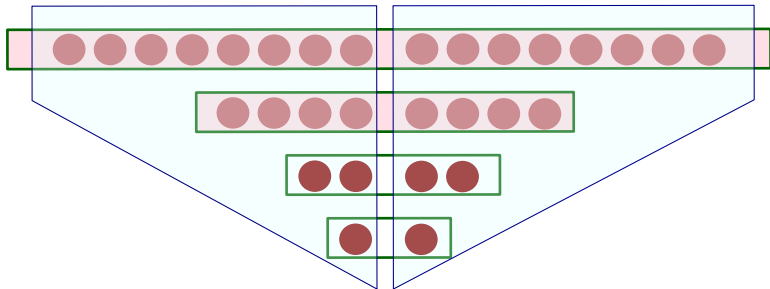
Θεώρημα

Ο άπληστος αλγόριθμος *SC-Greedy* που κάθε φορά επιλέγει το μεγαλύτερο, ως προς πλήθος νέων στοιχείων, σύνολο επιτυγχάνει (με k επαναλήψεις) λόγο προσέγγισης

$$1 - (1 - \frac{1}{k})^k > 1 - \frac{1}{e}$$

για το πρόβλημα **Maximum Coverage**.

Σημείωση: Το πρόβλημα ανήκει στην κλάση **APX**, παρ'ότι το **Set Cover** δεν ανήκει!



Εκτέλεση SC-Greedy για **Maximum Coverage** με $k = 2$:

$$n_{opt} = 30, \quad n_{sol} = 24$$

Γενική περίπτωση, $|U| = 2 \cdot (2^t - 1)$:

$$n_{opt} = |U|, \quad n_{sol} = 3 \cdot 2^{t-1} > \frac{3}{4}n_{opt}$$

Απόδειξη λόγου προσέγγισης $1 - \frac{1}{e}$ για το **Maximum Coverage** (i)

- ▶ Έστω $\mathcal{S}_{opt}$ μια βέλτιστη λύση που καλύπτει πλήθος στοιχείων n_{opt} . Αφού το πλήθος των συνόλων της $\mathcal{S}_{opt}$ είναι k θα πρέπει να υπάρχει σύνολο στην $\mathcal{S}$ με πληθικότητα $\geq \frac{n_{opt}}{k}$. Επομένως, ο άπληστος αλγόριθμος στο 1ο βήμα θα βρεί και θα επιλέξει σύνολο τουλάχιστον τόσων στοιχείων.

Απόδειξη λόγου προσέγγισης $1 - \frac{1}{e}$ για το **Maximum Coverage** (i)

- ▶ Έστω S_{opt} μια βέλτιστη λύση που καλύπτει πλήθος στοιχείων n_{opt} . Αφού το πλήθος των συνόλων της S_{opt} είναι k θα πρέπει να υπάρχει σύνολο στην $\mathcal{S}$ με πληθικότητα $\geq \frac{n_{opt}}{k}$. Επομένως, ο άπληστος αλγόριθμος στο 1ο βήμα θα βρεί και θα επιλέξει σύνολο τουλάχιστον τόσων στοιχείων.
- ▶ Με άλλα λόγια, στο 1ο βήμα, καλύπτεται πλήθος στοιχείων που αντιπροσωπεύουν **τουλάχιστον το $\frac{1}{k}$ του n_{opt}** ή, ισοδύναμα, μένει **ακάλυπτο το πολύ ένα μέρος $1 - \frac{1}{k}$** των στοιχείων της βέλτιστης λύσης S_{opt} .

Απόδειξη λόγου προσέγγισης $1 - \frac{1}{e}$ για το **Maximum Coverage** (i)

- ▶ Έστω S_{opt} μια βέλτιστη λύση που καλύπτει πλήθος στοιχείων n_{opt} . Αφού το πλήθος των συνόλων της S_{opt} είναι k θα πρέπει να υπάρχει σύνολο στην $\mathcal{S}$ με πληθικότητα $\geq \frac{n_{opt}}{k}$. Επομένως, ο άπληστος αλγόριθμος στο 1ο βήμα θα βρεί και θα επιλέξει σύνολο τουλάχιστον τόσων στοιχείων.
- ▶ Με άλλα λόγια, στο 1ο βήμα, καλύπτεται πλήθος στοιχείων που αντιπροσωπεύουν **τουλάχιστον το $\frac{1}{k}$ του n_{opt}** ή, ισοδύναμα, μένει **ακάλυπτο το πολύ ένα μέρος $1 - \frac{1}{k}$** των στοιχείων της βέλτιστης λύσης S_{opt} .

Σημείωση: Κάθε φορά που ο άπληστος αλγόριθμος καλύπτει στοιχεία εκτός βέλτιστης λύσης μπορούμε (για την απόδειξη) να ‘διαγράψουμε’ ισάριθμο πλήθος στοιχείων από τη βέλτιστη λύση, θεωρώντας ότι το αντίστοιχο μέρος έχει καλυφθεί.

Απόδειξη λόγου προσέγγισης $(1 - \frac{1}{e})$ — για το **Maximum Coverage** (ii)

- ▶ Με ανάλογα επιχειρήματα, μπορούμε να δείξουμε ότι στο i -οστό βήμα το ακάλυπτο μέρος της βέλτιστης λύσης μειώνεται τουλάχιστον κατά $\frac{1}{k}$ (αφού τα k σύνολα της $\mathcal{S}_{opt}$ αρκούν για να το καλύψουν πλήρως) και επομένως (επαγωγικά) απομένει ακάλυπτο το πολύ ένα μέρος $(1 - \frac{1}{k})^i$ της βέλτιστης λύσης.

Απόδειξη λόγου προσέγγισης $(1 - \frac{1}{e})$ — για το **Maximum Coverage** (ii)

- ▶ Με ανάλογα επιχειρήματα, μπορούμε να δείξουμε ότι στο i -οστό βήμα το ακάλυπτο μέρος της βέλτιστης λύσης μειώνεται τουλάχιστον κατά $\frac{1}{k}$ (αφού τα k σύνολα της $\mathcal{S}_{opt}$ αρκούν για να το καλύψουν πλήρως) και επομένως (επαγωγικά) απομένει ακάλυπτο το πολύ ένα μέρος $(1 - \frac{1}{k})^i$ της βέλτιστης λύσης.
- ▶ Τελικά, σε k βήματα έχει καλυφθεί τουλάχιστον το $1 - (1 - \frac{1}{k})^k$ της βέλτιστης λύσης. Επομένως:

$$SOL \geq (1 - (1 - \frac{1}{k})^k)n_{opt} > (1 - \frac{1}{e})n_{opt}$$

Απόδειξη λόγου προσέγγισης $(1 - \frac{1}{e})$ — για το **Maximum Coverage** (ii)

- ▶ Με ανάλογα επιχειρήματα, μπορούμε να δείξουμε ότι στο i -οστό βήμα το ακάλυπτο μέρος της βέλτιστης λύσης μειώνεται τουλάχιστον κατά $\frac{1}{k}$ (αφού τα k σύνολα της $\mathcal{S}_{opt}$ αρκούν για να το καλύψουν πλήρως) και επομένως (επαγωγικά) απομένει ακάλυπτο το πολύ ένα μέρος $(1 - \frac{1}{k})^i$ της βέλτιστης λύσης.
- ▶ Τελικά, σε k βήματα έχει καλυφθεί τουλάχιστον το $1 - (1 - \frac{1}{k})^k$ της βέλτιστης λύσης. Επομένως:

$$SOL \geq (1 - (1 - \frac{1}{k})^k)n_{opt} > (1 - \frac{1}{e})n_{opt}$$

(Χρήσιμη ιδιότητα: $\forall x \in \mathbb{R}, 1 + x \leq e^x$)

- ▶ Το αποτέλεσμα ισχύει και στην περίπτωση που τα στοιχεία έχουν βάρος και ζητείται λύση που καλύπτει **μέγιστο βάρος** στοιχείων.

- ▶ Το αποτέλεσμα ισχύει και στην περίπτωση που τα στοιχεία έχουν βάρος και ζητείται λύση που καλύπτει **μέγιστο βάρος** στοιχείων.
- ▶ Το αποτέλεσμα ισχύει ακόμη και σε περιπτώσεις όπου **δεν δίνονται τα σύνολα της $\mathcal{S}$ αναλυτικά** (μπορεί να είναι και εκθετικά πολλά), αλλά μπορούμε να βρίσκουμε το “καλύτερο” σύνολο σε πολυωνυμικό χρόνο.

Εφαρμογή: **k-Activity Selection** (aka **Max Interval (Graph) Coloring**).

- ▶ Το αποτέλεσμα ισχύει και στην περίπτωση που τα στοιχεία έχουν βάρος και ζητείται λύση που καλύπτει **μέγιστο βάρος** στοιχείων.
- ▶ Το αποτέλεσμα ισχύει ακόμη και σε περιπτώσεις όπου **δεν δίνονται τα σύνολα της $\mathcal{S}$ αναλυτικά** (μπορεί να είναι και εκθετικά πολλά), αλλά μπορούμε να βρίσκουμε το “καλύτερο” σύνολο σε πολυωνυμικό χρόνο.

Εφαρμογή: **k-Activity Selection** (aka **Max Interval (Graph) Coloring**).

- ▶ Εάν μπορούμε να βρίσκουμε το “καλύτερο” σύνολο σε πολυωνυμικό χρόνο με προσέγγιση $\rho (\leq 1)$ τότε ο άπληστος αλγόριθμος επιτυγχάνει λόγο προσέγγισης

$$1 - (1 - \frac{\rho}{k})^k > 1 - \frac{1}{e^\rho}$$

για το πρόβλημα **Maximum Coverage**.

- ▶ Το αποτέλεσμα ισχύει και στην περίπτωση που τα στοιχεία έχουν βάρος και ζητείται λύση που καλύπτει **μέγιστο βάρος** στοιχείων.
- ▶ Το αποτέλεσμα ισχύει ακόμη και σε περιπτώσεις όπου **δεν δίνονται τα σύνολα της $\mathcal{S}$ αναλυτικά** (μπορεί να είναι και εκθετικά πολλά), αλλά μπορούμε να βρίσκουμε το “καλύτερο” σύνολο σε πολυωνυμικό χρόνο.

Εφαρμογή: **k -Activity Selection** (aka **Max Interval (Graph) Coloring**).

- ▶ Εάν μπορούμε να βρίσκουμε το “καλύτερο” σύνολο σε πολυωνυμικό χρόνο με προσέγγιση ρ (≤ 1) τότε ο άπληστος αλγόριθμος επιτυγχάνει λόγο προσέγγισης

$$1 - (1 - \frac{\rho}{k})^k > 1 - \frac{1}{e^\rho}$$

για το πρόβλημα **Maximum Coverage**.

- ▶ Ενδιαφέρουσα γενίκευση: **monotone submodular functions**.

Το Πρόβλημα του Πλανόδιου Πωλητή

Traveling Salesman problem (TSP)

Ορισμός του προβλήματος TSP

Δίνεται: *πλήρης γράφος* $G(V, E)$ με μη αρνητικά κόστη (βάρη) στις ακμές του.

Ζητείται: *κύκλος ελαχίστου κόστους* που να επισκέπτεται κάθε κόμβο ακριβώς μία φορά (Hamilton Cycle).

Θεώρημα

Το πρόβλημα *TSP* δεν μπορεί να προσεγγιστεί με παράγοντα $\alpha(n)$, όπου $n = |V|$, για οποιαδήποτε πολυωνυμικά υπολογιστή συνάρτηση $\alpha : \mathbb{N} \rightarrow \mathbb{Q}^+$, εκτός εάν **P** = **NP**.

Θεώρημα

Το πρόβλημα TSP δεν μπορεί να προσεγγιστεί με παράγοντα $\alpha(n)$, όπου $n = |V|$, για οποιαδήποτε πολυωνυμικά υπολογιστή συνάρτηση $\alpha : \mathbb{N} \rightarrow \mathbb{Q}^+$, εκτός εάν $\mathbf{P} = \mathbf{NP}$.

Απόδειξη.

Αναγωγή από το **Hamilton Cycle**: συμπληρώνουμε τον αρχικό γράφο G με ακμές ώστε να κατασκευάσουμε πλήρη γράφο G' . Στις αρχικές ακμές δίνουμε βάρος 1, στις υπόλοιπες δίνουμε βάρος $\alpha(n) \cdot n$. Ισχύει ότι:

- ▶ Αν ο G είναι **Hamilton** τότε υπάρχει κύκλος TSP κόστους n στον G' , ενώ
- ▶ Αν ο G δεν είναι **Hamilton** τότε ο βέλτιστος κύκλος TSP στον G' έχει κόστος $> \alpha(n) \cdot n$.



Αναγωγές Εισαγωγής Χάσματος (i)

(Gap Introducing Reductions)

Μια αναγωγή h από το πρόβλημα απόφασης Π στο πρόβλημα ελαχιστοποίησης Π' (που απεικονίζει κάθε στιγμιότυπο I του Π σε κάποιο στιγμιότυπο $I' = h(I)$ του Π') λέγεται αναγωγή εισαγωγής χάσματος όταν υπάρχουν συναρτήσεις f, α ώστε:

- ▶ Αν το I είναι ‘yes’-instance του Π τότε $OPT_{\Pi'}(I') \leq f(I')$, ενώ
- ▶ Αν το I είναι ‘no’-instance του Π τότε $OPT_{\Pi'}(I') > \alpha(|I'|) \cdot f(I')$.

Θεώρημα

Αν το πρόβλημα Π είναι **NP**-complete και υπάρχει αναγωγή εισαγωγής χάσματος με παραμέτρους f, α από το Π στο πρόβλημα Π' τότε το Π' δεν προσεγγίζεται με παράγοντα $\alpha(n)$, εφ'όσον $\mathbf{P} \neq \mathbf{NP}$ (όπου n το μήκος της αναπαράστασης της εισόδου του Π').

Αναγωγές Εισαγωγής Χάσματος (ii) (Gap Introducing Reductions)

Μια αναγωγή h από το πρόβλημα απόφασης Π στο πρόβλημα μεγιστοποίησης Π' (που απεικονίζει κάθε στιγμιότυπο I του Π σε κάποιο στιγμιότυπο $I' = h(I)$ του Π') λέγεται αναγωγή εισαγωγής χάσματος όταν υπάρχουν συναρτήσεις f, α ώστε:

- ▶ Αν το I είναι ‘yes’-instance του Π τότε $OPT_{\Pi'}(I') \geq f(I')$, ενώ
- ▶ Αν το I είναι ‘no’-instance του Π τότε $OPT_{\Pi'}(I') < \alpha(|I'|) \cdot f(I')$.

Το πρόβλημα **Metric TSP**

Επιπλέον υπόθεση: βάρη ικανοποιούν τριγωνική ανισότητα.

Το πρόβλημα παραμένει **NP**-complete (γιατί;)

Το πρόβλημα **Metric TSP**

Επιπλέον υπόθεση: βάρη ικανοποιούν τριγωνική ανισότητα.

Το πρόβλημα παραμένει **NP**-complete (γιατί;)

2-προσεγγιστικός αλγόριθμος για το **Metric TSP**

- Βρες ελάχιστο συνδετικό δέντρο T στον G .
- Διπλασίασε ακμές του T .
- Βρες κύκλο Euler C στο διπλασιασμένο T .
- Επίστρεψε κύκλο που επισκέπτεται κόμβους με σειρά εμφάνισής τους στον C (**short-cutting**).

Ο παραπάνω αλγόριθμος είναι **2-προσεγγιστικός**:

Το πρόβλημα **Metric TSP**

Επιπλέον υπόθεση: βάρη ικανοποιούν τριγωνική ανισότητα.

Το πρόβλημα παραμένει **NP**-complete (γιατί;)

2-προσεγγιστικός αλγόριθμος για το **Metric TSP**

- Βρες ελάχιστο συνδετικό δέντρο T στον G .
- Διπλασίασε ακμές του T .
- Βρες κύκλο Euler C στο διπλασιασμένο T .
- Επίστρεψε κύκλο που επισκέπτεται κόμβους με σειρά εμφάνισής τους στον C (**short-cutting**).

Ο παραπάνω αλγόριθμος είναι **2-προσεγγιστικός**:

$$cost(C) \leq cost(C_T) \leq 2cost(T) \leq 2OPT$$

Καλύτερη προσέγγιση για το **Metric TSP** (Christofides' algorithm)

$\frac{3}{2}$ -προσεγγιστικός αλγόριθμος για το **Metric TSP**

```
/* Εύρεση ακόμη 'φτηνότερου' κύκλου Euler */  
- Βρες Eulerian completion του δέντρου  $T$ , μέσω  
minimum cost perfect matching  $M$  πάνω στους  
κόμβους περιττού βαθμού του  $T$ .  
- Συνέχισε όπως στον προηγούμενο αλγόριθμο.
```

Λήμμα: $\text{cost}(M) \leq \frac{1}{2}OPT$

Καλύτερη προσέγγιση για το **Metric TSP** (Christofides' algorithm)

$\frac{3}{2}$ -προσεγγιστικός αλγόριθμος για το **Metric TSP**

```
/* Εύρεση ακόμη 'φτηνότερου' κύκλου Euler */  
- Βρες Eulerian completion του δέντρου  $T$ , μέσω  
minimum cost perfect matching  $M$  πάνω στους  
κόμβους περιττού βαθμού του  $T$ .  
- Συνέχισε όπως στον προηγούμενο αλγόριθμο.
```

Λήμμα: $\text{cost}(M) \leq \frac{1}{2}OPT$

Απόδειξη: με short-cutting στον βέλτιστο κύκλο, παίρνουμε κύκλο C_{odd} κόστους $\leq OPT$ στους κόμβους περιττού βαθμού του T . Κόστος M το πολύ το μισό του κόστους του C_{odd} (γιατί;).

Καλύτερη προσέγγιση για το **Metric TSP** (Christofides' algorithm)

$\frac{3}{2}$ -προσεγγιστικός αλγόριθμος για το **Metric TSP**

```
/* Εύρεση ακόμη 'φτηνότερου' κύκλου Euler */  
- Βρες Eulerian completion του δέντρου  $T$ , μέσω  
minimum cost perfect matching  $M$  πάνω στους  
κόμβους περιττού βαθμού του  $T$ .  
- Συνέχισε όπως στον προηγούμενο αλγόριθμο.
```

Λήμμα: $cost(M) \leq \frac{1}{2}OPT$

Απόδειξη: με short-cutting στον βέλτιστο κύκλο, παίρνουμε κύκλο C_{odd} κόστους $\leq OPT$ στους κόμβους περιττού βαθμού του T . Κόστος M το πολύ το μισό του κόστους του C_{odd} (γιατί;).

$$cost(C) \leq cost(C_{T,M}) = cost(T) + cost(M) \leq \frac{3}{2}OPT$$

Το πρόβλημα **Metric TSP**_{(s,t)-path}

Ορισμός του προβλήματος **Metric TSP**_{(s,t)-path}

Δίνονται: γράφος με βάρη (όπως για το **Metric TSP**) και επιπλέον 2 κόμβοι s, t .

Ζητείται: **Hamilton path** ελαχίστου κόστους από s σε t .

Αλγόριθμος: συνδυασμός 2 ανεξάρτητων αλγορίθμων, καθένας δημιουργεί γράφο με **μονοπάτι Euler** με διαφορετικό τρόπο (βλ. επόμενη διαφάνεια).

Η επιλογή της καλύτερης από τις 2 λύσεις δίνει $\frac{5}{3}$ -προσεγγιστικό αλγόριθμο:

$$\min\{SOL_1, SOL_2\} \leq \frac{5}{3}OPT_{s,t}$$

Το πρόβλημα **Metric TSP**_{(s,t)-path} (συν.)

$\frac{5}{3}$ -προσεγγιστικός αλγόριθμος για το **Metric TSP**_{(s,t)-path}

1. Βρες ελάχιστο συνδετικό δέντρο T στον γράφο G . Διπλασίασε τις ακμές του T . Αφαίρεσε (s,t) -path από το διπλασιασμένο δένδρο. Βρες (s,t) -Euler path $P_{s,t}$, εκτέλεσε short-cutting για να βρείς (s,t) -Hamilton path κόστους:

$$SOL_1 \leq 2OPT_{s,t} - c_{s,t}$$

2. Με μικρή παραλλαγή του αλγόριθμου του Χριστοφίδη (Eulerian completion ώστε να προκύπτει (s,t) -Euler path με short-cutting), βρες (s,t) -Hamilton path με κόστος

$$SOL_2 \leq (3OPT_{s,t} + c_{s,t})/2$$

3. Επίστρεψε $SOL = \min(SOL_1, SOL_2)$

Ορισμός του προβλήματος

Δίνεται: γράφος $G(V, E)$ με μη αρνητικά βάρη στις ακμές του οι κόμβοι του οποίου χωρίζονται σε δύο σύνολα: **απαραίτητοι** και **Steiner**.

Ζητείται: δέντρο ελαχίστου κόστους που να περιέχει όλους τους απαραίτητους κόμβους.

Το Πρόβλημα **Metric Steiner Tree**

Επιπλέον υπόθεση: ο γράφος είναι πλήρης και τα δοθέντα βάρη ικανοποιούν την τριγωνική ανισότητα.

Ισοδυναμία προσεγγισιμότητας **Steiner Tree** και **Metric Steiner Tree**

Θεώρημα

*Δοθέντος ρ -προσεγγιστικού αλγορίθμου για το **Metric Steiner Tree** μπορούμε να κατασκευάσουμε ρ -προσεγγιστικό αλγόριθμο για το **Steiner Tree**.*

Απόδειξη: με αναγωγή διατήρησης του παράγοντα προσέγγισης από το **Steiner Tree** στο **Metric Steiner Tree**.

- Συμπλήρωση του αρχικού γράφου G σε πλήρη γράφο G' . Οι ακμές του G' έχουν το βάρος των συντομότερων μονοπατιών στον G (*metric closure*). Οι απαραίτητοι κόμβοι είναι ίδιοι.

Steiner Tree $\leq_{\text{apx.pres.}}$ Metric Steiner Tree

- ▶ Συμπλήρωση του αρχικού γράφου G σε πλήρη γράφο G' . Οι ακμές του G' έχουν το βάρος των συντομότερων μονοπατιών στον G (*metric closure*). Οι απαραίτητοι κόμβοι είναι ίδιοι.
- ▶ $OPT(I') \leq OPT(I)$ (γιατί;)

Steiner Tree $\leq_{\text{apx.pres.}}$ Metric Steiner Tree

- ▶ Συμπλήρωση του αρχικού γράφου G σε πλήρη γράφο G' . Οι ακμές του G' έχουν το βάρος των συντομότερων μονοπατιών στον G (*metric closure*). Οι απαραίτητοι κόμβοι είναι ίδιοι.
- ▶ $OPT(I') \leq OPT(I)$ (γιατί;)
- ▶ Κάθε λύση του I' με κόστος $SOL(I')$ δίνει λύση του I με κόστος $SOL(I) \leq SOL(I')$ (πώς;).

- ▶ Συμπλήρωση του αρχικού γράφου G σε πλήρη γράφο G' . Οι ακμές του G' έχουν το βάρος των συντομότερων μονοπατιών στον G (*metric closure*). Οι απαραίτητοι κόμβοι είναι ίδιοι.
- ▶ $OPT(I') \leq OPT(I)$ (γιατί;)
- ▶ Κάθε λύση του I' με κόστος $SOL(I')$ δίνει λύση του I με κόστος $SOL(I) \leq SOL(I')$ (πώς;).
Επομένως:

$$SOL(I) \leq SOL(I') \leq \rho OPT(I') \leq \rho OPT(I)$$

Σημείωση: Ισχύει επιπλέον ότι $OPT(I) = OPT(I')$, αλλά δεν το χρειαζόμαστε.

Αναγωγές Διατήρησης Παράγοντα Προσέγγισης (Approximation Factor Preserving Reductions)

Μια *αναγωγή διατήρησης παράγοντα προσέγγισης* από το πρόβλημα ελαχιστοποίησης Π στο πρόβλημα ελαχιστοποίησης Π' είναι ένα ζεύγος συναρτήσεων πολυωνυμικού χρόνου h, g , όπου η h απεικονίζει κάθε στιγμίοτυπο I του Π σε κάποιο στιγμίοτυπο $I' = h(I)$ του Π' και η g απεικονίζει λύσεις του I' σε λύσεις του I , ώστε:

- ▶ $OPT(I') \leq OPT(I)$
- ▶ για κάθε λύση S' του I' με κόστος $SOL(I', S')$ η $S = g(S')$ είναι λύση του I με κόστος $SOL(I, S) \leq SOL(I', S')$.

Θεώρημα

Μια αναγωγή διατήρησης παράγοντα προσέγγισης από το πρόβλημα Π στο πρόβλημα Π' μαζί με έναν ρ -προσεγγιστικό αλγόριθμο για το Π' δίνουν έναν ρ -προσεγγιστικό αλγόριθμο για το Π .

2-προσεγγιστικός αλγόριθμος για το **Metric Steiner Tree** (και το **Steiner Tree**)

Αλγόριθμος

Βρες και επίστρεψε ελάχιστο συνδετικό δένδρο στον παραγόμενο (node-induced) υπογράφο των απαραίτητων κόμβων (R).

2-προσεγγιστικός αλγόριθμος για το **Metric Steiner Tree** (και το **Steiner Tree**)

Αλγόριθμος

Βρες και επίστρεψε ελάχιστο συνδετικό δένδρο στον παραγόμενο (node-induced) υπογράφο των απαραίτητων κόμβων (R).

Απόδειξη λόγου προσέγγισης: η λύση είναι εφικτή για το **Metric Steiner Tree**, για προσεγγισιμότητα παίρνουμε κάτω φράγμα στο OPT με χρήση short-cutting:

Από οποιαδήποτε λύση για το **Metric Steiner Tree** μπορούμε να κατασκευάσουμε συνδετικό δέντρο στους κόμβους του R μόνο, διπλάσιου το πολύ κόστους (short-cutting στον αντίστοιχο κύκλο Euler). Επομένως και στη βέλτιστη λύση αντιστοιχεί ένα συνδετικό δέντρο T_R^* , με κόστος το πολύ $2 \cdot OPT$. Αρα:

$$cost(MST_R) \leq cost(T_R^*) \leq 2 \cdot OPT$$

Είναι γενικεύσεις του προβλήματος min-cut (θυμηθείτε το θεώρημα max-flow / min-cut):

Στο πρόβλημα **Multiway Cut** δίνεται γράφος G και k κόμβοι (terminals). Ζητείται η ελάχιστη τομή τέτοια ώστε κάθε terminal να βρίσκεται σε διαφορετική συνεκτική συνιστώσα (connected component). **NP**-hard, ακόμη και για fixed $k \geq 3$. Για $k = 2$;

Multiway Cut και Minimum k -Cut

Είναι γενικεύσεις του προβλήματος min-cut (θυμηθείτε το θεώρημα max-flow / min-cut):

Στο πρόβλημα **Multiway Cut** δίνεται γράφος G και k κόμβοι (terminals). Ζητείται η ελάχιστη τομή τέτοια ώστε κάθε terminal να βρίσκεται σε διαφορετική συνεκτική συνιστώσα (connected component). **NP-hard**, ακόμη και για fixed $k \geq 3$. Για $k = 2$;

Στο πρόβλημα **Minimum k -Cut** δίνεται γράφος G και ζητείται η ελάχιστη τομή τέτοια ώστε ο γράφος να διασπάται σε k συνεκτικές συνιστώσες. Πολυωνυμικά επιλύσιμο για fixed k , **NP-hard** για k που είναι μέρος της εισόδου.

Αλγόριθμος για το **Multiway Cut**

- Βρες isolating cut C_i για κάθε terminal s_i
- Επίστρεψε την ένωση των isolating cuts, εκτός από την βαρύτερη

Ορθότητα και λόγος προσέγγισης: Έστω A η βέλτιστη λύση, χωρίζοντας τον γράφο σε συνιστώσες $V_1, \dots, V_k$. Αν A_i είναι η τομή/υποσύνολο της A που χωρίζει το V_i από τις υπόλοιπες συνιστώσες, τότε είναι isolating cut για το s_i .

Άρα, $w(C_i) \leq w(A_i)$.

Κάθε edge της A , περιλαμβάνεται σε **δύο τομές A_i, A_j** , οπότε

$$\sum_{i=1}^k w(A_i) = 2w(A) = 2OPT$$

Επομένως:

$$w(C) \leq (1 - 1/k) \sum_{i=1}^k w(C_i) \leq (1 - 1/k) \sum_{i=1}^k w(A_i) = 2(1 - 1/k)OPT$$

Αλγόριθμος GH- k -cut

1. Compute Gomory-Hu tree T .
2. Output the union of cuts corresponding to lightest $k-1$ edges of T .

Με απόδειξη παρόμοια (κάπως πιο δύσκολη) αυτής για το multiway cut αποδεικνύεται ότι η λύση είναι $\leq 2(1 - 1/k)OPT$.

Δένδρο Gomory-Hu

Σημαντικές ιδιότητες: κάθε ακμή του αντιστοιχεί σε μια ελάχιστη τομή στον γράφο, και για κάθε ζεύγος (u, v) υπάρχει μια ακμή του δένδρου που αντιστοιχεί στην **ελάχιστη (u, v) -cut**

Κατασκευή δένδρου Gomory-Hu

1. construct tree T with unique vertex $S_0 = V$
2. while $\exists S_i \in T: |S_i| \geq 2$ do
 - choose two vertices x, y in S_i
 - compute minimum $x-y$ cut $C_{x,y}$ in G'
 - /* $G' = G$ with subtrees of S_i in T collapsed */
 - split S_i accordingly to S_i^x, S_i^y
 - add edge (S_i^x, S_i^y) to T
 - $w(S_i^x, S_i^y) \leftarrow w(C_{x,y})$ /* weight of edge (S_i^x, S_i^y) */
 - attach each subtree of S_i in T to S_i^x or S_i^y according to the cut