Cursul 4

Demonstrații și Justificări

Lema 1. Slide 15
Fie $G=(V,E)$ un graf neorientat și OPT cardinalul unei acoperiri de grad minim a lui G. Fie $E^*\subset E$ o mulțime de muchii nod disjuncte.
Atunci avem că **OPT≥|E*|**

Demonstrație:

Lema 1. Slide 15
Fie G=(V,E) un graf neorientat și OPT cardinalul unei acoperiri de grad minim a lui G. Fie $E^* \subset E$ o mulțime de muchii nod disjuncte.
Atunci avem că **OPT≥|E*|**

Demonstrație:

Fie **S** - o acoperire pentru G.

Lema 1. Slide 15
Fie $G=(V,E)$ un graf neorientat și OPT cardinalul unei acoperiri de grad minim a lui G. Fie $E^*\subset E$ o mulțime de muchii nod disjuncte.
Atunci avem că **OPT≥|E*|**

Demonstrație:

Fie **S** - o acoperire pentru G.
Deoarece E* este o multime de muchii nod disjuncte, înseamna ca orice nod din S poate acoperi cel mult o singura muchie din E*.

Lema 1. Slide 15
Fie G=(V,E) un graf neorientat și OPT cardinalul unei acoperiri de grad minim a lui G. Fie $E^*\subset E$ o mulțime de muchii nod disjuncte.
Atunci avem că **OPT≥|E*|**

Demonstrație:

Fie **S** - o acoperire pentru G.
Deoarece E* este o multime de muchii nod disjuncte, înseamna ca orice nod din S poate acoperi cel mult o singura muchie din E*.

Deci |S|>= |E*| pentru orice acoperire S (inclusiv cea optima) și orice multime de muchii nod disjuncte E*.

Lema 1. Slide 15
Fie G=(V,E) un graf neorientat și OPT cardinalul unei acoperiri de grad minim a lui G. Fie $E^* \subset E$ o mulțime de muchii nod disjuncte.
Atunci avem că **OPT≥|E*|**

Demonstrație:

Fie **S** - o acoperire pentru G.
Deoarece E* este o multime de muchii nod disjuncte, înseamna ca orice nod din S poate acoperi cel mult o singura muchie din E*.

Deci |S|>= |E*| pentru orice acoperire S (inclusiv cea optima) și orice multime de muchii nod disjuncte E*.
OPT>=|E*|

Teorema 2 (slide 16):

algoritmul descris alături este un algoritm 2-aproximativ

Teorema 2 (slide 16):
algoritmul descris alături este un algoritm 2-aproximativ

Înainte de a demonstra teorema - facem o justificare rapida cum ca S-ul rezultat este o acoperire a grafului G.:

Teorema 2 (slide 16):
algoritmul descris alături este un algoritm 2-aproximativ

Înainte de a demonstra teorema - facem o justificare rapida cum ca S-ul rezultat este o acoperire a grafului G.:

Practic, la fiecare pas al algoritmului, E’ va fi mulțimea muchiilor care nu sunt acoperite de S în acel moment.

La sfârșitul algoritmului E’ va fi vida.

Teorema 2 (slide 16):
algoritmul descris alături este un algoritm 2-aproximativ

Înainte de a demonstra teorema - facem o justificare rapida cum ca S-ul rezultat este o acoperire a grafului G.:

Practic, la fiecare pas al algoritmului, E’ va fi mulțimea muchiilor care nu sunt acoperite de S în acel moment.

La sfârșitul algoritmului E’ va fi vida.

Deci toate muchiile grafului sunt acoperite de nodurile din S.

Teorema 2 (slide 16):

algoritmul descris alături este un algoritm 2-aproximativ

Să demonstrăm ca algoritmul este 2-aproximativ!

Teorema 2 (slide 16):
algoritmul descris alături este un algoritm 2-aproximativ

Fie E* - mulțimea de muchii selectate la linia a3 a algoritmului (aleg (x,y)∈E*;)

Cum este E*?

Teorema 2 (slide 16):

algoritmul descris alături este un algoritm 2-aproximativ

Fie E* - mulțimea de muchii selectate la linia a3 a algoritmului (aleg (x,y)∈E*;)

Cum este E*?

E* este o mulțime de muchii nod disjuncte.

Teorema 2 (slide 16):
algoritmul descris alături este un algoritm 2-aproximativ

Fie E* - mulțimea de muchii selectate la linia a3 a algoritmului (aleg (x,y)∈E*;)

Cum este E*?

E* este o mulțime de muchii nod disjuncte.

Deoarece daca aleg o muchie xy care va fi inclusa în E*, toate celelalte muchii cu un capăt in x sau y sunt eliminate, deci nu vor fi incluse niciodată in E*.

Teorema 2 (slide 16): algoritmul descris alături este un algoritm 2-aproximativ

Fie E* - mulțimea de muchii selectate la linia a3 a algoritmului (aleg (x,y)∈E*;)

Cum este E*?

E* este o mulțime de muchii nod disjuncte.

Deoarece daca aleg o muchie xy care va fi inclusa în E*, toate celelalte muchii cu un capăt in x sau y sunt eliminate, deci nu vor fi incluse niciodată in E*.

OPT>=|E*| (din lema 1)

Teorema 2 (slide 16):

algoritmul descris alături este un algoritm 2-aproximativ

Fie E* - mulțimea de muchii selectate la linia 3 a algoritmului (aleg (x,y)∈E*;)

Cum este E*?

E* este o mulțime de muchii nod disjuncte.

Deoarece daca aleg o muchie xy care va fi inclusa in E* toate celelalte muchii cu un capăt in x sau y sunt eliminate, deci nu vor fi incluse niciodată in E*.

OPT>=|E*| (din lema 1)

2OPT>=2|E*|

Teorema 2 (slide 16):

algoritmul descris alături este un algoritm 2-aproximativ

Fie E* - mulțimea de muchii selectate la linia 3 a algoritmului (aleg (x,y)∈E*;)

Cum este E*?

E* este o mulțime de muchii nod disjuncte.

Deoarece daca aleg o muchie xy care va fi inclusa in E* toate celelalte muchii cu un capăt in x sau y sunt eliminate, deci nu vor fi incluse niciodată in E*.

OPT>=|E*| (din lema 1)

2OPT>=2|E*|=|S|

deci algoritmul este 2-aproximativ!

Discutie libera.

Daca un asemenea algoritm simplu este 2-aproximativ, inseamna ca se poate mai bine, nu?

Discutie libera.

Daca un asemenea algoritm simplu este 2-aproximativ, inseamna ca se poate mai bine, nu?

Raspuns:

Nu s-a gasit niciun algoritm cu un factor de aproximare <(2-epsilon)

S-a demonstrat ca orice algoritm aproximativ cu un factor de aproximare <1.4 nu poate rula in timp polinomial.

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \ldots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.

Slide 23:

Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.

Trebuie să minimizăm funcția:

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.

Trebuie să minimizăm funcția:

$$\sum_{1 \le i \le n} f(v_i) * x_i$$

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \ldots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.

Trebuie să minimizăm funcția:

$$\sum_{1 \leq i \leq n} f(v_i) * x_i$$

Constrângerile arată în felul următor: pentru fiecare muchie (v_i, v_j) trebuie să avem:

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \ldots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.

Trebuie să minimizăm funcția:

$$\sum_{1 \le i \le n} f(v_i) * x_i$$

Constrângerile arată în felul următor: pentru fiecare muchie (v_i, v_j) trebuie să avem:

$$x_i + x_j \geq 1, \forall\, i, j \in \{1, ..n\}$$
$$0 \leq x_i \leq 1, \forall\, i \in \{1, ..n\}$$

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

Ce se întâmpla cu nodul v_3 dacă $x_3 = ⅓$? In acoperirea S intra doar o treime din v_3?!

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

Ce se întâmpla cu nodul v_3 dacă $x_3 = ⅓$? In acoperirea S intra doar o treime din v_3?!

Probleme de tipul 1/0 linear programming (integer linear programing) - NU pot fi rezolvate cu algoritmi simplex, și sunt ele însele probleme NP-hard.

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \ldots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

Aproximăm soluția.

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

Aproximăm soluția.
Cum trebuie să fie x_i astfel încât v_i sa fie în acoperirea S?

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

Aproximăm soluția.
Cum trebuie să fie x_i astfel încât v_i sa fie în acoperirea S?
Răspuns:dacă $x_i \geq \frac{1}{2}$ atunci adaug pe v_i in acoperire.

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

Aproximăm soluția.
Cum trebuie să fie x_i astfel încât v_i sa fie în acoperirea S?
Răspuns:dacă $x_i \geq \frac{1}{2}$ atunci adaug pe v_i în acoperire.

De demonstrat că o astfel de acoperire este o soluție 2-aproximativă pentru WVCP.

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \ldots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

De demonstrat ca o astfel de acoperire este o soluție 2-aproximativă pentru WVCP.

$$ALG =$$

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

De demonstrat ca o astfel de acoperire este o soluție 2-aproximativă pentru WVCP.

$$ALG = \sum_{1 \le i \le n} f(v_i) * \begin{cases} 1, dacă\ x_i \ge 1/2 \\ 0, dacă\ x_i < 1/2 \end{cases} \le$$

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

De demonstrat ca o astfel de acoperire este o soluție 2-aproximativă pentru WVCP.

$$ALG = \sum_{1 \le i \le n} f(v_i) * \begin{cases} 1, dacă\ x_i \ge 1/2 \\ 0, dacă\ x_i < 1/2 \end{cases} \le \sum_{1 \le i \le n} f(v_i) * 2x_i$$

$$=$$

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

De demonstrat ca o astfel de acoperire este o soluție 2-aproximativă pentru WVCP.

$$ALG = \sum_{1 \le i \le n} f(v_i) * \begin{cases} 1, dacă\ x_i \ge 1/2 \\ 0, dacă\ x_i < 1/2 \end{cases} \le \sum_{1 \le i \le n} f(v_i) * 2x_i$$
$$= 2 * \sum_{1 \le i \le n} f(v_i) * x_i \le$$

Slide 23:
Putem scrie problema de weighted Vertex cover ca o problema de programare liniara.

$X = \{x_1, x_2, x_3, \dots, x_n\}$ - cu ce proprietate?
$x_i = 1$ daca acoperirea aleasă conține nodul v_i, si $x_i = 0$ altfel.
Observație: Algoritmul simplex îmi va oferi numere reale pt variabilele din X.

De demonstrat ca o astfel de acoperire este o soluție 2-aproximativă pentru WVCP.

$$ALG = \sum_{1 \le i \le n} f(v_i) * \begin{cases} 1, dacă\ x_i \ge 1/2 \\ 0, dacă\ x_i < 1/2 \end{cases} \le \sum_{1 \le i \le n} f(v_i) * 2x_i$$
$$= 2 * \sum_{1 \le i \le n} f(v_i) * x_i \le 2OPT$$