נושא 6 - אלגוריתמי קירוב

א) הגדרה

אלגוריתמי קירוב הם אלגוריתמים המוציאים בזמן פולינומי פתרון לבעיה כך שהפתרון יחסית קרוב לפתרון האופטימלי של הבעיה. משתמשים בהם לבעיות השייכות ל-NP, שאיננו יודעים למצוא להם פתרון אופטימלי בזמן פולינומי. אלגוריתמי הקירוב נותנים להם פתרון קרוב יחסית לאופטימלי.

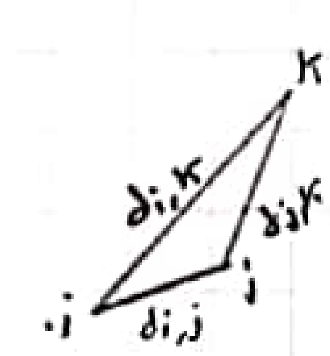
ב) יחס הקירוב

בהינתן בעיה Π ומופע של הבעיה I, נסמן את הפתרון האופטימלי ל-I כך $OPT(I)$.
אלגוריתם קירוב A עבור Π הוא אלגוריתם שעבור כל מופע I של Π יחזיר פתרון $A(I)$ בזמן פולינומי, כך שאם Π בע"מ מינימום יתקיים: $A(I)\leq\alpha\cdot OPT(I)$, $\forall I\in\Pi$,
ואם Π בע"מ מקסימום יתקיים: $OPT(I)\leq\alpha\cdot A(I)$ $\forall I\in\Pi$.
α נקרא "יחס הקירוב" של A. ככל ש-α יותר קרוב ל-1 כך A נחשב אלגוריתם קירוב טוב ומדויק יותר, מפני שהוא קרוב יותר לאופטימלי. ואילו ככל ש-α גדל מ-1 האלגוריתם קירוב פחות יעיל. בדרך כלל $1\leq\alpha\leq 2$.
נראה בהמשך הפרק מספר בעיות שיש להם אלגוריתמי קירוב, ובתוך כך את השיטות למציאת אלגוריתם קירוב.

ג) בע"ת K-CENTER

בהינתן n נקודות מידע, נגדיר מטריצת מרחק ריבועית ומכילה ערכים ממשיים, $D_{n\times n}\in M_{n\times n}(\mathbb{R})$, שתחזיק את המרחק בין כל שתי נקודות מידע. כלומר, כל איבר $d_{i,j}$ יכיל את המרחק מ-i ל-j. המטריצה תמיד שומרת על המטריקה הבאה:

- $\forall i,j\in[n]\ d_{i,j}\geq 0$ - מרחק תמיד חיובי.
- $\forall i\in[n]\ d_{i,i}=0$ - מרחק מנקודה לעצמה 0.
- $\forall i,j\in[n]\ d_{i,j}=d_{j,i}$ - D סימטרית.
- $\forall i,j,k\in[n]\ d_{i,j}+d_{j,k}\geq d_{i,k}$ - אי-שוויון המשולש. המרחק בין כל שתי נק' תמיד קטן מכל מסלול אחר בין הנק'.



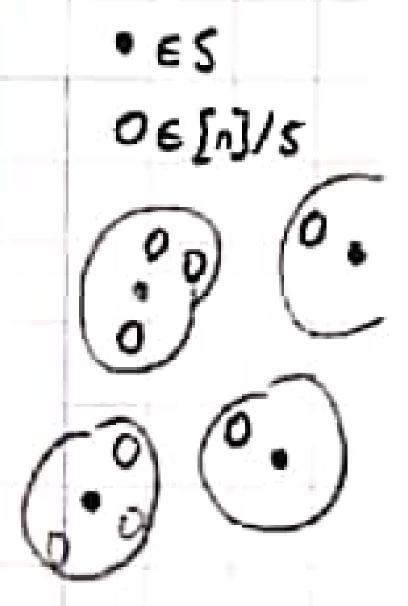
ניתן להציג מטריצה זו גם כגרף מלא K_n שמשקל הצלעות הם המרחקים. לכל קבוצה $S\subseteq[n]$ ולכל איבר $i\in[n]$, נגדיר את המרחק $d(i,S)$ להיות המרחק המינימלי בין i לאיבר (כלשהו) ב-S. $d(i,S)=\min_{j\in S} d_{i,j}$

<u>הגדרת הבעיה:</u> בהינתן מספר $k\in\mathbb{N}$, n נקודות מידע ומטריצת מרחק D כמו שתיארנו, נרצה למצוא קבוצת דמתים $S\subseteq[n]$ בגודל k ($|S|=k$), שעבורה $\max_{i\in[n]} d(i,S)$ מינימלי. כלומר, עבור כל תת-קבוצה של $[n]$ בגודל k, נגדיר את המרחק לנקודה הכי רחוקה מהתת-קבוצה. אנו מעוניינים למצוא את

11/5

בס"ד

התת-קבוצה שעבורה ערך זה מינימלי. הסיבה שנרצה למצוא S כזאת, היא שבאמצעותה נוכל לחלק את $[n]$ הנקודות ל-k קבוצות, כך שכל נקודה נמצאת בקבוצה של האיבר מ-S שהוא הכי קרוב אליו. כל איבר ב-S ישמש נקודת מרכז כובד (k-means) בקבוצה שלו.



חלוקה כזו מאוד שימושית להרבה בעיות מציאותיות. לדוגמה: n חברות דורשות שירות ויכולתנו להקים רק k מפעלים. אנו מעוניינים שהמפעלים יהיו כמה שיותר קרובים לחברות.

בעיות מסוג זה נקראות בעיות 'ניתוח אשכולות' (clustering). פונקציית המטרה כאן ($\max_{i\in[n]}(i,S)$) משתנה בין סוגי הבעיות ולכן לכל סוג בעיה אלגוריתם שונה.

<u>פתרון חמדני:</u> נמצא בכל איטרציה את הנקודה הכי רחוקה מ-S ונכניס אותה ל-S. בסוף נקבל קבוצה בגודל k.

```
K-CENTRE {
    initially S = ϕ
    set S={i} where i∈[n] chosen arbitrarily
    while (|S|<k) {
        j = max_{i∈[n]} d(i,S)   // הנק' הכי רחוקה מ-S
        S = S ∪ {j}
    }
    return S
}
```

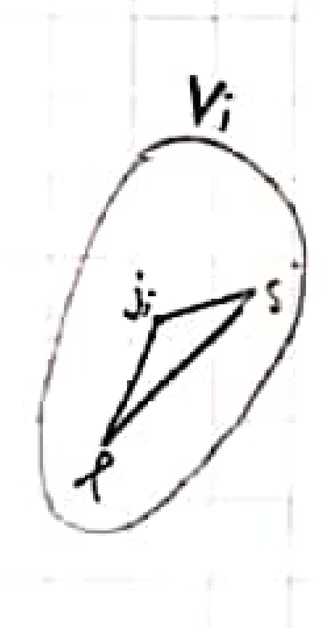
<u>הוכחת 2-קירוב:</u> נוכיח שאלגוריתם זה הוא 2-מקרב. כלומר שלכל מופע I של K-CENTRE. האלגוריתם החמדני יחזיר $S\subseteq[n]$, $|S|=k$, המקיים: $\max_{i\in[n]} d(i,S) \le 2\cdot OPT(I)$.

הוכחה - נגדיר $OPT=S^*=\{j_1,j_2,\dots,j_k\}$ פתרון אופטימלי לבעיה, ו-$r^*=\max_{i\in[n]} d(i,S^*)$ להיות המרחק המקסימלי מנקודה ב-S^* לנקודה אחרת כלשהי. S^* משרה חלוקה של $[n]$ ל-k קבוצות $V_1,V_2,\dots,V_k$ שהם הקלאסטרים כמו שהגדרנו לעיל. נשים לב כי עבור כל שני נקודות $x,y\in V_i$ כלשהן, המרחק ביניהם מקיים $d_{x,y}\le 2r^*$, וזאת לפי אי-שוויון המשולש $d_{x,y}\le d_{x,j_i}+d_{j_i,y}\le 2r^*$.

כעת נגדיר $S=\{s_1,s_2,\dots,s_k\}$ להיות הקבוצה שהוחזרה מהאלגוריתם החמדני, ו- $r=\max_{i\in[n]} d(i,S)$ להיות המרחק המקסימלי מנקודה ב-S לנקודה כלשהי אחרת. צריך להוכיח $r\le 2\cdot r^*$ כדי להוכיח 2-קירוב.

נחלק לשני מקרים אפשריים:

(1) אם בכל קלאסטר $V_1,V_2,\dots,V_k$ של הפתרון האופטימלי, יש בדיוק איבר אחד מ-S, כלומר $|S\cap V_i|=1,\ \forall i\in k$, אזי עבור כל נקודה $l\in[n]$ המרחק ממנה ל-S אכן קטן מ-$2r^*$, שהרי l חייבת להימצא בקלאסטר V_i כלשהו, שבו נקודה $s'\in S$, ומתקיים: $d_{l,s'}\le d_{l,j_i}+d_{j_i,s'}\le 2\cdot r^*$.



(2) אם לא בכל הקלאסטר יש איבר מ-S, אזי בהכרח ישנו קלאסטר V_i שבו שתי נקודות $x,y\in S$. נניח שהאלגוריתם החמדני בחר קודם את x ומתישהו לאחר מכן את y. אזי כאשר y נבחר המרחק המקסימלי מ-S כולה ל-y חייב להיות קטן מ-$2\cdot r^*$, שהרי לפי אבחנה שעשינו מתקיים $d_{x,y}\le 2r^*$. מכאן נוכל להסיק שכל הנקודות שנותרו מחוץ ל-S גם כן מרחקן מ-S קטן מ-$2\cdot r^*$, שהרי המרחק ל-y היה מקסימלי בעת בחירתו. מש"ל.

Scanned with CamScanner

k-suppliers

דומה לבעיית K-CENTRE גם כן מקבלת [n] נקודות מידע, מטריקת מרחקים D על
אותה מטריקה, ומספר $k\in\mathbb{N}$. אבל שכן [n] הנקודות מחולקות לשתי קבוצות $A\cup B=[n]$.
כאשר A היא קבוצת ספקים ו-B קבוצת לקוחות. המטרה היא למצוא קבוצת נקודות
$S\subseteq A$ בגודל k שעבורה $\max_{b\in B} d(b,S)$ מינימלי. כלומר למצוא קבוצה בגודל k מתוך A
שהמרחק המקסימלי ממנה ל-b הוא הקטן ביותר מכל קבוצה אחרת ב-A בגודל k.
נסמן לכל $b\in B$ את a_b להיות הנקודה ב-A שהכי קרובה ל-b.

```
K-SUPPLIER (A∪B, D, k) {
    apply greedy algorithem from K-CENTRE on B only and set S' the result
    ...
    return S = {a_b : b∈S'}
}
```

האלגוריתם מוצא את k הנקודות ב-B שהכי רחוקות לשאר הנקודות ב-B. לאחר מכן לוקח מ-A
את k הנקודות שהכי קרובות ל-k הנקודות שמצאנו ב-B.

<u>הוכחת 3-קירוב:</u> נסמן ב-r^* את המרחק המקסימלי שלה קבוצה האופטימלית מנקודה ב-B.

צריך להוכיח: $\max_{b\in B} d(b,S)\le 3\cdot r^*$. נשקול שני מקרים.

- אם לכל $b\in B$ יש לקוח $b'\in S'$ במרחק לכל היותר $2\cdot r^*$ ממנו, אז לפי אי-שוויון המשולש:

$$\max_{b\in B} d(b,S) \le d(b,s')+d(s',S) \le 2\cdot r^*+r^* = 3\cdot r^*$$

- אם יש $b\in B$ כך שהמרחק ממנה ל-S' גדול מ-$2\cdot r^*$, כלומר $d(b,S')\ge 2\cdot r^*$, אז כיוון שהאלגוריתם התמיד בוחר את הנקודה ב-B הרחוקה ביותר, משמע שאם כל הנקודות ב-S' רחוקות אחת מהשנייה ביותר מ-$2\cdot r^*$. לכן ב-$S'\cup\{b\}$ יש $k+1$ נקודות הרחוקות מהשנייה מרחק גדול מ-$2\cdot r^*$.
אמנם אז לא יכול להיות שתהיה נקודה $a\in A$ כך ש-a תהיה במרחק r^* או פחות משתי
נקודות ב-$S'\cup\{b\}$. לכן צריך לפחות $k+1$ נקודות ב-A כדי לקבל קירוב מקסימלית של r^*. סתירה!

לכן רק המצב הראשון אפשרי שכבר הוכחנו 3-מקרב.

3) METRIC-TSP

נתונים n ערים ומטריצת מרחקים $C_{n\times n}\in M_{n\times n}(\mathbb{R})$ סימטרית ריבועית, עם אותה המטריקה כמו בבעיית K-CENTRE. נסתכל על מטריצה זו כגרף מלא ממושקל K_n, שבו עלות הקשת בין שני ערים שווה לעלות המרחק ביניהם לפי מטריצת המרחקים.* הבעיה היא שבהינתן גרף G כזה למצוא בו מעגל המילטוני מינימלי, כלומר מעגל שעובר בכל הצמתים ועלותו מינימלית. נציג 3 אלגוריתמים בסעיפים הבאים המציגים אלגוריתם קירוב לבע"ת METRIC-TSP.

*פונקציית המשקל $c:[n]\to\mathbb{R}$

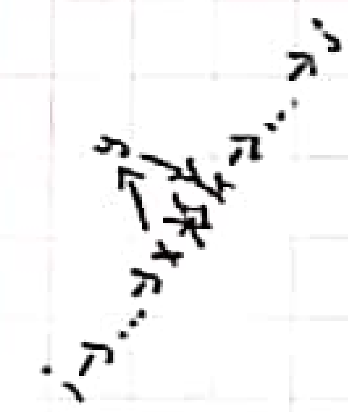
ה) Nearest Extension

נשים לב שבמעגל המילטון אם מחסירים ממנו קשת כלשהי, נקבל עץ פורש. לכן תמיד עלות מעגל המילטון קטנה מ- MSP (עץ פורש מינימלי).* לפי הנחה זו נציג את האלגוריתם החמדני

*לא הורדנו דווקא ... בהכרח נקבל MSP

```
Nearest-Extension(graph G) {
    let i,j ∈ [n] be the closest cities in G
    set T = ({i,j}, (i,j))   // T הוא תת-גרף המהווה מעגל ב-G
    while (V(T) ≠ V(G)) {
        let x,y = min{c_{x,y} : x ∈ V(T) and y ∉ V(T)}
        let k ∈ V(T) be the vertex after x in T
        set T = (V(T) ∪ {y}, E(T) - (x,k) + (x,y) + (y,k))
    }
    E(T) + (j,i)
    return E(T)
}
```

הסבר: האלגוריתם דומה לאלגוריתם פרים למציאת MSP, הוא מוצא בכל איטרציה את החתך המינימלי, הקשת המינימלית שיוצאת מצומת ב-T אל צומת שאינה ב-T, ומכניס אותה במקום לצומת ב-T.*



בסוף מוסיף את הקשת (j,i) כדי לסגור את המעגל, שהרי התחיל מ- (i,j).

הוכחת 2-קירוב: כאמור לעיל האלגוריתם עושה את אותה העבודה כמו אלגוריתם פרים למציאת MSP. נסמן ב-F' את כל הקשתות (x,y) שנבחרו בלולאה שהם חתך מינימלי. יש n-2 קשתות כאלו. F' ביחד עם (i,j) שנבחרה ראשונה מהווה MSP, שהרי זה מה שפרים היה בוחר. נסמן $F=F'+(i,j)=MSP$. נתחיל בלמצוא חסם עליון לעלייה בעלות של $E(T)$ בכל איטרציה. האלגוריתם בכל איטרציה מוצא מוחק על קשת ב-$E(T)$, הוא מוריד אותה ומכניס במקומה שתיים אחרות. תהי $(u,v)\in E(T)$. נניח כי באיטרציה i הקשת המינימלית שחוצה את החתך היא (u,w). נשים לב $(u,w)\in F'$. באיטרציה i השינוי בעלות המשקל הכולל של $E(T)$ הוא: $c_{u,w}+c_{w,v}-c_{u,v}$. לפי אי-שוויון המשולש $c_{w,v}\le c_{w,u}+c_{u,v}$. לכן נוכל לחסום את השינוי בעלות בדרך הבאה: $c_{u,w}+c_{w,v}-c_{u,v}\le c_{u,w}+c_{w,u}+c_{u,v}-c_{u,v}=2\cdot c_{u,w}$.

כעת נוכל לחסום את העלות של $E(T)$

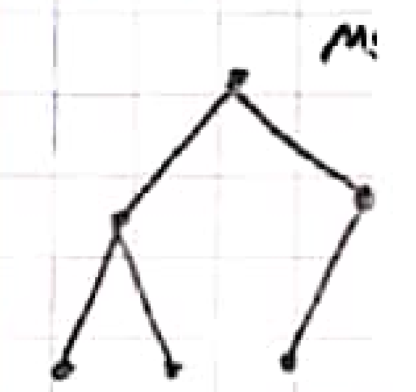
$$cost(E(T)) = cost\left(\text{כל המעקפים על קשת ב-}F'\right)+2\cdot c_{i,j} \underset{\text{לפי מסקנה אחרונה}}{\le} \sum_{f\in F'} 2\cdot c(f)+2\cdot c_{i,j} \underset{F=F'+(i,j)}{=} 2\cdot\sum_{f\in F} c(f) \underset{F=MSP}{=} 2\cdot cost(MST) \underset{\text{מסקנה ראשונה מתחילת הסעיף}}{\le} 2\cdot OPT$$

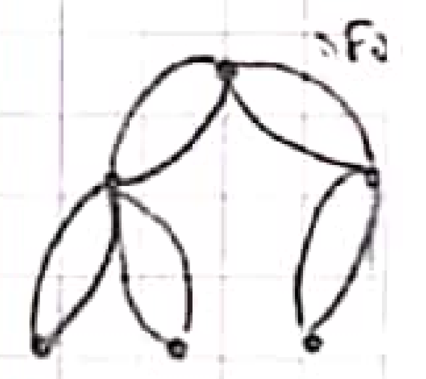
מ.ש.ל

18/5 אסף

1) Doubling Trees

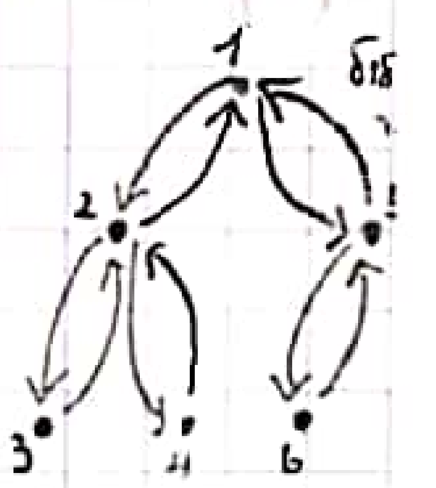
פתרון זה ל-METRIC-TSP מתבסס על משפט בקשר למסלול אוילר. מסלול אוילר זהו מסלול שעובר בכל הקשתות בגרף פעם אחת בדיוק. הוא יכול לעבור באותה צומת מספר פעמים. אנו מתעניינים במשפט הבא: במולטי-גרף קשיר (גרף שבין כל שני צמתים יכולים להיות מספר קשתות) יש מסלול אוילר אם ורק אם דרגת כל הצמתים זוגית. מאשפט זה נובע שאם T הוא עץ, אזי אם נכפיל כל צלע בגרף, כך שאם בין שני צמתים הייתה קשת כעת יש ביניהם שתי קשתות, אזי ב-T יש מסלול אוילר.





על בסיס עקרון זה נבנה את האלגוריתם הבא:

```
Doubling-Trees(graph G) {
    Find MST T of G
    double every edge of T. let DT be the result multigraph
    Find Euler tour E in DT
    construct TSP tour C from E by arranging vertices according
    to the order of their first appearance along E   // נבחר רק את הפעם הראשונה שכל צומת מופיע ב-E. נעבור ב-G שגרף מלא
    add to the end of C its first node
    return C
}
```



$1,2,3,2,4,2,1,5,6$

TSP: 1,2,3,4,5,6,1

הוכחת 2-קירוב: נשים לב כי $c(DT)=2\cdot c(T)$ שהרי DT מכפיל כל צלע ב-T. בנוסף, הפתרון C המוחזר מהאלגוריתם מקיים $c(C)\le c(DT)$, שהרי C רק מוצא קיצורי דרך ב-DT. כלומר, הוא לוקח מסלולים של מספר קשתות ב-DT, ומחליף אותם בקשת אחת. לפי אי-שוויון המשולש משקלה של קשת זו בוודאי יותר קטן מהמסלול שהחליפה. מכאן נקבל:

$c(C)\le c(DT)=2\cdot c(T)=2\cdot c(MST)\le 2\cdot OPT$. שהרי T הוא MST, וכפי הסברנו ש-OPT קטן מ-MST.

ב) Christofides

אלגוריתם קירוב נוסף ל-METRIC-TSP. אלגוריתם זה דומה ל-Doubling Trees, אלא שטוען שהכפלת MST כולו פעולה בזבזנית כדי ליצור מסלול אוילר, ויש פעולה טובה יותר לשם כך. השיטה מתבססת על המשפט הבא: בכל גרף מספר הצמתים עם דרגה אי-זוגית הוא זוגי. לכן נוכל ליצור גרף שכל דרגותיו זוגיים אם נוסיף לגרף שידוך מושלם של כל הצמתים האי-זוגיים, כלומר נוסיף לכל צומת שדרגתו אי-זוגית קשת ובכך נקבל שדרגות כל הצמתים זוגיות. יש שידוך מושלם על הצמתים שדרגתם אי-זוגית שכן הגרף מלא ומספרם זוגי. בגרף (או מולטי-גרף) שהתקבל כל הדרגות זוגיות ולכן יש בו מסלול אוילר.

```
Christofides(graph G) {
    Find MST T of G
    let O be all vertices in T having odd degree
    let M be minimum cost perfect matching in G[O]   // G[O] גרף מושרה המכיל רק צמתים ב-O.
    add M to T to attain MT (may be multigraph)
    Find Euler tour E in MT
    construct a TSP tour C from E by arranging vertices according
    to the order of their first appearance along E   // נבחר רק את הפעם הראשונה שכל צומת מופיע ב-E. נעבור ב-G שגרף מלא
    add to the end of C its first node
    return C
}
```

הוכחת $\frac{3}{2}$ קירוב: נוכיח קודם משפט עזר:

משפט עזר: יהי $W\subseteq V(G)$ קבוצת צמתים בגודל זוגי. ויהי M שידוך מושלם בעלות מינימלית על $G[W]$. אזי העלות הכוללת של M קטנה בחצי מעלות הפתרון האופטימלי של METRIC-TSP על G, כלומר $c(M)\le\frac{OPT}{2}$.

הוכחה: נסמן ב-H את המסלול של הפתרון האופטימלי העובר בכל הצמתים בעלות מינימלית, $c(H)=OPT$. עוד נסמן ב-H' את המסלול שעובר רק בצמתים מ-W שהושג ע״י קיצורי דרך ב-H בין כל שני צמתים מ-W. כלומר ניקח מ-H את סדר הצמתים וראשונות בהם מופיעה צמתים מ-W, ונוסיף ביניהם קשתות. נזכור ש-G גרף מלא. מכיוון שקיצורי דרך לא מוסיפים עלות הכוללת כתוצאה מאי-שוויון המשולש, מתקיים: $c(H')\le c(H)$. נשים לב שגם H' הוא איחוד שני שידוכים מושלמים על W. ומכאן:

$$2\cdot c(M)\underset{\text{מינימליות של M}}{\le} c(H')\le c(H)=OPT \Rightarrow c(M)\le\frac{OPT}{2}$$

נחזור להוכחה: נסמן ב-C את המסלול שהוחזר מהאלגוריתם. עלות מסלול זה קטנה מעלות MT, שהרי היא מכילה רק קיצורי דרך מ-MT. בנוסף, נזכור שעלות MST קטנה מעלות OPT.

$$c(C)\le c(MT)\underset{\text{MT איחוד זרים}}{=}c(T)+c(M)=c(MST)+c(M)\underset{\text{משפט עזר ו-}MST\le OPT}{\le}OPT+\frac{OPT}{2}=\frac{3}{2}OPT$$ מ.ש.ל.

STEINER-TREES

נתון גרף G ממושקל חיובי $c:E(G)\to\mathbb{R}_{\ge0}$, וחלוקה של הצמתים לשתי קבוצות $V(G)=R\cup S$, כאשר R הם הצמתים המבוקשים ו-S נקראים צמתי Steiner. המטרה היא למצוא עץ שמכיל את כל הצמתים ב-R עם משקל מינימלי.

Metric Steiner-Trees - אותה בעיה עם גרף מלא ופונקציית מחיר המקיימת מטריקה עם אי-שוויון המשולש.

<u>נוכיח רדוקציה מבעיית Steiner-Trees לגרסה המטרית:</u> כלומר בהינתן פתרון לגרסה המטרית נוכל למצוא פתרון עם אותו יחס לבעיה הרגילה. בהינתן מופע (G,c) לבעיית Steiner-Trees נגדיר G' גרף מלא הדומה ל-G עם השלמת הקשתות החסרות ופונקציית משקלים $c':E(G')\to\mathbb{R}_{\ge0}$ הנותנת לכל קשת (u,v) את משקל המסלול הכי נמוך ב-G. בהינתן T' פתרון ל-G', נגדיר פתרון T ל-G ע״י החלפת כל קשת ב-T' במסלול עם משקל הכי נמוך ב-G. נדאג שב-T לא יהיו מעגלים ע״י הורדת צלעות. קיבלנו עץ T שעדיין פורש את כל R גם עם אותו משקל או פחות ממנו, שכן יכולים להיות ב-T צלעות כפולות או מעגלים שגרמו לכך שהעלות תקטן.

מסקנה: $cost(T)\le cost(T')$. נראה אלגוריתם ל-Metric Steiner Trees שהוא 2-מקרב.*

לכן גם עבור Steiner-Trees זה 2 מקרב:

```
Metric Steiner-Trees(G,c) {
    return MST of G[R]   // עץ פורש מינימלי לגרף המושרה G[R]
}
```

$c(MST)\le c(H)\le c(\mathcal{E})=2\cdot OPT$ מ.ש.ל.

<u>הוכחת 2 קירוב:</u> הפוך ל-Doubling-Trees, נכפיל את OPT כדי לקבל מסלול אוילר $\mathcal{E}$, עם קיצורי דרך נקבל מעגל המילטון H. H הוא עץ פורש, לכן:

ח) MINIMUM SET-COVER

מופע של בעיית min set-cover חום מורכב מ- (Ω,ש). כאשר Ω היא קבוצה המכילה n איברים, (הגיית c)
ו- ש מורכבת מתת' קבוצות של Ω, $P(\Omega) \subseteq$ ש. תנאי חשוב שחייב להתקיים הוא שאיחוד כל תת'
הקבוצות ב- ש מרכיבים ביחד את Ω. כלומר $\bigcup_{c \in ש} c = \Omega$. יש שני סוגי בעיות לכל מופע:

(1) למצוא מתוך ש את כל תת' הקבוצות שאיחודם שווה ל- Ω וגם מספר הקבוצות שנבחרו מינימלי.
כלומר למצוא $T \subseteq ש$ כך ש- $\bigcup_{T \in T} T = \Omega$ וגם $|T|$ מינימלי. בעיה זו נקראת: min cardinality set-cover חום

(2) נגדיר לכל תת-קבוצה ב- ש משקל חיובי. אנו מעוניינים למצוא את כל תת' הקבוצות ב-ש שאיחודם
שווה ל- Ω וגם משקלם הכולל מינימלי. כלומר למצוא $T \subseteq ש$ כך ש $\bigcup_{T \in T} T = \Omega$ וגם $\sum_{T \in T} c(T)$ מינימלי.
בעיה זו נקראת min cost set-cover חים.

<u>נוכיח ש- min set-cover ∈ NPC חום:</u> ע"י רדוקציה מ- vertex-cover חום. עבור כל מופע של vc חום, שהוא
גרף רגיל $G=(V,E)$, נגדיר $\Omega = E(G)$ ו- $ש = \{C_u, u \in V(G)\}$ כאשר $C_u = \{vu, vu \in E(G)\}$. כלומר ש מכילה
עבור כל צומת ב- G את קבוצת הקשתות שיוצאות ממנה. נשים לב שאכן ש מכיל את כל הקשתות
בגרף ולכן התנאי מתקיים.

נוכל להסתכל על כל מופע של sc חום (ש,Ω) כגרף דו-צדדי שבו בצד אחד $E(G)$,
ובצד השני $V(G)$. נמתח קשת בין $v_i \in V(G)$ ל- $e_i \in E(G)$ אם ורק אם e_i הקשת שיוצאת מ- v_i.
אנו מחפשים תת-קבוצה של $V(G)$ הכי קטנה שתיתן לנו כיסוי של כל הקשתות $E(G)$ בגרף הדו-צדדי.
זה בדיוק מה שיתן לנו בעיית vc חום. ולכן הפוך, פתרון ל- sc חום עבור הגרף הדו-צדדי יהיו גם
פתרון ל- vc חום. לכן min sc ∈ NPC חום.

נראה אלגוריתם קירוב חמדני ל- sc min cardinality חום ונציג שני אלגוריתמי קירוב ל- cost sc חום.

ט) Min cardinality set-cover

אלגוריתם קירוב חמדני לפתרון הבעיה:

```
Min-cardinality-set-cover חום (Ω, ש) {
    A = ∅          // A קבוצה של תת'-קבוצות מ-ש
    while (∪A ≠ Ω) {
        c = max{ |(∪A)∪c| : c∈ש }
        A = A ∪ {c}
    }
    return A
}
```

האלגוריתם בוחר בכל איטרציה את התת קבוצה ב-ש $c \in$
שאם נוסיפה ל- A יתווספו ל- $\cup A$* הכי הרבה איברים
מ- Ω. כלומר c היא התת קבוצה שהכי מכסה את $\Omega \setminus \cup A$.

$\cup A = \bigcup_{A \in A} A$
*איחוד כל תת'-הקבוצות ב- A.

נסמן ב- m את גודל התת-קבוצה הכי גדולה ב- ש. $m = \max\{|c| : c \in ש\}$.

<u>הוכחת $O(\ln(m))$ קירוב:</u> כלומר להוכיח שמתקיים לכל מופע של הבעיה: $|A| \le O(\ln(m)) \cdot OPT$.

נשים לב שמספר התת קבוצות ב-A הוא בדיוק מספר האיטרציות של האלגוריתם החמדני, לכן נרצה לחשב מהו מספר המקסימלי של איטרציות. לשם כך קודם נחשב מהו המספר מינימלי של איברים שמתווספים ל-$\cup A$ בכל איטרציה. נסמן ב-A_i את הקבוצה A בתחילת האיטרציה ה-i, נרצה לחשב $|\cup A_{i+1}|-|\cup A_i|$, כלומר מספר האיברים שנוספו באיטרציה ה-i.

טענת עזר: לכל $i\in\mathbb{N}$ מתקיים: $|\cup A_{i+1}|-|\cup A_i| \ge \frac{|\Omega\setminus\cup A_i|}{OPT}$.

הוכחה: נשים לב שכל האיברים החדשים ש-A מכסה לאחר האיטרציה ה-i לקוחים מ-$\Omega\setminus\cup A_i$. נתבונן בפתרון האופטימלי $OPT=\{C_1,C_2,\dots,C_{OPT}\}$ של הבעיה המכילה OPT קבוצות. ב-OPT חייבת להיות תת קבוצה $C_j\in$ המקיימת $|C_j\cap(\Omega\setminus\cup A_i)|\ge\frac{|\Omega\setminus\cup A_i|}{OPT}$, שהרי אם לא כן אזי כל תתי הקבוצות ב-OPT לא היו מכסות את Ω. כיוון שהאלגוריתם לוקח את התת-קבוצה $C\in$ שמכסה הכי הרבה איברים חדשים מ-$\Omega\setminus A_i$, הוא בהכרח ישקול את C_j. לכן מספר האיברים החדשים שיתווספו ל-A בעקבות איטרציה i הוא לפחות $\frac{|\Omega\setminus\cup A_i|}{OPT}$.

נסמן ב-U_i את כל האיברים שנותר לכסות לפני האיטרציה ה-i, $U_i:=\Omega\setminus\cup A_i$. נרצה לחשב את $|U_i|$.

טענת עזר: לכל $i\in\mathbb{N}$ מתקיים: $|U_i|\le e^{-\frac{i}{OPT}}\cdot n$.

הוכחה: $|U_{i+1}|=|\Omega\setminus\cup A_{i+1}| = |\Omega|-|\cup A_{i+1}| \underset{\text{מטענת עזר}}{\le} |\Omega| - \frac{|\Omega\setminus\cup A_i|}{OPT} - |\cup A_i| = |\Omega| - \frac{|\Omega|}{OPT} + \frac{|\cup A_i|}{OPT} - |\cup A_i| =$

$\underset{\text{הוצאת איברים משותפים}}{=} (1-\frac{1}{OPT})|\Omega| - (1-\frac{1}{OPT})|\cup A_i| = (1-\frac{1}{OPT})|\Omega\setminus\cup A_i| = (1-\frac{1}{OPT})|U_i|$

קיבלנו נוסחה רקורסיבית: $|U_i|\le(1-\frac{1}{OPT})|U_{i-1}|$. נשתמש ברקורסיה ובעובדה ש-$n=|\Omega|=|U_0|$

$$|U_i|\le(1-\tfrac{1}{OPT})|U_{i-1}|\le(1-\tfrac{1}{OPT})^2|U_{i-2}|\le\dots\le(1-\tfrac{1}{OPT})^i|U_0|=(1-\tfrac{1}{OPT})^i n\le e^{-\frac{i}{OPT}}\cdot n$$

האי-שוויון האחרון נובע מהכלל הידוע $1-x\le e^{-x}$ $\forall x\in[0,1]$.

נחזור להוכחה. כעת כיוון שהסדרה $U_0,U_1,\dots$ היא סדרה יורדת בכל איטרציה, בהכרח יש איטרציה i^* שלאחריה מספר האיברים שנותרו לכסות קטן מ-OPT, כלומר $|U_{i^*}|\ge OPT > |U_{i^*+1}|$. במקרה הגרוע ביותר לאחר ה-i^*+1 יישארו $OPT-1$ איברים ובכל איטרציה מכאן והלאה נוסיף איבר 1 חדש בלבד. לכן נקבל שמספר האיטרציות (שהוא מספר הקבוצות בפתרון החמדני) הוא: $i^*+1+OPT-1=\underline{i^*+OPT}$.

נמצא חסם עליון לביטוי זה. לפי הגדרת $|U_{i^*}|$ וטענת עזר, נקבל:

$$OPT\le|U_{i^*}|\le e^{-\frac{i^*}{OPT}}\cdot n \Rightarrow OPT\le\frac{1}{e^{\frac{i^*}{OPT}}}\cdot n \;\Big/\; \frac{e^{\frac{i^*}{OPT}}}{OPT} \Rightarrow e^{\frac{i^*}{OPT}}\le\frac{n}{OPT} \;/\ln \Rightarrow \ln\left(e^{\frac{i^*}{OPT}}\right)\le\ln\left(\frac{n}{OPT}\right)\Rightarrow$$

$$\Rightarrow \frac{i^*}{OPT}\le\ln\left(\frac{n}{OPT}\right) \Rightarrow i^*\le\ln\left(\frac{n}{OPT}\right)\cdot OPT \;/+OPT \Rightarrow$$

$$\Rightarrow i^*+OPT\le\ln\left(\frac{n}{OPT}\right)\cdot OPT+OPT\le\left(\ln\left(\frac{n}{OPT}\right)+1\right)OPT\le(\ln(m)+1)OPT=O(\ln(m))OPT$$

האי-שוויון האחרון נובע מכך ש-$\frac{n}{OPT}\le m$ שהרי כל פתרון אופטימלי מכיל לפחות קבוצה אחת בגודל $\frac{n}{OPT}$, שאחרת לא היה מכסה כל האיברים ב-Ω. נמצא כי $i^*+OPT\le O(\ln(m))\cdot OPT$, ולכן האלגוריתם $O(\ln(m))$ מקרב.

מ.ש.ל

18/5

1) Min Cost Set-Cover - אלגוריתם חמדני

נרצה להשתמש באותו אלגוריתם חמדני כמו בשאלה הקודמת, אלא שכעת נצטרך להתחשב במשקל הקבוצה ולא רק במספר האיברים החדשים מ-Ω שמוסיפה לכיסוי $\mathcal{A}$. נעשה זאת בדרך הבאה: בכל תחילת איטרציה i שבה כבר כיסינו $\bigcup A_i$ איברים, נגדיר לכל $c\in\Psi$ את האפקטיביות של c באיטרציה i להיות:

$$eff_{\bigcup A_i}(c)=\begin{cases}\infty & c\subseteq \bigcup A_i\\ \frac{c(c)}{|c\setminus \bigcup A_i|} & \text{otherwise}\end{cases}$$

כלומר, כאשר $c\subseteq \bigcup A_i$ אזי בחירת c אינה מוסיפה כלום לכיסוי, ולכן האפקטיביות שלה ∞. אחרת, האפקטיביות שלה שווה לעלות שלה חלקי כמות האיברים החדשים שמוסיפה לכיסוי.

```
Min-Cost-set-cover-greedy(Ω,Ψ) {
    A = ϕ
    while(∪A ≠ Ω) {
        c = min{ eff∪A(c) : c∈Ψ }
        A = A ∪ {c}
    }
    return A
}
```

כך בעצם העלות עבור כל איבר חדש. אנו נבחר את ה-c שהאפקטיביות שלו הכי נמוכה.

<u>הוכחת $O(\ln(n))$ קירוב:</u> נוכיח קודם טענת עזר.

טענת עזר: נגדיר $T=\{c\in OPT: c\setminus \bigcup A_i\neq\emptyset\}$, כלומר T הוא איחוד כל הקבוצות ב-OPT שאם נוסיף ל-$\mathcal{A}$ באיטרציה ה-i הם יכסו איברים חדשים. הטענת עזר אומרת: ק"מ $c\in T$ כך ש- $\frac{c(c)}{|c\setminus \bigcup A_i|}\le\frac{OPT}{|\Omega\setminus \bigcup A_i|}$

הוכחה: נניח בשלילה שהטענה לא נכונה, כלומר $\frac{c(c)}{|c\setminus \bigcup A_i|}>\frac{OPT}{|\Omega\setminus \bigcup A_i|}$. אזי:

$$OPT\underset{\text{עלות חלק מקבוצות OPT}}{\ge}\sum_{c\in T}c(c)\underset{\text{יוסבר בהמשך}}{=}\sum_{c\in T}\sum_{e\in c\setminus\bigcup A_i}\frac{c(c)}{|c\setminus\bigcup A_i|}\underset{\text{הנחה בשלילה}}{>}\sum_{c\in T}\sum_{e\in c\setminus\bigcup A_i}\frac{OPT}{|\Omega\setminus\bigcup A_i|}\underset{\text{הוצאת גורם שלא תלוי ב-T}}{=}\frac{OPT}{|\Omega\setminus\bigcup A_i|}\cdot\sum_{c\in T}\sum_{e\in c\setminus\bigcup A_i}1\underset{\text{סכום כל הקשתות ב-OPT שאינם ב-}\bigcup A_i}{\ge}\frac{OPT}{|\Omega\setminus\bigcup A_i|}\cdot|\Omega\setminus\bigcup A_i|=OPT$$

בשוויון השני אנו מחלקים עלות כל קבוצה בצורה שווה לכל האיברים החדשים שמוסיפה לכיסוי. סכום כל עלויות אלו הוא כמובן עלות הקבוצה. שיטה זו לפירוק העלות נקראת "ניתוח פחת" (amortized analysis). קיבלנו $OPT>OPT$ שזוהי סתירה, ולכן הטענת עזר מתקיימת.

נחזור להוכחה. נשים לב שהביטוי $\frac{c(c)}{|c\setminus \bigcup A_i|}$ זהה להגדרת אפקטיביות של קבוצה. כיוון שהאלגוריתם מחפש בכל איטרציה קבוצה שהאפקטיביות שלה מינימלית, ובטענת עזר הוכחנו שקיימת קבוצה שהאפקטיביות שלה חסומה מלמעלה, בוודאי בכל איטרציה תיבחר קבוצה $c\in\Psi$ שהאפקטיביות שלה לכל היותר $\frac{OPT}{|\Omega\setminus\bigcup A_i|}$.

עבור כל קבוצה c שהאלגוריתם בוחר נגדיר לכל איבר חדש $e\in c\setminus\bigcup\mathcal{A}$ שהצטרף לכיסוי, משקל $ac(e)$ השווה לאפקטיביות של c. $ac(e)=eff_{\bigcup\mathcal{A}}(c)=\frac{c(c)}{|c\setminus\bigcup A_i|}$. ננסה למצוא את עלות הפתרון באמצעות משקלים אלו.

נסמן ב-$A=\{c_1,c_2,\dots,c_m\}$ את הפתרון שהוחזר מהאלגוריתם החמדני. נשים לב כי $A_i=\{c_1,c_2,\dots,c_{i-1}\}$ כל הקבוצות שנבחרו עד תחילת איטרציה i. העלות הכוללת של A היא:

$$c(A)=\sum_{c\in A}c(c)=\sum_{i=1}^{m}c(c_i)\underset{\text{ניתוח פחת}}{=}\sum_{i=1}^{m}\sum_{e\in c_i\setminus\bigcup A_i}\frac{c(c_i)}{|c_i\setminus\bigcup A_i|}=\sum_{i=1}^{m}\sum_{e\in c_i\setminus\bigcup A_i}ac(e)=\sum_{e\in\Omega}ac(e)$$

השוויון האחרון נובע מכך שאנו סוכמים את עלות המשקל של כל האיברים החדשים שמתווספים בכל איטרציה, שבסופו מתכנסים ב-Ω. ננסה למצוא ל-$\sum_{e\in\Omega}ac(e)$ חסם עליון.

בכל איטרציה $i\in[m]$ נסמן את מספר האיברים החדשים שאנו מכסים $k(i)$, ונסמן אותם כך:
$e_1^{(i)}, e_2^{(i)}, \dots, e_{k(i)}^{(i)}$. כל איברים אלו בכל איטרציה משלמים שווה: $ac(e_1^{(i)})=\dots=ac(e_{k(i)}^{(i)})=eff_{\cup A_i}(c_i)=\frac{c(c_i)}{|c_i\setminus \cup A_i|}$.

כעת נסדר את ה האיברים ב-Ω לפי סדר הכנסתם ל-A לפי האלגוריתם החמדני.

$$\Omega=\{e_1,e_2,\dots,e_n\}=\{e_1^{(1)},e_2^{(1)},\dots,e_{k(1)}^{(1)},e_1^{(2)},\dots,e_{k(2)}^{(2)},\dots,e_1^{(m)},\dots,e_{k(m)}^{(m)}\}$$

כדי למצוא את העלות הכוללת של הפתרון המוחזר מהאלגוריתם, נדרש לסכום משקל כל האיברים ב-Ω.
ננסה למצוא חסם עליון למשקל כל איבר כזה.

עבור איטרציה $i\in[m]$ בה בחרנו את הקבוצה $c_i\in\mathcal{C}$, נסמן את מיקום האיבר הראשון בקבוצה זו $e_k=e_1^{(i)}$, כאשר k המיקום של איבר זה בסדרה שעל Ω. מכאן יוצא שבתחילת איטרציה i כיסינו $k-1$ איברים, $|\cup A_i|=k-1$. מטענה זו והשילוב עם טענת העזר שעל, נוכל להניח שקיים $c\in OPT$ המקיים $eff_{\cup A_i}(c)=\frac{c(c)}{|c\setminus\cup A_i|}\le\frac{OPT}{|\Omega\setminus\cup A_i|}=\frac{OPT}{|\Omega|-|\cup A_i|}=\frac{OPT}{n-k+1}$. לכן באיטרציה i, שבה נבחרה הקבוצה c_i שהאפקטיביות שלה מינימלית, מתקיים: $eff_{\cup A_i}(c_i)\le eff_{\cup A_i}(c)\le\frac{OPT}{n-k+1}$. ומכאן $ac(e_k)\le\frac{OPT}{n-k+1}$. טענה זו נכונה לכל האיברים $e_{k+\ell}$, כאשר $0\le\ell\le k(i)$, שאנו מכסים באיטרציה i. $ac(e_{k+\ell})\le\frac{OPT}{n-k+1}$.
מכיוון ש-$n-(k+\ell)+1\le n-k+1$, גם נכון לומר כי $ac(e_{k+\ell})\le\frac{OPT}{n-(k+\ell)+1}$, שהרי הקטנת המכנה רק מגדילה את השבר. כיוון שטענה זו הוכחה עבור איטרציה i כללית ועבור מיקומו של האיבר בסדרה שעל Ω, נסיק כי הטענה נכונה עבור כל איבר ב-Ω. $ac(e_r)\le\frac{OPT}{n-r+1}$, כאשר r הוא מיקומו של האיבר e_r בסדרה שעל Ω.

כעת נוכל לחשב את העלות הכוללת של הפתרון המוחזר מהאלגוריתם החמדני.

$$c(A)\underset{\text{הראנו קודם}}{=}\sum_{i=1}^{n}ac(e_i)\underset{\text{טענה אחרונה}}{\le}\sum_{j=1}^{n}\frac{OPT}{n-j+1}\underset{\text{שינוי סדר סכימה}}{=}OPT\cdot\sum_{j=1}^{n}\frac{1}{j}\underset{\text{סכום טור הרמוני חסום}}{\le}(\ln(n)+1)OPT=O(\ln(n))OPT$$ מש"ל

א') Min cost set-cover - עיגול ה-primal

אלגוריתם קירוב נוסף לבעיית min cost SC באמצעות תכנון לינארי. נתחיל בהגדרת הבעיה בתכנון בשלמים ותכנון לינארי, לאחר מכן נציג אלגוריתם שפותר את הבעיה באמצעות אלגוריתם לפתרון LP (סימפלקס או אליפסואיד), ואז נעגל את הפתרון כדי לקבל פתרון בשלמים.

הגדרת הבעיה ב-LP/IP היא:

$$\min \sum_{c\in\mathcal{C}} c(c)\cdot x_c$$

$$\text{s.t.}\quad \sum_{c\in C_e} x_c\ge 1\quad \forall e\in\Omega: C_e=\{c\in\mathcal{C}: e\in c\}$$

$$\overset{IP}{x_c\in\{0,1\}\ \forall c\in\mathcal{C}}\ \Big/\ \overset{LP}{x_c\ge 0\ \forall c\in\mathcal{C}}$$

בוקטור המשתנים x, יהיה משתנה x_c לכל קבוצה $c\in\mathcal{C}$, שתקבל 1 אם c נמצאת בפתרון המוחזר מהאלגוריתם, או 0 אם אינה נמצאת. עבור האילוצים נגדיר לכל $e\in\Omega$ קבוצה C_e המכילה את כל הקבוצות ב-$\mathcal{C}$ ש-e מופיעה בהן, האילוץ הוא שהסכימה של כל המשתנים ב-x המתאימה לקבוצה ב-C_e, יהיה לפחות 1. כלומר בחרנו לפחות קבוצה 1 שמכילה את e. כך עבור כל $e\in\Omega$.

אנו רוצים שעלות כל הקבוצות שנבחרו תהיה מינימלית ולכן זוהי פונקציית המטרה.

נגדיר את עלות הפתרון לבעיה בתכנון בשלמים OPT, ועלות הפתרון בתכנון לינארי OPT_f. נשים לב כי $OPT_f\le OPT$ שהרי אנו מחפשים מינימום וב-LP אנו שוקלים יותר אפשרויות.

נסמן לכל איבר $e\in\Omega$ את מספר הקבוצות ב-$\mathcal{C}$ בהן הוא מופיע ב-f_e. מתקיים $f_e=|C_e|$. עוד נסמן ב-f את הערך f_e המקסימלי מבין כל האיברים, $f=\max\limits_{e\in\Omega} f_e$.

```
Deterministic Rounding LP Set-Cover (Ω, 𝒞) {
    let x* be the primal solution to the LP problem above
    A = {c ∈ 𝒞 : x*_c ≥ 1/f}
    return A
}
```

אלגוריתם זה "מעגל" את הפתרון ב-LP בכך שהוא מתייחס לכל x^*_c שגדול מ-$\frac{1}{f}$ כאילו כתוב שם 1 ולכן כולל אותו ב-A.

<u>**הוכחת f קירוב:**</u> נוכיח את השני שלבים.

(1) הוכחה ש-A הוא set-cover תקני: צריך להוכיח $\Omega=U_A$. יהי $e\in\Omega$. כיוון ש-x^* פתרון ל-LP הוא מקיים את האילוץ עבור e, ולכן $\sum\limits_{c\in C_e} x^*_c\ge 1$. כתוצאה מכך, חייב להיות $c\in C_e$ עבורו מתקיים $x^*_c\ge\frac{1}{|C_e|}=\frac{1}{f_e}$, שהרי אם כל $c\in C_e$ מקיים $x^*_c<\frac{1}{|C_e|}$ אזי סכום כולם לא היה מגיע ל-1 והאילוץ לא היה מתקיים. כעת, מכיוון ש-$f_e\le f$ שהרי f מקסימלי, מתקיים $x^*_c\ge\frac{1}{f_e}\ge\frac{1}{f}$. לכן c תיכלל ב-A, וכתוצאה מכך A אכן מכסה את e, $e\in U_A$.

(2) נוכיח שהאלגוריתם f-מקרב: נגדיר וקטור z עבור כל $c\in\mathcal{C}$, כך ש: $z_c=\begin{cases}1 & c\in A\\ x^*_c & c\notin A\end{cases}$.

נשים לב שלכל $c\in\mathcal{C}$ מתקיים $z_c\le f\cdot x^*_c$. שהרי אם $c\in A$ אזי $f\cdot x^*_c\ge f\cdot\frac{1}{f}=1$. ואם $c\notin A$ אזי $f\cdot x^*_c\ge x^*_c$ שהרי תמיד $f\ge 1$. נחשב את העלות הכוללת של A:

$$c(A)=\sum_{c\in A}c(c)\underset{\substack{z_c \text{ שווה } 1\\ \text{בקבוצות אלו}}}{\underset{c\in A}{=}}\sum c(c)\cdot z_c\underset{\text{הוספת קבוצות}}{\underset{c\in\mathcal{C}}{\le}}\sum c(c)\cdot z_c\underset{\substack{\text{הבחנה}\\ \text{קודמת}}}{\underset{c\in\mathcal{C}}{\le}}\sum c(c)\cdot f\cdot x^*_c\underset{f \text{ הוצאת}}{=}f\cdot\sum_{c\in\mathcal{C}}c(c)\cdot x^*_c=f\cdot OPT_f\le f\cdot OPT$$

מ.ש.ל

צרך הוכחה זו היא סטנדרטית עבור בעיות קירוב עם LP ובצורה דומה מוכיחים בעיות נוספות.*

מאת עבודה 6 שאלה 1. בדרך אם נגא בניטה $\frac{1}{x}$ אז קירוב הוא x.

<u>מסקנה:</u> יש לנו שני אלגוריתמי קירוב. עבור כל מופע של בעיית min cost set-cover, נחליט איזה אלגוריתם להשתמש בהתאם לחישוב $\min\{\ln(n),f\}$. אם $\ln(n)$ קטן יותר נשתמש בחמדני, ואם f קטן יותר, נשתמש בעיגול LP.

Min Vertex-Cover חום

נראה מספר שיטות למציאת קירוב ל-Min Vertex-cover חום.

(1) באמצעות שידוך

```
Min vertex-cover via Matching(G) {
    let M be maximal matching in G
    return V(M)
}
```

שידוך מקסימלי ניתן למצוא בזמן פולינומי. $V(M)$ הוא VC שהרי אם יש קשת ש-M לא מכסה, אזי הוא לא מקסימלי.

הוכחת 2-קירוב: מטענה אחרונה נובע כי תמיד מתקיים $|M| \le \tau(G)$ לכל שידוך. ונקבל $2|M| \le 2\cdot\tau(G)$. $2|M|$ זהו מספר הצמתים בכל שידוך ובפרט בשידוך המקסימלי. לכן $|V(G)| \le 2\cdot\tau(G) = 2\cdot OPT$. מ.ש.ל.

(2) אלגוריתם savage's

כל הצמתים הפנימיים בעץ DFS של G הם VC.

```
Savag's algorithm(G) {
    let T be DFS tree
    I(T) all vertex in T that are non-leaf vertices
    return I(T)
}
```

הוכחת 2 קירוב: נתחיל בטענת עזר;

טענת עזר: ב-T יש שידוך בגודל לפחות $\frac{|I(T)|}{2}$.

הוכחה: באינדוקציה על מספר הצמתים.

בסיס - עבור $V(G)=1$ השידוך בגודל 0.

צעד - נניח עבור $n-1$ ונוכיח עבור n. נסמן את תתי העצים של השורש $T_1,\dots T_k$, את תתי העצים של T_1 ב- $S_1,\dots,S_t$. בכולם יש פחות מ-n צמתים, לכן מתקיימת הטענה הבאה המסתמכת על הנחת האינדוקציה $|M| = \sum_{i=2}^{k}\frac{|I(T_i)|}{2} + \sum_{i=1}^{t}\frac{|I(S_i)|}{2} + 1$

בנוסף סכום כל הצמתים הפנימיים שווה לסכום הצמתים הפנימיים ב-T מלבד השורש והקודקוד הראשון T_1. כלומר, $\sum_{i=1}^{t}|I(S_i)| + \sum_{i=2}^{k}|I(T_i)| = |I(T)|-2$, נחלק משוואה זו ב-2 ונציב במשוואה שלנו ונקבל $|M| = \frac{|I(T)|-2}{2} + 1 \Rightarrow |M| = \frac{|I(T)|}{2}$ מ.ש.ל.

כעת מטענה משיטה קודמת נקבל: $|M| = \frac{|I(T)|}{2} \le \tau(G)$, ומכאן, $|I(T)| \le 2\cdot\tau(G) = 2\cdot OPT$.

(3) אלגוריתם קירוב באמצעות ה- Primal LP עבור הע"ת min VC ממושקלת במטלה 6.

גם אלגוריתם זה 2-מקרב.