

# **בティוחות רשות**

**פרויקט סוף**

**סמסטר א', 2024**

## **חלק מחשב**

**מגישים:**

**אסתר מרגוליס 337624670**

**נמרוד ארקוסין 208665554**

**גולן הרץ 207070319**

**רז זהר 206816555**

**נועם פרץ 313598823**

## **תוכן עניינים**

3 .....	שאלה 1 – אנגליתית
5 .....	שאלה 2 + רקע כללי לסימולטור
8 .....	שאלה 3
12 .....	שאלה 4
20 .....	שאלה 5
26 .....	שאלה 6
29 .....	שאלה 7

## שאלה 1 – אנליזית

בשאלה זו נתבקשנו לפתור בצורה אנליטית את פתרון בעיית NUM תחת קרטוריונים מסוימים ולעורך השוואה כיצד הדבר מתישב עם ההגדרה והפתרונו שהוצעו כחלק מההרצאות בקורס. נשתמש בnotations שהוצעו כחלק מההרצאות בקורס.

$$U_r(X_r) = \frac{X_r^{1-\alpha}}{1-\alpha}, \text{ given that } \alpha \geq 0$$

So the optimization problem will be:

$$\max_{X_r} \sum_r U_r(X_r) \text{ s.t. } \begin{cases} 1^T X \leq C_l \\ X \geq 0 \end{cases}$$

so  $X_0 + X_l \leq C_l$ ; for  $l \in \{1, \dots, L\}$

given that  $C_l = 1$  for every link  $l$

$$X_0 + X_l \leq 1; \forall l \in \{1, \dots, L\}$$

Because this is convex optimaztion problem we will write the Lagranzian function:

$$L(x, \lambda) = \sum_r U_r(X_r) - \sum_{l=1}^L \lambda_l \cdot (X_0 + X_l - 1)$$

Revoke that  $\nabla_x L(x, \lambda) = 0$

$$X_l^\alpha = \begin{cases} \frac{1}{\sum_{l=1}^L \lambda_l}, & l = 0 \\ \frac{1}{\lambda_l}, & \text{otherwise} \end{cases}$$

From KKT third condition:

$$\lambda_l(X_0 + X_l - 1) = 0; \forall l \in \{1, \dots, L\}, \lambda_l \geq 0$$

Given that  $X_l \geq 0$  (otherwise the logarithm will approcaches  $-\infty$ )

$$X_0 + X_l - 1 = 0; \forall l \in \{1, \dots, L\} \rightarrow X_i = X_j \text{ for } i, j \neq 0$$

so then  $\lambda_i = \lambda_j$  for every  $i, j \neq 0 \rightarrow$

$$X_l = \begin{cases} \frac{1}{\sqrt[\alpha]{L\lambda_1}}, & l = 0 \\ \frac{1}{\sqrt[\alpha]{\lambda_1}}, & \text{otherwise} \end{cases} \text{ for convinince we use } \lambda_1 \text{ but that correct for every } \lambda$$

$$\text{so } \frac{1}{\sqrt[\alpha]{L\lambda_1}} + \frac{1}{\sqrt[\alpha]{\lambda_1}} - 1 = 0 \rightarrow \lambda_1 = \left( \frac{\sqrt[\alpha]{L}}{1 + \sqrt[\alpha]{L}} \right)^\alpha$$

כעת נוכל לחשב עבור כל אחד מהערכים על מנת לקבל את הקритריון המתאים כמפורט.

$$\lim_{\alpha \rightarrow 1} X_l = \lim_{\alpha \rightarrow 1} X_l = \begin{cases} \frac{1}{1 + \sqrt[\alpha]{L}}, & l = 0 \\ \frac{\sqrt[\alpha]{L}}{1 + \sqrt[\alpha]{L}}, & otherwise \end{cases} = \begin{cases} \frac{1}{L + 1} \\ \frac{L}{L + 1} \end{cases}$$

נשים לב שגם גם התוצאה שקיבלנו בחישוב מההרצאה .

כעת עבור קרטריון מינימום שהייה נרצה לחשב את הגבול עבור  $\alpha = 2$

$$\lim_{\alpha \rightarrow 2} X_l = \lim_{\alpha \rightarrow 2} X_l = \begin{cases} \frac{1}{1 + \sqrt[\alpha]{L}}, & l = 0 \\ \frac{\sqrt[\alpha]{L}}{1 + \sqrt[\alpha]{L}}, & otherwise \end{cases} = \begin{cases} \frac{1}{\sqrt{L} + 1} \\ \frac{\sqrt{L}}{\sqrt{L} + 1} \end{cases}$$

עבור קרטריון מקס-מינ נקבל:

$$\lim_{\alpha \rightarrow \infty} X_l = \lim_{\alpha \rightarrow \infty} X_l = \begin{cases} \frac{1}{1 + \sqrt[\alpha]{L}}, & l = 0 \\ \frac{\sqrt[\alpha]{L}}{1 + \sqrt[\alpha]{L}}, & otherwise \end{cases} = \begin{cases} \frac{1}{L^0 + 1} = \frac{1}{2} \\ \frac{L^0}{L^0 + 1} \end{cases}$$

כעת נשים לב שתחת קרטריון מינ-מקס מסתדרת עם ההגדרה שנלמדה בהרצאה כיון שבניתו שבירצנו בהרצאה וגם בשאלת זו על כל אחד מהلينקים נמצאים שני משתמשים (המשתמש 0 ומשתמש נוסף בリンק) אז הקרטריון מתקיים כאשר הקצבים יהיו שווים בין המשתמשים ותחת הנחת קיבול זהה בכל אחד מהلينקים ושווה ל 1 נקבל שהקצב המתאים הינו 0.5 ולכן ההגדרה מסתדרת עם הקצת הקצבים שהושבה והוצגה מעלה.

## שאלה 2 + רקע כללי לSIMULATOR

נשים לב שבאften כללי SIMULATOR הרשות שבנו מאפשר גמישות מירבית בכמות הLINKS, ותוכנות רשות כדוגמת קביעת פונקציות רוחב פס, קביעת פונקציות הספק ועוד כאשר השתמשנו בפונקציות ברירת מחדל על מנת להגדיר את התנהלות הרצiosa עבור שאלה 1 ואופציה של פונקציות ארגומנטים לשאר השאלות.

על מנת לקבל תוצאות רשות בעלות משמעות – יצאנו מנקודת הנחה כי הגרף אשר מתאר את הרשות הינו גוף קשור שכן אחרת לא ניתן יהיה להגעה מכל אחת מהצמתים לכל אחת אחרת.

לכן עבור שאלה 2 הרצינו את בניית SIMULATOR הרשות עבורה רשות שבה מוגלים הצמתים ומוחברים לפיה מרחקם מצמתים אחרים. וזאת תוך הקפדה על כך שהגרף קשור – נשים לב שישנה אפשרות של טעינה ושמירה של מצב הSIMULATOR (גרף – קלומר קשתות וצמתים) לשימוש עתידי במידת הצורך.

לכן, עבור שאלה מס' 2, כל הרצה של הSIMULATOR תיתן תוצאה אחרת בהתאם לגרף ומהוות פתרון לשאלה זו.

נשים לב שעלה מנת לקבל SIMULATOR ריאლיסטית יותר הקפדו לכלול בכל אחת מהרציות הSIMULATOR את ערכיו ההפרעה שונים מאפס.

בשאלה זו נתקבשנו למשם את האלגוריתם הפרימאלי אשר פותר את בעיית Network Utility Maximization תחת קרטריון . alpha-fairness

זכיר כי האלגוריתם ממוש בצהה איטרטיבית – נציג את כלל האיטרציה תוך שימוש בnotations מההרצאות:

$$X_r^{t+1} = X_r^t + K_r(X_r^t) \left( U'_r(X_r^t) - f_l \left( \sum_r X_r \right) \right)$$

where  $f_l(\sum_r X_r) = \sum_{l:l \in r} e^{\beta(c_l - \sum_r x_r)}$

גרפים:

- Network Simulator Graph instance1 Figure
- גוף הקצבים בהרצה
- Figure 2 Primal Algorithm for NUM problem with alpha=1
- Figure 3 Primal Algorithm run for NUM problem for alpha = 2
- Figure 4 Primal Algorithm for NUM problem for alpha = 20
- נשים לב שבחינה חישובית עבור ערך של alpha=20 נקבל את התנהלות הרצiosa בגבול של אין סוף.

מהגרפים ניתן להסיק כי:

- חלוקת הקצבים מתחכמת בצהה שמתאימה לניתוח האנליטי שביצענו עבור שאלה 1 ובוצע בהרצאות כמו כן אנחנו שמים לב כי בהערכתו שונה בהינתן צורה הגרף נקבל הקצתה קצבים שונה כמו שהיוו מצפים.
- כמו שהיוו מצפים לכל ערך שונה של alpha אנחנו מקבלים קצב שונה שכן הCARTRIOON שונה ובצהה אנליטית alpha משפיע על הקצבים.
- נשים לב שהיוו צרכים לבחור פונקציית מחיר כזו שיעבורה יתקבלו ערכיהם הגיוניים – גם נומריים עבור עדכון הקצב בכל שלב.
- נשים לב שעבור כלל המקרים קיבלנו התוכנות של הקצבים כפי שהיוו מצפים.

- נשים לב שעבור כל אחד מהמקרים הינו צריכים לפחות קבועים אחרים על מנת לקבל התוכניות וחישובים נומריים חוקיים.
- נשים לב כי בחלוקת הקצבים יש חלוקה לרמות קצבים לפי כמות הקשתות המחברות של כל אחד מהצטמים.

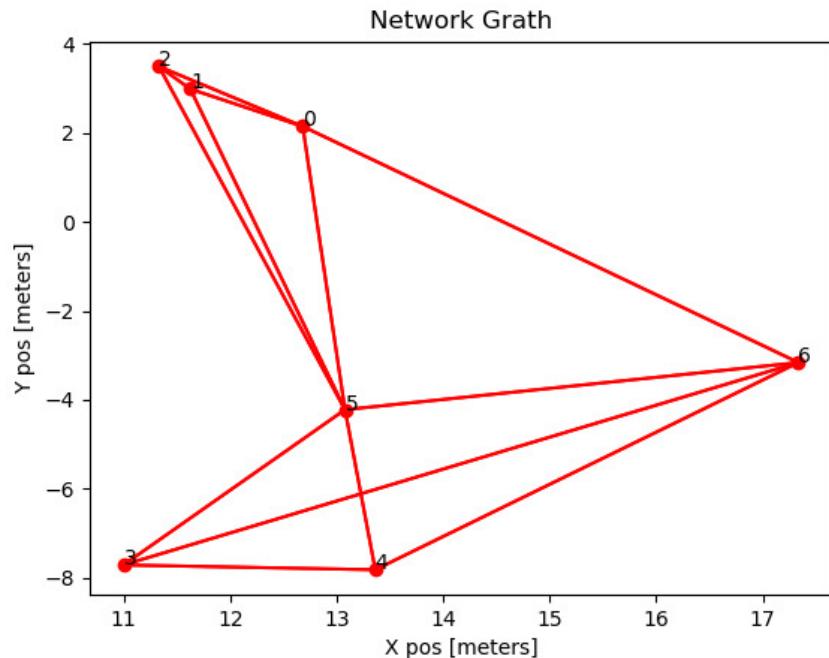
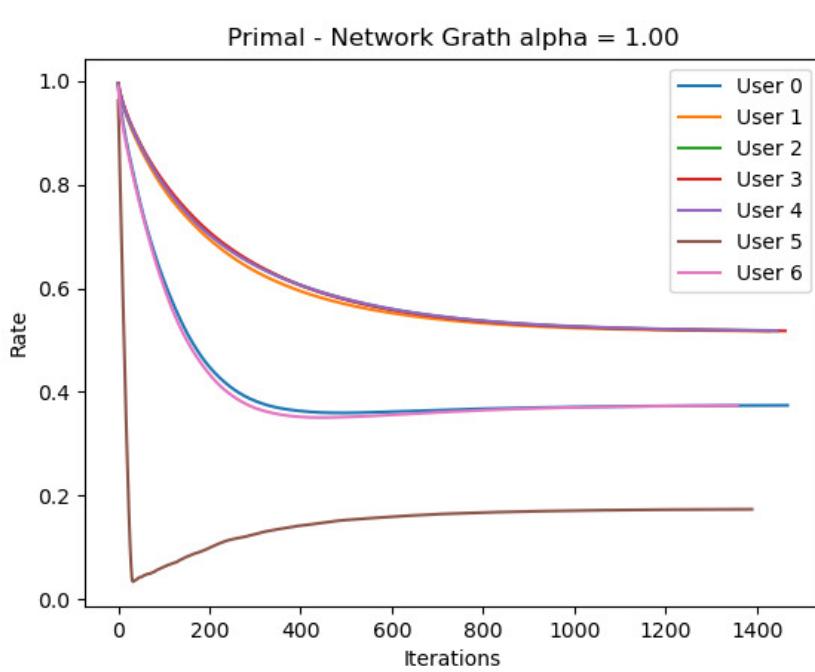


Figure 1 Network Simulator Graph instance



```
-----
user = 0, user_state.rate :0.3747351378278674
user = 1, user_state.rate :0.5171836328686725
user = 2, user_state.rate :0.5164743161462635
user = 3, user_state.rate :0.5163210890460735
user = 4, user_state.rate :0.5175208257888224
user = 5, user_state.rate :0.17436578875440462
user = 6, user_state.rate :0.37465971643667906
```

Figure 2 Primal Algorithm for NUM problem with alpha=1

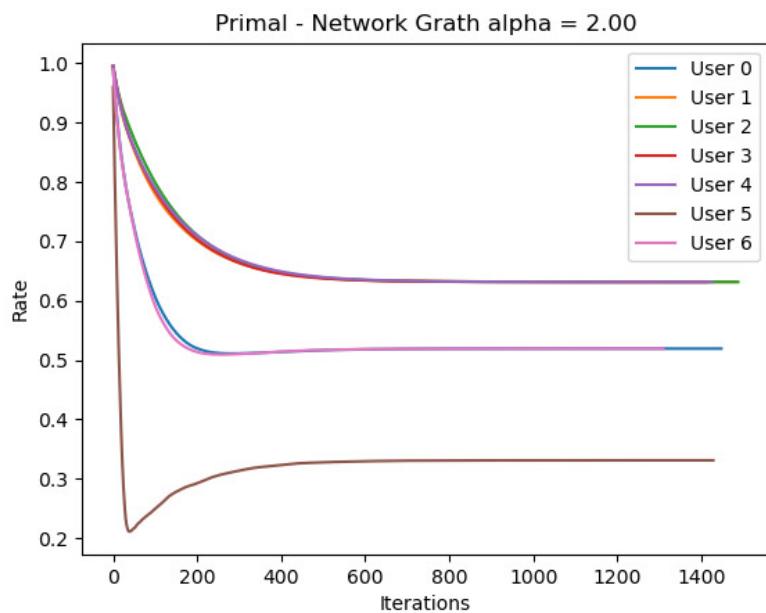


Figure 3 Primal Algorithm run for NUM problem for alpha = 2

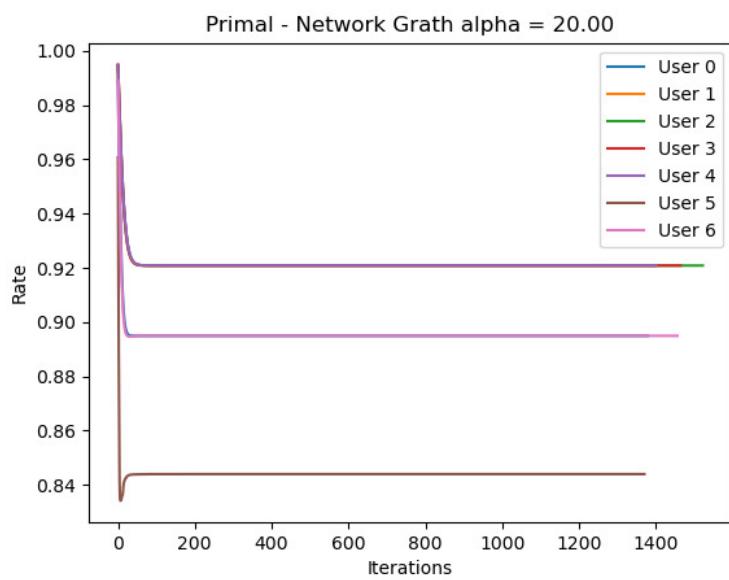


Figure 4 Primal Algorithm for NUM problem for alpha = 20

### שאלה 3

בשאלה זו התקשנו למש את האלגוריתם הדואלי אשר פותר את בעיית Network Utility Maximization קרטריון alpha-fairness .

נשים לב כי בכלל איטרציה אנו מעדכנים את כופלי לגראנץ ואו מתעדכנים הקצבים בכלל אחת מהאיטרציות.

זכור כי האלגוריתם ממוש בצורה איטרטיבית – נציג את כלל האיטרציה תוך שימוש בnotations מההרצאות :

$$X_r^{t+1} = \left( \sum_{l \in r} \lambda_l^t \right)^{-\frac{1}{\alpha}}$$

where for lagarnz multiplier iteration update rule:

$$\lambda_l^{t+1} = \lambda_l^t + h_l \cdot f_l(\lambda_l)$$

$$\text{where } f_l(\lambda_l) = \begin{cases} \sum_{r: l \in r} X_r^t - C_l, & \lambda_l > 0 \\ \max \left( 0, \sum_{r: l \in r} X_r^t - C_l \right), & \lambda_l = 0 \end{cases}$$

גרפים:

Figure 5 Simulator Network Graph at run •

◦ נשים לב שהשתמשנו באותו הגרף אשר מוצג גם בשאלה 2

◦ גראף הקצבים בהרצתה

◦ Figure 6 Dual Algorithm run for NUM problem with alpha=1

◦ Figure7 Dual Algorithm run for NUM problem with alpha=2

◦ Figure 8 Dual Algorithm run for NUM problem with alpha=20

◦ נשים לב שUMB הישובית עבר ערך של alpha=20 נקבל את התנהוגות הרצויה בגבול של אינסוף .

מהגרפים ניתן להסיק כי:

• חלוקת הקצבים - אנחנו שמים לב כי בהינתן צורת הגרף נקבל הקצתה קצבים שונה כמו שהיוו מצפים.

• כמו שהיוו מצפים לכל ערך שונה של alpha אנחנו מקבלים קצב שונה שכן הקרטריון שונה ובצורה אנליזית alpha משפיע על הקצבים.

◦ נשים לב שעבור כל המקרים קיבלנו התכונות של הקצבים כפי שהיוו מצפים.

◦ נשים לב שעבור כל אחד מהמקרים היוו ציריכים לבחור קבועים אחרים על מנת לקבל התכונות והישובים נורמיים חוקיים.

◦ נשים לב כי בחלוקת הקצבים יש חלוקה לרמות קצבים לפי כמות הקשתות המחברות של כל אחד מהצמתים.

◦ נשים לב שהיוו מצפים שהגרפים והתוצאות ייצאו דומות לתוצאות ההרצתה של האלגוריתם הפרימלי ואצלנו הן ייצאו שונות אנחנו מעריכים כי זו תוצאה של תזרורת הגראפים בהרצת האלגוריתם וכמו כן תוצאה של רנדומליות ברעשי ההפרעה כי לפי החומר התיאורטי מתקיימת דואליות חזקה.

◦ ע"מ לאמת את נכונות האלגוריתם והקווד הרצינו אותו עבור הגרף המתואר בהרצתה אותו ניתחנו

Figure 9 Regular dual algorithm run -

- נשים לב שעבור גרף זה קיבלנו את התוצאות אשר ציפנו להן.

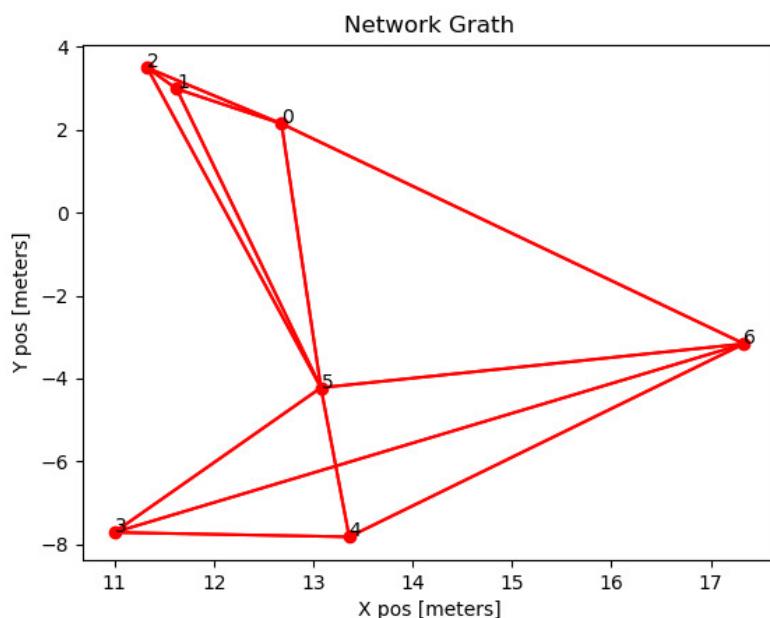
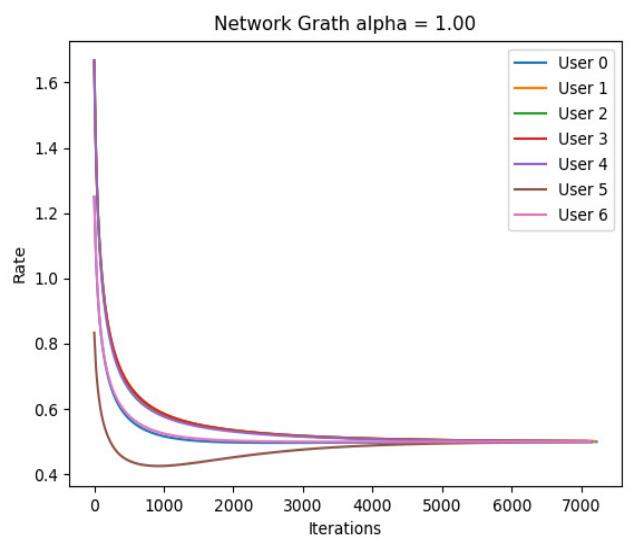


Figure 5 Simulator Network Graph at run



```

user = 0, user_state.rate :0.5000035268566309
user = 1, user_state.rate :0.5000031450206546
user = 2, user_state.rate :0.5000090174053519
user = 3, user_state.rate :0.5000126097744854
user = 4, user_state.rate :0.5000151374382342
user = 5, user_state.rate :0.49999103290951125
user = 6, user_state.rate :0.4999872879466277

```

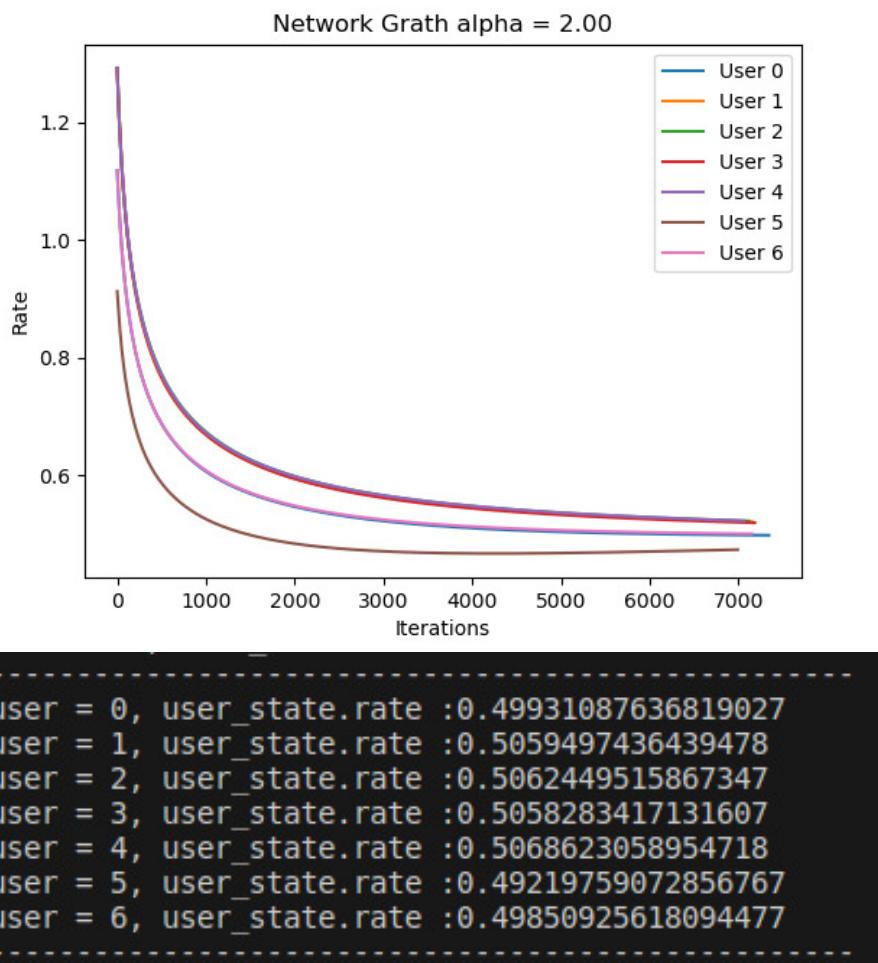
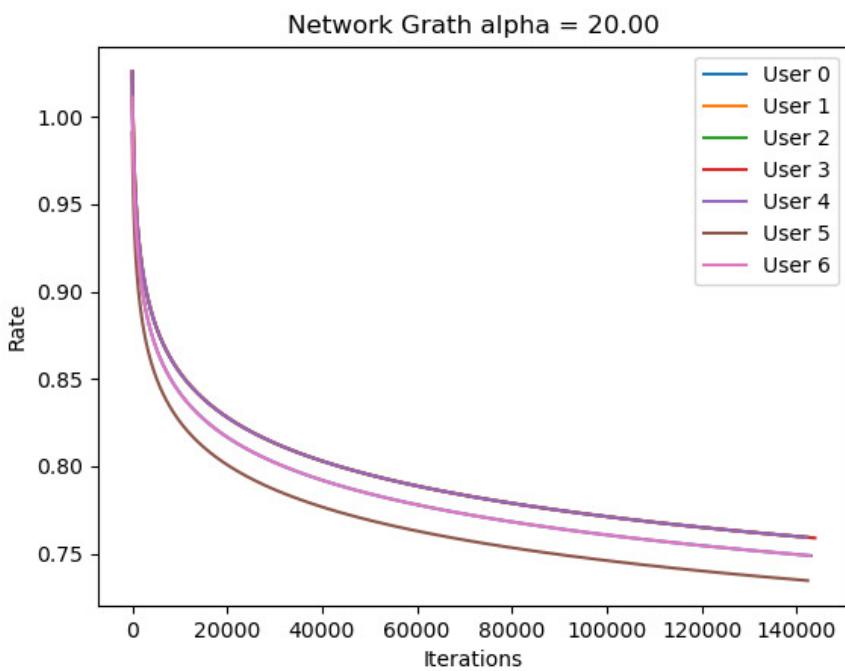


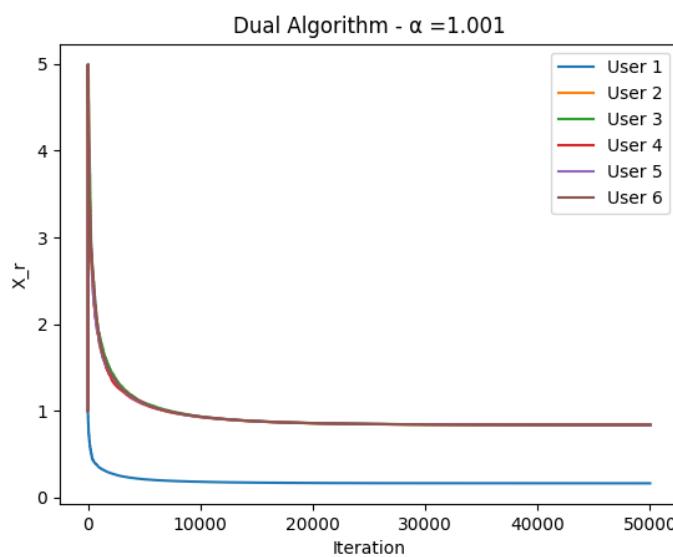
Figure 7 Dual Algorithm run for NUM problem with  $\alpha=2$



```

-----
user = 0, user_state.rate :0.8287142826271084
user = 1, user_state.rate :0.8411522103620177
user = 2, user_state.rate :0.8401995301357986
user = 3, user_state.rate :0.8405680554595979
user = 4, user_state.rate :0.8401666630962284
user = 5, user_state.rate :0.8129985667358183
user = 6, user_state.rate :0.8284097763667349
-----
```

Figure 8 Dual Algorithm run for NUM problem with alpha=20



```
Final rates are [0.16311315 0.83722127 0.837263 0.83718468 0.83728055 0.83724323]
```

```
Process finished with exit code 0
```

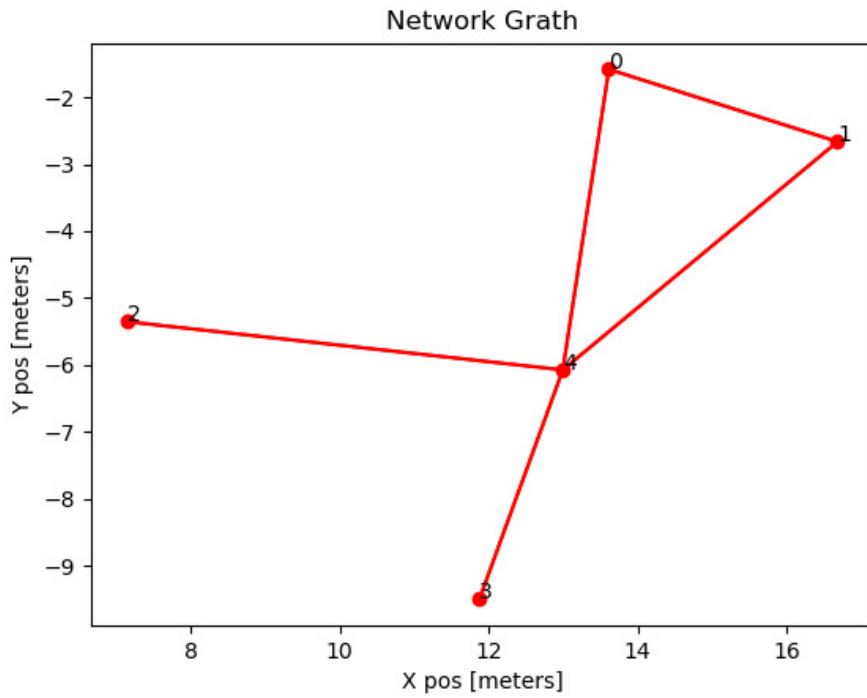
*Figure 9 Regular dual algorithm run*

## שאלה 4

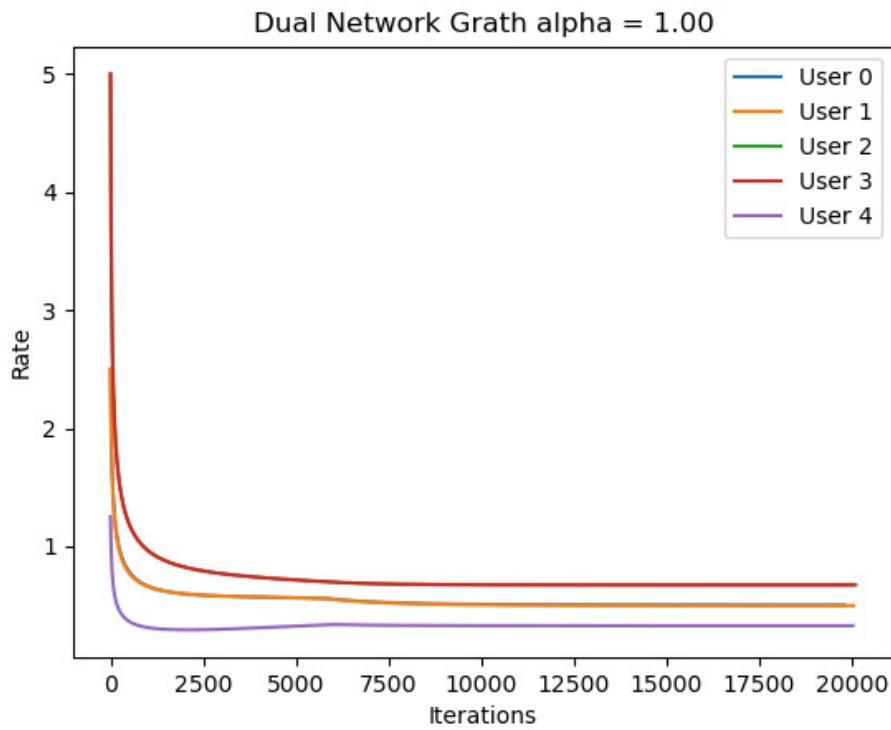
בשאלה זו התקשנו להריץ את האלגוריתמים הפרימלי והדואלי עבור רשת בה  $L=5$  ועבור ערכי  $\alpha$  מתאימים.

- Figure11 Network Simulator Graph for  $L=5$
- Figure 10 Dual Algorithm run for NUM with  $L=5$ ,  $\alpha=1$
- Figure 12 Dual Algorithm Run for NUM with  $L=5$ ,  $\alpha=2$
- Figure13 Dual Algorithm run for NUM with  $L=5$ ,  $\alpha=20$ 
  - הוספנו את הגרפים עבור המקרה בו  $L=5$ .
  - כתה הוספנו את הקצבים גם עבור האלגוריתם הפרימאלי:
    - Figure14 Primal Algorithm for NUM with  $L=5$ ,  $\alpha=1$
    - Figure15 Primal Algorithm run for NUM with  $L=5$ ,  $\alpha=2$
    - Figure16 Primal Algorithm run NUM ,  $L=5$ ,  $\alpha=20$
- נשים לב שהיינו מצפים והתוצאות ייצאו דומות לתוצאות ההרצאה של האלגוריתם הפרימלי ואצלנו הן ייצאו שונות אנחנו מעריכים כי זו תוצאה של הצורת הגרפים בהרצאת האלגוריתם וכמו כן תוצאה של רנדומליות בرعשי הפיעעה כי לפי החומר התיאורטי מתקיים דואליות חזקה.
  - ע"מ לאמת את נכונות האלגוריתם והקוד הרצינו אותו עבור הגרף המתואר בהרצאה אותו ניתחנו.
  - נשים לב שעבור גרף זה קיבלנו את התוצאות אשר ציפנו להן.

Figure 9 Reguler dual alghorithm run -



*Figure 11 Network Simulator Graph for L=5*

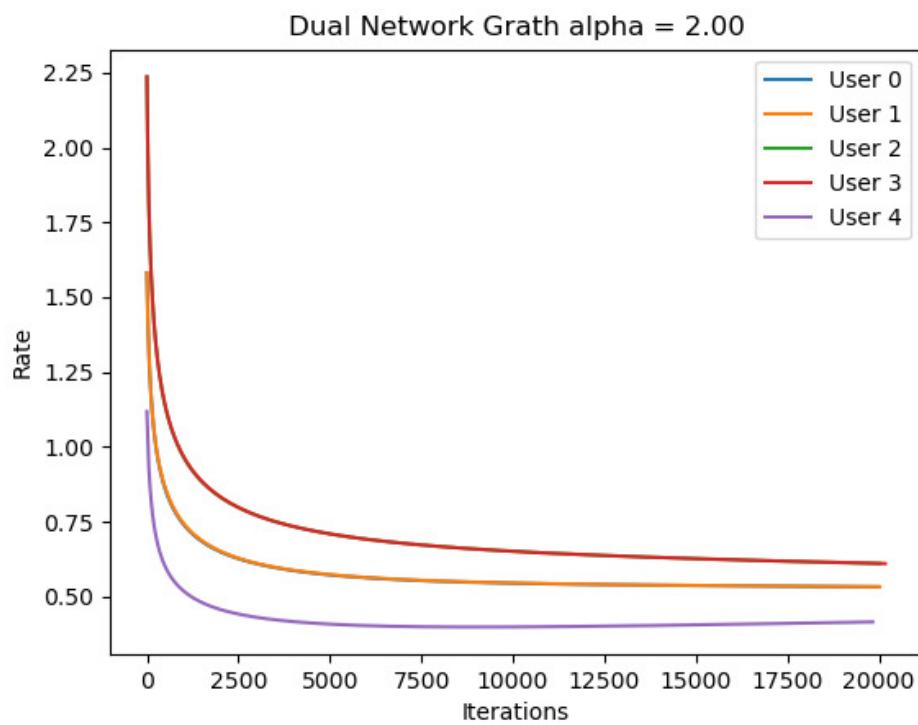


```

user = 0, user_state.rate :0.5025109883235233
user = 1, user_state.rate :0.49756956024728793
user = 2, user_state.rate :0.6726387881954617
user = 3, user_state.rate :0.6726387452093868
user = 4, user_state.rate :0.3273639677547512

```

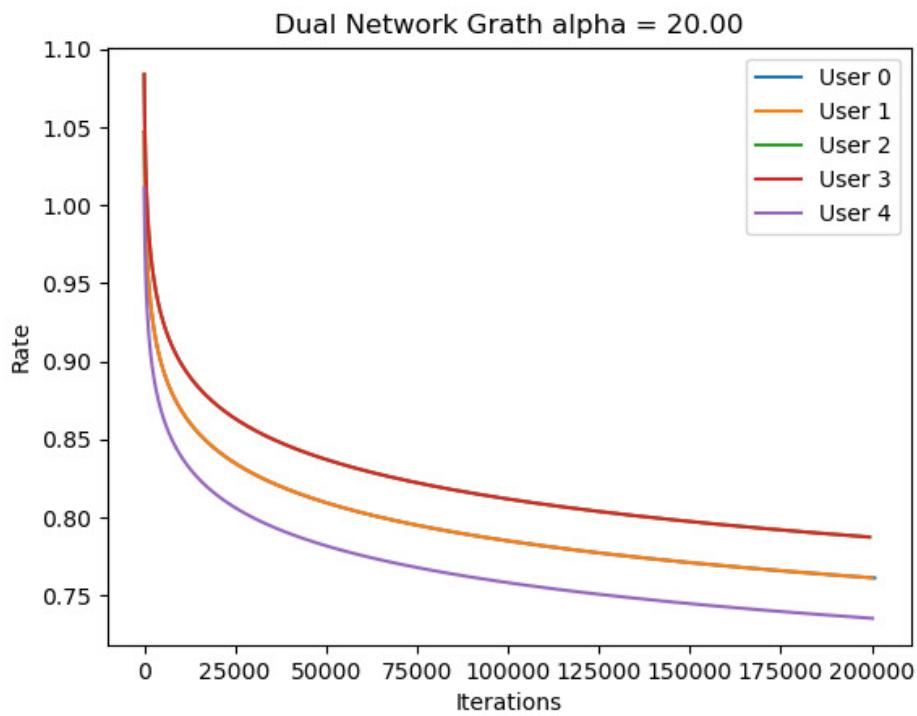
*Figure 10 Dual Algorithm run for NUM with L=5, alpha=1*



```

-----
user = 0, user_state.rate :0.531547569345951
user = 1, user_state.rate :0.5326838996225074
user = 2, user_state.rate :0.6101476743246282
user = 3, user_state.rate :0.6100620516937214
user = 4, user_state.rate :0.41414249648842794
-----
```

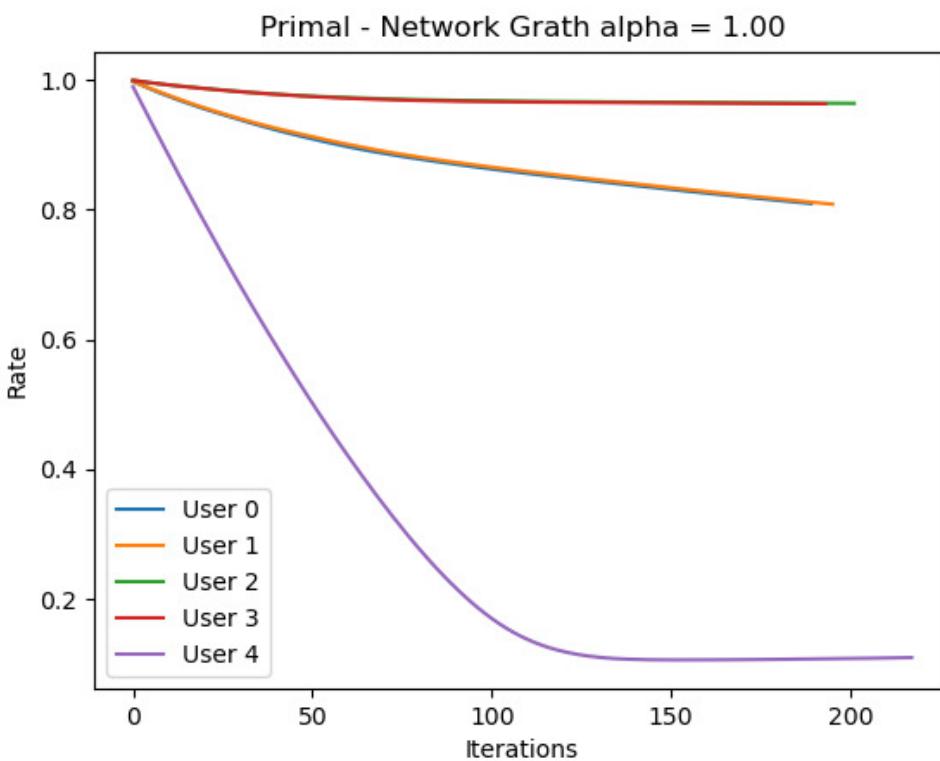
Figure 12 Dual Algorithm Run for NUM with  $L=5$ ,  $\alpha=2$



```

-----
user = 0, user_state.rate :0.842971966808776
user = 1, user_state.rate :0.8430435720371913
user = 2, user_state.rate :0.8722921621834313
user = 3, user_state.rate :0.8719571818139881
user = 4, user_state.rate :0.814721451569145
-----
```

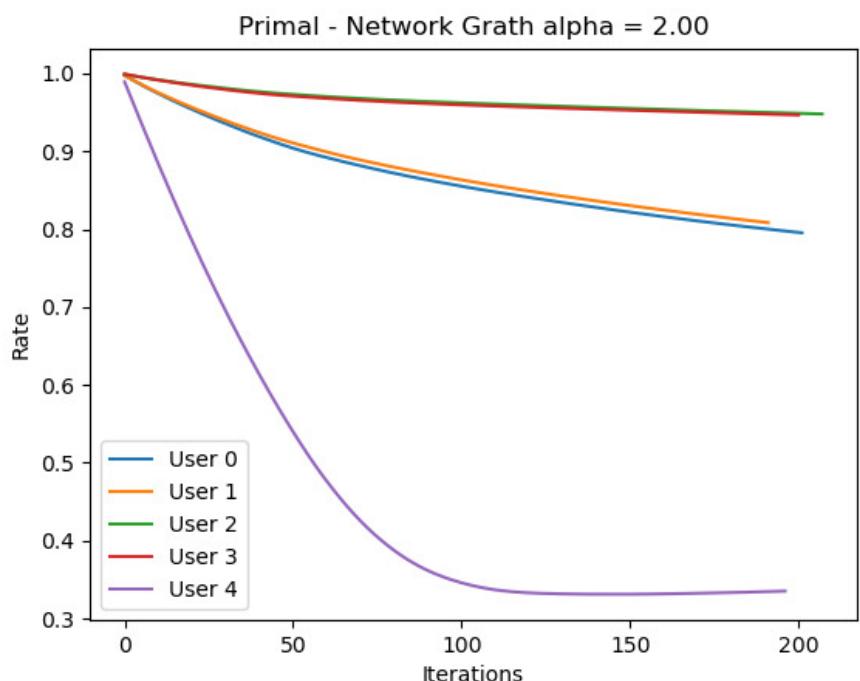
Figure13 Dual Algorithm run for NUM with L=5, alpha=20



```

-----
user = 0, user_state.rate :0.8033589426583145
user = 1, user_state.rate :0.770573120600916
user = 2, user_state.rate :0.9489292807287296
user = 3, user_state.rate :0.9543977235347174
user = 4, user_state.rate :0.11454214365662727
-----
```

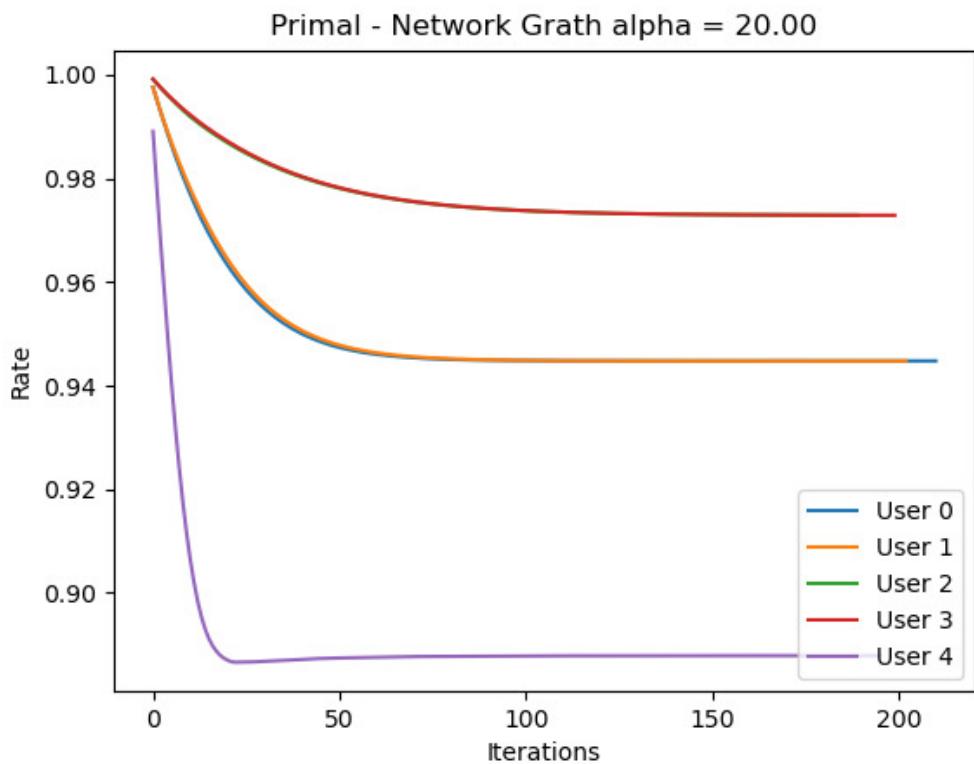
Figure 14 Primal Algorithm for NUM with  $L=5$ ,  $\alpha=1$



```

-----
user = 0, user_state.rate :0.8000211508158362
user = 1, user_state.rate :0.7841373824247351
user = 2, user_state.rate :0.952770111409925
user = 3, user_state.rate :0.9375764816031122
user = 4, user_state.rate :0.3389570852657199
-----
```

*Figure 15 Primal Algorithm run for NUM with L=5, alpha=2*



```

user = 0, user_state.rate :0.9447708840606925
user = 1, user_state.rate :0.9447713303417951
user = 2, user_state.rate :0.9728725112863478
user = 3, user_state.rate :0.9728808343256568
user = 4, user_state.rate :0.8879016415658405
-----
```

Figure16 Primal Algorithm run NUM , L=5, alpha=20

## שאלה 5

חזרו על שאלות 2 ו- 3 כך שהמלולים יוחשבו על ידי אלגוריתם דיקסטרה בשכבה הרשת. תכננו את האלגוריתם כך שיביא את הרשת לנקודת עבודה טובה. נמקו בצירוף תוצאות שקיבלו.

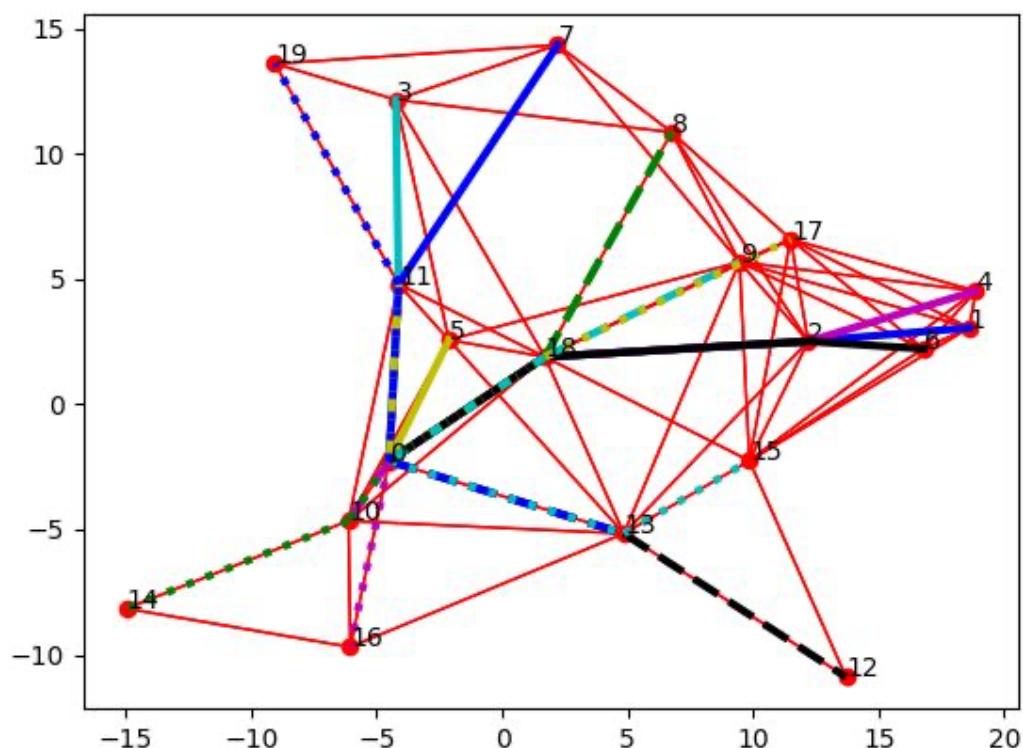
### פתרון

החלנו למש את אלגוריתם הדיקסטרה כך שמלול יבחר לפי מרחק מינימלי ליעד. המשקל של כלリンク הינו המרחק האוקלידי בין שני נקדים.

ubo הרצתה למרחק הכى קצר ליעד

מסלול לדוגמא שנבחר:

הרשת שבה השתמשנו –

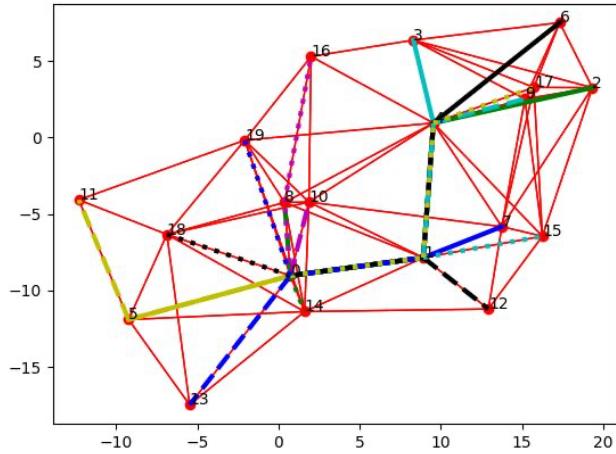


כל צבע או צבע מקווקו מסמל מהמקור ליעד.

– אלו המסלולים עצם, מלינק 0 לכל אחד מהיעדים האחרים (מוחשב לפי כמות hop'ים הכי קטן) –

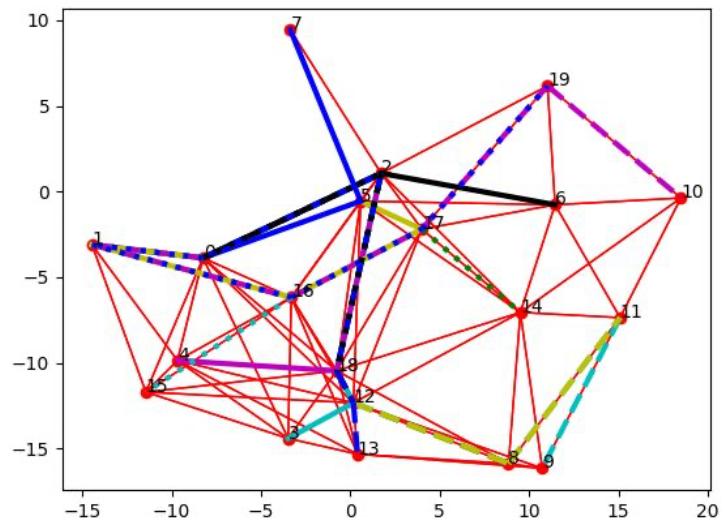
```
dijkstra from node 0
(source:0 ,destination:1 ,Weight:24.412438713262357)
Way: 0 --> 18 --> 2 --> 1
(source:0 ,destination:2 ,Weight:17.97649248933269)
Way: 0 --> 18 --> 2
(source:0 ,destination:3 ,Weight:14.43014946993573)
Way: 0 --> 11 --> 3
(source:0 ,destination:4 ,Weight:24.953029378568754)
Way: 0 --> 18 --> 2 --> 4
(source:0 ,destination:5 ,Weight:5.392413579956098)
Way: 0 --> 5
(source:0 ,destination:6 ,Weight:22.57809344137852)
Way: 0 --> 18 --> 2 --> 6|
(source:0 ,destination:7 ,Weight:18.526152542762055)
Way: 0 --> 11 --> 7
(source:0 ,destination:8 ,Weight:17.76017268941815)
Way: 0 --> 18 --> 8
(source:0 ,destination:9 ,Weight:16.099779408672035)
Way: 0 --> 18 --> 9
(source:0 ,destination:10 ,Weight:2.881196893019436)
Way: 0 --> 10
(source:0 ,destination:11 ,Weight:7.073259009926197)
Way: 0 --> 11
(source:0 ,destination:12 ,Weight:20.331172752300866)
Way: 0 --> 13 --> 12
(source:0 ,destination:13 ,Weight:9.78822681012882)
Way: 0 --> 13
(source:0 ,destination:14 ,Weight:12.402093940517187)
Way: 0 --> 10 --> 14
(source:0 ,destination:15 ,Weight:15.569964163474216)
Way: 0 --> 13 --> 15
(source:0 ,destination:16 ,Weight:7.578445704523273)
Way: 0 --> 16
(source:0 ,destination:17 ,Weight:18.326889462452787)
Way: 0 --> 18 --> 17
(source:0 ,destination:18 ,Weight:7.470443542900191)
Way: 0 --> 18
(source:0 ,destination:19 ,Weight:17.153086196703278)
Way: 0 --> 11 --> 19
```

## ubo הרצה עם כמות מינימלית של Links/מינימום HOPS



```
dijkstra from node 0
(source:0 ,destination:1 ,Weight:1)
Way: 0 --> 1
(source:0 ,destination:2 ,Weight:3)
Way: 0 --> 1 --> 4 --> 2
(source:0 ,destination:3 ,Weight:3)
Way: 0 --> 1 --> 4 --> 3
(source:0 ,destination:4 ,Weight:2)
Way: 0 --> 1 --> 4
(source:0 ,destination:5 ,Weight:1)
Way: 0 --> 5
(source:0 ,destination:6 ,Weight:3)
Way: 0 --> 1 --> 4 --> 6
(source:0 ,destination:7 ,Weight:2)
Way: 0 --> 1 --> 7
(source:0 ,destination:8 ,Weight:1)
Way: 0 --> 8
(source:0 ,destination:9 ,Weight:3)
Way: 0 --> 1 --> 4 --> 9
(source:0 ,destination:10 ,Weight:1)
Way: 0 --> 10
(source:0 ,destination:11 ,Weight:2)
Way: 0 --> 5 --> 11
(source:0 ,destination:12 ,Weight:2)
Way: 0 --> 1 --> 12
(source:0 ,destination:13 ,Weight:1)
Way: 0 --> 13
(source:0 ,destination:14 ,Weight:1)
Way: 0 --> 14
(source:0 ,destination:15 ,Weight:2)
Way: 0 --> 1 --> 15
(source:0 ,destination:16 ,Weight:2)
Way: 0 --> 8 --> 16
(source:0 ,destination:17 ,Weight:3)
Way: 0 --> 1 --> 4 --> 17
(source:0 ,destination:18 ,Weight:1)
Way: 0 --> 18
(source:0 ,destination:19 ,Weight:1)
Way: 0 --> 19
```

בחירה מסלולים כאשר המשקלים הם קיבולים, ורצוים קיבול מקסימלי



```

dijkstra from node 0
(source:0 ,destination:1 ,Weight:9)
Way: 0 --> 1
(source:0 ,destination:2 ,Weight:8)
Way: 0 --> 2
(source:0 ,destination:3 ,Weight:8)
Way: 0 --> 2 --> 18 --> 12 --> 3
(source:0 ,destination:4 ,Weight:8)
Way: 0 --> 2 --> 18 --> 4
(source:0 ,destination:5 ,Weight:8)
Way: 0 --> 1 --> 16 --> 17 --> 5
(source:0 ,destination:6 ,Weight:8)
Way: 0 --> 2 --> 6
(source:0 ,destination:7 ,Weight:6)
Way: 0 --> 5 --> 7
(source:0 ,destination:8 ,Weight:7)
Way: 0 --> 2 --> 18 --> 12 --> 8
(source:0 ,destination:9 ,Weight:7)
Way: 0 --> 2 --> 18 --> 12 --> 8 --> 11 --> 9
(source:0 ,destination:10 ,Weight:8)
Way: 0 --> 1 --> 16 --> 17 --> 19 --> 10
(source:0 ,destination:11 ,Weight:7)
Way: 0 --> 2 --> 18 --> 12 --> 8 --> 11
(source:0 ,destination:12 ,Weight:8)
Way: 0 --> 2 --> 18 --> 12
(source:0 ,destination:13 ,Weight:7)
Way: 0 --> 2 --> 18 --> 12 --> 13
(source:0 ,destination:14 ,Weight:8)
Way: 0 --> 1 --> 16 --> 17 --> 14
(source:0 ,destination:15 ,Weight:8)
Way: 0 --> 1 --> 16 --> 15
(source:0 ,destination:16 ,Weight:8)
Way: 0 --> 1 --> 16
(source:0 ,destination:17 ,Weight:8)
Way: 0 --> 1 --> 16 --> 17
(source:0 ,destination:18 ,Weight:8)
Way: 0 --> 2 --> 18
(source:0 ,destination:19 ,Weight:8)
Way: 0 --> 1 --> 16 --> 17 --> 19

```

הקבולים שהגRELנו:

(16, 0) 2	(12, 0) 1	(5, 0) 2	(0, 1) 9
(16, 1) 2	(12, 3) 10	(5, 2) 7	(0, 2) 8
(16, 2) 1	(12, 4) 5	(5, 6) 2	(0, 3) 4
(16, 3) 4	(12, 5) 1	(5, 7) 6	(0, 4) 5
(16, 4) 3	(12, 8) 7	(5, 12) 1	(0, 5) 7
(16, 5) 3	(12, 9) 1	(5, 14) 7	(0, 12) 5
(16, 12) 5	(12, 13) 7	(5, 16) 2	(0, 15) 6
(16, 13) 2	(12, 14) 7	(5, 17) 1	(0, 16) 5
(16, 15) 10	(12, 15) 2	(5, 18) 3	(0, 18) 7
(16, 17) 10	(12, 16) 10	(6, 2) 10	(1, 0) 3
(16, 18) 8	(12, 17) 1	(6, 5) 5	(1, 4) 6
(17, 2) 1	(12, 18) 4	(6, 10) 3	(1, 15) 2
(17, 5) 10	(13, 3) 9	(6, 11) 4	(1, 16) 8
(17, 6) 8	(13, 4) 2	(6, 14) 5	(2, 0) 2
(17, 12) 5	(13, 8) 3	(6, 17) 6	(2, 5) 7
(17, 14) 9	(13, 9) 1	(6, 19) 5	(2, 6) 8
(17, 16) 5	(13, 12) 6	(7, 2) 4	(2, 7) 1
(17, 18) 5	(13, 16) 4	(7, 5) 6	(2, 14) 1
(17, 19) 8	(13, 18) 1	(8, 9) 2	(2, 16) 3
(18, 0) 9	(14, 2) 10	(8, 11) 10	(2, 17) 7
(18, 2) 2	(14, 5) 8	(8, 12) 8	(2, 18) 8
(18, 3) 6	(14, 6) 1	(8, 13) 6	(2, 19) 6
(18, 4) 10	(14, 8) 4	(8, 14) 5	(3, 0) 4
(18, 5) 2	(14, 9) 1	(8, 18) 5	(3, 4) 7
(18, 8) 6	(14, 10) 2	(9, 8) 8	(3, 12) 6
(18, 12) 9	(14, 11) 5	(9, 11) 6	(3, 13) 4
(18, 13) 4	(14, 12) 8	(9, 12) 3	(3, 15) 4
(18, 14) 8	(14, 17) 8	(9, 13) 5	(3, 16) 1
(18, 15) 10	(14, 18) 9	(9, 14) 1	(3, 18) 4
(18, 16) 2	(15, 0) 1	(10, 6) 9	(4, 0) 8
(18, 17) 9	(15, 1) 1	(10, 11) 6	(4, 1) 3
(19, 2) 2	(15, 3) 7	(10, 14) 1	(4, 3) 1
(19, 6) 9	(15, 4) 10	(10, 19) 10	(4, 12) 2
(19, 10) 10	(15, 12) 10	(11, 6) 1	(4, 13) 5
(19, 17) 10	(15, 16) 5	(11, 8) 4	(4, 15) 5
	(15, 18) 7	(11, 9) 9	(4, 16) 2
		(11, 10) 10	(4, 18) 10
		(11, 14) 7	

## שאלה 6

עבור שאלה זו  $K=1$ , אך כעת ערכוי ההפראה גדולים מאוד. הסימולטור מקבל את המסלולים על ידי אלגוריתם דיקסטרה בשכבה הרשות.

כעת נתון האלגוריתם הבא להקצאת הקצבים: קיובל כלリンク מחולק על ידי TDMA בין ה flows שעוברים דרכו לפי יחס הדטה ש策יר להעביר. הקצב עבור כל flow יקבע לפי הלינק בעל קצב השירות המינימלי במסלול שלו. השבו את הקצבים המתאפשרים.

בחישוב הקצבים, צרפו גורפים כתלות בפרמטרים שונים עבור מספר המשתמשים, גודל הדטה, וכו' ופרטו.

### פתרון

ראשית, השתמשנו באלגוריתם דיקסטרה בשכבה הרשות כפי שצוין שאלה 5, על מנת לקבל את המסלולים הקצרים ביותר עבור כל הצמתים.

שנית, יצרנו פונקציה בסימולטור שמייצרת רשימה אקרואית של F זרמי מידע (flows) המוגדרים לפי צומת מקור, יעד, כמות DATA הנדרשת לשידור, link\_index capacity ועליהם נרחב בקרוב.

לאחר שהרצינו את דיקסטרה וקיבלו את כל הflows בדרכו עבור כל link כמה rzimim עליו ויחסבנו את סכום כל המידע שעובר על אותו link.

כדי לחשב את הקיבול היחסי עבור flow מסוים העובר בלינק ספציפי, המוחושב לפי כמות הflows גודל הדטה, השתמשנו בנוסחה הבאה:

$$flow_{capacity} = \frac{data_{amount}}{data_{sum}} \cdot link\ capacity$$

Flow – כמות המידע של flow

Link – Data Sum

בצורה זו, כל flow קיבל את capacity היחסית המתאימים לו על הלינק, כאמור באלגוריתם TDMA.

ובסיוף הגדרנו את קצב השידור של כל flow להיות capacity המינימלי שהתקבל עבור המסלול שלו.

לאחר מכן הרצינו על רשות בעלת 5 צמתים:

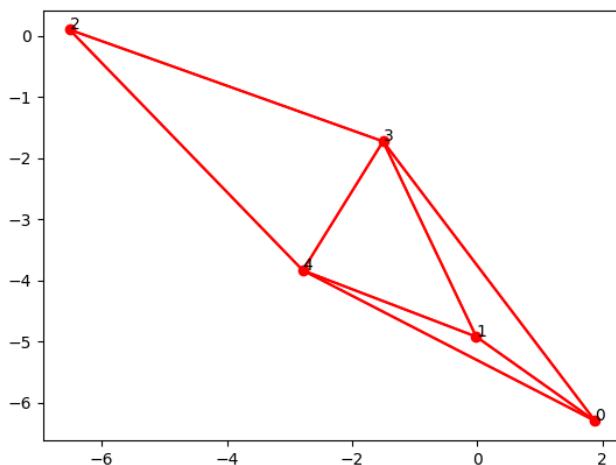


Figure 17: Graph with 5 nodes

הרצינו את האלגוריתם עבור מספר flows בין 5-50 וחשבנו את הקצבים המתקבלים ולאחר מכן שרטטנו גרף עבור הקצב המינימלי שהתקבל כפונקציה של כמות הflows. להלן חישוב הקצבים עבור 5 :flownum=5

```
Flow from 1 to 4 has rate 579.2730673234805 and data amount 238
Flow from 2 to 3 has rate 212.1410806542913 and data amount 811
Flow from 2 to 3 has rate 173.16571565122175 and data amount 662
Flow from 3 to 1 has rate 334.23957366365704 and data amount 870
Flow from 3 to 2 has rate 660.8494396064665 and data amount 768
```

Figure 18 : rate of all flows and their data amount

עבור מספר זה הקצב המינימלי הוא עבור flow 2 ל- 173.16Mbps נצפה לקבל קצב נמוך ככל שכמות הflows גדול, מכיוון שעבור מספר גדול של flows, הקיבול על כל לינק יתחלק ליותר משתמשים וכתוואה לכך כל משתמש יקבל קצב נמוך יותר.

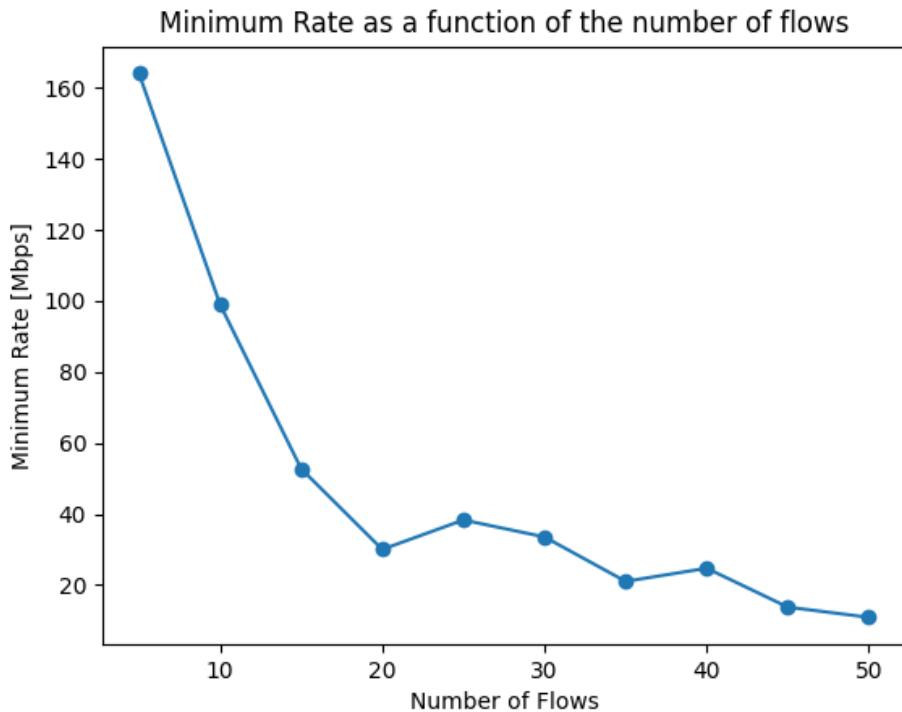


Figure 19 : TDMA algorithm with increasing number of flows

כפי שציפינו, אכן קיבלנו גרף יורך. כמובן, ככל שמספר הflows גדול, כך הקצב של כל משתמש קטן. לאחר מכן רצינו לבדוק עבור לינק מסוים, מה קורה כאשר כמות הדאטा לשידור קטנה רק בflow מסוים וכייזד הוא משפיע על שאר הflows שימושיים באותו הלינק.

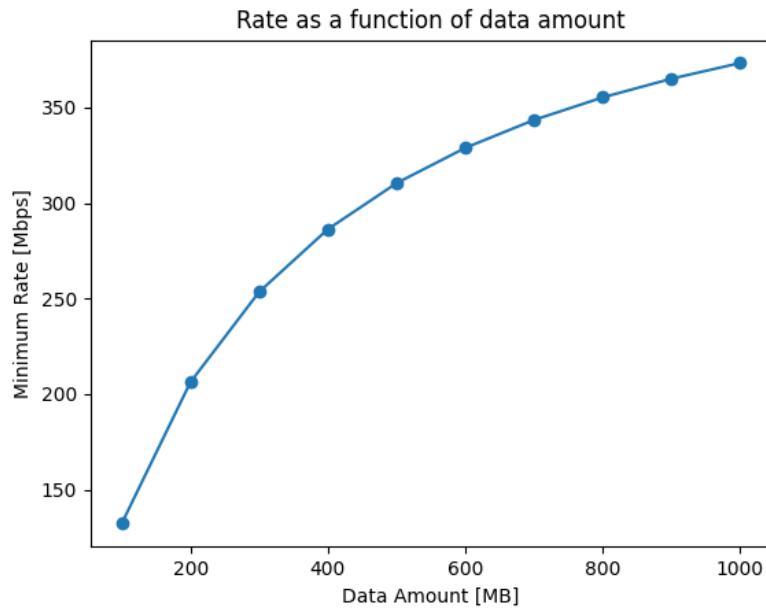


Figure 20 : Rate as a function of data amount for a specific flow (flow 0)

כפי שניתן לראות, קיבלנו שככל שכמות המידע שעברה על flow מסויים גדולה, כך גם גודל הקיבול של אותו flow וכתוצאה מכך הקצב המינימלי גדול. תוצאה זו توأمאת את ציפיותינו. כעת נצפה שעבור flow שונה משמש באותו הLINK, שהקיבול שלו קטן וכתוצאה מכך הקצב המינימלי יקטן עוד יותר.

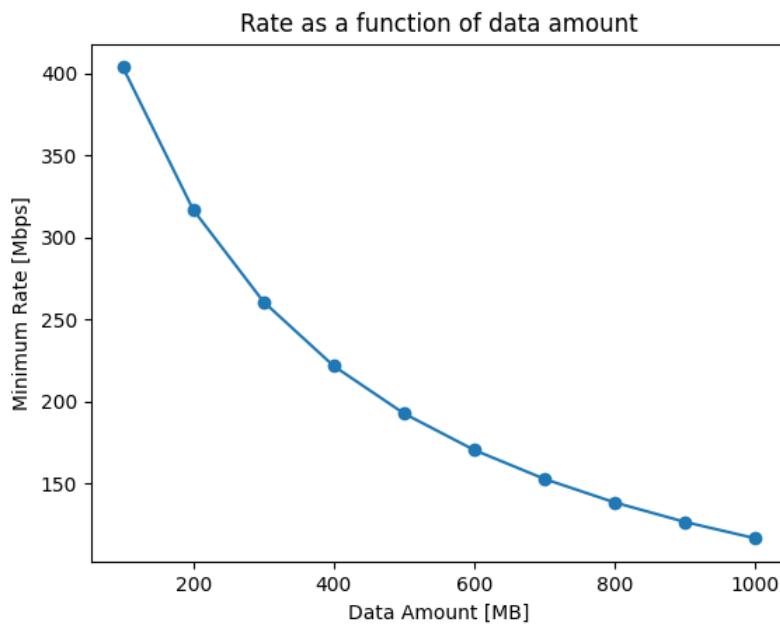


Figure 21: Rate as a function of data amount for a specific flow (flow 1)

כפי שציפינו, קיבלנו גרפ' יורך עבור flow אחר שמשתמש באותו הLINK. מכיוון שהקיבול של flow השני קטן כתוצאה מהגדלת המידע עברו flow הראשון.

## שאלה 7

עבור שאלת זו  $K > 1$ , וגם ערכיו הפרעה גדולים מאוד. הSIMOLUTOR מקבל את המסלולים על ידי אלגוריתם דיקסטרה בשכבה הרשות.

כעת נתון האלגוריתם הבא להקצאת הקצבים: קיבול כלリンク מחולק על ידי TDMA בין ה flows שעוברים דרכו לפי יחס הדאטה שצורך להעברה. הקצב עבור כל flow יקבע לפי הלינק בעל קצב השירות המינימלי במסלול שלו.

ניתן לבחור ערכאים שונים לשידור לאורך המסלול. הציעו אלגוריתם לבחירת ערכאים. חשבו את הקצבים המתקבלים.

בבחירה הקצבים, צרפו גורפים כתלות בפרמטרים שונים עבור מספר המשמשים, מספר ערכאים, גודל הדאטה, וכו' ופרטו.

### פתרון

כעת ביצענו את אלגוריתם TDMA בדומה לשאלה 6 אך כעת עבור מספר ערכאים גבוה מ-1, המחשבנו עבור  $k=1,2,3,4,5$ .

כפי שמתואר הSIMOLUTOR, כל משדר רשאי לשדר באחד מתוך  $k$  ערכאים אורטוגונליים (תדרי גל הנושא מופרדים ספקטרלית ומינימום שידור ברוחב פס צר מספיק כך שאין הפרעה בין שידורים בערכאים שונים). מכיוון שהערכאים אורטוגונליים, נרצה לפזר את העומס על הלינק בין כל הערכאים באופן אחיד, כדי ליעיל את פעולה המערכת.

האלגוריתם שהצענו לבחירת הערכאים פועל כך:

1. אתחול של `flows_num` ו-`data_sum` ל-0 עבור כל link ועבור כל ערך.
2. עבור כל משתמש, נחשב את המסלולים באמצעות אלגוריתם דיקסטרה.
3. אם קיימים מסלולים, עבור כל flow היוצא מהמשתמש, נוסיף את `flow` לכל הלינקים במסלול.
4. עבור הערך הנבחר לכל link, נגדיל את `flows_num` ונוסיף את כמות הנתונים של `flow` ל-`.data_sum`.
5. לאחר שככל ה`flows` עברו, נחשב את הקיבול של כל `flow` כפי שתיארנו בשאלה 6.

כעת סימלצנו על רשת בעלת 20 צמתים בכך לראות את חלוקת המידע לערכאים שונים בצורה הטובה ביותר:

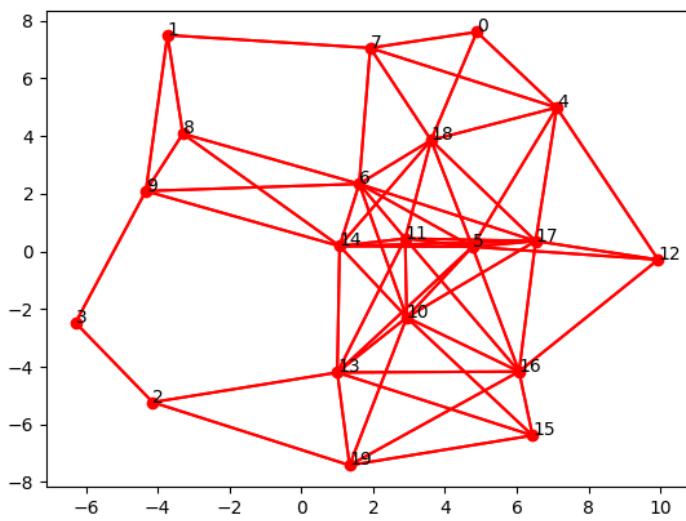


Figure 22 : Graph with 20 nodes

ושרטטנו גרפים של הקצב המינימלי כפונקציה של כמות ה-flows עבור  $k=1,2,3,4,5$

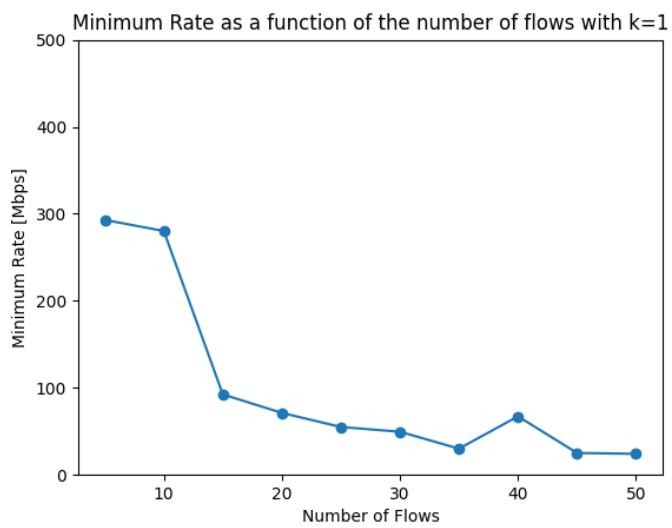


Figure 23 :  $k=1$ , rate as a function of the number of flows

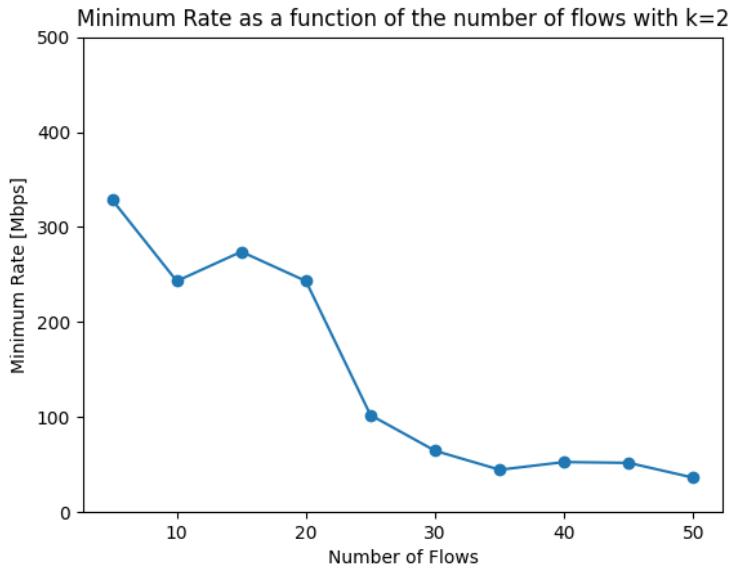


Figure 24:  $k=2$ , rate as a function of the number of flows

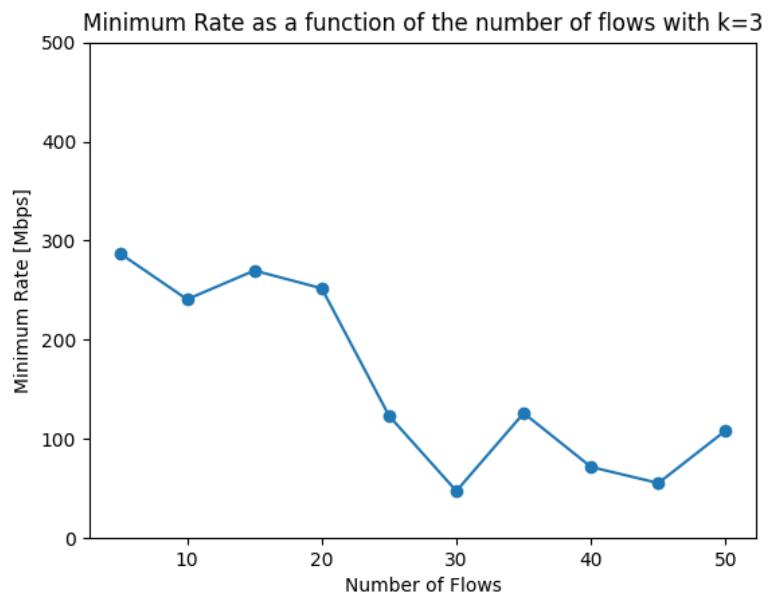


Figure 25:  $k=3$ , rate as a function of the number of flows

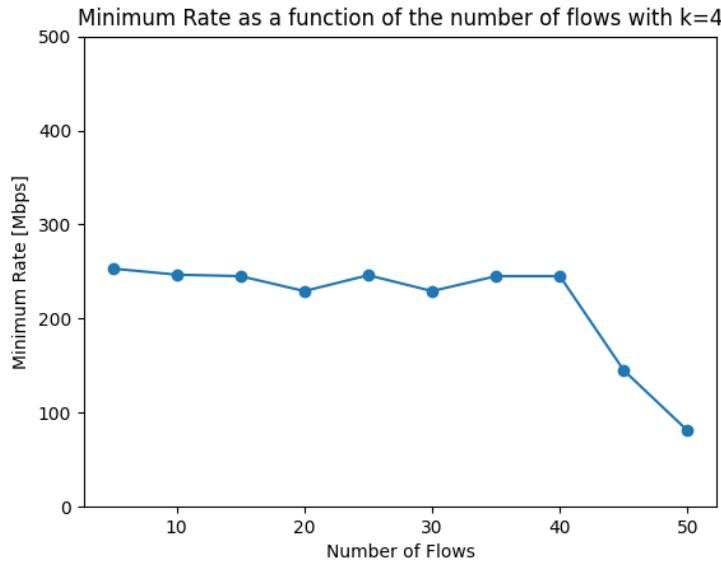


Figure 26:  $k=4$ , rate as a function of the number of flows

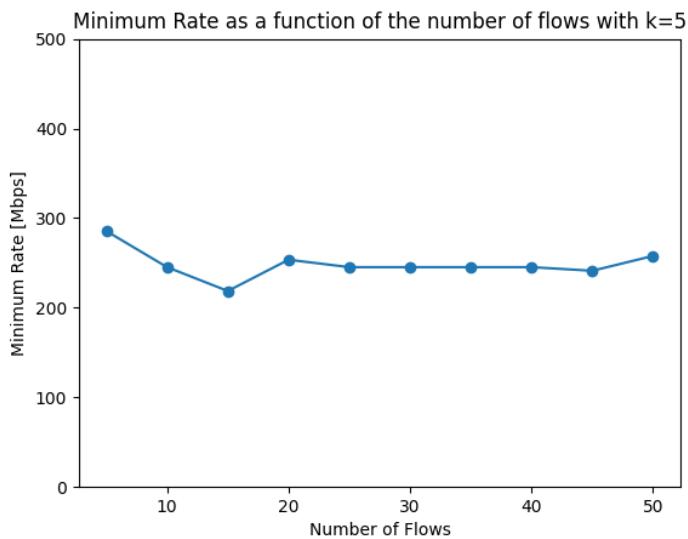


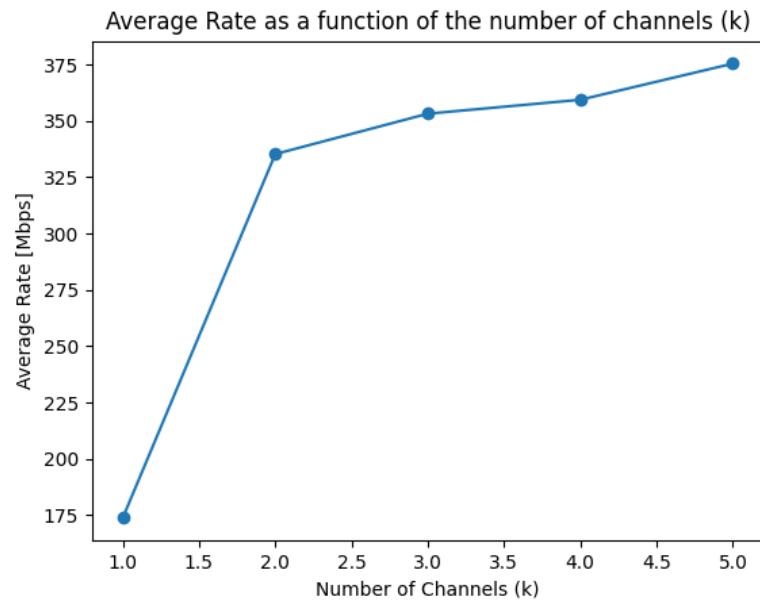
Figure 27:  $k=5$ , rate as a function of the number of flows

נבחן שככל שמספר העروצים גדול, כך העומס מתפלג בזרה שווה יותר על כל לינק.

כלומר, הקצב המינימלי הכללי גדול עד שהוא כמעט שווה בכלם.

במקרה שלנו הקצבים המינימליים הם בקירוב 275Mbps עבור מספר עروצים גובה של 5.

לאחר מכון הדפסנו את הקצב הממוצע כפונקציה של כמות העروצים, נזפה שהקצב יעלה ככל שכמות העروצים גדלה:



*Figure 28 : Avg rate vs k*

כפי שציפינו, אכן ככל שמספר הערוצים גדול, כך העומס מתפלג אחד יותר וכתוצאה מכך הקצב הממוצע עולה.