**הבנת האלגוריתם של Biswas ו-Barman**

**גרף הקנאה:**

שיטת גרף הקנאה עוזרת במקרה שהחפצים בדידים, כלומר לא ניתנים לחלוקה.

במקרה כזה, לא תמיד קיימת חלוקה ללא קנאה (EF), ולכן הוגדר קירוב לחלוקה כזו: חלוקה כמעט ללא קנאה (EF1).

הקצאת EF1 היא הקצאה בה לכל שני סוכנים, קיים חפץ כלשהו שאם נוריד אותו מהסל של השני, אז הראשון לא יקנא בו. במילים אחרות, הקנאה של כל סוכן כלפי כל סוכן אחר מוגבלת בערך המקסימלי של utility, בעיני הראשון, של חפץ בסל של השני.

גרף הקנאה מניח שלכל סוכן יש cardinal utility function על פני חבילת הסחורות. בנוסף, הפונקציה צריכה להיות מונוטונית (במקרה של goods, התועלת על פני קבוצה מסוימת של סחורות היא לפחות התועלת על פני כל תת קבוצה שלה). אבל, הutilities לא חייבות להיות additive (כלומר הפריטים לא חייבים להיות בלתי תלויים).

גרף הקנאה הוא גרף מכוון, כאשר אם i מקנא ב-j, יש קשת מכוונת מ-i ל-j בגרף.

האלגוריתם:

1. מסדרים את החפצים בצורה שרירותית.
2. כל עוד קיימים חפצים שטרם הוקצו:

* נותנים את החפץ לשחקן שאף אחד לא מקנא בו.
* אם אין כזה: סימן שיש מעגל קנאה.

כל עוד יש מעגל קנאה: נחליף סלים במעגל בניגוד לכיוון הקנאה.

* חוזרים ל1.

סיבוכיות זמן הריצה של האלגוריתם: O(mn^3)

האלגוריתם מחזיר בסופו של דבר חלוקה EF1, כי כאשר נותנים חפץ לשחקן שאף אחד לא מקנא בו, לאחר מכן כולם יכולים לקנא בו עד לכל היותר חפץ אחד (הזה).   
בנוסף, כשמסירים מעגל מהגרף, לא פוגעים בתועלות של כל הסוכנים מחוץ למעגל, והתועלות של כל הסוכנים בתוך המעגל משתפרות.

**מיון טופולוגי** (תזכורת):

סידור של קבוצת איברים שקיימות ביניהם תלויות, כך שאף איבר לא יופיע לפני איבר בו הוא תלוי. כלומר אם בגרף מכוון (חסר מעגלים) קיימת קשת בין a ל-b, אז המספר שניתן ל-a בהכרח קטן מהמספר שניתן ל-b.

לכל גרף מכוון חסר מעגלים קיים מיון טופולוגי (אחד לפחות), שניתן למצוא אותו בסיבוכיות של O(|V|+|E|).

במקרה שלנו: גרף הקנאה הוא מכוון חסר מעגלים (כי נפטרנו מהם). המיון הטופולוגי יבטיח שסוכן a אשר מקנא בסוכן b (יש קשת ab בגרף), יהיה לפניו בסידור הסוכנים מחדש, וכך יבחר לפניו פריט שיתווסף לסל שלו בהפעלה הבאה של round-robin algo (על הקטגוריה הבאה).

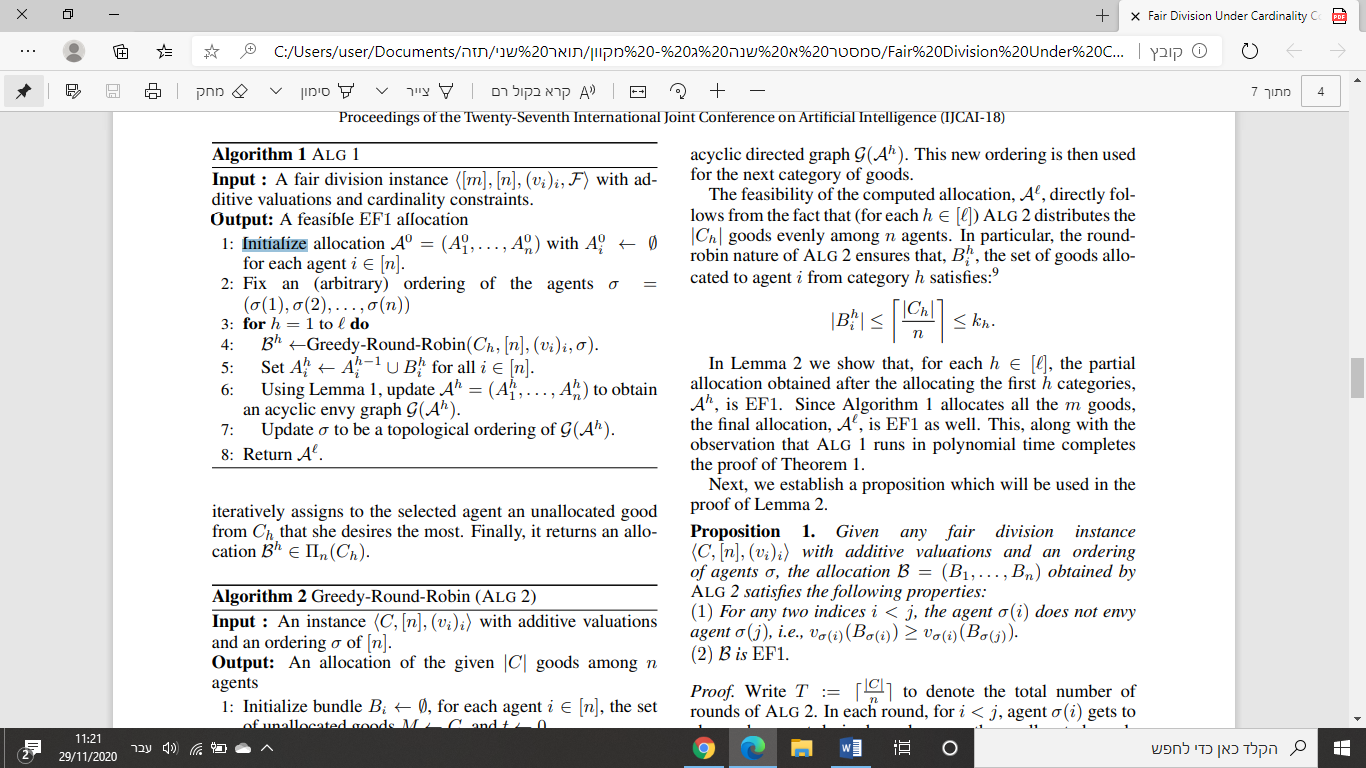
**ALGO 1 של Fair Division Under Cardinality Constraints:**

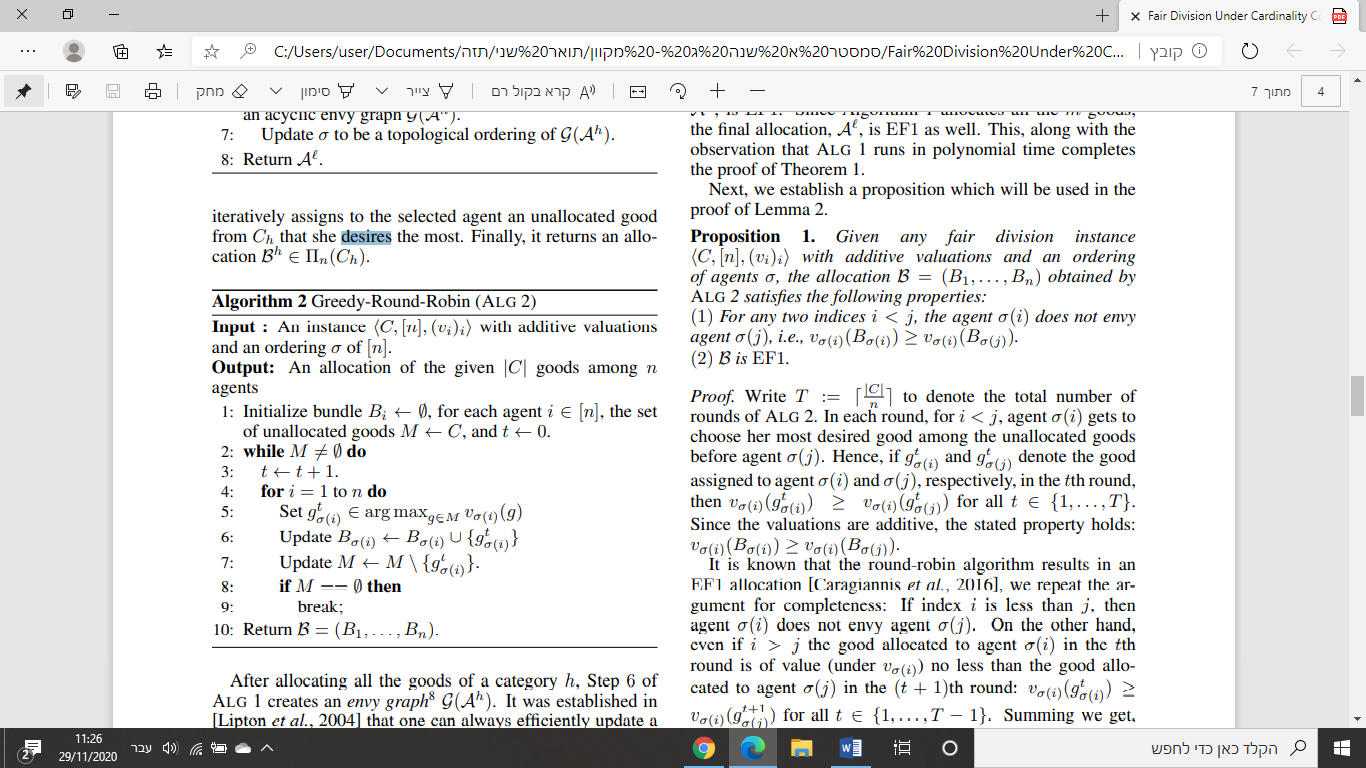
האלגוריתם מקבל את סט הסחורות (indivisible goods) [m], סט הסוכנים [n], ההעדפות של הסוכנים (התועלת של כל סוכן על פני כל תת קבוצה של סחורות) (v\_i)i, ואת קבוצת ההקצאות ה-feasible. כאשר הקצאה A=(A\_1,…,A\_n) היא feasible אמ"ם לכל A\_i ולכל קטגוריה h מתקיים A\_i ∩ C\_h| ≤ k\_h|.   
[כמובן שההקצאה היא לכל n הסוכנים, וכן מחלקים את כל m הסחורות].

כדי להבטיח ש-F לא ריקה, דורשים כי לכל קטגוריה h מתקיים: k\_h ≥ |C\_h| / n.   
(אם המגבלה תהיה קטנה מהערך הזה, אז לא יהיה ניתן לחלק את כל הסחורות).

האלגוריתם מחזיר הקצאה הוגנת שהיא EF1.

האלגוריתם:





מאתחלים הקצאה בה כל סוכן לא מקבל כלום.

סיגמא מסמלת את הסדר בו נעבור על הסוכנים באלגוריתם round-robin. בהתחלה קובעים סדר אקראי על פני הסוכנים.

עוברים על כל הקטגוריות, ומפעילים את ALGO 2 כדי למצוא חלוקה של הסחורות מהקטגוריה הספציפית הזו (h) עבור כל סוכן.

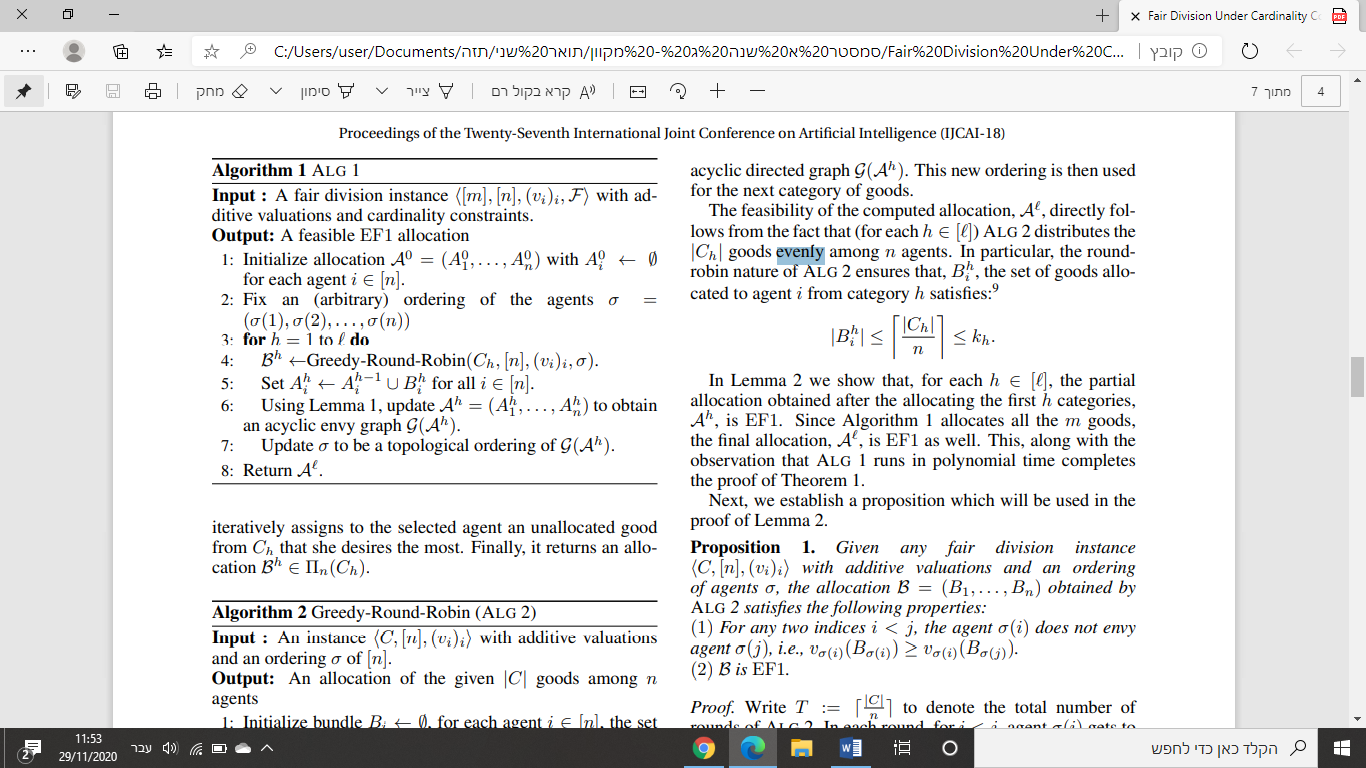
מוסיפים את מה שחזר להקצאה שיש לו עד עכשיו (של הקטגוריות שקטנות מh).

אם בגרף הקנאה יש מעגל, ניפטר ממנו ונעדכן את ההקצאה בהתאם.

לבסוף נעדכן את סיגמא לפי מיון טופולוגי על גרף הקנאה, ונמשיך (עד שנגמרות הקטגוריות).

לבסוף נחזיר את ההקצאה הסופית (חלוקה של כל הקטגוריות לכל הסוכנים).

נכונות:

* ההקצאה המוחזרת היא feasible מכיוון שעבור כל קטגוריה, ALGO 2 מחלק את הפריטים באופן שווה בין n הסוכנים. כלומר, לפי איך ש- round robin עובד, הסל של הסחורות מקטגוריה h עבור סוכן i מקיים:
* ההקצאה הסופית היא EF1.

כל הקצאה של סחורות מקטגוריה מסוימת היא EF1, כי round robin מחזיר הקצאה כזו כשמדובר בהעדפות שהן additive. (כי אם i<j, i בהכרח לא יקנא בj, כי הוא בחר לפניו. ואם i>j, אז בסיבוב 1 הוא לא יקנא במה שj בחר בסיבוב 2, ב-2 לא יקנא ב3 וכן הלאה עד הסוף... כלומר במקסימום הוא יקנא במה ש-j בחר בסיבוב 1, כי לא הייתה לו הזדמנות לבחור לפניו).

לכל h in [l], ההקצאה החלקית של כל הסחורות עד הקטגוריה הh היא EF1, לכן גם עבור h=l (מה שמוחזר), ההקצאה הזו היא EF1. הוכחה באינדוקציה על h.

* אם בהקצאה עד הקטגוריה הh סוכן a מקנא בסוכן b, אז לפי המיון הטופולוגי a יקבל כעת אינדקס קטן משל b, ואז לפי מה שמובטח ב ALGO 2, בהכרח בחלוקת הסחורות של הקטגוריה הh+1 בלבד, a לא יקנא בb. לכן הוא במקסימום ימשיך לקנא בb בגלל הסחורה g הזו (שהוא קינא בו בגללה מקודם).

דוגמת הרצה:

טבלת ההעדפות (התועלות) של הסוכנים על פני כל פריט.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | פריט 1 | פריט 2 | פריט 3 | פריט 4 | פריט 5 | פריט 6 |
| סוכן 1 | 10 | 5 | 0 | 2 | 2 | 3 |
| סוכן 2 | 8 | 6 | 5 | 3 | 7 | 8 |
| סוכן 3 | 10 | 3 | 1 | 4 | 6 | 4 |

קטגוריה 1 – הפריטים הכחולים. אפשר לקבל מקסימום 1.

קטגוריה 2 – הפריטים הכתומים. אפשר לקבל מקסימום 2.

--- נפעיל את ALGO 1:

אתחול:

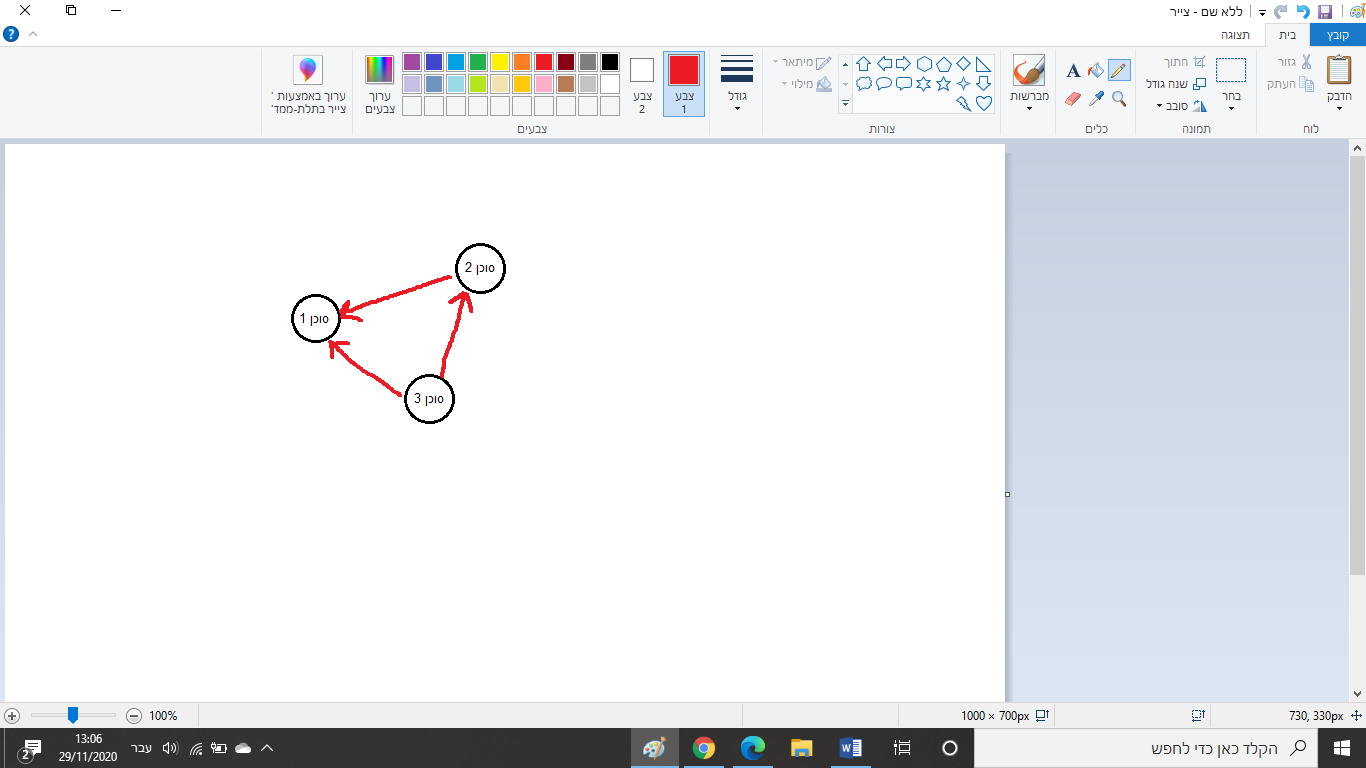
|  |  |
| --- | --- |
| הסוכנים | הפריטים שלהם עד כה |
| סוכן 1 | None |
| סוכן 2 | None |
| סוכן 3 | None |

נקבע את סיגמא להיות: {1,2,3}.

קטגוריה 1 – סוכן 1 מקבל את פריט 1, סוכן 2 מקבל את פריט 3.

|  |  |
| --- | --- |
| הסוכנים | הפריטים שלהם עד כה |
| סוכן 1 | 1 |
| סוכן 2 | 3 |
| סוכן 3 | None |

גרף הקנאה:



הוא חסר מעגלים. נעדכן את סיגמא: {3,2,1}

קטגוריה 2 – סוכן 3 מקבל את 5, סוכן 2 מקבל את 2, סוכן 1 מקבל את 6, סוכן 3 מקבל את 4.

|  |  |
| --- | --- |
| הסוכנים | הפריטים שלהם עד כה |
| סוכן 1 | 1,6 |
| סוכן 2 | 3,2 |
| סוכן 3 | 5,4 |

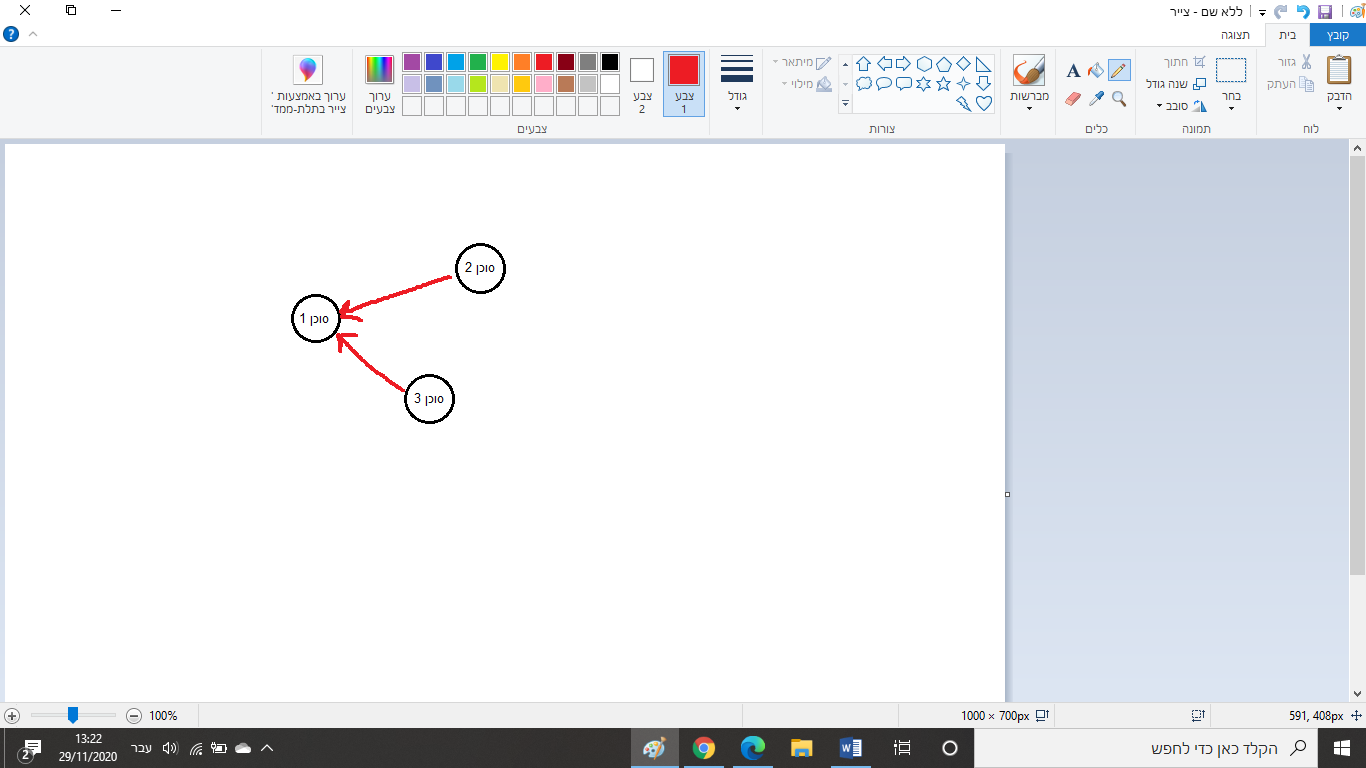
זו ההקצאה הסופית כי סיימנו עם כל הפריטים בכל הקטגוריות.

בעיני סוכן 1: הסל שלו שווה 13, הסל של 2 שווה 5, והסל של 3 שווה 4. הוא לא מקנא.

בעיני סוכן 2: הסל שלו שווה 11, הסל של 1 שווה 16 (מקנא בו), והסל של 3 שווה 10.

בעיני סוכן 3: הסל שלו שווה 10, הסל של 1 שווה 14 (מקנא בו), והסל של 2 שווה 4.

גרף הקנאה:



אם סוכן 2 יוריד ל-1 מהסל את פריט 1 הוא כבר לא יקנא בו. כך גם סוכן 3. לכן זה EF1.

בנוסף, ההקצאה עומדת בcardinality constraints.

(הערה: אם הגרף הזה היה עם מעגלים, היינו מסירים אותם).

**בgoogle doc יש את השינוי של האלגוריתם לגבי מטלות.**