

**פרויקט מסכם לתואר בוגר במדעים (B.Sc)
במתמטיקה שימושית**

שיטת ההפרה המינימלית לדירוג קבוצות ספורט

עמאשה באסל

**Minimum Violation Method For Ranking Sports
Teams**

Amasha Basel

**פרויקט מסכם לתואר בוגר במדעים (B.Sc)
במתמטיקה שימושית**

שיטת ההפרה המינימלית לדירוג קבוצות ספורט

עמאשה באסל

**Minimum Violation Method For Ranking Sports
Teams**

Amasha Basel

Advisor:
Assoc.Prof.Haggai Katriel

מנחה:
פרופ' כתריאל חגי

Karmiel

2020

כרמיאל

שיטת ההפרה המינימלית לדירוג קבוצות ספורט

עמאשה באסל

30 בדצמבר 2020

תוכן עניינים

4	מבוא	1
5	שיטת ההפרה המינימלית (MVR)	2
5	2.1 הגדרת מושגים חשובים	
5	2.2 דוגמאות	
	2.3 הגדרה פורמלית של שיטת ההפרה המינימלית לדירוג קבוצות ספורט	
6		
7	3 אלגוריתם חמדן-אקראי לקירוב MVR	3
7	3.1 שיטת העבודה של האלגוריתם	
7	3.1.1 שיטה בסיסית:	
7	3.1.2 שיטת מתקדמת:	
8	3.2 הוכחה פרקטית לנכונות האלגוריתם	
9	3.3 דוגמאות לעבודת האלגוריתם עם ניתונים שונים	
10	3.3.1 שיטת MVR בסיסית (דירוג התחלתי רנדומלי):	
	3.3.2 שיטת MVR מתקדמת (דירוג התחלתי אינו רנדומלי):	
14		
19	4 תוצאות שיטת MVR על נתונים אמיתיים	4
20	4.1 הפעלת שיטת ההפרה המינימלית על ניתונים אמיתיים	
21	4.2 השוואה בין שיטה בסיסית למתקדמת	
21	4.3 השוואה לשיטת הדירוג שמשתמשים בה היום	
22	5 סיכום	
22	6 רשימת מקורות	
23	7 נספחים	

תקציר

בעבודה זו נדון בבעיית דירוג קבוצות או שחקנים בטורניר כלשהו, על בסיס מערך השוואות שנעשות בין כל זוג קבוצות בנפרד. הקריטריון הכי חשוב בשיטות מסוג הזה, הוא שיהיו כמה שפחות מקרים שבהם קבוצה א מדורגת מעל קבוצה ב בזמן שקבוצה ב ניצחה את קבוצה א במשחק שהיה ביניהן בטורניר, מצב כזה נקרא הפרה (violation) בדירוג, לכן המטרה בשיטה שאנחנו מפתחים ומגדירים בעבודה היא למצוא את הדירוג בעל מספר הפרות (violation) מינימלי. לשיטה הזאת קוראים שיטת ההפרה המינימלית (MVR), בעבודה זו אנו מפתחים אלגוריתם מינימיזציה סטוכסטי כדי לחפש דירוג בעל מספר הפרות מינימלי, אחר כך אנו מדגימים (באופן פרקטי) שהאלגוריתם פועל

היטב ובסוף מנסים להפעיל את האלגוריתם על נתונים של הליגה הספרדית
לכדורגל ומשווים בין הדירוג שנקבל לדירוג הרשמי של הליגה.

1 מבוא

בעיית דירוג קבוצות ספורט בטורניר היא בעייה מפורסמת שנחקרה על ידי הרבה מדענים בעשרות השנים האחרונות, כאשר כל אחד מהם ניסה למצוא את שיטת הדירוג הכי טובה לפי דעתו לדירוג קבוצות או מספר שחקנים שמתחרים בטורניר כלשהו. מרבית השיטות שפותחו בעבר לדירוג היו מבוססות על "וקטור כוח" כלשהו או כל גרסה אחרת שלו, וקטור המציין את החוזק הכללי של כל קבוצה בודדת ביחס לכל השחקנים האחרות. בעבודה זו אנו נדון בשיטה הכי יעילה והכי מדויקת לדירוג קבוצות שנקראת שיטת ההפרה המינימלית שבעיקר היא מבוססת על תוצאת המשחקים בין כל שתי קבוצות בנפרד באותו הטורניר.

קיימות הרבה שיטות ידועות לדירוג קבוצות ספורט, כל שיטה יש לה את הקריטריון והרעיון מאחוריה, לכן בהינתן טבלת תוצאות של משחקים בין קבוצות, כל שיטה נותנת לנו דירוג אחר.

שיטת Massey: ניתן לסכם את הרעיון מאחורי שיטת Massey במשוואה עיקרית אחת $R_i - R_j = Y_k$, כאשר Y_k זה שולי ניצחון עבור משחק k , ו R_i ו R_j אלה הדירוגים של הקבוצות i, j . בהתאם. המשוואה מניחה, ששולי הניצחון במשחק כלשהו זה הפרש הדירוגים של שתי הקבוצות היריבות. [3]

שיטת Colley: בשנת 2001 כתב ד"ר ווסלי קולי, אסטרופיזיקאי בהכשרתו מאמר על השיטה החדשה שלו לדירוג קבוצות ספורט. הפרוייקט הצדדי שלו הפך לפרוייקט מוצלח מאוד בדומה לזה של Massey, והוא בתורו גם שימושי מאוד היום בדירוג קבוצות כדורגל של מכללות בארה"ב. השיטה שלו, שנקראת שיטת הדירוג של קולי, מתבססת על הרעיון הכי פשוט והכי ישן שיש בתחום של דירוג קבוצות, וזהו הסיכויים של הניצחון לכל קבוצה. [3]

כל שיטה יש לה את הקריטריון והרעיון מאחוריה, לכן בהינתן טבלת תוצאות של משחקים בין קבוצות, ורמת החוזק של הקבוצות, כל שיטה נותנת לנו דירוג אחר.

בעיית דירוג קבוצות המתחרות בליגה נחקרה על ידי חוקרים רבים בעשורים האחרונים, למשל Bent ו Ford and johnston, והרבה חוקרים אחרים. בשנות ה-60 הרבה חוקרים כמו קנדל ו גודארד התמקדו בבעיית מציאת דירוגים בעלי מספר הפרות מינימלי, ובסוף הביאו את הפתרון המוסבר בעבודה הזאת, שיטת ההפרות המינימלית (MVR) המפורסמת. [1]

בפרק 2 נגדיר את שיטת ההפרה המינימלית באופן פורמלי, נביא דוגמאות, נגדיר מושגים חשובים, ונציג את שיטת החישוב של ההפרה עבור דירוג כלשהו.

בפרק 3 נגדיר את שיטת העבודה של האלגוריתם החמדן-אקראי שלנו, בדיקת תפקוד האלגוריתם והוכחה פרקטית ליעילותו, אחר כך נעסוק בדוגמאות שונות עם מספר קבוצות שונה.

בפרק 4 נפעיל את שיטת ההפרה המינימלית על תוצאות הליגה הספרדית לכדורגל (תוצאות אמיתיות), ונשווה עם שיטת הדירוג שמתמשים בה בליגה הספרדית, וגם עם שיטת הדירוג לפי מספר הניצחונות.

בפרק 5 נערוך סיכום קצר, ונציג מסקנות.

2 שיטת ההפרה המינימלית (MVR)

2.1 הגדרת מושגים חשובים

הפרה (violation): ההפרה מתרחשת כאשר יש לנו מקרים בדירוג שבהם קיים זוג קבוצות א, ב כך שקבוצה א מדורגת מעל קבוצה ב למרות שקבוצה ב ניצחה את קבוצה א במשחק שהיה ביניהן בטורניר.

דירוג אופטימלי: זהו דירוג שבו $\text{violation} = 0$ (המשתנה violation מייצג את מספר ההפרות בדירוג נתון).

דירוג עם הפרה מינימלית: זהו דירוג, שבהינתן תוצאות של משחקים בין קבוצות בטורניר כלשהו, יש בו מספר הפרות מינימלי מכל הדירוגים האפשריים.

* מטרתנו בשיטה שלנו (MVR) היא למצוא את הדירוג בעל ההפרה המינימלית.

2.2 דוגמאות

דוגמא 1: דוגמה שבה קיים דירוג אופטימלי

בהינתן 5 קבוצות A, B, C, D, E וטבלת תוצאות משחקים:

	A	B	C	D	E
A	-	0-3	0-2	0-1	0-3
B	1-0	-	2-1	3-2	4-3
C	2-0	1-2	-	4-0	5-2
D	1-0	2-3	0-4	-	2-0
E	3-0	3-4	2-5	0-2	-

כלומר קבוצה B ניצחה את כל הקבוצות, קבוצה A הפסידה לכל הקבוצות, קבוצה C ניצחה את הקבוצות D, E וקבוצה D ניצחה את הקבוצה E, במקרה הזה אפשר לראות שקיים דירוג אופטימלי עם violations אפס כלומר בלי הפרות והוא:

B

C

D

E

A

לכן זהו הדירוג עם מספר ההפרות המינימלי וזהו הדירוג האופטימלי עבור הטורניר הנתון.

דוגמא 2:

בהינתן 5 קבוצות A, B, C, D, E וטבלת תוצאות משחקים:

	A	B	C	D	E
A	-	3-0	1-2	2-4	3-1
B	0-3	-	2-1	2-3	4-3
C	2-1	1-2	-	4-0	5-2
D	4-2	3-2	0-4	-	2-0
E	1-3	3-4	2-5	0-2	-

ברור שעבור הטבלה הנתונה, לא קיים דירוג אופטימלי, כי בכל דירוג שנמצא יהיה לפחות זוג קבוצות x, y כך ש x מדורגת מעל y לעומת ש y ניצחה את x , כלומר קיים לפחות violation אחד. אומנם שהבעיה (מציאת דירוג בעל מספר הפרות מינימלי) נראית בעייה קלה וניתן לפתור אותה בכמה שלבים, אך בפועל הבעיה היא בעייה קשה מאד, כלומר אין לה אלגוריתם שפותר אותה בזמן סביר כשמספר הקבוצות גדל. לשם כך, אנחנו צריכים למצוא שיטה היוריסטית, שמנסה למצוא את הדירוג עם מספר ההפרות המינימלית (או הדירוג שמספר ההפרות שלו קרוב למספר ההפרות המינימלי האפשרי) בסיבוכיות זמן סבירה כך שהאלגוריתם יהיה שימושי.

2.3 הגדרה פורמלית של שיטת ההפרה המינימלית לדירוג קבוצות ספורט

בהינתן N קבוצות בטורניר כלשהו, כל קבוצה משחקת פעם אחת נגד קבוצה אחרת בלי תיקו, כלומר Knock-out, וטבלת תוצאות משחקים ביניהן, המטרה שלנו היא למצוא את הדירוג עם מספר הפרות מינימלי. ראינו משתי הדוגמאות הקודמות שלא תמיד קיים דירוג אופטימלי, ולכן אנחנו צריכים למצוא את הדירוג הכי קרוב לדירוג האופטימלי, וכמובן בזמן סביר.

לשם כך בשיטת MVR אנחנו משווים בין זוג קבוצות שנבחר באקראי, אם החלפת המקומות שלהן בדירוג מקטינה את מספר ההפרות אז מחליפים וממשיכים עם זוג אחר, אם לא מקטינה את מספר ההפרות אז לא מחליפים ובחרים זוג אחר.

יש שתי גרסאות לשיטת ה MVR שלנו, והן:

- שיטת MVR בסיסית
- בשיטה הזאת, אנחנו מתחילים מדירוג התחלתי רנדומלי ואז משווים באופן רנדומלי בין הקבוצות, באופן זוגי בנפרד.
- שיטה MVR מתקדמת

השיטה הזאת היא שיטת ה MVR המתקדמת, כלומר יותר יעילה מהבסיסית, היא עובדת באותה צורה אבל בנגד לשיטה הבסיסית, אנחנו לא מתחילים מדירוג התחלתי רנדומלי, אלא מדירוג התחלתי המבוסס על המידע שלנו על רמות החוזק של הקבוצות מעונות קודמות, כלומר אם מהשנים הקודמות ידוע לנו שקבוצה יותר חזקה מקבוצה אחרת, היא תהיה מדורגת מעל לקבוצה החלשה יותר בדירוג ההתחלתי שלנו. אנחנו נראה שהשיטה הזאת מקטינה את מספר ההפרות משמעותית מכיוון שהאלגוריתם שלנו היוריסטי, כלומר הוא רץ מספר קבוע של פעמים ואז עוצר, כי כמו שצויין מקודם הבעיה הינה בעיה קשה מאוד ואין לה אלגוריתם שפותר אותה בסיבוכיות זמן סבירה כאשר מספר הקבוצות גדל.

3 אלגוריתם חמדן-אקראי לקירוב MVR

3.1 שיטת העבודה של האלגוריתם

3.1.1 שיטה בסיסית:

בהינתן מספר קבוצות המשתתפות בטורניר, וטבלת התוצאות של המשחקים שהיו ביניהן:

התוכנית מאתחלת מערך ששומר דירוג רנדומלי של הקבוצות, כלומר אם מספר הקבוצות 20, המערך יאותחל במספרים מ 1 עד 20 בלי חזרות, באופן לא סדרתי, אחר כך אנחנו מגדירים כמה סימולציות לעשות, כלומר כמה פעמים התוכנית תרוץ, בכל ריצה יש לנו דירוג אחר ולוקחים את הדירוג הכי טוב מבין כל הדירוגים כלומר הדירוג בעל הכי פחות violations.

3.1.2 שיטת מתקדמת:

ההבדל המרכזי בין שתי השיטות, זה שבשיטה המתקדמת המערך ההתחלתי הוא לא מאותחל באופן רנדומלי, אך הוא מאותחל בדרך הבאה:

נאתחל מערך, לשם הנוחות (עם ניתונים לא אמיתיים) נניח שאיברי המערך מ-1N הן הקבוצות שלנו, ונניח שהם ממוינים מהכי חזק (1) ולהכי חלש (N), נגדיר סיכוי לכל קבוצה, למשל, קבוצה שבעונות שעברו היתה מדורגת הכי הרבה פעמים מספר 1 יש לה סיכוי 70% להיות מדורגת מספר 1 עוד הפעם, ואם לא אז הסיכוי שהיא תהיה מספר 2 גדל בקבוע כלשהו עד לסיכוי שהיא תהיה מספר N והוא יהיה 100%, כלומר אם היא לא מבין כל ה N-1 הרשאונות אז היא בהחלט מספר N.

נראה בהמשך שהדירוג הזה של הקבוצות מאוד עוזר למצוא דירוג יותר טוב כלומר דירוג עם מספר הפרות יותר קטן מאשר בשיטה הבסיסית, מכיוון שהתוכנית רצה מספר קבוע של פעמים ולא מכסה את כל האפשרויות הנתונות (שזה לוקח בערך 100 שנה), ולכן כל שיפור באתחול המערך ההתחלתי של הקבוצות, יוביל אותנו למספר הפרות התחלתי מינימלי, שעוזר לתוכנית להקטין את מספר ההפרות בצעד הבא.

אחר כך בשתי השיטות:

- התוכנית שולחת את הדירוג הרנדומלי ההתחלתי לפונקציית עזר, שהתפקיד שלה הוא לחשב violations בדירוג, את מספר ה violations אנחנו שומרים במשתנה initial_violations.
- בשלב הזה בוחרים כמה פעמים התוכנית רצה (באותה הסימולציה) על הפונקציה ViolationDec שהתפקיד שלה הוא להפחית violations בדירוג הנתון ככל האפשר על ידי:

1. בחירת זוג קבוצות באופן רנדומלי
2. החלפת מקומות ביניהן בדירוג
3. אם מספר ה-violations קטן, אז נחליף בין זוג הקבוצות ונחזור ל-1, אם גדל אז מחזירים זוג כמו שהיה בהתחלה וחוזרים ל-1

- בסוף יש לנו עבור דירוג רנדומלי ומטריצת תוצאות רנדומלית, בכל סימולציה יש את מספר ההפרות המינימלי שהאלגוריתם מצא בה, אנו נבחר את הדירוג עם מספר ההפרות הכי קטן וזה יהיה הדירוג הכי קרוב לדירוג עם מספר ההפרות המינימלי שאפשר למצוא עבור מטריצת המשחקים.

3.2 הוכחה פרקטית לנכונות האלגוריתם

להוכחת האלגוריתם של השיטה שלנו (MVR), אנחנו נבנה תוכנית מטלב עם טבלת/מטריצת תוצאות רנדומלית אך שיש בה דירוג אופטימלי, אחר כך נפעיל את האלגוריתם שלנו ונראה שהוא כן מוציא את הדירוג עם הכי פחות violations, כלומר במצב שלנו את הדירוג האופטימלי (violations0) :

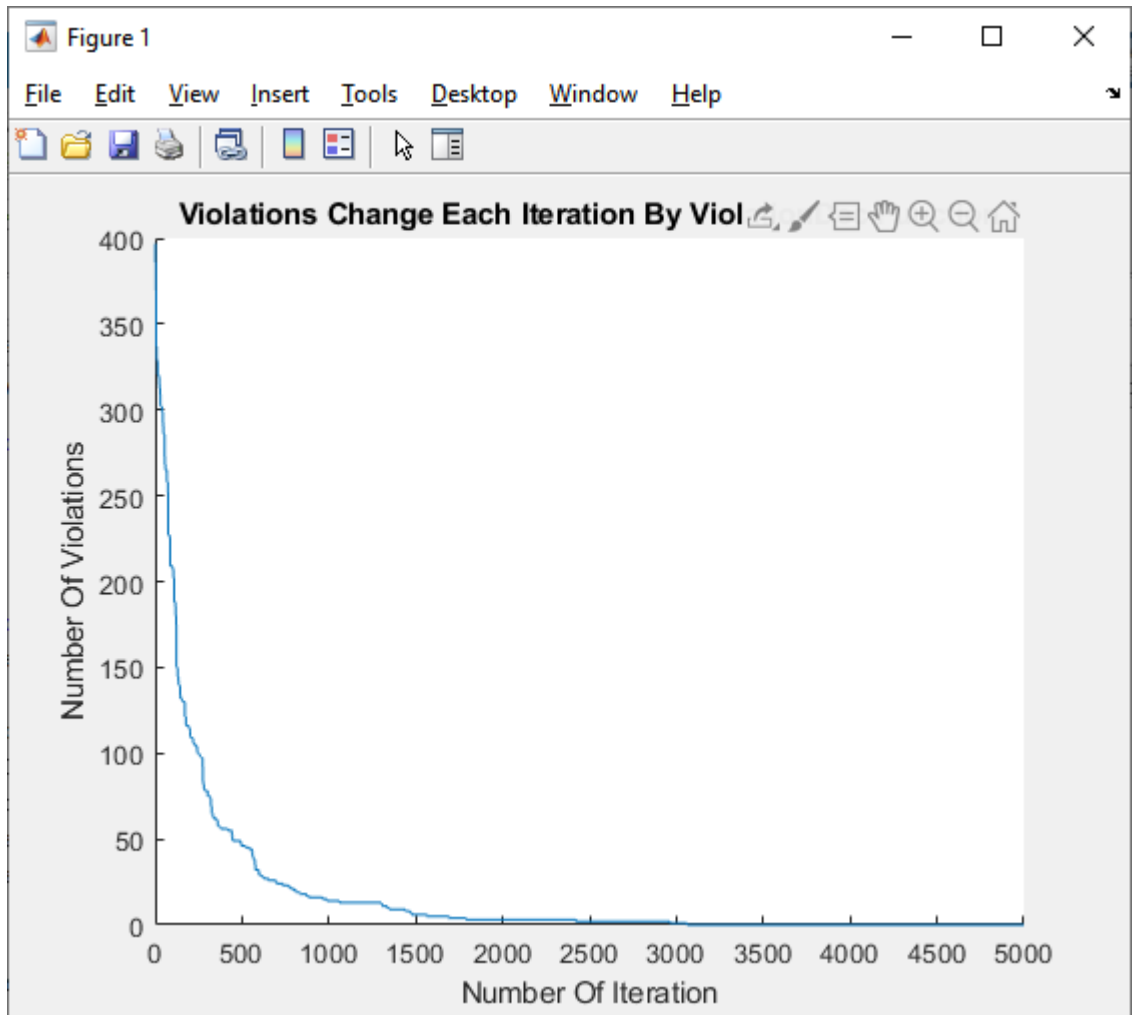
*איך מוצאים מטריצת משחקים כלשהיא שקיים עבורה דירוג אופטימלי בהכרח?

$$\text{TeamsMatrix}[i,j]=\begin{cases} 1 & \text{if } i > j \\ 0 & \text{else} \end{cases} \text{ : לוקחים את המטריצה הבאה:}$$

כלומר, מי שחזק יותר מנצח בסיכוי של 100%, בהגדרה הזאת לא יתקיים במטריצה שום מצב של : A מנצחת את B, B מנצחת את C, C מנצחת את A.

אחר כך אנחנו מתחילים מדירוג רנדומלי כלשהוא ועוברים על שלבי האלגוריתם כמו שהוסבר בפרק 3.1, ורואים שהאלגוריתם כן מוצא את הדירוג האופטימלי. התוצאות שהתקבלו:

מספר ההפרות המינימלי שהגענו אליו הוא `MinimumViolationsAccomplished=0` :



אנחנו יכולים לראות מהגרף שמספר ההפרות ההתחלתי עבור הדירוג הרנדומלי ומטריצת התוצאות שבנינו להוכחה הפרקטית, היה כ-400, ואחרי 4700 איטרציות בערך בפונקציה ViolationDec שהיא מפחיתה מספר violations בדירוג נתון כמו שהוסבר מקודם, הגענו לדירוג עם מספר violations 0 שזה הוא הדירוג האופטימלי, ולכן התוכנית והאלגוריתם שלנו עובדים בצורה נכונה.

3.3 דוגמאות לעבודת האלגוריתם עם ניתונים שונים

- משום שעדיין אנחנו לא בשלב של ניסיון על נתונים אמיתיים, התוכנית מאתחלת באופן רנדומלי מטריצה $N \times N$, כך ש N הוא מספר הקבוצות בטורניר (שאנחנו בוחרים), באופן הבא:

$$\text{TeamsMatrix}[i,j] = \begin{cases} 1 & \text{if team } i \text{ won against team } j \\ 0 & \text{else} \end{cases}$$

- בבדיקת האלגוריתם ובדוגמאות, אנחנו קובעים דירוג "אמיתי" שמשמש ליצירת הנתונים המסומלים שניתנים לאלגוריתם.

כלומר, המטריצה תאותחל במספרים 1 או 0 בלבד, מספר 1 בתא $[i,j]$ פירושו שקבוצה i ניצחה את קבוצה j במשחק שהיה ביניהן, ומספר 0 אומר שקבוצה j ניצחה את קבוצה i , לכן אחרי זה התוכנית עוברת על המטריצה פעם שנייה, ובכל תא $[i,j]$ שיש בו מספר 1, התוכנית שמה 0 בתא $[j,i]$, כי אם קבוצה i ניצחה את קבוצה j , קבוצה j הפסידה מול קבוצה i , ולכן לפי הגדרת המטריצה שלנו צריך שהתא $[j,i]$ יהיה בו 0, ובסוף התוכנית עוברת עוד פעם על המטריצה ושמה 0 בכל איברי האלכסון הראשי, משום שקבוצה לא יכולה לשחק עם עצמה.

3.3.1 שיטת MVR בסיסית (דירוג התחלתי רנדומלי) :

- **דוגמא 1:**

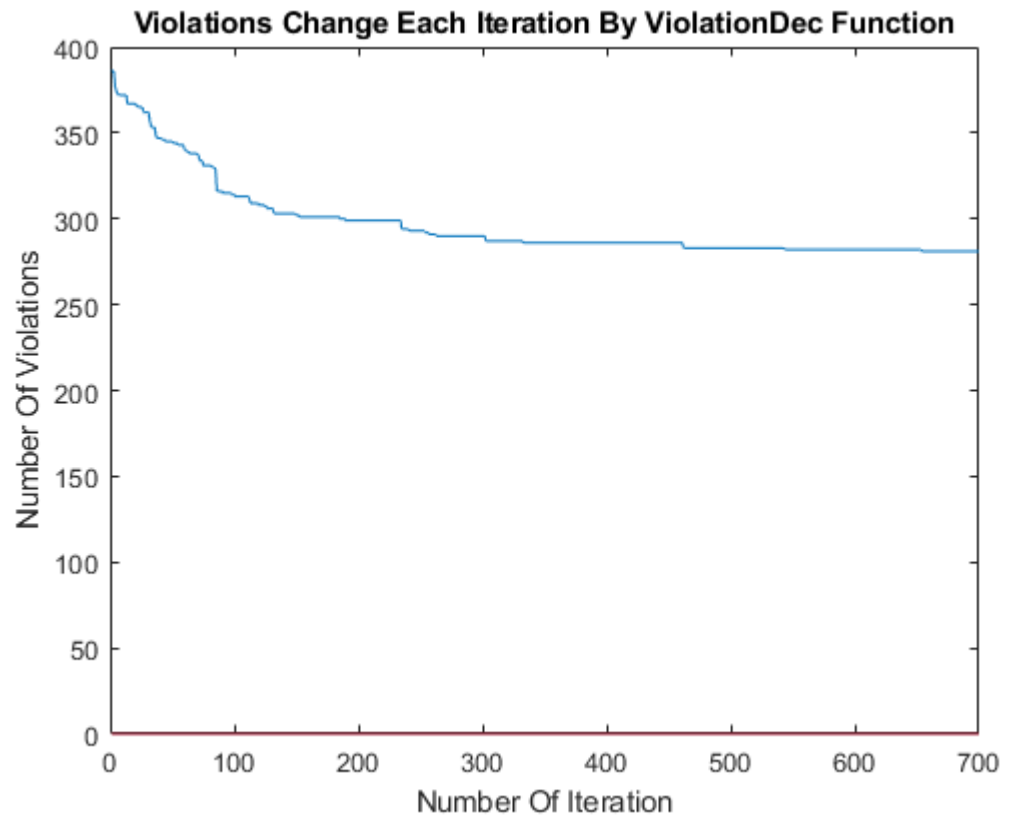
מספר הקבוצות הוא 40, סיכוי שקבוצה תנצח קבוצה שנייה הוא 50%, כלומר כל הקבוצות בעלות חוזק שווה.

תוצאות האלגוריתם:

Initial violations : 386

Minimum violation accomplished: 281

גרף לדעיכה של מספר ה violations על ידי הפונקציה violationDec:



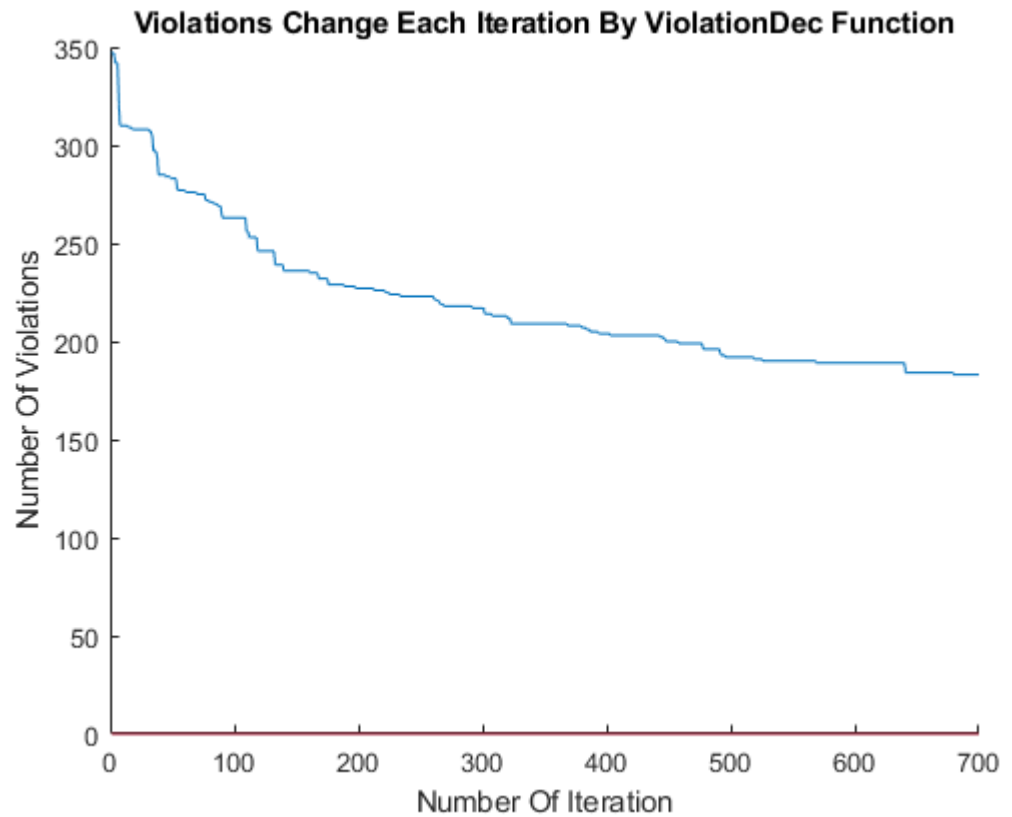
• דוגמא 2:

מספר הקבוצות הוא 40, קיים ידע על רמות החוזק של קבוצות, קבוצה יותר חזקה מנצחת קבוצה יותר חלשה בסיכוי של 70%.
 תוצאות האלגוריתם:

Initial violations : 347

Minimum violation accomplished: 183

גרף לדעיכה של מספר הviolations על ידי הפונקציה violationDec:



אנחנו יכולים לראות שהשינוי המרכזי בין הדוגמאות 1 ו 2 הינו 2 במספר ההפרות המינימלי שהיגענו אליו, ולא במספר ההפרות ההתחלתי שהוא בערך אותו דבר כי התחלנו בשתי הדוגמאות מדירוג רנדומלי.

השיפור מדוגמא 1 לדוגמא 2 היה מאד גדול והוא בערך 100 violations, שזה כמעט 36% ממספר ההפרות המינימלי שהיגענו אליו בדוגמא הראשונה.

• **דוגמא 3:**

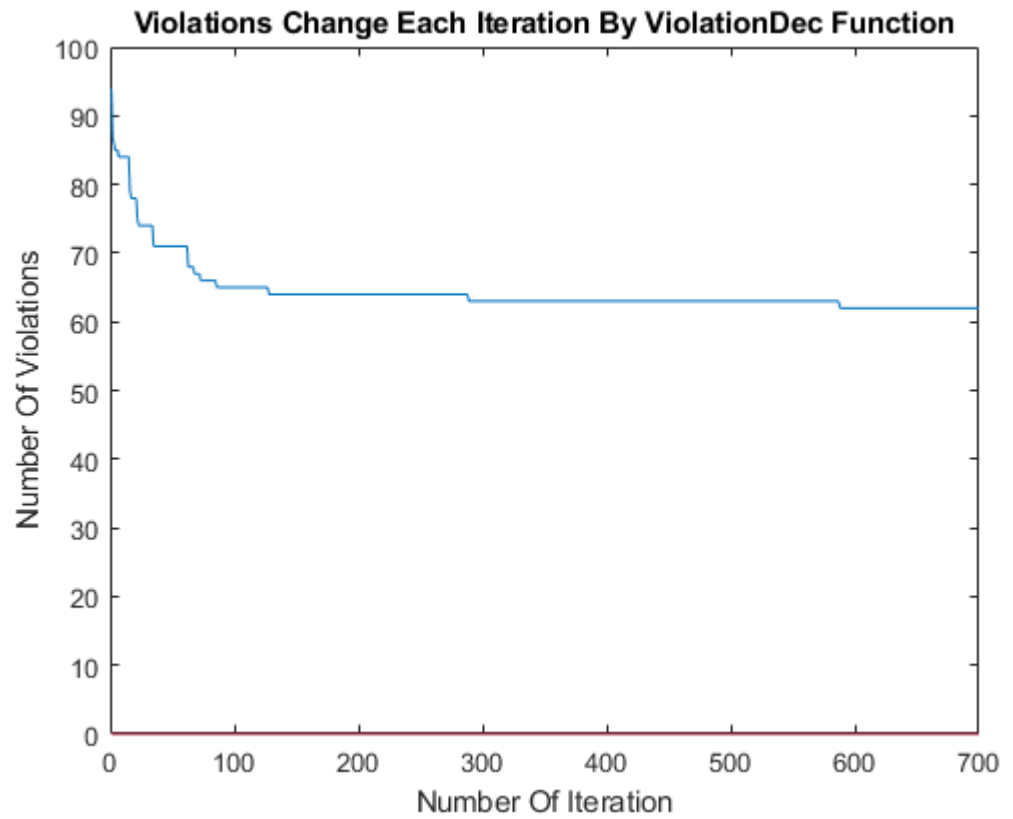
מספר הקבוצות הוא 20, סיכוי שקבוצה תנצח קבוצה שנייה הוא 50%, כלומר אין ידע על רמת החוזק של קבוצות.

תוצאות האלגוריתם:

Initial violations : 94

Minimum violation accomplished: 62

גרף לדעיכה של מספר הviolations על ידי הפונקציה violationDec:



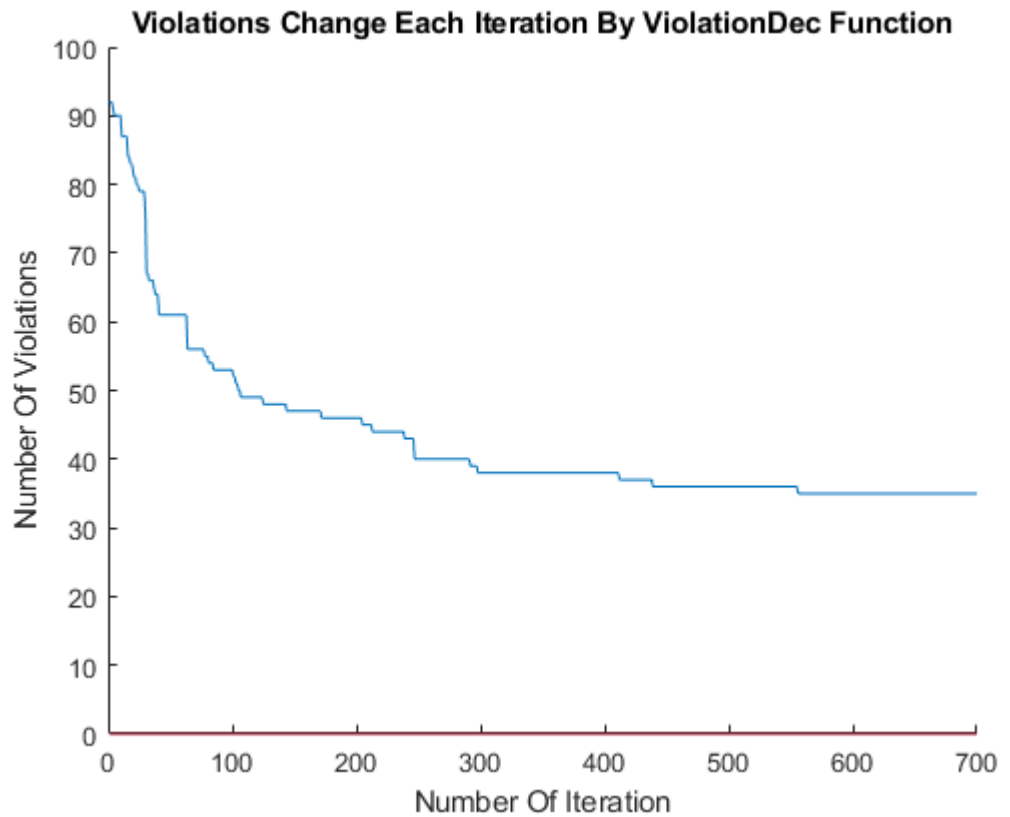
• דוגמא 4:

מספר הקבוצות הוא 20, קיים ידע על רמות החוזק של קבוצות, קבוצה יותר חזקה מנצחת קבוצה יותר חלשה בסיכוי של 70%. תוצאות האלגוריתם:

Initial violations : 92

Minimum violation accomplished: 35

גרף לדעיכה של מספר הviolations על ידי הפונקציה violationDec:



3.3.2 שיטת MVR מתקדמת (דירוג התחלתי אינו רנדומלי) :

- דוגמא 1:

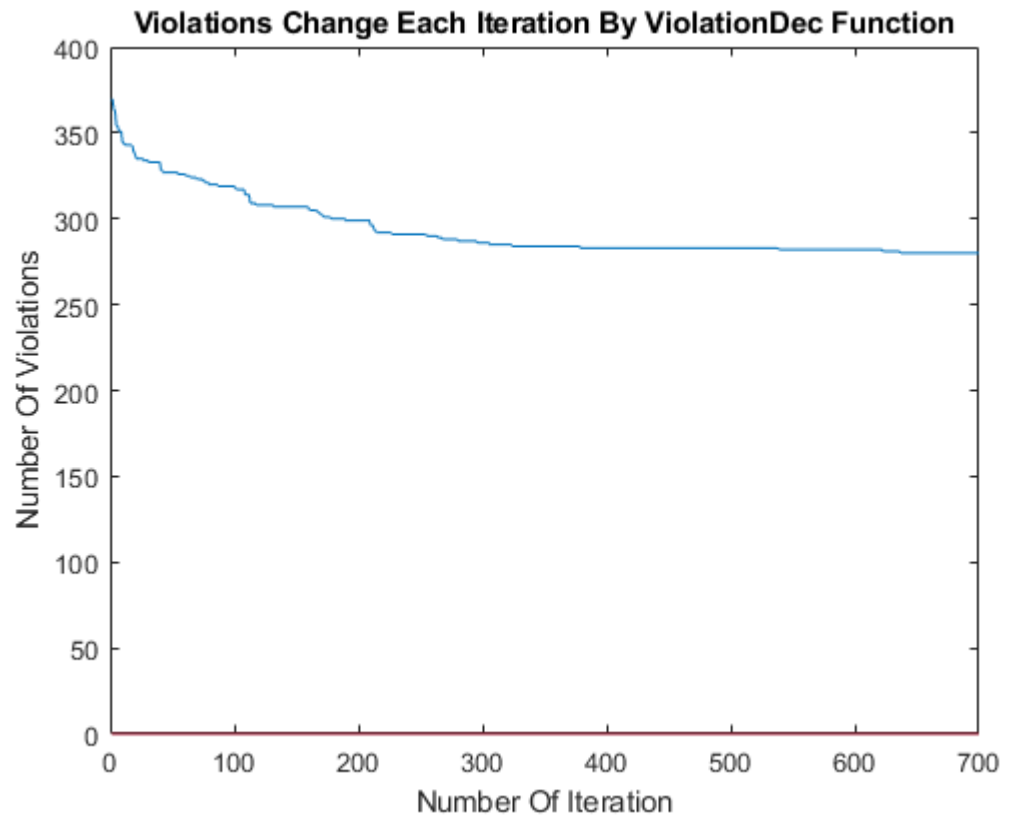
מספר הקבוצות הוא 40, סיכוי שהדירוג הסופי יהיה זהה לדירוג שהיה בעונה הקודמת הינו 75%, לכן אנחנו מתחילים מדירוג התחלתי שבו כל קבוצה תהיה במקום שהיתה בו בשנה שעברה בסיכוי של 75%, אם לא התקיים הסיכוי אז הוא יגדל עבור המקום הבא בסיכוי קבוע (כמו שהוסבר בפרקים קודמים על הדירוג ההתחלתי בשיטה המתקדמת), סיכוי שקבוצה תנצח קבוצה שנייה הוא 50%, כלומר אין ידע על רמות החוזק של הקבוצות בעונה הנוכחית.

תוצאות האלגוריתם:

Initial violations : 370

Minimum violation accomplished: 280

גרף לדעיכה של מספר violations על ידי הפונקציה violationDec:



• דוגמא 2:

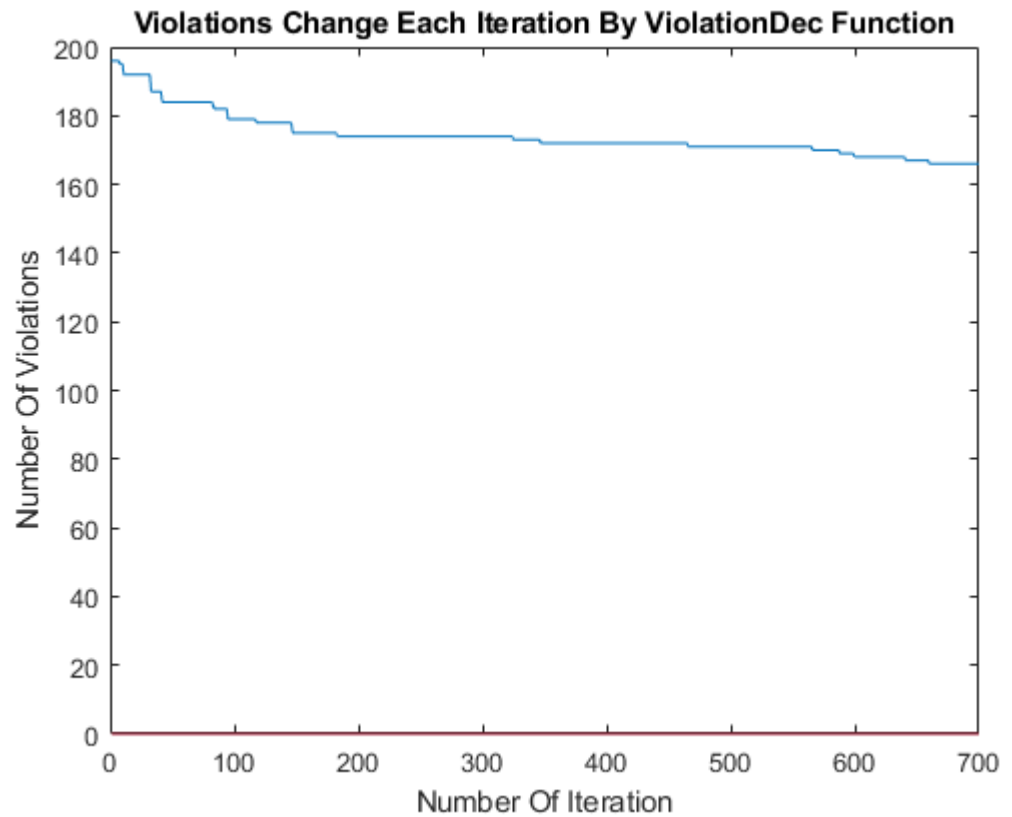
מספר הקבוצות הוא 40, סיכוי שהדירוג הסופי יהיה זהה לדירוג שהיה בעונה הקודמת הינו 75%, לכן אנחנו מתחילים מדירוג התחלתי שבו כל קבוצה תהיה במקום שהיתה בו בשנה שעברה בסיכוי של 75%, אם לא התקיים הסיכוי אז הוא יגדל עבור המקום הבא בסיכוי קבוע (כמו שהוסבר בפרקים קודמים על הדירוג ההתחלתי בשיטה המתקדמת), קיים ידע על רמות החוזק של הקבוצות בעונה הזאת, קבוצה יותר חזקה מנצחת קבוצה יותר חלשה בסיכוי של 70%.

תוצאות האלגוריתם:

Initial violations : 196

Minimum violation accomplished: 166

גרף לדעיכה של מספר הviolations על ידי הפונקציה violationDec:



• דוגמא 3:

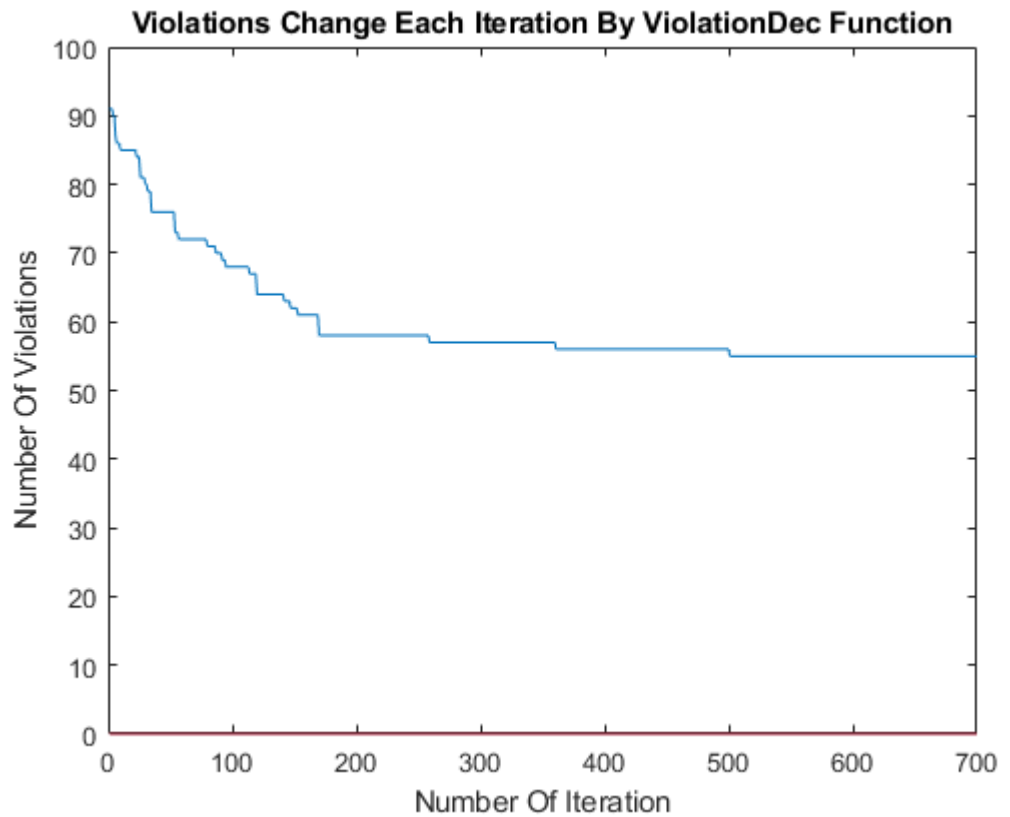
מספר הקבוצות הוא 20, סיכוי שהדירוג הסופי יהיה זהה לדירוג שהיה בעונה הקודמת הינו 80%, לכן אנחנו מתחילים מדירוג התחלתי שבו כל קבוצה תהיה במקום שהיתה בו בשנה שעברה בסיכוי של 80%, אם לא התקיים הסיכוי אז הוא יגדל עבור המקום הבא בסיכוי קבוע (כמו שהוסבר בפרקים קודמים על הדירוג ההתחלתי בשיטה המתקדמת), סיכוי שקבוצה תנצח קבוצה שנייה הוא 50%, כלומר אין ידע על רמת החוזק של קבוצות.

תוצאות האלגוריתם:

Initial violations : 91

Minimum violation accomplished: 55

גרף לדעיכה של מספר violations על ידי הפונקציה violationDec:



• **דוגמא 4:**

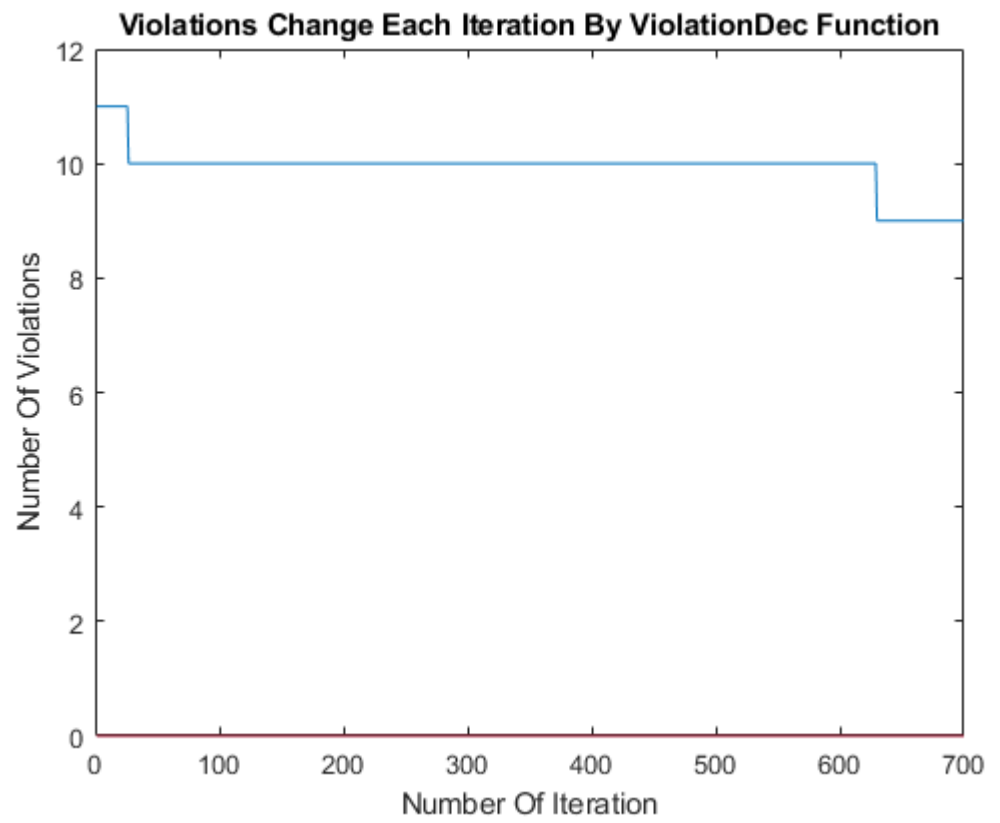
מספר הקבוצות הוא 20, סיכוי שהדירוג הסופי יהיה זהה לדירוג שהיה בעונה הקודמת הינו 80%, לכן אנחנו מתחילים מדירוג התחלתי שבו כל קבוצה תהיה במקום שהיתה בו בשנה שעברה בסיכוי של 80%, אם לא התקיים הסיכוי אז הוא יגדל עבור המקום הבא בסיכוי קבוע (כמו שהוסבר בפרקים קודמים על הדירוג ההתחלתי בשיטה המתקדמת), קיים ידע על רמות החוזק של קבוצות, קבוצה יותר חזקה מנצחת קבוצה יותר חלשה בסיכוי של 95%.

תוצאות האלגוריתם:

Initial violations : 11

Minimum violation accomplished: 9

גרף לדעיכה של מספר הviolations על ידי הפונקציה violationDec:



4 תוצאות שיטת MVR על נתונים אמיתיים

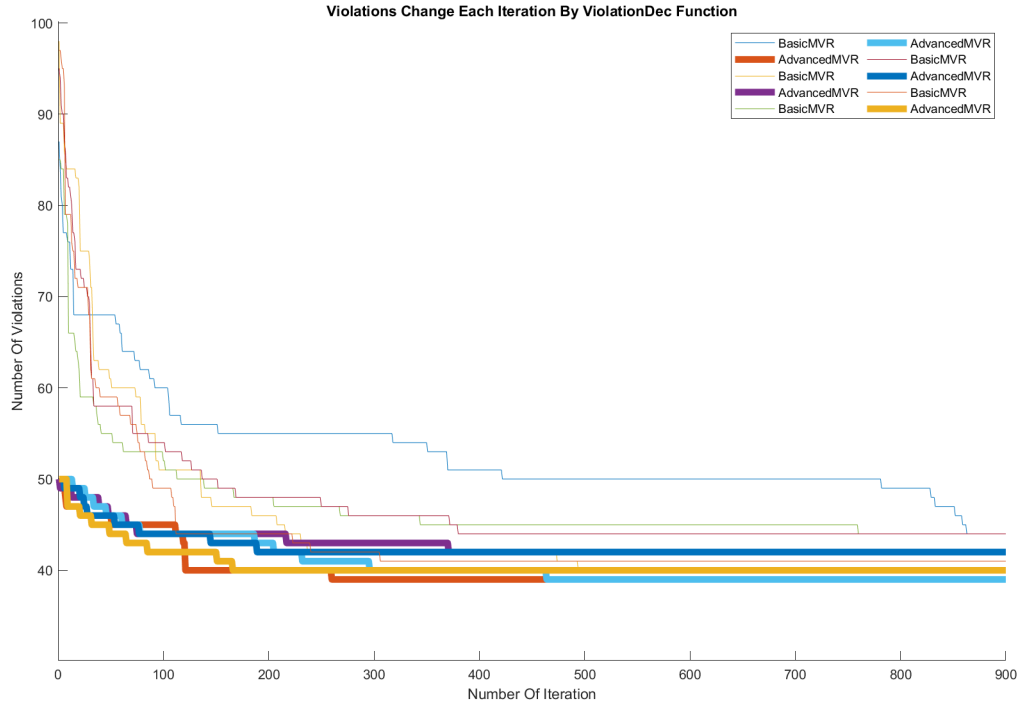
לשם הפעלת השיטה MVR על נתונים אמיתיים, (במקרה שלנו עם תוצאות הליגה הספרדית בשנת 2017), ולא על נתונים רנדומלים או מחושבים במטלב, צריך להכין תוכנית מטלב חדשה (מצורפת בנספחים מתחת לשם תוכנית MVR לניתונים אמיתיים), וגם צריך להניח כמה דברים ולהגדיר כמה הסכמים:

קודם כל צריך להגדיר רמות חוזק לקבוצות שמבוססת על העונות הקודמות, לשם כך אנחנו מגדירים את הדירוג הזה בהסתמכות על רמות החוזק של קבוצות הליגה בשנים האחרונות:

1	Barcelona	11	Espanyol
2	Atletico_Madrid	12	Real_sociedad
3	Real_Madrid	13	Celta_Vigo
4	Valencia	14	Alaves
5	Villareal	15	Levante
6	Real_Betis	16	Athletic_bilbao
7	Sevilla	17	Leganes
8	Getafe	18	Deportivo_La_Coruna
9	Eibar	19	Las_Palmas
10	Girona	20	Malaga

בטבלה הזאת, מצוינות דירוגי הקבוצות מהכי חזקה להכי חלשה לפי העונות הקודמות.

4.1 הפעלת שיטת ההפרה המינימלית על ניתונים אמיתיים



הדירוג עם מספר violations הכי נמוך התקבל בשיטה המתקדמת והוא :

Barcelona
 Atletico_Madrid
 Real_Madrid
 Valencia
 Girona
 Villarreal
 Sevilla
 Eibar
 Leganes
 Getafe
 Real_Sociedad
 Real_Betis
 Celta_Vigo
 Espanyol
 Levante
 Athletic_Bilbao
 Deportivo_La_Coruna
 Alaves
 Las_Palmas
 Malaga
 VIOLATIONS=37.

4.2 השוואה בין שיטה בסיסית למתקדמת

בשיטה הבסיסית מספר violations ממוצע היה כ-42, ובשיטה המתקדמת ממוצע violations שהתקבל היה כ- 37.

averageBasic=41.6000.

averageAdvanced=36.8000.

4.3 השוואה לשיטת הדירוג שמשמשים בה היום

הדירוג הרשמי שהיה בסוף העונה שהשתמשנו בניתונים שלה היה:

Barcelona
Atletico_Madrid
Real_Madrid
Valencia
Villareal
Real_Betis
Sevilla
Getafe
Eibar
Girona
Real_sociedad
Celta_Vigo
Espanyol
Alaves
Levante
Athletic_bilbao
Leganes
Deportivo_La_Coruna
Las_Palmas
Malaga

אנו יכולים לראות כי יש הרבה פחות הבדלים במקומות של הקבוצות בהשוואה לדירוג הכי טוב שהתקבל בעזרת שיטת MVR לבין הדירוג הרשמי שהתקבל בשיטה הרשמית שמקובלת בליגה הספרדית, וזאת עובדה מאוד חשובה שמדגימה ששיטת ה-MVR הינה שיטה יעילה ומדויקת למרות הסטוכסטיות שיש באלגוריתם שלה.

5 סיכום

לסיכום, שיטת MVR הינה שיטה מאוד יעילה לדירוג של קבוצות או דברים שיש ביניהם הבדלים, כמו שראינו מהדוגמאות שבסופו של התהליך אנחנו מגיעים למספר הפרות קטן וזה אומר שהדירוג שמצאנו זהו דירוג טוב ביחס לטבלת המשחקים שהיתה לנו, גם כן בהשוואה בין שיטת MVR הבסיסית למתקדמת, אנחנו רואים שיפור משמעותי לכמות ההפרות הסופית שהיגענו אליה בסוף, וזה נובע משני דברים :

1. התחלה יותר טובה: הדירוג ההתחלתי של השיטה המתקדמת הוא מבוסס על ידע קודם ברמות החוזק של הקבוצות, לכן (בסיכוי מאוד גבוה) אנחנו מתחילים מדירוג בעל מספר הפרות התחלתי יותר קטן, והאלגוריתם שלנו משנה את הדירוג רק אם זה מוביל למספר הפרות יותר קטן, לכן אנחנו אף פעם לא מגדילים את מספר הפרות.
2. האלגוריתם עובד בצורה איטירטיבית, כלומר הוא מבצע מספר איטרציות סופי, ולכן אם נתחיל מנקודה יותר טובה אנחנו (בסיכוי מאוד גבוה) מגיעים לנקודה יותר טובה, כלומר לדירוג סופי עם מספר הפרות יותר קטן.

את שתי הסיבות האלה, גרמו לכך שהשיטה המתקדמת הינה שיטה שמוציאה דירוג סופי עם פחות הפרות במספר סופי של איטרציות, וזה מוביל לדירוג יותר קרוב לדירוג הסופי שהיה במציאות בליגה.

6 רשימת מקורות

- [1] FORD, L. R. AND S. M. JOHNSTON, "A Tournament Problem," Amer. Math. Monthly, 66 (1959), 387-389.
- [2] Iqbal Ali, Wade D. Cook, and Moshe Kress. "On the minimum violations ranking of a tournament". Management Science, 32(6):660-672, 1986.
- [3] Amy N. Langville, Carl D. Meyer - Who's 1 The Science of Rating and Ranking (2012, Princeton University Press)

7 נספחים

(1) קוד תוכנית עבור הוכחה פרקטית:

```
clear all
clc
NumberOfTeams=40;
RandomRankArray=randperm(NumberOfTeams,NumberOfTeams);
%generating numbers from 1 to N without repeating
TempTeamsArray=1:NumberOfTeams;
NumberOfTimesToRun=1;
FinalViolationsArray=zeros(NumberOfTimesToRun,1);
Temporary=20000;
for f=1:1:NumberOfTimesToRun
TeamsMatrix=zeros(NumberOfTeams,NumberOfTeams);
initialProbability_i=100; %this means the stronger team will always win
MoreProbability_i=0;
%Creating the matrix with probabilities
for i=1:1:NumberOfTeams
for j=i+1:1:NumberOfTeams
TotalProbability_i=initialProbability_i+MoreProbability_i;
RandomNumber=randperm(100,1); %choosing random number between 1-
100
if (RandomNumber<=TotalProbability_i) %it means that the stronger team
won
TeamsMatrix(i,j)=1;
else
TeamsMatrix(i,j)=0;
end
MoreProbability_i=MoreProbability_i+(20/(NumberOfTeams-2));
end
MoreProbability_i=0;
TotalProbability_i=initialProbability_i;
end
%making the main diagonal of the matrix = 0 because i==j (team against
%itself)
for m=1:1:NumberOfTeams
TeamsMatrix(m,m)=0;
end
%now we fix the random filled matrix in this way: %we go on the matrix
horizontally and each 1 we discover,
suppose in (i,j), we %put 0 in (j,i)
for i=1:1:NumberOfTeams
for j=1:1:NumberOfTeams
if(i~=j)
if(TeamsMatrix(i,j)==1)
```



```

TeamsMatrix(j,i)=0;
end
if(TeamsMatrix(i,j)==0)
TeamsMatrix(j,i)=1;
end
end
end
end
ZeroViolationsArray=zeros(NumberOfTeams,1);
for i=1:1:NumberOfTeams
for j=1:1:NumberOfTeams
if(TeamsMatrix(i,j)==1)
ZeroViolationsArray(i)=ZeroViolationsArray(i)+1;
end
end
end
[Temp,I]=sort(ZeroViolationsArray,'descend');
ViolationBasedOnWinsNum=ViolationCalculator(I,TeamsMatrix);
initial_violation=ViolationCalculator(RandomRankArray,TeamsMatrix);
ViaolationBasedOnStrength=ViolationCalculator(Temp TeamsArray,TeamsMatrix);
NumberOfIterationForVioDec=10001;
%now we have a random rank saved in the array RandomRankArray
New_Rank=RandomRankArray;
ViolationArray=zeros(NumberOfIterationForVioDec);
ViolationArray(1)=initial_violation;
% A=ones(NumberOfTeams);
% ZeroViolations_Matrix=triu(A);
% for m=1:1:NumberOfTeams
% ZeroViolations_Matrix(m,m)=0;
% end
for i=1:1:NumberOfIterationForVioDec-1 [New_Rank,violation]=ViolationDec(New_Rank,TeamsMatrix);
ViolationArray(i+1)=violation;
end
if(violation<Temporary)
Temporary=violation;
MinimumViolationRank=New_Rank;
minVio=violation;
end
[temp2,MinimalVioRank]=sort(New_Rank);
hold on
plot(ViolationArray(1:5000));
title('Violations Change Each Iteration By ViolationDec Function')
xlabel('Number Of Iteration')
ylabel('Number Of Violations')
Final_Violation=violation;
FinalViolationsArray(f)=violation;

```

```

MinimumViolationAccomplished=min(FinalViolationsArray);
end
%histogram(FinalViolationsArray);
%hold on
FinalViolationsArray_Mean=mean(FinalViolationsArray); TempArray=zeros(length(FinalViolationsArray))
%for plot
%for x=1:1:length(FinalViolationsArray)
%TempArray(x)=FinalViolationsArray_Mean;
%end
% plot(TempArray);
%hold off

```

2) קוד מטלב לשיטת MVR בסיסית:

```

clear all clc
NumberOfTeams=20;
RandomRankArray=randperm(NumberOfTeams,NumberOfTeams);
%generating numbers from 1 to N without repeating
NumberOfTimesToRun=1;
FinalViolationsArray=zeros(NumberOfTimesToRun,1);
for f=1:1:NumberOfTimesToRun
TeamsMatrix=randi([01],NumberOfTeams,NumberOfTeams);
%making the main diagonal of the matrix = 0 because i==j
for m=1:1:NumberOfTeams
TeamsMatrix(m,m)=0;
end
%now we fix the random filled matrix in this way: %we go on the matrix
horizontally and each 1 we meet, suppose in (i,j), we %put 0 in (j,i)
for i=1:1:NumberOfTeams
for j=1:1:NumberOfTeams
if(i~=j)
if(TeamsMatrix(i,j)==1)
TeamsMatrix(j,i)=0;
end
if(TeamsMatrix(i,j)==0) TeamsMatrix(j,i)=1; end
end
end
end
%now we have the true ranks of the team based on the comon method %each
team rank is saved in OriginalRankArray array respectively
initial_violation=ViolationCalculator(RandomRankArray,TeamsMatrix);
%now we have a random rank saved in the array RandomRankArray
NumberOfIterationForVioDec=1001;
New_Rank=RandomRankArray; ViolationArray=zeros(NumberOfIterationForVioDec);
ViolationArray(1)=initial_violation;
% A=ones(NumberOfTeams); % ZeroViolations_Matrix=triu(A);

```

```

% for m=1:1:NumberOfTeams
% ZeroViolations_Matrix(m,m)=0;
% end
for i=1:1:NumberOfIterationForVioDec-1 [New_Rank,violation]=ViolationDec(New_Rank,TeamsMatrix);
ViolationArray(i+1)=violation;
end
%hold on %plot(ViolationArray);
FinalViolationsArray(f)=violation;
end
histogram(FinalViolationsArray);
% hold on % FinalViolationsArray_Mean=mean(FinalViolationsArray);
%TempArray=zeros(length(FinalViolationsArray),1);
%for plot
%for x=1:1:length(FinalViolationsArray)
%TempArray(x)=FinalViolationsArray_Mean;
%end
%%
%%plot(TempArray);
%hold off

```

3 קוד מטלב לשיטת MVR בסיסית עם מטריצת תוצאות כוללת חשיבות לחוזק הקבוצות (לשם נוחות: תא 1 במערך הינה הקבוצה הכי חזקה ותא אחרון הקבוצה הכי חלשה):

```

clear all
clc
NumberOfTeams=20; RandomRankArray=randperm(NumberOfTeams,NumberOfTeams);
%generating numbers from 1 to N without repeating
TempTeamsArray=1:NumberOfTeams; NumberOfTimesToRun=5;
FinalViolationsArray=zeros(NumberOfTimesToRun,1);
for f=1:1:NumberOfTimesToRun
TeamsMatrix=zeros(NumberOfTeams,NumberOfTeams);
initialProbability_i=80; MoreProbability_i=0;
%Creating the matrix with probabilities
for i=1:1:NumberOfTeams
for j=i+1:1:NumberOfTeams TotalProbability_i=initialProbability_i+MoreProbability_i;
RandomNumber=randperm(100,1);
%choosing random number between 1-100 if (RandomNumber<=TotalProbability_i)
%it means that the stronger team won TeamsMatrix(i,j)=1;
else
TeamsMatrix(i,j)=0;
end MoreProbability_i=MoreProbability_i+(20/(NumberOfTeams-2)); end
MoreProbability_i=0; TotalProbability_i=initialProbability_i;
end
%making the main diagonal of the matrix = 0 because i==j (team against
%itself)

```

```

    for m=1:1:NumberOfTeams TeamsMatrix(m,m)=0;
    end
    %now we fix the random filled matrix in this way: %we go on the matrix
    horizontally and each 1 we discover, suppose in (i,j), we %put 0 in (j,i)
    for i=1:1:NumberOfTeams
    for j=1:1:NumberOfTeams
    if(i~=j)
    if(TeamsMatrix(i,j)==1) TeamsMatrix(j,i)=0;
    end
    if(TeamsMatrix(i,j)==0) TeamsMatrix(j,i)=1;
    end
    end
    end
    end
    ZeroViolationsArray=zeros(NumberOfTeams,1);
    for i=1:1:NumberOfTeams
    for j=1:1:NumberOfTeams if(TeamsMatrix(i,j)==1) ZeroViolationsArray(i)=ZeroViolationsArray(i)+1;
    end
    end
    end [Temp,I]=sort(ZeroViolationsArray,'descend');
    ViolationBasedOnWinsNum=ViolationCalculator(I,TeamsMatrix); initial_violation=ViolationCalculator(R
ViolationBasedOnStrength=ViolationCalculator(TempTeamsArray,TeamsMatrix);
    NumberOfIterationForVioDec=10001;
    %now we have a random rank saved in the array RandomRankArray New_Rank=RandomRankArray;
ViolationArray=zeros(NumberOfIterationForVioDec);
    ViolationArray(1)=initial_violation;
    %A=ones(NumberOfTeams);
    %ZeroViolations_Matrix=triu(A);
    %for m=1:1:NumberOfTeams
    %ZeroViolations_Matrix(m,m)=0;
    %end
    for i=1:1:NumberOfIterationForVioDec-1 [New_Rank,violation]=ViolationDec(New_Rank,TeamsMatrix);
    ViolationArray(i+1)=violation;
    end [
temp2,MinimalVioRank]=sort(New_Rank);
    % hold on
    % plot(ViolationArray);
    Final_Violation=violation;
    FinalViolationsArray(f)=violation;
    end
    histogram(FinalViolationsArray);
    % hold on
    %
    %FinalViolationsArray_Mean=mean(FinalViolationsArray);
    %TempArray=zeros(length(FinalViolationsArray),1);
    % %for plot

```

```

%for x=1:1:length(FinalViolationsArray)
%TempArray(x)=FinalViolationsArray_Mean;
%end
% %
% plot(TempArray);
% hold off

```

4 קוד מטלב לשיטת MVR מתקדמת:

```

clear all
clc
Probability=70;
NumberOfTeams=40;
ArrayWithStrengthCalculated=ArrayBuilder(Probability,NumberOfTeams);
NumberOfTimesToRun=100;
TeamsMatrix=randi([01],NumberOfTeams,NumberOfTeams);
%making the main diagonal of the matrix = 0 because i==j
for m=1:1:NumberOfTeams
TeamsMatrix(m,m)=0;
end
%now we fix the random filled matrix in this way: %we go on the matrix
horizontally and each 1 we meet, suppose in (i,j), we %put 0 in (j,i)
for i=1:1:NumberOfTeams
for j=1:1:NumberOfTeams
if(i~=j)
if(TeamsMatrix(i,j)==1)
TeamsMatrix(j,i)=0;
end
if (TeamsMatrix(i,j)==0)
TeamsMatrix(j,i)=1;
end
end
end
end
FinalViolationsArray=zeros(NumberOfTimesToRun,1);
for f=1:1:NumberOfTimesToRun
%now we have the true ranks of the team based on the comon method %each
team rank is saved in OriginalRankArray array respectively
initial_violation=ViolationCalculator(ArrayWithStrengthCalculated,TeamsMatrix);
%now we have a random rank saved in the array ArrayWithStrengthCalcu-
lated
NumberOfIterationForVioDec=1001;
New_Rank=ArrayWithStrengthCalculated; ViolationArray=zeros(NumberOfIterationForVioDec);
ViolationArray(1)=initial_violation;
% A=ones(NumberOfTeams);
% ZeroViolations_Matrix=triu(A);

```

```

% for m=1:1:NumberOfTeams
% ZeroViolations_Matrix(m,m)=0; % end
for i=1:1:NumberOfIterationForVioDec-1 [New_Rank,violation]=ViolationDec(New_Rank,TeamsMatrix);
ViolationArray(i+1)=violation;
end
%hold on
%plot(ViolationArray);
final_violation=violation;
FinalViolationsArray(f)=violation; end
histogram(FinalViolationsArray);
% hold on % % FinalViolationsArray_Mean=mean(FinalViolationsArray);
% TempArray=zeros(length(FinalViolationsArray),1);
% %for plot
% for x=1:1:length(FinalViolationsArray)
% TempArray(x)=FinalViolationsArray_Mean;
% end
% %
%plot(TempArray);
% hold off

```

5) קוד מטלב לשיטת MVR מתקדמת עם מטריצת תוצאות כוללת חשיבות לחזיק הקבוצות (לשם נוחות: תא 1 במערך הינה הקבוצה הכי חזקה ותא אחרון הקבוצה הכי חלשה):

```

clear all
clc
Probability=80;
NumberOfTeams=40;
ArrayWithStrengthCalculated=ArrayBuilder(Probability,NumberOfTeams);
TempTeamsArray=1:NumberOfTeams; NumberOfTimesToRun=100;
Temporary=20000;
FinalViolationsArray=zeros(NumberOfTimesToRun,1);
for f=1:1:NumberOfTimesToRun
TeamsMatrix=zeros(NumberOfTeams,NumberOfTeams);
initialProbability_i=60;
MoreProbability_i=0;
%Creating the matrix with probabilities
for i=1:1:NumberOfTeams
for j=i+1:1:NumberOfTeams TotalProbability_i=initialProbability_i+MoreProbability_i;
RandomNumber=randperm(100,1);
%choosing random number between 1-100
if (RandomNumber<=TotalProbability_i) %it means that the stronger team
won
TeamsMatrix(i,j)=1;
else TeamsMatrix(i,j)=0;
end

```

```

    MoreProbability_i=MoreProbability_i+(20/NumberOfTeams-2));
end
MoreProbability_i=0; TotalProbability_i=initialProbability_i;
end
%making the main diagonal of the matrix = 0 because i==j (team against
%itself)
for m=1:1:NumberOfTeams
    TeamsMatrix(m,m)=0;
end
%now we fix the random filled matrix in this way: %we go on the matrix
horizontally and each 1 we discover, suppose in (i,j), we %put 0 in (j,i)
for i=1:1:NumberOfTeams
    for j=1:1:NumberOfTeams
        if(i~=j)
            if(TeamsMatrix(i,j)==1)
                TeamsMatrix(j,i)=0;
            end
            if(TeamsMatrix(i,j)==0)
                TeamsMatrix(j,i)=1;
            end
        end
    end
end
ZeroViolationsArray=zeros(NumberOfTeams,1);
for i=1:1:NumberOfTeams
    for j=1:1:NumberOfTeams
        if(TeamsMatrix(i,j)==1) ZeroViolationsArray(i)=ZeroViolationsArray(i)+1;
        end
    end
end
[Temp,I]=sort(ZeroViolationsArray,'descend');
ViolationBasedOnWinsNum=ViolationCalculator(I,TeamsMatrix); initial_violation=ViolationCalculator(A
ViolationBasedOnStrength=ViolationCalculator(TempTeamsArray,TeamsMatrix);
NumberOfIterationForVioDec=10001;
%now we have a random rank saved in the array ArrayWithStrengthCalcu-
lated New_Rank=ArrayWithStrengthCalculated; ViolationArray=zeros(NumberOfIterationForVioDec);
ViolationArray(1)=initial_violation;
% A=ones(NumberOfTeams);
% ZeroViolations_Matrix=triu(A);
% for m=1:1:NumberOfTeams
% ZeroViolations_Matrix(m,m)=0; % end
for i=1:1:NumberOfIterationForVioDec-1 [New_Rank,violation]=ViolationDec(New_Rank,TeamsMatrix);
ViolationArray(i+1)=violation;
end
if(violation<Temporary)
    Temporary=violation;

```

```

MinimumViolationRank=New_Rank;
minVio=violation;
end
[temp2,MinimalVioRank]=sort(New_Rank);
% hold on
% plot(ViolationArray);
Final_Violation=violation;
FinalViolationsArray(f)=violation; MinimumViolationAccomplished=min(FinalViolationsArray);
end
histogram(FinalViolationsArray);
hold on
FinalViolationsArray_Mean=mean(FinalViolationsArray); TempArray=zeros(length(FinalViolationsArray))
%for plot
for x=1:length(FinalViolationsArray) TempArray(x)=FinalViolationsArray_Mean;
end
%
plot(TempArray);
hold off

```

:ViolationClaculator עזר פונקציית (6 קוד

```

%here is a function for calculating violation of a rank
%input: random rank and the matches matrix
%output: violation value
function [violation]= ViolationCalculator(RandomRank,TeamsMatrix)
violation=0;
for i=1:length(RandomRank)
for j=1:length(RandomRank) if(RandomRank(i)>RandomRank(j))
%it means that team i rank is higher than team j rank (higher = worst in
ranks)
if(TeamsMatrix(i,j)==1)
%it means that team i win against team j
violation=violation+1; %if both conditions true, then its a violation
end
end
end
end
end
end
end

```

:ViolationDec עזר פונקציית (7 קוד

```

function [Rank1,violation] = ViolationDec(Rank,Matrix)
Old_violation=ViolationCalculator(Rank,Matrix);
Teams=randperm(length(Rank),2);
i=Teams(1);
j=Teams(2);
NewRank=Rank;

```



```

NewRank(j)=NewRank(i);
NewRank(i)=Rank(j);
New_violation=ViolationCalculator(NewRank,Matrix);
if(New_violation<Old_violation)
Rank1=NewRank;
violation=New_violation;
else
Rank1=Rank;
violation=Old_violation;
end
end
end

```

8 קוד פונקציית עזר ArrayBuilder:

```

function [ArrayWithStrengthCalculated] = ArrayBuilder(Prob,N)
    numberofteams=N;
    ArrayWithStrengthCalculated=zeros(1,numberofteams); initial_prob=Prob;
    prob_inc=(initial_prob)/(numberofteams-1);
    TeamIndex=1;
    changableNumberOfTeams=numberofteams+2;
    changableProb=initial_prob;
    for i=1:1:numberofteams
        RandomNumber=randperm(100,1);
        %choosing random number between 1-100
        for j=1:1:numberofteams
            if (RandomNumber<=changableProb) if(ArrayWithStrengthCalculated(j)==0)
                ArrayWithStrengthCalculated(j)=TeamIndex;
                break;
            end
        end
        changableProb=changableProb+prob_inc; RandomNumber=randperm(100,1);
    end
    changableProb=initial_prob;
    TeamIndex=TeamIndex+1; changableNumberOfTeams=changableNumberOfTeams-1;
    prob_inc=(initial_prob)/(changableNumberOfTeams-1);
end

```

9 קוד מטלה לתוכנית MVR עם ניתונים אמיתיים:

```

%MVR For Real Data (Spanish League)
clear all
clc
%Reading The Data:
r=sym('r', [20 1]); %2017 ranks
disp('***** Loading Data *****')
[~, text] = xlsread('season-1718.xlsx',1,'C2:D381');
%reads from C(upper left corner) to D(lower right corner).

```

```

AllSeasonResults = xlsread('season-1718.xlsx',1,'E:F');
%reads from E(upper left corner) to F(lower right corner).
disp('***** The Data has been successfully loaded *****')
%changing the text cells to be the rank of the team instead of the name of
%the team
textWithIndicies=zeros(380,2);
for i=1:1:380
for j=1:1:2
if(text(i,j)=="Barcelona")
textWithIndicies(i,j)=1;
end
if(text(i,j)=="AtleticodeMadrid")
textWithIndicies(i,j)=2;
end
if(text(i,j)=="RealMadrid")
textWithIndicies(i,j)=3;
end
if(text(i,j)=="Valencia")
textWithIndicies(i,j)=4;
end
if(text(i,j)=="Villarreal")
textWithIndicies(i,j)=5;
end
if(text(i,j)=="RealBetis")
textWithIndicies(i,j)=6;
end
if(text(i,j)=="Sevilla")
textWithIndicies(i,j)=7;
end
if(text(i,j)=="Getafe")
textWithIndicies(i,j)=8;
end
if(text(i,j)=="Eibar")
textWithIndicies(i,j)=9;
end
if(text(i,j)=="Girona")
textWithIndicies(i,j)=10;
end
if(text(i,j)=="Espanyol")
textWithIndicies(i,j)=11;
end
if(text(i,j)=="RealSociedad")
textWithIndicies(i,j)=12;
end
if(text(i,j)=="CeltadeVigo")
textWithIndicies(i,j)=13;

```

```

end
if(text(i,j)=="DeportivoAlaves")
textWithIndicies(i,j)=14;
end
if(text(i,j)=="Levante")
textWithIndicies(i,j)=15;
end
if(text(i,j)=="AthleticdeBilbao")
textWithIndicies(i,j)=16;
end
if(text(i,j)=="Leganes")
textWithIndicies(i,j)=17;
end
if(text(i,j)=="DeportivoLaCoruna")
textWithIndicies(i,j)=18;
end
if(text(i,j)=="LasPalmas")
textWithIndicies(i,j)=19;
end
if(text(i,j)=="Malaga")
textWithIndicies(i,j)=20;
end
end
end
end
FirstSeasonHalfMatrix=zeros(20,20);
SecondSeasonHalfMatrix=zeros(20,20);
halfLength=190;
seasonLength=380;
ArrayForRankingBasedOnWins=zeros(20,1);
for i=1:1:halfLength
HomeTeam=textWithIndicies(i,1);
AwayTeam=textWithIndicies(i,2);
if(AllSeasonResults(i,1)>AllSeasonResults(i,2))
%home team won
FirstSeasonHalfMatrix(HomeTeam,AwayTeam)=1;
FirstSeasonHalfMatrix(AwayTeam,HomeTeam)=-1;
end
if(AllSeasonResults(i,1)<AllSeasonResults(i,2))
%away team won
FirstSeasonHalfMatrix(HomeTeam,AwayTeam)=-1;
FirstSeasonHalfMatrix(AwayTeam,HomeTeam)=1;
end
if(AllSeasonResults(i,1)==AllSeasonResults(i,2))
%Tie
FirstSeasonHalfMatrix(HomeTeam,AwayTeam)=99;
FirstSeasonHalfMatrix(AwayTeam,HomeTeam)=99;

```

```

end
end
%filling the first half season matrix
for i=halfLength+1:1:seasonLength
HomeTeam=textWithIndicies(i,1);
AwayTeam=textWithIndicies(i,2);
if(AllSeasonResults(i,1)>AllSeasonResults(i,2))
%home team won
SecondSeasonHalfMatrix(HomeTeam,AwayTeam)=1;
SecondSeasonHalfMatrix(AwayTeam,HomeTeam)=-1;
end
if(AllSeasonResults(i,1)<AllSeasonResults(i,2))
%away team won
SecondSeasonHalfMatrix(HomeTeam,AwayTeam)=-1;
SecondSeasonHalfMatrix(AwayTeam,HomeTeam)=1;
end
if(AllSeasonResults(i,1)==AllSeasonResults(i,2))
%Tie
SecondSeasonHalfMatrix(HomeTeam,AwayTeam)=99;
SecondSeasonHalfMatrix(AwayTeam,HomeTeam)=99;
end
end
end
%mixing up the two matrixs:
FinalSeasonMatrix=zeros(20,20);
%if in firstmatrix(i,j)=99=secondmatrix(i,j)
%then in finalmatrix(i,j)=99;
for i=1:1:20
for j=1:1:20
if(FirstSeasonHalfMatrix(i,j)==99)
if(SecondSeasonHalfMatrix(i,j)==99)
FinalSeasonMatrix(i,j)=99;
end
end
end
end
%now we deal with lose-win or tie-win/tie-lose situations,
%for example if team a won against team b in the first
%season half and team b won against team a in the second season
%half, what to do ?
%the team with the most goals considered won in final matrix
%if they scored same goals in the 2 games, for example:
%first game between a and b a won 1:0
%second game between b and a b won 1:0
%in this situation we compare with goals scored and if equal
% we use coinflip and give the win for the team who win the coinflip
FirstTeamGoalsCounter=0;

```

```

SecondTeamGoalsCounter=0;
for i=1:1:20
for j=1:1:20
if(FirstSeasonHalfMatrix(i,j)==1)
%won first game
if(SecondSeasonHalfMatrix(i,j)==1)
%won second game
FinalSeasonMatrix(i,j)=1;
FinalSeasonMatrix(j,i)=-1;
end
if(SecondSeasonHalfMatrix(i,j)==99)
%tie second game
FinalSeasonMatrix(i,j)=1;
FinalSeasonMatrix(j,i)=-1;
end
if(SecondSeasonHalfMatrix(i,j)==-1)
%lost second game
for m=1:1:380
if(textWithIndicies(m,1)==i && textWithIndicies(m,2)==j)
break;
end
end
for f=1:1:380
if(textWithIndicies(f,2)==i && textWithIndicies(f,1)==j)
break;
end
end
%now m and f is the indecies we want
FirstTeamGoalsCounter=AllSeasonResults(m,1)+AllSeasonResults(f,2);
SecondTeamGoalsCounter=AllSeasonResults(m,2)+AllSeasonResults(f,1);
if(FirstTeamGoalsCounter>SecondTeamGoalsCounter)
FinalSeasonMatrix(i,j)=1;
FinalSeasonMatrix(j,i)=-1;
end
if(FirstTeamGoalsCounter<SecondTeamGoalsCounter)
FinalSeasonMatrix(i,j)=-1;
FinalSeasonMatrix(j,i)=1;
end
if(FirstTeamGoalsCounter==SecondTeamGoalsCounter)
coinflip=randi(100);
if(coinflip>50)
FinalSeasonMatrix(i,j)=1;
FinalSeasonMatrix(j,i)=-1;
end
if(coinflip<50)
FinalSeasonMatrix(i,j)=-1;

```

```

FinalSeasonMatrix(j,i)=1;
end
end
end
end if(FirstSeasonHalfMatrix(i,j)==-1)
%lost first game
if(SecondSeasonHalfMatrix(i,j)==1)
%won second game
for m=1:1:380
if(textWithIndicies(m,1)==i && textWithIndicies(m,2)==j)
break;
end
end
for f=1:1:380 if(textWithIndicies(f,2)==i && textWithIndicies(f,1)==j)
break;
end
end
%now m and f is the indecies we want
FirstTeamGoalsCounter=AllSeasonResults(m,1)+AllSeasonResults(f,2);
SecondTeamGoalsCounter=AllSeasonResults(m,2)+AllSeasonResults(f,1);
if(FirstTeamGoalsCounter>SecondTeamGoalsCounter)
FinalSeasonMatrix(i,j)=1;
FinalSeasonMatrix(j,i)=-1;
end
if(FirstTeamGoalsCounter<SecondTeamGoalsCounter)
FinalSeasonMatrix(i,j)=-1;
FinalSeasonMatrix(j,i)=1;
end
if(FirstTeamGoalsCounter==SecondTeamGoalsCounter)
coinflip=randi(100);
if(coinflip>50)
FinalSeasonMatrix(i,j)=1;
FinalSeasonMatrix(j,i)=-1;
end if(coinflip<50)
FinalSeasonMatrix(i,j)=-1;
FinalSeasonMatrix(j,i)=1;
end
end
end
if(SecondSeasonHalfMatrix(i,j)==99)
%tie second game
FinalSeasonMatrix(i,j)=-1;
FinalSeasonMatrix(j,i)=1;
end
if(SecondSeasonHalfMatrix(i,j)==-1)
%lost second game

```

```

FinalSeasonMatrix(i,j)=-1;
FinalSeasonMatrix(j,i)=1;
end
end
if(FirstSeasonHalfMatrix(i,j)==99)
%tie first game
if(SecondSeasonHalfMatrix(i,j)==1)
%won second game
FinalSeasonMatrix(i,j)=1;
FinalSeasonMatrix(j,i)=-1;
end
if(SecondSeasonHalfMatrix(i,j)==-1)
%lost second game
FinalSeasonMatrix(i,j)=-1;
FinalSeasonMatrix(j,i)=1;
end
end
end
end
%now we have the final result matrix of the season
for x=1:1:20
for y=1:1:20
if(FinalSeasonMatrix(x,y)==99)
coinflip=randi(100);
if(coinflip>50) FinalSeasonMatrix(x,y)=1;
FinalSeasonMatrix(y,x)=-1;
end
if(coinflip<=50)
FinalSeasonMatrix(x,y)=-1;
FinalSeasonMatrix(y,x)=1;
end
end
end
end
%TESTING THE MATRIX
error=0;
for i=1:1:20
for j=1:1:20
if(FinalSeasonMatrix(i,j)==1 && FinalSeasonMatrix(j,i)~-1)
error=error+1;
disp(i);
disp(j);
end
end
end
if(error>0)

```

```

disp('**** ERROR IN THE MATRIX ****');
return;
end
for x=1:1:20
for y=1:1:20
if(FinalSeasonMatrix(x,y)==1)
ArrayForRankingBasedOnWins(x)=ArrayForRankingBasedOnWins(x)+1;
end
end
end
NumOfWins=sum(ArrayForRankingBasedOnWins);
NumberOfSimulations=5;
%for the Advanced_MVR_Method
ArrayBasedOnStrength=ArrayBuilder(60,20);
NumberOfIterationForVioDec=900;
min_violations_AdvancedArray=zeros(NumberOfSimulations,1);
min_violations_BasicArray=zeros(NumberOfSimulations,1);
min_violations_AdvancedMatrix=zeros(NumberOfSimulations,20);
min_violations_BasicMatrix=zeros(NumberOfSimulations,20);
for index=1:1:NumberOfSimulations
%initial violations
initialVioaltion_AdvancedMVR=ViolationCalculator(ArrayBasedOnStrength,FinalSeasonMatrix);
Advanced_MVR_FINAL_RANK=ArrayBasedOnStrength;
BasicMVR_ViolationArray=zeros(NumberOfIterationForVioDec,1);
AdvancedMVR_ViolationArray=zeros(NumberOfIterationForVioDec,1);
AdvancedMVR_ViolationArray(1)=initialVioaltion_AdvancedMVR;
for i=1:1:NumberOfIterationForVioDec-1
[Advanced_MVR_FINAL_RANK,violation]=ViolationDec(Advanced_MVR_FINAL_RANK,FinalSeasonMatrix);
AdvancedMVR_ViolationArray(i+1)=violation;
end
MinimumVio_AdvancedMVR=AdvancedMVR_ViolationArray(NumberOfIterationForVioDec);
min_violations_AdvancedArray(index)=MinimumVio_AdvancedMVR;
min_violations_AdvancedMatrix(index,:)=Advanced_MVR_FINAL_RANK;
%for the Basic_MVR_Method
InitialRandomRankArray=randperm(20,20);
%generating numbers from 1 to N without repeating
initialVioaltion_BasicMVR=ViolationCalculator(InitialRandomRankArray,FinalSeasonMatrix);
Basic_MVR_FINAL_RANK=InitialRandomRankArray;
BasicMVR_ViolationArray(1)=initialVioaltion_BasicMVR;
for i=1:1:NumberOfIterationForVioDec-1
[Basic_MVR_FINAL_RANK,violation]=ViolationDec(Basic_MVR_FINAL_RANK,FinalSeasonMatrix);
BasicMVR_ViolationArray(i+1)=violation;
end
MinimumVio_BasicMVR=BasicMVR_ViolationArray(NumberOfIterationForVioDec);
min_violations_BasicArray(index)=MinimumVio_BasicMVR;

```



```

min_violations_BasicMatrix(index,:)=Basic_MVR_FINAL_RANK;
hold on
plot(BasicMVR_ViolationArray,'DisplayName','BasicMVR')
title('Violations Change Each Iteration By ViolationDec Function')
xlabel('Number Of Iteration')
ylabel('Number Of Violations')
hold on
plot(AdvancedMVR_ViolationArray,'LineWidth',5,'DisplayName','AdvancedMVR')
lgd = legend;
lgd.NumColumns = 2;
end
minBasic=min(min_violations_BasicArray);
minAdvanced=min(min_violations_AdvancedArray);
for f=1:1:NumberOfSimulations
if(min_violations_BasicArray(f)==minBasic)
break;
end
end
for m=1:1:NumberOfSimulations
if(min_violations_AdvancedArray(m)==minAdvanced)
break;
end
end
Basic_MVR_FINAL_RANK=min_violations_BasicMatrix(f,:);
Advanced_MVR_FINAL_RANK=min_violations_AdvancedMatrix(m,:);
disp('***** BASIC MVR BEST RANK *****')
disp(minBasic);
for f=1:1:20
Temp=Basic_MVR_FINAL_RANK(f);
switch Temp
case 1 disp('Barcelona')
case 2 disp('Atletico_Madrid')
case 3 disp('Real_Madrid')
case 4 disp('Valencia')
case 5 disp('Villarreal')
case 6 disp('Real_Betis')
case 7 disp('Sevilla')
case 8 disp('Getafe')
case 9 disp('Eibar')
case 10 disp('Girona')
case 11 disp('Real_Sociedad')
case 12 disp('Celta_Vigo')
case 13 disp('Espanyol')
case 14 disp('Alaves')
case 15 disp('Levante')
case 16 disp('Athletic_Bilbao')

```

```

case 17 disp('Leganes')
case 18 disp('Deportivo_La_Coruna')
case 19 disp('Las_Palmas')
case 20 disp('Malaga')
end
end
disp('***** ADVANCED MVR BEST RANK *****')
disp(minAdvanced);
for f=1:1:20
Temp=Advanced_MVR_FINAL_RANK(f);
switch Temp
case 1 disp('Barcelona')
case 2 disp('Atletico_Madrid')
case 3 disp('Real_Madrid')
case 4 disp('Valencia')
case 5 disp('Villarreal')
case 6 disp('Real_Betis')
case 7 disp('Sevilla')
case 8 disp('Getafe')
case 9 disp('Eibar')
case 10 disp('Girona')
case 11 disp('Real_Sociedad')
case 12 disp('Celta_Vigo')
case 13 disp('Espanyol')
case 14 disp('Alaves')
case 15 disp('Levante')
case 16 disp('Athletic_Bilbao')
case 17 disp('Leganes')
case 18 disp('Deportivo_La_Coruna')
case 19 disp('Las_Palmas')
case 20 disp('Malaga')
end
end
disp('***** NUMBER OF WINS RANK *****')
for f=1:1:20
Temp=max(ArrayForRankingBasedOnWins(f));
switch Temp
case ArrayForRankingBasedOnWins(1) disp('Barcelona')
case ArrayForRankingBasedOnWins(2) disp('Atletico_Madrid')
case ArrayForRankingBasedOnWins(3) disp('Real_Madrid')
case ArrayForRankingBasedOnWins(4) disp('Valencia')
case ArrayForRankingBasedOnWins(5) disp('Villarreal')
case ArrayForRankingBasedOnWins(6) disp('Real_Betis')
case ArrayForRankingBasedOnWins(7) disp('Sevilla')
case ArrayForRankingBasedOnWins(8) disp('Getafe')
case ArrayForRankingBasedOnWins(9) disp('Eibar')

```

```
case ArrayForRankingBasedOnWins(10) disp('Girona')
case ArrayForRankingBasedOnWins(11) disp('Real_Sociedad')
case ArrayForRankingBasedOnWins(12) disp('Celta_Vigo')
case ArrayForRankingBasedOnWins(13) disp('Espanyol')
case ArrayForRankingBasedOnWins(14) disp('Alaves')
case ArrayForRankingBasedOnWins(15) disp('Levante')
case ArrayForRankingBasedOnWins(16) disp('Athletic_Bilbao')
case ArrayForRankingBasedOnWins(17) disp('Leganes')
case ArrayForRankingBasedOnWins(18) disp('Deportivo_La_Coruna')
case ArrayForRankingBasedOnWins(19) disp('Las_Palmas')
case ArrayForRankingBasedOnWins(20) disp('Malaga')
end
ArrayForRankingBasedOnWins(f)=-9;
end
averageBasic=mean(min_violations_BasicArray);
averageAdvanced=mean(min_violations_AdvancedArray);
```