המחלקה למתמטיקה שימושית

חקירת משחק האורות

מאת:
ולדיסלב ברקנס

מנחה:
אלכס גולוורד

11 בדצמבר 2021

תוכן עניינים

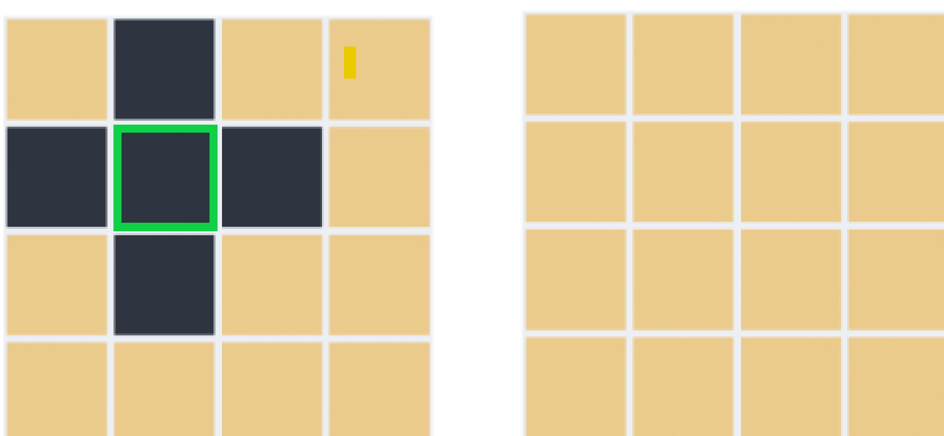
1 הקדמה

2 רקע על משחק האורות

משחק האורות או Lights Out בלועזית, זהו משחק בו יש לוח משבצות ריבועי.
כל משבצת הינה לחצן על הלוח שיכולה להיות בשתי מצבים: דלוק או כבוי.
כאשר לוחצים על משבצת, משבצת הנלחצת וכל משבצות הסמוכות לה כלומר, כל המשבצת בעל צלע משותפת משנות את מצב נוכחי.
משחק מתחיל כשהלוח כולו עם משבצות דלוקות והמטרה לכבות את כל המשבצות על הלוח כולו.

נתאר זאת ויזואלית:

מצב התחלתי (a)   לחיצה על משבצת מסומנת (b)



נבחין כי המשחק 4×4 מתחיל במצב (a).
בלוח (b) נתאר מצב בו לחצו על משבצת המסומנת, בירוק כל המשבצות השכנות והיא משנות מצבן, היות ומצב של כולן היה דלוקות לכן הן נכבו

המשחק במקור היה צעצוע אלקטרוני על לוח 5×5 ששוחרר ב 1995.
המשחק יכול להראות פשוט אבל כפי שתואר במאמר **[1]** "devilish invention".
קיים קושי רב בלמצוא שיטה לפתרון אינטואיטיבי, הקושי של משחק מתבלט בשאלה כיצד כדי להתחיל את המשחק?
בנוסף אציין מניסיון האישי שהמשחק קשה כבר על לוח 5×5 ולרוב אנשים שמשחקים אותו מכירים מצבים על הלוח שיודע עליהם משם את הפתרון.

פרויקט זה באה בעקבות הקושי של המשחק והניסוי להציע שיטות לפתרון, בעקבות

ניסיונות עלו נעזרנו במספר רב של כלים מתמטיים מתקדמים.
אחת המטרות במחקר למצוא הסבר לתופעות במשחק שנתקלנו.

נציין כי קיימים עוד המון שאלות שמשחק מעלה ולא לכולם קיים פתרון, נשמח בפרויקט זה פתרון לכמה מהשאלות שעולות.
חוץ מאתגר של המשחק עצמו קיים אתגר מתמטי שנרצה בפרויקט זה להציג ולהתעניין.

2.1 משחק האורות על גרף

אחרי שכללי המשחק על לוח הובנו אפשר לנסות להכליל את המשחק כמשחק על גרף.
קיימים הרבה סיבות בהם תירצה להגדיר את הבעיה על מבנה כללי שכזה:

1. ככול שמבנה כללי יותר תאוריה שאתה מפתח מתאימה ליותר בעיות.
2. קיימת תאוריה רחבה שפותחה על גרפים ואתכן שנעזר בחלק מהטענות מהתאורה שכזה.
3. מבליט את מהות הבעיה והגדרה הבסיסית ביותר של המשחק.

ארצה להתייחס לנקודה אחרונה, החשיבות הגדולה שאפשר לתאר את הבעיה של משחק כאוסף של כללים על גרף, מרכזת אותנו לבעיה ובסופו של דבר כשנראה את שיטה למציאת הפתרון, השיטה עצמה תזכיר לנו מיד את הייצוג הגרפי.

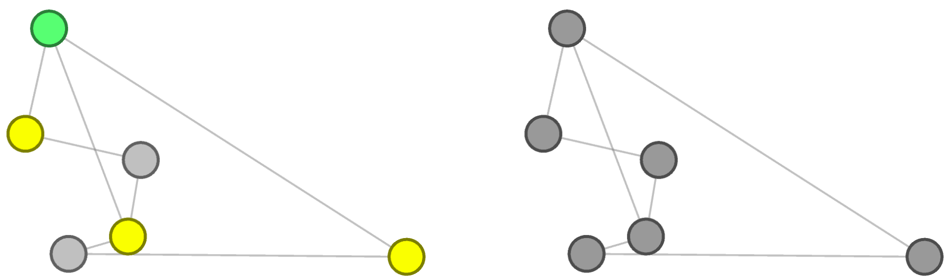
כדי לתאר את משחק האורות על גרף נשתמש באותם כללים שהגדרנו פרט לעובדה שצמתים הם הלחצנים או המשבצות במקרה של הלוח וכל לחיצה הופכת את המצב של הצומת והשכנים שלה.
נזכיר כי צמתים שכנים הם צמתים שיש קשת ביניהם.

נציין כי כאשר כל צומת יכולה להיות בשתי מצבים, דלוקה או כבויה המטרה היא לעבור מכל הצמתים במצב מסוים דלוק למצב אחר כבוי.
העובדה שמצב התחלתי הינו דלוק או כבוי אינה תשנה את המשחק עלה רק לאיזה מצב סופי צריך לעבור לכבוי או דלוק.

נמחיש זאת על דוגמה:

על גרף התחלתי (a) ניתן לראות 6 קודקודיים צבועים באפור ומטרה לצבוע את כולם לצהוב.
בשלב (b) מציגים לחיצה על צומת ירוקה היא ושכניה נצבעים בצהוב.

מצב התחלתי (a)   לחיצה על משבצת מסומנת (b)



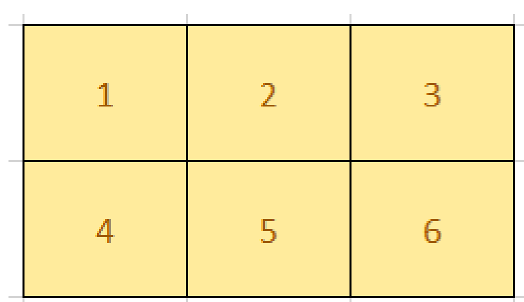
הערה 2.1 *בפועל צומת ירוקה גם נצבעת לצהוב צביעה לירוק נועדה להדגשה על מי נלחץ*

משחק על גרף הינה הכללה של משחק על לוח כלומר, כל משחק לוח ניתן לתאר בעזרת משחק על גרף.

נמחיש זאת על דוגמה:

ניקח לוח למשל 2×3 נמספר את המשבצות כמו באיור 3

איור 3: לוח 2×3

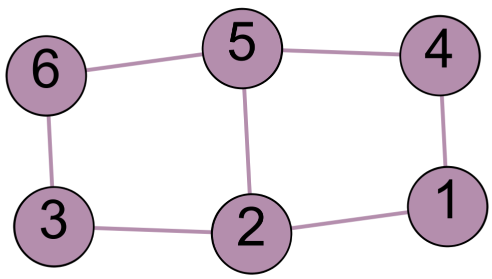


כדי לתאר את הלוח על על משחק גרף נשתמש בשני הכללים הבאים:

1. כל משבצת על משחק לוח נהפוך לנקודה.
2. כל זוג משבצות סמוכות על לוח נחבר את נקודות בקו

הגרף שנקבל עבור לוח באיור 3 מתואר באיור 4

איור 4: גרף 2×3



נשים לב שלגרף בו יש צומת אם יותר מ 4 שכנים לא ניתן לתאר לוח שכזה כיוון שלכל היות במשחק על לוח לכל משבצת יש לכל יותר 4 משבצות סמוכות.

לכן לסיכום אפשר למדל כל משחק לוח על משחק גרף שמייצג אותו אבל ההפך זה לא נכון.

3 אלגוריתם למציאת פתרון

לפני שנציעה לעלות פתרון, נשאל את עצמנו מדוע בכלל צריך למצוא פתרון. הרי בסופו של דבר זה משחק ולהציע פתרון למשחק יפגע במהותו משחק הרי אף אחד לא ירצה לשחק במשהו שידוע מה הפתרון שלו.

הצורך למצוא פתרון הוא נוראה טבעי וזה בעקבות שמששחק עצמו מעניין, כשאתה מתחיל את המשחק על לוח 3×3 המשחק ניראה תמים ופשוט אתה מתחיל לצפות לאיזושהי חוקיות.
בשלב הזה שאתה כבר מנסה לוח 4×4 המשחק מתגלה כלא פשוט כשאתה מנסה לוח בקונפיגורציה כלשהי לא בהכרח התחלתית מהר מאד אתה נעבד.
בשלב מסוים גם לוח 4×4 נהיה מוכר ובאופן תמים תנסה לעבור ללוח 5×5 ומהר מאד הלוח שובר את רוחה. קיימים כל כך הרבה מכירים שנשארת לך משבצת אחת שנותרה לסדר ואינה נעלמת האינטואיציה שחשבת שפיתח על לוח 4×4 נעלמת כאילו למדת לשחק משחק חדש לגמרי.

התופעה הזאת ששינוי גודל מרגיש שהתחלת משחק אחר עוד מורגשת בשלב שאתה מנסה לפתור את מצב התחלה בלוחות שונים

איור 5: פתרונות של משחק על לוחות שונים

איור 5 באה להמחיש את חוסר אינטואיציה כאשר האיור מתאר את פתרון של משחק על הלוח כאשר הלוח במצב התחלתי בו כל נורות דלוקות. כדי שהשחקן ינצח עליו ללחוץ על המשבצות הירוקות.
איור באה להראות שלוחות על קטנים מ 5×5 אתכן ותחשוב שפתרון נוצר על לחיצות סימטריות והאיור ממשיך שזה לא כך כי כאשר מסתכלים על הלוח 5×5 מיד אפשר לראות שפתרון לא ניראה סימטרי.

חוסר האינטואיציה מתבלט גם מהעבודה שכמות הפתרונות משתנה לכל לוח. עבור לוח 3×3 קיים פתרון יחיד, אבל ללוח 4×4 קיים 16 פתרונות. כמה פתרונות היה ללוח 5×5 , האם זה יותר או פחות מלוח 4×4 בהפתעה רבה ללוח 5×5 יש רק 4 פתרונות שזה מפתיע כי אפשר היה לצפות שמספר פתרונות על לוח גדול יותר אגדל.

אפשר להוסיף שעבור לוחות ריבועים כלומר nxn כמות הפתרונות כל כך לא צפויה כי עברו $n \in [1,20]$ מספר הפתרונות הגדול ביותר הוא ללוח $19x19$ ומספר פתרונות הוא 65536. מספר הפתרונות השני הגדול ביות הוא רק 256.

חוץ מבעיית חוסר אינטואיציה לחיפוש פתרון טבעי אפשרי לנסות פתרון נאיבי המנסה כל לחיצה .
פתרון הנאיבי נפסל ברגע הזה שחושבים על כמה קומבינציות לחיצה קיימות.

למה 3.1 *כמות האפשרויות לחיצה על לוח $m \times n$ הוא* $2^{m \cdot n}$

נומר שאפשרויות לחיצה זה חסם על כמות המצבים האפשריים שמשחק יוכל להיות. חסימה זאת נובעת משאלה אם שחקן לוחץ על לחצן כמה יכול להשפיע על הלוח. מובן שאם שחקן לחץ על לחצן מספר זוגי של פעמיים המצב יחזור למצב שהיה. לכן, כל לחצן משפיע על הלוח אם הוא נלחץ או לא כלומר, יכול להיות בשתי מצבים. היות וללוח $m \times n$ קיים $m \cdot n$ לחצנים , היות וכל לחצן יכול להיות בשתי מצבים שונים לכן נקבל שמספר אפשרויות לחיצה ללחצן 2 ול mxn לחצנים $2^{m \cdot n}$.

כבר בלוח $6x6$ כמות אפשרויות לחיצה גדולה מכמות הסטנדרטית שמציגים מספר שלמים, 4 בתים או 2^32 מספרים, המטרה של המחשה זה להדגיש כמה לא פרקטית אופציית הפתרון שכזה.

עכשיו שיש לנו מוטיבציה למצוא פתרון השאלה היא באיזה כלים בעבודה זה נציע להסתכל על שיטה הממדל את הבעיה לשדה לינארי ולעזר בכלים של אלגברה לינארית למצוא מספר פתרונות שונים.

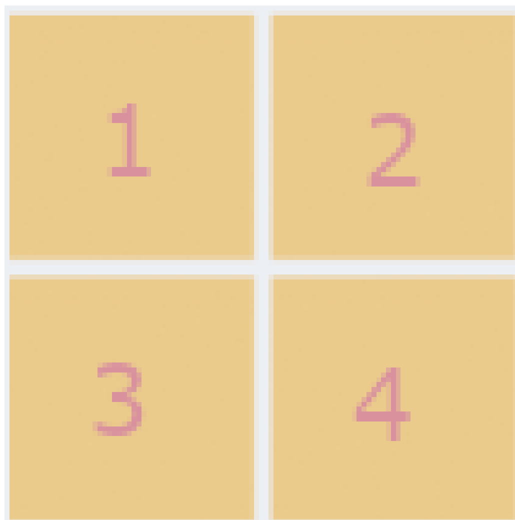
אתכן ויש כמה דרכים להגיע לאותה מודל לינארי שנציע, נציג בעובדה זה שני דרכים אחת בעזרת וקטורי שינוי ומניסיון למצוא צירוף לינארי של וקטורים עלו נמצא את פתרון, דרך שניה תהיה לפי מערכת משוואות שמתארת את הבעיה.
בשני הדרכים נראה שהגנו לאותה מערכת משוואות ונפתור את המשחק בעזרת חיפוש פתרון של המערכת.
בחרנו להציג קודם בעזרת וקטורי שינוי משום שהגדרת וקטורים פשוטה יותר להסבר לאחר שניראה את הדרך הראשונה הדרך השנייה קלה יותר להסבר.

כדי למדל את הבעיה על שדה לינארי נזכר בייצוג גרפי שאומר כי לחיצה על צומת משנה את הצומת ושכניה אם נסמן את צמתים ב n_i אז אפשר לתאר כי המצב אתחלתי של משחק על גרף הוא שכל צומת אם הערך התחלתי $n_i = 0$ וכל צומת יכול לקבל 2 ערכים שנסמן אותם ב $0, 1$ כאשר 0 מצב התחלתי שכל צמתים התחילו ו 1 מצב סופי של משחק המשחק מסתיים כשכל הצמתים מקיימים $n_i = 1$. לכן אנחנו עובדים על שדה בינארי.

אפשר לתאר לחיצה על צומת i כווקטור שינוי ערכי צמתי שכנים

לדוגמה ניקח משחק בגודל $2x2$ נמספר את הצמתים שורות ואז עמודות מלמעלה למטה כלומר כמו מתואר באיור **6**

איור 6: מספור לוח



הערה 3.1 *מספור לפי שורה עליונה על כל עמודות עד לשורה תחתונה תהיה שיטת המספור לאורך כל ספר*

לאחר מספור שכזה נוכל לומר שלחיצה על משבצת 1 וקטור שינוי שלה היה $t_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$

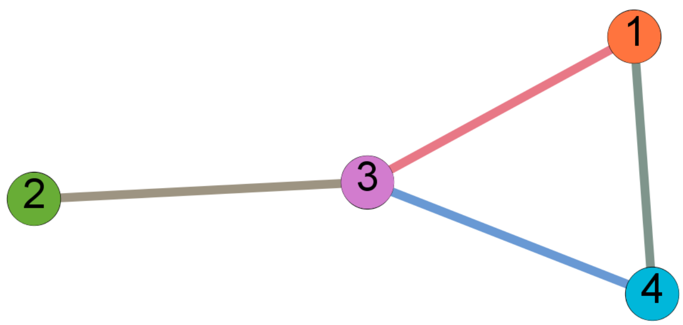
t_1 מתאר את עלו צמתים יכול שינוי

עבור גרף וקטור שינוי הי כדומה.

עבור גרף באיור מתקבל וקטור שינוי של צומת 1 כך **7** $t_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

הגדרה 3.1 *וקטור השינוי שנסמן ב $\vec{t_j}$ של לחצן ממוספר i.*
וקטור שייך לשדה ${F_2}^n$ כאשר F_2 שדה בינארי ו n הינו מספר הלחצנים. נגדיר שערכיו $t_{i,j} = 1$ עבור לחצנים j שכנים עליו ועל עצמו ושאר ערכי וקטור שווים ל 0.

איור 7: גרף ממוספר



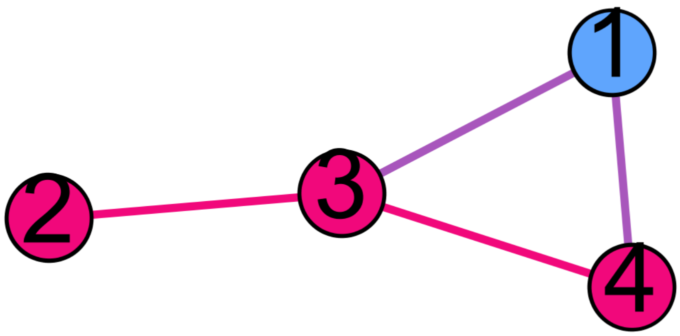
הערה 3.2 *היות ווקטור שינוי שדה* $F_2{}^n$ *חיבור בין וקטורים הינו חיבור בין האינדקסים מודול* 2 *וכפל בסקלר הוא לכפול את כל ערכי וקטור בסקלר כאשר הסקלרים יכולים להיות* 0 *או* 1

נזכיר שלחצנים שכנים הם לחצנים שמשתנים אתו באת לחיצה אם זה במקרה הגרפי צמתים שכנים כלומר בעלי צומת משותפת או במקרה הלוח זה לחצנים בעלי צלע משותפת.

בעזרת וקטורי השינוי אפשר לתאר תוצאה של קומבינציות של לחיצות בעזרת צירוף לינארי של וקטורי שינויים. את המצב המתקבל נוסיף למצב הקיים ונקבל את השינוי שנוצר בלחיצה של כפתורים. נדגים רעיון זה על איור 8 .

נניח שצומת 1 היא יחידה שדלוקה.
נרצה להראות איך הגרף יראה עם ילחצו על כפותרים $1, 3$.

איור 8: מצב התחלתי



$$t_1 + t_3 = \begin{bmatrix}1\\0\\1\\1\end{bmatrix} + \begin{bmatrix}1\\1\\1\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$

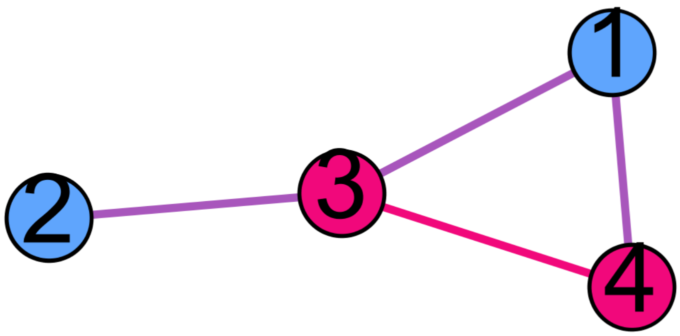
ומצב התחלתי שמתואר באיור נסמן ב S_0 לכן מתקבל

$$S_0 + t_1 + t_3 = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\1\\0\\0\end{bmatrix} = \begin{bmatrix}1\\1\\0\\0\end{bmatrix}$$

וקטור התוצאה שהתקבל אכן תואם לתוצאה המצופה מתואר באיור 9.

נשים לב שעבור איך שהגדרנו את המשחק Light out המשחק מתחיל כשכול נורות דלוקות או כבויות ומטרה היא לכבות או להדליק את כל נורות.
כפי שאמרנו שני המשחקים שקולים ההבדל רק מה המשמעות שנותנים לערכים $0, 1$ ומהו מצב הלוח התחלתי בהגדרה זה.

איור 9: מצב לאחר ביצוע הלחיצות



מפרק זה נגדיר שאנחנו פותרים את המשחק שלוח התחלתי כולו באפסים ומטרה להגיע ללוח שכולו אחדים. תיאור הטבעי למשחק שכזה להתחיל מלוח כבוי ומטרה להדליק את כל הנורות

מתיאור שכזה מובן כי S_0 זהו וקטור אפסים לכן כדי לתאר את ממצב התחלתי למצב לצירוף לינארי של וקטורי שינוי אין צורך לחבר בין מצב התחלתי וצירוף לינארי.

היות ומתקיים:

$$S_0 + \sum_{j=1}^{n} a_i \vec{t_j} = \sum_{j=1}^{n} a_j \vec{t_j} \quad \textbf{(1)}$$

בעקבות כך ניתן לתאר את בעיית המשחק לצורה הבאה:

$$\sum_{j=1}^{n} a_j \vec{t_j} = \vec{1} \tag{2}$$

כאשר $\vec{1}$ וקטור שכל ערכיו אחדים ו n מספר הצמתים בגרף.

תיאור שכזה מדגיש מספר תכונות

למה 3.2 *סדר הליחצות לא משנה את התוצאה הסופית*

נשים לב שאם ידוע קבוצת לחיצות ממצב התחלתי הערך של משבצת מסוימת נקבע לפי הערך של וקטור תוצאה בנוסחה 1. נשים לב שתוצאת הסכום איננה תלויה בסדר הלחיצות של הוקטור.

מערכת משוואות שמתוארת בנוסחה 2 אפשר לתאר במספר צורות ונפוצה מבינהם היא בעזרת מטריצה כמו שמתואר בנוסחה 3.

$$\begin{bmatrix} t_1 & t_2 & \cdots & t_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ t_{i,j} & t_{i,2} & \cdots & t_{i,n} \\ \cdots & \cdots & \cdots & \cdots \\ t_{n,1} & t_{n,2} & \cdots & t_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix} \tag{3}$$

הגדרה 3.2 *תהי משחק ומערכת משוואת שמתארת אותו מצורה 3 נגדיר את וקטור פתרון כאוסף הלחצנים ומצב להגיע לפתרון ונסמן אותו ב* $\vec{x}$

הגדרה 3.2 מתכוונת שאם מתקבל $\vec{x}$ וקטור פתרון של המערכת ו $x_i = 1$ אז המשמעות שכדי לפתור את המשחק צריך ללחוץ על לחצן i.
בנוסף $\vec{x}$ אחד הפתרונות אתכן והיו כמה.

תוצאה דומה אפשר לראות בפרק מהספר [2].
מרגע שהצלחנו לתאר את הבעיה מערכת משוואות לינארית על שדה ${F_2}^n$ מכן נוכל

להעזר בכלים של אלגברה לינארית כדי למצוא את הפתרון כמו מציאת פתרון בעזרת דירוג, מציאת מטריצה פסאדו הפוכה וכולי.

הערה 3.3 *עבור לוח* $[n \times n]$ *כמות הלחצנים* n^2 *ולכן גודל המטריצה שתוארה במשוואה 3 היא* $[n^2 \times n^2]$

כלומר כמות הזיכרון שצריך כדי לשמור מטריצה ללוח ריבועי שאורך צלע הוא n דורש לשמור מטריצה עם n^4 ערכים .

3.1 פתרון בעזרת שיטה הספרדית

הצגנו גישה פתרון בעזרת תיאור המערכת בעזרת וקטורים וחיפוש פתרון על ידי פתירת מערכת משוואות לינאריות. כפי שתואר בעבודה [2] החידה הוצגה כצעצוע ומתאימה למודל שקראנו לו משחק על לוח ריבועי. לפי הערה **3.3** כמות המידע שצריך לשמור גדל בקצב n^4 כאשר n כמות הלחצנים לשורה.
במאמר [1] מציג שיטה שמציאה פתרון עם מערכת משוואות לינארית אחרת שמובילה גם היא לפתרון. מערכת משוואת לינארית שמאמר [1] מתאר ממדי $n \times n$. כלומר כמות הערכים המטריצה המתקבלת שווה ל n^2 שקטנה בריבוע ממערכת שנתונה במשוואה 3.

בפרק זה נציג את הגישה שמתוארת [1] נתאר כמה הבחנות שיסתמכו ששני הגישות שקולות ושגישה החדשה הינה רק אופטימיזציה ספציפית לצורה של מטריצה הנתונה. את הגישה החדשה ניקרא לאורך כל הפרק הגישה הספרדית.

נתאר את הגישה הספרדית בעזרת דוגמה על לוח 3×3 זה הינה גישה כללית ללוח משחק בגודל n כלשהו פשוט מקל על ניסוח והבנה.

נניח שאיור **10** מתאר את הלוח הנתון כאשר המשבצות הינן הלחצנים ומספור זה אינדקסים הערה **3.1** כאשר מתחילים את המספור מ 0 נפוץ בשפות תוכנה להתחיל מאינדקס 0.

שיטת הפתרון של מאמר הספרדי הינה להתחל את הלוח עם משתנים ולמלאות את הלוח במשתנים עלו. לאחר שכל הלוח מלא מתקבלים n משוואות במקרה של דוגמה 3 וכל משוואה תהיה עם n נעלמים ולכן נקבל מערכת משוואת שמטריצה המייצגת הינה מסדר $n \times n$

שיטת הספרדית מתחיל בכך שממלאים את השורה העליונה במשתנים כפי שמתואר באיור **11**. לאחר מכן עוברים שורה שורה וממלאים אותה במשתנים שמשפיעים על הלחצן.

איור 10: לוח 3×3 עם אינדקסים

0	1	2
3	4	5
6	7	8

בשלב זה נסביר את אופן המילוי ומשמעות המשתנים.

ונסמן ב x_i אם נילחץ על לחצן i. אם נתייחס לכל הלחצנים כווקטור מסודר לפי אינדקס i נקבל $\vec{x}$ כפי שהגדרנו בהגדרה **3.2**.

נתסכל על איור **10** היות ומטרה להפוך את מצב הלחצנים כלומר צריך שכמות לחיצות על לחצנים שמשפיעים על משבצת סכום במודול 2 היה 1 כי הלחצן אשאר דלוק.

נדגים על כמה לחצנים מאיור **10**. מצב הלחצן תלוי האם הלחצנים סמוכים לו ועצמו לחוצים .
עבור לחצן 0. מתקבלת המשוואה:

$$x_0 + x_1 + x_3 = 1$$

ועבור לחצן 3:

$$x_0 + x_3 + x_4 + x_6 = 1$$

וכך ניתן להגדיר אילוצים לכל הלחצנים.

הגדרה 3.3 *המשוואה המתקבלת עבור לחצן i מחיבור מודל 2 עם לחצות על הלחצנים המשפעים על לחצן תקראה משוואת אילוץ על לחצן i*

עבור משחק לוח ריבועי באורך שורה n שהלצנים ממספרים לפי הערה **3.1** ניתן לנסח

בנוסחה פשוטה:

$$x^*_{i-n} + x^*_{i-1} + x^*_i + x^*_{i+1} + x^*_{i+n} = 1 \quad x^*_i = \begin{cases} x_i & \text{if } i \in [1, n^2] \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

איור 11: לוח 3×3 מאותחל

x0	x1	x2
3	4	5
6	7	8

לאחר שהגדרנו את משוואת האילצים נסביר כיצד למלאה את השורות לפי השיטה הספרדית.
כל לחצן ימלאה לפי משוואת האילוצים של הלחצן שמעליו.
כלומר כדי למלאה את לחצן 3 נסתכל על משוואת האילצים של לחצן 0.

$$x_0 + x_1 + x_3 = 1$$

ניזכר כי המשוואה שהתקבלה הינה על שדה מודלו 2. לכן בעזרת העברת אגפים מתקבל.

$$x_3 = 1 + x_0 + x_1$$

ונכתוב ערך זה בלחצן 3 כמו שמתואר באיור **12**

נשים לב שבעזרת גישה שתיארנו כרגע נוכל למלאה כל השורה שניה. כל שורה תלויה בשורה מלפניך לכן כך נוכל למלאה את כל השורות כמו שמתואר באיור **13**

נדגים מילוי משבצת 6 משורה שלישית לכן נצתרך ששורה שניה חושבה.
ניסתכל על משבצת מעל כלומר משבצת 3 וניסתכל למה שווה משוואת האילצים שלה:

איור 12: לוח 3×3 מילוי משבצת 3

x0	x1	x2
1 + x0 + x1	4	5
6	7	8

$$x_0 + x_3 + x_4 + x_6 = 1$$

לכן

$$x_6 = 1 + x_0 + x_3 + x_4$$

היות ושורה שניה מולת וידוע שערך משבצות באותה שורה:

$$x_3 = 1 + x_0 + x_1$$
$$x_4 = 1 + x_0 + x_1 + x_2$$

נציב ערכים אילו

$$x_6 = 1 + x_0 + (1 + x_0 + x_1) + (1 + x_0 + x_1 + x_2)$$
$$x_6 = 1 + x_0 + x_2$$

כדומה נעשה לשאר הערכים. התוצאה מתקבלת מתוארת באיור 13

איור 13: לוח 3×3 מלאה

x0	x1	x2
1 + x0 + x1	1 + x0 + x1 + x2	1+ x1 + x2
1 + x0 + x2	0	1 + x0 + x2

איור 14: לוח 3×3 מלאה כולל שורה וירטואלית

x0	x1	x2
1 + x0 + x1	1 + x0 + x1 + x2	1+ x1 + x2
1 + x0 + x2	0	1 + x0 + x2
1 + x1 + x2	x0 +x1 + x2	1 + x0 + x1

נבחין שלאחר שמילאנו את כל הלוח כמו שמתואר באיור 13 נותרו לנו עוד n משוואות אילוץ שתלויות בשורה אחרונה ולכן מאמר [1] מציאה להוסיף שורה וירטואלית כדי להשתמש במשוואות עלו כמו שמתואר באיור 14

המאמר טוען שהיות שורה זה לא באמת קיימת לכן ערכי של משבצת המתקבלות חייב להיות שווה ל 0

ולכן קיבלנו n משוואות על n נעלמים כאשר בדוגמה שלנו $n = 3$ לכן אפשר לנסות לפתור את המערכת הנתונה.

המאמר [1] מתאר מספר רב של פתרונות בלוחות ריבועים בגדלים שונה ואפילו על לוחות מלבניים.
האתגר המרכזי בשיטה הספרדית היא להצדיק אותה למה יש שורה וירטואלית והאם יש קשר בין שני השיטות.
בשלב זה נתרכז להראות את הקשר בין שיטה הספרדית ושיטה שהצגנו בפרק הקודם.

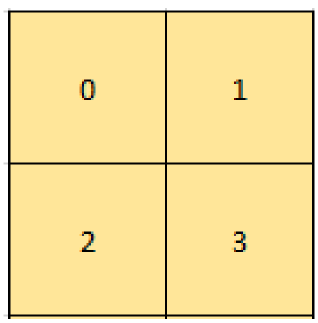
משפט 3.1 *משוואות האילוצים מתארות את מערכת המשוואות 3.*

משוואות האילוצים פורמלית היא לב השיטה הספרדית מכיוון שהתקדמות בשורות מבוססת על המשוואות עלו.

אם נפרוס את משוואת האילצים נקבל גם מערכת משוואת שפותרת את המשחק אבל כמות המשוואות הינה n^2. נבחין שאם נציג אותם כמטריצה נקבל את מטריצה במשוואה 3 וזאת מכיוון שאם נמספר ב j את כל המשוואות לפי האינקס i.
מתקבל משוואת האילצים שמשתנה x_i מופיעה בהם באותם אינדקסי j שבהם וקטור השינויי של לחצן i מקבל ערכים שווים ל 1.

נדגים זאת על לוח 2×2 שמתואר באיור

איור 15: לוח 3×3 מלאה כולל שורה וירטואלית



וקטורי השינויים:

$$t_0 = \begin{bmatrix}1\\1\\1\\0\end{bmatrix}, t_1 = \begin{bmatrix}1\\1\\0\\1\end{bmatrix}, t_2 = \begin{bmatrix}1\\0\\1\\1\end{bmatrix}, t_3 = \begin{bmatrix}0\\1\\1\\1\end{bmatrix},$$

לכן המטריצה מהצורה:

$$\begin{bmatrix}1 & 1 & 1 & 0\\1 & 1 & 0 & 1\\1 & 0 & 1 & 1\\0 & 1 & 1 & 1\end{bmatrix},$$

ומשוואת האילוצים מהצורה

$$x_0 + x_1 + x_2 = 1$$
$$x_0 + x_1 + x_3 = 1$$
$$x_0 + x_2 + x_3 = 1$$
$$x_1 + x_2 + x_3 = 1$$

תעלומה נוספת בשיטה הספרדית היא למה צריך שורה וירטואלי, למה ערכה שווה ל 0 ואיך המערכת משוואת שלו מצטמצמת ל n משתנים
הסבר לתופעה זה ניתן בעזרת הצגה המטריצה נראה זאת על מטריצה של משחק על לוח 3×3

נשים לב ששיטה הספרדית מדרג את המטריצה מורחבת $[M|\vec{1}]$.

$$\left[\begin{array}{ccccccccc|c}
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1\\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1\\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1\\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1\\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1\\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1\\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1\\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1
\end{array}\right]$$

נבחין כי מילוי משבצת בשיטה הספרדית היא שקולה להצבה שורה קודמת. פעולה דומה להצבה היא דירוג שורה לפי משתנה זה.
כלומר מתקבל שזה אותה ששיטה הספרדית הינה שיטה חכמה לדרג את הבעיה עד שנותר n משוואות.

המטריצה המדורגת המתקבלת מלוח 3×3

1	1	0	1	0	0	0	0	0	1
1	1	1	0	1	0	0	0	0	1
0	1	1	0	0	1	0	0	0	1
1	0	1	0	0	0	1	0	0	1
0	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0	1	1
0	1	1	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	1

נבחין שמשבצת שחושבו שהם משבצות מ 3 עד 8 באיור **10** בשיטה הספרדית שקולה לשורה $i-3$ במטריצה המורחבת המדורגת.

לסיכום השיטה הספרדית היא שקולה לדירוג חכם של המטריצה. אפשר לומר שמספיק היה לדרג רק n שורות אחרונות ומטריצה תהיה מדורגת אין צורך את כל השלבים.

לפי חישוב סיבוכיות לדרג מטריצה בגודל $n^2 \times n^2$ זה $O(n^4)$ דירוג בעזרת שיטה הספרדית אומרת שעל כל עמודה וקטור שינויים יש לכל יותר 5 ערכים ששווים 1.
לכן הסיבוכיות $O(n^2)$.
אם נעבוד בשיטה שרק צריך לדרג את n שורות אחרונות הסיבוכיות $O(n^3)$.

ננסה להראות זאת בפועל על ידי חישוב זמני חישוב.

4 הוכחת קיום פתרון עבור כל גרף

4.1 מספר הפתרונות עבור כל גרף

5 פתרון מינימלי עבור לוחות מלבניים

6 תוצאות ומסקנות

7 נספחים

רשימת מקורות

[1] ALL LIGHTS AND LIGHTS OUT An investigation among lights and shadows by SUMA magazine's article by Rafael Losada Translated from Spanish by Ángeles Vallejo

[2] Lecture 24: Light out Puzzle , SFU faculty of scienc department of mathematics