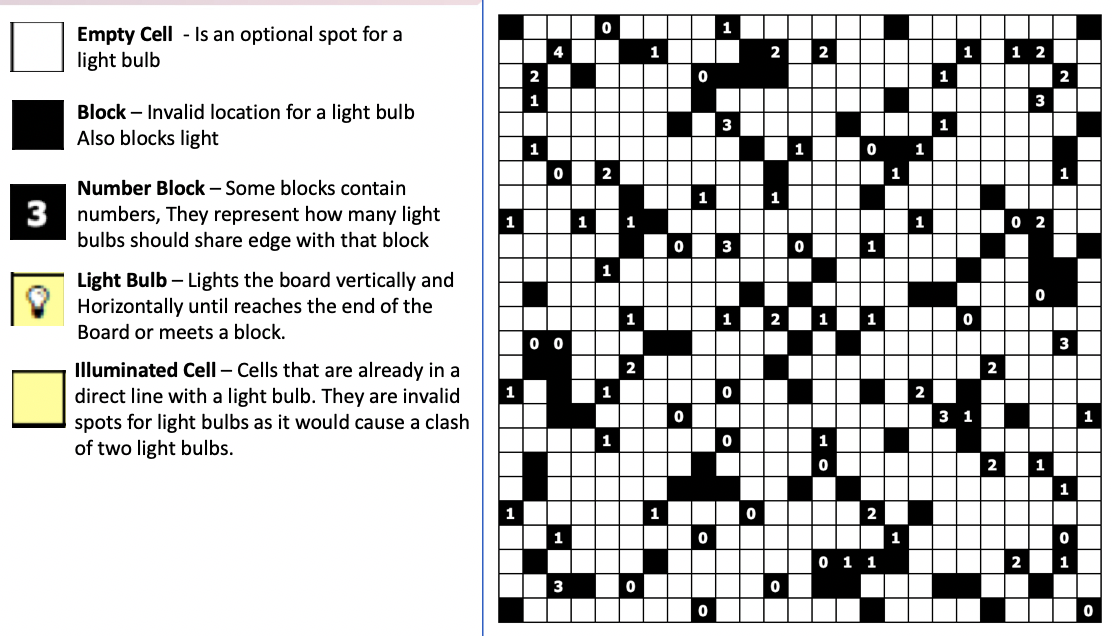
אלגוריתמים אבולוציונים – דו״ח מסכם

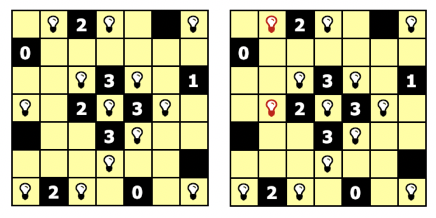
**הקדמה:**

בחרנו לבצע את הפרויקט בקורס על המשחק Lights Up Puzzle, למשחק זה מספר דרכי פתרון ביניהן אלגוריתמי חיפוש וחיפוש היוריסטי, אלגוריתמים אבולוציוניים ופתרונות היברידיים שמשלבים יתרונות משני הסוגים כשלמעשה מדובר בבעיית סיפוק אילוצים מורכבת. לוח המשחק בגודל NxN ומכיל משבצות ממספר סוגים, משבצות ריקות, בלוקים, בלוקים עם מספר, נורות ומשבצות מוארות. התנאים ההכרחיים לפתרון המשחק הם כדלהלן:

1. לכל בלוק עם מספר ישנן בדיוק אותו מספר של נורות צמודות אליו.
2. כל התאים מוארים.
3. אף נורה לא מאירה (מתנגשת) עם נורה אחרת.



**תרשים 1:** לוח המשחק וסוגי המשבצות השונים

****

**תרשים 2:** פתרון לא תקין (ימין) ופתרון תקין (שמאל) לאותו לוח משחק

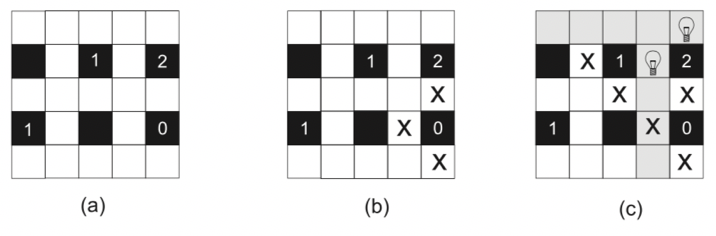
**סקר ספרות:**

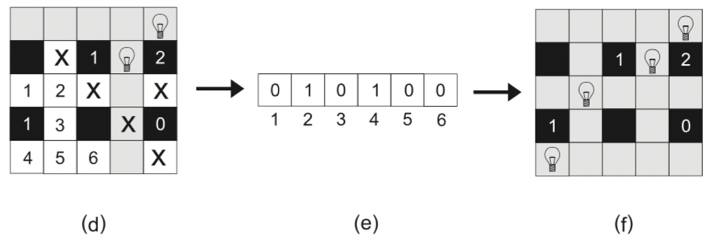
1. Np complete

ישנם מספר מחקרים קודמים שנערכו על המשחק והדרכים האפשריות לפתור אותו. מציאת פתרון ללוח המשחק נחשבת לבעיה np-שלמה כמו שציינו קנדל ושותפיו במחקר שערכו על משחקים np שלמים [1]. מרחב החיפוש של הבעיה הוא עצום ומכיל מספר רב מאוד של מצבים או פתרונות אפשריים (לא בהכרח תקינים), מרחב החיפוש של לוח כלשהו בגודל NxN תחום ב - . ניתן למעשה לבצע אופטימיזציות והקטנות למרחב החיפוש ולהגיע למרחב חיפוש בגודל -. *מדובר כמובן בהקטנה משמעותית של לוח החיפוש, אך בלוחות גדולים גם הצמצום הזה של מרחב החיפוש עשוי להשאיר מרחב חיפוש עצום בגודלו.*

1. פתרונות באמצעות חיפוש ויוריסטיקות
2. פתרונות גנטיקה טהורה
3. פתרונות היברידיים

סנצ׳ו ושותפיו מציעים דרך נוספת לפתרון המשחק במחקרם [2], תחילה נעשה עיבוד מקדמים ללוח המשחק, ומוצבות נורות אשר מיקומן ודאי כבר בשלב זה – למשל סביב בלוק עם המספר 4 (כלומר נורה בכל אחד מצלעותיו של הבלוק), או בלוקים עם המספר 3 שקיימים רק שלושה מיקומים לנורות סביבם (למשל צמודים לאחת מדפנות הלוח). בנוסף מסמנים ב -  X משבצות שאינם יכולות להכיל נורות (למשל כל המשבצות הצמודות לצלעותיו של בלוק 0). לאחר ביצוע העיבוד בצורה איטרטיבית ולאחר שלא נשארו נורות שמיקומן על הלוח ודאי, משתמשים במספר התאים החסרים לקביעת אורכו של הגנום בו ישתמש האלגוריתם האבולוציוני בשלב הבא. למשל, במקרה בו נשארו שישה תאים פנויים, יהיה הגנום באורך של 6 תאים, כך ש – 0 מסמן תא ריק ו – 1 מסמן נורה.





**הניסוי:**

1. מימוש – למעשה בחרנו לממש אלגוריתם עם עקרונות דומים לזה שהציעו סנצ׳ו ושותפיו, בחרנו לבצע עיבוד מקדים ללוח המשחק ובעזרת התאים שנשארו לקבוע את אורך הגנום. בנוסף שמרנו את מיפוי התאים החסרים על מנת להלביש את הגנומים חזרה על הלוח על מנת לבחון את הפנוטיפ. במאמר המוזכר לעיל מצאו פתרונות למספר לוחות 14x14 (בנוסף ללוחות קטנים יותר). אנחנו מעוניינים למצוא פתרונות גם ללוחות גדולים יותר (למשל 25x25(. תחילה בחרנו לממש את האלגוריתם באמצעות Heuristic Lab, פריימוורק מוכר על בסיס C# למימוש אלגוריתמים אבולוציוניים, אך לאחר מכן בין היתר עקב קשיים טכניים (שנינו עובדים על מערכות הפעלה שאינן וינדוס ו – Heuristic Lab לא עובד בצורה טובה בסביבות וירטואליות) בחרנו לממש את כל סביבת ההרצה בעצמנו באמצעות Node JS. הסיבה לבחירה ב - Node JS היא שידענו שאין חבילות שאנו צריכים ולא קיימות (לא השתמשנו בחבילות יעודיות כלשהן), ועל מנת לתרגל שפת תכנות ששנינו לא כתבנו בה זמן רב. מעבר לכך, אין סיבות טכניות לבחרה ב – Node JS לפרויקט זה.
2. לוחות – ידענו שנרצה להריץ את האלגוריתם על מספר מגוון ורחב של לוחות בגדלים שונים. לשם כך כתבנו סקריפט שלמעשה מעתיק לוחות מהאתר - <https://www.puzzle-light-up.com/> ומפרסר אותם לאובייקט איתו נוכל לעבוד, כאשר הלוח מיוצג על ידי מערך דו מימדי של מספרים שלמים ותאיו מיוצגים כלהלן:

**let** CellType = { "BLOCK":-1, //A Bock without a number  
 "ZERO\_LIGHTS":0, //Block with the number 0  
 "ONE\_LIGHT":1, //Block with the number 1  
 "TWO\_LIGHTS":2, //Block with the number 2  
 "THREE\_LIGHTS":3,//Block with the number 3  
 "FOUR\_LIGHTS":4, //Block with the number 4  
 "NO\_LIGHT":5, //Empty cell  
 "LIGHT\_BULB":6, //Light Bulb  
 "LIGHT":7, //Illuminated cell  
 "INVALID":8}; //Invalid bulb location (pre-processing only)

1. אבולוציה
2. פרמטרים
3. תוצאות
4. גרפים

**מסקנות:**

**מקורות:**

[1] KENDALL, Graham; PARKES, Andrew; SPOERER, Kristian. A survey of NP-complete puzzles. *ICGA Journal*, 2008, 31.1: 13-34.

[2] SALCEDO-SANZ, Sancho, et al. A nested two-steps evolutionary algorithm for the Light-up puzzle. *Icga Journal*, 2009, 32.3: 131-139.‏

‏