Light-Up(Akari) Solver

מיני פרויקט "נושאים במשחקי חשיבה"

מנחה: פרופסור דניאל ברנד
המחלקה למדעי המחשב
אוניברסיטת בן גוריון בנגב

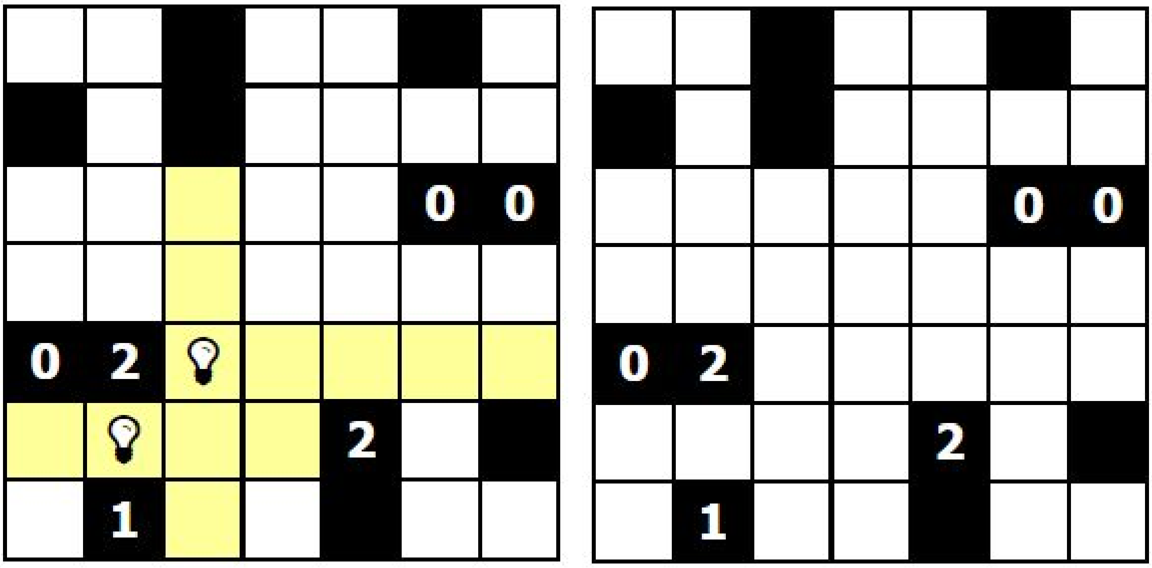
מגישים:
בר סימן טוב
מאיר יאגייאב

על המשחק

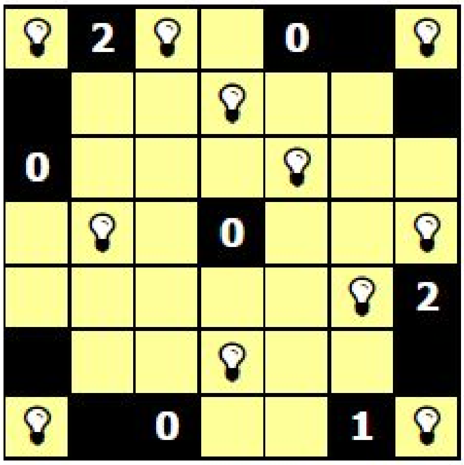
לוח המשחק הינו מטריצה ריבועית (n*n), המחולקת ל n^2 ריבועים קטנים אותם נכנה "תאים". "תאים" יכולים להיות (במצב התחלתי) לבנים (בהם ניתן למקם נורות), שחורים ממוספרים או שחורים לא ממוספרים. תאים שחורים מהווים מחסומים בין התאים הלבנים. כאשר ממקמים נורה במשבצת לבנה, היא מאירה את השורה והעמודה בה היא נמצאת על למחסום הקרוב, והופכת את התאים הללו לצהובים.מטרת השחקן היא למקם נורות בלוח תחת ההגבלות הבאות

1. על מנת לסיים את המשחק על כל המשבצות להיות מוארות (כלומר צהובות)
2. מסביב לכל משבצת ממוספרת יהיו מספר נורות בדיוק כמספר הרשום במשבצת
3. שתי נורות לא תארנה אחת את השנייה (כלומר, בכל רצף אופקי/אנכי של משבצות לבנות, לא תהיה יותר מנורה אחת)

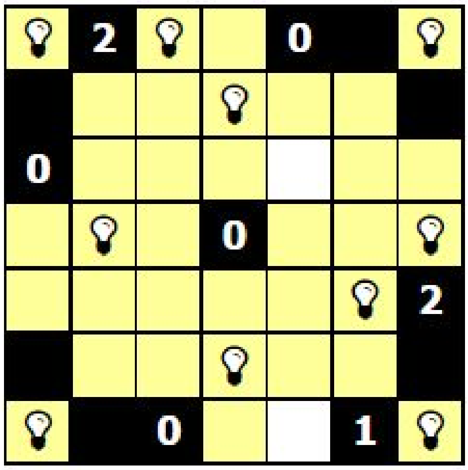
דוגמא ללוח התחלתי, ולאחר הצבה של שתי נורות:



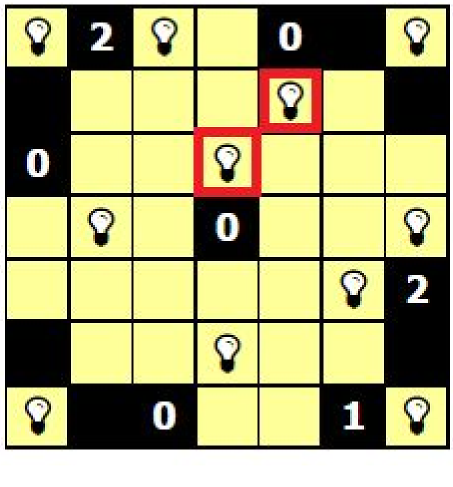
דוגמא לפתרון חוקי:



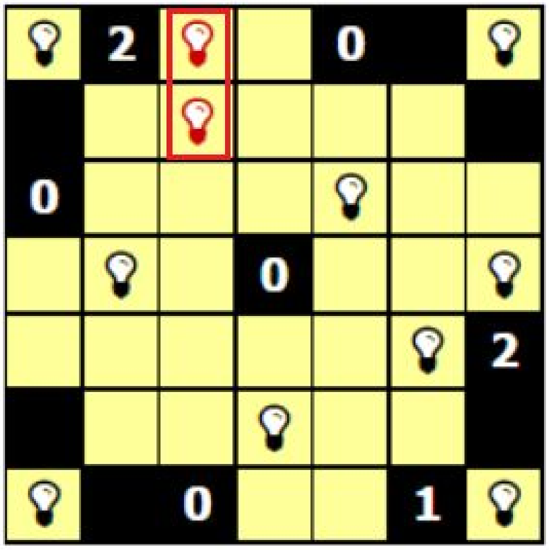
דוגמא להצבה שאינה מהווה פתרון בגלל כלל 1:



דוגמא לפתרון לא חוקי בגלל כלל 2 (הנורות שמוקפות באדום סותרות את המספר שרשום במשבצת הממוספרת הסמוכה אליהן):



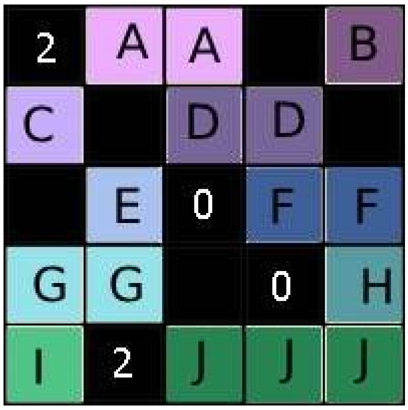
דוגמא לפתרון לא חוקי בגלל כלל 3 (הנורות הצבועות באדום ומוקפות מאירות זו את זו):



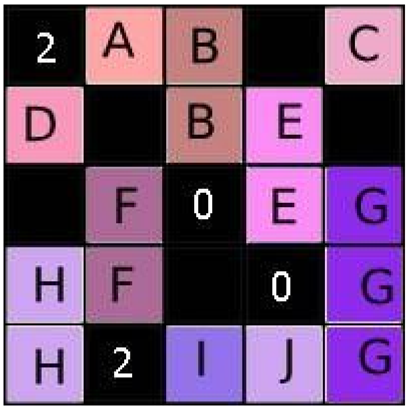
המשחק הוכח כNP שלם באמצעות רדוקציה לבעית Circuit-SAT בזמן פולינומי[1].

<u>הגדרות</u>

בלוק אופקי - רצף כל המשבצות הלבנות באותה שורה, בין גבול/משבצת שחורה לבין גבול/משבצת שחורה, שאינו מופרד על ידי משבצת שחורה נוספת. כלומר - כל המשבצות הלבנות בעלות אותם ערכי y, וערכי x עוקבים, אשר מתחיל במשבצת בעלת ערך ה-x הנמוך ביותר שאינו גבול/משבצת שחורה, ומסתיים במשבצת בעלת ערך ה-x הגבוה ביותר שאינו גבול/משבצת שחורה.



בלוק אנכי - רצף כל המשבצות הלבנות באותו טור, בין גבול/משבצת שחורה לבין גבול/משבצת שחורה, שאינו מופרד על ידי משבצת שחורה נוספת. כלומר - כל המשבצות הלבנות בעלות אותם ערכי x, וערכי y עוקבים, אשר מתחיל במשבצת בעלת ערך ה-y הנמוך ביותר שאינו גבול/משבצת שחורה, ומסתיים במשבצת בעלת ערך ה-y הגבוה ביותר שאינו גבול/משבצת שחורה.



בניית לוח

בהנתן המספרים n, b, r, ניתן לבנות בקלות לוח כלשהו בגודל n*n, עם b משבצות שחורות לא ממוספרות, ו-r משבצות שחורות ממוספרות:

1. נגריל את המיקום של כל משבצת שחורה לא ממוספרת מבין המשבצות שעדיין לא מוקמה בהן משבצת שחורה.
2. נגריל את המיקום של כל משבצת שחורה ממוספרת מבין המשבצות שעדיין לא מוקמה בהן משבצת שחורה (בין שממוספרת או לא ממוספרת).
3. נגריל את המספר לכל משבצת שחורה ממוספרת (מבין המספרים 0-4).

במרבית המקרים נקבל לוח לא פתיר. מכיוון שנעסוק בעיקר במציאת פתרונות עבור לוחות בהם קיים פתרון, נרצה להגריל לוח שהוא בהכרח **פתיר**.

נעשה זאת באופן הבא:

1. נגריל את המיקום של כל משבצת שחורה לא ממוספרת מבין המשבצות שעדיין לא מוקמה בהן משבצת שחורה.
2. נגריל את המיקום של כל משבצת שחורה ממוספרת מבין המשבצות שעדיין לא מוקמה בהן משבצת שחורה (בין שממוספרת או לא ממוספרת), אך לא נסמן בה מספר עדיין.
3. כל עוד יש משבצות לבנות:
   3.1. נגריל משבצת לבנה w.
   3.2. נציב ב-w נורה.
   3.3. נעדכן את כל המשבצות באותו טור והמשבצות באותה שורה עם w, שאין בינן לבין w משבצת שחורה, להיות צהובות.
4. לכל משבצת שחורה ממוספרת s:
   4.1. נספור בכמה מהמשבצות הסמוכות לs יש נורה, ונגדיר את מספר המשבצות הללו בתור המספר המסומן במשבצת

   שלב ביניים: במידה ומעוניינים לשמור פתרון קיים עבור הלוח(לצורך השוואה), שומרים את מיקומי הנורות בשלב הזה.
5. "מסירים" את כל הנורות (מחליפים כל משבצת צהובה ומשבצת עם נורה למשבצת לבנה ללא נורה)

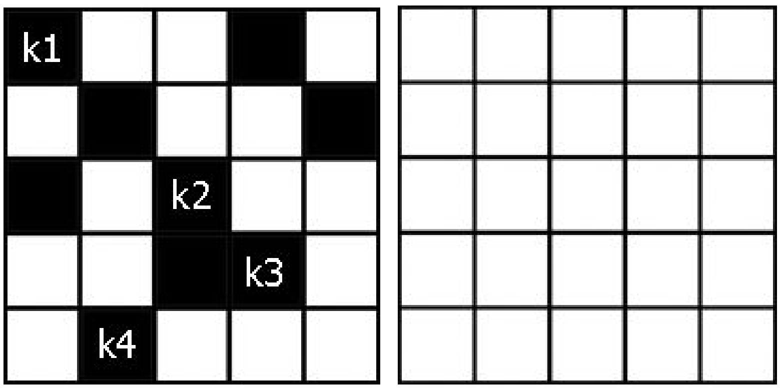
מכיוון שבשלב 6 נשארנו עם לוח שיש בו רק משבצות לבנות, משבצות שחורות ממוספרות ומשבצות שחורות לא ממוספרות, מדובר בהכרח בלוח משחק לפי ההגדרה. מספר המשבצות השחורות הלא ממוספרות נקבע בשלב 1, ומספר המשבצות השחורות הממוספרות נקבע בשלב 2, ושניהם לא משתנים בשום שלב

אחר, לכן בהכרח קיבלנו לוח עם התנאים שרצינו (n*n, עם b משבצות שחורות לא ממוספרות ו-r משבצות שחורות ממוספרות).
כעת נראה כי הלוח שבנינו פתיר:
טענה: הצבה של הנורות בלוח לפי המיקום ששמרנו בשלב 5 מהווה פתרון.
טענה א: לאחר ההצבה, כל המשבצות הלבנות מוארות.
תהי משבצת לבנה v כלשהי. בסוף שלב 3 היא בהכרח או מכילה נורה או צהובה(אחרת, היא היתה לבנה, ושלב 3 לא היה מסתיים כי עדיין קיימת משבצת לבנה). בשלב 4 ו5 לא שינינו את המשבצות הלבנות/צהובות/נורות, לכן היא נשארה צהובה/נורה גם בשלב 5. אם היא נורה אז היא מוארת על ידי עצמה. אם היא צהובה, אז קיימת משבצת אחרת שבה יש נורה, w, שכאשר הצבנו ב-w נורה (בשלב 3.2), צבענו את v בצהוב בשלב 3.3. אז v נמצאת באותו טור או שורה עם w, ואין בינהן משבצת שחורה, לכן w מאירה את v.

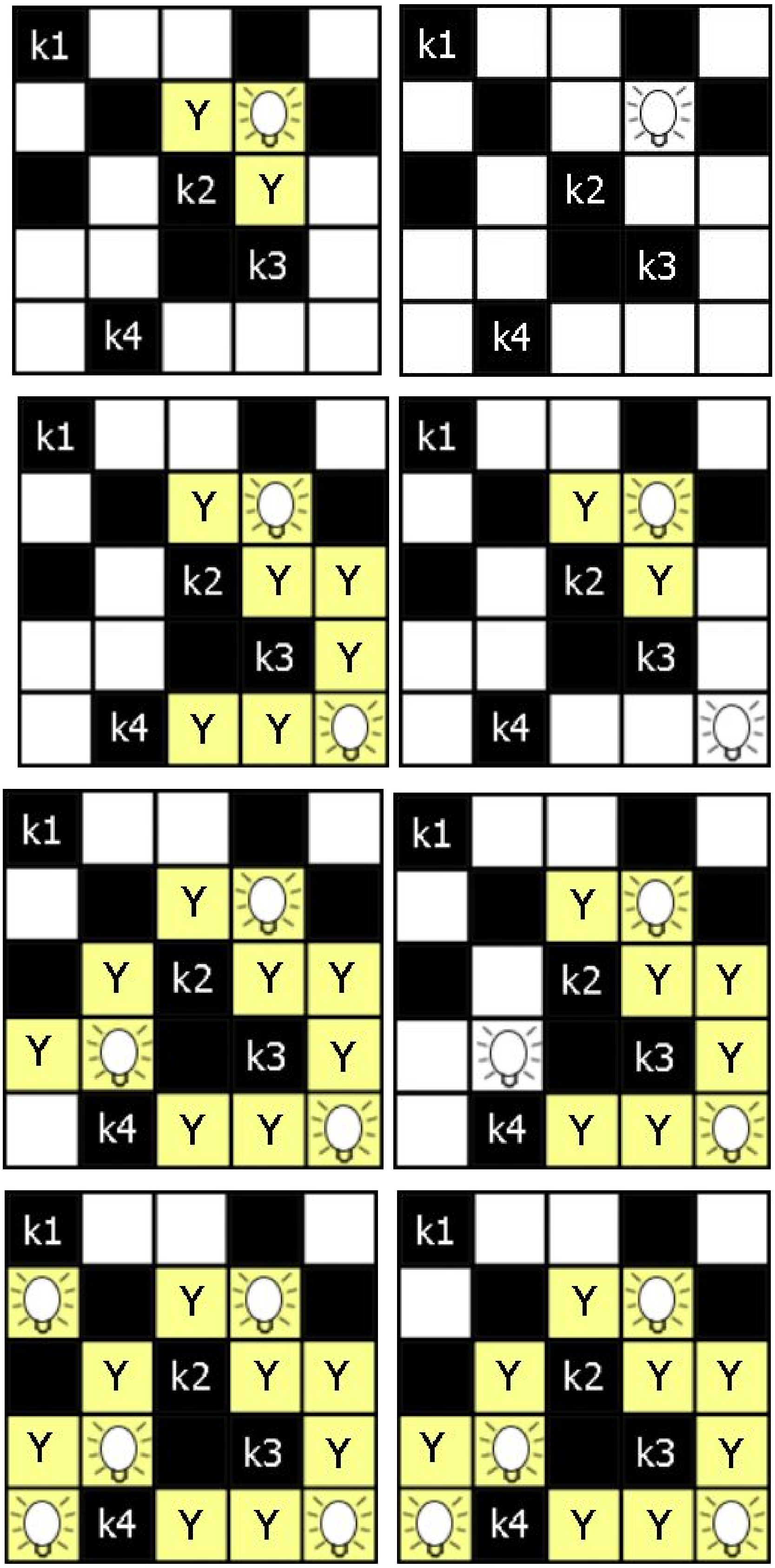
טענה ב: לאחר ההצבה, לכל משבצת ממוספרת, מספר השכנים המכילים נורות זהה למספר שרשום על המשבצת - נובע ישירות מכך שהגדרנו את המספר על המשבצת להיות מספר הנורות מבין השכנים שלה, בשלב 4.

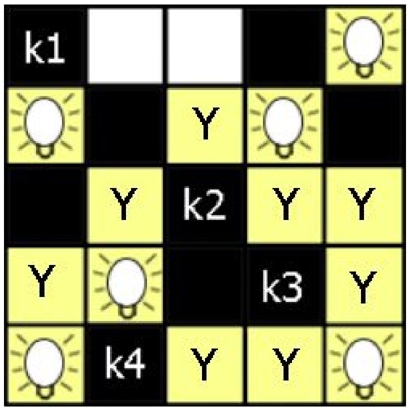
טענה ג: אין שתי מנורות שמאירות אחת את השנייה.
נניח בשלילה שקיימות משבצות u,v, ששתיהן מכילות נורה, שתיהן באותה שורה/טור, ואין משבצת שחורה המפרידה בינהן. שתיהן היו צריכות לקבל את ההצבה של הנורה בשלב 3.2(כיוון שרק בשלב זה מציבים נורות), ועל כן היו חייבות להבחר בשלב 3.1. אחת מהן היתה צריכה להבחר ראשונה, בלי הגבלת הכלליות u. לאחר ש-u נבחרה, בשלב 3.3 v נצבעה בצהוב (מההנחה - הן באותה שורה/טור ואין בינהן משבצת שחורה). ולאחר מכן לא ייתכן שv נבחרה בשלב 3.1, כיוון שבשלב 3.1 בוחרים רק מבין המשבצות הלבנות, לכן לא ייתכן שהוצבה נורה ב-v בשלב 3.2 - סתירה.

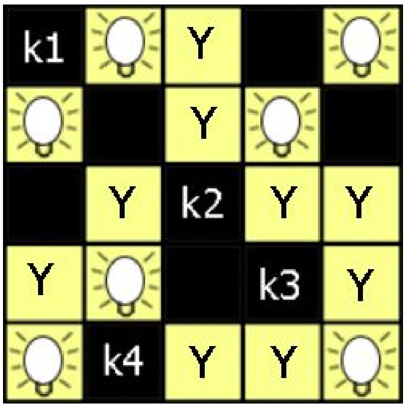
המחשת האלגוריתם:
שלבים 1-2 - מגרילים את המיקום של המשבצות השחורות:

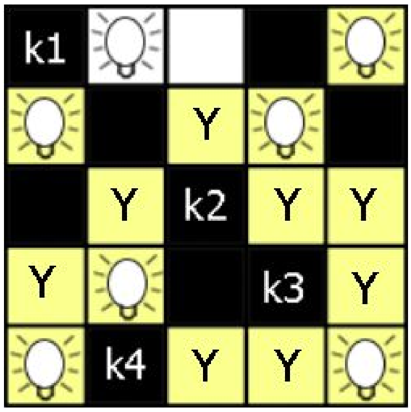


ריצת שלב 3 - מגרילים משבצת לבנה ומציבים בה נורה (כל שורה מהווה איטרציה, הסימון בצבע צהוב ובאות Y ממחישים את העדכון של המשבצות עליהן כבר יש נורה שמאירה, ועל כן לא יכולה להיות בהן נורה חוקית):

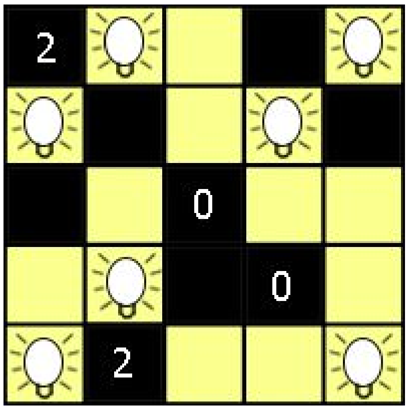




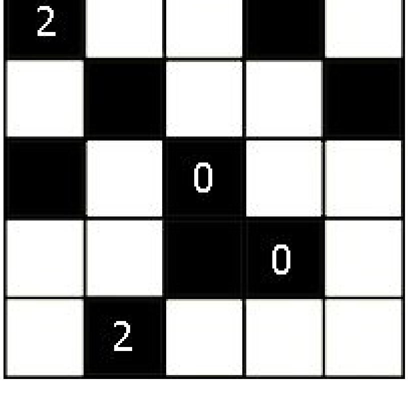




שלב 4 - מציבים לכל משבצת ממוספרת את מספר הנורות בסמוך אליה בפועל:

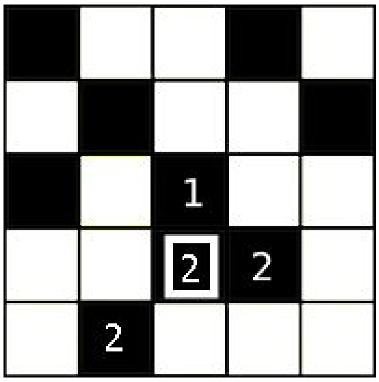


שלב 5 - מסירים את כל הנורות:

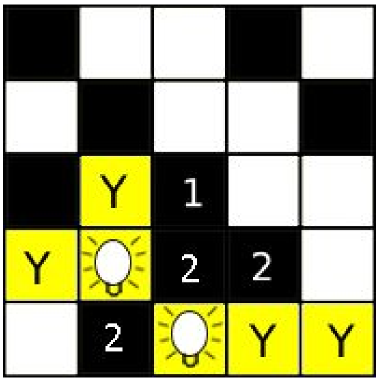


הפותר הנאיבי

הפותר הנאיבי מדמה במידה מסוימת את הדרך בה פותר אנושי ייגש לפתור את הפאזל. ראשית נשים לב לריבועים השחורים הממוספרים. ישנם מקרים בהם מספר התאים הפנויים (כלומר הלבנים) מסביב למשבצת כזו, זהה למספר הנורות שנותר לנו לשבץ מסביב לאותה משבצת . במצב זה , בטוח שכל אחד מהתאים הללו מכיל נורה.

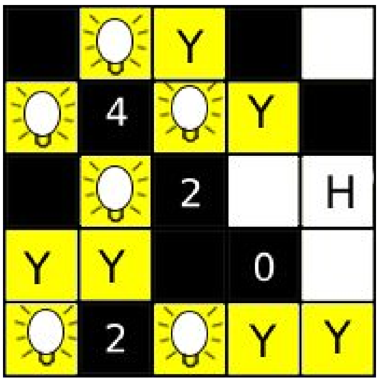


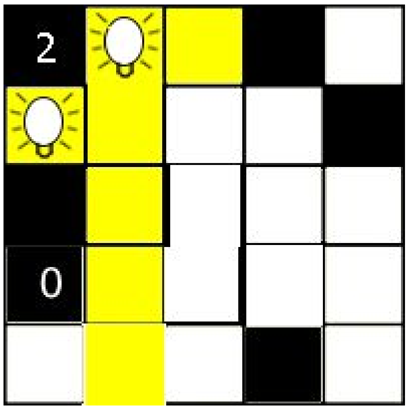
לדוגמא,בפאזל מימין, נבחין כי מסביב לספרה 2 באינדקס (4,3) יש בדיוק 2 משבצות לבנות , כמו כן לאחר הצבת הנורות מסביב למשבצת זו- מסביב למשבצת עם המספר 2 שלידה יש בדיוק 2 משבצות לבנות ולכן רק 2 מקומות בהן ניתן למקם נורות- בדיוק כמספר המופיע על המשבצת



מצב נוסף בו נטפל, הוא מצב בו בעבור משבצת מסויימת (A), קיימת משבצת אחת בלבד באחד הבלוקים שלה (האנכי או האופקי) שיכולה להאיר אותה. משבצת זו יכולה להיות A עצמה או משבצת אחרת - B השונה מ A . במצב זה נשבץ נורה ב-B על מנת להאיר את A, כיוון שכל פתרון חוקי חייב להכיל נורה כלשהי שתאיר על A.

לדוגמא, המשבצת המסומנת באות H. התא שאיתה בשורה לא יכול לקבל נורה,גם לא זה שמתחתיה והשלישי בטור צהוב ולכן גם לא יכול לקלוט מנורה. לכן המשבצת היחידה שיכולה להאיר את H היא H עצמה- ולכן נמקם בה נורה.





דומא נוספת, נבחין במשבצת מתחת לספרה 0. לא ניתן למקם בה נורה ויש רק משבצת אחת שיכולה להאיר עליה. מצב זה שונה מהמצב הקודם שכן המשבצת שמאירה עליה יכולה להיות מוארת גם על ידי משבצת אחרת

האלגוריתם הנאיבי עובר על המשבצות הממוספרות במטריצה,ובודק את קיום המצב הראשון בעבור כל אחת מהן. במידה והצליח למקם נורות הוא חוזר על שלב זה, אחרת עובר לשלב השני בו הוא סורק את הלוח, וכאשר מגיע למשבצת לבנה, הוא בודק אילוץ לשים נורה על פי המצבים השני וממקם נורה במשבצת המתאימה במידה ומצא שאפשר. בסיום האטרציה, אם הוא מיקם נורה חדשה במהלך הריצה, הוא עובר על הלוח שוב (החל מהשלב הראשון) - אחרת מסיים

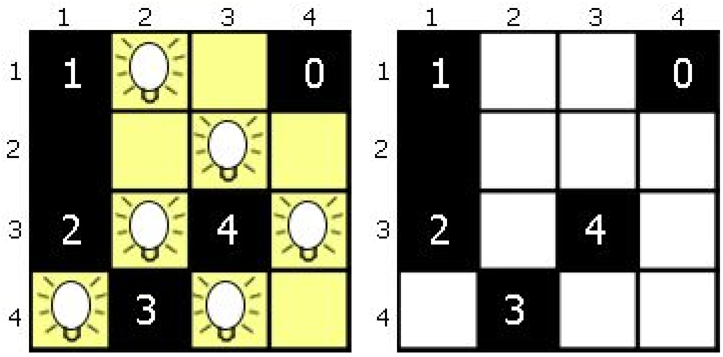
הפותר הנאיבי הוא פולינומיאלי על גודל הקלט [O(n^3)] - אך לא תמיד מוצא פתרון מלא

ננסה למצוא פותר אחר- פותר SAT

רדוקציה ל-SAT

הרדוקציה ל-SAT מבוצעת באופן הבא:

לכל משבצת לבנה (y,x), נגדיר משתנה בוליאני $b_{y,x}$, שהמשמעות שלו תהיה האם במשבצת המתאימה יש נורה או לא.
להמחשת הדוגמאות, נתבונן בלוח הבא(מימין - מצב התחלתי, משמאל - פתור):



נגדיר פסוק CNF המורכב מפסוקיות בוליאניות מ-3 סוגים:

1. לכל משבצת לבנה, הגדרנו פסוקית רגילה(כלומר or בין כל המשתנים) המורכבת מכל המשתנים בבלוק האופקי של המשבצת, ובנוסף כל המשתנים בבלוק האנכי של המשבצת.
לדוגמא: עבור המשבצת 2,3: $atLeast(1,\ \{b_{2,2}, b_{2,3}b_{2,4}, b_{1,3}\})$

2. לכל משבצת שחורה ממוספרת, בעלת המספר k, נגדיר פסוקיות עבור המשבצות הלבנות הסמוכות אליה של "בדיוק k".
לדוגמא: עבור המשבצת 3,3: $exactly(4,\ \{b_{2,3}, b_{3,2}, b_{3,4}, b_{4,3}\})$

3. לכל בלוק(אופקי או אנכי) בלוח, נגדיר פסוקית "לכל היותר 1" בין כל המשבצות בבלוק.
לדוגמא:
עבור הבלוק האופקי בשורה 2(כלומר, משבצות (2,2), (2,3),(2,4)):
$atMost\{1,\ \{b_{2,2},\ b_{2.,3}, b_{2,4}\})$
עבור הבלוק האנכי בעמודה 4: $atMost(1,\ \{b_{2,4}, b_{3,4}, b_{4,4}\})$

בפועל, בשביל פסוקיות מסוג "בדיוק k" השתמשנו בפסוקיות מסוג "לפחות k" ביחד עם פסוקיות "לכל היותר k", שמובנות ב-SAT4J, הספרייה בה השתמשנו[2].
מסקירה של קוד המקור של הספרייה[3], עולה כי האופן בו מיוצרות בפועל פסוקיות כאלה עבור j משתנים הוא:
"לפחות k" - אם j-k=1 אז הופכים את זה לפסוקית רגילה. אחרת, הפסוקית נשמרת כאובייקט מסוג "פסוקית לפחות"(AtLeast), המכיל שדה של "מספר האילוצים הלא מסופק המירבי"(maxUnsatisfied), שמקבל את הערך j-k, ומונה של אילוצים לא מסופקים(counter) שמאותחל בערך 0. בכל פעם שמבצעים השמה לאחד המשתנים הבולאניים, במקום לבדוק האם עדיין קיים משתנה שיכול להיות מסופק(כמו בפסוקית בינארית רגילה), בודקים האם ההשמה הזאת סותרת את אחד המשתנים בפסוקית - אם כן, מגדילים את המונה ב-1; ולבסוף כשבודקים האם פסוקית ספיקה, בודקים האם counter≤maxUnsatisfied.

"לכל היותר k" - הופכים את כל הסימנים של המשתנים, ועם האוסף החדש, יוצרים פסוקית חדשה מסוג "לפחות j-k".

נגדיר: הצבה של נורות בלוח והשמה של פסוק ה-CNF מתאימות אם ורק אם: לכל משבצת לבנה, יש בה נורה בהצבה אם ורק אם המשתנה המתאים למשבצת מקבל ערך אמת בהשמה.

טענה: השמה מספקת לפסוק ה-CNF אם ורק אם ההצבה המתאימה של נורות בלוח היא פתרון חוקי ללוח.

מסקנות מהטענה:
1. קיים פתרון חוקי ללוח אם ורק אם פסוק ה-CNF ספיק.
2. בהינתן השמה מספקת לפסוק ה-CNF, ניתן להמירה לפתרון חוקי ללוח בזמן לינארי על גודל הלוח($O(n^2)$).

לצורך נוחות, עבור משבצת ממוספרת b, נסמן ב k_b את המספר שמסומן עליה.
הוכחת הטענה:
יהי S פתרון חוקי ללוח. נניח בשלילה שההשמה המתאימה אינה מספקת את פסוק ה-CNF, כלומר קיימת פסוקית שאינה מסתפקת בהשמה זו. נחלק ל3 מקרים:
<u>**מקרה ראשון -**</u> הפסוקית היא מסוג 1. אז עבור המשבצת w שהיא המשבצת המתאימה לפסוקית, אף משבצת לא מכילה נורה. הבלוק האופקי והבלוק האנכי של

המשבצת מכילים את כל המשבצות שנורה בהן תאיר את w. לכן w נשארת לבנה לאחר הצבת כל הנורות בלוח. אז לא כל המשבצות הלבנות מוארות בסיום הצבת הנורות, סתירה לכך שS הוא פתרון ללוח(כלל 1).

מקרה שני - הפסוקית היא מסוג 2. אז עבור המשבצת b שהיא המשבצת המתאימה לפסוקית שלא מסופקת, אחת מבין הפסוקיות המרכיבות את פסוק ה-"לפחות k_b" או פסוק ה-"לכל היותר k_b" אין מסתפקת בהשמה המתאימה לS. כלומר, מספר המשתנים שערכיהם הוא true בהשמה מבין השכנים של b שונה מ- k_b. אז מספר הנורות בS מבין השכנות של b שונה מ- k_b, אז S מפר את כלל 2 עבור פתרון חוקי - בסתירה לכך שS הוא פתרון ללוח.

מקרה שלישי - הפסוקית היא מסוג 3. כלומר, קיים בלוק בו יש לפחות 2 משבצות w,v, אשר המשתנים המתאימים עבורן מקבלים ערך אמת בהשמה המתאימה לS. אז במשבצות w וv יש נורה בהשמה S. המשבצות באותו בלוק, לכן הן באותה שורה או טור, ואין משבצת שחורה שמפרידה בינהן - לכן הנורות במשבצות אלה מאירות אחת את השנייה, ומפרות את כלל 3, בסתירה לכך שS פתרון חוקי.

שלושת המקרים מכסים את כל הפסוקיות בפסוק ה-CNF, ובכל המקרים הגענו לסתירה, לכן ההשמה המתאימה לS כן מספקת את הפסוק.

תהי H השמה מספקת לפסוק ה-CNF.
תהי משבצת לבנה w, בפסוק ה-CNF קיימת פסוקית מסוג 1, עבור איחוד הבלוקים של w, כלומר שהמשתנה של לפחות אחת המשבצות לפחות באחד מהבלוקים של w בהשמה H(בגלל ש-H השמה מספקת). לכן בהצבה המתאימה ל-H, יש נורה במשבצת הזאת, ובגלל שהיא חולקת בלוק עם w, מתקיים ש-w מוארת בהצבה המתאימה ל-H. בחרנו במשבצת כלשהי, לכן זה נכון לכל משבצת לבנה, לכן תנאי 1 לפתרון חוקי של הלוח מתקיים עבור ההצבה המתאימה ל-H.

תהי משבצת ממוספרת b. בפסוק ה-CNF קיים אוסף של פסוקיות הקובע כי בדיוק b_k מהמשתנים המתאימים לשכנים שלהם מקבלים ערך אמת, ומכיוון ש-H השמה מספקת תנאי זה מתקיים ב-H. מהגדרת הצבה מתאימה, המשבצות שיש בהן נורה הן אלה שמקבלות ערך אמת בהשמה, לכן בדיוק אותן b_k משבצות שכנות של b מכילות נורות. כלומר, לכל משבצת ממוספרת, מספר השכנים שלה שמכילים נורה זהה למספר המסומן עליה, לכן תנאי 2 לפתרון חוקי של הלוח מתקיים עבור ההצבה המתאימה ל-H.

יהיו משבצות לבנות w,v המכילות נורה בהצבה המתאימה ל-H. נניח בשלילה שהן מאירות אחת על השנייה. הן חייבות להיות באותה שורה או טור, ובלי שמשבצת שחורה כלשהי תפריד בינהן, אחרת לא יוכלו להאיר אחת על השנייה, לכן הן באותו בלוק. לכל בלוק יש אוסף פסוקיות המגדירות "לכל היותר 1", ומכיוון ש-H השמה מספקת הן כולן מסופקות. לכן לא קיימות בבלוק 2 משבצות אשר המשתנה המתאים להן מקבל ערך אמת. אבל w ו-v מכילות נורות בהשמה המתאימה ל-H, לכן מהגדרת השמה מתאימה, המשתנה המתאים לכל אחת מהן מקבל ערך אמת בהשמה H, סתירה. לכן לכל 2 נורות מתקיים שהן לא מאירות אחת על השנייה, אז תנאי 3 לפתרון חוקי של הלוח מתקיים עבור ההצבה המתאימה ל-H.

שלושת התנאים מתקיימים, לכן ההצבה המתאימה ל-H מהווה פתרון חוקי של הלוח.

זמני הריצה של הרדוקציה

עבור לוח בגודל n*n:

יצירת כל פסוקית מסוג 1(לכל משבצת לבנה, פסוקית של כל המשבצות באותו בלוק אופקי או אנכי) לוקח $O(n)$. יש לכל היותר $O(n^2)$ משבצות לבנות, לכן סה"כ הזמן לבניית כל הפסוקיות מסוג 1 הוא $O(n^3)$.

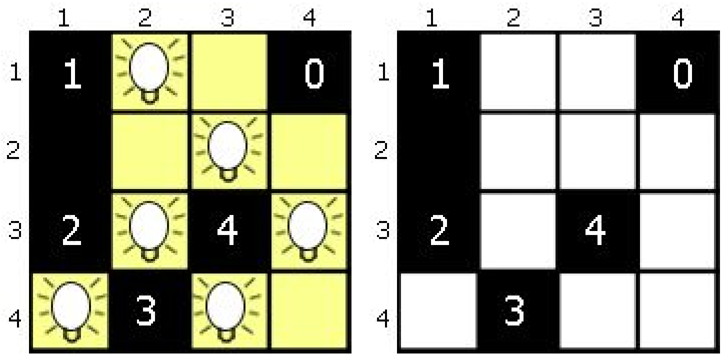
יצירת כל פסוקית מסוג 2(לכל משבצת ממוספרת b, פסוקית "בדיוק k_b מבין השכנים שלה), לוקח $O(1)$ (יש לכל היותר 4 שכנים להוסיף והמספר נתון). יש לכל היותר $O(n^2)$ פסוקיות כאלה, לכן סה"כ הזמן לבניית כל הפסוקיות מסוג 2 הן $O(n^2)$.

יצירת הפסוקיות מסוג 3(לכל בלוק אופקי ואנכי יכולה להיות לכל היותר משבצת אחת שמשתנה שלה מקבל ערך "אמת") היא $O(n)$ לכל פסוקית(יכולות להיות לכל היותר n משבצות בבלוק). יש לכל היותר $O(n^2)$ בלוקים אופקיים ו $O(n^2)$ בלוקים אופקיים(כל בלוק חייב להכיל לפחות משבצת לבנה אחת), אז סה"כ יש $O(n^2)$ בלוקים. לכן סה"כ הזמן לבניית כל הפסוקיות מסוג 3 הוא $O(n^3)$.

סה"כ הזמן לרדוקציה $O(n^3) + O(n^2) + O(n^3) = O(n^3)$

דוגמא לרדוקציה

נתבונן שוב בלוח לדוגמא ממקודם(מימין - לא פתור, משמאל - פתור):



1. פסוקיות מסוג 1:
   1.1. עבור המשבצת 1,2: $atLeast(1,\ \{b_{1,2}, b_{1,3}, b_{2,2}, b_{3,2}\})$
   1.2. עבור המשבצת 1,3: $atLeast(1,\ \{b_{1,2}, b_{1,3}, b_{2,3}\})$
   1.3. עבור המשבצת 2,2: $atLeast(1,\ \{b_{2,2}, b_{2,3} b_{2,4}, b_{1,2}, b_{3,2}\})$
   1.4. עבור המשבצת 2,3: $atLeast(1,\ \{b_{2,2}, b_{2,3} b_{2,4}, b_{1,3}\})$
   1.5. עבור המשבצת 2,4: $atLeast(1,\ \{b_{2,2}, b_{2,3}, b_{2,4}, b_{3,4}, b_{4,4}\})$
   1.6. עבור המשבצת 3,2: $atLeast(1,\ \{b_{1,2}, b_{1,3}, b_{2,2}\})$
   1.7. עבור המשבצת 3,4: $atLeast(1,\ \{b_{2,4}, b_{3,4}, b_{4,4}\})$
   1.8. עבור המשבצת 4,1: $atLeast(1,\ \{b_{4,1}\})$
   1.9. עבור המשבצת 4,3: $atLeast(1,\ \{b_{4,3}, b_{4,4}\})$
   1.10. עבור המשבצת 4,4: $atLeast(1,\ \{b_{4,3}, b_{4,4}, b_{2,4}, b_{3,4}, \})$
2. פסוקיות מסוג 2:
   2.1. עבור המשבצת 1,1:
      2.1.1. $atLeast(1,\ \{b_{1,2}\})$
      2.1.2. $atMost(1,\ \{b_{1,2}\})$
   2.2. עבור המשבצת 1,4:
      2.2.1. $atLeast(0,\ \{b_{1,3}, b_{2,4}\})$
      2.2.2. $atMost(0,\ \{b_{1,3}, b_{2,4}\})$
   2.3. עבור המשבצת 3,1:
      2.3.1. $atLeast(2,\ \{b_{3,2}, b_{4,1}\})$
      2.3.2. $atMost(2,\ \{b_{3,2}, b_{4,1}\})$
   2.4. עבור המשבצת 3,3:
      2.4.1. $atLeast(4,\ \{b_{2,3}, b_{3,2}, b_{3,4}, b_{4,3}\})$
      2.4.2. $atMost(4,\ \{b_{2,3}, b_{3,2}, b_{3,4}, b_{4,3}\})$

2.5. עבור המשבצת 4,2:

2.5.1. $atLeast(3,\ \{b_{4,1}, b_{3,2}, b_{4,3}\})$

2.5.2. $atMost(3,\ \{b_{4,1}, b_{3,2}, b_{4,3}\})$

3. פסוקיות מסוג 3:

3.1. עבור הבלוק האופקי בשורה 1:

3.1.1. $atMost(1,\ \{b_{1,2},\ b_{1,3}\})$

3.2. עבור הבלוק האופקי בשורה 2:

3.2.1. $atMost\{1,\ \{b_{2,2},\ b_{2.,3}b_{2,4}\})$

3.3. עבור הבלוקים האופקיים בשורה 3:

3.3.1. $atMost(1,\ \{b_{3,2}\})$

3.3.2. $atMost(1,\ \{b_{3,4}\})$

3.4. עבור הבלוקים האופקיים בשורה 4:

3.4.1. $atMost(1,\ \{b_{4,1}\})$

3.4.2. $atMost(1, \{b_{4,3}, b_{4,4}\})$

3.5. עבור הבלוק האנכי בעמודה 1:

3.5.1. $atMost(1,\ \{b_{4,1}\})$

3.6. עבור הבלוק האנכי בעמודה 2:

3.6.1. $atMost(1,\ \{b_{1,2}, b_{2,2}, b_{3,2}\})$

3.7. עבור הבלוקים בעמודה 3:

3.7.1. $atMost(1, \{b_{1,3}, b_{2,3}\})$

3.7.2. $atMost(1,\ \{b_{4,3}\})$

3.8. עבור הבלוק האנכי בעמודה 4:

3.8.1. $atMost(1,\ \{b_{2,4}, b_{3,4}, b_{4,4}\})$

שימוש ברדוקציה באמצעות SAT Solver:

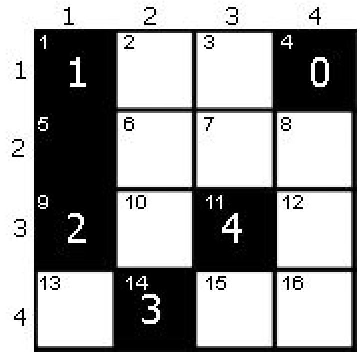
השתמשנו בספרייה SAT4J, שהיא פותרן בעיות SAT לשפת Java בה השתמשנו[2].

בספרייה כל משתנה מסומן על ידי ערך של int: אם זה בצורתו החיובית אז זה המספר החיובי(למשל, 7), ואם זה בצורתו השלילי אז זה אותו מספר עם סימן הפוך (למשל, 7-). על כן, לכל משבצת (x, y), הגדרנו את ה-id שלה: $id(x,\ y)\ =\ n * y + x$

.

החישוב בכיוון ההפוך הוא $x(id) = id\ modulo\ n,\ y(id) = floor(id/n)$

כך, למשל, למשבצת $(2,\ 3)$ בלוח 7*7, ה-id מוגדר להיות $id(2,3) = 3 * 7 + 2 = 23$, ואז אם נרצה להציב את המשתנה המייצג אותה בצורתה החיובית נשתמש במספר 23, ובצורתה השלילית נשתמש במספר 23-.
והחישוב בכיוון ההפוך הוא $x(23) = 23\ modulo\ 7\ =\ 2,\ y(23) = floor(23/7) = 3$.
לדוגמא: עבור הלוח מלמעלה, האינדקסים יהיו(מסומנים בקטן בפינה השמאלית העליונה של כל משבצת):



וכך, על מנת לייצג את אוסף הפסוקיות $atLeast(4,\ \{b_{2,3}, b_{3,2}, b_{3,4}, b_{4,3}\})$, נוסיף את אוסף הפסוקיות $addAtLeast(4,\ \{7, 10, 12, 15\})$

הפותר ההיברידי

הרדוקציה ל-SAT ושימוש ב-SAT Solver בהכרך ימצאו פתרון, אך מכיוון שבעייה זו היא NP-שלמה, היינו רוצים לנסות לצמצם את התלות ב-SAT. לצורך כך כתבנו את הפותר ההיברידי.

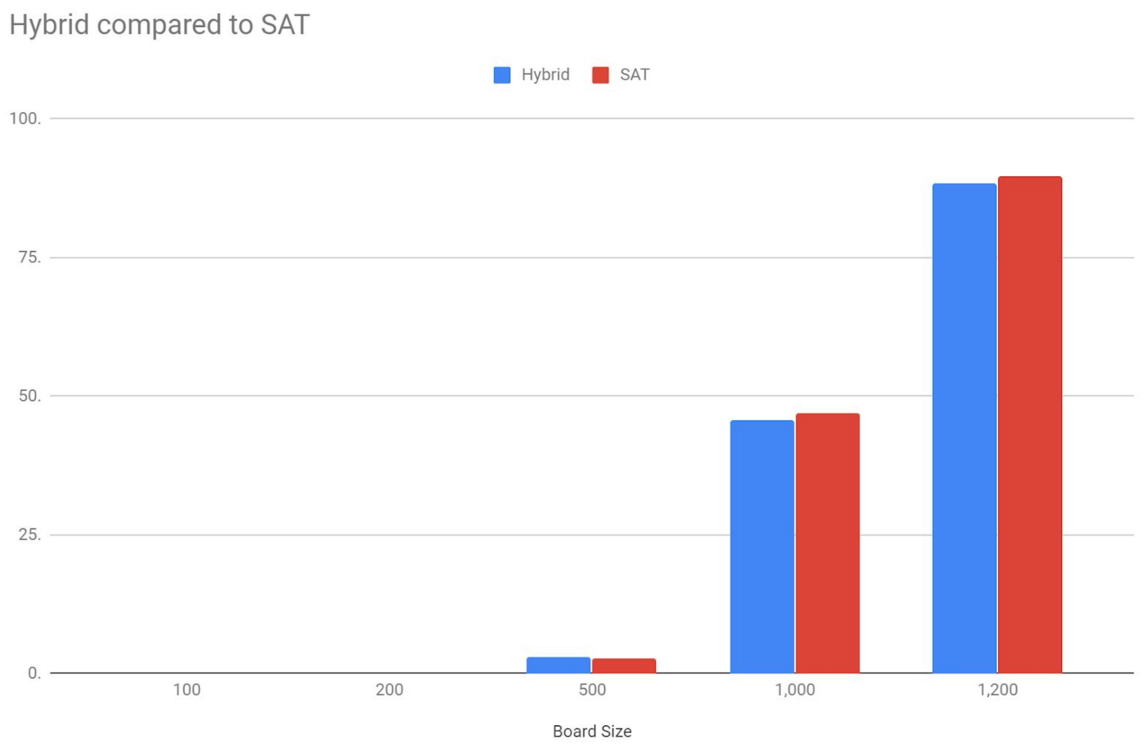
הפותר ההיברידי מורכב משני חלקים:
חלק ראשון: הפותר הנאיבי, כפי שתואר בפרק 4
חלק שני: רדוקציה ל-SAT, כפי שתוארה בפרק 5, עם השינויים הבאים:
עבור פסוקיות מסוג 1(כלומר, לכל משבצת לבנה b):
אם המשבצת מכילה נורה(לאחר הריצה של הנאיבי), נוסיף פסוקית המכילה רק את המשתנה המתאים למשבצת, בצורתו החיובית.
אם המשבצת היא צהובה(כלומר, לאחר הריצה של הנאיבי יש נורה במשבצת אחרת שמאירה עליה), נוסיף פסוקית המכילה רק את המשתנה המתאים למשבצת, בצורתו השלילית.
אחרת, נוסיף פסוקית מסוג 1 כמו מקודם, אך לא נוסיף משבצות צהובות. כלומר: נגדיר פסוקית רגילה התורכב:
לכל משבצת c בבלוק האופקי של b: אם c היא לבנה(ולא צהובה) נוסיף אותה לפסוקית.
לכל משבצת c בבלוק האנכי של b: אם c היא לבנה(ולא צהובה) נוסיף אותה לפסוקית.

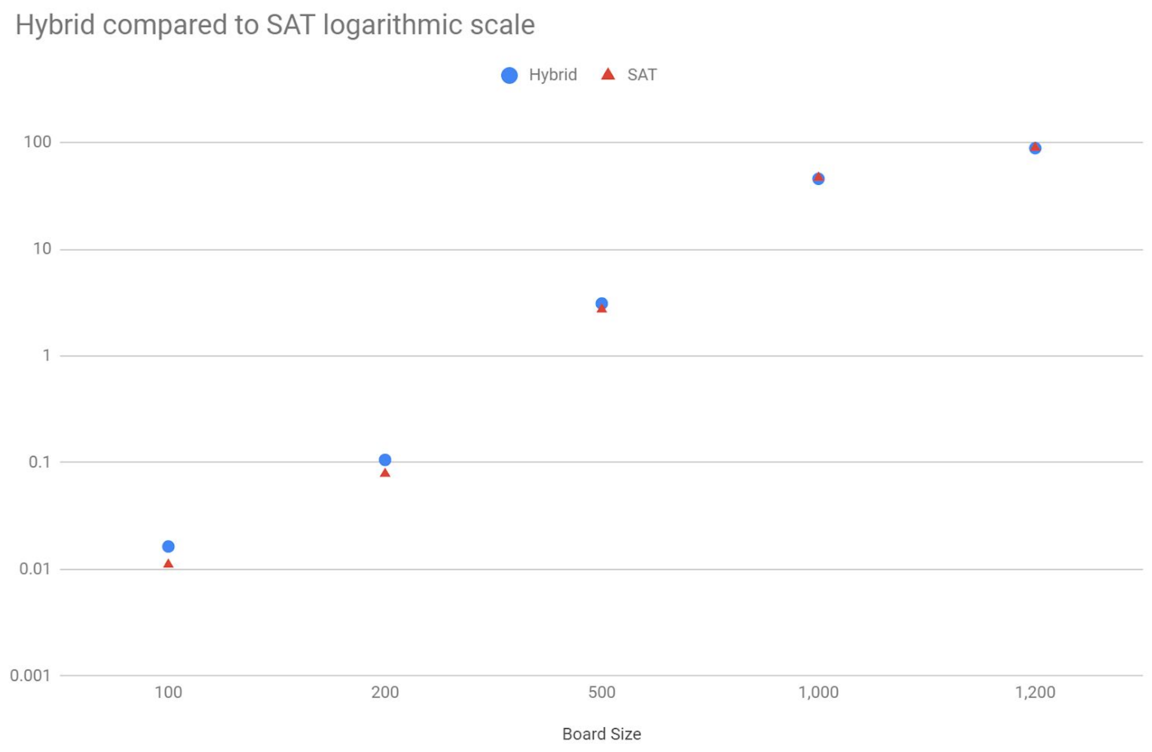
הבנייה נובעת מכך שבפותר הנאיבי מצאנו מנורות שחייבות להיות בפתרון.
כל משבצת שיש בה נורה שקולה לערך חיובי במשתנה המתאים ברדוקציה ל-SAT.
כל משבצת צהובה שקולה לערך שלילי במשתנה המתאים(כיוון שאסור להציב נורה במשבצת שכבר יש נורה שמאירה עליה), ומאידך לא צריך להכריח שתהיה נורה שתאיר עליה(כי אנחנו יודעים שכבר יש כזאת).
עבור משבצות לבנות, במקום להוסיף את כל המשבצות שבבלוקים שלה בתור אלה שיכולות להאיר עליה, הוספנו רק את אלה שלא הוכחנו(בזמן הריצה של הנאיבי) שבהכרח לא יכולה להיות בהן נורה.

הרצות

הרצנו את 3 הפותרים שהצגנו על לוחות בגדלים: 100*100, 200*200, 500*500, 1000*1000, ו-1200*1200, עם מספרים משתנים של של משבצות שחורות ומשבצות ממוספרות.
תוצאות ההרצה, עם זמני הריצה הממוצעים(בשניות) על כל גודל לוח של הפותרים השונים:

Board Size	Hybrid	SAT	number of boards	% of lamps naive found	Naive	Hybrid SAT part
100	0.016372	0.01111	153	0.821265	0.005766	0.010606
200	0.106256	0.078805	181	0.810353	0.0355	0.070755
500	3.098508	2.734934	253	0.808266	0.425207	2.673301
1,000	45.686456	46.872064	153	0.8266	1.578449	44.108007
1,200	88.453658	89.622266	78	0.809605	2.356465	86.097193





ניתן לראות כי בכל הגדלים זמן הריצה של ה-SAT בפותר ההיברידי הוא קצר מזמן הריצה של ה-SAT לבד, וככל שהלוחות גדלים כך זמן הריצה של האלגוריתם הנאיבי נהיה זניח בהשווה לזמן הריצה של ה-SAT, ככה שכבר בn=1000 אנו רואים שיפור בזמן הריצה הכולל של האלגוריתם ההברידי ביחס לSAT.
בנוסף, בכל ההרצות, האלגוריתם הנאיבי מוצא כ-81% מהנורות בממוצע.

את תוצאות הריצה ניתן לראות במלואן בנספח המצורף.

ביבליוגרפיה

1. Light Up is NP-complete(Brandon McPhail, February 28, 2005) http://mountainvistasoft.com/docs/lightup-is-np-complete.pdf
2. SAT4Jdoc: http://www.sat4j.org/maven23/org.sat4j.core/apidocs/index.html
3. SAT4Jdoc source code: https://svnlegacy.ow2.org/sat4j/maven/branches/REDUCED_CORE_PACKAGES/