

Guard Scheduling Optimization

מגישות:

שם: גיל אברהם ת.ז: 318662939

שם: חן לארי ת.ז: 209192798

רקע:

בפרויקט שלנו שילבנו שני נושאים שניתנו: למדנו כיצד לשלב באלגוריתם תכנות בשלמים יותר מפונקציית מטרה אחת (במקרה שלנו 2) ובנוסף השונו את האלגוריתם לאלגוריתם hill climbing with random restart. עוד דבר שראינו הוא כיצד להשתמש בהיוריסטיקת "מינימום סתירות" עבור hill climbing על-מנת שיספק כמה שיותר אילוצים של הבעיה.

תיאור הבעיה:

ישנה חברת אבטחה שמספקת שומרים לבניין מסוים שצריכים לשמור עליו 24/7. השומרים עובדים במשמרות כך שישנם 21 משמרות ושומר לא יכול לקחת משמרת כפולה על-מנת שלא יתעייף. הבניין מאוד חשוב ולכן נדרשים לפחות 2 שומרים בכל משמרת אך אין צורך ביותר מ-6. חברת האבטחה תומכת בכל האוכלוסיות ועל כן מאפשרת לעובדיה לעבוד במשרה חלקית או מלאה, ומנסה לקיים שכל עובד יוכל לבחור יום שבו לא ירצה לעבוד.

נכתוב זאת בצורה נוחה לפישוט:

- בכל משמרת חייבים להיות לפחות 2 שומרים.
- בכל משמרת לא יהיו יותר מ-6 שומרים.
- שומר לא יכול לעבוד ביותר ממשמרת אחת ביום.
- שומר לא יכול לעבוד בשתי משמרות רצופות.
- לכל שומר יש אחוז משרה שאותו הוא צריך למלא (אחוז משרה מוגדר לפי מספר המשמרות לכל שומר)
- כל שומר יכול לבקש ימים בהם לא רוצה לעבוד.

האילוץ האחרון הוא אילוץ "רך" ואילו האילוצים האחרים הם אילוצים "קשים".

המטרה של החברה היא למקסם את אחוז המשרה עבור כל שומר תחת האילוצים הקיימים.

גישת הפתרון:

נראה כמה מודלים לפתרון הבעיה:

1. תכנות בשלמים integer programming.

2. Hill climbing with random start

3. Genetic algorithm

תיאור הבעיה כבעיית אופטימיזציה:

משתני ההחלטה:

ישנם 21 משתני החלטה בינאריים. $x_{i,j} = 1$ אם השומר ה- i עובד במשמרת ה- j .

ו- $x_{i,j} = 0$ אחרת.

לכל מודל נגדיר פונקציית מטרה ואילוצים שונים. נתאר אותם עבור כל אחד מהמודלים.

תכנות בשלמים:

פונקציית המטרה:

נמקסם את אחוז המשרה: $\max \sum_{i,j} x_{i,j}$

נכניס את האילוץ הרך לפונקציית מטרה נוספת: $\min \sum_{i,j} I[x_{i,j} = 0]$

אילוצים:

1. בכל משמרת לפחות 2 שומרים: $\forall j: \sum_i x_{i,j} \geq 2$

2. בכל משמרת לכל היותר 6 שומרים: $\forall j: \sum_i x_{i,j} \leq 6$

3. שומר לא עובד ביותר ממשמרת אחת ביום:

$$\forall i: x_{i,1} + x_{i,2} + x_{i,3} \leq 1$$

$$\forall i: x_{i,4} + x_{i,5} + x_{i,6} \leq 1$$

$$\forall i: x_{i,7} + x_{i,8} + x_{i,9} \leq 1$$

$$\forall i: x_{i,10} + x_{i,11} + x_{i,12} \leq 1$$

$$\forall i: x_{i,13} + x_{i,14} + x_{i,15} \leq 1$$

$$\forall i: x_{i,16} + x_{i,17} + x_{i,18} \leq 1$$

$$\forall i: x_{i,19} + x_{i,20} + x_{i,21} \leq 1$$

4. שומר לא יכול לעבוד ב-2 משמרות רצוף: $\forall i, j: x_{i,j} + x_{i,j+1} \leq 1$

5. לכל שומר נקיים שאחוז המשרה שלו היא c_i $\forall i: \sum_j x_{i,j} \leq c_i$ כאשר c_i הוא כמות

המשמרות שמגדירות את אחוז המשרה שלו.

[Hill climbing with random restart](#):

פונקציית המטרה:

נשים לב שאלגוריתם "טיפוס הרים" רגיל אינו מספק אילוצים, אך אצלנו ישנם אילוצים. לכן נשתמש בהיוריסטיקת "מינימום סתירות" כך שנכניס פונקציית מטרה סינטטית שתתאר לנו כמה אילוצים אינם מסופקים מהפתרון. אילוצים קשים יקבלו "עונש" חמור יותר על אי-סיפוקם, ואילו אילוצים רכים יקבלו עונש קל יותר.

האילוצים הקשים הם אותם אילוצים שתיארנו קודם, נסמן את הסכום שלהם ב- HC (hard constraint) ואת העונש עליהם נגדיר 10. האילוץ הרך יסומן ב- SC (soft constraint) ואת העונש עליהם נגדיר 1.

$$\min 10 * HC + 1 * SC$$

אילוצים:

אין

[תיאור האלגוריתמים בכלליות + פסאודו קוד:](#)

תכנות בשלמים:

כל המשתנים חייבים להיות שלמים בפתרון.

על מנת לפתור את הבעיה הזו נשתמש בכלי של גוגל: or-tools.

[Hill climbing with random start](#):

נתחיל בכל פעם מפתרון חוקי אחר שנשפר אותו עד שניתקע ולבסוף נחזיר את הפתרון הטוב ביותר שמצאנו. מציאת השיפור (שכן) של הפתרון הנוכחי תתבצע כך: מחליפים מקום שבו יש 1 עם מקום שבו יש 0.

1. נבחר פתרון רנדומלי ונשפר אותו עד שניתקע.
2. אם הפתרון שמצאנו טוב יותר מהפתרון הטוב ביותר שמצאנו עד כה, נשמור אותו כפתרון הטוב ביותר.
3. נחזור ל-1.

[נשווה בין האלגוריתמים:](#)

בכל פעם נבצע בדיקה של הבעיה כך שנציב נתונים חדשים ונראה את התוצאות. אנחנו עשינו מעל ל-100 בדיקות של נתונים והחלטנו לכתוב כאן שלושה בדיקות שבעינינו היו הכי מעניינות שמצאנו.

השוואה ראשונה:

בחברה ישנם 30 שומרים, ולכל שומר יש את הבקשות שלו. זה המקרה הקלאסי של הבעיה. נראה את הפתרון שהאלגוריתמים נותנים.

תכנות בשלמים מצא פתרון אופטימלי עם סיפוק של 100 משמרות מתוך 101 משמרות

שביקשו השומרים. סידור המשמרות נראה כך:

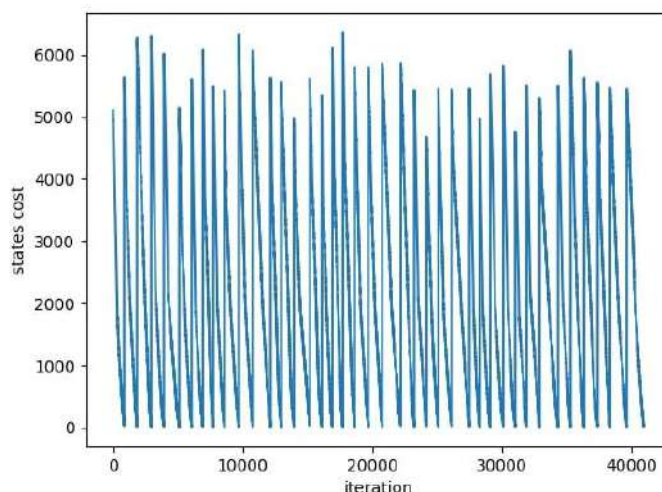
שבת	שישי	חמישי	רביעי	שלישי	שני	ראשון	משמרת
שומר 8 שומר 9 שומר 23 שומר 28	שומר 3 שומר 6 שומר 8 שומר 13 שומר 25 שומר 26	שומר 1 שומר 23 שומר 29	שומר 1 שומר 9 שומר 10 שומר 11	שומר 11 שומר 14 שומר 15 שומר 22 שומר 24 שומר 29	שומר 12 שומר 14	שומר 6 שומר 8 שומר 13 שומר 21 שומר 25 שומר 27	בוקר
שומר 1 שומר 2 שומר 12 שומר 15 שומר 19 שומר 30	שומר 2 שומר 4 שומר 11 שומר 19 שומר 27 שומר 30	שומר 3 שומר 13 שומר 14 שומר 16 שומר 20 שומר 28	שומר 7 שומר 14 שומר 16 שומר 26 שומר 27 שומר 29	שומר 6 שומר 18	שומר 6 שומר 20 שומר 22 שומר 27 שומר 29 שומר 30	שומר 5 שומר 9 שומר 12 שומר 24	צהריים
שומר 6 שומר 18 שומר 24 שומר 26 שומר 27	שומר 1 שומר 16 שומר 20 שומר 21 שומר 24 שומר 29	שומר 10 שומר 12 שומר 15 שומר 17 שומר 30	שומר 4 שומר 18 שומר 21 שומר 22 שומר 24 שומר 28	שומר 2 שומר 3 שומר 5 שומר 13 שומר 17 שומר 26	שומר 7 שומר 18	שומר 18 שומר 26 שומר 28	ערב

Hill climbing עם 40 random restart מצא פתרון מינימלי של 9, כלומר 9 אילוצים לא

סופקו. סידור המשמרות נראה כך:

שבת	שישי	חמישי	רביעי	שלישי	שני	ראשון	משמרת
שומר 3 שומר 6 שומר 14 שומר 24 שומר 25 שומר 27	שומר 3 שומר 6 שומר 13 שומר 24 שומר 30	שומר 6 שומר 7 שומר 9	שומר 14 שומר 20 שומר 24	שומר 22 שומר 24 שומר 27	שומר 1 שומר 23 שומר 29 שומר 30	שומר 18 שומר 25	בוקר
שומר 1 שומר 9 שומר 10 שומר 12 שומר 17 שומר 19	שומר 2 שומר 4 שומר 7 שומר 9 שומר 12 שומר 26	שומר 3 שומר 13 שומר 14 שומר 22 שומר 26 שומר 28	שומר 2 שומר 16 שומר 26 שומר 27 שומר 28	שומר 4 שומר 6 שומר 11 שומר 18 שומר 20	שומר 15 שומר 18 שומר 20	שומר 21 שומר 26	צהריים
שומר 8 שומר 13 שומר 16 שומר 30	שומר 1 שומר 16 שומר 19 שומר 21 שומר 28 שומר 29	שומר 5 שומר 8 שומר 10 שומר 17 שומר 18 שומר 21	שומר 11 שומר 12 שומר 13 שומר 22 שומר 29	שומר 2 שומר 15 שומר 26 שומר 29 שומר 30	שומר 8 שומר 11 שומר 12 שומר 14	שומר 5 שומר 27 שומר 28	ערב

כאשר התנודות של הערכים שנמצאו:



השוואה שנייה:

השוואה זו התבצעה על 20 שומרים כאשר שומר אחד לא רוצה לעבוד. כלומר, לקחנו שומר אחד ושמנו אצלו בקשות לא לעבוד בכל הימים. שומר זה הוא שומר מספר 1.

תכנות בשלמים מצא פתרון אופטימלי עם סיפוק של 68 משמרות מתוך 71 משמרות

שביקשו השומרים. סידור המשמרות נראה כך:

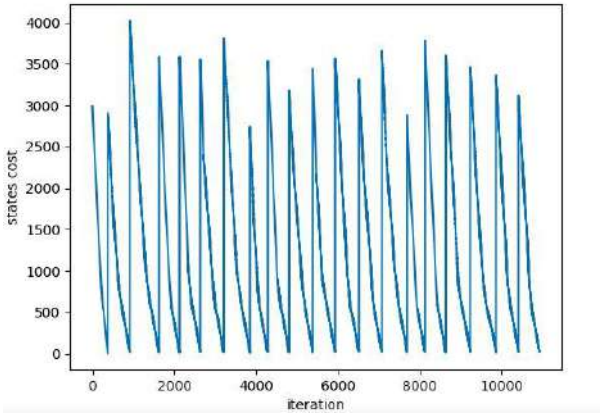
שבת	שישי	חמישי	רביעי	שלישי	שני	ראשון	משמרת
שומר 10 שומר 18	שומר 8 שומר 10	שומר 2 שומר 4 שומר 8 שומר 17 שומר 20	שומר 4 שומר 8 שומר 13 שומר 15 שומר 16	שומר 3 שומר 20	שומר 6 שומר 7	שומר 5 שומר 6 שומר 13 שומר 14 שומר 16	בוקר
שומר 6 שומר 7 שומר 16 שומר 17 שומר 20	שומר 2 שומר 12 שומר 16 שומר 17	שומר 3 שומר 10	שומר 3 שומר 18	שומר 7 שומר 9 שומר 15 שומר 16 שומר 19	שומר 4 שומר 10	שומר 2 שומר 20	צהריים
שומר 2 שומר 9 שומר 13 שומר 14 שומר 19	שומר 3 שומר 6 שומר 11 שומר 13 שומר 14	שומר 12 שומר 13 שומר 14 שומר 19	שומר 11 שומר 12	שומר 5 שומר 11 שומר 17	שומר 5 שומר 11	שומר 3 שומר 9	ערב

Hill climbing עם 40 random restart מצא פתרון מינימלי של 2, כלומר 2 אילוצים לא

סופקו. סידור המשמרות נראה כך:

שבת	שישי	חמישי	רביעי	שלישי	שני	ראשון	משמרת
שומר 4 שומר 8 שומר 10 שומר 15	שומר 4 שומר 7 שומר 8 שומר 10 שומר 11	שומר 4 שומר 17	שומר 3 שומר 16	שומר 13 שומר 20	שומר 10 שומר 12	שומר 1 שומר 16 שומר 18	בוקר
שומר 5 שומר 6 שומר 7 שומר 16 שומר 17 שומר 20	שומר 9 שומר 12 שומר 17 שומר 20	שומר 3 שומר 5 שומר 6 שומר 9 שומר 10	שומר 11 שומר 14	שומר 15 שומר 16	שומר 1 שומר 2 שומר 19	שומר 12 שומר 20	צוהריים
שומר 2 שומר 9 שומר 13 שומר 14 שומר 19	שומר 3 שומר 6 שומר 13 שומר 14	שומר 2 שומר 13 שומר 14	שומר 6 שומר 13	שומר 5 שומר 11 שומר 17	שומר 16 שומר 18	שומר 3 שומר 19	ערב

כאשר התנודות של הערכים שנמצאו:



השוואה שלישית:

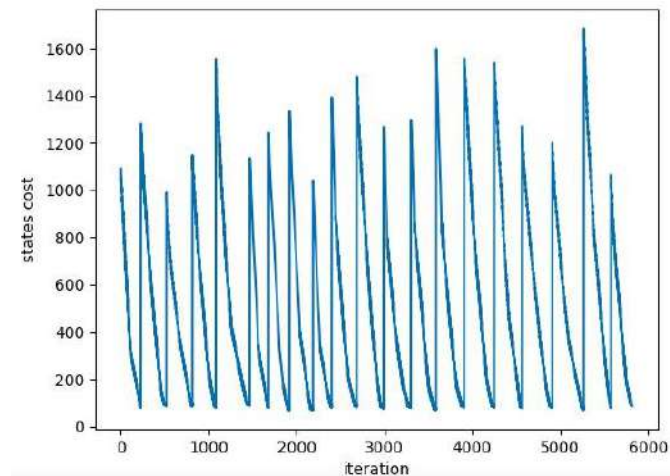
עשינו בדיקה על 10 שומרים כך שאחוז המשרה שלהם גורם שיהיה משמרת אחת בלי 2 שומרים ולכן לא מתקיים התנאי שבכל משמרת יהיה לפחות 2 שומרים. כלומר יש Hard constraint שלא מתקיים.

תכנות בשלמים לא מצא פתרון!

Hill climbing עם 20 random restart מינימלי של 69, כלומר ישנם גם אילוצים HC שלא סופקו וגם אילוצים SC. סידור המשמרות נראה כך:

שבת	שישי	חמישי	רביעי	שלישי	שני	ראשון	משמרת
שומר 4 שומר 9	שומר 10 שומר 8	שומר 9 שומר 10	שומר 3 שומר 4	שומר 6	שומר 3 שומר 4	שומר 1 שומר 7	בוקר
שומר 1 שומר 2	שומר 2 שומר 3	שומר 2 שומר 5		שומר 4 שומר 9	שומר 9	שומר 3 שומר 8	צוהריים
שומר 6 שומר 8	שומר 5 שומר 7	שומר 6 שומר 7	שומר 3	שומר 5 שומר 10	שומר 2 שומר 10	שומר 1 שומר 9	ערב

כאשר התנודות של הערכים שנמצאו:



השוואה 4:

עשינו בדיקה על 20 שומרים כאשר אף אחד מהשומרים לא רוצה לעבוד.

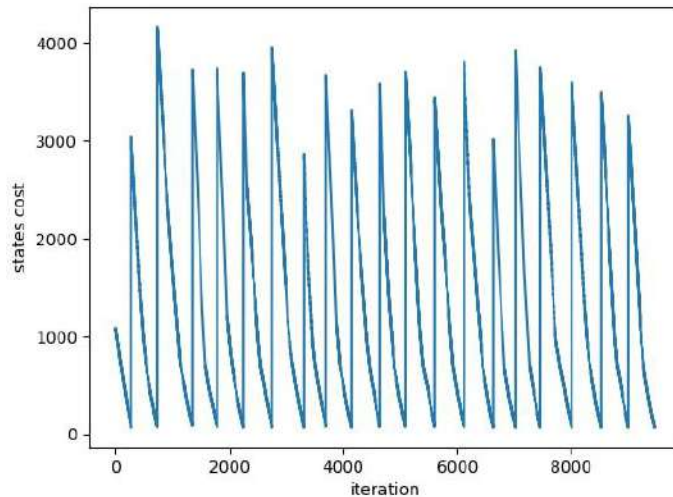
תכנות בשלמים לא מצא פתרון!

Hill climbing עם 20 random restart מנימלי של 101, כלומר ישנם גם אילוצים

HC שלא סופקו וגם אילוצים SC. סידור המשמרות נראה כך:

משמרת	ראשון	שני	שלישי	רביעי	חמישי	שישי	שבת
בוקר	שומר 3 שומר 5 שומר 13 שומר 14 שומר 17	שומר 1 שומר 7 שומר 8 שומר 12 שומר 14 שומר 20	שומר 4 שומר 11 שומר 13 שומר 14	שומר 2 שומר 5	שומר 2 שומר 7 שומר 13 שומר 19	שומר 2 שומר 3 שומר 6	שומר 3 שומר 12 שומר 13 שומר 16 שומר 18 שומר 19
צהריים	שומר 7 שומר 9	שומר 13 שומר 18	שומר 3 שומר 5 שומר 6	שומר 3 שומר 7 שומר 13 שומר 14	שומר 9 שומר 14 שומר 15	שומר 7 שומר 10 שומר 11 שומר 5 שומר 16	שומר שומר 6 שומר 9 שומר 17
ערב	שומר 2 שומר 13 שומר 18	שומר 3 שומר 5 שומר 9 שומר 17	שומר 7 שומר 12 שומר 13	שומר 8 שומר 11 שומר 12 שומר 18	שומר 5 שומר 7 שומר 8 שומר 12 שומר 16	שומר 1 שומר 5 שומר 9 שומר 14	שומר 1 שומר 5 שומר 7 שומר 8 שומר 14 שומר 18

כאשר התנודות של הערכים שנמצאו:



תובנות ומסקנות:

1. כאשר ישנו פתרון, אלגוריתם תכנות בשלמים ימצא פתרון אופטימלי חוקי, ויהיה עדיף על אלגוריתם hill climbing שיביא קירוב לפתרון האופטימלי.
2. אלגוריתם תכנות בשלמים נותן תוצאה טובה בזמן מהיר יותר מאשר אלגוריתם טיפוס הרים ולכן עדיף גם מהבחינה הזו.
3. אלגוריתם תכנות בשלמים יבטיח תוצאה אשר תספק את כל האילוצים שהם hard constraint. לעומתו, אלגוריתם hill climbing ינסה לעשות זאת וישתפר ככל שעוברים האיטרציות. זה גורם לכך שכאשר אין תוצאה אופטימלית, אלגוריתם תכנות בשלמים יחזיר שאין תוצאה ולעומתו אלגוריתם טיפוס הרים יגיע לתוצאה הקרובה ביותר שהצליח.

מה למדנו מהפרויקט:

למדנו מתי כדאי להשתמש בכל אלגוריתם:

- אם רוצים מודל "קשוח" שמספק באופן מוחלט את האילוצים הקשים ורק לאחר מכן מנסה לספק כמה שיותר אילוצים רכים – נבחר במודל של תכנות בשלמים.
- כאשר יש פתרון המספק את ה-hard constraints מודל התכנות בשלמים שמביא פתרון אופטימלי עדיף.
- כאשר אין אפשרות לספק את האילוצים הקשים, מודל טיפוס הרים נותן תוצאות טובות מאוד.