תזמון משימות לחישוב מקבילי בסביבה הטרוגנית

מאת: אליה אטלן, אביב זבולוני.

הבעיה בקטן

מה עושים כיום?

אלגוריתם זה משבץ שרירותית משימות למעבדים פנויים, כלומר דואג שלא יהיה מעבד חסר תעסוקה אם יש אפשרות

אם ישנם P מעבדים מסוגים שונים, ניתן להראות שמתזמן זה עלול להיות גרוע פי P מהפתרון האופטימלי.

כיום בתעשייה וב Mobileye משתמשים באלגוריתם "חמדן" על מנת לתזמן את המשימות.

אנחנו התמקדנו בבעיית התזמון הבאה:

מספר מעבדים מסוגים שונים. כל משימה רצה על מעבד אחד.

תלות בין תחילת משימה אחת לסיום משימה אחרת.

מתחילת המשימה הראשונה -> סיום המשימה האחרונה.

איזה מתזמנים יוכלו לשפר את זמן הריצה של האלגוריתם החמדן?

Q – <u>המעבדים</u>

C_{MAX} - <u>המטרה</u>

Precedence – <u>האילוצים</u>

 $\mathsf{type}: \ 1 \quad P \quad Q \quad R \quad O \quad F \quad J$

number of machines : \emptyset 2 3 4 5 m

₽₽

Objective function : $C_{
m max}$ $\sum C_j$ $\sum w_j C_j$ $L_{
m max}$

רכז פרויקטים: ד"ר אלי אנגלברג מנחה אקדמי: ד"ר צור לוריא



הבעיה בגדול

בעיית תזמון מעבדים היא בעיה שבה יש לקבוע באיזה סדר ומתי משימות יתבצעו במעבד, מתוך מטרה לייעל את ביצועיהם במערכת הפעלה

הבעיה מתעוררת כאשר יש מספר תהליכים שצריכים להתבצע על גבי מעבד אחד או יותר. אך המשאבים מוגבלים, ולכן צריך לתזמן אותם בצורה חכמה.

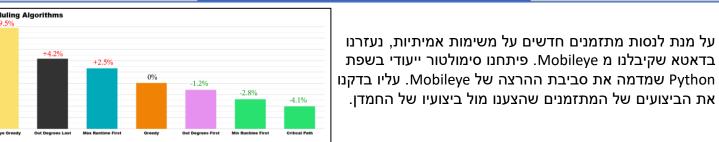
עם בעייה זו מתמודדת חברת Mobileye אשר עוסקת בפיתוח מערכות לנהיגה אוטונומית.



שיפור

יכול לרוץ על מעבד מסוג 1 🦲

יכול לרוץ על מעבד מסוג 2 יכול לרוץ על מעבד מסוג 3 🛑 יכול לרוץ על מעבד מסוג 4 🌑



Out Degrees First

להריץ משימות לפי מספר המשימות שתלויות בה - משימות ש"תוקעות" הרבה משימות ירוצו קודם.

Min Runtime First להריץ משימות לפי אורך זמן הריצה שלהם מהקטן לגדול - מונע תקיעה של המעבדים.

Critical Path

להריץ משימות לפי "הסדר הקריטי" שלהן - תזמון זה דואג לקדימות השיבוץ לפי המסילה הקריטית (בגרף הקדימויות) בכל

מסקנה: שיפור משמעותי במספר מתזמנים שהם לא החמדן!

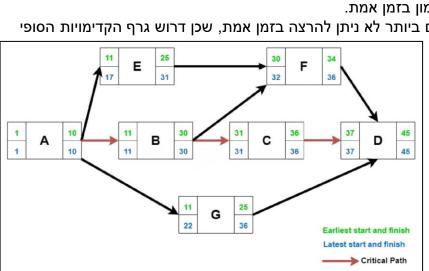
רב-משימתית (Multi-tasking).

התאמת המתזמן לפי "זמן קריטי" לתזמון בזמן אמת.

מתזמן זה שהיה בעל הביצועים הטובים ביותר לא ניתן להרצה בזמן אמת, שכן דרוש גרף הקדימויות הסופי על מנת לחשב זמן קריטי למשימות.

Online Critical Path

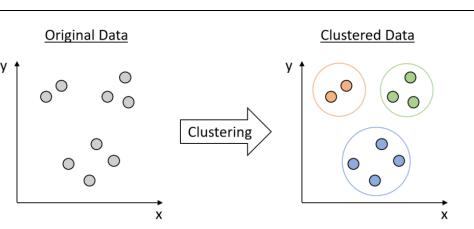
בעת ריצה חדשה, נסתמך על ריצות "קודמות בשביל לחשב "סדר קריטי עבור הריצה החדשה.



Clustering

חישוב מראש בזמן ארוך כרצוננו תזמון אופטימלי למספר פרופילים שכיחים.

בזמן אמת ברגע שתתחיל ריצה חדשה, ננסה להתאים אותה לאחד הפרופילים הקיימים ולתזמן אותה לפי



מתזמן פרקטי

?האם ניתן להציג שיפור גם על החומרה של Mobileye ואילוצי התזמון בזמן אמת

נדרשנו בשלב זה להטמיע יכולות חדשות בסימולטור שלנו על מנת להתמודד עם האילוצים הנוספים, כדי שנוכל לענות על השאלה.

התאמנו את המתזמנים הקודמים לחומרה החדשה בעזרת ערכי סף - משימה שמעל ערך סף מסויים היא תחשב דחופה אחרת היא לא.

מסקנה: אין שיפור מובהק. מניתוח התוצאות עולה שאין מתזמן פרקטי בשיטת ערכי הסף היוכל להחליף את המתזמן החמדן, בהינתן האילוצים הנוספים.

- נציע שני כיוונים מבטיחים: Clustering .1
- Online Critical Path .2

Offline

2. ידוע גרף התלויות בין המשימות

במתזמן כזה:

1. ידועים זמני הריצה של המשימות

<u>עד עכשיו</u>

<u>עכשיו יש א</u>ילוצים

2. לא ידוע גרף התלויות בין המשימות 3. לא ידוע סדר ההופעה של המשימות 3. וידוע סדר ההופעה של המשימות

יכלנו למיין כרצוננו, ללא שום הגבלות על זמן הריצה וסיבוכיות מקום.

החומרה של Mobileye תומכת בשני תורים בלבד – דחוף, ולא דחוף.

בנוסף לכך, עד עכשיו עסקנו בלנתח אלגוריתמי Offline.

בנוסף, המתזמן במציאות הוא מתזמן Online.

במתזמן כזה: 1. לא ידועים זמני הריצה

Online