הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל ע״ש אנדרו וארנה ויטרבי
המעבדה לבקרה, לרובוטיקה וללמידה חישובית

ספר פרויקט

אלגוריתם זמן-אמת לניהול שדה תעופה

מבצעים:

יואב כהן Yoav Cohen

הודיה כהן-אדיב Hodaya Cohen-Adiv

מנחה:

אייל טייטלר Eyal Taitler

סמסטר רישום: חורף תשפ״א

תאריך הגשה: אוקטובר, 2021

תוכן עניינים

רשימת איורים

תקציר

בפרויקט זה נממש אלגוריתם online המסתמך על אלגוריתם offline קיים.

אלגוריתם ה-online מקבל תוכנית אופטימלית של המראות ונחיתות מטוסים ליום עבודה, ובמהלך היום יצטרך להתמודד עם תקלות ושינויים בזמן-אמת. זאת כאשר ההיבט המרכזי הוא התמודדות עם התקלות והשינויים בזמן-אמת תוך סטייה מינימלית מהתוכנית המקורית.

במקרה שהאלגוריתם לא ימצא פתרון בזמן-אמת, הוא ישלח את המצב הקיים עם האילוץ לאלגוריתם ה-offline שיבצע תכנון מחדש לתוכנית.

את האלגוריתם פיתחנו בסביבת python על סמך אלגוריתם ה-offline הקיים. בבסיסו אלגוריתם ה-online צריך לדעת לקבל את נתוני הפלט של אלגוריתם ה-offline אודות התוכנית האופטימלית של זמני ההמראה והנחיתה של המטוסים. כמו כן, עליו לתמוך בקבלת התקלות והשינויים בזמן-אמת. בנוסף, עליו לספק החלטה אודות המשך התוכנית בהתאם לתקלות ולשינויים.

Abstract

In this project we will implement an online algorithm which is based on an existing offline algorithm.

The online algorithm receives an optimal program of airplane takeoffs and landings for the workday, and during the day the algorithm will need to handle incidents and changes in real time. The main focus is to handle incidents and changes in real time with minimal deviation from the original program.

If the algorithm does not find a real time solution, it will send the existing situation with the constraint to the offline algorithm which will reprogram the program.

We developed the algorithm in the python environment based on the existing offline algorithm.

The basis of the online algorithm should know to accept the offline algorithm data printout about the optimal program for airplane takeoff and landing times. In addition, it must be able to support the receipt of real time incidences and changes. It must also supply a decision about program continuation based on the incidents and changes.

1 מבוא

כידוע, שדה תעופה מורכב מפעולות רבות במהלך היום – הוא מכיל מספר רב של מטוסים ומספר רב של מסלולים, ועליו לסנכרן בין כל הפעולות השונות ולפקח עליהן. שדה תעופה מכיל סוגים שונים של פעולות לדג׳: פעולות רבות המתרחשות בו-זמנית, פעולות אשר תלויות אחת בשנייה, פעולות אשר יוצרות הפרעה ועלולות לשבש את סדר היום שנקבע מראש.

פיקוח ובקרה על שדה תעופה מתבצע בדרך כלל ע״י קביעת תוכנית לסדר-יום. תוכנית זו מכילה את לוח הזמנים עבור כל יום מראש. כאשר לוח הזמנים כולל את סדר ההמראות והנחיתות ואת זמנם, בנוסף הוא מציין את מספר המסלול עבור ההמראה והנחיתה של כל מטוס. כמו כן, הפיקוח והבקרה דורשים מעקב אחר התוכנית בזמן אמת, וטיפול בתקלות ובשינויים המתרחשים במהלך היום.

2 הגדרת הבעיה

בניית תוכנית לסדר-יום עבור סדר ההמראות והנחיתות של המטוסים, פיקוח ובקרה אחריה בזמן-אמת וטיפול בבעיות המתרחשות במהלך היום, היא בעיה מורכבת.

ניתן לחלק את הבעיה לשתי תתי-בעיות:

1. בניית תוכנית סדר-יום – זה הוא אלגוריתם ה-offline[1], אלגוריתם זה מתבסס על נתונים שמוזנים כקלט לבעיה.
2. פיקוח ובקרה אחר תוכנית סדר-היום בזמן אמת – זה הוא אלגוריתם ה-online שמימשנו במסגרת פרויקט זה. אלגוריתם זה מתבסס על תוכנית יום קיימת שמוזנת כקלט לבעיה.

2.1 אלגוריתם ה-OFFLINE

נרחיב מעט על אלגוריתם ה-offline, כיוון שאנו משתמשים בפלט שלו, כקלט לאלגוריתם ה-online.

אלגוריתם ה-offline מבוסס על תורת הגרפים, על יוריסטיקות ומקבל boost מרשתות נוירונים.

האלגוריתם מקבל כקלט קובץ קונפיגורציה המכיל את מספר המטוסים ואת מספר המסלולים בבעיה. בנוסף, עבור כל מטוס מכיל וקטור של נתונים התחלתיים. אלגוריתם ה-offline מחשב את סדר היום האופטימלי בהינתן הנתונים שהוגדרו לו. האלגוריתם מוציא כפלט את תכנית סדר-היום אשר חישב כגרף STN בצורת טבלה.

[1] אלגוריתם זה מומש בפרויקט קודם ע"י הסטודנטים בר מימרן ותום שפירא, בהנחיית איל טייטלר.
הפרויקט הנוכחי מתבסס על פרויקט ה-offline.

2.1.1 קובץ קונפיגורציה

קובץ קונפיגורציה חוקי מכיל את השדות הבאים (לפי הסדר משמאל לימין):

1. מזהה המטוס.
2. זמן המינימלי (יחסית לתחילת הבעיה) בו המטוס חייב להתחיל להתיישר על מסלול.
3. הזמן המקסימלי (יחסית לתחילת הבעיה) בו המטוס חייב להתחיל להתיישר על מסלול.
4. הזמן שלוקח למטוס לבצע את משימתו באוויר.
5. הזמן המקסימלי בו המטוס יכול להיות באוויר.
6. הזמן המקסימלי (יחסית לתחילת הבעיה) בו המטוס חייב לנחות ולפנות את מסלול הנחיתה.
7. סטטוס המציין האם בתחילת הבעיה המטוס נמצא על הקרקע (1) או באוויר (5).

נציין כי כל הזמנים בתת סעיפים 1-7 מוגדרים בדקות. בנוסף, קובץ הקונפיגורציה מכיל את מספר המטוסים, מספר המסלולים וחסם עליון לזמן מציאת פתרון.

דוגמא לקובץ קונפיגורציה חוקי:

number_of_planes = 3
number_of_lanes = 2
max_run_time = 3

plane id	start day min	start day max	mission duration	max fuel	end day	status
plane0	0	10	40	90	200	1
plane1	00	00	10	20	00	5
plane2	00	00	20	31	00	5

איור 1: קובץ קונפיגורציה חוקי

2.1.2 תכנון עם אילוצי זמנים

בשונה מתכנון קלאסי, תכנון עם זמנים ואילוצים מתמקד בבעיה בה לכל פעולה יש התחלה וסוף, כלומר לכל פעולה קיים ממד נוסף והוא זמן הביצוע של הפעולה. בתכנון עם זמנים, יש צורך לקבוע את זמני ההתחלה והסיום של הפעולות וכן את האילוצים ואת התלויות בין הפעולות השונות. לצורך פשטות, נהוג לקבוע את כלל הזמנים בבעיה ביחס לזמן תחילת הבעיה אשר נקבע להיות 0.

דוגמא לאילוץ אפשרי: בבעיה בה כלל הזמנים מוגדרים ברזולוציה של דקות. כאשר מטוס ממריא במשך 20 דקות, בין זמן 0 לזמן 20 לא אפשרי שמטוס אחר ינחת באותו המסלול ובאותו חלון זמנים. ולכן, צריך להיות סדר בין הפעולות והאילוצים בין המטוסים.

נציין כי בעיית הזמנים המאולצת היא בעיית אופטימיזציה עם אילוצים לכל דבר ועניין. לכן להגדרת ולמציאת פתרון חוקי כנדרש נעשה שימוש בכלי אופטימיזציה מתאים.

בנוסף נציין, כי האילוצים כשלעצמם בדרך-כלל פוסלים הרבה כיווני פתרון ובכך מקטינים משמעותית את מרחב המצבים האפשרי של הבעיה.

2.1.3 גרף STN (Simple Temporal Network)

את פתרון בעיית האופטימיזציה שתוארה ב-1.2.3 נהוג להציג בגרף, הנקרא גרף STN. זהו למעשה גרף המייצג את הממד הזמני של הפעולות שמבצע בפתרון הבעיה, כאשר כל פעולה מיוצגת על-ידי זמן ההתחלה היחסי שלה והזמן הכולל הדרוש לביצועה.

חשוב לציין כי מעצם היותו פתרון בעיית אופטימיזציה, הפתרון המוצג ב-STN הינו פתרון אופטימלי עבור מסלול הפתרון עבורו נפתרה בעיית האופטימיזציה.

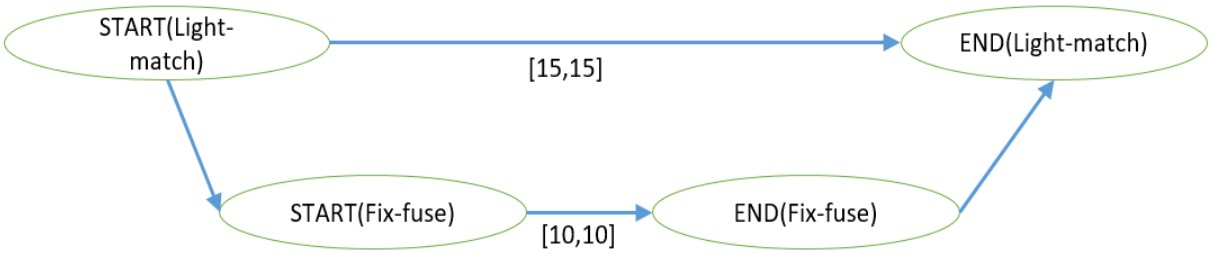
נדגים בעזרת 2 בעיות אופטימיזציה – אחת בעיה בסיסית, והשנייה בעיה מורכבת יותר המדגימה מתיחה של גרף STN.

- בעיית אופטימיזציה בסיסית:

תיקון פיוז לוקח בדיוק 10 דקות. התיקון חייב להתבצע כאשר הוא מלווה באור הבוקע מנר אשר יכול להיות דלוק למשך 15 דקות לכל היותר, זהו האילוץ.

יש למצוא פתרון אופטימלי הממזער את הזמן היחסי בו תיקון הפיוז הושלם.

פתרון הבעיה הנ"ל מתואר בגרף ה-STN באיור 4 מטה:



איור 2: פתרון בעיית האופטימיזציה הנ"ל מוצג בגרף STN

ניתן לראות כי לפי הפתרון שהתקבל בגרף ה-STN באיור 4, תיקון הפיוז החל אפסילון זמן לאחר הדלקת הנר ונמשך 10 דקות.

- בעיית אופטימיזציה מורכבת:

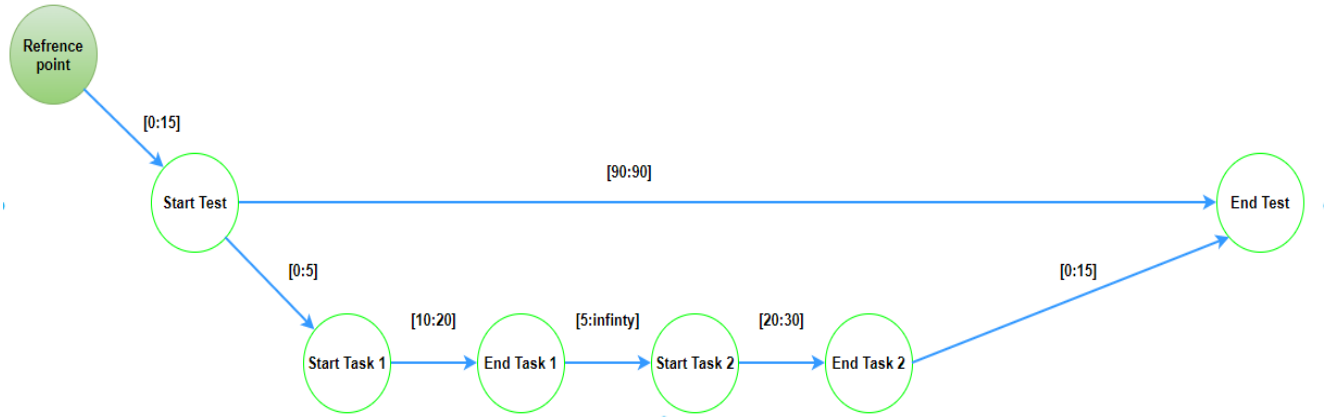
נדגים כעת מקרה יותר מורכב של גרף STN, בו נראה את היתרונות של שימוש במבנה נתונים זה:

הבעיה אותה נדגים היא ביצוע מבחן כניסה על ידי זוג סטודנטים לפני ניסוי במעבדה לבקרה.

את המבחן עליהם להתחיל יחד בזום ולסיים תוך 90 דקות. המבחן כולל 2 מטלות כאשר מטלה מספר 2 תלויה בסיום של מטלה מספר 1.

זוג הסטודנטים ישב והכין תכנית לביצוע המטלות בהתבסס על הידע שלהם והעריכו כי:

- תחילת המבחן תהיה עד 15 דקות החל מרגע תחילת המפגש בזום.
- ביצוע המטלה הראשונה תתחיל עד 5 דקות מתחילת המבחן.
- משך המטלה הראשונה יהיה בין 10 ל-20 דקות.
- המטלה השנייה תתחיל כ-5 דקות לאחר סיום המטלה הראשונה.
- משך המטלה השנייה יהיה בין 20 ל-30 דקות.
- סיום המבחן והגשתו ייקחו עד כ-15 דקות מסיום המטלה השנייה.

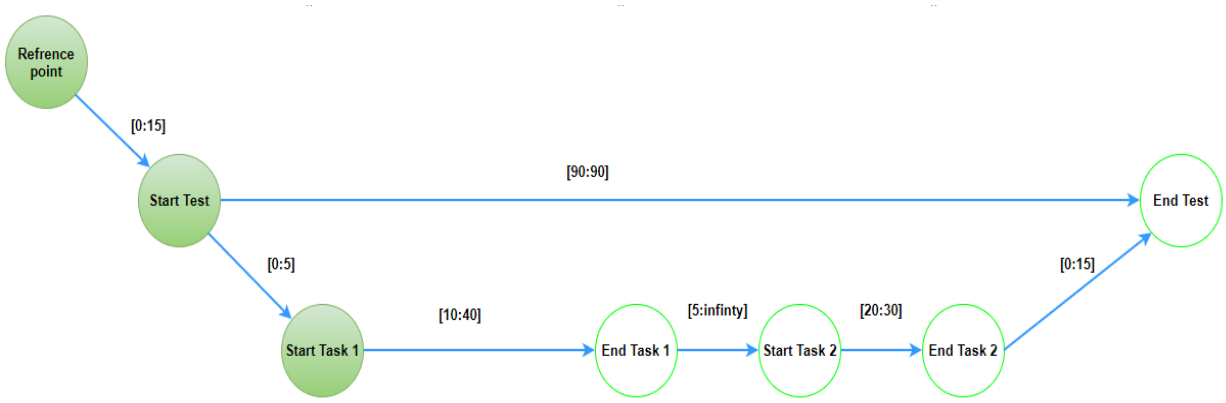


איור 3: פתרון בעיית המבחן ע"י הסטודנטים מוצג כגרף STN

כתוצאה מנפילת החיבור האינטרנטי פסק הזום בין שני הסטודנטים למשך 20 דקות.

המשמעות היא מתיחה של גרף ה-STN בבעיה על מנת לבצע תיקון מקומי.

ניתן לראות בגרף החדש שהחסם העליון עלה ל-40 אך הבעיה נשארה פיזיבילית.



איור 4: גרף ה-STN לאחר המתיחה

2.2 אלגוריתם ה-ONLINE

אלגוריתם ה-online מקבל כקלט את אותו קובץ קונפיגורציה שהוכנס לאלגוריתם ה-offline, ואת גרף ה-STN המיוצג כטבלה (פלט ה-offline).

פלט האלגוריתם הוא סדר הפעולות שקרו במהלך היום, כולל ההפרעות והאילוצים שהתרחשו.

אלגוריתם ה-online משמש כ-"מגדל פיקוח", אשר מפקח ומבקר על תוכנית סדר-היום שהתקבלה. האלגוריתם שולח את הפעולות לסביבת ההרצה ומבצע החלטות על בסיס מצב העולם וההפרעות המתרחשות.

3 מטרת הפרויקט

מטרת הפרויקט היא בניית אלגוריתם online לניהול המראות ונחיתות של מטוסים. האלגוריתם מנהל את שדה התעופה בהתאם לתוכנית סדר-היום שהתקבלה מאלגוריתם ה-offline. זאת תוך מעקב והתמודדות עם שינויים ואילוצים המתרחשים בזמן-אמת.

כאשר מתרחשת הפרעה אלגוריתם ה-online צריך לקבוע בזמן-אמת בין האפשרויות הבאות:

1. הוא יכול לרוץ עם ההפרעה והאילוצים שהתרחשו.
2. יש לבצע תיכנון מחדש ביחס למצב העולם בו הוא נמצא.
3. להחליט שהתוכנית לא אפשרית ולשלוח שגיאה.

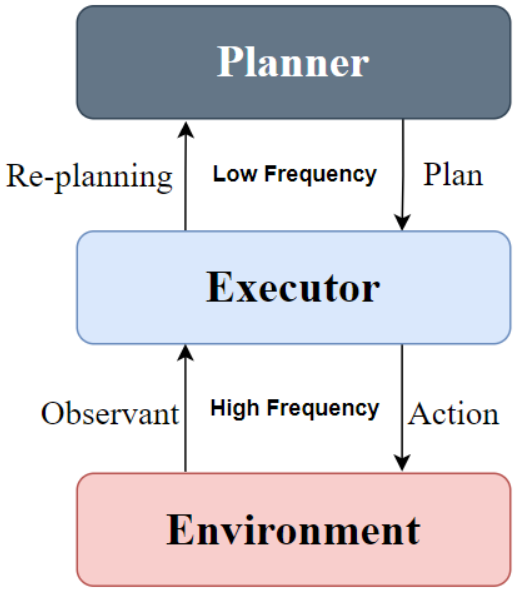
4 הנחות שבוצעו במסגרת הפרויקט

במסגרת הפרויקט הנחנו מספר הנחות:

- לא מתבצעות המראות ונחיתות בו-זמנית. כלומר, במרחב האווירי יש מטוס יחיד בכל רגע נתון.
- זמן התיכנון מחדש זניח ביחס לזמניי הבעיה. לכן, כאשר נדרש לבצע תיכנון מחדש, הקפאנו את מצב העולם ביחס למצב בו הוא נמצא.
- ההפרעה היחידה שביצענו היא עיכוב/דחייה של פעולה. כלומר, מתיחה של גרף ה-STN.

5 תיאור כללי של הבעיה

זוהי סכמה כללית של הבעיה המלאה:



איור 5: סכמה כללית של הבעיה

כאשר:

- ה-Planner הוא אלגוריתם ה-offline שמומש בפרויקט הקודם.
- ה-Executor הוא אלגוריתם ה-online.
- ה-Environment הוא הסביבה אשר מדמה את שדה התעופה כולל ההפרעות והאילוצים בזמן-אמת. בסביבה זו, בדקנו את אלגוריתם ה-online.

בנוסף כפי שצוין בסכמה – התקשורת בין ה-Planner ל-Executor היא בתדר נמוך, התקשורת מתבצעת מעט פעמים במהלך ריצת אלגוריתם ה-online. לעומת זאת, התקשורת בין ה-Executor ל-Environment היא בתדר גבוה, יש תקשורת רבה בין מגדל הפיקוח לשדה התעופה, בדומה לעולם האמיתי.

5.1 Executor

ה-Executor הוא אלגוריתם ה-online, אותו מימשנו במסגרת הפרויקט. האלגוריתם מקבל כקלט גרף STN, אשר מציג את תוכנית סדר-היום בצורת טבלה. ההיבט המרכזי של האלגוריתם הוא התמודדות עם תקלות ושינויים בזמן-אמת תוך סטייה מינימלית מהתוכנית המקורית. האלגוריתם עוקב ומפקח אחר התוכנית המקורית, הוא שולח את הפעולות לביצוע לפי תוכנית סדר-היום תוך מעקב אחר ההפרעות המתרחשות בזמן-אמת. עבור כל פעולה שנשלחת, האלגוריתם מבצע שלוש בדיקות:

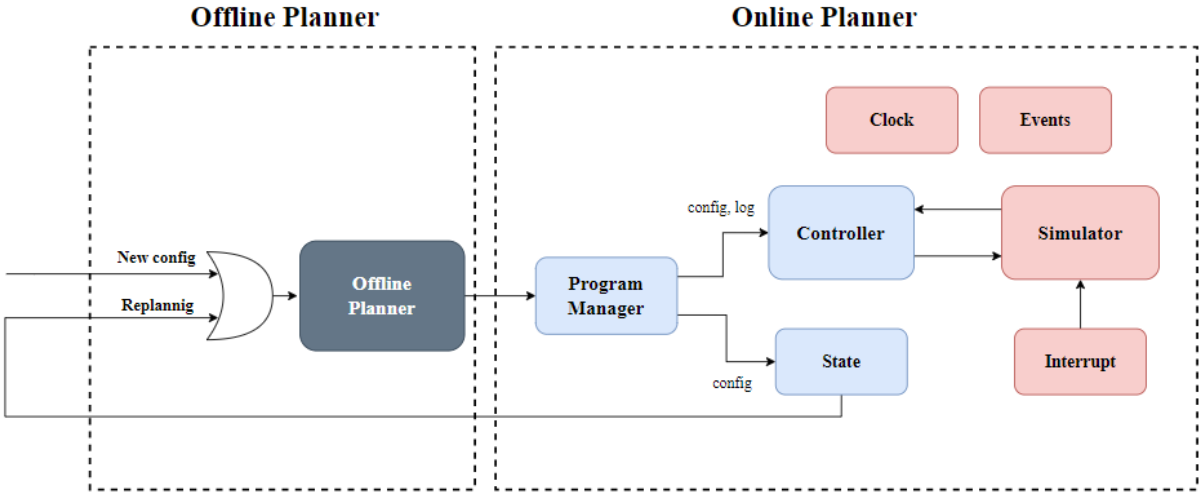
1. האם כל האבות של אותה פעולה אכן הסתיימו.
2. האם ביצוע הפעולה חוקי מבחינת מצב העולם, לדג' כאשר מטוס צריך לנחות, עלינו לוודא כי קיים מסלול פנוי שיאפשר את המצב הנ"ל.
3. האם הגיע זמן תחילת הפעולה לפי תוכנית סדר-היום.

כאשר מתרחשת הפרעה, היא עלולה לפגוע בכל אחת מהדרישות הנ"ל, ועל האלגוריתם להחליט על דרך התמודדות, ישנן 2 דרכים:

1. תיקון לוקאלי – התוכנית יכולה להמשיך למרות ההפרעה שהתרחשה ללא צורך בתכנון מחדש, התוכנית עדיין פיזיבילית. התהליך של תיקון לוקאלי היא המתנה של פעולות בזמן אמת ועקיפה של פעולות אחרות מה שגורם לשינוי מתוכנית סדר-היום המקורית.
2. תכנון מחדש – התוכנית לא יכולה להמשיך בעקבות ההפרעה, התוכנית איננה פיזיבילית. האלגוריתם שולח את מצב העולם הנוכחי בתוספת האילוץ לאלגוריתם ה-offline, לצורך תכנון מחדש. התהליך של תכנון מחדש מתבצע החל מנקודת ההפרעה, ולאחריה אנו ממשיכים את סדר הפעולות החדש שלנו.

6 תיאור כללי

להלן הסכמה הכללית של הפרויקט, ניתן לראות כי צבעי המודולים באיור 6 הינם בהתאמה לאיור 5.



איור 6: סכמה כללית של אלגוריתם ה-online

תקציר תפקידו של כל בלוק בסכמה:

1. **Offline Planner** – זה אלגוריתם ה-Offline. הוא מקבל כקלט קובץ קונפיגורציה היכול להיות אחד משני מקרים:
   - קובץ קונפיגורציה חדש המסמל על יום חדש.
   - קובץ קונפיגורציה שנבנה מתכנון מחדש (משוב).

   כמו כן, הוא מוציא כפלט את תוכנית סדר-היום (גרף ה-STN).
2. **Program Manager** – מקבל כקלט את תוכנית סדר-היום. בנוסף, הוא מייצר את המודולים השונים ואת הקשרים ביניהם.
3. **Controller** – מקבל כקלט את קובץ הקונפיגורציה שהוכנס ל- offline planner ואת תוכנית סדר-היום. תפקידו לשמש כמגדל פיקוח ובקרה.
4. **State** – מייצג את מצב העולם. הוא מציג בכל רגע נתון את מצב המטוסים, המסלולים, המרחב האווירי.
5. **Simulator** – משמש כמודל לעולם, אשר מריץ את הפעולות. בנוסף, הוא נותן אינדיקציה לתחילה ולסיום של כל פעולה.
6. **Interrupt** – אחראי על ההפרעות המתרחשות במהלך היום.

7. **Events** – מודול גלובלי המשמש כאמצעי תקשורת בין המודולים השונים.
8. **Clock** – מודל גלובלי אשר סופר זמן באופן דיסקרטי ומשמש כשעון בסביבה שיצרנו.

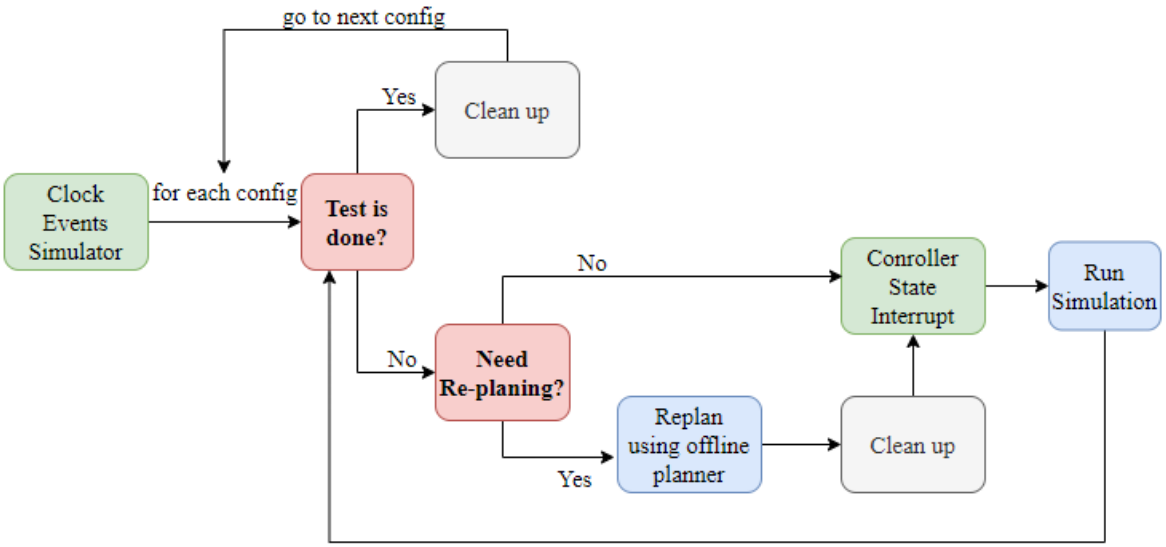
7 תיאור מפורט

7.1 Program Manager

ה-Program manager הוא החולייה המקשרת בין אלגוריתם ה-Offline לאלגוריתם ה-Online.

המודול מקבל את הפלט של ה-offline, ובונה את המודולים הרלוונטיים לאלגוריתם ה-online. המודול אחראי על יצירת המודולים הגלובליים ועל ניקיונם כאשר מתקבלת תוכנית חדשה. בנוסף, הוא אחראי על יצירת המודולים הלוקאלים עבור כל טסט ועל הקישור בניהם. כמו כן, המודול מבצע בדיקה האם יש צורך בתכנון מחדש, ובמידה שכן מריץ את אלגוריתם ה-offline.

בנוסף נציין כי בכל סוף סימולציה, ה-Program manager עובר לתוכנית סדר-היום הבאה.



איור 7: Program Manager Flow

7.2 Controller

ה-Controller משמש כמגדל פיקוח ובקרה, הוא חלק מה-Executer. הוא מחליט אילו פעולות יישלחו לסביבה, במטרה לבצע אותן. הוא מחזיק reference למודולים ה-clock וה-events, על מנת שיוכל לתקשר עם שאר המודולים בתוכנית. בנוסף, המודול משתמש בשני מבני נתונים:

1. גרף ה-STN.
2. Priority queue.

7.2.1 גרף ה-STN

גרף ה-STN נגזר מהפלט של אלגוריתם ה-offline, אשר מציג את התוכנית בצורת טבלה.

הטבלה מסודרת כך שכל שורה מייצגת פעולה מסוימת, והעמודות עבור כל פעולה מכילות את השדות הבאים (לפי הסדר משמאל לימין):

1. שם הפעולה.
2. זמן תחילה הפעולה.
3. רשימת ההורים של הפעולה – הפעולה הנוכחית יכולה להתחיל רק בתנאי שפעולות אלו אכן סיימו להתבצע.
4. רשימת בנים של הפעולה – פעולות אלה חייבות לחכות לסיום ביצוע הפעולה הנוכחית על מנת להתחיל.

דוגמא לפלט חוקי[2]:

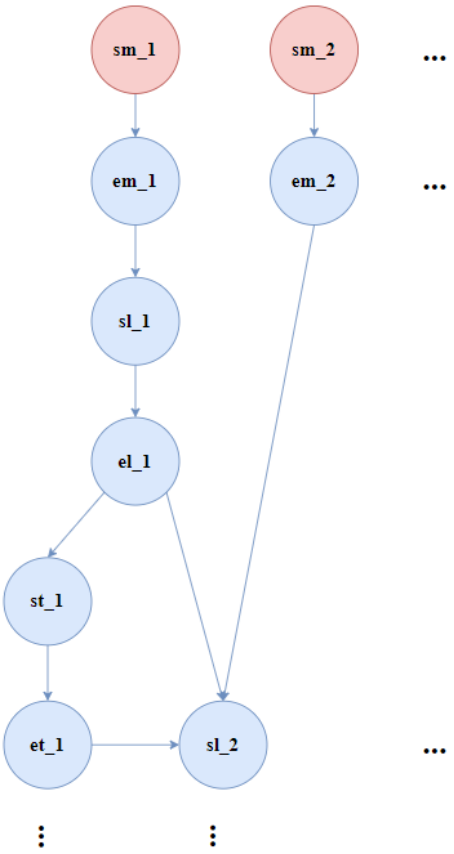
```
t_sm_1    : 0.0      :    parents: []                          :    childs: ['t_em_1']
t_sm_2    : 0.0      :    parents: []                          :    childs: ['t_em_2']
t_sctto_0 : 0.0      :    parents: []                          :    childs: ['t_ectto_0']
t_em_1    : 10.0     :    parents: ['t_sm_1']                  :    childs: ['t_sl_1']
t_ectto_0 : 10.0     :    parents: ['t_sctto_0']               :    childs: ['t_sto_0']
t_sl_1    : 10.001   :    parents: ['t_em_1']                  :    childs: ['t_el_1']
t_el_1    : 20.001   :    parents: ['t_sl_1']                  :    childs: ['t_sl_2', 't_st_1']
t_st_1    : 20.002   :    parents: ['t_el_1']                  :    childs: ['t_et_1']
t_et_1    : 30.002   :    parents: ['t_st_1']                  :    childs: ['t_sl_2']
t_em_2    : 30.002   :    parents: ['t_sm_2']                  :    childs: ['t_sl_2']
t_sl_2    : 30.003   :    parents: ['t_em_2', 't_et_1', 't_el_1']  :    childs: ['t_el_2']
t_el_2    : 40.003   :    parents: ['t_sl_2']                  :    childs: ['t_st_2', 't_sto_0']
```

איור 8: פלט של אלגוריתם ה-offline

במטרה לנהל את אלגוריתם ה-online, בחרנו להעביר את הפלט המופיע באיור 8 למבנה נתונים של גרף מכוון חסר מעגלים. בחרנו במבנה נתונים זה, על מנת לבצע בקלות חיפוש בגרף ולעקוב אחר התלויות. התלויות מיוצגות כקשרי אב ובן.

[2] פירוט מלא מופיע תחת קטגוריה נספחים

דוגמא לגרף STN שהומר מטבלה:



איור 9: גרף STN שהומר מטבלה

בגרף זה ניתן לראות לצומת sl_2 יש שני אבות - צומת el_1 וצומת em_2, ולכן צומת sl_2 תצטרך לחכות לסיום של שתי פעולות אלה, על מנת להתחיל להתבצע.

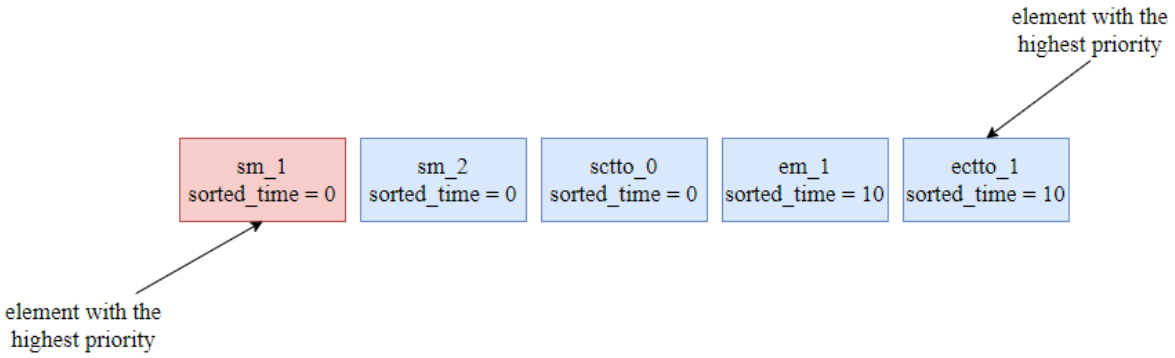
כל צומת מחזיק את הפרמטרים הבאים עבור הפעולה:

1. שם הפעולה.
2. מספר המטוס.
3. זמן תחילת הפעולה ביחס לתחילת הבעיה, על סמך אלגוריתם ה-offline.
4. Sorted-time – הזמן לפיו אנו ממיינים את הפעולה בהתאם להפרעות ולאילוצים[3].
5. Airtime – מציין כמה זמן המטוס נמצא באוויר.
6. Action ended – דגל המציין האם הפעולה הסתיימה או לאו.
7. רשימת האבות של הפעולה.
8. רשימת הבנים של הפעולה.

7.2.2 Priority Queue

ה-priority queue הוא תור העדיפויות שלפיו ה-controller שולח את הפעולות לביצוע. התור מאותחל לפי התוכנית המקורית שהתקבלה מאלגוריתם ה-offline.

התור ממוין לפי המשתנה sorted-time, שמאותחל לזמן תחילת הפעולה כפי שמוגדר בקלט מה-offline. משתנה זה יכול להשתנות בהמשך התוכנית, בהתאם לתיקונים הלוקאליים. פעולה בעלת ה-sorted-time הכי נמוך, תהיה בעלת העדיפות הגבוהה ביותר. כל תא בתור מחזיק מצביע לצומת הרלוונטי של הפעולה בגרף ה-STN.



איור 10: תור העדיפויות של ה-controller

[3] פירוט בהמשך

לפני שה-controller שולח פעולה לסביבה במטרה לבצע אותה, הוא מבצע מספר בדיקות:

1. האם האבות של הפעולה סיימו – מתבצע ע״י חיפוש בגרף.
2. האם הגיע הזמן של הפעולה להתבצע, כלומר sorted-time = 0.
3. בדיקת חוקיות הפעולה מול מודול מצב העולם. לדוגמא, כאשר הפעולה הבאה היא נחיתה, פונקציית החוקיות בודקת האם אכן יש מסלול פנוי כדי לבצע את הפעולה.

פעולות שהגיע זמנן להתבצע (sorted-time = 0), אך לא מוכנות עקב הפרעה או שינוי הפעולות ע״י ה-controller. מקבלות קנס של יחידת זמן אחת המתווסף ל-sorted-time, ובכך אנו משנים את סדר הפעולות של התוכנית המקורית בהתאם להפרעות ולאילוצים. זהו הוא התיקון הלוקאלי של גרף ה-STN, כלומר, המתיחה שלו.

נציין כי כל פעולה שהסתיימה מרימה את דגל Action ended בצומת שלה בגרף, וכאשר כל הפעולות הסתיימו ותור העדיפויות ריק, הבקר מאותת כי הוא סיים את פעולתו.

7.3 Simulator

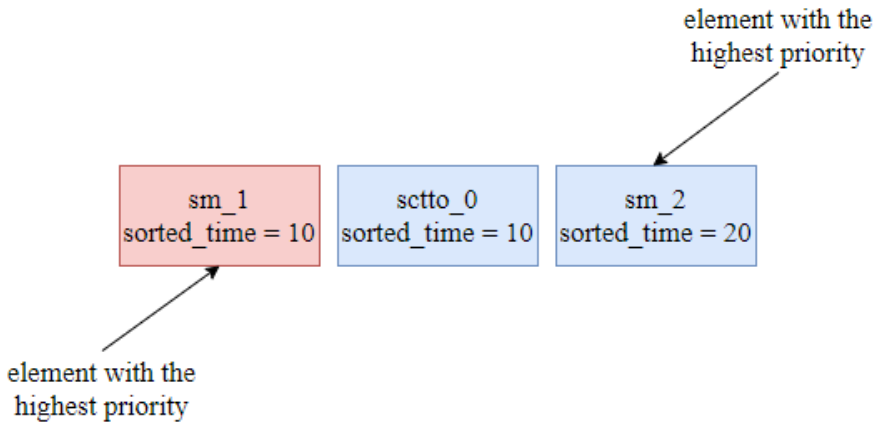
ה-Simulator הוא הגוף המבצע, הוא חלק מה-Environment. אין לו יכולת החלטה, אלא הוא מקבל פעולות ופקודות ומבצע אותן. ה-Simulator מדמה את העולם.

בדומה ל-controller, ה-simulator מחזיק reference למודולים ה-clock וה-events, על מנת שיוכל לתקשר עם שאר המודולים בתוכנית. בנוסף, המודול משתמש במבנה נתונים priority queue הממוין באופן שונה מה-controller.

7.3.1 Priority Queue

ה-priority queue הוא תור העדיפויות שלפיו ה-simulator עוקב אחר הפעולות. לפי תור עדיפויות זה ה-simulator יודע אילו פעולות מתבצעות בכל רגע נתון.

התור ממוין לפי המשתנה sorted-time. בשונה מה-controller , משתנה זה מציין את זמן משך הפעולה. המשתנה מאותחל למשך הפעולה כפי שהוגדר בנתונים שהוכנסו לאלגוריתם ה-offline. אך, הוא יכול להשתנות בהמשך התוכנית, בהתאם להפרעות ולאילוצים המתווספים בזמן-אמת. הפעולות מסודרות לפי משך הפעולה, כלומר פעולה הקצרה ביותר, תהיה בעלת העדיפות הגבוהה ביותר.



איור 11: תור העדיפויות של ה-simulator

כאשר נשלחת הפרעה ע״י מודול ה-Interrupt, ה-simulator מוסיף את משך ההפרעה לפעולה המתאימה וממיין מחדש את התור בהתאם. עבור כל מחזור שעון, ה-simulator מוריד יחידת זמן מכלל הפעולות שמתבצעות, בכך מתבצע קידום של מצב העולם והזמן הדיסקרטי שיצרנו. בכל התחלה או סיום של פעולה, מתבצע עדכון במודול ה-state.

נציין כי ה-simulator ממומש באופן מקבילי[4], כלומר ייתכן שבאותו מחזור שעון יתחילו או יסיימו מספר פעולות במקביל.

7.3.2 State

ה-State מייצג את מצב העולם בכל רגע נתון, המודול מאזין ל-simulator במטרה להתעדכן כאשר פעולה מתחילה או מסיימת. המודול אחראי על פונקציית החוקיות, בה ה-controller משתמש, ולהחזיר האם המצב הבא חוקי או לא. בנוסף, הוא מבצע הקפאה של מצב העולם וכתיבה של קובץ קונפיגורציה חדש כאשר נדרש תכנון מחדש.

ה-state מחזיק שני מערכים – אחד למטוסים ואחד למסלולים.

- מערך המטוסים:

כל אינדקס במערך שייך למספר המטוס של אותו האינדקס וכל תא מחזיק את מצב המטוס בכל רגע נתון.

דוגמא למערך מטוסים:

sl	sm	ectto
Plane 0	Plane 1	Plane 2

איור 12: מערך המטוסים

מדוגמא זו, ניתן לראות כי מטוס מספר 0 נמצא במצב sl, ומטוס מספר 1 נמצא במצב sm.

[4] חשוב לציין כי המקביליות לא תוכנתית, אלא אלגוריתמית. כלומר, אין חוטים או מנגנון חומרתי, אלא תקשורת בין המודולים על מנת שיתבצעו מספר דברים במקביל.

המצבים האפשריים עבור המטוסים הוגדרו לנו לפי פרויקט ה-offline :

1. (SCTTO) Start Clear To Take Off – מטוס מתחיל את תהליך התיישרות על מסלול ההמראה.
2. (ECTTO) End Clear To Take Off – מטוס מסיים את תהליך ההתיישרות על מסלול ההמראה.
3. (STO) Start Take Off – מטוס מתחיל את תהליך המראה.
4. (ETO) End Take Off – מטוס מסיים את תהליך המראה.
5. (SM) Start Mission – מטוס מתחיל את משימתו באוויר.
6. (EM) End Mission – מטוס מסיים את משימתו באוויר.
7. (SL) Start Landing – מטוס מתחיל את תהליך נחיתה.
8. (EL) End Landing – מטוס מסיים את תהליך נחיתה.
9. (ST) Start Taxi – מטוס מתחיל את תהליך פינוי מסלול הנחיתה.
10. (ET) End Taxi – מטוס מסיים את תהליך פינוי מסלול הנחיתה.

- מערך המסלולים :

מערך המסלולים מייצג את המסלולים באותו אופן, כל תא מחזיק ערך מספרי כאשר :

- ערך (-1) – מייצג מסלול פנוי.
- כל מספר אחר – מייצג את מספר המטוס אשר תופס את המסלול ברגע הנתון.

דוגמא למערך מסלולים :

0	2	-1
Lane 0	**Lane 1**	**Lane 2**

איור 13: מערך המסלולים

מדוגמא זו, ניתן לראות כי מטוס מספר 0 תופס את מסלול 0, וכי מסלול 2 פנוי.

7.3.3 כתיבת קובץ קונפיגורציה חדש

כתיבת קובץ קונפיגורציה חדש מתבצע כאשר יש צורך בתכנון מחדש. מודול מצב העולם מבצע תרגום של מצב הבעיה לקובץ קלט לאלגוריתם ה-Offline.

המודול מתחיל במעבר על וקטור המטוסים, עבור כל מטוס הוא בודק:

- אם המטוס על הקרקע (מצבים 0-3) - המטוס עובר למצב 1.
- אם המטוס סיים המראה או התחיל משימה (מצבים 4-5) - המטוס עובר למצב 5.
- אם המטוס התחיל נחיתה או על הקרקע (מצבים 7-10) - המטוס עובר למצב הסופי, מצב 11.
- מצב מיוחד הוא מצב 6 - מצב זה מייצג מטוס שסיים את משימתו, ולכן לאחר התכנון מחדש מקבל עדיפות גבוהה לנחיתה מיידית.

לקובץ הקונפיגורציה יודפס מספר המטוסים העדכני – מספר המטוסים של הבעיה המקורית פחות מספר המטוסים במצב 11.

לבסוף המודול עובר שוב על ווקטור המטוסים ועבור כל מטוס שאינו במצב 11 מכניס ווקטור של ערכים מעודכנים בקובץ הקונפיגורציה החדש.

7.4 Clock

ה-clock מדמה לנו את ציר הזמן אשר מיוצג באופן דיסקרטי. הוא משמש כמודול גלובלי כחלק מה-Environment.

נציין כי בתחילת העבודה ייצגנו את השעון בזמן רציף, אך לאחר בעיות מימוש והתייעצות עם מנחה הפרויקט, החלטנו לעבור לזמן בדיד.

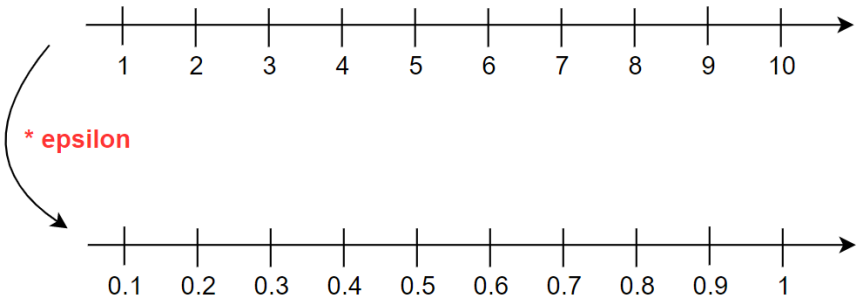
היתרונות בזמן בדיד הן:

- יכולת לנרמל את ציר הזמן בקלות.
- יכולת מעקב אחר הפעולות בצורה יעילה.
- הרצת תוכניות סדר-היום בהילוך מהיר.

הפרמטרים העיקריים במודול זה הם:

- ערך השעון – מספר יחידות הזמן שעברו.

- אפסילון – ערך זה הוא הפקטור לפיו ציר הזמן מתכווץ או נמתח. ערך האפסילון נקבע שרירותית לפני תחילת הריצה. במהלך ריצת ה-online מתבצעות מספר המרות על סמך ערך האפסילון.



איור 14: כיווץ ציר זמן בעזרת שימוש באפסילון

השעון מבצע ריצה אינסופית, כאשר בכל איטרציה הוא בודק שני דברים:

1. האם תוכנית סדר-היום הסתיימה.
2. האם יש צורך בקידום השעון (כתלות בפעולות שהתבצעו ב- simulator וב-controller).

אחת התקלות המשמעותיות שנתקלנו בהן במהלך הפרויקט, הייתה תוצאת לוואי של כיווץ ציר הזמן. כיוון שהכפלנו את הזמנים בתוכנית במספרים ממשיים ולמחשב יש יכולת ייצוג מוגבלת של מספרים מסוג זה, במהלך הריצות נתקלנו בתופעות לא מוסברות ובסימולציות שנכשלו כאשר ציפינו שהן יצליחו.

לאחר מעבר על התוצאות, הבנו שהבעיה הייתה בהשוואת זמני הפעולות שהסתיימו ל-0 מסוג שלם (integer).

בעיה זו יצרה עיכובים בסיום הפעולות, דבר שגרר כישלון של סימולציות.

תיקנו את הבעיה בכך שיצרנו אפס לוגי חדש. הגדרנו את האפס להיות קטן בשני סדרי גודל מהאפסילון, במטרה שהוא יהיה המספר הקטן ביותר בתוכנית, ובכך פתרנו את בעיית ההשוואה שנוצרה.

7.5 Events

ה-events הוא פרוטוקול התקשורת אשר דרכו כל המודולים מתקשרים. הוא משמש כמודול גלובלי כחלק מה-Environment. ה-events יוצר את התדירות הגבוהה בתקשורת בין ה-online ל-environment.

נציין כי בתחילת העבודה מימשנו את התקשורת ישירות בין המודולים, אך זה גרר צימוד גבוה של התכנית שלנו (תלויות רבות), ולכן עלה הצורך ביצירה של מודול מסוג זה. המודול הוא רשימה של סיגנלים אשר מוגדרים באופן הבא:

Signal	Type	Source	Destination
sa	List	Controller	Simulator
cena	Boolean	Clock	Controller, Simulator
cfa	Boolean	Simulator	Clock
fa	List	Simulator	Controller
cd	Boolean	Controller	Simulator, Clock
sd	Boolean	Simulator	Clock
rand	Float / Boolean	Interrupt	Simulator
rp	Integer / Boolean	Controller	Clock
test	Integer / Boolean	Clock	Program manger

משמעות הסיגנלים:

- **Sa** – ה-controller מאותת ל-simulator שנשלחה אליו פעולה.
- **Cena** – השעון מאותת שמתבצע תהליך סנכרון בעקבות מספר פעולות שמסתיימות\מתחילות בו זמנית.
- **Cfa** – ה-simulator מאותת לשעון כי הסתיימה פעולה.
- **Fa** – ה-simulator מאותת לבקר כי הסתיימה פעולה.
- **Cd** – ה-controller מאותת ל-simulator לשעון כי הוא סיים את פעולתו.
- **Sd** – ה-simulator מאותת לשעון כי הוא סיים את פעולתו.
- **Rand** – מודול ההפרעות מאותת ל-simulator על הפרעה חדשה שנוצרה.
- **Rp** – ה-controller מאותת לשעון על צורך בתכנון מחדש.
- **Test** – השעון מסמן ל-program manager על צורך בתכנון מחדש או מעבר לטסט הבא.

7.6 Interrupt

ה-Interrupt אחראי על יצירת הפרעות, במטרה לדמות עולם מציאותי. הוא שייך לחלק ה-Environment.

המודול מייצר הפרעות אשר נשלחות ל-simulator, הפרעה יכולה להתפרש כשני מצבים:

1. דחייה של פעולה מסוימת, כלומר תחילת זמן ההתחלה שלה.
2. עיכוב של פעולה מסוימת, כלומר הארכת משך זמן הפעולה.

ה-Interrupt רנדומלי בשני מובנים:

1. תדירות יצירת ההפרעה הוא משתנה אקראי אחיד בין 1 ל-4 פעולות. בחרנו במספרים אלו כדי למצוא איזון בהופעת ההפרעות.
2. משך ההפרעה הוא משתנה אקראי מפולג יוניפורמי בין מחצית משך הפעולה שנבחרה לבין משך הפעולה כולו.

כאשר נוצרת הפרעה המודול מאותת ל-simulator על ההפרעה שנוצרה ויאפס את תדירות יצירת ההפרעה.

8 תוצאות

במהלך בדיקת אלגוריתם ה-online ביצענו מספר תרחישים:

1. מודול ההפרעות כבוי – בדיקה שאלגוריתם ה-online מריץ באופן מדויק את התוכנית המתקבלת מאלגוריתם ה-offline.

כאשר מודול ההפרעות דלוק, ישנם שני תרחישים:

2. אלגוריתם ה-online מריץ את התוכנית עם תיקונים לוקאליים ללא תכנון מחדש בהתאם להפרעות המתקבלות.
3. אלגוריתם ה-online מקבל הפרעה שגורמת לתכנון מחדש.

כלל התרחישים המוצגים התבצעו עבור קובץ קונפיגורציה מספר 0:

ProjectA > BT_ProA > configs > config0.txt

```
number_of_planes = 3
number_of_lanes = 2
max_run_time = 3
plane0 0  10 40 90 200   1   3 4
plane1 00 00 10 20 00    5   1
plane2 00 00 20 31 00    5   2
```

איור 15: קובץ קונפיגורציה מספר 0

תוכנית סדר-היום שהתקבלה מאלגוריתם ה-offline:

ProjectA > BT_ProA > logs > log_output_0.txt

```
t_sm_1    : 0.0     :  parents: []                          :  childs: ['t_em_1']
t_sm_2    : 0.0     :  parents: []                          :  childs: ['t_em_2']
t_sctto_0 : 0.0     :  parents: []                          :  childs: ['t_ectto_0']
t_em_1    : 10.0    :  parents: ['t_sm_1']                  :  childs: ['t_sl_1']
t_ectto_0 : 10.0    :  parents: ['t_sctto_0']               :  childs: ['t_sto_0']
t_sl_1    : 10.001  :  parents: ['t_em_1']                  :  childs: ['t_el_1']
t_el_1    : 20.001  :  parents: ['t_sl_1']                  :  childs: ['t_st_1', 't_sl_2']
t_st_1    : 20.002  :  parents: ['t_el_1']                  :  childs: ['t_et_1']
t_et_1    : 30.002  :  parents: ['t_st_1']                  :  childs: ['t_sl_2']
t_em_2    : 30.002  :  parents: ['t_sm_2']                  :  childs: ['t_sl_2']
t_sl_2    : 30.003  :  parents: ['t_em_2', 't_et_1', 't_el_1']  :  childs: ['t_el_2']
t_el_2    : 40.003  :  parents: ['t_sl_2']                  :  childs: ['t_st_2', 't_sto_0']
t_st_2    : 40.004  :  parents: ['t_el_2']                  :  childs: ['t_et_2']
t_sto_0   : 40.004  :  parents: ['t_ectto_0', 't_el_2']     :  childs: ['t_eto_0']
t_et_2    : 50.004  :  parents: ['t_st_2']                  :  childs: []
t_eto_0   : 50.004  :  parents: ['t_sto_0']                 :  childs: ['t_sl_0', 't_sm_0']
t_sm_0    : 50.005  :  parents: ['t_eto_0']                 :  childs: ['t_em_0']
t_em_0    : 90.005  :  parents: ['t_sm_0']                  :  childs: ['t_sl_0']
t_sl_0    : 90.006  :  parents: ['t_eto_0', 't_em_0']       :  childs: ['t_el_0']
t_el_0    : 100.006 :  parents: ['t_sl_0']                  :  childs: ['t_st_0']
t_st_0    : 100.007 :  parents: ['t_el_0']                  :  childs: ['t_et_0']
t_et_0    : 110.007 :  parents: ['t_st_0']                  :  childs: []
```

איור 16: פלט אלגוריתם ה-offline עבור קובץ קונפיגורציה מספר 0

תוצאות אלגוריתם ה-online עבור התרחישים שהוצגו לעיל:

1. פלט אלגוריתם ה-online ללא מודול ההפרעות:

```
ProjectA > HY_ProA > Logs > log_output0.txt
Simulator: started - sctto 0, time is - 0
Simulator: started - sm 2, time is - 0
Simulator: started - sm 1, time is - 0
Simulator: action done - sctto 0, time is - 10
Simulator: action done - sm 1, time is - 10
Simulator: started - ectto 0, time is - 10
Simulator: started - em 1, time is - 10
Simulator: action done - ectto 0, time is - 10
Simulator: action done - em 1, time is - 10
Simulator: started - sl 1, time is - 10
Simulator: action done - sl 1, time is - 20
Simulator: action done - sm 2, time is - 20
Simulator: started - el 1, time is - 20
Simulator: action done - el 1, time is - 20
Simulator: started - st 1, time is - 20
Simulator: started - em 2, time is - 30
Simulator: action done - st 1, time is - 30
Simulator: action done - em 2, time is - 30
Simulator: started - et 1, time is - 30
Simulator: action done - et 1, time is - 30
Simulator: started - sl 2, time is - 30
Simulator: action done - sl 2, time is - 40
Simulator: started - el 2, time is - 40
Simulator: action done - el 2, time is - 40
Simulator: started - sto 0, time is - 40
Simulator: started - st 2, time is - 40A
Simulator: action done - sto 0, time is - 50
Simulator: action done - st 2, time is - 50
Simulator: started - eto 0, time is - 50
Simulator: started - et 2, time is - 50
Simulator: action done - eto 0, time is - 50
Simulator: action done - et 2, time is - 50
Simulator: started - sm 0, time is - 50
Simulator: action done - sm 0, time is - 90
Simulator: started - em 0, time is - 90
Simulator: action done - em 0, time is - 90
Simulator: started - sl 0, time is - 90
Simulator: action done - sl 0, time is - 100
Simulator: started - el 0, time is - 100
Simulator: action done - el 0, time is - 100
Simulator: started - st 0, time is - 100
Simulator: action done - st 0, time is - 110
Simulator: started - et 0, time is - 110
Simulator: action done - et 0, time is - 110
```

איור 17: פלט אלגוריתם ה-online ללא מודול ההפרעות

ניתן לראות כי ה-simulator מסיים כצפוי לפי התוכנית.

2. פלט אלגוריתם ה-online כאשר התבצעו תיקונים לוקאליים ללא תכנון מחדש, בהתאם להפרעות שהתקבלו:

```
ProjectA > HY_ProA > Logs > log_output0.txt
Interrupt: Added 6.5859 time units to action sctto
Simulator: started - sctto 0, time is - 0
Simulator: started - sm 2, time is - 0
Simulator: started - sm 1, time is - 0
Simulator: action done - sm 1, time is - 10
Simulator: started - em 1, time is - 10
Simulator: action done - em 1, time is - 10
Interrupt: Added 7.2489 time units to action sl 1
Simulator: started - sl 1, time is - 10
Simulator: action done - sctto 0, time is - 17
Simulator: started - ectto 0, time is - 17
Simulator: action done - ectto 0, time is - 17
Simulator: action done - sm 2, time is - 20
Simulator: action done - sl 1, time is - 28
Interrupt: Added 0.0007 time units to action el 1
Simulator: started - el 1, time is - 28
Simulator: action done - el 1, time is - 28
Simulator: started - st 1, time is - 28
Interrupt: Added 0.0008 time units to action em 2
Simulator: started - em 2, time is - 30
Simulator: action done - em 2, time is - 30
Simulator: action done - st 1, time is - 38
Simulator: started - et 1, time is - 38
Simulator: action done - et 1, time is - 38
Interrupt: Added 9.8883 time units to action sl 2
Simulator: started - sl 2, time is - 38
```

```
Simulator: action done - sl 2, time is - 58
Simulator: started - el 2, time is - 58
Simulator: action done - el 2, time is - 58
Interrupt: Added 7.3232 time units to action sto 0
Simulator: started - sto 0, time is - 58
Simulator: started - st 2, time is - 58
Simulator: action done - st 2, time is - 68
Simulator: started - et 2, time is - 68
Simulator: action done - et 2, time is - 68
Simulator: action done - sto 0, time is - 76
Simulator: started - eto 0, time is - 76
Simulator: action done - eto 0, time is - 76
Interrupt: Added 28.3852 time units to action sm 0
Simulator: started - sm 0, time is - 76
Simulator: action done - sm 0, time is - 145
Interrupt: Added 0.0006 time units to action em 0
Simulator: started - em 0, time is - 145
Simulator: action done - em 0, time is - 145
Simulator: started - sl 0, time is - 145
Simulator: action done - sl 0, time is - 155
Simulator: started - el 0, time is - 155
Simulator: action done - el 0, time is - 155
Interrupt: Added 7.8836 time units to action st 0
Simulator: started - st 0, time is - 155
Simulator: action done - st 0, time is - 173
Simulator: started - et 0, time is - 173
Simulator: action done - et 0, time is - 173
```

איור 18: פלט אלגוריתם ה-online עבור תיקונים לוקאליים ללא תכנון מחדש

ניתן לראות כי ה-simulator מסיים לבצע את תוכנית סדר-היום ללא תכנון מחדש לאחר 173 יחידות זמן, ולא לאחר 110 יחידות זמן כפי שמופיע בתוכנית סדר-היום המתקבלת מאלגוריתם ה-offline. בנוסף, ניתן לראות כי ההפרעות מתווספות באופן רנדומלי כמצופה.

3. פלט אלגוריתם ה-online כאשר ה-simulator מקבל הפרעה שגורמת לתכנון מחדש:

```
ProjectA > HY_ProA > Logs > log_output0.txt
Interrupt: Added 7.9250 time units to action sctto 0
Simulator: started - sctto 0, time is - 0
Simulator: started - sm 2, time is - 0
Simulator: started - sm 1, time is - 0
Simulator: action done - sm 1, time is - 10
Interrupt: Added 0.0007 time units to action em 1
Simulator: started - em 1, time is - 10
Simulator: action done - em 1, time is - 10
Interrupt: Added 7.5444 time units to action sl 1
Simulator: started - sl 1, time is - 10
Simulator: action done - sctto 0, time is - 18
Interrupt: Added 0.0005 time units to action ectto 0
Simulator: started - ectto 0, time is - 18
Simulator: action done - ectto 0, time is - 18
Simulator: action done - sm 2, time is - 20
Simulator: action done - sl 1, time is - 28
Interrupt: Added 0.0009 time units to action el 1
Simulator: started - el 1, time is - 28
Simulator: action done - el 1, time is - 28
Simulator: started - st 1, time is - 28
Interrupt: Added 0.0008 time units to action em 2
Simulator: started - em 2, time is - 30
Simulator: action done - em 2, time is - 30
Simulator: action done - st 1, time is - 38
Simulator: started - et 1, time is - 38
Simulator: action done - et 1, time is - 38
Interrupt: Added 9.5346 time units to action sl 2
Simulator: started - sl 2, time is - 38
Simulator: action done - sl 2, time is - 58
Interrupt: Added 0.0006 time units to action el 2
Simulator: started - el 2, time is - 58
Simulator: action done - el 2, time is - 58
Interrupt: Added 131.5400 time units to action sto 0
Simulator: started - sto 0, time is - 58
Simulator: started - st 2, time is - 58
Simulator: action done - st 2, time is - 68
Interrupt: Added 0.0009 time units to action et 2
Simulator: started - et 2, time is - 68
Simulator: action done - et 2, time is - 68
Simulator: action done - sto 0, time is - 200
Interrupt: Added 0.0007 time units to action eto 0
Simulator: started - eto 0, time is - 200
Simulator: action done - eto 0, time is - 200
Controller: next state is illegal
State: Starting Replaning for config 0
```

איור 19: פלט אלגוריתם ה-online עבור הפרעה הדורשת תכנון מחדש

ניתן לראות כי בזמן 33 נוצרה הפרעה לפעולת ההמראה של מטוס 0, הפרעה זו גרמה לעיכוב בתוכנית שהפך אותה ללא פיזיבלית. כתוצאה מכך התבצע תכנון מחדש וקיבלנו את קובץ הקונפיגורציה הבא עבור מצב העולם ברגע הנתון:

```
ProjectA > BT_ProA > configs > config0.1.txt
number_of_planes = 1
number_of_lanes = 2
max_run_time = 3
plane0 00 00 40 90 200 5 1
```

איור 20: קובץ קונפיגורציה חדש עבור התכנון מחדש

בעקבות קובץ קונפיגורציה זה, אלגוריתם ה-offline חישב תוכנית סדר-יום חדשה:

```
ProjectA > BT_ProA > logs > log_output_0.1.txt
t_sm_0    : 0.0       :    parents:  []          :    childs:  ['t_em_0']
t_em_0    : 40.0      :    parents:  ['t_sm_0']  :    childs:  ['t_sl_0']
t_sl_0    : 40.001    :    parents:  ['t_em_0']  :    childs:  ['t_el_0']
t_el_0    : 50.001    :    parents:  ['t_sl_0']  :    childs:  ['t_st_0']
t_st_0    : 50.002    :    parents:  ['t_el_0']  :    childs:  ['t_et_0']
t_et_0    : 60.002    :    parents:  ['t_st_0']  :    childs:  []
```

איור 21: פלט ה-offline עבור קובץ הקונפיגורציה החדש

פלט אלגוריתם ה-online עבור תוכנית סדר-היום החדשה:

```
ProjectA > HY_ProA > Logs > log_output0.1.txt
Simulator: started - sm 0, time is - 0
Simulator: action done - sm 0, time is - 40
Interrupt: Added 0.0009 time units to action em 0
Simulator: started - em 0, time is - 40
Simulator: action done - em 0, time is - 40
Simulator: started - sl 0, time is - 40
Simulator: action done - sl 0, time is - 50
Simulator: started - el 0, time is - 50
Simulator: action done - el 0, time is - 50
Interrupt: Added 7.9249 time units to action st 0
Simulator: started - st 0, time is - 50
Simulator: action done - st 0, time is - 68
Interrupt: Added 0.0005 time units to action et 0
Simulator: started - et 0, time is - 68
Simulator: action done - et 0, time is - 68
```

איור 22: פלט ה-online עבור תוכנית סדר-היום החדשה

ניתן לראות כי התכנית הסתיימה בהצלחה ללא תכנון מחדש נוסף, אך עם הפרעות נוספות שנוצרו במהלך הריצה.

9 רשימת מקורות

1. Offline Algorithm:
https://drive.google.com/file/d/1vAdCvLn5NhJQlFqSfsDZVQqb6amQi6PS/view

2. Path finding algorithms:
https://medium.com/omarelgabrys-blog/path-finding-algorithms-f65a8902eb40

3. A* Algorithm:
https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2
https://en.wikipedia.org/wiki/A*_search_algorithm
https://www.youtube.com/watch?v=g024lzsknDo

4. Linear programming problems:
https://eur01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fmedium.com%2F%40geekrodion%2Flinear-programming-introduction-e0547f3db30d&data=02%7C01%7Cyoavneta1%40campus.technion.ac.il%7C9a8a708d00a44179a5c708d85ad61e8a%7Cf1502c4cee2e411c9715c855f6753b84%7C1%7C0%7C637359222967797894&sdata=aQZNrzDxEgMqZxG5NWcu1cNuONbHEK8gF00D%2Fp36PbI%3D&reserved=0

5. Causal Graph Planning
https://docs.google.com/presentation/d/12_sedRHMg1G81Usm1MppfbiooQAHPEy5vnC-HyIeiR4/edit#slide=id.p42
https://docs.google.com/presentation/d/12_sedRHMg1G81Usm1MppfbiooQAHPEy5vnC-HyIeiR4/edit#slide=id.p42

6. Simple Temporal Networks
https://docs.google.com/presentation/d/1uVhc5xsp25RTV2C8ZJdHDnwQQVCHT46zqKU6Rb9cJE0/edit#slide=id.g221c7f3818_1_0

7. Temporal planning :
https://icaps20subpages.icaps-conference.org/wp-content/uploads/2020/10/Temporal-Planning.pdf
https://www.youtube.com/watch?v=zhbh-NtWMPE&feature=youtu.be

8. Object oriented programming in python:
https://www.youtube.com/watch?v=JeznW_7DlB0
9. Python singleton:
https://www.tutorialspoint.com/python_design_patterns/python_design_patterns_singleton.htm

10. Priority queue in python:
https://docs.python.org/3/library/heapq.html
https://www.geeksforgeeks.org/binary-heap /
https://www.geeksforgeeks.org/heap-queue-or-heapq-in-python /
11. Observer in python:
https://www.geeksforgeeks.org/observer-method-python-design-patterns /