

# AI FINAL PROJECT

SARF – Smart AI Route Finder

# הצגת הבעיה

## תיאור הבעיה

ניתן לחלק את התיירות העולמית לשני אופנים - תיירות מודרכת ותיירות עצמאית. אלו הבוחרים במדריך הילווה אותם במהלך החופשה זוכים להדרכה וארגון מההמראה ועד לנחיתה בחזרה בבית תוך בחירת האתרים הטובים ביותר ביעד וניצול מיטבי של הזמן.

לעומתם, הבוחרים בתיירות עצמאית נאלצים לבנות בעצמם את החופשה: תכנון מסלול הטיול, בתי המלון הרבים והאתרים אותם יראו ויבקרו במהלך שהותם בחופשה. בעבר היו אל הספרים ומדריכי הטיולים הכתובים שעזרו במלאכה זו. מאז פרץ האינטרנט לחיינו, המידע זמין יותר מאי פעם וכעת נולדה בעיה חדשה - אינפלציית המידע. לכל אתר בו נרצה לבקר קיימים מאות פרטי מידע אודותיו, החל מהאתר הרשמי ועד לבלוגים וביקורות על ידי גולשים. אתרים שונים כגון TripAdvisor ודומיו מכילים רשימות ארוכות של אתרים מומלצים ודירוגים שונים.

טיול במדינה זרה הינו עסק יקר, יום טיול במדינה זרה מתמחר בין 50 יורו ל 100 יורו לאדם. זאת הסיבה שהתיירים העצמאיים המתכננים בעצמם את הטיול מנסים להגיע למסלול אופטימלי, כזה המביא את הניצולת של זמנם היקר בחו"ל למקסימום.

על אף מקורות המידע האיכותיים, עדיין המעוניינים לצאת לחופש או טיול במדינה זרה מוצאים את עצמם טובעים באתרים, ספרים ושאר מקורות מידע בנסיון לבנות את המסלול האולטימטיבי. מסקר קצר שביצענו בקרב סטודנטים ומשפחותיהם נראה כי בתכנון טיול מושקעים בתכנון כיומיים עבור כל יום טיול. רבים ציינו שהסיבה לכך היא הנסיון לאסוף את המידע בצורה מיטבית ממקורות המידע השונים והרכבתם לכדי מסלול קוהרנטי.

אופטימלי הוא מושג מתעתע - נשים לב שבעת טיול במדינה זרה, כל אדם יעדיף מסלול טיול שונה. הדוגמא הקלאסית היא טיול "בטן-גב" באתר נופש לעומת טיול אקסטרים העובר בין אתרים שונים. במערכת זו הגדרנו שהאופטימלי הינו האמצע בין אלו - מסלול בו נוסעים בין אתרים שונים במסלול, ונתנו עדיפות לאתרים מסוגים מסוימים. ניתן לשנות זאת בעזרת הגדרות בקוד האפליקציה (כגון משקלי יוריסטיקות ציונים, שאליות אל מול מקור הנתונים ועוד), אך ברור כי לצורך בחינת האלגוריתמים והיכולת של AI לתת מענה לבעיה מספיק לבחור אופי מסויים ולבצע בדיקה אל מול אופטימליות זו.

בהנתן הגדרה של אופטימליות, מציאת המסלול הטוב ביותר הינה בעיה קשה לפתרון שכן היא דורשת התחשבות במספר רב של גורמים המתחלקים לאיכות הנקודות בהם מבקרים, ואיכות המסלול. מעבר ובדיקה של כלל האפשרויות וכלל המסלולים האפשריים הינו קשה ולא אפקטיבי. אלגורמי AI של Local Search מאפשרים לנו לנסות לפתור את הבעיה בצורה אפקטיבית יותר.

# הפתרון המוצע

## קווים לדמותו של הפתרון

### SARF ~ Smart AI Route Finder

הפתרון שפיתחנו מאפשר מציאת מסלול אופטימלי, או קרוב לאופטימלי בהתבסס על מקורות מידע המכילים מידע גיאוגרפי על כלל האתרים בעולם. בעזרת אלגוריתמי AI המבצעים Local Search ובהתבסס על המידע הקיים במקורות אנו בונים מסלול טיול מוצא עבור כל יעד ומשך זמן.

בעת הזנת מיקום יציאה לטיול, מיקום סיום ומשך זמן הטיול האלגוריתמים ירצו לחישוב המסלול המתאים ביותר לפרמטרים תוך התחשבות ביוריסטיקות ציון שונות.

## בחירות טכנולוגיות

בחרנו לממש את הפתרון בשפת JavaScript מעל שרת Web זאת על מנת לאפשר עבודה נוחה אל מול שירותי ה API של Google וכן לאפשר הצגה נוחה של מסלולים ותוצאות בעזרת רכיב מפה של Google. השימוש ב API של Google ושימוש בשפת Javascript דורש עבודה אסינכרונית. מכך שכלל האלגוריתמים נכתבו צורה אסינכרונית והותאמו לעבודה שכזו.

## גישות לפתרון

כדי לפתור את הבעיה, התייחסנו אליה כבעיית חיפוש מקומי.

ייצגנו קודקוד (Node) בתהליך החיפוש באופן הבא:

```
function Node(originalTime, time, pois, distances)
  this.pois = pois;
  this.distances = distances;
  this.originalTime = originalTime;
  this.timeRemainingHours = time;
  this.scoreFunc = null;
```

כל קודקוד מכיל את רשימת כל נקודות הציון בהם הוא עובר, לפי סדר מעבר. בהכרח מתקיים שנקודת הציון הראשונה היא תמיד נקודת המוצא של הטיול והנקודה האחרונה היא נקודת סיום הטיול.

בנוסף, לנוחות המימוש, כל קודקוד גם מכיל מערך כל המרחקים בין כל שני נקודות ציון, זמן החיפוש הכולל שאפשרי בטיול, זמן שנותר לטיול ופונקציית המשקל שמתעדכנת דינאמית תוך כדי ריצה.

במהלך מציאת הפיתרון, ישנם מספר הנחות שנלקחות בחשבון:

- ביממה של טיול ישנם 9 שעות אפקטיביות.
- בניית המסלול מתייחסת לזמן הטיול כרציף. למשל 3 ימים ועוד 6 שעות שקול ל 33 שעות רצופות של טיול, ללא התחייסות ללינה.
- אין התייחסות לשעות פתיחה\סגירה של אתרים.
- בכל נקודת ציון מבלים שעתיים, ללא התייחסות לסוג נקודת הציון - הנחה זו לוקחת בחשבון שזהו הממוצע הכללי.
- אין התייחסות לגבולות בין מדינות, כלומר ייתכן והמסלול ימליץ לבקר בנקודת ציון במדינה שכנה.

### מתן ציון למסלול ע"ב פונקציות משקל

הגדרנו את פונקציית המשקל של קודקוד נתון כשקלול משקלים שונים:

```
{weight: 1, scoreFunc: new RatingScore()};  
{weight: 0.4, scoreFunc: new TypesScore()};  
{weight: 0.5, scoreFunc: new PhotosScore()};  
{weight: -0.7, scoreFunc: new LongestDistanceScore()};  
{weight: -0.4, scoreFunc: new CloseToEnd()};  
{weight: 0.1, scoreFunc: new SimpsonsDiversityScore()};  
{weight: -0.1, scoreFunc: new ClusterFactor()};
```

בכל ריצה מתבצעת הערכה על כלל המסלול כולו על פי פונקציות אלו. לכל פונקציה שונה נתנו מקדם כדי לנרמל את גודל המשקלים ביחס למשקלים האחרים.

פונקציות המשקל מתחלקות לשתי קבוצות:

- הפונקציות Rating, Types, Photos, Simpson's ממשקלות את טיב נקודות הציון בלבד.
- הפונקציות Longest, Shortest, Cluster ממשקלות את איכות כל המסלול, תוך התעלמות מטיב נקודות הציון.

קבוצה	שם	תיאור
איכות האתרים בהם מבקרים	Rating Score	סכום כל הרייטינגים של נקודות הציון במסלול
	Types Score	כמה הטיול מגוון, על פי הטיוגים של הנקודות
	Photos Score	כמה תמונות יש לכל נקודות הציון
	Simpson's Diversity Score	מתוך קבוצת הסוגים של נקודות הציון, כמה הפיזור הוא אחיד. יותר אחיד, טוב יותר. <a href="https://en.wikipedia.org/wiki/Diversity_index#Simpson_index">https://en.wikipedia.org/wiki/Diversity_index#Simpson_index</a>
איכות המסלול בין האתרים	Longest Distance Score	המרחק הרחוק ביותר בין שתי נקודות ציון בטיול. ככל שקצר יותר, טוב יותר.
	Shortest Distance Score	המרחק בין נקודת הציון הלפני אחרונה לבין נקודת הסיום. ככל שקצר יותר, טוב יותר.
	Cluster Factor	מספר וגודל האשכולות (clusters) בטיול. יותר אשכולות \ אשכולות גדולים יותר זה פחות טוב.

# אלגוריתמי AI במימוש הפתרון

## HILL CLIMBING

בשיטה זו, קודקוד ההתחלה מכיל את נקודות ההתחלה והסיום בלבד.

בכל איטרציה, באופן אקראי מגרילים את אחד הנקודות שכבר קיימות בטיול, ומחפשים נקודת ציון חדשה שניתן להוסיף לטיול. מתוך קבוצת נקודות הציון שה API מספק לנו, בוחרים את הנקודה שתמקסם את הציון של טיול באותו רגע ועוברים לאיטרציה הבאה.

סוף החיפוש הוא כאשר מתוך כל הנקודות שקיבלנו בחיפוש, לא קיימת שום נקודה שנוכל להוסיף מבלי לחרוג מהזמן שמוקצה לטיול.

פונקציית השכנות - בחיפוש זה, בהינתן קודקוד כלשהו עם  $k$  נקודות ציון, נסמן את קבוצת נקודות הציון שלו להיות:

$$[start, p_1, \dots, p_{k-2}, end]$$

שכן שלו יהיה כל קודקוד עם  $1+k$  נקודות ציון מהצורה:

$$[start, p_1, \dots, p_i, p_{new}, p_{i+1}, \dots, p_{k-2}, end]$$

כאשר  $P_{new}$  היא נקודת ציון זרה לקודקוד המקורי.

כאמור, מאופי אפיון הבעיה, תמיד נעדיף טיול עם יותר נקודות ציון מאשר פחות נקודות ציון (כי נרצה למקסם את תועלת הטיול), לכן אין צורך בהגדרת שכן עם  $k-1$  נקודות ציון.

כדי לשמור על סדר הגיוני של נקודות הציון, בזמן בקשת נקודות ציון חדשות עבור יצירת שכן, נגביל את רדיוס החיפוש להיות תלוי בזמן הטיול הנותר ותלוי בנקודות הציון שכבר קיימות בטיול.

## GENETIC ALGORITHM

בשיטה זו, האוכלוסיה ההתחלתית של החיפוש היא שני קודקודים התחלתיים (זהים לחלוטין אחד לשני, ולקודקוד ההתחלתי של חיפוש ה hill climbing) המציינים מסלולים המכילים שתי נקודות בלבד, התחלה וסיום.

שיטת הריצה היא לפי ההגדרה של אלגוריתם גנטי כפי שהוגדר בכיתה, עם אימוץ הגדרות המתאימות לבעיה זו.

## כשירות (FITNESS)

שימוש ישיר בפונקציית המשקל שהגדרנו למעלה המשכללת את כלל היוריסטיקות.

## מוטציה

$$\left(1 - \frac{\text{Current Generation}}{\text{Max Generation}}\right)^2 \text{ היא } \text{ההסתברות למוטציה}$$

מוטציה של קודקוד באה לידי ביטוי בהוספת נקודת ציון אקראית במקום אקראי ברשימת נקודות הציון הנוכחיות של הקודקוד. ההשלכה של שימוש בסתברות מעריכית היא שבתחילת הריצה, האלגוריתם מנסה להוסיף כמה שיותר נקודות ציון, ולקראת סיום הריצה הוא מנסה למצוא את הסדר הטוב ביותר של הריצה שממקסם את הציון הכולל.

#### בחירה (SELECTION)

כאשר נרצה לזווג זוג קודקודים כדי לייצר דור חדש, מתוך כל הדור נוכחי, נבחר באופן הסתברותי זוג הורים כך שקודקוד בעל ציון גבוה יותר, זוכה להסתברות גבוהה יותר להיבחר.

#### זיווג (REPRODUCTION)

בהיתן זוג הורים, נייצר את קבוצת הילדים שלהם להיות בחירה אקראית של כל נקודות הציון של שני ההורים ביחד, בסדר אקראי. גודל הדור החדש יהיה פי 1.5 מגודל הדור הקודם עד למקסימום הקבוע בקונפיגורציה.

#### הערות

- גודל האוכלוסיה בכל דור מוגבל ל 50 פריטים ע"פ קונפיגורציה.
- מספר הדורות המקסימלי הוא כמספר השעות בטיול. עקב ההנחה בא נבלה בכל נקודת ציון שעתיים, גידול מספר הדורות מעבר למספר זה היווה שינוי קטן.

#### רנדומליות

עקב הבחירה הרנדומית לגבי סביב איזו נקודה לחפש כעת שתי ריצות בעלות פרמטרים זהים עלולות לתת תוצאה שונה כמעט לחלוטין. הרנדומליות הינה חלק ממימוש האלגוריתם, והמשתמש רשאי להריץ מספר פעמים אם יראה לנכון. כיוון שמדובר במציאת מסלול טיול הרי שהמסלול הינו בגדר המלצה בלבד, וייתכן שהמשתמש ייקח אותו כבסיס לשינויים, על כן ריצה נוספת המאפשרת תוצאות שונות במקצת אינה דבר פסול.

# מקורות מידע בשימוש

## APIS

לצורך העבודה השתמשנו במספר APIs של Google Maps המכילים את המידע אודות המקומות השונים.

**Google Places API** - מקור המידע המרכזי המכיל נקודות עניין ברחבי העולם. המקור מאפשר לתשאל אותו על ידי מקום, רדיוס, וסוגי מקומות ולקבל את כלל המקומות העונים על התנאים. עבור כל מקום נקבל מידע בסיסי אודותיו כגון שם, מיקום, רייטינג (לפי דרוג משתמשים), תמונה ועוד...

בנוסף ה-API מאפשר לבצע Autocomplete על מקומות המוקלדים על ידי המשתמש ועל ידי כך לשפר את חווית השימוש באפליקציה.

**Google Direction API** - מאפשר לחשב מסלול בין נקודות (עד 8 נקודות ביניים) ולהציג אותו על מפה.

**Google Geocode Library** - חבילה זו מאפשרת חישוב מרחק אווירי בין שתי נקודות וכן פעולות שונות על אובייקטי גיאוגרפיה. השימוש בספרייה זו מאפשר חישוב מהיר של מרחק בין נקודות והערכה של זמן הנסיעה בין אחד לשני.

## הגבלות

יש לציין שהשימוש במקורות המידע של Google היה נוח, ברור ואפקטיבי מאוד. כמות המידע הנגיש היא עצומה, ומאפשרת הגעה למסלולים ברמה גבוהה. יחד עם זאת ישנם מספר מגבלות, מגבלות אלו לא שינו את אופן המימוש שתכננו מראש, ולא גרמו לקושי במימוש האלגוריתמים.

1. הגבלת כמות השאילתות בשניה - בשימוש חנימי אנו מוגבלים ב-5 שאילתות לשניה, דבר הגורם לנו להמתין מספר מילישניות בין כל שאילתה על מנת לא לעבור את מגבלה זו. הדבר מתבטא אך רק בעיכוב בהרצת האלגוריתם.
2. איכות הנקודות המתקבלות - על אף היותו של Google Places מקור מהימן ומלא, הוא אינו מיועד לתיירות בלבד ואינו מכוון באופן מלא לתחום זה. אנו מאמינים כי שימוש עתידי במקור ייעודי כגון TripAdvisor ייתן נקודות איכותיות יותר. הסיבה היא איכות סקירות המשתמשים, הדירוגים הניתנים על ידם וכו'. הקוד נכתב כך שהחלפת מקור המידע במקור אחר הינה פעולה קלה ותלויה אך ורק בזמינות של אותו מקור. (לצערנו, לא יכולנו להשיג נגישות ל-API של TripAdvisor)



# הוראות הפעלה

## קימפול

על מנת להריץ את האפליקציה לא נדרש לקמפל דבר. דרוש חיבור אינטרנטי עובד.

האפליקציה נבדקה על דפדפן Chrome גרסא 49.0.02. על אף שלא אמורה להיות בעיה להפעיל את האפליקציה על כל דפדפן מודרני התומך ב JavaScript ו CSS, מומלץ להריץ את Chrome, ובמידה וישנן בעיות בהרצה או בתצוגה מומלץ לעדכן לגרסא העדכנית ביותר, או להעביר לגרסא 49.

## הרצה

כדי להריץ את הפרוייקט, יש להפעיל את קובץ index.html הנמצא בתקיית הפרוייקט. נכון לימן כתיבת הפרוייקט, הוא גם מאוחסן בכתובת <http://www.rotenberg.co.il/AI/>: [הבאה](#)

לאחר פתיחת הקובץ או גלישה לכתובת יעלה החלון הבא:

The screenshot shows the SARF web application interface. At the top, there is a navigation bar with links for SARF, Planner, and About. The main content area is divided into several sections:

- Route Parameters:** This section contains dropdown menus for 'Start Location' (set to Tel Aviv-Yafo, Israel) and 'End Location' (set to Jerusalem, Israel). It also has input fields for 'Days' (set to 1) and 'Hours' (set to 0), and a dropdown for 'Algorithm' (set to Genetic Search). A 'Find Best Route!' button is located at the bottom of this section.
- Route Information:** This section contains a table with two columns: 'Func' and 'Score'.
- Map:** A Google Map showing the route from Tel Aviv-Yafo to Jerusalem. The map includes various road numbers and city names.
- Recommended POIs:** A list of recommended points of interest.
- Log:** A text area for logging.

## תחילת החיפוש

כדי למצוא את המסלול האופטימלי, יש לבחור מקום סיום והתחלה בעזרת אלגוריתם AutoComplete או בעזרת תפריט dropdown של מקומות שכחים, בחירת מספר ימי הטיול (יתורגם ל 9 שעות טיול

לכל יום) ותוספת מספר שעות. לבסוף יש לבחור מתוך אלגוריתמי החיפוש, גנטי או Hill Climbing, וללחוץ על כפתור החיפוש.

## מהלך החיפוש

התנהגות הממשק תהיה שונה בהתאם לאלגוריתם הנבחר:

במידה ונבחר באלגוריתם Hill Climbing, כאשר תתווספנה נקודות למסלול, יקרו מספר שינויים:

בחלון המפה (המרכזי), יתווסף סמן לנקודה החדשה והכיתוב על הסמנים יתעדכן לשקף את הסדר החדש (לפי סדר ה A-Z). בחלון Recommended POIs יתווספו צעדי החיפוש, שיתעדכנו לפי הסדר. בנוסף, חלון ה Route Information יציג את ציוני המסלול תוך כדי החיפוש. ניתן לראות את אופן פעולת האלגוריתם על ידי העדכון העיתי.

במידה ונבחר באלגוריתם גנטי, כתוצאה מהתנהגות האלגוריתם, חלונות המפה ורשימת נקודות הציון לא יתעדכנו עד סוף הריצה וזאת כי אין מחויבות לנקודות מסוימות עד לסיום החיפוש.

בשני המקרים, חלון ה Log יתעדכן עם מידע רלוונטי בנוגע לתהליך החיפוש.

בראש החלון ניתן לראות Progress Bar המציין את משך תהליך החיפוש וכמה זמן מוערך נשאר עוד.

## סיום החיפוש והצגת התוצאות

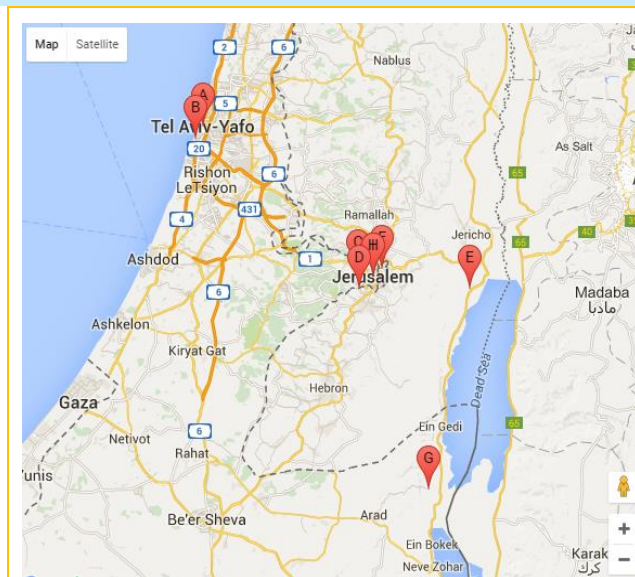
בסיום הריצה, חלון ה Recommended POIs יכיל את כל נקודות הציון של המסלול, לפי סדר הביקור בהן. חלון המפה יכיל את כל הסמנים של המסלול וחלון Route Information יאוכלס עם פירוט הציונים השונים של המסלול, לפי פונקציית המשקל שהגדרנו. ניתן להתחיל חיפוש חדש.

# ניתוח תוצאות ריצת האלגוריתמים

## בניית הציון לפי יוריסטיקות

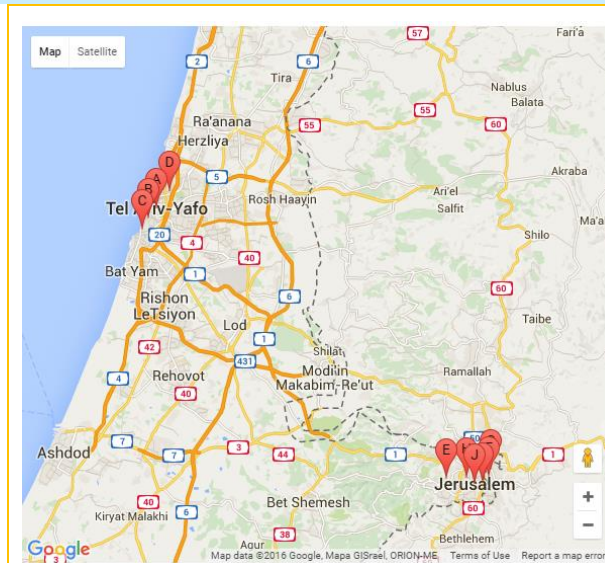
בתהליך הבניה של ציון עבור מסלול משתתפות מספר יוריסטיקות כפי שהוצג מקודם. על מנת להבין את החשיבות שלהן בקביעת המקסימום נסתכל על ריצות בהם הסרנו יוריסטיקות ספציפיות ועל ידי כך הסרנו את התתייחסות לפרמטרים אלו בבניית המסלול. הריצות נעשו בעזרת אלגוריתם Hill Climbing.

### תל אביב ירושלים – יומיים – עם כלל היוריסטיקות



- |     |                                       |
|-----|---------------------------------------|
| 4.7 | מזיאון בתי האוסף לתולדות צה"ל         |
| 4.8 | Rockefeller Archeological Museum      |
| 4.6 | Yad Vashem                            |
| 3.6 | עין יעל                               |
| 4.7 | גן לאומי קומראן - Qumran              |
| 4.2 | L. A. Mayer Institute for Islamic Art |
| 4.7 | Masada National Park                  |

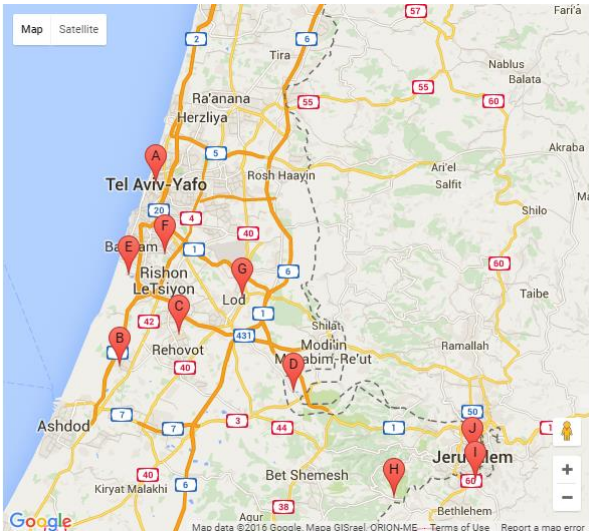
### תל אביב ירושלים – יומיים – ללא CLUSTERING



- |     |                                  |
|-----|----------------------------------|
| 4.7 | The Temple Institute             |
| 4.5 | Beit Ha'ir                       |
| 4.7 | מוזיאון בתי האוסף לתולדות צה"ל   |
| 4.8 | Palmach Museum                   |
| 4.6 | Yad Vashem                       |
| 4.8 | Rockefeller Archeological Museum |
| 4.7 | Menachem Begin Heritage Center   |
| 4.6 | Bible Lands Museum               |

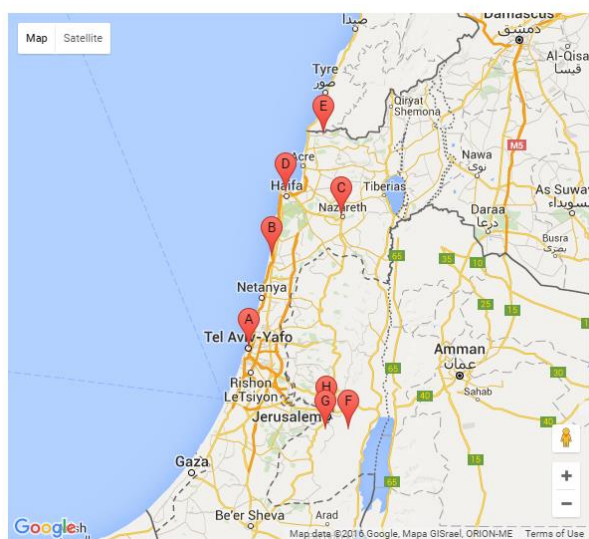
## תל אביב ירושלים – יומיים – ללא RAITING

ניתן לראות שהמסלול מתחשב בשאר הפרמטרים המשפיעים על איכות המסלול ואנחנו מקבלים מסלול מגוון, במספר מוקדים שונים ללא צבירים כלל. יחד עם זאת, אנחנו מקבלים נקודות באיכות נמוכות באופן יחסי, וכאלו ללא דירוג כלל.

	3.6	Mini Israel
	N/A	Waterpark
	4.8	גן המדע ע"ש קלור
	N/A	ספדי כיף
	4.3	סופרלנד
	4.1	Yamit 2000
	N/A	כנסיית ג'ורג' הקדוש קוטל הדרך
	N/A	חוות אייל

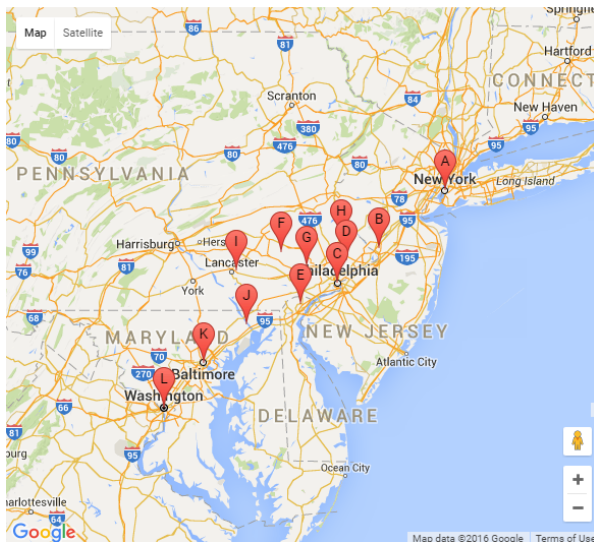
## תל אביב ירושלים – יומיים – ללא LONGEST DISTANCE

על כך נאמר – לנסוע לירושלים דרך חיפה. יחד עם זאת ניתן לשים לב לאיכות הנקודות שהתקבלו, וכן לכמות נמוכה יותר של אתרים – זאת מכיוון שזמן רב מתבזבז בנסיעה מאשר בביקור באתרים.

	4.9	מוזיאוני ראלי, קיסריה
	4.6	Nazareth Village
	4.9	Bahai Gardens in Haifa
	4.9	Keshet Cave
	4.6	Church of Nativity
	4.8	Mar Saba Monastery

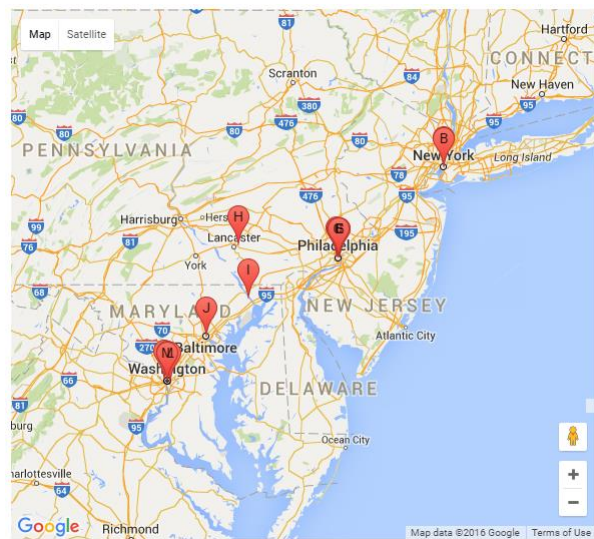


ניו יורק – וושינגטון – 3 ימים – עם כלל הוורסטיקות

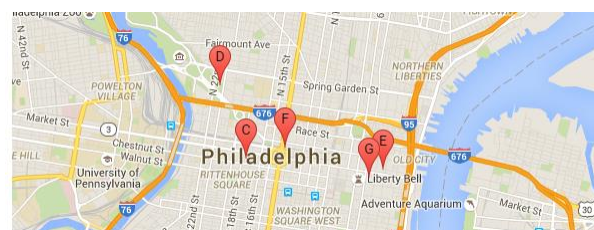


4.2	<b>Discover Lancaster</b>
4.7	<b>Grounds For Sculpture</b>
4.5	<b>Visit Philadelphia</b>
4.6	<b>Glencairn Museum</b>
4.6	<b>Nemours Mansion and Gardens</b>
4.6	<b>Hopewell Furnace National Historic Site</b>
4.8	<b>Wharton Esherick Museum</b>
4.6	<b>Mercer Museum</b>
4.1	<b>Ripley's Believe It or Not</b>
4.7	<b>Steppingstone Farm Museum</b>

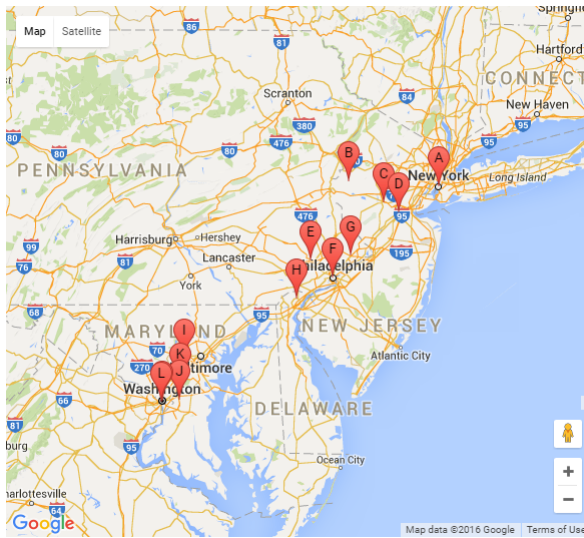
ניו יורק – וושינגטון – 3 ימים – ללא CLUSTERING



4.2	<b>Discover Lancaster</b>
4.8	<b>Museum of American Finance</b>
4.5	<b>Visit Philadelphia</b>
4.8	<b>Rodin Museum</b>
5	<b>Free Tours by Foot - Philadelphia Tours</b>
4.9	<b>Masonic Temple</b>
3.8	<b>Philadelphia Sightseeing Tours Inc.</b>
2.6	<b>Big Bus Washington DC</b>
4.7	<b>Steppingstone Farm Museum</b>
4.8	<b>The Star-Spangled Banner Flag House</b>
4.8	<b>Dumbarton Oaks Museum</b>
4.6	<b>Decatur Carriage House</b>

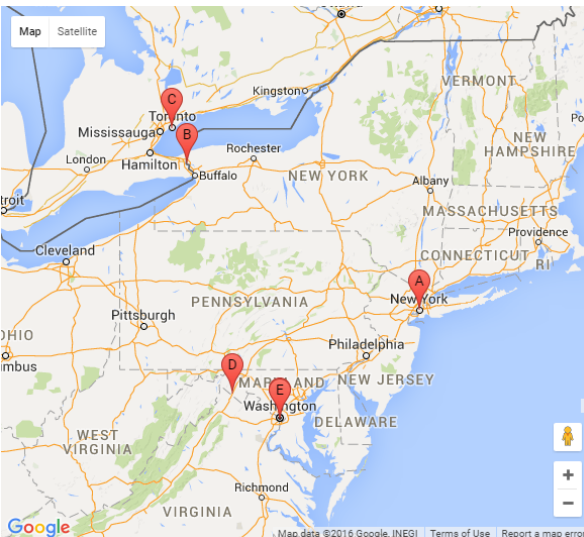


ניו יורק – וושינגטון – 3 ימים – ללא RATING



N/A	Howard County Tourism & Promotion
N/A	Warren County Convention & Visitors Bureau
N/A	Somerset County Tourism
N/A	Central Jersey Convention & Visitors Bureau
N/A	Valley Forge Tourism and Convention Board
4.5	Visit Philadelphia
N/A	Caribbean Travel Service
4.6	Nemours Mansion and Gardens
N/A	Prince George's County MD Conference & Visitors Bureau
5	City Segway Tours - Washington, DC
N/A	Montpelier Mansion

ניו יורק – וושינגטון – 3 ימים – ללא LONGEST DISTANCE



4.2	i-81 Northbound West Virginia Welcome Center
4.6	Niagara Tourism and Convention Corporation: Niagara USA Official Visitor Center
4.2	Tourism Toronto
4.6	USA Guided Tours

## השוואת GENETIC עם HILL CLIMBING

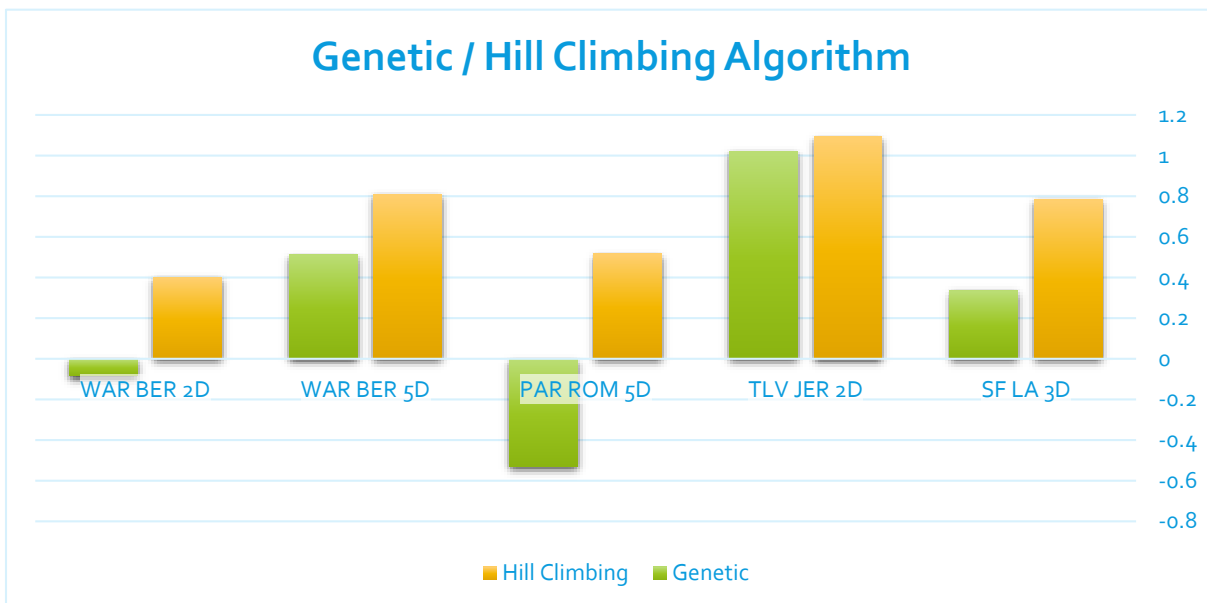
על מנת להשוות את האלגוריתמים אחד אל מול השני נשאיר את פונקציית הציון זהה, ונבצע ריצות שונות ונשווה אותן. נבצע מספר ריצות באורכים שונים ובמדינות שונות, וברדיוס שונה כך שמזכה לתוצאות מגוונות אשר ייבחנו את האלגוריתמים בסביבה רלוונטית.

### הריצות שבוצעו

מקור	יעד	משך זמן
סן פרנסיסקו	לוס אנג'לס	3 ימים
תל אביב	ירושלים	2 ימים
פריז	רומא	5 ימים
וורשה	ברלין	5 ימים
וורשה	ברלין	2 ימים

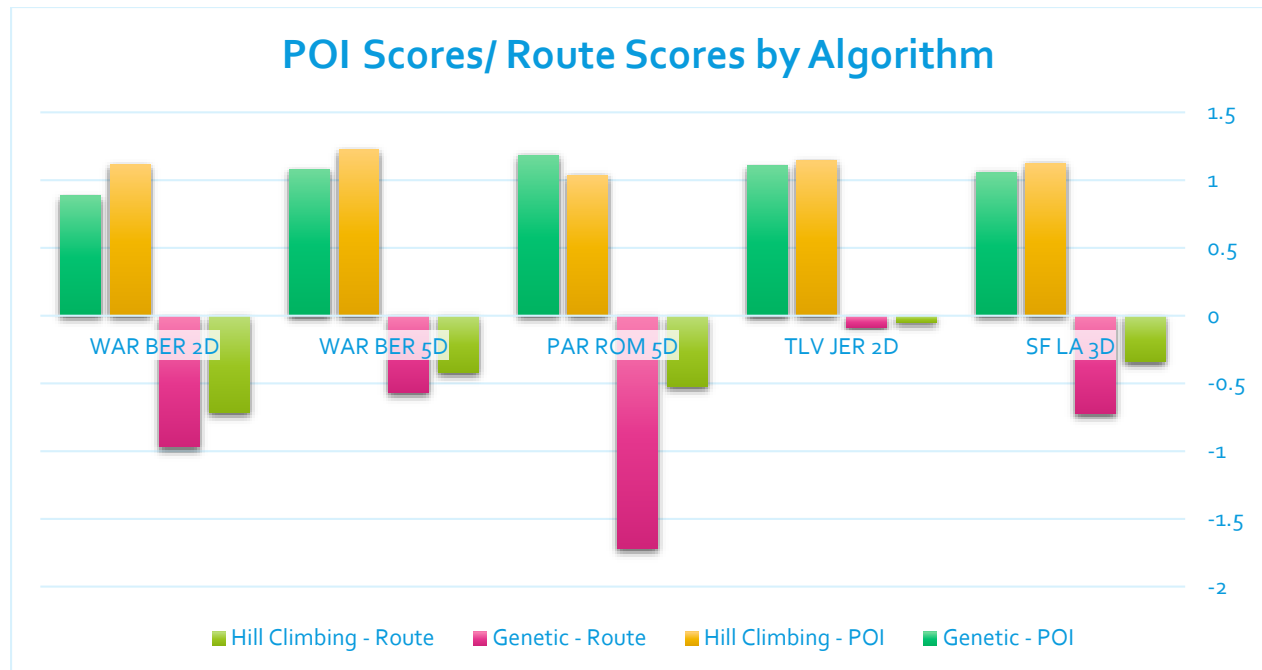
לכל שורה בוצעה ריצה של אלגוריתם גנטי, וריצה של Hill Climbing כאשר כל הסביבה עברה איפוס לפני הריצה על מנת להבטיח את אמינות התוצאות.

### ציון כולל



הריצה מתוארת בפורמט D <Start City> <End City> <Number Of Days>

נשים לב שבכל הריצות אלגוריתם Hill Climbing מנצח בהשגת מסלול עם תוצאה גבוהה יותר. התוצאות בחלק מהמקרים שליליות, אך אין לכך חשיבות גדולה כיוון שהתוצאות מנורמלות כקבוצה.



ננסה להבין את ההבדל בתוצאות על ידי פירוק התוצאות לשתי קבוצות היורסטיקות למתן הציון. הציון הכולל מורכב מחיבור של התוצאות של כלל היורסטיקות.

נוכל לשים לב שבאופן כמעט מוחלט התוצאות של האלגוריתם הגנטי בראי איכות הנקודות הנבחרות זהה כמעט לאיכות הנקודות הנבחרות באלגוריתם ה Hill Climbing. כלומר, אין ייתרון לאלגוריתם זה או אחר ברמת איכות הנקודות המומלצות.

לעומת זאת, ניתן לראות את ההבדל המשמעותי ביותר בטיפול במסלול כולו ובחירת מסלול העונה על הקריטריונים שהוצבו. בכל הריצות קיבלנו תוצאה ברורה של עדיפות אלגוריתם Hill Climbing על פני האלגוריתם הגנטי. ניתן להסביר זאת עקב כך שמספר האופציות עבור האלגוריתם הגנטי למסלולים כל כך גדולה שקשה להגיע למסלול אופטימלי, גם לאחר מספר דורות גדול מאוד, וזאת לעומת אופן הפעולה של אלגוריתם Hill Climbing בה הבחירה של הנקודות מבתצעת מראש כך שיספקו גם את הציון של אופי המסלול.

בחירת הנקודות מכילה מספר נקודות קטן ובחירתם לא תלוי בסדר, על כן מספר הסידורים אינו פונקציה והבחירה של נקודות באיכות גבוהה מתאפשרת בצורה מהירה יחסית גם באלגוריתם הגנטי.

במהלך הנסיונות שביצענו הרצנו את האלגוריתם הגנטי במספר רב של דורות, אך הציון לא השתפר בצורה משמעותית אם בכלל. ייתכן כי במידה והגדרת האופטימליות הייתה שונה ולא הייתה כוללת חלק מהיוריסטיקות היינו מקבלים שהציון דומה בשונה ממה שמוצג כאן. יחד עם זאת, ברור שבדוגמא זו ביצועיו של האלגוריתם הגנטי נמוכים באופן משמעותי מאלו של אלגוריתם Hill Climbing.



## סיכום

הרעיון לפרוייקט זה עלה עקב הרצון לטוס לטייל בסיום התואר. כאשר ניגשנו לפתרון הבעיה לא היה שום ערבון לכך שהאלגוריתמים שלמדנו הינם ברי מימוש עבור בעיה זו. האתגרים והחשש מפני תוצאות שאינן מספקות, ואינן רלוונטיות היה גדול למדי. כיום, לאחר מימוש הפתרון אנו בוחנים את המסלולים והתוצאות אל מול הטיולים המתוכננים לסיום התואר, ומודעהמים כל פעם מחדש מאיכות התוצאות שהאלגוריתמים נותנים – התוצאות רלוונטיות ביותר.

צינו כבר כי סוגיית האופטימליות הינה סובייקטיבית, וייתכן כי אחרים היו בוחרים לממש אופטימליות שונה (כלומר, פונקציות ציון ומשקל שונות). יחד עם זאת, השינויים הקטנים שביצענו במהלך העבודה הוכיחו כי שינו שכזה אינו משפיע על מימוש האלגוריתמים, כך שניתן לקנפג את האפליקציה כרצון המשתמש.

האלגוריתמים הוכחו את עצמם כעונים על הבעיה שהוצגה. ראינו שאלגוריתם Hill Climbing פועל בצורה יותר טובה להשגת מסלול אופטימלי כמו שהוגדר בפרוייקט זה.

השימוש ב API של Google מאפשר שימוש במסד הנתונים האדיר של Google המכיל מקומות מכל העולם, ומאפשר הרצת האלגוריתמים על כל מקום בעולם ובכל פרמטר.

Func	Score
Rating Score	0.770
Type Count Score	0.136
Photos Score	0.019
Longest Distance Score	-0.059
Close to End Score	-0.000
Diversity Score	0.185
Clustering Score	-0.017
Total Score	1.034

חלק ממסלול הטיול המתוכנן בסיום התואר – Yellowstone National Park & Grand Teton National Park, USA