|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | |  |
| אוניברסיטת תל-אביב | | **Tel-Aviv University** |
| הפקולטה להנדסה  בי"ס להנדסת חשמל | |  | | Faculty of Engineering  School of Electrical Engineering |
| ויזואליזציה של מרחקים ברשת | | | | |
| פרויקט מס' 10-1-1-53  ספר פרויקט | | | | |
| מבצעים: | | | | |
|  | אמיר לזבניק | | 039099783 | |
|  | אריאל חרונטמן | | 039623053 | |
| מנחים: | | | | |
|  | פרופסור בועז פת שמיר | | אוניברסיטת ת"א | |
|  | גברת נעה זילברמן | | אוניברסיטת ת"א | |
| מקום ביצוע הפרויקט:  המעבדה לתקשורת מחשבים | | | | |

**תוכן עניינים**

1 תקציר 3

2 הקדמה 3

3 רקע תיאורטי 4

4 סימולציה 6

5 מימוש 7

6 ניתוח תוצאות 15

6.1 השוואות בין תוצאות הסימולציה לעבודה בזמן אמיתי 15

6.2 ביצועי המערכת מבחינת זמן אמיתי 16

7 סיכום, מסקנות והצעות להמשך 16

מקורות 17

# תקציר

הפרוייקט מתרכז בויזואליזציה של עיוותי מרחקים בין IPs. הויזואליזציה היא הצגה של העיוותים בין המרחק הגיאוגרפי האמיתי בין שני אתרים (IPs) לעומת המרחק שנמדד ביניהם באצמעות ריצת traceroute במערכת DIMES.

הפרוייקט יוכל לשמש לצרכים מחקריים שונים, כגון – הערכת טיב תקשורת באזור מסויים, הבנת המסלול בו עובר המידע (לא יוצג מסלול מלא אך ניתן להבין אם המסלול הוא עקיף או ישיר), השוואת טיב התקשורת לאיזור גיאוגרפי מסויים בימים, שבועות או שנים מסויימות ועוד.



# הקדמה

בפרק זה יתוארו:

* מטרות הפרויקט
* המוטיבציה
* הגישה לפתרון הבעיה
* השוואה כנגד עבודות קיימות בנושא

# רקע תיאורטי

### חישוב מרחקים אמיתיים

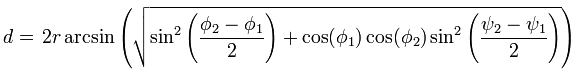
בהינתן נקודות הציון הגיאוגרפיות של האתרים, ניתן לחשב את המרחקים בין האתרים כמו מרחק בין 2 נקודות על פני שטח של כדור.

בחישוב זה השתמשנו בנוסחת "האברסין" - <http://en.wikipedia.org/wiki/Haversine_formula>



כאשר: d-המרחק הפיסי, r – רדיוס המרחק ממרכז הכדור, ושאר הזויות הן נקודות הציון אשר מסומנות בזויות.

כמו כן מתוך הנוסחה נרצה לחלץ את d המרחק הפיסי, כלומר:



אולם חישוב זה אינו פשוט משום שכדור הארץ אינו עגול, אלא אליפטי, כלומר שהרדיוס בחישוב מרחק על פני כדור הארץ הוא דינמי ומשתנה מנקודה לנקודה, לדוגמא לנקודה הנמצאת על קו המשווה מרחקה ממרכז כדוה"א הוא 6,378.1קמ (כלומר רדיוס של 6,378.1קמ)

ואילו לנקודה בקוטב מרחק ממרכז כדוה"א הוא 6,356.8קמ (כלומר רדיוס של 6,356.8קמ), כלומר ל2 הנקודות רדיוס שונה

נפתור אתגר זה ע"י שימוש ברדיוס הממוצע בין 2 הנקודות.

### חישוב מרחקים מתוך המדידות

בהינתן מדידת הזמן שלקח לאות לעבור בין 2 אתרים ניתן לחשב את המרחק "הווירטואלי" בין 2 הנקודות, ע"י שימוש בכלל האצבע 1ms=100km, כלומר אם נמדדו 10ms בין 2 אתרים, אז המרחק "הווירטואלי" בניהם הוא 1,000 ק"מ.

### הצגת העיוותים בצורה ויזואלית

את תצוגת העיוותים נראה על פני גרף תלת מימדי, שבסיסו X,Y הוא בעצם הקואורדינאטות של הנקודות במרחב, וציר Z הוא בעצם תוצאת העיוות או במילים אחרות - תוצאת העיוות היא למעשה היחס שבין המרחק הפיסי למרחק הווירטואלי תוצאה זו מביאה למקסימום ביטוי את העיוות שנוצר בין המרחקים.

ערך עיוות של 1 יעיד על מרחק ווירטואלי זהה למרחק הפיסי, כלומר אין עיוות.

ערך גדול מ1, מרחק ווירטואלי גבוה ביחס לפיסי, וככל שיהיה גבוה יותר כך העיוות יהיה גדול יותר ובולט יותר.

ערך קטן מ1, מרחק פיסי גבוה ביחס לווירטואלי (מצב שיהיה נדיר יותר כנראה) ויבטא בעצם עיוות שלילי (ויש להניח שיהיה גם מתון יותר)

כמו כן, הצגת העיוותים אינה מתחשבת בגורמים חיצוניים כמו המיקום עצמו.

אפשרויות אחרות שהועלו היו:

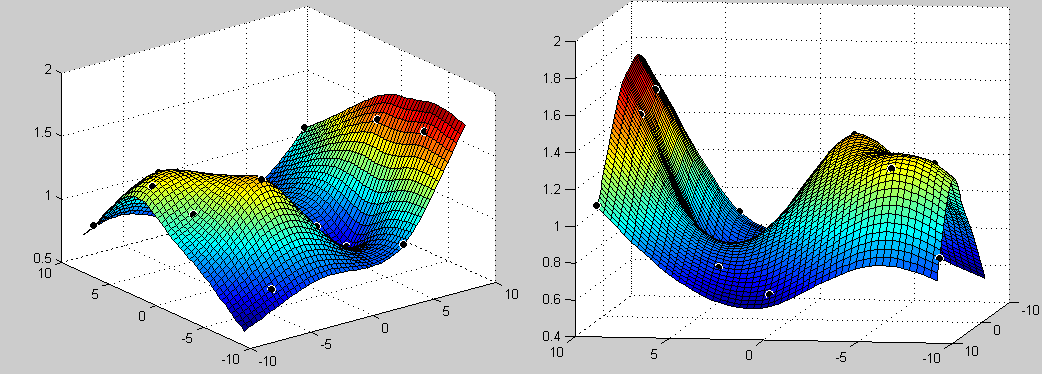
* הצגת המרחק הווירטואלי על ציר Z, וכך למעשה הגובה(Z), מתרומם בהתאם לעיוות, אולם היא נפסלה משום שככל שהמרחק בין האתרים גדל, כך הגובה גדל גם הוא, והתוצאה למעשה אינה אובייקטיבית.
* הצגת ההפרש בין המרחק הווירטואלי למרחק הפיזי, וכך למעשה הגובה משתנה ביחס לגודל העיוות, אולם גם היא נפסלה משום שככל שמרחק בין אתרים גדול יותר, כך ההפרש יכול להיות גדול יותר וגם תוצאה זו אינה אובייקטיבית.

כך למעשה גרף התוצאה נותן ביטוי אובייקטיבי ומקסימאלי לעיוות עצמו ובגלל שהתוצאה מוצגת ביחס למשטח XY שהוא למעשה משטח המיקומים הפיסיים של האתרים, ניתן לפענח אזורי עיוות שונים על פני השטח.

בעזרת פונקצית TPAPS של Matlab נוכל להסיק מאוסף נקודות המדגם אל כלל הגרף.

פונקצית TPAPS – היא פונקציה שבהינתן נקודות על משטח וערכים מתאימים להן, יכולה לחשב בעזרת שימוש באנרגיה מינימאלית בין הנקודות, את הSPLINE דרכן, כלומר במקרה שלנו, ממצעת את הגרף על כל שטחו כשלרשותה אוסף נקודות מדגם.

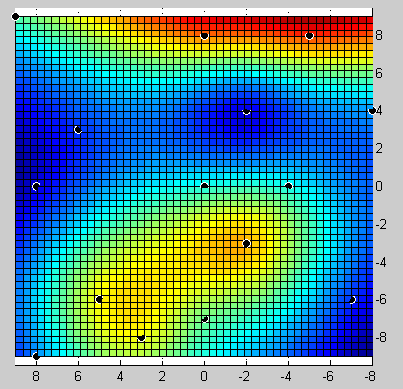
### דוגמאות לאופן בו נציג את התוצאות



איור 1 - תצוגה תלת-מימדית של הנקודות ועיוותי המרחקים, יחד עם ביטוי כרומטי של התוצאות

באיור למעלה ניתן לראות אוסף נקודות שחורות שהן נקודות המדגם, אליהן נמדד המרחק מאתר המקור הממוקם במרכז הגרף. כל הנקודות מפוזרות על פני משטח XY לפי קואורדינטות האורך והרוחב שלהן, גובה הגרף - Z הוא תוצאת היחס בין המרחק הווירטואלי למרחק האמיתי (הגיאוגרפי).

ניתן להבחין(בגרף בצד שמאל ולמטה) כי באזור החלק הקדמי והאחורי (בצבעים אדום וצהוב) ישנו עיוות גבוה, ואילו באזור הצדדים (בצבע כחול) כמעט ואין עיוות.



איור 2 - תצוגה דו-ממדית של הנקודות ועיוותי המרחקים, עם ביטוי כרומטי של התוצאות

# סימולציה

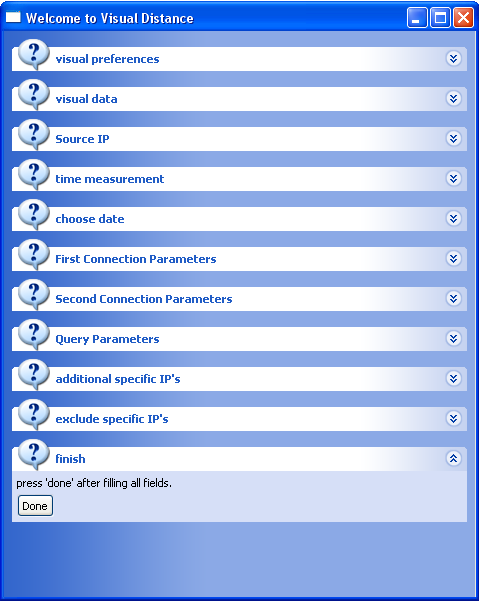
בפרק זה תתואר סביבת הסימולציה (אם רלבנטי)

# מימוש

המודול ההתחלתי של המערכת הוא הקובץ runJava.bat זהו קובץ שרץ בסביבת WINDOWS, אשר מריץ את קבצי הJAVA המקומפלים (\*.class), כך שלמעשה בלחיצה כפולה על קובץ זה ייפתח לנו ממשק המשתמש.

**GUI –**

ממשק המשתמש מומש באמצעות חבילת SWT בשפת התכנות JAVA, כאשר המסגרת הכללית היא ExpandBar המכיל סעיפים מסוג ExpandItem – הסיבה לשימוש שכזה היא שניתן במסך אחד לראות את כל האופציות ואין צורך להחליט לפני שעוברים שלב, ואז קשה לחזור אחורה, כאן המשתמש יכול לפתוח את ה ExpandItem המתאים לו ואף לחזור אליו בהמשך בהתאם להחלטות שלו בסעיפים אחרים. כמו כן, צורה זו מאפשרת חלוקה לוגית של האופציות הרבות שקיימות למשתמש, וכך קל יותר לנווט בין האפשרויות השונות.



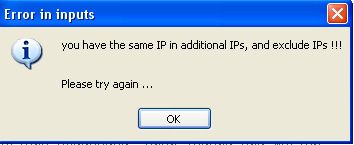
איור 4.1 – מסך הפתיחה של ממשק המשתמש

המשתמש מזין את הערכים בשדות השונים ולאחר מכן כאשר הוא לוחץ על כפתור הDONE, מאזין לכפתור מזהה את הלחיצה ומעביר את הנתונים לתוך אובייקט מיוחד מסוג DETAIL אשר קודד במיוחד כך שיכיל את שלל הנתונים הנחוצים לאגירה ולשימוש בהמשך, לאחר שהנתונים נאגרו באובייקט הDETAIL, נפתח חלון הודעה ובתוכו סיכום שלל הנתונים שהמשתמש הזין (אשר נשלפו מאובייקט הDETAIL) וכך המשתמש יכול לבחון שוב את בחירותיו ולהחליט האם אכן אלו הערכים על פי הם הוא מעוניין לקבל תוצאות, משתמש יכול לבחור OK ולהמשיך או CANCEL על מנת לעשות שינויים בנתונים אותם הזין.



איור 4.2 – חלון סיכום, עם כפתורי OK ו CANCEL

לאחר לחיצה על OK, המערכת תבצע סדרת בדיקות אשר בודקות האם יש נתונים חסרים או סתירות בערכים שהוזנו ע"י משתמש, ומתריעה על כך בהתאם בעזרת חלון התרעה, לדוגמא:



איור 4.3 – חלון המתריע על כך שישנה כתובת משותפת ברשימת ההכנסה ורשימת ההוצאה של IP לרשימה

אם המערכת לא זיהתה תקלות, היא ממשיכה אל החלק של עבודה בפועל מול ה-DB של DIMES.

**DIMES DB –**

כעת, כאשר המשתמש בחר והזין את כל בחירותיו לטופס הויזואלי, מתחיל החלק של העבודה מול ה-DB של DIMES. חלק זה מחייב חיבור ל-DB באחת משתי דרכים:

* חיבור ישיר - כאשר המחשב שמריץ את האפליקציה שלנו נמצא בתוך הרשת של DIMES. במצב זה ניתן פשוט להתחבר ל-DB באופן ישיר
* חיבור מרחוק ע"י SSH Tunnel - כאשר המחשב שמריץ את האפליקציה נמצא מחוץ לרשת של DIMES, עליו לפתוח SSH Tunnel ל-DB של DIMES על מנת לאפשר הרצת שאילתות.

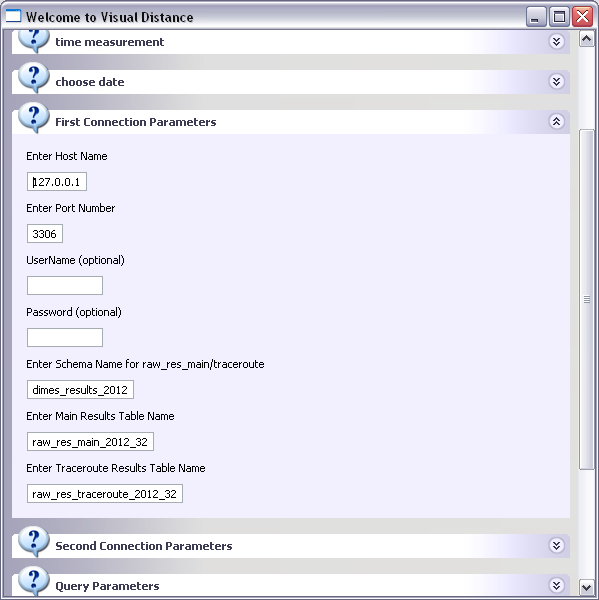
אנו זקוקים לשני חיבורים על מנת להחזיר את המידע המבוקש. חיבור אחד ישמש אותנו לעבודה מול השרת שמכיל את טבלאות תוצאות הניסויים (טבלה כזו לדוגמא היא raw\_res\_main\_2012\_28). החיבור השני ישמש אותנו על מנת להתחבר לשרת שמכיל את טבלת הקישור בין IP לבין נקודת הציון שלו על הגלובוס (בדרך כלל מדובר בטבלה IPsTblFull, אך ניתן לבחור טבלה אחרת במקומה אם רוצים, ויש לספק את פרטי הסכמה ושם הטבלה). בנקודות הציון על הגלובוס, נעשה שימוש אחר-כך לצורך חישוב המרחק הגיאוגרפי (האמיתי) בין שתי נקודות.

את החיבור עצמו אנו משיגים באמצעות שימוש בממשק MySQL שמסופק ע"י קובץ JAR שניתן להוריד בחינם מהרשת (הגרסה בה השתמשנו היא mysql-connector-java-5.1.21-bin.jar, אותה ניתן להשיג בכתובת האינטרנט - <http://www.mysql.com/downloads/connector/j/>, והיא כמובן כלולה בתיקיות האפליקציה המסופקות למשתמש הקצה. קובץ JAR זה מספק לנו ממשק ליצירת חיבור ל-DB של DIMES, ממשק להרצת שאילתות ועוד.

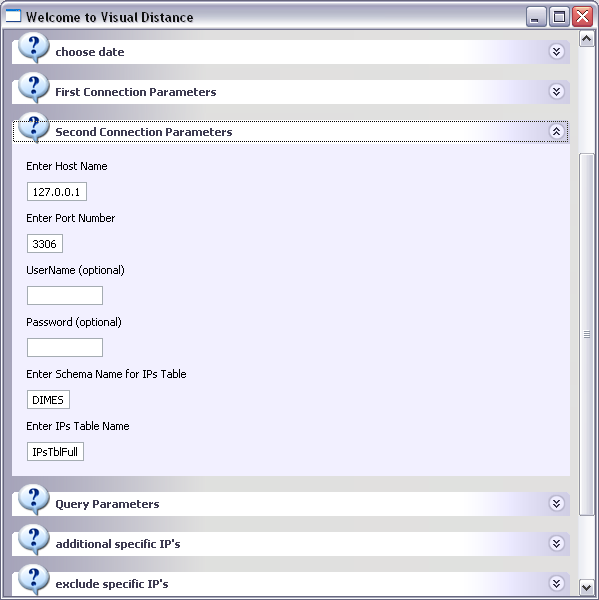
אנו מחזיקים שני אובייקטים של חיבור ל-DB של DIMES. אנו זקוקים לשני אובייקטים, כיוון שאנו מתחברים לשני שרתים שונים של ה-DB. באחד נמצאות הסכמות והטבלאות של תוצאות הניסויים (כמו הסכמה dimes\_results\_2012 והטבלאות המתאימות בה raw\_res\_main\_2012\_xx, ו- raw\_res\_traceroute\_2012\_xx, כאשר xx מייצג את מספר השבוע בשנה), ובשרת השני נמצאת הטבלה IPsTblFull המכילה מיפוי מ-IP למיקום שלו על פני כדור הארץ.

את המידע הדרוש ליצירת חיבורים אלו אנו מבקשים מהמשתמש דרך הממשק הגרפי. אנו ממלאים את השדות הנחוצים במידע של ברירת מחדל אך המשתמש יכול ורשאי לשנות מידע זה. באיורים הבאים ניתן לראות את מסכי הזנת הנתונים לחיבורים אלו. הנתונים שניתן להזין:

* כתובת ה-Host בו נמצא שרת ה-DB. בדר"כ יהיה זה על המחשב המקומי או דרך Tunnel שנוצר בעזרת PuTTY, וגם אז נפנה למחשב המקומי. לכן בחרנו שברירת המחדל תהיה כתובת המחשב המקומי.
* מספר פורט - פורט ברירת המחדל ב-MySQL, כאשר ה-DB נמצא על המחשב המקומי הוא 3306. אם משתמשים ב-Tunnel, יש להגדיר את הפורט דרכו יוצאים לסכמה הרלוונטית.
* במידה ויש צורך בשם משתמש ו/או סיסמא, ניתן להזין גם אותם. אם לא יוזן דבר בשדות אלו, הם יישארו ריקים וייעשה שימוש בערכי ברירת מחדל (user = codeLimited, password=, כלומר סיסמא ריקה).
* שם הסכמה ושם הטבלה או טבלאות - ברירת המחדל כאן היא השנה והשבוע הנוכחיים עבור הסכמה וטבלאות התוצאות הראשיות. עבור הסכמה וטבלת התאמת ה-IP למיקום הגיאוגרפי בכדור הארץ, ברירת המחדל היא הסכמה DIMES והטבלה IPsTblFull.



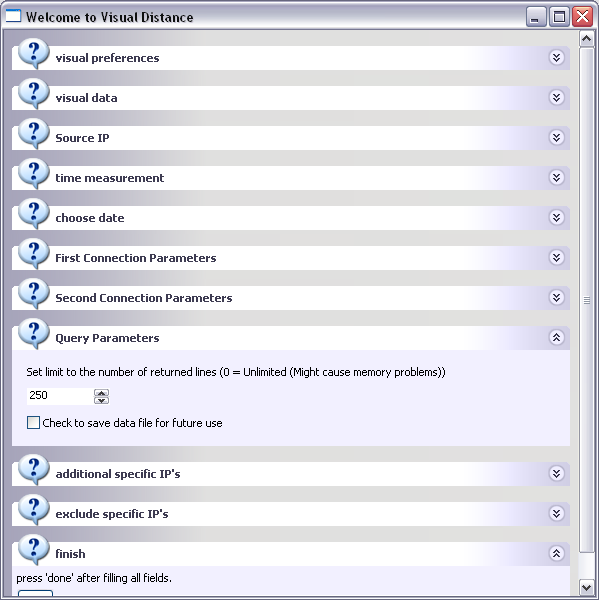
איור 4.4 – חלון ממשק המשתמש בחלק בו מגדירים את מאפייני החיבור הראשון



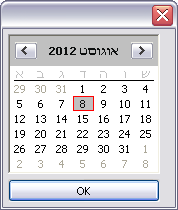
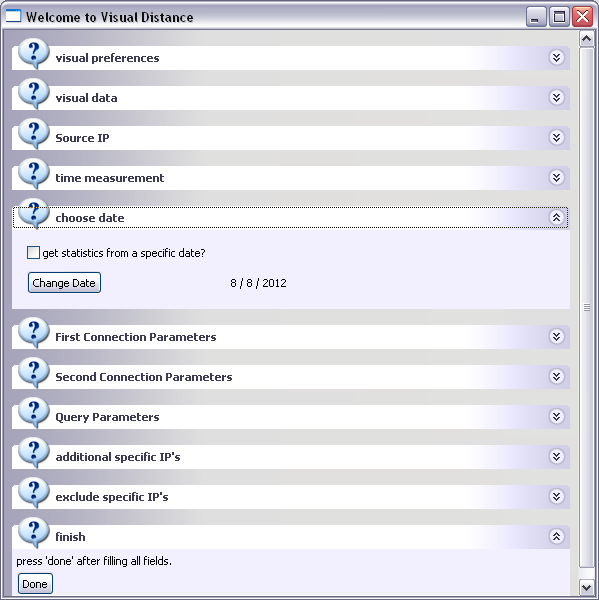
איור 4.5 – חלון ממשק המשתמש בחלק בו מגדירים את מאפייני החיבור השני

בעזרת שני אובייקטים אלו אנו מריצים שאילתות שונות מול הטבלאות השונות. עבור החיבור עם תוצאות הניסויים יש לנו מספר סוגי שאילתות שאנו מריצים. יש לציין כי השאילתות שלנו בנויות כך שרק ניסויים שהצליחו (כלומר ה-traceroute הגיע ליעדו בהצלחה) ייכללו בתוצאות שחוזרות.

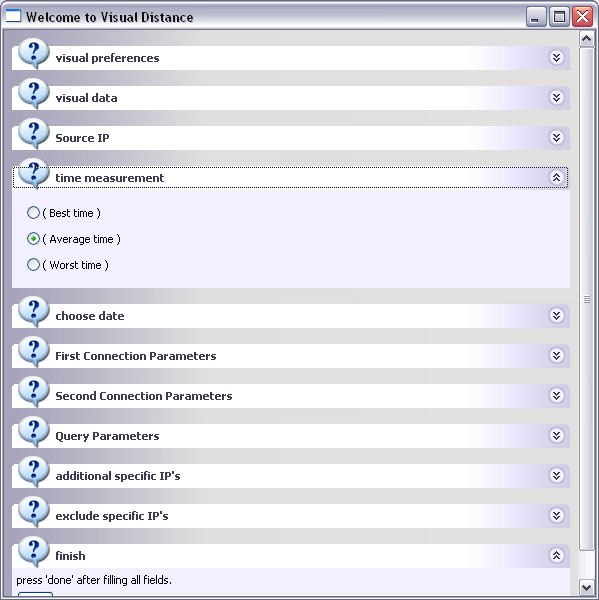
* שאילתה כללית - כאשר המשתמש לא מגדיר הגדרות מיוחדות כמו רשימת IP שיש לכלול בתוצאות נבצע שאילתה רגילה זו. לשאילתה זו מספר וריאציות התלויות בקלט מהמשתמש:
  + LIMIT - אם המשתמש הגדיר הגבלה על סך התוצאות המוחזרות, אנו מוסיפים הגבלה על מספר התוצאות שתחזיר השאילתה. התוצאות שיחזרו במקרה של הגבלה שכזו יהיו התוצאות הראשונות שיתאימו. באופן עקרוני, התוצאות יגיעו לפי סדר ה-sequence number, מספר סידורי רץ שמייצג את מספר הניסוי במערכת DIMES.  
    אם לא הוגדר LIMIT ע"י המשתמש (הזנת 0 בתור ה-LIMIT במסך הזנת הנתונים), אזי לא נגביל את מספר התוצאות שיחזרו, ולמעשה נקבל את כל התוצאות המתאימות לשאר ההגדרות שהזין המשתמש.  
    כאן ניתן לראות את החלק בממשק המשתמש בו מגדירים את ה-LIMIT  
      
    איור 4.6 – חלון ממשק המשתמש בחלק בו מגדירים LIMIT



* + תאריך ספציפי - ניתן להגדיר תאריך ספציפי (יום ספציפי) ולקבל תוצאות עבור יום זה בלבד. יש לשים לב ולהגדיר תאריך שנמצא בשבוע של השנה שמתאים לטבלה שממנה הוגדר ששולפים מידע, אחרת נקבל אפס תוצאות. אם לא מסמנים את תיבת הסימון עבור תאריך ספציפי, באופן דיפולטיבי אנו שולפים תוצאות ללא תלות בתאריך ביצוע הניסוי.  
      
      
    איור 4.7 – חלון ממשק המשתמש בחלק בו ניתן להגדיר שימוש בתאריך ספציפי ואת התאריך עצמו  
      
    בעת לחיצה על הכפתור Change Date ייפתח חלון עם לוח שנה ובו ניתן לבחור תאריך של יום ספציפי. בעת לחיצה על OK, ייסגר החלון והתאריך הנבחר יחליף את התאריך שהיה כתוב קודם לכן. ברירת המחדל היא התאריך הנוכחי במערכת המריצה את האפליקציה.  
      
    איור 4.8 – חלון שנפתח בעת לחיצה על הכפתור Change Date שניתן לראות באיור הקודם



* + סוג הזמנים הנמדדים - ניתן להגדיר האם מעוניינים לקבל תוצאות עבור הזמן הממוצע שנמדד בין המקור ליעדים השונים, או לחילופין עבור הזמן הטוב ביותר או הזמן הגרוע ביותר. המידע הזה נמצא בטבלאות של תוצאות הניסויים ולכן ראינו לנכון לאפשר למשתמש לבחור איזה סוג מידע לקבל. ברירת המחדל שלנו היא הזמן הממוצע, אך דרך המסך המתואר באיור למטה ניתן לשנות בחירה זו  
      
    איור 4.9 – חלון ממשק המשתמש בחלק בו מגדירים את סוג הזמנים הנמדדים



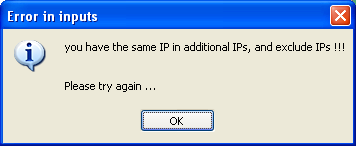
* שאילתה על רשימת IP-ים ספציפיים - אנו מאפשרים למשתמש להגדיר רשימה של עד 10 IP-ים ספציפיים שהמשתמש מעוניין לוודא שיהיו בתוצאות שחזרו. המשתמש יכול לבחור אם הוא רוצה לכלול את ה-IP הללו בתור ה-IP היחידים שיחזרו (לא תרוץ שאילתה כללית) או בתור IP שיהיו בנוסף לרשימה הכללית.  
  על מנת לתמוך באפשרות זו היה עלינו להגדיר שאילתה מיוחדת שתשלוף מידע עבור IP ספציפי, ואז עלינו להריץ אותה כמספר הפריטים שיש ברשימה שהעביר המשתמש, כלומר לכל היותר 10 פעמים. הגבלנו את הרשימה ל-10 פריטים היות ושאילתה זו לוקחת זמן רב יותר מהשאילתה הכללית, וגם אנו מאמינים ש-10 פריטים שכאלה מספיקים לצרכי המשתמשים. ניתן לשנות גודל רשימה זו במידת הצורך אך הדבר יכול לגרום להארכת זמן העיבוד מול ה-DB.  
  במצב שמתייחסים לרשימה בתור IP נוספים, אנו חוסכים שאילתות ע"י בדיקה של כל IP ברשימה אם הוא כבר נמצא ברשימת היעדים שכבר חזרו, ואם כן, פשוט לא נחפש אותו, כי חבל על השאילתה והזמן.
* שאילתה לבירור מיקום גיאוגרפי של IP כלשהו - כפי שהוסבר קודם, הסכמה והטבלה שמכילה את הקישור בין IP למיקומו הגיאוגרפי ברשת הקואורדינאטות של כדור הארץ נמצאת בשרת אחר. לכן עלינו להקצות אובייקט חיבור אחר וגם שאילתה אחרת לצורך בירור המיקום של כל IP. יצרנו שאילתה שכזו, שלא היה בה צורך בפרמטרים, פרט לפרמטרים ההגיוניים של שם הסכמה, שם הטבלה וה-IP עליו אנו רוצים לברר. כך בעצם עם אותה שאילתה, בשינוי של ה-IP בלבד נוכל לברר את המיקום של ה-IP של מקור הניסוי וגם של ה-IP-ים של יעדי הניסוי.  
  שאילתה זו מתרחשת מאחורי הקלעים, ורק לאחר שכבר שלפנו את המידע מהטבלאות הראשיות של הניסויים. לפני הכנסה של מידע על יעד כלשהו לרשימת היעדים שיעברו למודול ה-Matlab, אנו מבררים את מיקומו הגיאוגרפי.  
  ניסינו לחשוב על דרכים לייעל שאילתה זו, כיוון שלמעשה מריצים שאילתה מחדש על כל יעד בנפרד. זה מייצר הרבה overhead לכל יעד. לצערנו, בנייה של שאילתה אחת מקיפה עבור כל ה-IP נראית לנו כמו יצירה של שאילתה מאוד מסובכת שגם תקשה עלינו מבחינה תכנותית, גם תקשה על אדם חיצוני שיקרא את הקוד וגם עשויה בסופו של המאמץ להכביד על החישוב בצד השרת של ה-DB. שילמנו על זה מחיר מסוים במהירות העבודה, אך מבדיקות שערכנו, בירור מיקומים על רשימה של כ-50 יעדים לוקחת בערך שתי שניות, ולכן אם לא מתעקשים לקבל את כל היעדים של מקור מסוים בשבוע מסוים בשנה כלשהי (עשוי להגיע למעל 10,000 תוצאות), אזי אין לנו חשש מכך שמהירות העבודה של האפליקציה תיפגע בצורה ניכרת.

**נקודות חשובות בתהליך העבודה מול ה-DB והעבודה על השאילתות, יחד עם מספר אתגרים עליהם התגברנו:**

* ישנה תופעה ידועה של traceroute שגורמת לכך ש-IP היעד מופיע יותר מפעם אחת. כלומר נוצר שכפול של השורה שבה מוצג הזמן שלקח להגיע ל-IP היעד. ניתן לזהות תופעה זו בטבלאות של DIMES כאשר רואים עבור ניסוי מסוים (ניסוי מזוהה חד-חד-ערכית לפי מספר sequence number) את IP היעד מופיע יותר מפעם אחת בודדת (לעיתים אפילו 12 פעמים אך לרוב בסביבות 4 עד 5 שכפולים). תחילה חשבנו לעשות שימוש ב-DISTINCT בשאילתה שלנו, אך אופציה זו סיננה רק את השורות שבהן היה מספר ניסוי זהה, IP זהה וגם זמן נמדד זהה. לצערנו ישנם ניסויים בהם התרחשה תופעת השכפול הזו גם עם זמני מדידה שונים בכל שורה. לבסוף השתמשנו בפקודת GROUP BY בסוף השאילתה שלנו, והפעלנו אותה על המספר המזהה של הניסוי. מכיוון שמספר זה הוא כאמור ייחודי, נשארנו עם תוצאה אחת ויחידה עבור כל ניסוי, ותוצאה זו היתה תמיד התוצאה הראשונה שהתקבלה בניסוי - בדיוק ההתנהגות לה ציפינו ואותה רצינו לקבל.  
  מידע נוסף על התופעה ב-traceroute ניתן למצוא באינטרנט, לדוגמא:
  + <http://www.velocityreviews.com/forums/t38637-tracert-returns-same-ip-address-two-or-three-times.html>  
    או
  + <http://www.digitalpoint.com/lists/47778.html>  
    וגם כאן
  + <http://www.qa.downappz.com/questions/last-3-hops-of-a-traceroute-are-the-exact-same-address-why.html>
* ציינו קודם שהיינו זקוקים לשני אובייקטים של חיבור לשרת ה-DB של DIMES, מכיוון שהיינו צריכים לפנות לטבלה נוספת שנמצאת בסכמה אחרת על שרת אחר - טבלת הנ"צ-ים, הטבלה שמאפשרת לנו לברר מהו המיקום הגיאוגרפי המדויק של כל IP. עובדה זו הקשתה עלינו מאוד. השאיפה הייתה לייצר שאילתות כמה שיותר כלליות, שנוכל להריץ פעם אחת או מספר מצומצם של פעמים (עבור רשימת IP ספציפיים שהוזכרה קודם). לכן השאיפה הייתה שהמידע שמתקבל על מיקום ה-IP ייכנס לשאילתה העיקרית. מכיוון שהסכמה נמצאת בשרת אחר, נאלצנו להתמודד עם הבעיה בדרך מקורית של הרצת שאילתה לכל IP מול השרת השני וכך לברר לכל IP את מיקומו הגיאוגרפי על כדור הארץ.
* Private Source IP - במבנה של כתובת IP מסוג IPv4, יש שלוש קבוצות של טווחי כתובות שמשמשים בתור כתובות פרטיות. הכוונה בכתובות פרטיות היא שכל אחד יכול לעשות בהן שימוש בתוך הרשת הפנימית שלו (לדוגמא NAT או רשת פנימית של משרד מסוים), וידוע שאין להן משמעות מחוץ לרשת הפנימית. כתובות כאלו לדוגמא 10.0.0.42 או 192.168.1.101. הקושי שהתעורר בעקבות כתובות אלו מבחינתנו היה באיתור המיקום הפיסי של המקור. מובן מאליו שלא ניתן להריץ ניסוי traceroute לעבר כתובת שנמצאת בטווח הכתובות הללו, מכיוון שיש הרבה מאוד Host-ים עם כתובת זהה. בנוסף, לרוב כתובות אלו מוסתרות מאחורי רשת פנימית ובעצם לא חשופות לעולם החיצון. כלומר מהצד של היעדים אנו בטוחים שלא ניתקל בכתובת פרטית. מה לגבי כתובת המקור? כאן נתקלנו בקושי. ראינו שיש בטבלאות של תוצאות הניסויים לא מעט ניסויים שהמקור שלהם כתוב בתור כתובת ששייכת לטווח הכתובות הפרטיות. הפתרון הגיע ע"י כתיבת פונקציה בוליאנית שמחזירה true אם הכתובת נכללת בטווח הכתובות הפרטיות או false אחרת. השתמשנו בפונקציה זו על מנת לאתר כתובת מקור פרטית. ברגע שנמצאה כתובת שכזו, ייצרנו שאילתה חדשה שהסתמכה על ההנחה הבאה - כתובת פרטית של מקור כלשהו תצא בדר"כ מהרשת הפנימית דרך ממשק עם כתובת כלשהי שאיננה פרטית, ותמיד מכתובת זו. תחת הנחה זו, ביצענו שאילתה לטבלה המכילה את פירוט ניסוי ה-traceroute כך שקיבלנו את כל רשימת הכתובות שעברו בדרך מהמקור (בעל הכתובת הפרטית) ליעד. חיפשנו ברשימה זו את הכתובת הראשונה שאיננה פרטית, ולקחנו אותה בתור כתובת ה-IP של היעד. הנחה הנוספת שמסתתרת כאן היא שמקור בעל כתובת פרטית נמצא (מבחינה גיאוגרפית פיסית) במרחק קטן מאוד מציוד הקצה (ראוטר, סוויצ' וכו') שמספק לו מצד אחד את הרשת הפנימית ומצד שני ממשק בעל כתובת ציבורית. לדוגמא, המחשב הנייד שיש לי בבית נמצא ברשת הפנימית שמייצר הנתב שלי ושניהם נמצאים באותה דירה. דוגמא אחרת, המחשב שלי בעבודה נמצא באותו בניין (בקומה אחרת) שבו נמצא הנתב שמאפשר לו להתחבר לשאר העולם (יחד עם שאר מחשבי העובדים שגם להם יש כתובות פרטיות).  
  יש לציין כי את ההנחות שהנחנו, בדקנו עבור מספר כתובות פרטיות ומספר ניסויים מאותן כתובות וראינו כי ההנחות הללו מתקיימות. אנו משאירים אותן בגדר הנחות, מכיוון שאין לנו דרך לעבור על כל ה-DB ולבדוק את נכונותן.  
  עוד נציין כי עשינו שימוש בהמרת כתובת IP מצורת dotted notation למספר מטיפוס long על מנת לאפשר לנו לבדוק אם כתובת נמצאת בטווח הכתובות הפרטיות בצורה קלה יותר. קל יותר לבדוק תנאי שכזה על מספרים מאשר על מחרוזות.  
  מידע נוסף ורשימת טווחי הכתובות הפרטיות ניתן למצוא כאן:  
  <http://en.wikipedia.org/wiki/IPv4#Private_networks>
* תהליך בניית השאילתות - לא פשוט להריץ שאילתות מתוך קוד Java, ועל אחת כמה וכמה לא פשוט לתכנן שאילתות עם פרמטרים משתנים. תהליך העבודה לבניית השאילתות השונות היה צריך להיות כזה שננסה להריץ את השאילתה על פלטפורמה שידוע לנו שהיא עובדת, להגיע למצב שהשאילתה מחזירה את התשובות להן אנו מצפים ורק אז להכניס אותה לקוד שלנו. לכן תהליך העבודה שסיגלנו לעצמנו היה כזה:  
  + התעורר צורך בשאילתה מסוג חדש או וריאציה על שאילתה קיימת
  + התחברנו ל-DB של DIMES בעזרת כלי MySQL Workbench, אותו ניתן להוריד בחינם מהאתר:  
    <http://dev.mysql.com/downloads/workbench/>



* + כתבנו את השאילתה ללא פרמטרים (השתמשנו בערכים קבועים כמו IP מסוים, תאריך מסוים וכו') והרצנו ב-Workbench עד שראינו שקיבלנו תוצאות שמתאימות לצורך שלנו. בדרך היינו נתקלים בשגיאות סינטקטיות בניסוח השאילתות, ומתוך ידיעה שהבעיה היא אצלנו ולא ב-Workbench יכולנו למצוא את הבעיות יותר בקלות.
  + לאחר שהתקבלה שאילתה עובדת ב-Workbench, לקחנו אותה כמו שהיא והרצנו דרך הקוד שלנו. רצינו לראות שמתקבלת אותה תוצאה מהקוד ומה-Workbench.
  + לאחר שוידאנו שמתקבלת אותה התוצאה, החלפנו את הערכים הקבועים שהכנסנו בפרמטרים שמתקבלים מהמשתמש דרך הממשק הגרפי.
  + ניסינו להריץ את השאילתה דרך הקוד עם פרמטרים חדשים ולהריץ אותה עם אותם ערכי פרמטרים (בתור קבועים) גם ב-Workbench, על מנת לוודא סופית שמתקבלות גם עכשיו תוצאות זהות.
* ייעול השאילתה הכללית - בתחילת הדרך השאילתה הכללית הייתה מסורבלת למדי. ראינו שלוקח לה מספר דקות (סביב 5) להחזיר תשובה על שאילתה עם הגבלה ל-1000 יעדים ברשימה. התייעצנו עם פורומים וחברים שעוסקים או עסקו בתחום ה-DB ויחד איתם ייעלנו את השאילתה ע"י שימוש ב-LEFT JOIN. השימוש באופרטור הזה מאפשר לנו לסנן חלקית את התוצאות מהטבלה המשנית, זו שמכילה את הפירוט לכל ניסוי וניסוי, ולקבל ממנה רק את השורות שמכילות את כתובת היעד (כלומר סינון הכתובות בדרך אל היעד) ורק שורות שמכילות ניסוי מסוג traceroute (סינון השורות של ה-ping המקדים ל-traceroute). הוספנו בדיקה על התאמת ה-sequence number כדי למנוע כפילויות, היות וזהו מזהה ייחודי לניסוי.  
  שינוי זה ייעל את השאילתה לריצה של מספר שניות עבור מגבלה של מאות יעדים ועד דקה או שתיים עבור מגבלות של אלפי יעדים.
* ניפוי כתובות IP לא רצויות - כאשר נתקלנו בתוצאות שנראו לנו קיצוניות מבחינת הזמן שנמדד אליהן, הגענו למסקנה שיש לאפשר למשתמש להכניס רשימה של כתובות IP של יעדים שהוא מעוניין שלא ייכללו ברשימת התוצאות. בדומה לרשימת הכתובות הנוספות עליה הסברנו קודם, אפשרנו לצרף עד 10 כתובות IP שאותן רוצים להבטיח שלא ייכללו ברשימת התוצאות.  
  כעת, אם המשתמש סימן שברצונו לסנן רשימה של כתובות IP כלשהן, התלבטנו איפה יבוצע הסינון. האם בזמן השאילתה, ואז הסינון יבוצע ע"י הרצת השאילתה ב-DB ואנחנו היינו מקבלים חזרה את התוצאות המסוננות. דבר זה היה מקל את מלאכת ה-parsing של המידע שחוזר, אך מקשה מאוד את אופן כתיבת השאילתה. האפשרות השנייה הייתה לבצע את הסינון בעת הכנסת יעד חדש שחזר מהשאילתה הכללית. כל יעד שמועמד להיכנס לרשימת היעדים שחזרו נבדק אם הוא מופיע ברשימת הכתובות הלא רצויות, ורק אם הוא לא ברשימה הלא רצויה, הוא נכנס לרשימת התוצאות.  
  סינון זה עלול ליצור מצב שבו אם הוגדר ע"י המשתמש LIMIT כלשהו, הוא עשוי לקבל עד 10 תוצאות פחות ממה שביקש. כדי לפתור מצב זה, מנינו את מספר היעדים שסוננו בפועל, והרצנו שאילתת המשך על מנת לקבל יעדים נוספים כדי למלא את המכסה שהוגדרה ע"י LIMIT. מניה זו מתרחשת גם על השאילתה הנוספת שתוארה כרגע, מכיוון שגם בתוצאות שחוזרות ממנה ייתכן שנפגוש כתובות IP שעלינו לסנן.  
  קושי נוסף שדרש מענה ברגע שהכנסנו את האופציה לסנן כתובות IP לא רצויות הוא המצב שבו משתמש הכניס (בכוונה או בטעות) את אותה כתובת IP גם ברשימה של הכתובות הרצויות וגם ברשימה של הכתובות הלא רצויות. מצב זה נמנע ע"י בדיקה שמתבצעת בעת הלחיצה על כפתור ה-Done, ומקפיצה התראה על מצב זה למשתמש, בטרם הופעל החיבור ל-DB ובטרם הורצה אפילו שאילתה אחת.  
    
  איור 4.10 – חלון השגיאה שמוצג למשתמש במקרה שאותה כתובת IP הוכנסה לשתי הרשימות - הרצויים והלא רצויים.



**כתיבת המידע שנאסף לקובץ שייקרא ע"י Matlab**

לאחר שאספנו את המידע הנחוץ על כל היעדים (מי הם היעדים, מיקומם והזמן שלקח להגיע אליהם) ועל המקור (מיקומו והאם הוא כתובת פרטית או לא), עלינו להעביר אותו למודול ה-Matlab בדרך כלשהי. הדרך שנבחרה היא ע"י קובץ txt במיקום ובשם ידועים מראש.

במקביל לכתיבה לקובץ זה, ניתנת למשתמש האפשרות לסמן V בתיבת בחירה אם ברצונו לשמור עותק נוסף של המידע שיישלף. אם המשתמש בחר באופציה זו, יישמר עותק נוסף של הקובץ שמועבר ל- Matlab, תחת תיקיית OutputFiles שבתיקיית האפליקציה. שם הקובץ הוא לפי התאריך והשעה בהם לחץ המשתמש על כפתור ה-Done. הסיבה לשמירת הקובץ בצד היא רצון לאפשר פתיחה של תוצאות הריצה ע"י Matlab גם ללא צורך בחיבור ל-DB. בנוסף, ניתן לעשות בצורה זו השוואות בין ריצות שונות.

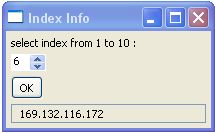
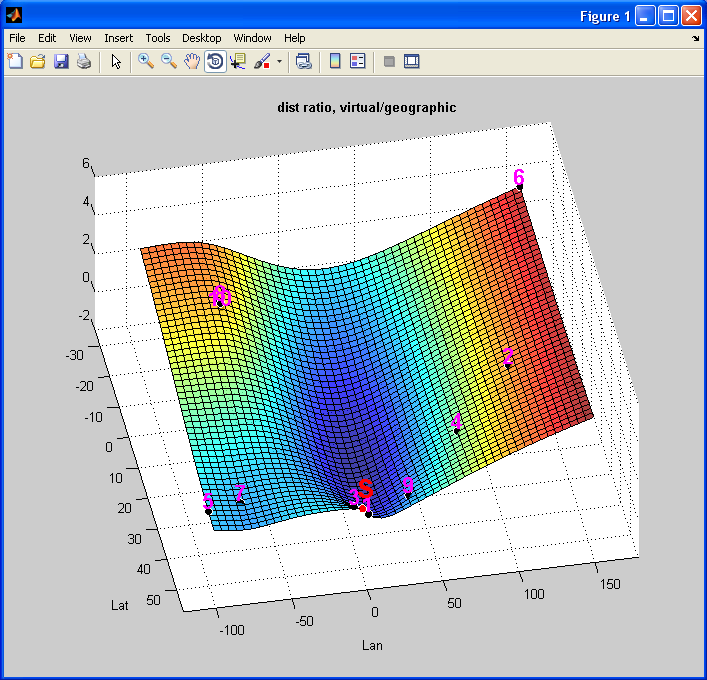
**ויזואליזציה של התוצאות –**

לאחר שהנתונים הוזנו לתוך קובץ הנתונים, המערכת מפעילה את הקובץ runMatlab שגם הוא מסוג bat, קובץ זה פותח את ה-Matlab ומריץ עליו את המתודה הראשית של מערכת הויזואליזציה,חלקי המתודה הם:

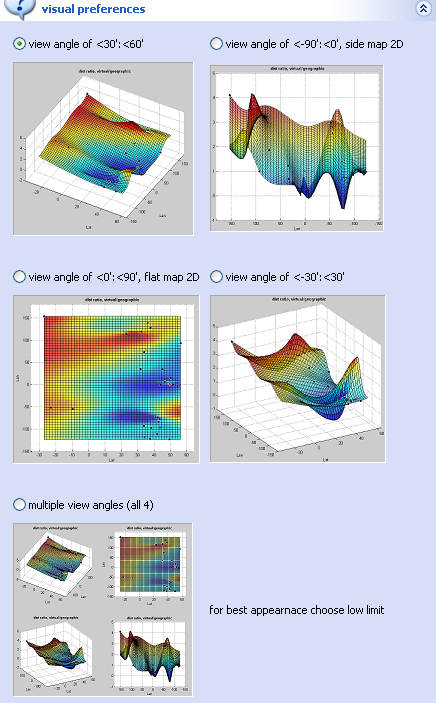
* קריאת הנתונים מהקובץ
* הכנת הנתונים בצורה נכונה למתודת התצוגה הגרפית
* אלגוריתם למציאת המרחקים הגאוגרפים האמיתיים במימוש לאלגוריתם Haversine
* חישוב מרחק וריטואלי מתוך מדידות הזמנים לפי כלל האצבע
* מיצוע הנתונים ע"פ משטח לפי נוסחת\מתודת TPAPS, והצגה גרפית שלהם על גרף D3, בהתאם לבחירת המשתמש באיזה אופן יוצגו הנתונים.

כמו כן, במקביל לפתיחת ה-Matlab, קוד הJAVA יקרא למתודה אשר מריצה מנשק GUI נוסף אשר מציג למשתמש מקרא אשר מציג עבור כל נקודה את המידע שלה.

משתמש יבחר את מספר הנקודה וילחץ OK והמקרא יציג לו את המידע עבור הנקודה.



איורים 4.4, 4.5 – גרף תוצאות ומקרא לצידו אשר מציג את המידע (כתובת IP) עבור נקודה 6



הסבר שימוש בתפריטי ממשק המשתמש:

Visual preferences – הלשונית הראשונה בממשק משתמש,

מאפשר למשתמש לבחור באחד מבין 4 אופציות הצגת תוצאות,

הראשונה והרביעית הן בתלת מימד, השניה והשלישית הן

בדו-מימד של תצוגה עילית או צדדית, כמו כן, משתמש יוכל לבחור

באופציה החמישית והיא להציג את כל 4 האופציות יחדיו, בחירה זו

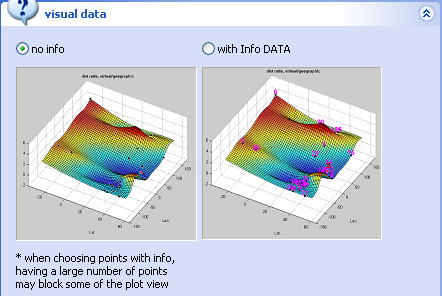
אינה מומלצת כאשר ישנו מידע רב בגרף, וכך למעשה התצוגה

המרובה תציג כל אחת מ4 התצוגות בגודל קטן יותר.

כמו כן חשוב לציין שבכל אחת מהאופציות המשתמש עדיין יכול

לאחר קבלת התוצאות לשנות את זווית הראיה בעצמו.

Visual data – מאפשרת למשתמש לבחור האם להציג את התוצאות



עם מידע או בלי מידע נוסף על הגרף, המידע הוא בעצם הוספה של

מספר ליד כל נקודה, על מנת שיהיה לנקודות סימן זיהוי,

יש לשים לב שאם לגרף יש מספר רב של נקודות, אזי ייתכן שיהיו

המון מספרים על הגרף בצורה כזו שהם כבר יסתירו זה את זה ואולי

אף את הגרף.

source ip – לשונית שבה מתבקש המשתמש לבחור



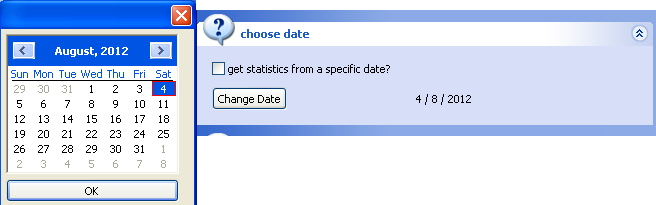
את הכתובת IP של המקור ממנו נמדדו תוצאות.



Time measurement – לשונית שבוחרים בה לפי איזה מדידת זמנים

מעוניינים לקבל תוצאות.

לפי הזמן בממוצע, הטוב ביותר או הגרוע ביותר.

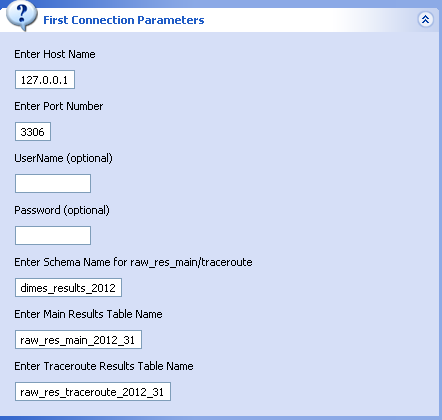


Choose date – בלשונית זו מתבקש משתמש

להחליט האם הוא מעוניין בתוצאות מתאריך מסויים

ואם כן, אז גם לבחור תאריך מתוך חלון בחירת תאריך.

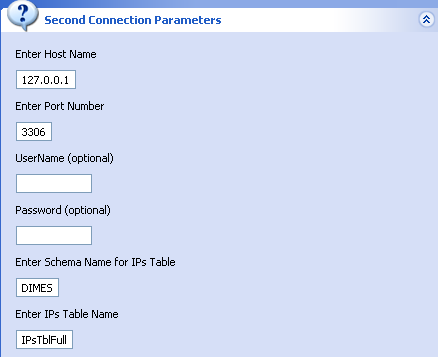
first connection parameters – בלשונית זו יבחר המשתמש



נתונים עבור חיבור ראשון לDB כמו:

Host, מספר של הPort, שם משתמש וסיסמה, שם הסכימה

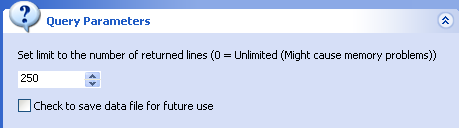
והטבלה וגם שם טבלת הtraceroute ממנה יילקחו הנתונים.



Second connection parameters – לשונית דומה

לקודמת, עבור חיבור נוסף עם שינויים קלים לעומת

החיבור הקודם.

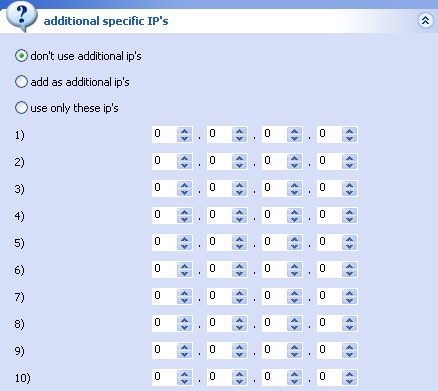


Query parameters – אופציות לשאילתה, כמו כמות

נקודות שמשתמש מעוניין שיוחזרו מהDB, והאם

לשמור את הנתונים לאחר הריצה.

additional & exclude IPs – משתמש יכול לבחור האם



להוסיף IP, כך שיובטח שיהיה בגרף (כל עוד הip קיים

עבור הנתונים שהוזנו ע"י המשתמש), כמו ניתן גם

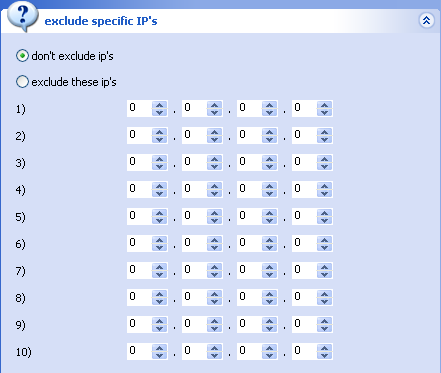
להסיר IP מנקודות שמעוניינים לקבל בתוצאות

(לדוגמא עבור IP חריג שסוטה מהממוצע תוצאות)

בכל אחד מהמקרים ניתן לרשום עד 10 IPs, כאשר

המערכת לא מאפשרת למשתמש לרשום IP זהה בשתי

הרשימות.



# ניתוח תוצאות

## השוואות בין תוצאות הסימולציה לעבודה בזמן אמיתי

להלן דוגמא של טבלה:

טבלה 1 – השוואת ביצועים

|  |  |  |  |
| --- | --- | --- | --- |
|  | **פרמטר** | **סימולציה** | **זמן אמיתי** |
|  | **הגבר** | **8 dB** | **7.5 dB** |
|  | עוצמת רעש | **-30 dBm** | **-50 dBm** |
|  |  |  |  |
|  |  |  |  |

## ביצועי המערכת מבחינת זמן אמיתי

# סיכום, מסקנות והצעות להמשך

זהו הפרק החשוב ביותר. בפרק זה יש לכלול:

* בחינת תוצאות הפרויקט מול המטרות שהוגדרו מלכתחילה
* הצעות לשיפור ביצועי המערכת
* אפשרויות להמשך פעילות (פיתוח/מחקר) עתידית

אפשר להוסיף אלגוריתם למציאת חריגים בגרף שהורסים לאחרים, כי הויזואליזציה בגרף בנויה על שינויים – אם יש אחד ששונה בהרבה מהשאר, לא רואים את ההבדלים בין האחרים בבירור.

# מקורות

בפרק זה יש לכלול את כל מקורות הספרות שהסתמכתם עליהם. ציון המקורות חייב לאפשר איתור מדוייק של המסמך, כנהוג בספרות המקצועית. להלן כמה דוגמאות:

**פרסום בעברית:**

1. י. פיינגלרנט, "עיבוד תמונות אולטרה-סאונד למטרות רפואיות", חיבור על מחקר לשם מילוי חלקי של הדרישות לקבלת התואר מגיסטר למדעים בהנדסת חשמל, טכניון 1982

**פרסומים באנגלית:**

**ספר**

1. A. M. Bronstein, M. M. Bronstein, and R. Kimmel. "Numerical geometry of non-rigid shapes”, Springer-Verlag New York Inc, 2008.

**מאמר:**

1. G. B. Giannakis , "Highlights of Signal Processing for Communications", IEEE Signal Processing Magazine, Vol. 16, no 2, pp. 14-49, March 1999

**דף נתונים של רכיב:**

1. “Spartan-3A DSP FPGA Family Data Sheet”, XILINX Product Specification DS610, October 4, 2010. <http://www.xilinx.com/support/documentation/data_sheets/ds610.pdf>

**Application Note :**

1. X. Zhang, "Developing a CCStudio 2.0 DSP/BIOS Application for FLASH Booting on the TMS320C5402 DSK", TI Application Report SPRA661A, November 2000. <http://www.ti.com/lit/an/spra661a/spra661a.pdf>

**User's Guide:**

1. "TMS320C6201/6701 Evaluation Module User's Guide", SPRU269F, August 2002
2. Code Composer Studio IDE 2.0 online help.

**קישורים למקורות באינטרנט:**

1. "Dolby E Multichannel Coding for DTV Audio Production and Distribution", <http://www.dolby.com/tech/m.br.9903.epaper.pdf>
2. G. Welch and G. Bishop – "An Introduction to the Kalman Filter", [http://www.cs.unc.edu/~welch/kalman/kalman\_filter/kalman.html - pgfId-11854](http://www.cs.unc.edu/~welch/kalman/kalman_filter/kalman.html#pgfId-11854) <http://www.cs.unc.edu/~welch/kalman/kalman_filter/kalman.html#pgfId-11854>
3. C. Mercer, "Smoothing Spectral Data", *The PROSIG Digital Signal Processing Tutorials*, <http://www.prosig.com/signal-processing/smoothingspectra.html>

קשיים בדרך, ופתרונם:

