**מעבדה בבינה מלאכותית**

**דו"ח תרגיל בית 1**

In a\* with heurestic1:

In 7000 milSec one level was solved

In 310000 milSec all the levels was solved

In a\* with heurestic2:

In 9000 milSec one level was solved

In 355000 milSec all the levels was solved

In IDLS:

In 9000 milSec one level was solved/\*\*

In 355000 milSec all the levels was solved/\*\*

**FIGURE 1**

זהו הגרף שמייצג את הרמות (LEVEL) ביחס למספר הצעדים בכל רמה.  
עבור הפעלת אלגוריתם A\* עם הפונקציה ההיוריסטית הראשונה שקראנו לה DISTANCE התוצאות היו קרובות לתוצאות הנתונות, וה branching factor היה יורד ונתן תוצאות נמוכות (בסביבות של 1).

ניסינו לשפר ביצועים על ידי מציאת פונקציה היורסטית אחרת שחשבנו שהיא יכולה לשפר ביצועים, ולכן הגדרנו הפונקציה CONTRA\_MOBILITY והתחלנו לראות איך התוצאות הולכות להשתנות.  
מה שיצא לנו שעבור הפנקציה CONTRA\_MOBILITY נתנה ביצועים זהות לאלה הנתונות מבחינת מספר צעדים, אבל ה TRADE OF שנוצר זה זה branching factor גדל יחסית לזה שהיה בפונקציה DISTANCE.

**FIGURE 2**

אפשר לראות מ **FIGURE 2** שה branching factor בפונקצית DISTANCE הוא יותר טוב מזה שב CONTRA\_MOBILITY ובעיקר ברמות הבסיסיות (Beginner).  
ה branching factor הוא פונקציה יורדת "ועל זה צריך לחשוב ולהוסיף לוגיקה".

כדי לשפר הביצועים מבחינת ה branching factor נסינו לשלב בין שתי הפונקציות היוריסטיכות שיצרנו והשוואנו ביצועים ביניהם.  
השילוב נעשה בפונקציה CONTRA\_MOBILITY+DISTANCE ונשינו כל פעם לתן ו שונות ולראת מה יהיו התוצאות.  
התוצאות מבחיננת מספר צעדים ומבחינת branching factor אפשר לראות ב **FIGURE 3** וב **FIGURE 4** בהתאם:

**FIGURE 3**

**FIGURE 4**

אפשר לראות שהשילוב שנעשה הוא כן הוריד את ה branching factor אבל לא נתן בדיוק תוצאות טובות מבחינת מספר צעדים.

עבור ו האלגוריתם נתן שילוב מאוד טוב בין התוצאות ואפשר לראת את זה בגרפים,  
כך ש ב **FIGURE 3** אפשר לראות שעבור פרמטרים אלו הפונקציה מאוד קרובה להתוצאות הנתונות וגם עכשיו ה branching factor ירד בצורה משמעית כמו שמיוצג ב **FIGURE 4**.