

Ben-Gurion University of the Negev
Department of Computer Science

התפתחות אינטליגנציה מלאכותית של קבוצות על
ידי תכנות גנטי
Evolving AI Strategies of Groups using
Genetic Programming



Project submitted as part of the requirements for the
B.Sc. degree of Ben-Gurion University of the Negev

By

Erez Yaakov & Avihu Shaban

The work for this project has been carried out at
Ben-Gurion University of the Negev
Under the supervision

Of

Mr. Michael Orlov & Prof. Moshe Sipper
November 2007

תוכן העניינים

3.....	מבוא
.....	המשחק
5	
.....	האפליקציה
6	
.....	תפריט
6.....	התחלה
7.....	לשונית תצוגת הדורות
.....	פאנל
7.....	האוכלוסייה
.....	פאנל
8.....	הקרב
.....	פאנל שמירה
9.....	וטעינה
.....	לוח
10.....	בקרה
.....	חלון טעינת
10.....	אוכלוסייה/מומחים
.....	השחקן
11	
11.....	אובייקט האינדיבידואל
11.....	מראה השחקן Avatar
11.....	הדקדוק
14.....	מרכיבי התהליך האבולוציוני
14.....	תנאי עצירה
14.....	XO
14.....	Mutation
14.....	Self-XO
14.....	Fitness
14.....	Selection
.....	סוגי
15.....	ניסויים
15.....	טורניר
15.....	Arena
15.....	Arena CLL
.....	אופטימיזציות והנדסת
16.....	תוכנה
17.....	ניסויים
.....	מסקנות
27.....	וסיכום
30.....	References
31.....	Appendix A

מבוא

מטרת הפרויקט שלנו היא לבחון דרכים שונות להיווצרות אינטליגנציה מלאכותית של קבוצות על ידי תכנות גנטי. הקבוצות נלחמות אחת מול השנייה בסביבה דו-ממדית על ידי יריות ושליפת הרבות.

אינטליגנציה מלאכותית הינו מושג אשר מגדיר יכולת של "מכשיר" לא ביולוגי להתנהג בצורה אינטליגנטית.

אינטליגנציה מלאכותית הינו מושג מאוד רחב, כיום קיימות גישות רבות ושונות לאינטליגנציה מלאכותית. בעבר גישות אילו התרכזו בחיפוש הפתרון הטוב ביותר מתוך מרחב הפתרונות. גישה זו נתקלה בבעיות פרקטיות של זמן ומקום מאחר ומרחב הפתרונות יכול להיות מאוד גדול, אם בכלל אפשר להגדיר את המרחב הזה. בעיות נוספות של חוסר ודאות או דטרמיניסטיות הביאו חוקרים למסקנה שבינה מלאכותית לא יכולה להתבסס רק על חיפוש. גישות נוספות של אופטימיזציות חיפוש התפתחו, יושמו היוריסטיקות והערכות לגבי כיוון החיפוש ועומקו על מנת להקטין את כמות המשאבים הנדרשים לחיפוש. גישה חדשה ל אינטליגנציה מלאכותית הם האלגוריתמים הגנטיים אשר באים לפתור בעיות אשר מרחב הפתרונות שלהם עצום או לא מוגדר. גישה זו מיישמת את העיקרון האבולוציוני בו המותאם לסביבתו שורד ומעמיד צאצאים.

Genetic algorithms^[1] (GA) הם דוגמא ל evolutionary computation שהם תת תחום של אלגוריתמי אופטימיזציה. בהינתן אוכלוסיה של פתרונות לבעיה, תכנות אבולוציוני מרחיב/מחליף את האוכלוסייה באינדיבידואלים "טובים" יותר בפתרון הבעיה. בבסיסו התהליך האבולוציוני שואב את השראתו מעולם הביולוגיה, שם לאורך זמן האבולוציה מייצרת את היצור ה"טוב" ביותר. הכרומוזומים אשר מורכבים מה DNA הם המודל האבסטרקטי לפיו אורגניזמים חיים בנויים. בכל יצור כזה יש כרומוזומים אשר תת חלקים מהם הם הגנים, גנים אלו מגדירים את צורתו, יכולותיו, תכונותיו והתנהגותו של היצור. במהלך הרבייה גנים של ההורים מורכבים יחד כדי ליצר את הדור הבא.

עקרונות ביולוגים אלו ניתן ליישם גם על האוכלוסייה של הפתרונות לבעיה. נצטרך להגדיר את ההתאמה של העולם הביולוגי לעולם של מרחב הפתרונות לבעיה שלנו. מה צריך:

- אלף-בית לגנום (ראה Appendix A)
- מבנה של אינדיבידואל אשר יודע להתנהג בהתאם לגנום
- אוכלוסייה
- שיטות רבייה
- דרך להעריך את איכות האינדיבידואל
- הצורה בה יבחרו אינדיבידואלים כדי לייצר את הדור הבא.

על אלמנטים אלו נריץ את האלגוריתם האבולוציוני (הכללי):

```
Generate a random population
Evaluate each individual in the population and assign fitness
While termination condition not met
    Selection
    Xo
    Mutation
    Assign fitness
End
```

Genetic Programming^[2] (GP)¹ הוא תת תחום של GA שבו אוכלוסיית הפתרונות מורכבת מתוכניות מחשב אשר משתנות ומשתפרות לאורך התהליך האבולוציוני, בכל שלב איכות הקוד נאמדת בעזרת אבולוציה של התוכנית והשוואתה לשאר האוכלוסיה.

תוכנית מחשב יותר "טובה" תזכה בסיכוי טוב יותר להעביר את המידע הגנטי (הקוד) שלה לדור הבא.

אנו מפעילים GP לתהליך אבולוציוני של התפתחות אסטרטגית משחק, למשחק מרובה שחקנים שיובא בהמשך. השתמשנו בסוג של Strongly Typed GP¹⁴ אשר מתבצע על ידי שימוש בדקדוק (מוסבר בהמשך).

מוטיבציה:

בתוכניות אשר מדמות תהליך אבולוציוני ישנם פרטים רבים אשר רלוונטיים לשם מעקב אחר הניסוי. שפע הפרטים וההתקדמות הבלתי פוסקת של התהליך מקשים מאוד על המתכנת לראות מה קורה בניסוי. בנוסף, בגלל האופי הסטוכסטי של התוכנית קשה מאוד לאתר שגיאות תכנותיות. כמו כן, העבודה של המתכנת יכולה להיעשות קלה בהרבה כאשר התוכנית האבולוציונית ניתנת לשליטה תוך כדי ריצתה.

על מנת להתמודד עם אתגרים אלו שמנו דגש על מספר נושאים:

- הוויזואליזציה והאינטראקטיביות של התהליך האבולוציוני.
- באפליקציה יש שליטה על מהירות הסימולציה, אפשרות להשהיה והמשך הריצה.
- סימולציית הקרב נראית בצורה גרפית.
- מראה היצורים הוא ייחודי ומצביע על מקורו האבולוציוני.
- יש מעקב אחרי כל הדורות בריצה כולל מי ההורים של כל יצור והאם התבצע xo , mutation, וכו'.
- ניתן לראות בצורה גרפית את עץ התוכנית.
- יש אפשרות של שמירה, טעינה ועריכה של יצורים בודדים ואוכלוסיות שלמות. בעזרת מנגנון זה יצרנו ושמרנו יצורים שנוצרו על ידינו ובהם השתמשנו לצורך אימון והשוואה בניסויים שערכנו.

השתמשנו באפליקציה שיצרנו כדי לערוך מספר ניסויים. בניסויים שערכנו ביצענו השוואות בין סוגי selection, ובין סוגי ניסויים. בחנו בניסויים למידה בשלבים ולמידה בשלבים מקבילים (^{13}CLL).

מה במסמך:

תיאור המשחק, עולם המשחק והאפליקציה, תיאור השחקן – המידע הגנטי, המראה ורכיביו הפנימיים, האופרטורים הגנטיים ושיטות הערכת וברירת האוכלוסיה. ולבסוף תיאור הניסויים, תוצאות ומסקנות.

המשחק

עולם המשחק הוא עולם דו מימדי בגודל קבוע, בו שתי קבוצות (אדום/כחול) אשר מורכבות ממספר שחקנים (בניסויים אשר הרצנו התמקדנו בקרבות של שתיים על שתיים).

שחקן יכול לנוע לכל כיוון (360 כיוונים) צעד אחד בכל תור.

השחקן נלחם בשחקני הקבוצה השנייה, בכך שהוא יורה בכיוון מסוים או מכה בחרב בכיוון מסוים (360 כיוונים). לאחר ששחקן מתקיף ישנה השהיה בהתאם לסוג ההתקפה (השהיה: מספר התורות הבאים בהם לא יתקוף או יזוז).

ישנם 3 סוגי התקפה:

חרב – בעלת הנזק הכי גדול, השהיה נמוכה, טווח קצר.

ירייה קטנה – בעלת נזק קטן, השהיה נמוכה, טווח ארוך.

ירייה גדולה – בעלת נזק בינוני, השהיה גבוהה, טווח ארוך.

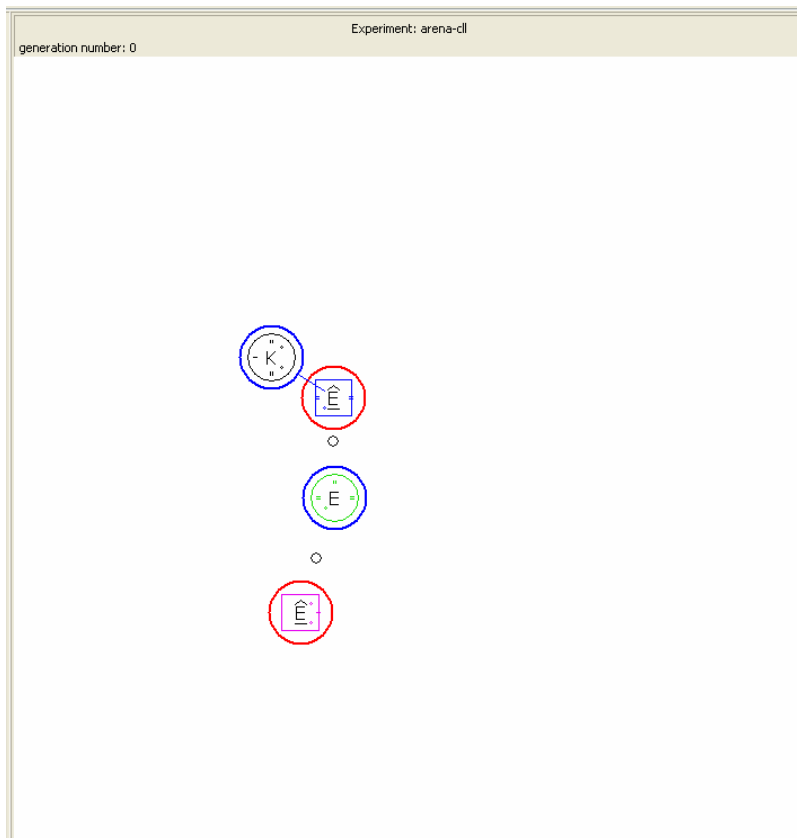
שחקן אשר עומד בדרך של ירייה או חרב נפגע ומד החיים שלו יורד בהתאם לעוצמת המכה. שחקן שמד החיים שלו עומד על 0 "מת" ולא יקבל יותר תורות משחק.

גבולות המגרש מוגדרים היטב ושחקן לא יוכל לחרוג מהגבולות, בנוסף שחקנים לא יכולים "לדרוך" אחד על השני, תנועה לתוך קיר או שחקן אחר תגרור חוסר תנועה של אותו שחקן באותו תור.

כל סימולציה של קרב מוגבלת במספר קבוע של תורות אשר יש לכל שחקן לשחק (fight duration).

עוד על מאפייני השחקן ראה את החלק על ה [אינדיבידואל](#) בהמשך.

בתמונה נלחמים הקבוצה האדומה והקבוצה וכחולה. השחקן הכחול מכה בחרב את האדום.



האפליקציה

תפריט התחלה:



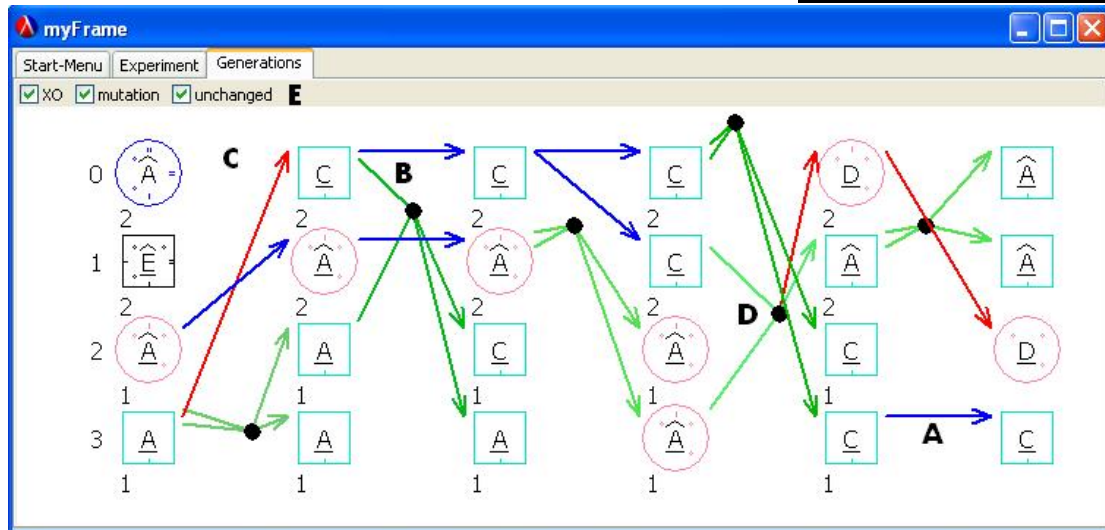
זה החלון הראשון שעולה בתוכנית, פה נקבע אופי הניסוי (החלקים בתמונה ממוספרים בהתאם להסברים):

1. רזולוציית המסך לפי המחשב שבו עובדים
2. סוג הניסוי והדרך בה ישתנה מבנה הניסוי ותוצאותיו (ראה הסבר על כל ניסוי)
3. דרך החישוב של איכות השחקן:
 - Life&Score / Score / Life (ראה [Fitness](#))
4. הדרך בה מתבצע ה Selection במעבר בין הדורות:
 - Fitness proportionate / Ranking (ראה [Selection](#))

5. פרמטרים כללים לניסוי ספציפי:

1. גודל האוכלוסיה עליה נבצע את הניסוי.
2. מספר הדורות, זהו תנאי העצירה של הניסוי.
3. מספר תורות לכל שחקן בקרב יחיד.

לשונית Generations:



- זוהי לשונית מעקב אחר התפתחות האוכלוסייה לאורך הדורות.
- בכל דור מופיעים כל השחקנים לפי הסדר בו התנהלו הקרבות.
- עבור כל שחקן מופיע ה [Avatar](#) (צלמית), הניקוד אותו קיבל (לפי סוג ניסוי ו fitness type), חץ נכנס עבור צאצא, וחץ יוצא עבור הורה
 - הערה: לשחקן חדש לא יהיו חצים נכנסים, לשחקן שלא נבחר לדור הבא לא יהיו חצים יוצאים.
- ישנם חצים בשלושה צבעים שונים:
 - כחול: הצאצא עבר כמו שהוא ללא שינוי המידע הגנטי. (ראה **A**)
 - אדום: הצאצא עבר מוטציה. (ראה **C**)
 - גווני ירוק: על שני הורים התבצע xo, מהם נוצרו שני צאצאים. (ראה **B**)
- הערה: גם לאחר xo יכולה להתבצע על הצאצאים מוטציה (ראה **D**)
- מאחר שברוב הניסויים האוכלוסייה הינה גדולה בהרבה מהדוגמא ישנה אפשרות להציג רק חלק מהחצים עבור מעקב יותר נוח (ראה **E**)

פאנל population:

- החלון הימני בלשונית Experiment, מוצג בו עבור כל דור בניסוי את האוכלוסייה הנוכחית.
- כאשר זוג או שני זוגות (תלוי בניסוי) משתתפים בקרב כרגע, יופיעו נקודות בצבעים שונים על מנת לציין באיזה קבוצה הזוג משתתף.
- לאחר סיום הסיבוב יקבע עבור כל זוג אם הוא המנצח, אם כן יופיע $\sqrt{\quad}$ ירוק, אחרת יופיע X שחור.
- עבור ניסוי Tournament לכל שלב בטורניר יופיעו סימנים נוספים בצד בהתאם לתוצאות הסיבובים (כמו בדוגמא).
- לחיצה על צלמית שחקן תיתן את המידע הגנטי של השחקן בחלונית load/save אשר בצד שמאל.

0		X
1		X
2		✓ ●
3		✓ ●
4		X
5		X
6		✓ ●
7		✓ ●

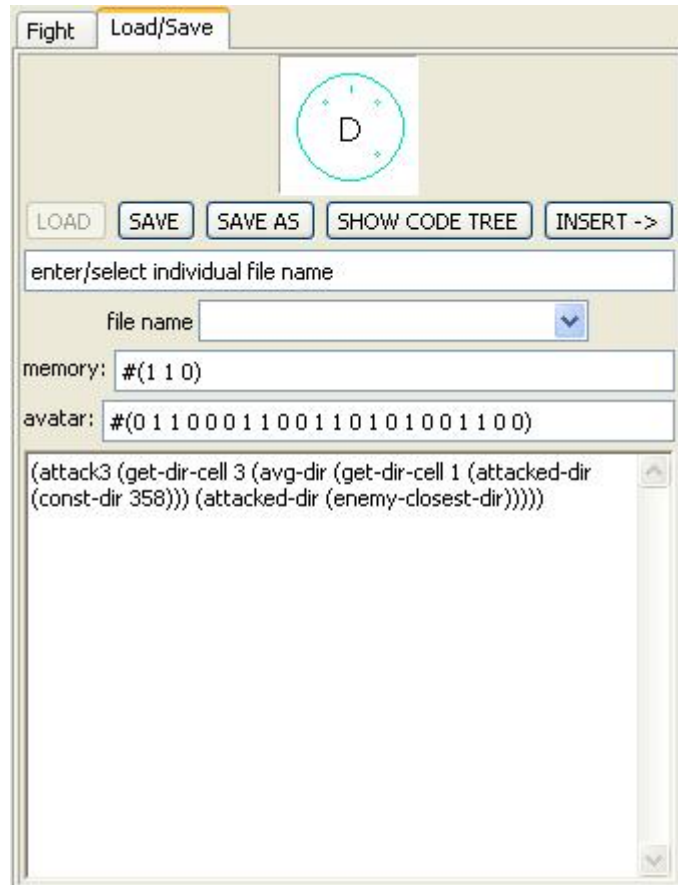
פאנל fight:

- חלופית זו נמצאת בצד שמאל בלשונית Experiment, מוצגים בה המשתתפים לפי קבוצות, עבור כל סיבוב קרבות.
- לצד כל שחקן יופיע הניקוד שצבר בסיום כל קרב. (ראה 1)
- בתחתית כל צלמית מופיע "מד הבריאות", המראה עד כמה השחקן נפגע בקרב הנוכחי (ראה 2)
- בקרבות של arena/arena-cll experts השונים יופיעו כקבוצת הכחולים.

RED TEAM	
	-70 1 680
	-440 -350
2	
BLUE TEAM	
	880 690
	140 -310

פאנל Load/Save:

- בפאנל זה אפשר ניתן לשמור/לטעון/לערוך אינדיבידואלים יחידים.
- פה ניתן לראות את המידע הגנטי של כל שחקן באוכלוסייה, או בבסיס הנתונים הקיים.
- שינוי בווקטור הצלמית יגרור שינוי מתאים בצלמית המופיעה בראש החלון.
- שחקן שיטען יופיע בדור הבא.
- שינויים במידע הגנטי לא ישפיעו על שחקן עד אשר הוא ישמר ויטען לדור הבא.
- שינויים שמתבצעים בזיכרון לאורך הקרב ישפיעו על ווקטור הזיכרון (לא בזמן אמת).



כפתורים:

- **LOAD:** לאחר שנבחר שם קובץ מתוך תפריט הגלילה לחיצה על הכפתור תציג את המידע הגנטי מתוך הקובץ.
- **SAVE:** לאחר שנבחר או נטען שחקן לחיצה על הכפתור תשמור את המידע הגנטי על קובץ בספרייה הייעודית. שם הקובץ יראה כך - `.ind[DD-MM]_[HH#MM#SS][SHAPE-LETTER].txt`
- **SAVE AS:** לאחר שנבחר או נטען שחקן לחיצה על הכפתור תשמור את המידע הגנטי על קובץ בספרייה הייעודית. שם הקובץ יהיה המחרוזת בשדה הטקסט מתחת לכפתורים.
- **SHOW CODE TREE:** לאחר שנבחר או נטען שחקן לחיצה על הכפתור תעלה חלון המציג את הקוד אשר בשדה הטקסט למטה בצורת עץ.
- **INSERT ->** לאחר שנבחר או נטען שחקן לחיצה על הכפתור תכניס את האינדיבידואל לרשימת המתנה לסיום הדור הנוכחי, כל השחקנים שהוכנסו יופיעו בדור הבא ראשונים באוכלוסייה.

:Control-Panel

- חלק זה בלשונית Experiment נותן שליטה על מהלך הניסוי.
- ניתן להוריד את מהירות הסימולציה על ידי שינוי סקלת ה Speed.
- מספר התורות בקרב ניתן לשינוי על ידי Duration.

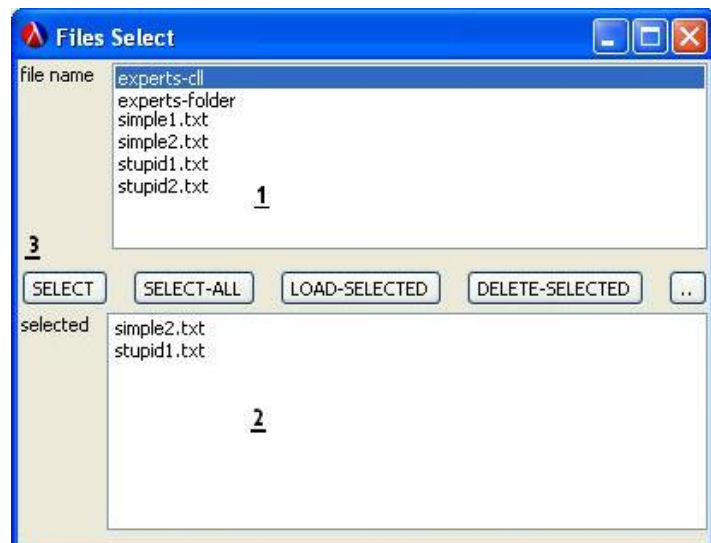


כפתורים:

- **SAVE-ALL**: לחיצה על כפתור זה תשמור את כל האוכלוסייה בספרייה נפרדת בבסיס הנתונים. שם הספרייה יראה כך - `save-all-[DD-MM]_[HH#MM#SS]`.
- **STOP**: עוצר את הניסוי בצורה נכונה כאילו מספר הדורות הסתיים.
- **Pause/Resume**: כפתורים אלה עוצרים ומחדשים את הניסוי.
- **LOADER**: ראה הסבר בהמשך.

חלון :LOADER

- חלון זה עולה כאשר לוחצים על LOADER ב control panel (ראה למעלה), או כאשר הניסוי הוא `arena/arena-cll` וצריך להגדיר Experts מתוך הספרייה.
- החלון הזה משמש בעיקר לפני הלחיצה על start- experiment על מנת להגדיר אוכלוסיה התחלתית ו Experts לניסויים הרלוונטיים.



1. רשימת הקבצים לבחירה.
2. רשימת הקבצים שנבחרו לטעינה.
3. כפתורי השליטה:

- **SELECT**: בוחר את הקובץ המודגש בחלון 1 ומכניסו ל-2, אם הקובץ הנבחר הוא ספרייה יופיעו הקבצים מהספרייה בחלון 1.
- **SELECT-ALL**: בוחר את כל הקבצים מחלון 1.
- **LOAD-SELECTED**: טוען את כל הקבצים מתוך חלון 2 לרשימת המתנה לדור הבא, וסוגר את החלון.
- **DELETE-SELECTED**: מוחק קובץ מודגש מחלון 2.
- **כפתור (..)**: חזרה לספרייה הראשית.

השחקן

אינדיבידואל:

האלמנט המרכיב את האוכלוסייה, והאובייקט המוצג בלוח המשחק.

חלקים עיקריים:

הורים/הורה:

לכל יצור (לפי שיטת הרבייה (...xo, mutation) הורה/הורים שממנו קיבל את המידע הגנטי שלו (אם היצור אינו מוגרל בצורה רנדומאלית).

גנום:

Code: עץ ב lisp-like סקריפט לפי הדקדוק המוגדר.
Memo: וקטור של n ביטים (בדרך כלל בין 3-5).
Avatar: וקטור של 22 ביטים עבור המראה הייחודי של השחקן (למראה אין משקל בביצועים).

מראה האינדיבידואל – Avatar:

התפתחות מראה האינדיבידואל מתבצעת על ידי התהליך האבולוציוני במקביל להתפתחות ההתנהגות. מראה האינדיבידואל נקבע על ידי חלק מהגנום המיוצג על ידי bit-string. חלקים שונים ב bit-string אחראיים על אספקטים שונים במראה, כגון צבע, צורה וכו'.
Bit-string באורך 22:



0 1 1 - 0 - 0 0 0 0 - 1 - 0 - 0 0 0 0 - 1 1 - 0 1 - 1 0 - 0 1



0 0 1 - 1 - 1 0 0 0 - 1 - 0 - 0 0 0 1 - 1 1 - 0 1 - 1 0 - 0 1



הדקדוק:

הדקדוק על פיו גוזרים תוכנית מחשב המהווה את המוח של היצור, ומשמשת לקביעת מהלכיו. הדקדוק משמש בעת ביצוע XO/mutation ככלי לשמירת ההתאמה בין הטיפוסים. הקוד כתוב בצורת Lisp כך שניתן להראות את הקוד כעץ גזירה של הדקדוק.

הדקדוק מגדיר מה שנקרא ¹⁴STGP (Strongly-Typed-GP) הtypes העיקריים:

- If:** בדיקה בוליאנית ופעולות
- Command:** נגזר לפעולות התקפה תזוזה, שינוי תאי זיכרון ובדיקות בוליאניות
- Set:** פעולות השמה של ערכים בתאי זיכרון, ובתאי כיוון
- Numeric:** קבועים או פעולות חשבון על מרחקים בלוח
- Direction:** כיוון במעלות, פעולות חשבון על כיוון, גישה לתאי כיוון
- Boolean:** true/false, פעולות לוגיות - על תאי זיכרון, מצבי סביבה, וסנסורים.
- Const:** קבועים של מרחק/כיוון/זווית.

Program → Command;
 Command → | If | Set | Move | Attack | Move&Attack
 If → (Boolean Command Command)
 Set → (Set-mem Command) | (Set-dir Command)
 Move → Direction
 Attack → (Attack-Type Direction)
 Move&Attack → (Move Attack)
 Boolean → And | Or | Not | < | > | = | ...

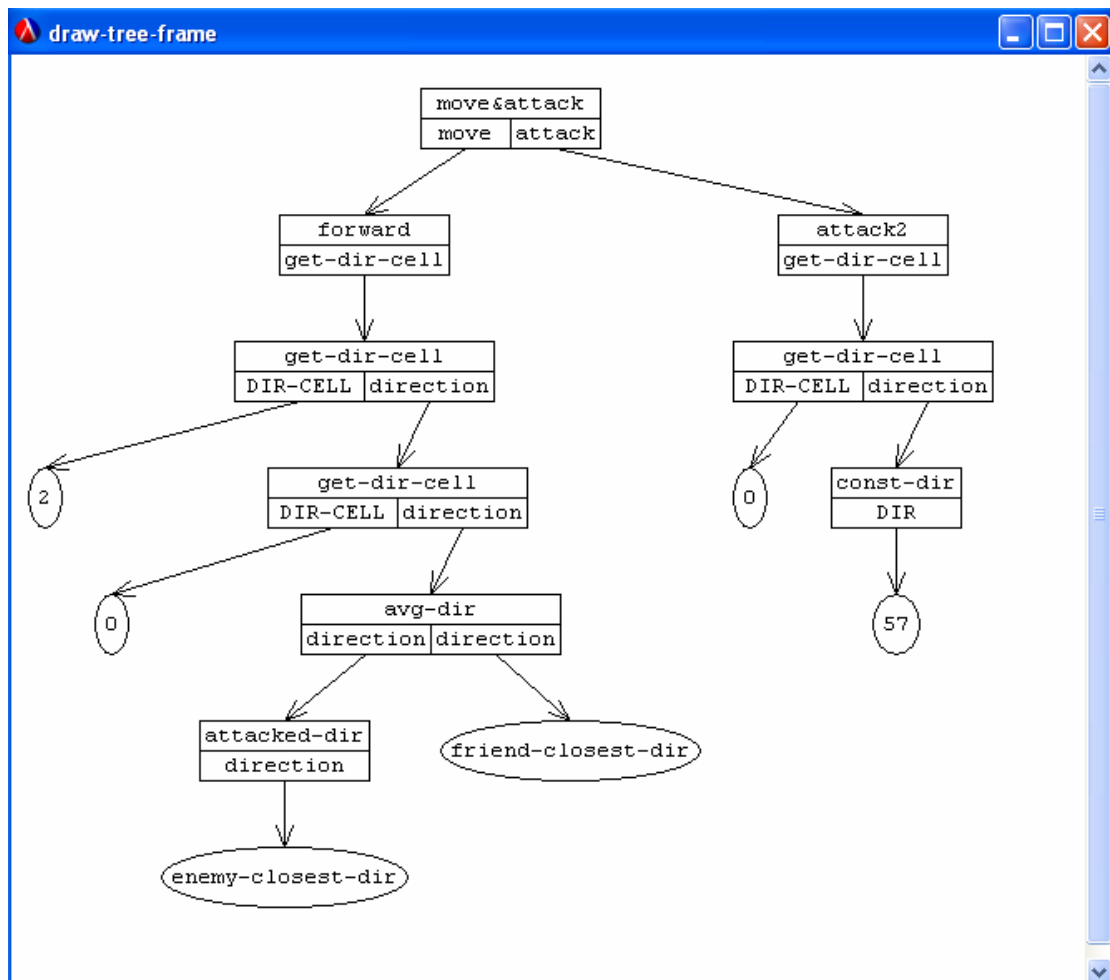
הערה: הדקדוק המלא הרבה יותר מורכב, הצגנו פה תמצות של כללי הגזירה.
 (על מנת לראות את כללי הגזירה המלאים ראה את [Appendix A](#))

דוגמא:

בדוגמא, ניתן לראות קוד של שחקן, בחלון למטה ניתן לראות את עץ התוכנית של קוד זה,
 כפי שהאפליקציה ייצרה.

```

(move&attack
  (forward
    (get-dir-cell 2
      (get-dir-cell 0
        (avg-dir (attacked-dir (enemy-closest-dir)
          (friend-closest-dir))))))
    (attack2 (get-dir-cell 0 (const-dir 57))))
  
```



הערה: על מנת להציג את עץ התוכנית, בלשונית experiment בחר load/save לאחר שבחרת שחקן מקובץ או מהאוכלוסייה לחץ על SHOW CODE TREE.

הסבר קצר על ריצת התוכנית:

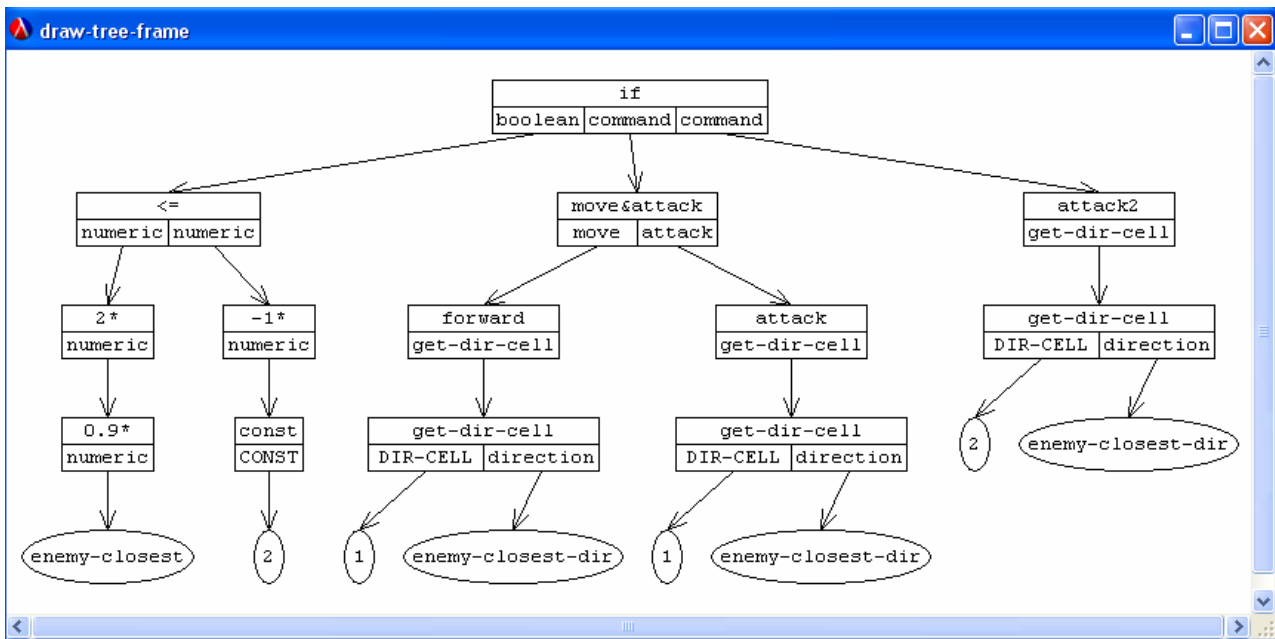
התוכנית מבצעת התקפה ותזוזה.

התזוזה מורכבת מבדיקת תא כיוון מספר 2, אם התא מלא יזוז בכיוון אשר בתא, אם התא ריק ינסה את תא 0, אם התא מלא יזוז בכיוון אשר בתא.

אחרת יחשב ממוצע בין הכיוונים של החבר הכי קרוב ובין הכיוון שממנו הותקף לאחרונה, ואם לא הותקף יחשב את הממוצע לפי האויב הכי קרוב.

ההתקפה היא ירי רגיל (לפי attack2) ומורכבת מבדיקת תא 0, אם התא מלא יתקוף בכיוון אשר בתא אחרת יתקוף בכיוון קבוע בזווית של 57

דוגמא נוספת לתוכנית:



מרכיבי התהליך האבולוציוני

תנאי עצירה

לפי מספר דורות.

Xo

בחירת תת-עץ מהאינדיבידואל ראשון. על פי ה"טיפוס" של תת העץ נבחר תת-עץ מתאים מהאינדיבידואל השני ונעתיק אותו לראשון. באותו האופן נבחר תת-עץ באינדיבידואל השני ונעתיק במקומו תת-עץ מהאינדיבידואל הראשון המתאים בטיפוס.

mutation

בחירת תת עץ מאינדיבידואל. גידול תת עץ חדש על פי גזירה של הדקדוק החל מה-NonTerminal או Terminal המתאים לתת העץ.

Self-XO

סוג של מוטציה המתבצע על ידי שימוש באופרטור ה- XO על אותו האינדיבידואל. כך יכול להתבצע שיכפול של תתי עצים מעץ התוכנית, או השמטה של חלק מעץ התוכנית.

Fitness

נקבע על פי סוג ניסוי (ראה [סוגי ניסויים](#)).

דרך החישוב של איכות השחקן:

1. Life: בטורניר נקבע הפיטנס על ידי השלב בטורניר אליו הגיע השחקן.
2. Score: הפיטנס נקבע על ידי סכום הניקוד שמקבל השחקן בקרבות באותו הדור. פגיעה באויב מעניקה נקודות ואילו פגיעה בחבר מורידה נקודות, בנוסף השרדות ומוות של השחקן בקרב משפיע על הניקוד.
3. Life&Score: שילוב של שני החישובים הקודמים: מיקום בטורניר כפול הניקוד (רלוונטי בעיקר ל-Tournament).

Selection (משתנה ידנית בחלון הפתיחה)

הדרך בה מתבצע ה Selection במעבר בין הדורות:

1. Ranking: לפי המיקום של השחקן במיון האוכלוסיה לפי fitness. אם יש שחקן שהוא הרבה יותר טוב משאר האוכלוסייה או יותר טוב במעט משאר האוכלוסייה, הדבר לא יבוא לידי ביטוי בסיכוי בחירתו לדור הבא.
2. Fitness proportionate: לפי היחס בין fitnesses של היצורים כך יהיה סיכוי בחירתם לדור הבא.

סוגי ניסויים

ציון הכשירות (fitness) מחושב בהתאם לניסוי אותו אנחנו מריצים.

טורניר:

טורניר אלימינציה. כל האוכלוסייה מתחלקת לזוגות של קבוצות המתחרות ביניהן. הקבוצה המנצחת מבין כל שתיים עולה לשלב הבא בטורניר, עד שנשארת רק קבוצה אחת. ציון הפיטנס ניתן לפי השלב בטורניר אליו הגיעה הקבוצה, בחזקות של שתיים: 1, 2, 4, 8 וכו'.

טורניר מורכב משלבים, בהם כל זוג באוכלוסייה נלחם סיבוב אחד בזוג אחר. בכל סיבוב זוג מהאוכלוסייה נלחם מספר קרבות (בדרך כלל 3) מול זוג אחר באוכלוסייה. בכל קרב עבור כל תור תבצע אבולוציה של קוד השחקנים המשתתפים בקרב. מספר התורות נקבע על פי *fight duration*. (הרצת התורות נראית כתנועה רציפה). המנצח בסיבוב הוא זה שניצח ביותר קרבות ויעלה לשלב הבא בטורניר, המפסיד לא ימשיך והציון שלו בטורניר יהיה 2 בחזקת השלב אליו הגיע.

שיטה זו להתפתחות היא על ידי מתן פיטנס פנימי, כלומר יחסית לחברי האוכלוסייה האחרים.

:Arena

בניסוי זה נבחרת קבוצה של "מומחים". כל קבוצה באוכלוסייה נלחמת מול קבוצת המומחים. ציון הכשירות של כל קבוצה נקבע לפי כמה טוב התמודדה מול המומחים. שיטה זו להתפתחות היא על ידי מתן פיטנס חיצוני, כלומר יחסית ליצורים אחרים שהם לא מתוך האוכלוסייה.

ניסוי זה מורכב מסיבובים של כל זוג באוכלוסייה מול זוג של שחקנים ממאגר המומחים אשר מוגדרים מראש לניסוי.

המומחים נבחרים בשיטת ה *round robin* כך שכל מומחה נבחר מספר שווה של פעמים. (הערה: בניסויי arena שערכנו המומחים היו תמיד שני מומחים קבועים).

כל סיבוב מורכב ממספר קרבות כאשר מספר זה נקבע בהתאם ל *fitness* הממוצע של הדור הקודם. ככל שהממוצע עולה, עולה מספר הקרבות כדי להשיג דיוק רב יותר בהערכת ה *fitness*.

:Arena CLL

(*CLL*^[3] – Concurrent Layered Learning)

בקצרה על *Layered Learning*: שיטה זו באה לבדוק האם "אימון" היררכי (כלומר בשלבים) של השחקנים יביא לתוצאות יותר טובות, המוטיבציה היא שבכל שלב "ילמד" השחקן תכונה אחת ספציפית. *CLL*: הוא *Layered Learning* כך שהשלבים הם מתבצעים במקביל ולא אחד אחרי השני.

בניסוי *Arena CLL* נבחרת קבוצה של "מומחים" ברמות שונות, חלק יותר טובים וחלק פחות טובים. כל קבוצה באוכלוסייה נלחמת מול כל קבוצת המומחים. דבר זה מאפשר התפתחות בשלבים. בהתחלה היצורים ילמדו לנצח את המומחים הפחות טובים ואחר כך את המומחים הייחודיים יותר טובים בהדרגה. נשתמש בשיטה זו על מנת לנסות להאיץ את תהליך ההתפתחות או להגיע לאיכות שחקן גבוהה יותר.

ניסוי זה מורכב מסיבובים של כל זוג באוכלוסייה מול כל זוג של שחקנים ממאגר המומחים אשר מוגדרים מראש לניסוי.

(הערה: בניסויי *Arena CLL* שערכנו היו תמיד ארבעה זוגות קבועים של מומחים).

כל סיבוב מורכב ממספר קרבות מול כל זוג מומחים כאשר מספר זה נקבע בהתאם ל *fitness* הממוצע של הדור הקודם. ככל שהממוצע עולה, עולה מספר הקרבות כדי להשיג דיוק רב יותר בהערכת ה *fitness*.

אופטימיזציות והנדסת תוכנה

:Analyzer

התוכנית אשר נמצאת במוח היצור מוערכת (מתבצעת) כ- 400 פעמים בכל קרב עבור כל יצור. בכל ניסוי מתבצעים עשרות אלפי קרבות. דבר זה אומר שחשוב מאוד שהערכת התוכנית אשר נמצאת במוח תהיה מהירה ביותר. לשם כך הכנסנו את ה-analyzer. תפקיד ה-analyzer הוא לקמפל את עץ התוכנית לפונקציה (ביטוי למבדא מוערך). לשם הרצת הקרב נפעיל את הפונקציה המקומפלת של כל יצור מספר פעמים כאורך הקרב. דבר זה חוסך את הפעולה היקרה של ביצוע parsing לעץ מאות פעמים בכל קרב, ולכן מהירות הריצה משתפרת רבות.

:Garbage Collection

עבור אובייקטים גדולים בתוכנית, אשר מוקצים ונהרסים בתדירות גבוהה (למשל אובייקטים מסוג אינדיבדואל, גנום, ירייה ועוד) השתמשנו בניהול מאגר אובייקטים עבור כל סוג אובייקט. אובייקט אשר הסתיים השימוש בו הוכנס למאגר. כאשר נדרשנו ליצר אובייקט חדש, בדקנו האם קיים אובייקט פנוי במאגר. אם כן, מחזרנו אותו על ידי אתחול בערכים המתאימים (לא מוקצה זיכרון נוסף). אם לא, אז אובייקט זה נוצר מחדש (מוקצה זיכרון חדש). אופי התוכנית הוא כזה שלאחר זמן מסוים בריצתה התוכנית כמעט ולא תקצה זיכרון נוסף ורק תמחזר אובייקטים מהמאגר.

המנגנונים של ה Analyzer וה GC שיפרו משמעותית את זמן הריצה וכמות הזיכרון הנדרשת להרצת האפליקציה. שיפור זה בעל חשיבות גבוהה מאחר וסוג כזה של תהליך ידוע בתור זולל משאבים. מאחר ואחת המטרות של הפרויקט הייתה ליצור אפליקציה ידידותית למשתמש ביתי, בעזרת השיטות האלו הצלחנו להריץ את הניסויים הכבדים ביותר על כל מחשב אפילו נייד בלי בעיות.

:Module System

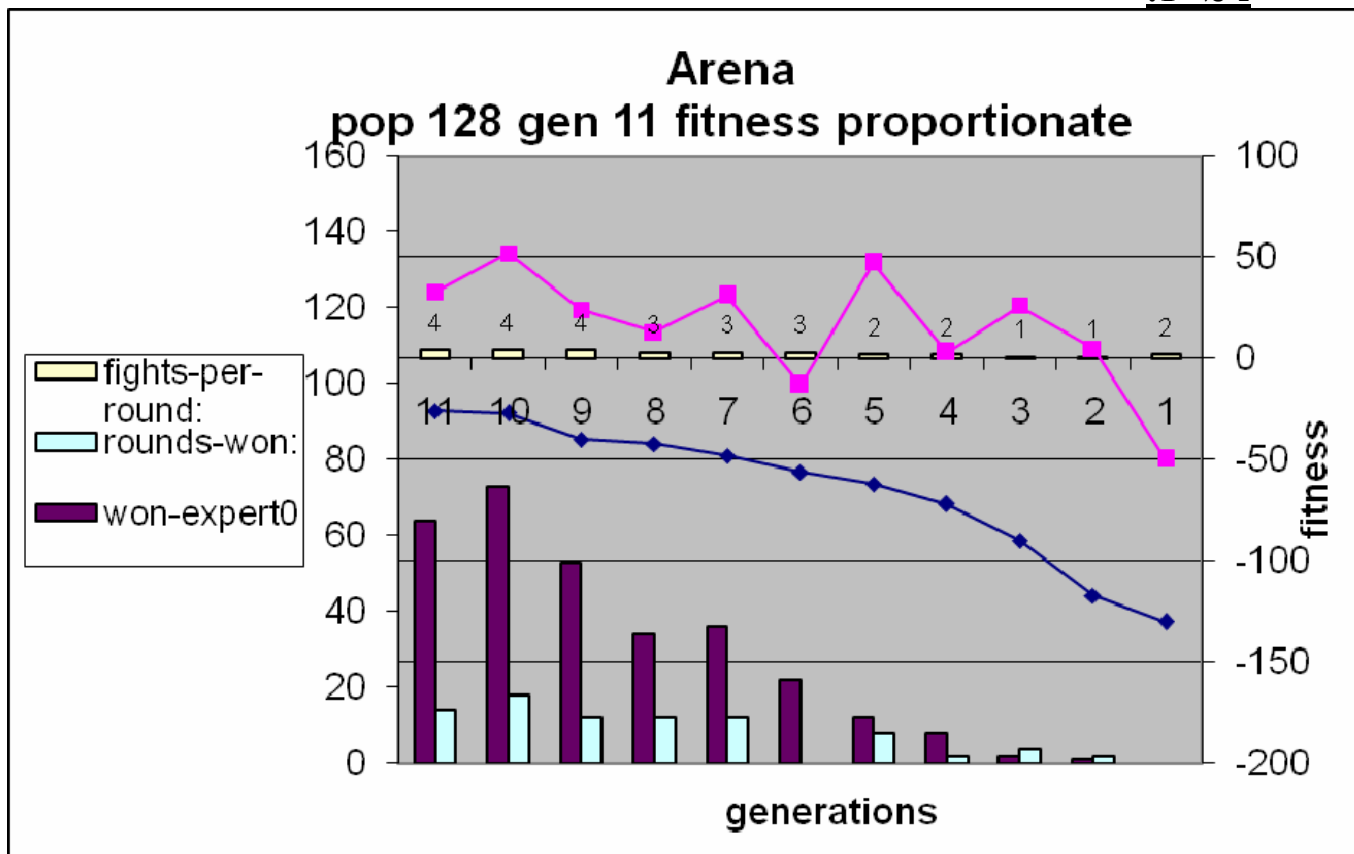
מאחר והשפה בה כתבנו היא Plt Scheme שהיא שפה פונקציונאלית ולא מונחית עצמים ממשנו בעצמנו אובייקטים בשיטת message passing ומנגנון של מחלקות (בסגנון של package או static class) אשר מגדיר מרחב שמות נפרד לכל מחלקה. בעזרת מנגנון זה ניתן לייבא או לייצא משתנים/פונקציות בין המחלקות. שיטת המחלקות מתבצעת באופן יעיל מאחר ולכל משתנה/פונקציה מועברת ונשמרת כתובת לקסיקוגרפית והשימוש באותו משתנה/פונקציה הוא בזמן קבוע $O(1)$.

ניסויים

הקדמה

בחלק זה נתאר את הניסויים שערכנו, נציג את התוצאות שלהם, וננסה לנתח ולהסיק מסקנות. הגרפים מוצגים בפורמט הבא: ציר ה-y הנמצא בצד ימין מייצג פיטנס ואליו משויכות עקומת הפיטנס הכי טוב והפיטנס הממוצע. בעמודות מוצגות מספר הקרבות בסיבוב (fights-per-round). לציר ה-y המשני הנמצא בצד שמאל משויכות העמודות שמתחילות מתחתית הגרף ומייצגות את כמות הניצחונות מול כל קבוצת מומחים.

ניסוי 1:



פרמטרים:

arena : סוג ניסוי

128 : אוכלוסיה

11 : מס' דורות

0.5 : xo

0.2 : אחוז מוטציה: מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo

fitness proportionate : Selection

מומחים: קבוצה אחת של שני מומחים. אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

תוצאות הניסוי:

1. אנו רואים עלייה של הפיטנס הממוצע ושל הפיטנס הכי טוב.

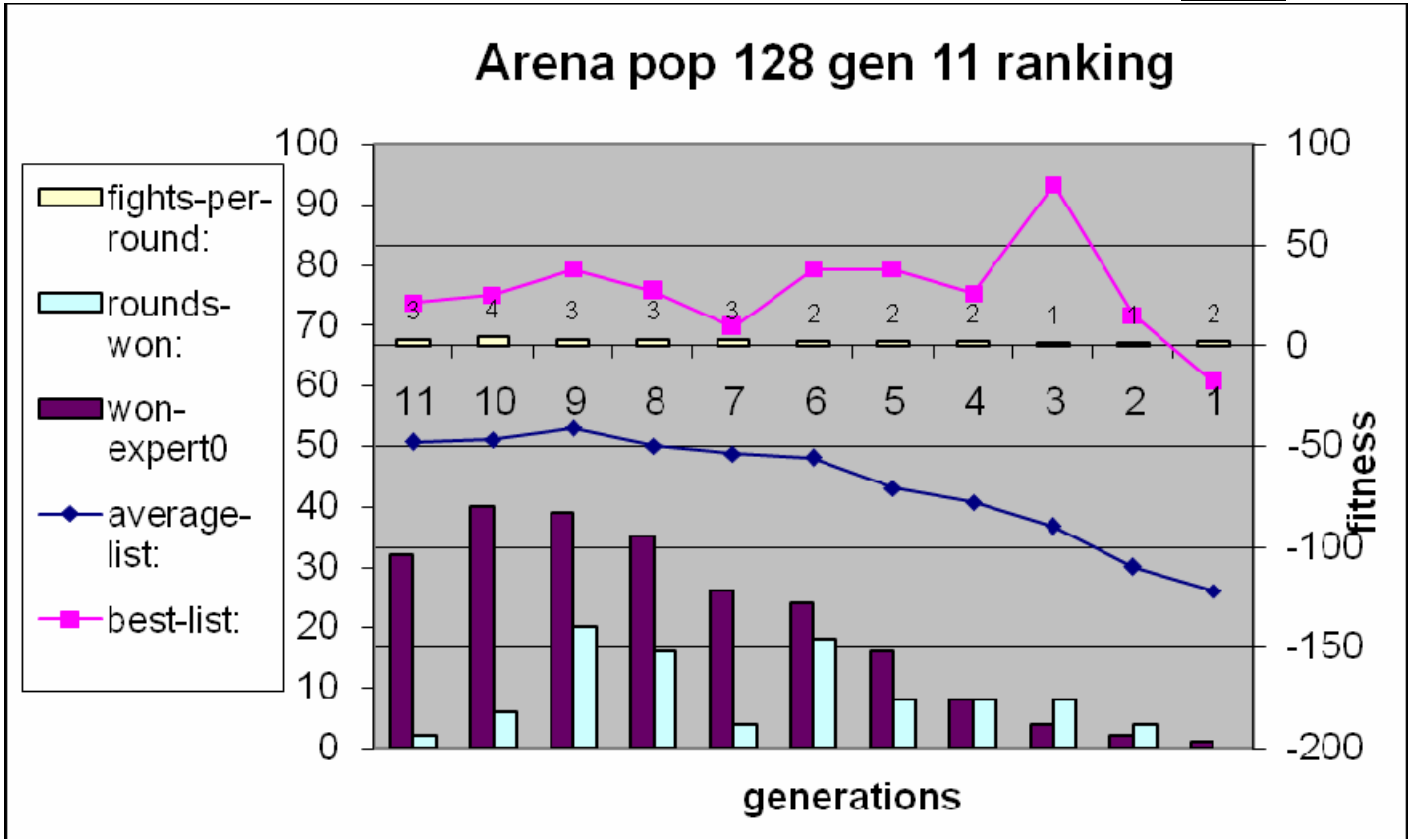
2. אנו רואים עלייה במספר הניצחונות מול המומחים.

3. פיטנס ממוצע מגיע ל -26 ופיטנס הכי טוב מגיע ל 52.

ניתוח ומסקנות מהניסוי:

אנו שמים לב שבמעברים למספר קרבות גדול יותר (מ-1 ל-2 ומ-2 ל-3) הפיטנס הכי טוב יורד, למרות שהפיטנס הממוצע עולה. אנו מסיקים שדבר זה אומר שהערכת הפיטנס נעשית מדויקת יותר ויש פחות טעויות בהערכה הנובעות ממקריות.

ניסוי 2:



פרמטרים:

סוג ניסוי: arena
 אוכלוסיה: 128
 מס' דורות: 11
 אחוז xo: 0.5

אחוז מוטציה: 0.2 מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo

Selection: ranking

מומחים: קבוצה אחת של שני מומחים. אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

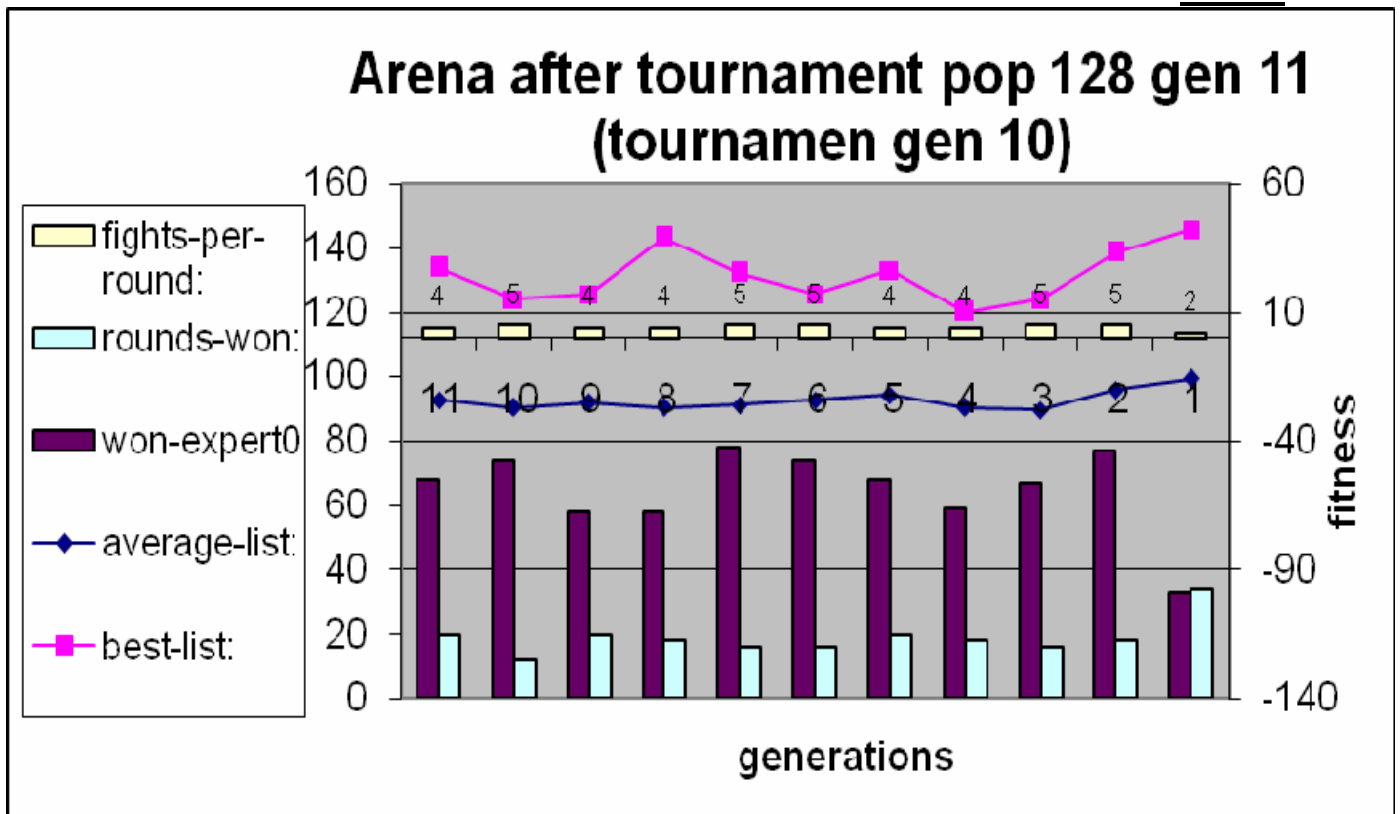
תוצאות הניסוי:

1. אנו רואים עליה של הפיטנס הממוצע ושל הפיטנס הכי טוב. ירידה קטנה בדורות האחרונים.
2. אנו רואים עליה במספר הניצחונות מול המומחים. ירידה קטנה בדורות האחרונים.
3. פיטנס ממוצע מגיע ל -41 ופיטנס הכי טוב מגיע ל 25.
4. הערה: won-expert0 לא מנורמל ויש לחלק בכמות הקרבות.

ניתוח ומסקנות מהניסוי:

1. אנו שמים לב שבמעברים למספר קרבות גדול יותר (מ-1 ל-2 ומ-2 ל-3) הפיטנס הכי טוב יורד, למרות שהפיטנס הממוצע עולה. אנו מסיקים שדבר זה אומר שהערכת הפיטנס נעשית מדויקת יותר ויש פחות טעויות בהערכה הנובעות ממקריות.
2. אנחנו רואים שהתוצאות פחות טובות בצורה משמעותית מאותו הניסוי אשר התבצע עם fitness proportionate selection, שם התקבלו תוצאות של 26- עבור פיטנס ממוצע ו 52 עבור פיטנס הכי טוב.

ניסוי 3:



בניסוי זה הרצנו עשר דורות של arena, את היצורים שכבר התפתחו במשך 10 דורות של tournament.

פרמטרים:

- סוג ניסוי: arena אשר מורץ לאחר tournament
- אוכלוסיה: 128
- מס' דורות: 11
- אחוז xo: 0.5
- אחוז מוטציה: 0.2 מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo
- Selection: fitness proportionate
- מומחים: קבוצה אחת של שני מומחים. אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

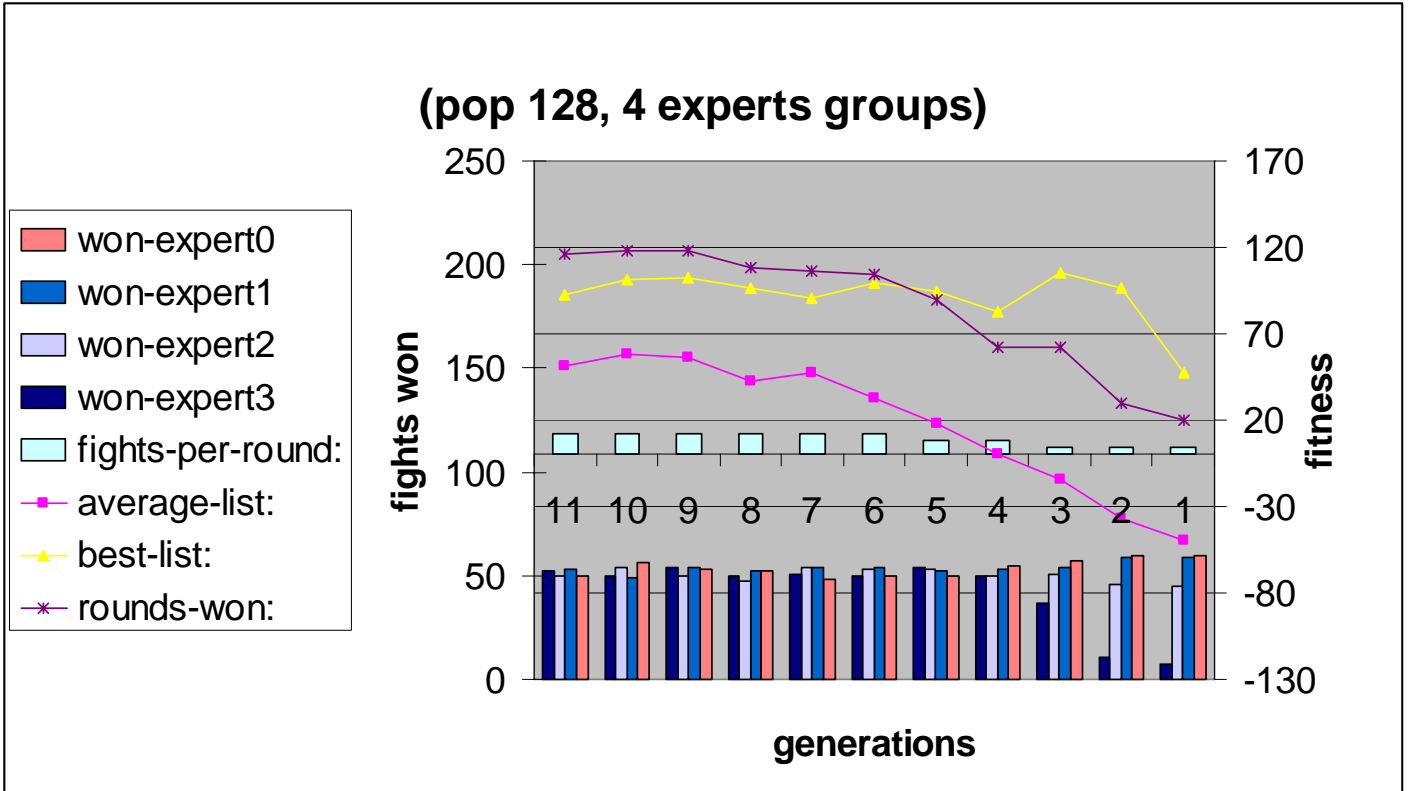
תוצאות הניסוי:

1. אנו רואים שאין עליה משמעותית של הפיטנס הממוצע ושל הפיטנס הכי טוב.
2. פיטנס ממוצע מתחיל מ 16- ולא עולה מעבר לזה במהלך הריצה. פיטנס הכי טוב מגיע ל 39 במספר קרבות גדול (4 קרבות ומעלה).

ניתוח ומסקנות מהניסוי:

1. אנו שמים לב שבמעברים למספר קרבות גדול יותר הפיטנס הכי טוב לפעמים יורד ולפעמים עולה.
2. אנחנו רואים שכבר בדור הראשון יש תוצאות טובות, יותר מאשר הניסוי הזה בו לא אימנו את היצורים בטורניר לפני.

ניסוי 4:



פרמטרים:

סוג ניסוי: arena cll

אוכלוסיה: 128

מס' דורות: 11

אחוז xo: 0.5

אחוז מוטציה: 0.2 מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo

Selection: fitness proportionate

מומחים: 4 קבוצות מומחים ברמת קושי עולה.

כמות קרבות: כמות הקרבות נקבעת על פי ה- fitness הממוצע. מתחת ל- fitness ממוצע של -30 מתבצע קרב אחד מול כל קבוצת מומחים (זה אומר 4 קרבות), בין -30 ל- 10 שני קרבות מול כל קבוצת מומחים, ומעל 10 שלושה קרבות.

קבוצת המומחים מחולקת לארבע קטגוריות לפי רמת קושי עולה: stupid, simple, novice, experts.

Stupid: יורים בצורה רנדומאלית לכיוונים ל-4 כיוונים: למעלה, למטה, ימינה ושמאלה.

Simple: זזים במעגלים ויורים לכיוון התנועה.

Novice: יורים בהפסקות לאויב הכי קרוב וזזים לאויב/חבר הכי קרוב.

Experts: אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

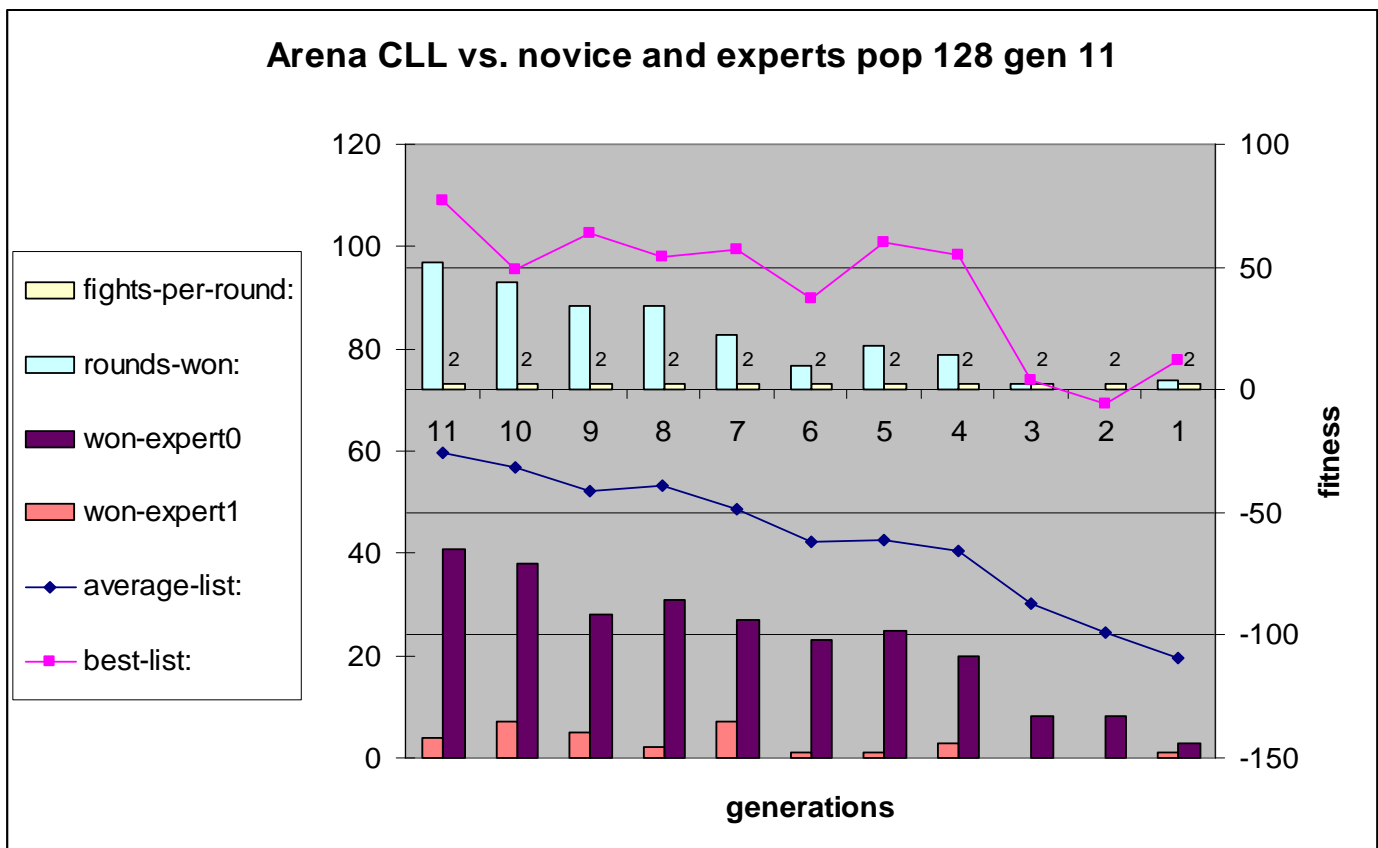
תוצאות הניסוי:

1. זהו ניסוי בסדר גודל גדול אשר לקח יותר מ-7 שעות להריץ.
2. אנו רואים עליה של הפיטנס הממוצע ושל הפיטנס הכי טוב (לא ניתן להשוות פרמטרים אלו לניסויים קודמים מאחר שבניסוי זה יש מומחים שחלקם אחרים מהמומחים בניסוי arena).
3. עליה הדרגתית בדורות הראשונים בכמות הקרבות המנוצחים מול המומחים בקטגוריות expert ו novice.
4. אין שיפור בקרבות המנוצחים מול המומחים בקטגוריות simple ו stupid.

ניתוח ומסקנות מהניסוי:

1. אנו שמים לב שבמעברים למספר קרבות גדול יותר הפיטנס הכי טוב יורד, למרות שהפיטנס הממוצע עולה. אנו מסיקים שדבר זה אומר שהערכת הפיטנס נעשית מדויקת יותר ויש פחות טעויות בהערכה הנובעות ממקריות.
2. אולי אין טעם להשתמש במומחים ברמות הקושי הנמוכות stupid ו simple כי אין שיפור בכמות הניצחונות מולם, דבר זה יכול להביא להורדה של זמן הריצה.

ניסוי 5:



פרמטרים:

arena cll סוג ניסוי:

אוכלוסיה: 128

מס' דורות: 11

אחוז xo: 0.5

אחוז מוטציה: 0.2 מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo

Selection: fitness proportionate

מומחים: 2 קבוצות מומחים ברמת קושי עולה.

כמות קרבות: כמות הקרבות נקבעת על פי ה- fitness הממוצע. מתחת ל- fitness ממוצע של -30 מתבצע קרב אחד מול כל קבוצת מומחים (זה אומר 2 קרבות), בין -30 ל 10 שני קרבות מול כל קבוצת מומחים, ומעל 10 שלושה קרבות.

קבוצת המומחים מחולקת לשתי קטגוריות לפי רמת קושי עולה: novice, experts. (הורדנו את הרמות החלשות (stupid, simple).

Novice: יורים בהפסקות לאויב הכי קרוב וזזים לאויב/חבר הכי קרוב.
 Experts: אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

תוצאות הניסוי:

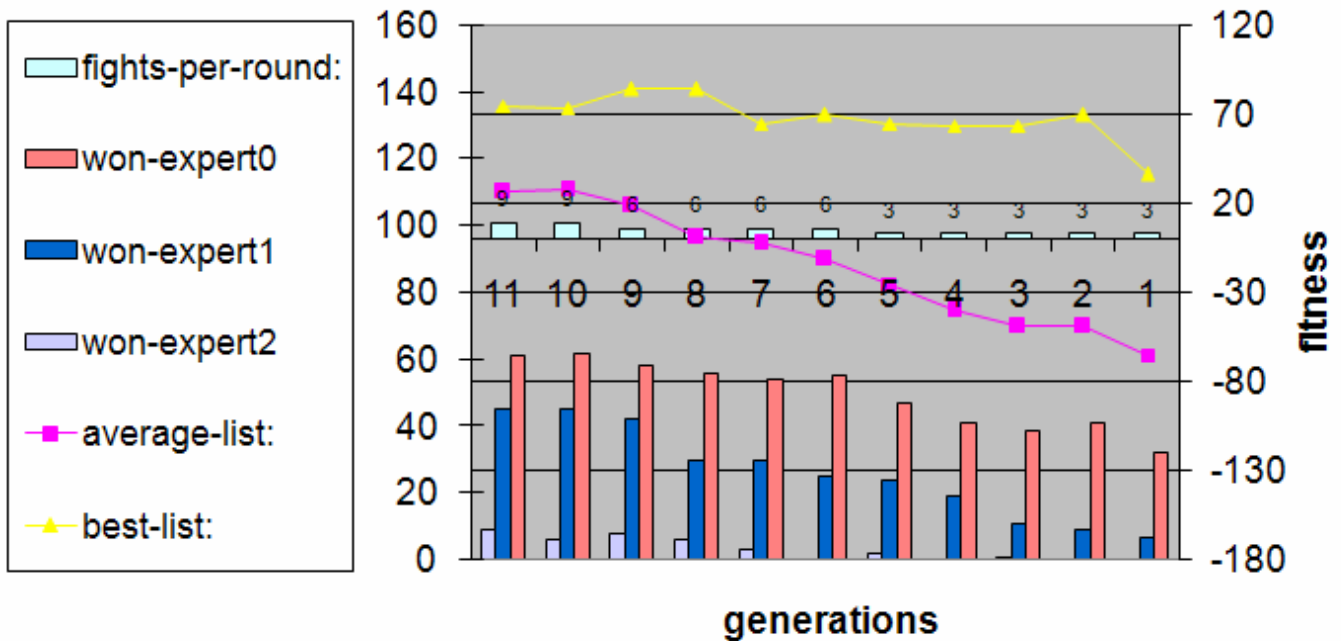
1. אנו רואים עליה של הפיטנס הממוצע ושל הפיטנס הכי טוב.
2. עליה הדרגתית בדורות הראשונים בכמות הקרבות המנוצחים מול המומחים בקטגוריות expert ו novice.
3. יש שיפור גדול בקרבות המנוצחים מול המומחים בקטגורית novice.
4. השיפור מול המומחים בקטגורית ה- expert קטן למדי.

ניתוח ומסקנות מהניסוי:

1. כנראה יש חשיבות לכך שיש מדרג באוכלוסיית המומחים והדבר תורם להתפתחות האבולוציה.

ניסוי 6:

Arena vs. simple novice expert pop 128 gen 11



פרמטרים:

arena cli סוג ניסוי:
 אוכלוסיה: 128
 מס' דורות: 11
 אחוז xo: 0.5
 אחוז מוטציה: 0.2 מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo
 Selection: fitness proportionate
 מומחים: 3 קבוצות מומחים ברמת קושי עולה.

כמות קרבות: כמות הקרבות נקבעת על פי ה- fitness הממוצע. מתחת ל- fitness ממוצע של 30- מתבצע קרב אחד מול כל קבוצת מומחים (זה אומר 3 קרבות), בין 30- ל 10 שני קרבות מול כל קבוצת מומחים, ומעל 10 שלושה קרבות.

קבוצת המומחים מחולקת לשלוש קטגוריות לפי רמת קושי עולה: simple, novice, experts. (הורדנו את הרמה החלשה (stupid)).

Simple: זזים במעגלים ויורים לכיוון התנועה.
 Novice: יורים בהפסקות לאויב הכי קרוב וזזים לאויב/חבר הכי קרוב.
 Experts: אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

תוצאות הניסוי:

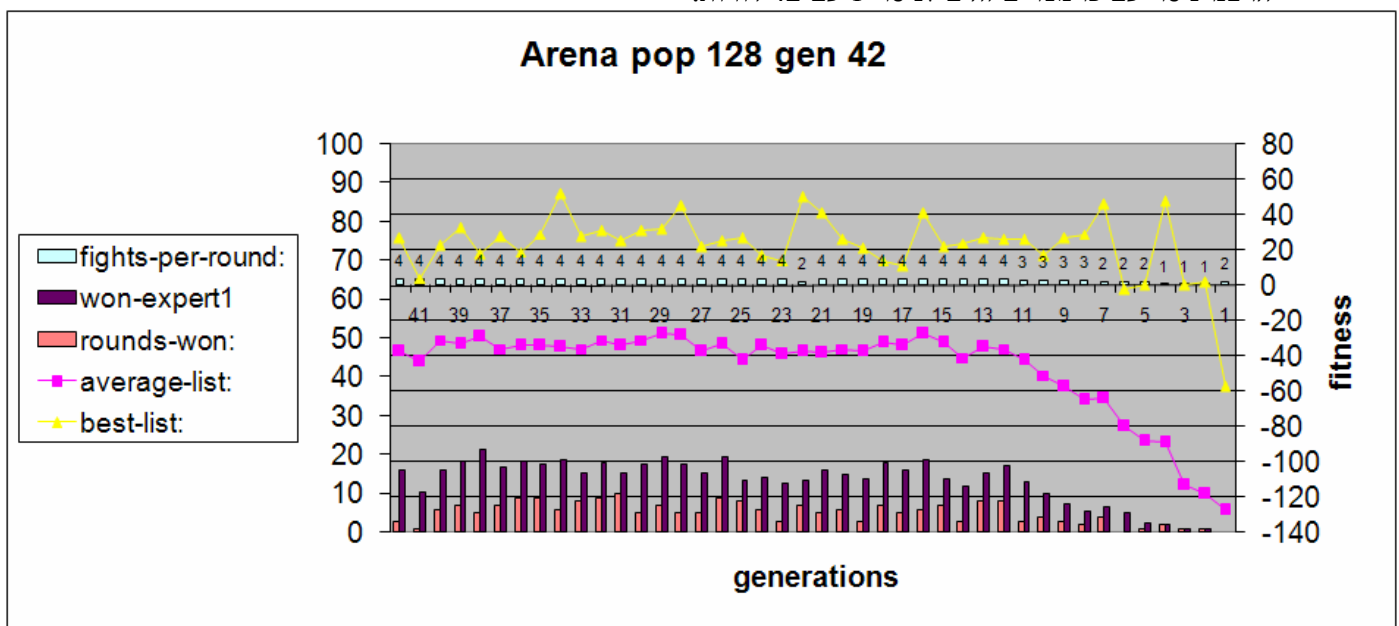
1. אנו רואים עליה של הפיטנס הממוצע ושל הפיטנס הכי טוב.
2. גרף ה-"פיטנס הכי טוב" יותר מונוטוני ופחות קופצני מאשר בניסויים הקודמים.
3. יש עליה בכמות הקרבות המנוצחים מול המומחים בקטגוריות simple ו novice ו expert.
4. העלייה הגדולה ביותר היא מול ה simple, עליה בינונית מול ה novice, ועליה קטנה יחסית מול ה experts.
5. השיפור מול קבוצת ה experts יותר גדול במקצת מאשר בניסוי הקודם בו לא הייתה קבוצת ה simple.

ניתוח ומסקנות מהניסוי:

1. ניסוי זה מחזק את ההשערה שכנראה יש חשיבות לכך שיהיו יצורים ברמות נמוכות בקבוצת המומחים.

ניסוי 7:

הרצנו ניסוי עם פרמטרים זהים לניסוי 1 עם 42 דורות.



פרמטרים:

arena: סוג ניסוי

128: אוכלוסיה

42: מס' דורות

0.5: xo אחוז

0.2: אחוז מוטציה: מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo

Selection: fitness proportionate

מומחים: קבוצה אחת של שני מומחים. אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

תוצאות הניסוי:

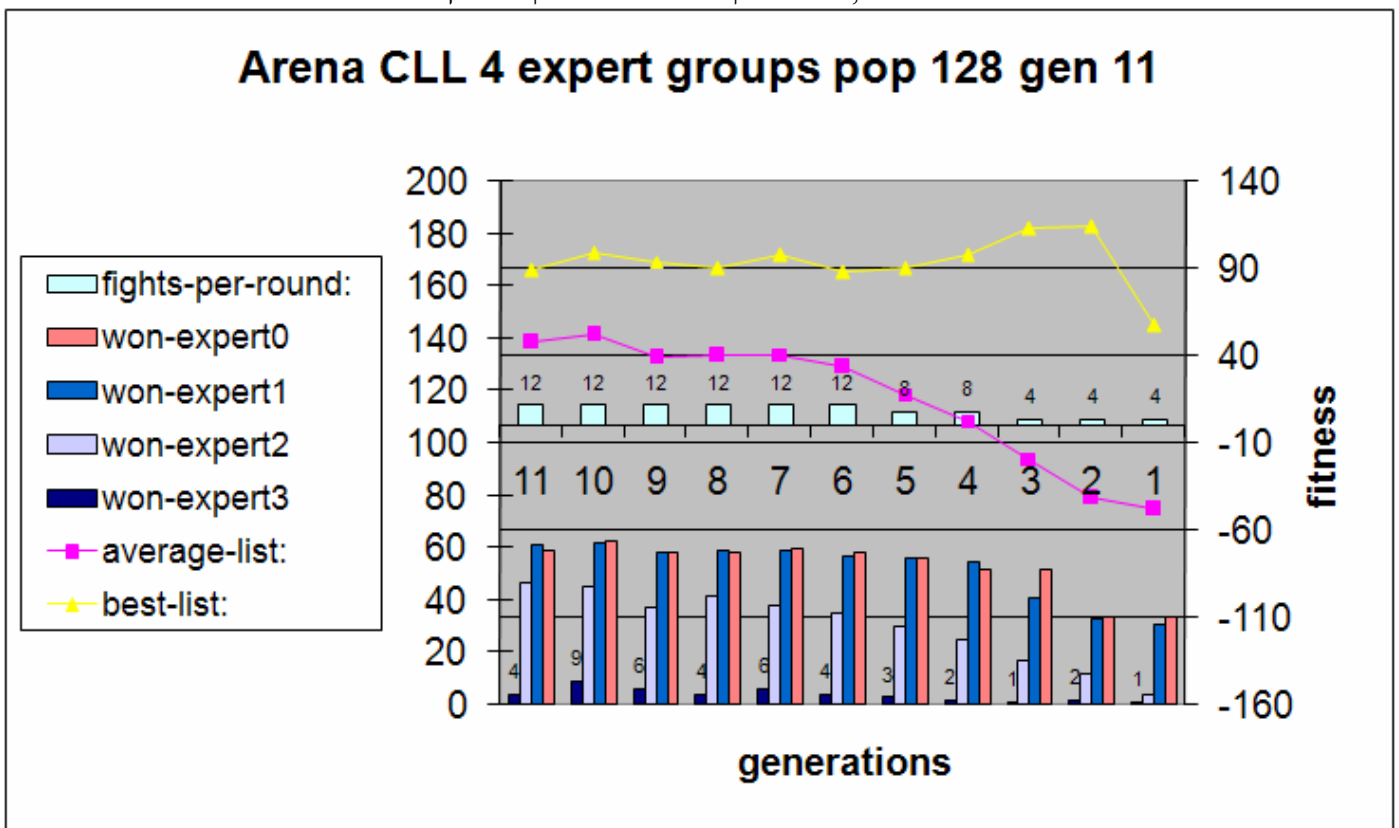
1. אנו רואים עליה של הפיטנס הממוצע ושל הפיטנס הכי טוב עד הדור ה-13. לאחר מכן לא עולים באופן משמעותי.
2. אנו רואים עליה במספר הניצחונות מול המומחים עד הדור ה-13.
3. פיטנס ממוצע מגיע ל -29 ופיטנס הכי טוב מגיע ל 52. (בניסוי 1: ממוצע -26, הכי טוב 52)

ניתוח ומסקנות מהניסוי:

1. אנו רואים כי לא השגנו תוצאות יותר טובות בניסוי זה מאשר בניסוי 1.
2. רוב השיפור באוכלוסיה בריצה זו התרחש ב 13 הדורות הראשונים.
3. נראה כי בניסוי מסוג זה קשה ליצורים להשתפר לאחר כ 13 דורות ולכן כנראה כדאי בהרצות ניסויים אלו להריץ את הניסוי מספר הרצות של כ- 15 דורות, ואת הריצה הטובה מביניהם להמשיך עוד דורות.

ניסוי 8:

הרצנו ניסוי עם פרמטרים זהים לניסוי 4, כדי לבדוק האם התוצאות שקיבלנו הן תוצאות מייצגות.



פרמטרים:

arena cli: סוג ניסוי

128: אוכלוסיה

11: מס' דורות

0.5: xo

0.2 מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo

Selection: fitness proportionate

מומחים: 4 קבוצות מומחים ברמת קושי עולה.

כמות קרבות: כמות הקרבות נקבעת על פי ה- fitness הממוצע. מתחת ל- fitness ממוצע של -30 מתבצע

קרב אחד מול כל קבוצת מומחים (זה אומר 4 קרבות), בין -30 ל 10 שני קרבות מול כל קבוצת מומחים,

ומעל 10 שלושה קרבות.

קבוצת המומחים מחולקת לארבע קטגוריות לפי רמת קושי עולה: stupid, simple, novice, experts.

Stupid: יורים בצורה רנדומאלית לכיוונים ל-4 כיוונים: למעלה, למטה, ימינה ושמאלה.
 Simple: זזים במעגלים ויורים לכיוון התנועה.
 Novice: יורים בהפסקות לאויב הכי קרוב וזזים לאויב/חבר הכי קרוב.
 Experts: אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

תוצאות הניסוי:

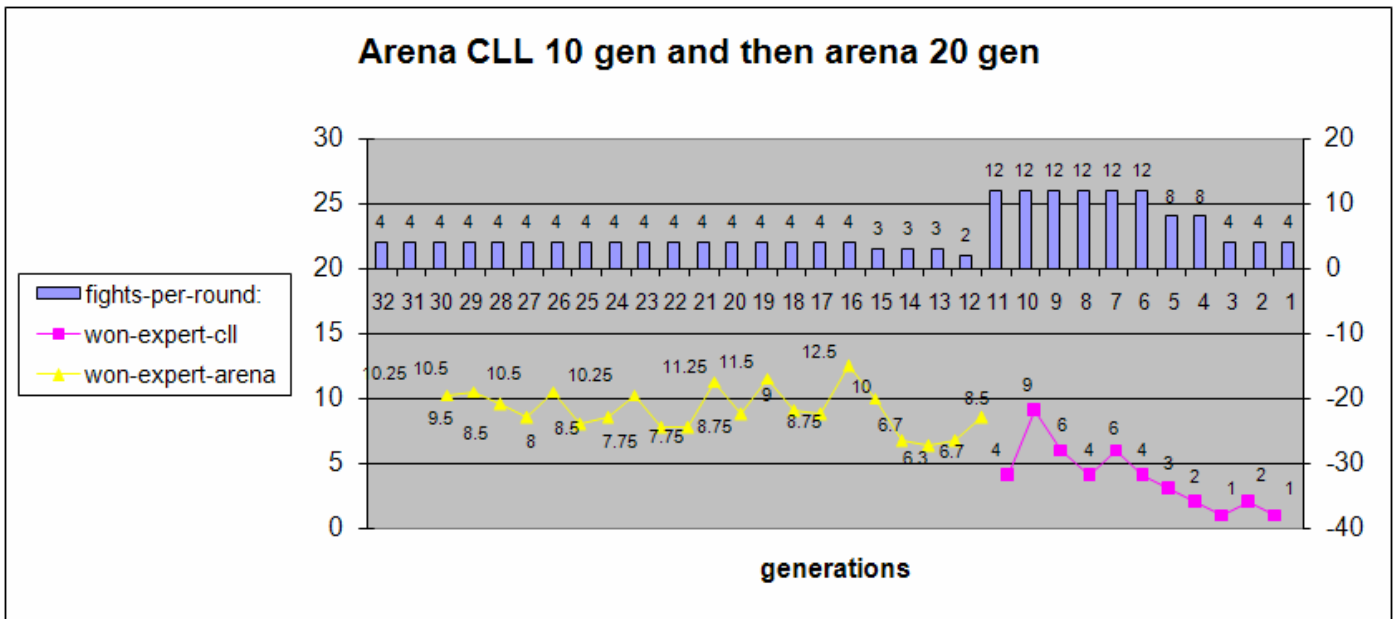
1. אנו רואים עליה של הפיטנס הממוצע. הפיטנס הכי טוב יורד במקצת בגלל העלייה בכמות הקרבות שמקטינה את השונות.
2. עליה הדרגתית בכמות הקרבות המנוצחים מול המומחים בארבע הקטגוריות.

ניתוח ומסקנות מהניסוי:

1. תוצאות פחות טובות מאשר בניסוי 4.

ניסוי 9:

בניסוי זה המשכנו את ניסוי 8 בעוד 20 דורות של arena.



פרמטרים של חלק ראשון:

מפורט בניסוי 8.

פרמטרים של חלק שני:

arena: סוג ניסוי

אוכלוסיה: 128

מס' דורות: 21

אחוז xo: 0.5

אחוז מוטציה: 0.2 מתוך סיכוי זה יש סיכוי של 0.33 ל- self-xo

Selection: fitness proportionate

מומחים: קבוצה אחת של שני מומחים. אחד יורה לאויב הכי קרוב אם החבר לא נמצא בדרך. השני רץ לאויב ושולף חרב.

תוצאות הניסוי:

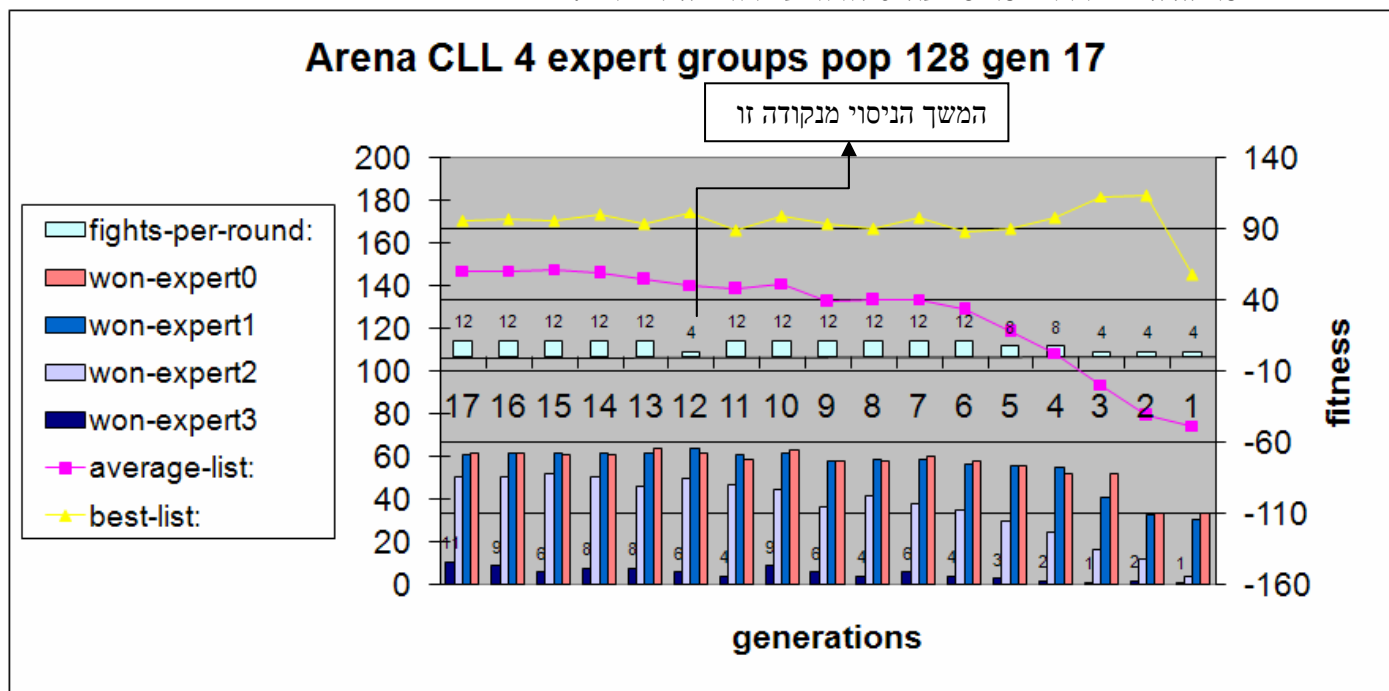
1. יש מגמה של שיפור במספר הניצחונות בחלק השני של ההרצה (arena) אולם עם הרבה קפיצות.
2. הפיטנס הממוצע מגיע ל -30 והפיטנס הכי טוב מגיע ל- 50 (לא מוצגים בגרף).

ניתוח ומסקנות מהניסוי:

1. הרצת arena cll לפני arena, במקרה הזה, לא הביאה לתוצאות גבוהות יותר.

ניסוי 10:

בניסוי זה המשכנו את ניסוי 8 בעוד 6 דורות עם אותם הפרמטרים.



תוצאות הניסוי:

1. יש שיפור קטן מאוד במספר הניצחונות, בפיטנס הממוצע ובפיטנס הכי טוב, בחלק השני של ההרצה.

ניתוח ומסקנות מהניסוי:

1. ראינו שהיה קשה לאוכלוסיה להתפתח מעבר לנקודה מסוימת בסוג הניסוי הזה.

מסקנות וסיכום

:Selection

תוצאות:

תוצאות הניסויים שהתבצעו עם fitness proportionate selection היו יותר טובות בצורה משמעותית מאשר תוצאות הניסויים עם ranking selection.

מסקנות:

כאשר אינדיבידואל קיבל ניקוד פיטנס שהיה גבוה באופן משמעותי משאר האוכלוסייה היה זה חשוב שיבחר באופן גבוה באותו היחס. כאשר אינדיבידואל אחד נבחר הרבה לדור הבא השונות באוכלוסייה לא קטנה משום שהאופרטורים הגנטיים בהם השתמשנו (mutation, xo, self-xo) שמרו על המגוון והשונות באוכלוסייה. על ידי כך לא נוצרה השתלטות של סופר-אינדיבידואל ולא נקלענו לנקודות מקסימום מקומי (ככל האפשר). ההשתפרות של האוכלוסייה כאשר השתמשנו ב- ranking selection הייתה איטית יחסית, משום שיצורים שהתבלטו בקרבות לא קיבלו יחס מקביל בבחירתם לדור הבא.

:טורניר

תוצאות:

בשיטת ניסוי זו קשה להעריך את טיב השחקנים ולהשוות לשיטות ניסויים אחרות. לכן הרצנו ניסוי arena שהיצורים ההתחלתיים בו היו כאלה שאומנו בעזרת הטורניר. התוצאות שהתקבלו היו טובות יותר באופן משמעותי ועלו על כל שיטה אחרת של ניסוי בה השתמשנו.

מסקנות:

שיטה טובה לאימון היצורים אך קשה להעריך את התוצאות. בניסוי בו הרצנו טורניר ואז arena לא היה שיפור של היצורים במהלך השלב השני של הניסוי (arena). אנו מסיקים מזה שהטורניר הביא את היצורים לרמת לחימה גבוהה, אשר הייתה יותר טובה מאשר זוג המומחים בהם השתמשנו בניסוי ה- arena, ולכן לא יכלו להמשיך להתפתח בניסוי זה.

:Arena

תוצאות:

זהו סוג ניסוי בו אפשר להתייחס ולהשוות נתונים שונים המתקבלים מהריצה: פיטנס ממוצע, פיטנס הכי טוב, ומספר ניצחונות מול המומחים. ראינו כי הניסוי הביא לשיפור בכל אותם הפרמטרים. ראינו שהניסויים הגיעו בדרך כלל לנקודה ממנה לא השתפרו כמעט, גם אם ממשיכים את הניסוי עוד מספר רב של דורות.

מסקנות:

אם אין השתפרות משמעותית בדורות הראשונים אין טעם להמשיך את הניסוי עוד דורות, אלא להתחיל ניסוי חדש. בשיטה זו בניגוד לטורניר איכות השחקנים נבחנת אל מול המומחים ולא מול האוכלוסייה, לכן ראינו התפתחות טובה של יכולת קבוצתית להתמודד נגד המומחים, עד רמה של ניצחון ברור. אך ככל שרמת האוכלוסייה עולה המומחים אינם משתפרים בהתאם מה שמביא את ההתפתחות הפוטנציאלית של האוכלוסייה לעצירה מסוימת. זאת בניגוד לטורניר שם ההתפתחות הפוטנציאלית תיאורטית אינה חסומה. התכונות שהתפתחו אצל השחקנים היו ספציפיות להתמודדות עם המומחים. כך התפתחו יצורים המתמחים בהתמודדות מול המומחים, ולא בהכרח מול כל יצור טוב אחר.

:Arena-CLL

תוצאות:

ראינו בהתחלה ניסוי מוצלח בו השתתפו מומחים מ-4 רמות קושי בו ההשתפרות הייתה גדולה מאוד. העלנו השערה שהמומחים ברמות הנמוכות אינם נחוצים להתפתחות. בניסויים הבאים של arena cll בדקנו את השערה זו על ידי השמטה של הרמות הנמוכות של המומחים. ראינו כי ההשתפרות הייתה קטנה יותר בניסויים אלה. כלומר, גם הרמות הנמוכות תורמות להתפתחות.

מסקנות:

כמו ב- arena גם בניסוי arena-cll ההתפתחות הפוטנציאלית חסומה, משום שהמומחים קבועים במהלך הריצה. כמו ב- arena גם כאן ניסויים אשר ההתפתחות בדורות הראשונים לא הייתה טובה, לא עזר

להריץ אותם עוד דורות. ראינו כי ב- arena-cll ניתן להגיע לתוצאות טובות בהרבה מ- arena, אך לא כל הרצה אכן מגיעה לתוצאות טובות.

אורך הניסוי:

גילינו כי בניסויים arena ו arena-cll בשלב מוקדם בריצה כדאי לבחון את רמת היצורים, ולא להמשיך את הניסוי אם ההתפתחות לא הייתה גדולה. ראינו כי ברוב הניסויים לא היה שיפור מורגש אחרי כ- 12 דורות.

Fitness ממוצע:

ראינו כי הפיטנס הממוצע הוא מדד טוב לאיכות האוכלוסייה, וקבענו לפיו את כמות הקרבות בסיבוב.

עליה במספר הקרבות:

השתמשנו בפיטנס הממוצע כמדד לקביעת מספר הקרבות, ובכך יש התאמה אוטומטית של הניסוי על פי ההתקדמות. כאשר האוכלוסייה היא בשלבים הראשונים של ההתפתחות נבצע מספר נמוך של קרבות (הפיטנס הממוצע נמוך). כאשר האוכלוסייה היא בשלבים יותר מתקדמים והפיטנס הממוצע גבוה נבצע מספר גבוה יותר של קרבות. כך נשקיע הרבה זמן בהערכה מדויקת של הפיטנס בדורות המתקדמים יותר, כי אז חשוב הדיוק בהערכה, בעוד שבשלבים המוקדמים יותר של ההתפתחות דיוק זה פחות חשוב, וזמן ההערכה יהיה קטן יותר.

התנהגות:

ראינו מספר סוגי התנהגויות שהתפתחו במהלך הניסויים. נסקור כאן את ההתנהגויות המעניינות.

- א. ירייה תוך כדי תנועה ושמירה על מרחק מהיריב.
- ב. תנועה סיבובית סביב עצמו או סביב שחקן אחר.
- ג. שחקן שזז מהר (מבלי להתקיף) יחמוק כמעט מכל ירי לעברו.
- ד. התנהגויות של בריחה מאויב יורה או התרחקות מאויב עם חרב.
- ה. שחקן אחד יורה בזמן שהשני בורח ומעסיק את האויבים.
- ו. החלפת נשק לסירוגין.
- ז. התאמת הנשק למרחק מהאויב.
- ח. שינויי התנהגות ברורים כאשר אחד השחקנים (אויב או חבר) "מתים".

אסטרטגיות משחק מוצלחות:

- א. שני השחקנים רצים אל האויבים עם חרב. התנהגות זו אולי לא הכי מתוחכמת אבל ניצחה ברוב המקרים את המומחים שלנו של arena.
 - ב. שחקן אחד מתקדם עם חרב ליריב והשני יורה. זאת התנהגות הזוהה למומחים שלנו אשר התפתחה על ידי התהליך האבולוציוני.
 - ג. אם השחקן רחוק מהיריב הוא עומד ויורה, ואם קרוב לאויב הוא מתקרב ושולף חרב.
 - ד. שחקן שיורה ומתקרב ושולף חרב כשהוא בטוח.
- (שילובים רבים מתוך ההתנהגויות הללו הביאו לתוצאות יפות אך לא ניתן היה בהכרח להבין בדיוק מה האסטרטגיה)

לסיכום:

- האפליקציה שכתבנו מהווה פלטפורמה להרצת ניסויים אבולוציוניים תוך כדי שליטה על פרמטרים שונים תוך כדי ריצת התוכנית.
- במהלך הריצה ניתן לעקוב ולראות את מהלך האבולוציה.
- ניתן להתאים תוכנית זאת עבור בעיות אחרות הדורשות פתרון אבולוציוני.
- הצלחנו לייצר בינה מלאכותית אשר התעלתה על שחקנים שכתבנו בעצמנו.
- בחנו והשוונו סוגי ניסויים שונים ושיטות לשיפור קצב ההתפתחות.
- למדנו והתנסו בהרבה תחומים: תכנות והנדסת תוכנה, תכנות אבולוציוני, הסתברות וגיאומטריה ומחקר.

רעיונות להמשך:

- כל אינדיבידואל באוכלוסייה הוא קבוצה. כך יהיה דגש על ההתאמה ושיתוף הפעולה של היצורים בקבוצה.
- כל אינדיבידואל באוכלוסייה יהיה קבוצה המורכבת מכמה יצורים. לכל קבוצה תהיה כמות קבועה של כסף, אותו היא יכולה לחלק בין היצורים השונים בקבוצה, בשביל כלי נשק וכלי הגנה. כך כל יצור בקבוצה יהיה בעל יכולות שונות הנקבעות על פי תהליך האבולוציה. גנום הקבוצה יהיה מורכב מכמה חלקים. חלק אחד יהיה אופן חלוקת הכסף בין היצורים. החלקים הנוספים יהיו המוח של כל אחד מן היצורים.
- לשנות את ניסוי ה- CLL כך שקבוצת המומחים יכולה לגדול או להתחלף בהדרגה בקבוצות טובות מן האוכלוסייה. בצורה זו תמיד יהיה אתגר חדש לאוכלוסייה, כי גם אוכלוסיית המומחים מתפתחת ומשתפרת. זהו סוג של קו-אבולוציה.

References

1. (GA) Evolutionay Algorithms, M. Tomassini
<http://www.cs.bgu.ac.il/~sipper/courses/papers/ga-ehw-final-SV.pdf>
2. (GP) GENETIC ALGORITHMS AND GENETIC PROGRAMMING, John R. Koza, <http://www.genetic-programming.com/c2003lecture1modified.ppt>
INTRODUCTION TO GENETIC PROGRAMMING, John R. Koza, July 2003, <http://www.genetic-programming.com/gecco2003tutorial.pdf>
3. (CLL) Concurrent Layered Learning, Shimon Whiteson and Peter Stone, <http://www.cs.utexas.edu/~pstone/Papers/bib2html-links/AAMAS03.pdf> ,
<http://citeseer.ist.psu.edu/575124.html> ,
<http://citeseer.ist.psu.edu/hsu02genetic.html>
4. (STGP) Montana, D. J. 1995. Strongly Typed Genetic Programming. In *Evolutionary Computation*. 3:2, 199–230. The MIT Press, Cambridge MA. ,
Strongly Typed Genetic Programming in Evolving Cooperation Strategies, Thomas Haynes, Roger Wainwright, Sandip Sen & Dale Schoenefeld, July 1995, <http://www.mcs.utulsa.edu/~rogerw/papers/Haynes-icga95.pdf>

Appendix A

הדקדוק המלא:

```
Program -> Command
Command -> Function | Terminal
Function -> If | ("begin" Set Command)
Terminal -> Move&Attack | Move | Attack
Attack -> ("Attack" Get-dir-cell) |
    ("Attack2" Get-dir-cell) |
    ("Attack3" Get-dir-cell)
Move -> ("forward" Get-dir-cell)
Move&Attack -> ("Move&Attack" Move Attack)
Numeric -> Numeric-Function | Numeric-Terminal
Numeric-Function -> + | - | x*-1 | x*0.9 |
    x*0.5 | x*2 | x*1.1
Numeric-Terminal -> Const | ("erc") |
    ("sensor" Direction) | ("friend-closest") |
    ("enemy-closest")
Direction -> Direction-Function |
    Direction-Terminal
Direction-Function -> ("Attacked-dir" Direction) |
    Get-dir-cell | ("dir+" Direction Dir-offSet) |
    ("avg-dir" Direction Direction)
Get-dir-cell ->
    ("Get-dir-cell" DIR-CELL Direction)
Direction-Terminal -> ("Const-dir" DIR) |
    ("friend-closest-dir") |
    ("enemy-closest-dir")
Dir-offSet -> Dir-offSet-Function |
    Dir-offSet-Terminal
Dir-offSet-Function ->
    ("dir-" Direction Direction) |
    ("Dir-off+" Dir-offSet Dir-offSet) |
    ("Dir-off*2" Dir-offSet) |
    ("Dir-off*-1" Dir-offSet) |
    ("Dir-off*0.5" Dir-offSet)
Dir-offSet-Terminal -> ("Dir-offSet" DIR-OFF)
Set -> Set-mem | Set-dir
Set-mem -> ("Set0" MEM-CELL) | ("Set1" MEM-CELL) |
    ("flip" MEM-CELL)
Set-dir -> ("Set-dir" DIR-CELL Direction) |
    ("add-to-dir-cell" DIR-CELL Dir-offSet)
Const -> ("Const" CONST)
```

```

Boolean -> Boolean-Function | Boolean-Terminal
Boolean-Function -> ("and" Boolean Boolean) |
  ("or" Boolean Boolean) | ("not" Boolean) |
  ("null?" Direction) | ("not-null?" Direction)
Boolean-Terminal -> ("<=" Numeric Numeric) |
  ("<" Numeric Numeric) |
  (">=" Numeric Numeric) |
  (">" Numeric Numeric) |
  ("=" Numeric Numeric) |
  ("mem-cell" MEM-CELL) |
  ("env-cond" ENV-CONDITION) |
  ("rand-bool") | ("true") | ("false") |
  ("is-sensor" Direction ENV-ELEMENT)
If -> ("If" Boolean Command Command) |
  ("If-Set" Boolean Set Command)
+ -> ("+" Numeric Numeric)
- -> ("- " Numeric Numeric)
x*-1 -> ("-1*" Numeric)
x*0.9 -> ("0.9*" Numeric)
x*0.5 -> ("0.5*" Numeric)
x*2 -> ("2*" Numeric)
x*1.1 -> ("1.1*" Numeric)

```

Constants:

```

MEM-CELL - memory cell number [0, size-1]
DIR-CELL - direction cell number [0, size-1]
DIR - direction number [1,360]
ENV-CONDITION - ['hit-wall, 'hit-somebody,
  'attacked]
CONST - distance [-5,5]
ENV-ELEMENT - ['enemy, 'friend, 'wall, 'null]
DIR-OFF - direction offset (1 30 90 180)

```

(erc):

זהו משתנה רנדומאלי ולא קבוע אשר בכל הרצה של התוכנית יקבל ערך חדש בטווח [-5,5], בעיקר מתייחס למרחק על לוח המשחק.