Description: Beskrivning: hd_vertikal_farg WORD.wmf

Thesis report

Master thesis in Microdata Analysis

Business Intelligence Program

School for Technology and Business Studies

Dalarna University

Description: StreckTiteln

All for one one for All

Author: Bamshad Shirmohammadi

Supervisor: Professor Arend Hintze

Examiner:

Dalarna University

791 88 Falun

Sweden

Tel 023-77 80 00

Subject:

Course code: MI4002

Points: 15/30hp

Date of the examination: 20XX-XX-XX

**Abstract**

**Contents**

[Introduction 1](#_Toc57969534)

[Background 1](#_Toc57969535)

[Purpose 3](#_Toc57969536)

[Literature review 3](#_Toc57969537)

[Material and Methods 3](#_Toc57969538)

[Experiments 4](#_Toc57969539)

[Data description 5](#_Toc57969540)

[LOD.csv 5](#_Toc57969541)

[Movement.csv 6](#_Toc57969542)

[Methods 7](#_Toc57969543)

[LOD analyzer 7](#_Toc57969544)

[Movement analyzer 8](#_Toc57969545)

[Results 10](#_Toc57969546)

[Discussion 15](#_Toc57969547)

[Conclusions 15](#_Toc57969548)

[References 15](#_Toc57969549)

# Introduction

The state of the art machine learning technique to optimize neural network AI controllers (agents) is back-propagation and deep learning. This method assumes that solutions and examples for such solutions are known. A different approach to this is evolutionary computation and genetic algorithms. Instead of rewarding proper responses to specific situations and hoping for proper generalization, a genetic algorithm uses an objective function. This objective function evaluates the total performance of a system and can select controllers that perform better. Over many generations such system optimizes controllers, often finds more creative solutions [1], and most importantly does not require knowledge about specific actions but only the desired outcome. This method works well when optimizing individual controllers as their objectives do not conflict with the objectives of other controllers. When agents need to interact or work in teams [2], the situation becomes much more complicated, and often the goals of the individual are contrary to the success of the group. Imagine self driving cars. Optimizing them to reach their goal as fast as possible might induce negative side effects. How to overcome such issues? Group level selection plays an important role in evolution. Often individuals do not act alone, but in groups. As such they can achieve together more than alone. Collaborative hunting is one of those examples, where individuals are not selected individually but as a group [3]. This group level selection scheme typically pools the resources the group collected and redistributes them back equally, which implies that groups are evaluated by their average performance. What if this scheme is altered? Imagine a group is rewarded according to the performance of its best individual (all for one) or its worst performer (one for all)? In the all for one case, you would pool all resources on one individual, and in the case of the one for all you would distribute the resources as fairly as possible. The big question is, which of the two groups collected the most resources in total? This research will use neuro-evolution where agents are simulated in a virtual environment and controlled by so called Markov Brains. Group level selection regimes such as the ones described above will be tested with respect to their effect on individual and group level performance.

## Background

The people’s collaboration in the societies can be considered as one of the main aspects of the development in terms of economy, health, education and so on. For example, in a country the citizens would have a better health care system, if all of them pay their taxes. However, it is always possible that some of people can bypass the rules and stop paying it. There are different approaches to address this issue for example, a government can deprive those who do not pay their taxes from welfare system benefits or another way can be educating people in a way that it works based on trust.

In this regard, there are some practices to simulate the real societies using small group of men and women and test their way of collaboration. For instance, one of these experimental economics is “public goods game”. In this game the participants are given equal amount of money and there is a public pot that they can put some or all their money. When they put their tokens in the pot, it will be multiplied by a factor and then, it will be divided in the players. So, if a player is careful about the society benefits, he/she will put more money and those who are selfish and only consider their personal benefits will not put anything. The scenario that individuals only think about their personal gain is called “Tragedy of the Commons”.

This problem or tragedy can be discussed in the field of artificial intelligence as well. For example, if we make fisher robots and reward them based on the number of fishes they can bring, they will do it as much as possible and eventually the ecosystem will be damaged. So, maybe we should change their rewarding schema in a way that they must also consider the community benefits and do not ruin the ecosystem. In this study, we are going to work in this problem and check the possibility of forcing artificial agents to cooperate based on their common welfare and not only their individual interest.

## Purpose

This research seeks to improve the way we train groups of AI controllers (agents) to perform better individually and in teams at the same time. While this is a basic research question in optimization of neural networks using genetic algorithms, it has direct applications to robotics and other autonomous AI decision making systems that need to work in groups.

## Literature review

# Material and Methods

This research will use the MABE (C++ Modular Agent Based Evolution Framework [4]) to implement virtual test environments. Agents are controlled using Markov Brains [5] which are a particular evolvable type of neural network. After replicating evolutionary experiments are completed, data will be analyzed and visualized.

## Experiments

Simple example:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  | Icon  Description automatically generated | | Icon  Description automatically generated | Icon  Description automatically generated | Icon  Description automatically generated |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | | Turn 1 Scores | | | | | | | | | | A1 0 | | | A2  0 | | A3  0 | | A4  0 | | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  | Icon  Description automatically generated | |  | Icon  Description automatically generated |  |  |  | |  |  |  | |  |  |  |  |  | |  |  | Icon  Description automatically generated | |  |  |  | Icon  Description automatically generated |  | |  |  |  | |  |  |  |  |  | | Turn 2 Scores | | | | | | | | | | A1  1 | | | A2  0 | | A3  1 | | A4  0 | | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  | Icon  Description automatically generated |  | |  | Icon  Description automatically generated |  |  |  | |  |  |  | |  |  |  |  |  | |  |  | Icon  Description automatically generated | |  |  |  | Icon  Description automatically generated |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | | Turn 3 Scores | | | | | | | | | | A1  1 | | | A2  0 | | A3  2 | | A4  0 | |   First generation’s genetics: A1: G1 A2: G2 A3: G3 A4: G4  IDs: A1: 1 A2: 2 A3: 3 A4: 4 |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  | Icon  Description automatically generated | | Icon  Description automatically generated | Icon  Description automatically generated | Icon  Description automatically generated |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | | Turn 1 Scores | | | | | | | | | | A1 0 | | | A2  0 | | A3  0 | | A4  0 | | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | | Icon  Description automatically generated |  | Icon  Description automatically generated |  |  | |  |  |  | |  |  |  |  |  | |  | Icon  Description automatically generated |  | |  | Icon  Description automatically generated |  |  |  | |  |  |  | |  |  |  |  |  | | Turn 2 Scores | | | | | | | | | | A1 0 | | | A2  1 | | A3  0 | | A4  1 | |   Second generation’s genetics: A1: G1\*V1 A2: G3\*V2 A3: G3\*V3 A4: G3\*V4  IDs: A1: 5 A2: 6 A3: 7 A4: 8 | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  | Icon  Description automatically generated |  |  | |  |  |  | |  |  |  |  |  | |  | Icon  Description automatically generated |  | |  | Icon  Description automatically generated |  |  |  | |  |  |  | |  |  |  | Icon  Description automatically generated |  | |  |  |  | |  |  |  |  |  | | Turn 3 Scores | | | | | | | | | | A1 0 | | | A2  1 | | A3  0 | | A4  2 | | |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  | Icon  Description automatically generated | | Icon  Description automatically generated | Icon  Description automatically generated | Icon  Description automatically generated |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | | Turn 1 Scores | | | | | | | | | | A1 0 | | | A2  0 | | A3  0 | | A4  0 | | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  | Icon  Description automatically generated | | Icon  Description automatically generated | Icon  Description automatically generated | Icon  Description automatically generated |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | | Turn 2 Scores | | | | | | | | | | A1 1 | | | A2  1 | | A3  1 | | A4  1 | |   Third generation’s genetics: A1: G3\*V2 \*V5 A2: G3\*V4\*V6 A3: G3\*V4\*V7 A4: G3\*V4\*V8  IDs: A1: 9 A2: 10 A3: 11 A4: 12 | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  | |  |  |  |  |  | |  | Icon  Description automatically generated |  | | Icon  Description automatically generated |  |  | Icon  Description automatically generated |  | |  |  |  | | Icon  Description automatically generated |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | |  |  |  | |  |  |  |  |  | | Turn 3 Scores | | | | | | | | | | A1 1 | | | A2  2 | | A3  1 | | A4  1 | | |

**Game settings:**

Variable settings:

|  |  |
| --- | --- |
| Reward mode | Meaning |
| 0 | Individual reward |
| 1 | Mean score (What they got on average) |
| 2 | Maximum score (What their best performer received) |
| 3 | Minimum score (What their worst performer received) |

|  |  |
| --- | --- |
| Group mode | Meaning |
| 0 | Clone |
| 1 | Four different organisms |

Constant parameters:

Other 11 points such as field’s dimension, …

## Data description

### LOD.csv

|  |  |
| --- | --- |
| Column name | Explanation |
| Generation | Generation number |
| ID | ID of each agent |
| Score | This field based on rewarding scheme value has the following meanings:   |  |  | | --- | --- | | Reward mode | Score Meaning | | 0 | Individual score | | 1 | Mean score of the four agents | | 2 | Max score of the four agents | | 3 | Minimum score of the four agents | |
| rawScores | Scores of all the group members |
| ownScore | The own score of the agent which is selected in the line of decent |

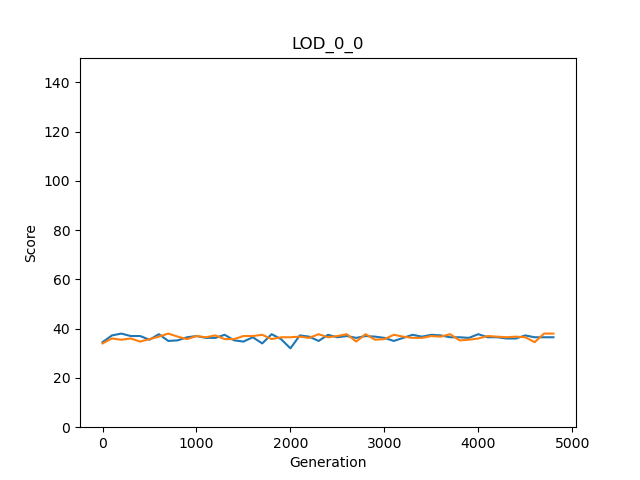
### Movement.csv

|  |  |  |
| --- | --- | --- |
| Column name | Explanation | |
| t | Turn number | |
| x | Position in X axis | |
| y | Position in Y axis | |
| d | Direction | Meaning |
| 0 | Up |
| 1 | Right |
| 2 | Bottom |
| 3 | left |
| e | Number of grasses that the agent gathered. It can be named as energy as well. | |
| a | Action type | Meaning |
| 0 | Do nothing |
| 1 | Turn left |
| 2 | Turn right |
| 3 | Move forward |
| 4,5,6,7 | giving grasses to an agent or putting it on a tile. |
| b | Beep (yes or no) | Meaning |
| 0 | No |
| Any other number | Beep |

## Methods

### LOD analyzer

Testing: using 0.25% of data for LOD\_0\_0\_0 and LOD\_0\_0\_1

Graph test (step 3)

LOD statistics test (step 4)

Step 4.1

Firstly, we have checked if minimum, maximum, sum and average of each row are calculated correctly. For this case, two of rows from LOD\_0\_0\_0 and LOD\_0\_0\_1 are tested manually.

Step 4.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| rewardGroupModes | meanOwnScores | meanMinimums | meanMaximums | meanAverages | meanSums |
| LOD\_0\_0 Replicate 0 & 1  (Excel) | 36.41326531 | 25.83673469 | 48.3775510 | 36.413265 | 145.653061 |
| LOD\_0\_0  Replicate 0 & 1  (Python) | 36.41327 | 25.83673 | 48.37755 | 36.41327 | 145.6531 |

### Movement analyzer

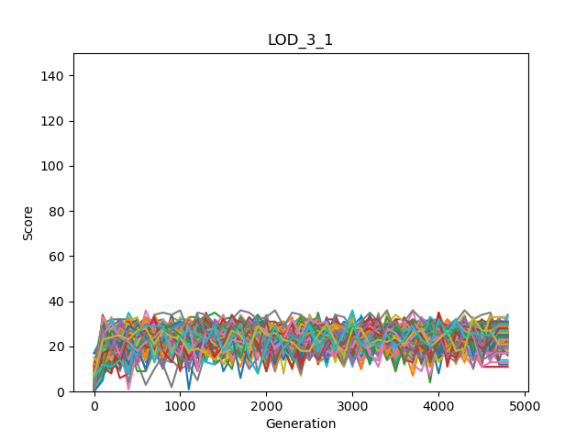
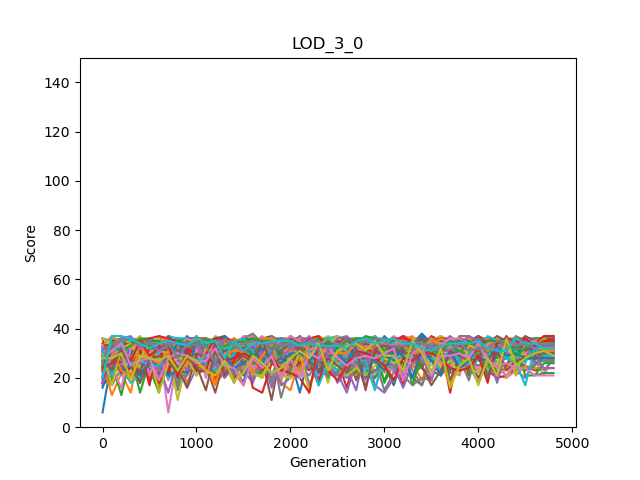
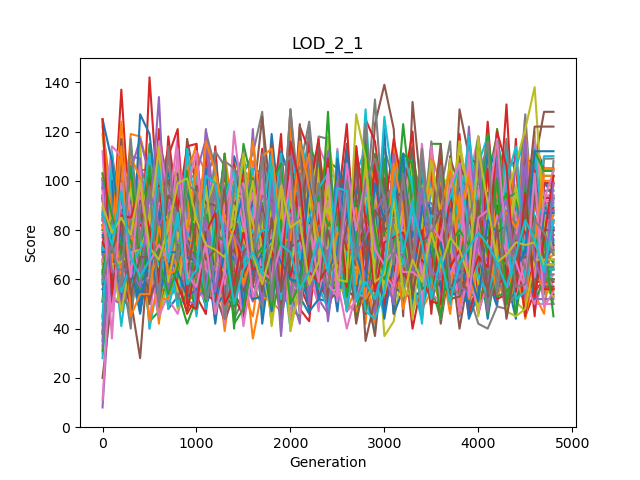
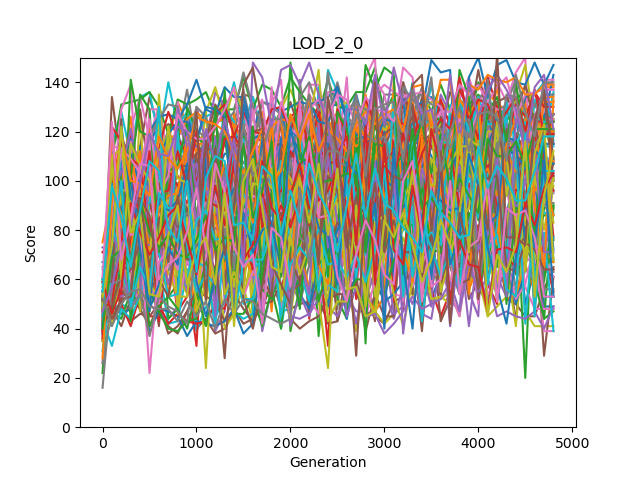
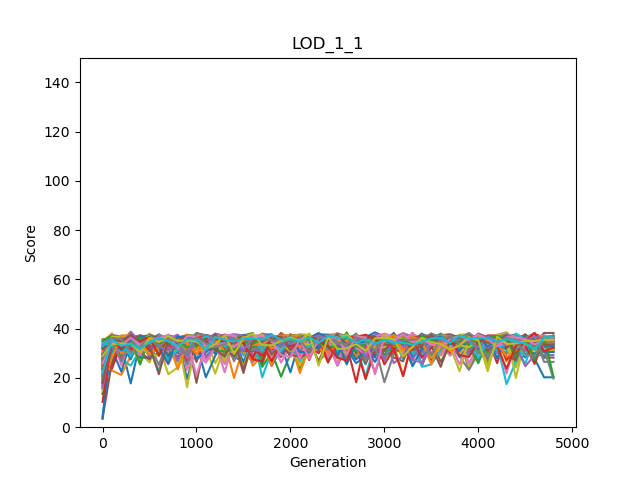
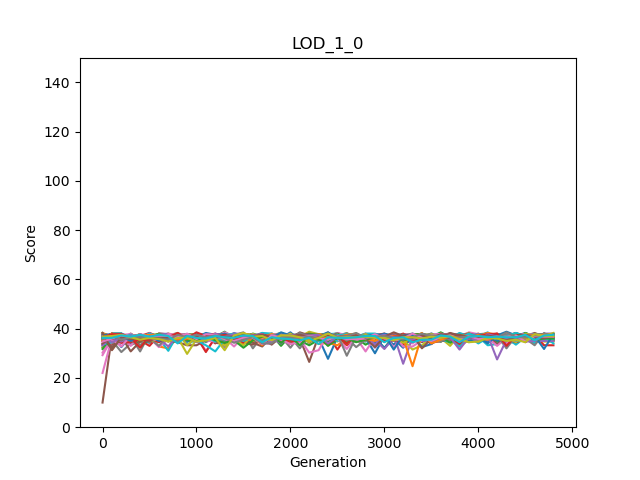
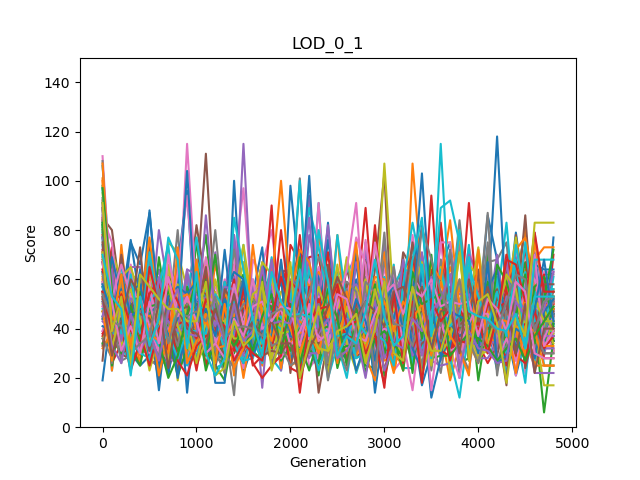
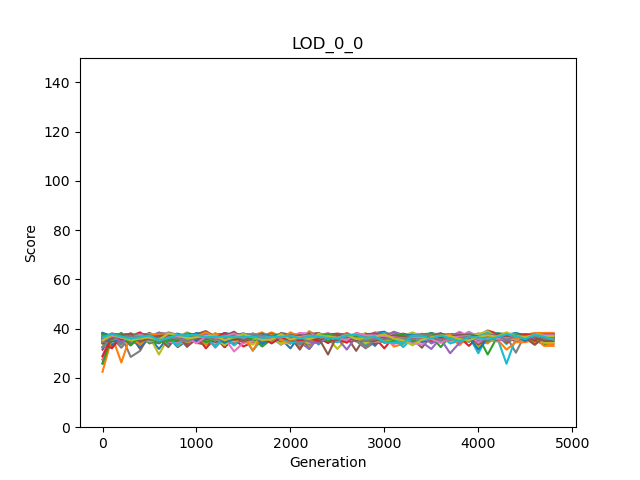
Testing:

First, we should get sure that the percentages for each movement file is calculated correctly. For this propose we selected to of the files and compared the results using python and excel.

The next testing part is about the averages which again we compared python and excel results.

|  |  |  |  |
| --- | --- | --- | --- |
| Reward scheme and  group mode | Mean of *giving or putting grasses* percentage | Mean of *do nothing*  percentage | Mean of *beep* percentage |
| Movement\_1\_1 Excel | 0.002 | 2.0775 | 21.978 |
| Movement\_1\_1  Python | 0.002 | 2.0775 | 21.978 |
| Movement\_2\_1  Excel | 2.006 | 0.8965 | 21.8615 |
| Movement\_2\_1  Python | 2.006 | 0.8965 | 21.8615 |

# Results



Individual | Clone

Individual | Group

Mean | Clone

Mean | Group

Maximum | Clone

Maximum | Group

Minimum | Clone

Minimum | Group

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| rewardGroupModes | Mean  OwnScores | Mean  Minimums | Mean  Maximums | Mean  Averages | Mean  Sums |
| LOD\_0\_0  Group mode: Clone  Reward mode: Individual | 36.4275510 | A picture containing shape  Description automatically generated26.0175510 | 47.4516326 | 36.4275510 | 145.710204 |
| LOD\_0\_1 3  Group mode: Not Clone  Reward mode: Individual | 44.7497959 | 20.1351020 | 52.0906122 | Logo  Description automatically generated36.5244898 | Logo  Description automatically generated146.097959 |
| LOD\_1\_0 2  Group mode: Clone  Reward mode: Average | 36.4372959 | Logo  Description automatically generated25.8336734 | 47.5702040 | A picture containing shape  Description automatically generated36.4372959 | A picture containing shape  Description automatically generated145.749183 |
| LOD\_1\_1  Group mode: Not clone  Reward mode: Average | 47.0285714  Logo  Description automatically generated | 11.6457142 | Logo  Description automatically generated56.7748979 | 33.9260204 | 135.704081 |
| LOD\_2\_0  Group mode: Clone  Reward mode: Maximum | 91.9987755  Logo  Description automatically generated | 2.70897959 | Logo  Description automatically generated91.9987755 | 30.8365306 | 123.346122 |
| LOD\_2\_1  Group mode: Not clone  Reward mode: Maximum | 67.5675510  A picture containing shape  Description automatically generated | 1.56244898 | A picture containing shape  Description automatically generated75.5930612 | 31.0185204 | 124.074081 |
| LOD\_3\_0 1  Group mode: Clone  Reward mode: Minimum | 30.7763265 | Logo  Description automatically generated30.7763265 | 42.3973469 | Logo  Description automatically generated36.4311734 | Logo  Description automatically generated145.724693 |
| LOD\_3\_1  Group mode: Not clone  Reward mode: Minimum | 34.1008163 | 23.0051020 | 44.5406122 | 33.0433673 | 132.173469 |

|  |  |  |  |
| --- | --- | --- | --- |
| Reward scheme and  group mode | Mean of *giving or putting grasses* percentage | Mean of *do nothing*  percentage | Mean of *beep* percentage |
| Individual  Clone | 0 | 0.7005 | 13.4225 |
| Individual  Not clone | 0 | 0.136 | 24.147 |
| Average  Clone | 0 | 0.2965 | 17.156 |
| Average  Not clone | 0.002 | 2.0775 | 21.978 |
| Maximum  Clone | 24.948 | 3.304 | 53.814 |
| Maximum  Not clone | 2.006 | 0.8965 | 21.8615 |
| Minimum  Clone | 0 | 0.963 | 51.2315 |
| Minimum  Not clone | 0 | 8.1085 | 9.5385 |

Always the minimum scores are higher in clone mode in relative to different individuals. So, if we are interested to have good minimum scores, we should use clone mode.

Except maximum rewarding scheme, the maximum score is higher when the agents are not clone. Therefore, if we do not consider setting the rewarding scheme, different agents are better than clones.

Except maximum rewarding scheme, the own score is higher when the agents are not clone. Therefore, if we do not consider setting the rewarding scheme, different agents are better than clones.

There is no clear pattern however, when clone modes have bigger averages the differences are bigger. So, we can say that for average also the clones are better.

Again, there is no clear pattern and when clone modes have bigger sum, the differences are bigger. Like the average for total sum also clones are slightly better.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reward  And Group | Beep  Mean Minimums | Mute  Mean Minimums | Beep  Mean Maximums | Mute  Mean Maximums |
| Individual  Clone | 26.09 | 25.24 | 47.19 | 46.11 |
| Individual  Not clone | 26.958 | 26.954 | 47.53 | 47.43 |
| Mean  Clone | 26.14 | 25.27 | 47.45 | 46.20 |
| Mean  Not clone | 25.41 | 25.45 | 46.97 | 46.99 |
| Maximum  Clone | 1.56 | 3.74 | 92.88 | 34.00 |
| Maximum  Not clone | 25.13 | 25.57 | 46.89 | 47.41 |
| Minimum  Clone | 31.51 | 8.72 | 41.49 | 15.19 |
| Minimum  Not clone | 23.72 | 23.46 | 43.85 | 43.47 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reward  And Group | Beep  Mean Averages | Mute  Mean Averages | Beep  Mean Sums | Mute  Mean Sums |
| Individual  Clone | 36.40 | 35.40 | 145.61 | 141.63 |
| Individual  Not clone | 36.97 | 36.96 | 147.88 | 147.84 |
| Mean  Clone | 36.52 | 35.46 | 146.10 | 141.86 |
| Mean  Not clone | 35.88 | 35.92 | 143.55 | 143.69 |
| Maximum  Clone | 30.00 | 16.24 | 120.03 | 64.97 |
| Maximum  Not clone | 35.75 | 36.21 | 143.03 | 144.87 |
| Minimum  Clone | 36.46 | 11.89 | 145.84 | 47.56 |
| Minimum  Not clone | 33.44 | 33.15 | 133.77 | 132.61 |

# Discussion

# Conclusions

# References

|  |  |
| --- | --- |
| [1] | L. Joel, C. Jeff and M. Dusan, "The surprising creativity of digital evolution," in *Artificial Life Conference Proceedings 14, pages 76-83. MIT Press*, 2017. |
| [2] | H. Arend and M. Masoud, "Evolution of autonomous hierarchy formation and maintenance," in *Artficial Life Conference Proceedings 14, pages 366-367. MIT Press*, 2014. |
| [3] | B. Gergely and S. Szabolcs, "Beneficial laggards: multilevel selection, cooperative polymorphism and division of labour in threshold public good games," *BMC evolutionary biology,* 2010. |
| [4] | B. Clifford and H. Arend, "Mabe (modular agent based evolver): A framework for digital evolution research," in *Artificial Life Conference Proceedings*, 2017. |
| [5] | H. Arend, A. E. Jeffrey, S. O. Randal, B. K. David, S. Jory, A. Larissa, T. S. Ali, K. Peter, S. e. Leigh och G. Heather, ”Markov brains: A technical introduction”. |