Hipsters in the Swarm of Robots

Ameya Wagh
Robotics Engineering
Worcester Polytechnic Institute
Worcester, USA
aywagh@wpi.edu

Milan Shah
Robotics Engineering
Worcester Polytechnic Institute
Worcester, USA
mrshah@wpi.edu

***Abstract*—In the social networks like Facebook, Twitter, or Instagram the spread of opinions, memes, news or in the network of people the spread of diseases or rumors depends on its structure as well as the spreading process. However, heterogeneous behavior among different type of agents can substantially change this spreading dynamics. For example, anti-establishment nodes also known as hipsters can significantly affect the spreading dynamics of two competing products when only one product was present initially. Here, we aim to investigate such behavior in the Swarm of Robots by introducing hipster robots in the network.**

***Index Terms*—Swarm Robotics, Hipsters, Conformist**

I. Introduction

In recent years, the use of social media sites like Facebook, Twitter, or Instagram has increased a lot. It is affecting marketing strategies, political opinions, etc at very high scale and it depends on the structure of the network. A normal agent conforms to its neighbor following the majority rule [4]. This can be seen in social circles who adopt a common opinion. However, anti-establishment agents, also called hipsters [2], who do not accept or adopt less popular opinion or product, play a significant role. They can change the behavior of the entire network partially or completely based on their proportion. Similar results can be seen during the time of elections i.e. such people prefer to vote for the political party who is less famous. As this behavior is contagious, it can result incompletely unexpected results. Same way, in case of two competing products, final fraction of people who adopt each product: even when only one of the two products exists in the beginning of the simulation, small fraction of hipsters in a network can cause the other product to eventually become more popular one [2].

In our project, we try to replicate this behavior in the swarm of robots and analyze how it affects the propagation of a certain opinion. These behaviors are not similar to faulty robot behavior or when they are lying. Faulty agents are those robots in the swarm of robots which are not working properly because of a software bug, power failure, sensor failure, or motor failure [5]. On the other hand, lying agents are those malicious agents that deliberately give a false reading(s) in a competitive case for personal gain [1]. However, hipsters are the agents who defy the popular opinion and would always do the opposite of the majority.

II. Background Work

A. Propagation Models

In the system of Swarm robots, each robot is independent entity. In such system, robots do not have much memory to store large data. Considering, one robot to have a message or an opinion and it is continuously moving, whenever other robots come into its vicinity it transmits the information and thus propagating the information. This way a message propagates among all the robots. It's not necessary to spread a message constantly. A robot can transmit periodic bursts by turning on and off its LEDs. The robots in its neighborhood turn on their own LEDs following their neighbors. This kind of propagation model is generally seen in fireflies which use it for synchronization [3] [7]. Similar propagation trends are seen when a rumor is spread in a network.

B. Majority Rule

Majority rule [4] states that a majority of a given entity is considered. This is modeled by a max function. In swarm, majority rule is generally used to decide between multiple entities and a majority vote of those entities is considered. For example, let say a person is planning to buy a new mobile phone of either of the two brands, e.g. iPhone or android. If most of his friends are using a mobile phone from one particular brand, e.g. iPhone, then that person would also buy the same brand following a majority rule.

C. Hipster Behaviors

Hipster behavior is loosely defined as exhibiting a behavior to do exactly opposite. This can be a behavior opposite to a condition or a state. It can be tied up with the majority rule as exhibiting behavior exactly opposite to majority rule, i.e. minority rule. For example, in the above-mentioned scenario, if a person is a hipster then he would buy a mobile phone which is used by least of his friends.

III. Methodology

A. Experimental Setup

We performed experiments on the Khepera IV robot [8] in ARGoS simulation environment [6]. A population of N robots was randomly initialized in an environment. This experiment was repeated for different values of N i.e. for different number of robots. All the robots would move randomly in the environment following an obstacle avoidance algorithm. Each robot had an active and a passive state and was initialized with a passive state.

B. Binary Opinion System

A robot with id = 0 was initialized with an opinion. For example, in our implementation, the robot had one opinion, represented by RED color (value = 1) or an opposite opinion, represented by GREEN color (value = -1). Initially, all the robots had no opinion, represented by turning off all the LEDs. Robots with hipster behaviors were shown with BLUE color led.

C. Acquiring Opinion

If the robot was a normal behaving robot or say conformist robot, it would follow majority rule and take the color of the maximum active neighbors as shown in fig 1. Alternately, if the robot was a hipster robot, it would take the opposite color of the maximum active neighbors e.g. if the majority was RED, it would take GREEN as shown in fig 2. The opinion was found out by adding the opinions of all the neighbors in the field of view given as Cumulative Neighbour opinion (CNO), refer equation 1. The passive robot chooses the opinion by checking the condition given by equation 2.

$$CNO = \sum_{n \in neighbors} opinion[n] \tag{1}$$

$$opinion = \begin{cases} 1 & CNO > 0 \\ 0 & CNO == 0 \\ -1 & CNO < 0 \end{cases} \tag{2}$$

If the CNO is positive, it takes opinion represented by 1. Similarly, if the CNO is negative, it takes opinion represented by -1. if the CNO is equal to 0, it stays in the passive mode. A robot acquires the opinion only once and does not follow the opinion acquisition state after it has acquired either of the opinion, i.e moves to active state. Alternately, the robots with passive state only can acquire an opinion.

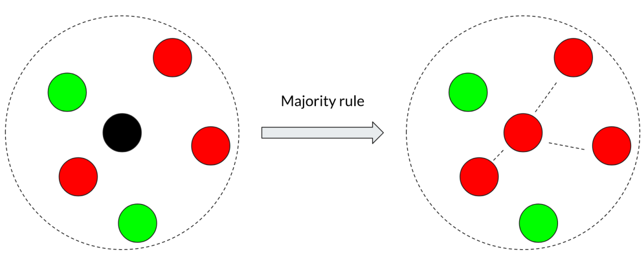


Fig. 1: Conformist Robot Behavior

In our simulation, to differentiate among conformist robots and hipster robots, we have followed the convention:

- Conformist Robots (see fig 3):
  - No opinion i.e. opinion value 0: BLACK color
  - Opinion value -1: GREEN color
  - Opinion value 1: RED color

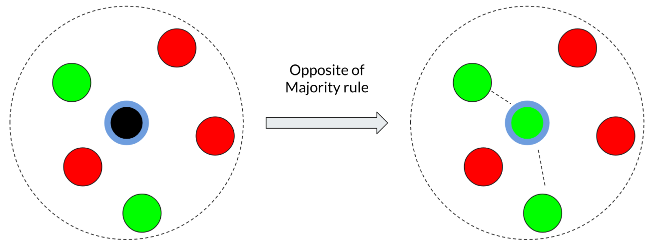


Fig. 2: Hipster Robot Behavior

- Hipster Robots (see fig 4):
  - No opinion i.e. opinion value 0: BLUE color
  - Opinion value -1: CYAN color
  - Opinion value 1: MAGENTA color

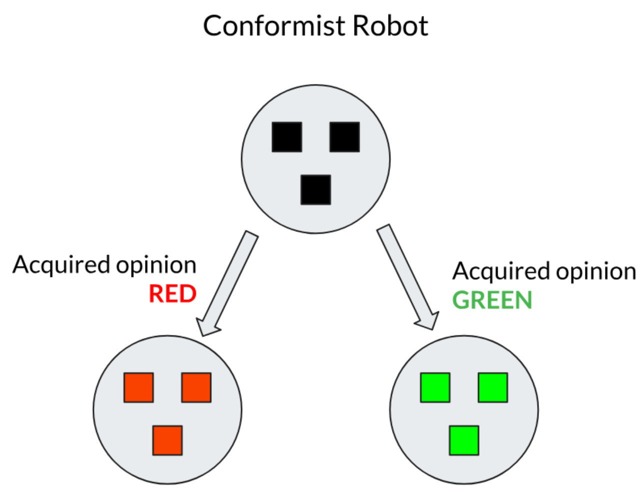


Fig. 3: Conformist Robot

The Magenta color is a combination of BLUE (robot being hipster) and RED (the robot acquired opinion RED). Similarly, the LED's indicating CYAN signifies BLUE + GREEN.

To study the effect of hipsters in the final distribution of opinions, we experiment with varying hipster percentage in a given population. We vary two parameters, the population of the swarm and the percentage of hipsters in the given population. To derive conclusions concretely, we perform the given experiments multiple times with different random configurations.

Algorithm 1 describes the algorithm followed by individual robot. $OPINION == 0$ signifies that the robot is in passive state. An active state is indicated by a non-zero opinion. Every robot follows 2 main procedures, $Init$ which initializes the attributes and $Step$ which is executed at every simulation tick.

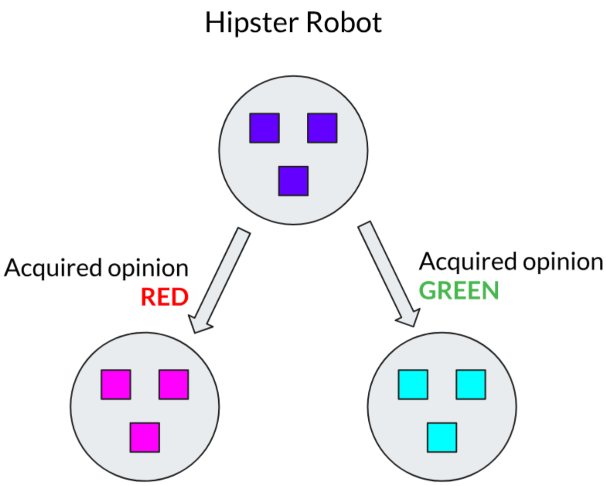


Fig. 4: Hipster Robots

Algorithm 1 Behavior of a Robot

```
procedure INIT()                          ▷ Initialize the robot
    OPINION=0
    if id==0 then
        OPINION=1
    if id < (hipster_percentage · population) then
        isHipster=1
    else
        isHipster=0
    neighbor_opinion_listener.start()
procedure STEP()                ▷ Performed at every time step
    neighbors.broadcast("opinion", OPINION)
    avoid_obstacle()
    if OPINION==0 then
        update_opinion()
```

The robot follows a simple obstacle avoidance controller for random walk. The Khapera robot uses its distance sensors to form a distance vector of obstacle from the robot. If the magnitude of that vector crosses a certain threshold, the robot turns either right or left depending on the angle of the obstacle made with the robot.

Algorithm 2 Obstacle avoidance

```
procedure AVOID_OBSTACLE()
    if sensor.magnitude < COLLISION_THRESH then
        go_straight()
    else
        if sensor.angle > 0.0 then
            turn_right()
        else
            turn_left()
```

The robot updates its opinion when it is in passive state or when the $OPINION == 0$. The robot uses majority rule to update its opinion. If the robot behavior is conformist, the robot updates the opinion given by majority rule, whereas if the robot is a hipster, it takes the opposite opinion as that given by majority rule.

Algorithm 3 Update opinion

```
procedure UPDATE_OPINION()
    if sum(neighbor_opinion) > 0 then
        if isHipster then
            OPINION=-1
        else
            OPINION=1
    else if sum(neighbor_opinion) < 0 then
        if isHipster then
            OPINION=1
        else
            OPINION=-1
```

For total 20 robots and 40% hipster robots, this can be seen in the fig 5, 6, and 7. A robot once in an active state can not change its opinion, thus a robot can only take one opinion.

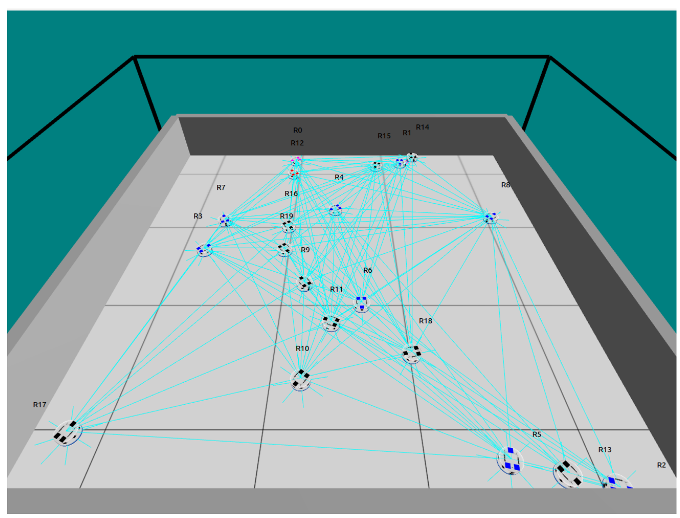


Fig. 5: Initial configuration in ARGoS

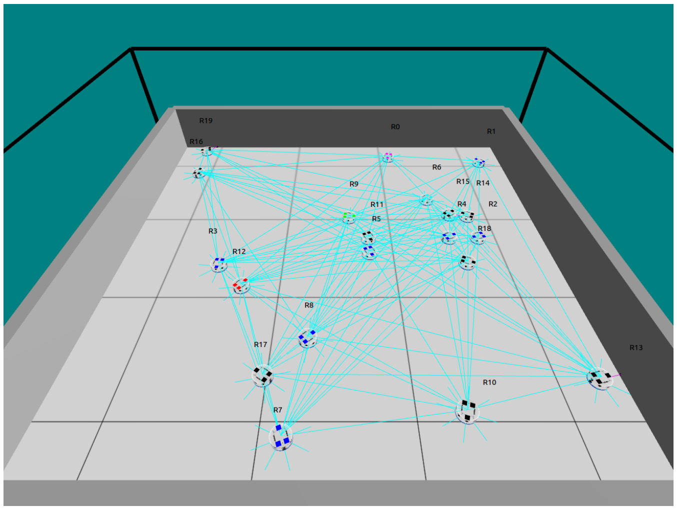


Fig. 6: Intermediate configuration in ARGoS after 253 steps

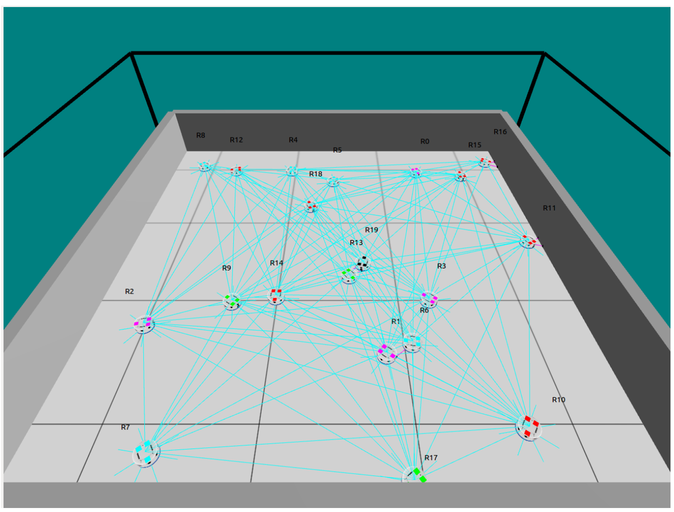


Fig. 7: Final configuration in ARGoS after 1100 steps

IV. Results

A. Trends of opposite opinion and hipster percentage

We carried out this experiment for the combination of different total population i.e. 10, 20, 30, 40, and 50, and for every population we evaluated for different number of hipster robots i.e. 5%, 10%, 15%, ..., 95%, and 100% of total population. To generalize the experiment, we ran the experiment for 10 times for a given set of configuration of population and hipster percentage. Each time the robots were placed at random locations and were assigned random ids.

From the figure 9 it can be inferred that the lesser number of robots show high variance in the percentage of opinion, but eventually converge to a value. As the number of robots increase, the variance in the opinion percentage reduces. This can be intuitively reasoned by the space obtained for the robots for random walk. As the size of the arena is fixed, the density of the robots changes with change in population. Lesser number of robots have less density in the arena. As we use majority rule, we take majority votes of the neighbour in a fixed radius. With less number of robots in the neighbourhood, this majority rule fluctuates significantly. with increased number of neighbours, the majority rules holds good and stable outcomes are observed.

Another observation from figure 9 shows the convergence of the opinion percentage at 50%. This is statistically consistent as the population tends to 100% hipsters, each individual robot will have an opinion opposite to its neighbour. Thus the probability of each opinion would be 50%.

We show the box plots of the trends of opposite opinion percentage and hipster percentage in figure 10. It can be observed from the statistical information in the box plots that the variance in data decreases with increase in the density of population. Moreover, the outliers in the denser populations have significantly decreased.

B. Modelling the effect of hipster percentage on the total opinion distribution

From the above trends, we model the relation of the hipster percentage with the opinion percentage in a network. We consider the results with population 50 as they had the least variance and would be conducive to fit a curve.

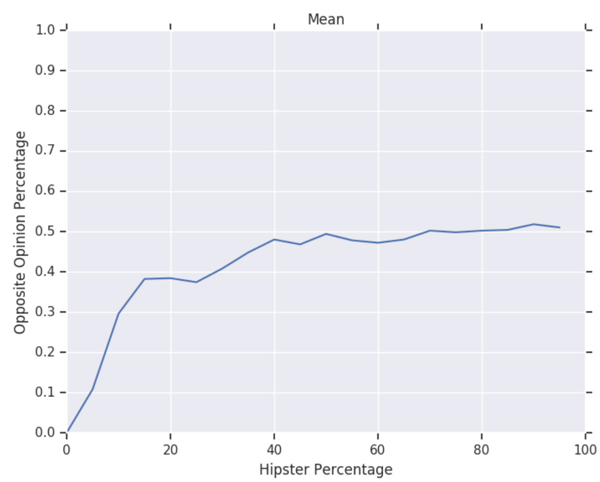


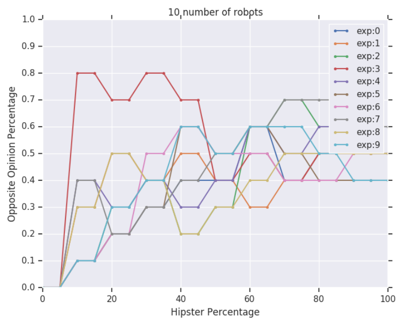
Fig. 11: plot of the mean data points for the trends with 50 robots

Figure 11 shows the mean of the data points in all the 10 trials for 50 robots. This curve is a non-linear function of hipster percentage which converges at 0.5. By observation similar looking models like $tanh$, log, $sigmoid$. We found the model given by equation 3 to be the best fit by least square method, where y is the percentage of opposite opinion and x is the precentage of hipsters.

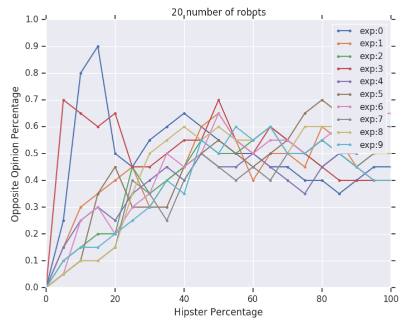
$$y = a \cdot tanh(b \cdot x) \tag{3}$$

Fitting the given model by minimizing the least squares of the data points, values of a and b were obtained as follows.

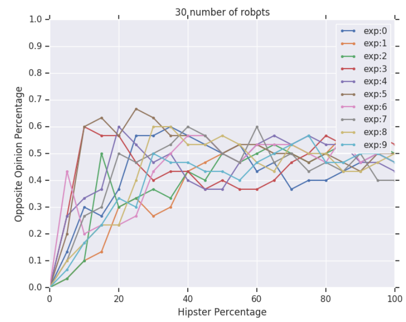
- a = 0.489



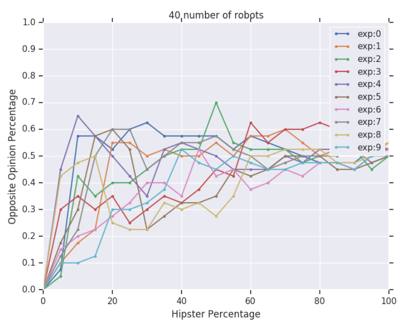
(a) trend with population 10



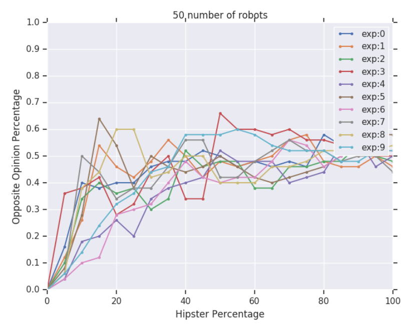
(b) trend with population 20



(c) trend with population 30



(d) trend with population 40



(e) trend with population 50

Fig. 9: Trends of opposite opinion percentage in the population w.r.t the hipster percentage

- b = 0.056

Thus the best fitting model was obtained and is given by equation 4

$$y = 0.489 \cdot tanh(0.056 \cdot x) \tag{4}$$

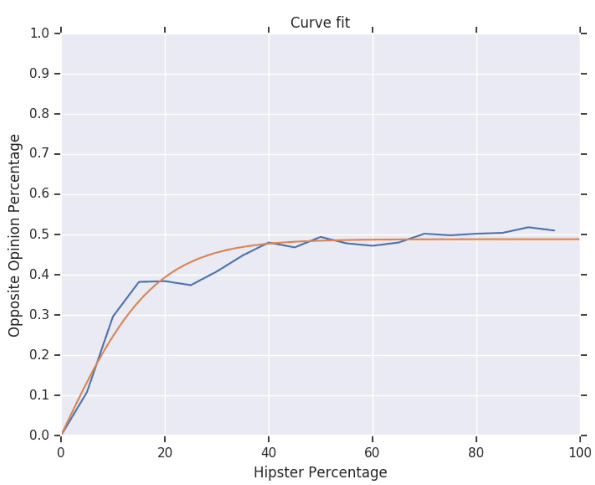


Fig. 12: Mathematical model for the relation of hipster percentage to opposite opinion percentage

The figure 12 shows the fit of the model with the actual data points.

V. Conclusion

We implemented a propagation algorithm in swarm robots and simulated it in Argos simulator. We studies the behavior of hipster robots in a network and the effect of difference in opinion with respect to the percentage of hipster robots in a given population. We acquired statistical data to show a trend of percentage of opposite opinion in a given population as with respect to the percentage of hipsters. Lastly, we found a best fitting mathematical model to represent this trend.

VI. Future Work

These experiments assumed that the robots would acquire the opinion only once. We would like to explore a more probabilistic approach in which the robot changes its opinion over time or due to certain thresholds. We would also like to explore the trends of these behaviors with more than 2 opinions and if so, how can they be modeled.

References

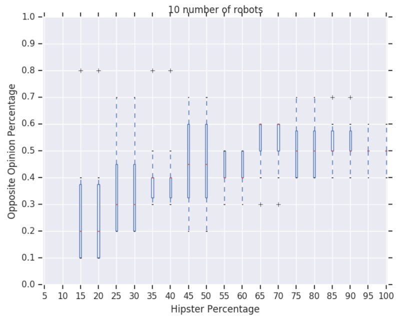
[1] Evan Ackerman. Swarm robots evolve deception. accessed Mar 27, 2019.

[2] Jonas S. Juul and Mason A. Porter. Hipsters on Networks: How a Small Group of Individuals Can Lead to an Anti-Establishment Majority. 2017.

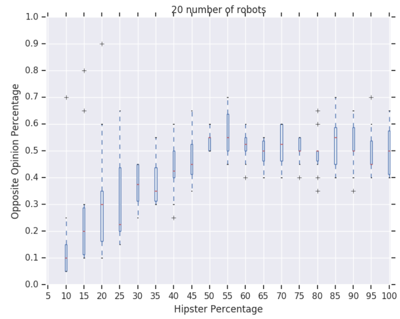
[3] Renato E. Mirollo and Steven H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, 1990.

[4] Marco A. Montes de Oca, Eliseo Ferrante, Alexander Scheidler, Carlo Pinciroli, Mauro Birattari, and Marco Dorigo. Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5(3):305–327, Dec 2011.

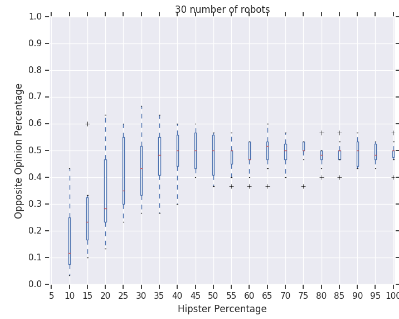
[5] James O'Keeffe, Danesh Tarapore, Alan G. Millard, and Jon Timmis. Adaptive Online Fault Diagnosis in Autonomous Robot Swarms. *Front. Robot. AI*, 5, 2018.



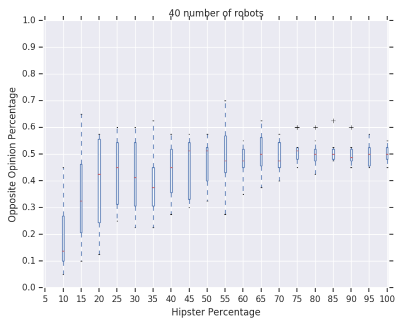
(a) trend with population 10



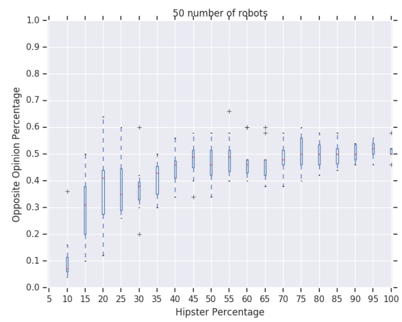
(b) trend with population 20



(c) trend with population 30



(d) trend with population 40



(e) trend with population 50

Fig. 10: Box plot of trends of opposite opinion percentage in the population w.r.t the hipster percentage

[6] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Timothy S. Stirling, Álvaro Gutiérrez, Luca Maria Gambardella, and Marco Dorigo. ARGoS: A pluggable, multi-physics engine simulator for heterogeneous swarm robotics. *IRIDIA - Technical Report Series*, (December 2010):1–22, 2011.

[7] Gonzalo Marcelo Ramírez-Ávila, Jürgen Kurths, Stéphanie Depickère, and Jean-Louis Deneubourg. *Modeling Fireflies Synchronization*, pages 131–156. Springer International Publishing, Cham, 2019.

[8] Jorge M. Soares, Iñaki Navarro, and Alcherio Martinoli. The khepera IV mobile robot: Performance evaluation, sensory data and software toolbox. *Advances in Intelligent Systems and Computing*, 417(January 2015):767–781, 2016.