

Microrobots

*To what extent do the different boards that can be composed with
Microrobots have the same level of difficulty?*

Contents

1	Introduction	2
2	Concepts	3
2.1	Graph Theory	3
2.2	Combinatorial analysis	3
2.3	Statistical calculus	3
2.4	Programming	3
3	Methodology	4
3.1	Calculus of the number of boards	4
3.2	Difficulty of a board	5
3.3	Calculus of distance between cells	5
3.4	Sampling Theory	6
3.5	Proof Statistical samples	6
4	Results	8
4.1	Results of Analysis of the Variance of one Factor (ANOVA)	8
4.2	Description of ANOVA method	9
4.3	Board frequency distribution	10
4.4	Data Analysis	12
5	Conclusions	14
5.1	Further work	14
6	Bibliography	15
7	Annex	16
7.1	Statistical calculus	16
7.2	Program	17

1 Introduction

Last summer, while I was with some friends during a hot afternoon, we discovered a new game that entertained us a lot. The game was called "Microrobots", and it was played the following way:

There is a board of 6x6 cells. On each cell there is a number and a color. Two cells are picked randomly and the players need to find the minimum distance between them. The movements can only be made in horizontal or vertical, and you can only move from the cell you are to another one that shares either the same color or the same number. As we can see, the rules were quite easy, but once we started to play we saw that the game had much more things to give us than just some fun in one afternoon. First of all, the board was composed of four non-symmetrical pieces that you had to put forming a square. It was said that with all the combinations that could be formed, the game could still be played, so that it was always a connected graph (Graph that from any node you can arrive to any other node). Moreover, we soon discovered that the game itself was much harder than expected; finding the shortest path was quite a tough activity for us. A lot of boards could be formed, and while in ones it seemed very easy to solve any combination, in others seemed nearly impossible.



Figure 1: A quick view of the different parts of the game

In the end, Microrobots raised too many questions inside me to leave them unsolved, and I decided to analyze the game in a more severe and rigorous way, in order to solve the problems that I had thought about. I decided that I wanted to combine different branches of mathematics, as it would be easier to obtain more interesting results: graph theory, counting principles, statistics and programming were the main foundations behind my project.

I narrowed my research, and the question that I finally asked myself was, *to what extent do the different boards that can be composed with Microrobots have the same level of difficulty?*

2 Concepts

2.1 Graph Theory

Once a board was created, I needed a mathematical way to express this board and operate with it. It was a very direct association for me to see the board of this game as a graph; more precisely, an undirected graph of 36 nodes (with every node associated with a cell). This is why, to do all the mathematical operations my work needs, I will be using graph theory, a discipline within the topic of discrete mathematics. All the concepts I will use are going to be either proved shortly or explained if they cannot be proved (for example, a definition).

2.2 Combinatorial analysis

Another question that quickly arose in me when I first started to play Microrobots is the total amount of different boards that can be created. Although it is not a major issue, determining this number requires careful inspection of the rotations and symmetries as you could create duplicates. This situation reminds me some of the problems that I have solved during my maths course.

The main importance of this figure (total amount of boards) is linked to the statistical analysis that will focus the determination of the difficulty of different boards. Due to the huge number of boards, I need to define a sample of boards to analyze. In order to have significant results for my research, the sample should be based on sample theory (a branch of statistics) and to define the sample I needed the total amount of boards.

2.3 Statistical calculus

I have just identified a first approach required in my work to the world of statistics. Nevertheless, there is a much more central point that requires statistical analysis. As I will define later, difficulty of a concrete board is the mean of all the steps used to solve all the different paths, so the mean of all the different possibilities that someone can play in the same board. Therefore, to analyze if there are significant differences in the difficulty of boards I need to determine if there are differences that are statistically significant amongst them.

In my case I will use modeling techniques to determine the distribution of paths in a board and statistical analysis to determine if there are significant differences. Although the concepts I need in that discipline are beyond the syllabus that I have studied in my maths course, all the basic concepts (moments, distribution, hypothesis test, etc.) were a significant help for me when I did my own research in these topics.

2.4 Programming

As we start to notice, my project implies managing a large amount of data -we will later see that there exists lots of different boards, and lots of operations need to be done with each one-. With this in mind I decided to write programs to help me with this task. Everything that the program does is explained and based on the theory learned during the course of Mathematics HL and some extra topics that I have been researching on; I created the program merely for the fact that it can do calculations much faster than manually. The base of the program is on the annex and it is programmed with Python 3.6. Moreover, to do the statistical calculus I will use XLSTAT (<https://www.xlstat.com/en/>).

3 Methodology

3.1 Calculus of the number of boards

To calculate the number of boards, it is necessary to explain how the boards were created. The game has 4 pieces, each piece being a 3x3 sub-board. You should place these four pieces forming a square of 2x2. The pieces can be put in any position or orientation if they are forming a square.

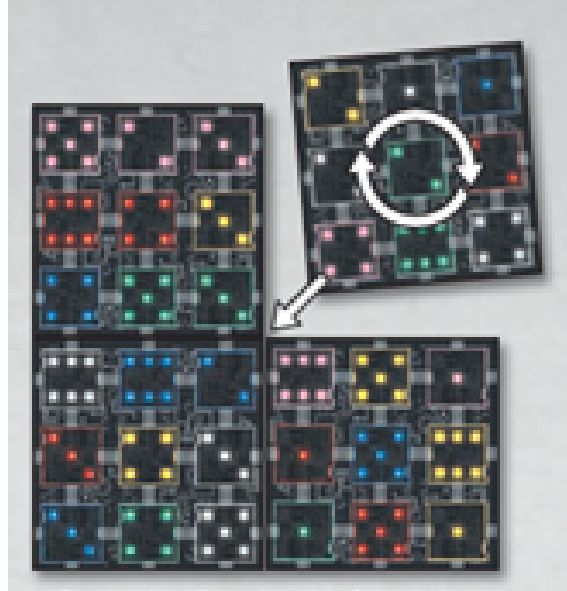


Figure 2: Scheme of how the boards can be created

A first calculation that seems interesting to understand the amount of data we are managing in our problem is the number of different boards that can be created. There are 4 pieces, that can be put in four different positions, with 4 different orientations. Combining elements (in our case pieces) in different positions is the classical problem of combinatorial analysis. In the first position you can put any of the 4 pieces, in the next position, you can only put 3 as you have already put one, in the next 2... which finally results in a total combination of $4!$ possible boards ($4 \cdot 3 \cdot 2 \cdot 1$). This number should be multiplied by the four different rotations of every piece to obtain the following result.

$$4! \cdot 4^4 = 6144$$

This is the total number of boards that we can create. But it is not our final result, as we are looking for the different boards that exist, and in those 6144 there are lots of boards which are repeated. A rotation of board will not change the distances between the points, it will just change the players perspective, a thing that does not influence our work at all. Each board has exactly 16 boards with the same connections, which are the four rotations of the complete board times the 4 rotations of half a board. All this cases generate the same graph, so we will subtract them. Obtaining than, that there only exist the following number of different boards.

$$\frac{6144}{16} = 384$$

As we can see this number is quite large taking into consideration that we have just 4 pieces for each board. As we have said before, to tackle with the problem we are going to analyze only a sample, so we will describe later how to get a significant sample in our case.

3.2 Difficulty of a board

Since the beginning of this project, we have talked about 'difficulty' of boards. To start working, let's propose a formal definition. For me, a board was harder if the minimum distance between random points were longer. Taking that into account and referring to what is mentioned in the subsection 2.3, let's assume a general definition for this work.

The difficulty of a board is the average of the minimum distance between all the cells in the same board assuming that the distance is the shortest path between two points, following the rules of the game.

We can see that a difference in the difficulty of a board would require that there are many distances that change in order to modify the average distance of any board.

3.3 Calculus of distance between cells

To make all this relations, one of the first steps is being able to find the minimum distance between two cells following the rules of the game. Applying the theory studied in discrete mathematics I came up with several different options to do, for example Dijkstra's algorithm (which appears in the Pearson Mathematics HL book). Although it seemed a very good option, I finally decided to apply the concept of adjacency matrices. The reason for it was quite simple; although Dijkstra was quite more efficient to find distances between two single points, in my project I need to look for all the distances of the graph that represents each board. Taking this into account and applying that an adjacency matrix of a graph powered up to n is equal to the connections of the same graph with n steps (see demonstration below), it was much more efficient to do it with this method. It is relatively easy to prove this property using Induction.

Lemma 1. *An adjacency matrix powered up to n is the connections of the same graph with n steps*

Proof. Base Case: If we take an adjacency matrix and we power it to 1, the property is true as the matrix doesn't change, and for definition the adjacency matrix shows the direct (1 step) connections in a graph.

Inductive Step: Assuming it is true for any given n , we want to prove it true for $n + 1$. This implies that we want to prove that if a matrix up to the power of n represents the different paths between the cells, multiplied to the original matrix will give a matrix that represents the different paths with $n + 1$ steps.

Let A be the adjacency matrix of a graph G . Let V_i and V_k be different points from the graph. Let F_{ik}^n be the different path of n length between V_i and V_k . We can express a walk of length $n+1$ steps from v_i to v_j as a n steps path from v_i to v_k and a walk of length 1 from v_k to v_j .

That means, the number of paths of $n+1$ steps from v_i to v_j is the sum over of all walks from v_i to v_k times the number of ways to walk in one step from v_k to v_j . Thus:

$$F_{ij}(n+1) = \sum_{k=1}^V A_{ik} F_{kj}^n = \sum_{k=1}^V A_{ik} A_{kj}^n$$

Which is the definition of the formula for the product used in the multiplication of matrices. Therefore we have proved it true for $n+1$ assuming it true for n .

Conclusion: As we have proven it true for a base case and assuming it true for $n=k$, true for $k+1$, it is true for all natural numbers.

□

Once I made an operation with one adjacency matrix, I could use it to find the distances between all the points I wanted from the same matrix. If I had used Dijkstra it would have been much harder as I would need to apply the algorithm several times in order to find the different distances.

3.4 Sampling Theory

The number of boards obtained in the subsections above clearly exceeds our capacity to compute. Therefore, I decided to take a sample of boards to make my research. In order to achieve sound results, I used the conclusions of Sampling Theory to determine the size of the sample to obtain statistically significant results.

According to the Sample Theory, the size of the sample to identify possible differences amongst groups of data with a confidence level (the level of certainty that a group of data can give us among the total) is determined by the following expression:

$$n = \frac{N\sigma^2 Z_\alpha^2}{e^2(N-1) + \sigma^2 Z_\alpha^2}$$

n: size of the required

N: total number of population (in our case 384 boards)

Z_α : represents the level of confidence we want in our results (for 95 percent of confidence the value is 1,96). This 95 percent level of confidence means that in less than 5% of the cases the results of take a sample of n results differs from real results (the results of the total of the cases) in more than e (the error of our estimation). This level of confidence is the most usual when designing an experiment with samples.

σ : variance of the population. In case of ignorance, as is our case, $p=0,5$ and $q=1-p=0,5$

e: represents the error in our estimation, this is, the difficulty of each board. As we can see afterwards, the steps of each path that defines the difficulty varies in a discrete way in the range [2,10] so a value of $e = 10\%$ (that represents in our case round 1) is discriminant enough for our purposes.

Applying this formula I arrived to the conclusion that I needed a sample of 76 boards to be significant. This sample also needed to be completely random, to obtain this randomness I made the computer chose the boards based on a random time seed, a random value which is obtained from a list depending on the time since the computer was turned on.

3.5 Proof Statistical samples

The calculation for the significant statistical sample is obtained through the calculation of the confident estimated interval for the mean, which is the following formula.

$$X - Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}} \leq u \leq X + Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}}$$

From where the error is the following one

$$e = Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}}$$

From this formula of the error of the confident interval of the mean, we need to isolate the n , to do it we follow the next procedure.

We square everything

$$e^2 = \left(Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}} \right)^2$$

$$e^2 = Z^2 \frac{\sigma^2}{n} \frac{N-n}{N-1}$$

Multiplying the fraction we obtain the following

$$e^2 = \frac{Z^2 \sigma^2 (N-n)}{n(N-1)}$$

From here we do basic algebra to obtain what is next

$$e^2(N-1)n = Z^2 \sigma^2 (N-n)$$

$$e^2 n N - e^2 n = Z^2 \sigma^2 N - Z^2 \sigma^2 n$$

$$e^2 n N - e^2 n + Z^2 \sigma^2 n = Z^2 \sigma^2 N$$

$$n(e^2 N - e^2 + Z^2 \sigma^2) = Z^2 \sigma^2 N$$

Now we finally isolate the n

$$n = \frac{Z^2 \sigma^2 N}{e^2 N - e^2 + Z^2 \sigma^2}$$

And we achieve the formula we have used to determine the size of the sample

$$n = \frac{Z^2 \sigma^2 N}{e^2 (N-1) + Z^2 \sigma^2}$$

4 Results

Before applying the statistical test that would made me obtain a real conclusion, I decided to obtain a first sample of the different difficulties in 76 different boards (significant statistical sample) to start having an idea of their behavior.

The following table presents the most relevant statistical measures obtained from 76 random boards and some basic statistic operations to start having a first idea of the behaviour of the game.

Statistical measure	value
\bar{X} , mean of the difficulty	3,43133
Standard Deviation	0,37816
Percentage of deviation	8,8818
Minimum	2,79784
Maximum	4,25772
Range	1,45988
Range Percentage	34,288

Table 1: Some interesting data to have a first approach to the problem in which we see the results from 76 different boards analyzed

As we can see, the results of the difficulty of different boards should be analyzed. There are differences of the means, the standard deviation is significant (10% of the mean value) and range is wide (35% of the mean value). The mean is located nearly in the middle between the minimum and the maximum so we can conclude there are the same number more or less of upper outlayers than of lower outlayers.

In this point I saw that my data analysis was making me unable to obtain deep conclusions, I could see there were some differences between the boards, but I couldn't determine whether these differences were significant or not due to the fact I had not done any tests. I decided then to go further in the statistical analysis to determine if the differences were significant enough to answer the question I established at the beginning of my work.

Doing some research I finally found that this is a classical problem in statistical analysis: determine if some samples come from the same collective or not. Each sample has a specific value (with differences amongst them). The technique used to solve this question is called ANOVA TEST, a test designed to tell you whether or not some different samples were or not equivalent (are part of the same collection).

4.1 Results of Analysis of the Variance of one Factor (ANOVA)

ANOVA is designed to test the significance of difference between three or more samples to test the null hypothesis, that the sample means are equal. In my case I have used a confidence level of 95 percent, due to the facts it's a very common one and because it gives me a sufficient level of confidence to distinguish between one more step and on step less.

The main conception of ANOVA analysis lies on:

1. Different sampling (in our case different board configuration)
2. Null Hypothesis (H_0): the mean of each sampling is the same (in our cases all boards have same difficulty)
3. Alternative Hypothesis (H_1): the means of different sampling are different (there are differences among the difficulties of different boards)

Initially the test is designed based on the premise that the variable follows a normal distribution. So far we have no clue about the distribution law of our variable (although later on we will discuss this point), so we can have some doubt about the applicability of the test. Nevertheless as this situation is quite normal, there are literature (see GLASS et al. 1972) about the robustness of the test and the resilience of the results even in case of the failure of the premises.

Before getting into the details of the method and the results, we can conclude that the differences of the difficulty amongst boards are significant enough, rejecting the null hypothesis that all boards have the same level of difficulty.

This results is quite aligned with our first approach of the results of our sample, when we saw that the standard deviation was about 10% of the mean value. The present test just confirmed our initial suspect with a scientific method that allows us to ensure that some combinations are more difficult than others.

4.2 Description of ANOVA method

Now let's try to explain how the method works. The main idea is based on identify the sources of variation amongst samples (boards). The total variation (as the sum of the squares of the deviation of each measurement from the grand mean) is expressed as a variance:

$$TotalVariation = V = \sum_{i=1}^{jk} (x_i - \bar{X})^2$$

Where j represents each board and k a specific measure of a board.

It can be proved (we will follow SPIEGEL (1999) Chapter 16 – *Analysis of Variance* for the rest of this explanation) that total variation can be written as the sum of V_w (variation of the distance of each path in a board regarding the mean distance of this board) and V_b (the variation of the mean distance of each board mean regarding the global mean distance).

$$V = V_w + V_b$$

We can consider each measurement from board j as the result of adding to the mean of this board Z_j a chance factor ϵ_{jk}

$$X_{jk} = \epsilon_{jk} + Z_j$$

Let Z be the global mean.

$$\alpha_j = Z_j - Z$$

Than we arrive to

$$X_{jk} = Y + \alpha_j + \epsilon_{jk}$$

The null hypothesis implies all boards have equal difficulty (mean) son $\alpha_1 = \alpha_2 = \dots = \alpha_j = 0$ Therefore H_0 implies $Y_j = Y$ for all j.

It can be demonstrated that

1. $\frac{V_w}{\sigma^2}$ is χ^2 distributed with a(b-1) degrees of freedom
2. And additionally under the null hypothesis $\frac{V_b}{\sigma^2}$ and $\frac{V}{\sigma^2}$ are χ^2 distributed with (a-1) and (ab-1) degrees of freedom.

Where χ^2 represents the distribution of the addition of independent standardized squared variables. In our case this variable is the difference between the difficulty of each single path in a board and the mean difficulty of this board

Let's define a as the number of boards and b as the number of distances (measurement) in each board

So using estimators of the variance of each board based on variation previously defined

$$S_w^2 = \frac{V_w}{a(b-1)}$$

$$S_b^2 = \frac{V_b}{a-1}$$

$$E(S_b^2) = \sigma^2 + \frac{b}{a-1} \sum_{i=1}^j \alpha_j$$

If the null hypothesis H_0 is not true we can expect S_b^2 be greater than σ^2 while S_w^2 will be σ^2 if the means are equivalent.

The ratio $\frac{S_b^2}{S_w^2}$ it is a good statistic to prove the Ho. If the statistic is significantly large, we can conclude there are differences amongst board variances (due to α_i not null).

The theorem that ensure that $\frac{S_b^2}{S_w^2}$ follows a F distribution with a-1 and a(b-1) degrees of freedom will provide use with the ANOVA test result.

EXCEL (and XLSTAT) will provide the library to develop a ANOVA test (Analysis of the variance of one factor).

Variance Analysis

Source of variance	Sum of squares	Freedom degrees	Average of squares	F	Probability	Critical value of F
Intragroup	2213,075	69	76,31293	34,9977	2,90E-182	1,46777
Intergroup	84712,9166	28850	2,18051			
Total	86925,9916	28919				

Table 2: Label: In this table we observe the results of the ANOVA analysis

Since we obtain that $F = 34,99$ which is much bigger than 1,46 (critical value) the null hypothesis can be rejected and the difference between difficulty of boards is proven.

4.3 Board frequency distribution

Although I had already an answer for my research question, as a further step I decided to analyze the frequency distribution of the distances of all different paths in a board. Additionally the fact that our conclusion is based on ANOVA analysis, one of its premises is not ensured -the normal distribution- reinforces the interest of identify which type of distribution follows the board difficulty.

With this purposes, I have used Distribution Analysis, a statistical technique that allows you to identify the best fitted distribution pattern for the sample data.

The underlying concept in this technique is to compare the frequency distribution observed with the one which results from a theoretical (expected) distribution and determine if the differences are significant enough. Usually the test is done with cumulative frequencies.

I do not have any clue of the candidate to compare our sample results, and it would be a long and complex process to compare with all possible distributions. Fortunately, statistical software as XLSTAT provides us with tools to make the comparison with main statistical distributions in a very short period of time, and this is what I will use. (The results of the individual test for each board are not plotted as there are too many information and would not be relevant for our purpose. We will work with the result of the distribution of the *average* of all the board sample -see Annex-).

As a result of the analysis, nearly all boards in the sample present a Poisson distribution with mean in [3-4] range. Let's try to find an explanation to that finding. As we have studied in the IB syllabus, a Poisson distribution is a limit case of a binomial distribution when the number of trials becomes very large (tends to infinite) and the probability of success is small. A binomial distribution describes the action of choosing between two options (each one with its complementary probability) several times. If we analyze this game from a new perspective, we can find similarities to these probability distributions.

Let's imagine a concrete path from a cell a to a cell b. So our variable will be the number of steps we need to arrive to the cell b. In every additional step, we have a probability to have arrived to b, and a probability that we can't arrive there yet. So the probability of success in each step is the product of binomial distributions (arrived - not arrived) where the probability of each step increases as the number of intermediate points increases as the number of steps.

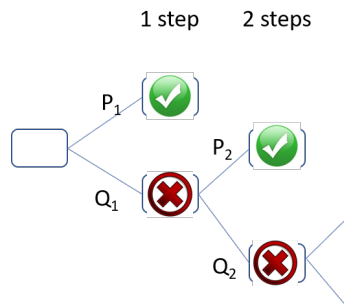


Figure 3: Graphical representation of the probability of distances regarding the number of steps

So, it resembles a Poisson approach as a limit cases of a binomial distribution (with a low probability of success in first step). The main difference is the increasing probability of success as the number of steps increases (due to the new paths added at each step). My calculation of the increase of probability (P1 to P2) results on a non-significant variation, that will be reflected on the graphs that will be shown in next section.

Once we have justified the Poisson distribution, let's try to figure out the value of the mean [3-4]. The whole board is composed of 36 nodes, and each one of them is linked to 10 other nodes. So from the initial node we can eventually move to 10 nodes (5 horizontally + 5 vertically) in the first move and 9 (bearing in mind there is nonsense to return to the previous node). A first glance of how many steps are required to link two nodes could be simply the ratio of number of nodes to be fulfilled (36) by the number of nodes accessible in each step (9 except 10 in first step), that results in 3,88 steps as average.

This result is consistent with the average (3,43) we have found in our sample of 76 boards (see graph below), and the difference is not statistically significant. This approach is clearly a simplification

but a quite straight form to justify the mean of the Poisson distribution we have found.

4.4 Data Analysis

We have already seen with the ANOVA test that all boards are not equivalent in terms of difficulty but lets try to find more information about the difficulties in the different boards.

We will first plot some results in a graphical way, so we can notice that the distribution of average distance of the board sample paths has a clear concentration in the range [2,6] steps, where all the paths above distance 6 represent less than 5 percent of total, with a huge concentration in the range [2-4]. In orange we have plotted a Poisson distribution (with the same mean that we have found in previous section) just to see in a more graphical way the results we obtained in the Board Frequency Distribution.

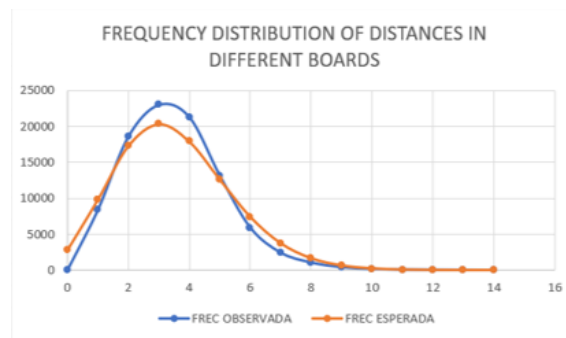


Figure 4: This graphic shows the relative distribution of the average distance of boards: in blue we have the observed frequency, and in orange the expected one (Poisson distribution)

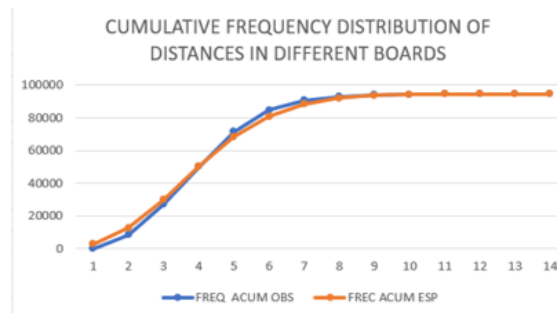


Figure 5: This graphic shows the cumulative frequency of the average distances of boards: in blue we have the cumulative observed frequency, and in orange the cumulative expected one (Poisson distribution)

In fact, the board Mean Distance distribution has a narrow range [3-4] with some exceptions (one of them with Mean above 5 could be considered as an outlier). As we can notice from a visual inspection, our reasoning about the Poisson behaviour of distance distribution is clearly verified in this comparative graph. We have not done any statistical analysis of the differences between observed and expected values, but we can anticipate the differences are not significant to reject our reasoning.

So as a first analysis, we can see the different boards have a mean distance quite similar [3,4] The game is designed in a way that most of the challenges (calculate the distance between points) can be solved in an easy way (implying 4 or less steps). The concentration of distances around mean and the skewness is part of the success of the game: 3 - 4 steps make it neither so easy to be bored

nor so difficult to discourage the players.

Further analysis of the results provided another curious property which I plot down here.

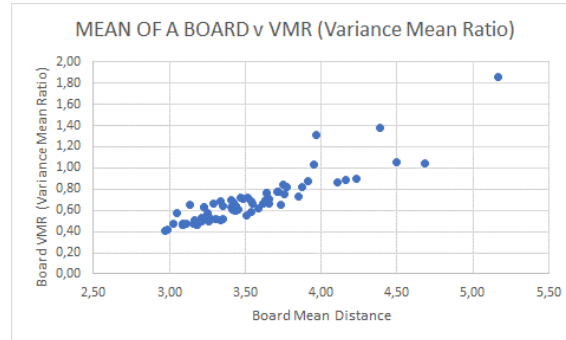


Figure 6: This graphic shows how does the VMR change when the mean increases

We can observe there is a positive relationship between Board Mean Distance and Board Variance (we used Variance Mean Ratio -VMR- as the ratio between variance and mean in each board in order to normalize the data and make all the boards comparable). Although it is not directly related to the work, it's more information that can perhaps help has take some conclusions. After this first approach, I have analyzed the distributions of each board and the combined distribution of all boards in a more detailed way to be able to obtain as much conclusions as I could about my topic.

5 Conclusions

As a first conclusion, and answering to my initial question, I have finally concluded that there are differences between the boards in terms of difficulty.

Once answering the initial question, let's put together a list of different findings that we have been obtaining throughout the work. First of all, you must remember that since we have used sample theory with level of confidence 95 percent and significant error less than 10 percent all our conclusions are statistically sound.

- The different boards do not have the same difficulty level. The variance that we have calculated denies the hypothesis that all boards have the same variance. We have confirmed the intuition that we can have after looking at the results of our sample (Table 1) with the outcomes of ANOVA test (Table 2).
- The mean difficulty of all boards is in a narrow range [3,4], so they are similar.
- The distribution of path difficulty for each board follows a Poisson distribution.
- The mean difficulty of the boards also follows a Poisson distribution.

Referring to conclusion 3, the Poisson distribution, we will recall the argument we have stated before in the Board Frequency Distribution.

We can conclude that although there are differences each time you play with different boards, in the long term, playing several times, the difference vanishes becoming all plays quite are similar: there is no specific a priori rule to determine for a specific board if it will be more or less difficult. What you can expect is quite regular in terms of difficulty, and probably this is one of the main virtues of the game.

5.1 Further work

My work has focused on basic theory (graph, statistics, sampling). When I reached my conclusions, I had the opportunity to discuss this work with Dr Sergio Gómez, an specialist in networks working currently at University Rovira i Virgili. He opened me to leading-edge research in the field of networks, specifically the problem of communicability, a concept quite aligned with my work, and that I had not discovered at the moment when he told me.

Communicability is defined as the shortest path connecting nodes, and in general terms, as a measure of the ability of a network to connect nodes. Immediately I understood this technique could be applied to my work and it is my intention to follow on both in this area and with a more detailed analysis of the different board complexity to identify pattern in the construction of the boards. Although I do not have space to continue with this research on this assessment, it is a very interesting point from where to continue in future projects related with this topic.

6 Bibliography

- GLASS, G.V., PECKHAM, P.V. and SANDERS, J.R. (1972) Consequences of failure to meet assumptions underlying fixed effects analyses of variance and covariance. *Rev. Educ. Res.* 42: 237-288.
- NEWMAN, M. (2010) *Networks. An introduction.* Oxford. Oxford University Press
- SPIEGEL, M. and STEPHENS, L. (2014). *Schaum's outline of statistics.* 3rd ed. New York: McGraw-Hill.
- WILSON, R.J. (1996) *Introduction to graph theory,* 4 ed. Essex, Longman

7 Annex

7.1 Statistical calculus

$$Mean = X = \frac{\sum x_i}{n}$$
$$Deviation = \sqrt{\frac{\sum (x_i - X)^2}{n}}$$

newline

Results of the distribution analysis (provided by XLSTAT) to know what kind of distribution best fits the difficulties of 72 different boards of our sample.

Distribución	valor-p
Binomial negativa(1)	0,000
Binomial negativa(2)	0,009
Exponencial	0,031
Fisher-Tippett(1)	0,033
Fisher-Tippett(2)	0,394
Gamma(1)	0,000
Gamma(2)	0,622
Gumbel	0,000
Log-normal	0,692
Logística	0,172
Normal	0,242
Normal(estándar)	0,000
Poisson	0,911
Student	0,000
Weibull(1)	0,000
Weibull(2)	0,602

Table 3: Results of the different board distributions

7.2 Program

```
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
class Peca(object):
def init(self,llista):
self.caselles = llista
self.orientacio = 'N'
def dalt(self):
if self.orientacio == 'N':
return self.caselles[0:3]
elif self.orientacio == 'E':
return [self.caselles[6],self.caselles[3],self.caselles[0]]
elif self.orientacio == 'O':
return [self.caselles[2],self.caselles[5],self.caselles[8]]
else:
return [self.caselles[8],self.caselles[7],self.caselles[6]]
def migh(self):
if self.orientacio == 'N':
return self.caselles[3:6]
elif self.orientacio == 'E':
return [self.caselles[7],self.caselles[4],self.caselles[1]]
elif self.orientacio == 'O':
return [self.caselles[1],self.caselles[4],self.caselles[7]]
else:
return [self.caselles[5],self.caselles[4],self.caselles[3]]
def baix(self):
if self.orientacio == 'N':
return self.caselles[6:9]
elif self.orientacio == 'E':
return [self.caselles[8],self.caselles[5],self.caselles[2]]
elif self.orientacio == 'O':
return [self.caselles[0],self.caselles[3],self.caselles[6]]
else:
return [self.caselles[2],self.caselles[1],self.caselles[0]]
def esquerra(self):
if self.orientacio == 'N':
return [self.caselles[0],self.caselles[3],self.caselles[6]]
elif self.orientacio == 'E':
return self.caselles[6:9]
elif self.orientacio == 'O':
return [self.caselles[2],self.caselles[1],self.caselles[0]]
else:
return [self.caselles[8],self.caselles[5],self.caselles[2]]
def migv(self):
if self.orientacio == 'N':
```

```

return [self.caselles[1],self.caselles[4],self.caselles[7]]
elif self.orientacio == 'E':
return self.caselles[3:6]
elif self.orientacio == 'O':
return [self.caselles[5],self.caselles[4],self.caselles[3]]
else:
return [self.caselles[7],self.caselles[4],self.caselles[1]]
def dreta(self):
if self.orientacio == 'N':
return [self.caselles[2],self.caselles[5],self.caselles[8]]
elif self.orientacio == 'E':
return self.caselles[0:3]
elif self.orientacio == 'O':
return [self.caselles[8],self.caselles[7],self.caselles[6]]
else:
return [self.caselles[6],self.caselles[3],self.caselles[0]]
p1 = Peca(['5P','2P','3P','6R','4R','3Y','4B','5G','3G'])
p2 = Peca(['1Y','5R','1G','6Y','5B','1R','1P','5Y','6P'])
p3 = Peca(['5W','4G','3B','3W','4Y','3R','2B','6B','6W'])
p4 = Peca(['4P','2W','2Y','6G','2G','1W','4W','2R','1B'])
class Taulel (object):
def init (self,llistapeces,llistaor):
self.peces = llistapeces[:]
for i in range(len(llistapeces)):
self.peces[i].orientacio = llistaor[i]
def repr(self):
fila1 = cadena(self.peces[0].dalt() + self.peces[1].dalt())
fila2 = cadena(self.peces[0].migh() + self.peces[1].migh())
fila3 = cadena(self.peces[0].baix() + self.peces[1].baix())
fila4 = cadena(self.peces[2].dalt() + self.peces[3].dalt())
fila5 = cadena(self.peces[2].migh() + self.peces[3].migh())
fila6 = cadena(self.peces[2].baix() + self.peces[3].baix())
return fila1 + fila2+ fila3+ fila4+ fila5+ fila6
def columna(self,n):
if (n==1):
return cadena1(self.peces[0].esquerra() + self.peces[2].esquerra())
if (n==2):
return cadena1(self.peces[0].migv() + self.peces[2].migv())
if (n==3):
return cadena1(self.peces[0].dreta() + self.peces[2].dreta())
if (n==4):
return cadena1(self.peces[1].esquerra() + self.peces[3].esquerra())
if (n==5):
return cadena1(self.peces[1].migv() + self.peces[3].migv())
if (n==6):
return cadena1(self.peces[1].dreta() + self.peces[3].dreta())
def fila(self,n):

```

```

if (n==1):
return cadena1(self.peces[0].dalt() + self.peces[1].dalt())
if (n==2):
return cadena1(self.peces[0].migh() + self.peces[1].migh())
if (n==3):
return cadena1(self.peces[0].baix() + self.peces[1].baix())
if (n==4):
return cadena1(self.peces[2].dalt() + self.peces[3].dalt())
if (n==5):
return cadena1(self.peces[2].migh() + self.peces[3].migh())
if (n==6):
return cadena1(self.peces[2].baix() + self.peces[3].baix())
def graph (taulel):
p = []
for i in range(1,7):
p += taulel.fila(i).split()
g =nx.Graph()
g.addnodesfrom (p)
for i in range (6):
f = taulel.fila(i+1).split()
for x in range (5):
n = f[x][0]
c = f[x][1]
for j in range (x+1,6):
if n==f[j][0]:
g.addedge(f[x], f[j])
if c==f[j][1]:
g.addedge(f[x], f[j])
for i in range (6):
col = taulel.columna(i+1).split()
for x in range (5):
n = col[x][0]
c = col[x][1]
for j in range (x+1,6):
if n==col[j][0]:
g.addedge(col[x], col[j])
if c==col[j][1]:
g.addedge(col[x], col[j])
return g
def esgroc(x):
return 'Y' in x
def esvermell(x):
return 'R' in x
def esrosa(x):
return 'P' in x
def esverd(x):
return 'G' in x

```

```

def esblanc(x):
return 'W' in x
def esblau(x):
return 'B' in x
def dibuixa (graf):
vermells = list(filter(esvermell,graf.nodes()))
grocs = list(filter(esgroc,graf.nodes()))
verds = list(filter(esverd,graf.nodes()))
roses = list(filter(esrosa,graf.nodes()))
blaus = list(filter(esblau,graf.nodes()))
blancs = list(filter(esblanc,graf.nodes()))
pos = nx.circular_layout(graf)
nx.draw(graf,pos,with_labels = True)
nx.draw_networkx_nodes(graf,pos,nodelist=grocs,node_color = 'y')
nx.draw_networkx_nodes(graf,pos,nodelist=vermells,node_color = 'r')
nx.draw_networkx_nodes(graf,pos,nodelist=blaus,node_color = 'b')
nx.draw_networkx_nodes(graf,pos,nodelist=verds,node_color = 'g')
nx.draw_networkx_nodes(graf,pos,nodelist=roses,node_color = 'm')
nx.draw_networkx_nodes(graf,pos,nodelist=blancs,node_color = 'w')
plt.show()
def matriu (graph):
m = nx.adjacency_matrix(graph)
m= m.todense()
return m
def passos(m): trobat = False matriu_inicial = m
n = 0
while not trobat :
n += 1
for i in range(36):
for j in m.tolist()[i] :
break else :
return n
m = sum([matriu_inicial * i for i in range(n)])
def cadena (llista):
string=""
for element in llista:
string+=(element + ' ')
string+=" "
return string
def cadena1 (llista):
string=""
for element in llista:
string+=(element + ' ')
return string.strip()
def dificultat (morig): Aux=np.full((36,36), False, dtype=bool)
a= []
b=0
l=[]
power = 1
while not np.all(Aux):
b = 0
m = np.linalg.matrix_power(morig, power)
for i in range(36):
for j in range(36):
if m[i,j] != 0 and Aux[i][j] == False:
Aux[i][j]= True
l.append(power)
power += 1

```

```
a.append(b)
print (l)

if name == 'main':
    T = Taulel ([p1,p2,p3,p4],[ "N", "N", "N", "N"])
    G = graph(T)
    print(dificultat (matriu (G)))
    np.setprintoptions(threshold=np.nan)
```