Report for the Research Track II course.

Ramona Ferrari
Department of Computer Science and Technology, Bioengineering, Robotics and System Engineering
DIBRIS
Genoa, Italy
s4944096@studenti.unige.it

***Abstract*—This document serves as a report for the Research Track II course. It focuses on the comparative analysis of two distinct algorithms and conducts a statistical examination of the initial assignment in the Research Track I course. The primary objective of both algorithms is to enable a robot to retrieve all tokens and transport them to a designated destination. Following an introduction to the background and problem statement, the author presents a comprehensive overview of the methodologies employed, alongside a detailed description of the experimental procedures. Subsequently, the report delves into the analysis of the obtained results and concludes with a discussion of the findings.**

I. INTRODUCTION ABOUT RESEARCH TRACK I FIRST ASSIGNMENT.

A. Description of the enviroment.

The simulation environment utilized for this study is a 2D simulator, meticulously crafted to emulate the movement dynamics of a robotic agent traversing a fixed playground strewn with stationary tokens.
Developed by Student Robotics, this simulator offers a controlled setting wherein robots can detect, grasp, relocate, and release tokens with precision.
The primary objective of the assigned task is to program a Python node capable of orchestrating the robotic agent to consolidate all golden tokens within the playground. The initial conditions of the simulation depict the robot positioned in the upper left corner, facing towards the center, with a circular arrangement of six golden tokens distributed around it.
Notably, the algorithm's design is conceived to be agnostic to the precise number of tokens present, ensuring versatility and adaptability across varying scenarios.

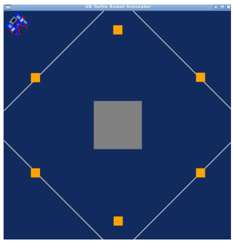

Fig. 1: The initial configuration of robot and tokens

II. THE ALGORITHMS.

Let's proceed to describe the algorithms used for conducting the experiments. The algorithms were developed by the author of this report and Alberto Di Donna. Both algorithms were developed independently of each other, with their respective authors never having engaged in direct collaboration or comparison.

A. Algorithm A

The basic idea of this code is enabling the robot to achieve comprehensive visibility of all tokens within its vicinity by executing a rotational motion, thus scanning its environment in its entirety. Subsequently, the robot selects the initial token encountered during its rotational sweep and proceeds to transport it to a predefined destination, namely, the position of the last token observed. This process iterates successively, as the robot systematically locates, retrieves, and relocates each token to the vicinity of the destination point. Such a procedural strategy ensures the methodical consolidation of all tokens in close proximity to the designated destination.

B. Algorithm B

In this algorithm, the following strategy is employed: the first step involves identifying the nearest token, moving to it, and grasping it. Subsequently, this token is transported to the nearest token, serving as the collection point, and released there. The next phase entails identifying another token, disregarding the one already deposited at the collection point, and repeating the process of reaching, grasping, and transporting it to the collection point.

This sequence of actions is iterated until all tokens have been gathered at the designated collection point.

III. RESEARCH TRACK II ASSIGNMENT.

A. *Hypotheses made*

For conducting a statistical analysis it's fundamental to define the hypothesis the author wants to prove. An hypothesis is a statement which makes a prediction about something which is not proven. In this case, the **Null Hypothesis** is H_0: it is not possible to declare the two algorithms, A and B, present any relevant difference in time execution, so they are equally efficient.
The **Alternative Hypothesis** is H_a: one of the algorithms performs in different time with respect to the other one, so their performances are different. Another important concept is the **Level of Significance**, a percentage representing how it's possible to trust in the conclusions.
All experiments are conducted employing distinct environmental configurations to introduce variability (and complexity in some cases), thereby elucidating the advantages and disadvantages of the algorithms. Modifications are made in the file "*two_colours_assignment_arena*".
H_0 is the hypothesis to prove and H_a is the one to reject; this means this report would prove the two algorithms performs equally with respect to time execution.

B. *Description and motivation of the experimental setup*

When comparing the two algorithms directly, without any modifications, it becomes evident that algorithm A outperforms algorithm B due to its predefined velocity settings. Roughly, algorithm A requires approximately 55 seconds, whereas algorithm B necessitates 378 seconds for completion. However, these time measurements alone are insufficient to assert the superiority of algorithm A over algorithm B.
Given that the algorithms are founded on two **distinct approaches** to solving the problem, it is necessary to **standardize the velocity** of the robot across both algorithms. Otherwise, the results may be unduly influenced by the chosen velocities, potentially compromising the integrity of the conclusion. Therefore, whenever the robot is required to turn or drive in a straight line, the functions $turn(10, 0.1)$ and $drive(50, 0.5)$ are respectively invoked.
These values have been chosen arbitrarily, yet with the aim of ensuring that the robot does not take an excessive amount of time to execute the task while still retaining the capability to fulfill the task of bringing all tokens close together.

EXPERIMENT NUMBER	ALGORITHM A (s)	ALGORITHM B (s)	FAILURES
1	116	175	N
2	115	222	N
3	115	140	N
4	117	199	N
5	115	210	N
6	114	189	N

Fig. 2: Data from experiments without changes

In the image displayed in Figure 2, experimental data are presented within a table format. Notably, the values are denoted in seconds, truncated to whole numbers, and have been measured utilizing the *time* library. Furthermore, the presence of "Y" or "N" in the final column indicates whether the robot fails the task; in this case **failure** means the robot does not see all the boxes or does not carry them in the target position or the node does not finish. This report does not focus on collision between robot and tokens.
Changing the number of gold tokens, the following data are collected:

NUMBER OF BOXES	ALGORITHM A (s)	ALGORITHM B (s)	FAILURES A	FAILURES B
1	5	infinite	N	Y
2	31	25	N	N
3	53	3	N	Y
5	100	29	N	Y
7	131	79	N	Y
8	149	42	N	Y
10	186	14	N	Y

Fig. 3: Data from experiments changing number of boxes

Another kind of experiment can be done changing the the value of the inner and outer radius in the code (so, the position of the tokens); it's possible to see data collected in the following table:

INNER RADIUS	OUTER RADIUS	ALGORITHM A (s)	ALGORITHM B (s)	FAILURES A	FAILURES B
0.1	1.9	93	178	N	N
0.8	0.8	75	66	N	N
1	2.6	121	19	N	Y
2.6	0.9	76	64	N	N
2	2	108	79	N	N
0.3	1.9	93	177	N	N
0.6	2.5	118	120	N	N

Fig. 4: Data from experiments changing inner and outer radius from the code

In the last kind of experiments, the number of tokens is fixed to 3 and the position of the tokens are random in the arena. The collected data are the following:

EXPERIMENT VALUE	ALGORITHM A (s)	ALGORITHM B (s)	FAILURES A	FAILURES B
1	38	36	N	N
2	29	37	N	N
3	29	48	N	N
4	44	26	N	N
5	56	30	N	N
6	41	13	N	Y
7	51	47	N	N
8	46	35	N	N
9	24	16	N	N
10	39	42	N	N

Fig. 5: Data from experiments with 3 tokens and random position of tokens

C. Results

The results are showed in a summary table and whenever there is a failure, the algorithm is penalized with $+200$ seconds and the cell is marked with red colour.

EXPERIMENT NUMBER	ALGORITHM A	ALGORITHM B
1	116	175
2	115	222
3	115	140
4	117	199
5	115	210
6	114	189
7	5	200
8	31	25
9	53	203
10	100	229
11	131	279
12	149	242
13	186	214
14	93	178
15	75	66
16	121	219
17	76	64
18	108	79
19	93	177
20	118	120
21	38	36
22	29	37
23	29	48
24	44	26
25	56	30
26	41	213
27	51	47
28	46	35
29	24	16
30	39	42

Fig. 6: Summary of all experiments

In addition, it's possible to compare the time execution of the algorithms thanks to the following graph:

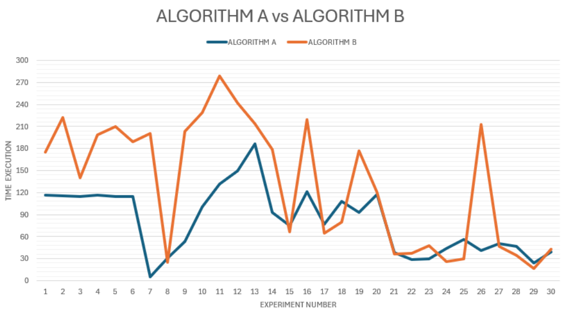

Fig. 7: Time comparison of algorithms

In most experiments, as it is possible to see in Figure 7, the execution time of algorithm B is consistently greater than that of algorithm A, thus suggesting that the two algorithms perform differently in time execution.

D. Discussion of the results with statistical analysis

Concerning the statistical analysis, it is possible to prove what said previously and confirm or reject the alternative/null hypothesis, using **T-test**. The first step is to compute the **arithmetic mean** for both algorithms, dividing the sum of all time execution by the total number of experiments.

$$\overline{x}_1 = \frac{\sum_{n=1}^{N_1} X_i}{N_1} = 80,970701832$$

$$\overline{x}_2 = \frac{\sum_{n=1}^{N_2} X_i}{N_2} = 132,040571872$$

where N_1 and N_2 = 30 and X_i are the values of the time.

Then, the **standard deviation** is computed to provide how close the values are to the mean value.

$$\sigma_1 = \sqrt{\frac{\sum_{n=1}^{N_1}(X_i - \overline{x}_1)^2}{N_1}} = 44,239871392$$

$$\sigma_2 = \sqrt{\frac{\sum_{n=1}^{N_2}(X_i - \overline{x}_2)^2}{N_1}} = 85,136463824$$

At this point, it is suitable to use ***two-Sample T-test***, calculating the **Pooled Variance** and the **Pooled Estimated Standard Error**:

$$\hat{\sigma}^2_{\text{pooled}} = \frac{(N_1 - 1)\cdot\sigma_1^2 + (N_2 - 1)\cdot\sigma_2^2}{N_1 + N_2 - 2} = 4449,268785081$$

$$\hat{\sigma}_{\overline{x}_1 - \overline{x}_2} = \sqrt{\frac{\sigma^2_{\text{pooled}}}{N_1} + \frac{\sigma^2_{\text{pooled}}}{N_2}} = 17,222599078$$

The **t-statistics** turns into:

$$t_{\overline{x}_1 - \overline{x}_2} = \frac{\overline{x}_1 - \overline{x}_2}{\hat{\sigma}_{\overline{x}_1 - \overline{x}_2}} = -2,965282406$$

The computed value is taken in absolute value: $2,965282406$.

E. Conclusion

With 58 degrees of freedom and a significance level of 5%, the critical value of the two-tailed t-distribution is approximately $2,001$. Therefore, if the absolute value of t-test exceeds 2.001, the *null hypothesis is rejected* at a significance level of 5%. Since, in this case, t-test value is $2,965$, which is more than 2.001, it is correct to reject the null hypothesis.

In contrast, it is possible to *confirm the alternative hypothesis*.

IV. Further Comments

The tables, graphs, and all computational analyses were conducted using the *Excel* software. Furthermore, the value of the *two-tailed T-test* was computed utilizing an online calculator accessible at the following link: Calculator Website. This calculator facilitated the statistical computations necessary for this analysis. Moreover, an additional statistical consideration involves examining the frequency of failures as opposed to focusing solely on execution time. This avenue presents an intriguing opportunity for further investigation and analysis within the scope of this research.