

INSTITUTO TECNOLÓGICO DE AERONÁUTICA



Arthur José de Sousa Rodrigues

**MULTI-AGENT GRAPH EXPLORATION WITHOUT
COMMUNICATION**

Bachelor's Thesis
2023

Computer Engineering

Arthur José de Sousa Rodrigues

**MULTI-AGENT GRAPH EXPLORATION WITHOUT
COMMUNICATION**

Advisor

Prof. Dr. Luiz Gustavo Bizarro Mirisola (ITA)

Co-advisor

Prof. Dr. Vitor Venceslau Curtis (ITA)

COMPUTER ENGINEERING

SÃO JOSÉ DOS CAMPOS
INSTITUTO TECNOLÓGICO DE AERONÁUTICA

2023

Cataloging-in Publication Data
Documentation and Information Division

de Sousa Rodrigues, Arthur José
Multi-agent graph exploration without communication / Arthur José de Sousa Rodrigues.
São José dos Campos, 2023.
26f.

Bachelor's Thesis – Course of Computer Engineering– Instituto Tecnológico de Aeronáutica,
2023. Advisor: Prof. Dr. Luiz Gustavo Bizarro Mirisola. Co-advisor: Prof. Dr. Vitor Venceslau
Curtis.

1. Maze. 2. Graph. 3. Search. 4. Multi-agent. I. Instituto Tecnológico de Aeronáutica. II. Title.

BIBLIOGRAPHIC REFERENCE

DE SOUSA RODRIGUES, Arthur José. **Multi-agent graph exploration without communication**. 2023. 26f. Bachelor's Thesis – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSION OF RIGHTS

AUTHOR'S NAME: Arthur José de Sousa Rodrigues

PUBLICATION TITLE: Multi-agent graph exploration without communication.

PUBLICATION KIND/YEAR: Bachelor's Thesis / 2023

It is granted to Instituto Tecnológico de Aeronáutica permission to reproduce copies of this work and to only loan or to sell copies for academic and scientific purposes. The author reserves other publication rights and no part of this work can be reproduced without the authorization of the author.

Arthur José de Sousa Rodrigues
Rua H8B 226
12.228-461 – São José dos Campos–SP

MULTI-AGENT GRAPH EXPLORATION WITHOUT COMMUNICATION

Bachelor's Thesis approved in its non-final version by the signatories below:

Arthur José de Sousa Rodrigues
Author

Luiz Gustavo Bizarro Mirisola (ITA)
Advisor

Vitor Venceslau Curtis (ITA)
Co-advisor

Prof. Dr. Marcos Ricardo Omena de Albuquerque Máximo
Course Coordinator of Computer Engineering

São José dos Campos: Jun 16, 2023.

To my mother Ana Paloma, daughter of
Antônio and Vandei. To my father Jairo
José, son of José and Rosinete.

Acknowledgments

I thank my family for invest time and financial resources in my studies. And to give me unconditionally love and motivation.

I thank my H8 friends for always support me with positive words.

I thank my advisor Prof. Dr. Luiz Gustavo Bizarro Mirisola and my co-advisor Prof. Dr. Vitor Venceslau Curtis for all patience and attention that has been given to me.

I thank to Brazilian Air Force for support my studies since I joined the Air Cadet Preparatory School (EPCAr).

“Now to him who is able to do immeasurably more than all we ask or imagine, according to his power that is at work within us, to him be glory in the church and in Christ Jesus throughout all generations, for ever and ever”

— SAINT PAUL

Resumo

Algoritmos computacionais de busca têm sido estudados por cientistas e companhias de engenharia desde o último século devido às aplicações de tais algoritmos em situações reais, como agendamento de linhas aéreas, planejamento de rota em mapas, algoritmos de busca na Internet, planejamento de rotas em redes de computadores, robótica, etc. Usualmente, esses métodos se baseiam em tipos abstratos de dados, como grafos e árvores, para transferir um problema real para dentro de um contexto delimitado e mais compreensivo. No campo da Ciência da Computação, grafos são tipos abstratos de dados que podem servir como suporte ferramental para algoritmos de busca.

De uma abordagem relacionada a algoritmos de busca em labirintos, este trabalho de graduação propõe um método para explorar grafos em um âmbito multiagente, em que os agentes não são capazes de se comunicarem entre eles, o que faz referência a problemas da realidade, como exploração no fundo do mar, busca em estruturas compostas por muros maciços, busca em locais inóspitos por agentes com restrição energética, etc.

Abstract

Computational search algorithms have been studied by many scholars and engineering companies since the last century due to the real-life applications of such algorithms, such as in airline scheduling, planning path on maps, search engine algorithms, social media marketing, Internet routing protocols, robotics, etc. Usually these methods rely on abstract data types, including graphs and trees, to transfer a real-world problem into a delimited and more comprehensive environment. In the context of computer science, graphs are abstract data type that might support computational search algorithms.

From the maze-solving algorithms approach, this work aims to propose a method to explore a graph in a multi-agent situation, where the agents cannot communicate with each other, that makes reference to real-life problems, like deep sea exploration, search in large wall structures, search with low energy-based agents in inhospitable environments.



List of Figures

FIGURE 2.1 – Blue and red agents traversing a 6×6 maze. The goal is represented by the green entity. This maze is a perfect maze (MUHAMMAD, 2021).	20
FIGURE 2.2 – Maze cell indices. Source: Muhammad (2021).	21
FIGURE 2.3 – Red agent traversing a maze toward the green goal. This maze is not a perfect maze.	21
FIGURE 2.4 – Graph representation of the maze presented in Figure 2.3. The cell indices are pointed out. Furthermore, the visited cells by the red agent are also pointed out.	21
FIGURE 2.5 – Graph representation of the maze presented in Figure 2.3, with intervals related to the agent's range of action.	23

List of Tables

List of Abbreviations and Acronyms

DFS Depth First Search

BFS Breath First Search

List of Symbols

Contents

1	INTRODUCTION	14
1.1	Motivation	14
1.2	Related work	15
1.3	Definitions	16
1.3.1	Graph	16
1.3.2	Maze	16
1.3.3	Agent	17
1.3.4	Maze-solving algorithms	17
1.3.5	Mixed Radix	17
2	MODELS	19
2.1	Maze and multi-agent exploration	19
2.2	Maze from a graph topology perspective	20
2.3	Multi-agent exploration without communication	20
3	RESULTS	24
4	NEXT STEPS	25
	BIBLIOGRAPHY	26

1 Introduction

This graph exploration work aims to discuss the main maze exploration algorithms and then propose a method where multi-agents must find the maze solution without any type of communication, only working with probability distributions to guide their behaviors. The main challenge of this research is to find a distributed exploration approach with total communication restriction since an agent's partial knowledge cannot be shared with another agent and, at the same time, a single agent must avoid repeating a branch of another agent. The proposed maze structure abstraction is a traditional regular grid that can be generalized into graphs.

This report is the initial part of the bachelor's thesis, and this chapter intends to introduce the general concept of the related thesis and present the motivation (Section 1.1), the related work (Section 1.2), and the main definitions about maze-solving algorithms (Section 1.3).

1.1 Motivation

Graph exploration has been the target of studies since Leonhard Euler proved that Seven Bridges of Königsberg (SHIELDS, 2012) has no solution. It has been researched not only in academia but also in the industry due to several practical applications, like airline scheduling, planning path on maps, search engine algorithms, social media marketing, Internet routing protocols, and robotics.

Specifically in robotics, graph exploration can be used to explore a maze with a single agent through a bunch of traditional algorithms: random mouse, wall follower, Trémaux, etc (SADIK et al., 2010). And it can be useful to guide many real-life problems such as search in nuclear plant disasters, burning buildings, and extraterrestrial environments. In these previous examples, the multi-agent exploration is more interesting than the single-agent approach since it can speed up the exploration. Recent studies have explored multi-agent maze-solving algorithms as seen in the Multi-Agent Maze Exploration paper (KIVELEVITCH; COHEN, 2010), where authors proposed a Tarry's algorithm generalization. It is important to emphasize that maze-solving algorithms consider that the structure is

unknown, and then traditional search algorithms, such as Dijkstra and A*, cannot be used to solve the maze.

Traditional multi-agent maze exploration approaches is based on internal communication between agents, where each agent knows about visited cells by another agent. It avoids a second exploration in a useless path and thus it decreases computational costs. However, there are some real situations where communication is limited or impossible, such as deep sea exploration, search in large wall structures, or search with low energy-based autonomous agents.

However, the incommunicable approach between agents have not been concretely found in the literature despite it may have real applications and may guide search plans in real-world problems. In order to explore it, this work presents some ways to achieve the solution of a maze based on agents without communication, that might be generalized to graph exploration algorithms.


1.2 Related work


Multi-agent cooperative system approaches are common in literature, especially when it comes from robotics researches. In the context of communicable agents, mainly in robotics, this chapter presents some related works.


Mataric (1995) established common properties across different scenarios of mobile multi-agent interactions, such as dispersion - “the ability of a group of agents to spread out in order to establish and maintain some minimum interagent distance” -, aggregation - “the ability of a group of agents to gather in order to establish and maintain some maximum interagent distance” -, homing - “the ability of an agent to find a particular region or location” -, etc. The author proposes a synthetic structure in order to abstract different types of interagent basis behaviors.

Burgard et al. (2005) pointed out that an exploration group of robots takes several advantages over single agent exploration, despite a coordinating group might introduce redundancy. The authors present an algorithm to efficiently explore an environment by mobile and autonomous robots within a communication range.

Sadik et al. (2010) presents, as seen in the title of the paper, a comprehensive and comparative study of maze-solving algorithms techniques by implementing graph theory. The research is delimited in the “Micromouse competition” context, which is a famous maze competition that has been performed worldwide since late 1970s. The authors compared maze-solving methods based on graph theory algorithms, such as DFS (Depth First Search) and BFS (Breath First Search) flood-fill, to common algorithms in the

“Micromouse” context, such as Wall Follower. They concluded that, despite graph theory algorithms demands higher computational complexity, they are more proficient compared to non-graph theory methods. 


Based around maze-solving algorithms, Kivelevitch & Cohen (2010) proposed a generalization of Tarry’s algorithm, but the new approach is that all visited cells of the maze are known by each agent, since each agent shares its knowledge with all the others. In that sense, each one holds an dynamic map of the maze excluding redundant information, allowing information sharing. The authors presents in the article the performance of the proposed solution, where a group of virtual coordinate agents is required to find the goal without an a-priori knowledge of the maze, so-called “maze exploration”. 

Beisel (2014), similarly to aforementioned, worked in simulation and mathematical analysis for strategies related to cooperative autonomous robots, that can share messages with each other. The author concluded that a cooperative approach might result in significant performance improvements. 

1.3 Definitions

Mainly considering these previous references, this chapter intends to define common terms about maze-solving algorithms, graph theory, and mathematical tools that were useful to abstract the proposed solution. Furthermore, it intends to clarify the approach domain of this work.

1.3.1 Graph


In the context of computer science, a graph is a abstract data type that can be represented topologically as a interconnected node network. Several computational problems are discussed in terms of graphs. 

It is worth to mention that there is a subdomain of graph data type called tree. A tree is also a abstract data type, which can be represented topologically as a hierarchical structure of nodes. In fact, a tree is a subset of graph theory, therefore a tree is a graph under a different perspective. In Section 2.2, a perspective of a maze from a tree and a graph topologies will be explored.


1.3.2 Maze

This report define a maze at the same perspective presented in Kivelevitch & Cohen (2010). A maze is a n-dimensional gridded space of any size, usually rectangular. The

gridded space is composed by a set of cells, while a cell is the elementary item of a maze, defined as a delimited n-dimensional space. Cells might be connected or not connected to another adjacent cell, separated by a “wall” in the latter case. Without losing generality, as will be presented in Section 2.1, this work considered a maze, for simulation purposes, as a two-dimensional gridded space composed by two-dimensional bounded cells.

Thus, from the above definition, a graph or a tree are good candidates to abstract a maze, which might be described as a interrelated nodes (cells) network. 

1.3.3 Agent


As defined in Kivelevitch & Cohen (2010), an agent is an autonomous entity that can traverse the maze obeying the interrelation of the cells. 

In a maze context, a multi-agent approach describes the coordinating behavior of several autonomous agents.

1.3.4 Maze-solving algorithms

A maze has a goal to be achieved, which is usually a single marked cell or a path for an agent to exit the maze. Since the last century, many scholars have been studied algorithms to solve the maze computationally, and the proposed solutions are so-called “maze-solving algorithms”.

1.3.5 Mixed Radix

Mixed radix is a numerical representation that does not rely on standard positional numeral systems. In a traditional numerical system, there is not a base variation through the numbers, nevertheless, in a mixed radix representation, a number is represented by a sequence of numerals in different bases, where each numeral is a multiple of the previous numerical sequence, however relied on a different base. 

For example, Arndt (2011) establishes a arithmetical method to manipulate a specific subset of mixed radix numbers. The mixed radix representation $A = [a_0, a_1, a_2, \dots, a_{n-1}]$ of a number x with respect to a radix vector $M = [m_0, m_1, m_2, \dots, m_{n-1}]$ is given by:

$$x = \sum_{k=0}^{n-1} a_k \prod_{j=0}^{k-1} m_j \quad (1.1)$$

where $0 \leq a_j < m_j$ (and $0 \leq x < \prod_{j=0}^{n-1} m_j$, so that n digits suffice). In a traditional positional numerical system, the vector M is like $M = [r, r, r, \dots, r]$, and then the relation


is simply given by:


$$x = \sum_{k=0}^{n-1} a_k r^k \tag{1.2}$$


Despite the aforementioned representation, this work presents a slightly different approach in Section 2.3.


2 Models

2.1 Maze and multi-agent exploration

For simulation purposes, this work modeled a maze under  an interrelated cells perspective, where each cell is a square with its edges composed by a wall or not. If there is a wall, an agent cannot traverse the maze across the related edge. On the other hand, if there is not a wall, an agent has a free way to traverse the maze through the related edge. It is worth to mention that, if an edge has a wall, the edge of the adjacent corresponding cell also has necessarily a wall.

The maze has a goal that is a single marked cell, and an agent inside the maze aims to find the marked cell, traversing the maze cell by cell. This agent is an autonomous entity that follows a specific algorithm according the current explored path and its programmed initial rules. So that it doesn't go through the same path more than one time, it stores the visited cells. Thus, every time the agent find a branched cell (a cell with at least one edge without a wall), it ignores the already visited branches by itself, and, if there is not a candidate to be a possible branch, the agent go back to the previous visited cell. Furthermore, specifically to this research, differently from some approaches presented in Beisel (2014), Burgard et al. (2005), and Kivelevitch & Cohen (2010), there is no intercommunication between agents in a multi-agent situation to solve the maze. 

 Muhammad (2021) developed an open-source maze generator. It is a python module that creates randomly mazes and enables the user to simulate its own maze-solving algorithm. In this context, this work presents a multi-agent maze-solving algorithm that has been simulated over Muhammad (2021) open-source code, with modifications. The Figure 2.1 presents, for example, 2 agents traversing a 6×6 maze toward the goal.

Muhammad (2021) software creates by default a “perfect maze”, which means that there is one and only one path to the goal from any cell. However, it is possible to set the code to generate a imperfect maze. 

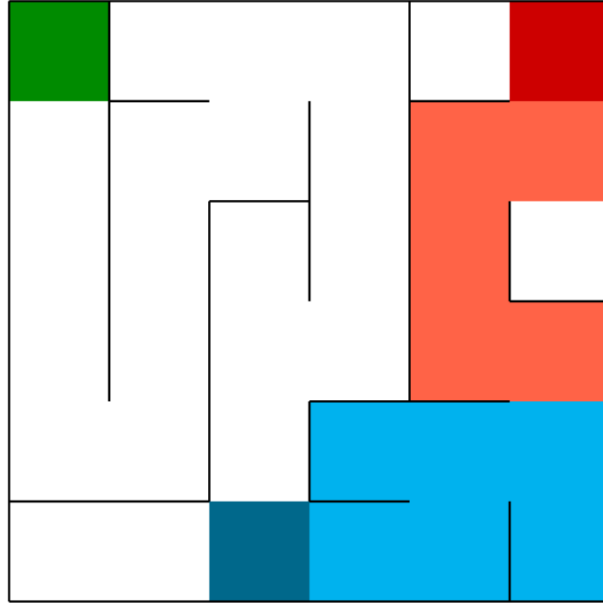


FIGURE 2.1 – Blue and red agents traversing a 6×6 maze. The goal is represented by the green entity. This maze is a perfect maze (MUHAMMAD, 2021).

2.2 Maze from a graph topology perspective

As pointed out in Section 2.1, given that an agent ignores visited cells, there are important statements related to this work:

- if there is only one path to the goal from any cell, it is valid to consider a maze as a tree, i.e., a perfect maze (MUHAMMAD, 2021);
- if there is more than one path to the goal from any cell, the maze cannot be considered as a tree, but it might be considered as a graph;
- despite the last statement, and also considering that an agent ignores visited cells, an agent path always might be individually interpreted as a tree.

Figure 2.4 gives an example of the graph representation of the maze presented in Figure 2.3, where a red agent is traversing the maze toward the green goal. As established by Muhammad (2021), the maze cells are programmatically addressed as indices of a matrix, such as represented in Figure 2.2.

2.3 Multi-agent exploration without communication

The goal of this work is to present a maze-solving algorithm in a multi-agent environment, where an agent cannot communicate with the other agents. Thus, each agent must

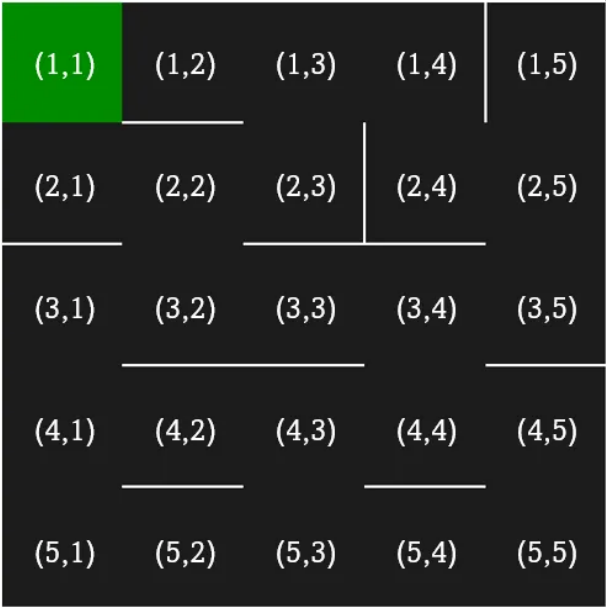


FIGURE 2.2 – Maze cell indices. Source: Muhammad (2021).

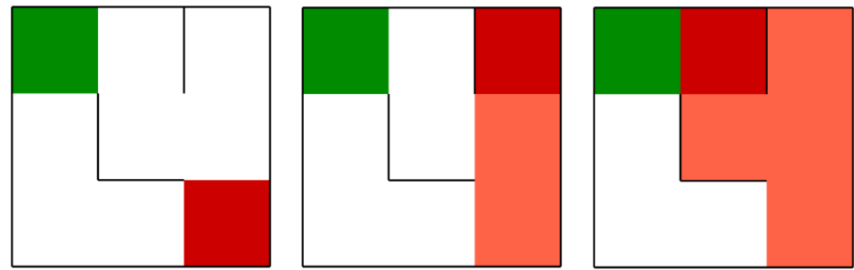


FIGURE 2.3 – Red agent traversing a maze toward the green goal. This maze is not a perfect maze.

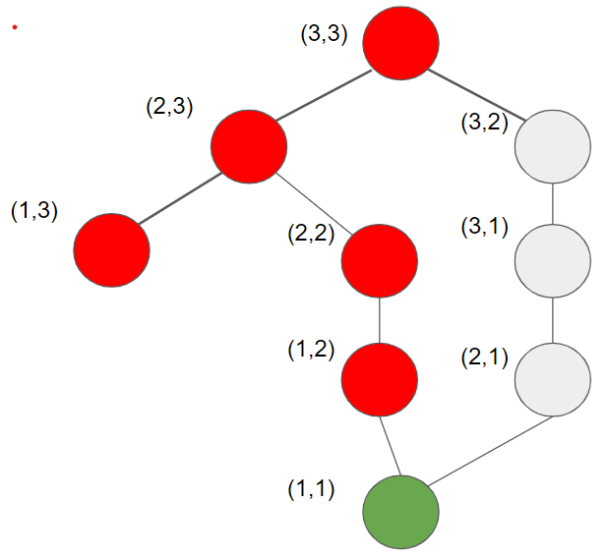





FIGURE 2.4 – Graph representation of the maze presented in Figure 2.3. The cell indices are pointed out. Furthermore, the visited cells by the red agent are also pointed out.

be previously programmed to avoid traversing the maze through the same way related to another agent. 


First of all, this research considers a maze as a graph, where each node is a cell representation of the maze. Given a initial node that the agent starts its path through it, the agent checks if the node has children. Supposing that a node of the graph has at least one child, i.e., there are not completely walled cells, important statements are established:

- if an agent finds the node where is the goal, the agent finishes its path;
- if an agent is not in the node where is the goal and the node has only one child, the agent necessarily goes through this node child;
- if an agent is not in the node where is the goal and the node has more than one child, the agent needs to make a decision about which child to choose; 

To establish an decision algorithm to the last statement, this work proposes firstly a interval division related to the agents and the children. Supposing that there are n agents a_1, a_2, \dots, a_n to explore the maze, each agent will have a corresponding and proportional range of action related to the graph, as presented below:

$$\begin{aligned}
 d &= 1/n \\
 a_1 &: [0, d[\\
 a_2 &: [d, 2d[\\
 a_3 &: [3d, 4d[\\
 &\dots \\
 a_n &: [(n-1)d, 1]
 \end{aligned} \tag{2.1}$$


where d it the size of the partition corresponding to each agent.

On the other hand, this work has been handled corresponding intervals to each child of a node, as presented in Figure 2.5, that represents the maze in Figure 2.3. Thus, the goal of this work aims to establish an efficient algorithm that interrelates the agent intervals to the graph intervals. Furthermore, a mixed radix representation to the agent path has been studied. 

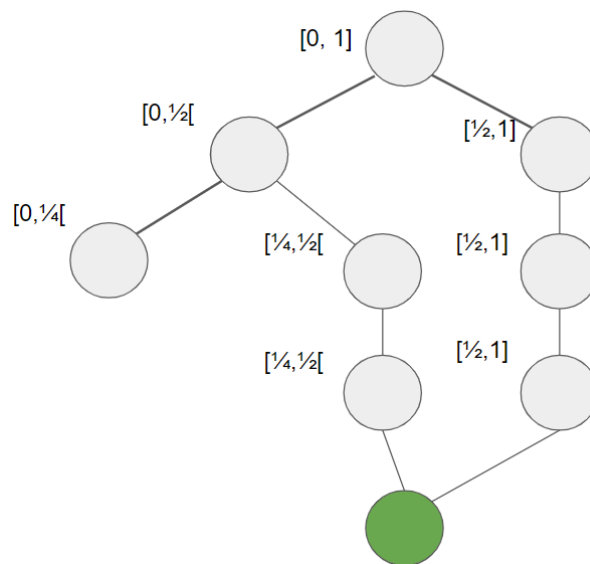


FIGURE 2.5 – Graph representation of the maze presented in Figure 2.3, with intervals related to the agent's range of action.

3 Results

A code has been developed to compare the proposed algorithm with other maze-solving algorithms. The code was delivered to the advisor and co-advisor by email.

4 Next steps

Next steps has been discussed with the advisor and the co-advisor, and the main points are:

- to compare the proposed algorithm with well-known algorithms in the context of maze-solving challenges;
- to generalize the proposed algorithm to search algorithms in graphs;
- to establish a mixed radix representation to the agent path.

Bibliography

Arndt, J. (2011). **Mixed radix numbers**. In: Matters Computational. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14764-7_9

Beisel, B. W. (2014). **Distributed Maze Solving By Cooperative Robotic Platforms**. University of Maryland, Department of Electrical and Computer Engineering.

Burgard, W., Moors, M., Stachniss, C., Schneider, F. E. (2005). **Coordinated multi-robot exploration**. IEEE Transactions on Robotics, 21(3), 376-386. doi:10.1109/tro.2004.839232

Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. and Stein, Clifford. **Introduction to Algorithms**. 4th : The MIT Press, 2022.

Kivelevitch, Elad & Cohen, Kelly. (2010). **Multi-Agent Maze Exploration**. Journal of Aerospace Computing Information and Communication. 7. 391-405. 10.2514/1.46304.

Matarić, M. J. (1995). **Designing and Understanding Adaptive Group Behavior**. Adaptive Behavior, 4(1), 51-80. doi:10.1177/105971239500400104

Muhammad Ahsan Naeem. **pyamaze**. GitHub, 2021. Available at: <<https://www.github.com/MAN1986/pyamaze>>. Accessed on: June 6, 2023.

Sadik, A. M. J., Dhali, M. A., Farid, H. M. A. B., Rashid, T. U., Syeed, A. (2010). **A Comprehensive and Comparative Study of Maze-Solving Techniques by Implementing Graph Theory**. 2010 International Conference on Artificial Intelligence and Computational Intelligence. doi:10.1109/aici.2010.18.

Shields, R. (2012). **Cultural Topology: The Seven Bridges of Konigsburg, 1736**. Theory, Culture & Society, 29(4-5), 43-57. doi:10.1177/0263276412451161.

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO TC	2. DATA 06 de junho de 2023	3. DOCUMENTO Nº DCTA/ITA/DM-000/2023	4. Nº DE PÁGINAS 26
5. TÍTULO E SUBTÍTULO: Multi-agent graph exploration without communication			
6. AUTOR(ES): Arthur José de Sousa Rodrigues			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA			
8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: Maze; Graph; Search; Multi-agent			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Maze; Graph; Search; Multi-agent			
10. APRESENTAÇÃO: (X) Nacional () Internacional ITA, São José dos Campos. Curso de Engenharia de Computação. Orientador: Prof. Dr. Luiz Gustavo Bizarro Mirisola. Coorientador: Prof. Dr. Vitor Venceslau Curtis. Defesa em 00/00/2023. Publicada em 00/00/2023.			
11. RESUMO: <p>Algoritmos computacionais de busca têm sido estudados por cientistas e companhias de engenharia desde o último século devido às aplicações de tais algoritmos em situações reais, como agendamento de linhas aéreas, planejamento de rota em mapas, algoritmos de busca na Internet, planejamento de rotas em redes de computadores, robótica, etc. Usualmente, esses métodos se baseiam em tipos abstratos de dados, como grafos e árvores, para transferir um problema real para dentro de um contexto delimitado e mais compreensivo. No campo da Ciência da Computação, grafos são tipos abstratos de dados que podem servir como suporte ferramental para algoritmos de busca.</p> <p>De uma abordagem relacionada a algoritmos de busca em labirintos, este trabalho de graduação propõe um método para explorar grafos em um âmbito multiagente, em que os agentes não são capazes de se comunicarem entre eles, o que faz referência a problemas da realidade, como exploração no fundo do mar, busca em estruturas compostas por muros maciços, busca em locais inóspitos por agentes com restrição energética, etc.</p>			
12. GRAU DE SIGILO: (X) OSTENSIVO () RESERVADO () SECRETO			