Title	パズルの解法アルゴリズムと情報推移に伴うエンターテインメントの解析
Author(s)	劉, 暢
Citation	
Issue Date	2022-09
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/18136
Rights	
Description	Supervisor:飯田 弘之, 先端科学技術研究科, 博士

Solving Puzzles and Its Entertaining Analysis in Informational Progresses

by

Liu Chang

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Supervisor: Hiroyuki Iida

School of Information Science
Japan Advanced Institute of Science and Technology

September,2022

Abstract

As a human activity with an ancient history, games not only serve for fun but also promote the development of entertainment technology. The vast majority of puzzle games are known as single-agent games in the area of mental exercises, such as N-puzzle and Rubik's Cube. Solving puzzles helps to understand better information variation and stochastic characteristics in the solving progress. Puzzles can be divided into two categories: puzzles without hidden information and puzzles with hidden information, representing the certainty and uncertainty factors in the puzzle-solving process, respectively.Uncertainty in a puzzle affects the way players experience entertainment and affects the solvability of the puzzle. In general, the purpose of a puzzle is to allow the player to explore for the optimal solution. Recent related work on applying search algorithms to the puzzle domain can be divided into two directions. The first direction is using puzzles as experimental platforms to verify the performance of the algorithm, while the other is using the search algorithm to find the optimal solution to the puzzle. Few studies have focused on hidden information in puzzles on solvability. However, observing the information in the puzzle-solving process may lead to a link between the puzzle and the game.

In this thesis, the A* algorithm was used to solve N-puzzle and dynamic information in the solving progress. It explores information by solving puzzles in an optimal way and entertaining analysis way. In addition, a solving strategy based on Gauss–Jordan Elimination and Constraint Satisfaction Problem was proposed to solve Minesweeper to explore the link between puzzles and games. The winning rate based on this strategy provides a new perspective on the definition of puzzles.

This thesis focuses on solving puzzles, and its entertaining analysis of information progresses. (1) Solving an 8-puzzle with the randomly generated initial position using the A* algorithm as the AI player. By adopting total steps to solve the game and the success rate as the game progress model, the attractiveness and sophistication of this puzzle have been discussed. Such findings could contribute to the evolutionary changes in

sliding puzzle games. (2) To develop an AI solver of Minesweeper with the configuration of $9\times9|10$, $16\times16|40$, and $16\times30|99$, based on the obtained information on the board, called the 'PAFG' strategy, which stands for the primary reasoning, the advanced reasoning, the first action strategy, and the guessing strategy. The first two strategies take advantage of knowledge-based rules and linear system transformation (Gauss-Jordan elimination algorithms) to determine the probability of making a move independently. The last two strategies explore the beginning and ways to determine hidden puzzle states to enhance the winning rate of the AI solver. Such an AI solver could contribute to classifying single-agent stochastic puzzles and establishing the boundary of the puzzle-solving and game-playing paradigm. (3) To explore puzzle categories based on the perspective of Minesweeper solvability and find the border between puzzles and games, as well as study the motion in mind to the entertainment analysis of solving puzzle field. Moreover, to discover significant characteristics from the perspective of information dynamics in the solving process and reveal the internal laws behind players' behavior. The experiment demonstrates the link between solving puzzles and playing games from an entertaining analysis view. Even more, it has become indispensable in the field of puzzle-solving and entertainment analysis and influences the way of puzzle-solving and the solving experience.
Keyword: *Puzzle Solver*; *Information Dynamic*; *A* algorithm*; *Gauss–Jordan Elimination*; *Constraint Satisfaction Problem*

Acknowledgments

First and foremost, I would like to say thank you to my motherland, the People's Republic of China. For only her strong support and backup that I have the opportunity and courage to become who I am today.

I would like to dedicate my most sincere and deepest gratitude to my supervisor, Professor Hiroyuki Iida. I would like to thank him for giving the opportunity to continue my study and for giving me the experience of life at JAIST. His serious and meticulous approach to research, his guidance and encouragement to me led me to become a better researcher. His generosity, kindness, and wisdom have led me to want to be a better person. The sharing from him in the weekly Newsletters has always led us to better enjoy our studies and life at JAIST, and I believe it will always encourage me to embrace a better life in the future.

Great thanks to my second supervisor, Professor Kokolo Ikeda. He has always given me valuable advice and suggestions on my research. His guidance and help have made my research better and more concrete, and I have benefited a lot from his serious and rigorous research attitude. Thanks to my minor research supervisors, Professor Kaneko Mineo and Dr. Saajid Abuluaih. With their support, I was able to complete on my minor research. The encouragement and support from Professor Kaneko Mineo allowed me to learn many skills that will benefit me for life. The rich experience and always enthusiastic sharing from Dr. Saajid Abuluaih made me realize the wonderful world and want to help more people within my ability in the future.

Sincere thanks and appreciate go to Dr. Mohd Nor Akmal Khalid. Not only for his guidance and help in my research work and publications, but also for always giving me guidance when I was worried about my future and troubles in my life. Like a patient and friendly big brother of mine, he gave me more confidence to enjoy my study life at JAIST. Thanks to the committee members of my Ph.D defense: Professor Kokolo Ikeda, Professor Kiyoaki Shirai, Professor Shun-Chin Hsu and Professor Tsuyoshi Hashimoto.

Next, special thanks to Professor. Jijun Zhao, Ms. Hongyu Zhao and Ms. Yuexian Gao. It is their encouragement and trust that gave me the opportunity and courage to continue my studies abroad. Thanks to Mrs. Asakura for filling my life in our lab with warmth and laughter like a friend. Thank you to Mr. Huang Shunqi for his assistance in developing the AI program. Thanks to all the students in the Iida lab that making my student life colorful. I appreciate Ms. Xiaohan Kang, Ms. Naying Gao, Ms. Xinyue Wang, and Mr. Yuanting Mu for helping me a lot with my life and study when I first came to JAIST. I would like to thank Ms.Tingting Wu and Mr. Yicong Wu for a richer academic life experience. Thanks to Papa, Nazhif and all members in Iida lab for the friendship and teamwork.

I will always be grateful to Jianjian, Miao, Lily, Mao, and Tingzi for always supporting and being by my side. They have accompanied me through the happiest and most toughest times in the past few years. The time we laughed, went to the beach, had a barbecue, went for a bike ride and attended a wedding together were the most unforgettable and precious memories of my Ph.D life.

Gratitude to my beloved family: my parents and my brothers, they gave me warm love and strong backing, so that I always have courage to try. A big thanks and the best wishes to my best friends: Huan, Chun, Cheng. They always encouraged and comforted me, which gave me more courage to face my future life.

Furthermore, I would like to thank JAIST and the Japan government for their support, especially during the particularly difficult COVID-19 times. Thanks to my Japanese teachers: Ms. Tsutsui and Ms. Kubo for providing me with a wonderful experience of learning Japanese, and leading me to experience a more wonderful life. I love the mountains, trees, streams, sky and clouds on this side of JAIST.

Finally, I would like to thank all those who have supported me directly or indirectly to complete my studies. Thank you again to all of you mentioned or not mentioned above, without you, I would not be who I am today. Thank you for being there for imperfect me, and I will strive to become a better me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Chapter Introduction

In this chapter, we first introduce the history of solving algorithms and strategies using the puzzle as test-beds and how it grows with the development of the puzzle. As the major content of this thesis consists in analyzing the generated solvers and analyzing its results to the entertaining analysis the information progresses based on the solving information. Then, we overview the solving method of puzzles and its impact on entertaining analysis. Finally, we outline the contents and summarize the contributions of this thesis.

1.2 Background

Games are an indispensable human activity for fun. Researchers have tried to defined game as a fundamental part of human existence to analyze the art of game design, and explore the reason why do people play games. [2] provided the relationship between games and culture, Adrienne defined the video game culture which permeate areas such as education, social business activities, mobile technology, and family interactions, and people who play these games do not differentiate between age, gender, sexual orientation, race, religion and nationality. Video games use electronic devices as a medium for various gaming experiences, which could provide entertainment for a broad population. A study [3] proposed an iterative method to solve two-person game, which has been proofed in mathematical way and has laid the promising theoretical foundation for the future of solving games.

However, in the 21st century, the electronic game industry that targets "casual" game players has emerged [4]. Puzzle game as a classical casual game is well known to help logic-based brain stimulation that enjoys relaxation and entertainment during the game-playing process. Unlike other games that aim to make the gameplay longer, such as flappy bird, a puzzle game focuses on finding the optimal solution under specific rules [5]. In some cases, the game and the puzzle can be interchangeable. Games can be composed of small puzzles, and once the puzzles are solved, there is no uncertainty in game. Moreover, researchers have studied the puzzle as a single decision maker and single-player game in decades [6] [7], they defined a solution of the puzzle should be aesthetically pleasing and gives the user satisfaction, where 24 puzzles were used as test-beds to explore the complexity and methods to solve.

In general, solving a puzzle means finding a solution to a puzzle. Researchers have explored many methods and algorithms for solving kinds of puzzles [8] [9]. A study proposed the first application of a metaheuristic technique for solving the popular sudoku puzzle. Notably, they pointed out that the relationship between logic-based and random search algorithms is actually likely to be symbiotic; that is, they are likely to be able to benefit from each other's strengths: random search algorithms can help logic-based algorithms solve a wider range of problem instances, while logic-based algorithms have the potential to significantly reduce/adjust the search space to reduce the work of random search algorithms. And, a hybrid algorithm for sudoku might be even further promise [10]. Similar to Sudoku, Nonogram is called logic puzzle, Japanese puzzle, number weave, puzzle puzzle, number cross puzzle. In addition, an algorithm based on shape and image merging was proposed to solve jigsaw puzzles by using images containing dozens of piece [11].

Moreover, a puzzle often has more than one solution, advanced players seek the optimal solution, which takes the least number of steps or has the highest chance of solving. The optimal solution consists of a sequence of steps, the optimal solution at each step determines the final optimal solution, which is related to the solving rate in the solving process. However, there is still no study explored well aboout the puzzle-solving process in mind from the view of information dynamics during the solving process.

The motion in mind concept uncovers the motion changes of game's and players' information for various game uncertainty over time has been discussed by Iida et al [1]. As quoted by Albert Einstein, "Most people see what is, and never see what can be." In a sense, a similar notion is induced in this thesis from the context of the puzzle. By understanding the information progress in puzzle-solving, we can have a better comprehension of human life, including the interpretation of entertainment, training, and thinking. This thesis investigates the dynamic uncertainty of problem-solving progression in different puzzles and the interplays that a player experienced between solving puzzles and playing games. Exploring the boundaries of games and puzzles while establishing the differences and connections between puzzles and games is also a promising research direction that warrants further investigation. Nevertheless, further exploration in contemporary data collection technology may provide fertile grounds for further justification of the thesis's findings.

Puzzle is an ancient single-agent game with the purpose of finding the optimal solution to solve problems under specific rules [4]. With the artificial intelligence (AI) developments, researchers focus on solving puzzles automatically by various algorithms and strategies. Each puzzle has a unique characteristic that involves solving information delivery and provides a solving experience for players. The relationship between motions for the player and the game, can reasonably correspond to the relationship between physics for humans and nature, which has been verified to some extent in the fields of sports games and board games through motion in mind measure. Combined with game refinement theory, both are promising in studying the uncertainty characteristics of games in the field of entertaining analysis. This thesis focusing on exploring the deterministic and stochastic elements of puzzle mechanics based on developing AI solvers of target puzzles, as well as its impact of information dynamics during the puzzle-solving process on engagement and entertainment analyses by using the game refinement theory and motion in mind concept. Moreover, finding the border between puzzles and games from the solvability point of view, as well as the winning rate various on the puzzle configurations.

1.3 Statement of Research Question

The main purpose of this thesis is to develop puzzle AI solvers to explore the internal mechanism of puzzle based on the information during the solving progress, as well as to do entertaining analysis together with game refinement theory and the motion in mind concept. To this end, the main contributions of this thesis are as follows: (1) To develop some AI solvers for solving some target puzzles (N-puzzle, Minesweeper, etc.). (2) To establish puzzle categories from the perspective of their solvability and find the border between puzzles and games. (3) To explore the entertainment analysis of solving puzzle filed with the game refinement theory and motion in mind measure, discover the characteristics of each puzzle from the perspective of information dynamic in solving process, and reveal the internal laws behind player's behavior.

This thesis focuses on dynamic information in puzzle-solving process as well as its impact on engagement and entertainment analysis of puzzle development. The design concept of AI solvers in this thesis is based on the solving logic of players, which is modeled mathematically to facilitate future entertainment analysis of players with different abilities. Furthermore, more strategies correspond to the increased player's ability, which implies a higher winning rate, verified by proposed knowledge-driven reasoning, strategies, and simulation conducted in this thesis. Promising future work includes exploring the current strategies in other related single-agent games, to expand further and verify the definition established.

1.4 Structure of the Thesis

This thesis firstly introduced the background of the research field and stated the target research questions. Then, related works for the evolutionary of puzzles and the complexity of solving puzzles as well as measure of engagement and entertainment in Chapter 2. An ancient single-agent puzzle N-puzzle as the test-bed to study the entertainment analysis based on various puzzle configurations in Chapter 3. Besides, this research proposes a strategy to solve single-agent stochastic puzzle by takes Minesweeper as a test-bed, where the comparison with different methods are in Chapter 4. This thesis digs into the single-agent puzzles in 5, which shows the informational progression of solving puzzle process from the searching and solvability points of view, the border between games and puzzles has been defined as well. In general, this thesis tries to explore the solving difficulty in deterministic and stochastic puzzles. It indicates the significant features relevant to figure out the engagement and entertainment in our life.

This thesis comprises of 6 main chapters:

- **Chapter 1: Introduction**

  The objective of this chapter is to introduce the frame of this research, such as a brief evolutionary development in this domain, the outline of this research, the definitions, as well as the relationship among keywords in the research considered. The introduction chapter also includes a statement of the research problem, it explains the main problem that the research aims to solve, as well as the objective and significance of this thesis. At the end of this chapter, the structure of the thesis will be explained.

- **Chapter 2: Related Works**

  Literature Review: The chapter serves as a review of the theoretical background related to this research as well as presenting the state-of-the-art research in the field. The content of this chapter will be the background of puzzle evolutionary, the complexity of solving puzzles, a review of the solving algorithms, as well as the motion in mind, a measure of entertainment in the game domain.

- **Chapter 3: Solving Single-Agent Deterministic Puzzle**

  As a single-agent game, the sliding puzzle game had been rated as one of the Chinese video programs that exceeded 10 million downloads in 2018. A sliding puzzle is a combination puzzle that challenges a player to slide (frequently flat) pieces along specific routes (usually on a board) to establish a specific end configuration. It examines the player's comprehensive ability to balance a whole range of dimensions, including reasoning, computing, observation, memory, space, and creative ability. As early as the 1970s, Doran collected the data through experiments for solving 8-puzzle games using three heuristics algorithms and a weighted parameter. Since then, many efforts have been directed towards research in sliding puzzle games, focusing on finding the optimal solution or using a sliding puzzle game as an experimental platform to verify the time and space complexity of algorithms [12]. However, these studies are concerned with finding the optimal solution or improving the solving algorithms. Limited studies have been conducted on evaluating the attractiveness of an 8-puzzle for players. Therefore, the A* algorithm is adopted in this chapter as the AI player to randomly generate the 8-puzzle solution and explore the reason why the game is challenging but accessible among some people.

- **Chapter 4: Informational Dynamic of Single-Agent Stochastic Puzzle**
  Proposing an efficient puzzle solver has been a significant research paradigm in Artificial Intelligence (AI). Efficient solving mechanisms and algorithms had been exclusively conducted on the domain of two-player zero-sum games since 1994 [13]. Based on such domain, various methodologies have been proposed which take advantage of the deterministic nature of such games, which is associated with predicting a game-theoretic value and determining optimal strategy. Nevertheless, a clear definition of puzzle categories is needed from the perspective of its solvability. For such purpose, the Minesweeper puzzle was utilized as the benchmarking testbed. In such a condition, what distinguishes a deterministic puzzle from other types of puzzles, i.e., a stochastic puzzle? Therefore, this chapter proposed a definition of a stochastic puzzle from its solvability, which formed the foundation for the proposed AI solver. Moreover, the proposed AI solver takes advantage of both the deterministic and stochastic elements of the puzzle, such strategies also combine mathematical models, knowledge-driven rules, and linear transformation to provide conducive moves in solving the Minesweeper, comparable to previously proposed Minesweeper AI solvers.

- **Chapter 5: Finding the Border Between Games and Puzzles**
  Nowadays, researchers focus on studying and developing algorithms to analyze the relationship between branching factors and difficulty in puzzle games. However, the play experience would differ depending on the players' ability, experience, to solve the puzzle which causes some variant in their decision-making ability. Then, how can this ability affect players' decision-making during the puzzle game process? This chapter aims to analyze how the reduction of branching factor affect the relation

between game process and player experience from objective and subjective perspectives among different puzzles. Also, the motion in mind concept is adopted to find the border between playing games and solving puzzles.

- **Chapter 6: Conclusion**

  Chapter 6 gives the general conclusion of this thesis and illustrates the significance findings both in theoretical and practical sense. The deficiency of the current thesis and promising future direction are given at the end of the chapter.

Chapter 2

Literature Review

2.1 Chapter Introduction

In this chapter, we introduce the theoretical background related work to this research as well as illustrate the uncertainty measurement in the field of entertainment of solving puzzles field. The first section serves to review the history of puzzles evolution and the relationship between the category of puzzles and their characteristics. Studies of the complexity of solving puzzles from ancient times are covered in the second section. Different scholars may have different views on the same problem, and their research is also listed in this chapter. Then, the measurement of the uncertainty analysis and motions in mind have been presented in this chapter, as well as the state-of-the-art research in the field.

2.2 Evolution of Puzzles

Research on the effects and benefits of playing games has been going on for decades. Vygotsky argued for a strong theoretical link between games and factors that promote social cognition, where the immediate and concrete feedback, such as dead ends in puzzles serve to the reward continual effort and keep players or solvers within the proximal development zone [14]. Researcher Granic said the core of all kinds of video games is problem solving. As an ancient video games, puzzle game designers often offer little guidance on how to solve problems, and instead provide a near-blank board for players to explore a large number of possible solutions based on their own experience and intuition [15]. In addition, a research has shown that play puzzle games with characteristic of minimal, short-term, and highly accessible, can helps to feel better, promote relaxation, and combat feelings of anxiety, like Angry Birds, Bejeweled II [16].

The Tangram is a puzzle consisting of seven put-together pieces. It was invented in China between 1796-1802, brought to England through the "China Trade", and soon became popular in Europe and America. History has labeled it the world's first puzzle craze in the book [17]. Several kinds of arrangement puzzles had been invented before the advent of the jigsaw puzzle, and many after that, but the jigsaw puzzle is by far the most popular one. These seven pieces are all simple shapes: two small triangles, a medium-sized triangle, two large triangles, a diamond, and a square. It is unique in its ability to turn these simple geometric shapes into fascinating, elegant, complex, and sometimes contradictory figures. Or you can create your own designs, limited only by your imagination. This simple puzzle is most gripping: How do seven simple tans create such extraordinary images and bewildering challenges?

Moreover, most puzzles we have ever played belonged to NP-complete puzzles [7]. Researcher Kendall analyzed 24 puzzles in terms of the presence or absence of behavioral order and hidden information in order to promote the research attention in the field of puzzles, and gave references to most of the puzzles. Minesweeper, Mastermind, Solitaire, and Tetris are thought of as a puzzle with hidden information. While Blocks World, Clickomania, Lemmings, Mastermind, Minesweeper, N-Puzzle, Peg Solitaire, Rush Hour, Solitaire, and Tetris are considered as a puzzle with the action order matters. At the end of the study, many promising research directions based on puzzle research are given, such as are the puzzles stochastic or deterministic? How large is the search space? Are the puzzles single-goal or multi-goal? A lot of important inspirations are given to further researchers. From this perspective, the Minesweeper has hidden information, while N-puzzle does not, giving us some inspiration to distinguish between puzzle classifications.

"HuaRongDao" is a traditional sliding puzzle game in China, based on the "Romance of the Three Kingdoms", and is known as one of the "three incredible games of ancient

China" along with the Tangram and the Nine Linked Rings for its richness of puzzle and fun; also known as "the three incredible games in the world of intellectual games" by foreign intellectual experts together with Rubik's Cube and Independent Diamond Chess for its many variations and never-ending play. N-puzzle is a kind of sliding puzzle which the Fifteen Puzzle has been popular for more than 100 years, originated in December 1879 when Mr.Rice placed 15 wooden blocks on a cardboard box and surrounded the pieces with a wooden frame called Gem. It took only a few months for it to become popular in February 1880. The rule is to place the blocks in the box irregularly, then move the blocks until the order is regular. These simple and somewhat ambiguous instructions were a key factor in the development of the Fifteen Puzzle craze [18]. No matter old or young, black or white, they tried to solve the Fifteen Puzzle all day and night. Later, it evolved into 8-puzzle and 24-puzzle, which became popular around the world as a different difficulty division.

Throughout human history, puzzle games have their origins in the advent of brain teasers and puzzles. Many educational games were created in the early days of handheld gaming, where they created a template for games that required only thought and strategy without any action or adventure. Erno Rubik created Rubik's Cube in the late 1970s. The $3 \times 3 \times 3$ Rubik's cube is a classic three-Dimensional combination puzzle with 3×3 planes in 6 faces that can be rotated 90 degrees in each direction, with an ample state space of about 4.3×10^{19} possible configurations and only a single goal state [19]. It can be solved when all planes on each face of the cube are the same color. The game length of the optimal solution and nodes generated for random Rubik's Cube problem has been discussed, which has successfully increased the cost-effectiveness of search algorithms. Tetris is considered to be the game's revolution and popularization of the puzzle game genre. Tetris is a tile-matching video game created in 1984 by Russian software engineer Alexey Pajitnov for the Electronika 60 computer [20]. Researchers said that Tetris, one of the ancestors of the matching tiles game, combined with the chain shot (1985) have contributed to the iterative development of Match-3 puzzles, with the representative nature of the popular trend being Bejeweled faced to the audience of "casual" gamers in 2001 [21].

In addition, with Minesweeper became a feature of Microsoft Windows in 1989, the hidden information puzzle gained it its popularity [22,23]. With more versions, players can design the size of the board and the density of mines independently. It is worth mentioning that the rules of the first step of minesweeper are extremely important, because it is related to the winning rate of Minesweeper. i.e. whether the cell opened in the first step is likely to be a mine or not, and how many of cells are opened in the first step. And in general, the rules of the first step are determined by the puzzle designer.

2.3 Complexity of Solving Puzzles

From the time the puzzle appeared, people were trying to solve it. As more and more attention was paid to the field of scientific research, researchers started to analyze the type and complexity of the puzzle and tried to solve it in different ways until they found the optimal solution to the puzzle. Indeed, an article of Stephen Cook [24] begins succinctly by showing the problems faced when considering computational complexity: "In general, it is much harder to find a solution to a problem than to recognize one when it is presented." This idea can be applied to most of the current popular puzzles. Three possible correlations between P, NP, NPC, and co-NP are shown in Figure 2.1.

Polynomial-time (P) problems are those for which a method exists to solve them in polynomial time. In the computational complexity theory, Polynomial time means that the calculation time of a problem is not greater than a Polynomial multiple of the problem size. Any abstract machine has a complexity class that includes problems that can be solved by the machine in polynomial time. An optimization problem is called polynomial-time solvable if a polynomial-time algorithm has been found, and the set of such problems is denoted as P. Therefore, polynomial-time solvable problems are called P problems.

Non-deterministic polynomial time (NP) problems exist when an algorithm validates the response "YES" in polynomial time for problem instances. Non-deterministic algorithms decompose the problem into two phases: guessing and verifying. The guessing phase of the algorithm is non-deterministic, and the verification phase of the algorithm is deterministic, which verifies the correctness of the solution given in the guessing phase. Let an algorithm M be a non-deterministic algorithm for solving a decision problem Q. If the verification phase of M can be completed in polynomial time, then M is said to be a polynomial-time non-deterministic algorithm. Some computational problems are deter-

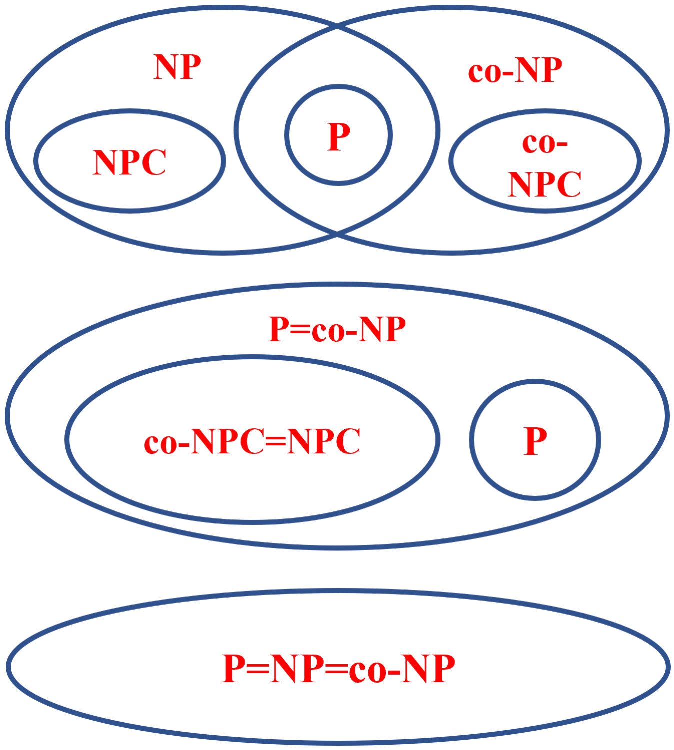


Figure 2.1: An illustration of three possible relationships among P, NP, NPC, and co-NP

ministic, such as those for which the result can be obtained by following the derivation of the formula. However, some problems are not computable in a straightforward way. For example, the problem of finding large prime numbers. Is there a formula to find out what the next prime number is? The answer to such a question cannot be calculated directly, but can only be obtained by indirect "guessing". This is a non-deterministic problem. These problems usually have an algorithm that does not tell the answer directly, but can tell whether a possible result is correct or incorrect. This algorithm, which can show whether the "guessed" answer is correct or not, is called polynomial non-deterministic time problem if it can be computed in polynomial time.

A problem p is an NP-complete problem if and only if p is an NP problem and every NP problem can be reducible to p in polynomial time [25], which means that NPC problems are the most challenging problems in NP problems. The complexity of certain problems in NP is related to the complexity of the entire class. If there is a polynomial time algorithm for any of these problems, then all NP problems are polynomial time solvable. These problems are called NP-complete problems. Nowadays, in the pure scientific research, communication, transportation, industrial design and enterprise management departments, in the social military, political and commercial struggle emerged a large number of NP problems. If it is solved by the exhaustive method familiar to classical pure mathematicians, the calculation time often reaches astronomical figures and has no practical value at all. Many experienced mathematicians believe that there is no complete, accurate, but not too slow solution to these problems. NP = P? Probably the most important mathematical problem of the century.

Moreover, co-NP problems are complements of NP problems. In other words, co-NP problems are those for which a polynomial-time solution exists to validate the answer "NO" for problem instances. A problem p is a co-NP-complete problem if and only if it is a co-NP problem, and every co-NP problem can be reduced to p in polynomial time, implying that co-NP-complete problems are the most challenging co-NP problems.

Many research experiences have been put into studying the complexity of puzzle-solving field, and lots of the puzzles that have created a popular craze have proven to be NP problems. However, there are many classifications of the characteristics of these puzzles, such as the hidden information, and the order of behavior is important. This thesis focuses on focuses on analyzing the mechanism and characteristics of solving puzzles from the perspective of uncertainty (hidden information) in puzzles, with N-puzzle as the representative deterministic puzzle, and Minesweeper as the representative uncertainty puzzle, as well as the significance of their information dynamics in the solving process in the entertainment field.

2.3.1 N-puzzle

N-puzzle first appeared in the scientific literature in 1879 [7]. The N-puzzle game consists of n numbered, movable tiles set in a $m \times m$ frame, where $m \times m - 1 = n$, each of which has a unique number from $1...n$. The goal of the puzzle is to disarrange the initial state cells out of order, then through legal movement, make all cells discharge in the order of the goal state. It is interesting to note that not all N-puzzles are solvable. The states of N-puzzle can be divided into two states, half of which cannot be legally moved to reach the target state, while the other half can. This research also showed that the generalization to the N-puzzle is NP-complete problem, but the 15-puzzle is not.

The A* algorithm is well-known heuristic algorithm for solving N-puzzle, especially for 8-puzzle, which has a relatively small search space. The IDA* algorithm as a linear space version of A* algorithm, was the first to obtain optimal solutions of 15-puzzle. Korf and Schultze provide an improvement on the best first search, enabling them to complete the first breadth first search of 15-puzzle [26]. Analysis and experiments in a study based on IDA* algorithm for solving Rubik's cube and N-puzzle show that the asymptotic heuristic branching factor is the same as the brute force branching factor on the exponential tree. This thesis shows that the effect of heuristics is to reduce the effective search depth by a constant, rather than reducing the effective branching factor, as opposed to brute force search. This result lays a theoretical and practical foundation for the algorithm and complexity of N-puzzle [12].

As the generation of 24-puzzle, the depth of the search tree increases as the number of cells in the puzzle increases. An survey of puzzle games provided many of the methods used by researchers trying to solve 24-puzzle [7]. Among them, researchers tried to use IDA* algorithm for experiments to find the optimal solution, some tried to propose an algorithm to find the sub-optimal solution in polynomial time, and some researchers provided a technology to prune repeated nodes from the search. Besides, researchers proposed a method based on IDA* algorithm and pattern database heuristics to find the optimal solution of 24-puzzle. However, the purpose of this thesis on deterministic puzzles is to find the optimal solution with the minimum number of steps. The IDA* algorithm may not get the optimal solution because of insufficient iteration depth, while the A* algorithm always gets the optimal solution if it takes long enough.

2.3.2 Minesweeper

Minesweeper is a popular computer puzzle game that comes with some Microsoft Microsoft Windows operating systems. It is played on an $R \times L \mid b$, all of which are initially hidden. $R \times L \mid b$, where R and L are the number of rows and columns on the board, respectively, and b is the number of mines. From the computational complexity perspective, previous studies found that Minesweeper is too hard to solve in polynomial time. Moreover, as one of the NP-complete problems, Minesweeper is the most difficult NP problem as found by [27]. From Minesweeper consistency problem, given a game of Minesweeper configuration and a board state with some visible numbers and flagged mines, does there exist a possible distribution of unknown mines to satisfy known information? [25] proposed that the Boolean satisfiability problem (SAT) could reduce to Minesweeper consistency problem, which was described as Boolean formula (i.e., all variables take TRUE or FALSE assignments).

Furthermore, based on a theorem by [24] that Boolean satisfiability problem (SAT) is NP-complete, which proved that the Minesweeper consistency problem is NP-complete. Then, solving the Minesweeper consistency problem provides a way to Minesweeper inference problem. For instance, given a game of Minesweeper configuration and a board state, does a hidden cell exist that the player can infer undoubtedly? [25] and [28] studied Minesweeper inference problem from the perspective of computational complexity theory. They proved that the Minesweeper inference problem is a co-NP-complete problem (the complements of NP problem), which is the most challenging problem in co-NP and reduces the complement of SAT to the Minesweeper inference problem.

2.4 Game Refinement Theory

Game refinement (GR) theory has been studied the game outcome uncertainty, where game dynamics are evaluated based on an innovative view on the outcome uncertainty of the game simulated via the analogy of Newton's law of motion [29] [1]. It has been evaluated in the domain of game such as board games and sports games, also studied in non-game domain such as education and business. Game refinement theory fundamentally involves the measures that define the game sophistication that converges towards a common range, where the most stochastic game located in $GR \in [0.07, 0.08]$, corresponds to the lower bound (fairness) and upper bound (engagement), respectively. Later, it has been involved to measure the attractiveness of a game [30], where the lower bound and upper bound are corresponds to the game that more relies on skill and chance, respectively.

From game playing point of view, the information on reaching a game outcome for a player is regarded as a function of time t, and the information on the game results is regarded as the solved uncertainty (information) $x(t)$. In other words, the process of solving the uncertainty is an increasing function of time achieving such an outcome. Then, (2.1) is obtained to illustrate the velocity in game, where the parameter $n(1 \leq n \in \mathbb{N})$ is the number of possible options (branching factors), the parameter t is the game length, which is depends on players.

$$x'(t) = \frac{n}{t}x(t) \tag{2.1}$$

However, such a formulation implies that the game outcome is known. In reality, the game outcome is unknown until the game ends. As such, a realistic formulation considering the uncertainty of the game outcome is given by (2.2). Note that $0 \leq t \leq T$ and $0 \leq x(t) \leq 1$, and $x(0) = 0$ and $x(T) = 1$. Here, from the game objective point of view, the game length is assumed as T, and the game outcome is regarded as $x(T) = 1$.

$$x(t) = \left(\frac{t}{T}\right)^n \tag{2.2}$$

The velocity in a game process can be seen as Equation (2.1), based on the accelerated velocity in physics is used to describe changes in velocity, Equation (2.3) is given to illustrate the rate of change of the solved information $x(t)$ of the game progress, where the solved information of (2.2) is assumed to be twice derivable at $t \in [0, T]$. This implies that game is fascinating if this value increases or decreases, it will make the game even more fascinating and entertaining. thus, this character is considered to be the one that deserves the most attention in a well-refined game domain. This thesis is studied in determining the deterministic and stochastic characters in puzzle field, which is a popular single agent game domain.

$$x''(t) = \frac{n(n-1)}{T^n}t^{(n-2)} \mid_{t=T} = \frac{n(n-1)}{T^2} \tag{2.3}$$

Then, the GR measure is given as (2.4) in the root square of Equation (2.3), the accelerated velocity. This measure has been verified to reflect some aspects of the entertainment of games, such as attractiveness, engagement, and playing comfort. This thesis focus on doing entertaining analysis with the game refinement theory in the puzzle domain.

$$GR = \frac{\sqrt{n(n-1)}}{T} \tag{2.4}$$

2.4.1 Gamified experience for board games and sports games

Based on the study of game refinement theory, the ratio of solving uncertainty at different depths is given as v, and the solved uncertainty of the game $y(t)$ is an increasing function of time t, which can be given by (2.5). Let p be the probability of selecting the best choice among n number of options (branching factors). Hence, $p = \frac{1}{n}$ holds the moving velocity in a game. Based on such notation, the risk frequency ratio m (risk frequency over the whole game length) is defined as $m = 1 - p = 1 - v = \frac{n-1}{n}$. Then, gamified experience is gained only when the risk of failure occurs with $m \geq \frac{1}{2}$, which implies $n \geq 2$, and has been verified kinds of fun games.

$$y(t) = vt \tag{2.5}$$

The slope (v) with the time (t) of a game progress model in (2.5) and mass in the game playing have been determined in two domains: (1) scoring sports games, and (2) board games. For scoring sports game, let G and T be the total scores of goals and shoot attempts per game, respectively. Score rate v (the total scores of goals over the shoot attempts per game) is given by (2.6), where the slope v ($v = p$) of game progress model is equivalent with score rate in (2.5). Note that the score rate v in some sports (e.g., table tennis, badminton, soccer) is given by $v = \frac{1}{2}$, this situation is because one would have a point with the possibility of $\frac{1}{2}$ at each round.

$$v = \frac{G}{T} \quad \text{and} \quad m = 1 - v \tag{2.6}$$

For board games, let B and D be the average number of possible moves and game length. Score rate p is approximated as (2.7), by which p is equivalent with the slope v ($v = p$) of game progress model in (2.5). Note that the v in board games is approximated based on the number of plausible moves b, where $n \simeq \sqrt{B}$ is used in the best-case analysis of an efficient $\alpha\beta$ algorithm that is useful for pruning.

$$v \approx \frac{1}{2}\frac{B}{D} \quad \text{and} \quad m = 1 - v \tag{2.7}$$

2.5 Motion in Mind Measure

When players play games from the beginning to the end, the game progress can be treated as solving uncertainty. In other words, the game is full of uncertainty at the beginning, as the player play the game and the game process moves forward, the game's uncertainty information becomes less until zero at the end. Over time, the process of playing the game is one of decreasing uncertainty. In a puzzle game, the game information becomes certain when the player gets solutions to solve the puzzle.

Similar to physics in the world, vital physical quantities in mind are the velocity and mass, with the assumptions of $v + m = 1$ which are based on the zero-sum assumption, where gain or loss utility of one player is exactly balanced by the losses or gains of the utility of its opponent; thus deriving a reliable measurement of players' game experience, such as engagement and comfort [1]. Moreover, in puzzle games, different levels players may choose differently at each step based on skills, differing in velocity to move and solutions to solve the puzzle; portraying different solve experiences, such as attractiveness and engagement.

By analogically defining the game-winning (or success) rate and winning hardness (or difficulty) as the velocity (v) and mass (m), respectively, various motions in mind quantities can be determined [1]. Table 2.1 provides the analogical link of the related physics in mind notations and its in-game context (specific to the current study).

Table 2.1: Analogical link between physics and game (adopted from [1])

Notation	Physics context	Game context
y	Displacement	Solved uncertainty
t	Time	Progress or length
v	Velocity	Win rate (p)
M	Mass	Win hardness (m)
g	Acceleration (gravity)	Acceleration, a
F	Newtonian force	Force in mind
$\vec{p}$	Momentum	Momentum
U	potential energy	Potential energy, E_p

As the table shown, the displacement (y) in physic corresponds to the solved uncertainty in game context, and time (t) stands for the game progress or length in game domain. Force is determined as a product of mass and acceleration ($F = ma$), which relates to the acquiring engagement of player's movement ability in the game playing, from Newton's second law of motion. In classical physic, the gravitational potential energy U is given by (2.8) where g and h stand for gravitational acceleration and height (or displacement), respectively. Then, the potential energy (E_p) given by (2.9) can be obtained by the correspondence of $M = m$, $g = a$, and $h = y(t)$, where m and a stands for the win hardness and acceleration in game, respectively. A game's energy is defined as the amount of the required information (energy) needed by the playing in the game process, which is equivalent to the expectation of player to finish the game or the anticipation that the player expect the game give.

$$U = Mgh \tag{2.8}$$

$$E_p = ma\left(\frac{1}{2}at^2\right) = \frac{1}{2}ma^2t^2 = 2mv^2 \tag{2.9}$$

Meanwhile, the notion of momentum in game-playing process is given by (2.10), which defines the product of m and v, which is the moving difficulty (or hardness) and ability to move, respectively. This equation states that momentum (p_1) is directly proportional to the velocity of a game, and directly proportional to the mass of a game. In other words, such quantities describe the freedom magnitude of the player to use their ability to address the difficulty in games. Note that momentum in game playing is relied on two factors: the game progress ratio v and the hardness to move in a game m.

$$\vec{p_1} = mv \tag{2.10}$$

The game experience depends on the game itself (objective), but also on the player (subjective) such as skill, experience. Assumptions of both momentum and mass as the manifestation of energy lead to the discussion on the notion of potential energy (E_p) being conserved over time [31]. Then, such energy is transformed into the game's momentum ($\vec{p_1}$) and the mind's momentum ($\vec{p_2}$) of players, as given by (2.11). And, the $\vec{p_1}$ is considered the objective point of view, whereas the $\vec{p_2}$ is from the subjective point of view. The former is associated with the game's motion, while the latter is associated with the player's play experience [31], which is obtained based on equations (2.10), (2.11) and

(2.12).

$$E_p = \vec{p_1} + \vec{p_2} \tag{2.11}$$

$$\vec{p_2}(m) = E_p - p_1 = 2m^3 - 3m^2 + m \tag{2.12}$$

Then, (5.2) is obtained by the first derivative of (2.12). Solving $\vec{p_2} = 0$, then $m = \frac{3 \pm \sqrt{3}}{6}$ is obtained. It was conjectured that $m \simeq 0.79$ is the upper limit for competitive play mode, where $m \simeq 0.21$ is the lower limit for easy-win mode associated with the addictive zone (Figure 2.2). Each limit value corresponds to risk-taking engagement and profit-winning engagement, respectively. Interestingly, the cross point of $\vec{p_2} = E_p$ occurred when $m = 0.5$, which implies the moment where the game's motion is the greatest while the mind's motion is non-existence since E_p reflects energy conservation of objective and subjective motions. That means the game experience becomes fully stochastic, and predicting the game outcome becomes impossible.

Puzzles are a classic single-player game about problem solving. This thesis focuses on the player's experience of solving puzzles, and figure out the difference and connection to playing a game. As well as, the dynamics in energy (motion) based on deterministic and uncertain information during the solving process.

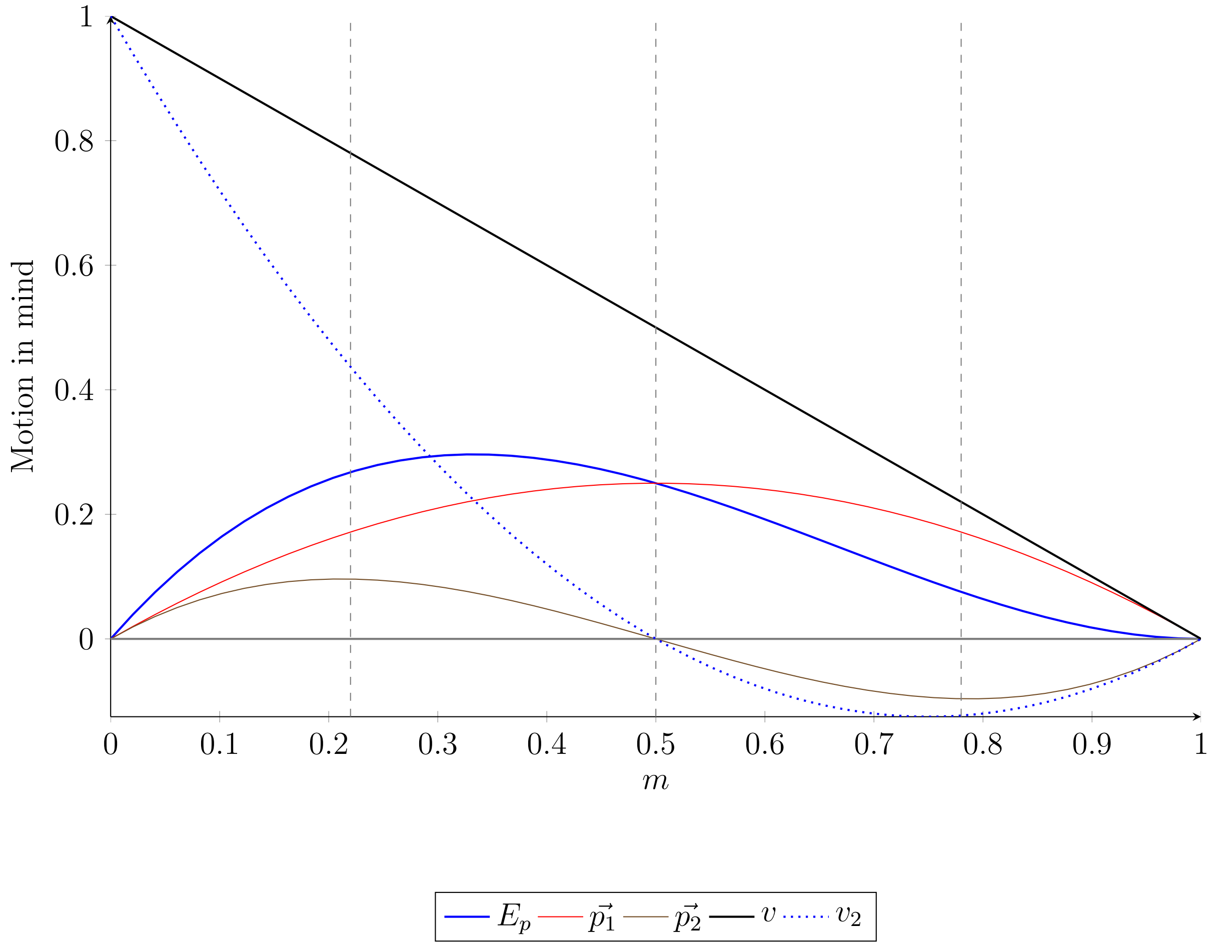


Figure 2.2: Illustration of law of motion in mind over various mass (m). The subjective motion(p_2) is derived from the objective ones(p_1), where subjective velocity (v_2) was established. $\vec{p_2}$ is derived based on the conservation of E_p.

2.6 Chapter Summary

In this chapter, related works to this thesis were illustrated. Works related to some ancient puzzles, namely the Tangram, HuaRongDao, Rubik's Cube, and Minesweeper are reviewed. Related work on problem classification, as well as enumerating complexity and solution analysis using minesweeper and N-puzzle games as experimental platforms. Moreover, in the field of entertaining analysis, game refinement theory which relies on the uncertainty in the process of game is introduced. Meanwhile, the measurement method of analyzing the change of motions in game-playing from the objective and subjective points of view of player is expounded, which serves as the base to the linking between solving puzzles and playing games.

Chapter 3

Entertaining Analysis of Solving Single-Agent Deterministic Puzzle

This chapter is based on an integrated, updated and condensed version of the following publication:

- Liu Chang, Huang Shunqi, Mohd Nor Akmal Khalid, Hiroyuki Iida, Attractiveness of Single-Agent Game: Case Study Using Sliding Puzzle. 2020 International Conference on Advanced Information Technologies (ICAIT). 2020. pp. 76-81.

3.1 Chapter Introduction

With the characters always having certain solutions and no hidden information, researchers have studied deterministic puzzles for decades. As early as the 1970s, Doran collected the data through experiments for solving 8-puzzle games using three heuristics algorithms and a weighted parameter [32]. Since then, many efforts have been directed toward research in sliding puzzle games, focusing on finding the optimal solution or using a sliding puzzle game as an experimental platform to verify the time and space complexity of algorithms [12]. A study [33] proposed that the game-tree search algorithm and the evaluation function are two core components of a two-player game playing program.

These are also important in single-agent games. And, Hyper-Heuristic concept is an promising way in designing puzzle-solving methods, as provided in [33], a hyper-heuristic is "a search method or learning mechanism for selecting or generating heuristics to solve computational search problems". Nevertheless, little work has been done to focus on the attractiveness of the game itself.

As a single-agent game, the N-puzzle game had been rated as one of the Chinese video programs that exceeded 10 million in 2018. A sliding puzzle, sliding block puzzle, or sliding tile puzzle is a combination puzzle that challenges a player to slide (frequently flat) pieces along specific routes (usually on a board) to establish a specific end configuration. It examines the player's comprehensive ability to balance a whole range of dimensions, including reasoning, computing, observation, memory, space, and creative ability.

Game refinement theory was first used to measure the attractiveness of multi-player incomplete-information games for pointing out the critical factor in calculating the number of options and game length [29]. In recent years, the game refinement theory has evaluated increasing attractiveness and the sophistication of game theoretical aspects, such as sports games and board games [34].

In recent studies, it has been shown that the game information in the brain is determined by neurons, which transmit information to other parts of the body through electrical signals, and the "strength" of such signals plays a role in understanding the entertainment aspects of the game [35]. Hence, it is interesting to know the physics measurements in mind of game playing. As indicated by many modern gaming principles for many years, the 8-puzzle game is considered a mixture of skill and chance. Although the puzzle game has been studied for decades, there is no publicly accessible database with real playing data of 8-puzzle to evaluate its entertaining characteristics. Recent findings based on motion in minds derived from the game progress model are utilized to quantify the attractiveness of an 8-puzzle game [1]. Hence, this thesis uses 8-puzzle as a deterministic testbed to explore entertaining analysis by developing AI solvers.

3.2 Game Testbed: 8-puzzle

The N-puzzle game consists of n numbered, movable tiles set in a $m \times m$ frame, where

$m \times m - 1 = n$, each of which has a unique number from $1...n$. This thesis using 8-puzzle as a testbed to explore the solving algorithm on various game length, as well as the entertaining analysis for AI and human players. The 8-puzzle game consists of eight numbered, movable tiles set in a 3×3 frame. One cell of the frame is always empty, thus making it possible to move an adjacent numbered tile into the empty cell. The purpose of this game is to move from the initial position to the target position with a minimum number of steps. An example of a randomly generated 8-puzzle with the initial and the final state is depicted in Figure 3.1.

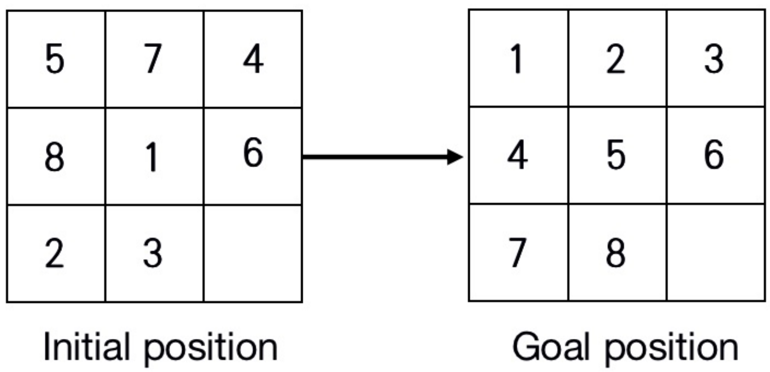


Figure 3.1: An example of 8-puzzle

The 8-puzzle was used as a prominent platform to apply the heuristic search algorithms in 1979 [36]. For several decades, researchers are focused on solving an 8-puzzle with heuristic search algorithms. In 1993, [37] used iterative-deepening search in the 8-puzzle to evaluate the benefit of the node where the algorithm performs better with a random operator selection scheme.

Some studies analyzed the pathology of the 8-puzzle, where several heuristic functions and various search tree properties were evaluated [38]. The authors concluded that dynamic search depth is necessary to determine the right positions for a more efficient in-depth search. Kevin adopted genetic programming to solve an 8-puzzle to solve 20 puzzle instances, which then compared with other traditional algorithms (A* algorithm, breadth-first search, and depth-first search) [39]. Although genetic programming could produce solutions for all 20 puzzle instances, the A* algorithm provides a better solution. More recently, Hamming, Manhattan, and Chebyshev heuristics were compared, where the space-time complexity of the A* algorithm has been improved with Chebyshev heuristics [40].

However, these studies are concerned about finding the optimal solution or improving the solving algorithms. Limited studies have been conducted on evaluating the attractiveness of an 8-puzzle for players. Also, the A* algorithm is adopted in this thesis as the AI player to randomly generate the 8-puzzle solution and explore the reason why the game is challenging but accessible among some people.

3.3 Experimental Setup and Its Results

This section divides the 8-puzzle-based experimental simulation into two parts: the first is the construction of the 8-puzzle AI solver using the A* algorithm, which is known for solving puzzles; the second is the data collection based on the human player's puzzle-solving. Both are performed entertaining analyses by using game refinement theory and motion in mind to discover the most sophisticated mechanics corresponding to players of different abilities. In addition, we also try to analogize 8-puzzle to board games and scoring sport games for entertaining analysis, providing promising directions for the field of puzzle mechanics analysis.

3.3.1 A* Algorithm

The A* algorithm is a direct search method for solving the shortest path in a static road network most efficiently and is an effective heuristic algorithm for solving many search problems. The closeness of the distance estimation value in its algorithm to the actual value determines the final search speed. A study compares the completeness and average path length of Breadth-First Search, Depth First Search, Best First Search, and A* Search. Experiments show that the average path length of algorithm A* is the shortest. It is similar to the Breadth-First search in terms of technique but may be slower because it is exhaustive. According to this characteristic, the single-agent AI system in this thesis for the 8-puzzle game simulation is implemented using the python compiler with the A* algorithm. It can be made faster by limiting the depth, where its procedure is given as in Algorithm 1. It shows the game's process while including essential parameters such as the number of nodes in the open list, the number of nodes in the close list, and game length.

For solving 8-puzzle, A* algorithm is an effective method to solve the shortest path in a static road network, two lists have been created based on the principle: the open list and the close list, which are related to the searching nodes. The open list saved all nodes that have been generated and not examined, and the close list recorded nodes that have been visited. In the searching process, the evaluation function is shown as:

$$f(n) = g(n) + h(n) \tag{3.1}$$

where $f(n)$ is the evaluation function of node n from the initial position to the goal position, $g(n)$ is the actual cost of going from the initial node to the n node, and $h(n)$ is the estimated cost of the optimal path from n to the goal position. The purpose of this research is to find the data about nodes who has the minimum value, which means it would be equally selected to get the optimal solution.

3.3.2 Analysis of the 8-puzzle for AI player

As such, an AI system equipped with the well-known A* algorithm was utilized to simulate the optimal play of human players for data collection. The AI system randomly generates the initial position of an 8-puzzle and moves the tiles in four ways: Left, Up, Right, Down to find an optimal solution to move to the goal position. The simulation was conducted up to 10,000 times for $9!/2$ solvable problem instances and computed the optimal solutions for all problem instances.

According to the game refinement theory, a good and sophisticated game always finds the balance between chance and skill [29]. Since the sophisticated zone of game refinement value for most popular games has been verified to be $GR \in [0.07, 0.08]$, the 8-puzzle game is not the most sophisticated game version among the sliding puzzle games according to the manual simulation. Table 3.1 shows that the $GR = 0.064$, which simulated through 10,000 runs, where the n and T are the average plausible options of each step and the average steps to solve the puzzles, respectively. It implies that an 8-puzzle would be enjoyable to professional or skillful players.

Based on the result in Table 3.1, considering only the average steps to solve as the

Algorithm 1 A* algorithm of N-puzzle case

```
First put initial start node in open list
while open != null do
    choose the best node from the open list
    delete n node from the open list
    add n node into the close node
    for all son node of X do
        if X is in close list then
            Continue;
        end if
        if X is in open list then
            compare the value of g in the open list and g
        end if
        if c_n is empty then
            add into open list.
        end if
    end for
    check for end node in the open list
    break.
end while
```

Table 3.1: Game refinement value of 10000 simulated times of 8-puzzle, where n and D stands for the average number of plausible options and steps to solve respectively

Simulated Times	n	D	GR
10000	1.94	21.03	0.064

game length (D) may not be a viable solution to determine the underlying mechanisms of the puzzle's attractiveness. In the 8-puzzle game, each step made by the player may or may not lead to a successful attempt to solve it. Hence, the game length (D) is defined as the total steps to solve for the 8-puzzle game, and the 8-puzzle is divided into 29 levels depending on the game length. The v was called the successful solving rate: The ratio of the number of optimal solutions within D steps and all instances. For example, $D = 5$ implies all runs that successfully solved the puzzle in five or less are included in calculating the success rate (v) for that particular game length.

Table 3.2 depicted the calculated GR and motion in mind values based on various D value. It can be observed that the GR value and the success rate (v) increase gradually with the increasing the total steps to solve the puzzle. According to the game refinement theory, when $D = 14$ GR value is 0.0737 which located in the comfortable zone of $GR \in [0.07, 0.08]$, where we obtain $m = 0.962$. Such D for the 8-puzzle would provide comfortable game sophistication that can be enjoyed by beginner players.

Table 3.2: Motion in mind measures and GR value over different total steps to solve (D) for the AI player (A* algorithm).

D	v	m	GR	F	p	E_p
10	0.0053	0.9947	0.0326	0.0011	0.0053	0.0001
11	0.0090	0.9910	0.0405	0.0016	0.0089	0.0002
12	0.0148	0.9852	0.0497	0.0024	0.0146	0.0004
13	0.0255	0.9745	0.0626	0.0038	0.0248	0.0013
14	0.0380	0.9620	**0.0737**	0.0052	0.0366	0.0028
15	0.0602	0.9398	0.0896	0.0075	0.0566	0.0068
16	0.0913	0.9087	0.1068	0.0104	0.0830	0.0151
17	0.1421	0.8579	0.1293	0.0143	0.1219	0.0346
18	0.2001	0.7999	0.1491	0.0178	0.1601	0.0641
19	0.2921	0.7079	0.1753	0.0218	0.2068	0.1208
20	0.3899	0.6101	0.1975	**0.0238**	0.2379	0.1855
21	0.5231	0.4769	0.2232	**0.0238**	**0.2495**	0.2610
22	0.6399	0.3601	0.2412	0.0209	0.2304	**0.2949**
23	0.7724	0.2276	0.2592	0.0153	0.1758	0.2716
24	0.8587	0.1413	0.2675	0.0101	0.1213	0.2084
25	0.9393	0.0607	0.2741	0.0046	0.0570	0.1071
26	0.9756	0.0244	0.2739	0.0018	0.0238	0.0464
27	0.9959	0.0041	0.2716	0.0003	0.0041	0.0081
28	0.9993	0.0007	0.2672	0.0002	0.0007	0.0014
29	1	0	0.2626	0	0.0007	0

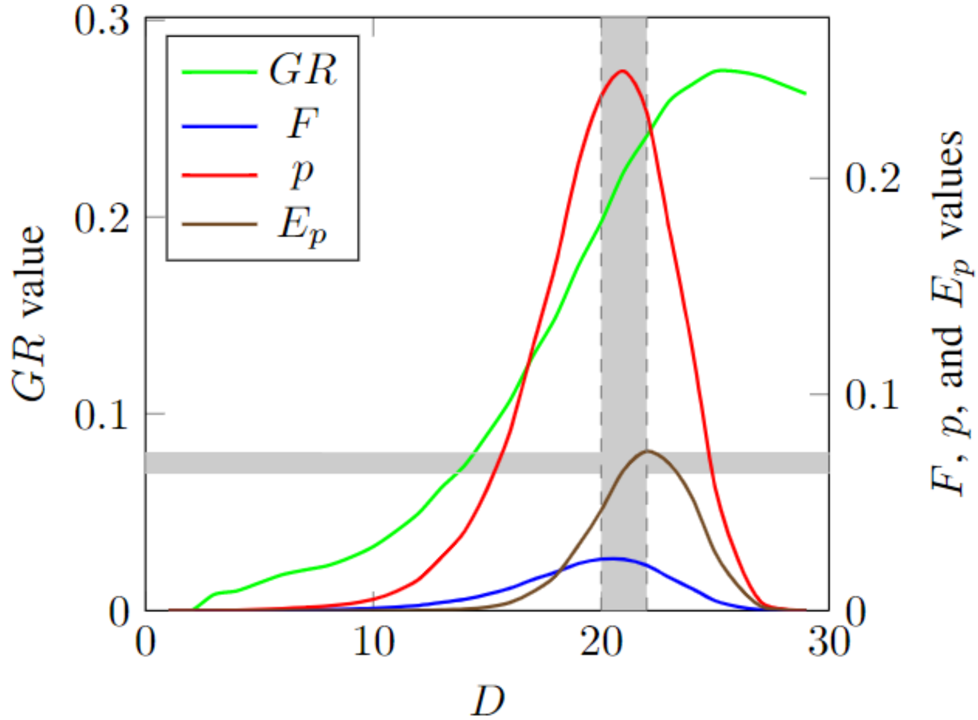


Figure 3.2: The value of GR, F, p, and E_p, with respect to D in 8-puzzle. The gray area indicates the peak range of F, p, and E_p and the GR zone

Meanwhile, Figure 3.2 shows that $D \in [21, 22]$ provides the best game playing experience for the 8-puzzle game. When $D = 21$, p reaches its peak value and F is the highest, where both values of v and m are near to 0.5. This configuration implies that the game provides players with sufficient magnitude of movement freedom and balances the amount of skill needed to the challenge given. When $D = 22$, E_p is maximum, which implies that it is the most engaged to play, which encourages movement potential and player's motivation in the game. Interestingly, one cross point provided a border interpretation of game playing behaviors. The cross point was found between $D = 21$ and $D = 22$, which indicates the moment where the 8-puzzle game is the freedom to move limits the playing ability. Moreover, the potential energy in mind have been observed with almost no growth in the initial 14 game steps, which then increases quickly to a peak at the game depth of 22, which is appropriate depth to enjoy the sliding puzzle the most.

Based on the GR values, the situation implies that the play experience balances the required skill and the element of chance, which makes the game felt more entertaining and sophisticated. According to the force, making a move in the game is challenging, yet engaging when D is larger. A similar situation can be observed in a low-score game like soccer [41]. As D increases, the engagement reduces (decrease in mass), and the ability to move becomes trivial.

Observing the peaks of p and maximum F, the results of the game tend to be unpredictable ($v \simeq m \simeq 0.5$), where the player felt enjoyable thrills because it is a fair game. For example, it is like two players of a similar level playing against each other, where the probability of winning and losing is equal (observing a seesaw effect), making the game enjoyable, thrilling, and fair.

Based on the potential energy, such a situation implies that most people may enjoy it since its peak value is reached at a higher D, implying that the game is more relaxed and comfortable to solve (i.e., a higher number of total steps were allowed). Also, $v = 0.64$ implies a little risk, and players enjoy winning the game while feeling relaxed and comfortable. For example, if two players play three sets of games and one of the players wins two games; thus, the potential energy to play a game is high due to having higher winning possibilities and making the player felt comfortable.

3.3.3 Analysis of the 8-puzzle for human player

Another experiment consisting of 290 samples for 10 volunteer players of an 8-puzzle game was conducted. The total steps to solve the game (D) relate the solution to the puzzle, which means that the game's difficulty will increase as the total step to solve increases. Unlike the experiments of AI solver, the experiments for human players were based on the difference in the known optimal solution (game length), and the puzzles were divided into 29 levels. The v was called the successful solving rate: The ratio of the number of players solved in D steps to the total number of human players. For example, when $D = 10$, $v = 1$ implies all can solve the game length of 10 for this level of 8-puzzle, and the $m = 1 - v = 0$. The results of game refinement and motion in mind measure for the human player are shown in Table 3.3.

Comparing the D of the human player with the AI player, we see in most cases that $GR \geq 0.08$ holds, which is out of the comfortable zone (Figure 3.3), implying that the 8-puzzle game would be highly stochastic as well as being dependent on the player intuition, that means more skill needed to be learned to enjoy the game more. For AI

Table 3.3: Total steps to solve 8-puzzle games for human players, as well as various motion in mind measures

D	v	m	GR	F	p	E_p
10	1	0	0.4472	0	0	0
11	0.9	0.1	0.4045	0.0164	0.09	0.1897
12	0.9	0.1	0.3873	0.0150	0.09	0.1897
13	0.7	0.3	0.3282	0.0323	0.21	0.5020
14	**0.5**	**0.5**	0.2673	**0.0357**	**0.25**	0.7071
15	0.4	0.6	0.2309	0.0320	0.24	0.7589
16	0.2	0.8	0.1581	0.0200	0.16	0.7155
17	0.2	0.8	0.1534	0.0188	0.16	0.7155
18	0.4	0.6	0.2108	0.0267	0.24	**0.7688**
19	0.2	0.8	0.1451	0.0168	0.16	0.7155
20	0.2	0.8	0.1414	0.0160	0.16	0.7155
21	0.1	0.9	0.0976	0.0086	0.09	0.5692
22	0.1	0.9	0.0953	0.0082	0.09	0.5692
23	0.1	0.9	0.0953	0.0082	0.09	0.5692
24	0.1	0.9	0.0913	0.0075	0.09	0.5692
25	0.1	0.9	0.0913	0.0075	0.09	0.5692
26	0.1	0.9	0.0877	0.0069	0.09	0.5692
27	0.1	0.9	0.0861	0.0067	0.09	0.5692
28	0.1	0.9	0.0845	0.0064	0.09	0.5692
29	0	1	0	0	0	0

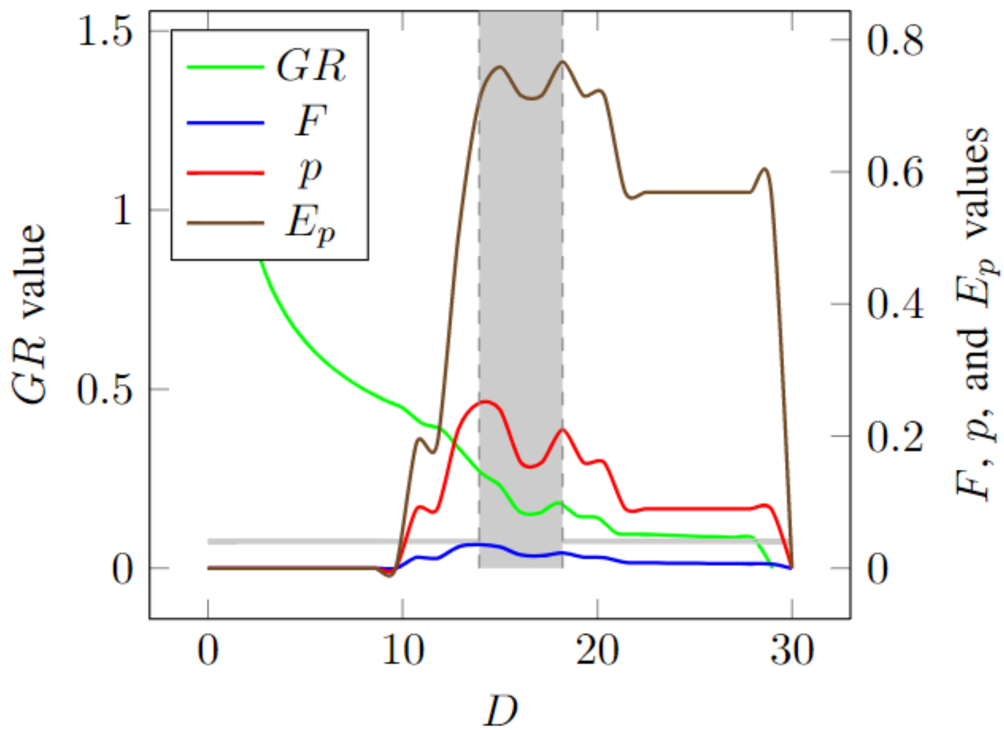


Figure 3.3: The GR value, F, p, and E_p, with respect to the game depth in 8-puzzle based on human players

players, $GR = 0.064$ (less than 0.07) means that they rely on more skill to find the optimal solution, implying that the 8-puzzle might be too easy, and the game would be enjoyable to professional or skillful players. It can be observed that $D \in [14, 18]$ is the region where $\vec{p}$ and E_p is maximum, implying the best game playing experience for the 8-puzzle. At $D = 14$, the $\vec{p}$ value is maximum while $m = 0.5$ with $GR > 0.08$, which implies that it is the most interesting to play with equiprobable risk to move (risky play). At $D = 18$, the E_p value is maximum with $m > 0.5$ implies that the game is the least difficult and easy to retain the player's motivation by having sufficient challenge.

3.3.4 Further analysis of the results

As the total steps to solve the game (D) can also be analogously interpreted as the difficulty of solving the 8-puzzle game, further analysis is conducted by dividing the player levels into five levels, designated between 1 to 5. Level 1 represents the expert players who could solve the most challenging problems, which means that the player has the ability to solve puzzles that have an optimal solution of more than 26 steps. In contrast, level

5 represents the players with the lowest ability who can only solve positions within ten steps. Table 3.4 shows the analysis of the game progress model of an 8-puzzle as a board game. It can be observed that level 5 has the highest $\vec{p}$ and E_p, which implies that players at such level would feel more freedom to move while retaining their motivation to move the game. Meanwhile, according to the GR value of Level 3, the game's difficulty has sufficiently sophistication for most players.

Table 3.4: Analysis of 8-puzzle game as a board game based on GR and motion in mind measure

Level	D	B	p	m	GR	F	$\vec{p}$	E_p
1	$(26-\infty)$	1.94	0.0461	**0.9539**	0.0663	0.0042	0.0440	0.0041
2	$(21-25)$	1.93	0.0467	0.9533	0.0671	0.0043	0.0445	0.0042
3	$(16-20)$	1.94	0.0547	0.9453	**0.0787**	0.0058	0.0517	0.0057
4	$(11-15)$	1.98	0.1184	0.8816	0.1682	0.0250	0.1044	0.0247
5	$(1-10)$	2.08	**0.2364**	0.7636	0.3277	**0.0820**	**0.1805**	**0.0853**

$m = 1 - v$ and $p = v = \frac{B}{2D}$; $a = \frac{2v}{D}$; $E_p = 2mv^2$; $F = ma = m(GR)^2$;

On the other hand, Table 3.5 shows the analysis of the game progress model of an 8-puzzle as a scoring sports game. For such measures, the major difference is found on the v value which is determined by $p = v = \frac{1}{n}$, and n stands for the average number of plausible options. Level 4 and Level 5 has the highest E_p and $\vec{p}$ respectively, which implies that the player will be retaining their high motivation and engagement, while felt thrilling due to a fair game and great movement freedom. The GR value indicates that the game sophistication at Level 3 is suited for most players, similar to what was found based on the 8-puzzle analysis as a board game.

Table 3.5: Analysis of 8-puzzle game as a scoring sports based on GR and motion in mind measure

Level	D	n	p	m	GR	F	$\vec{p}$	E_p
1	$(26-\infty)$	1.94	0.5155	0.4845	0.0642	0.0311	0.2575	0.2498
2	$(21-25)$	1.93	**0.5181**	0.4819	0.0648	0.0312	**0.2587**	0.2497
3	$(16-20)$	1.94	0.5155	0.4845	**0.0762**	0.0369	0.2575	0.2498
4	$(11-15)$	1.98	0.5051	0.4949	0.1666	0.0825	0.2525	**0.2500**
5	$(1-10)$	2.08	0.4808	**0.5192**	0.3406	**0.1769**	0.2400	0.2496

$m = 1 - v$ and $p = v = \frac{1}{n}$; $GR = \frac{\sqrt{n(n-1)}}{T}$; $E_p = 2mv^2$; $F = ma = m(GR)^2$

Comparing the two observations (Table 3.4 and Table 3.5), it can be inferred that

8-puzzle game as a scoring sports game makes more sense, where Level 5 implies highest difficulty (m) where the total steps to solve is the least (low D), and having high F value indicates high ability requirement as well as its attractiveness (to watch). Also, Level 2 showed the greatest $\vec{p}$ (movement freedom) since it has $p \simeq 0.5$, indicating that the game is fair and every step made is equiprobable to be better or worst in solving the puzzle. Interestingly, E_p is the highest for Level 4, indicating that the game is exciting and motivating. Since the GR value given at Level 3 is also the known zone value, Level 3 provides the best sophistication for an 8-puzzle where sufficient challenge, thrills, motivation, and movement freedom are satisfied; thus, being the most attractive D setting, aligned with the notion of being in the "noble uncertainty" [42]. Therefore, it is interesting to note that having infinite game steps does not translate with the game experience being engaging or exciting but took a long time instead.

3.4 Chapter Conclusion

The 8-puzzle game is a kind of single-agent game that could be beneficial to logical training, which is famous worldwide. Meanwhile, people focused on finding its optimal solutions and used the game as prominent platforms to analyze heuristic search algorithms in the AI context. Hence, it is essential to know that the reason why the game is attractiveness.

Based on the measure of game refinement theory and motion in mind, this thesis analyzed the attractiveness of an 8-puzzle game by adopting A* algorithm as the AI player to play randomly generated initial position of the puzzle. Meanwhile, the player's engagement, movement freedom, and entertainment from the aspect of the total steps to solve the game were evaluated for both AI and human players. Thus, the entertaining and sophisticated zone (total steps to solve) for different player's levels were determined.

Improving the mechanics of the 8-puzzle game (such as setting the time limits, step bonus, and different difficulties) to observe its attractiveness for various level players would be the crucial agenda in the future work. Comparing with different types of sliding puzzles, such as the one with more visual (such as an 8-pictures sliding puzzle) and challenging ones (such as Sokoban), would also be an exciting future venture. Also, the reason behind the sliding puzzle game's evolution would be another interesting prospect of future studies.

Chapter 4

Informational Dynamic of Single-Agent Stochastic Puzzle

This chapter is based on an updated and condensed version of the following publication:

- Liu Chang, Huang Shunqi, Gan Naying, Mohd Nor Akmal Khalid, Hiroyuki Iida. (2022). A Solver of Single-Agent Stochastic Puzzle: A Case Study With Minesweeper. Knowledge-Based System, 2022, 246: 108630.

4.1 Chapter Introduction

Proposing an efficient puzzle solver has been a significant research paradigm in Artificial Intelligence (AI). Since the work by [13] in 1994, efficient solving mechanisms and algorithms had been exclusively conducted on the domain of two-player zero-sum games. Based on such domain, various methodologies have been proposed which take advantage of the deterministic nature of such games, which is associated with predicting the game-theoretic value [43] [44] and determining optimal strategy [45].

Due to the wide variety of puzzles of different properties, many research efforts had been dedicated to investigating the computational complexity [7] and the possible classification of puzzles [46]. Moreover, a recent attempt at classifying puzzles had been mathematically formulated by simplifying the essential components of the puzzle into a Boolean satisfiability problem (SAT) or Constraint Satisfaction Problem (CSP) to determine classes of the puzzle [47]. Thus, a problem-specific Monte-Carlo tree search (MCTS) can effectively be applied to the respective puzzle classes based on such mathematical definitions. However, such an investigation was still limited to deterministic puzzles.

As defined by [47], there are three types of puzzle classes: class A, class B, class AB, and class C. The class A puzzle is a puzzle that can be statistically solved with simple constraints. Meanwhile, class B puzzle is a puzzle that is directly affected by the step or sequence of moves with an additional time-related dimension. Moreover, the class AB is the combination of class A and class B where the solver can address both classes. Finally, class C is the puzzle that is outside of the other classes and has at least one random feature and/or inputs.

The 2048 and Minesweeper are examples of the class C puzzle, since the former had random input variables in every state while the latter contains random placement of hidden mines. Recently, a 2×2 version of 2048 had been solved by [48]. A similar puzzle, namely Minesweeper, corresponds with a single-agent incomplete-information game [22] that fits the criteria of the class C puzzle, which had been recently investigated to determine the reason for its attractiveness [49]. With the growing community around solving Minesweeper, various AI agents or solvers have been introduced in the literature [50–55].

Nevertheless, a clear definition of puzzle categories is needed from the perspective of its solvability. For such purpose, the Minesweeper puzzle was utilized as the benchmarking testbed. In such a condition, what distinguishes a deterministic puzzle from other types of puzzle, i.e., a stochastic puzzle? Therefore, this thesis proposes a definition of a stochastic puzzle from its solvability, which forms the foundation for the proposed AI solver. Moreover, the proposed AI solver takes advantage of both the deterministic and stochastic elements of the puzzle, which called PAFG: the primary reasoning strategy ("P"), the advanced reasoning strategy ("A"), the first action strategy ("F"), and the guessing strategy ("G"). Such strategies also combine mathematical model, knowledge-driven rules, and linear transformation to provide conducive moves in solving Minesweeper, comparable to previously proposed Minesweeper AI solvers.

4.2 Game Testbed: Minesweeper

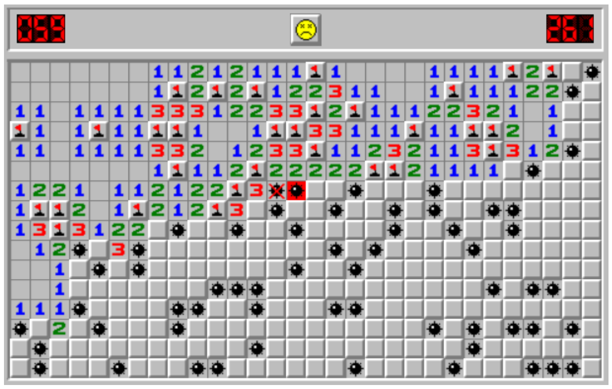
Minesweeper is a classic single-agent incomplete-information game, which gains its popularity as it became a feature of Microsoft Windows in 1989 [22, 23]. In the 1980s, various Minesweeper-type video games were designed, such as Mined-Out, Yomp, and Relentless Logic[1]. After that, Minesweeper was integrated into the entertainment pack in many generations of Microsoft Windows, like 3D, hexagonal, triangular, and multiple Minesweepers. This thesis focuses on the size of the standard board of the Minesweeper: 9×9, 16×16, and 16×30, which are the three furthest boards that have survived from the emergence of the Minesweeper until now. And, based on these three puzzle configurations, experimental simulations are performed using changing the number of mines.

The goal of Minesweeper is to open all non-mine hidden cells on the board and flag all the mines on the board. If the player opens instead of flagging a mine cell incorrectly, the game is lost. For example, Figure 4.1 (a) and (b) show a losing state and a winning state of Minesweeper ($16 \times 30|99$ mines), respectively. In this chapter, we use three classic Microsoft Minesweepers as the testbed: beginner ($9 \times 9|10$ mines), intermediate ($16 \times 16|40$ mines), and expert ($16 \times 30|99$ mines). To simplify its reference throughout the process, these three puzzles were designated as board labelled as X, Y, Z, respectively.

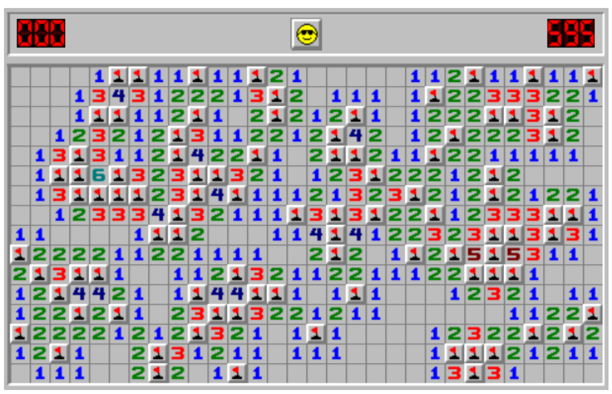
The discrete model-based cellular automaton was first used to solve Minesweeper proposed by [56] in 1951. Various distribution and mines density were recorded, and several agents would automatically solve the problem mapped to specific networks by making random choices using an adaptive algorithm. Besides, [27] found that solving Minesweeper is an NP-complete problem which means that Minesweeper is hard to be solved in polynomial time. Meanwhile, [50] formulated Minesweepers as Constraint Satisfaction Problem (CSPs) with finite constraints or limitations to be satisfied. A heuristic strategy with a backtracking algorithm was used to find the best guess in solving the game.

Based on the researcher's study of Minesweeper, this thesis is directed at proposing solving strategies applicable to the characteristics of stochastic puzzles, with the intention of developing the AI solver with high winning rates state-of-the-art, as well as analyzing the game mechanics and characteristics of Minesweeper. More importantly, it explores the

[1] http://www.minesweeper.info/wiki/Windows_Minesweeper#Windows_3.1



(a) A losing state of Minesweeper



(b) A winning state of Minesweeper

Figure 4.1: Two final states of Minesweeper 16 × 30|99 mines, the purpose is to find all hidden mines on the board without opening them. (a) is a losing state because the player opened on a mine in the game process, (b) is a winning state while the player revealed all mines on the board.

classification of puzzles and the border between puzzles and games based on the solvability of Minesweeper.

4.3 Research Methodology

The overall of the PAFG strategy is given by Algorithm 2. The first cell is opened according to the first action rule, which ensures the first opened cell and the surrounding cells are safe. Then, the primary reasoning strategy is simple and effective, the advanced reasoning strategy is used based on the primary reasoning strategy, and the problem switches to finding invariables in a linear system of equations. Moreover, the Gauss-Jordan elimination method is used to get general solutions, and some invariables can be found since the solutions satisfy certain constraints. Furthermore, the guessing strategy is based on the result of Gauss-Jordan elimination, and the binary tree search based on the linear system model is used, where the binary tree is pruned according to the constraints on the solutions of linear systems.

The time complexity for the primary reasoning strategy is $O(n)$, n is the number of unsolved number cells. For the advanced reasoning strategy, the time complexity is $O(n^3)$, and n is the number of variables in the system of equations. Moreover, the time complexity for the guessing strategy is $O(2^n)$, where n is the number of variables, and setting the number of free variables $n \leq 20$. The size of the entire search tree is capped at 2^{20}, and tree pruning dramatically reduces the size of the search tree; thus, significantly reducing the expected complexity.

4.3.1 Building Minesweeper AI

The configuration of Minesweeper is denoted by $R \times L \mid b$, where $R > b$ and $R, L, b \in \mathbb{N}^*$, R and L are the number of rows and columns on the board, respectively, and b is the number of mines. For any cell c, it can be represented by a pair of unique position (x, y), i.e., $c = (x, y)$, where $1 \leq x \leq R$, $1 \leq y \leq L$ and $x, y \in \mathbb{N}^*$, x and y are the row and column number, respectively. The cell type $I(c)$ is in the range of $[-1, 10]$, the meaning is as follows:

Algorithm 2 The Overall Algorithm

Input B: game board

open the first cell according to the first action rule
while the game is not end **do**
    $S_1, S_2 \leftarrow PRS(B)$
    if $S_1 and S_2$ are not empty **then**
        open the cells in S_1
        flag the cells in S_2
    end if
    $S_1, S_2, C, V \leftarrow ARS(B)$
    if $S_1 and S_2$ are not empty **then**
        open the cells in S_1
        flag the cells in S_2
    end if
    $S_1, S_2 \leftarrow GS(B, C, V)$
    open the cells in S_1
    flag the cells in S_2
end while

- $I(c) = -1$: c is an exploded mine.
- $I(c) = 0$: c is the number "0" displayed as a blank opened cell.
- $1 \leq I(c) \leq 8$: c is a number that equals $I(c)$.
- $I(c) = 9$: c is a hidden cell.
- $I(c) = 10$: c is a flagged cell.

Here, the $I(\cdot)$ is a function that its input is the position of a cell, and the output is a type integer. The cells set $C = \{\, (x, y) \mid 1 \leq x \leq R,\ 1 \leq y \leq L,\ x, y \in \mathbb{N}^* \,\}$. The neighborhood $N(c)$ is a set surrounding cells(adjacent or diagonally adjacent) of c, i.e, $N(c) = \{\, (x, y) \mid x_0 - 1 \leq x \leq x_0 + 1,\ y_0 - 1 \leq y \leq y_0 + 1,\ (x, y) \in C,\ (x, y) \neq (x_0, y_0) \,\}$. And, we define the $N(\cdot)$ as a function that its input is the position of a cell, and the output is a set of positions of surrounding cells, and it is impossible that two different cells have one same neighborhood. A corner cell has $|N(c)| = 3$, a border cell has $|N(c)| = 5$, and other unsolved number cell if $1 \leq I(c) \leq 8$, and there exists a cell $c^* \in N(c)$ such that $I(c^*) = 9$, and an unsolved block U is defined as a set of cells that satisfying two conditions:

- For any $c \in U$, c is an unsolved number cell.
- If $|U| \geq 2$, for any two different $c_1, c_2 \in U$, there exists a path connecting c_1 and c_2.

For an unsolved block U, the frontier $F(U)$ is the set of cells that for any $c \in F(U)$, there exists an unsolved number cell $c^* \in U$ such that $c \in N(c^*)$. A cell c is an uninformed cell if $I(c) = 9$ and for any $c^* \in N(c)$, c^* is not a number cell. The uninformed set is the set of uninformed cells. Figure 4.2 is an example of Minesweeper 9×9 board size. The gray blank cells have $I(c) = 0$. The cells with black "f" are flagged cells, and the cells with numbers are numbered cells. The set of orange blank cells is the frontier $F(U_1)$, the set of blue and purple blank cells is the frontier $F(U_2)$, the set of green and purple blank cells is the frontier $F(U_3)$, and the white cells are uninformed cells.

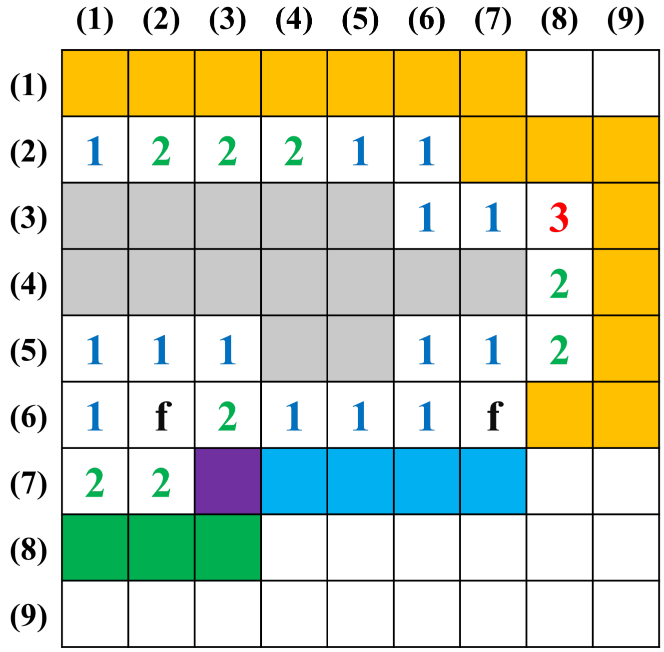


Figure 4.2: The basic information of the cells on the board for minesweeper $(9 \times 9 \mid 10)$, there are three unsolved blocks on the board: $U_1 = \{(2,1), ..., (2,6), (3,6), ..., (3,8), (4,8), (5,7), (5,8)\}$, $U_2 = \{(6,3), ..., (6,6)\}$, $U_3 = \{(7,1), (7,2)\}$.

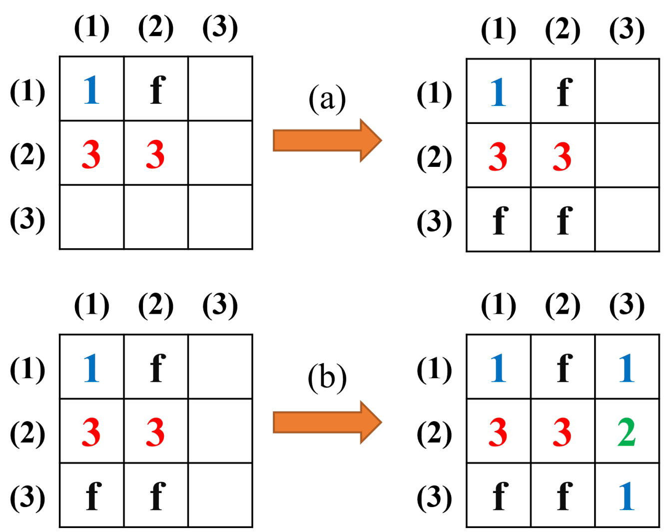


Figure 4.3: An illustration of primary reasoning strategy: The cells with "f" and numbers are flagged cell number cells, respectively.

Primary Reasoning Strategy

The primary reasoning strategy has two rules. Firstly, the surrounding hidden cells are safe for any cell when their value equals the number of surrounding flagged mines. Secondly, the surrounding hidden cells are mines for any cell when the number of surrounding hidden cells equals its value minus the number of surrounding marked mines. These reasoning strategies were devised under the mathematical solving logic of players with the basic knowledge and rules, whose general flow is described in Algorithm 3. For an unsolved number cell c, suppose that f^* is the set of flagged cells in $N(c)$ and h^* is the set of hidden cells in $N(c)$, i.e., $f^* = \{c^* \mid I(c^*) = 10, c^* \in N(c)\}$ and $h^* = \{c^* \mid I(c^*) = 9, c^* \in N(c)\}$. The line of $6 - 9$ is one of the key ideas in Algorithm 3. It shows if $|h^*| = I(c) - |f^*|$, then for any $\bar{c} \in h^*$, $\bar{c}$ must be a mine.

Algorithm 3 Primary Reasoning Strategy

Input B: game board

Output S_1: the set of safe cells, S_2: the set of mine cells

1: **Function** PRS(B)
2:     S_1, $S_2 \leftarrow \{\}$ ▷ empty set
3:     **for** each unsolved number cell c on the board B **do**
4:         $f^* \leftarrow$ the set of flagged cells in $N(c)$
5:         $h^* \leftarrow$ the set of hidden cells in $N(c)$
6:         **if** $|h^*| = I(c) - |f^*|$ **then**
7:             $S_2 \leftarrow h^*$
8:         **else if** $I(c) = |f^*|$ **then**
9:             $S_1 \leftarrow h^*$
10:         **end if**
11:     **end for**
12:     **Return** S_1, S_2
13: **end Function**

Since c is an unsolved number cell, $I(c)$ indicates the number of mines in $h^* \cup f^*$, the cells in f^* are flagged cells (or flagged mines), thus, there are $I(c) - |f^*|$ mines in h^*. There are $\binom{a_1}{a_2}$ way(s) to distribute mines, where $a_1 = |h^*|$, $a_2 = I(c) - |f^*|$, then, $\binom{a_1}{a_2} = 1$ since $|h^*| = I(c) - |f^*|$ (i.e., for any $\bar{c} \in h^*$, $\bar{c}$ must be a mine cell). If $I(c) = |f^*|$, then for any $\bar{c} \in h^*$, $\bar{c}$ must be a number. Since there are $I(c) - |f^*|$ mines in h^*, $I(c) = |f^*|$, there is no mines in h^*, i.e., for any $\bar{c} \in h^*$, thus, $\bar{c}$ must be a number cell. Figure 4.3 is an example of primary reasoning strategy in Minesweeper case. Then, the Figure 4.2 can be obtained since the cell $(2, 1)$ has a number "3," so $(3, 1)$ and $(3, 2)$ are mines that are flagged, and the Figure 4.3 is obtained since the cell $(2, 2)$ has a number "3"; thus, $(1, 3)$, $(2, 3)$ and $(3, 3)$ are safe to be open.

Advanced Reasoning Strategy

The advanced reasoning strategy is based on the primary reasoning strategy with a deeper understanding of the mechanics of Minesweeper and knowledge-driven methods that exploit existing information. Firstly, the frontier division is conducted to get independent frontiers. Then, we see each hidden cell of this frontier as a Boolean variable for each independent frontier [50]. Subsequently, linear equations are proposed to find solution invariability in the constrained linear system, where it always has the same value on all possible solutions; thus, determining the corresponding cell is safe or not.

Frontier Division: Suppose that there exists some unsolved blocks $U_1, \dots, U_k$ $(k \in \mathbb{N}^*)$ on the board, and the frontiers of these unsolved blocks are $F_i = F(U_i)$ $(i = 1, \dots, k)$. Then, two different frontier F_1 and F_2 can be defined, where there exists an edge between F_1 and F_2 if $F_1 \cap F_2 \neq \phi$, and F_1 is dependent to F_2 if there exists some edges that can connect F_1 and F_2. Moreover, for three different frontier F_1, F_2, and F_2, if F_1 is dependent to F_2 and F_2 is dependent to F_3, then F_1 is dependent to F_3. Thus, hidden cells are divided into different independent frontiers $\hat{F}_1, ..., \hat{F}_n$. Figure 4.4 is an example of Frontier Division where the blue, orange and green parts are independent (i.e., The distribution of mines in one part would not affect other parts).

Boolean Modeling: Suppose that $\hat{F} = \{c_1, ..., c_k\}$ is an independent frontier, for any cell $c_i \in \hat{F}$, let $W(c_i) = \{c^* \mid c^* \in N(c_i), 1 \leq I(c^*) \leq 8\}$, where $W(c_i)$ is a set of number cells that are in the neighborhood of c_i. And, let $W = W(c_1) \cup \cdots \cup W(c_k)$, where W is a union of all the $W(c_i)$. Then, each $c_i \in \hat{F}$ is a variable $x_i \in \{0, 1\}$, where "0" means c_i is a safe cell, and "1" means c_i is a mine cell. Then, a system of linear equations can be obtained according to the information $W = \{w_1, \dots, w_q\}$. Each equation of E has the form as (4.1), where $i = 1, ..., k$, $a_i \in \{0, 1\}$, $x_i \in \{0, 1\}$, $b \in \{1, 2, 3, 4, 5, 6, 7, 8\}$.

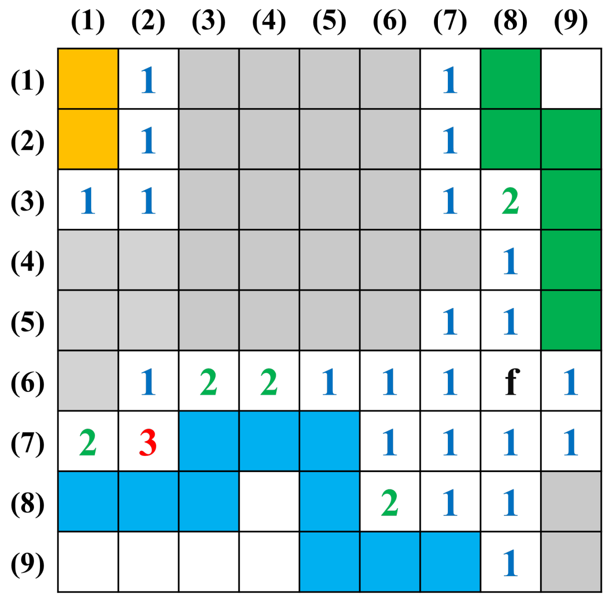


Figure 4.4: Frontier division: an example of 9×9 Minesweeper configuration with the cells marked with blue, orange, and green colors to signify different and independent frontiers.

$$a_1x_1 + ... + a_kx_k = b \tag{4.1}$$

V can be seen as a set of all bijective mappings between c_i and x_i. A mapping is bijective if and only if it is injective and surjective. For any $w \in W$, the left side of the equation is the sum of $V[\{c \mid c \in N(c), I(c) = 9\}]$, the right side is $I(w) - |\{c \mid c \in N(c), I(c) = 10\}|$. Figure 4.5 is an example of modeling the hidden cells (blue color) into Boolean variables. We can see $(4,1)$ as x_1, $(4,2)$ as x_2, ..., $(4,6)$ as x_6, then $V[1] = (4,1)$, ..., $V[6] = (4,6)$. The system of linear equations is shown as (4.2), where $x_1, ..., x_6 \in \{0,1\}$.

$$\begin{cases} x_1 + x_2 & = 1 \\ x_1 + x_2 + x_3 & = 2 \\ x_2 + x_3 + x_4 & = 2 \\ x_3 + x_4 + x_5 & = 2 \\ x_4 + x_5 + x_6 & = 2 \end{cases} \tag{4.2}$$

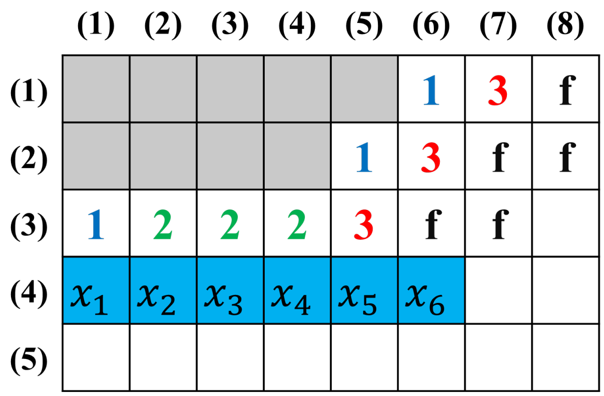


Figure 4.5: Boolean model of hidden cells: an example of a partial 9 × 9 Minesweeper configuration (top part) with each hidden cells were modelled with Boolean variable $x_i \in \{0, 1\}$, where "0" means c_i is a safe cell, and "1" means c_i is a mine cell.

Gauss–Jordan Elimination Algorithm: After modeling hidden cells in an independent frontier to Boolean variables, a system of linear equations can be established as $\Phi : \boldsymbol{Ax} = \boldsymbol{b}$, where for any element in $\boldsymbol{A}$ or in $\boldsymbol{x}$, the value is 0 or 1. Here, the Gauss–Jordan elimination algorithm is used to solve the system of linear equations by transforming the augmented matrix of the linear system to its reduced row echelon form, which transforms the original linear system into a simpler equivalent system [57].

Elementary row operations are needed to find the reduced row echelon form of a matrix. Suppose that $\boldsymbol{B}[i]$ denotes the i-th row of a matrix $\boldsymbol{B}$, there are three types of elementary row operations:

- **Row swapping**: Swap the two different rows $\boldsymbol{B}[i]$ and $\boldsymbol{B}[j]$, i.e., $\boldsymbol{B}[i] \leftrightarrow \boldsymbol{B}[j], i \neq j$.
- **Row multiplication**: Multiply each element in $\boldsymbol{B}[i]$ by a non-zero constant c, i.e., $\boldsymbol{B}[i] \leftarrow c\boldsymbol{B}[i], c \neq 0$.
- **Row addition**: Add a multiple of a row $\boldsymbol{B}[j]$ on another row $\boldsymbol{B}[i]$, i.e., $\boldsymbol{B}[i] \leftarrow \boldsymbol{B}[i] + c\boldsymbol{B}[j], c \neq 0$.

If a matrix $\boldsymbol{C}$ can be obtained from another matrix $\boldsymbol{B}$ by elementary row operations, then $\boldsymbol{B}$ and $\boldsymbol{C}$ are equivalent. The function $JS : \boldsymbol{B} \rightarrow \boldsymbol{C}$ is the function that does Gauss–Jordan elimination to $\boldsymbol{B}$, where $\boldsymbol{B}$ is the augmented matrix of a system of linear equations, $\boldsymbol{C}$ is the reduced row echelon form of $\boldsymbol{B}$. The size of $\boldsymbol{B}$ and $\boldsymbol{C}$ are same. Let $\boldsymbol{B}$ be the augmented matrix of Φ, and $\boldsymbol{C} = JS(\boldsymbol{B})$, i.e., $\boldsymbol{C}$ is the reduced row echelon form of $\boldsymbol{B}$. Suppose that $\boldsymbol{B}$ and $\boldsymbol{C}$ are $\alpha \times \beta$ matrices. For any $i \in \{1, \ldots, \alpha\}$ and $j \in \{1, \ldots, \beta\}$, $\boldsymbol{B}[i][j]$ denotes the element of the i-th row and j-th column of $\boldsymbol{B}$.

Conjecture 1 *For the i-th row $\boldsymbol{C}[i]$, if there exists $k \in \{1, ..., \beta-1\}$ such that $\boldsymbol{C}[i][k] = 1$ and $\boldsymbol{C}[i][j] = 0$ for any $j \in \{1, ..., k-1, k+1, ...\beta-1\}$, then the variable $x_k = \boldsymbol{C}[i][\beta]$.*

Remark 1 *Since $\boldsymbol{C}$ is the reduced row echelon form of $\boldsymbol{B}$, where $\boldsymbol{B}$ is the the augmented matrix of a system of linear equations, $\boldsymbol{C}[i][j]$ $(1 \leq j \leq \beta - 1)$ is the coefficient of the variable x_j in the i-th equation, thus, if only one element $\boldsymbol{C}[i][k] = 1$ in the first $\beta - 1$ elements, $x_k = \boldsymbol{C}[i][\beta]$.*

Conjecture 2 *For the i-th row $\boldsymbol{C}[i]$, suppose that $\delta = \{j \mid j = 1, ..., \beta - 1, \boldsymbol{C}[i][j] \neq 0\}$, if $\boldsymbol{C}[i][\beta] = 0$, and $\boldsymbol{C}[i][k] > 0$ for any $k \in \delta$ or $\boldsymbol{C}[i][k] < 0$ for any $k \in \delta$, then, $x_k = 0$.*

Remark 2 *Suppose that $\Delta = \{j \mid j = 1, ..., \beta - 1, \boldsymbol{C}[i][j] \neq 0\} = \{\delta_1, ..., \delta_t\}$. Since $x_{\delta_1}, ..., x_{\delta_t} \in \{0, 1\}$, if $\boldsymbol{C}[i][\delta_1] > 0, ..., \boldsymbol{C}[i][\delta_t] > 0$, then*

$$\boldsymbol{C}[i][\delta_1]x_{\delta_1} + ... + \boldsymbol{C}[i][\delta_t]x_{\delta_t} \geq 0 \tag{4.3}$$

where equality $x_{\delta_1} = ... = x_{\delta_t} = 0$. (The case that $\boldsymbol{C}[i][\delta_1] < 0, ..., \boldsymbol{C}[i][\delta_t] < 0$ is similar, only need to replace "$\geq$" with "$\leq$"). Since $\boldsymbol{C}[i][\beta] = 0$, the equation is:

$$\boldsymbol{C}[i][\delta_1]x_{\delta_1} + ... + \boldsymbol{C}[i][\delta_t]x_{\delta_t} = 0 \tag{4.4}$$

From (4.3) and (4.4), $x_{\delta_1} = ... = x_{\delta_t} = 0$ is obtained.

Conjecture 3 *For the i-th row $\boldsymbol{C}[i]$, suppose that $\Delta = \{j \mid j = 1, ..., \beta - 1, \boldsymbol{C}[i][j] > 0\}$, $\Gamma = \{j \mid j = 1, ..., \beta - 1, \boldsymbol{C}[i][j] < 0\}$. If $\boldsymbol{C}[i][\beta] = \sum_{k \in \Delta} \boldsymbol{C}[i][k]$, then for any $\delta \in \Delta$, $x_\delta = 1$, and for any $\gamma \in \Gamma$, $x_\gamma = 0$. If $\boldsymbol{C}[i][\beta] = -\sum_{k \in \Gamma} \boldsymbol{C}[i][k]$, then for any $\gamma \in \Gamma$, $x_\gamma = 1$, for any $\delta \in \Delta$, $x_\delta = 0$.*

Remark 3 *Suppose that $\Delta = \{j \mid j = 1, ..., \beta - 1, \boldsymbol{C}[i][j] > 0\} = \{\delta_1, ..., \delta_t\}$, $\Gamma = \{j \mid j = 1, ..., \beta - 1, \boldsymbol{C}[i][j] < 0\} = \{\gamma_1, ..., \gamma_h\}$. Since $x_{\delta_1}, ..., x_{\delta_t} \in \{0, 1\}$, with equality*

$x_{\delta_1} = ... = x_{\delta_t} = 1$. Since $x_{\gamma_1}, ..., x_{\gamma_h} \in \{0, 1\}$, with equality $x_{\gamma_1} = ... = x_{\gamma_h} = 0$.

$$\boldsymbol{C}[i][\delta_1]x_{\delta_1} + ... + \boldsymbol{C}[i][\delta_t]x_{\delta_t} \leq \sum_{k=1}^{t} \boldsymbol{C}[i][\delta_k] \tag{4.5}$$

$$\boldsymbol{C}[i][\gamma_1]x_{\gamma_1} + ... + \boldsymbol{C}[i][\gamma_h]x_{\gamma_h} \leq 0 \tag{4.6}$$

From (4.5) and (4.6), we can get equation(4.7). Otherwise, since $\boldsymbol{C}[i][\beta] = \sum_{k=1}^{t} \boldsymbol{C}[i][\delta_k]$, the equation is shown as (4.8):

$$\boldsymbol{C}[i][1]x_1 + ... + \boldsymbol{C}[i][\beta - 1]x_{\beta-1} \leq \sum_{k=1}^{t} \boldsymbol{C}[i][\delta_k] \tag{4.7}$$

$$\boldsymbol{C}[i][1]x_1 + ... + \boldsymbol{C}[i][\beta - 1]x_{\beta-1} = \sum_{k=1}^{t} \boldsymbol{C}[i][\delta_k] \tag{4.8}$$

From (4.7) and (4.8), $x_{\delta_1} = ... = x_{\delta_t} = 1$ and $x_{\gamma_1} = ... = x_{\gamma_h} = 0$.

The algorithm of the advanced reasoning strategy is shown as Algorithm 4.

Algorithm 4 Advanced Reasoning Strategy

Input B: game board

Output S_1: the set of safe cells, S_2: the set of mine cells

Function ARS(B)
    S_1, $S_2 \leftarrow \{\}$ ▷ empty set
    Do frontier division to get independent frontiers
    for each independent frontier $\hat{F}$ **do**
        Switch cells in $\hat{F}$ to variables
        Establish a system of linear equations Φ
        $V \leftarrow$ bijective mappings between cells and variables
        $B \leftarrow$ the augmented matrix of Φ ▷ $\alpha \times \beta$ matrix
        $C \leftarrow GJ(B)$ ▷ C is the reduced row echelon form of $\boldsymbol{B}$
        for $i = 1, ..., \alpha$ **do**
            $\Delta \leftarrow \{j \mid j = 1, ..., \beta - 1, \boldsymbol{C}[i][j] > 0\}$
            $\Gamma \leftarrow \{j \mid j = 1, ..., \beta - 1, \boldsymbol{C}[i][j] < 0\}$
            if $\Delta = \phi$ AND $\Gamma = \phi$ **then**
                Go to next loop
            end if
            if $|\Delta \cup \Gamma| = 1$ **then**
                if $\boldsymbol{C}[i][\beta] = 0$ **then**
                    Add $V[j](j \in \Delta \cup \Gamma)$ into S_1
                else
                    Add $V[j](j \in \Delta \cup \Gamma)$ into S_2
                end if

Algorithm 4 Advanced Reasoning Strategy (continued)

else if $\boldsymbol{C}[i][\beta] = \sum_{k \in \Delta} \boldsymbol{C}[i][k]$ **then**
Add $V[j](j \in \Gamma)$ into S_1
Add $V[j](j \in \Delta)$ into S_2
else if $\boldsymbol{C}[i][\beta] = -\sum_{k \in \Gamma} \boldsymbol{C}[i][k]$ **then**
Add $V[j](j \in \Delta)$ into S_1
Add $V[j](j \in \Gamma)$ into S_2
end if
end for
end for
Return S_1, S_2
end Function

The First Action Strategy

The first action strategy aims to find the best position of the first opened cell and acquire more information that determines the safety of the following cells. In other words, choosing the best initial position can significantly improve the winning rate. A study by [58] defined the neighborhood as a set of cells relative to the first cell while [59] supposed that none of the first cell's neighbors are mines. Here, the safe neighborhood rule is proposed with the basis of the first action rule and the neighbor cell, and both are for the first cell opened. The discussion of the safe first action rule and the safe neighborhood rule is as follows.

Safe First Action Rule: The safe first action rule is that the first opened cell must be safe [59]. Suppose that c_0 is the first opened cell, mines will be placed in other cells $\{c \mid c \neq c_0\}$. Then, it is assumed that the game configuration is $R \times L \mid b$, and $R - 8 \geq b$, then probability c_0 is given by (4.9).

$$Pr(I(c_0) = 0) = \frac{\binom{b}{RL-|N(c_0)|}}{\binom{b}{RL}} \tag{4.9}$$

$$= \frac{(RL - |N(c_0)| - b + 1)...(RL - |N(c_0)|)}{(RL - b + 1)...(RL)} \tag{4.10}$$

To get the best first action, we need to find the first opened cell c_0 that maximize the probability that c_0 is a "0". The cardinality of c_0 is defined as (4.11). Since $R - 8 \geq m$, $|N(c_0)| = 3$ can maximize $Pr(I(c_0) = 0)$. Thus, the corner cells are the best first action under the safe first action rule of $\frac{1}{4}$ of the board size.

$$|N(c_0)| = \begin{cases} 3, & c_0 \text{ is a corner cell} \\ 5, & c_0 \text{ is a border cell} \\ 8, & \text{other cases} \end{cases} \tag{4.11}$$

Safe Neighborhood Rule: The first action strategy also explores the safe neighborhood rule to enhance the winning rate, which means the first opened cell and its neighborhood must be safe. Suppose that c_0 is the first opened cell, mines will be placed in other cells $\{c \mid c \neq c_0, c \notin N(c_0)\}$. In this case, the probability c_0 is given by (4.12).

$$Pr(I(c_0) = 0) = 1 \tag{4.12}$$

Such condition allowed for more information to be obtained when c_0 was found. Suppose that F is the frontier after doing the first action, and $Z = \{c \mid c \in F, I(c) = 0\}$. The problem switches to: if $|Z|$ is a constant, how to maximize $|Z|$? This implies that the closer c_0 to the center of the board is, the larger the value of $|Z|$.

Guessing Strategy

The guessing strategy extends the two preceding reasoning strategies, which corresponds to the stochastic elements during the solving process of Minesweeper, where each cell on the board has uncertainty, and the player cannot determine the safe cell from the available information. However, players of different abilities use the guessing strategy differently; for example, advanced players avoid using the guessing strategy whenever possible, while beginners may use this strategy all the time. The key to the guessing strategy is to find a hidden cell that maximizes its safety probability. The mathematical solving logic of expert players is inspired by the skill of computing possibilities based on limited information. After dividing variables into free and leading variables by Gauss-Jordan elimination, the binary tree search searches all possible combinations of free variables to find numerical solutions. Finally, to reduce complexity, we pure the binary tree according to the constraints on the general solutions of linear systems.

Constrained Search: Suppose that there is one independent frontier on the board, and the advanced strategy is used to get a system of linear equations Φ. Let $\boldsymbol{B}$ be the augmented matrix of Φ, where $\boldsymbol{B}$ is a $\alpha \times \beta$ matrix. $\boldsymbol{C}$ is the reduced row echelon form of $\boldsymbol{B}$, which is got by Gauss–Jordan elimination [60]. The purpose is to find a hidden cell c with minimal risk and provide more information. The Gauss–Jordan elimination divided variables into free variables V_f and leading variables V_l. Any variable $v_l \in V_l$ is a linear combination of free variables in V_f and a constant, so the solutions are presented by free variables.

Since the value of all variables are in $\{0, 1\}$, the binary tree search is used to find all possible numerical solutions of Φ, where the critical point is need to search free variables that can significantly reduce the scale of the search. The binary tree has f free variables, and the depth of the root node is 0. The nodes with the depth less than f are non-leaf nodes, which variable equals the depth $d(1 \leq d \leq f-1)$. The left/right child node of a non-leaf node n, and the variable is 0/1. Figure 4.6 shows an illustration of binary tree search for free variables. Here, the completed node is defined as a node n that satisfies either of these conditions: (a) n is a leaf node; (b) each child node of n is a completed node or an illegal node. Moreover, a node n is an illegal node if one of the conditions is satisfied: (a) the number of free variables equals 1 is significantly greater than r (remaining mines); (b) there exists a leading variable not in $\{0, 1\}$ under the variable state of n. Thus, when pruning a binary tree, if a node is an illegal node, then no need to search its child nodes. Otherwise, it stores the states of nodes from the root node to the current node, and each legal leaf node presents a possible free variable. Thus, all possible numerical solutions of Φ are obtained.

Additional Enhancements: The additional technique is to choose a variable that would provide more information when the values of variables $\hat{x_1}, ..., \hat{x_t}$ are near to "0". Then, it is assumed that $x_i = 0$ for $i = 1, ..., t$, substitute $x = 0$ back into Φ by back-

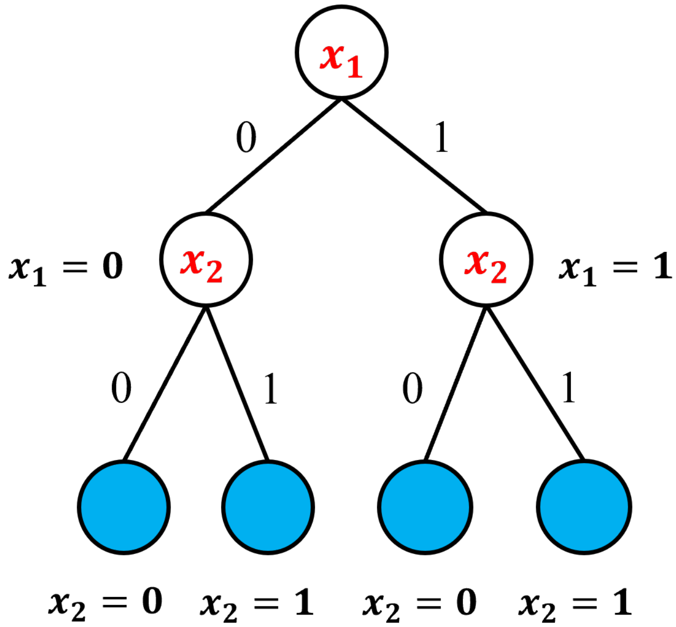


Figure 4.6: Binary tree search for free variables: x_1 and x_2 are free variables, blue nodes are leaf nodes, which represents four variable states: $x_1 = 0, x_2 = 0$; $x_1 = 0, x_2 = 1$; $x_1 = 1, x_2 = 0$; and $x_1 = 1, x_2 = 1$.

tracking method, and denote $Inf(x_i)$ as the number of invariables of $\bar{\Phi}$. Finally, the x_i is chose with the largest $Inf(x_i)$. However, all possibilities of hidden cells are searched by brute-force search when the number of hidden cells on the board is insufficient. Since there is no information on the board, the best choice is to open a cell far from the solved area since it has a higher probability of opening a "0" cell. The algorithm of the guessing strategy is as Algorithm 5.

Algorithm 5 Guessing Strategy

Input B: game board, S: the set of solutions of linear systems

Output c_0: the cell that will be opened

```
Function GS(B)
    if S = φ then
        Choose a hidden cell c_0 on B far from the solved areas randomly
    else if the number of hidden cells on B is less than 6 then
        Do a brute-force search to choose the safest c_0 on B
    else
```

Algorithm 5 Guessing Strategy (continued)

$Cand \leftarrow \{\}$
for each $s \in S$ **do**
  $V_f \leftarrow$ a list of free variables
  Create the root node n_r
  $n_c \leftarrow n_r$ ▷ current node
  $P \leftarrow \{\}$
  while n_r is not a completed node **do** ▷ Search free variables
    if n_c is a leaf node **then**
      Substitute the value of the free variables into s
      Add a possible numerical solution into P
      $n_c \leftarrow$ the parent node of n_c
      Update the state of n_c
    else if n_c is a completed node OR n_c is illegal **then**
      $n_c \leftarrow$ the parent node of n_c
      Update the state of n_c
    else if the left child node of n_c was not created **then**
      Create a left child node n_0
      $n_c \leftarrow n_0$
    else
      Create a right child node n_1
      $n_c \leftarrow n_1$
    end if
  end while
  Calculate probabilities that cells in p are safe
  Add the safest cells and their probabilities into $Cand$
end for
$c_0 \leftarrow$ the safest cell in $Cand$
Return c_0
end if
end Function

4.4 Experiment Results and Analysis

In this section, we compare the winning rates of Minesweeper under sub-strategies in the PAFG strategy and illustrate the similarities and differences with other methods or strategies, as well as the winning rate comparisons between the PAFG strategy and existing methods, and the discussion of the future enhancement of the strategy.

4.4.1 First action rule and safe first action rule

The first action rule is used for the first step on the board, and the safe first action rule means the first opened cell is safe. This paper conducted experiments on both rules to increase the probability of game to proceeds and avoid the player failing at the first step. The experiment results under the first action rule and safe first action rule are reported in Table 4.1. The AI agent proposed based on the "PAFR" strategy were running for 10,000 games for the beginner (9×9 | 10 mines), intermediate (16×16 | 40 mines) and expert (16×30 | 99 mines) configurations of the Minesweeper. It can be observed that the winning rate of the safe first action rule achieves 87.9%, 78.2%, 39.7%, which is 10.9%, 12%, 8.1% higher than the first action rule.

Table 4.1: Comparison for first action rule and safe first action rule based on different Minesweeper configuration.

Rule	9×9 | 10	16×16 | 40	16×30 | 99
First Action Rule	77%	66.2%	31.6%
Safe First Action Rule	87.9%	78.2%	39.7%

9×9 | 10: 9×9 board size with 10mines;
16×16 | 40: 16×16 board size with 40 mines;
16×30 | 99: 16×30 board size with 99 mines.

4.4.2 Safe first action rule and safe neighborhood rule

More information obtained means a higher probability of finding more safe cells and mines, which was the basic idea of the safe neighborhood rule. The specific rule is that the neighbor cells around the first opened cell are safe to obtain more board information. The difference between the safe first action rule and the safe neighborhood rule is the different location of the first action cell. Figures 4.7 and 4.8 show the winning rate of choosing different first opened cells under different first action rules based on the game configuration of $8 \times 10 \mid 12$, which is based on a quarter of the whole board. To better demonstrate the results under these rules, this experiment takes Minesweeper in configuration $8 \times 10 \mid 12$ as an example and 10,000 runs per cell as the first open cell for both safe first action rule and safe neighborhood rule.

The AI agent is based on the "PAR" strategy, where "R" is the random guessing strategy. The result shows that the cell on the top left corner has the highest winning rate of 71.5% based on the safe first action rule, while the cell closer to the center with the winning rate around 85% under the safe neighborhood rule. Furthermore, it can obtain from Figure 4.8 that the cell closer to the center has a higher winning rate, which suggests that the AI solver following the safe neighborhood rule has a better chance of getting a high winning rate. Since the correctness of the safe neighborhood rule has been proven by the experimental results above, all the simulation experiments utilizing the PFAG strategy are based on the safe neighborhood rule.

	(1)	(2)	(3)	(4)	(5)
(1)	71.5	69.0	67.3	68.5	67.4
(2)	69.3	67.4	64.5	63.6	64.0
(3)	67.7	64.6	62.2	61.4	61.8
(4)	67.0	63.4	61.1	60.7	59.8

Figure 4.7: Safe first action rule. The winning rate of different first opened cells (game configuration: $8 \times 10 \mid 12$ mines with "PAR" AI agent strategy on safe first action rule).

4.4.3 Guessing strategy and random guessing strategy

The difference between PAFG and PAFR strategy is that "G" and "R" are different, where "R" randomly guessing cells to open the cells, while "G" chooses the most promising safe

	(1)	(2)	(3)	(4)	(5)
(1)	77.9	81.2	82.6	83.0	82.7
(2)	81.4	82.1	83.4	83.7	83.7
(3)	82.5	83.8	84.5	85.0	84.5
(4)	82.8	83.5	84.6	84.6	84.8

Figure 4.8: Safe neighborhood rule. The winning rate of different first opened cells (game configuration: 8×10 | 12 mines with "PAR" AI agent strategy on safe neighborhood rule).

one. The game configuration in this paper: beginner 9×9 | 10, intermediate 16×16 | 40 and expert 16×30 | 99, which is a feature of Microsoft Windows. In order to compare the experimental data with the previous studies, the experiment was conducted 10,000 times with PAFG* AI agent with three Minesweeper configurations: beginner (8×8 | 10 mines), intermediate (16×16 | 40 mines) and expert (16×30 | 99 mines), listed in Table 4.2. Comparing with PAFR strategy, the PAFG strategy improved the winning rate for three configurations (9×9 | 10, 16×16 | 40 and 16×30 | 99) of minesweeper by 2.6%, 6.7% and 18.1%, respectively. In this case, it can be seen that for all three puzzle configurations, the "G" strategy has a significantly higher win rate than "R" strategy. For the guessing strategy, the key is to use the information obtained, and to pay attention to the existing information and search for unknown information is also a future direction.

4.4.4 Comparison of methods and strategy

To the best of our knowledge, the strongest AI in previous works is PSEQ-D256 [54], where the proposed strategy has better performance for beginner level, the same for intermediate, and 0.4% less for expert; thus, generally better performed than PSEQ strategies. The difference between PSEQ-D256 and PSEQ is that it uses several situations to choose OPTIMAL procedure or HEURISTIC strategies. Moreover, it is a solver composed of heuristic strategy and the quasi-optimal procedure while using the criterion of the number of situations; that is also why the performance at the expert level is slightly higher than the strategy in this paper.

Table 4.2: Comparison of various methods and strategies from the previous works against the proposed PAFG strategy performance (winning rate in percentage) in solving the Minesweeper.

Strategy	**Beginner**	**Intermediate**	**Expert**
PAFR⋆ (this paper)	93.8%	79.6%	27.5%
PAFG⋆ (this paper)	96.4%	86.3%	45.6%
PAFG (this paper)	82%	86.3%	45.6%
PSEQ-D256 [54]	81.8%	78.2%	40.1%
PSEQ [54]	81.6%	78.1%	39.6%
OH [53]	80.2%	74.4%	38.7%
cSimEnuLoClf [52]	80.0%	75.6%	37.5%
LSWPE [51]	N.A.	67.7%	25.0%
Lordeiro (ϵ-greedy) [55]	77%	57.9%	4.1%
Lordeiro (UCB) [55]	76.4%	44.5%	0.62%
CSPStrategy† [50]	80.0%	44.3%	33.9%
Pedersen⋄ [51]	92.5%	67.7%	25%

beginner: $8 \times 8 \mid 10$, intermediate: $16 \times 16 \mid 40$, and expert: $16 \times 30 \mid 99$;
†: the configuration is: $8 \times 8 \mid 10$, $15 \times 13 \mid 40$, and $16 \times 30 \mid 99$;
⋆: the configuration is: $9 \times 9 \mid 10$, $16 \times 16 \mid 40$, and $16 \times 30 \mid 99$;
⋄: the configuration is: $10 \times 10 \mid 10$, $16 \times 16 \mid 40$, and $16 \times 30 \mid 99$.

It is worth noting that [54] had probed the corner blocks to be the first step to improving their strategies' success rate. This condition is well validated by the safe first action rule in this paper. Moreover, this thesis proposed the safe neighborhood rule to enhance the performance of the PAFG strategy. As a result, even with the random guessing strategy (R), the winning rate shown in Table 4.1 is higher than guessing strategies "P" and "T" in PSEQ. After that, [54] discussed the PSEQ strategies based on the maximum probability of the block and heuristic methods; this optimal and heuristic-based strategy has a much higher win rate than the previous strategy. However, a remote similarity between the "G" strategy in PAFG and PSEQ maximizes the probability of the hidden safe cell. At the same time, the constrained search and Gauss-Jordan elimination was used to reduce the scale of the search, together with the backtracking method to enhance the strategy. Thus, even without using heuristic methods, PAFG provides a high-level solver of Minesweeper compared with PSEQ-D256 state of the art. Nevertheless, the difficulty of solving Minesweeper increases as the density of mines increases, which means the "G" strategy needs to be used more often and choose the safest cell from the cell with currently available information, without considering the part of the cell with unknown information.

4.5 Discussion

This chapter focuses on classic Microsoft Minesweeper by proposing an AI solver based on the PAFG strategy that reaches the reported winning rate (see Table 4.2) to facilitate the study of the puzzle-solving process. The four strategies proposed were adopted to imitate different player abilities in solving problems, specifically, addressing a single-agent game, such as Minesweeper. With the presence of stochastic elements (e.g., hidden information of the mine location from all available cells on the board), the solvability is dependent on improving the condition that can be regarded as a 'puzzle' while mitigating the condition that can be regarded as a 'game.' In another word, mitigating the most on the ratio of certainty against uncertainty.

Moreover, the definition of solvability that distinguishes 'puzzle' and 'game' from the context of the stochastic single-agent game had been introduced and expanded from the two-player zero-sum game. By adopting an appropriate Minesweeper AI strategy, the experiment conducted on the various board size achieved adequate and comparable performance to the strategy of the related works. With the reported winning rate of the proposed PAFG given in Table 4.2, it is clear that based on the configuration of the experiments, for instance, the 9×9 Minesweeper was solvable 82% of the time as a puzzle. Meanwhile, Minesweeper was played as a game for the remaining 18% of the time. This condition can be associated with the information dynamics of the solving process where the random/hidden information of the single-agent game affected the solvability of the positions/moves, which makes it either solvable or not.

From the solvability point of view, the strategy of Minesweeper AI in solving problems is based on cells with information on the board, since in most cases, the safety probability of cells with information is more distinctive, which decides to be more informed compared to the cells without any information. Moreover, in some cases, it is easy to determine whether the hidden cells on the board are mines or not. Such a situation is when there is a relationship between the number of hidden cells and the number of mines minus the number of flag cells, which constitutes the primary reasoning strategy ("P" strategy). However, the purpose is to find all hidden mines on the board. As the information increases, the problem is switched to finding variables in a Boolean variable linear system of equations, which is over-determined in theory. Here, Gauss-Jordan elimination helped reduce the augmented matrix into the row echelon form. The general solutions to the system can be obtained with the advanced reasoning strategy ("A" strategy). Moreover, the guessing strategy ("G" strategy) was used to find a hidden cell and maximize its safe probability when all cells have a probability of being mines instead of open cells randomly. In addition, the rule proposed in the first action strategy ("F" strategy) always keeps both the first cell opened and the surrounding cells safe to obtain more information in the first step.

The strength of the PAFG strategies was the complementary nature of the four strategies to achieve a high winning rate in solving Minesweeper. In other words, each strategy provides a balance between what is currently known and what is unknown of the puzzle state to decide the best possible future state of the puzzle. Based on the reported experimental results, this condition showed that having such a balance is well-suited for Minesweeper, where stochastic elements are always present, inducing insufficient state of the art approaches in dealing with such cases.

Nevertheless, if there is no available information on the board, opening a cell far from the solved area is the best choice. The current maximum probability of safe cells was obtained using a conditional probability based on known information. However, there are cases where the maximum probability of safe cells based on known information is lower than that of cells in regions of unknown information, such as when the density of the remaining mines is low. Thus, balancing exploration and utilization based on known information is an expecting direction of future improvement.

4.6 Chapter Conclusion

A knowledge-based rule verified by the experimental simulation had found that cells closer to the center are the best positions for the "F" strategy with the safe neighborhood rule. Furthermore, other strategies were used to obtain a high winning rate based on available information alongside game progression. For example, the safety probability of each surrounding cell was independently calculated. In addition, the advanced reasoning strategy is used to invariably find a linear system of equations based on the primary reasoning strategy mathematically. Moreover, the Gauss-Jordan elimination is adopted to obtain general solutions in a linear system of equations, exploring the guessing strategy to independently reduce the augmented matrix into the row echelon form under the information of the considered cell. Compared with the solving strategy state of the art, this AI solver reached a high winning rate with: 96.4% for $9 \times 9(10 mines)$ board size, 86.3% for $16 \times 16(40 mines)$ board size, and 45.6% for $16 \times 30(99 mines)$ board size.

The design concept of this AI solver is based on the solving logic of players, which is modeled mathematically to facilitate future entertainment analysis for different abilities players to solve Minesweeper. Furthermore, more strategies correspond to the increased player's ability, which implies a higher winning rate, verified by proposed knowledge-driven reasoning, strategies, and simulation conducted in this thesis. Promising future work includes exploring the PAFG strategy in other related single-agent games, such as 2048 and similar games, to expand further and verify the definition established from this thesis. Moreover, adopting the PAFG strategy as a tool to analyze and measure the entertainment aspects of the single-agent stochastic game and other deterministic puzzles is exciting prospects for future investigation.

Chapter 5

Finding the Border Between Games and Puzzles

5.1 Chapter Introduction

Many games and puzzles can be converted into each other. We can easily turn many puzzles into games, and vice versa. For example, Go as a game is composed of many small puzzles. Games can have puzzles as a subset that together make up the game, as many games do. And in most cases, the puzzles are an integral part of the overall game, because games that place most of their challenge value on the puzzles they contain quickly lose the uncertainty once all the puzzles have been solved. So, what exactly distinguishes the game from the puzzle? This chapter describes the differences and connections between games and puzzles, as well as current researches on solving puzzles and playing games. By listing studies in the field of puzzles, the gap between current research on puzzles and players is pointed out. furthermore, a model of information change during puzzle solving is proposed, using N-puzzle and Minesweeper as testbeds to analyze players' entertainment based on solving information. More importantly, the classification of puzzles is defined from the perspective of solvability, leading to finding the border between puzzles and games.

5.2 Games and Puzzles

The earliest distinction made on games and puzzles had been previously discussed [61], which emphasized the ability for ludic activity changes where a puzzle presents a singular thing to "solve" (i.e., rule-based), while a game changes and responds based on the player's action and might be unpredictable (i.e., mechanics). Meanwhile, several researchers (e.g., [62], [63] and [64]) have mixed opinions on such distinction where the terminology such as "puzzle game" was used in a broader sense while argues that all "games are puzzles to solve." Building on this understanding, it was noted that "almost every game has some degree of puzzle-solving," but finally concluded that there is a distinction, summarized that a puzzle is static, while a game is interactive [65]. A recent study outlined that puzzles and games shared distinctive traits based on the experience they provide to the players, where puzzles being a subset of games that are rooted in a playable experience, their affordances and relationships with their players, and their places in the context of "casual" games [4].

Studies indicate that emotions play an essential role in decision-making by utilizing functional magnetic resonance imaging in Ultimatum game [66]. Also, decision-making ability was measured for soccer players based on two levels: the player's technical-tactical skills (subjective way) and the tactical context (objective way). Similar to the board game or sports game that uses the best move selection as a critical point of winning it [67] [68], the average branching factor is the critical elements in 'playing' games or 'solving' puzzle. While brute-force methods can be adopted to solve games, its consumption in space and time would be infeasible [67]. Therefore, studies about the optimal solution of puzzle games focused on search space reduction, like Breadth-First Search (BFS) and Depth First Search algorithms (DFS) [69]. Usually, an optimal solution was found with the least steps or cost in a game, which depends on reducing the branching factor of different heuristic evaluation functions [70] [71].

A puzzle is an enjoyable game with a right or fixed answer to solve [1]. As one of the problem-solving modes, the puzzle consists of several sub-problems that could independently be solved depending on players' abilities (such as hand-eye coordination and logic)

[1] https://scottkim.com/

and engaged players to accomplish challenging goals [72]. This thesis focuses on two types of puzzles: N-puzzle, Minesweeper. These puzzles represent different problem-solving under the setting of different state information dynamics (sequencing in N-puzzle, stochastic in Minesweeper).

However, what all these puzzles have in common is discovering solutions to solve subproblems that make up the goal state. It is interesting to know the dynamic solving rate in puzzle solving progress and players' experience. A recent study based on a physics-based puzzle analyzes game levels design from two dimensions of axiality and density from the perspective of game designers. One study suggests generating playable content through defining better fitness functions or developing an effective AI agent to play the game more effectively [73]. Generating the more playable content in a puzzle is essential from the game design point of view, which is also vital to understand players' behavior and how people think in the game [74]. Such a study pointed out that perceptual speed is a critical factor in solving Tangram puzzles and analyzing the correlation between selecting puzzle pieces over time.

Based on previous work, the motion in mind concept uncover the motion changes of game's and players' information [1] [31]. Such studies describes the player's motion in mind for various game uncertainty (m) over time. This thesis investigates the dynamic uncertainty of problem-solving progression in different puzzles, and the interplays that a player experience between solving and playing such puzzles. It can be said that the information of the puzzle moves forward with different solving rates (or velocity) relatives to players' abilities. This thesis explores how the solving rate (v) changes in different puzzles, with motion in mind from both objective and subjective views.

5.2.1 Current gaps in puzzle studies

Studies on puzzles had mainly been rooted in the application of puzzles as a tool to understand the process of learning better. It was also agreed that puzzles had been a commonly accepted method to introduce students to spatial thinking [75], computational thinking [76] [77], and even musical thinking [78]. For instance, some studies had examined the learning and enjoyment aspect during game-playing by determining the best pace challenges being presented from a game design perspective [63], in which a pragmatic approach of problem-solving taken from behavioral psychology where three metrics were used (unit of analysis, actions to completion, and presentation of novel skill). Meanwhile, another study conducted a mixed-method approach in which problems with learning structure within a game are identified along with the game progress, and the different strategies players adapt to overcome that problem is determined by comparing the optimum solutions in puzzles against observed player performance [79].

Another aspect of puzzle studies involves understanding player's strategies in solving puzzles. For example, one study proposed a visualization technique to compare behavior sequences and capture strategies enacted by players in solving a puzzle, allowing the verification of specific hypotheses about those strategies [80]. In contrast, another study presented human-interpretable rules of puzzle-solving strategy, where a rich structure of domain-specific knowledge and strategies that are not obvious from the description of a game's rules to be generalized for better understanding [81]. In another perspective, the impact of strategy and dexterity on video games is that a player must use strategies to decide between the number of moves and the dexterity to execute those moves correctly [82]. An artificial intelligence (AI) agent simulates human-like strategy and dexterity using player modeling by integrating them into a single framework that quickly measures score distributions in interactive puzzles.

A causal relationship has been suggested in action video games to improve various visual and attention skills by doing experiments in the habitual video-game and non-video-game players [83]. Recent research experiments have discussed an action video game, a puzzle game, or a real-time strategy game on the skills differences between players, including attention, memory, and executive control [84]. It illustrated players' skills and specific abilities that could be enhanced during the game's progress [84] [85]. This thesis sought to identify significant factors related to players' ability in the puzzle game domain.

Players' ability dramatically influences the game progress and resulted in differences among players. However, it is challenging to determine the puzzle's velocity of progression from the motion in mind perspective. How can the change of solved uncertainty, associated with the velocity (or the winning rate in general), be defined in this context? In both the board or sports game domain, the best move selection model is adopted [1]. Meanwhile, solutions in most puzzles can be seen as a sequence of moves. In addition, as opposed to a two-player game tree, the puzzles' game tree is viewed as indistinguishable tree breadths, and the optimal solution is selected at each step [86]. Moreover, changes of game-playing experience between different puzzle progression relative to the player's ability are analyzed, and its essential entertainment elements are established.

5.3 Informational Progress Model in Puzzle-solving

The model of move candidate selection based on skill and chance has been illustrated as Figure 5.1, which has been previously proposed in the domain of board games [41] [1] [29]. This illustration shows that skillful players would consider a set of fewer plausible candidates (b) among all possible moves (B) to find a move to play. A core part of the original game with branching factor B can be transformed into a stochastic game with a smaller branching factor b since it is assumed that each candidate may be equally selected.

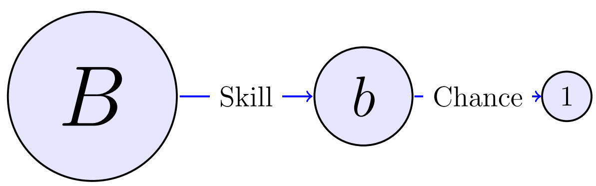


Figure 5.1: An illustration of move selection model based on skill and chance (adopted from [1])

Similarly, such a branching factor b exists in a puzzle game case, where the player could select one among b with equal probability to get the optimal solution at each step of the game process. The more b at one step, the greater the movement freedom to select in that step and the easier it is to move the game forward. Hence, reducing the candidates' solution (B) down to a few (b) is crucial in puzzle game cases, where each candidate has an equal probability of being selected. Unlike two-player board games with solving uncertainty, a puzzle game is a classic single-agent problem-solving game with a definite goal. The game can be solved if a solution to this problem is found, and usually, the expert player can find the optimal solution. It is assumed that the game length of a solvable puzzle game equals 1; otherwise, the game length is 0.

The proposed game progress model of game uncertainty is based on board games work by [87], where the uncertainty on the game result is the number of moves in board games

n. Here, the information on the solution of puzzle games is defined as the amount of solved uncertainty (or information obtained) $y(x)$, as given by (5.1).

$$y(x) = \left(\frac{x}{B}\right)^n \tag{5.1}$$

Where $y(x)$ $(0 \leq y(x) \leq 1)$ and x $(0 \leq x \leq B)$ are the uncertainty solved and uncertainty represented by the branching factors, respectively. Here, n denotes the number of plausible moves at a given position of the puzzle under the consideration. There is no uncertainty after solving a puzzle, so the velocity of the solving process is given by (5.2), which is the first derivation of the solved uncertainty.

$$y'(x) = \frac{n}{B^n} x^{n-1} \mid_{x=B} \quad = \frac{n}{B} \tag{5.2}$$

Thus, the probability of solving a puzzle at each step p_i is obtained as (5.4), where b_i is candidate options of equally being selected for optimal solutions, and B is all branching factors. When $b_i = B_i$, then $p_i = 1$, which means that all branching factors are optimal or has no information to solve at the current step, and the player has complete freedom to select any of them. Conversely, $b_i = 1$ means that only one search tree branch can be selected to reach the optimal solution. Moreover, if $v = 0$, there is no solution for the current situation. In this case, the average of solving a puzzle v and m are obtained as (5.3), where D is the game length (total steps of solving), and m is the risk of solving a puzzle.

$$p_i = \frac{b_i}{B_i} \quad \text{or} \quad p_i = 1 \tag{5.3}$$

$$v = \frac{1}{D} \sum_{i=1}^{D} p_i \quad \text{and} \quad m = 1 - v \tag{5.4}$$

For deterministic puzzles, the ratio of b to B is related to the probability of getting the optimal solution. Take N-puzzle as an example, the puzzle is a deterministic puzzle, meaning that there is a definite solution to the puzzle, but the solution obtained may not be optimal depending on the player's ability. Therefore, this informational progress model is significant to measure the uncertainty of obtaining the optimal solution to a deterministic puzzle. But for stochastic puzzles, the ratio of b to B is about whether there is a deterministic solution to the puzzle. This is because the hidden information in an uncertainty puzzle increases the probability that the puzzle is unsolvable. Take Minesweeper as an example, in general, an increase in the number of mines will lead to an increase in uncertain information. How to measure uncertain information in stochastic puzzles is one of the further purposes of the proposed informational progress model.

5.4 Game Testbeds: N-puzzle and Minesweeper

In this section, we use N-puzzle and Minesweeper as representatives of deterministic and stochastic puzzles, respectively, and conduct simulation experiments for information analysis in solving puzzles and exploring puzzle mechanisms. At the same time, the classification of puzzles is defined based on the winning rate of Minesweeper, and the boundaries of puzzles and games are found.

5.4.1 Informational progression of N-puzzle

N-puzzle is a single agent game that consists of $n-1$ numbered, movable tiles set in a frame because one cell in the frame is always empty so that the cells around it can be moved. The simulation result based on 8-puzzle, 15-puzzle, and 24-puzzle are summarized in Table 5.1. It is observed that N-puzzle are easily solvable by AI agent, making its win rate always 1. However, depending on the board size, the dynamics of the solving rate would differ during the process.

Table 5.1: Analysis of n-puzzle game on motion in mind

N-puzzle	v	E_p	p_1	p_2	v_2
8	0.3877	0.1841	0.2374	-0.0533	-0.0871
15	0.2930	0.1214	0.2072	-0.0858	-0.1213
24	0.2365	0.0854	0.1805	-0.0952	-0.1246

In general, it was observed that an increase in the board size decreases the solving rate, illustrating the increase in difficulty as the N-puzzle board size increases. Comparing these three board sizes, 8-puzzle has the most E_p and v_2, indicating the highest motivates in the game-playing experience with some errors (negative v_2) in reaching the goal state (associated with the notion of worry). Such experiences diminished when the board size increases into 15-puzzle and 24-puzzle, where reductions of E_p and v_2 were observed.

This research focuses on understanding the experience of the puzzle-solving process by utilizing the game progress model of the N-puzzle. The dynamic solving rate (v) of the 8-puzzle game progress is shown in Fig 5.2. It can be observed that the solving rate drops from peak to trough and repeats as the solving process. A common pattern observed is that all the peak values are 1 and the trough is close to 0, and the speed drops more and more slowly as the game progress. As the branching factor b value increases or decreases, the corresponding v value would accelerate or decelerates. This dynamic v value reflects the player solving the puzzle's heuristic searching process and indicates the difficulties of solving this puzzle at the current step. For example, if $v_i = 0.2$, the difficulty of not finding the optimal solution is $m_i = 0.8$ for step i.

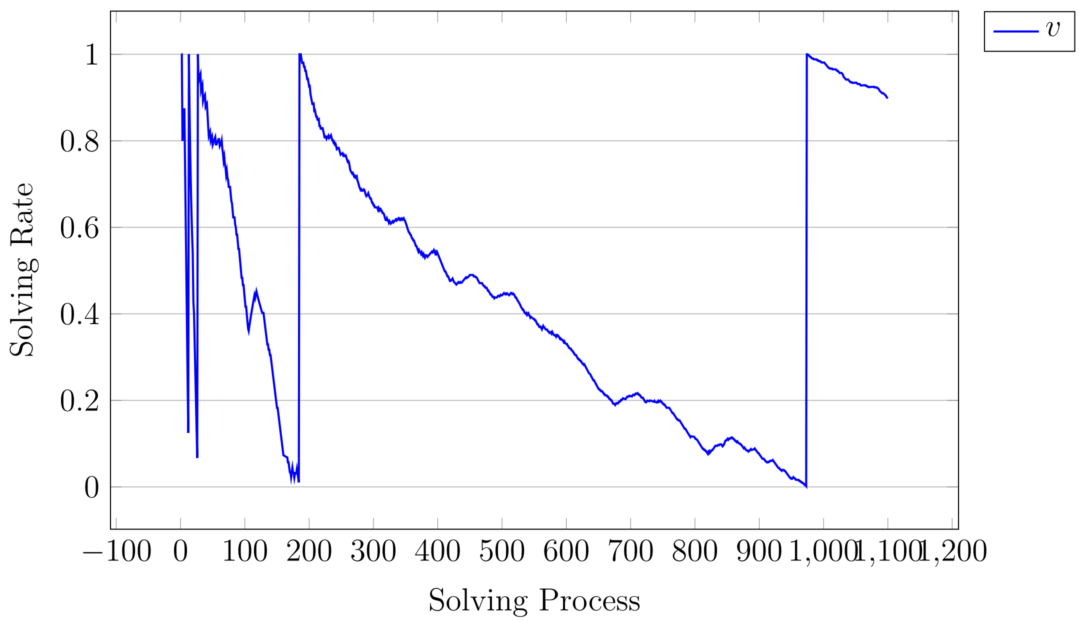


Figure 5.2: Dynamic process of solving rate of 8-puzzle

Moreover, Figure 5.3 showed the distribution of v value in the solving 8-puzzle process within 500 runs. As we can see, value $v<0.2$ accounted for the largest proportion, and the number of nodes searched decreases with the increase of the solving rate, while the number of $v = 0.1$ was the least. This situation illustrates the difficulty of finding the optimal solution in solving 8-puzzle since there are many cases where the player is uncertain, as opposed to $v = 1$.

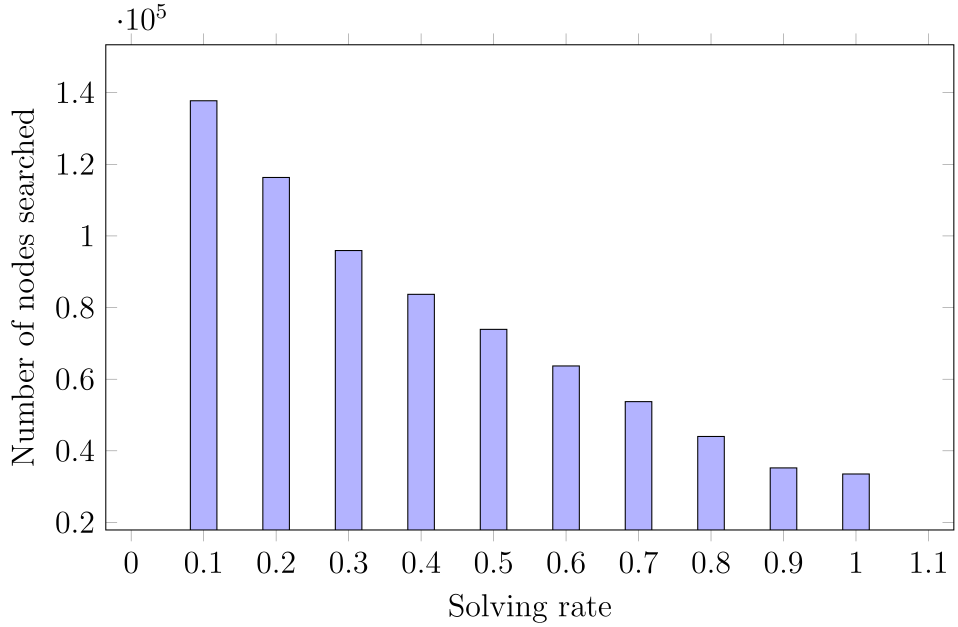


Figure 5.3: The distribution of the solving rate in the search process when solving 8-puzzle.

Based on the study's findings, the puzzle provides a different entertainment experience where the information perceived and received by the players was efficiently manipulated and carefully refined. For instance, N-puzzle explores the information space where the solving rates were in a "seesaw" effect slowly throughout the play processes (see Figure 5.2), making both the motivation and challenge of the puzzle marginal most of the time, while being surprising fresh at certain moments. It was an exciting take on information aspects that involves process sequencing. At the same time, it can be associated with strategic decision-making in business and engineering, where some decision was valid and feasible for the long-term interest while in some rare occasions, requires a drastic measure (e.g., application in industrial supply-chain and process planning problems).

5.4.2 Informational progression of Minesweeper

The distribution of the winning rates for the standard $9 \times 9|10$, $16 \times 16|40$, $16 \times 30|99$ configurations of Minesweeper are depicted in Figure 5.4. The experimental results were based on 5000 instances each, where the winning rate was divided into three parts: $0 - 0.9$, $0.9 - 1$, and 1. As we can see, the 9×9 instances with a winning rate of 1 take up the largest fraction of the three, which means that this Minesweeper has the highest proportion of certainty. On the contrary, 16×30 board size have the highest uncertainty in solving it and simultaneously making it the most difficult among the three configurations. Moreover, as the board size increases, the ratio of $0.9-1$ significantly increases, while the ratio of $v<0.9$ increases the least, indicating the standard Minesweepers board sizes were expected to provide sufficient play experience for non-novice players. Moreover, all three configurations of Minesweeper show a large number of winning rates in $0.9-1$ compared to a winning rate $v<0.9$ as a percentage of the total number in the presence of uncertain information, which can indicate that the solving strategy proposed in this thesis works well for different configurations of Minesweeper.

Also, from the above results, we can observe that a large board and a high number of mines will lead to a lower winning rate for Minesweeper. This means that a large board and a high number of mines will generate more hidden information, leading to more unsolvable factors for Minesweeper and thus a lower win rate. Since Minesweeper involves stochastic elements in the puzzle-solving process, two factors were considered to corroborate the essential findings from such puzzles: the number of mines and board sizes. Detailed discussion on such factors is given in the following sections.

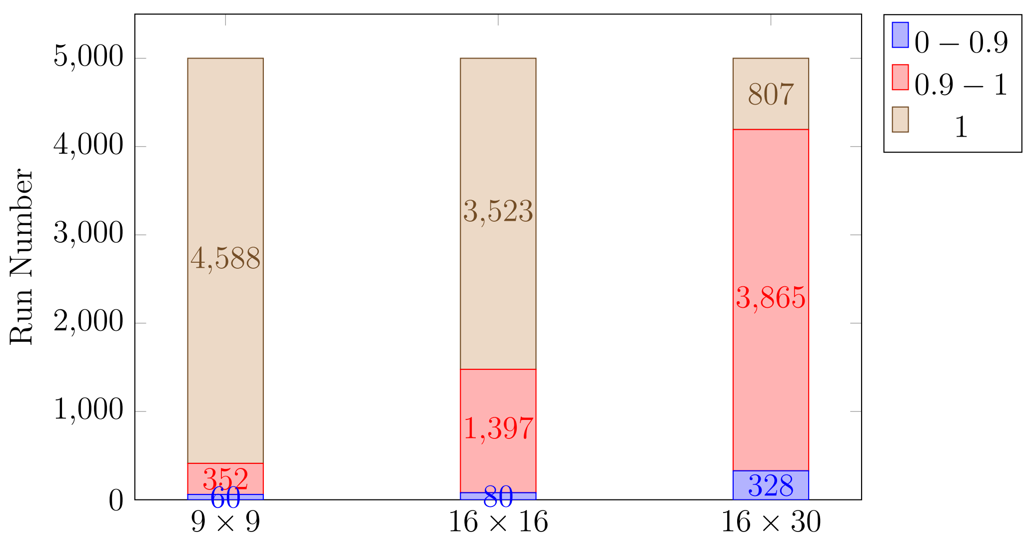


Figure 5.4: The distribution of solving rates for the three standard Minesweepers board sizes

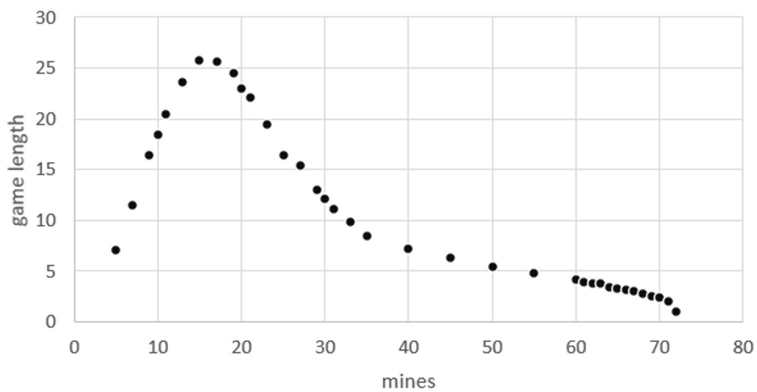
Moreover, together with Figure 5.5 shows the game length and the number of guess times for solving Minesweeper 9×9 board size various the number of mines $M \in [1, 72]$, each mine 2000 runs. We can observe that as the number of mines increases, the curves of both the game length and the number of guesses increase and then decrease. In particular, the game length peaks in subfigure (a) when the number of mines is 17, and after that it keeps decreasing to the end. This situation means that after a certain threshold of hidden information is reached, the probability that the puzzle can be solved continues to decrease until finally Minesweeper is solved with just one step. Also, in subfigure (b), although the number of guesses first increases and then decreases, after the number of mines is 17, the number of guesses is mostly greater than the value of 17, which shows that the increase in the number of mines leads to an increase in uncertainty, and the increase in the hidden information also leads to an increase in the uncertainty of whether the puzzle is solvable or not.

Similarly, Figure 5.4 shows that as the size of the board increases and the number of mines increases, the proportion of puzzles with a solving rate (winning rate) of less than 1 increases. There is a relationship between the uncertainty information in the puzzle and these two values, and it may be that the density of the number of mines reaches a certain threshold, leading to a certain threshold of uncertainty information, which in turn affects the solvability of the puzzle.

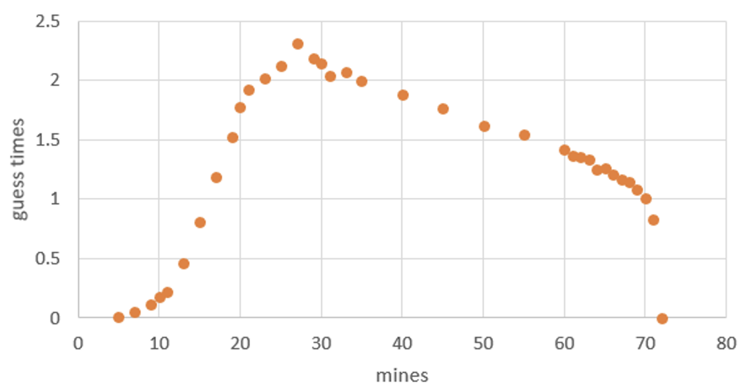
The study based on the solvability of deterministic puzzle and stochastic puzzles, gives inspiration for the next to define about the different categories of puzzles, and to search the border between puzzles and games. Allis et al. [13] proposed a definition of game solvability based on ultra-weakly solved, weakly solved, and strongly solved. Puzzles can be composed into games, and the fun of solving them can be enjoyed in solving them, and similarly, games can be interchanged with puzzles, so it is interesting to study the boundary between puzzles and games. Moreover, the classification of puzzles offers a new perspective, no longer based on the complexity of the mitosis, but on the winning rate due to uncertainty information.

5.4.3 The solvability of single-agent game: evidence from Minesweeper

Generally, defining the solvability of a game depends on the observable outcome of the game. In the context of two-player zero-sum games, [13] has defined solvability from the perspective of game-theoretic values, which have three different levels: ultra-weakly solved, weakly solved, and strongly solved. Such a solvability definition revolves around the legal positions (or moves) made in the game (initial or all). As a result, a varying degree of game-theoretic value is achieved under reasonable resources. Such a degree of



(a) Game length various on the number of mines



(b) Guess times various on the number of mines

Figure 5.5: Game length and the number of guesses for solving Minesweeper 9×9 board size various on the number of mines, 2000 runs each mine. (a) is the scatter diagram of the average game length, and (b) is the scatter diagram of the guess times for solving Minesweeper 9×9 board size.

game-theoretic determination and consideration of the legal positions were the features that differentiate the three-level of solvability of the two-player zero-sum games. Based on such understanding, this thesis develops the definition of a single-agent game's solvability based on such a foundation.

In the context of the solvability definition of two-player zero-sum games, solving the game corresponds to finding a solution via a solution tree using tree search methods (such as AND/OR tree search or equivalent). In this thesis, the concept of solving games is extended, which focuses on stochastic single-agent games. To define stochastic single-agent games, the deterministic single-agent games must first be established from the solvability perspective. As such, the solvability of a deterministic single-agent game can be defined as finding a solution with 100% probability for any initial position. Furthermore, as discussed by [47], a deterministic single-agent game is solvable when a solution can be statistically determined with simple (class A) or time/state-dependent (class B) constraints.

A survey conducted on the different single-agent games had described its solvability with features such as random, and hidden information is called a stochastic game [7] [47]. For example, Minesweeper and Tetris are stochastic single-agent games, while N-puzzle is a deterministic single-agent game. As puzzle games evolved, stochastic single-agent games are NP-hard problems such as Threes, and 2048 puzzle [88]. For a stochastic puzzle, the solvability of such a puzzle is defined as finding a solution with less than 100% probability but also being more than 0%.

Let p be the probability of a win (or winning rate) of a single-agent game. Thus, a stochastic single-agent game is considered a 'puzzle' if and only if $0 < p < 1$ in finding a solution for any initial position. Otherwise, it is regarded as a 'game.' These two distinction is established as the possible border between puzzle-solving and game-playing. In this thesis, puzzle-solving is regarded as exhibiting more than 0% possibility of finding a solution for any initial position of a single-agent game. Meanwhile, game-playing is regarded as exhibiting no certainty of a solution to be found, as shown in (5.5).

$$p = \begin{cases} 1, & \text{deterministic puzzle} \\ 0 < p < 1, & \text{stochastic puzzle} \\ 0, & \text{game} \end{cases} \tag{5.5}$$

A single-agent AI solver of Minesweeper was conducted in 9×9 board for 144,000 runs, where the mines are $1 \leq M \leq 72$. Figure 5.6 shows the winning rate $p = 1$ when $M \leq 3$; thus, indicating that such a situation corresponds to a deterministic puzzle. Relative to the definition given by [13], such a situation indicates a strongly solved instance since the strategy adopted by the AI agent can determine the game-theoretic value for all positions. As the number of mines increases, the winning rate decreases for $M \in [4, 39]$, which corresponds to a weakly solved situation. In other words, the AI agent can determine at least some game-theoretic value for an initial position since some uncertainty occurred that makes certain positions in the game unsolvable. When $M \in [40, 66]$, $p = 0$ that a solution is unavailable, a puzzle becomes a game. Finally, $p \to 1$ for $M \in [67, 72]$ as available information is sufficient to determine a solution (or determine game-theoretic game) after opening only several cells (similar situation as a strongly solved instance).

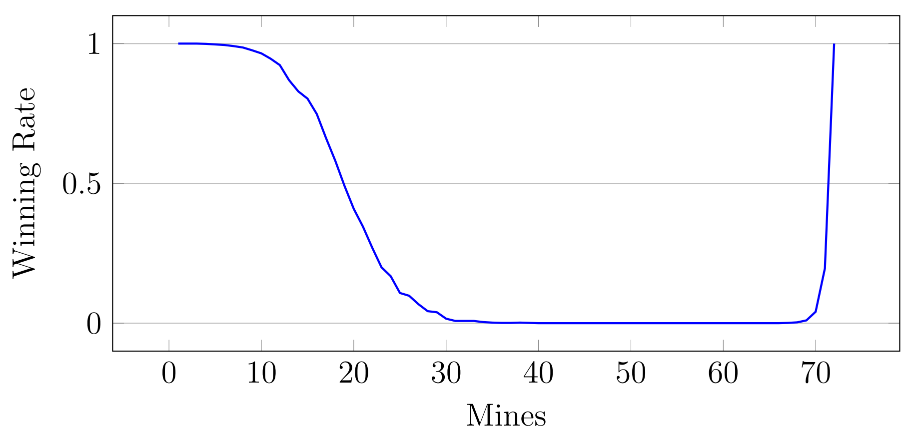


Figure 5.6: The winning rate (p) of the 9×9 board size Minesweeper is based on the number of mines $M \in [1, 72]$, with 2000 runs per mine. The winning rate $p = 1$ for $M \leq 3$ indicates a deterministic puzzle; $0<p \leq 1$ for $M \in [4, 39]$ and $M \in [67, 71]$ indicates a stochastic puzzle; otherwise, $p = 0$ for $M \in [40, 66]$ indicates a game.

The experiment conducted on the 9×9 Minesweeper, which 37.5% (54,000 instances) of the total instances are unsolvable. Note that such instances may vary depending on different factors; nevertheless, they constitute what can be defined as the “uncertainty” element of the single-agent game.

Moreover, 16×16 board size of Minesweeper was used as a testbed for additional verification of the definition of puzzles, where the mines are $1 \leq M \leq 247$. Figure 5.7 shows the experimental results based on 2000 instances per mine. The winning rate $p = 1$ when $M \leq 13$ indicating this situation corresponds to a deterministic puzzle. As the number

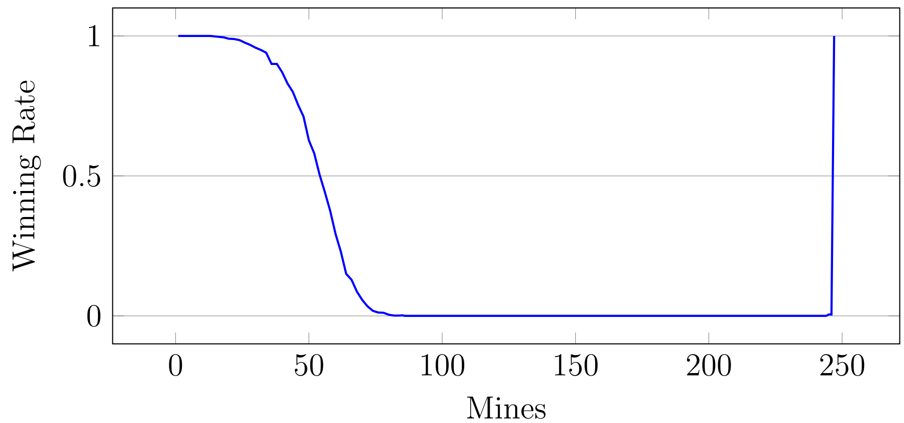


Figure 5.7: The winning rate (p) of the 16×16 board size Minesweeper is based on the number of mines $M \in [1, 247]$, with 2000 runs per mine. The winning rate $p = 1$ for $M \leq 13$ indicates a deterministic puzzle; $0<p \leq 1$ for $M \in [14, 84]$ and $M \in [245, 246]$ indicates a stochastic puzzle; otherwise, $p = 0$ for $M \in [86, 244]$ indicates a game.

of mines increases, the winning rate decreases for $M \in [14, 84]$ and $[245, 246]$, which corresponds to a weakly solved situation with $0<p \leq 1$ with some uncertainty occurred that makes certain positions in the game unsolvable. When $M \in [40, 66]$, $p = 0$ that a solution is unavailable, a puzzle becomes a game. The overall trend of the win rate for the 16×16 Minesweeper is similar to that of the 9×9. Both are deterministic puzzles when the number of mines is small, and as the number of mines increases, the uncertainty in the puzzle begins to increase until it increases to the point where only a few clicks are needed to solve the entire puzzle. This experiment verifies to some extent the positive certainty of our proposed solvability based puzzle definition, and demonstrates that puzzles and games can be transformed into each other according to changing conditions, pointing to a new direction for the boundary between puzzles and games.

Meanwhile, stochastic puzzle-like such as Minesweeper provides several exciting perspectives in the thesis. Firstly, observation of different settings and ability levels provide insights on the original or standard Minesweeper 16×16 and 16×30 board sizes being suited for experienced players, while the 9×9 board size is suited for the beginner; thus, corroborating the previous findings of similar nature [89]. It is curious as to how the designer in the first place decided such setting. On the one hand, the potential reason is that the original purpose of Minesweeper was the mastery of using the mouse peripheral; thus, being educational. On the other hand, the mastery of mouse peripheral also requires continuous effort; thus, it requires the Minesweeper with enough challenge to be played even by experienced players. Hence, the design allowed for "leeways" for different player abilities to match the difficulty expected from the play experience while catering to as many different play audiences as possible. Nevertheless, Minesweeper provides a platform to explore open problems, albeit in a limited sense.

5.5 Chapter Conclusion

Solvability is particularly important in the field of puzzle solving, which is about having a definite solution to the puzzle. If the winning rate of a puzzle is equal to 1, it means that the puzzle is deterministic and always solvable. If the winning rate is between 0 and 1, it means that there is uncertain information in the puzzle solving process leading to an unsolvable puzzle, which we call weakly solvable. If the winning rate is equal to 0, then the puzzle becomes a game, which indicates that the puzzle is full of uncertainty information without a definite solution to solve it.

Meanwhile, N-puzzle and Minesweeper are used as experimental platforms to analyze the informational progression during the puzzle solving process. N-puzzle as a deterministic puzzle which implies a specific solution that allows N-puzzle to move from the initial state to the ordered state of the target. The goal of the puzzle is to solve the puzzle in as few steps as possible, so the choice of each step in the solution process is related to the optimal solution of the puzzle. If each step is chosen from the optimal solution, what is obtained together is the overall optimal solution to the puzzle. In general, for puzzles like N-puzzle, there is no uncertainty of finding a solution, only uncertainty of finding the optimal solution.

For Minesweeper, the size of the board and the number of mines are related to the uncertainty information of the puzzle. From 9×9 and 16×16 Minesweeper, the uncertainty in the solving process increases with the number of mines until the winning rate is 0, and the puzzle becomes a game full of uncertainty. This means that there is no way to know the output of the puzzle, and there are stochastic factors in the puzzle solving process that cause the puzzle to be unsolvable, leading the researcher to find the border between the puzzle and the game.

From the solved games domain, various two-player zero-sum perfect information games have been classified into ultra-weakly solved, weakly solved, and strongly solved [13]. For example, ultra-weakly solved means the game can be solved but does not require the player to achieve the optimal solution. Weakly solved is that player needs to achieve a draw in a game, and strongly solved requires an optimal strategy in all legal positions to achieve the optimal solution. It can be remarked that most solved games were equivalent to puzzles, which were associated with v_0. Meanwhile, weakly-solved and ultra weakly-solved were associated with $v_0 \leq v \leq v_1$ and $v_1 \leq v \leq v_2$, respectively. However, such associations were based on the reported results, and the information during the puzzle-solving progress varies in players' solving experience, which requires further experimentation and justification. Therefore, it is interesting to compare different solving levels in a single-agent puzzle or puzzle-like domain and establish relations to the two-player game solving in future studies.

Chapter 6

Conclusion

6.1 General Conclusion

Solving puzzles helps better understand the process's informational progression and explore entertaining analysis deeper. This thesis used the A* algorithm to find the optimal solutions for N-puzzle and analyze heuristic searches in the artificial intelligence (AI) context. Regarding solvability, N-puzzles are solvable (deterministic) puzzles because there exist specific solutions to solve the puzzle, implying no uncertain information that makes the puzzle unsolvable. However, the uncertainty in solving a deterministic puzzle lies in finding the optimal solution to the puzzle. A single-agent puzzle, namely 8-puzzle, does the observation of the uncertain information obtained from the optimal solution. It concluded that uncertainty of the optimal solution in each step affects the ability of the A* algorithm to find the optimal solution in all steps. This relates to the probability of the player finding the optimal solution to a deterministic puzzle, which, if achieved at each step of the solution process, will lead the player to the optimal solution to the entire puzzle.

The search methods and algorithms are crucial for obtaining the optimal solution to a deterministic puzzle. However, for stochastic puzzles, where the winning rate is in favor of obtaining more information during the solving process, the balance between the information obtained and the search for hidden information is an important topic. Besides, the research on uncertainty analysis of solving stochastic puzzles relies on a popular single-player puzzle known as Minesweeper. The uncertainty of solving it lies in stochastic factors that make the puzzle unsolvable in the solving progress, which depends on the number of mines and the size of the board. An AI solver was proposed based on the obtained information on the board, called the 'PAFG' strategy, which stands for the primary reasoning, the advanced reasoning, the first action strategy, and the guessing strategy. More importantly, the first step is crucial to the winning rate when solving a stochastic puzzle. A safe first step with more information will probably increase the probability of a higher winning rate. The first action strategy in this thesis aims to find the best position of the first opened cell and acquire more information that determines the safety of the following cells. In other words, choosing the best initial position can significantly improve the winning rate.

Game refinement theory and the concept of motion in mind have been verified by kinds of sports games and board games of the sophisticated zone and entertaining analysis. The experimental results of the 8-puzzle imply that it would be enjoyable for professional or skillful players to find the optimal solution. Moreover, different game lengths would provide comfortable game sophistication zones for players with different abilities. The informational progress model in puzzle-solving is aligned with the velocity in the measure of Motion in Mind. Applying the plausible candidates (b) over all possible moves (B) values in the puzzle as the velocity (solving rate), corresponding to motion in mind measure, enable the solving experience of simulated players. Although for players, the gaming experience lies in the uncertainty about the uncertain information of game output. For deterministic puzzles, the uncertainty in solving progress comes from finding the optimal solution, as for the N-puzzle. This solving experience depends on the player's ability and the searching exploration of the puzzle, resulting in various solving experiences for the same puzzle. Beginners are drawn to solving the puzzle, while higher-level players are fascinated by finding the optimal solution to the puzzle.

Furthermore, applying the winning rate (p) as the velocity value for a puzzle with stochastic information and motion in mind measure allows the simulated players to solve the puzzle and play game experiences to be engaged. When the value of $p = 1$, the player is allowed to have the experience of solving a puzzle, this situation indicates a deterministic puzzle. When the win rate is $0<p<1$, the player is allowed to have the experience of solving a puzzle while experiencing uncertainty, which indicates a stochastic puzzle. When $p = 0$, the player is allowed to have the whole game experience to solve the puzzle. Results from these experiments leading the link between the border between puzzles and games.

Overall, the future works in this research will focus on a deeper understanding of the mechanics of puzzles by developing AI solvers and generating the proposed solving strategy for the stochastic puzzle field. Moreover, improving puzzle mechanics for players with different abilities, such as puzzle difficulty and hints of solution strategies. Further investigation of the established link between puzzles and games would also be another promising prospect for future studies.

6.2 Answers to Research Questions

Research objectives laid out at the beginning of this thesis as follows: (1) To develop some AI solvers for solving some target puzzles (N-puzzle, Minesweeper, etc.). (2) To establish puzzle categories from the perspective of solvability and find the border between puzzles and games. (3) To explore the entertainment analysis of solving puzzle filed with the game refinement theory and motion in mind measure, discover the characteristics of each puzzle from the perspective of information dynamic in solving process, and reveal the internal laws behind player's behavior.

The results achieved for each problems are as follows:

- An AI system has been developed based on the A* algorithm, a well-known search algorithm. It automatically generates the N-puzzle's initial state and solves the puzzle with the optimal solution (least steps). It is observed that players with different abilities enjoy different steps of the puzzle, which relies on the game refinement theory and motion in mind measure. Such a situation may reflect the fact that length (steps) reflects the difficulty of a deterministic puzzle. Moreover, an AI solver was proposed based on the obtained information of Minesweeper, the solving strategy called the 'PAFG,' which stands for the primary reasoning, the advanced reasoning, the first action strategy, and the guessing strategy. The experimental simulation of various configurations yielded a high-level winning rate of 96.4%, 86.3%, and 45.6% for the $9 \times 9|10$, $16 \times 16|40$, and $16 \times 30|99$ Minesweeper board configurations.

- The research in this thesis on finding the border between puzzles and games from solvability was based on the puzzle with hidden information, namely Minesweeper (p). The configurations of 9×9 board various on mines $\in [1, 72]$ was used as a testbed to illustrate the dynamics of the winning rate on the basis of the solvability definition of two-player zero-sum games, where $p = 1$ corresponds to a deterministic puzzle, $0 < p < 1$ indicates a stochastic puzzle, and $p = 0$ implies a game. Besides, the 16×16 board various mines $\in [1, 247]$ was verified as a supplementary experiment simulation.

- To discover the characteristics of puzzles from the perspective of information dynamics in solving puzzle process, the informational progress model in puzzle-solving was first applied to the puzzle-solving field to analyze the uncertainty of finding an optimal solution and solving a puzzle with stochastic elements. Then, to explore the field of puzzle-solving for entertainment research, game refinement theory combined with the motion in mind measure was used to analyze the impact of different difficulty puzzles on players. The uncertainty of finding the optimal solution allows players of different abilities to enjoy finding a solution to a deterministic puzzle. At the same time, hidden information is a crucial factor that impacts the winning rate of stochastic puzzles and allows players to engage.

Bibliography

[1] Hiroyuki Iida and Mohd Nor Akmal Khalid. Using games to study law of motions in mind. *IEEE Access*, 8:138701–138709, 2020.

[2] Adrienne Shaw. What is video game culture? cultural studies and game studies. *Games and culture*, 5(4):403–424, 2010.

[3] Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

[4] Mark R Johnson. Casual games before casual games: Historicizing paper puzzle games in an era of digital play. *Games and Culture*, 14(2):119–138, 2019.

[5] Yuexian Gao, Chang Liu, Naying Gao, Mohd Nor Akmal Khalid, and Hiroyuki Iida. Nature of arcade games. *Entertainment Computing*, 41:100469, 2022.

[6] Juris Hartmanis. Computers and intractability: a guide to the theory of np-completeness (michael r. garey and david s. johnson). *Siam Review*, 24(1):90, 1982.

[7] Graham Kendall, Andrew Parkes, and Kristian Spoerer. A survey of np-complete puzzles. *ICGA Journal*, 31(1):13–34, 2008.

[8] Owen WS Huang, Hercy NH Cheng, and Tak-Wai Chan. Number jigsaw puzzle: A mathematical puzzle game for facilitating players' problem-solving strategies. In *2007 First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL'07)*, pages 130–134. IEEE, 2007.

[9] R Rahim, R Dijaya, MT Multazam, A Daengs GS, and D Sudrajat. Puzzle game solving with breadth first search algorithm. In *Journal of Physics: Conference Series*, volume 1402, page 066040. IOP Publishing, 2019.

[10] Rhyd Lewis. Metaheuristics can solve sudoku puzzles. *Journal of heuristics*, 13(4):387–401, 2007.

[11] Feng-Hui Yao and Gui-Feng Shao. A shape and image merging technique to solve jigsaw puzzles. *Pattern Recognition Letters*, 24(12):1819–1835, 2003.

[12] Richard E Korf, Michael Reid, and Stefan Edelkamp. Time complexity of iterative-deepening-a*. *Artificial Intelligence*, 129(1-2):199–218, 2001.

[13] Louis Victor Allis et al. *Searching for solutions in games and artificial intelligence.* Ponsen & Looijen Wageningen, 1994.

[14] Lev Semenovich Vygotsky and Michael Cole. *Mind in society: Development of higher psychological processes*. Harvard university press, 1978.

[15] Isabela Granic, Adam Lobel, and Rutger CME Engels. The benefits of playing video games. *American psychologist*, 69(1):66, 2014.

[16] Carmen V Russoniello, Kevin O'brien, and Jennifer M Parks. Eeg, hrv and psychological correlates while playing bejeweled ii: A randomized controlled study. *Annual review of cybertherapy and telemedicine*, 7(1):189–192, 2009.

[17] Jerry Slocum. Tangram: The world's first puzzle craze, 2003.

[18] Jerry Slocum and Dic Sonneveld. The 15 puzzle: how it drove the world crazy. *The puzzle that started the craze of*, 1880.

[19] Richard E Korf. Finding optimal solutions to rubik's cube using pattern databases. In *AAAI/IAAI*, pages 700–705, 1997.

[20] Alexey Pajitnov and Roger Dean. History of tetris. accessed on June 22, 2021.

[21] Luciano Guala, Stefano Leucci, and Emanuele Natale. Bejeweled, candy crush and other match-three games are (np-) hard. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8. IEEE, 2014.

[22] Blai Bonet and Hector Geffner. Belief tracking for planning with sensing: Width, complexity and approximations. *Journal of Artificial Intelligence Research*, 50:923–970, 2014.

[23] Kenneth M Bayer, Josh Snyder, and Berthe Y Choueiry. An interactive constraint-based approach to minesweeper. In *AAAI*, pages 1933–1934, 2006.

[24] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

[25] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[26] Richard E Korf and Peter Schultze. Large-scale parallel breadth-first search. In *AAAI*, volume 5, pages 1380–1385, 2005.

[27] Richard Kaye. Minesweeper is np-complete. *The Mathematical Intelligencer*, 22(2):9–15, 2000.

[28] Allan Scott, Ulrike Stege, and Iris Van Rooij. Minesweeper may not be np-complete but is hard nonetheless. *The Mathematical Intelligencer*, 33(4):5–17, 2011.

[29] Hiroyuki Iida, Kazutoshi Takahara, Jun Nagashima, Yoichiro Kajihara, and Tsuyoshi Hashimoto. An application of game-refinement theory to mah jong. In *International Conference on Entertainment Computing*, pages 333–338. Springer, 2004.

[30] Arie Pratama Sutiono, Ayu Purwarianti, and Hiroyuki Iida. A mathematical model of game refinement. In *International Conference on Intelligent Technologies for Interactive Entertainment*, pages 148–151. Springer, 2014.

[31] Mohd Nor Akmal Khalid and Hiroyuki Iida. Objectivity and subjectivity in games: Understanding engagement and addiction mechanism. *IEEE Access*, 9:65187–65205, 2021.

[32] James Doran. *An Approach to Automatic Problem Solving*, volume 1, pages 105–124. 01 1967.

[33] TS Hsu, SC Hsu, JC Che, YT Chiang, BN Chen, YC Liu, HJ Chang, SC Tsai, TY Lin, and GY Fan. Computers and classical board games: An introduction, 2017.

[34] Shuo Xiong, Long Zuo, and Hiroyuki Iida. Quantifying engagement of electronic sports game. *Advances in Social and Behavioral Sciences*, 5:37–42, 2014.

[35] Sakshi Agarwal, Rinel Ram, and Hiroyuki Iida. Measuring force of game information in the brain: Linking game refinement theory and neuroscience. In *2016 8th International Conference on Knowledge and Smart Technology (KST)*, pages 291–294. IEEE, 2016.

[36] John Gaschig. Performance measurement and analysis of certain search algorithms. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1979.

[37] Alexander Reinefeld. Complete solution of the eight-puzzle and the bene t of node ordering in ida*. In *International Joint Conference on Artificial Intelligence*, pages 248–253, 1993.

[38] Rok Piltaver, Mitja Luštrek, and Matjaž Gams. The pathology of heuristic search in the 8-puzzle. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(1):65–94, 2012.

[39] Kevin Igwe, Nelishia Pillay, and Christopher Rae. Solving the 8-puzzle problem using genetic programming. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, pages 64–67, 2013.

[40] Anca-Elena Iordan. A comparative study of three heuristic functions used to solve the 8-puzzle. *Journal of Advances in Mathematics and Computer Science*, pages 1–18, 2016.

[41] Hiroyuki Iida. Where is a line between work and play? 2018.

[42] Wu Yicong, Htun Pa Pa Aung, Mohd Nor Akmal Khalid, and Hiroyuki Iida. Evolution of games towards the discovery of noble uncertainty. In *2019 International Conference on Advanced Information Technologies (ICAIT)*, pages 72–77. IEEE, 2019.

[43] Hayato Kita, Cincotti Alessandro, and Hiroyuki Iida. Theoretical value prediction in game-playing. *SIG-GI, IPSJ (GI)*, 87:71–77, 2005.

[44] Hiroyuki Iida. On games and fairness. In *12th Game Programming Workshop in Japan*, pages 17–22, 2007.

[45] Athanasios Kehagias and Georgios Konstantinidis. Some game-theoretic remarks on two-player generalized cops and robbers games. *Dynamic Games and Applications*, pages 1–18, 2021.

[46] Cameron Browne. The nature of puzzles. *Game & Puzzle Design*, 1(1):23–34, 2015.

[47] Mohammad Sina Kiarostami, Mohammadreza Daneshvaramoli, Saleh Khalaj Monfared, Aku Visuri, Helia Karisani, Simo Hosio, Hamed Khashehchi, Ehsan Futuhi, Dara Rahmati, and Saeid Gorgin. On using monte-carlo tree search to solve puzzles. In *2021 7th International Conference on Computer Technology Applications*, pages 18–26, 2021.

[48] Anggina Primanita, Mohd Nor Akmal Khalid, and Hiroyuki Iida. Computing games: Bridging the gap between search and entertainment. *IEEE Access*, PP:1–1, 05 2021.

[49] Wu Tingting, Huang Shunqi, Htung Pa Pa Aung, Mohd Nor Akmal Khalid, and Hiroyuki Iida. Analysis of single-agent game: Case study using minesweeper. In *2020 International Conference on Advanced Information Technologies (ICAIT)*, pages 105–110. IEEE, 2020.

[50] Chris Studholme. Minesweeper as a constraint satisfaction problem. *Unpublished project report*, 2000.

[51] Kasper Pedersen. The complexity of minesweeper and strategies for game playing. *Project report, univ. Warwick*, 2004.

[52] Marc Legendre, Kévin Hollard, Olivier Buffet, and Alain Dutech. *MineSweeper: Where to probe?* PhD thesis, INRIA, 2012.

[53] Shahar Golan. Minesweeper strategy for one mine. *Applied Mathematics and Computation*, 232:292–302, 2014.

[54] Jinzheng Tu, Tianhong Li, Shiteng Chen, Chong Zu, and Zhaoquan Gu. Exploring efficient strategies for minesweeper. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[55] Igor Lordeiro, Diego Haddad, and Douglas Cardoso. Multi-armed bandits for minesweeper: Profiting from exploration-exploitation synergy. *IEEE Transactions on Games*, PP:1–1, 05 2021.

[56] Andrew Adamatzky. How cellular automaton plays minesweeper. *Applied mathematics and computation*, 85(2-3):127–137, 1997.

[57] Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[58] John Von Neumann. The general and logical theory of automata. 1951.

[59] David J Becerra. *Algorithmic approaches to playing minesweeper*. PhD thesis, 2015.

[60] G Peters and James Hardy Wilkinson. On the stability of gauss-jordan elimination with pivoting. *Communications of the ACM*, 18(1):20–24, 1975.

[61] Chris Crawford. The art of computer game design. 1984.

[62] Raph Koster. *Theory of fun for game design.* " O'Reilly Media, Inc.", 2013.

[63] Conor Linehan, George Bellord, Ben Kirman, Zachary H Morford, and Bryan Roche. Learning curves: analysing pace and challenge in four successful puzzle games. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*, pages 181–190, 2014.

[64] Luke Arnott. Unraveling braid: Puzzle games and storytelling in the imperative mood. *Bulletin of Science, Technology & Society*, 32(6):433–440, 2012.

[65] Greg Costkyan. I have no words & i must design: toward a critical vocabulary for games. In *Proceedings of the computer games and digital cultures conference, Finland*, pages 9–33, 2002.

[66] Alan G Sanfey, James K Rilling, Jessica A Aronson, Leigh E Nystrom, and Jonathan D Cohen. The neural basis of economic decision-making in the ultimatum game. *Science*, 300(5626):1755–1758, 2003.

[67] Taichi Ishitobi, Aske Plaat, Hiroyuki Iida, and Jaap van den Herik. Reducing the seesaw effect with deep proof-number search. In *Advances in Computer Games*, pages 185–197. Springer, 2015.

[68] Anggina Primanita and IIDA Hiroyuki. Nature of probability-based proof number search. In *Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*, pages 485–491. Atlantis Press, 2020.

[69] H Jaap Van Den Herik, Jos WHM Uiterwijk, and Jack Van Rijswijck. Games solved: Now and in the future. *Artificial Intelligence*, 134(1-2):277–311, 2002.

[70] Harpreet Kaur. Algorithms for solving the rubik's cube: A study of how to solve the rubik's cube using two famous approaches: The thistlewaite's algorithm and ida* algorithm., 2015.

[71] Deepika Mohan, Baruch Fischhoff, Derek C Angus, Matthew R Rosengart, David J Wallace, Donald M Yealy, Coreen Farris, Chung-Chou H Chang, Samantha Kerti, and Amber E Barnato. Serious games may improve physician heuristics in trauma triage. *Proceedings of the National Academy of Sciences*, 115(37):9204–9209, 2018.

[72] Mable B Kinzie and Dolly RD Joseph. Gender differences in game activity preferences of middle school children: implications for educational game design. *Educational Technology Research and Development*, 56(5):643–663, 2008.

[73] Mohammad Shaker, Mhd Hasan Sarhan, Ola Al Naameh, Noor Shaker, and Julian Togelius. Automatic generation and analysis of physics-based puzzle games. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, pages 1–8. IEEE, 2013.

[74] Ivenulut Rizki Diaz, Ahmad Supianto, and Herman Tolle. Log data analysis of player behavior in tangram puzzle learning game. 2018.

[75] Emel ÇİLİNGİR ALTINER. Relationship between spatial thinking and puzzle games of elementary school students. *International Online Journal of Educational Sciences*, 10(1), 2018.

[76] Xina Jiang, Casper Harteveld, Xinyuan Huang, and Anthony YH Fung. The computational puzzle design framework: A design guide for games teaching computational thinking. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–11, 2019.

[77] Marco Scirea. Adaptive puzzle generation for computational thinking. In *International Conference on Human-Computer Interaction*, pages 471–485. Springer, 2020.

[78] Jordan BL Smith, Jun Kato, Satoru Fukayama, Graham Percival, and Masataka Goto. The crosssong puzzle: Developing a logic puzzle for musical thinking. *Journal of New Music Research*, 46(3):213–228, 2017.

[79] Monica Visani Scozzi, Ioanna Iacovides, and Conor Linehan. A mixed method approach for evaluating and improving the design of learning in puzzle games. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pages 217–228, 2017.

[80] Truong-Huy D Nguyen, Magy Seif El-Nasr, and Alessandro Canossa. Glyph: Visualization tool for understanding problem solving strategies in puzzle games. In *Foundations of Digital Games 2015*, 2015.

[81] Eric Butler, Emina Torlak, and Zoran Popović. Synthesizing interpretable strategies for solving puzzle games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pages 1–10, 2017.

[82] Aaron Isaksen, Drew Wallace, Adam Finkelstein, and Andy Nealen. Simulating strategy and dexterity for puzzle games. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 142–149. IEEE, 2017.

[83] C Shawn Green and Daphne Bavelier. Action video game modifies visual selective attention. *Nature*, 423(6939):534–537, 2003.

[84] Walter R Boot, Arthur F Kramer, Daniel J Simons, Monica Fabiani, and Gabriele Gratton. The effects of video game playing on attention, memory, and executive control. *Acta psychologica*, 129(3):387–398, 2008.

[85] Haruka Kurono and Yu Shibuya. Difference between expert's and novice's gaze point in playing competitive puzzle game. In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pages 149–152. IEEE, 2019.

[86] Erik D Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. In *International Symposium on Mathematical Foundations of Computer Science*, pages 18–33. Springer, 2001.

[87] Hiroyuki Iida, Nobuo Takeshita, and Jin Yoshimura. A metric for entertainment of boardgames: its implication for evolution of chess variants. In *Entertainment Computing*, pages 65–72. Springer, 2003.

[88] Stefan Langerman and Yushi Uno. Threes!, fives, 1024!, and 2048 are hard. *Theoretical Computer Science*, 748:17–27, 2018.

[89] Wu Tingting, Huang Shunqi, Htung Pa Pa Aung, Mohd Nor Akmal Khalid, and Hiroyuki Iida. Analysis of single-agent game: Case study using minesweeper. In *2020 International Conference on Advanced Information Technologies (ICAIT)*, pages 105–110, 2020.

Publications

Journal Papers

1. Liu, C., Huang, S., Naying, G., Khalid, M. N. A., and Iida, H. (2022). A solver of single-agent stochastic puzzle: A case study with Minesweeper. Knowledge-Based Systems, 2022, 246: 108630.

2. Gao Y., Liu, C., Naying, G., Khalid, M. N. A., and Iida, H., Nature of arcade games, Entertainment Computing, 2022, 41.

International Conferences (Proceedings)

1. Liu, C., Huang, S., Khalid, M. N. A., and Iida, H., Attractiveness of Single-Agent Game: Case Study Using Sliding Puzzle, the 2020 International Conference on Advanced Information Technologies (ICAIT) Online, Japan.

2. Yicong, W., Liu, C., Khalid, M. N. A., and Iida, H., Informational Analysis of Go, Part 2: Zuozi and Huanqitou, the 2020 International Conference on Advanced Information Technologies (ICAIT) Online, Japan.

3. Gao, Y., Liu, C., Gao, N., Khalid, M. N. A., and Iida, H., The Entertainment Appeal of Rhythm Games, the 5th International Multi-Conference on Artificial Intelligence Technology (MCAIT 2021) Online, Japan.

4. Liu, C., Rizani, M. N., Khalid, M. N. A., and Iida, H., Entertainment Analysis of Single-Agent Game: Case Study in Match-3 puzzle Game, the Artificial Intelligence and Entertainment Science Workshop (AIES 2021) Online, Japan.

Presentations at Academic Conferences (Oral)

1. Liu, C., Huang, S., Khalid, M. N. A., and Iida, H., (2020, May). Attractiveness of Single-Agent Game: Case Study Using Sliding Puzzle, presented at the 2020 International Conference on Advanced Information Technologies (ICAIT) Online, Japan.

2. Yicong, W., Liu, C., Khalid, M. N. A., and Iida, H., (2020, May). Informational Analysis of Go, Part 2: Zuozi and Huanqitou, presented at the 2020 International Conference on Advanced Information Technologies (ICAIT) Online, Japan.

3. Gao, Y., Liu, C., Gao, N., Khalid, M. N. A., and Iida, H., (2020, Aug). The Entertainment Appeal of Rhythm Games, presented at the 5th International Multi-Conference on Artificial Intelligence Technology (MCAIT 2021) Online, Japan.

4. Liu, C., Rizani, M. N., Khalid, M. N. A., and Iida, H., (2021, Nov). Entertainment Analysis of Single-Agent Game: Case Study in Match-3 puzzle Game, presented at the Artificial Intelligence and Entertainment Science Workshop (AIES 2021) Online, Japan.

Other

- JAIST Research Grants 2021. Analysis of Evolutionary Changes of Puzzle Games.