IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 12, 2021, accepted May 4, 2021, date of publication May 11, 2021, date of current version May 20, 2021.
Digital Object Identifier 10.1109/ACCESS.2021.3079356

Computing Games: Bridging the Gap Between Search and Entertainment

ANGGINA PRIMANITA[1], MOHD NOR AKMAL KHALID[2], (Member, IEEE), AND HIROYUKI IIDA[2]
[1]Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology, Nomi 923-1292, Japan
[2]Research Center for Entertainment Science, Japan Advanced Institute of Science and Technology, Nomi 923-1292, Japan
Corresponding author: Anggina Primanita (s1820431@jaist.ac.jp)

This work was supported by a grant from the Japan Society for the Promotion of Science, in the framework of the Grant-in-Aid for Challenging Exploratory Research under Grant 19K22893.

ABSTRACT Over history, games have served multiple purposes. It serves as a fun activity for players who need the entertainment to become test-beds for artificial intelligence. Solving games is beneficial in providing a better understanding of how information is progressing throughout the game. Uncertainty in games affects the way a game is solved and the way the game is experienced. Previous works have interpreted uncertainty in the game progress through various means, but there have been no clear links among those interpretations. In this paper, the probability-based proof number search (PPNS) and single conspiracy number (SCN) were used as the domain-independent indicators to analyze how uncertainty affects various game elements. PPNS exploits information from certain and uncertain information to reach convergence in solving games. Meanwhile, SCN evaluates the game states' difficulty and describes game-playing patterns to understand play positions better. The study's objective focuses on finding the optimal difficulty ordering of a game solver, defining the indicator for entertainment, and linking game-tree search and entertainment in different game environments. Experiments results demonstrate the link between the search indicators and the measure of entertainment where uncertainty plays a vital role in both contexts, verified from both two-person and single-agent games. Such a situation is also crucial for both computation and entertainment measures since it impacts both the quality of information and the expected game-playing experience.

INDEX TERMS Search algorithm, game tree, entertainment, single agent game, puzzle game, single conspiracy number.

I. INTRODUCTION

Game is one of the most sought-after activities for a human as a medium of entertainment. Currently, the game-playing culture has been proven as a place of comfort when human is not allowed to enjoy physical, social activity [1]. The essence of play that included the elements of fun [2], and purpose [3] had been coincided with the transition of human wisdom and sensibilities for several millennia. With the advancement of technology and communication systems availability (such as the internet, personal computers, and fast computing resources), design and research in games had been accelerated at a rapid pace alongside the domain of artificial intelligence (AI) [4].

Many problems encountered in games reflect real-world problems, which made the game's uses much more impactful [5]. Based on such notions, research in games had branched into many sub-fields, encompassing many disciplines. Among the most studied is game theory and evolutionary games. Game theory involves mathematically model conflicts and cooperation between intelligent, rational decision-makers [6], [7]. Such a theory was widely pervaded in economic theory while being widely adopted to better understand sociology, political, and biological phenomena. Meanwhile, the evolutionary game is the adaptation of the traditional game theory, which expanded the strategic interactions to a large population of agents driven by natural selection (population-wise) or by myopic decision rules (individual-wise) [8]. The evolutionary game had been applied to a variety of fields, such as transportation science, computer science, sociology [9], environmental science [10], ecology [11], and even epidemiology [12].

In regards to game studies in general, the notion of gamification and ludification had been gained attention relative to educational studies [13], cultural anthropology and

The associate editor coordinating the review of this manuscript and approving it for publication was Rupak Kharel.

philosophy [14], as well as game studies' influences on societal culture [15]. These studies emphasize that game-playing and game designs' behavioral aspects coincide with some effects of societal interests, such as learning, thinking, decision-making, and interactions within the digital spheres. Moreover, purposeful design in games had been increasingly being considered since it has been proposed by von Anh back in 2006 [16] and had a close relation to the concept of serious games [13]. The concept of game with purpose leads to crowd-sourced computation that leverages vast resources of human interactions with computers over the Internet. Meanwhile, the concept of serious games adopted game design elements in non-game contexts (i.e., healthcare [17]). Purposeful design in games had been seen as a successful adaptation of game-related concepts into automating data labeling, and categorization [16], intention-based design [18], and social-based and location-based data tagging [19].

The domain of games remains a fascinating ground for AI development and test-bed purposes. The main reason for games being used as AI test-beds is because it is cheap, deterministic, easily repeatable and controllable, as well as an entertaining environment [20]. In recent years, AI continues making breakthroughs in games through Deep Blue in chess [21], Chinook in checkers [22], AlphaGo in Go [23], and AlphaZero in chess again [24]. These rapid AI developments in games are highly affected by (end)game solvers in a game. A game is considered solved when its outcome can be predicted from any state, assuming that both players play perfectly. The concept of solving games is usually applied to full-information games without any chance element. The idea of solving games is a part of achieving the general goal of creating a good game-playing program [25].

The goal of solving games is to find the game's theoretical value. However, it is also proven beneficial in providing a better understanding of how a game works [26]. This condition ultimately leads to a better understanding of how information is progressing throughout the game. Uncertainty in games does affect how a game is solved and the way the game is experienced. During game-playing, a game started in a state of uncertainty where there was no information about the winner of the game [27]. As the game progresses, more information is obtained. Throughout this study, the notion of "uncertainty" is defined as the amount of information currently unknown and required to be traversed throughout the processes (or state spaces) in determining (or reaching) the win state (or target goal) of the game.

Currently, the game progress can be observed using two different approaches. The first approach is by observing its game progress pattern using a search indicator such as the Probability-based Proof Number Search (PPNS) [28] and the Single Conspiracy Number (SCN) [29]. The second approach is by analyzing and observing the game information progress model, called the motion in mind [27], [30]. Although the two approaches exploited information differently, both measure the game progress, where the link between the two approaches is investigated.

The aim of adopting PPNS and the Single Conspiracy Number (SCN) as the main tools for the experimentation of this study is that these search indicators provide better insights into the influence of uncertainty in the games. On the one hand, such insight involves computing the optimal play by the PPNS and the influence of uncertainty in the tree-search framework. On the other hand, the insight provided by the SCN encompasses reasonable interpretation of uncertainty and their expected entertainment values from the games. Hence, uncertainty can be established as the link between the optimal play domain and the game entertainment measure, where its implications in games computation and game-playing experience can be identified. As such, the main aim of this study revolves on the following objectives:

1) To find the optimal difficulty ordering procedure for game solver for different game-tree structures.
2) To define the indicator for entertainment using a game-tree search framework.
3) To define the link between the game-tree search result and entertainment indicator in different game environments.

It is important to note that, in the context of optimization, PPNS and SCN are categorized as recursive, tree-traversal algorithms that enumerate the subset of the solution set (called branches or leaves) from a candidate solution set (called root) through exploring the adversarial state-space search (a game-tree search framework). Hence, PPNS and SCN were deterministic global optimization/exact strategies, specialized in a game-tree search that provided the measures of decision complexity [31] and decision difficulty [29], [32], respectively, in identifying the optimal (or winning) game state.

The paper is organized as follows. Section II reviews some of the important works related to PPNS and SCN relative to other state-of-the-art in the domain of game tree search. Then, Section III provides an overview of the PPNS application in the domains of two-person and single-agent games. Then, Section IV provides an overview of the SCN and the current related findings for two-person and single-agent games. The link between optimal play and game entertainment is established in Section V. The findings and discussion from the perspective of entertainment and information science were provided in Section VI. Finally, Section VII concludes the paper.

II. DOMAIN-INDEPENDENT INDICATORS

A. CONSPIRACY NUMBER

Conspiracy number search (CNS) [33] is a MIN/MAX tree search algorithm that attempts to guarantee the accuracy of the MIN/MAX value of a root node. The likelihood of the root taking a particular value is reflected in that value's associated conspiracy number. The conspiracy number is the minimum number of leaf nodes in the tree that must change their score (by searching deeper) to result in the root taking on that new value [34].

CNS was proposed to design a strong computer player but suffers from a low search efficiency because of its slow convergence and the high cost of computing conspiracy numbers. However, CNS provides a promising concept for measuring stability using its conspiracy numbers. In the current context, the measure of "stability" refers to the ability of the conspiracy number to indicate the root node convergence towards stable states within the game-tree framework. Some variants inspired by CNS have been proposed as game-independent heuristics. The most successful among them is the proof number search (see Section II-B).

Meanwhile, another study of CNS rooted on improving the performance of the CNS, where Lorenz *et al.* [35] expanded the necessary steps of CNS, where the selection is made by assigning demands (called en targets) on the nodes of a game subtree in a top-down fashion, and a set of leaves is selected in a single action. Then, Lorenz [36] proposed controlled conspiracy-2 search (CC2S) to improve root stability against single faulty leaf evaluation, which applied in the Chess domain.

A different approach toward conspiracy numbers since previous methods only use conspiracy number of a single (as in the original CNS [33]) or two (as in $\alpha\beta$-conspiracy search [37]) evaluation values to determine the direction of the search. Then, conspiracy number was adopted to identify the critical positions in simple games (Tic-tac-toe and Othello) to apply speculative play [31], [38]. Another study adopted conspiracy number for improving the move selection was conducted by Vu *et al.* [39], where conspiracy number in a game situation is regarded as the probability distribution of evaluating a situation (i.e., high conspiracy number indicates the difficulty of achieving and vice versa).

An experiment using game transcripts from the United States 2015 National Open tournament of Othello professional games, where a novel move selection policy was proposed that treated conspiracy number as a whole; thus, enables a better understanding of a game situation and determine a better decision [39]. Finally, Pawlewicz and Hayward [40] developed a global heuristic evaluation that reliably scores relative strengths of node siblings via siblings comparison evaluation function (SCEF) and incorporating algorithm-wide enhancements, such as Rapid Action Value Estimate (RAVE) statistics [41], transposition tables, and paralleling.

B. PROOF NUMBER

Proof number search [42] is one of the most efficient algorithms for solving games and complex endgame positions inspired by the concept of conspiracy numbers. Proof number search focuses on the AND/OR tree and tries to establish the game-theoretical value in a best-first manner. In proof number search, each node has a proof number and a disproof number (DN). The proof number (DN) represents the scale of difficulty of proving (disproving) a node. The proof number search process involves expanding the most-proving node, which is the most efficient node for proving (disproving) the root.

Compared with conspiracy number search, proof number search is more successful in practical use. Such consideration reduces the relevant numbers to two, the proof number and the disproof number, thereby improving the search efficiency. Nagai [43] proposed a depth-first proof number (df-pn) search by adopting iterative deepening, which improves the original proof number search to utilize less computational resources by lessening the expansion on the interior node and reducing the number of proof number and DN that have to be recomputed. A df-pn enhancement via the $1 + \varepsilon$ method reduces the search's tendency to jump around the tree [44], where it was found to be more robust than proof number search in its application to Atari Go and Lines of Action. A scalable parallel version of df-pn search (SPDFPN) had also been proposed by Pawlewicz and Hayward [45] based on the serial version of enhanced df-pn [44], where SPDFPN solved all previously intractable 9×9 Hex opening moves and the first to solve the 10×10 Hex opening move.

The proof number and DN are highly instrumental in solving games when proving (disproving) a search-tree position. Such indicators had been leveraged alongside MCTS, where an improved game solver's quality is expected. An early example of such a combination is the Monte-Carlo proof number search (MCPNS) [46], a best-first search designed to work in an AND/OR game-tree. Employing MIN/SUM rules to backpropagate information from the simulation, node information was accessed based on its Monte-Carlo which was infused with proof number and DN, called the Monte-Carlo proof number (*pmc*) and the Monte-Carlo disproof number (*dmc*). Similarly, an algorithm called Product Propagation (PP) was proposed by Saffidine and Cazenave [47] that combines the idea of proof number search, and probabilistic reasoning to solve three perfect information game test-beds (Y, Domineering, and NoGo) where it had outperformed other proof number search variants.

The most recent improvements to the df-pn are the DFPN-E introduced by Kishimoto *et al.* [48], where an added edge cost initialization to the df-pn had been proposed. DFPN-E was applied to the chemical synthesis problem with an unbalanced search space where the tree is identified to be lopsided, and the DFPN-E is proven to be successful where it traversed the least number of nodes [48].

C. MONTE-CARLO TREE SEARCH (MCTS)

The Monte-Carlo tree search (MCTS) is a best-first search algorithm that has been commonly employed to solve games. First proposed by Coulom in 2006 [49], the MCTS framework consists of four steps: (1) selection, (2) expansion, (3) simulation, and (4) backpropagation. The algorithm starts with selecting the next action based on a stored value (selection). When it encounters a state that cannot be found in the tree, it expands the node (expansion). The node expansion is based on multiple, randomly simulated games (simulation or playout). The value is then stored and backpropagated to the tree's root, where the algorithm continues to repeat the steps until the desired outcome is reached (backpropagation).

MCTS utilizes simulations to gain information from unexplored nodes, and this was expanded by introducing upper confidence bound applied to trees (UCT) by Kocsis and Szepesvári [50]. Although it has been proven that MCTS converges to minimax when evaluating the available moves [51], MCTS converges only in so-called ''Monte Carlo Perfect'' games [52].

A new best-first search algorithm derived from MCTS and UCT was then introduced by Chaslot *et al.* [53]. The algorithm's goal was to overcome the difficulty of building heuristic knowledge for a non-terminal game state by employing stochastic simulations. Determining a game-playing strategy is useful when using multiple simulations. Its usage has become well known, especially in Go, and in part led to the AlphaGo's wins against top grandmasters [23].

III. CREATING AN EFFICIENT SOLVER VIA PROBABILITY-BASED PROOF NUMBER SEARCH

Intuition and expertise are the main essences of game-playing, which are also crucial in decision-making. Understanding the game intricacies had spawned a popular study known as game-solving. The purpose of such a study is to estimate the possible game outcome by analyzing the game information. In this study, the proposed probability-based proof number search (PPNS) algorithm is adopted for such a purpose.

PPNS is a best-first search algorithm that is set to be used in an AND/OR game tree structure. The proof number and disproof number are highly instrumental when the branching factor varies, giving distinguishable information to indicate the shortest path of proving or disproving a node [28]. However, for games with a balanced tree (almost fixed depth and an almost fixed branching factor), the proof number search is relatively weak because the proof numbers and disproof numbers are too similar to give distinguishable information.

Instead, PPNS uses an indicator called a probability-based proof number to indicate a node's probability leading to the expected position. The idea originates from the concept ''searching with probabilities'' [54], where the core idea involves that proving a node is computed from the probabilities of proving its children while following the AND/OR rules of probability events. The probability-based proof number specifies the probability of a node (terminal nodes, leaf nodes, or internal nodes) to be proven in an AND/OR tree [28]. The probability-based proof number assigned to an OR and AND internal node is the product of its child values. As such, the probability-based proof number of a node contains two different pieces of information derived from the current game-tree. All of these nodes have their probability-based proof number (n_{ppn}) value that is calculated as follows:

- If n is a terminal leaf node, then

$$n_{ppn} = \begin{cases} 1 & \text{if } n = \text{winning node;} \\ 0 & \text{if } n \neq \text{winning node;} \end{cases} \tag{1}$$

- If a node n is a leaf node, then let R be the winning rate as a result of game playout, and θ is a small positive number close to 0, then

$$n_{ppn} = \begin{cases} R & \text{if } R \in (0, 1); \\ R - \theta & \text{if } R = 1; \\ R + \theta & \text{if } R = 0; \end{cases} \tag{2}$$

- If a node n is an internal node, then
  - if n is an OR node,

$$n_{ppn} = 1 - \prod_{n_c \in \text{ children of } n} (1 - n_c) \tag{3}$$

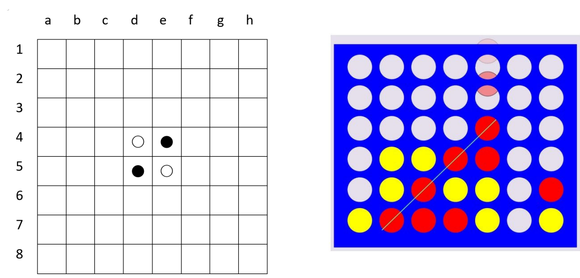
  - if n is an AND node,

$$n_{ppn} = \prod_{n_c \in \text{ children of } n} n_c \tag{4}$$

For experiments conducted in this section, a specialized program is written in C++ programming language. The experiment was conducted on a computer with an Intel i5-8400 processor running at 2.81 GHz using 8 GB of RAM, running Windows 10, on a 64-bit machine.

A. TWO-PERSON GAME

In the two player game domain, solving game means being able to predict the outcome when both players conducted a perfect play. In this case, the solver that can give the most prediction is indicated by its ability to converge in the root node. In the domain of two-person game, Connect Four and Othello (Figure 1) were adopted as the test bed [55].



(a) Initial position of 8 × 8 Othello board

(b) Illustration of Connect Four with a 7×6 board (adopted from [56])

FIGURE 1. Illustration of (a) the standard 8 × 8 Othello board and (b) the standard 7 × 6 Connect Four board. Both games are distinctive in the sense where Othello could flip the discs (board pieces) to any side based on the "capture" rule, and the game is won by having more discs at the end of the game. Meanwhile, the Connect Four game mechanics utilizes gravity to play and can be characterized as a *k*-in-a-row family of games where the game is won when either side achieved 4-in-a-row pieces in any direction.

PPNS performance had been investigated in P-game tree (each move is randomly assigned a chosen value in a MIN/MAX tree [50]) and compared with PNS, MCPNS [46], UCT (MCTS solver equipped with the Upper Confidence Bounds applied to Trees), and pure MCTS solver [57]. Two experiments were conducted on 400 P-game tree, where the first 200 having two branches with 20 layers, while the second involves the remaining 200 having eight branches and eight layers [28]. The study found that PPNS outperformed others, while on average, taking less time and fewer iterations to prove or disprove a game tree, converges faster to the correct solution compared to PNS, MCPNS, and the pure MCTS solver, and the error rate of selected moves decreases more smoothly as the number of iterations increases.

Recent application of PPNS was conducted on Connect Four, a perfect information k-in-a-row game where the game is won by lining up four chips in either horizontal, vertical, or diagonal[1]). Gravity is an essential element of play in the game where pieces fall as far to the bottom as possible, and the game can be won between 13-ply to 42-ply (the board is filled), implying that the game search tree highly varied in the number of depth and the game-tree structure is unbalanced. The experiment with 200 generated Connect Four positions (each position contains 12-ply of randomly generated moves) had found that using real-numbers causes unnecessary prolonging of the search procedure of PPNS [58]. The results demonstrated that PPNS with a pr value reduces the amount of explored nodes needed to solve up to 57% (Figure 2), implying that even a small amount of explored nodes allowed information from an unexplored area to be exploited and combined to reach the desired goal.

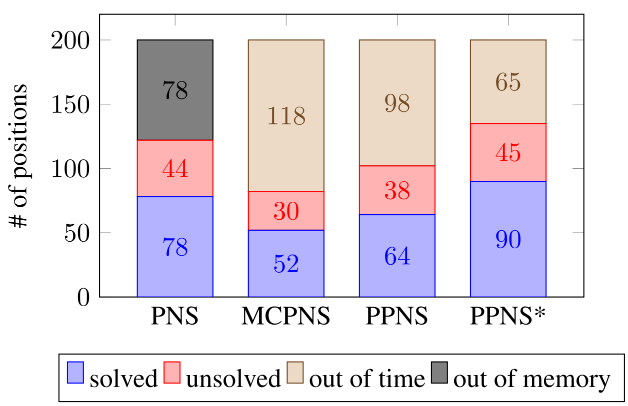


FIGURE 2. Number of 200 generated Connect Four positions solved, unsolved, and out of bounds (memory and time) by PNS, MCPNS, and PPNS. The application of the *pr* parameter in PPNS (denoted as PPNS*) increases the total positions that are concluded (solved or unsolved) where the optimal configuration of $\theta = 0.001$ and $pr = 0.001$ with a total of 135 positions was found [55].

A further experiment was conducted on Othello,[2] which is a board game that played on an 8×8 board where two disks of each color (black and white) are initially placed diagonally in the center of the board [55]. Othello has a balanced game-tree structure (required to play the game until the board is full) with the state-space size of Othello is approximately 10^{28} [59]. An experiment on 600 Othello positions was performed for three randomly generated moves (18, 26, or 32 moves of the game) which are given in Table 1. The experiment was conducted considering the precision rate ($pr = 0.00001$) parameter, which found that PPNS performs the best by solving most positions in every stage of the game (72% positions on average).

TABLE 1. Experimental result of different algorithms applied to 600 Othello positions (200 for randomly generated moves of 18, 26, or 32) [55].

Moves	Algorithm	Solved	Unsolved	Out	Completion*
18	PPNS	52	56	92	54.00%
	PNS	12	1	187	6.50%
	MCPNS	5	4	191	4.50%
26	PPNS	63	88	49	75.50%
	PNS	19	16	165	17.50%
	MCPNS	10	10	180	10.00%
32	PPNS	82	91	27	86.50%
	PNS	39	39	122	39.00%
	MCPNS	6	37	157	21.50%

*Completion = (Solved + Unsolved) / 200; even if the result concludes that it is unsolved, the algorithm is still able to conclude.

Probability-based proof number is an instrumental indicator in PPNS, where its ability to solve games was successfully demonstrated in two-person games. Current insight suggests that probability-based proof number is suited for solving a game that requires a long look-ahead strategy. The probability-based proof number in Connect Four showed that the quality of information is critical, where even a small amount of explored nodes with appropriate statistical information of unexplored nodes can vastly improve the solver's performance. Furthermore, the application of probability-based proof number in Othello affirms that PPNS is an optimal solver for games with a balanced tree structure, where an increased amount of information allowed for more positions to be solved. The probability-based proof number also demonstrates the importance of considering the appropriate "moment" to take advantage of both the explored and unexplored nodes to solve more positions faster and earlier.

B. SINGLE-AGENT GAME

Expanding the PPNS framework to the single-agent game requires revisiting its general idea of such a framework. In a two-person search framework, the OR nodes represent the maximizing agent (select the child with maximum probability-based proof number) while the AND nodes represent the minimizing agent (select the child with minimum probability-based proof number) before back-propagating its result to the root [28]. In the single-agent game realm, while the maximizing agent is the sole player of the game who tries to gain the most upper hand and reach their objective, the minimizing agent can be elusive as there is no clear opposing player that tries to gain the upper hand in the game. Such

[1]Connect four game rules can be found at http://www.ludoteka.com/connect-4.html

[2]Othello game rules are described at https://www.mastersofgames.com/rules/reversi-othello-rules.htm

games' objectives can be in the form of a score, point, or an end position. Single-agent games usually possess a mechanic that is used to hinder the player from reaching their objective. The inherent inhibiting mechanics in single-agent games (Table 2) were meant for the player, which is equivalent to the minimizing agent. Thus, the feature of these inhibitor mechanics was adopted in place of the second player.

TABLE 2. Inhibiting mechanics in the single-agent games.

Game	Mechanics
SameGame	Gravity between blocks [61]
8-puzzle	Limited move direction [62]
Sokoban	Shape of the map [63]
2048	Randomly spawn tiles [64]

The game of 2048 was first published by its creator, Gabriele Cirulli, in 2014 [64]. The main objective of 2048 is to join identical tiles currently available on the board to obtain the highest tile number. The game of Three inspired the first iteration of the game [65] and played in a 4×4 board. Moving a tile in a particular direction pushes it in such a direction until it is stopped by either another tile or the grid's edge. If there are two tiles with the same number next to each other, they will be merged, creating a tile with a number equal to the sum of the two original tiles. A turn is only valid if it causes a change in the board, whether it is moving tiles or merging tiles. If a move that the player chooses does not change the board's state, it is not considered a turn.

In the current context, a 2×2 version board of the game 2048 was adopted as the testbed where the highest available tile number is tile with value 32 (via brute force enumeration [66]) and player mobilization is very limited. The rule of the game's initial position remains the same, in which there are two tiles with a value of 2 placed arbitrarily on the board. The board states on the OR nodes consist of at most four states (move directions), the AND nodes consist of at most six states, and there can be three empty tiles. However, there are two possible tiles, numbered 2 and 4, which doubled the total number of available states. As the game nature is different from the previous two-person counterpart, minor modifications in the playout step were conducted on the PPNS, which does not affect its domain-independent feature.

As 2×2 2048 only has six enumerable opening positions, all the positions were adopted in the experiment disregarding the mirroring positions. Two win objectives were set where the first one being the highest number tiled equal to 16 and the second being the highest number tiled equal to 32. Three different algorithms, PNS, MCPNS, and PPNS, were compared to solve each position independently. The experiment was simulated in 60 moves for both MCPNS and PPNS, while the PPNS utilizes $\theta = 0.01$ and $pr = 0.0001$.

The three algorithms' effectiveness was compared where all of the algorithms could solve all the initial positions of 2×2 2048, where the number of visited nodes and iterations were given in Table 3. Such a result can be categorized as ultra-weakly solved with the game-theoretical value being a win for the player. This situation implies that given no change in rules and handicaps, a player that employs a perfect play strategy would be able to achieve the two objectives for the game. The highest numbered tile equals to 16 and 32, which aligns with the combinatorial enumeration of the game [66]. The end of the game can be reached as early as the highest available tile value of 8 and the highest quality tile equal to 32; thus, making its game-tree highly varied and exhibits the behavior of the unbalanced game-tree. All algorithms were able to converge while traversing less than 1% of all states.

TABLE 3. Average iterations and nodes visited by the PNS, MCPNS, and PPNS applied to 2 × 2 of 2048.

Algorithm	Pos. Solved	Objective	Iteration	Nodes
PNS	12	16	6	15.67
		32	11	32.33
MCPNS	12	16	4	12.33
		32	4.67	15.67
PPNS	12	16	2	7.5
		32	2	8.67

*Pos. solved: all the solved initial position;

The result of the quality measure between the algorithms showed that PPNS took the least resources to converge from the table comparison. Nodes traversed by each of the search algorithms represent the board's state where the combinatorial bounds of game states in 2×2 2048 are computed to be of 537 states inclusive of the six initial positions [66]. When mirroring positions and repeated positions are removed, it is calculated to be 59 possible states with two initial positions [67]. Compared to the nodes traversed to reach convergence, the best-first search algorithms all traversed fewer nodes than the available nodes. This situation shows the efficacy of generalizing the PPNS to the single-agent domain, where it is considered as "vanilla" since it disregards the possibility of enhancements (i.e., transposition table).

IV. MEASURING GAMING EXPERIENCE VIA SINGLE CONSPIRACY NUMBER

The ability to use conspiracy numbers to improve the play experience leads to a hypothesis that these numbers can be used to assess a certain position in a game. One shortcoming of the conspiracy number approach, however, is that there are multiple numbers for each node. To enable its use as an indicator, the SCN was proposed. It incorporates the idea of the conspiracy number, but rather than one node having several associated numbers, those numbers are all merged into one value called the SCN.

The SCN reflects the difficulty of a node obtaining a value of no less than T, where T is a threshold on the legal MIN/MAX values. When T equals the maximum legal MIN/MAX value, the SCN is equivalent to the proof number. When T equals the minimum legal MIN/MAX value, the SCN is zero because there is no difficulty for a node to obtain a value that is no less than the minimum. When T is between the maximum and minimum legal MIN/MAX values, the SCN indicates the difficulty of a position obtaining a score of no less than T.

Compared with the conventional evaluation method using MIN/MAX values, the SCN is more game-independent and is expected to yield more information on game progress patterns while indicating the potential change in the MIN/MAX values. Therefore, the SCN is expected to be a good supplement to evaluation function values for analysing game progress patterns.

Let n_{scn} be the SCN of node n, let m be the MIN/MAX value of node n, and let T be a threshold on the legal MIN/MAX values. Then, the formalism of the SCN is given as follows:

- When n is a terminal node:

$$n_{scn} = \begin{cases} 0 & \text{if } m \geq T; \\ \infty & \text{if } m < T; \end{cases} \tag{5}$$

- When n is a leaf node (not terminal):

$$n_{scn} = \begin{cases} 0 & \text{if } m \geq T; \\ 1 & \text{if } m < T; \end{cases} \tag{6}$$

- When n is an internal node:
  - If n is MAX node, then

$$n_{scn} = \min_{n_c \in \text{child of } n} n_c \tag{7}$$

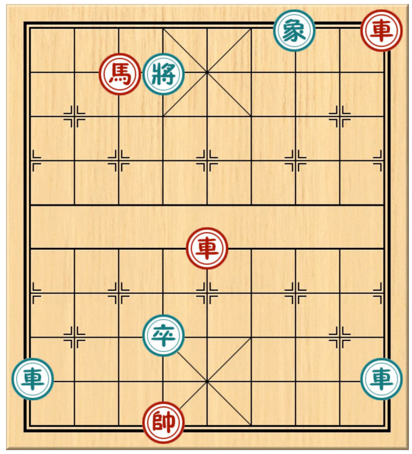
  - If n is MIN node, then

$$n_{scn} = \sum_{n_c \in \text{child of } n} (n_c) \tag{8}$$

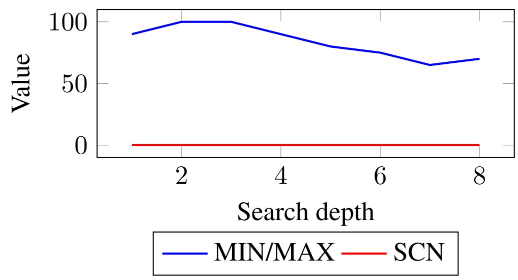
For experiments conducted in this section, a specialized program is written in Python programming language. The experiment was conducted on a computer with an Intel i5-8400 processor running at 2.81 GHz using 8 GB of RAM, running Windows 10, on a 64-bit machine.

A. TWO-PERSON GAME

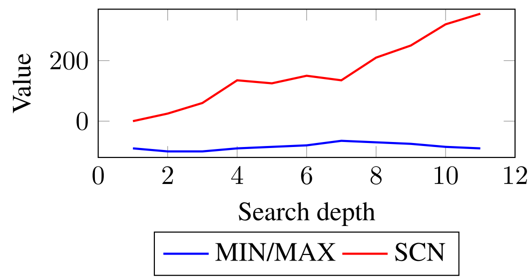
Previous applications of SCN had been applied to abstract strategy two-person board games. For example, Song and Iida [29], [32] have tested the effectiveness of the SCN by examining it in Xiang Qi (or Chinese chess), which is a family of abstract strategy board games similar to (Western) chess and shogi (Japanese chess), through self-play experiments using the ElephantEye (Light) program. In addition to the differences in the rules, piece behavior, and cultural reflections in Xiang Qi compared to chess [68], Xiang Qi is an extremely tactical game that requires the evaluation of positions far ahead into the future. As such, Xiang Qi is the perfect testbed to determine the effectiveness of the SCN in estimating long-term outcomes based on its ability to reflect the game progress pattern [32]. It was found that long-term positions require a precise evaluation technique (see Figure 3), for which the SCN is more consistent and accurate than conventional heuristics (such as MIN/MAX values) [29], [32].



(a) An example of Xiang Qi's tactical position where Red is to move (Red wins)



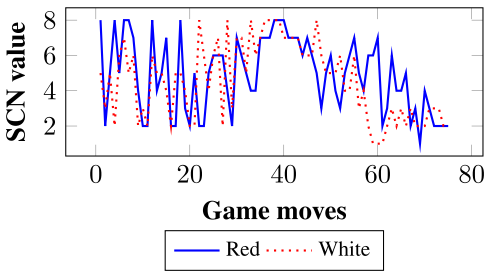
(b) Red's MIN/MAX value and SCN in the initial position with different search depth ($T = 600$)



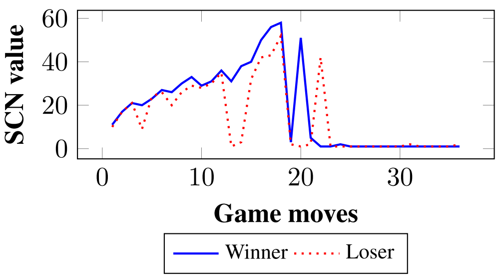
(c) Black's MIN/MAX value and SCN in the first position with different search depth ($T = 600$)

FIGURE 3. Depiction of an example tactical position in Xiang Qi, where the SCN and MIN/MAX values were compared. In such a position, the Red player is considered to have an advantage over the Black player, where its next move leads to better board states based on the high MIN/MAX value; hence, associated with a very low SCN value. Meanwhile, the MIN/MAX value of the Black player does not provide enough information to judge the next moves but better illustrated by the SCN, where the next move to be made is a 'challenging' state correspond to the search depth. Hence, it can be observed that SCN can demonstrate the implication of such a tactical position more effectively.

Then, SCN is further implemented on top of a popular open-source checkers program; namely, Samuel AI. Checkers has an unbalanced tree-search structure [69] and a high decision complexity [22], where the SCN stabilizes for medium-term and short-term positions depending on the relative differences in the SCN values and the tree-search structure. A recurring pattern identified is that a low SCN value ($\rightarrow$ 0) implies a winning position, while a high SCN value ($\rightarrow$ ∞) implies a losing position (see example in Figure 4(a)). Additionally, a constant intermediate SCN value indicates a "loop" or repetitive play.



(a) Players' moves in a draw game in checkers (Samuel AI)



(b) Players' moves in GoBang (based on past tournament data)

FIGURE 4. Analysis of SCN results for two board games: (a) checkers and (b) GoBang. The SCN results of both games indicated the game-playing conditions (favorable or unfavorable) and progress continuity (ending or condition changes), where a situation of repetitive play in checkers and the adversarial struggles of players in GoBang was visualized.

Recently, the game progress in GoBang was analyzed through an experiment conducted on authoritative data [70], called a "chess manual", which were downloaded from a professional GoBang official website (http://game.onegreen.net/Soft/HTML/47233.html). The collected data recorded the game scores from various competitions ranging from the global to the national level, associating the game scores with every player's moves. In this experiment, a GoBang program was implemented to calculate both the MIN/MAX and SCN values, and each move was defined as one play. The SCN value was observed to indicate both the game conditions (favorable or unfavorable) and progress continuity (ending or condition changes) [70]. Additionally, opposing and similar SCN trends imply "defensive" and "neutral" game situations (see example in Figure 4(b)). The selection of a higher threshold (T) value relative to the MIN/MAX value also allows the game condition and situation to be reflected better than a lower T.

SCN is a heuristic-free informative indicator for analyzing game progress patterns that had been successfully achieved in two-person games. It is mainly used to distinguish whether positions are favourable or unfavourable to the player. Additionally, fluctuation patterns indicate a seesaw turnover frequency, which is indicative of a "tough" game between the players. Therefore, for two-person games, the SCN typically indicates the stability of a position in a game. A stable state is indicated by fewer fluctuations (or a high frequency of low values) of the SCN, implying a lower difficulty of reaching a specific position in a game (high certainty). On the other hand, an unstable state is indicated by more fluctuations (or a high frequency of high values) of the SCN, implying more difficulty in reaching a particular game position (high uncertainty). Additionally, a high SCN implies good prospects for a better position and vice versa. Because the SCN is a game-independent indicator, it is expected that the SCN can be extended to a single-agent problem, for which it can potentially be used to find the cause of failure in the game.

B. SINGLE-AGENT GAME

The original version of the game is played on a 4 $\times$ 4 board [64]. The original game's starting position consists of three tiles with the number 2 and a single tile with the number 4 (Figure 5 (left)). Each turn, the player has at most four options: to move the tile to the right, left, up, or down. After a valid turn, a new tile will pop up at a random location on the board (Figure 5 (right)). The new tile can be a number 2 tile or number 4 tile.

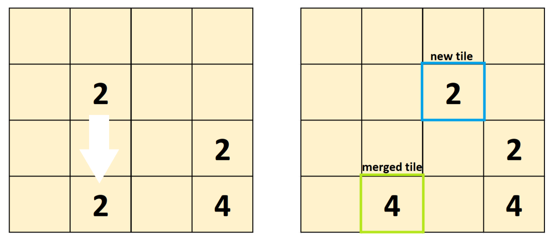


FIGURE 5. Illustration of tile merging from an initial state of the game of 2048, where the player chooses to move "down" (left). Then, a random tile appears in a single turn of the game immediately after the merging of tiles (right).

The game of 2048 is a stochastic game with complete information. The player can observe the entire state of the board at any point in the game. However, the player cannot predict the progress of the entire game from the board's current state because, in every turn, a random tile will appear [71]. The appearance of random tiles gives a unique character to the game, as for each turn, there are only at most four options, but there are a large number of possible board states.

Compared to the previous implementation of the SCN in the minimax framework, the treatment for obtaining the m value is different. In this experiment, the m value is calculated based on the Expectimax values of the nodes. The Expectimax or Expectiminimax algorithm is a variation of the minimax algorithm. It was developed specifically for games that depend not only on player skill but also on random chance [72]. The random factors are represented by special nodes, known as chance nodes. Thus, along with MIN and MAX nodes, there are three kinds of nodes in the Expectimax game tree structure.

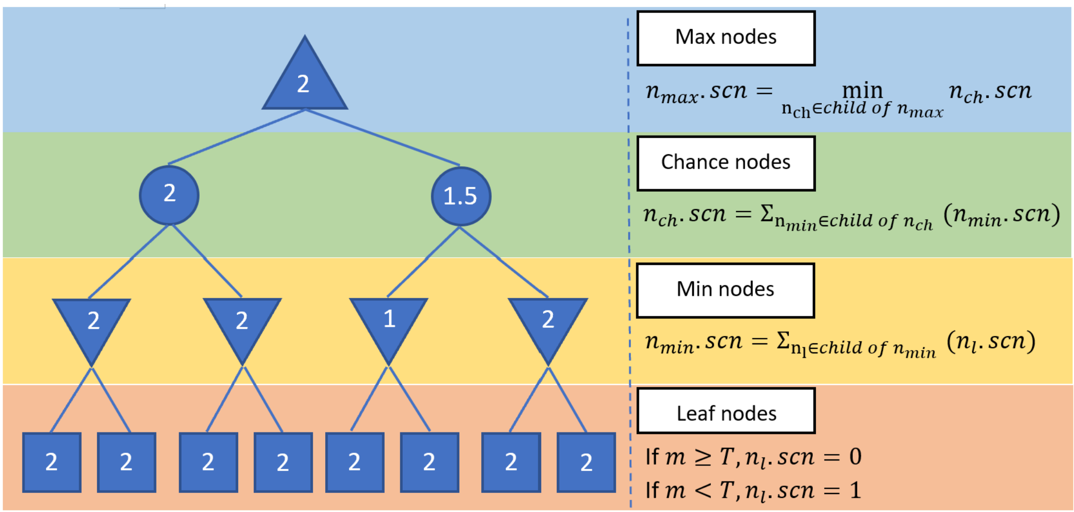


FIGURE 6. Illustration of the SCN calculation in the Expectimax framework. The SCN value is obtained by considering the MIN/MAX value at the leaf or terminal node corresponding to the T value, which then propagated to the root node through the tree structure considering three layers of different node types and their associated propagation rule.

However, the definition still holds, as the m value is a node's inherent minimax value. For every leaf and terminal node in the tree, the Expectimax values of the nodes are calculated based on (9), where $E(s)$ is the evaluation function, w and b is the weight and board indexes, respectively.

$$E(s) = \sum_{i=0}^{3}\sum_{j=0}^{3} w[i][j] \times b[i][j] \tag{9}$$

In a single-agent game, the SCN value is calculated only for MAX and MIN nodes. This is because the chance nodes do not represent the 'difficulty' or 'stability' of a node [31], [38]; rather, they represent the random chance. In this case, the chance nodes are treated the same way as MIN nodes.

Compared to the previously considered board games, it was observed that the T value is vital for observing the game progress and represents an intermediate stage between stable and unstable state [69]. In a single-agent game, T cannot be defined with only one single value; as the game progresses, the m value will continuously increase for the same state difficulty. Thus, a new way to calculate T is proposed by (10). The SCN calculation in 2048 is illustrated in Figure 6.

$$T_{new} = T_{old} * 2^n \tag{10}$$

An experiment was conducted to test the effectiveness of the SCN in representing the game progress for 2048. Every appearance of a new highest-number tile can be thought of as a progress milestone. Thus, every step taken by the player is a step to achieve this milestone. Accordingly, the T value in the 2048 case is changed every time such a milestone is achieved.

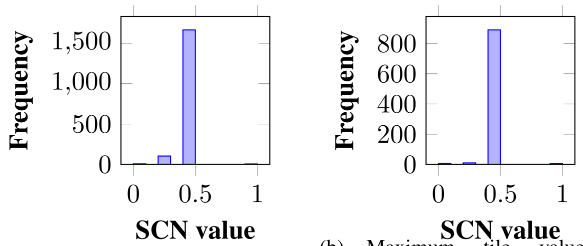
In this experiment, the algorithm was set to play 2048 20 times in 3 different configurations based on the Expectimax search depth (d). The 3 different configurations represent different player capabilities, ranging from a rookie player ($d = 2$) to a casual player ($d = 3$) to an expert player ($d = 4$). For each configuration, the single-agent SCN for every turn was recorded. Implementing the Expectimax algorithm with the three different configurations yields the results displayed in Table 4. It can be observed that the deeper the search depth is (the higher d is), the higher the chance of obtaining higher-number tiles (2048 or greater). This situation also affects the average score and the average number of steps taken to reach the result. Therefore, it can be said that the three configurations representing rookie, casual, and expert players perform as expected.

TABLE 4. Results of Expectimax implementation for the game of 2048.

	Highest Tile				Average	
d	512	1024	2048	4096	Score	Steps taken
2	5	7	7	1	20212.00	1031.00
3	0	9	10	1	25163.00	1244.00
4	2	4	10	4	32866.25	1537.05

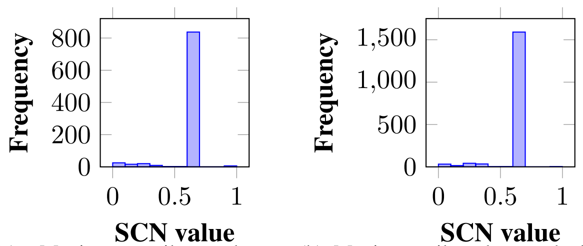
The SCN results for a simulated rookie player ($d = 2$) are shown in Figure 7. Because the T value changes every time a new milestone is reached, the SCN will decrease to a low number when such a transition occurs (high frequency of low SCN values). This situation can be translated to mean that the game is always stable, making the player take a constant gamble to end the game prematurely. In the single-agent game context, the high frequency of low SCN value implies an unstable position. The rookie player does not have sufficient capacity to effectively overcome the stable game state (in other words, "stability trap") player is continually choosing unstable positions. Effectively, this player takes constant gambles to reach the game objective.

For games with a simulated casual player ($d = 3$), each category's game results are shown in Figure 8. For these games, it can be seen that the SCN fluctuates among several



(a) Maximum tile value reached is 1024 (b) Maximum tile value reached is 2048

FIGURE 7. Illustration of games of 2048 with $d = 2$ (SCN values normalized between 0 and 1). A high frequency of middle to low SCN values was observed, which indicates the game is stable (or in a constant state of a gamble between better and worst game state).

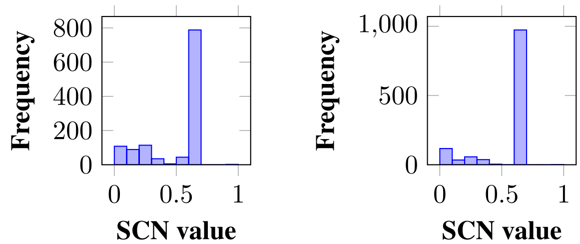


(a) Maximum tile value reached is 1024 (b) Maximum tile value reached is 2048

FIGURE 8. Illustration of games of 2048 with $d = 3$ (SCN values normalized between 0 and 1). A high frequency of high SCN values was observed, which indicates the game continuously transitions between stable and unstable, which implies a speculative game state and the play is longer.

ranges of values, indicating that the player is employing particular strategies in achieving milestones. For example, arranging higher-numbered pieces into one specific area on the board to improve movement flexibility. A player with sufficient play experience will gain the knowledge to traverse both unstable and stable state and retain to play longer. This condition corresponds to a high frequency of high SCN value, indicating that the game is unstable and becomes speculative. However, such a player may still lead to a deadlock, forcing the game to end earlier, indicated by the ratio of the frequency of low SCN values being greater than the frequency of high SCN values.

For games with a simulated expert player ($d = 4$), the results are shown in Figure 9. For these games, the fluctuations in the SCN are the highest (the SCN values show the most significant variety). These fluctuations show that an experienced player progresses through the game while frequently changing stance from stable to unstable positions and vice versa. Similarly, an expert player would choose a more stable position (high frequency of low SCN values). As the SCN value lowered, a less stable position will be selected, which returned into a more unstable state. However, expert players always overcome unstable positions (match high-valued tiles as quickly as possible, resulting in more empty board spaces), leading to more prolonged gameplay and possibly better scores. Nevertheless, the frequency of high SCN values is also the highest, indicating that the game



(a) Maximum tile value reached is 1024 (b) Maximum tile value reached is 2048

FIGURE 9. Illustration of games of 2048 with $d = 4$ (SCN values normalized between 0 and 1). A high frequency of low and high SCN values was observed, which indicates the game was able to be retained more unstable states compared to stable ones, and the game rapidly changes from stable to unstable positions and vice versa, leading to a prolonged play.

remains in such a turnover position until the game cannot continue (is forced to end because the board has filled up to the point where no further move can be made).

V. MOTION IN MIND AND SINGLE CONSPIRACY NUMBER

The game refinement (GR) theory plays an essential role in quantifying game sophistication by determining the rate of solved uncertainty along the game length where the appropriate amount of fairness, excitement, and thrills were empirically quantified [27], [73]. Expanding GR theory further by analogously defining the notion of success rate as velocity (v) and difficulty rate as mass (M), then various "motion in mind" can be measured (Table 5) [30].

TABLE 5. Analogical link between motion in mind and motion in physics [30].

Notation	Physics	Games
y	Displacement	Solved uncertainty
t	Time	Total score or game length
v	Velocity	Winning rate
M	Mass	Winning hardness (m)
g	Gravitational acceleration	Acceleration in mind (a)
$\vec{p}$	Momentum	Momentum in game
U	Potential energy	Energy in game (E_p)

Such analogies have been adopted to measure sophistication of player's entertainment in board games and scoring sports games. It was found that the game's motion in mind is closely related to the cultural aspects of the games' origin (in board games) and game's popularity (in sports games). The relationship between game-playing and rewarding experience had also been established via operant conditioning [74].

SCN featured the stability indicator to determine the difficulty of reaching a state value not less than the threshold (T). Based on the motion in mind concepts from the aspects of entertainment [30], the velocity (v) corresponds to the normalized SCN values, given by (11) and (12). Since v is a vector quantity, it describes not only the progression of

TABLE 6. Results of the application of various average of motion in mind measures in 2048.

d	Highest Tile	v	M	$\vec{p}$	E_p
2	1024	0.32782	0.67218	0.21517	0.14447
2	2048	0.32727	0.67273	0.21383	0.14411
3	1024	0.49763	0.50237	0.24464	0.24881
3	2048	0.45281	0.54719	0.22709	0.22439
4	1024	0.36248	0.63752	0.18039	0.16753
4	2048	0.44654	0.55346	0.21766	0.22072
5	1024	0.38209	0.61791	0.17264	0.18042
5	2048	0.28728	0.71272	0.18343	0.11764

reaching the winning state (or goal) but also the 'direction,' which refers to the two-dimensional invariant property that corresponds to the gain (positive) or loss (negative) of such winning rate.

$$scn_{\text{norm}} = \frac{1}{k}\sum_{n=1}^{k}\frac{n_i - \min(n)}{\max(n) - \min(n)} \tag{11}$$

$$v = scn_{\text{norm}} \tag{12}$$

The applications of motion in mind measures based on SCN were conducted to the 2048 games with Expectimax search depth, $d \in [2, 5]$, where $d = 5$ simulates a master player. Interesting results were observed for the 2048 games, albeit calculated from different depths, achieved quite similar results (Table 6). In average, $v = 0.38549$ ($M = 0.61451$) gives the average $\vec{p} = 0.20686$ and $E_p = 0.18101$. According to Iida and Khalid [30], the game is motivating enough to play because the information expectation is high (high E_p). In addition, the game is considered sophisticated enough where it has a sufficient challenge level fitting the effort made by the player (high $\vec{p}$) except for the rookie player with $d = 2$ by having $M > \frac{1}{2}$. This situation explains the popularity of 2048, as players with different degrees of skills can play the game with equal levels of fairness.

However, such a result does not show the different expertise representation between the skill levels as the SCN suggested due to the changing threshold. Thus, the scn_{norm} data were then observed separately. The separation is made based on the previously defined threshold. Each data point consists of the data from one threshold to the next, depicted as in Figure 10. Based on the momentum and potential energy of player with different skill levels, different game-playing experience can be observed.

For a rookie player, the value is concentrated at a certain point. The player starts their game when the game is considered high-tension ($v \simeq 0.2$ with fair E_p and small $\vec{p}$). The peak momentum (competitive balance) was never reached, which means that their game-playing experience relies on the board's random tiles. Meanwhile, a casual player starts their game in a similar position with a rookie player but later reaches the peak of the game's potential energy and momentum. Such a condition showed that their game-playing experience does not rely only on chance but also on their skill.

The game-playing experience for expert and master players has a distinctive similarity, in which both initially have very low $\vec{p}$ and E_p values (challenging). However, the game becomes more interesting when several milestones were overcome. The game also prolonged and peaked $\vec{p}$ and E_p due to the large gap between milestone changes. An expert player's $\vec{p}$ and E_p were at the cross point, which implies that competitive balance was stroked in most of their game-playing experience and makes the game more exciting. The expert player game ended while they are at the peak of $\vec{p}$ (balanced game-playing experience). Meanwhile, the game ended later for a master player, in which the game is considered much more challenging.

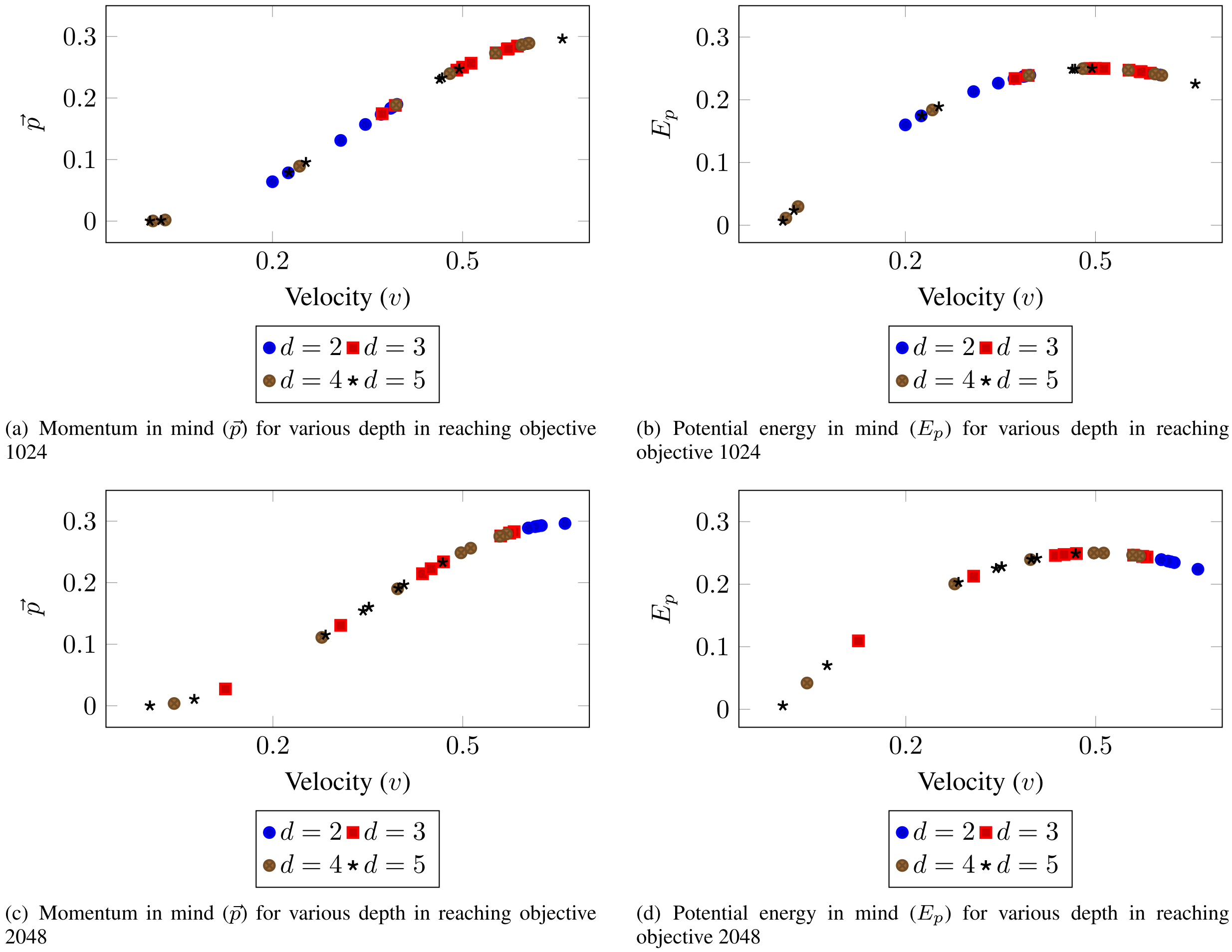
VI. FINDINGS AND DISCUSSIONS

A. INFORMATION QUALITY AND INTUITIVE CAPACITY

Three experiments were conducted on three games with distinctively different game tree structures, namely, Connect Four, Othello, and 2 × 2 2048. These experiments were done to explore and examine the quality of PPNS. The Connect Four game was chosen to represent the real case with an unbalanced game tree structure, while the Othello game was chosen to represent a balanced game tree structure. The last game, 2 × 2 2048, is an extension where the concept of searching in a two-person game tree structure is expanded into a single-agent game tree structure. The results of those experiments are then compared to two related solver algorithms, namely, PNS and MCPNS, to examine the impact of combining uncertainties into getting the game's theoretical value.

The solver framework's general idea was revisited, which results in finding the opponent's representation, making it possible to solve a single-agent game. The possible game-theoretical value, in the end, is similar to that of a two-person game (win, lose, or draw). Comparison results from the experiments show that the PPNS performs better than the other two algorithms. In both two-person and single-player games, it demands less resource yet can converge successfully. It is likely that the result of the difference in information quality is stored in the probability-based proof number. The MPN in PPNS is chosen based on the highest probability-based proof number of the current tree. Experiments with two-person games show that information quality is most critical, where the best solver is the one that utilizes the smallest amount of explored nodes.

The probability-based proof number of a node stores information from both its predecessors and likely successors, relieving the need to explore unnecessary nodes. A human player should make an educated guess to win a single-agent game with random mechanics. Such a condition is reflected by the probability-based proof number, considering certain information from its predecessors and uncertain information from its likely successors similar to a human player using their prior knowledge to make an educated guess. The way the algorithm explores a game tree is similar to that



(a) Momentum in mind ($\vec{p}$) for various depth in reaching objective 1024

(b) Potential energy in mind (E_p) for various depth in reaching objective 1024

(c) Momentum in mind ($\vec{p}$) for various depth in reaching objective 2048

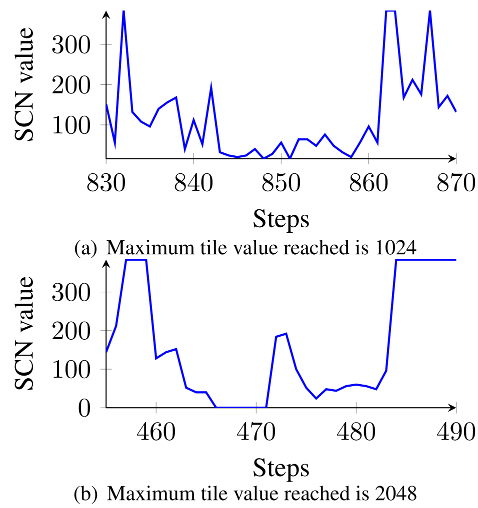
(d) Potential energy in mind (E_p) for various depth in reaching objective 2048

FIGURE 10. **SCN and motion in mind measures for various depths ($d \in [2, 5]$). It can be observed that the player's skill levels (increasing d) are described by the capability to reach the peak momentum and high potential energy, implying excitement and challenge.**

of human intuition. This insight is further emphasized with the implementation of PPNS in a single-agent game. The 2048 game requires creating a higher numbered tile piece and considering the placements of the tile. In the 2×2 board, the player's mobility becomes limited, with only four grids available. It needs a clear look-ahead tile placement strategy for the player to reach the desirable highest tile. PPNS requiring the least resource shows that it is the best-suited solver for the game that requires such intuition.

B. PLAYER EXPERIENCE AND RISK MANAGEMENT

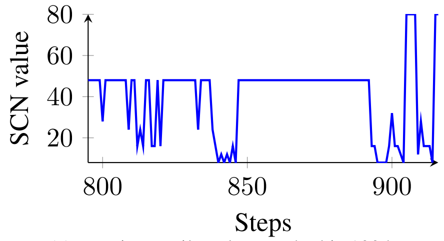
Comparing the results of the simulated casual and expert players reveals the difference in their intuition and decision-making ability. The simulated expert player can escape from multiple "pitfalls" in the game, while the casual player will succumb to them. This situation leads to the "stability trap" in the game of 2048. In the current context, the "stability trap" refers to a position on the board that is considered stable (promising) but does not favour the player (i.e., will cause the game to end early). This trap can be demonstrated when a player chooses a more stable position on the board, constantly creating multiple tiles (with high numbers) and filling the board with untouchable tiles. An expert player would be able to escape this position (Figure 11). However, casual and rookie players have a lower



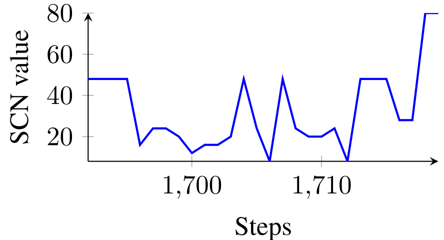
(a) Maximum tile value reached is 1024

(b) Maximum tile value reached is 2048

FIGURE 11. **Depiction of the occurrence of the "stability trap" in the simulated expert player games ($d = 4$). Expert players will be able to escape from trap positions and continue their games, leading to higher scores and longer games.**



(a) Maximum tile value reached is 1024



(b) Maximum tile value reached is 2048

FIGURE 12. Depiction of the occurrence of the "stability trap" in the simulated casual player games ($d = 3$). Casual players have a higher chance of being trapped in unfavourable positions, leading to an early end to the game.

chance of escaping from the trap, which lessens their chance of progressing (Figure 12).

Based on the experimental results and discussions, players with different ability levels can be observed using the proposed variant of the SCN. Using the proposed method, the stability trap was identified, which should be approached differently by the players. An expert player will deliberately approach this position, as such a player has sufficient ability to escape from the trap. However, a player with less knowledge is unable to escape from the trap immediately. Such a situation will cause the game to converge to a position that is typically regarded as unfavourable, causing the game to end prematurely.

The case of trapping positions is not unique to single-agent games such as 2048. The term has been used for years in research on games, related to a problematic position in a two-person game such as chess [75], to refer to a position or state that seems to be promising at first but ends up in a reversal, leading to states that are more favourable to the opponent. Jansen [75] discussed two circumstances that are crucial in recognizing the opponent's strategy (speculative play). One circumstance involves the presence of a strong indication causing a good move to be underestimated, leading the opponent to play poorly (*swindle* position). The other circumstance involves making a bad move to induce the opponent to overestimate the available opportunity (*trap* position).

C. "ROLLER COASTER" IN MIND

A game-theoretic value is obtainable in a two-person game context based on the best possible play by both players. A better result is not possible unless the opponent makes a move that reduces his or her game-theoretic value (i.e., the opponent makes a mistake) [76], [77]. In such a case, exploiting the opponent's mistake would allow for move decisions that lead to better game-theoretic values (to win from a draw, or to draw or win from a loss).

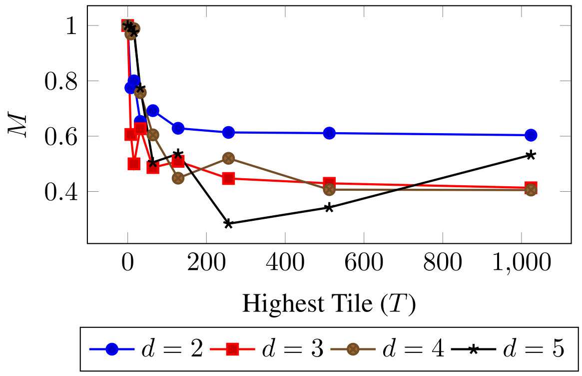
With the extension of the SCN to single-agent games, it displays an informative aspect of the game progress patterns, albeit from a different perspective. Since no apparent "opponent" is present, the SCN reflects the game progress relative to the player's own ability and own chances to progress through the game. A "mistake" here corresponds to the frequency of the stability trap, where a greater frequency of staying in the trap induces the effect of the player choosing positions from which the player overestimates the ability to progress.

Analyzing further from the perspective of the M dynamics is demonstrated by the situation that can be interpreted similarly to a "roller-coaster" experience when a player faces adverse conditions in their game-playing progression (Figure 13). It can be observed that the SCN changeover gap (from one milestone to another) is large in the first few milestones, demonstrating the transitions from uncertainty to various unstable (or unfavorable) states. This situation is similar to the roller coaster ride going down (gain velocity) from the top ramp (significant change of v). Then, it slowed down with some fluctuation (change in v), implying the existence of the acceleration (thrilling experience). This situation implies the game changes from being challenging in the short-term aspect of game-playing (high M) into a more challenging in the long-term aspect of game playing (middle M), which requires some level of strategic planning and knowledge-based play.

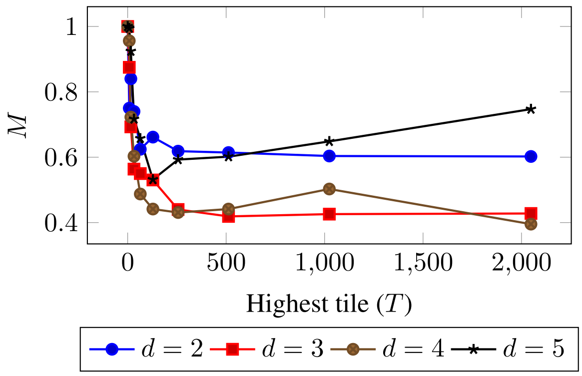
However, the "ride" experience started to be distinctive when the milestone transitioned from $8 \rightarrow 16$, $32 \rightarrow 64$ and $64 \rightarrow 128$, simulated by the different skill levels (or depth, d). This situation can be observed from the fluctuation of the M value where the SCN changeover gap is medium (or low), and the SCN value is high (or middle). It implies that the game state is situated in a favorable position but may become stable or unstable depending on the skill levels (i.e., $d = 2$ and $d = 3$). For $d = 4$ and $d = 5$, the unstable game state was utilized as an advantage to stay and play longer (utilizing appropriate heuristic from more access to information, i.e., applying strategy gained from play experience) instead of trapped in a stable state and end the game.

Table 7 illustrates the player's flow-experience promotion model via the bridging of motion in mind and SCN. A general interpretation of SCN based on the application of motion in mind had found that any game's initial state (S_0) starts with low-value SCN and a medium to high amount of SCN changeover gap.

Novice player tends to let their game state in an unstable position due to apathy, denoted by unchanging high SCN. However, an abrupt change in their position, indicated by the high or medium changeover gap, shows that they are falling into the stability trap. The sudden change could be attributed to the sense of worry while playing the game as they fell into the stability trap. Casual player tends to be situated at



(a) The M dynamics for various depth in reaching objective 1024



(b) The M dynamics for various depth in reaching objective 2048

FIGURE 13. ***M* dynamics of 2048 games for various depth ($d \in [2, 5]$) where the fluctuations of *M* is greater when *d* is higher.**

TABLE 7. **Generalized interpretation of the SCN relative to positions in a single-agent game. A low changeover gap of high (or mid) SCN values indicates the best outcome, while a high (mid) changeover gap of high (low) SCN values indicates chances to move forwards.**

		SCN Changeover Gap		
		Low	**Medium**	**High**
SCN	**Low**	Trap	Favourable	Favourable
	Middle	Uncertain	Uncertain	Uncertain
	High	Favourable	Trap⋆	Trap⋆

⋆ : thrilling;

the high SCN with a high (or medium) SCN changeover gap due to anxiety or arousal. However, an expert player would be situated at average low SCN with medium (or low) SCN changeover gap due to flow or control, whereas a master player would be situated at low SCN.

However, most players would experience a middle SCN value where its uncertainty is most significant (low SCN changeover gap), where the probability of success or failure to enter or leave the state of stability trap is uncertain. However, the middle to high SCN changeover gap requires observing the magnitude of change, where rapid increase means a favorable condition that usually happens at the end of the stability trap or rapidly decreases, which implies entering the stability trap.

VII. CONCLUSION

This study explores the idea of search "indicator," whose idea was originated from the scalar versions of the original conspiracy number search (CNS) framework [33]. However, expanding from such an idea leads to the various domain-independent indicators, which are useful in the search process of the AI adopted in the domain of games. It was found that such domain-independent indicators were an effective tool for solving and understanding search-tree computation that has strong relations with the informational uncertainty of the game states. Such condition generates two related but distinct veins of research: solving and understanding games.

In another direction, the inspiration of conspiracy numbers was reflected in SCN for understanding games based on the stability change of the root node via the SCN. Adopting SCN as the change of stability (equivalent to velocity in mind, v), dynamics of its changeover (v-dynamics) is essential for engagement from a long-term perspective (i.e., repeated play). In contrast, a considerable change of v (i.e., acceleration) is crucial for the sense of thrills in the short-term (in the 2048 game, random tile appearance caused such dynamics). Having visual and empirical evidence of such dynamics can better understand entertainment moments from a search process (i.e., a roller coaster in mind).

In essence, PPNS and SCN, both inspired from conspiracy number as domain-independent indicators, provided two essential directions on game-related computing, which relies on the uncertainty of root stability in the framework of the search-tree. On one end, PPNS leverage known information and unknown information via probability, representing appraisal of information quality and uncertainty that can simulate the notion of intuition. On the other end, SCN is a functional measure of game-playing expertise and risk management by providing the possibility of identifying risk (i.e., of losing or failure) and appraise judgment as early as possible.

The implications of both directions are crucial for preserving valuable resources and saving precious time while potentially suited for addressing high-stakes decision-making and long-term planning to optimize values and minimize risks. In similar directions, it is interesting to investigate using a family of AI algorithms for global optimizations, such as evolutionary computation,[3] where similar effects can be achieved, particularly in determining the optimal solution in long-term planning and risks minimization in decision-making.

Future research efforts should also emphasize establishing search and entertainment in a multidisciplinary perspective to bridge information and computation in conjunction with psychological study and affective sciences. Other interesting direction worth venturing includes verifying further the feasibility of PPNS and SCN in multiplayer games and non-game

[3]A comprehensive review on the families of global optimization algorithms can be found in [78]

contexts (i.e., education and business application) and exploring the impact of the incomplete-information game.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable feedback and suggestions. They would also like to thank Jaap van den Herik for his invaluable comments on the development of the probability-based proof number search (PPNS).

REFERENCES

[1] N. Dyer-Witheford and G. de Peuter, "Postscript: Gaming while empire burns," *Games Culture*, vol. 16, no. 3, Sep. 2020, Art. no. 1555412020954998.

[2] J. Huizinga, "Homo ludens: A study of the play element in culture," *Leisure Ethics*, vol. 168, p. 237, 1955.

[3] K. Masukawa, "The origins of board games and ancient game boards," in *Simulation and Gaming in the Network Society*. Singapore: Springer, 2016, pp. 3–11.

[4] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*, vol. 2. Cham, Switzerland: Springer, 2018.

[5] M. Świechowski, "Game AI competitions: Motivation for the imitation game-playing competition," in *Proc. 15th Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2020, pp. 155–160.

[6] M. O. J. Neumann Von, *Theory of Games and Economic Behavior*. Princeton, NJ, USA: Princeton Univ. Prress, 1944.

[7] R. B. Myerson, *Game Theory*. Cambridge, MA, USA: Harvard Univ. Press, 2013.

[8] W. H. Sandholm, "Evolutionary game theory," in *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*. New York, NY, USA: Springer, 2020, pp. 573–608.

[9] M. Askarizadeh, B. T. Ladani, and M. H. Manshaei, "An evolutionary game model for analysis of rumor propagation and control in social networks," *Phys. A, Stat. Mech. Appl.*, vol. 523, pp. 21–39, Jun. 2019.

[10] Y. Yang and W. Yang, "Does whistleblowing work for air pollution control in China? A study based on three-party evolutionary game model under incomplete information," *Sustainability*, vol. 11, no. 2, p. 324, Jan. 2019.

[11] M. Zhang and H. Li, "New evolutionary game model of the regional governance of haze pollution in China," *Appl. Math. Model.*, vol. 63, pp. 577–590, Nov. 2018.

[12] M. Alam, K. M. A. Kabir, and J. Tanimoto, "Based on mathematical epidemiology and evolutionary game theory, which is more effective: Quarantine or isolation policy?" *J. Stat. Mech., Theory Exp.*, vol. 2020, no. 3, Mar. 2020, Art. no. 033502.

[13] S. De Freitas, "Are games effective learning tools? A review of educational games," *J. Educ. Technol. Soc.*, vol. 21, no. 2, pp. 74–84, 2018.

[14] A. Dippel and S. Fizek, "Ludification of culture: The significance of play and games in everyday practices of the digital era," in *Digitisation: Theories and Concepts for the Empirical Cultural Analysis*. Evanston, IL, USA: Routledge, 2017, pp. 276–292.

[15] R. Y. Bayeck, "Exploring video games and learning in south africa: An integrative review," *Educ. Technol. Res. Develop.*, vol. 68, no. 5, pp. 2775–2795, Oct. 2020.

[16] L. V. Ahn, "Games with a purpose," *Computer*, vol. 39, no. 6, pp. 92–94, Jun. 2006.

[17] E. Brox, L. Fernandez-Luque, and T. Tøllefsen, "Healthy gaming–video game design to promote health," *Appl. Clin. Informat.*, vol. 2, no. 2, p. 128, 2011.

[18] K. Mitgutsch and N. Alvarado, "Purposeful by design?: A serious game design assessment framework," in *Proc. Int. Conf. Found. Digit. Games (FDG)*, 2012, pp. 121–128.

[19] I. Celino, D. Cerizza, S. Contessa, M. Corubolo, D. DellAglio, E. D. Valle, and S. Fumeo, "Urbanopoly—A social and location-based game with a purpose to crowdsource your urban data," in *Proc. Int. Conf. Privacy, Secur., Risk Trust Int. Conf. Social Comput.*, Sep. 2012, pp. 910–913.

[20] S. Risi and M. Preuss, "From chess and atari to StarCraft and beyond: How game AI is driving the world of AI," *KI–Künstliche Intelligenz*, vol. 34, no. 1, pp. 7–17, Mar. 2020.

[21] M. Campbell, A. J. Hoane, Jr., and F.-H. Hsu, "Deep blue," *Artif. Intell.*, vol. 134, nos. 1–2, pp. 57–83, Jan. 2002.

[22] J. Schaeffer, N. Burch, Y. Björnsson, A. Kishimoto, M. Müller, R. Lake, P. Lu, and S. Sutphen, "Checkers is solved," *Science*, vol. 317, no. 5844, pp. 1518–1522, Sep. 2007.

[23] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, and S. Dieleman, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[24] M. Sadler and N. Regan, *Game Changer: AlphaZero's Groundbreaking Chess Strategies and the Promise of AI*. Alkmaar, The Netherlands: New in Chess, 2019.

[25] J. Kloetzer, H. Iida, and B. Bouzy, "A comparative study of solvers in amazons endgames," in *Proc. IEEE Symp. Comput. Intell. Games*, Dec. 2008, pp. 378–384.

[26] J. Uiterwijk, "Predictions and perfection: The art of solving games," Tech. Rep. [Online]. Available: https://www.researchgate.net/publication/228915062_Predictions_and_Perfection_The_Art_of_Solving_Games

[27] H. Iida, K. Takahara, J. Nagashima, Y. Kajihara, and T. Hashimoto, "An application of game-refinement theory to mah jong," in *Proc. Int. Conf. Entertainment Comput.* Berlin, Germany: Springer, 2004, pp. 333–338.

[28] Z. Song, H. Iida, and H. van den Herik, "Probability based proof number search," in *Proc. 11th Int. Conf. Agents Artif. Intell.*, 2019, pp. 661–668.

[29] Z. Song and H. Iida, "Using single conspiracy number to analyze game progress patterns," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2017, pp. 219–222.

[30] H. Iida and M. N. A. Khalid, "Using games to study law of motions in mind," *IEEE Access*, vol. 8, pp. 138701–138709, 2020.

[31] M. N. A. Khalid, U. K. Yusof, H. Iida, and T. Ishitobi, "Critical position identification in games and its application to speculative play," in *Proc. Int. Conf. Agents Artif. Intell.*, vol. 2. Setubal, Portugal: Science and Technology Publications, 2015, pp. 38–45.

[32] Z. Song and H. Iida, "Using single conspiracy number for long term position evaluation," *ICGA J.*, vol. 40, no. 3, pp. 269–280, Mar. 2019.

[33] D. A. McAllester, "Conspiracy numbers for min-max search," *Artif. Intell.*, vol. 35, no. 3, pp. 287–310, Jul. 1988.

[34] J. Schaeffer, "Conspiracy numbers," *Artif. Intell.*, vol. 43, no. 1, pp. 67–84, Apr. 1990.

[35] U. Lorenz, V. Rottmann, R. Feldmann, and P. Mysliwietz, "Controlled conspiracy-number search," *ICGA J.*, vol. 18, no. 3, pp. 135–147, 1995.

[36] U. Lorenz, "Controlled conspiracy-2 search," in *Proc. Annu. Symp. Theor. Aspects Comput. Sci.* Berlin, Germany: Springer, 2000, pp. 466–478.

[37] D. McAllester and D. Yuret, "Alpha-beta-conspiracy search," *ICGA J.*, vol. 25, no. 1, pp. 16–35, Mar. 2002.

[38] M. N. A. Khalid, E. M. Ang, U. K. Yusof, H. Iida, and T. Ishitobi, "Identifying critical positions based on conspiracy numbers," in *Proc. Int. Conf. Agents Artif. Intell.* Cham, Switzerland: Springer, 2015, pp. 100–127.

[39] Q. Vu, T. Ishitobi, J.-C. Terrillon, and H. Iida, "Using conspiracy numbers for improving move selection in minimax game-tree search," in *Proc. 8th Int. Conf. Agents Artif. Intell., Sci. Technol. Publications*, Setubal, Portugal, 2016, pp. 400–406, doi: 10.5220/0005753004000406.

[40] J. Pawlewicz and R. B. Hayward, "Conspiracy number search with relative sibling scores," *Theor. Comput. Sci.*, vol. 644, pp. 127–142, Sep. 2016.

[41] S. Gelly and D. Silver, "Monte-carlo tree search and rapid action value estimation in computer go," *Artif. Intell.*, vol. 175, no. 11, pp. 1856–1875, Jul. 2011.

[42] L. V. Allis, M. van der Meulen, and H. J. van den Herik, "Proof-number search," *Artif. Intell.*, vol. 66, no. 1, pp. 91–124, Mar. 1994.

[43] A. Nagai, "DF-PN algorithm for searching and/or trees and its applications," Ph.D. dissertation, Dept. Inf. Sci., Univ. Tokyo, Tokyo, Japan, 2002.

[44] J. Pawlewicz and L. Lew, "Improving depth-first pn-search: 1+ε trick," in *Proc. Int. Conf. Comput. Games*. Berlin, Germany: Springer, 2006, pp. 160–171.

[45] J. Pawlewicz and R. B. Hayward, "Scalable parallel dfpn search," in *Proc. Int. Conf. Comput. Games*. Cham, Switzerland: Springer, 2013, pp. 138–150.

[46] J.-T. Saito, G. Chaslot, J. W. Uiterwijk, and H. J. Van Den Herik, "Monte-carlo proof-number search for computer go," in *Proc. Int. Conf. Comput. Games*. Berlin, Germany: Springer, 2006, pp. 50–61.

[47] A. Saffidine and T. Cazenave, "Developments on product propagation," in *Proc. Int. Conf. Comput. Games*. Cham, Switzerland: Springer, 2013, pp. 100–109.

[48] A. Kishimoto, B. Buesser, B. Chen, and A. Botea, "Depth-first proof-number search with heuristic edge cost and application to chemical synthesis planning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 7226–7236.

[49] R. Coulom, "Efficient selectivity and backup operators in Monte-Carlo tree search," in *Proc. Int. Conf. Comput. Games*. Berlin, Germany: Springer, 2006, pp. 72–83.
[50] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, 2006, pp. 282–293.
[51] J. Kloetzer, H. Iida, and B. Bouzy, "The monte-carlo approach in Amazons," in *Proc. Comput. Games Workshop*, 2007, pp. 185–192.
[52] I. Althöfer, "On board-filling games with random-turn order and Monte Carlo perfectness," in *Proc. Adv. Comput. Games*. Berlin, Germany: Springer, 2011, pp. 258–269.
[53] G. M. J.-B. Chaslot, M. H. M. Winands, H. J. V. D. Herik, J. W. H. M. Uiterwijk, and B. Bouzy, "Progressive strategies for Monte-Carlo tree search," *New Math. Natural Comput.*, vol. 4, no. 3, pp. 343–357, 2008.
[54] A. J. Palay, "Searching with probabilities," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, Jul. 1983.
[55] A. Primanita, M. N. A. Khalid, and H. Iida, "Characterizing the nature of probability-based proof number search: A case study in the othello and connect four games," *Information*, vol. 11, no. 5, p. 264, May 2020.
[56] E. Maby, M. Perrin, O. Bertrand, G. Sanchez, and J. Mattout, "BCI could make old two-player games even more fun: A proof of concept with 'Connect Four,'" *Adv. Hum.-Comput. Interact.*, vol. 2012, pp. 1–8, Jan. 2012.
[57] M. H. M. Winands, "6x6 LOA IS SOLVED," *ICGA J.*, vol. 31, no. 4, pp. 234–238, Dec. 2008.
[58] A. Primanita and H. Iida, "Nature of probability-based proof number search," in *Proc. Sriwijaya Int. Conf. Inf. Technol. Appl. (SICONIAN)*. Dordrecht, The Netherlands: Atlantis Press, 2020, pp. 485–491.
[59] H. J. van den Herik, J. W. H. M. Uiterwijk, and J. van Rijswijck, "Games solved: Now and in the future," *Artif. Intell.*, vol. 134, nos. 1–2, pp. 277–311, Jan. 2002.
[60] M. P. Schadd, M. H. Winands, H. J. Van Den Herik, G. M.-B. Chaslot, and J. W. Uiterwijk, "Single-player monte-carlo tree search," in *Proc. Int. Conf. Comput. Games*. Berlin, Germany: Springer, 2008, pp. 1–12.
[61] A. Reinefeld, "Complete solution of the eight-puzzle and the bene t of node ordering in ida," in *Proc. Int. Joint Conf. Artif. Intell.*, 1993, pp. 248–253.
[62] Y. Shoham and J. Schaeffer, "The FESS algorithm: A feature based approach to single-agent search," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2020, pp. 96–103.
[63] R. Mehta, "2048 is (PSPACE) hard, but sometimes easy," 2014, *arXiv:1408.6315*. [Online]. Available: http://arxiv.org/abs/1408.6315
[64] G. Cirulli. (2014). *2048 Game Official Website*. Accessed: Aug. 24, 2020. [Online]. Available: http://https://play2048.co/
[65] B. Kuchera. (2015). *Why it Took a Year to Make, and Then Break Down, an Amazing Puzzle Game*. [Online]. Available: http://www.posted 06-february-2013
[66] J. L. Miller. (2017). *The Mathematics of 2048: Counting States With Combinatorics*. [Online]. Available: http://www.posted 17-september-2017
[67] J. L. Miller. (2017). *The Mathematics of 2048: Counting States by Exhaustive Enumeration*. [Online]. Available: http://www.posted 12-december-2017
[68] L. Ma, "Xiangqi vs chess—The cultural differences reflected in Chinese and western games," *Open J. Social Sci.*, vol. 8, no. 3, p. 52, 2020.
[69] S. Busala, Z. Song, H. Iida, M. Khalid, and U. Yusof, "Single conspiracy number analysis in checkers," in *Proc. 11th Int. Conf. Agents Artif. Intell.*, 2019, pp. 539–546.
[70] Y. An, A. Primanita, M. N. Akmal Khalid, and H. Iida, "Ascertaining the play outcome using the single conspiracy number in GoBang," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2020, pp. 658–661.
[71] J. Sijtsma, N. H. Jansen, and F. W. Vaandrager, "Creating a formal model of the game 2048," Ph.D. dissertation, Radboud Univ., Nijmegen, The Netherlands, 2020.
[72] P. Rodgers and J. Levine, "An investigation into 2048 AI strategies," in *Proc. IEEE Conf. Comput. Intell. Games*, Dortmund, Germany, Aug. 2014, pp. 1–2, doi: 10.1109/CIG.2014.6932920.
[73] H. Iida, N. Takeshita, and J. Yoshimura, "A metric for entertainment of boardgames: Its implication for evolution of chess variants," in *Entertainment Computing*. Boston, MA, USA: Springer, 2003, pp. 65–72.
[74] K. Xiaohan, M. N. A. Khalid, and H. Iida, "Player satisfaction model and its implication to cultural change," *IEEE Access*, vol. 8, pp. 184375–184382, 2020.
[75] P. Jansen, "Problematic positions and speculative play," in *Computers, Chess, and Cognition*. New York, NY, USA: Springer, 1990, pp. 169–181.
[76] P. J. Jansen, "Using knowledge about the opponent in game-tree search," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, Sep. 1992.
[77] H. Iida, J. W. H. M. Uiterwijk, H. J. van den Herik, and I. S. Herschberg, "Potential applications of opponent-model search1: Part 1: The domain of applicability," *ICGA J.*, vol. 16, no. 4, pp. 201–208, Dec. 1993.
[78] J. Stork, A. E. Eiben, and T. Bartz-Beielstein, "A new taxonomy of global optimization algorithms," *Natural Comput.*, vol. 19, no. 3, pp. 1–24, Nov. 2020.

ANGGINA PRIMANITA received the M.IT. degree in game informatics from James Cook University (JCU), Singapore, in 2013. She is currently pursuing the Ph.D. degree with the Graduate School of Advanced Science and Technology, JAIST, Nomi, Japan. Her current work is focused on establishing the link between game computation and game refinement theory in the area of entertainment technology, under the supervision of Prof. Hiroyuki Iida. Her research interests include game tree search, game solvers, single-agent game playing, pathfinding search algorithms, and data science in gaming.

MOHD NOR AKMAL KHALID (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from the University of Science, Malaysia, in 2013, 2015, and 2018, respectively. He is currently working as an Assistant Professor with the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), and a member of the Research Center for Entertainment Science. He specializes in artificial intelligence techniques, game informatics, evolutionary algorithms, and decision support systems. His work focuses specifically on methods of optimization and game informatics in the fields of operation research and entertainment technology. His research interests include manufacturing systems, artificial intelligence techniques, game analysis and informatics, search algorithms, optimization techniques, advanced scheduling and planning, and machine learning methods.

HIROYUKI IIDA received the Ph.D. degree in heuristic theories of game tree search from the Tokyo University of Agriculture and Technology, Tokyo, in 1994. He was affiliated with Shizuoka University, Hamamatsu, and a Guest Researcher with Maastricht University. He is currently a Japanese Computer Scientist and a Computer Game Researcher with a focus on game refinement theory, opponent-model search, and computer shogi. He is also the Trustee and the Vice President of educational and student affairs with the Japan Advanced Institute of Science and Technology (JAIST), the Director of the Global Communication Center and the Research Center for Entertainment Science, and Head of the Iida Laboratory. He is a professional 7-dan shogi player, a coauthor of the shogi program Tacos, and a four-time gold medal winner at the Computer Olympiad. His research interests include artificial intelligence, game informatics, game theory, mathematical models, search algorithms, game refinement theory, game tree search, and entertainment science. He is a member of the Board of the ICGA as a Secretary-Treasurer and a Section Editor of the *ICGA Journal*.