

## Towards a Characterization of Chance in Games The Case of Two-Player Zero-Sum Games with Perfect Information

Jakob Erdmann<sup>1</sup>

Lehrstuhl für Mathematische Optimierung,  
Institut für Angewandte Mathematik,  
Fakultät für Mathematik und Informatik,  
Friedrich-Schiller-Universität, 07737 Jena, Germany.

### ABSTRACT

The distinction between games of chance and games of skill is not well-defined at present. We introduce the concept of *chanciness* for non-deterministic two-player zero-sum games with sequential moves and perfect information. *chanciness* quantifies how strongly a game is influenced by chance events. We demonstrate that the relative influence of chance on game outcomes varies with the skill of the playing agents. Therefore, we assign a *chanciness* value to the combination of a game and a specific set of players. The effective computability of *chanciness* is demonstrated for exemplary games.

**Key words:** games of chance, chance and skill, automatic evaluation, computer aided game inventing

### 1. INTRODUCTION

Games are conventionally classified as games of chance, games of skill or games with mixed characteristics. The desire for such a classification stems from various sources.

- Many countries have laws to regulate gambling, in other words, playing games of chance with money at stake. It is then up to legislators and lawyers to decide which games are within the scope of such laws. In Germany, the gambling law “Glücksspielgesetz” defines games of chance. A game of chance is one, in which outcomes are fully or mostly determined by chance (Staatsvertrag, 2007). The term “mostly” is not elucidated. This distinction becomes even less clear when the accumulated outcomes of numerous game rounds are considered. When the game in question involves even a tiny bit of skill, the central limit theorem may be used to argue that the accumulated result of a sequence of games is determined by skill alone (Alon, 2007).

The controversial nature of the skill / chance distinction is apparent from US-American court rulings regarding poker (Billings, Davidson, and Szafron, 2002). Poker was ruled a *game of skill* in Pennsylvania (Cypra, 2009) and a *game of chance* in North Carolina (Burton, 2007). This contradiction leaves open the task of quantifying the skill and chance inherent in a game.

- Naturally, the people who play games are interested in matters of chance and skill. Some players prefer games of chance over games of skill while others have opposite tastes. In this context, a precise classification of games may not be important. However, the results of individual matches (instances) lends itself to analysis. Having won a game with a chance component we may ask ourselves whether our victory was due to our skillful moves or simply caused by a string of lucky dice rolls (Sno, 2009). The answer to this question is central to the feeling of accomplishment of the players and thus to the enjoyment of the game.

- A third group of people, who concern themselves with the balance of chance and skill in games, are game

---

<sup>1</sup>jakob.erdmann@uni-jena.de

inventors. When designing a game for a particular audience, the inventor may wish to realize a particular level of chance and skill. Unfortunately, it is hard to measure the effects that subtle rule variations have on this balance.

We are concerned with *computer-aided game inventing* as proposed by Althöfer (Althöfer, 2003). We wish to use computers in place of human players to test newly designed games. A prerequisite of such schemes is the existence of computer-readable rule descriptions. The field of *general game playing* (Love *et al.*, 2008; Pell, 1994; Finnsson and Björnsson, 2008) provides computer algorithms, capable of using such rule descriptions to play games against each other.

Our eventual goal is an algorithm for automatically determining the extent of chance and skill in a game. In this paper we present the concept of *chanciness*  $C$ , our theoretical groundwork for such an undertaking. The *chanciness* of a game is high if game results are strongly influenced by chance and it is low if the game allows for skillful play. We will show that our *chanciness* depends on the game players (agents). This leads to a measure of  $C(\text{game}, X, Y)$  where  $X$  and  $Y$  are specific agents who compete in a two-player game. We narrow our focus to two-player zero-sum games of perfect information, sequential moves, finite length, and a finite number of moves in every game state.

In Section 2 we discuss previous works on formalizing the extent of chance and skill in games. We introduce terms and concepts for our discussion in Section 3. Our concept of *chanciness* rests on the analysis of individual game instances. To what extent did player moves and random events influence the game result? We deal with the subject of *influence* in Section 4. Based on our notion of influence we define the *relative influence of chance* in Section 5. In Section 6 we generalize from game instances to games and define *chanciness*. Section 7 contains exemplary measurements of *chanciness* for different games that are played by various computer agents. Section 8 relates our concept of *chanciness* to games played by humans. In Section 9 we summarize our findings and propose further lines of research.

## 2. PREVIOUS WORK

The question whether games should be legally classified as games of skill or games of luck has been addressed by Borm and Dreef since 1997 (Borm and van der Genugten, 1997; Dreef, Borm, and van der Genugten, 2004; Dreef, 2005). Based on three types of playing agents they introduce the concept of *Relative Skill (RS)* and give values for the games of Roulette, Black Jack, Poker and other games. Games that allow for skillful play receive a higher rating of *RS* than games in which chance dominates the result. Thus, *RS* expresses the opposite idea of *chanciness* and is closely linked to our question.

*RS* is measured by comparing expected game rewards of beginners, optimal players and fictive players. “Fictive players” know in advance the result of any kind of random event.

$$RS = \frac{\text{gain}_{\text{optimal}} - \text{gain}_{\text{beginner}}}{\text{gain}_{\text{fictive}} - \text{gain}_{\text{beginner}}} \quad (1)$$

Dreef recommends that the playing style of the beginner is to be modelled in one of three ways depending on what is called “the structure” of the game.

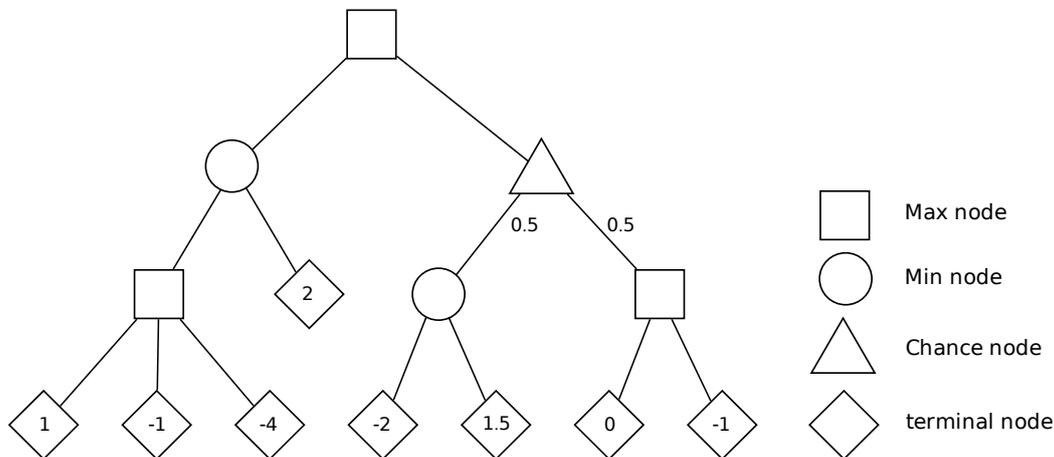
- random move selection
- quantitative analysis of human beginners playing style
- using domain expert knowledge

This approach, however, is not applicable to our problem. First of all, the modelling of the beginner depends on subjective judgements by the researcher and precludes the use of automatic algorithms. Secondly, the optimal behavior for a given game can in general not be discovered by human analysis, much less by an automated process (in reasonable time). Thirdly, the Borm and Dreef notion of “fictive players” allows the creation of games, that receive ratings of *RS* which contradict common sense (see the game of *lottery chess* in Appendix A).

However, as our most important objection we wish to argue that an assessment of *Relative Skill* or *chanciness* for a given game can not be obtained independently of the particular agents.

### 3. TERMS AND CONCEPTS

In this section, we introduce the necessary terms and concepts for our discussion of *chanciness*. The starting point is the central term *game*. We wish to discuss non-deterministic<sup>2</sup> finite zero-sum games of two players, perfect information and sequential moves<sup>3</sup>.



**Figure 1:** An exemplary game tree with chance nodes

As our game model we use the extensive form with chance moves (Shoham and Leyton-Brown, 2008). A game  $G$  is represented by a finite tree as in Figure 1. Nodes of the tree correspond to game states. The leaves of the tree represent terminal game states and are labeled with the deterministic result  $r$  of the game. The game result  $r$  is a real number that gives the reward for player *Max* (his goal is to maximize  $r$ ). As  $G$  is a zero-sum game, the other player, *Min*, receives  $-r$  as reward. The inner nodes represent non-terminal states that require a move. They are labeled as either *Max* or *Min* nodes if that player is to move. If the game state requires a “chance move”, such as a roll of the dice or the drawing of cards from a randomized deck, the node is labeled as a “chance node”. The links in the tree correspond to the available moves by either players (“player moves”) or chance (“chance moves”). Chance moves are labeled with their respective probabilities. For every chance node  $s$  we define  $\mathcal{C}(s)_i$  as the probability that chance will move to the  $i$ th child of  $s$ .

A match  $g$  of game  $G$  is given by a tuple  $g = (X, Y, (s_0, s_1, \dots, s_k))$ .  $X$  and  $Y$  are agents who take on the roles of players *Max* and *Min*.  $(s_0, \dots, s_k)$  is a path of nodes (game states) through the tree from the root to a leaf node. As we are interested in *computer-aided game inventing*, our agents are computer algorithms. The terms “player” and “agent” are somewhat synonymous. We’ll use the term “player” to emphasize the roles in a game (*Max* and *Min*). The use of “agent” emphasizes the algorithm that is used to fill the role of a player.

Since we are not interested in *how* our agents form their move decisions we can use a very simple agent model. It suffices that the agents are somehow able to select their move from the finite set of available moves. To reflect our ignorance, we use a stochastic model. For every game  $G$  let  $S_G$  denote the set of all possible states of  $G$ . Let  $S_G^{Max} \subseteq S_G$  denote the set of all states in which player *Max* has to move. For every game state  $s \in S_G$  let  $\text{Children}(s) = (c_1, c_2, \dots, c_n)$  denote the set of states that can be reached by one legal move. An agent  $X$  capable of playing  $G$  as *Max* is a function that assigns to every state  $s \in S_G^{Max}$  a discrete probability distribution  $(p_1, p_2, \dots, p_n)$  over  $\text{Children}(s)$ .

$$X(s) = (p_1, \dots, p_n), \quad p_i \geq 0, \quad \sum p_i = 1, \quad |\text{children}(s)| = n \quad (2)$$

Let  $\text{Children}(s) = (c_1, \dots, c_n)$ .  $X(s)_i = p_i$  is the probability that  $X$  will move to node  $c_i$  in state  $s$ . Likewise, agents capable of playing  $G$  as *Min* provide probability distributions for every state  $s \in S_G^{Min}$ . Deterministic agents are handled as a special case of stochastic agents. For simplicity’s sake our agents only consider the current state  $s$  when selecting their move. This stochastic independence of moves is not a requirement for the

<sup>2</sup>We treat deterministic games as a special case of non-deterministic games.

<sup>3</sup>Games with sequential moves are also known as *Dynamic Games*.

calculation of *chanciness*. In principle, agents could consider previous states and opponent moves when making their move<sup>4</sup>

Using the terms defined above, we can describe a wide selection of board games such as *chess*, *backgammon* and *pachisi* (reduced to two players). When viewed as a two-player game between player and bank, even *Roulette* can be modeled in these terms. However, some care must be taken when modelling these games as trees of finite depth. While all games have finite length in practice, formal proof for finiteness may not be available. To satisfy the formal constraint, slight rule changes are usually sufficient<sup>5</sup>

#### 4. THE CONCEPT OF INFLUENCE

When describing a game  $G$  as a game of chance or a game of skill humans employ an intuitive understanding of influence. To qualify as a game of chance,  $G$  must involve some sort of randomizing device and that device must have an influence on the game result. Likewise, to deserve the label “game of skill”, there must be room for making “good” and “bad” move decisions by the players. In other words, players must be able to influence game results through skillful decisions. If the result of a game is subject to influence of both chance and skill, we can try to compare their relative importance. For a scientifically meaningful comparison we must develop a rigorous definition of *influence*.

In Subsection 4.1 we explain our approach for formalizing the concept of *influence*. This leads us to the need for evaluating game states. We discuss problems related to positional evaluation in Subsection 4.2 and introduce *relative positional values* in Subsection 4.3. We explain how *relative positional values* solve the aforementioned problems in Subsection 4.4 and finally present our definition of *influence* in Subsection 4.5.

##### 4.1 Approach to Formalization

To understand influence we must distinguish between the game  $G$  and a match  $g$  of the game  $G$  that took place between specific players. When talking about the game  $G$  we actually discuss *possible* influence. When we say  $G$  is a game of chance, we are saying, a typical match  $g$  of  $G$  is strongly influenced by chance. However, not every match plays the same. If one player gets very lucky, he may win without effort. In another match, the influence of dice rolls may favor both players equally and the result is then decided by the skill of the players. Therefore, we will define *influence* for matches  $g$  of game  $G$ . After we have understood how chance and skill affect matches, we generalize our findings to talk about the properties of  $G$ .

What exactly is influence? When we talk about the influence of moves (player moves or chance moves), we talk about their influence on the game result. A single move rarely decides the game. Instead, moves lead to game states in which a player has improved or reduced chances of winning. From the perspective of player *Max*, a move has a positive influence if it improves his position and hence increases his expected reward from the game. This interpretation is crucial because it allows us to quantify influence. We evaluate consecutive game states and define *influence* as the change in evaluation. This approach leads us to the problem of evaluating game states.

##### 4.2 Difficulties of Positional Evaluation

For deterministic 2-player zero-sum games such as *chess*, the *minimax* algorithm (Zermelo, 1913; von Neumann, 1927) allows us to compute the game-theoretic value of every position by backward induction. These values represent the game result for agents who play optimally. For non-deterministic games, the *expectimax* algorithm (a generalization of minimax for trees with chance nodes) computes the expected values for optimal agents

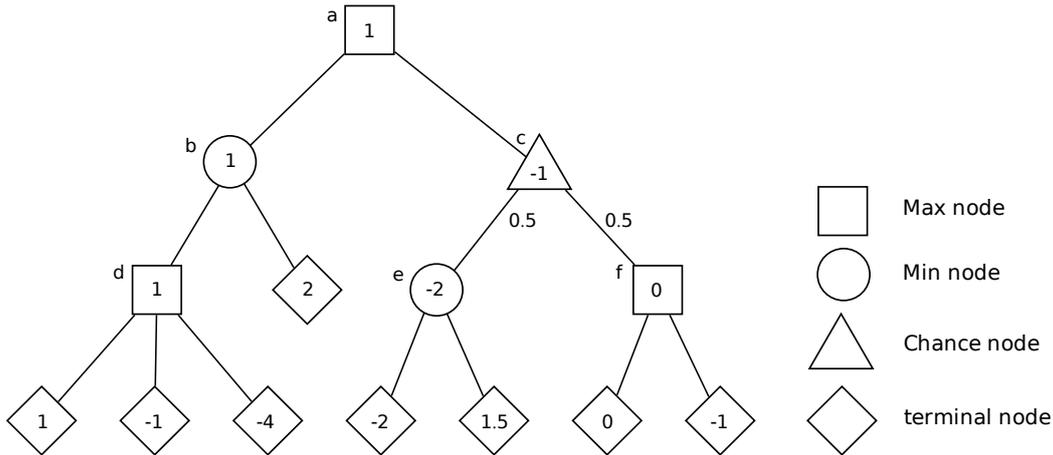
<sup>4</sup>This assumes that the game is represented by a real tree. (Two chess positions with the same board and different histories would be represented by two different nodes).

<sup>5</sup>Tournament chess rules include the *50 move rule* to avoid games of arbitrary length (FID, 2008). For backgammon a limit of 10000 on the total number of moves in a match has little consequence to the way the game is played (after 10000 moves a draw would be declared).

(Shoham and Leyton-Brown, 2008). The expectimax value  $V(s)$  of a position  $s$  is given by a recursive formula:

$$V(s) := \begin{cases} r(s) & \text{if } s \text{ is terminal} \\ \max_i V(c_i) & \text{if Max moves in } s \\ \min_i V(c_i) & \text{if Min moves in } s \\ \sum_i \mathcal{C}(s)_i * V(c_i) & \text{if Chance moves in } s \end{cases} \quad (3)$$

Here  $r(s)$  is the game result for a terminal position  $s$ ,  $\text{Children}(s) = (c_1, \dots, c_n)$  and  $\mathcal{C}(s)_i$  is the probability that chance selects  $c_i$ . By starting at the leaves and backing up towards the root, the expectimax value can be computed for every node. Figure 2 gives the expectimax values for all game states of our example game from Section 3.



**Figure 2:** A game tree with expectimax values

The minimax and expectimax algorithm are widely used to analyze games. They also form the basis for many game playing agents. However, in light of our goals, the positional values computed by these algorithms have two important drawbacks.

- The *Interpretation Problem* arises when using expectimax values to reason about games between imperfect agents. Consider the game tree in Figure 2. In the root position  $a$ , an optimal agent playing as *Max* moves to position  $b$  which leads to the maximum reward  $r = 1$ . For that agent, moving to state  $c$  instead would be a “bad” move (adverse influence) because his expected reward would drop from 1 to -1. However, if an imperfect agent was playing as *Max*, that assessment might change. Consider an imperfect agent  $X$  who does not play optimal in position  $d$ . This could be modeled as  $X(d) = (0.25, 0.25, 0.5)$ . Thus, in position  $d$ , agent  $X$  would have an expected reward of  $-2$ . If we assume further that  $X$  plays optimal in position  $f$  and  $Y$  plays optimal as well, we must conclude that the move to  $b$  is “bad” for agent  $X$  while the move to  $c$  is “good” (beneficial influence). This example demonstrates that positions must be evaluated with respect to the playing agents. As another example, imagine a chess master giving advice to a novice. A risky gambit that the chess master would play is not advisable for the novice who cannot exploit the situation. Thus, the gambit would be a “good” move for the master, but a “bad” move for the novice. When using expectimax values to calculate the influence of moves, the interpretation problem crucially effects our results.

- The practical use of expectimax values is further hampered by the *Resource Problem*. Since the expectimax algorithm requires the evaluation of almost<sup>6</sup> the complete game tree, it is unfeasible to compute these values for complex games. In practice expectimax values are approximated by using heuristics. These heuristic methods can be very efficient but must often be tuned for every new game. General game playing algorithms avoid this tuning but they give no guarantees as to the accuracy of their approximations.

In the next section we introduce a different approach to positional evaluation which avoids these problems.

<sup>6</sup>A tree may allow safe pruning of some branches (See (Hauk, Buro, and Schaeffer, 2006)).

### 4.3 Relative Positional Values

We define the *relative positional value*  $\tilde{V}(s, X, Y)$  for a game state  $s$  and the agents  $X, Y$ . The term *relative* emphasizes that the positional value depends on the agents.

The result  $r$  of a match that is started in state  $s$  and then played out by the agents  $X$  (as *Max*) and  $Y$  (as *Min*) can be seen as the realization of a random variable  $R$ . When  $s$  is a terminal state the realization of  $R$  is a constant. Otherwise the value of  $r$  depends on the stochastic behavior of  $X, Y$  and on chance. We define the *relative positional value* as the expected value of  $R$ .

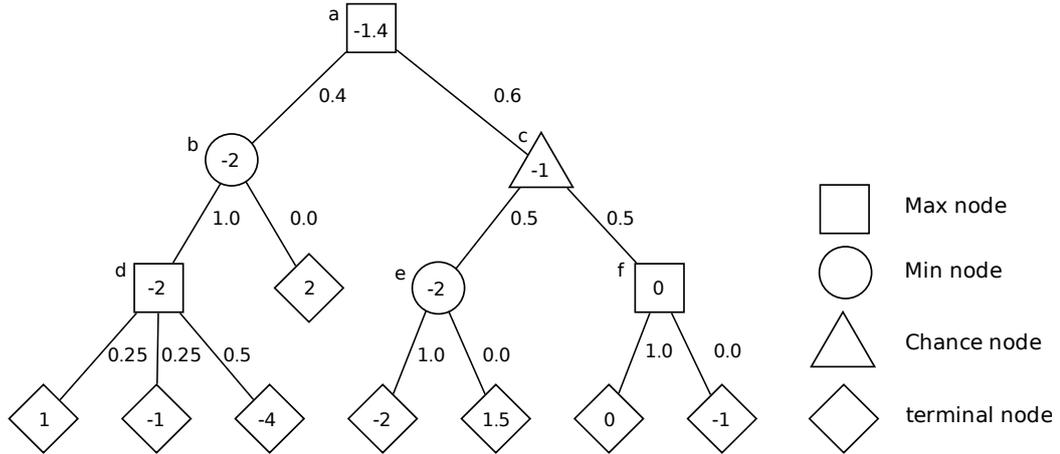
**Definition 4**

$$\tilde{V}(s, X, Y) := \mathbb{E}(R)$$

Figure 3 shows an annotated game tree. The move probabilities for chance and for exemplary agents  $X, Y$  are given for every move in the tree. The *relative positional values* for every node in the tree were computed by a recursive formula that looks similar to Formula 3 but replaces the *min* and *max* terms with expectation terms.

$$\tilde{V}(s, X, Y) := \begin{cases} r(s) & \text{if } s \text{ is terminal} \\ \sum_i X(s)_i * \tilde{V}(c_i, X, Y) & \text{if } X(\text{Max}) \text{ moves in } s \\ \sum_i Y(s)_i * \tilde{V}(c_i, X, Y) & \text{if } Y(\text{Min}) \text{ moves in } s \\ \sum_i \mathcal{C}(s)_i * \tilde{V}(c_i, X, Y) & \text{if Chance moves in } s \end{cases} \quad (5)$$

Here  $r(s)$  is the game result for a terminal position  $s$ ,  $\text{Children}(s) = (c_1, \dots, c_n)$ , and  $X(s)_i$  is the probability that  $X$  selects  $c_i$  ( $Y(s)_i$  is the probability that  $Y$  selects  $c_i$ ).



**Figure 3:** A game tree with move probabilities for specific agents  $X$  and  $Y$ . The *relative positional values* for these agents are given for every node.

### 4.4 Relative Positional Values Compared to Expectimax Values

Our new concept of *relative positional values* solves the two problems from Subsection 4.2. The *Interpretation Problem* is resolved by taking the behavior of imperfect agents into account. Figure 3 above shows the same game tree as Figure 2, this time annotated with *relative positional values* for the imperfect agents  $X$  and  $Y$ . Our intuitions of “good” and “bad” moves for  $X$  are reflected by increasing and decreasing positional values.

The move from root node  $a$  to node  $c$  is “good” because  $\tilde{V}(c, X, Y) = -1 > -1.4 = \tilde{V}(a, X, Y)$ . Likewise, the move from  $a$  to the other successor  $b$  is “bad” because  $\tilde{V}(b, X, Y) = -2 < -1.4 = \tilde{V}(a, X, Y)$ . It must be stressed that these values (as well as the judgements of “good” and “bad”) are specific to the agents  $X$  and  $Y$ .

The *Resource Problem* from Subsection 4.2 stems from the algorithmic complexity of computing minimax and expectimax values. The exact computation of *relative positional values* is just as expensive as the computation of expectimax values.

However, it is possible to approximate *relative positional values* with Monte-Carlo methods thus saving computing time. As in Subsection 4.3 let  $R$  be a random variable that gives the result of a match between  $X$  and  $Y$  with start in state  $s$ . Each simulation of such a match yields a realization  $R_i$  of  $R$ . The average over  $n$  such realizations  $R_i$  can be used to approximate the expected value of  $R$  and hence the *relative positional value* of  $s$  for  $X$  and  $Y$ .

$$\tilde{V}(s, X, Y) \approx \tilde{V}_{\text{approx}} = \frac{1}{n} \sum_{i=1}^n R_i \quad (6)$$

Since we intend to use computer algorithms as agents, simulation games can be repeated as often as desired (given enough computing resources).

In Subsection 4.2 we noted that approximations of expectimax values either rely on game specific knowledge or fail to give guarantees on accuracy.

In contrast, approximations of *relative positional values* via Monte-Carlo methods are general and give such guarantees. They are general because we already assume computer agents that can be used for simulation games (these agents are generally imperfect and therefore unsuitable for approximating expectimax values).

Monte-Carlo methods also give guarantees on the accuracy of approximation. The standard deviation of  $\tilde{V}_{\text{approx}}$  (and thus the error in approximation) varies inversely with the square root of the number of observations:  $\sigma(\tilde{V}_{\text{approx}}) = \sigma(R)/\sqrt{n}$  (central limit theorem).

In conclusion we deem *relative positional values* suitable for evaluating the influence of game moves.

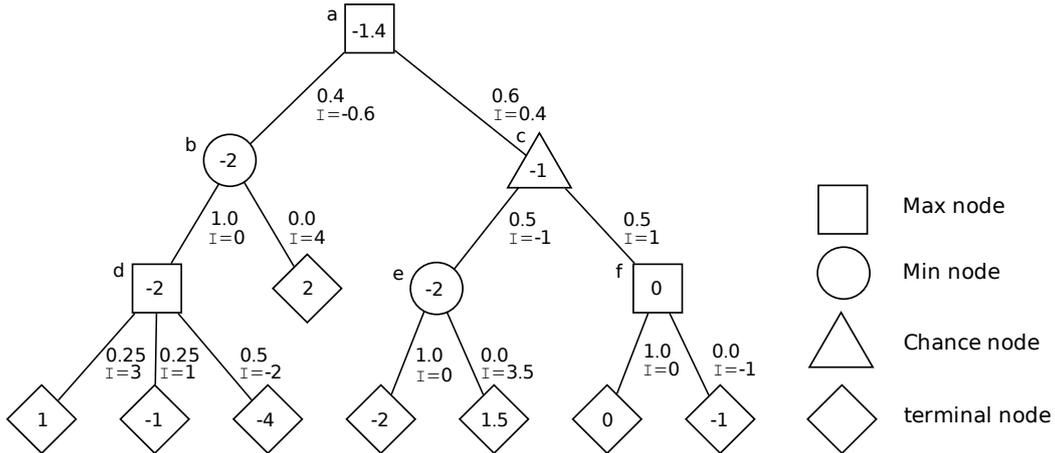
#### 4.5 Influence Defined

Our concept of *relative positional values*, from Subsection 4.3 allow us to define the *influence*  $I$  of moves. Let  $m = (s, s')$  be a move that transforms position  $s$  into  $s'$ . We define the influence of that move in a game between the agents  $X, Y$  as follows.

##### Definition 7

$$I(m, X, Y) := \tilde{V}(s', X, Y) - \tilde{V}(s, X, Y)$$

This definition does not distinguish between “player moves” and “chance moves”. Figure 4 is based on the same game tree and agents as Figure 3. Additionally, it shows the influence of all moves.



**Figure 4:** A game tree with move probabilities, *relative positional values* and the influence  $I$  for all moves.

As a consequence of Definition 7 we get

**Lemma 8** “Deterministic moves” have no influence.

We call a move  $m = (s, s')$  “deterministic move” for agent  $X$  if state  $s$  is always followed by the specific move  $s'$ . In other words  $X(s) = (0 \dots 0, 1, 0 \dots 0)$ : move  $s'$  follows with probability 1. As an example see the move  $(b, d)$  in Figure 4. Here, the agent  $Y$  playing as *Min* moves from  $b$  to  $d$  with probability 1.

**Proof:** Let  $s$  be a game state with  $\text{Children}(s) = (c_1 \dots c_n)$  and  $X$  to move. Let  $Y$  be an arbitrary agent and  $X(s)_k = 1$  (move  $(s, c_k)$  is a deterministic move). Using Equation 5, the expected game result in position  $s$ , can be expressed as a weighted sum of the expected results of the children of  $s$ .

$$\begin{aligned} \tilde{V}(s, X, Y) &= \sum_{i=1}^n X(s)_i \cdot \tilde{V}(c_i, X, Y) \\ &= 1 \cdot \tilde{V}(c_k, X, Y), \text{ hence} \\ I((s, c_k), X, Y) &= \tilde{V}(c_k, X, Y) - \tilde{V}(c_k, X, Y) = 0 \end{aligned}$$

□

## 5. THE RELATIVE INFLUENCE OF CHANCE

In the previous section we have defined the *influence*  $I(m, X, Y)$  of a move  $m$  in a game between the agents  $X$  and  $Y$ . Using this definition we can compare the influence of chance moves with the influence of player moves on the result of a match.

Let  $g = (X, Y, (s_0, s_1, \dots, s_n))$  be a match of game  $G$  between  $X$  and  $Y$  ( $X$  playing as *Max*,  $Y$  as *Min*). Let  $m_i$  denote the move  $(s_i, s_{i+1})$ . The result of  $g$  is the relative value of its final position  $\tilde{V}(s_n, X, Y)$ . As the value of a terminal game state does not depend on the agents anymore *relative positional value* and expectimax value are the same, hence  $\tilde{V}(s_n, X, Y) = V(s_n)$ . Using our definition of *influence* we can express the result of  $g$  by the influences of the moves  $s_i$ .

### Lemma 9

$$V(s_n) = \tilde{V}(s_n, X, Y) = \tilde{V}(s_0, X, Y) + \sum_{i=0}^{n-1} I(m_i, X, Y)$$

**Proof:** Applying Definition 7 leads to a telescoping sum that leaves only the term  $\tilde{V}(s_n, X, Y)$ :

$$\begin{aligned} &\tilde{V}(s_0, X, Y) + \sum_{i=0}^{n-1} I(m_i, X, Y) \\ &= \tilde{V}(s_0, X, Y) + \sum_{i=0}^{n-1} \tilde{V}(s_{i+1}, X, Y) - \tilde{V}(s_i, X, Y) \\ &= \tilde{V}(s_n, X, Y) + \sum_{i=0}^{n-1} \tilde{V}(s_i, X, Y) - \tilde{V}(s_i, X, Y) \\ &= \tilde{V}(s_n, X, Y) \end{aligned}$$

□

Lemma 9 “explains” how the combined influence of all moves leads to the result of match  $g$ . The term  $\tilde{V}(s_0, X, Y)$  is special because it is not the influence of a move but the expected value of the starting position. It reflects the “fairness” of the game  $G$  combined with the skill difference of the agents  $X$  and  $Y$ . We call  $\tilde{V}(s_0, X, Y)$  the *match expectation* ME. We can group all the moves that were made in  $g$  according to their mover. This gives use the set of player moves  $M_{\mathcal{P}}$  and the set of chance moves  $M_{\mathcal{C}}$ .

$$\begin{aligned} M_{\mathcal{P}} &= \{(s_i, s_{i+1}) \mid \text{Max or Min to move in } s_i\} \\ M_{\mathcal{C}} &= \{(s_i, s_{i+1}) \mid \text{chance to move in } s_i\} \end{aligned}$$

The sets  $M_{\mathcal{P}}$  and  $M_{\mathcal{C}}$  allow us to sum the combined influences of player moves and chance moves as  $I_{\mathcal{P}}$  and  $I_{\mathcal{C}}$ .

$$I_{\mathcal{P}} = \sum_{m \in M_{\mathcal{P}}} I(m, X, Y)$$

$$I_{\mathcal{C}} = \sum_{m \in M_{\mathcal{C}}} I(m, X, Y)$$

The sets  $M_{\mathcal{P}}$ ,  $M_{\mathcal{C}}$  and the values  $ME$ ,  $I_{\mathcal{P}}$ ,  $I_{\mathcal{C}}$ , all depend on a concrete match  $g$ .

We can now rephrase Lemma 9 as

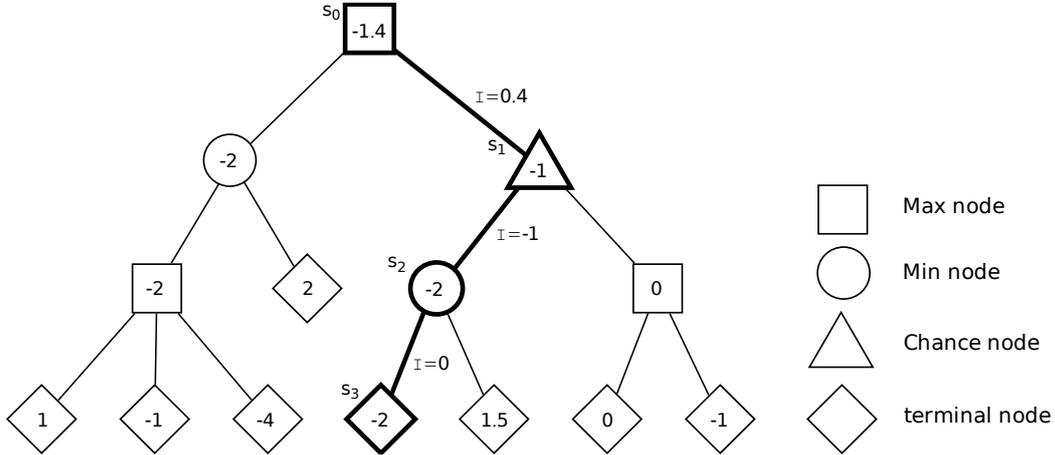
$$V(s_n) = ME + I_{\mathcal{P}} + I_{\mathcal{C}} \quad (10)$$

Based on these terms we define  $c(g)$ , the *relative influence of chance* on the result of match  $g$ .

**Definition 11**

$$c(g) := \frac{|I_{\mathcal{C}}|}{|ME| + |I_{\mathcal{P}}| + |I_{\mathcal{C}}|} \text{ if } |ME| + |I_{\mathcal{P}}| + |I_{\mathcal{C}}| \neq 0$$

We intentionally leave the special case of  $|ME| + |I_{\mathcal{P}}| + |I_{\mathcal{C}}| = 0$  undefined. Section 6 will show that the definition of this special case has no influence on our measure of *chanciness*.



**Figure 5:** A game tree with and a single match (path) in that tree with move influence for the agents  $X$  and  $Y$

Figure 5 shows an exemplary match  $g = (X, Y, (s_0 \dots s_3))$ . We observe that Lemma 9 holds:

$$V(s_3) = -2, \quad ME = -1.4, \quad I_{\mathcal{P}} = 0.4, \quad I_{\mathcal{C}} = -1$$

$$-2 = -1.4 - 1 + 0.4$$

We calculate the *relative influence of chance* for  $g$ .

$$c(g) = \frac{|I_{\mathcal{C}}|}{|ME| + |I_{\mathcal{P}}| + |I_{\mathcal{C}}|}$$

$$= \frac{1}{1.4 + 1 + 0.4} \approx 0.36$$

### 5.1 Properties of $c(g)$

For a match  $g = (X, Y, (s_0 \dots s_n))$ , the *relative influence of chance*  $c(g)$  is a real number from the interval  $[0, 1]$ . We can analyze how the components of Definition 11 are affected by match properties and how they affect the value  $c(g)$ .

•  $I_{\mathcal{C}}$ : if a game does not allow for chance moves, the influence of chance  $I_{\mathcal{C}}$  will be zero for any match  $g$  and hence  $c(g)$  will be zero as well. For a game that does allow chance moves  $I_{\mathcal{C}}$  will generally vary from match

to match. It may still be zero (or close to zero) for some matches if the influence of chance moves cancel out each other. For example, if a sequence of lucky dice rolls for player *Max* is followed by another sequence of dice rolls that favor player *Min*. Thus, even for games of chance, some matches may be completely decided by player actions.

- $I_{\mathcal{P}}$ : the term  $I_{\mathcal{P}}$  quantifies the combined influence of the players for a match  $g$ . Whereas games without chance moves are common, games without player moves are uncommon and may not be at all recognized as “games”. Even though player moves occur they may not be influential to the game result. For example, picking red or black in *Roulette* has no impact on the expected result. Thus, games that do not allow for “bad” and “good” moves will have low values of  $I_{\mathcal{P}}$  for any match  $g$ . Because of Lemma 8, a game where “good” moves are obvious to the agents will also allow for little player influence. This resonates strongly with a depreciation of games by humans who have fully mastered those games. Few people return to the game of *Noughts and Crosses* after mastering its “strategy”. High values of  $I_{\mathcal{P}}$  indicate that a game allows for influential moves and that the agents have not yet mastered the strategy. Consequently such matches will receive lower values of  $c(g)$ .

- ME: the *match expectation* ME, is defined as the expected result of a match (starting at the root position) between specific agents. Its value does not depend on the moves of a match because all matches for a game share the same root position. It is effectively a function of the game  $G$  and the agents  $X, Y$ . This function  $ME(G, X, Y)$  combines two different aspects of a match: the skill difference between the agents and the asymmetry of game  $G$ . If two agents of differing skill play a game that allows for influential moves, the better player can expect a higher reward. This usually leads to a higher value of  $|ME|$ . If the asymmetry of the player positions confers an advantage to one of the players, this also tends to increase the value of  $|ME|$ . Either effect is independent of chance and tends to increase the predictability of game results. Consequently,  $|ME|$  appears in the denominator of Definition 11.

As a remarkable consequence of combining skill difference and game asymmetry in the term ME, the playing order of the agents is relevant for the value of  $c(g)$ . In general  $|ME(G, X, Y)| \neq |ME(G, Y, X)|$ . To understand this, we must imagine an asymmetric game  $G$  (favoring player *Max*), a skilled agent  $X$  and a less skilled agent  $Y$ . If  $X$  plays as *Max*, ME will be high because the stronger agent plays in an advantageous position. If  $X$  plays as *Min*, ME will be closer to zero because skill differential and game asymmetry cancel each other out. In this case we have  $|ME(G, X, Y)| > |ME(G, Y, X)|$ .

## 6. CHANCINESS

In the previous section we have defined the *relative influence of chance* to analyze a specific match  $g$  of the game  $G$ . The analysis of several matches can be combined to reveal properties of the game  $G$ .

Before the agents  $X$  and  $Y$  play a match  $g$  of game  $G$ , the moves that will be taken and the result of the match are generally unknown. Since moves and result depend on stochastic processes we view ME,  $I_{\mathcal{P}}$ ,  $I_{\mathcal{C}}$  and likewise the *relative influence of chance*  $c(g)$  as random variables.

It might seem intuitive to define *chanciness* as the expected value of  $c(g)$ . However, that approach would lead to counterintuitive results for some games. Imagine a game  $G$  with the following structure:

$$\begin{aligned} c(g) &= \frac{\epsilon}{\epsilon + \epsilon \cdot 10^{-10}} \approx 1 && \text{for 90\% of the matches} \\ c(g) &= \frac{\epsilon}{\epsilon + \epsilon \cdot 10^{10}} \approx 0 && \text{for 10\% of the matches} \end{aligned}$$

Most single matches are decided by chance and our naive definition of *chanciness* would assign a value of  $\approx 0.9$  to this situation. This would be undesirable because the long term result of a series of matches is quite independent of chance.

To avoid this problem we define the *chanciness*  $C$  of a game when played by  $X$  and  $Y$  as a weighted expectation of the *relative influence of chance*.

### Definition 12

$$C(G, X, Y) = \frac{\mathbb{E}(|I_{\mathcal{C}}|)}{|ME| + \mathbb{E}(|I_{\mathcal{P}}|) + \mathbb{E}(|I_{\mathcal{C}}|)}$$

The following considerations motivate this definition: Every match  $g$  increases our knowledge about the game  $G$ . The gain in knowledge varies with the denominator of Definition 12,  $\mathbb{E}(|I_{\mathcal{P}}| + |I_{\mathcal{C}}|)$ . If that denominator is zero, we do not gain any knowledge about  $G$  at all (and thus,  $c(g)$  is undefined). We therefore weigh our values  $c(g)$  by this denominator before combining them in our definition of *chanciness*.

$$\begin{aligned} w(g) &:= \mathbb{E}(|I_{\mathcal{P}}| + |I_{\mathcal{C}}|) \\ C(G, X, Y) &:= \mathbb{E} \left[ \frac{c(g) \cdot w(g)}{\mathbb{E}(w(g))} \right] \\ &= \mathbb{E} \left[ \frac{|I_{\mathcal{C}}|}{\mathbb{E}(|I_{\mathcal{P}}|) + \mathbb{E}(|I_{\mathcal{C}}|)} \right] \\ &= \frac{\mathbb{E}(|I_{\mathcal{C}}|)}{\mathbb{E}(|I_{\mathcal{P}}|) + \mathbb{E}(|I_{\mathcal{C}}|)} \end{aligned}$$

Single matches where the *relative influence of chance*  $c(g)$  is undefined because  $w(g)$  is zero, do not present a problem with this definition of *chanciness*. In theory, the denominator of Definition 12 might be zero as well. In our opinion, this is not a problem either, because games where  $\mathbb{E}(|I_{\mathcal{P}}|)$ , and  $\mathbb{E}(|I_{\mathcal{C}}|)$  all equal zero are rather uninteresting.  $\mathbb{E}(|I_{\mathcal{C}}|) = 0$  implies that  $I_{\mathcal{C}}$  is always zero and thus dice moves are meaningless for the outcome of the game. Likewise  $\mathbb{E}(|I_{\mathcal{P}}|) = 0$  implies that  $I_{\mathcal{P}}$  is always zero and hence each player move is either deterministic or meaningless for the outcome of the game. In fact every state in the game tree that is ever visited would have the same *relative positional value* of zero.

Our *chanciness*  $C(G, X, Y)$  defined by 12 measures the propensity of matches to be (relatively) influenced by chance. The *chanciness* for a specific game and specific agents can be computed exactly by calculating the influence and probability of every possible move in  $G$ . For the game and agents from Diagram 4 we receive a *chanciness* of  $\approx 0.197$ .

For obvious reasons, this exhaustive method of computation quickly becomes unfeasible when considering larger game trees. Instead, we approximate *chanciness* via Monte-Carlo sampling. A number of matches  $g_i$  is played and the weighted sample mean of  $c(g_i)$  is taken as approximation of *chanciness*.

$$C(G, X, Y) \approx \frac{\sum_{g_i} |I_{\mathcal{C}}|}{\sum_{g_i} |\mathbb{E}(|I_{\mathcal{P}}|) + \mathbb{E}(|I_{\mathcal{C}}|)} \quad (13)$$

In the next section we present the results of such measurements.

## 7. SIMULATION RESULTS

In this Section we present measurements of *chanciness* for various combinations of games and agents. The combinations of games and agents are chosen so as to highlight important properties of *chanciness*. *Relative positional values* were calculated with the *Lookup Sampling* algorithm (Appendix B, a more efficient version of regular Monte-Carlo Sampling).

### 7.1 PaCRaWa

We introduce the game PaCRaWa (*Partially Controlled Random Walks*) to serve as a very simple game model. More precisely PaCRaWa $[d, T, a]$  is a class of games, as the rules contain several parameters: the power of the dice  $d \in \mathbb{R}_0^+$ , the number of turns  $T \in \mathbb{N}$ , and the beginners advantage  $a \in \mathbb{R}$ . The playing material consists of a counter variable  $c \in \mathbb{R}$  and a two-sided “dice”. At the beginning of the game,  $c$  is set zero. The players *Max* and *Min* take turns modifying the counter  $c$ . In every turn there are exactly two legal moves: increment  $c$  by one or decrement  $c$  by one. After a player move, the dice is “rolled” and according its outcome,  $c$  is either incremented or decremented by  $d$  (the dice power parameter). Player *Max* takes the first move and the game ends after  $T$  turns ( $2 \times T$  moves). At the end of the game, player *Max* is awarded  $c + a$  as reward (*Min* receiving  $-(c + a)$ ). For  $d > 0$  this game contains both elements of skill and elements of chance as its result is influenced by player

choices as well as random events. The trivial choice of moves in *PaCRaWa* serves to illustrate an important point. It is possible for agents to master the strategy of a game, whereupon the game may turn into a pure game of chance. Thus, the perceived nature of the game changes with the skill of the agents. This again justifies our belief that a game should be judged in reference to the agents.

We conduct sets of experiments with three different agents. Agent *X* plays perfectly, agent *Y* plays the correct move with probability 0.75 and the wrong move with probability 0.25. Agent *Z* plays the correct move with probability 0.5.

In a first set of experiments we test the effect of different agent combinations and dice powers on *chanciness*. Figure 6 generally shows a positive correlation between dice power and *chanciness*, which is consistent with our intuition regarding the influence of chance. The effect of different agents on *chanciness* is a bit more complicated. In the two-player matches, stronger agents experience more *chanciness*. This is to be expected, as the strong moves cancel each other out and gains can only be made with luck of the dice. When a strong agent faces a weak opponent the game has little *chanciness*, because the skill difference results in a definite score advantage for the stronger agent.

$(Max, Min) \setminus d$	0.1	0.5	1.0
$(X, X)$	0.97	0.98	0.98
$(Y, Y)$	0.11	0.37	0.53
$(Z, Z)$	0.09	0.33	0.47
$(X, Y)$	0.06	0.24	0.37
$(Y, Z)$	0.05	0.21	0.34
$(X, Z)$	0.04	0.16	0.27

**Figure 6:** *chanciness* of  $PaCRaWa[d, T = 10, a = 0]$  for different dice powers  $d$ , and different agents (1000 game instances per value,  $\sigma < 0.01$ ).

In a second set of experiments we test the effect of different game lengths  $T$  on *chanciness*. Figure 7 shows the *chanciness* for different game lengths and agents. We observe that games with an odd number of turns are asymmetric and give an advantage to player *Max*. If *Max* is played by a strong agent, that advantage translates into a high *match expectation* ME and thus reduced *chanciness*. ME also increases with game length, when the agents have differing skills.

$(Max, Min) \setminus T$	1	2	5	6	31	32
$(X, X)$	0.50	0.97	0.65	0.97	0.80	0.98
$(Y, Y)$	0.45	0.54	0.48	0.58	0.51	0.54
$(Z, Z)$	0.49	0.52	0.50	0.48	0.47	0.52
$(X, Y)$	0.50	0.45	0.37	0.41	0.29	0.30
$(Y, Z)$	0.45	0.38	0.36	0.36	0.27	0.27
$(X, Z)$	0.50	0.33	0.32	0.31	0.18	0.18

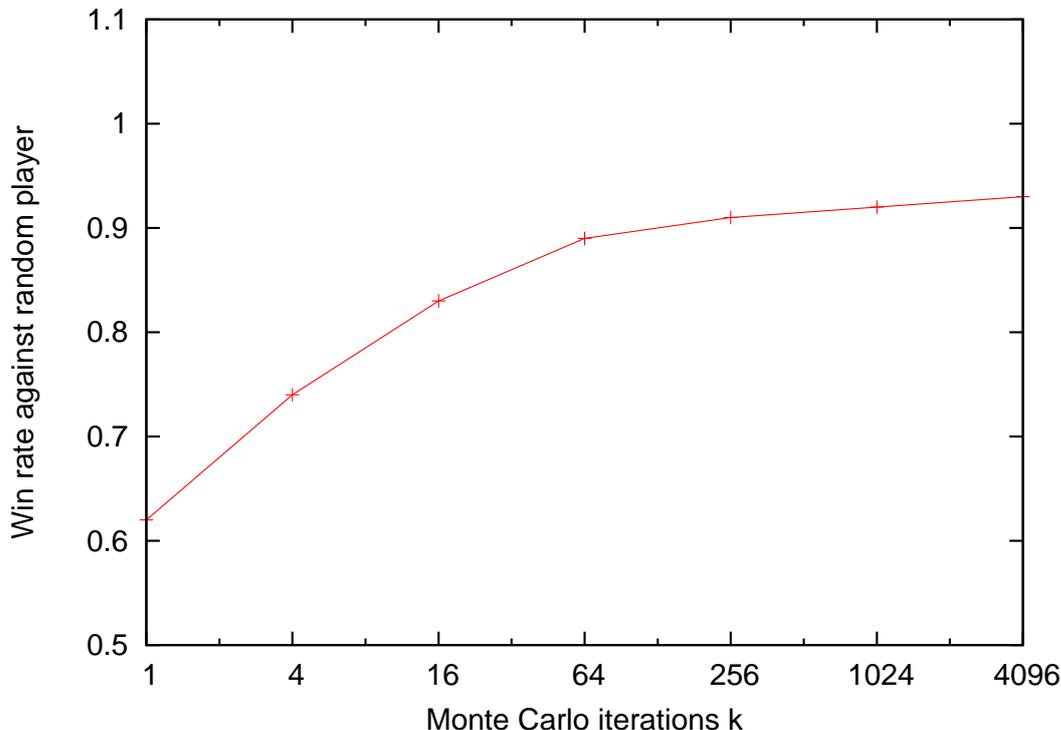
**Figure 7:** *chanciness* of  $PaCRaWa[d = 1, T, a = 0]$  for different game lengths  $T$  and various players  $\mathcal{P}$  (1000 game instances per value,  $\sigma < 0.01$ ).

The error in our approximations of *chanciness* is obvious in games between the agents  $(X, X)$  and even game lengths  $T$ . We would expect the player influence  $I_{\mathcal{P}}$  to be zero because *Max* and *Min* moves always cancel each other out. Likewise ME should be zero because both players have the same number of moves and the dice is fair. Consequently, *chanciness* should equal 1. However, in Figure 7 the highest value of *chanciness* is 0.98.

Even though *PaCRaWa* is an exceedingly simple game, it took 200 minutes to compute the data tables in this section (albeit with unoptimized code).

## 7.2 EinStein Würfelt Nicht

“EinStein würfelt nicht!” is a popular (Alden, 2004) board game designed by Ingo Althöfer (Althöfer, 2004). The title can be translated either as “A single stone does not play dice” or “Einstein does not play dice”. This play on words alludes to the game’s rules and is also a reference to Albert Einstein’s famous quotation about God and dice. The game is played by two players who rely on chance and skill to fulfill their objectives. “EinStein würfelt nicht!” can be played online for free at (Malaschitz, since 2002).



**Figure 8:** Win rates in EWN of Monte Carlo Players with different strengths versus an opponent who moves at random.

We introduce a game with slightly different rules and examine the effects of that rule change on *chanciness*. We will denote the original game as EWN and our variant as  $\overline{\text{EWN}}$ .

In EWN, players alternate in moving their pieces which are numbered from 1 to 6. Before a move can be undertaken, a six-sided dice is cast and the number on the dice indicates the piece that must be moved. During the course of the game pieces are removed from play. If the active player has lost piece  $i$  and the dice shows that number, the player may choose between the piece with the closest number to  $i$ , higher than  $i$  and the piece with the closest number lower than  $i$  (for detailed rules see (Althöfer, 2004)).

In our variant  $\overline{\text{EWN}}$ , this choice is abolished. The player always has to move the piece with the next available piece after  $i$  in the sequence  $(1, 2, \dots, 6, 1, 2, \dots)$ . Because of this rule change, some situations in  $\overline{\text{EWN}}$  leave the player with fewer move choices than in EWN.

For our experiments we have used agents that employ the *Monte Carlo* algorithm (Bruegmann, 1993; Gelly and Wang, 2006; Coulom, 2006) to select their moves. We denote these *Monte Carlo Players* as  $\text{MC}_k$ , with  $k$  indicating the number of Monte Carlo iterations. The playing strength in EWN grows with  $k$  as shown in Figure 8.

Figure 9 shows the *chanciness* of different player match-ups for the games EWN and  $\overline{\text{EWN}}$ . It can be seen that the *chanciness* is higher when the opponents are more skilled and closer to each other in strength. The *chanciness* of  $\overline{\text{EWN}}$  is consistently higher than the *chanciness* of the original game EWN. We propose that the lack of movement choices in our variant lessens the influence of the players and thus makes for a more chance-dependant game.

$(Max, Min) \setminus game$	EWN	$\overline{EWN}$	$\sigma$
$(MC_1, MC_1)$	0.51	0.57	0.028
$(MC_8, MC_8)$	0.62	0.68	0.025
$(MC_{64}, MC_{64})$	0.71	0.76	0.022
$(MC_{512}, MC_{512})$	0.79	0.84	0.015
$(MC_{512}, MC_{1024})$	0.82	0.85	0.013
$(MC_1, MC_8)$	0.44	0.53	0.023
$(MC_1, MC_{64})$	0.35	0.41	0.020
$(MC_1, MC_{512})$	0.30	0.40	0.017
$(MC_8, MC_1)$	0.43	0.47	0.021
$(MC_{64}, MC_1)$	0.33	0.35	0.018
$(MC_{512}, MC_1)$	0.24	0.32	0.017

**Figure 9:** *chanciness* of EWN and  $\overline{EWN}$  for different agents (100 game instances per value. For each row the larger standard deviation is given).

The least *chanciness* is found for matches between agents of greatly differing skill ( $MC_1, MC_{512}$ ). This can be explained by a high value for the *match expectation*  $ME$ . The table also shows a consistently higher *chanciness* for matches in which the weaker agent takes the first move rather than the second move. This can be explained by the different values of  $ME$  as explained in Subsection 5.1.

Each data point in Figure 9 is the result of 100 simulation games for the respective agents. The combined simulations required about a week of processing time. The high resource requirements when compared to our measurements for the game of PaCRaWa can be explained by the increased complexity of EWN and of the *Monte Carlo Players* (especially those with high  $k$ ).

## 8. RELEVANCE OF CHANCINESS FOR HUMAN AGENTS

In the preceding sections we have introduced our concept of *chanciness* for games played by computer agents. What relevance does this concept have for matches between humans?

This difficult question has philosophical and practical aspects. Is the expected result of a game between human agents and thus the *relative positional value* (Subsection 4.3) well defined? The answer to this philosophical question ultimately depends on one's interpretation of probability (Hájek, 2009).

One important interpretation is the *frequentist interpretation*. It defines probability as the relative frequency of an event for a large number of trials. A game between humans cannot be repeated because the agents change as they learn from previous games. For that reason our definition of *chanciness* is meaningless under a *frequentist* interpretation of probability.

On the other hand, the *Bayesian* interpretations allows probabilities for non-repeatable events. This gives meaning to *relative positional values* for human agents and hence to our definition of *chanciness*.

An equally pressing issue is the matter of computability of *chanciness* for human agents. Given a game  $G$  and a pair of human agents  $X, Y$ , can we discover the *chanciness*  $C(G, X, Y)$  with an algorithm that uses the results of matches involving  $X$  and  $Y$ ?

As in the case for computer agents we should not expect an exact calculation of *chanciness* for most games. We therefore rephrase the question: given a game  $G$  and a pair of human agents  $X, Y$ , can we approximate the *chanciness*  $C(G, X, Y)$  with an algorithm that uses the results of matches involving  $X$  and  $Y$ ?

As humans play much slower than computers, are less willing to repeat games and generally learn while they play, direct mass simulations (as with computer agents) can be ruled out. Instead, we suggest an indirect approach: if we had access to computer agents  $U, V$  that played very similar in style to our human agents  $X$  and  $Y$ , we

could approximate *chanciness* for these computer agents.  $C(G, U, V)$  would arguably be an approximation for  $C(G, X, Y)$ .

Of course, a general approximation of “style” is in itself a daunting task. If taken to the extreme it would be akin to passing the Turing Test (Turing, 1950). In fact in an earlier version of his famous test, Turing explicitly talks about comparing the playing style of humans and machines for the game of chess (Turing, 1947).

We conjecture that for some games it is sufficient to approximate the playing strength of human agents to approximate their style. We would consider an approximation of style sufficient, if matches between the computer agents  $U, V$  would cover similar regions of  $G$ 's game tree as matches between the humans  $X, Y$ . How this *similarity* should be defined is an open question. It should somehow involve the distribution of chance nodes and their potential influence. Games for which chance nodes are distributed homogeneously in the tree would then automatically fulfill the condition of match similarity.

When faced with such a homogeneous game  $G$ , we would measure the playing strengths of the humans  $X$  in  $G$  by conducting a limited number of matches between  $X$  and computer agents of different strengths. We then pick a computer agent  $U$  that matches the strength of  $X$  (and likewise a computer agent  $V$  that matches the strength of human  $Y$ ). We would then take  $C(G, U, V)$  as an approximation for  $C(G, X, Y)$ .

While we have no clear concept of homogeneity, we can construct games in which equal playing strength does not imply equal style (and equal *chanciness*). As an example, consider a hybrid of chess and backgammon: The first move that is taken is a decision whether chess or backgammon should be played. Obviously the tree of this game is not homogeneous as it has a chess subtree and a backgammon subtree which are quite different in their distribution of chance nodes.

The move of the first player decides whether the chess- or the backgammon-subtree is used. Given agents with the right combination of skills, this first move decision is independent of playing strength and only a matter of “style”. Yet, the first decision determines whether the influence of chance for the match will be zero (chess) or non-zero (backgammon). Fortunately, common games appear to be for more homogeneous in this regard.

Eventually we wish to use *chanciness* to automatically compare games that are played by larger groups of agents (e.g. buyers of board games). This forces us to generalize from particular agents to larger groups. We point to a list of games that were played by general<sup>7</sup> computer agents (based on Monte-Carlo methods).

**Würfelingo** (Wueppen, 2007) Agents match the playing strength of average human players. Group trials by the author indicate indistinguishability from equally skilled humans.

**EinStein würfelt nicht** Agents match the playing strength of average human players. Monte Carlo agents played against humans on the game server [www.inetplay.de](http://www.inetplay.de) from 2005 to 2008. User feedback indicates indistinguishability from human style.

**Gelb Voran** (Althöfer, 2005) Agents match the playing strength of advanced human players. Personal experience by the author suggests indistinguishability in style from human play.

**Seasons of the Sun** (Althöfer, 2007b) Agents match the playing strength of advanced human players. Reports from various players indicate indistinguishability in style from human play.

**Hex** (Hein and Nash, 1942) Agents play considerably below human levels. They displayed on tendency to play obvious losing moves that only prolonged the match (“chains of resignation” (Guenther, 2008)). This assessment only refers to a plain UCT algorithm. Various heuristics have been successfully combined with UCT (MoH, ).

**Big Balls** (Althöfer, 2007a) In group trials by the author, agents played below average human levels. This allows distinguishing playing style for a large group of humans.

Based on these examples, approximating the playing style of casual game players with general game playing agents of sufficient strength appears plausible to us.

Therefore, we think that the concept of *chanciness* is useful in *computer-aided game inventing*. Althöfer lists several game features that are amenable to automatic evaluation: game length (average and maximum), drawing quota, balancedness, and deepening positivity (the impact of search depth on playing strength) (Althöfer,

<sup>7</sup>Not tuned to the specific properties of a game.

2003). Browne adds further features such as rule complexity, convergence, and drama<sup>8</sup> (Browne, 2008). We add *chanciness* to this list of features for automatic game evaluation.

## 9. CONCLUSIONS AND OPEN PROBLEMS

We have defined the *chanciness* of a non-deterministic<sup>9</sup> perfect-information zero-sum game in extensive form that is played between two specific agents. This definition is based on the *relative influence of chance* for specific matches of that game by those same agents. We have approximated those *chanciness*-values for exemplary games played by computer agents.

The *chanciness* of a game  $G$  for some agents  $X, Y$  is a real number from the interval  $[0, 1]$ . It is close to 1 if the influence of chance on the outcome of the game is high. In this case,  $G$  played by  $X$  and  $Y$  is a game of chance. The *chanciness* is low or close to zero for games where chance has little influence on the game result. In that case,  $G$  played by  $X$  and  $Y$  is a game of skill. The *relative influence of chance* can be used to analyze specific matches of a game. If the *relative influence of chance* is low, the match result should be attributed to the skill difference of the agents.

A necessary condition for the calculation of *chanciness* is the (frequent) repeatability of games between the agents. This does not exclude the use of learning agents as long as those agents can be reset to a previous internal state. As human agents can neither be reset nor do they tolerate many repeated games, *chanciness* can not be computed directly for human agents.

For computer-aided game development, the existence of repeatable automatic players can be assumed. The calculation of *chanciness* aids in the process of game inventing when adjusting a game's position on the skill-chance continuum. If, for example a game inventor wishes to reduce the influence of chance in one of his invented games he can compute the *chanciness* for various rule variations and some set of agents. He can then discard those rule variations which do not lower the *chanciness* value.

An open problem of our method is the resource requirement for approximating *relative positional values*. The Algorithm described in Appendix B might be further improved through the use of hash tables. Measuring the variance during approximation would allow us to cut off some simulations while maintaining overall accuracy.

*chanciness* can be calculated for one-player games (puzzles) by modeling them as zero-sum games against a passive opponent. Additional research is undertaken to generalize the concept of *chanciness* to other game models:

- **Games with more than two players** These games require a vectorial representation of positional values and influence. Furthermore, these games may force an agent to select among moves which are of equal value for him but differ in their influence on his opponents. It may be appropriate to attribute such influence to chance.
- **Non-zero-sum games** These games can be mapped onto zero-sum games with an additional player and present all the challenges of games with more than two players.
- **Games in normal form (matrix games)** Players typically employ randomized (mixed) strategies. This gives an additional source of chance which has to be taken into account.
- **Sequential games with imperfect information** Players have to make educated guesses regarding the game state they are in. The unpredictability of some game state aspects is from their perspective a form of randomness. This should be reflected in a generalization of *chanciness*.

As we continue to explore these questions, we expect that the concept of *relative positional values* will be useful as it is easily applied to a large class of games with any number of players.

<sup>8</sup>*Convergence* measures the trend in the number of (good) move choices as a match progresses. *Drama* measures how well agents can recover from self-perceived disadvantageous positions.

<sup>9</sup>Deterministic games are a special case of non-deterministic games (Nowakowski, 1996). Their *chanciness* is always zero.

## 10. REFERENCES

- Alden, S. (2004). "EinStein würfelt nicht!" on boardgamegeek.com. [www.boardgamegeek.com/boardgame/18699](http://www.boardgamegeek.com/boardgame/18699). accessed 2009/07/21.
- Alon, N. (2007). Poker, Chance and Skill. [www.cs.tau.ac.il/~nogaa/PDFS/publications.html](http://www.cs.tau.ac.il/~nogaa/PDFS/publications.html). accessed 2009/01/19.
- Althöfer, I. (2003). Computer-Aided Game Inventing. [www.minet.uni-jena.de/preprints/althoef03/](http://www.minet.uni-jena.de/preprints/althoef03/). [accessed 2009/01/19].
- Althöfer, I. (2004). Board game "EinStein würfelt nicht". [www.en.wikipedia.org/wiki/EinStein\\_würfelt\\_nicht](http://www.en.wikipedia.org/wiki/EinStein_würfelt_nicht). 3-Hirn-Verlag [[www.3-hirn-verlag.de/games.html](http://www.3-hirn-verlag.de/games.html)].
- Althöfer, I. (2005). Board game "Gelb Voran". 3-Hirn-Verlag. Description at [[www.ludorium.at/SPIELE/0\\_english\\_version/3\\_hirn/gelb\\_voran\\_e.htm](http://www.ludorium.at/SPIELE/0_english_version/3_hirn/gelb_voran_e.htm)] accessed 2009/05/26.
- Althöfer, I. (2007a). Big Balls. 3-Hirn-Verlag [[www.3-hirn-verlag.de/games.html](http://www.3-hirn-verlag.de/games.html)].
- Althöfer, I. (2007b). Board game Seasons of the Sun. 3-Hirn-Verlag [[www.althofer.de](http://www.althofer.de)]. playable online at [[www.devilplay.de/sots/sots.html](http://www.devilplay.de/sots/sots.html)].
- Billings, D., Davidson, A., and Szafron, D. (2002). The challenge of poker. *Artificial Intelligence*, Vol. 134, pp. 201–240.
- Borm, P. and Genugten, B. van der (1997). On a relative measure of skill for games with chance elements. *Journal of the Spanish Society of Statistics and Operations Research*, Vol. 9, No. 1, pp. 91–114.
- Browne, C. (2008). *Automatic Generation and Evaluation of Recombination Games*. Ph.D. thesis, Queensland University of Technology.
- Bruegmann, B. (1993). Monte Carlo Go. [www.ideaenest.com/vegos/MonteCarloGo.pdf](http://www.ideaenest.com/vegos/MonteCarloGo.pdf). Unpublished technical report.
- Burton, E. (2007). North Carolina Appeals Court Says Poker 'Game Of Chance'. [www.pokernews.com/news/2007/5/north-carolina-appeals-game-chance.htm](http://www.pokernews.com/news/2007/5/north-carolina-appeals-game-chance.htm). accessed 2009/01/19.
- Coulom, R. (2006). Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. *Proceedings of the 5th International Conference on Computer and Games* (eds. J. H. van den Herik, P. Ciancarini, and J. H. H. L. M. Donkers), Vol. 4630/2007 of *Lecture Notes in Computer Science*, pp. 72–83, Springer. <http://remi.coulom.free.fr/CG2006/CG2006.pdf>.
- Cypra, D. (2009). Pennsylvania Judge Rules Poker is a Game of Skill. [www.pokernewsdaily.com/pennsylvania-judge-rules-poker-is-a-game-of-skill-941](http://www.pokernewsdaily.com/pennsylvania-judge-rules-poker-is-a-game-of-skill-941). accessed 2009/01/19.
- Dreef, M. (2005). *Skill and Strategy in Games*. Ph.D. thesis, Tilburg University, Faculty of Econometrics and Operations Research.
- Dreef, M., Borm, P., and Genugten, B. van der (2004). Measuring skill in games: several approaches discussed. *Mathematical Methods of Operations Research*, Vol. 59, No. 3, pp. 375–391.
- (2008). FIDE Laws of Chess. [www.fide.com](http://www.fide.com).
- Finnsson, H. and Björnsson, Y. (2008). Simulation-Based Approach to General Game Playing. *AAAI*, Vol. 23, pp. 259–264.
- Gelly, S. and Wang, Y. (2006). Exploration exploitation in Go: UCT for Monte-Carlo Go. *NIPS: Neural Information Processing Systems Conference On-line trading of Exploration and Exploitation Workshop*. MIT Press.
- Guenther, J. (2008). Evaluation of Monte-Carlo agents for the board game "Hex". unpublished private correspondence.
- Hauk, T., Buro, M., and Schaeffer, J. (2006). Rediscovering \*-MINIMAX Search. *Lecture Notes in Computer Science*, Vol. 3846, pp. 35–50. Springer Berlin, Heidelberg.

Hayward, R. (2009). MoHex wins at the 14th Computer Olympiad. <http://www.cs.ualberta.ca/~hayward/hex/>. accessed 2009/07/16.

Hein, P. and Nash, J. (1942). Hex. [en.wikipedia.org/wiki/Hex\\_\(board\\_game\)](http://en.wikipedia.org/wiki/Hex_(board_game)).

Hájek, A. (2009). Interpretations of Probability. *The Stanford Encyclopedia of Philosophy* (ed. E. N. Zalta). The Metaphysics Research Lab, Stanford University, spring 2009 edition.

Love, N., Hinrichs, T., Haley, D., Schkufza, E., and Genesereth, M. (2008). General Game Playing: Game Description Language Specification. Technical report, Stanford University.

Malaschitz, R. (since 2002). Little Golem Online Board Games. [www.littlegolem.net](http://www.littlegolem.net).

Neumann, J. von (1927). Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, Vol. 100, pp. 295–320.

Nowakowski, R. J. (1996). *Games of No Chance*. Cambridge University Press. ISBN 9780521646529.

Pell, B. (1994). A strategic metagame player for general chess-like games. *AAAI'94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 2)*, pp. 1378–1385, American Association for Artificial Intelligence, Menlo Park, CA, USA.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, Cambridge. ISBN 978-0521899437.

(2009). Analysis of Backgammon matches by the Snowie engine. [www.bgsnowie.com/backgammon/matches.dhtml](http://www.bgsnowie.com/backgammon/matches.dhtml). accessed 2009/02/23.

Staatsvertrag (2007). Staatsvertrag zum Glücksspielwesen in Deutschland (GlüStV). Hamburgisches Gesetz- und Verordnungsblatt.

Turing, A. (1947). *Intelligent Machinery*. Oxford University Press. ISBN 978-0408566506.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, Vol. 59, pp. 433–460.

Wueppen, H. (2007). Board game "Würfel Bingo". Ravensburger, item number: 27131.

Zermelo, E. (1913). Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. *Proceedings of the Fifth International Congress of Mathematicians*, Vol. 2, pp. 501–504. English translation at [[www.math.harvard.edu/~elkies/FS23j.03/zermelo.pdf](http://www.math.harvard.edu/~elkies/FS23j.03/zermelo.pdf)] accessed 2009/04/22.

## 11. APPENDICES

### APPENDIX A: LOTTERY CHESS

*Lottery chess* is an artificially designed class of games to reveal inconsistency in the notion of *Relative Skill* (Dreef, 2005). *lottery chess* has two parameters  $n, k \in \mathbb{N}$ . There exist realizations of the parameters  $n, k$  for which the resulting games receive a rating of *Relative Skill* which in our opinion contradicts common sense on the notions of skill and chance.

*Lottery chess* is almost identical to the classical game of *chess*. As the only difference, each player secretly selects an integer from  $[0, 1, \dots, n]$  before the start of the chess match. The winner of the match receives a score of 1. Additionally a lottery takes part. A random integer from  $[0, 1, \dots, n]$  is drawn. Should a player have guessed that number he receives an additional score of  $k$ .

The influence of chance on the game outcome can be made arbitrarily small by increasing  $n$ . As  $n$  grows, the likelihood that one of the players correctly guesses the lottery number decreases to zero. At the same time the *Relative skill* (in the sense of Dreef) can be made arbitrarily small by increasing  $k$ . This follows because the fictive player *always* wins the lottery no matter its odds. As  $k$  grows this advantage of the fictive player overshadows any advantage of being good at chess.

By choosing large values for  $n$  and  $k$  (such as  $n = 10^6, k = 10^3$ ) we can create a game that has a *Relative Skill* rating near zero. Equation 1 gives

$$\begin{aligned} RS &= \frac{\text{gain}_{\text{optimal}} - \text{gain}_{\text{beginner}}}{\text{gain}_{\text{fictive}} - \text{gain}_{\text{beginner}}} \\ &= \frac{1.001 - 0.001}{1000001 - 0.001} < 10^{-6} \end{aligned}$$

However, the result of that game when played by any human or algorithm will be mostly determined by skill at chess and will therefore be a game of skill. As a game cannot be a game of skill and a game of no skill at the same time, the definition of *Relative Skill* is contradictory.

It must be noted that this extreme behavior is highly unlikely to show up in a real game played by real people.

## APPENDIX B: LOOKUP SAMPLING

In Subsection 4.3 we introduced *relative positional values* and in Subsection 4.4 we explained how to approximate them via Monte-Carlo Sampling. The efficiency of that algorithm can be improved when calculating *relative positional values* for consecutive game states of a match. We call that improved algorithm *Lookup Sampling*.

Let  $g = (X, Y, (s_0, \dots, s_n))$  be a match of game  $G$  between the agents  $X$  and  $Y$ . To calculate the *relative influence of chance* of  $g$ , we need the values  $\tilde{V}(s_i, X, Y)$  for all  $s_i$  in  $g$ .

According to Equation 5 we can calculate the  $\tilde{V}(s_i, X, Y)$ , given the *relative positional values* for all child nodes of  $s_i$  and the probabilities of all child nodes to be selected in a match between  $X$  and  $Y$ . *Lookup Sampling* exploits this Equation to speed up our sampling algorithm.

We approximate the *relative positional values* in the backward sequence for  $(s_n, s_{n-1}, \dots, s_0)$ . When approximating  $\tilde{V}(s_i, X, Y)$  we start a simulation match and sample a child node  $s'_i$  for  $s_i$  (by querying an agent or simulating a chance move according to the rules of  $G$ ). If that child node  $s'_i$  is identical to the node  $s_{i+1}$  from  $g$  we immediately return  $\tilde{V}(s_{i+1}, X, Y)$  as the value of that simulation match. Otherwise we continue the simulation match until a leaf node is reached. The average result of all simulation matches is then used as our approximation of  $\tilde{V}(s_i, X, Y)$ .

Because the comparison of  $s'_i$  and  $s_{i+1}$  can be accomplished in constant time (via Hashing) we can save time by stopping our simulation early. This total gain in computing speed depends on the probability of  $s_{i+1}$  to be sampled. If  $(s_i, s_{i+1})$  happens to be a “deterministic move” (Subsection 4.5) the sampling for  $\tilde{V}(s_i, X, Y)$  always stops after the first move.

Moreover *Lookup Sampling* improves the accuracy of influence calculations. Let  $m = (s_i, s_{i+1})$  be a “deterministic move” by agent  $X$ . According to Lemma 8 the influence of  $m$  is zero. With regular Monte-Carlo Sampling the influence of  $m$  is the difference of two approximated *relative positional values*. It is very unlikely that both approximations return the same value and thus yield an influence of zero. With *Lookup Sampling*,  $\tilde{V}(s_i, X, Y)$  equals  $\tilde{V}(s_{i+1}, X, Y)$  exactly, because every simulation match for  $s_i$  returns  $\tilde{V}(s_{i+1}, X, Y)$  as its result. Even though  $\tilde{V}(s_{i+1}, X, Y)$  is only an approximation, the influence of  $m$  is calculated as zero.

The game PaCRaWa[ $d = 1, T = 10, a = 0$ ] (Subsection 7.1) played by two optimal (and deterministic) agents  $X$  is fair (ME = 0). Because chance generally has an influence on the match result and the deterministic agents have no influence at all, the *chanciness*  $C(\text{PaCRaWa}[d = 1, T = 10, a = 0], X, X)$  is 1. Given 1000 samples per game state and 100 matches, regular Monte-Carlo Sampling approximates the *chanciness* as 0.88 while *Lookup Sampling* achieves an approximation of 0.98.