

# 不確定性を含むデジタルカーリングにおけるゲーム木探索

加藤 修<sup>1,a)</sup> 飯塚 博幸<sup>1,b)</sup> 山本 雅人<sup>1,c)</sup>

受付日 2016年2月20日, 採録日 2016年9月6日

**概要:** 二人零和有限確定完全情報ゲームにおいてすでに AI は人間に匹敵する強さとなっており, 最近では麻雀や人狼など, 多人数ゲーム, 不完全情報ゲーム, 不確定ゲームが新たな研究対象として注目を集めている. そのような中で不確定ゲームとしてデジタルカーリングがある. デジタルカーリングはカーリングの二人用コンピュータゲームであり, AI どうしを競わせカーリングの戦略を解析することを目的とした不確定ゲームのテストベッドとして開発された. 本研究ではデジタルカーリングにボードゲームの探索手法を適用し, 投球目標座標と回転方向を候補手, 盤面状態を局面としてデジタルカーリングに Expectimax によるゲーム木探索を適用する手法を提案する. Expectimax ではゲーム木内で確率的に推移するノードを用いており, ゲームの不確定性を考慮した探索が可能となっている. 提案手法の有効性を検証するため, 不確定性を考慮する場合としない場合それぞれにおいて探索の深さを変化させ既存 AI との対戦実験を行った. その結果, 提案手法による不確定性の考慮を行った場合に勝率が上昇し, また不確定性を考慮した場合のみ探索の深さを増やすことで勝率が上昇したことから, デジタルカーリングにおける提案手法による不確定性を考慮した先読みの有効性が明らかとなった.

キーワード: デジタルカーリング, 不確定ゲーム, ゲーム木探索, Expectimax

## A Method of Game Tree Search in Digital Curling Including Uncertainty

SHU KATO<sup>1,a)</sup> HIROYUKI IIZUKA<sup>1,b)</sup> MASAHITO YAMAMOTO<sup>1,c)</sup>

Received: February 20, 2016, Accepted: September 6, 2016

**Abstract:** Computer AI game players have beaten human expert players in two players zero-sum perfect information games such as Go and Japanese Chess and the focus is moving to multiple players, imperfect information games and even with uncertainty such as Mah-jong and Werewolf. Digital curling is one of the games with uncertainty. It has been developed as a testbed of the games with uncertainty to compete computer AI programs and to analyze the curling strategies of human players. This paper proposes an expectimax game tree search method for the digital curling where candidate moves are represented as the combination of the throwing-target position and rotational direction of stones and the discretized board states are nodes of the tree. The method can take into account uncertainty using stochastic transition in the game tree. To evaluate our proposed method, we compare the winning rates of the methods with and without the stochasticity while changing the searching depth. Our results show that our proposed method with the stochasticity is better than without the stochasticity and can only show the effective look ahead. From those results, it is shown that our proposed method could handle uncertainty and perform effective tree search on digital curling.

**Keywords:** digital curling, stochastic game, game tree, expectimax

<sup>1</sup> 北海道大学大学院情報科学研究科  
Graduate School of Information Science and Technology  
Hokkaido University, Sapporo, Hokkaido 060-0814, Japan

a) shu.kato@complex.ist.hokudai.ac.jp

b) iizuka@complex.ist.hokudai.ac.jp

c) masahito@complex.ist.hokudai.ac.jp

### 1. はじめに

チェスや将棋などのボードゲームにおけるコンピュータ AI の開発・研究は昔からさかんに行われている. 1997 年には IBM の開発したディーブブルーがチェスの世界チャ

ンピオンに勝利した [1]. また, 2016 年 1 月にはディープラーニングを用いた Google の “AlphaGo” が囲碁のプロ棋士に勝利したと報じられ話題を集めている [2]. このように二人零和有限確定完全情報ゲームにおいてすでに AI は人間に匹敵する強さとなっている.

一方で, 最近では麻雀や人狼など, 多人数ゲーム, 不完全情報ゲーム, 不確定ゲームが新たな研究対象として注目を集めている [3], [4]. そのような中で, 不確定ゲームとしてビリヤードなどの現実世界のスポーツを題材としたコンピュータゲームが存在し [5], [6], デジタルカーリングもその 1 つである. カーリングは戦略性の高いスポーツであるが [7], 戦略の決定には選手の力量やリンクに張られた氷の状態など様々な要因による誤差が発生するため, これまでカーリング戦略に関する科学的な研究はあまりされていない. デジタルカーリングはカーリングにおける戦略のみを切り出し議論するために電気通信大学の伊藤らのグループが開発したカーリングの二人用コンピュータゲームである [8], [9]. デジタルカーリングでは, プレイヤが交互に投球する石の初速度と回転方向を入力し石を投球することでゲームが進行する. この際, プレイヤの入力に乱数を加えることで結果に不確定性を発生させている. これは現実世界のカーリングにおける氷の状態などの不確定な要素を総合的に再現しており, この点がデジタルカーリングの大きな特徴である.

過去には公式大会が開催されており, 2015 年 9 月に台湾で行われたデジタルカーリングの世界大会「IEEE-CIG 2015 ミニ大会」では著者らの開発した AI「じりつくん」が優勝を果たしている [10]. また国内大会では 2015 年 11 月の「GPW 杯」においてモンテカルロ木探索をベースとした AI「歩」が優勝し, 同月の「第 2 回デジタルカーリング大会」では「歩」をベースとして序盤にカーリング戦略の知識を組み込んだ AI「GCCS」が優勝している.

本論文の目的は, 不確定ゲームであるデジタルカーリングにおいて著者らの開発した AI「じりつくん」による探索の先読みの有効性を明らかにすることである. 「じりつくん」はデジタルカーリングにおいて Expectimax によるゲーム木探索を行う [11]. ここで, デジタルカーリングへのゲーム木探索の適用にあたり, 2 つの解決すべき点がある. 1 つは, 候補手である投球目標座標が二次元直交座標系の実数値ベクトルであり, 候補手が無限に存在することである. このため実時間内に探索を終わらせるためには, 候補手を有限個に絞る必要がある. 2 つ目はプレイヤがデジタルカーリングへ投球する石の初速度と回転方向を入力する際, 石の初速度に実数値の乱数が加えられることである. これにより 1 つの候補手から生成される結果の局面は毎回異なるため, 想定するノードは無限に存在し, 通常のゲーム木探索は適用できない. 本論文ではこれら 2 点を解決し, 提案手法の有効性を検証するため,

不確定性を考慮する場合としない場合それぞれにおいて探索の深さを変化させ既存 AI との対戦実験を行い結果を議論する.

## 2. カーリング

カーリングは 2 チームが交互にストーンを氷上で滑らせ, ハウスとよばれる円状領域の中心に近い場所を確保し合うことで得点を競うスポーツである. 自分のストーンを相手にはじき出されないようにしつつハウス中央に近い位置を確保するという点において, 非常に高い戦略性が存在する. デジタルカーリングは, コンピュータ内で実行する物理シミュレーションによってカーリングの試合を再現したもので, プレイヤからの入力に従って投球シミュレーションを行い投球後の局面を出力し, 両チームのプレイヤの投球を繰り返し, ゲームが進行する. なお, 現段階では単純化のためブラシで石の滑る先を磨くスウィーピングについては考慮されておらず, また氷の摩擦係数は一定である.

本章ではカーリングのルールを簡単に説明し, その後デジタルカーリングの仕組みについて述べる.

### 2.1 カーリングのルール

#### 2.1.1 試合の流れ

カーリングはエンドとよばれる部分ゲームを繰り返すことでゲームが進行する. 各エンドでは両チーム 8 投ずつストーンを交互に投球し, 計 16 個の石の投球後に得点計算を行う. 得点計算では, 相手チームの石の中で最もハウス中央に近い石より内側にある自チームの石の個数が得点となる. そのため片方のチームは必ず 0 点となる. 図 1 の例では, 黄チームが 2 点, 赤チームが 0 点となる. ただし, ハウスの外にある石は得点計算の対象外となり, ハウス内に石が 1 つも存在しない場合は両チームともに 0 点となる. 通常の大会では 10 エンド行い, 総得点によって勝敗を決定する.

#### 2.1.2 手番の決定

カーリングでは得点計算直前に投球を行う後攻が非常に有利であり, 各エンドにおいて先攻か後攻かという手番は非常に重要となる. 1 エンド目の手番はじゃんけんや抽選などにより決定し, 2 エンド目以降は前エンドで得点したチームが先攻となる. 前エンドの得点が両チーム 0 点の場合には手番を交代せずに次のエンドを行う. よって, 状況

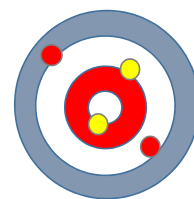


図 1 得点計算の例 (黄チーム 2 点, 赤チーム 0 点)

Fig. 1 Example of scoring (yellow 2 points, red 0 point).

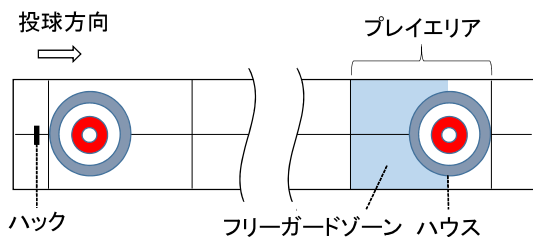


図 2 カーリングリンク  
Fig. 2 Curling Rink.

によっては後攻で 1 点取るよりもハウス内のストーンをすべてはじき出し意図的に 0 点にすることで次のエンドでの後攻権を得るといった戦略も存在する。

### 2.1.3 投球

プレイヤーは踏み台（ハック）から踏み出し、プレイエリアに向かってストーンを投球する（図 2）。代表的な投球方法として、投球したストーンをプレイエリア内部に止めるドロウとプレイエリア内にあるストーンをはじき出すテイクアウトがある。

### 2.1.4 フリーガードゾーンルール

フリーガードゾーンとは図 2 の水色で示される領域であり、フリーガードゾーンルールとは後攻の 2 投目までフリーガードゾーン内にある相手のストーンをテイクアウトしてはならないというルールである。フリーガードゾーン内の相手のストーンをテイクアウトした場合は、相手のストーンは元の位置に戻し、自分のストーンはプレイエリアから取り除かれる。序盤でフリーガードゾーンにストーンを溜めた場合、盤上のストーン配置が複雑になり本来不利な先攻側が得点する可能性や後攻側が大量点を得る可能性が高くなることが知られている。

### 2.1.5 用語まとめ

- エンド：先攻後攻互いに 8 投ずつ投球し、その後得点計算を行う部分ゲーム
- ドロウ：ストーンをプレイエリア内に止める投球
- テイクアウト：盤上にあるストーンをはじき出す投球

## 2.2 デジタルカーリング

デジタルカーリングは二人用のカーリングコンピュータゲームである。ルールや試合進行はすべて実際のカーリングに沿って行われるが、ストーンの投球のみコンピュータ内のシミュレーションにより行う。デジタルカーリングにおける投球のシミュレーションは、物理シミュレータおよび乱数生成器により実現されている。

物理シミュレータは投球するストーンの初速度  $\mathbf{v}$ 、回転  $R$ 、さらに現在の局面  $s_t$  を入力とし、投球後のストーンの運動および衝突に関する物理演算を行い、投球後の局面  $s_{t+1}$  を一意に生成する（式 (1)）。

$$s_{t+1} = f(\mathbf{v}, R, s_t) \quad (1)$$

$\mathbf{v}$  は二次元直交座標系の実数値ベクトル、 $R$  は右回転  $R_R$ 、左回転  $R_L$  の 2 値、回転の速さは一定値をとる。 $s_t$  は現在までに行われた投球数、現在の手番、現在のエンド数、全エンド数、各エンドの得点、各ストーンの座標を情報として持つ。なお、任意の  $R$  において  $\mathbf{v}$  と、衝突が起これなかった場合にストーンが到達する座標  $\mathbf{q}$ （すなわち、投球目標座標）は 1 対 1 で対応しており、以下のように変換可能である。

$$(\mathbf{v}, R) = g(\mathbf{q}, R) \quad (2)$$

$$(\mathbf{q}, R) = g^{-1}(\mathbf{v}, R) \quad (3)$$

乱数生成器はプレイヤーがデジタルカーリングに入力する投球するストーンの初速度  $\mathbf{v}_m$  へ加算する二次元確率ベクトル  $\mathbf{v}_r$ （以降、誤差ベクトルとよぶ）を生成する。これにより現実世界のカーリングにおける氷の状態などの不確定な要素を総合的に再現しており、同一の  $\mathbf{v}_m$  から異なる結果が生じうる。 $\mathbf{v}_r$  生成の手順は以下のとおりである。

- (1) 平均 0 標準偏差  $\sigma_x, \sigma_y$  の正規分布に従う確率変数  $r_x, r_y$  を要素とする二次元確率ベクトル  $\mathbf{r}$  を生成。
- (2) ハウス中央の座標  $\mathbf{p}_h$  を投球目標座標とする初速度  $\mathbf{v}_h$  を算出。

$$(\mathbf{v}_h, R) = g^{-1}(\mathbf{p}_h, R) \quad (4)$$

- (3)  $\mathbf{p}_h$  から  $\mathbf{r}$  ずれた点を投球目標座標とする初速度  $\mathbf{v}'_h$  を算出。

$$(\mathbf{v}'_h, R) = g^{-1}((\mathbf{p}_h + \mathbf{r}), R) \quad (5)$$

- (4)  $\mathbf{v}'_h$  と  $\mathbf{v}_h$  の差分ベクトルを誤差ベクトル  $\mathbf{v}_r$  とする

$$\mathbf{v}_r = \mathbf{v}'_h - \mathbf{v}_h \quad (6)$$

実際のゲームにおける投球シミュレーションの流れは、プレイヤーがデジタルカーリングに投球するストーンの初速度  $\mathbf{v}_m$  と回転  $R_m$  を入力した後、乱数生成器により生成した  $\mathbf{v}_r$  を  $\mathbf{v}_m$  へ加算し、物理シミュレータへ入力することで次の局面  $s_{t+1}$  を生成する（式 (7)）。

$$s_{t+1} = f((\mathbf{v}_m + \mathbf{v}_r), R, s_t) \quad (7)$$

$\mathbf{v}_m$  がハウス中央を投球目標座標とする初速度である場合、 $g(\mathbf{v}_m, R_m)$  により得られる投球目標座標  $\mathbf{q}_m$ （ハウス中央の点）に対する  $g((\mathbf{v}_m + \mathbf{v}_r), R_m)$  により得られる投球目標座標  $\mathbf{q}'_m$  のずれ  $\mathbf{q}'_m - \mathbf{q}_m$  は  $\mathbf{r}$  に従う。 $\mathbf{v}_m$  がハウス中央以外を投球目標座標とする初速度である場合もほぼ同様となる。なお、 $\mathbf{r}$  の各要素  $r_x, r_y$  が従う正規分布の標準偏差  $\sigma_x, \sigma_y$  は、公式大会においては事前に公開されており、既知の情報として利用可能である。2015 年 11 月 22 日に行われた第 2 回 UEC 杯までの公式大会では  $\sigma_x, \sigma_y$  はともにストーンの半径  $0.145m$  である。

また、デジタルカーリングにおける座標系は図 3 中の

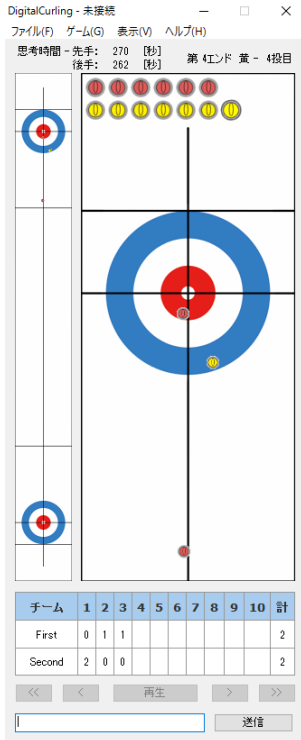


図 3 デジタルカーリングの画面  
Fig. 3 Snapshot of digital curling.

カーリングリンクの左上端を原点とし、水平方向を  $x$  軸、垂直方向を  $y$  軸としている。  $x$  軸は右向き、  $y$  軸は下向きが正の方向である。

### 3. 提案手法におけるゲーム木探索

本論文では、デジタルカーリングに Expectimax を適用し、ストーン配置や残り投球数などの盤面状態を局面、投球目標座標  $q$  と回転  $R$  を候補手、候補手実行時の確率的な分岐点をチャンスノードとしたゲーム木探索を行う。 Expectimax とは、通常の Minimax 法に対しチャンスノードとよばれる確率的な分岐点を導入し、不確定ゲームにおいて Minimax 探索を行う手法である。チャンスノードの評価値は、チャンスノードから展開される各子ノードの生起確率と評価値の積を足し合わせたものとなる。デジタルカーリングでは、投球目標座標  $q$  が連続値であり、候補手  $m$  が無限に存在する。またプレイヤーの決定した候補手には連続値の乱数が加算されるため、1つの候補手から生成される乱数加算後の候補手（以降、実行手  $m'$  とよぶ。チャンスノードからの分岐1つ分に相当する）も無限に存在する。そのため、実時間内に探索を終了させるためには、  $m$  と  $m'$  を有限個で代表させる必要がある。本論文ではそれらの集合を各々、代表候補手集合  $M$ 、代表実行手集合  $M'(m)$  とよぶ（図 4）。

また、通常の Minimax 法と同様、末端ノードの評価には局面の有利不利を定量化する局面価値関数を使用する。

本章で述べる内容は以下のとおりである。

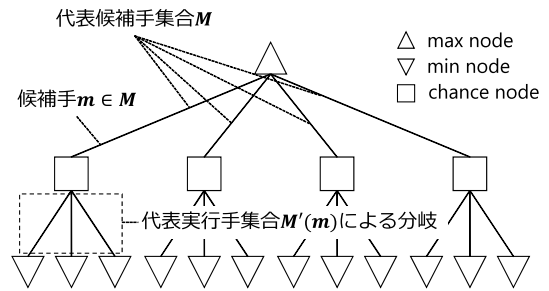


図 4 Expectimax におけるゲーム木  
Fig. 4 Game tree of Expectimax.

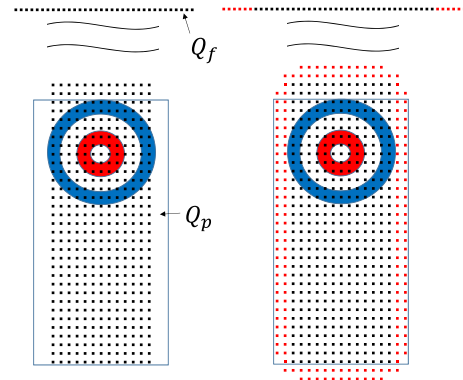


図 5 左図：投球目標座標集合  $Q = Q_p + Q_f$  の模式図  
右図： $Q$  を拡張して生成した  $U_q$  の模式図

Fig. 5 Left: Discretized target positions for delivering stones.  $Q_p$  is mainly for draw shots and  $Q_f$  is for take-out shots.  
Right: Discretized possible result positions,  $U_q$ , by candidate moves because of noise.

- (1) 代表候補手集合  $M$  の生成
- (2) 代表実行手集合  $M'(m)$  の生成
- (3) チャンスノードの評価値計算
- (4) 局面評価関数

#### 3.1 代表候補手集合 $M$ の生成

代表候補手集合  $M$  の生成にあたっては、まず投球目標座標  $q$  を一定範囲内の領域で等間隔に離散化した投球目標座標集合  $Q$  を回転  $R$  ごとに生成する（式 (8)）。  $Q$  は、初速の小さい候補手の生成を想定した投球目標座標集合  $Q_p$  と、初速の大きい候補手の生成を想定した投球目標座標集合  $Q_f$  の 2 つの和集合である（図 5 左図）。  $Q_p$  はプレイヤー周辺を投球目標座標とし（式 (9)）、  $Q_f$  はプレイヤー遠方の投球位置から遠い座標の横 1 列を投球目標座標とする（式 (10), (11)）。投球開始位置から投球目標座標までの軌道間にストーンが存在する場合、テイクアウトにもなりうる。  $Q_p$  からはドローとテイクアウト、  $Q_f$  からはテイクアウトのみの候補手を生成する。座標を離散化する間隔は、  $Q_p$  は  $x, y$  方向ともにストーン半径  $r_s$ 、  $Q_f$  は  $0.5r_s$  としている。また、以下の式 (9) 中の  $ph_x, ph_y$  はハウス中央の  $x, y$  座標であり、式 (9), (10), (11) 中の  $Z$  は

整数の集合を表す.

$$\begin{aligned}
 M &= \{m \mid m = (q, R), q \in Q(R), \\
 Q(R) &= Q_p(R) + Q_f(R), \\
 R &\in \{R_R, R_L\}\}
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 Q_p(R_R) &= Q_p(R_L) \\
 &= \{q \mid q = (x, y), \\
 x &= p_{hx} + r_s \times s, \\
 y &= p_{hy} + r_s \times t, \\
 -13 &\leq s \leq +13, \\
 -17 &\leq t \leq +41, \\
 s, t &\in \mathbf{Z}\}
 \end{aligned} \tag{9}$$

$$\begin{aligned}
 Q_f(R_R) &= \{q \mid q = (x, y), \\
 x &= 0.5 + 0.5r_s \times s, \\
 y &= 6.039, \\
 0 &\leq s \leq 71, \\
 s &\in \mathbf{Z}\}
 \end{aligned} \tag{10}$$

$$\begin{aligned}
 Q_f(R_L) &= \{q \mid q = (x, y), \\
 x &= 4.25 - 0.5r_s \times s, \\
 y &= 6.039, \\
 0 &\leq s \leq 71, \\
 s &\in \mathbf{Z}\}
 \end{aligned} \tag{11}$$

$$|Q_p(R_R)| = |Q_p(R_L)| = 1593 \tag{12}$$

$$|Q_f(R_R)| = |Q_f(R_L)| = 72 \tag{13}$$

### 3.2 代表実行手集合 $M'(m)$ の生成

デジタルカーリングと同様に候補手へ連続値の乱数が加算されるゲームとして、ピリヤードのコンピュータゲームがある. 2016年5月時点で世界1位のピリヤードAIである CueCard では、候補手  $m$  ごとに試合時と同一の乱数を用いて代表実行手集合  $M'(m)$  を生成している. この手法では、候補手  $m_1, m_2$  の候補手空間上での距離が近く  $M'(m_1), M'(m_2)$  の領域が互いに重なる場合にも、各々を独立に生成している (図6 上段).

しかし、そのような候補手  $m_1, m_2$  では、 $M'(m_1), M'(m_2)$  間で互いに  $m'$  を共有させることで、 $m_1, m_2$  (のチャンスノード) から同一の子ノードへ分岐させ、生成される子ノードの総数を削減可能である (図6 中段, 図7).

本論文における提案手法では、候補手  $m$  ごとに  $m' \in M'(m)$  を生成するのではなく、あらかじめ全候補手間で共有することを目的に  $m'$  を等間隔 ( $M$  と同一の間隔) に離散化した  $m'$  の全体集合  $U_{m'}$  を定義し (式(14)),  $M'(m)$  をすべて  $U_{m'}$  の部分集合とすることで (式(15)) 異なる候補手間で  $m' \in M'(m)$  を共有する (図6 下段).

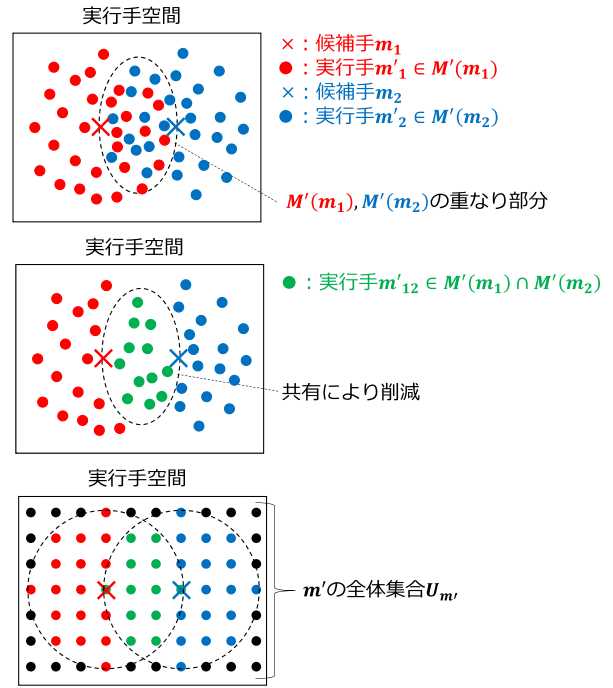


図6 上段：乱数により候補手  $m$  ごとに独立に実行手  $m'$  を生成  
中段：候補手間で実行手が共有  
下段：あらかじめ全候補手間で  $U_{m'} \ni m'$  を定義

Fig. 6 Top: Possible results by candidates moves.  
Middle: Sharing the possible results among different candidates moves.  
Bottom: Our sharing method for digital curling.

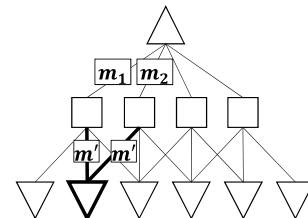


図7 候補手  $m_1, m_2$  で共通の  $m'$  を使用 (ゲーム木)  
Fig. 7 A resulting move is shared between candidate move  $m_1$  and  $m_2$  (game tree).

$$\begin{aligned}
 U_{m'}(R) &= \{m' \mid m' = (q', R), \\
 q' &\in U_{q'}(R)\}
 \end{aligned} \tag{14}$$

$$\begin{aligned}
 M'(m = (q, R)) &= \{m' \mid m' = (q', R), \\
 |q' - q| &\leq 3r_s \\
 q' &\in U_{q'}(R)\}
 \end{aligned} \tag{15}$$

式(14), (15)中の  $U_{q'}$  は、代表候補手集合  $M$  の生成に使用した投球目標座標集合  $Q$  を拡張した集合である (図5 右図).  $Q$  からの拡張は、すべての  $m$  において  $M'(m)$  を生成するため、 $Q$  の端から  $3r_s$  の範囲で新たな投球目標座標を追加する. また式(15)では、 $m'$  を  $M'(m)$  の要素とする条件を  $|q' - q| \leq 3r_s$  としているが、4章の対戦実験の設定 ( $\sigma_x = r_s, \sigma_y = r_s$ ) の場合、ハウス中央へ投球する候補手  $m_h = (q_h, R)$  と  $m_h$  から生成される実行

手  $m'_h = (q'_h, R)$  において  $|q'_h - q_h| \leq 3r_s$  となる確率は 99.7%であり、ハウス中央以外を投球目標座標とした場合もほぼ同様である。

なお、プレイエリア遠方の投球目標座標集合  $Q_f$  を投球目標座標とした候補手  $m_f$  においては、ストーンの初速の  $y$  方向成分が誤差ベクトル  $v_r$  の  $y$  方向成分に比べ十分に大きいと仮定し、代表実行手集合  $M'(m_f)$  は  $x$  軸方向に 1 列のみとして、 $y$  方向のずれは考慮しない。

また、提案手法では  $M'(m)$  生成時に乱数を使用しないため、 $M'(m)$  の要素数が少ない場合にも偏りが生じないというメリットもある。

### 3.3 チャンスノードの評価値算出

チャンスノードの評価値は、チャンスノードから展開される各子ノードの生起確率と評価値の積を足し合わせた期待値であるが、各子ノードの生起確率は各候補手  $m \in M$  (チャンスノード) から各実行手  $m' \in M'(m)$  へ分岐する確率  $p(m, m')$  に相当する。本論文では、すべての  $m \in M$ ,  $m' \in M'(m)$  の組において  $p(m, m')$  を算出した確率テーブルを事前に生成する。確率テーブルの作成にあたっては、試合時の乱数を用いて候補手  $m = (q, R)$  から実行手  $m'_r = (q'_r, R)$  を 10 万回生成し、 $m' = (q', R) \in M'(m)$  ごとに  $|q' - q'_r|$  が  $M'(m)$  内で最小となった回数をカウントするという操作をすべての  $m \in M$  に対して行った。

### 3.4 局面評価関数

カーリングはエンドとよばれる部分ゲームを繰り返すことで試合が進行する。このような部分ゲームを繰り返すゲームでは、必ずしも部分ゲーム内の得点を最大化することが最善な行動であるとは限らない。実際にカーリングでは後攻は次のエンドでの後攻権獲得のため、1 点を取るよりも意図的に両チーム 0 点とする戦略をとることがある。

カーリングと同様に部分ゲームを繰り返すゲームとしてバックギャモンがある。バックギャモンでは Match Equity Table という各試合状況ごとの勝率を記録した勝率テーブルが存在し [12]、戦略立案に役立っている。本論文では、デジタルカーリングにおけるゲーム木探索にこの勝率テーブルを適用する (作成方法は本章の最後に述べる)。その際、残りエンド数、現在の得点差 (自分の得点 - 相手の得点)、自分の手番 (先攻 or 後攻) の 3 つの要素をデジタルカーリングにおける試合状況として定義する。ゲーム木探索における末端ノードの評価時に、そのノードのストーン配置で得点計算を行い次のエンドに進んだときの試合状況を確定させ、勝率テーブルを参照することで最終的な勝率を最大化する手を選択することが可能となる。

以上のことをふまえて、最終的な勝率を考慮した局面評価関数  $e(s)$  を設計した (式 (16))。

$$e(s) = W(s) + \mu \sum_i u(s, i) N(s, i) \quad (16)$$

$W(s)$  は局面  $s$  のストーン配置で本来後攻 8 投目投球後に行われる得点計算を行い次のエンドに進んだ場合の勝率を出力する項である。ただし局面  $s$  が先攻投球後の局面である場合には、先攻のストーンの中でハウス中央に最も近いストーン (先攻のナンバーワンストーン) を後攻のストーンに置き換えて得点計算を行う。これは本来の得点計算は必ず後攻投球後に行われるため、先攻のナンバーワンストーンは後攻の投球によりはじき出されやすいという点を考慮している。

また、 $W(s)$  は局面  $s$  のストーン配置で得点計算を行うが、まだエンド内の投球回数が残っている場合には現在のストーン配置で計算した得点と本来得点計算が行われる後攻 8 投目投球後のストーン配置で計算した得点は異なる可能性があり、出力する勝率は正確な値にはならない。そこで局面  $s$  を各ストーンの位置やストーンどうしの位置関係の面から評価するために、局面  $s$  のストーン配置を評価する項  $\mu \sum_i u(s, i) N(s, i)$  を導入する。 $\mu$  は結合の重み、 $u(s, i)$  は局面  $s$  において  $i$  番目に投球されたストーン  $b(s, i)$  が自分のストーンの場合は +1、相手のストーンの場合は -1 を出力する関数である。 $N(s, i)$  は  $b(s, i)$  の位置や他のストーンとの位置関係を評価する関数であり、以下の式 (17) で表される。

$$N(s, i) = J(s, i) \times T(s, i) \times D(s, i) \quad (17)$$

$J(s, i)$  は  $b(s, i)$  の得点への結びつきやすさに関する評価指標、 $T(s, i)$  は  $b(s, i)$  のテイクアウトされにくさに関する評価指標、 $D(s, i)$  は  $b(s, i)$  のダブルテイクアウトされにくさに関する評価指標を表す。

#### 3.4.1 $b(s, i)$ の得点への結びつきやすさ $J(s, i)$

$J(s, i)$  は以下の式 (18) で表される。

$$J(s, i) = w_1 k(s, i) + w_2 h(s, i) \quad (18)$$

$w_1, w_2$  は結合の重みである。 $k(s, i)$  は  $b(s, i)$  のハウス中央への近さの順位に関する評価指標、 $h(s, i)$  は  $b(s, i)$  の盤上の位置に関する評価指標であり各々以下の式 (19)~(22) で表される。

$$k(s, i) = \frac{1}{1 + n_s(s, i)} \quad (19)$$

$$h(s, i) = h_x(s, i) \times h_y(s, i) \quad (20)$$

$$h_x(s, i) = \begin{cases} 1 - \frac{|x - p_{hx}|}{\alpha} & (|x - p_{hx}| < \alpha) \\ 0 & (\text{otherwise}) \end{cases} \quad (21)$$

$$h_y(s, i) = \begin{cases} \frac{(y - p_{hy} + \beta)^2}{\beta^2} (p_{hy} - \beta \leq y < p_{hy}) \\ 1 - \frac{(y - p_{hy})^2}{\gamma^2} (p_{hy} \leq y < p_{hy} + \gamma) \\ 0 & (\text{otherwise}) \end{cases} \quad (22)$$

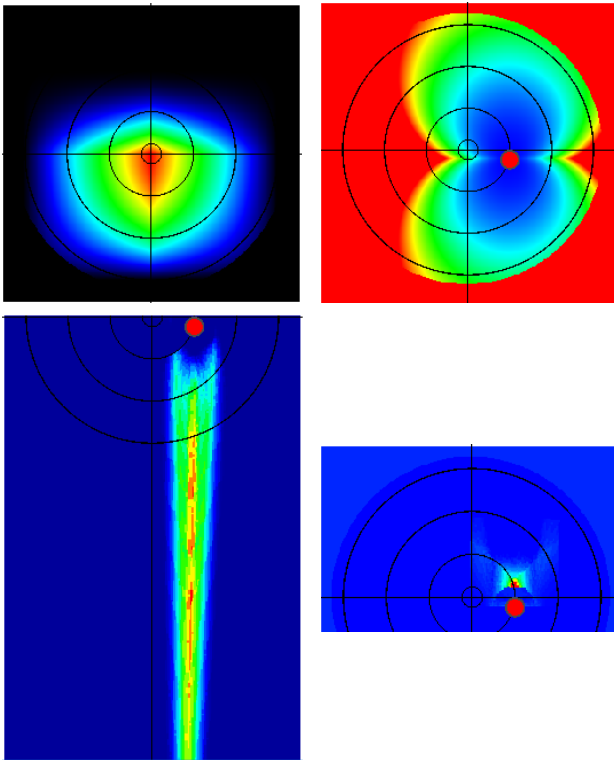


図 8 左上図は各座標ごとの  $b(s, i)$  を置いたときの  $J(s, i)$  の値, 他の図はすでに  $b(s, i)$  が盤面上にある場合に各座標にストーンを置いたときの  $D(s, i)$  の値 (右上図) と  $T(s, i)$  の値 (左下図, 右下図) を座標ごとにヒートマップで表した図

Fig. 8 Heat map of value of  $J(s, i)$ ,  $T(s, i)$  and  $D(s, i)$ , figure in the upper left is value of  $J(s, i)$  when putting  $b(s, i)$  on each coordinates, and the other pictures are value when  $b(s, i)$  is already put and putting another stone on each coordinates ( $D(s, i)$ : figure in the upper right,  $T(s, i)$ : figure below left and right).

式 (19) 中の  $n_s(s, i)$  は  $b(s, i)$  よりもハウス中央に近い相手ストーンの数である. 式 (20) 中の  $h_x(s, i)$ ,  $h_y(s, i)$  は各々  $b(s, i)$  の  $x$ ,  $y$  座標に関する評価値であり,  $h(s, i)$  はそれらの積である. 式 (21), (22) 中の  $x$ ,  $y$  は  $b(s, i)$  の  $x$ ,  $y$  座標,  $p_{hx}$ ,  $p_{hy}$  はハウス中央の  $x$ ,  $y$  座標,  $\alpha$ ,  $\beta$ ,  $\gamma$  はパラメータである.  $h(s, i)$  は  $b(s, i)$  の位置がハウス中央に近づくほど高くなる. また, カーリングでは盤上のストーンは投球されたストーンにはじかれた場合は必ず投球方向 (図 2) に移動する. そのためハウス後方 (ハウスの中心より投球位置から遠い側) よりハウス前方 (ハウスの中心より投球位置に近い側) に位置している方がエンド終了時にハウス中央に近くなる可能性は高い. よって  $h(s, i)$  の値はハウス前方の方が大きくなるように設計している (図 8 の左上).

### 3.4.2 $b(s, i)$ のテイクアウトされにくさ $T(s, i)$

$T(s, i)$  は以下の式 (23) で表される.

$$T(s, i) = \min(o(s, i, R)) + c \quad (23)$$

$o(s, i, R)$  は相手が回転  $R$  で投球した場合の  $b(s, i)$  のテ

イクアウトされにくさに関する評価指標であり,  $c$  は  $T(s, i)$  を 0 にしないための定数である.

一般にカーリングでは前後に他のストーンが存在することで相手にテイクアウトされにくくなる. 別なストーンが自身よりも投球位置側にある場合はそのストーンがテイクアウトの軌道を塞ぎ, 反対に別なストーンが自身よりも奥側にある場合は自身がテイクアウトされたときにそのストーンにより止められる可能性がある.

本論文ではデジタルカーリングにおいてストーン  $b$  と別なストーン  $b'$  の相対位置と,  $b$  のテイクアウトされにくさの関係をあらかじめシミュレーションにより調査した.  $b'$  を  $b$  より投球位置に近い位置に配置し,  $b$  をテイクアウトする軌道で投球する. テイクアウトは  $b$  の左端にぎりぎり接触する軌道からはじめて軌道をストーン半径  $r_s/6$  ずつ右にずらして計 24 軌道行う. 結果はテイクアウトの軌道ごとに, 投球したストーン  $b_t$  が  $b$  に接触した場合はテイクアウト成功で 1, 接触しなかった場合はテイクアウト失敗で 0 として 24 bit の bit 列で保存する. これを  $b_t$  の回転を  $R_R, R_L$  の 2 通りで, さらに  $b$  の位置を固定して  $b'$  位置を少しずつ変えて行い,  $b$  と  $b'$  の相対位置ごとにテイクアウトの成功と失敗に関する 24 bit の bit 列のデータベース  $O_f$  を作成した. 同様のことを,  $b'$  を  $b$  より投球位置から遠い位置に配置して行い, データベース  $O_b$  を作成した. ただしこの場合には,  $b_t$  は 24 軌道すべての軌道で  $b(s, i)$  と必ず接触するため, テイクアウトの成功条件は  $b(s, i)$  が元の位置から一定以上移動することとした.

$$o(s, i, R) = w_3 o'(o_f(s, i, R)) + w_4 o'(o_b(s, i, R)) \quad (24)$$

$$o'(o_f(s, i, R)) = \frac{1}{o''(o_f(s, i, R)) + 1} \quad (25)$$

$$o'(o_b(s, i, R)) = \frac{1}{o''(o_b(s, i, R)) + 1} \quad (26)$$

$w_3, w_4$  は結合の重みである.  $o_f(s, i)$  は  $b(s, i)$  と局面  $s$  における  $b(s, i)$  以外のすべてのストーン,  $o_b(s, i)$  は  $b(s, i)$  と局面  $s$  における  $b(s, i)$  から見た相手チームのすべてのストーンとの相対位置ごとに各々  $O_f, O_b$  から bit 列を参照し, それらを AND 演算した結果を出力する関数であり,  $o''$  は bit 列を入力とし, bit 列中で 1 が最も多く続いた回数を入力する.  $b(s, i)$  の前後に他のストーンが存在すると  $T(s, i)$  の値は大きくなる (図 8 の左下と右下).

### 3.4.3 $b(s, i)$ のダブルテイクアウトされにくさ $D(s, i)$

ダブルテイクアウトとは, 一度の投球で相手のストーンを 2 個まとめてテイクアウトすることである. ストーンどうしの位置が遠く, 横に並んでいるほどダブルテイクアウトされにくくなる.  $D(s, i)$  は以下の式 (27)~(29) で表される.

$$D(s, i) = \min(D'_{i,j}) \quad (27)$$

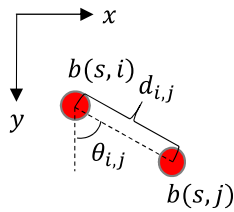


図 9 ダブルテイクアウトされにくさ  $D(s, i)$

Fig. 9 Difficulty of double takeout  $D(s, i)$ .

$$D'_{i,j} = \begin{cases} 0.5 + 0.5 \frac{d_{i,j}^2}{D''_{i,j}{}^2} & (d_{i,j} < D''_{i,j}) \\ 1 & (\text{otherwise}) \end{cases} \quad (28)$$

$$D''_{i,j} = a + b \cos \theta_{i,j} \quad (29)$$

$j$  は  $i$  を除く自チームのストーン番号である。式 (29) の  $a, b$  はパラメータ、 $\theta_{i,j}$  は  $y$  軸方向のベクトルと  $b(s, i), b(s, j)$  の座標の差分ベクトルがなす角度であり、式 (28) の  $d_{i,j}$  は  $b(s, i), b(s, j)$  間の距離である (図 9)。  $D'_{i,j}$  は  $b(s, i)$  と  $b(s, j)$  が遠ざかるほど、また並びが水平になるほど大きくなる (図 8 の右上)。

### 3.5 勝率テーブルの作成方法

残りエンド数 0~10, 得点差 -5~+5, 先攻 or 後攻の各試合状況における勝率のテーブルを以下の手順で作成した。

- (1) エンド内の得点を最大化するじりつくん (GPW 杯時) どうして 1 エンドの試合を複数回行い、得点分布を作成する。このじりつくんはすべての試合状況下で同一の挙動をするため、生成される得点分布はすべての試合状況で同一のものとなる。
- (2) 試合終了時 (残りエンド数 0) の勝率を、得点差が正の値のときは 100%, 負の値のときは 0%, 得点差なし (0 点) のときは延長戦を行ったときの勝率とし、(1) で作成した得点分布に従い残りエンド数 1 から順に各試合状況における勝率を算出しテーブル作成する。延長戦時の勝率は、得点差を正にすることを目的としたじりつくんどうしの 1 エンドの対戦を複数回行うことで求める。
- (3) (2) で作成した勝率テーブルを利用したじりつくんどうして各試合状況下で 1 エンドの試合を複数回行い、試合状況ごとの得点分布を作成する。(2) とは異なり、試合状況ごとに異なる得点分布が生成される。
- (4) (2) と同様の手順でテーブルを完成させる。

表 1 は後攻視点の勝率テーブルを一部抜粋したものである。

## 4. 実験

結果に不確実性を含むデジタルカーリングにおける提案手法による先読みの有効性を明らかにするために、不確実性を考慮する場合としない場合それぞれにおいて探索の深

表 1 勝率テーブル (後攻視点)

Table 1 Winning percentage table (second player).

	残りエンド数			
	3	2	1	0
+3	94%	97%	99%	100%
+2	88%	91%	96%	100%
得 +1	77%	80%	87%	100%
点 ±0	62%	61%	73%	73%
差 -1	45%	39%	47%	0%
-2	29%	22%	24%	0%
-3	16%	10%	8%	0%

さを変化させ既存 AI との対戦実験を行い性能を比較する。この際、探索の有効性を明らかにするためには、使用する評価関数の精度が重要となる。本実験では 3.4 節で述べた評価関数を使用するが、この評価関数は過去の大会で優勝したときのものをベースとしており、一定以上の精度があるものと考えられる。対戦に用いるじりつくんは、探索の深さが 1 の場合と 2 の場合、さらに不確実性を考慮する場合と不確実性を考慮しない場合の計 4 パターンとする。なお、不確実性の考慮の有無は 3.2 節で述べた代表実行手集合の要素数により決定される。不確実性を考慮しない場合には候補手に乱数を加算せずそのまま実行手とし、代表実行手集合の要素数を 1 として探索を行う。

### 4.1 実験設定

対戦相手には第 2 回 UEC 杯の優勝 AI である「GCCS」を用いる。GCCS は、モンテカルロ木探索を用いた第 1 回 UEC 杯と GPW 杯の優勝 AI「歩」をベースとし、序盤 (後攻 2 投目まで) は実際のカーリング戦略の知識を用いて候補手を選択する。探索時の局面の評価は、1 試合 10 エンドの場合には、前半の 1~5 エンドはエンド内の得点をベースに行い、後半の 6~10 エンドは最終的な勝率をベースとして行う。

また、じりつくんは公式大会参加時には、制限時間を考慮して探索時間を短縮するためにプレイエリア周辺の目標座標集合である  $Q_p$  のうち、図 10 の白抜き円の座標から生成された実行手を最も近い塗りつぶし円の座標から生成された実行手で代用して探索を行っており、本実験においても同様の設定で行っている。白抜き円の座標から最も近い塗りつぶし円の座標が 2 つ存在している場合は、 $y$  座標の値の小さい座標 (自身より上側の座標) を優先している。これにより、 $Q_p$  から生成される候補手は左右の回転ごとに 957 個となり、1 つの親ノードから生成する実行手の総数を元の 4,458 個から 2,082 個に減らしている。なお、白抜き円の座標は経験的に決定している。

なお、実験に用いた評価関数のパラメータは以下のとおりである。各パラメータの値は複数の局面において探索を行った際に意図どおりの候補手を選択するように手動で調



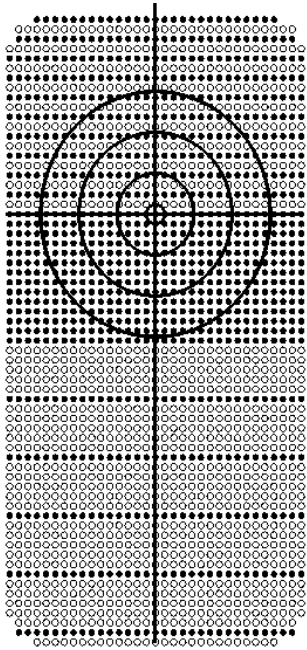


図 10 白抜き円の座標から生成される実行手をすべて最も近い塗りつぶし円の座標から生成される実行手で代用

Fig. 10 Resulting moves made from filled circle coordinates is substituted by resulting moves made from circle coordinates.

整した.

$$\mu = \begin{cases} 0.1 & (\text{if the first player's turn}) \\ 0.2 & (\text{otherwise}) \end{cases} \quad (30)$$

$$w_1 = 1.0 \quad (31)$$

$$w_2 = 1.0 \quad (32)$$

$$\alpha = r_{h1} + r_s \quad (33)$$

$$\beta = r_{h2} \quad (34)$$

$$\gamma = r_{h1} + r_s \quad (35)$$

$$w_3 = 1.0 \quad (36)$$

$$w_4 = 2.0 \quad (37)$$

$$a = 0.915 \quad (38)$$

$$b = r_{h1} \quad (39)$$

$$c = 0.3 \quad (40)$$

$r_{h1} = 1.83$  は図 3 の青い円の半径,  $r_{h2} = 1.22$  は白い円の半径,  $r_s = 0.145$  はストーン半径を表す.

対戦は 1 エンド目後攻をじりつくんに固定して, 1 試合 10 エンドで 100 試合ずつ行い, 10 エンド終了時の得点が等しい場合には延長戦を行う. 延長戦は直前のエンドで得点したチームを先攻として, 勝敗がつくまで 1 エンドの試合を繰り返し行う. 制限時間は 1 試合 1,440 秒, 延長戦では 144 秒する. 使用する計算機は Intel Xeon コア数  $4 \times 2$  (2 プロセッサ) クロック周波数 2.26 GHz である.

表 2 100 試合中のじりつくんの勝率. 括弧内は標準誤差を表す  
Table 2 Winning rate of victory of jiritsu-kun in 100 games.

	不確定性の考慮	
	あり	なし
深さ 1	39.1% ( $\pm 5.2\%$ )	10.9% ( $\pm 3.1\%$ )
深さ 2	82.3% ( $\pm 3.7\%$ )	7.9% ( $\pm 2.7\%$ )

表 3 じりつくんが先攻のエンドの得点分布  
Table 3 Score distributions when jiritsu-kun is the first player.

	不確定性の考慮あり		不確定性の考慮なし	
	深さ 1	深さ 2	深さ 1	深さ 2
3 点以上	1%	0%	0%	0%
2 点以上	3%	4%	1%	0%
1 点以上	11%	15%	6%	2%
0 点	4%	6%	0%	1%
-1 点以下	47%	50%	47%	46%
-2 点以下	24%	18%	28%	30%
-3 点以下	9%	5%	14%	15%
平均得点	-1.12	-0.79	-1.68	-1.93
分散	2.23	1.77	1.97	1.48
エンド数	226	235	164	163

表 4 じりつくんが後攻のエンドの得点分布  
Table 4 Score distributions when jiritsu-kun is the second player.

	不確定性の考慮あり		不確定性の考慮なし	
	深さ 1	深さ 2	深さ 1	深さ 2
3 点以上	4%	13%	5%	5%
2 点以上	21%	27%	16%	16%
1 点以上	52%	44%	37%	38%
0 点	5%	0%	3%	8%
-1 点以下	11%	8%	25%	22%
-2 点以下	3%	2%	8%	7%
-3 点以下	1%	0%	2%	3%
平均得点	0.93	1.51	0.35	0.40
分散	1.77	2.52	2.88	2.50
エンド数	294	270	341	342

## 4.2 実験結果

100 試合中の勝敗結果から算出した標準誤差を用いた検定を行った. その結果を表 2 に示し, エンドごとの得点分布と平均得点を表 3 と表 4 に示す. 対戦相手の GOCS は後半 6~10 エンドで大量得点がついている場合にはどの候補手を選択しても評価が等しくなり (勝率が 100% または 0%) 必ずしも有効でない手を選択するため, エンドごとの得点分布と平均得点の算出には各試合前半の 1~5 エンドまでの結果を用いている.

表 2 の結果から, 不確定性の考慮ありの場合は探索の深さ 1 の場合, 2 の場合ともに不確定性の考慮なしの場合に比べ勝率が有意に高く, また不確定性の考慮ありの場合は探索の深さ 1 より探索の深さ 2 の方が勝率が有意に高いことが確認できたが, 不確定性の考慮なしの場合は勝率に有

意差は見られなかった。

表 3 から、先攻エンドにおいて不確定性を考慮しない場合には深さ 2 の平均点が深さ 1 より有意に低くなる事が確認できた (片側有意水準 5% の Welch の t 検定)。

表 4 から、後攻エンドにおいて探索の深さ 1 では不確定性の考慮ありの方が考慮なしより 1 点以上である確率が有意に高かったが、2 点以上である確率と 3 点以上である確率では有意差はなかった (両側有意水準 5% の Z 検定)。

また、1 手の探索にかかった平均時間は以下のとおりである。

- GCCS : 17.97 秒
- 不確定性の考慮あり探索の深さ 1 : 1 秒未満
- 不確定性の考慮あり探索の深さ 2 : 13.88 秒
- 不確定性の考慮なし探索の深さ 1 : 1 秒未満
- 不確定性の考慮なし探索の深さ 2 : 6.84 秒

### 5. 考察

不確定性を考慮した場合に探索の深さを増やすことで勝率が上昇したという結果から、不確定ゲームにおいて提案手法により不確定性を考慮することで適切に先読みできていることが明らかとなった。また逆に不確定性の考慮なしで探索の深さを増やした場合に先攻エンドにおける平均得点が低下したことは、実現される可能性を考慮せずに先読みを行ったことで、実際には生起する可能性が非常に低い展開も無視できなかつたためであると考えられる。実際に不確定性の考慮なしの場合と不確定性の考慮ありの場合での着手を比較したところ、不確定性の考慮なしの場合は図 11 のような正確に実行できた場合には非常に有利となるが実現可能性が低いような手を選択する傾向が見られた。また、不確定性の考慮ありの場合に探索の深さを深くすることで、図 12 のように深さ 1 では次の相手の投球を十分に考慮できておらず自分のストーンに近い位置に投球しているが、深さ 2 まで読むことでその後の相手の投球によるダブルテイクアウトのリスクを考慮して自分のストーンから離れた位置への投球を行う傾向がみられた。

さらに、後攻の深さ 1 の探索において不確定性を考慮する場合としない場合では不確定性を考慮した方が 1 点以上である確率は高いが 2 点以上である確率と 3 点以上である確率に有意差はなかったことは、1 点を高確率でとれる局面において、不確定性を考慮しない場合には複数点を得られる確率が少しでも存在すれば失敗した場合をまったく考慮せずに複数点を狙うためであると考えられる。

### 6. おわりに

本論文では、デジタルカーリングにおいてゲーム木探索を適用する手法を提案し、その手法を用いて開発した AI 「じりつくん」と既存 AI の対戦実験を行った。その結果、不確定性を考慮した場合に勝率が上昇し、また不確定性を

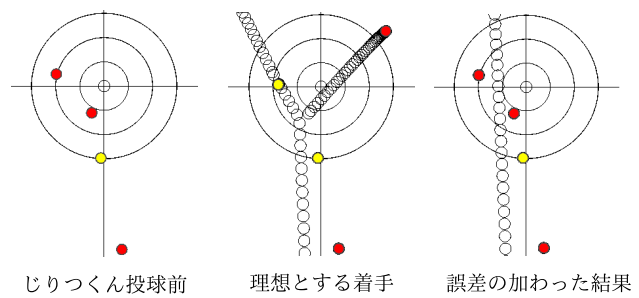


図 11 深さ 2 不確定性の考慮なしの着手例

Fig. 11 Example of move without considering uncertainty at depth 2.

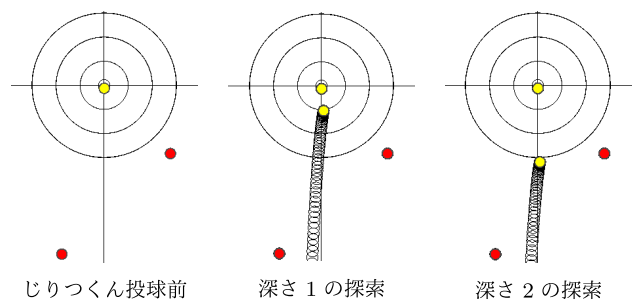


図 12 不確定性の考慮ありで探索の深さが異なる場合の着手比較

Fig. 12 Comparison of move between depth 1 and depth 2 (uncertainty is considered).

考慮した場合のみ探索の深さを増やすことで勝率が上昇した。このことから、結果に不確定性を含むデジタルカーリングにおける提案手法による先読みの有効性が明らかとなった。今後は、候補手の絞り込みを行うことでさらに深い探索を行い、不確定ゲームにおいて深く探索することの意義をより明らかにする予定である。

### 参考文献

- [1] Campbell, M., Hoane, J.A., Hsu, F., et al.: Deep blue, *Artificial Intelligence*, Vol.134, No.1-2, pp.57–83 (2002).
- [2] Silver, D., Huang, A., Maddison, J.C., et al.: Mastering the game of Go with deep neural networks and tree search, *Nature*, Vol.529, pp.484–489 (2016).
- [3] 水上直紀, 中張遼太郎, 浦 晃ほか: 多人数性を分割した教師付き学習による四人麻雀プログラムの実現, 情報処理学会論文誌, Vol.55, No.11, pp.2410–2420 (2014).
- [4] 篠田孝祐, 鳥海不二夫, 片上大輔ほか: 汎用人工知能の標準問題としての人狼ゲーム, 人工知能学会全国大会論文集, Vol.20, pp.1–3 (2014).
- [5] Smith, M.: Pickpocket: A computer billiards shark, *Artificial Intelligence*, Vol.171, pp.1069–1091 (2007).
- [6] Altman, A., Archibald, C. and Shoham, Y.: Analysis of awinning computational billiards player, *Artificial Intelligence*, Vol.171, pp.1069–1091 (2009).
- [7] Coleman, G.: Introduction to Curling Strategy (2014).
- [8] 北清勇磨, 岡田雷太, 伊藤毅志: デジタルカーリングサーバーの提案と紹介, 情報処理学会研究報告, Vol.GI31, No.2, pp.1–5 (2014).
- [9] 電気通信大学情報理工学研究所伊藤毅志研究室: デジタルカーリング, 入手先 (<http://minerva.cs.uec.ac.jp/curling/wiki.cgi>) (参照 2016-02-19).

- [10] 加藤 修, 飯塚博幸, 山本雅人: 不確実性を含むカーリングにおける先読み手法の提案と有効性検証, 第14回情報科学技術フォーラム (FIT2015) 講演論文集, No.14, pp.313–314 (2015).
- [11] Michie, D.: *Advances in programming and non-numerical computation*, pp.183–200 (1966).
- [12] Backgammon Galore: Match Equity Table, available from <http://www.bkgm.com/articles/GOL/demo/equity.htm> (accessed 2016-02-19).



加藤 修 (学生会員)

1990年生。2015年北海道大学工学部情報エレクトロニクス学科卒業。同年同大学大学院情報科学研究科修士課入学。



飯塚 博幸 (正会員)

2004年東京大学総合文化研究科博士課程修了(博士(学術)), 2005年日本学術振興会特別研究員(PD, はこだて未来大学), イギリスサセックス大学客員研究員, 2008年大阪大学大学院情報科学研究科助教。2013年北海道大学大学院情報科学研究科准教授。専門は人工生命, 複雑系科学, 人間情報工学。



山本 雅人 (正会員)

1968年生。1996年北海道大学大学院工学研究科システム情報工学専攻博士後期課程修了。同年日本学術振興会特別研究員(PD)。1997年北海道大学大学院工学研究科助手。2000年同大学院工学研究科助教授。同大学院情報科学研究科助教授, 2007年同大学院同研究科准教授を経て, 2012年同大学院教授。この間, 科学技術振興機構さきがけ研究員, デューク大学客員研究員, 博士(工学)。現在は, 進化型計算に基づく仮想ロボット開発, ゲームプログラミング, 複雑ネットワークの研究に従事。情報処理学会北海道支部長, 観光情報学会理事, 電子情報通信学会, 人工知能学会, 日本オペレーションズ・リサーチ学会, 精密工学会, 日本機械学会等, サービス学会, 廃棄物資源循環学会各会員。