

# GPCC 報告 (2022 年)

## Games and Puzzles Competitions on Computers

<http://hp.vector.co.jp/authors/VA003988/gpcc/gpcc.htm>

藤波順久\*

### 1 2022 年の課題

2022 年の GPCC では、以下の課題を取り上げた。

**GESCHENKT(ゲシェンク)** 3~7 人で行うカードゲームである。3~35 の 33 枚のカード (数字はマイナス点と考えられる) と、55 枚のチップを使用する。チップは人数が 3~5 なら 11 枚、6 なら 9 枚、7 なら 7 枚ずつ配る。カードはシャッフルしてから 9 枚抜いて (抜いたカードも伏せたまま) 残りを山札にし、山札を 1 枚めくってゲームを始める。

手番の人はカードを取るか、チップを 1 枚出すかの、どちらかを行う。前者の場合は出ているチップがあればすべて取り、山札を 1 枚めくって続ける (続けてカードを取ることもできる)。後者の場合は次の人の番になる。カードをすべて取り終わったら終了である。

終了後は、それぞれ取ったカードの数字を合計する。ただし、数字の連続するカードがあれば、その中の一番小さい数字だけ合計する。持っているチップの枚数を合計から引き、その値が一番小さい人が勝ちとなる。

山札をフルオープンして始めるバリエーションも考えられるが、その場合は 2 人での対戦にしてもよいかもかもしれない。

**BRIDGET** 二人で行うボードゲームである。8×8 の盤に白と黒の立体テトロミノを交互に置き、盤の任意の対辺をつないだ人が勝ちである。ピースはそれぞれ、L×4, S×4, T×4, O×2 の 14 個を使う。置き方には以下の制限がある。

- 各ピースの少なくとも一つの立方体が盤に接するように置く
- 宙に浮いている立方体があってはいけない

「つながっている」の定義は、以下の二通りを選択可能とする。

(2D ルール) 真上から見てつながっている

(3D ルール) 真上から見てつながっている、かつ、段差が発生する場合は壁がすべて同色

---

\*株式会社ソニー・インタラクティブエンタテインメント、GPCC chair

それぞれ14個のピースをすべて置いても対辺をつなげられなかった場合、ステイルメイトとなる。その場合は、真上から見て、盤の縁のマスを占める数が多いほうを勝ちとする(これは紙の説明書のルールに従っており、Webの説明<sup>1</sup>とは異なる)。

なお、ピースをすべて置く前であっても、ピースを置けなくなる場合がある。その場合、置けない人はパスで、二人とも置けない場合はステイルメイトと同じ扱いとすることにする。

## 2 2022年の進展

**BRIDGET**については、第64回プログラミング・シンポジウムで、天海良治さんと竹内郁雄さんが、以前開発したプログラムのその後の進展について報告予定である。詳細は同シンポジウム予稿集の「Lisp マシンエミュレータ Celis の並列化 - Bridget ゲームへの適用」を参照してほしい。以下はその概要である。

1980年代半ばに開発・市販された Lisp マシン TAO/ELIS の C 言語によるエミュレータ Celis は、最近の Intel や AMD CPU 上ですでに当時の約 300 倍の速度で動作している。当時の技術水準により、エミュレータ部分を含めても使用するメモリは 140MB 程度であり、現在の PC の GB オーダーのメモリには数十台分が十分に入る。最近の PC のマルチコアの性能向上は目覚しく、16 コアのマシンが容易に入手できるようになった。そこで、各コアで並列に Celis を走らせる実装をほとんどシェルプログラミングで行なった。これにより、一昨年のプロシンで報告した Bridget というゲームの AI を、現時点で 3 台 48 コア (49 並列 Celis) で、4 手先読みで走らせることが可能になった。シンポジウムまでには、72 コアに拡張する予定である。本報告では並列化の API や実装の工夫などについて述べる。

両氏より、上記報告に対する付加的な報告をいただいたので、3 節に掲載する。そこで使われている手の表記方法と盤面の表示方法の概略は本文中で説明されているが、詳細を知りたい場合は <https://www.nue.org/nue/bridget/bridget-notation.pdf> を参照してほしい。

同シンポジウムではまた、GPCC で昨年取り上げた **ハゲタカのえじき** について、志村伊生六さんと山口文彦さんによる報告が予定されている。詳細は同シンポジウム予稿集の「モンテカルロ木探索を用いたハゲタカのえじきエージェントの評価」を参照してほしい。以下はその概要である。

モンテカルロ木探索は、シミュレーションを用いることで、ゲームにおける最善手を効率良く求めることができ、逐次手番ゲームにおける有効なアルゴリズムとして普及している。しかし行動を同時に選択する同時手番ゲームにおいては、自分の行動を決定する際に他プレイヤーの行動を観察することが出来ないため、逐次手番ゲームとは異なり、各プレイヤー毎に求めた評価値をまとめた上でノードを選ぶ。本研究では、ハゲタカのえじきという同時手番ゲームの 2 人プレイで、モンテカルロ木探索の複数のバリエーションを用いた対戦エージェントを制作し、

---

<sup>1</sup><https://etgames.co.uk/how-to-play/bridget/>

対戦における強さはどの程度か、理想的な戦略と考えられるナッシュ均衡戦略にどのくらい近づけるかを調べる。またハゲタカのえじきの3人プレイにおいて、2人プレイのエージェントに用いたアルゴリズムを基にモンテカルロ木探索の手法を提案し、同様の検証を行う。

**GESCHENKT** については、特に進展はなかった。

### 3 Bridgetに関する報告と新しい次の1手問題（竹内郁雄，天海良治）

この1年間の進展は、プロシンの論文に記載の通りであるが、少し付加的な報告をする。

(1) 並列化による高速化のほかに、基礎部分のプログラムのリファクタリングにより、全体として約2倍の高速化を行った。リファクタリングの詳細は省略する。

(2) デバグはもっぱら自己対戦（のログ）を観察することで行った。自己対戦は、黒の初手として可能な174種類のすべて手について網羅的に何度も行った。最近の多数のコア並列だと、1試合平均12分弱を要するので（短いものは5分程度、長いものは20分程度にもなる）、1ラウンドぶっ続けで30数時間を要する。

このうち、あまりにも早く勝負がついてしまう試合はどこかにバグがあるものと見なして、序盤のほうの手の修正を行って、再試合を行う。この修正により、試合がもつれるようだと、その修正を「定跡候補」に追加する。

あきらかなバグがあった場合や、パラメータ調整を行った場合は、再度自己対戦ラウンドを実行する。コンピュータは疲れを知らないが、対戦ログのあちこちをピックアップして精査する作業自体はかなり疲れるものであった。

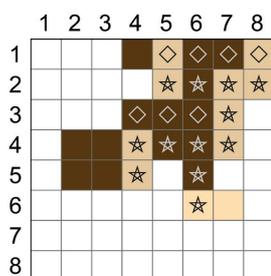
いずれにしてもこの自己対戦の観察による実力向上はまだ途上である。少しのんびりで行う予定である。そのうちモンテカルロ法などに追い抜かれてしまうかもしれない。

(3) 局面の評価関数はいまだに穴が多い。以下に紹介する「次の1手問題」の中には、評価関数をかなり抜本的にいじらないといけないことを示唆するものがある。しかし、評価関数を大幅に変更すると、これまででしっかり動いていた（いい手を指していた）ものを破壊する可能性もあり、慎重さが必要である。

(4) 分岐数の非常に大きいゲームなので、現在の候補手の生成はすで盤上にある駒に乗るか、接触するかの手（合わせて接触手と呼ぶ）に限定している。なお、一部例外はある。並列化により、非接触手、いわゆる positional move を担当するコアの計画を立てたが、それには着手しなかった。

以下、自己対戦の中で見つけた「次の一手問題」として面白そうなものを紹介する。解答は掲載しないので、ぜひ考えていただきたい。ここでは着手を87 DWのように座標と駒の向きを空白で区切って表す。局面の図示は真上から見たもので、☆印は3階、◇印は2階、

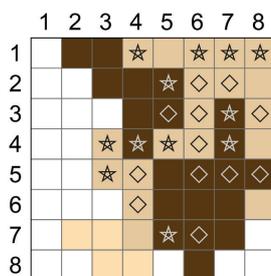
無印は1階であることを意味する。右側に書かれた情報は初手からの手数とそのときの着手である。その下の4桁はいわゆるLOSTで、黒と白の駒L, 駒O, 駒S, 駒Tの残り数を連続した4桁で表している。そのあとのカッコ内の数値は上から見えるそれぞれの駒の小体の数の総計を示している。右下の赤い数はこの局面での総候補手数である。



① 黒の必勝手は？

# 14 66 BE  
 1114 (14)  
 1141 (11)

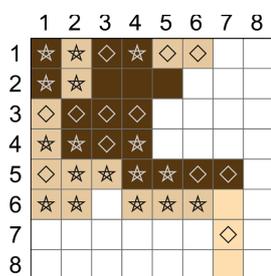
491



② 黒の必勝手は？

# 20 72 ZE  
 2002 (21)  
 0211 (23)

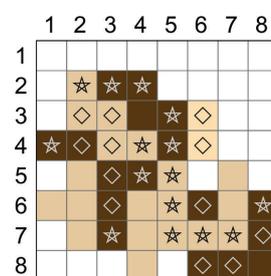
135



③ 次の黒の一手だけでなく、次の白、その次の黒も当ててください

# 20 77 VS  
 1111 (17)  
 0130 (17)

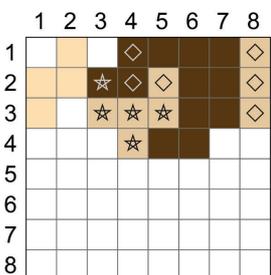
325



④ 黒は何を打つべきでしょうか？本文[a]参照

# 20 36 WS  
 1102 (17)  
 0121 (21)

185



⑤ 黒の必勝手は？  
 本文[b]参照

# 10 12 ZS  
 3033 (12)  
 2232 (12)

606

[a] この局面は実はプロシンの論文で紹介した bug63-の局面のあと、黒が 87 DW, 白が 36 WS と打った局面である。開発途上、まだ並列化による高速化がない時代、我々は白の 36 WS を見て「なんと天才的な！ こんな思いつかない！」と大感激したものである。しかし、4手先読みになり、これは幻想だったことが明らかに。しかし、ここで黒はうっかりしていると勝ち損ねる。

[b] ここでの必勝手は、現在の評価関数では候補手の第 41 位にしかない。これはなんとかしたいが、ここまで来ると評価関数の変更は結構な冒険である。