Title	不完全情報ゲーム『ガイスター』における詰め問題の提案と面白い問題生成
Author(s)	石井，岳史
Citation	
Issue Date	2020-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/16380
Rights	
Description	Supervisor：池田　心，先端科学技術研究科，修士（情報科学）

修士論文

不完全情報ゲーム『ガイスター』における詰め問題の提案と面白い問題生成

1810010　石井 岳史

主指導教員 池田 心

北陸先端科学技術大学院大学
先端科学技術研究科
（情報科学）

令和２年２月

Abstract

With the development of hardware and algorithms in past decades, artificial intelligence (AI) in games has shown impressive results. One possible way to make AIs involved more deeply in humans' daily lives is game content generation, which can be used to entertain human players and has become one of the popular fields among research in games. On the one hand, much research in content generation for perfect information games, such as tsume-shogi (shogi mating problems), has been well conducted. On the other hand, for imperfect information games, topics on how to entertain human players or content generation were less explored. However, for decision-making in real-world problems, people are not always provided with perfect information. Thus, it is expected that the research in imperfect information games can bring useful insights into solving real-world problems.

Geister is a two-player zero-sum deterministic imperfect information board game. At the beginning of a game, both players own the same sets of eight pieces, which contain four pieces of two types. Different types are labeled by different colors. One notable feature of the game is that players cannot observe the opponents' piece colors. Thus, players need to infer piece colors from past actions. For beginners of the game, inferring opponents' piece colors and misleading opponents about their piece colors are difficult skills to learn. Besides, it also happens that beginners miss the chances to win games where they could. From these aspects, we considered that it is necessary to provide players with environments that can assist them in improving playing skills. As tsume-shogi is to shogi, we thought that it is also possible to generate puzzles for Geister, which gives consideration to both entertainment and real plays. Considering that missing chances to win is one of the most important problems to overcome, we focused on endgame puzzles. Still, some other types of puzzles may also benefit beginners, such as "inferring the opponents' piece colors" and "indicating the best actions."

This research aimed to "propose Geister endgame puzzles" and "investigate how to generate interesting Geister endgame puzzles of proper difficulty quickly." First, we need to survey research in game content generation, particularly that related to endgame puzzles, and then define Geister endgame puzzles. Also, the difficulty and interestingness of the generated puzzles should vary. To train players, interesting puzzles or those whose difficulty is proper to the target players should be provided. Thus, we also need to investigate features that can represent the interestingness and the difficulty of puzzles.

In this research, we proposed Geister endgame puzzles and generated interesting puzzles. First of all, we defined two kinds of Geister endgame puzzles. "Normal version" followed the same rule as the original Geister about the observation of opponents' piece colors, i.e., players cannot observe opponents' piece colors.

"Partially-informed version" differed in that the players are informed of part of the opponents' piece colors. We expected the strategies for playing the two versions to be different. Since players can usually infer some of the opponents' piece colors near endgames, we considered the partially-informed version to be closer to real plays.

With regard to generating various puzzles efficiently, we tried two algorithms, random method and reverse method. The former randomly placed given numbers of both players' pieces on the board, while the later backtracked legal actions from existing puzzles. We analyzed each generated puzzle by the time cost and the length of the longest winning path. For simplicity of discussions, "game length" refers to the length of the longest winning path of a puzzle in the rest of this Abstract. Experiments on the random method showed that the numbers of generated puzzles decreased drastically as the game lengths increased. In more detail, we collected puzzles whose game lengths were between 9 and 19. The number of puzzles with game lengths of 19 was less than 0.1% of all collected puzzles. Besides, the experiments demonstrated that the generation time for one puzzle increased considerably as the numbers of pieces increased. For example, when both players had one piece of each color (i.e., each player had two pieces), generating one puzzle cost 2.7 seconds on average. The time became 1,299 seconds per puzzle when both players had three pieces of each color.

By applying the reverse method, we generated new puzzles from existing ones. The results showed that the probability of generating a new puzzle from an existing one was higher than 50%. Especially, we succeeded in generating puzzles with game lengths of 21 from those of 19, while the random method failed to generate any. Moreover, the generation time was about 60 times faster than that of the random method. We concluded that the reverse method could generate Geister endgame puzzles more efficiently than the random method, especially puzzles whose game lengths are long.

As the next step, we conducted a subject experiment to investigate how human players feel about difficulty and interestingness for Geister endgame puzzles. We prepared a variety of puzzles and asked human subjects to solve these puzzles and then rate the interestingness and the difficulty in five-grade evaluation (-2, -1, 0, 1, and 2). From the results, the correlation coefficient between the average values of the interestingness and the difficulty was 0.63, which showed a moderately positive correlation. We also found that puzzles too difficult made players feel less interesting. Based on the evaluations from humans, we applied supervised learning to predict the interestingness and the difficulty of Geister endgame puzzles. The root-mean-square errors of interestingness and difficulty were 0.52 and 0.66, respectively, which showed that rough predictions could be made. We also analyzed crucial factors in predicting interestingness and difficulty, respectively. More specifically, interestingness was strongly influenced by the numbers of opponent's pieces to capture during the solving process and the numbers of pieces that the player should move. We suspected because puzzles involving capturing opponents' pieces were more attractive to players than those just to win by moving their pieces to "exits." As for difficulty, the maximum proof numbers at root nodes of depth-first proof-number (df-pn) search greatly influenced. Df-pn was used to prove whether a player can win from a given board, and the proof number of a node represents the number of leaf nodes to be examined. We considered that it was natural that players felt difficult for puzzles with high proof numbers at the root nodes.

To sum up, this research proposed endgame puzzles for Geister, an imperfect information game. We then applied the random method and the reverse method to generate puzzles, where the later was much more efficient. Furthermore, predictions on human players' ratings on difficulty and interestingness of Geister endgame puzzles were reasonably accurate. Based on the results, we thought that it would be able to generate puzzles of some designate difficulty or interestingness easily. Unlimited to Geister, we expected this research to contribute to the whole game society since Geister shares some general properties with other games.

概要

近年，ハードウェアの進歩や新たなアルゴリズムの提案により，ゲームにおける人工知能の発展は目覚ましい．なかでも人間の感性を理解することでより AI を人間の身近な存在とするべく，人間を楽しませるコンテンツ生成の AI についての研究が盛んに行われている．しかし将棋などの完全情報ゲームにおけるコンテンツである詰将棋などの研究が多く確認できる一方で，不完全情報ゲームにおけるプレイヤーを楽しませる研究やコンテンツ生成についての研究は比較的数が少ないと感じる．実社会における意思決定も，全ての情報が意思決定者に与えられない場合が殆どである．そのため不完全情報ゲームについての研究は，次なる社会問題解決法の糸口として多くの期待がかけられている．

不完全情報ゲームである『ガイスター』というボードゲームがある．ガイスターは互いのプレイヤーが 2 種 4 個ずつの駒を用いて遊ぶ二人零和確定不完全情報ゲームである．駒はそれぞれの特徴を持つ青色と赤色のものを用いる．ゲームの大きな特徴として対戦相手の駒の種類がわからないことが挙げられ，対戦相手の駒の種類をそれまでの動き等から予測する必要がある．このゲームを始めたばかりの初心者にとって，対戦相手の駒種の予測や自分の駒種を誤認させるテクニックなどの技術を身に着けることは難しい．さらには本来では勝利が確定している盤面の見逃しなどが発生しうる．そのため上達を促すためには，練習や教育のような技術力向上支援を行う環境の提供が必要となる．なかでも将棋においての詰将棋のような娯楽と実戦における技術力向上の手段として用いられるパズルは，ガイスターにおいても有用だと考えられる．このようなパズルは様々なものが考えられ，ガイスターにおいても『駒色予測問題』や『次の一手問題』などが作りえるが，まずは必勝盤面の見逃しの克服が重要と考えのもと詰め問題に着目した．

本研究の目的は『ガイスターの部分問題として切り出せるものにはどんなものがあるか』『面白く適切な難易度の詰め問題を高速に生成するにはどうしたらよいか』を解明することである．そのためにまず，現存する部分問題コンテンツを調査・考察し，ガイスターに適切なルールを持った新コンテンツを提案しなければならない．コンテンツを生成するに際し，様々な難易度・面白さを持つものを多く生成する必要がある．さらに問題を教育に適用する場合，面白い問題やプレイヤーに適した特定難易度の問題を提供することが求められ，面白い・難しいと感じる特徴を知る必要がある．

本研究では，ガイスターにおける部分問題として詰めガイスター問題を提案し，面白い問題の生成を行った．まず詰め問題のルールを定義し，実際のガイスターのルール同様に対戦相手の駒についての情報を完全に非公開とした『一般問題』と，一部の対戦相手の駒色が公開されている『一部公開問題』に大別した．実戦におけるゲーム終盤では対戦相手の駒色がそれまでの行動等から予測できることも多く，それによって特定の戦略が必要となってくる．そのため，より実戦的な課題として一部公開問題を提案する．

多様性のある問題を効率よく生成するためのアルゴリズムとして，ランダム生

成と既存問題から手を戻すことで新たな問題を生成する逆順生成法の 2 種を試し，それぞれの手法でどのような問題がどれほどの時間で生成されるのかを検証した．その結果，盤面をランダムに生成するランダム生成法においては，長手数の問題ほど生成数は激減し，9〜19 手の問題の生成を試みた際に，駒数によっては 19 手問題が 0.1%程度しか生成されなかった．さらに生成時間においては，各駒 1 つずつの場合 2.7 秒/問だったのが各駒 3 つずつの場合 1,299 秒/問と，駒数が増えるほど 1 問あたりの生成時間が顕著に増えることが確認できた．

逆順生成法では，別に用意した問題から新たな問題を生成する実験を行った．結果，1 つの問題から 50%以上の確率で新たな問題が生成されることを確認した．特に， 19 手問題の一部から 21 手問題が生成されるなど，ランダム生成法では非現実的であった生成を可能にした．さらに生成時間においても，駒数と手数によってはランダム生成のみと比較して 60 倍近くの速度が出るなど，問題とならなかった盤面が少なかったためか大きな改善が見られた．これによりランダム生成法で苦手としていた長手数の問題を効率よく生成可能となった．

次に人間が面白い・難しいと感じる特徴を得るために，生成した問題を用い被験者実験を行うことで，面白さおよび難しさの推定・評価を行った．様々なカテゴリの問題を被験者に解かせ，面白さと難しさの 5 段階評価 (-2,-1,0,1,2) でアンケートを取った．被験者の感じた面白さと難しさの平均値には 0.63 の相関係数があり，ある程度の正の相関があることを確認した．一方で難しすぎる問題はその限りでないということも考察できた．次に教師あり学習を用いて，用意した特徴量から面白さと難しさを推定するモデルを生成した．結果，精度を示す二乗平均平方根誤差は面白さ 0.52，難しさ 0.66 と，どちらに対してもある程度の推定ができた．さらに，問題の面白さ推定に対して，解く過程で取る必要がある敵駒の数が大きく影響していることがわかった．これはプレイヤーが勝利のためにただ脱出を目指すだけの問題よりも駒の取りあいが必要となる問題に魅力を感じるためだと考えられる．一方で，問題の難しさに関してはルートノードが Df-pn 探索中に得た最大の証明数が大きく影響していた．Df-pn はプレイヤーが特定の盤面から勝つことができるかを証明するために使用され，探索中に得られる証明数はその局面で調べる必要のある局面の量を表している．これが大きくなるほど難しいと感じるのは自然なことである．

本研究は，不完全情報ゲームであるガイスターの詰め問題コンテンツ提案と生成を通して，ゲームにおける部分問題の切り出しと，面白く適切な難易度の詰め問題の高速生成を目指した．そして，プレイヤーがどんな問題に面白さや難しさを感じるのかを，ある程度の精度で推定することができた．これにより，特定の難しさを持つ問題や面白い問題を任意に生成することができると考えられる．これらはガイスターのみならず，様々なゲームにも共通する点もあることから，業界全体への貢献が期待できる．

目 次

図目次

表 目 次

第1章　はじめに

近年，ハードウェアの進歩や新たなアルゴリズムの提案により，ゲームにおける人工知能の発展は目覚ましい．人工知能は人間に代わる意思決定の基盤として様々な場で利用され社会問題の解決に貢献しており，既に我々の生活には欠かせないものとなっている．ゲームにおける人工知能は，エージェントの意思決定や状態生成などデジタル・アナログゲーム共に多くの研究が行われている．多くのゲームは定義が単純明確で再現コストも低いこと，さらにプレイヤーの意思決定の繰り返しによって進行していくという特徴を持つことから，人工知能を人間の知能と近づけていくためにエージェント AI は非常に優れた人工知能の題材とされている．そのためゲーム人工知能の世界から発展したアルゴリズムも多く，特に Google 社が開発した『AlphaGo[1]』が世界ランキング首位のプレイヤーに勝利するなど，囲碁を始めとした完全情報ゲームにおいてのエージェント AI は大きな成果を挙げている．

そういったエージェント AI の研究が隆盛する一方で，人間の感性を理解することでより AI を人間の身近な存在とするべく，近年は人間を楽しませるコンテンツ生成の AI についての研究も盛んに行われている．ゲームの状態などのコンテンツ生成はこれまで人間が手作業で作るしかなかったが，これを AI に置き換えることで人間の負担を大きく減らし，更には多様性やリアルタイム性を持った生成も可能としている．コンテンツはゲームの核となるもので，人間はゲームの面白さの大半をコンテンツから感じ取ると考えられる．そのため面白さとコンテンツ生成は非常に相性がよく，併せて行われている研究も多い．例として石飛らが面白さに重きを置いた詰将棋問題の自動生成 [2] について研究している．これはボードゲームのコンテンツとしては特に有名である詰将棋問題の生成手法と，問題の面白さに何が関わっているのかを分析した研究である．他にも，広瀬らは逆算法による詰将棋問題の生成 [3] を行い，ただ生成するだけでなく芸術性を評価する機構を組み込んだシステムを開発した．このように将棋などの完全情報ゲームのコンテンツ生成についての研究は広く行われている．

その一方で，不完全情報ゲームにおけるプレイヤーを楽しませる研究やコンテンツ生成についての研究は比較的数が少ないと感じる．不完全情報ゲームとは，将棋や囲碁のように全てのプレイヤーが全ての情報を与えられている完全情報ゲームとは違い，トランプのババ抜きや麻雀など与えられない情報があるゲームを指す．不完全情報ゲームは与えられた情報から与えられていない情報を推測する必要があるため，一般的に人工知能の開発では完全情報ゲームについてのものより

難しいとされている．実社会における意思決定も，全ての情報が意思決定者に与えられない場合が殆どである．そのため不完全情報ゲームについての研究は，次なる社会問題解決法の糸口として多くの期待がかけられている．

世界的に親しまれている将棋やチェスなどのゲームに近いルールを持つ『ガイスター [4]』 というボードゲームがある．ガイスターは互いのプレイヤーが 2 種 4 個ずつの駒を用いて遊ぶ二人零和確定不完全情報ゲームである．駒はそれぞれの特徴を持つ青色と赤色が存在する．ゲームの大きな特徴として対戦相手の駒の種類がわからないことが挙げられ，駒の種類をそれまでの動き等から予測する必要がある．そのことから，簡潔なルールでありながら心理戦の要素が多い．

このゲームの初心者にとって，対戦相手の駒種の予測や自分の駒種を誤認させるテクニックなどの技術を身に着けることは難しい．さらに考えることが多く，本来では勝利が確定している盤面の見逃しなどが発生しうる．これらのことから初心者は理詰めよりも運や心理戦に頼りがちで，そのことが上達を困難にさせる．そのため上達を促すためには，練習や教育のような技術力向上支援を行う環境の提供が必要となる．将棋には娯楽と実戦における技術力向上の手段として用いられる詰将棋というパズルがある．似たボードゲームであるガイスターにおいてもこのようなパズルは有用だと考える．このようなパズルは様々なものが考えられ，ガイスターにおいても『駒色予測問題』や『次の一手問題』などが作りえるが，まずは必勝盤面の見逃しの克服が重要と考え詰め問題に着目した．

本研究の目的は『ガイスターの部分問題として切り出せるものにはどんなものがあるか』『面白く適切な難易度の詰め問題を高速に生成するにはどうしたらよいか』を解明することである．そのためにまず，現存する部分問題コンテンツを調査し，ガイスターに適切なルールを持った新コンテンツを提案しなければならない．そしてコンテンツを生成するに際し，様々な難易度･面白さを持つものを多く生成する必要がある．その問題を教育に適用する場合，面白い問題やプレイヤーに適した特定難易度の問題を提供することが求められ，面白い・難しい問題が持つ特徴を知る必要がある．

なお，本研究の内容は第 41 回ゲーム情報学（GI）研究発表会にて発表した “不完全情報ゲーム『ガイスター』における 2 種の詰め問題の提案と考察”[5] および，ゲームプログラミングワークショップ 2019 にて発表した “難しい詰めガイスター問題の生成法”[6] らを基に整理・加筆したものである．

本章に続き，第 2 章では本研究で対象とする不完全情報ゲーム『ガイスター』のルール説明や戦略，第 3 章ではガイスターの AI プレイヤーとパズル生成・評価に関する研究の紹介を行う．第 4 章では本研究で提案する『詰めガイスター問題』について，第 5 章・第 6 章ではアプローチの中から本研究で主に試したランダム生成法，逆順生成法のアルゴリズムの説明と生成実験について述べる．第 7 章では生成した問題を用いた被験者実験についての説明と，面白さ・難しさ推定について述べる．第 8 章では他に考えられる生成アプローチについて述べる．最後に第 9 章で本研究の総括と今後の展望・課題を述べる．

第2章　ガイスター

2.1　ルール

ガイスター（Geister）[4] は，プレイヤーが青駒と赤駒，2 種類の駒をそれぞれ 4 つずつ用いて遊ぶ二人零和確定不完全情報ゲームである．盤面例を図 2.1 に示す．本ゲームの特徴として，プレイヤーは自身の駒色を確認できるが，対戦相手の駒色を確認できないことが挙げられる．盤面は縦横に 6 マス合計 36 マスとなっており，自身から見て一番奥の左右端マスをそのプレイヤーの脱出口としている．

各プレイヤーはゲーム開始前に自分から見て手前 2 × 4 の所定の範囲に 8 駒を自由に配置する．開始後は自身の手番で自身の駒のうち 1 つを縦横 4 方向のいずれかに 1 マス動かす．自身の駒が既にある方向に動かすことはできないが，動かした先に対戦相手の駒があればそれを取ることができる．その際に取られた駒の色は開示される．そうして勝利条件を目指して手番をお互いに繰り返す．勝利条件は以下の 3 種類である．

勝利条件 1. 自分の青駒を脱出させる（脱出口でもう 1 手番使う）
勝利条件 2. 相手の青駒を全て取る
勝利条件 3. 自分の赤駒を全て取らせる

よって敗北条件は以下のようになる．

敗北条件 1. 相手の青駒に脱出される
敗北条件 2. 自分の青駒が全て取られる
敗北条件 3. 相手の赤駒を全て取ってしまう

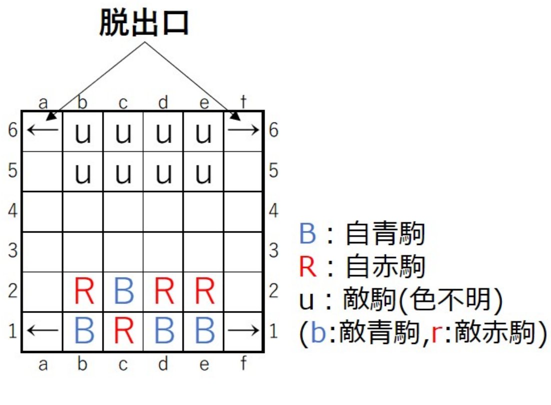


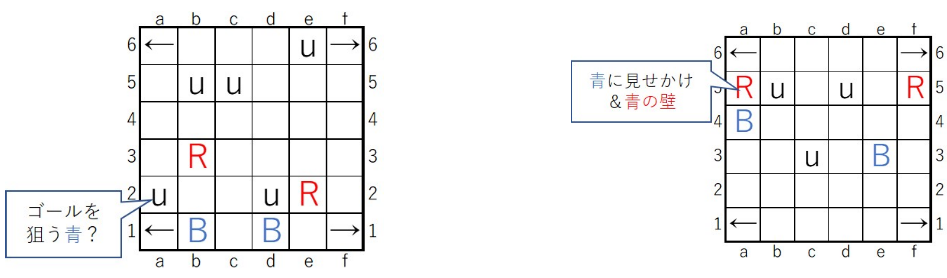
図 2.1: 盤面例

2.2 基本戦略

本ゲームにおける戦略は敵青駒の脱出を防ぎつつ，自青駒を脱出口に向けて動かすことを基本とする．だからと言って，前に出てくる敵駒を手あたり次第に取る，自青駒を無策に脱出口に向かわせる，ことは愚策となる．前者の場合，取った駒がいずれも敵赤駒で，それが 4 つになってしまうと対戦相手に勝利条件 3 を満たされてしまう．後者の場合，突出させた自青駒が孤立してしまうと，まずそのまま脱出できず取られてしまうだろう．これらのことから，対戦相手の駒色予測と，赤駒によるカモフラージュなど駒色を誤認させるテクニックが要求される．

例として図 2.2(a) では，a2 の敵駒は脱出口に近づいており，脱出を狙う敵青駒だと考えるほうが安全である．一方で d2 の敵駒は取られることを恐れずこちらの陣を荒らすような位置におり，取られても損が少ない敵赤駒だと予想できる．図 2.2(b) では，自赤駒をまるで脱出を狙う自青駒かのように振舞わせることで本命の脱出手段を隠している．脱出を狙うように見せる以外にも，取られないよう逃げるように動かすことでも自青駒に見せられるだろう．しかし対戦相手も同じ人間や賢い AI であれば，勿論同じように読んでくるだろう．

これらのことからガイスターは簡潔なルールでありながら，人間同士のプレイでは論理だけでなく勘やハッタリを駆使した複雑な心理戦になることが多い．一方でガイスターは対戦相手の駒が非公開である確率的ゲームであることから，本質的にはこのゲームにおける最適戦略は混合戦略となる．混合戦略とは，複数の戦略を確率的に混合して戦略を選択するプレイの仕方である．例としてじゃんけんでは,『この手を出すのが最適である』といったことは存在せず，相手の戦略によって自分が一番利得を得られる戦略は変わってくる [7]．よって自分の手を相手に読まれないように各手 1/3 の確率で出すことが最適と言える．ガイスターの場合は,『最初の駒配置をいつも同じにするとバレてつけこまれるからいろんなパターンを使う』『赤駒をつっこませて青駒が逃げ回るのが普通だけど，たまに逆のことをやってみる』といった混合戦略が必要となってくる．



(a) (b)

図 2.2: テクニックの例

2.3　ガイスターの面白さと難しさ

ガイスターにおける面白い点であり醍醐味は，複雑な心理戦にあると考える．前節で述べたような駒色の予測や誤認させるテクニックなどの心理戦は，ある一盤面からではなくそれまでの盤面の遷移履歴や，複数プレイを重ねる場合プレイ履歴も影響する．そのような積み重ねがゲームに影響するというのは面白い点であろう．

一方でガイスターにおける一番の難しい点は，様々な盤面や動きを想定する必要があるため考えることが膨大になってしまうことだろう．例として，非公開情報である対戦相手の初期配置は 70 通り，ゲーム中対戦相手の取れる行動は最高 32 通りもある．そのため全ての状況を考えず，予測を絡めていくこととなる．

これらのことから初心者は醍醐味である心理戦に重きを置きすぎてしまい，考えることの多さから理詰めを放棄しがちである．そのことから本来勝てるはずの盤面の見逃しなどが起こりがちで，その見逃しにすら気づかずにいると上達にも繋がらない．心理戦の度合いはあれど，これは様々なゲームにも共通することで，ゲームの上達における大きな課題である．

第3章　先行・関連研究

本研究に関連する研究を，ガイスターの AI プレイヤー，パズル生成・評価に分けて紹介する．

3.1　ガイスターの AI プレイヤー

ガイスターの AI プレイヤーについては現状ガイスターの研究の中でも特に多く見られ様々な手法が試されているが，高水準な強さを持つ AI プレイヤーは未だ確認できていない．多くの研究では，着手可能な手からランダムに選択する『ランダム AI』，ひたすら脱出のため脱出口へ向かわせる『猪突猛進 AI』が強さの比較対象として用いられている．さらに高レベルな比較対象として，モンテカルロ法を用いた『モンテカルロ法 AI』もよく用いられている．

不完全情報ゲームにおけるモンテカルロ法は，状態を仮定し状態それぞれにプレイアウトを割り振る必要がある．そのためプレイアウト回数不足や，評価のまとめ方などの問題が存在する．そのことから，モンテカルロ法 AI をただ比較対象とするだけでなく，何らかの要素を追加してそれらの問題を緩和する研究も多く行われている．

三塩・小谷は過去の着手等から不完全情報の推定を行う UPP というアルゴリズムを提案し，モンテカルロ法を用いたガイスター AI に応用した [8]．モンテカルロ法に存在する様々な問題に対して，過去のプレイアウト結果を利用することで可能性の高い状態を推測するアルゴリズム UPP を用いることで，問題の影響を緩和することを目指した．結果，猪突猛進 AI には圧勝し，UPP を用いなかったモンテカルロ法 AI プレイヤーに勝ち越し，ガイスターにおける UPP の有効性を示した．

佐藤は強化学習を用い，自己対戦を繰り返すことで行動価値関数を改善し，強い AI を開発することを試みた [9]．ある局面においてある行動を行った場合の勝率見積もりを計算する行動価値関数 $Q(S_t, a)$ では，これまでの着手から色予測を行うガイスターおいては推測を考慮した行動価値関数を求めることができない．そこで佐藤は，盤面の過去情報を駒の動きなどの特徴から求められる推測値を用いて近似し，特徴 f とした行動価値関数 $Q(f, S_t, a)$ を行動価値関数として用いた．そしてゲームが引き分けになりにくいような着手制限を与え，強化学習を行った．さらに必勝手の見逃しを防ぐために Df-pn アルゴリズムを用いた必勝手探索を用いた．結果，前述したモンテカルロ法 AI プレイヤーに UCT アルゴリズムを加えた

ものに対して勝ち越すことができた．

末續・織田はルールベースを用いて行動を決定する AI を開発した [10]．これは対戦相手の駒の青らしさを過去の動きや位置から評価し，それを基に予め決められたアルゴリズムで行動を決定するというものである．駒色評価の例として，脱出を狙うような前進や，相手駒に取られないように接敵数を減らす動きをする駒を青らしい，逆に積極的に接敵数を増やす動きをする駒を赤らしい，と評価している．さらに必勝手検索を実装しており，必勝形をテーブルにまとめて着手選択の際にテーブルと一致するかを調べることで必勝手かどうかを調べた．

川上・橋本は完全情報ゲームとして探索を行うことで強さが向上するかを検証した [11]．探索する盤面を互いの駒色が見えるものとして，ありえる完全情報盤面を探索，集計することで手を選ぶという手法を取った．探索には MinMax 探索を用いており，評価関数に『青駒の個数差』『駒の位置』を用いている．ガイスターにおいては青駒が相手より多ければ多いほど有利と考えられる．駒位置については，自分の各駒において敵陣に近いほどプラス，相手の各駒において自陣に近いほどマイナスとしている．さらに完全情報盤面の敵駒に紫駒 [9] を適用した実験も行った．紫駒を応用した MinMax 法は非常に強力で，評価関数を改善する前の AI プレイヤーに圧勝するほどの高い性能を持った．

Sehar らは機械学習を用い対戦相手の駒色の予測を行った [12]．1400 ゲーム以上の競技リプレイデータから特徴を抽出し，様々な機械学習アルゴリズムを適用して推論を行った．結果，一番高い性能を出したアルゴリズムでは推定精度 56% を超えるなど高い精度を実現した．

3.2　パズル生成・評価

本研究では，ガイスターの部分問題として詰めガイスター問題というパズルを提案・生成する．パズル生成や，パズルの面白さや難しさを推定・評価する研究は盛んに行われており，パズルのカテゴリに合わせて様々な手法が試されている．

広瀬らは詰め手順の逆算による詰将棋問題生成を行った [3]．詰将棋における解答プログラムでは，不詰めの状態から盤面を修正していくのが難しいとされる．そのため，既に詰めが証明できている盤面を基本とし肉付けを行う逆算法を用いている．さらに生成された問題を評価するために，内容のよさ，完成度の高さ，解き難さを評価基準とする評価関数を用いた．結果，9 手詰までの問題を生成でき，専門誌で紹介されるほどの高評価な問題を得た．しかし初期配置への依存が大きい，作品に狙いを持たせにくい，などの課題があった．

石飛らは証明数探索に用いられる証明数と反証数を用いて詰将棋問題の評価を行った [2]．詰将棋問題に対して探索を行い，高評価とされている問題と一般的な問題に対して証明数と反証数にどのような違いがあるかを解析した．その結果，証明数と反証数は面白さと関わりがあることを示した．

藤原らは 3×3 のマスに数字を入れていくパズルである数独の良質な問題を自動生成した [13][14]．自動生成される問題はこれまで品質に問題があったが，藤原らは問題作成の高速性と品質を両立させることを目指した．人工知能と知識工学をいかしてパズル作家の考え方を取り入れ，問題を評価して良問を選んでいる．現在は進化計算を活用するフレームワーク『ダーウィン』により，世界最大の数独問題を生成しギネス記録に認定された [15]．

大町らは不完全情報ゲームである上海ゲームを対象とし，適切な難易度かつ理不尽さが少ないインスタンスの生成を行った [16]．まずインスタンス生成をランダムに行い，それを性能の異なる複数の評価機で評価するといった手順を取る．評価機として未知牌の仮定数と探索深さが違う 2 つの仮想プレイヤーを用いて解答率を求め，解答率の違いの分布からインスタンスの分類を行った．結果,『確定行動が多く簡単すぎる』『読みが多く必要だが運要素が少なく理不尽でない』など望ましい問題と望ましくない問題を分けることを可能にした．

牧田らはぷよぷよにおける連鎖構成力を向上させるために用いられているコンテンツである『なぞぷよ』を対象とし，多様で面白い問題の提供システムを提案した [17]．問題の生成法として，ある程度の制約をつけつつもランダムにぷよを配置していくランダム生成法，そして最後の盤面から下から押し上げるように新たなぷよを配置していく手順を繰り返すことで盤面を作り出す逆向き生成法を用いている．問題の提供システムとして，まずプレイヤーに問題を出題しプレイデータを収集する，そのデータを基に教師あり学習を行い面白さと難易度推定モデルを生成，モデルによって選ばれた更に問題を解かせデータを収集しデータからモデルを更新，といった手順を繰り返す．これによりモデルの性能を上げながら面白く適切な難易度を持つ新たな問題を提供することができる．生成された問題の

中には，人間でも作るのが難しいと思われるような難しい問題もあり，研究の価値を示した．

及川らはテトリスにおける重要技術『T-spin』の詰め問題を生成した [18][19]．詰め T-spin 問題は，T-spin の構築パターン（定石型）を覚える練習をさせることを目的としているコンテンツである．生成法として，まず T-spin が可能な地形を用意し，そこから解かせたい手数分のテトロミノを抜き出すという逆向き生成法を用いている．そして面白さの推定を行い，22 個の盤面特徴量を用いて面白さの平均絶対誤差 0.27（5 段階評価）の推定を可能とした．さらにその推定モデルを用いることで，面白い問題のみを選別し，その問題を用いて練習させることで実際に T-spin 技術が向上し，勝率も上昇したことを被験者実験により示した．

Barbara らはパズルにおける PCG(Procedural Content Generation）について調査した [20]．まずパズルの特徴や目的を基に 11 のカテゴリに分類，そのカテゴリごとに使用されているパズルの生成・評価法について調査した．本研究と近いと思われる Sliding Puzzle で主に使用されている生成手法として，パズルの最後の状態から逆順に戻していくものがある．さらに評価方法として，エージェント AI に解かせることで難易度等の評価指標を計測している．本論ではパズルの評価指標として難しさ，多様性，鮮度，美学などが広く使われていると述べている．そして，最後にこれからの PCG 需要や，一般性を持たせることの難しさ等を示した．

第4章　詰めガイスター問題

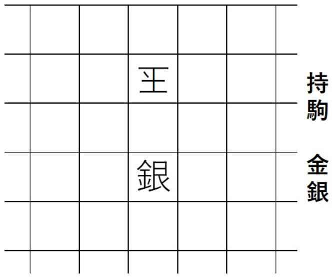
これまでの議論を踏まえ，本章ではガイスターの技量向上や普及を支援するため，ガイスターにおける部分問題となる『詰めガイスター問題』を提案する．

4.1　詰めガイスター問題の背景

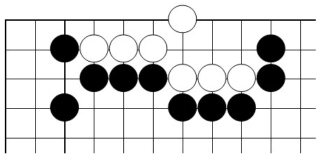
ボードゲームにおけるコンテンツとして，将棋を題材とした『詰将棋 [21](図 4.1(a))』や，碁を題材とした『詰碁 [22](図 4.1(b))』が存在する．一般的な詰将棋と通常の将棋の違う点として，王が存在するのは相手側のみで，自分側が詰まされることは想定しなくてもよい，持ち駒を使い切るように設計する必要がある，最短の勝ち方が本質的に一通りになるように設計しなくてはならない，などが挙げられる．詰碁は通常の碁と比べて目的が違い，通常の碁は所謂陣取りゲームのようなものだが，詰碁は石の死活を問う問題となっており，目的がはっきりとしていることが特徴である．さらにこれらは題材となったゲームよりサイズが小さい場合が多いことも特徴的である．

これらのコンテンツはパズルとしての楽しみを提供するだけでなく，解くことで詰ませる技術の向上は勿論，頭の中で駒や石を動かす訓練にもなるとされており，それが実際の対局での技術向上に繋がる．トップ棋士も訓練に使用しており，特に詰将棋を藤井聡太七段は得意にしていることが有名である．さらにこのようなコンテンツは題材元のゲームの普及という面でも大きな影響を与えている．題材となったゲームはどれも対戦相手がいてこそ成立するものであるが，詰め問題のようなものは本やアプリ上で遊べるものが多く，一人で遊べることが大きな利点である．さらに短時間で遊べることから暇つぶしや軽い頭の体操としても非常に優れたものとなっている．

このように詰将棋などの詰め問題は有用であるが，ガイスターを含め多くのボードゲームにおいてこのようなコンテンツは広く用いられていない．多くのゲームにおいて，実力をつけるために問題を解かせることは非常に有用だと考えられる．特に，先述したようにガイスターにおいて，初心者は心理戦に重きを置きすぎて理詰めを放棄，本来勝てる盤面を見逃してしまうことが多々ある．その見逃しの克服に役立つであろう．



(a) 詰将棋



(b) 詰碁

図 4.1: 詰め問題コンテンツ

4.2 詰めガイスター問題の定義

詰めガイスター問題を定義する前に，広く普及されている詰将棋を比較対象として挙げる．前述したように一般的な詰将棋と通常の将棋の違う点として，自分側が詰まされることは想定しなくてもよい，持ち駒を使い切るように設計，最短の勝ち方は本質的に一通り，などが挙げられる．これは詰将棋が実用性だけでなく，美しさを求められた結果，実戦とは異なる特徴を持たせられたためだと考えられる．

それに対し，今回提案する詰めガイスター問題はプレイヤーに解いて楽しんでもらうことと実力を向上させることを重視する．そのため前述した詰将棋のような特徴を持たせる必要はなく，より実戦的かつ高い自由度を持たせて設計することができる．以上のことより，本研究における詰めガイスター問題を以下のように定義する．

詰めガイスター問題はボードゲーム『ガイスター』のルールを用いたパズルであり，通常のガイスターのルールに加え以下のルールを持つ．

- 先手側が勝利する最短の手順を見つける
- 問題として盤面，後手側の青赤それぞれの駒数，最短の手数が公開される
- 最善，最短手順は複数種類ある場合がある
- 問題には必ず先手側の必勝手が存在する

この問題において，先手側の目的は非公開である後手側がどんな駒の色組合せであろうと，どんな行動であろうと勝利条件を満たすことができる最短の手順を見つけることが目的となる．

本研究では大きくわけて 2 種類の問題を生成した．1 つは後手側の駒色を完全に非公開とすることで実際のガイスターのルールに近づけた『一般問題』．もう 1 つは一部の駒色を公開することでゲーム終盤の駒色予測を反映し，より実戦的とした『一部公開問題』である．

以降，詰めガイスター問題は図 4.2 のような形で表現する．図 4.2(a) は一般問題であり，図 4.2(b) が一部公開問題である．盤面上部には後手側の残駒数と最短の手数を示している．残駒数は青が b，赤が r である．盤面には先手側の駒として青赤それぞれ B と R，後手側の色不明駒を u としている．さらに一部公開問題の場合はさらに公開された青駒を b，赤駒を r としている．最短の手数とは，いわゆる X 手詰めで，後手の手数を含めた数を示している．

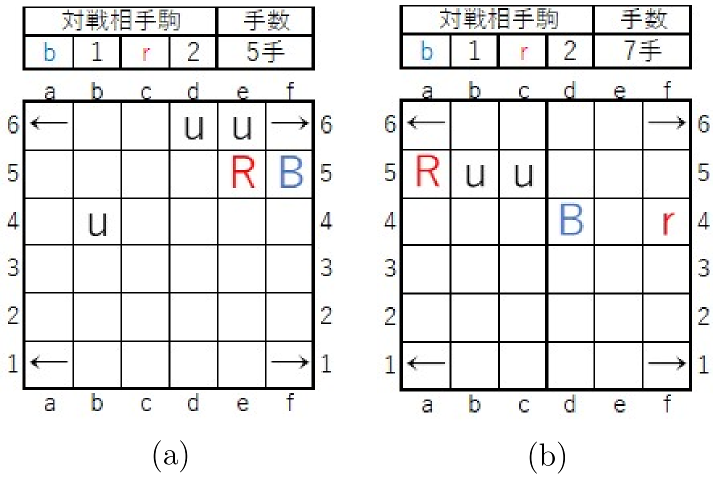


図 4.2: 詰めガイスター問題例

ここで図 4.2(a) の解き方を図 4.3 に沿って説明する．初期盤面は先手駒がそれぞれ 1 つずつ，後手駒が青駒 1 つ，赤駒 2 つである．先手側は，脱出口に近い自青駒を脱出させたいところではあるが，f5 から f6 に単純に動かすと [a] 次の後手番で取られてしまう [b]．そうすると勝利条件 2 を満たされ敗北してしまう．そこで先手番 1 手目で自赤駒を e5 から e6 に動かした場合 [c]，次の後手番でもし d6 から e6 に動かして敵赤駒を取ってしまうと先手側は勝利条件 3 を満たし勝利することができる [d]．そのため後手側は別の動きをするしかないのだが，そうしている間に先手側は自青駒を脱出口に近づけ [e]，次の手番を使って脱出することができる [f]．このように最短だと 2 手で決着 [d] は着くのだが，後手番がどんな動きであっても勝てる手順を見つける必要があるため，本問題では最短手数 5 手である．

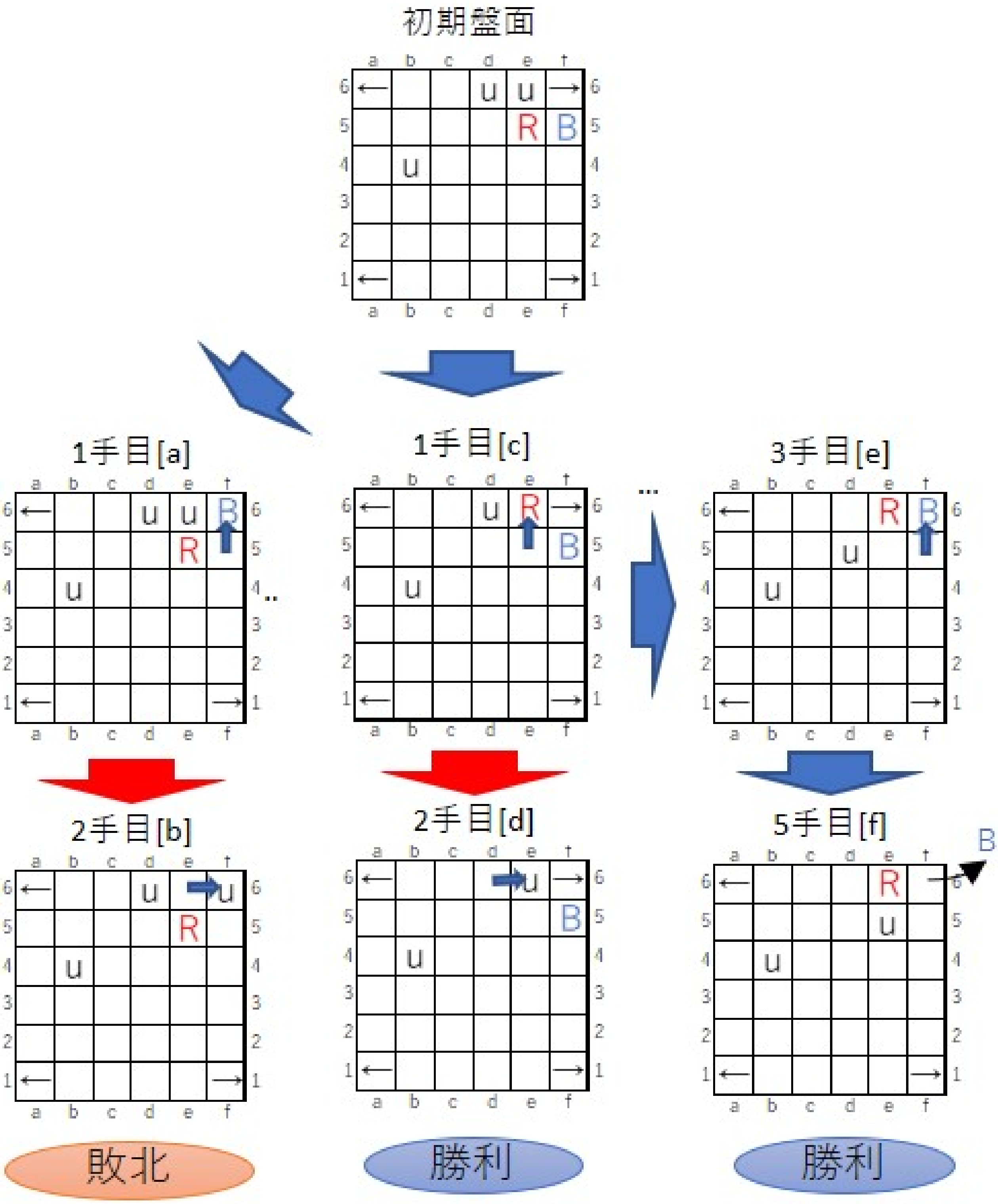


図 4.3: 図 4.2(a) の解き方

次に図 4.2(b) の解き方を図 4.4 に沿って説明する．問題では，f4 に後手側の赤駒が公開されている．先手側が勝利条件 2 を満たすためには b5，c5 の色不明駒を両方取ればよい．まず a5 の赤駒を b5 に移動させ駒を取る [a]．後手側は c5 の駒も取られたら敗北するため逃げる必要があるのだが，c4 か d5 に移動すると [b]d4 の青駒に取られてしまい [c]，b5 に移動すると最後の赤駒を取ってしまう [d]．よって c6 に逃げることになる [e]．先手側は次に d4 の青駒を d5 に移動する [f]．すると後手側は c6 の駒を 3 方向のいずれかに動かしても先手側に取られてしまう．よって f4 の赤駒を動かすしかないのだが，先手側は次の手番で b5 の赤駒を b6 に動かすと [g]，後手側は次に c6 の駒を動かしても動かさなくても先手側に取られることになるため，7 手で先手側の勝ちである [h]．

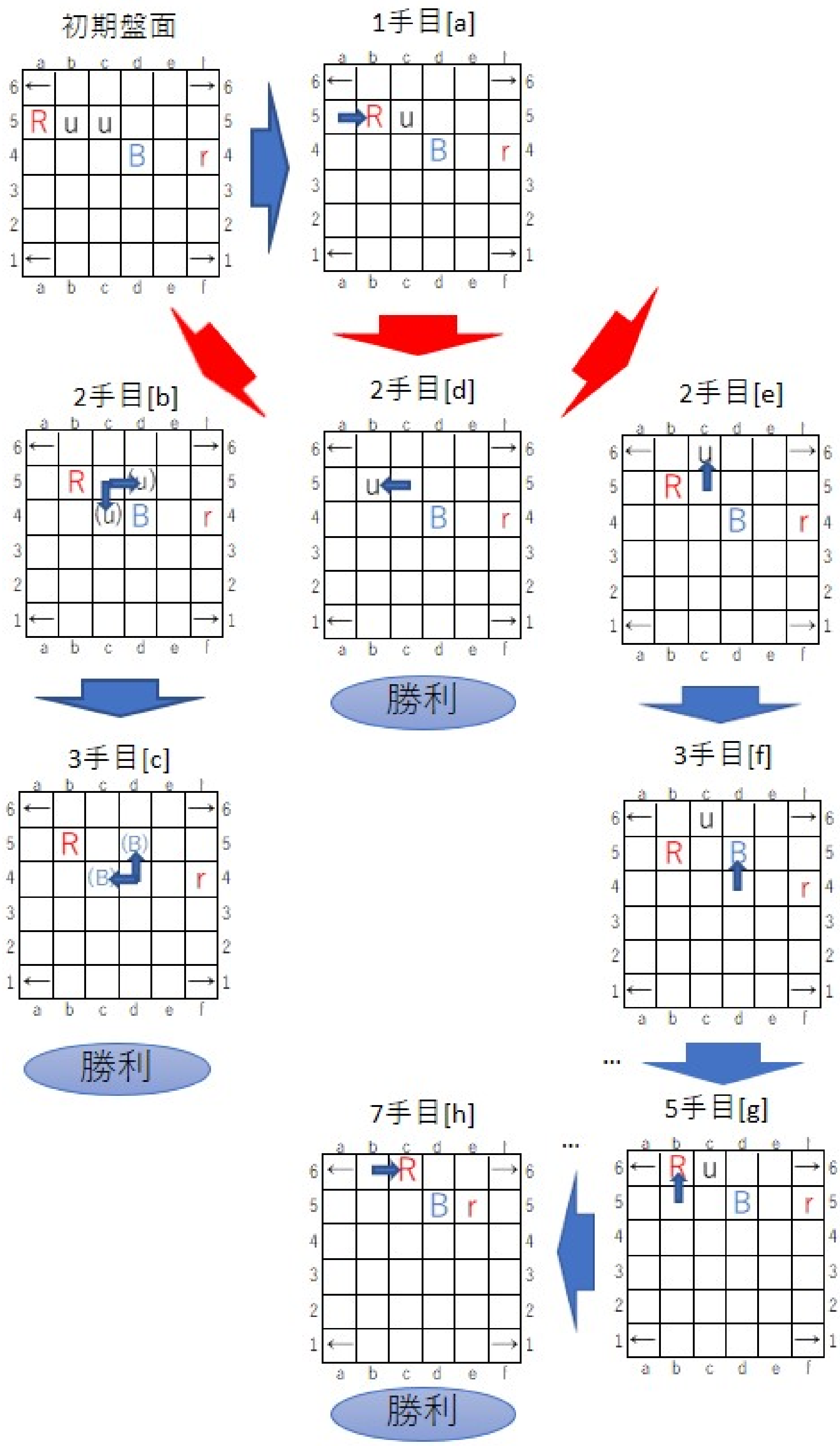


図 4.4: 図 4.2(b) の解き方

4.3 目指す勝利条件と問題の種類

ガイスターには前述した 3 種類の勝利条件がある．各詰めガイスター問題は，それらによって構成された問題になる．ここでは各勝利条件を目指した詰めガイスター問題がどのようなものになるのかを考察していく．

勝利条件 1．青駒を脱出させる

脱出口からの青駒の脱出により満たされる勝利条件である．実際の問題においては，青駒を脱出口に向かわせるだけでいい単純すぎる問題は極僅かで，敵の足止めや 2 駒以上でその敵の守りを突破する，などの要素が入ってくるため手数は増えてくる．この条件の特徴として，満たす際に基本的に後手側の色組合せに依存しない．さらに他の勝利条件と組み合わされて頻繁に登場する要素であることが挙げられる．

勝利条件 2．青駒を全て取る

詰めガイスター問題は後手側の駒色組合せに依存しない手順を見つけなければならない．そのため常に最悪な状況を想定する必要があり，後手駒を取った場合それは赤駒であると考えなければならない．なぜなら，それが青駒だったときに比べ，赤駒だったときには先手側に一切の得がないからである．つまり一般問題においては，先手側がこの勝利条件 2 を満たすために青駒を全て取ろうとしても，その前に必ず後手側の赤駒を全て取ってしまい勝利条件 3 を満たされることになるため，この勝利条件を用いた問題を生成することは不可能である．

一方で，一部公開問題においてこの勝利条件は重要な役割を果たすことがある．詳しくは 4.5.2 節で述べるが，後手赤駒が 1 つでも公開されている場合，この条件を満たせる可能性がある．よって，そうでない場合と比べ，敵駒を追い詰め取っていくという戦略が必要となってくる．

勝利条件 3．赤駒を全て取らせる

詰めガイスターでプレイヤーが満たすべき条件の一つは，後手側がどんな動きをしようとも勝てる最短手順を見つけることである．そのため後手側の手順は最善，最長を目指すものを考えることになる．後手側にとっては先手側の駒色が全てわかっていると仮定すべきで，つまり先手側の最後の赤駒は取ってもらえないということである．よってこの勝利条件によってしか勝利できない問題を生成することは不可能である．

しかしながらこの勝利条件を利用することで後手側の行動を制限することは可能である．例として図 4.5 の問題のような状況を考える．先手側の赤駒が最後の 1 つで，後手側が脱出（勝利条件 1）を狙っている状況である．そこで最後の赤駒である a2 を狙われている脱出口である a1 に配置すると，後手側は脱出のために赤駒を突破するしかないが，取ってしまうと先手側に勝利条件 3 を満たされ敗北してしまう．よって f1 からの脱出を狙うことになるのだが，その時間を使って先手側は脱出することが出来る．このように勝利条件 3 を利用することで行動を制限することは実戦でも非常に有効なテクニックである．

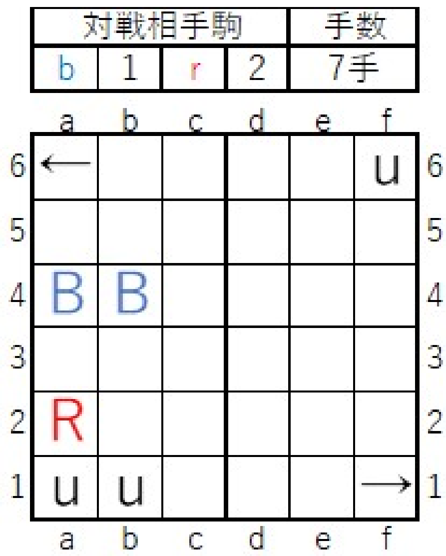


図 4.5: 勝利条件 3 の例

以上の考察を基に，問題を『一般問題』『一部公開問題』の二つに大別，更にその中で構成されている勝利条件等を基に，問題に含まれている要素を紹介する．

4.4　問題カテゴリ『一般問題』

実際のガイスターのルールに則り，後手駒の色情報を非公開とした詰め問題である．4.3 節で述べたように一般問題は原理上，勝利条件 2 を満たす問題や，勝利条件 3 によってしか勝利できない問題は生成されない．そのため勝利条件 1 を目指す問題のみとなる．

各問題を構成する要素として，ここでは先手側の青駒を脱出させることで勝利する『青駒脱出』の要素と，先手側の赤駒を利用する『赤駒壁利用』要素を紹介する．

4.4.1　青駒脱出

勝利条件 1（青駒の脱出）を利用する要素である．以下にこの要素によって構成された問題例（図 4.6，図 4.7）と解き方を示す．

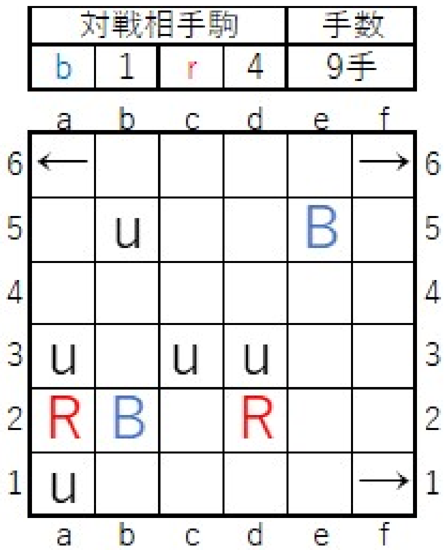


図 4.6: 『青駒脱出』を用いた問題例 1

図 4.6 の問題を見てみると，脱出口に近い e5 の青駒を脱出させることが目標だと予想できる．さらに先手側の一番脱出口に近い駒 e5 よりも，a1 の後手駒の方が脱出口に近いことがわかる．このことから，後手側の脱出を阻止しつつ，先手側の青駒を脱出させる問題であることが予想できる．手順としてはまず a2 の赤駒を a1 に移動させる．そうすると後手側の a1 駒を取ることができるので，眼前の脅威はなくなる．後手側が次に脱出口に近い a3 駒を a2 に移動してきても，次の手で b2 青駒か a1 赤駒を用い，取ることができる．詰めガイスター問題における最短手数は後手側の動きによって一番引き延ばされた場合になるため，後手側は a3 → a2

と動かし，先手側は b2（a1）→ a2，その後 e5 の青駒を f6 の脱出口から脱出させればよい．これで 9 手である．このように後手側の脱出を遅らせつつ，先手側の脱出を狙うのがガイスターの基本である．

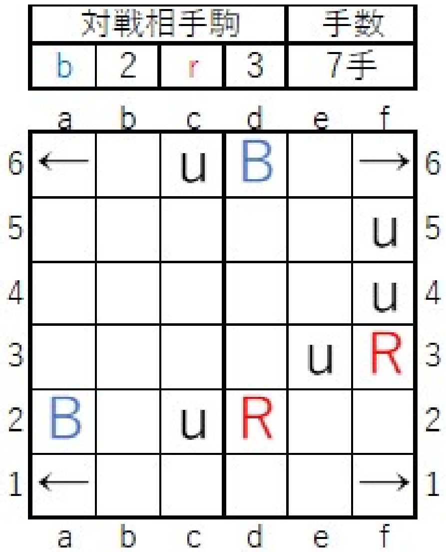


図 4.7: 『青駒脱出』を用いた問題例 2

図 4.7 の問題を見てみると，d6 の青駒を脱出させる問題であることがすぐに予想できる．しかし，青駒に一番近い脱出口である f6 に隣接している f5 に後手側の駒があるため，真っ直ぐに向かうわけにはいかない．一方，もう片方の脱出口である a6 には簡単に向かえてしまう．c6 に後手側の駒があるが，初手で d6 青駒によって取ることができてしまい，そのまま真っ直ぐ脱出口に向かえてしまう．このように，ただ一番近いだけの脱出口ではなく，他のルートを探す必要があるような問題も存在する．

4.4.2　赤駒壁利用

勝利条件 3（赤駒を全て取らせる）を利用する要素である．以下にこの要素によって構成された問題例（図 4.8，図 4.9）と解き方を示す．4.3 節で述べたように，勝利条件 3 は，これのみを満たす問題は生成されない．よって以下で紹介する問題には 4.4.1 の『青駒脱出』の要素も含んでいる．

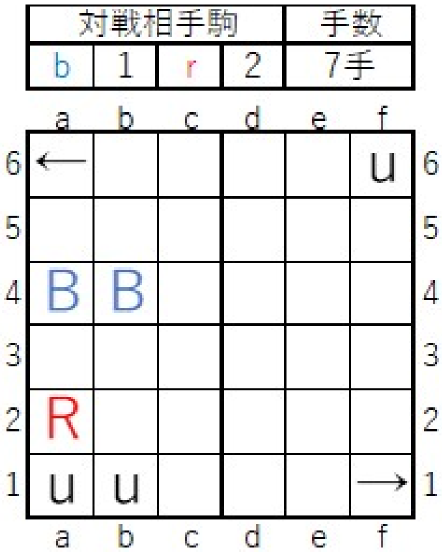


図 4.8: 『赤駒壁利用』を用いた問題例 1

図 4.8 の問題は 4.3 節での例と同じ問題ではあるが，ここではより詳しく解き方を示す．問題を見てみると，脱出口に近い a4 の青駒を脱出させること，a1b1 の後手側の駒を脱出させないことが目標だと予想できる．まずは a1 駒の脱出を阻止するべく赤駒を a2 から a1 に移動させて後手側の駒を取る．次に後手番で b1 から a1 に移動させてしまえば先手側の駒は取られ，そのままゴールされるように思えるが，取られる赤駒は先手側最後の赤駒であるため，勝利条件 3 を満たされ後手側は敗北してしまう．そのため，取ることはできず f1 の脱出口を目指さざるを得なくなる．このように，最後の赤駒を脱出口に置き，蓋をする戦法は実戦でも有効であり後手の行動に大きな制限をかけることができる．

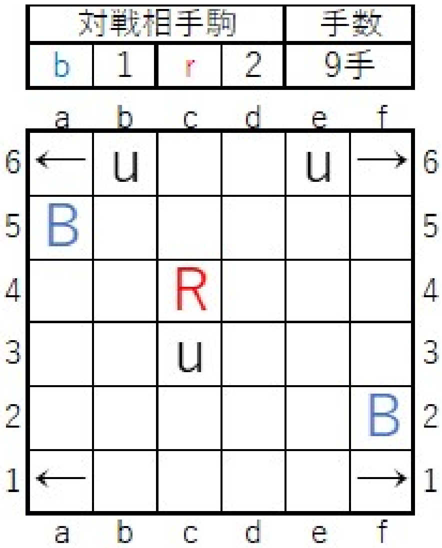


図 4.9: 『赤駒壁利用』を用いた問題例 2

次に図 4.9 の問題を見てみる．一見して a5 の青駒を a6 から脱出させることを目指すことが予想できる．しかし，b6 に後手側の駒があるため，脱出口に直行すれば取られてしまう．そのため脱出口に移動する前に b6 の駒を取るか移動させる必要がある．ここで先手側の赤駒は最後 1 つであるため，取られると勝利条件 3 を満たすことを利用する．赤駒を c4 から c5，c6 と動かしていくことで b6 の駒を取ることができる．その間，後手がどう b6 の駒を動かそうとも，先手側は赤駒か a5 の青駒で取ることができる．そうして b6 の駒を取った後，a5 の青駒で a6 の脱出口で脱出すればよい．このように，取られることが得になる最後の赤駒を後手側の駒に向かわせ道を開く，頭数を減らす戦法は実戦でも非常に有効である．

4.4.3 考察

後手側の駒が実ゲーム同様非公開である一般問題を提案した．構成要素の一つである『青駒脱出』を用いた問題は，後手側の脱出を阻止しつつ脱出を目指すものが基本であるが,『赤駒壁利用』を併せた問題はさらに複雑化して，最後の赤駒で脱出口に蓋をする問題や，後手側の駒に突撃させる問題が存在する．これらはガイスターにおいて非常に有効な戦法であるため解かせる価値は十分にあると考えられる．

しかし一般問題においては勝利条件 1,3 を利用した問題しか原理的に生成されないため，多様性が欠如している．そのうえ情報を完全に非公開としているため，実ゲームでの駒予測に基づく練習ができない．そこで，後述する『一部公開問題』を考案した．

4.5　問題カテゴリ『一部公開問題』

実際のガイスターのルールと違い，後手側の駒色情報を一部公開した問題である．ガイスターは対戦相手の動かし方から色組み合わせを予測するゲームである．例として，ゴールを積極的に狙ってきつつも取られないように動く敵駒は青駒，取られることを恐れずにこちらの駒を取ってくる敵駒は赤駒だと予測できる．そのような予測の積み重ねから終盤が成り立つため，実際のルールと乖離していても，完全に非公開となっている盤面よりも一部公開問題は現実的・実用的だと考える．

各問題を構成する要素として，ここでは青駒を脱出させることで勝利する『青駒脱出』の要素と，青駒を全て取る『青駒全取り』の要素を紹介する．なお，一般問題同様に『赤駒壁利用』によって構成される問題は存在するが,『青駒脱出』と共に構成される問題は一般問題のものと大きな違いが見られず省略する.『青駒全取り』と共に構成される問題は 4.5.2 節で紹介する．

4.5.1　青駒脱出

勝利条件 1（青駒の脱出）を利用する要素である．以下にこの要素によって構成された問題例（図 4.10）と解き方を示す．

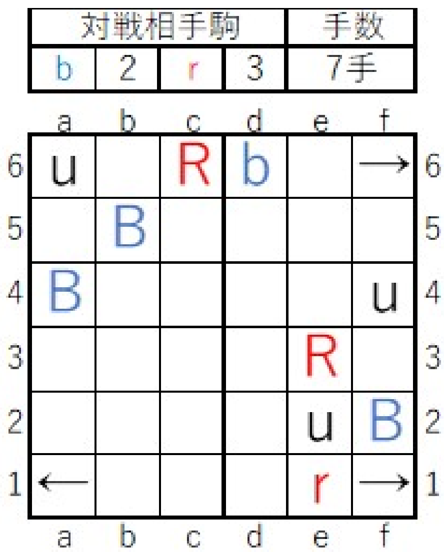


図 4.10: 『青駒脱出』を用いた問題例 3

図 4.10 の問題では，d6 に後手側の青駒，e1 に赤駒が公開されおり，a4 か b5 の青駒を a6 の脱出口に向かわせればよいと予想できる．一般問題であれば，e1 の赤駒の色がわからず，f1 からの脱出を防ぐ必要があったが，本問題では色が公開されているためその必要がない．こちらの脱出口に近い青駒は 2 つあるため，片方

を犠牲にするつもりで動かすことで後手側の e2 が脱出する前にこちらが脱出できる．このように実戦でも後手側の駒が予想できていれば，無駄な行動をとらずに動けることもある．

4.5.2 青駒全取り

勝利条件 2（青駒を全て取る）を利用する要素である．一般問題では勝利条件 1 と勝利条件 3 を利用する問題しか生成できなかった．しかし，一部公開問題においてはもう 1 つの勝利条件である勝利条件 2 を利用する要素を含んだ問題が生成可能となる．

後手側の駒色組合せについては，常に最悪な状況を考えなくてはならないため，取った後手側の駒は赤駒だとして数えるべきである．敵赤駒を全て取ってしまうと対戦相手に勝利条件 3 を満たされてしまい負けとなるため，一般問題においては後手側の青駒を取ることはできず，勝利条件 2 を満たすことはできない．一方，一部公開問題において後手側の赤駒が最低 1 つでも公開されていると，そのうち 1 つを避けて全駒を取ることで赤駒を 1 つ残しつつ青駒を全て取ることができる．つまり勝利条件 2 を満たす問題を生成することが可能である．

以下にこの要素によって構成された問題例（図 4.11）と解き方を示す．

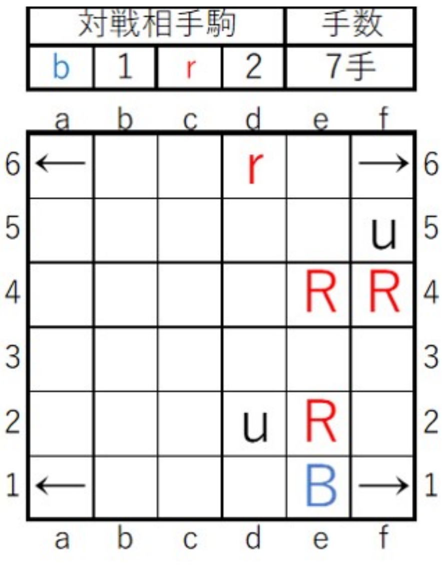


図 4.11: 『青駒全取り』を用いた問題例

図 4.11 では，d6 に後手側の赤駒が公開されている．勝利条件 2 を満たすためには d2，f5 の色不明駒を両方取ればよい．そのどちらを初手で取るかでは，逃げられた（あるいは反撃された）ときにより厄介なほうを先に取るべきである．f5 は上方向に逃げられてもすぐに追い詰めることができそうだが，d2 は左方向に逃げられたら取ることが難しそうである．よって e2 の赤駒を d2 に動かし取る．後手側は f5 の駒を取られるわけにはいかないため，f6 に逃がすことになる．先手側は e4 から e5 に動かす．すると後手側は追い詰められた f6 を動かすわけにはいかないので d6 の駒を動かすことになる．その後先手側は f4 の赤駒を f5 に動かせば，後手側が次にどう動かそうと最後の色不明駒を取ることができる．よって先手側の勝ちである．

4.5.3 考察

後手側の駒を一部公開した一部公開問題を提案した．4.5.1 の『青駒脱出』のみを用いた問題は，若干の違いがあるとはいえ一般問題のものと比べて多様性としては弱い．一方で 4.5.2 の『青駒全取り』要素によって構成された問題は一般問題ではありえなかった問題である．実際に生成を行ってみたところ，赤駒を壁として利用する問題が多く，これは『赤駒壁利用』の要素が後手側の行動を制限し，追い詰めるというこの問題の攻略法と相性がよいためであると考えられる．以上により，この問題を解くことにより，駒の追い詰め方等を学ぶことができると考えられる．

4.6 問題生成アプローチ

詰めガイスター問題の生成アプローチは様々なものが考えられる．以下では本研究で主に用いたアプローチ 2 種について述べる．他にも有用だと思われるアプローチがあるが，それらについては 8 章で述べる．

一つ目は初期盤面をランダムに生成するアプローチである．駒数などの条件を与えて駒配置を行うだけなので，盤面の生成自体に資源は殆ど使わない．しかし盤面を詰め問題とするには必ず必勝手が存在し，かつ最短手数がわからなければならない．そのため必勝手探索を行い詰め問題になっているかと最短手数の確認を行わなければならない．

二つ目は初期盤面に既存問題を用い，手を進めるのではなく戻すことで新たな問題とするアプローチである．基となる盤面を必要とするという欠点はあるが，狙った手数の問題を生成しやすいという利点が存在する．

本論ではこの 2 つのアプローチによる生成アルゴリズムを用い，それぞれのアルゴリズムがどのような問題をどれほどの時間で生成されるかを検証していく．そのため 5 章および 6 章では，この 2 つのアプローチを用いたアルゴリズムの説明および問題生成実験の方法と結果について述べる．

第5章　ランダム生成法

5.1　アルゴリズム

本手法のアルゴリズムは，盤面をランダムに生成し必勝手が存在するか検査，最後に生成したい問題に応じて絞り込みを行う，といった流れになる．

5.1.1　盤面のランダム生成

入力として，先手側と後手側の各駒数を指定し，配置および色組合せをランダムとした盤面を生成する．さらに生成する問題が一部公開問題だった場合，配置した駒のうちランダムでいくつかの色を公開する．この時点では盤面があっても，それが問題として成立しているかはわからない．さらに問題の必要項目である最短手数がこの時点では不明である．例として，図 5.1 のような盤面が生成されたとする．この盤面はこの時点で必勝手が存在するかも最短手数も不明である．実際，b1 を青駒であると考えた場合，先に脱出され 4 手で負けてしまうため，問題として成立しない．そのため，問題として成立しているかの検査のために，次節で述べる必勝手探索を行う必要がある．

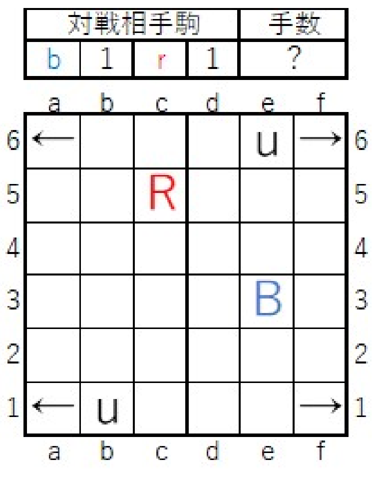


図 5.1: ランダム盤面例

5.1.2 必勝手探索

盤面を詰め問題とするためには必ず先手番の必勝手が存在しなければならないため，探索を行うことで検査する必要がある．さらに，この必勝手探索は必勝手の有無を検査するだけでなく，最短手数を求めることにも用いる．これは問題の目的が『後手駒がどんな駒色の組み合わせ，動きであろうとも最短で勝利できる手順の発見』であり，そのことからプレイヤーに最短手数の提示が必要となるためである．

本手法では探索に紫駒 [9] の概念を導入する．詰めガイスター問題は不完全情報ゲームであるため，考えられる色組合せを全て探索する必要があるように思える．しかし，後手駒の色組合せに依存しない必勝手が存在することを証明するためには，その状況での最悪の駒色のみを考えればよい．つまり取った後手駒は赤色，脱出した後手駒は青色と考える．このように常に最悪な状況に変化する駒を紫駒とし，非公開の後手駒全てを紫駒とすることで完全情報ゲームとして扱うことができる．こうすることで図 5.2 のように，本来 3 通りの駒色組合せがありえるような盤面でも，紫駒の概念を用いることで 1 種類の盤面のみを探索すればよく，探索コストを大幅に削減することができる．

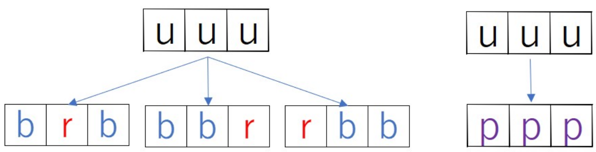


図 5.2: 紫駒の盤面削減例

探索手法は $\alpha\beta$ 法と Df-pn 法の 2 種類を試した．この 2 種の手法はどちらも詰将棋での詰探索に用いられており，初期では $\alpha\beta$ 法が用いられていたが計算量爆発により 27 手以上の問題を解くことが出来ず限界を迎え，その後は Df-pn のような探索木を AND/OR 木と見なす探索法が主に使われるようになり 300 手以上の問題を解くことに成功している [23]．

$\alpha\beta$ 法とは利得を最大，損害を最小になるように手を選ぶゲーム木の探索アルゴリズムの一つであるミニマックス法を改良した手法である．ミニマックス法では，駒の位置や数などから盤面を評価する必要がある．本手法では，詰みであるかを判定することが目的であるため，詰みであれば 9,999 などの膨大な評価値を設定した．さらに手数（深さ）が伸びれば伸びるほど評価値にマイナス修正を入れている．こうすることで，後手番は先手番が勝てない，もしくは詰み手が最長になるように手を選び，先手番は詰み手が最短になるように手を選ぶようになる．

Df-pn 法 (Depth-First Proof-Number Search)[24] とは，長井らにより提案され

た証明数探索を改良した手法である．証明数探索同様 AND/OR 木に証明数・反証数の概念を追加しているが，それらの両方を閾値として用いることが違いとして挙げられる．さらにハッシュテーブルを用いることで探索した盤面の証明数と反証数を保存，再度同じ盤面を訪れた際に保存していた証明数と反証数を参照，再度の探索を省略することで探索効率を向上させている．ハッシュテーブルの有効性を検証すべく，4 つの問題に対しハッシュテーブルの有無での探索ノード数と探索時間を比較する．対象とした問題 (図 5.3) は各駒 1 つずつの 9 手，15 手の問題，各駒 2 つずつの 9 手，15 手の問題である．表 5.1 に結果を示す．結果を見ると，どの問題においてもノード数と探索時間ともに大きく改善されることがわかる．特に手数が長くなればなるほどその改善は大きくなる．

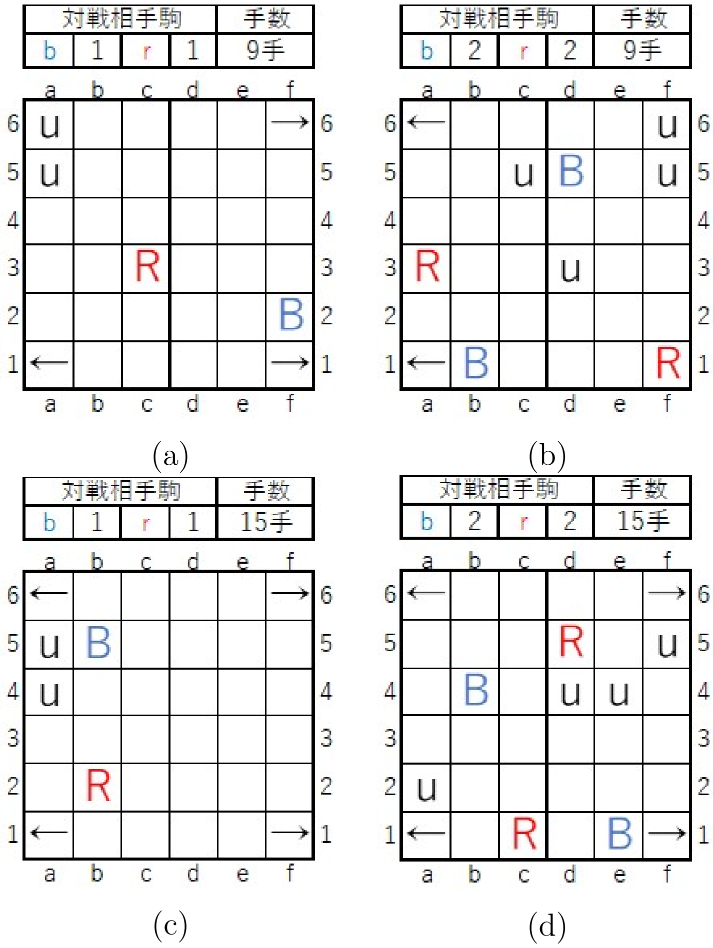


図 5.3: ノード数削減検証に用いた問題

表 5.1: ハッシュテーブルの有無による比較

		探索ノード数	探索時間 [ms]
問題 (a)	有	249	18
	無	1,480	20
問題 (b)	有	2,252	22
	無	13,266	31
問題 (c)	有	8,198	50
	無	355,151	510
問題 (d)	有	127,460	949
	無	5,148,463	27,699

5.1.3 Df-pn の工夫

本アルゴリズムでは，必勝手探索に用いる Df-pn に以下に示す幾つかの工夫を加えた．

最短手数の発見

Df-pn は本来詰み手が存在することの証明はできるが，特性上最短の手順を見つけるのではなく詰み手順を見つけた時点で探索を終了してしまう．そこで本アルゴリズムでは反復深化を用いて，探索深さを 2 手ずつ深くしていき（後手番で先手が詰ませることはできないため 2 手ずつとした），詰みが証明できた時点での深さから最短手数を見つけることとした．例として，5 手では詰み手が見つからず，次の 7 手で見つかった場合，その盤面は 7 手問題となる．

詰めない盤面の探索打ち切り

必勝手探索において計算するノードを減らすことは探索効率の向上に繋がる．本アルゴリズムでは，計算ノードを減らすために，残りの探索深さからして詰むことができない手があれば不詰としてそれ以降の探索を打ち切っている．詰めガイスター問題の一般問題において，自身の脱出口とそれに一番近い青駒とのマンハッタン距離は，脱出に必要な最低限の手数と相関がある．それを残りの探索深さと比較することで，詰むことが確実にできないかを判定する．これにより，図 5.4 の左盤面のような一定手数以内に詰むことができない手を深くまで無駄に探索することを防ぐことができ，探索ノード数を大幅に削減した．この探索打ち切りの有効性を検証すべく，4 つの問題に対し探索打ち切りの有無での探索ノード数と探索時間を比較する．対象とした問題はハッシュテーブルの有効性検証の際に用いたものと同じ問題 (図 5.3) である．表 5.2 に結果を示す．

結果を見ると，どの問題においてもノード数と探索時間ともに大きく改善されることがわかる．特に大きく改善されたのが (a)(b) の問題で，これらの問題は脱出口まで一番近い青駒と脱出口までの距離が，問題手数と比べて比較的遠いことが挙げられる．この工夫は脱出口までの距離によって打ち切るかどうかを判断しているため，距離を縮める猶予が多い問題ほど恩恵が大きいことがわかる．

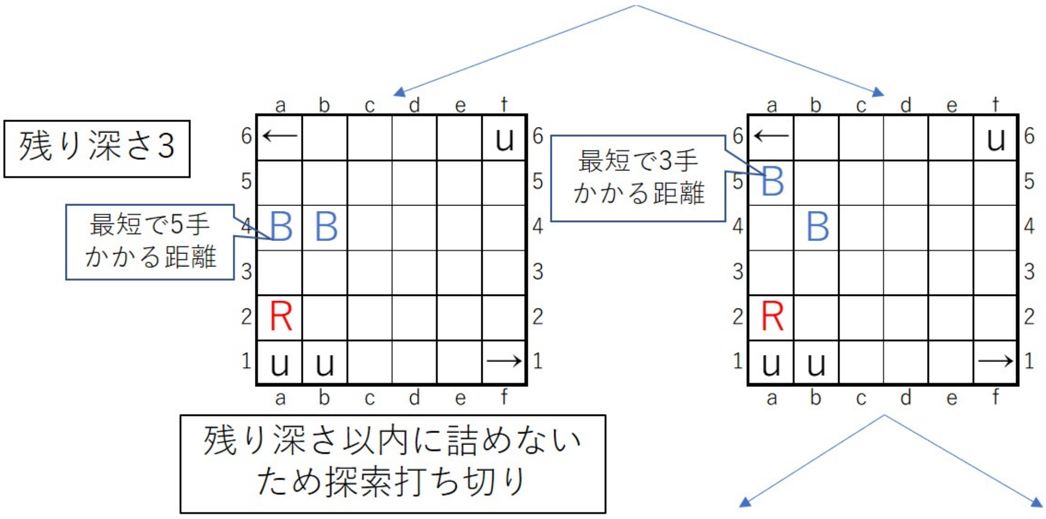


図 5.4: 探索打ち切り例

表 5.2: 探索打ち切りの有無による比較

		探索ノード数	探索時間 [ms]
問題 (a)	有	249	18
	無	7,090	30
問題 (b)	有	2,252	22
	無	71,107	339
問題 (c)	有	8,198	50
	無	14,965	80
問題 (d)	有	127,460	949
	無	814,862	8,264

5.1.4 問題の絞り込み

5.1.1〜5.1.3 により必勝手が存在する盤面を生成したが，その殆どが 3 手や 5 手などの短手数の問題や，脱出口に一番近い青駒が一直線に向かうだけの問題であった．そこで簡単・単純すぎる問題などを意図的に外すべく，あるいは逆に狙った条件の問題のみにするべく，盤面に様々な条件をかけ絞り込む．本研究では例として以下のような条件を用いることで，問題の質をある程度保証，狙った要素を持つ問題生成を可能とした．

問題の手数に下限を設ける

最短手数が一定数に満たない問題を切り捨てる．

直行防止

脱出口に一番近い先手側の青駒が真っ直ぐ向かうだけの問題を切り捨てる．その青駒と脱出口間のマンハッタン距離を測定し，最短手数と比較することで識別する．

赤駒壁利用

4.4.2 で示した赤駒壁利用の要素を用いた問題のみに絞る条件．通常通り必勝手を探索した後，先手側の赤駒を全て青駒に変化させ再探索させる．そうして最短手数が増えるか必勝でなくなれば赤駒の特性を利用している問題として，この条件を満たしているとする．

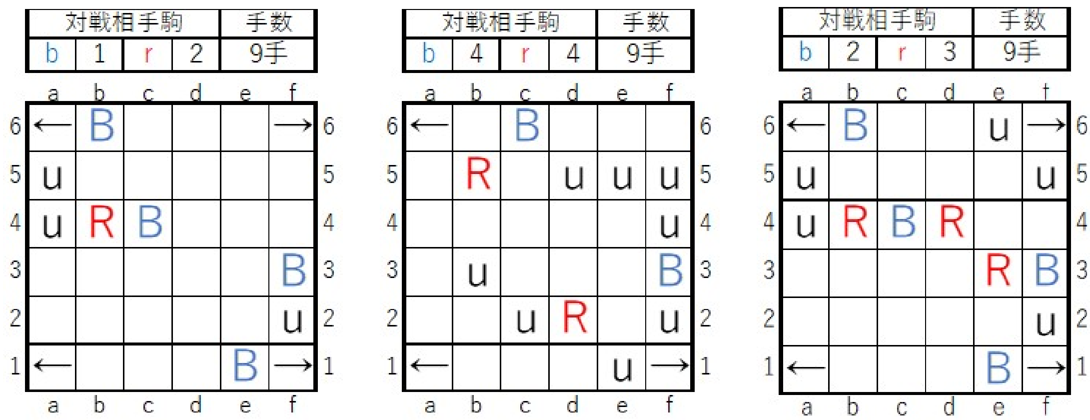
青駒全取り

4.5.2 で示した青駒全取りの要素を用いた問題のみに絞る条件．必勝手が見つかった際，その必勝手が全て勝利条件 2 を満たすものであった場合，この条件を満たしているとする．

5.2 生成実験

5.2.1 探索法による比較

4.1 節では必勝手探索の手法として，$\alpha\beta$ 法と Df-pn 法を挙げた．ここではその 2 手法のどちらの探索速度が優れているかを検証する．検証は 3 種類の盤面を用意し，それぞれの手法により盤面が詰みだと証明されるまでの時間を比較する．盤面はそれぞれ駒数の異なる 9 手問題 (図 5.5(a)〜(c)) である．



(a) (b) (c)

図 5.5: 探索速度比較に用いた問題

結果を表 5.3 に示す．Df-pn 法はどの問題に対しても大きく探索時間を改善していることがわかる．Df-pn はハッシュの読み書き時間がかかってしまうが，計算ノード数が大幅に削減されたことで大きく差ができた．他のさまざまな問題に対しても検証したところ，よほど短手数の問題でない限り Df-pn 法が優れた結果を出したことを確認した．よって，これ以降は Df-pn 法を探索法として用いていく．

表 5.3: 探索法による探索時間の比較

探索手法	探索時間 [ms]		
	問題 1	問題 2	問題 3
$\alpha\beta$	3452	127098	333467
Df-pn	90	503	467

5.2.2　様々な駒数における生成率，生成時間比較

必勝手探索手法として Df-pn 法を採用した本生成法を用いて詰め問題を生成し，生成速度および各手数における生成数に駒数がどのような影響を及ぼすかを検証した．問題の設定としてカテゴリを一般問題，駒数を 7 パターン，数としてカウントする問題手数を 9 手から 19 手（絞り込み条件により範囲外の問題を切り捨て）とした．駒数のパターンと生成数は以下のようになっている．各駒数パターンにおける生成数は，以下に示すように統一されていない．一部のパターンは既発表論文用に長時間の実験を行ったが，パターン 1，パターン 4 などは生成される問題の傾向が分かったところで実験を止めている．一方でパターン 5 以降は本論文用に実験を追加したものだが，生成に時間がかかることもあってやや少ない問題数にとどまっている．

1. 各駒 1 つずつ（3000 問）
2. 先手駒 1 つずつ，後手駒 2 つずつ （5816 問）
3. 先手駒 2 つずつ，後手駒 1 つずつ（10000 問）
4. 各駒 2 つずつ（2500 問）
5. 先手駒 2 つずつ，後手駒 3 つずつ（1597 問）
6. 先手駒 3 つずつ，後手駒 2 つずつ（1676 問）
7. 各駒 3 つずつ（572 問）

検証項目は各駒数パターンにおける各手数の生成比率と 1 問あたりの生成時間である．表 5.4 と表 5.5 に結果を示す．なお以降の文中，各駒 1 つずつなどの駒数パターンの場合 (1,1,1,1) などと表現する．

表 5.4: ランダム生成による各手数の生成比率

駒数	各手数の生成比率 [%]					
(先青, 赤, 後青, 赤)	9	11	13	15	17	19
1,1,1,1	73.84	16.13	6.53	3.00	0.50	0
1,1,2,2	63.21	25.50	9.30	1.72	0.28	0
2,2,1,1	88.32	8.31	2.49	0.59	0.24	0.05
2,2,2,2	67.13	24.79	5.84	1.76	0.40	0.08
2,2,3,3	56.86	32.81	6.76	2.63	0.63	0.31
3,3,2,2	70.94	23.75	4.41	0.72	0.18	0
3,3,3,3	54.55	33.04	8.22	2.80	1.40	0

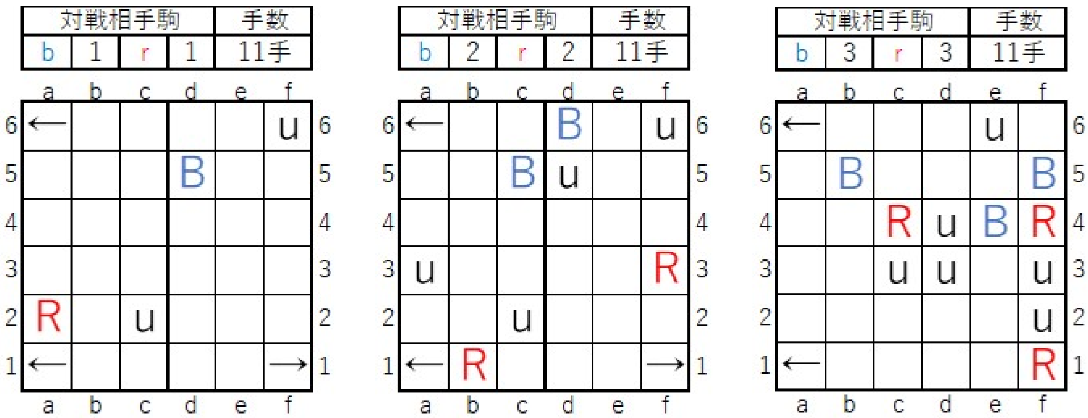
表 5.5: ランダム生成による 1 問あたりの平均生成時間

駒数	各手数の 1 問あたりの平均生成時間 [s/問]						1 問あたりの
(先青, 赤, 後青, 赤)	9	11	13	15	17	19	平均生成時間 [s/問]
1,1,1,1	3.65	16.7	41.3	90.0	539	—	2.7
1,1,2,2	21.7	53.7	147	796	4,977	—	13.7
2,2,1,1	19.1	203	677	2,856	7,020	33,698	16.9
2,2,2,2	92.5	251	1,064	3,531	15,535	77,674	62.1
2,2,3,3	263	456	2,213	5,691	23,901	47,802	150
3,3,2,2	989	2,955	15,895	98,021	392,085	—	702
3,3,3,3	2,382	3,932	15,813	46,451	92,902	—	1,299

各手数における生成比率を見ると，どの駒数においても手数が長くなるほど比率が激減していることがわかる．これはガイスターのような小さな盤面で長手数の問題となると，相当に複雑な盤面が要求されるためだと思われる．特徴的な点として，駒数が (2,2,1,1) と (3,3,2,2) のパターンが他パターンと比較して 9 手問題が多く生成されていることが挙げられる．これは先手駒の数が後手駒の数より多いほど容易に勝てる場合が多いためだと考えられる．

次に生成時間を見ると，駒数が増えるほど 1 問あたりの生成時間は増大していることがわかる．これは駒数と探索ノード数には相関関係があることから当然の結果だと言える．(3,3,2,2) が (2,2,3,3) よりも多くの時間を使っているのは，先述したように容易に勝てる場合が多いことから今回は問題として数えていない 7 手以下の盤面が多く見つかったためだと考えられる．(2,2,1,1) が (1,1,2,2) よりも多くの時間を使っていることも同じ理由であろう．

問題の傾向も駒数によって違いが見られた．ここで (1,1,1,1)(2,2,2,2)(3,3,3,3) の 3 パターンの問題（図 5.6) を 1 問ずつ見ていく．



[a] (1,1,1,1)　[b] (2,2,2,2)　[c] (3,3,3,3)

図 5.6: 生成された問題

まずは [a] の (1,1,1,1) の問題を見る．この問題は d5 の自青駒を a6 から脱出させる目的がわかりやすい問題である．だがそれだけではなく，盤面左下において最後の自赤駒で脱出口を塞ぐといった攻防も行われる．ここで注目すべきは敵赤駒の数で，本問題では 1 つとなっている．つまり，どんな駒色組合せでも勝てる手順を探さなくてはならない以上，1 つも敵駒を取ってはいけないのである．

次に [b] の (2,2,2,2) の問題を見る．この問題は c5 と d6 の自青駒を上手く使って脱出を狙うことが目的である．駒数が増えたことで一見どちらの自青駒で脱出するのか，どちらの脱出口に向かうのかが分かりづらくなっている．さらに盤面左下で脱出を防ぐための攻防が行われることが予想できる．さらに敵赤駒の数が 2 つであるため，敵駒を 1 つまで取っていい問題となっている．

最後に [c] の (3,3,3,3) の問題を見る．この問題も [a][b] 同様に敵駒の脱出を防ぎつつ，青駒の脱出を狙う問題となっているが，敵赤駒の数が [a][b] より多い 3 つである．つまり敵赤駒を 2 つまで取っていいことになる．実際にこの問題は盤面右下において敵駒を 2 つ取る動きが必要となるのだが，1 手目は f1 自赤駒で f2 の敵駒をすぐに取るのではなく，一旦 e1 に避けることにより待ち伏せする動きが正解となる．1 つの自駒で複数の敵駒の脱出を防ぐ場合，こういった独特な動きを取る以外にも [a] のように最後の赤駒を使って蓋をする動きがあるが，そういった問題は単純になりやすい．さらにこの問題も，[b] 同様にどの青駒をどの脱出口に向かわせるか直感的に分かりづらい問題となっている．

これらのことから，駒数は増えれば増えるほど盤面は複雑化していき，様々な勝ち方が現れる可能性があることがわかる．特に敵赤駒の数は敵駒を取ることができる数に直結する．しかし，長手数の駒数が多い問題を生成するには，(3,3,2,2) の 17 手問題を 1 問生成するのに 100 時間以上 (392,085 秒) かかるなど，ランダム性に依存することから非常に時間がかかる．よって，駒数が多くとも狙って長手数の問題を効率よく生成する手段が求められる．

第6章　逆順生成法

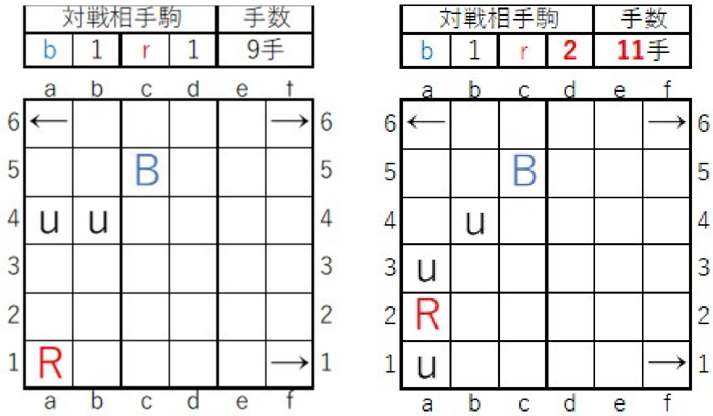
5章では生成アルゴリズムとしてランダム生成法を提案，検証した．ランダム生成法では多様な問題を一から生成することが可能であったが，ランダム性に大きく依存することから長手数の問題を狙って生成することが困難であった．そこで，既存の問題を利用することでランダム性を抑え，狙った手数の問題を生成できるのではないかと考えた．本章では既存の問題から手を戻すことで新たな問題を生成する逆順生成法を提案し，問題生成を行うことで有効性を検証する．

6.1　アルゴリズム

本手法のアルゴリズムは，後手番と先手番を交互に戻していき，手と手と間で逐一必勝手探索を行っていく，といった流れになる．

(1) 問題を用意

本手法では予め問題として成立している盤面を用意する必要がある．その盤面を元として逆順に手を戻していく．これ以降，図6.1に示す，ある9手詰の元盤面から新たな11手詰の盤面の生成を例として逆順生成法の流れを説明する．左図がランダム生成により得た9手詰問題であり，逆順生成した11手詰問題である右図は一部の駒の配置と後手赤駒の数が変化していることがわかる．これらの変化がどのような処理によって起こったことかを以下に示していく．



[元盤面] [逆順生成盤面]

図 6.1: 9 手詰問題から 11 手詰問題を逆順生成する例

(2) 後手駒を 1 つ戻す

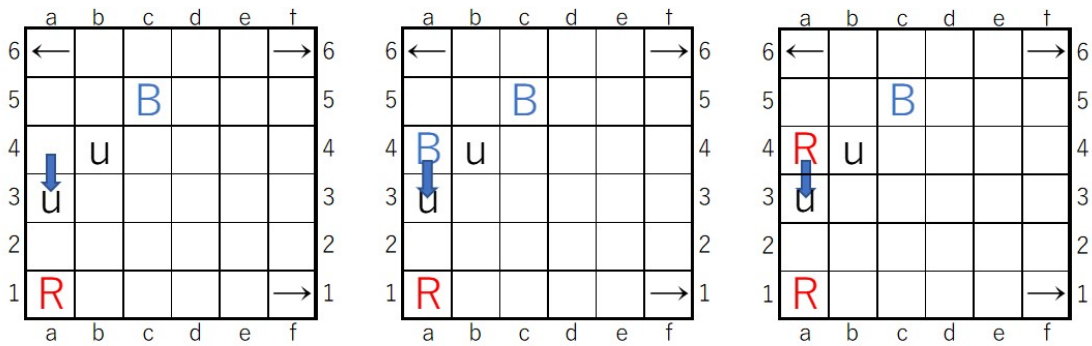
後手番の合法手を全て列挙し，ランダムに 1 手戻す．この際，通常の移動とは異なり移動先に何らかの駒があってはならない．移動後の盤面は一意的ではなく，

a 移動前のマスに先手側の駒がなかった場合

b 先手側の青駒があった場合

c 先手側の赤駒があった場合

の 3 パターンが考えられる．これは正順で見た場合，先手側の駒が取られたかを考えている．図 6.2 では 3 パターンそれぞれでの盤面を示している．



[a] [b] [c]

図 6.2: 後手番の戻し手パターン例

(3) 得られた盤面の最短手数を求める

(2) で得られた盤面の必勝手探索を行い，戻した手以外で先手番が元々の詰手数で詰めなくなるような手がないかを調べる．もしあれば (2) に戻り別の手を探す．必勝手探索は 5 章で述べた Df-pn 法を用いる．

図 6.3 では (2) で [a] のパターンを選んだ際の例を示している．[a] の盤面からは元盤面のみに派生するわけではなく，他の様々な盤面に派生する可能性がある．そのため 9 手詰みの元盤面から 1 手戻したからといって確実に後手番から数えて 10 手詰みになるとは限らない．そのための検査である．

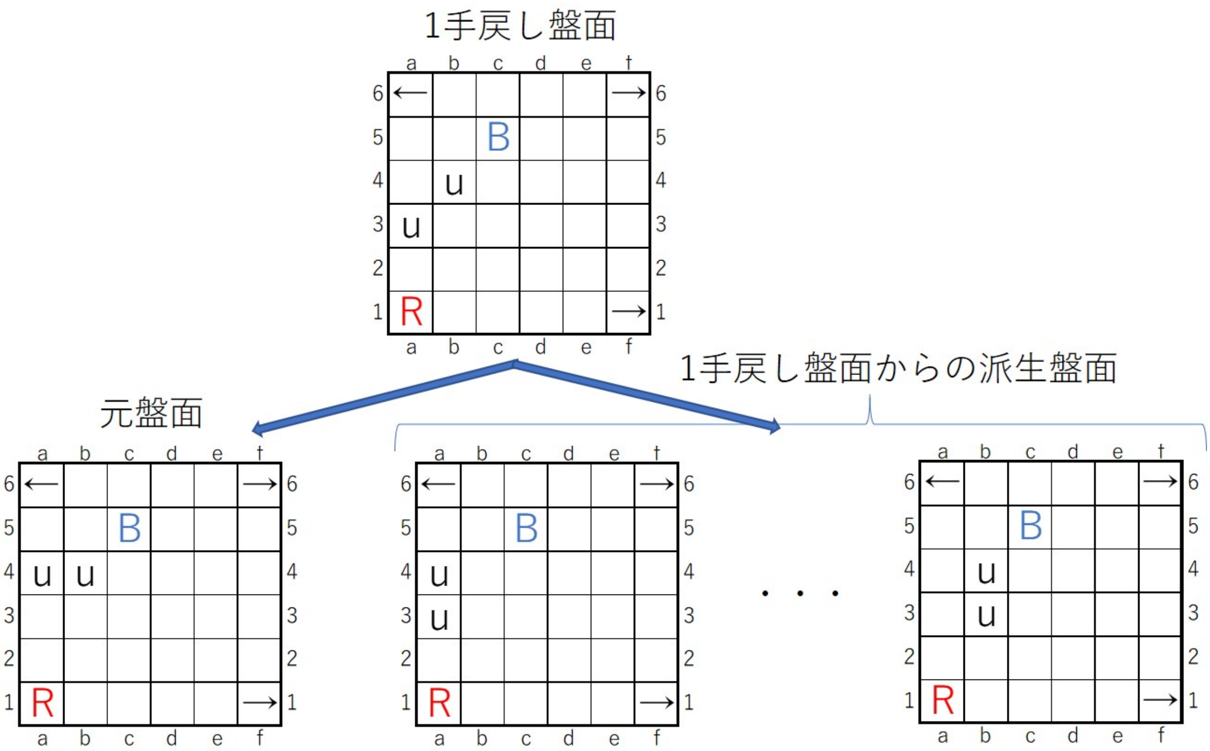


図 6.3: 派生盤面の例

(4) 先手駒を 1 つ戻す

先手番の合法手を全て列挙し，ランダムに 1 手戻す．こちらは先ほどの 3 パターンの手番を入れ替えたものに，非公開駒を加えた 4 パターンを考える必要がある．

a 移動前のマスに後手側の駒がなかった場合

b 後手側の青駒があった場合

c 後手側の赤駒があった場合

d 後手側の非公開駒があった場合

[d] の場合，紫駒の概念を導入しているため非公開駒は赤駒であると考える．なお一般問題の場合，[b][c] のような公開駒は考えないため，図 6.4 のように [a] と [d] の 2 パターンのみでよい．

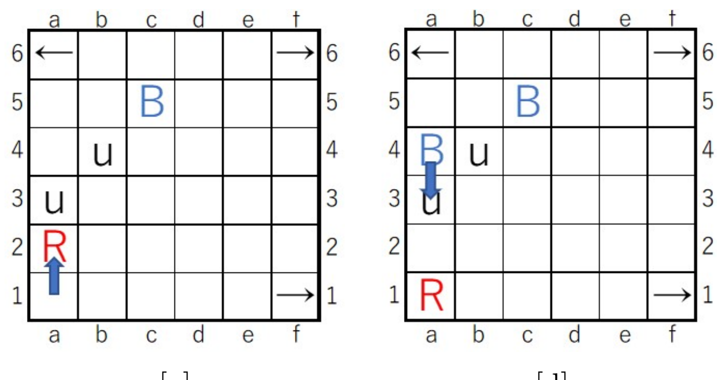


図 6.4: 先手番の戻し手パターン例

(5) 得られた盤面の最短手数を求める

(4) で得られた盤面の必勝手探索を行い，最短手数を求める．この際，元盤面での必勝手数+2 手の必勝手を見つける必要はなく，+1 手までの必勝手を見つけることができなければ，元盤面での必勝手数+2 手であることがわかる．なぜなら (4) の時点で，先手側は本来の詰手数+1 手で詰めることが確定しているため，そこから戻した手を辿れば基本的に+2 手で詰むことができるからである．しかしもし+1 手までに必勝手が見つかれば，それは手数を伸ばすための戻し手が逆に手数を縮めたことになるため (4) に戻り別の手を探す必要がある．

例のように 9 手問題から 11 手問題を生成する際に，11 手問題であることを証明するためには 10 手以内に詰めないことがわかればいい．これによって得られた盤面を新たな問題とする．

6.2 生成実験

本生成法を用いて詰め問題を生成し，生成速度および各手数における生成数を検証した．用いた元盤面は 5.2.2 節にて生成した問題を用いた．生成する問題は元の手数+2 手とし，1 つの元盤面につき最高 1 問の逆順問題しか生成していない．検証項目は，各駒数パターンにおける+2 手問題が生成された元盤面の比率（逆順生成成功率）と，1 問あたりの逆順生成に用いた時間である．本検証では前章同様に 7 つの駒数パターンについて行っているが，逆順生成においては (1,1,1,1) から (2,1,2,1) のように駒数が変化する問題が生成される場合がある．その際は (1,1,2,2) としてではなく (1,1,1,1) としてカウントしている．まず表 6.1 に，前者の結果を示す．

表 6.1: 逆順生成による+2 手問題が生成された元盤面の比率（逆順生成成功率）

駒数 (先青, 赤, 後青, 赤)	+2 手問題が生成された元盤面数 / 元盤面数 [%]					
	9 → 11	11 → 13	13 → 15	15 → 17	17 → 19	19 → 21
1,1,1,1	37.09	57.44	68.37	46.67	20.00	—
1,1,2,2	55.50	53.41	42.14	28.00	12.50	—
2,2,1,1	28.57	42.36	67.07	52.54	50.00	100
2,2,2,2	62.48	48.87	52.05	43.18	40.00	50.00
2,2,3,3	63.33	47.71	64.81	52.38	50.00	80.00
3,3,2,2	65.77	60.30	58.11	75.00	33.33	—
3,3,3,3	70.51	57.14	51.06	43.75	25.00	—

+2 手問題の生成比率を見ると，どの駒数・元盤面の手数においても生成比率に有意な差は見られない．そして殆どの場合で元盤面から 50%以上の確率で生成できている．

次に，表 6.2 に 1 問当たりの逆順生成に用いた時間を示す．

表 6.2: 逆順生成のみでの 1 問あたりの平均生成時間
元盤面の生成コストを含んでいない

駒数	各手数の 1 問あたりの平均逆順生成時間 [s/問]					
(先青, 赤, 後青, 赤)	9 → 11	11 → 13	13 → 15	15 → 17	17 → 19	19 → 21
1,1,1,1	0.08	0.15	0.27	1.1	8.3	—
1,1,2,2	0.30	1.4	6.0	23.0	76.0	—
2,2,1,1	0.29	1.2	3.1	3.9	12.2	23.2
2,2,2,2	1.0	7.2	29.2	98.9	261.0	452.1
2,2,3,3	1.9	12.9	37.5	190.8	348.7	936.1
3,3,2,2	2.1	14.8	113.0	243.2	1,474.0	—
3,3,3,3	3.7	33.7	207.2	914.1	7,026.6	—

結果を見ると，駒数が増えれば増えるほど 1 問あたりの生成時間は増大していることがわかる．これはランダム生成時と同様に，駒数と探索ノード数には相関関係があることから当然の結果だと言える．しかし，合計が同じ駒数でも (1,1,2,2) と (2,2,1,1) や，(2,2,3,3) と (3,3,2,2) には有意な差が表れている．これは，逆順生成アルゴリズムの仕様上，先に戻す後手番の駒数が多ければ多いほど計算量が膨大になるためだと考えられる．考えられる盤面全てを探索するのであれば，先手駒が多くとも後手駒が多くとも差は生まれないが，本アルゴリズムでは先に戻す後手番次第ではそこから派生する先手番の探索は行わない．そのため後手駒の駒数（行動数）から有意な差が現れたのだと考えられる．なお，この生成時間は元盤面が用意されていることが前提のもので，元盤面の生成コストは含まれていない．逆順生成法はこの手法単体では問題を生成できないことから，ランダム生成法などの一から問題を作る生成法と同時に運用するのが望ましい．

ランダム生成法と逆順生成法の同時運用による性能を評価する．まず，同時に運用した場合，どのように問題数が変化するかを述べる．図 6.5 にランダム生成の後に逆順生成を行った際の生成数の変化例を示す．例は駒数 (3,3,3,3) を 100 問生成した場合のものである．まずランダム生成を行った際に表 5.4 の生成率に従って生成される．それを元に逆順生成法を用いて手数の小さい問題から＋ 2 手の問題を生成していく．まずは 54.6 問の 9 手問題から表 6.1 の生成成功率に従って 38.5 問の新たな 11 手問題が生成される [a]．これによりランダム生成法により生成された 33.0 問と合わせて 71.5 問の 11 手問題が生成されたことになる [b]．同様に，11 手問題問題 71.5 問から 13 手問題が逆順生成により 40.9 問を生成され [c]，ランダム生成での 8.2 問と合わせて 49.1 問となる [d]．この手順を繰り返していくことになる．

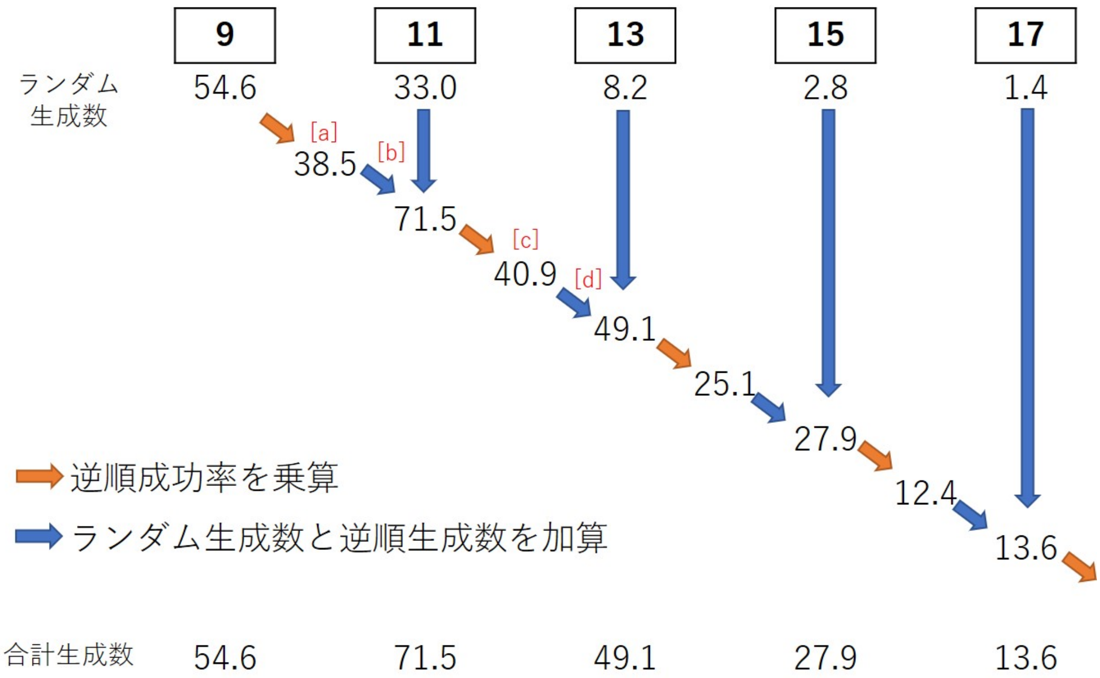


図 6.5: ランダム生成後の逆順生成による問題数の変化

次に生成時間を考える．この例の場合，まず駒数 (3,3,3,3) を 100 問最初に生成した時間は表 5.5 に従い．1,299 × 100 ＝ 129,900[s] となる．9 手から 11 手を生成する逆順生成の処理は，表 6.2 に従い，3.7 × 38.5 ＝ 143[s] となる．つまり，11 手問題を 71.5 問得るために 129,900 ＋ 143 ＝ 130,043[s] を要したことになるため，11 手問題を 1 問生成するための平均生成時間は 130,043/71.5 ＝ 1,818[s] となる．同様にこの手順を繰り返すことで全ての手数における平均生成時間を求めることができる．

表 6.3 に，ランダム生成後に逆順生成を行った際の 1 問当たりの生成時間を示す．さらに，本手法での生成速度がランダム生成のみのときと比較してどれだけ速いかを示す．

表 6.3: ランダム＋逆順生成による 1 問当たりの平均生成時間
元盤面の生成コストを含んでいる

駒数 (先青, 赤, 後青, 赤)	各手数の 1 問あたりの平均生成時間 [s/問]（ランダム生成より何倍速いか）					
	9 → 11	11 → 13	13 → 15	15 → 17	17 → 19	19 → 21
1,1,1,1	6.3	8.7	11.5	24.6	131.5	—
	(2.7 倍)	(4.7 倍)	(7.8 倍)	(21.9 倍)	—	—
1,1,2,2	22.8	34.2	79.3	291.3	2,406.8	—
	(2.4 倍)	(4.3 倍)	(10.0 倍)	(17.1 倍)	—	—
2,2,1,1	50.5	102.4	148.0	274.8	553.3	570.4
	(4.0 倍)	(6.6 倍)	(19.3 倍)	(25.5 倍)	(60.9 倍)	—
2,2,2,2	93.7	168.7	324.8	816.3	2,255.8	4,868.7
	(2.7 倍)	(6.3 倍)	(10.9 倍)	(19.0 倍)	(34.4 倍)	—
2,2,3,3	218.5	390.4	580.4	1,246.1	2,730.2	4,303.7
	(2.1 倍)	(5.7 倍)	(9.8 倍)	(19.2 倍)	(17.5 倍)	—
3,3,2,2	998.2	1,512.8	2,646.9	3,740.5	12,695.5	—
	(3.0 倍)	(10.5 倍)	(37.0 倍)	(105 倍)	—	—
3,3,3,3	1,819.1	2,678.6	4,905.2	10,877.7	50,537.3	—
	(2.2 倍)	(5.9 倍)	(9.5 倍)	(8.5 倍)	—	—

生成時間からはランダム生成法との大きな差が見られる．ランダム生成法と比較すると，どの駒数・手数においても 1 問あたりの生成時間は大きく短くなっている．逆順生成はアルゴリズムの仕様上，1 つの問題を生成するための探索数はランダム生成法よりも多い．しかし，ランダム生成法により生成された多くの盤面は問題として成立せず，さらには望んだ手数の問題が生成されるとは限らない．一方で逆順生成法では望んだ手数の問題に合わせた元盤面さえ用意できれば，高確率で問題として成立する盤面を生成できる．特に長手数の問題に関しては，ランダム生成ではごく少数しか生成されなかった 19 手問題を逆順生成法では 17 手問題を元とすることで多数生成することができ，さらには 19 手問題から 21 手問題までを生成することができた．これらのことから，逆順生成法は狙った手数の問題を効率よく生成できることができると考えられる．さらに，+2 手問題だけでなく，+4 手，+6 手と徐々に戻す手数を増やしていくことで，更なる効率化が期待できる．

第7章　面白さ・難しさ推定

生成した問題には様々な質の物が存在する．例として，手数は長いがほぼ脱出口に直行するだけの簡単な問題，駒数が少なく一見簡潔に見えるが非常に複雑な動きを要求する問題，などである．本研究の将来展望は，詰め問題コンテンツによってゲームの普及およびプレイヤーの実力向上を目指すことである．そのためには，面白い問題は勿論のこと，プレイヤーに適した特定難易度の問題の提供が求められる．そしてそれらの問題の生成のためには，面白い問題や難易度に影響を及ぼす特徴が何であるかを知る必要がある．そこで，生成した問題に対して面白さと難しさに焦点を当てた推定を行った．まずは被験者実験を行い，被験者に問題の面白さと難しさを評価させる．そして得られたデータを基に，問題の特徴量から面白さと難しさを推定するモデルを生成した．

7.1　被験者実験

プレイヤーが詰めガイスター問題を解く際に，どのような特徴を持った問題に面白さや難しさを感じるのか分析するべく，ガイスターを殆どプレイしたことがない 8 人を対象として被験者実験を行った．実験の手順は以下のようになる．

1. 実験概要，目的，ルールの説明
2. 予め用意した詰めガイスター問題を 1 問提示
3. 制限時間を過ぎるか，解けた時点で終了
4. 解答例を表示
5. 問題が面白かったか難しかったか，解けたかどうかをアンケート評価
6. 手順 2～5 を合計 100 問分行う

被験者には図 7.1 のようなツールを使用してもらい，問題の解答および評価を行わせる．問題に関わる表示項目は通常の詰めガイスター問題同様，最短手数と後手側の駒数と盤面である．一つ一つの問題に時間をかけすぎることを防ぐために，解答には 90 秒の時間制限を設けた．評価項目である面白さと難しさは 5 段階評価 (-2,-1,0,1,2) とした．さらに，任意ではあるがコメントの入力フォームを設け，面白さや難しさを感じた根拠，感想の記入を求めた．

なお，問題解答中に駒操作を行えるようなユーザーインタフェースは実装していない．詰めガイスター問題は実プレイでの必勝盤面の見逃しを防ぐことが目的の一つであり，実プレイでと同様に脳内で考えてもらうことが望ましいとしたからである．このこともあって,『プレイヤーがその問題を解けたかどうか』という情報は，表示された解答例を見た上での自己申告として集めた．

図 7.1: 被験者実験に用いたツール

提示問題は合計 100 問とし，問題の種類と構成する要素を基に,『青駒脱出』の要素を持つ一般問題,『青駒脱出＋赤駒壁利用』の要素を持つ一般問題,『青駒脱出』の要素を持つ一部公開問題,『青駒全取り』の要素を持つ一部公開問題，の 4 カテゴリに配分を行った．手数と駒数については長すぎる（多すぎる），短すぎる（少なすぎる）ような極端な問題を少なめとし，中程度の問題に多く割り振った．問題の配分について付録 A を参照されたい．提示順番は，全被験者で統一すると慣れや飽きが評価に影響してくると考え，一意的ではなく被験者ごとにランダムとした．

7.2 問題の特徴量

各詰めガイスター問題にはカテゴリや駒数や手数は勿論のこと，様々な特徴が存在する．そういった特徴が，人が問題に対して感じる面白さや難しさに関係すると仮定し，推定・評価に利用する．そこで問題の面白さや難しさに関係しうる特徴量を挙げ，各問題についてそれらを算出した．以下に本研究で用いた特徴量を示す．

各駒の数
先手側の青駒と赤駒数，後手側の青駒と赤駒数．

問題の最短手数
最短手順における手数．

先手側の脱出口と一番近い先手青駒との距離
先手脱出口と全ての先手側青駒とのマンハッタン距離を算出し，その中でも一番短い距離．

後手側の脱出口と一番近い後手青駒と紫駒との距離
後手脱出口と全ての後手側青駒（一部公開問題の場合）と紫駒とのマンハッタン距離を算出し，その中でも一番短い距離．

先手側の脱出口と一番近い後手駒との距離
先手脱出口と全ての後手駒とのマンハッタン距離を算出し，その中でも一番短い距離．

脱出口へ向かう以外の行動手数
問題の最短手数から，先手側が脱出口に移動する手数を除いた手数．これが多いほど，脱出以外の行動，例として妨害や脱出の援護などに手数を使うことになる．

後手陣側に入っている先手駒の割合，先手陣側に入っている後手駒の割合
盤面を上下に分けた際の互いのプレイヤー側半分をそのプレイヤーの陣としたとき，その陣に入っている敵駒の割合．これが高いほど，駒は敵陣に多く存在し攻め込んでいる（攻め込まれている）印象を受けると考えられる．

問題のカテゴリ
問題の種類（一般問題か一部公開問題）と問題を構成する要素を基に,『青駒脱出』の要素を持つ一般問題,『青駒脱出＋赤駒壁利用』の要素を持つ一般問題,『青駒脱出』の要素を持つ一部公開問題,『青駒全取り』の要素を持つ一部公開問題，の 4 カテゴリに分類したもの．

探索中のルートノードが得た最大の証明数，最大の反証数

必勝手探索を Df-pn を用いて行った際に，探索中に得られた最大の証明数と反証数．石飛らは詰将棋の研究においてそれぞれの値が，先手側が詰むために調べる必要のある局面数と後手側が詰みから逃れるために調べなければならない局面数としており，プレイヤーが感じる面白さや難しさに関係してくるのではないかと考察している [2]．ルートノードの証明数と反証数の再計算は本来，ルートノードに戻ってきた際にしか行われないが，本研究では石飛らと同様に末端ノードが展開される度にルートノードまでの証明数・反証数を再計算することで途中経過を得ている．なお，反証数においては詰みが確定した時点で∞となってしまうため，∞になる直前の値を使用している．

詰みまでに減る駒数の最小値

盤面が詰みとなるまでに盤面から減った駒数．詰めガイスター問題においては，詰み手順は複数存在しうることから減る駒数は様々ありえるため，本特徴量においては最小値を使用する．この駒数が多いほど，プレイヤーは取る必要がある駒が多いことを表す．

詰みまでに動かす先手駒の最小数

盤面が詰みとなるまでに動かす先手側の駒数．盤面から減った駒数同様に最小値を使用する．この駒数が多いほど様々な駒を動かす必要がある問題ということになり，全駒数からこの数を引いた駒数は役割の有無はともかく動かす必要がない駒の数ということになる．

7.2.1 特徴量と評価値の関係分析

特徴量の各値の面白さ評価値，難しさ評価値の関係として 2 つの例を示す．1 つ目は表 7.1 に示す，各『問題の最短手数』の平均難しさ評価値 (-2.0〜2.0) である．手数が増えると平均難しさ評価値が上昇していることがわかる．19 手問題のみ例外に 17 手問題より下がっているが，これはサンプル数が少なくたまたまそのような問題が選ばれたためだと思われる．

表 7.1: 各最短手数の平均難しさ評価値

最短手数							
5	7	9	11	13	15	17	19
-1.88	-1.44	-0.94	-0.36	0.05	0.32	1.42	1.00

2 つ目は表 7.2 に示す，各『問題のカテゴリ』の平均面白さ評価値 (-2.0〜2.0) である．一般問題と比べ一部公開問題は負の値ではあるものの平均面白さ評価値が高い．問題を構成する要素ごとに見ると，一般問題において,『青駒脱出』の要素を持つ問題に比べ『赤駒壁利用』を加えた問題は平均面白さ評価値が高い．さらに一部公開問題においては,『青駒全取り』の要素を持つ問題の平均面白さ評価値は他よりも高い.『赤駒壁利用』や『青駒全取り』を含む問題は一目見ただけではどの要素を含んでいるかわかりにくい傾向があると考えられ，プレイヤーはそういう点に解き甲斐を感じたのだと予想できる．

表 7.2: 各問題カテゴリの平均面白さ評価値

一般問題全体		-0.18
構成要素別	青駒脱出	-0.33
	青駒脱出+赤駒壁利用	0.04
一部公開問題全体		-0.02
構成要素別	青駒脱出	-0.18
	青駒全取り	0.14

7.3　面白さと難しさの相関

各問題における被験者全員の平均面白さと難しさを用い，横軸を難しさ，縦軸を面白さとしてプロット（図 7.2）することで，面白さと難しさに相関があるのかを調べた．相関係数は 0.63 となり，面白さと難しさには正の相関があることがわかる．この結果から，難しい問題は面白いと評価される場合が多いことがわかる．しかし，ここで図 7.2 の黄枠に囲まれたプロット点に着目する．これらの問題は非常に難しい（-2.0〜2.0 評価で 1.5〜2.0）問題と評価されているが，それほど面白いと評価されているわけではない．Malone はゲームを楽しくしている内発的動機付けとして『挑戦』を挙げており [25]，挑戦心は適切な難易度を持つ問題に抱くと考えられる．難しい問題は面白いが，難しすぎる問題は面白くないと感じられるのはこのためと思われる．特に回帰直線から離れているグラフ内 (a) と (b) を図 7.3 に示し考察を行う．

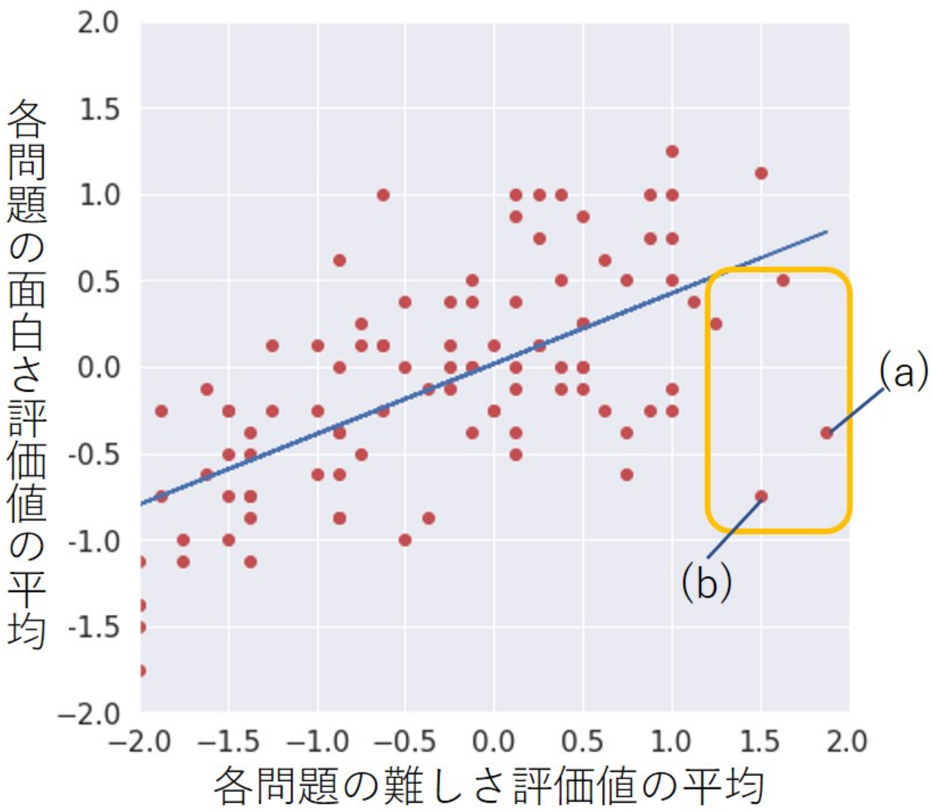
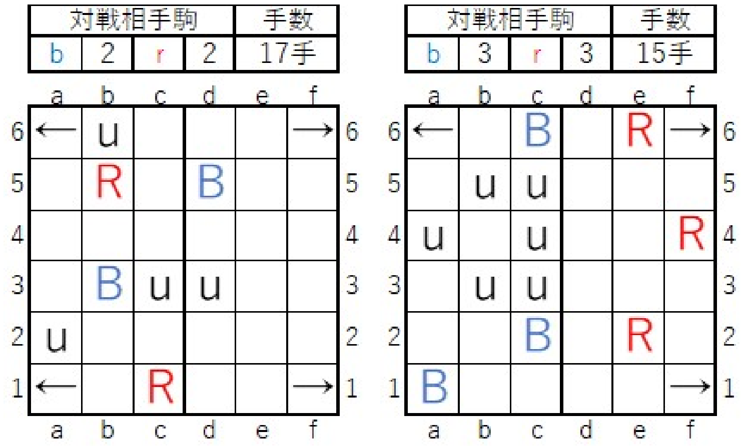


図 7.2: 各問題の平均面白さ（縦軸）と平均難しさ（横軸）評価値のプロット図



(a) 面白さ:-0.4, 難しさ:1.9　(b) 面白さ:-0.6, 難しさ:1.5

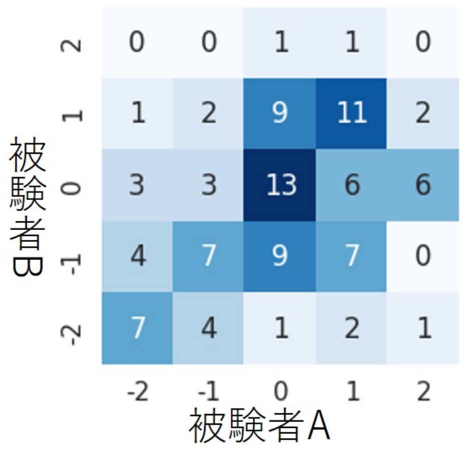
図 7.3: 難しいが面白くない問題

図 7.3(a) は駒数 (2,2,2,2) の 17 手問題である．一見 d5 の自青駒を f6 から脱出させるような簡潔な問題に見える．しかし実際は左下の脱出を止める，右上の脱出を目指し b6 の敵駒を引き付ける，引き付けた敵駒の横を縫うように b3 の自青駒で脱出を目指す，脱出を目指す青駒を赤駒で援護する，といった手順を取る問題である．長手数の問題でありながら，突き詰めると互いに無駄な行動が殆ど存在しない非常に高難易度な問題である．殆どの被験者は時間いっぱいまで使い切っており，コメントで『これは思いつかない』といった意見が散見された．これにより，攻略の糸口さえ見えないほどの高難易度の問題は万人に評価されにくいと考えた．こういった問題は，中級者以上に解かせるか，初級者でも 10 分程度の時間を与えるなどすればむしろ高評価となったかもしれない．

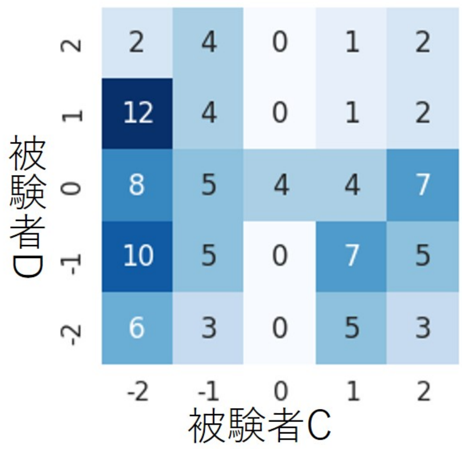
図 7.3(b) は駒数 (3,3,3,3) の 15 手問題である．c6 の自青駒を f6 から脱出させる問題であるが，e6 の自赤駒が邪魔になっており回り込むか移動させるしかない．この問題の場合は回り込むことになるのだが，自青駒の動きは非常に簡潔なもので，c6 → d6 → d5 →⋯→ f6 と動く．その道中で自赤駒が敵駒を取るような動きが挟まるが，先手側の動きは単調である．一方で，その自青駒を追いかける敵駒の動きは少々複雑で，初手 c5 から c6 か d5 への二択で c6 へ動くことになる．もし d5 から追いかけた場合は，d6 の青駒から脱出への道中で取られてしまう．このように一見似た結果になりそうな後手側の動きのどちらを選ぶか，先手側はどんな道順で回り込むかが重要な問題である．コメントでは『非現実的な盤面』『不自然な動き』といった意見が散見された．この問題は先手側の赤駒が右半分，青駒が左半分，後手側の駒は左半分に集中している，と非常に奇妙な形をしている．さらに，動かす必要がある駒は少なく，何かしらの役割があっても動かす必要がない駒が殆どであることが挙げられる．加えて，一見先手側に都合よく見える後手側の動きなどが，解答例を確認したプレイヤーを納得させられなかったと考えられる．

7.4　被験者同士の評価差

人間がゲームに感じる面白さなどの印象は，そのゲーム自体への理解度は勿論のこと，感性やこれまで蓄積されてきた様々な経験に左右されると予想できる．そこで，8 人の被験者同士の面白さ評価をヒストグラム化し，評価の傾向の差を調査した．ここでは相関係数が一番高いヒストグラムと一番低いヒストグラムを図 7.4 に示す．



(a) 相関係数最大 (0.38)



(b) 相関係数最小 (-0.12)

図 7.4: 被験者同士の面白さ評価

図 7.4(a) の相関係数は 0.38 と正の相関が見られ，2 人の評価傾向は僅かに似ていることがわかる．一方で図 7.4(b) の相関係数は-0.12 と極小ではあるが負の相関を持っている．ヒストグラムを細かく見ていくと，横軸の被験者 C が面白くないと評価した問題の中でも，縦軸の被験者 D は少し面白いと感じている問題が多いことがわかる．これらの被験者について分析したところ，被験者 C は全被験者でも最大の正答率 81%，被験者 D は最小の正答率 43%であり，約 1.8 倍もの開きがあった．考えた時間においても，被験者 D はほぼ全ての問題で時間を最大に使い切っていた．前節で述べたように，面白さと難しさにはある程度の相関がある．そのことから，被験者 D が難しく面白いと感じた問題が，被験者 C にとっては簡単でつまらない問題だった，というような差異があったと考えられる．これは面白さの感じ方は被験者の熟練度やパズルへの経験によって大きく左右されるという予想の裏付けとなる．

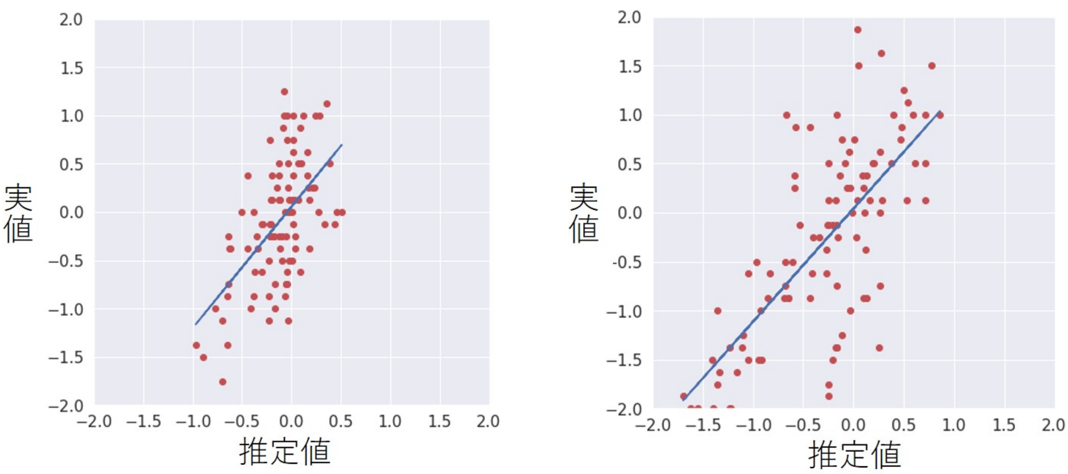
これにより，全てのプレイヤーにとって面白いと感じられる問題の生成は難しく，将来的に技術向上支援システムとして運用する際は，個人個人の熟練度や趣味嗜好に合わせた問題の生成が望ましいと考える．

7.5　面白さ・難しさ推定

続いて本研究では 7.2 節で示した各問題における特徴量を説明変数，被験者実験により得られた各問題の面白さと難しさを目的変数として，面白さと難しさそれぞれの推定モデルを生成した．学習には決定木アルゴリズムに基づいた勾配ブースティングの機械学習フレームワークである LightGBM[26] を用いた．LightGBM のバージョンは 2.2.3 であり，両モデルの学習に用いたパラメータは以下の通りである．

- metric(誤差関数の測定法)：rmse
- learning_rate(学習率)：0.05
- max_bin(分岐に入る最大データ数)：255
- num_iterations(木の数)：150

生成したモデルに対して，評価の過大評価・過小評価を抑制するために 10 分割交差検証を行った．推定を行った結果を，横軸を推定値，縦軸を実値としてプロットしたものを図 7.5 に示す．なお，青線は回帰直線である．



(a) 面白さ推定 (RMSE:0.52)　　(b) 難しさ推定 (RMSE:0.66)

図 7.5: 面白さ，難しさ推定

精度を示す二乗平均平方根誤差 (RMSE) は面白さ推定では 0.52，難しさ推定では 0.66 となった．多少の誤差は見られるものの，面白さと難しさ共に実値と推定値には正の相関がはっきりと出ており，ある程度推定できていることがわかる．

さらに，どの特徴量が推定に重要な役割を持っているのかを示す重要度を分析したが，7.3 節で述べたように面白さと難しさは正の相関がある．よって片方の推定に高い重要性を持つ特徴量はもう片方の推定にも対しても重要な特徴量，ということが多く見られた．

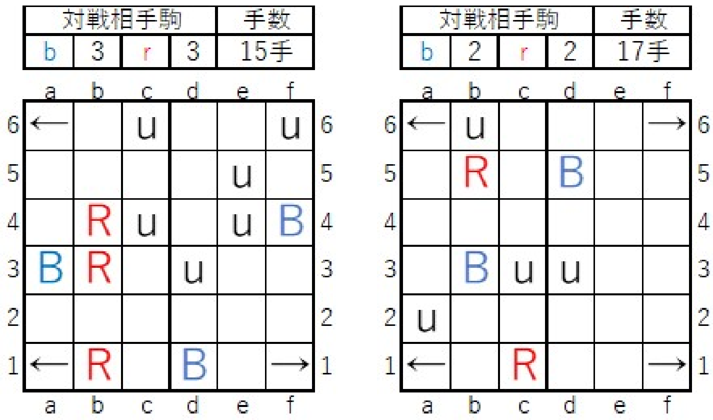
難しさ推定に高い重要度を持つ特徴量の一つに『探索中のルートノードが得た最大の証明数』があり，これは面白さにも比べても高い重要度を持っていた．証明数とはその局面で調べる必要がある局面の量を表しており，詰みが確定した時点で 0 となる数値だが，探索中のルートノードが得た最大の証明数を逐一更新していくことで特徴量としている．これが大きくなるほど調べる必要がある局面の量は増えることになるため，人間の場合はある程度の目星をつけて調べるであろうことを差し引いても，難しいと感じるのは自然なことである．

他に難しさ推定に高い重要度を持ちながら面白さではそれほど高くなかった特徴量として『脱出口へ向かう以外の行動手数』があった．これは一般問題において最短手数から脱出口に移動する手数を除いた手数であり，これが大きくなるほど脱出を目指す以外の手，例として敵駒の脱出の妨害，脱出を目指す自駒の援護，などの手が多いこととなる．この値が難しさ推定に高い重要度を持つのは自然なことであるが,『最短手数』よりも重要な特徴量であることから，プレイヤーは問題で示される最短手数というまず目に入る第一印象にあまり引きずられないということがわかる．面白さ推定に対してはそれほど高い重要度を持っていなかった理由としては，明らかな勝ちが見えているにも関わらず後手側の脱出妨害などで手数を稼がれても面白さを感じないプレイヤーが多かったのではないかと考えられる．

難しさに比べて面白さの推定に高い重要度を持つ特徴量としては『詰みまでに減る駒数の最小値』が挙げられる．これは盤面が詰みとなるまでに減る最小限の駒数を表す値であり，これが大きいほど面白いと感じるプレイヤーが多いように見受けられた．つまり，プレイヤーは駒の取り合いに大きな魅力を感じるのではないかと予想できる．

7.6 特徴的な問題

ここでは被験者に最も面白かった，難しかったと評価された問題を紹介する．この2問については付録Bにおいてより詳しい解説を載せているため参照されたい．



(a) 最も面白い
(面白さ：1.3)

(b) 最も難しい
(難しさ：1.9)

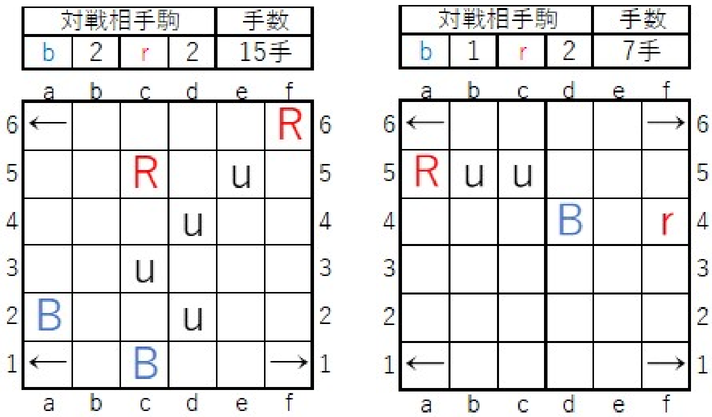
図 7.6: 最も面白い，難しい問題

図7.6(a) が最も面白かったとされる問題で，被験者全員の平均面白さ評価値は1.3であった．左上部での攻防が軸となる問題で，自赤駒をうまく立ち回らせ自青駒の脱出を支援することになる．取れる駒をすぐには取らず，動かれることが致命的になる駒を妨害することを第一に考えなくてはならない．コメントによると，その動きの意外性に魅力を感じたプレイヤーが見られた．こういった問題は，取れる駒数に余裕がある際は取りたくなる初心者の考えを覆す，非常に解かせる価値のある問題だと考えられる．

図7.6(b) が最も難しかったとされる問題で，これは図7.3(a) の同様の問題であり，詳しい問題の考察は7.3節および付録Bを参照されたい．被験者全員の平均難しさ評価値は1.9であり，正答率は0%であった．攻略の糸口さえ見えないほどの高難易度の問題は初心者に受け入れにくいことが考察できる．

その他の特徴的な問題を 2 問紹介する．図 7.7(a) は長手数だが簡単で面白くないとされた問題である．最短手数は 15 手もありながら，蓋を開けてみると敵駒 1 つの妨害と自青駒の脱出口への直行を合わせただけの問題である．その妨害も，周辺の駒数が少ないことから非常に見えやすいものであり，被験者はあまりの簡潔さにやりごたえを感じなかったのであろうと考えられる．

逆に図 7.7(b) は短手数だが面白いとされた問題であり，図 4.2(a) と同様の問題である．駒数が少ないことから考える局面も少ないように思えるが，敵駒の様々な逃がし方と，それを追い詰めるための自駒の動きを考える必要がある適度なやりごたえを持つ問題である．青駒全取り要素を含む問題の入門として最適な問題と考える．



(a) 長手数で面白くない
(面白さ：-1.2)

(b) 短手数で面白い
(面白さ：1.0)

図 7.7: 特徴的な問題

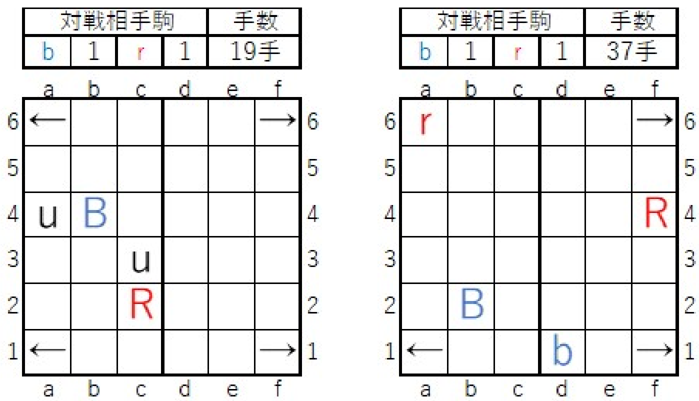
第8章　その他のアプローチ

5 章，6 章では詰めガイスター問題の生成法としてランダム生成と逆順生成のアプローチを取った．本章では本研究と強く関係するその他のアプローチとして，川上が提案した『後退解析』のアイデアおよび実験 [27] と，池田が提案した『部分問題の評価利用』のアイデア [28] を紹介する．

8.1　後退解析

後退解析とは，勝ち負け引き分けなどの評価が確定した局面から手を戻していき，戻した局面の評価を決定していく手法である [29]．この手法は膨大な盤面の列挙が必要となるため，川上は駒数を最小である (1,1,1,1) に抑えて全列挙を行い検証を行った [27]．(1,1,1,1) の場合, 敵駒色を 1 つ特定すればもう一方も定まるため，一部公開問題は存在せず一般問題・完全公開問題の 2 種類が考えられる．

結果，一般問題において必勝盤面が 191,992 つ，その他の盤面が 514,868 つとなり，最長勝ち盤面は図 8.1(a) で 19 手であった．一方，完全公開問題では必勝盤面が 783,232 つ，その他の盤面が 630,488 つとなり，最長勝ち盤面は図 8.1(b) で 37 手であった．完全公開問題の最長手数 37 手は，本研究で行ったランダム＋逆向き生成法の実験の中では最長手の 21 手よりも長い．よって，少ない駒数であれば後退解析の方が，長手数の問題をより効率的に生成できると考えられる．



(a)　(b)

図 8.1: 後退解析による生成問題

8.2　部分問題の評価利用

次に面白い問題の生成において，池田は部分問題（左上 3 × 3 マスなど）の評価を利用できないかと考えた [28]．例として，部分問題だけ見れば受けのない形をしているが，全体としては手数の稼ぎ合いで非自明な勝ち筋のある問題を構築できると面白いのではないかと考えた．問題例を図 8.2 に示す．

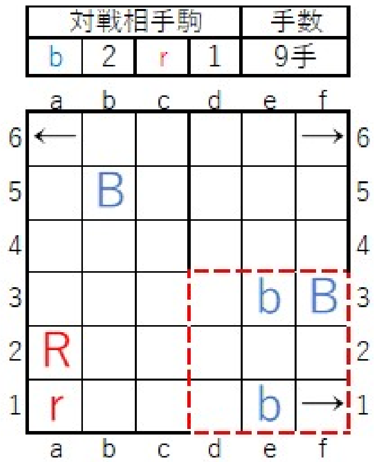


図 8.2: 部分問題の評価利用の例

図 8.2 の右下 3 × 3 マス（赤破線部）に注目すると，部分的にはいずれ脱出を許してしまうという受けがない形となっている．だからといって，右下を放置し b5 の青駒を脱出に向けて動かすと，自青駒の脱出 3 手に対し敵青駒の脱出 2 手が 1 手早いため負けが確定する．そこで f3 の自青駒を f2 に動かすと，敵青駒が脱出するためには 2 手増えて 4 手必要となり，左上の自青駒が先に脱出できる，といった仕掛けとなっている．このような問題を生成するために，部分問題の評価を利用できるのではないかと考えた．

第9章　おわりに

本研究では，不完全情報ゲーム『ガイスター』を対象とした部分問題として詰めガイスター問題を提案し，問題生成を行うことで，ゲームにおける部分問題の切り出しと，面白く適切な難易度の詰め問題の高速生成を目指した．結果，詰めガイスター問題を定義し，それぞれに特徴を持つ一般問題と一部公開問題を提案した．

そして問題生成アルゴリズムとして，ランダムに盤面を生成する『ランダム生成法』と，元となる盤面から手を戻すことで新たな問題を生成する『逆順生成法』の 2 種の生成法を用い，効率的な問題生成を目指した．さらに，プレイヤーがどんな問題に面白さや難しさを感じるのかを被験者実験と推定モデル生成を通して分析し，問題の持つ特徴量から面白さと難しさをある程度の精度で推定することに成功した．これにより，特定の難しさを持つ問題や面白い問題を任意に生成することができると考えられる．これらはガイスターのみならず，様々なゲームにも共通する点があることから，業界全体への貢献が期待できる．

一方で，個人個人によって感じる面白さや難しさなどの感性に違いが見られたため，今後技術力向上支援システムとして取り入れていくためには特定個人に合わせた高精度なモデルの生成が必要になってくるであろう．さらに，本研究では部分問題として詰め問題を対象として選んだが，他にもガイスターにおいて『色予測問題』『次の一手問題』などの様々なコンテンツが考えられる．ガイスターにおいて幅広い技術を身に着けるためにはそのような問題にも高い需要があると思われる．

参考文献

[1] David Silver et al. Mastering the game of go without human knowledge. *Nature*, Vol. 550, pp. 354–, 2017.

[2] 石飛太一. 詰め将棋問題の自動生成アルゴリズムに関する研究. 北陸先端科学技術大学院大学課題研究報告書, 2013.

[3] 広瀬正幸他. 逆算法による詰め将棋の自動創作. 人工知能学会誌, Vol. 13, pp. 452–460, 1998.

[4] ガイスター. [http://www.mobius-games.co.jp/Gester.htm].（アクセス：2020/01/10）.

[5] 石井岳史 et.al. 不完全情報ゲーム『ガイスター』における 2 種の詰め問題の提案と考察. 研究報告ゲーム情報学（GI）, Vol. 2019-GI-41, pp. 1–8, 2019.

[6] 石井岳史 et.al. 難しい詰めガイスター問題の生成法. ゲームプログラミングワークショップ 2019 論文集, pp. 12–19, 2019.

[7] ゲーム理論入門／（６）じゃんけんのナッシュ均衡. [https://himaginary.hatenablog.com/entry/20080825/game_6].（アクセス：2020/01/31）.

[8] 三塩武徳, 小谷善行. ゲームの不完全情報推定アルゴリズム upp とそのガイスターへの応用. 研究報告ゲーム情報学（GI）, Vol. 2014-GI-31, pp. 1–6, 2014.

[9] 佐藤佑史. ガイスターにおける自己対戦による行動価値関数の学習. 電気通信大学学術機関リポジトリ, 2015.

[10] 末續鴻輝, 織田祐輔. 機械学習を用いないガイスターの行動アルゴリズム開発. GAT2018 論文集, Vol. 2018, pp. 13–16, 2018.

[11] 川上直人, 橋本剛. 完全情報ゲームの探索を用いたガイスター ai の研究. ゲームプログラミングワークショップ 2018 論文集, pp. 35–42, 2018.

[12] Sehar Shahzad Farooq et al. Inference of opponent's uncertain states in ghosts game using machine learning. *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, Vol. 2, pp. 335–346, 2015.

[13] パソコン初心者の館. [http://www.pro.or.jp/~fuji/]. （アクセス：2020/01/30）.

[14] 高品質なナンプレ問題を自動生成する人工知能システム . [https://pc.watch.impress.co.jp/docs/2006/0906/yajiuma.htm]. （アクセス：2020/01/30）.

[15] 進化計算DARWIN. [https://www.timedia.co.jp/service/darwin/].（アクセス：2020/01/30）.

[16] 大町洋, 佐藤直之, 池田心. 複数ソルバを用いた上海ゲームのインスタンス生成. ゲームプログラミングワークショップ 2013 論文集, pp. 126–129, 2013.

[17] 牧田光平, 池田心. 連鎖構成力向上のための多様で面白いなぞぷよ提供法の提案. 研究報告ゲーム情報学（GI）, Vol. 2019-GI-41, pp. 1–8, 2019.

[18] Taishi Oikawa et.al. Improving human players t-spin skills in tetris with procedural problem generation. The 16th International Conference on Advances in Computer Games 発表論文, 2019.

[19] 及川大志, 池田心. テトリスにおける t-spin 構成力向上のための問題作成. ゲームプログラミングワークショップ 2018 論文集, pp. 175–182, 2018.

[20] Barbara De Kegel et.al. Procedural puzzle generation: A survey. *IEEE Transactions on Games*, 2019.

[21] 詰将棋. [https://ja.wikipedia.org/wiki/詰将棋]. （アクセス：2020/01/14）.

[22] 詰碁. [https://ja.wikipedia.org/wiki/詰碁].（アクセス：2020/01/14）.

[23] 長井歩. 詰将棋. *IPSJ Magazine*, Vol. 44, pp. 905–910, 2003.

[24] 長井歩, 今井浩. df-pn アルゴリズムの詰将棋を解くプログラムの応用. 情報処理学会論文誌, Vol. 43, pp. 1769–1777, 2002.

[25] Thomas W. Malone. Toward a theqry of intrinsically motivating instruct. *Cognitive Science*, Vol. 4, pp. 333–369, 1981.

[26] LightGBM. [https://lightgbm.readthedocs.io/en/latest/index.html].（アクセス：2020/01/28）.

[27] 川上直人. 後退解析による詰めガイスター問題の列挙. 第 43 回ゲーム情報学（GI）研究発表会（発表予定）, 2020.

[28] 池田心. *personal communication.*

[29] K.Thompson. Retrograde analysis for certain endgames. *ICCA Journal*, Vol. 9, pp. 131–139, 1986.

付録

A　被験者実験に用いた問題

表 A.1～表 A.4 に被験者実験に用いた詰めガイスター問題 100 問の『駒数』『手数』『問題種類』『構成要素』を示す.

表 A.1: 青駒脱出の要素を持つ一般問題

手数	先手駒数		後手駒数		問題種類	構成要素	問題数
	青駒	赤駒	青駒	赤駒			
5	1	1	1	1	一般	青駒脱出	1
7	2	2	2	2	一般	青駒脱出	1
7	4	4	4	4	一般	青駒脱出	1
9	2	2	2	2	一般	青駒脱出	2
9	3	3	3	3	一般	青駒脱出	2
9	4	3	1	4	一般	青駒脱出	1
11	2	2	2	2	一般	青駒脱出	4
11	3	3	3	3	一般	青駒脱出	3
13	2	2	2	2	一般	青駒脱出	3
13	3	3	3	3	一般	青駒脱出	3
13	4	4	4	4	一般	青駒脱出	1
15	2	2	2	2	一般	青駒脱出	6
15	3	3	3	3	一般	青駒脱出	4
17	2	2	2	2	一般	青駒脱出	2
19	3	3	3	3	一般	青駒脱出	1

表 A.2: 青駒脱出と赤駒壁利用の要素を持つ一般問題

手数	先手駒数		後手駒数		問題種類	構成要素	問題数
	青駒	赤駒	青駒	赤駒			
5	1	1	1	2	一般	青駒脱出+赤駒壁利用	1
9	2	1	2	1	一般	青駒脱出+赤駒壁利用	1
9	2	1	1	2	一般	青駒脱出+赤駒壁利用	1
9	2	2	2	2	一般	青駒脱出+赤駒壁利用	2
9	3	2	3	2	一般	青駒脱出+赤駒壁利用	3
11	2	1	2	1	一般	青駒脱出+赤駒壁利用	2
11	2	2	2	2	一般	青駒脱出+赤駒壁利用	2
11	3	2	3	2	一般	青駒脱出+赤駒壁利用	1
13	2	1	2	1	一般	青駒脱出+赤駒壁利用	3
13	2	2	2	2	一般	青駒脱出+赤駒壁利用	3
13	3	2	3	2	一般	青駒脱出+赤駒壁利用	1
15	2	1	2	1	一般	青駒脱出+赤駒壁利用	2
15	2	2	2	2	一般	青駒脱出+赤駒壁利用	1
17	2	2	2	2	一般	青駒脱出+赤駒壁利用	1
19	2	2	2	2	一般	青駒脱出+赤駒壁利用	1

表 A.3: 青駒脱出の要素を持つ一部公開問題

手数	先手駒数		後手駒数		問題種類	構成要素	問題数
	青駒	赤駒	青駒	赤駒			
7	2	2	2	2	一部公開	青駒脱出	3
7	3	2	2	3	一部公開	青駒脱出	1
9	2	2	2	2	一部公開	青駒脱出	4
11	2	2	2	2	一部公開	青駒脱出	4
13	2	2	2	2	一部公開	青駒脱出	4
15	2	2	2	2	一部公開	青駒脱出	4

表 A.4: 青駒全取りの要素を持つ一部公開問題

手数	先手駒数		後手駒数		問題種類	構成要素	問題数
	青駒	赤駒	青駒	赤駒			
7	1	1	1	2	一部公開	青駒全取り	1
7	1	3	1	2	一部公開	青駒全取り	1
7	3	3	1	2	一部公開	青駒全取り	2
9	2	2	2	2	一部公開	青駒全取り	2
9	3	3	1	2	一部公開	青駒全取り	1
11	1	2	1	2	一部公開	青駒全取り	2
13	1	2	1	2	一部公開	青駒全取り	7
13	2	2	2	2	一部公開	青駒全取り	2
13	3	3	1	2	一部公開	青駒全取り	2

B　最も面白い，難しいと評価された問題の解説

図 7.6 の 2 問について解説を行う．まず最も面白い問題と評価された図 7.6(a) を再び図 B.1 に示す．この問題は a3 の自青駒を a6 から脱出させることを目標とする問題であるが，まずは脱出のための障害が何かを考えなくてはならない．位置からして邪魔をしてくると考えられるのは c4 と c6 の敵駒であろう．特に障害となるのは c6 で，単純にまっすぐ脱出を目指した場合，先に脱出口前で待ち伏せされることになる．そのため，脱出を目指す前に取る必要がある．そこでまず 1 手目で b4 の自赤駒を b5 に移動させ，c6 の敵駒を待ち伏せする．敵駒が動く前にこの手を行わなかった場合，c6 の敵駒に b6 に動かれ逆に自赤駒を待ち伏せされることになる．次の後手番（2 手目）では，今動いた先手側の赤駒を取ることで c6 の駒を守ることを優先する，と考えるべきである．ここでは c4 の駒を c5 に動かすこととなる．これで赤駒を取るために隣接することになり，もし取られたとしても返しで取り返すか c6 を安全圏に逃がすことができる．次の先手番（3 手目）では，b3 の赤駒を b4 に動かすことになる．なぜならこうすることで，隣接された b5 の赤駒がもし取られても更に取り返しつつ c6 の敵駒に対して待ち伏せを継続することができるからである．ここまで進めると後は実質的な消化試合となり，後手側が青駒を取ろうと赤駒を取ったとしても，先手側は敵駒を取り返しつつもう片方の敵駒に対して待ち伏せを行うことができる．

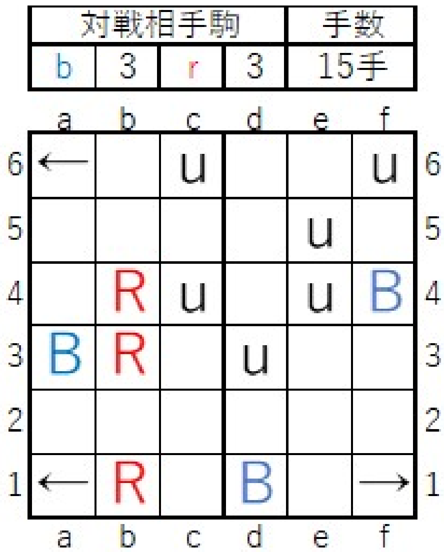


図 B.1: 最も面白いと評価された問題

次に最も難しい問題と評価された図 7.6(a) を再び図 B.2 に示す．最初の先手番（1 手目）は c1 から b1 に赤駒を動かすことは確定である．なぜなら，そうしなかった場合 a2 敵駒にすぐ脱出されてしまうからである．次の後手番（2 手目）も，d5 青駒の脱出を防ぐために b6 から c6 に動かすことは確定である．次の先手番（3 手目）だが，ここでどちらの青駒の脱出を目指すか考える必要がある．右上の d5 青駒の脱出を目指す場合，2 手目で動かれた c6 敵駒を排除する必要がある．b5 の赤駒を使用することでそれも可能だが，本問題は敵赤駒が 2 駒であり，最悪の状況を考えた場合取ることが出来る駒は 1 つとなる．その 1 つは a2 敵駒に使用すべきであるため，c6 敵駒は取るべきではない．b5 赤駒を敢えて取らせることで b1 赤駒を最後の赤駒とし脱出口に蓋をすることで敵駒を取らずして脱出の妨害を行うことも可能ではあるが，b5 赤駒を必ず取らせる状況を作ることは難しい．一方で，b3 青駒で脱出を目指した場合は，妨害しうる敵駒は a2 と c6 の敵駒である．a2 敵駒は脱出を狙っている駒であり，もしこの駒で妨害する場合，後手側の脱出までの手数を稼ぐことができる．c6 敵駒は基本的に右上の d5 青駒の待ち伏せに徹する必要があるため，d5 青駒の斜め 1 マスの位置を維持する必要がある．それを利用していけばよい．よってここでは青駒を b3 から b4 に動かす．次の後手番（4 手目）では c6 駒を動かすことができないため，a2 駒を a3 に動かすことで b4 青駒の待ち伏せを行う必要がある．次の先手番（5 手目）では b5 赤駒を a5 に動かすことで，a3 敵駒を待ち伏せしつつ b4 青駒の通り道を開ける．こうなると，後手番は a3 駒と c6 駒は共に身動きが取れなくなる．先手番（7 手目）で b4 青駒を b5 に動かすことで c6 駒の斜め 1 マスに二つの青駒を配置したことになる．あとは片方で c6 駒を引き付けることで，もう片方の脱出口までの道を開ければよい．

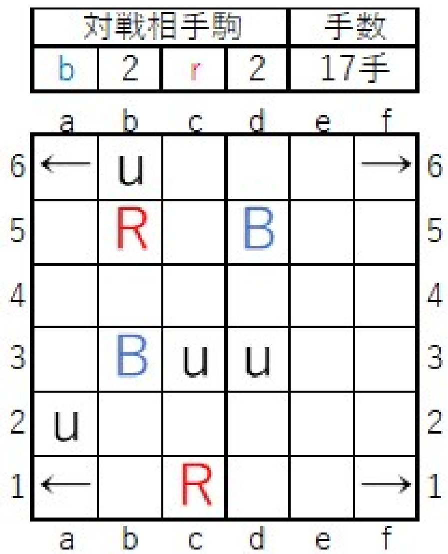


図 B.2: 最も難しいと評価された問題