

複数ソルバを用いた上海ゲームのインスタンス生成

大町洋 佐藤直之 池田心

北陸先端科学技術大学院大学

{h.omachi, satonao, kokolo} @jaist.ac.jp

数独を代表とする一人パズルに対する求解・インスタンス生成の研究はさかんに行われているが、そのうち“上海”やマインスイーパーなど不完全情報性を持つものにおける研究は依然少ない。本研究では、上海を題材に、不完全情報性を踏まえた着手決定法を考案し、その上で人間にとって面白いインスタンスを生成することを目的とする。具体的にはまず、モンテカルロ法と評価関数ベースの木探索を組み合わせ、高性能・低性能の仮想プレイヤーを作成する。その上でランダムに生成したインスタンスを解かせ、“高性能の仮想プレイヤーには解け、低性能の仮想プレイヤーには解けない”，つまり簡単あるいは難しすぎず、また不完全情報性による理不尽さの少ない、実力が物をいうインスタンスを提供する。

Instance Generator of Mahjong Solitaire using Multiple Solvers

HIROSHI OMACHI, NAOYUKI SATO, KOKOLO IKEDA

Japan Advanced Institute of Science and Technology

A lot of research has been done in the field of single player puzzles, especially for solving or creating instances. However, research is still limited for puzzles of imperfect information games like Shanghai solitaire (Mahjong solitaire) or Minesweeper. Our aim is to create interesting instances for human players in puzzles of imperfect information games. First, we propose a decision-making method in Shanghai solitaire, by applying Monte-Carlo method and action-value functions. Then, we implement two virtual players, which have high-performance and low-performance, and we use virtual players to solve instances of Shanghai solitaire puzzle. This allows us to pick out interesting instances, that is, the instances which high-performance players can solve but low-performance players cannot. These instances are interesting for human players because they reflect the level of the player at solving the puzzle.

1. 背景と目的

パズルゲームは古くから楽しまれているゲームの一つであるが、従来は用紙や本に書き込むか、専用の道具を用いて遊ぶことが主流であった。しかし、近年はコンピュータ等の普及により手軽にプレイすることが可能となり、従来人間が作成していたパズルの一問一問（インスタンス）もコンピュータによる自動生成が研究され、実際に自動生成によるパズル本も出版されている[1]。

また、コンピュータの普及により一人でも簡単に（見たくないものを見ずに）不完全情報ゲームをプレイすることが可能となり、広く楽しめるようになった。上海ゲーム (Mahjong Solitaire) は代表的な不完全情報パズルゲームの一種であり、Microsoft 社の Windows Vista 以降に標準ゲームとして搭載され、より手軽に楽しめる様になった。しかしながら、ゲーム開始時の牌の初期配置（インスタンス）による難易度の差が激しく、運次第で理不尽な場合や、簡単、あるいは難しすぎる場合など、プレイヤーが不満を感じてしまう場合がある。

一方で、これまでも数独を代表とするペンシルパズルに対する解答・自動生成・自動分類等の研究は盛んに

行なわれてきた。しかしながら、上海ゲームはペンシルパズルとは異なり、不完全情報性や、解が一意でないという特徴があり、従来手法を適用することが出来ない。

本研究ではプレイヤーにより良いインスタンスを提供することを目標とし、まず上海ゲームにて、プレイヤーと同じ条件下で木探索とモンテカルロシミュレーションを用いた仮想プレイヤーを実現した。そしてパラメータの異なる仮想プレイヤーを複数用いることで、インスタンスの難易度や特徴を評価する手法を提案し、実験を行った。

2. 上海ゲーム

上海ゲームは麻雀牌を使った一人有限確定不完全情報ゲームであり、36種の麻雀牌を各4枚ずつ、合計144枚を特定の型に沿って積み上げ、以下のルールに従って全ての牌を取り除くことをゴールとするゲームである。

- プレイヤーは2枚の同じ牌種の牌の組を一度に取り除くことが出来る
- 何らかの牌が上に乗っている牌は取り除けない
- 左右両方に牌が接している牌は取り除けない

牌の状態の例を **Figure 1** に示す。図中の白い四角形は着手可能な牌、薄い灰色の四角形は着手不可能の牌、濃

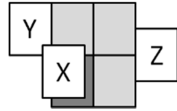


Figure 1 牌の状態の例

い灰色はプレイヤーにとって見えない牌（未知牌）でありかつ着手不能な牌を表している。例えば図中の X と Y がどちらも“一萬”なら、X と Y のペアを取り除ける。また、Y, Z の様な何にも乗っていない牌を 1 層目、X の様な 1 層目の牌に乗っている牌を 2 層目の牌とする。

このゲームの特徴として、不完全情報である点と、解までの着手数が確定という点がある。特に不完全情報であるため、プレイヤーが得られる情報を基に最もクリア不可能になり辛くなる様な着手（最善行動）を常に行ったとしても、クリア出来るかどうか確率的に決定される場合がある。本研究では、絶対に解けないインスタンスは除外し、さらに確率（運）に左右される度合いの小さいインスタンスを生成することを目標とする。

また、先行研究として上海ゲームは PSPACE 完全であることや、未知牌を知ることが出来れば NP 完全であることが知られており、未知牌を知ることが出来る状態で約 97% の解答率のソルバも作成されている[2].

3. 生成機

本研究では以下の 3 つのステップのアルゴリズムにより、プレイヤーへ提供するインスタンスを生成する。

- i. ランダムインスタンスを複数作成
- ii. 各インスタンスを性能の異なる評価機で評価
- iii. インスタンスを難易度や特徴を基に分類

ランダムインスタンス生成は、全ての牌が牌種未決定の状態から合法手のルールに従い着手を行い、全ての牌を取り終えたところで、着手履歴のそれぞれに牌種を割り当てる。この手法により、解答手順の存在が保障されたランダムインスタンスを生成することが可能である。

なお、インスタンスの難易度や特徴の分類は、異なる探索能力や重みパラメータを持った複数の評価機による平均解答率の違いを基に行う。例えば、二つの異なる評価機 α , β による、あるインスタンスに対する平均解答率の二次元マップの例を Figure 2 に示す。

図中右上の濃い灰色の領域は α , β が共通して高い解答率を出した場合で、簡単すぎるインスタンスであると判断する。逆に図中左下の薄い灰色の領域は α , β が共通して低い解答率を出した場合で、難しすぎる、もしくは運の影響が大きいインスタンスであると判断する。一方で図中の楕円形の領域は α , β で解答率に差がある場合で、 α , β の得意分野の違いから解答するのに特定の能力が必要となるインスタンスであると考えられ、インスタンスの特徴を表している。本研究ではこの様なインスタンスを“特定の能力があれば解けるインスタンス”とし、プレイヤーにとって解き応えのある物であると考えている。

4. 評価機

4-1. アルゴリズム

本研究ではあるインスタンスに対して評価機を用い

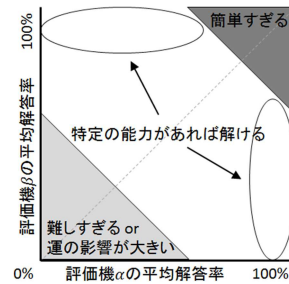


Figure 2 二つの評価機による平均解答率

た評価を行うことが重要となる。先行研究では未知牌の情報を得られるという条件でより厳密にインスタンスの解答を求めるソルバを作成していた[2]が、本研究ではよりプレイヤーに近い立場でインスタンスの評価を行うため、実際のゲームでプレイヤーが得られる情報と同じだけの情報を仮想プレイヤーに与えて着手決定を行わせ、仮想プレイヤーの着手決定に基づいて局面を推移させていく手法を用いて、インスタンスの平均解答率を求める。

実装した評価機のアルゴリズムを以下に示す。

- i. 合法手がある場合、仮想プレイヤーに着手を要求
- ii. 局面を推移
- iii. i-ii を合法手がなくなるまで繰り返す
- iv. クリアできたか調べる
- v. i-iv を一定回数繰り返し、平均解答率を出力

4-2. 仮想プレイヤー

本研究では 3 つの要素を用いた仮想プレイヤーにより、ある局面における着手決定を行う。その際に、入力された局面には未知牌が含まれている場合があるが、これは木探索の妨げになる。そこで、未知牌をランダムに仮定した上での木探索を n 回繰り返すモンテカルロ的アプローチ[3]を実装することで、未知牌の可能性を思慮した上での探索を行なうことを実現した。実装した仮想プレイヤーのアルゴリズムとそれぞれの要素を以下に示す。

- i. 確定着手がある場合は、確定着手を作成し v へ
- ii. 全ての未知牌をランダムで仮定する
- iii. 全ての合法手を、行動評価値を用いた深さ d の木探索を用いて評価
- iv. ii-iii を n 回繰り返し平均評価値最大の着手を選択
- v. 着手を出力

● 確定着手

確定着手とは、ある局面において、将来的に解答不能になる原因となりえない着手のことである。本研究では、以下の 4 つの確定着手を実装している。

- ① 残り枚数が 2 枚の牌種の牌は取る
- ② 同じ牌種が 4 枚着手可能な場合は取る
- ③ 同じ牌種が 3 枚着手可能で、かつその中に浮いている牌がある場合は、その牌以外の 2 枚の牌を取る
- ④ 同じ牌種が 3 枚着手可能で、かつその中に自身と同じ牌種の牌を直接支配している牌がある場合、その牌を含めた 2 枚を取る

ここで、ある牌 X が牌 Y に乗っている場合、もしくはある牌 X が牌 Z と共に牌 Y の左右にある場合、牌 X



Figure 3. “支配している牌”の例
XやZはYを支配している

にとって牌 Y は “支配している牌” とする. “支配している牌” の例を Figure 3 に示す. また “浮いている牌” とは, “支配してもされでもない牌” のこととする.

● 行動評価値

行動評価値とは, 仮想プレイヤーが確定着手を取ることが出来ない局面において着手決定を行う指標として用いるものである. ある局面 S から着手 a を取り局面 S' へと推移した場合の行動評価値を, 以下の 5 つの評価要素と, それぞれに対する重みを用いて求める. また, 最終的な行動評価値は (0, 1) で正規化している.

- I. 着手 a の 2 枚の牌が支配していた牌の総数
- II. 局面 S と S' の合法手数数の増減
- III. 2 層以上にある牌の数の増減
- IV. 浮いている牌の数の増減
- V. 未知牌の数の増減

なお各評価要素の重みは, レイアウト (初期配置の形) 毎にローカルサーチを行い, 最も平均解答率の高いものを利用している.

● 木探索

仮想プレイヤーは入力を局面として, 前述の確定着手と行動評価値を用いた木探索を行なうことで, 着手を決定して出力する. ゲーム木は入力された局面 S から一定の深さ D まで子ノードを生成するが, この時確定着手により推移したノードは深さとして見なさない.

また, ある深さ d の局面 S_d にて着手 a により局面が S_{d+1} に推移し, S_{d+1} が着手 {a_i} を持つとき, 局面 S_d における着手 a の行動評価値 E(S, a) を次の様に定義する.

$$E(S_d, a) = 1 : d < D \text{ かつ } \max_i E(S_{d+1}, a_i) = 1$$

$$E(S_d, a) = 0 : d < D \text{ かつ } \max_i E(S_{d+1}, a_i) = 0$$

$$E(S_d, a) = \frac{E(S_d, a) + \max_i E(S_{d+1}, a_i)}{2}$$

$$: d < D \text{ かつ } 1 > \max_i E(S_{d+1}, a_i) > 0$$

$$E(S_d, a) = E(S_d, a) : d = D$$

本手法では本来の行動評価値に子局面が持つ合法手の中で最も行動評価値が高いものの値による補正をかけている. 補正をかける際に親子ノード間での評価値の単純な平均化を行っているが, これはそれぞれの評価値の比率を調整する実験を行ったところ, 単純な平均を取る方法が最も高い性能を発揮したためである. また, 子局面の最大行動評価値が 0,1 であった場合は, 親局面の行動評価値も 0,1 にする.

4-3. 性能実験

本研究では, Figure 4 に示す 2 種類のミニサイズ (9 牌種 36 枚) レイアウト mTurtle と mSpider のそれぞれにおいて, 仮想プレイヤーのパラメータ n = 1, 4, 16,

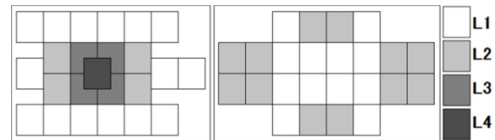


Figure 4. (左)mTurtle (右)mSpider (L=Layer)

Table1. mTurtle の平均解答率

n\d	1	2	3	4
1	0.8990	0.8802	0.8676	0.8399
4	0.9005	0.9097	0.9224	0.9160
16	0.9005	0.9222	0.9436	0.9494

Table2. mSpider の平均解答率

n\d	1	2	3	4
1	0.8769	0.8682	0.8764	0.8443
4	0.8804	0.8699	0.9024	0.9103
16	0.8806	0.8641	0.9114	0.9347

D = 1, 2, 3, 4 のそれぞれを組み合わせ, 1,000 個のインスタンスを 100 回ずつ評価する実験を行うことで, 実装した仮想プレイヤーの有効性を調査した.

実験結果を Table1, Table2 に示す. 結果を見ると, n = 1 では D が大きくなるにつれて性能が低下していることが分かる. これは, 1 度しか未知牌の仮定をしたうえで探索を行なわないため, いわゆる “思い込み” が発生し, その上で先読みを行うために悪い着手を行っているのではないかと考えている.

一方で, n が 1 より大きい場合は n, D それぞれが大きくなるにつれて性能が上昇し, 90% を超える高い解答率を得た. また, 牌の枚数を 2 倍の 72 枚にしたレイアウトでも実験を行ったところ, n = 16, D = 4 で 94.78% という高い解答率を得られており, 大きなサイズのインスタンスにおいても有効であるという結果が得られている.

5. 抽出

本稿では, 例えば二つの性能の異なる仮想プレイヤーによる解答率の二次元マップの分布から, 特徴的なインスタンスを抽出するという手法を提案している. 提案手法により期待通りのインスタンスを抽出することが可能かを調査するため, パラメータ n = 1, D = 1 と n = 16, D = 3 の二つの仮想プレイヤーの評価値の二次元マップを作成し, mTurtle の物を例として Figure 5 に示す.

Figure 5 の上部が示す様に, 二つの仮想プレイヤーを利用することで Figure 2 で示した “特定の能力があれば解けるインスタンス” が多量に存在することが確認できる. また Figure 5 の左側に注目すると, 低性能の仮想プレイヤーでは解答できなかったが, 高性能の仮想プレイヤーでは解答できるインスタンスの存在も確認できる.

この結果を基に, 二次元マップの分布から興味深いインスタンスをいくつかピックアップし, 著者の三名 (いずれも上海ゲーム経験者) が実際にプレイしたところ, 次の様な傾向が見られた. なお, 分類は平均解答率を大まかに high-middle-low の三つに分類している.

- high-high : 前者 : n = 16, D = 3 後者 : n = 1, D = 1 確定行動が多い. 簡単すぎる.

- middle-middle : 読みが必要となり解き応えがあるが, 運が絡んでしまう場合もある.

