

マイクロタスク型クラウドソーシングによるリバーシの試み

佐々木優^{*1} 平木理恵^{*1} 馬場雪乃^{*3} 森嶋厚行^{*2} 鹿島久嗣^{*3}
 Yu Sasaki Rie Hiraki Yukino Baba Atsuyuki Morishima Hisashi Kashima

^{*1}筑波大学大学院 図書館情報メディア研究科

Graduate School of Library, Information and Media Studies, University of Tsukuba.

^{*2}筑波大学 知的コミュニティ基盤研究センター/図書館情報メディア系

Research Center for Knowledge Communities, Faculty of Library, Information and Media Science, University of Tsukuba.

^{*3}京都大学大学院 情報学研究科 知能情報学専攻

Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University.

Crowdsourcing decision making is an interesting issue. This paper reports our attempt to crowdsource choosing moves in the reversi game. We chose this setting because the game is simple and populer, and we expect we can analyze the behavior of the crowd without involving complicated factors. This paper explains a variety of tasks for aggregating the inputs from the crowd, and some of the lessons learned.

1. はじめに

インターネットの出現により、多人数による意志決定に関して、単純な投票による多数決以外の様々な方法が可能になった。このような、インターネットを用いた多人数による意志決定プロセスの研究は近年注目を集めている [1]。本論文では、クラウドソーシングの一形態であるマイクロタスク型クラウドソーシングを用いた多人数による意志決定のあり方を議論するために、ルールが単純で広く知られているリバーシゲームを題材に、多様な方法による意志決定の性質を明らかにする事を目標とする。具体的な問題設定は次の通りである。リバーシ盤面と、次の手が黒か白かを与えられた時、次の一手の決定を、不特定多数の人々によって行う (図 1)。人々の入力には、マイクロタスクを利用する。3 節で説明するように、タスクは次に打つ手の単純投票とは限らない。

盤面 *Board* と次の手の色 *color* が与えられた時、次に打つ可能性がある手の候補の集合を $Moves(Board, color) = \{m_1, \dots, m_n\}$ とする。このとき、この問題は、 $m_x \in Moves(Board, color)$ なる m_x を一つ決定する関数 *Next* として定式化できる。

$$m_x = Next(Board, color)$$

リバーシを対象とした理由は、ルールが単純で、過去にやった事がない人でも理解しやすいため、今回の不特定多数の人々が取り組む作業の一例として適切と考えたためである。

2. 関連研究

1990 年代、当時のチェス世界王者とクラウドソーシングによる集合知「ワールド」の対戦が行われた。「ワールド」とは、世界中のチェス愛好家のべ 300 万人で構成されたチームで、彼らの次の一手は多数決によって決められた。この対戦は 2 回行われたが、「ワールド」の次の一手の決定方法は、それぞれ条件が異なり、例えば投票時間が、1 回目の対戦では 10 分、2 回目の対戦では 24 時間与えられた。また、2 回目の対決時には 24

連絡先: 佐々木 優, 筑波大学 図書館情報メディア研究科,
 〒 305-8550 茨城県つくば市春日 1-2, 029-859-1559,
 yuu.sasaki.2015b@mlab.info

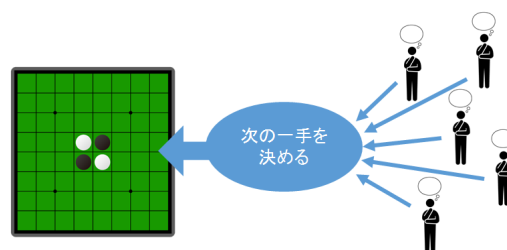
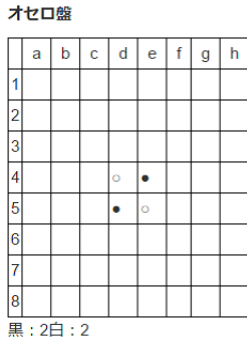


図 1: 不特定多数の人々によって次の一手を決定

時間の中に議論が行われるなどした。この違いは戦局の違いを与えたと言われている [2]。このように、「ワールド」の次の一手の決定方法には様々な方法が用いられ、方法次第で「ワールド」がより良い意思決定ができたかどうかが変わった。最近では、人と計算機がチームを組んで行う Advanced Chess の分野 [4] で、終盤に強くミスもしない計算機の強みと、形勢判断能力に優れた人間の強みを生かせば、人単独や、計算機単独よりも強いという結果が示されている。本研究は、リバーシというシンプルなゲームを対象に、多様なタスクデザインを試す事によって、どの要素が意志決定にどう影響を及ぼすのかを詳細に明らかにする事を目標とする。Michael Margel らの研究 [3] では、「Pocketmonster Red」を複数人で同時に操作できるように改造した「TwitchPlaysPokemon」を取り上げ、参加したプレイヤーの分析、タイプ分けを行っている。本研究は、この研究と同様に、次の操作を不特定多数の人々で決定するゲームを扱うが、プレイヤーのふるまいの分析だけではなく、ゲーム全体でプレイヤーの意思決定には何が影響しているかを考察する事で、より良い意思決定の方法を検討する。

3. マイクロタスク設計

本節では、リバーシの次の一手を決めるためのマイクロタスク設計について説明する。マイクロタスク設計に関しては様々な選択肢が考えられるが、今回は次の要素に着目して複数の設計を行う。



黒の手番です
もっとも良いと思う手を選んでください

図 2: 各自先読み-P-C のタスク画面

1. どう先読みするか: 次の手を打つための先読みをどう行うかに着目する. 各人が個々に先読みをする方法を, 各自先読み, 複数人数で分担する方法を分担先読みと呼ぶ.
2. なにをワーカが入力するか: 打つべき手を入力する方法をポジティブのみ (P), 打つべき手と打つべきではない手を入力する方法をポジティブネガティブ (PN) と呼ぶ.
3. なにをワーカに見せるか: Board のみ見せる (ϕ), 入力候補 (Candidates) である $Moves(Board, color)$ を見せる (C), 他人の意見を見せる (D)
4. どこまで分担先読みするか: 分担先読みの場合には, 先読みする幅 k と深さ l の数がパラメータ $WkDl$ となる.

次に, これらの組合せとしていくつかを説明する.

3.1 各自先読み-P

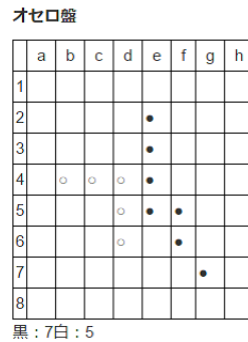
これは, 各自がそれぞれ先読みを行い, 次に置くべきマスを投票する方法である. 形式的には次のように定義される. $M = Moves(Board, color)$ としたとき, 各候補 $m_i \in M$ が得た投票数 $positive(m_i)$ の大小で決まる M 上の順序関係を \leq_p とする. さらに, 全ての M を, 順序関係 \leq で降順 (同点の場合にはランダム) に並べた列を返す関数を $Sort_{\leq} M$, 列 S の先頭から 1 番目の m_k を取り出す関数を $S[1]$ とする. この時, 次の手を決定する式は次のようになる.

$$Next(Board, color) \equiv (Sort_{\leq_p} M)[1]$$

各自先読み-P-C のタスク画面を図 2 に示す. 図 2 のように, ワーカは次に自分の色の石を置くべきだと思うマスを, あらかじめ提示された置けるマスの選択肢の中から回答する. 回答されたマスの中でもっとも投票数が多かったマスを次の一手に決定する. もっとも投票数が多かったマスが複数存在した場合は, それらのマスの中からランダムで決定する.

各自先読み-P- ϕ では, 置けるマスの選択肢を提示せず Board のみを表示し, ユーザが自由に座標を入力する. したがって, ユーザにとって気がつかない候補が存在する場合がある.

各自先読み-P-CD は, あらかじめ置ける場所の選択肢を提示し, そこから次に置くべきマスとそう考えた理由の回答を求める方法である. また, タスク画面に他のワーカのそのマスを選んだ理由を表示させる事で, 他のワーカの考えを知る事を



方法3 理由づけ投票
次に石を置くべきだと思うマスと, そう思った理由を回答してください.
(回答例) c行目, 3列目のマス→c3, 理由: 最も多く相手の石をとれるから

他の人の回答

よこ	たて	理由
c	3	白に角をとられるのを防ぐため

黒の手番です
● c3 ● c5 ● c7 ● c6 ● a4
理由:

図 3: 各自先読み-P-CD

可能にする. 各自先読み-P-CD のタスク画面を図 3 に示す. 図 3 のように, ワーカは先に回答した他のワーカの意見を見ながら, あらかじめ提示された置けるマスの選択肢の中から, 次に自分の色の石を置くべきだと思うマスと, そのマスを選んだ理由を回答する. 次に自分の色の石を置くマスは, 回答されたマスの中から投票数がもっとも多かったマスに決定する. もっとも投票数が多かったマスが複数存在した場合は, それらのマスの中からランダムで決定する.

3.2 各自先読み-PN

これは, 置くべきマスと置くべきでないマスの回答を求める方法である. m_i の「置くべき」の投票数を $positive(m_i)$, 「置くべきでない」の投票数を $negative(m_i)$ とする. 各候補 $m_i \in M$ の $positive(m_i) - negative(m_i)$ の大小で決まる M 上の順序関係を \leq_{pn} とする. この時, 次に置くマスを決定する式は次のようになる.

$$Next(Board, color) \equiv (Sort_{\leq_{pn}} M)[1]$$

各自先読み-PN-C のタスク画面を図 4 に示す. 図 4 のように, ワーカは次に自分の色の石を「置くべき」だと思うマスと「置くべきではない」と思うマスを選択し回答する. 複数のマスに同じ「置くべき」の票数が入っていた場合, 各自先読み-P ではそれらからランダムで決定したが, PN では「置くべきでない」の投票数を考慮する事により, より良いマスを選択させる事を狙う.

3.3 分担先読み

これは, 各自で先読みをして置くべきマスを選ぶのではなく, 分担して先読みを行う事によって, 次の手を決定する方法である. 分担先読みでは二種類のタスク A, B を用いる.

タスク A 石を置くべきマスを選択する. タスク画面は各自先読みのタスクと同じである. ただし, 各自先読みの場合

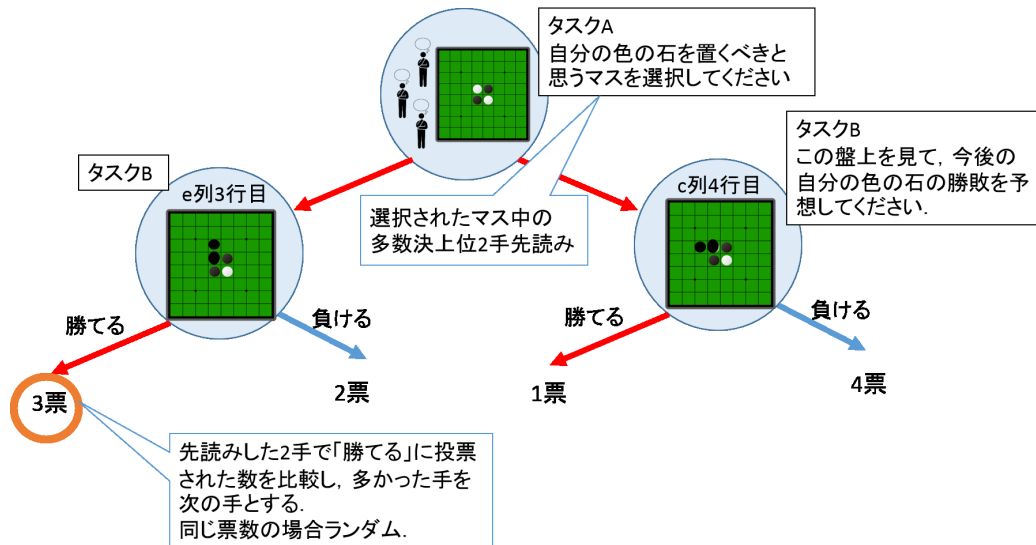


図 5: 分担先読み W2D1-P の流れ

オセロ盤

	a	b	c	d	e	f	g	h
1								
2								
3								
4				○	●			
5			○	○	○	●		
6						●		
7						●		
8								

黒 : 4白 : 4

黒の手番です
打つべき場所を選んでください
○ c4 ● b5 ○ c6 ● e6 ○ c3
打つべきでない場所を選んでください
○ c4 ● b5 ○ c6 ● e6 ○ c3

図 4: 各自先読み-PN-C タスク画面

オセロ盤

	a	b	c	d	e	f	g	h
1								
2								
3								
4			●	●	●			
5				●	○			
6								
7								
8								

黒 : 2白 : 2

この手番で黒は勝てますか? 負けますか?

図 6: 分担先読み タスク B

には、黒の担当の人は、黒の手を入力するだけであったが、分担先読みの場合は、自チームの立場だけではなく、相手の立場になった手も打つ事になる。

タスク B タスク A において投票されたマスのうち、投票数が多かった上位 N 個それぞれに、仮に石を置いた場合の盤上をタスク画面に表示する。その盤上をみて今後自分の色の石がどうなるか、「勝つ」「わからない」「負ける」のどれかを予想する。タスク画面を図 6 に示す。

各自先読みの場合には、黒の担当の人は、タスク A を利用して黒の手を入力するだけであったが、分担先読みの場合には、自チームの立場だけではなく、相手の立場になった手も打つ事になる。例えば、先読み数が 2 の時には、まず、システムは黒の手を入力するタスク A を生成する。黒の手が入力されると、その結果を見て多数決で上位 2 件のものに対して、今度は白の立場になっておこなうタスク A を生成する。その後、多数決で上位 2 件についてタスク B を生成し、その結果から最終

的な手を決定する。

各候補 $m_i \in M$ がタスク B において勝つと投票された数 $win(m_i)$ の大小で決まる M 上の順序関係を \leq_{win} と表記すると、分担先読み W_kD1-P は次のように定義される。

$$Next(Board, color) \equiv \max_{\leq_{win}} M'$$

$$\text{where } M' = (Sort_{\leq_p} M)[1, 2, \dots, k]$$

図 5 に、分担先読み $W2D1-P$ (幅 2, 深さ 1) の流れを示す。このように、先読み数が 1 の場合は、黒の手を入力して、その結果を見て多数決で上位 k 件 (今回は 2) のものに対して、タスク B を行う。

4. 予備実験

4.1 設定

不特定多数を対象とした実験を行う前段階として、実験協力者を募って予備実験を行った。タスクは Crowd4U [7] 上で

表 1: 最終的な石数と勝敗

方法	Crowd	COM	勝者
各自先読み-P-C	13	51	COM
各自先読み-P- ϕ	27	37	COM
各自先読み-P-CD	20	34	COM
各自先読み-PN-C	27	37	COM
分担先読み W2D1-P-C	7	57	COM

実装した。実験では、日本オセロ連盟の教材用オセロ [6](以下 COM) と、複数人のチーム (以下, Crowd) で対戦を行った。COM の難易度は、3 段階のうちもっとも易しいレベルに設定した。白 (後手) を COM が担当し、Crowd は黒 (先手) を担当した。

Crowd の構成人員は、リバーシの経験がある大学生・大学院生 10 人である。彼らは都合の良い時間にいつでも試合に参加できるため、必ずしも常に 10 人が参加したとは限らない。投票時間 2 分の間に先に投票した 5 人分の選択から次の一手を判断し、2 分以内に 5 人分の投票がなければ、すでに投票された中から、多数決で次の一手を決定した。

3 節で説明した全てのタスク設定で試合を行った。分担先読みは、各自先読み-P-C のタスクをタスク A として利用する分担先読み W2D1-P-C を用いた。

4.2 結果

まず、最終的な石数を表 1 に示す。今回は全て COM が勝者となった。

次に、Crowd が打った手の評価を行った。具体的には、リバーシ上級者 (元筑波大学オセロ同好会会員) 2 名が、実験で行った全ての試合の Crowd の手について、合議の上で評価を行った。評価基準は、5 (良い)、4 (やや良い)、3 (わからない)、2 (やや悪い)、1 (悪い) の 5 段階で行った。具体的には、リバーシでは一般的に自分の置く事ができるマスが多い方が有利とされている [5] ため、Crowd の打てるマスを増やし、COM の打てるマスを減らした手を高い評価 (評価 5) とし、Crowd の打てるマスを減らし、COM の打てるマスを増やした手を低い評価 (評価 1) とした。加えて、評価者には、評価 5 の中でも彼らならどこに置くかを選んでもらった。これをここでは「正解」と呼ぶ。

図 7 は、各手法による手の評価の結果である。各自先読み-P- ϕ より各自先読み-P-C の方が、評価 5 の割合と Crowd が打った手と正解が一致した割合が高かった。また、各自先読み-PN-C が、評価 5 の割合と Crowd が打った手と正解が一致した割合が、比較的高い結果になっている。興味深い結果としては、各自先読み-P-CD は、評価 5 も正解も少なく、分担先読み W2D1-P-C は、評価 5 は多いが、正解が少ないという結果になっている。

各手法の票の散らばりを比較するために、各手のジニ係数の平均を計算した (表 2)。各自先読み-P-CD において、特に散らばりが少ない事が読み取れる。これは、他人の意見を見る事ができるため、それに影響を受ける人が多かった事を示唆している。実際、各自先読み-P-CD のタスクで入力された理由には、「みんながそう言うなら」など他のワーカの意見に同調する内容がみられた。また、これは各自先読み-P-CD の正解が少なかった一つの要因とも考えられ、理由を示す事が必ずしもより良い意思決定になるとは限らない事を示しているが、この件に関しては、Crowd の中にリバーシが強い人が含まれるかどうかとも関係すると予想される。

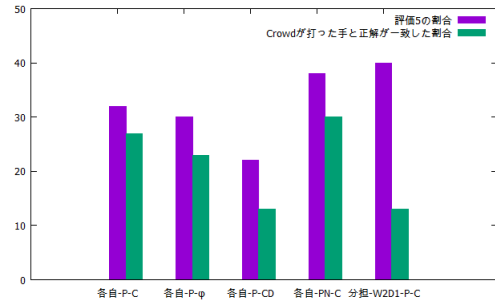


図 7: 各手の評価結果

表 2: ジニ係数の平均

方法	ジニ係数の平均
各自先読み P-C	0.356
各自先読み P- ϕ	0.342
各自先読み P-CD	0.359
各自先読み PN	0.282
分担先読み W2D1-P-C	0.281

5. まとめと今後の課題

本論文では、マイクロタスク型クラウドソーシングを用いた多人数による意志決定の知見を得る事を目標として、その題材に、リバーシを取り上げ、複数のタスクデザインを試みた。今後は、実際に不特定多数による実験、様々なパラメータを変更した実験を行った結果にもとづいて詳細な分析を行い、マイクロタスクを用いた不特定多数による意志決定のプロセスに関するデザインの原則を明らかにしたい。

謝辞. 本研究の一部は科研費基盤研究 (#25240012) の支援による。

参考文献

- [1] Katsuhide Fujita, Takayuki Ito, Minjie Zhang, Valentin Robu. Next Frontier in Agent-Based Complex Automated Negotiation. Studies in Computational Intelligence 596, Springer, Mach 30,2015, ISBN 978-4-431-55524-7
- [2] マイケル・ニールセン. オープンサイエンス革命. 紀伊國屋書店, 2013.
- [3] Margel, Michael. Association for Computing Machinery. Twitch Plays Pokemon: An Analysis of Social Dynamics in Crowdsourced Games.
- [4] Rasskin-Gutman, Diego. Chess metaphors: artificial intelligence and the human mind. MIT Press, 2009.
- [5] Seal Software. リバーシのアルゴリズム C++&Java 対応. 第 1 版 4 刷, 株式会社工学社, 2006, 207p.
- [6] “オセロ・無料ゲーム - Othello! JAPAN”. Othello! JAPAN. <http://www.othello.org/play/>, (2015-01-13).
- [7] “Crowd4U - クラウドフォーユー: 非営利・公益・学術目的のクラウドソーシング・マイクロボランティアプラットフォーム - Crowd4U”. Crowd4U. <http://crowd4u.org>, (2015-01-13).